UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Tese

Uma Contribuição para Avaliação de Custos do Uso de Nuvens Computacionais no Suporte a Aplicações Científicas

Maicon Ança dos Santos

Pelotas, 2023

Maicon Ança dos Santos

Uma Contribuição para Avaliação de Custos do Uso de Nuvens Computacionais no Suporte a Aplicações Científicas

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Gerson Geraldo H. Cavalheiro

Pelotas, 2023

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

S237c Santos, Maicon Ança dos

Uma contribuição para avaliação de custos do uso de nuvens computacionais no suporte a aplicações científicas / Maicon Ança dos Santos ; Gerson Geraldo Homrich Cavalheiro, orientador. — Pelotas, 2023.

146 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2023.

1. Infraestrutura de nuvem. 2. Workflow. 3. Bag-o--tasks. 4. Simulador. I. Cavalheiro, Gerson Geraldo Homrich, orient. II. Título.

CDD : 005

Elaborada por Simone Godinho Maisonave CRB: 10/1733

Maicon Ança dos Santos

Uma Contribuição para Avaliação de Custos do Uso de Nuvens Computacionais no Suporte a Aplicações Científicas

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 28 de fevereiro de 2023

Banca Examinadora:

Prof. Dr. Gerson Geraldo H. Cavalheiro (Orientador)
Doutor em Ciência da Computação pelo Institut National Polytechique de Grenoble.

Prof. Dr. Adenauer Correa Yamin
Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Claudio R. Geyer
Doutor em Informática pela Université Joseph Fourier.

Prof. Dr. Rafael Burlamaqui Amaral
Doutor em Computação pela Universidade Federal Fluminense.

Dedico este trabalho aos meus amados Rosane e Mathias.

AGRADECIMENTOS

Com mais esta etapa de aprendizado concluída, gostaria de deixar registrado meus sinceros agradecimentos a todos que, de uma forma ou outra, estiveram presentes nessa empreitada.

Primeiro, agradeço a Deus, fonte de toda a sabedoria e conhecimento. Sem Ele nada disto seria possível em minha vida.

Quero agradecer ao meu orientador, Prof. Dr. Gerson Cavalheiro, por ter apostado firmemente neste trabalho e ter incentivado e cobrado mesmo nos momentos mais difíceis.

À minha amada esposa, Rosane, por todo amor, carinho e paciência durante este tempo de estudos no Doutorado. Momentos de esforço e ausência que foram compensados pelo sucesso de mais esta conquista ao teu lado. Te amo.

Para tudo há um tempo,
para cada coisa há um momento debaixo dos céus:
tempo para nascer, e tempo para morrer;
tempo para plantar, e tempo para arrancar o que foi plantado.
— Ecl 3, 1-2

RESUMO

SANTOS, Maicon Ança dos. **Uma Contribuição para Avaliação de Custos do Uso de Nuvens Computacionais no Suporte a Aplicações Científicas**. Orientador: Gerson Geraldo H. Cavalheiro. 2023. 146 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2023.

Esta Tese discute a adoção de infraestruturas de nuvem para suporte à execução de aplicações científicas no meio acadêmico, no qual é comum o compartilhamento de recursos computacionais por diferentes grupos de pesquisa. Para o bom gerenciamento dos recursos compartilhados, o ambiente de nuvem computacional se apresenta como promissor no que se refere a universalização de acesso e o melhor aproveitamento de recursos financeiros. O objetivo geral deste trabalho é racionalizar, em relação aos custos financeiros, a adoção de nuvens computacionais para apoio à pesquisa acadêmica. Para isso, é necessário determinar os custos envolvidos para implantação de uma solução, atendendo a uma comunidade federada de instituições de ensino e pesquisa. Nesse contexto, são abordadas alternativas com ambientes de nuvens públicas, privadas e híbridas, identificando vantagens e desvantagens da adoção de cada uma e os respectivos impactos financeiros. Diferentes cenários de uso para infraestruturas de nuvem foram concebidos a fim de combinar opções de nuvens federadas e nuvens públicas com estratégias de escalonamento, dentre elas a de *cloud bursting*. Como contribuição científica, este trabalho traz a constatação de que nuvens públicas podem ser utilizadas como apoio a uma nuvem federada e o desenvolvimento de um simulador para ambientes de rede, o WCSim (*Workflow Cloud Simulator*), no qual foi possível utilizar aplicações modeladas como fluxos de trabalho, a partir de tarefas que se relacionam como *bag-of-tasks*. Para avaliar o comportamento das execuções de *workflows* sobre uma nuvem federada, juntamente com a técnica de *cloud bursting*, estudos de caso foram aplicados para uso no WCSim. Os resultados dos experimentos permitem verificar que a modalidade de precificação *spot* deve ser considerada nas avaliações de possíveis contratações de provedores de serviços de nuvem. Também observa-se que o balanceamento de carga intra e entre sítios pode ser bem explorado, potencializando a obtenção de ganhos de desempenho.

Palavras-chave: Infraestrutura de Nuvem. *Workflow*. *Bag-of-Tasks*. Simulador.

ABSTRACT

SANTOS, Maicon Ança dos. **A Contribution to Cost Evaluation of Using Cloud Computing in Support of Scientific Applications**. Advisor: Gerson Geraldo H. Cavalheiro. 2023. 146 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2023.

This Thesis discusses the adoption of cloud infrastructures to support the execution of scientific applications in academia, where it is common to share computational resources among different research groups. For the proper management of shared resources, the computational cloud environment appears promising in terms of universal access and better use of financial resources. The general objective of this work is to rationalize, in relation to financial costs, the adoption of cloud computing for academic research support. To do this, it is necessary to determine the costs involved in implementing a solution, serving a federated community of educational and research institutions. In this context, alternatives with public, private, and hybrid cloud environments are addressed, identifying the advantages and disadvantages of each, as well as their respective financial impacts. Different usage scenarios for cloud infrastructures were conceived in order to combine options of federated and public clouds with scaling strategies, including cloud bursting. As a scientific contribution, this work brings the finding that public clouds can be used to support a federated cloud and the development of a simulator for network environments, the WCSim (Workflow Cloud Simulator), in which applications modeled as workflows could be used from tasks that relate to a bag-of-tasks. To evaluate the behavior of workflow executions on a federated cloud, along with the cloud bursting technique, case studies were applied for use in WCSim. The results of the experiments allow us to verify that the spot pricing modality should be considered in the evaluations of possible cloud service provider contracts. It is also observed that intra and inter-site load balancing can be well explored, enhancing performance gains.

Keywords: Cloud Infraestructure. Workflow. Bag-of-Tasks. Simulator.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BoT	*Bag-of-Tasks*
CAPEX	*Capital Expenditure*
CPU	*Central Processing Unit*
DAG	*Directed Acyclic Graph*
DoB	*DAG of BoTs*
EC2	*Amazon Elastic Compute Cloud*
Gbps	*Gigabits per Second*
HPC	*High Performance Computing*
IaaS	*Infrastructure as a Service*
IEP	*Instituições de Ensino e Pesquisa*
LGPD	*Lei Geral de Proteção de Dados*
M	*Meta*
MPI	*Message Passing Interface*
NCBI	*National Center for Biotechnology Information*
OPEX	*Operational Expenditure*
OE	*Objetivo Específico*
QP	*Questão de Pesquisa*
ROI	*Return on Investment*
RSL	*Revisão Sistemática da Literatura*
TCO	*Total Cost of Ownership*
vCPU	*Virtual Central Processing Unit*
WCSim	*Workflow Cloud Simulator*

SUMÁRIO

1 INTRODUÇÃO

O modelo de computação em nuvem se apresenta como uma infraestrutura de suporte à execução sob demanda para uma vasta gama de aplicações, desde simples portais web a grandes fluxos de trabalho científicos. Neste ambiente computacional, recursos podem ser rapidamente provisionados e liberados com o mínimo esforço de gerenciamento ou interação com o provedor de serviços ((MELL; GRANCE, 2011), (THAKUR; GORAYA, 2022)).

Uma nuvem provê recursos computacionais de forma elástica. O modelo de implementação de nuvem pública oferece elasticidade por meio de um modelo de pagamento por utilização (*pay-as-you-go*), no qual a alocação de recursos aumenta ou diminui à medida que o usuário final disponibiliza mais ou menos recursos financeiros para instanciação de seu *data center*. Isto faz com que as demandas de processamento e disponibilidade, que antes faziam parte de estruturas dedicadas, sejam repassadas a provedores de serviços de nuvem externos responsáveis por garantir os acessos necessários, desonerando o usuário dos processos de gestão dos recursos.

Em implementações de nuvens privadas, por sua vez, a elasticidade é promovida pela ativação ou desativação de nós de processamento já pertencentes ao patrimônio das instituições, sem a necessidade de novos aportes financeiros durante as operações. Nestes ambientes também é possível programar o uso dos recursos em função da demanda. Com a diminuição dos custos de implementação das estruturas de nuvem, é cada vez mais comum encontrar instituições, acadêmicas ou comerciais, que operam seus *data centers* com suas próprias nuvens privadas. É comum a exploração de tais infraestruturas como suporte ao processamento das cargas de trabalhos geradas pelas aplicações pertinentes aos negócios destas instituições. Empresas e cooperativas, como a Unimed[1], já optaram por migrar seus serviços, antes alocados em infraestruturas de terceiros, para dentro de seus ambientes locais. No contexto acadêmico, como exemplo, a Universidade de São Paulo (USP) disponibiliza, tanto para comunidade acadêmica quanto comunidade externa, o projeto InterNuvem[2], o qual oferece acesso, por parte de pesquisadores, a serviços de

[1] `http://www.unimed.coop.br/portalunimed/relatorio2013/realizacoes.html`. Acesso em: 15 outubro 2020.

[2] `https://cetisp.sti.usp.br/competencias/internuvem/`. Acesso em: 13 fevereiro 2023.

armazenamento e processamento de dados de alto desempenho em nuvens computacionais interconectadas. Já a empresa Tecnologia Bancária (TecBan)[3], responsável pela rede de caixas eletrônicos Banco24Horas, utiliza infraestrutura de nuvem privada para hospedar sistemas críticos, que necessitam de maior garantia quanto à confiabilidade e segurança, exigidos pelo modelo de negócio ao qual pertence.

Cargas de trabalho, principalmente aquelas compostas por tarefas grandes e complexas, são predominantes em ambientes distribuídos como grades e nuvens. Com a popularização e constante evolução das nuvens computacionais, demandas pela execução de aplicações que exigem alto poder de processamento se tornam mais presentes, oportunizando frentes de pesquisa sobre a viabilidade de execuções *High Performance Computing* (HPC) em nuvens (públicas ou privadas). Consequentemente, também são realizados esforços objetivando identificar os tipos de aplicações que terão melhor desempenho em ambientes altamente distribuídos e consequências financeiras na adoção de soluções em cada um dos tipos de nuvem. Para os provedores de serviço de nuvem, o principal objetivo é a entrega de conteúdo e não aplicações que demandem alto poder de processamento. Com isso, é necessário analisar o comportamento das aplicações e o cenário de uso pretendido, a fim de que seja possível a escolha da infraestrutura mais adequada para execução ((ROLOFF et al., 2012), (KEHRER; BLOCHINGER, 2019)).

Um dos aspectos que promovem a adoção de infraestruturas de nuvem por centros de pesquisa ou instituições comerciais é o financeiro. No caso de nuvens públicas, observa-se um menor custo total de propriedade (TCO) e uma maior flexibilidade em termos de recursos. Isso vem por permitir, à instituição, foco nos negócios, ignorando problemas relacionados ao gerenciamento da infraestrutura (ACETO et al., 2013). Quando aplicado à solução de nuvem privada, a centralização dos recursos tem efeito similar, concentrando os custos de propriedade e de gestão em um setor especializado da instituição, mas distribuindo entre os demais setores, seu uso.

Para Filiopoulou et al. (2015), a estimativa do custo total de propriedade, em particular, é um procedimento que fornece os meios para determinar o valor econômico total de um investimento, incluindo as despesas iniciais de capital (CAPEX) e as despesas operacionais (OPEX). No contexto de computação em nuvem, corresponde à estimativa de valores necessários para implementação e operação de uma infraestrutura de nuvem.

Neste trabalho, o foco está em analisar os investimentos financeiros em infraestruturas de nuvem para execução de uma determinada aplicação. Para os usuários de nuvem, é importante obter o melhor desempenho a partir de custos monetários reduzidos. Assim, eventuais desperdícios relacionados às possíveis perdas financeiras decorrentes do baixo índice de produção ou do passivo construído em *hardware* não explorado, devem ser monitorados levando em consideração a satisfação do usuário, no que diz respeito ao tempo

[3] http://epocanegocios.globo.com/Revista/Common/0,,EMI96465-17453,00-NUVEM+PUBLICA+OU+PRIVADA.html. Acesso em: 15 outubro 2020.

de execução de suas demandas, TCO e o retorno sobre o investimento (ROI). Com isso, é possível analisar se as ofertas de computação em nuvem, principalmente aquelas focadas no modelo de Infraestrutura como Serviço (IaaS), são capazes de atender os diferentes tipos de demandas, levando em consideração os custos envolvidos nas operações.

Na literatura, alguns trabalhos tratam questões relacionadas aos parâmetros a serem considerados na implantação de infraestruturas de nuvem. Em seu estudo, Cui (2017), categorizou o modelo de custos em cinco grandes aspectos: infraestrutura, servidores, rede, energia e custo de manutenção, com o objetivo de identificar quais elementos consomem mais recursos financeiros e as principais tendências para itens como energia e refrigeração. Falcão e seu grupo (2019), buscou sintetizar informações referentes aos custos de uma nuvem computacional, levando em consideração as despesas com implementação e operação, destacando itens como ROI, eficiência dos *data centers*, relação custo/benefício e uma comparação das estruturas disponíveis no mercado. Nesse contexto, a questão que se coloca é o contraste e a comparação dos benefícios e dos limites, para o usuário, obtidos na solução adotada para implantação da infraestrutura de nuvem em ambiente público ou privado.

O contexto do presente trabalho está delimitado ao uso de recursos computacionais em ambientes acadêmicos para suporte a aplicações HPC. Tais aplicações possuem características específicas que expressam suas necessidades computacionais. Além de necessidades típicas ligadas à capacidade do processador e da disponibilidade de memória, em sistemas distribuídos requisitos ligados à comunicação também são determinantes para verificar sua adequação aos ambientes de nuvem. O uso de elementos de *hardware* não apropriados para alto desempenho e sobrecargas geradas pelos processos de virtualização constituem obstáculos na adoção de nuvens por parte dos usuários de computação de alto desempenho (HPC). Por exemplo, aritméticas de ponto flutuante, comunicação entre processos e até mesmo a escolha de *drivers* para máquinas virtuais podem afetar, significativamente, o desempenho do processo computacional (ALADYSHEV et al., 2018). Com relação às decisões de escalonamento, estas podem afetar o desempenho das tarefas a serem executadas. Escalonadores compatíveis com HPC podem melhorar o desempenho das aplicações em ambientes de nuvem, pois conseguem explorar as propriedades da infraestrutura e das próprias aplicações (NETTO et al., 2018). Assim, algumas questões relacionadas a quem se beneficiaria com a migração para ambientes de nuvem, qual, por que e como cada aplicação HPC pode ser submetida, devem ser consideradas.

Para apoiar o desenvolvimento do trabalho proposto, foi realizada uma Revisão Sistemática da Literatura (RSL). O objetivo, nesta RSL, é o de identificar abordagens que relacionem perdas e ganhos financeiros da solução adotada quanto ao desempenho obtido no suporte à aplicação. Como resultado, são elencados os trabalhos mais relevantes no contexto de análise de custos de implantação de nuvens que suportem a execução de aplicações HPC. De posse deste material é possível analisar, sob diferentes perspectivas, se as ofertas

de computação em nuvem (Infraestrutura como Serviço, principalmente) são capazes de atender os diferentes tipos de demandas, levando em consideração os custos envolvidos nas operações. Nesta abordagem são considerados tanto as soluções que adotam nuvens privadas quanto nuvens públicas.

1.1 Objetivos

O objetivo geral deste trabalho é racionalizar, em relação aos custos financeiros, a adoção de nuvens computacionais para apoio à pesquisa acadêmica, determinando os custos envolvidos para implantação de uma solução, atendendo a uma comunidade federada de instituições de ensino e pesquisa. Serão abordadas alternativas com ambientes de nuvens públicas, privadas e híbridas, identificando vantagens e desvantagens da adoção de cada uma e os respectivos impactos financeiros.

Para que o objetivo geral desta Tese seja concretizado, alguns objetivos específicos (**OE**) necessitam ser atingidos. São eles:

OE1 - Modelar diferentes configurações para soluções de nuvem, sejam elas privadas, públicas ou híbridas, para atender um conjunto de instituições de ensino e pesquisa;

OE2 - Identificar custos associados à implementação de diferentes estratégias de nuvem, verificando a demanda de recursos computacionais requerido pelas pesquisas de diferentes áreas;

1.1.1 Metas

Os objetivos específicos desta Tese serão atingidos a partir do cumprimento das seguintes metas (**M**):

M01 - Determinar qual modelo de implantação (pública, privada ou híbrida) melhor atende o uso pretendido pelas instituições de pesquisa (**OE1**);

M02 - Precificar componentes de uma nuvem no que diz respeito a itens de *hardware*, *software*, infraestrutura de suporte e recursos humanos (**OE2**);

M03 - Definir o modelo de custos que será utilizado na avaliação (**OE2**);

M04 - Aplicar o modelo de custos e identificar possíveis problemas de dimensionamento de infraestruturas (**OE2**);

M05 - Projetar o crescimento infraestruturas de nuvem para uso em pesquisas acadêmicas (**OE2**).

1.2 Metodologia

O presente trabalho tem por objetivo caracterizar as relações de custo e desempenho no uso de uma infraestrutura de nuvem federada apoiada por uma infraestrutura de nuvem pública para suporte ao apoio à pesquisa científica acadêmica. O trabalho se propõe a avaliar o comportamento da infraestrutura construída considerando diferentes cargas de submissão de trabalho e diferentes configurações dos recursos de processamento, assim como de diferentes estratégias de escalonamento das aplicações lançadas sobre os recursos disponibilizados.

Como modelo de aplicações para o contexto científico, foi assumido o modelo de aplicação *workflow*, sendo consideradas aplicações apresentadas no projeto Pegasus (PEGASUS, 2022) como modelo de aplicações submetidas pelos usuários. Nesta tese foi considerado que os *workflows* submetidos à nuvem são compostos de tarefas descritas em dois níveis: tarefas no nível *workflow* e tarefas no nível *bag-of-tasks*. As tarefas no nível *workflow* apresentam relações de dependência entre si, que devem ser obedecidas para correção da execução. No entanto, internamente, cada tarefa do nível *workflow* é composta por um conjunto de tarefas do tipo *bag-of-tasks*, em que não há dependência entre elas. O problema do escalonamento, portanto, se dá no atendimento às dependências do nível *workflow* e na exploração da concorrência das tarefas no nível *bag-of-tasks*.

Como modelo de infraestrutura, assumiu-se a existência de sítios, cada sítio pertencente a uma instituição federada a uma infraestrutura de nuvem, oferecendo um conjunto de recursos de processamento. A esta nuvem federada pode estar associada uma nuvem pública, cujos recursos podem ser explorados via *cloud bursting*.

A precificação do uso desta infraestrutura considera que cada instrução, tecnicamente, cada milhão de instruções, possui um custo associado. Os custos de execução sobre a nuvem federada e a nuvem pública são diferentes entre si, e devem ser considerados de forma separada.

A proposta de trabalho, então, baseia-se na concepção de cenários de uso de infraestruturas de nuvem combinando sítios federados e nuvem pública sob diferentes estratégias de escalonamento para exploração da configuração avaliada. Dentre estas estratégias de escalonamento incluem-se aquelas que realizam *cloud bursting*.

Para execução do trabalho foram identificados simuladores para nuvens populares na comunidade científica. Os estudos destes simuladores mostraram suas deficiências no que diz respeito à modelagem de aplicações do tipo *workflow*, além de requererem pacotes auxiliares para realizar a contabilização de custos. A decisão para apoiar o suporte para realização dos experimentos recaiu, assim, em retomar resultados prévios, documentados em (SANTOS, 2016), onde um simulador para aplicações *bag-of-tasks* foi apresentado. Este simulador foi estendido para incorporar diferentes funcionalidades requeridas nesta tese, em particular, a possibilidade de organizar as tarefas nos dois níveis citados anteriormente,

tarefas *workflow* e tarefas *bag-of-tasks*. Grande parte do desenvolvimento da tese foi tomado em conceber este simulador, tornando-o genérico para descrever o conjunto de cenários possíveis no contexto de avaliação ao que o trabalho se propôs.

A avaliação se deu pela especificação de diferentes casos de uso, nos quais foram variadas cargas de trabalho submetidas à nuvem, as configurações de infraestrutura implantadas nos sítios e as estratégias de escalonamento. O conjunto de cenários de avaliação se mostrou bastante amplo, graças ao simulador construído, o qual oferece uma grande gama de recursos de configuração de uso. A avaliação realizada permitiu verificar o desempenho e custos de diferentes configurações de infraestrutura de nuvem para diferentes cenários de carga de trabalho.

1.3 Contribuições Científicas

O principal resultado desta tese documentado no Capítulo 6 é a constatação de que uma nuvem pública, como apoio a uma nuvem federada, pode ser utilizada para melhorar tanto o desempenho em termos de tempo de execução das aplicações quanto o custo monetário destas execuções. Estes ganhos não ocorrem em todas as situações, uma vez que dependem do custo computacional submetido a nuvem e também da configuração da infraestrutura implantada. Este conjunto de resultados deve ser estendido pela realização de um número maior de experimentos, avaliando diferentes estratégias de escalonamento.

Outra contribuição importante desta tese é a construção de um novo simulador para ambientes de rede, o WCSim (*Workflow Cloud Simulator*), apresentado no Capítulo 5. Este simulador se diferencia dos demais simuladores presentes na literatura, notadamente SimGrid (CASANOVA et al., 2014) e CloudSim (CALHEIROS et al., 2009), por permitir a adaptação da configuração da infraestrutura e da carga de trabalho submetida por arquivos de entrada, não exigindo esforço de programação. Outro diferencial deste simulador é perceber que as aplicações submetidas a uma nuvem são compostas de tarefas caracterizadas em dois níveis. Em um nível, as tarefas relacionam-se entre si por uma relação de fluxo de dados, caracterizando um *workflow* entre elas. Em um nível interno às tarefas *workflow*, um novo conjunto de tarefas, relacionam-se entre si como um *bag-of-tasks*, podendo ser executadas concorrentemente, possibilitando explorar todo o paralelismo da arquitetura.

1.4 Conhecendo brevemente este trabalho

O presente trabalho tem como objetivo estender a discussão sobre a adoção de infraestruturas de nuvem para suporte a computação em ambiente científico acadêmico. Neste meio, o meio científico acadêmico, é comum grupos de pesquisa em diferentes áreas compartilharem infraestruturas físicas. No que diz respeito ao compartilhamento de recursos de processamento, infraestruturas como grades rapidamente se apresentaram

como uma alternativa ao compartilhamento de recursos computacionais. Em um momento de universalização do acesso aos recursos de processamento, assim como vislumbrando a possibilidade de melhor aproveitamento de recursos financeiros, o ambiente de nuvem computacional se apresenta. A viabilidade de construir uma nuvem federada, utilizando recursos de processamento de forma capilarizada, associado a um provedor externo capaz de estender o conjunto de recursos sob demanda, pagando pelo uso, passa a ser um grande atrativo.

O pano de fundo desta tese é justamente a discussão do uso de um provedor de processamento externo a uma nuvem composta por recursos de processamento de instituições federadas. Sobre este pano de fundo, se apresentam as contribuições materiais desta tese. A primeira se dá na caracterização da modelagem de aplicações como *workflows* de *bag-of-tasks*. Este modelo de aplicação considera que as aplicações são compostas de *bag-of-tasks* que se relacionam entre si por relações de dependência. A segunda contribuição material é o simulador WCSim desenvolvido para apoiar a realização dos estudos de caso. Este simulador encontra-se disponível[4] para contribuições.

1.5 Estrutura do texto

O restante do texto desta Tese está organizado como segue.

No Capítulo 2 são apresentados os requisitos de execução para aplicações HPC, juntamente com os diferentes custos associados à implantação de nuvens, assim como caracterizados os aspectos que influenciam nas suas respectivas variações. Neste capítulo são apresentados os modelos de aplicações submetidos à nuvens computacionais: *bag-of-tasks* e *workflow*.

O Capítulo 3 apresenta um apanhado, retirado da literatura, sobre questões que envolvem a precificação de uso de infraestruturas de nuvem.

O Capítulo 4 apresenta a Revisão Sistemática da Literatura, juntamente com os trabalhos relacionados ao tema deste estudo.

O simulador WCSim é apresentado no Capítulo 5. Além de apresentar o diagrama de classes no qual o código está organizado, este capítulo detalha os parâmetros de entrada, que descrevem a carga de trabalho a ser simulada e a configuração da infraestrutura de nuvem, e as saídas oferecidas.

O conjunto de casos de estudo é detalhado no Capítulo 6. Neste capítulo são tecidas as observações sobre desempenho e custo dos cenários avaliados.

O Capítulo 7 conclui o trabalho e apresenta desdobramentos para trabalhos futuros.

[4] `https://github.com/madsantos/WCSim`

2 APLICAÇÕES HPC SOBRE NUVENS COMPUTACIONAIS

Este capítulo apresenta uma visão sobre as aplicações que são lançadas sobre nuvens computacionais, algumas definições e conceitos importantes para este trabalho. As duas primeiras seções apresentam os modelos de aplicações que, segundo a literatura, são os mais frequentes e, ambientes de rede: *bag-of-tasks* (Seção 2.1) e *workflow* (Seção 2.2). A Seção 2.3 apresenta questões sobre o custo de execução destas aplicações, com vistas a precificar o custo de uma infraestrutura e a Seção 2.4 traz conceitos e definições sobre *cloud bursting*.

2.1 Bag of Tasks

De acordo com (CIRNE; BRASILEIRO; SAUVÉ, 2003), *Bag of Tasks* (BoT) são aplicações paralelas cujas tarefas são independentes entre si, não exigindo, portanto, cuidados de sincronização mútua em tempo de execução. A característica de independência das aplicações se apresenta em uma vasta gama de cenários, tais como mineração de dados, simulações, buscas massivas e muitas outras aplicações científicas com alta demanda de processamento. Outro ponto de destaque se deve ao fato da independência entre as tarefas tornar possível que aplicações BoT sejam executadas com sucesso em ambientes altamente distribuídos, como nuvens computacionais.

A infraestrutura provida por nuvens computacionais fornece suporte para a execução de aplicações do tipo BoT. Em muitos casos, as tarefas pertencentes às aplicações BoT possuem um alto custo computacional e para evitar grandes investimentos em recursos de *hardware*, usuários podem submeter às infraestruturas computacionais em nuvem, de forma oportunista, suas demandas de processamento para este tipo de aplicação.

Neste trabalho, uma aplicação BoT é descrita por um conjunto A composto por $n \geq 1$ quádruplas descrevendo, cada uma, um conjunto de tarefas τ, na forma:

$$A = \{q_1 \dots q_n\} \text{ onde } q_i = [a_i, d_i, b_i, c_i], \forall i | 1 \leq i \leq n \quad (1)$$

Os elementos destas quádruplas são: ***a***, o instante de tempo no qual o conjunto de tarefas chega para o processamento; ***d***, o tempo de duração do grupo de tarefas; ***b***, a

quantidade de tarefas descritas por quádrupla correspondente; e ***c***, a taxa de utilização de CPU para cada tarefa da quádrupla. O valor informado para o custo deve estar entre 1 e 100, indicando o percentual de uso da CPU de cada uma das tarefas quando em execução. As informações relacionadas a tempo, a e d, são informadas na unidade adotada. No modelo adotado, as informações de tempo consideram a existência de um número não limitado de processadores e sua execução em um ambiente livre de contenção. Assim, cada quádrupla q_i descreve um grupo de b_i tarefas $\tau_i^j, \forall j | 1 \leq j \leq b_i$ que são inseridas no *bag* no tempo a_i, com duração d_i e custo individual de c_i.

O número total de tarefas de uma aplicação A composta por n quádruplas é dado por:

$$N_A^n = \sum_{i=1}^{n} b_i \tag{2}$$

Não existe nenhuma restrição de ordem de execução entre as tarefas definidas em cada quádrupla, uma vez que, por definição, as tarefas em um BoT são independentes. No entanto, a data de chegada a_i das tarefas definidas pela tupla q_i impõe que nenhuma tarefa definida por q_i inicie antes do tempo a_i.

A seguir é apresentado um exemplo da descrição de uma aplicação conforme o modelo definido.

$$A = \{[0, 5, 3, 50], [0, 8, 5, 30], [3, 4, 6, 80], [4, 10, 2, 20]\} \tag{3}$$

O exemplo de aplicação apresentado consiste em uma aplicação A composta de um total de 16 tarefas, distribuídas em quatro grupos de tarefas idênticos. Os dois primeiros grupos definem tarefas que chegam no tempo 0 (zero) do processamento. As tarefas do terceiro e do quarto grupo são adicionadas ao *bag* nos tempos 3 e 4. No modelo adotado, estes tempos consideram a existência de um número não limitado de processadores e sua execução em um ambiente livre de contenção.

Para efeito de controle de evolução das aplicações BoT, o tempo é assumido discreto. Desta forma, os atributos descrevendo unidade tempo, os atributos a e d das quádruplas (tempo de chegada e duração) definem a relação de granularidade entre as tarefas, nas quais sendo maior o tempo, mais grossa a granularidade, podendo cada unidade ser mapeada em um valor qualquer, alterando, para uma mesma aplicação, a granularidade de todas as tarefas de modo uniforme.

Define-se *passo* a passagem de uma unidade de tempo. Desta forma, a duração d_i de uma tarefa τ_i descreve o número de passos necessários para executá-la. Ao conjunto de instruções de passo de uma tarefa se dá o nome de *job*. Assim, cada tarefa τ_i define d_i *jobs* pelo conjunto $\{w_1 \ldots w_{d_i}\}$, os quais devem ser executados na estrita sequência de ordem na qual foram definidos, garantindo que w_1 não seja executado antes do passo definido por a_i. O passo a_i deve ser interpretado, no caso da execução da aplicação em uma arquitetura não contingenciada, pelo número total de passos previamente executados pela aplicação.

Inexistem relações de ordem entre os *jobs* de diferentes tarefas, pelas razões já expostas. O número total de *jobs* de uma aplicação A descrita por n quádruplas é dado por:

$$N_A^* = \sum_{i=1}^{n} d_i \times b_i \tag{4}$$

Na Figura 1 são apresentados os 6 primeiros passos na evolução da aplicação A descrita no exemplo anterior. Observe que nesta representação está sendo assumido a existência de um número não limitado de recursos de processamento e inexistência de condições de contenção de execução. No passo inicial (passo=0), 8 tarefas estão presentes no bag e são lançadas para execução. Terminado o passo, avança o tempo (passo=1) e um *job* de cada uma das tarefas é completado. O mesmo ocorre ao término do passo 1 e do passo 2. No passo 3, um novo conjunto de tarefas é recebido no bag e lançado. Ao final do passo 3, todas as tarefas em execução finalizam a execução de um *job*. No passo 4, como no passo anterior, um novo conjunto de tarefas inicia a execução e todas as tarefas completam a execução de um *job*. Neste passo ocorre também que um conjunto de tarefas executa seu último *job*, finalizando sua execução. No passo seguinte (passo=5) as tarefas remanescentes avançam, concluindo mais um passo. A execução prossegue reproduzindo o padrão apresentado.

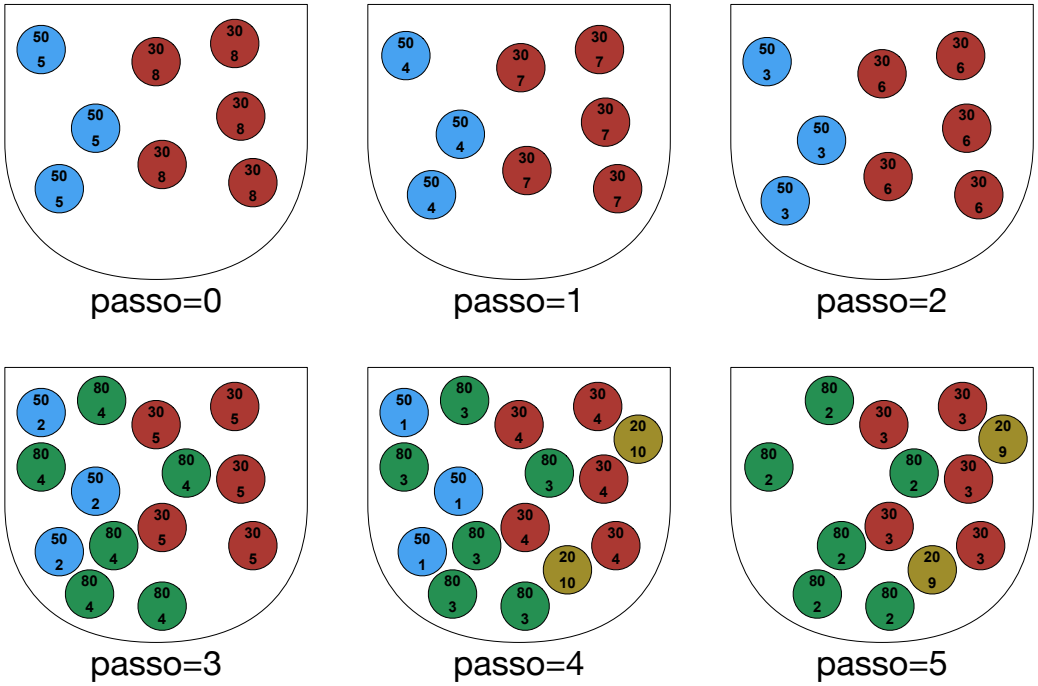


Figura 1 – Exemplo de aplicação *Bag of Tasks*.

2.2 Fluxos de Trabalho Científicos - *Workflows*

Os fluxos de trabalho científicos (*workflows*) estão presentes em diversos domínios de aplicação e pesquisa, como por exemplo, astronomia, biologia, física gravitacional, estudos

de terremotos, entre outros. Eles permitem que os usuários expressem facilmente tarefas computacionais de várias etapas, por exemplo, recuperar dados de um instrumento ou banco de dados, reformatar os dados e executar uma análise.

Um fluxo de trabalho científico descreve as dependências entre as tarefas e, na maioria dos casos, é descrito como um grafo acíclico direcionado (DAG), no qual os nós são tarefas e as arestas denotam as dependências entre tarefas. As tarefas em um fluxo de trabalho científico podem ser desde tarefas seriais curtas até tarefas paralelas muito grandes, cercadas por um grande número de tarefas seriais pequenas usadas para pré e pós-processamento.

Pesquisas científicas utilizam conjuntos de dados cada vez mais complexos a fim de executar simulações e análises. Neste cenário, tecnologias de fluxos de trabalho científicos buscam maneiras de gerenciar a complexidade das aplicações científicas e a maneira de lidar com grandes quantidades de dados.

À medida que os fluxos de trabalho científicos crescem em complexidade e importância, é necessário um maior entendimento com relação à gerência de tais estruturas. Torna-se importante conhecer os requisitos dos *workflows* (fluxos de trabalho), bem como seus comportamentos, que podem conduzir para significativas melhorias no que tange a algoritmos para provisionamento de recursos computacionais, escalonamentos de *jobs* e manuseio dos dados (JUVE et al., 2013).

Como exemplo de um fluxo de trabalho científico, a Figura 2 apresenta o SIPHT. Este fluxo representa uma aplicação que procura por pequenos RNAs não traduzidos, conhecidos como sRNA, que regulam processos como secreção e virulência em bactérias. Tal projeto de bioinformática, a cargo da Universidade de Harvard, utiliza um fluxo de trabalho para automatizar a busca de genes codificadores de sRNA para todos os *replicons* bacterianos em um grande banco de dados de informações sobre biotecnologia.

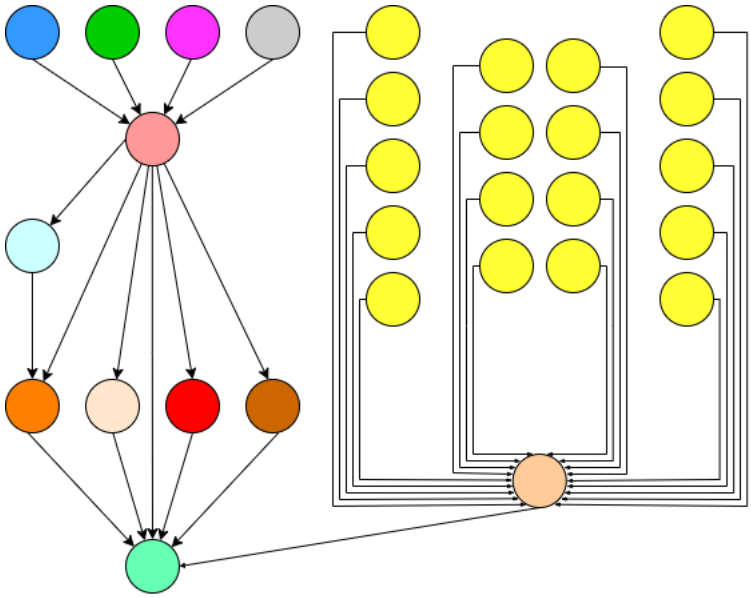


Figura 2 – Fluxo de trabalho SIPHT. Adaptado de (JUVE et al., 2013).

No contexto de fluxos de trabalho científicos, o projeto Pegasus (PEGASUS, 2022) surge com uma alternativa para a execução de aplicações baseadas em *workflows* em diferentes ambientes, que podem ser desde *desktops* ou *notebooks* até infraestruturas computacionais de grande porte como *clusters*, *grids* ou nuvens.

Com o Pegasus é possível conectar o domínio científico e o ambiente de execução, realizando o mapeamento automático das descrições do fluxo de trabalho em recursos de computação distribuídos. É possível localizar automaticamente os dados de entrada necessários e os recursos computacionais para a execução do fluxo de trabalho. O Pegasus permite que os cientistas construam fluxos de trabalho em termos abstratos sem se preocupar com os detalhes do ambiente de execução subjacente ou com os detalhes das especificações de baixo nível exigidas pelo *middleware* que estiver sendo utilizado.

Na Figura 3 é possível observar a arquitetura do sistema de gerenciamento de *workflows* que faz parte do Projeto Pegasus.

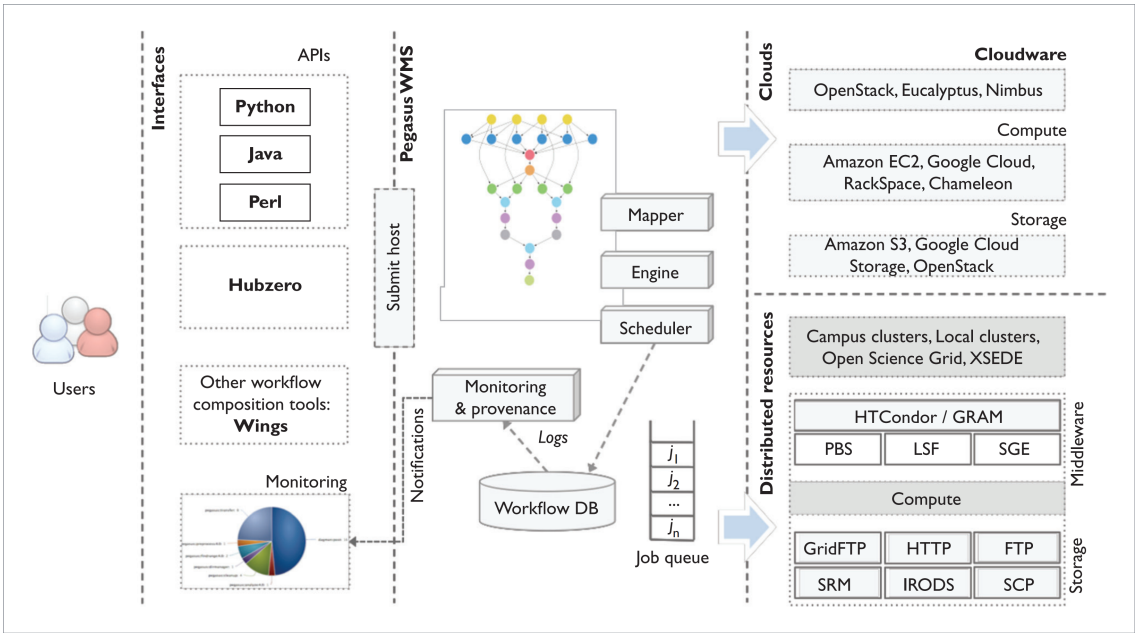


Figura 3 – Arquitetura do Pegasus Workflow Management System (WMS) (DEELMAN et al., 2016).

2.3 Aspectos de Custo

Aplicações para HPC possuem diferentes características que podem determinar sua adequação a um ambiente de nuvem. Questionamentos a respeito de por que e quem deve escolher uma nuvem para execução de aplicações HPC, quais destas aplicações e como a nuvem pode ser usada para HPC devem balizar os estudos de viabilidade de migração para ambientes remotos distribuídos.

Ambientes HPC são orientados a desempenho, ao passo que nuvens computacionais são orientadas pela relação custo (monetário) *vs.* utilização de recursos. Além disso, nuvens foram, originalmente, projetadas para a execução de aplicações comerciais e de serviços

para a internet. O uso de conexões de rede não apropriadas para alto desempenho, a sobrecarga das técnicas de virtualização e as limitações dos sistemas de armazenamento podem ser considerados barreiras para a adoção de nuvens para aplicações HPC. Cabe, então, identificar (GUPTA et al., 2016):

- *Quem* é o usuário candidato para um ambiente de nuvem;
- *Qual* é o tipo de aplicação que pode se beneficiar de uma execução em ambiente de nuvem;
- *Por que* o usuário terá benefícios na execução de sua aplicação em ambiente de nuvem; e,
- *Como* este benefício pode ser atingido.

Com base nestes pontos, as Tabelas 1 e 2 apresentam alguns posicionamentos com relação aos questionamentos originados a partir de duas diferentes abordagens que visam auxiliar nas decisões de migração para ambientes de nuvem computacional: *a*) considera-se os aspectos da execução em modelos de desempenho, custo e negócios; e *b*) são exploradas técnicas para preencher as lacunas entre nuvens e aplicações HPC. Estas abordagens têm por objetivo identificar as diferenças de demandas por HPC, por parte dos usuários, e quais alternativas eles dispõem para implantar suas aplicações. As alternativas que buscam preencher a lacuna entre aplicações HPC e nuvens podem ser classificadas em duas categorias (GUPTA et al., 2016): primeiro, aquelas que objetivam tornar as nuvens cientes das aplicações HPC e, segundo, aquelas que buscam tornar as aplicações HPC cientes das infraestruturas de nuvem nas quais serão executadas.

A Tabela 1 traz respostas para a o cenário no qual é pretendido tornar as nuvens cientes das aplicações HPC que serão executadas. Para isso, são exploradas técnicas de virtualização leve (por exemplo, contêineres) e determina-se quão próximo do desempenho de máquina física é possível chegar com as aplicações HPC. São exemplos, *hypervisors* otimizados para HPC e nuvens otimizadas para computação de alto desempenho como Amazon HPC.

Tabela 1 – Questões sobre nuvens cientes de aplicações HPC

Questionamento	**Resposta**
Quem	Pequenas e médias organizações ou empresas em crescimento.
Qual	Aplicações com padrões de comunicação menos intensos e menos sensíveis a interferências.
Por que	Pequenas e médias organizações que são sensíveis aos argumentos de CAPEX/OPEX.
Como	Tornar as nuvens cientes das aplicações HPC, por exemplo, com uso de virtualização leve e afinidade de CPU.

Fonte: Adaptado de (GUPTA et al., 2016).

Já na Tabela 2, as respostas tratam de pontos que devem ser considerados quando se busca por ambientes de execução capazes de trabalhar com aplicações HPC cientes de

nuvens computacionais. Esta alternativa, embora ainda pouco explorada, permite ajustes nos tempos de execução das aplicações HPC em nuvens para obtenção de um melhor desempenho com uso de tecnologias como balanceadores de carga com reconhecimento de nuvem para aplicações HPC e a implantação de topologias com reconhecimento de aplicações científicas na nuvem.

Tabela 2 – Questões sobre aplicações HPC cientes de nuvem

Questionamento	**Resposta**
Quem	Usuários com aplicações que tem melhor relação custo/desempenho em nuvens *vs.* outras plataformas.
Qual	Aplicações com necessidades de desempenho que podem ser atendidas em média escala (em termos de número de *cores*).
Por que	Nuvens permitem que vários usuários acessem estruturas compartilhadas, garantindo um melhor uso dos recursos.
Como	Abordagem híbrida supercomputador-nuvem com escalonamento ciente de aplicação e *cloud bursting*.

Fonte: Adaptado de (GUPTA et al., 2016).

O uso de nuvens para execução de aplicações HPC pode ser visto como um bom complemento para estruturas locais de supercomputadores e *clusters*, não podendo, ainda, substituí-las totalmente. Abordagens de utilização de nuvens híbridas, nas quais ocorre uma integração entre infraestruturas de nuvens públicas e privadas, permitem a ocorrência de *cloud bursting* (MANSOURI; PROKHORENKO; BABAR, 2020). Isto faz com que aplicativos utilizem toda a capacidade dos recursos computacionais de uma nuvem privada e, em reação a este aumento, migrem suas tarefas em ambientes de nuvem pública, à medida que os recursos locais se tornem escassos.

Aplicações científicas tem necessidades significativamente diferentes das aplicações comerciais típicas, executando suas tarefas de maneira fortemente acoplada e em escalas bem maiores de recursos computacionais. Tal comportamento leva a requisitos de largura de banda e latência mais exigentes do que a maioria dos usuários de nuvem. Aplicações científicas também requerem acesso a grandes quantidades de dados e isso pode levar a um grande custo de inicialização e armazenamento (YELICK et al., 2011).

Cargas de trabalho científicas podem ser classificadas em três categorias abrangentes de acordo com seus requisitos: fortemente acopladas em grande escala, médio alcance e alto rendimento. A Tabela 3 apresenta características que favorecem o entendimento da viabilidade de execução de aplicações HPC em nuvens. Também classifica, em alto nível, cargas de trabalho executadas pela comunidade científica.

A partir da classificação das cargas de trabalho científicas, caracterizadas anteriormente (Tabela 3), é possível identificar alguns exemplos de cada um dos tipos de aplicação, conforme descrito na Tabela 4.

Em seu estudo, o grupo de Parashar (PARASHAR et al., 2013) apresentou uma divisão dos ambientes de nuvem para HPC em três categorias: *a*) ***HPC in the Cloud***, que se

Tabela 3 – Recomendação de uso de nuvens em função do tipo de aplicação HPC

Tipo de Aplicação	Recomendação
Fortemente acoplada em grande escala	Tipicamente, aplicações MPI, que utilizam milhares de *cores* e exigem uma rede de comunicação de alto desempenho. Neste caso, qualquer gargalo de virtualização ou problemas de alta latência na rede terão impacto negativo no desempenho das aplicações. Logo, é recomendável a execução em ambientes tradicionais de supercomputação ou em nuvens privadas com servidores *bare metal* e redes de alta velocidade.
Médio alcance	Estas aplicações utilizam um número variado de *cores* (dezenas ou centenas) e têm requisitos de desempenho mais baixos do que as aplicações de grande escala fortemente acopladas. Consequentemente, são mais tolerantes à virtualização e redes tradicionais. É recomendado que estas aplicações explorem os benefícios do acesso rápido a recursos para a nuvem, especialmente a virtualização leve (contêineres).
Alto rendimento	Aplicações compostas por tarefas independentes que exigem pouca ou nenhuma comunicação. Tais aplicações podem se beneficiar de um grande número de recursos disponíveis e são tolerantes à heterogeneidade. Recomenda-se o uso de nuvens para estas aplicações, especialmente explorando mecanismos de elasticidade. Outros benefícios também podem ser alcançados por meio de ambientes de nuvem híbrida HPC, distribuindo tarefas em *clusters* locais e nuvens públicas.

Fonte: Adaptado de (YELICK et al., 2011).

Tabela 4 – Exemplos de aplicações HPC

Tipo de Aplicação	Exemplos
Fortemente acoplada em grande escala	Aplicações MPI; aplicações de geração de modelos climáticos, sísmicos.
Médio alcance	Aplicações de simulação orientada a eventos; aplicações escalonadas no tempo, cujos trabalhos são menos sensíveis a prazos.
Alto rendimento	Aplicações BoT; aplicações MapReduce; simulações Monte Carlo.

concentra em mover aplicações HPC para ambientes de nuvem; *b*) ***HPC Plus Cloud***, na qual usuários fazem uso de nuvens para complementar seus recursos de HPC, em situações de picos de demanda (*cloud bursting*); e *c*) ***HPC as a Service***, que expõe os recursos HPC por meio de serviços de nuvem. Estas categorias estão relacionadas a como os recursos são alocados e abstraídos para simplificar o uso da nuvem.

A execução de aplicações HPC em nuvem ainda possui vários problemas em aberto. Para exemplificar, a abstração da infraestrutura de nuvem limita o ajuste das aplicações. Além disso, a maioria das conexões de rede dos provedores de nuvem não é rápida o suficiente para aplicações fortemente acopladas em grande escala, com alta comunicação entre processadores.

O modelo de negócio para uma nuvem HPC também é um campo a ser bem explorado. Em nuvens públicas, provedores de serviços lançam várias cargas de trabalho sobre os mesmos recursos físicos a fim de explorar economia de escala, que nem sempre é apropriada para HPC. Além disso, mesmo que pequenas empresas se beneficiem do rápido acesso a recursos de nuvens públicas, isso nem sempre é verdadeiro para grandes usuários

de HPC (NETTO et al., 2018).

Por outro lado, em ambientes de nuvens privadas, usuários de HPC podem ter acesso direto ao gerenciamento dos componentes e o compartilhamento de recursos é reduzido por consequência de o modelo de múltiplos inquilinos ficar limitado a um grupo específico de usuários.

Questões referentes à troca de CAPEX por OPEX tem destaque no processo de decisão de migração de aplicações para ambientes de nuvem. CAPEX está relacionado com os investimentos feitos na aquisição de equipamentos, *softwares* e inicialização dos mesmos dentro do provedor de serviço. Já o OPEX diz respeito aos investimentos feitos com alocação de serviços, como por exemplo, manutenção de equipamentos e contratação de serviços de nuvem.

Aplicações que fazem uso variável dos recursos computacionais determinam uma menor utilização global dos equipamentos. Juntamente com os pontos referentes a CAPEX e OPEX, esta utilização variável de recursos serve de argumentos para provedores e usuários finais de nuvem. Os usuários podem se beneficiar caso suas execuções se caracterizem como aplicações de uso variável (GUPTA et al., 2016). Por outro lado, provedores de nuvem podem tirar benefícios de uma utilização agregada de recursos de todos os seus inquilinos. Para isso, é fundamental que a agregação possa sustentar um modelo de precificação lucrativo frente aos grandes investimentos iniciais necessários para oferecer recursos de computação e armazenamento por meio de uma interface em nuvem pública. No caso de uma nuvem privada, o lucro está em oferecer mais produtividade para os usuários, atendendo de modo satisfatório às demandas da instituição.

2.4 Escalonamento em Infraestruturas Híbridas

A execução em uma infraestrutura híbrida necessita uma estratégia de escalonamento que seja capaz de detectar quando os recursos locais estão saturados e uma alternativa deve ser buscada. Esta alternativa é buscada por uma estratégia de escalonamento chamada de *cloud bursting*

Cloud bursting pode ser definido como a utilização temporária de recursos de nuvens computacionais por meio de infraestruturas de nuvens híbridas (JIGSAW, 2021). Neste modelo de implantação, as aplicações que são executadas em ambientes locais, como nuvens privadas, são repassadas, geralmente uma parte, para nuvens públicas, com o objetivo de evitar a interrupção dos serviços.

De modo geral, as organizações reservam a opção de *cloud bursting* para aplicações que tendem a gerar picos de demanda e grandes flutuações na utilização de recursos computacionais. No entanto, aplicações muito complexas ou que dependam de recursos específicos de computação não costumam ser boas candidatas para a execução explodida em uma nuvem pública (NETAPP, 2020).

A utilização de nuvens públicas, como complemento para os recursos locais de computação, pode se tornar uma opção mais prática às organizações que já operam seus próprios sistemas computacionais *in-house*. Vantagens como escalabilidade, com o uso de escalonamento horizontal, na qual picos esporádicos de cargas de trabalho podem ser tratadas de forma mais eficaz; a utilização de recursos, na qual o nível de provisionamento pode ser ajustado para a carga de trabalho em níveis normais de uso; e a segurança que permite que uma política multinuvem possa ser aplicada, principalmente, devido à adoção de nuvens públicas. Em ambientes que fazem uso de *cloud bursting*, os dados confidenciais poderiam ser mantidos e tratados no ambiente privado (nuvem privada) ao passo que cargas de trabalho menos sensíveis à políticas de segurança poderiam ser transferidas para nuvens públicas sem comprometimento da confidencialidade dos dados (LEE; LIAN, 2017).

2.5 Discussão

De forma comparativa, aplicações descritas por BoTs, por não possuírem restrições de execução das tarefas, impõem um número menor de restrições ao uso de nuvens computacionais que aplicações descritas por *workflows*. A simples existência de dependências entre as etapas da aplicação implica na necessidade de considerações sobre os custos de comunicação entres os recursos de processamento para o lançamento das atividades. Por outro lado, aplicações do tipo *workflow* podem ser vistas como um modelo mais representativo do universo de aplicações HPC possíveis, uma vez que cada etapa de seu processamento, ou seja, cada nó que descreve uma atividade a ser executada pode, por sua vez, ser decomposta em um novo *workflow* ou mesmo em um conjunto de tarefas independentes, ou seja, em um BoT.

No presente trabalho, o modelo de aplicação HPC é considerado como um *workflow* cujas etapas de processamento descrevem BoTs. A este modelo deu-se a denominação de grafo dirigido de *bag-of-tasks*, sendo referido como DoB (acrônimo para *Directed acyclic graph of Bag-of-tasks*).

Aplicações DoB necessitam de muitos recursos computacionais e fornecê-los de maneira otimizada requer ajustes em vários aspectos (GANTIKOW et al., 2015). Em termos de desempenho, o uso de virtualização já está bem mais aprimorado, devido ao suporte à virtualização no nível de sistema operacional, fornecendo desempenho próximo às infraestruturas de *bare metal*.

Do ponto de vista do fluxo de trabalho, a transição de uma infraestrutura de *hardware* dedicado para serviços oferecidos em nuvem implica em uma modificação de processos. Esta mudança de processo é especialmente complicada devido ao armazenamento e à transferência de grandes quantidades de dados. Este aspecto pode ser simplificado com a hospedagem dos dados diretamente no provedor de serviços.

Os ambientes de nuvem e as estruturas clássicas para HPC têm maneiras distintas de

gerenciar recursos de computação. Para extrair o melhor desempenho de um ambiente em nuvem, os esforços concentraram-se em aumentar o isolamento entre máquinas virtuais e reduzir a sobrecarga imposta pelas técnicas de virtualização. As estruturas HPC, por sua vez, visam extrair o máximo de desempenho possível da infraestrutura (NETTO et al., 2018).

As solicitações de usuários para acessar exclusivamente partes de um *cluster* HPC são enfileiradas sempre que os recursos estiverem sobrecarregados. Em ambientes de nuvem isso não ocorre devido à disponibilidade "ilimitada" de recursos, também chamada de elasticidade. Além do mais, *hardware* para HPC, especificamente interfaces de rede, são consideravelmente mais caras do que exemplares utilizados para criação de nuvens tradicionais, com *hardware* de *commodity*[1]. Sendo assim, alocar usuários em *hardwares* específicos para HPC, configura um desperdício para o provedor de serviços, caso ele não atenda exclusivamente usuários com demandas HPC.

O desafio em gerenciar recursos HPC em nuvens está em desenvolver um modelo de negócio sustentável, com economia de escala e que ofereça processamento de alto desempenho aos usuários. Para avançar nesta área, esforços de pesquisa devem ser concentrados para que os provedores possam oferecer sistemas de filas e modelos de preços que levem em conta o tempo que os usuários estão dispostos a esperar para ter acesso aos recursos. Alguns modelos flexíveis de aluguel de recursos HPC em nuvens, como apresentado em (ZHAO; LI, 2012), são baseados em estratégias de planejamento que consideram instâncias sob demanda e *spot*, motivados pelos interesses das duas partes, usuário e provedor, envolvidas no processo de alocar recursos.

As alternativas atuais buscam viabilizar a implantação de nuvens trazendo o conceito de escalabilidade também para a implantação da infraestrutura. Uma estratégia é federar um conjunto de instituições para compartilhar não apenas os recursos de processamento, mas também os custos operacionais de manutenção. Outra alternativa consiste em realizar a implantação de uma infraestrutura híbrida, na qual a instituição mantém um conjunto de recursos mínimos e, na ocorrência de um pico de demanda, é buscado aprovisionamento de recursos em uma nuvem pública, em uma estratégia chamada de *cloud bursting*.

Neste trabalho, o modelo de nuvem considerado é o de uma nuvem composta por recursos de processamento oferecidos por instituições federadas, habilitado a realizar *cloud bursting* em uma nuvem pública. As infraestruturas de nuvens públicas já dispõe de equipamentos de *hardware* necessários para o atendimento exclusivo de aplicações HPC, porém, neste trabalho, assume-se o posicionamento de que as nuvens públicas exercem um papel complementar às infraestruturas disponibilizadas localmente pelas instituições que fazem parte da nuvem federada.

[1] Em um contexto de Tecnologia da Informação, é um dispositivo ou componente de dispositivo que é relativamente barato, amplamente disponível e, em alguns casos, intercambiável com outro *hardware* do mesmo tipo.

3 CONTABILIZAÇÃO DE CUSTOS

Em ambientes de nuvem, a precificação é o processo pelo qual se determina quanto um provedor de serviços receberá de um usuário final pelos serviços prestados (AL-ROOMI et al., 2013). O processo de precificação pode ser fixo, no qual o usuário final paga sempre o mesmo valor durante todo o tempo de uso dos serviços; dinâmico, em que os valores cobrados mudam com base em alterações de características; e dependente de mercado, que é quando o cliente é cobrado em função das condições de mercado em tempo real.

Este capítulo apresenta, a partir de um apanhado da literatura, elementos associados à viabilização econômica da exploração de um ambiente de nuvem computacional. Em um primeiro momento, na Seção 3.1, são apresentados quais elementos das aplicações caracterizam demandas de recursos de processamento e que, portanto, requerem aquisição destes recursos para que sejam efetivamente executadas. A Seção 3.2 caracteriza como podem se apresentar, em infraestruturas físicas reais, as opções de suporte de *hardware* às demandas de processamento requeridas pelas aplicações. As estratégias de precificação do uso de nuvens computacionais são discutidas na Seção 3.3.

3.1 Fontes de Custos

De um modo geral, os custos associados à utilização dos recursos computacionais disponibilizados por nuvens do tipo IaaS são difíceis de ser quantificados, fazendo com que seja necessária a utilização de modelos de decisão orientados a custos. Um dos modelos mais utilizados é o custo total de propriedade – TCO (*Total Cost of Ownership*) (STREBEL; STAGE, 2010).

O custo total de propriedade, inicialmente definido pela empresa Gartner, Inc.[1], é reconhecido como o método padrão para a análise financeira de recursos de Tecnologia da Informação (TI) e outros custos empresariais relacionados à TI (MIERITZ; KIRWIN, 2005). O modelo de TCO inclui aquisição, gerenciamento e suporte de *hardware* e *software*, comunicações, despesas do usuário final e os custos com tempo de inatividade, treinamento e outras perdas de produtividade. Para Filiopoulou e seu grupo (FILIOPOULOU et al.,

[1] `https://www.gartner.com/en` - empresa criada no final dos anos 1970, com atuação nos ramos de pesquisa, consultoria, eventos e prospecção do mercado de Tecnologia da Informação.

2015), a estimativa do custo total de propriedade, em particular, é um procedimento que fornece os meios para determinar o valor econômico total de um investimento, incluindo as despesas iniciais de capital (CAPEX, *capital expenditure*) e as despesas operacionais (OPEX, *operational expenditure*). No contexto de computação em nuvem, corresponde à estimativa de valores necessários para implementação e operação de uma infraestrutura.

Os benefícios do uso da abordagem de TCO estão na melhoria da comunicação entre cliente e provedor e na análise de todo o ciclo de vida dos artefatos de TI. Além disso, é possível analisar os custos ou componentes de custos individuais de um artefato de TI por meio de um esquema predefinido (WALTERBUSCH; MARTENS; TEUTEBERG, 2013). Como o objetivo do modelo TCO é fornecer uma visão abstrata e simplificada do "mundo real", em vez de incluir todos os custos relevantes na análise, a complexidade da realidade pode ser reduzida trabalhando com base em premissas e incluindo apenas um número limitado de fatores de custo cuidadosamente selecionados.

Métricas para os cálculos de custo e investimento em infraestruturas de nuvens computacionais necessitam de um modelo claro, juntamente com os fatores que influenciam estes custos. A Tabela 5 atribui fatores de custo f para cada tipo de custo t identificado. Os tipos de custo $t \in T$ e os fatores de custo $f \in F$ estão sujeitos aos conjuntos T e F:

$$T = \{dEst, ava, txIaaS, imp, sup, trein, manut, falha, bs\} \tag{5}$$

$$\begin{aligned} F = \{&dTempo, sCon, itDec, pComp, cArm, \\ &tEnt, tSda, tInt, nCon, dom, ssl, lic, txServ, \\ &pPort, cSup, rProb, tPrep, tPart, mInstr, perda\} \end{aligned} \tag{6}$$

Cada fator influenciador é atribuído a um tipo de custo e, em seguida, os elementos dos conjuntos T e F são agrupados em fórmulas. Cada fórmula é aplicada a um custo específico, possibilitando a obtenção dos valores para cada tipo de custo.

Tabela 5 – Tipos de custo e fatores de custo relacionados

Tipo de custo	Fatores de custo
Decisão estratégica, seleção de serviços de computação em nuvem e tipos de nuvem (*dEst*)	Despesas de tempo (*dTempo*), serviços de consultoria (*sCon*), informações para tomada de decisão (*itDec*).
Avaliação e seleção de prestador de serviços (*ava*)	Despesas de tempo (*dTempo*), serviços de consultoria (*sCon*), informações para tomada de decisão (*itDec*).
Taxa de serviço IaaS (*txIaaS*)	Poder de computação (*pComp*), capacidade de armazenamento (*cArm*), transferência de dados de entrada (*tEnt*), transferência de dados de saída (*tSda*), transferência de dados interna do provedor (*tInt*), número de consultas (*nCon*), domínio (*dom*), certificado SSL (*ssl*), licença (*lic*), taxa de serviço básica (*txServ*).
Implementação, configuração, integração e migração (*imp*)	Despesas de tempo (*dTempo*), processo de portabilidade (*pPort*).
Suporte (*sup*)	Despesas de tempo (*dTempo*), custos de suporte (*cSup*), resolução de problemas (*rProb*).
Treinamento inicial e permanente (*trein*)	Tempo de preparação dos funcionários internos (*tPrep*), tempo de participação dos funcionários internos (*tPart*), material de instruções (*mInstr*), serviços de consultoria externa (*sCon*).
Manutenção e modificação (*manut*)	Despesas de tempo (*dTempo*).
Falha no sistema (*falha*)	Perda por período (*perda*).
Backsourcing ou descarte (*bs*)	Despesas de tempo (*dTempo*), processo de portabilidade (*pPort*).

Fonte: Adaptado de (WALTERBUSCH; MARTENS; TEUTEBERG, 2013).

3.2 Fatores de Custos

Os custos relacionados à decisão estratégica, seleção de serviços de computação em nuvem e tipos de nuvem (*dEst*), juntamente com os custos de avaliação e seleção de prestadores de serviço (*ava*) são influenciados pelas despesas de tempo (*dTempo*) necessário para a tomada de decisão, as despesas com informações nas quais a decisão pode ser baseada (*itDec*), como, por exemplo literatura científica ou análises de mercado, bem como custos de serviços de consultoria externa (*sCon*). O custo total com as despesas de tempo resultam do tempo total gasto por todos os funcionários $\left(C^{dEst}_{dTempo} = \sum p^{dEst}_{dTempo,m} * a^{dEst}_{dTempo,m}\right)$, ou seja, é determinado pelo valor da hora de trabalho de cada empregado $\left(p^{dEst}_{dTempo,m}\right)$, multiplicado pelo tempo gasto $\left(a^{dEst}_{dTempo,m}\right)$, e somando os valores de todos os empregados (m) envolvidos. Custos com tomadas de decisão ocorrem em períodos $i < 1$ e, além disso, o custo total com a aquisição de informações (*itDec*) pode ser descrito como o somatório do total de preços (p) de todos os materiais adquiridos $\left(C^{dEst}_{itDec} = \sum p^{dEst}_{a}\right)$. Por fim, os custos

com serviços de consultoria $\left(C_{sCon}^{dEst}\right)$ são adicionados ao total e todos os gastos correspondentes aos fatores que influenciam o tipo de custo *dEst*, sumarizados pela fórmula $\left(C^{dEst} = C_{dTempo}^{dEst} + C_{sCon}^{dEst} + C_{itDec}^{dEst}\right)$. Para o processo de avaliação e seleção de prestadores de serviços (*ava*), os custos dependem da quantidade de tempo que os empregados dedicam a este processo (*dTempo*) e os custos de eventuais consultorias externas (*sCon*). Os cálculos para $\left(C_{dTempo}^{ava}\right)$ e $\left(C_{sCon}^{ava}\right)$ são análogos a $\left(C_{dTempo}^{dEst}\right)$ e $\left(C_{sCon}^{dEst}\right)$.

Para implementações de nuvem do tipo Infraestrutura como Serviço (IaaS), o tipo de custo que relaciona as taxas sobre os serviços (*txIaaS*) é composto por elementos como o custo com poder de computação (*pComp*), que pode ser calculado multiplicando o número de unidades de processamentos utilizadas $\left(a_{pComp,i}^{txIaaS}\right)$ por período *i*, pelo custo de uma unidade de processamento $\left(p_{pComp,i}^{txIaaS}\right)$. O preço varia de acordo com as características específicas do sistema, como, memória RAM, número de unidades de computação, capacidade de armazenamento, sistema operacional $\left(C_{pComp,i}^{txIaaS} = a_{pComp,i}^{txIaaS} * p_{pComp,i}^{txIaaS}\right)$ e o custo total deste fator *pComp* resulta do somatório de preços durante todos períodos *n* $\left(C_{pComp,i}^{txIaaS} = \sum_{i=1}^{n} C_{pComp,i}^{txIaaS}\right)$. Com a generalização do cálculo $\left(C_{f,i}^{t} = \sum_{i=1}^{n} a_{f,i}^{t} * p_{f,i}^{t}\right)$, que sumariza os custos unitários em função da quantidade consumida em um período de uso *i*, pode-se aplicar o mesmo raciocínio para os valores de capacidade de armazenamento (*cArm*), transferência de dados de entrada (*tEnt*), transferência de dados de saída (*tSda*) e a transferência de dados interna para outros serviços do mesmo provedor (*tInnt*) e o custo com o número de consultas (*nCon*). Os custos com manutenção de domínio para acesso Web (*dom*), certificados SSL (*ssl*), licenciamento de *software* (*lic*) e taxas básicas de serviços (*txServ*) podem ser determinados pela multiplicação do número de períodos utilizados *n* pelo respectivo preço p_f^t do fator de custo *f* do respectivo tipo de custo *t* $\left(C_f^t = n * p_f^t\right)$.

As despesas com o tempo (*dTempo*) necessário para cumprir as tarefas de implementação, configuração, integração e migração de serviços e dados influenciam o custo total tipo de custo (*imp*). Um fator de custo importante neste tipo de custo é o processo de portabilidade (*pPort*) dos dados do cliente para provedor de serviços. Conforme mencionado, os provedores cobram de seus clientes pela transferência de dados de entrada. Os custos da transferência inicial de dados para a nuvem para fins de migração do sistema pertencem a este tipo de custo. Eles são calculados multiplicando o volume de dados por unidade (ou seja, gigabyte) pelo preço de uma unidade. Alguns provedores oferecem serviços de envio de disco rígido para inserir os dados do cliente. No entanto, essa abordagem não se concentra no volume de dados, mas sim no número de discos rígidos e no tempo de carregamento de dados. O fator de custo de *pPort* não depende de mudanças temporais de preço porque se presume que o processo de portabilidade de dados pode ser concluído dentro de um período *t*: $\left(C_{pPort}^{imp} = a_{pPort}^{imp} * p_{pPort}^{imp}\right)$. As despesas de tempo C_{dTempo}^{imp} podem ser determinadas da mesma maneira que C_{dTempo}^{dEst}: $\left(C_{dTempo}^{imp} = \sum p_{dTempo,m}^{imp} * a_{dTempo,m}^{imp}\right)$.

O tipo de custo Suporte (*sup*) depende do custo dos serviços de atendimento via telefone,

e-mail, sistema de chamados ou *chat* durante todo ciclo de vida da infraestrutura de nuvem. Portanto, este tipo de custo depende do gasto de tempo (*dTempo*) necessário para interação com a equipe de suporte, bem como dos custos ocorridos. Alguns provedores de serviços cobram seus usuários com base no tempo necessário para a resolução de problemas e suporte. Os custos totais com suporte podem ser determinados pela multiplicação do preços de uma unidade pelo número total de unidades necessárias $\left(C^{sup}_{cSup} = p^{sup}_{cSup} * a^{sup}_{cSup}\right)$. Já os custos com resolução de problemas dependem do número de unidades consumidas e o preço por unidade $\left(C^{sup}_{rProb} = p^{sup}_{rProb} * a^{sup}_{rProb}\right)$.

Os custos totais do tipo de custo "treinamento inicial e permanente"(*trein*) podem ser subdivididos em treinamento interno, no qual os próprios colaborados atuam como treinadores, e treinamento externo, no qual são necessários treinadores externos à empresa. Os custos de um treinamento interno dependem da quantidade de tempo de preparação investido por um ou mais empregados (*tPrep*), o tempo de participação dos funcionários internos (*tPart*) e os custos com material para os treinamentos (*mInstr*):

$$
\begin{aligned}
C^{trein}_{int} &= \sum C^{trein}_{tPrep,m} + \sum C^{trein}_{tPart,m} + C^{trein}_{mInstr,m} && (7)\\
&= \sum \left(p^{trein}_{tPrep,m} * a^{trein}_{tPrep,m}\right) && (8)\\
&+ \sum \left(p^{trein}_{tPart,m} * a^{trein}_{tPart,m}\right) + C^{trein}_{mInstr} && (9)
\end{aligned}
$$

O total dos custos de um treinamento externo pode se calculado pela adição dos custos com serviços de consultoria que organizam o treinamento (*sCon*), a quantidade de tempo que os empregados investem na participação do treinamento (*tPart*) e os custos com materiais de treinamento (*mInstr*):

$$
\begin{aligned}
C^{trein}_{ext} &= \sum C^{trein}_{sCon} + \sum C^{trein}_{tPart} + C^{trein}_{mInstr,m} && (10)\\
&= \sum \left(p^{trein}_{sCon} * a^{trein}_{sCon}\right) && (11)\\
&+ \sum \left(p^{trein}_{tPart,m} * a^{trein}_{tPart,m}\right) + C^{trein}_{mInstr} && (12)
\end{aligned}
$$

Custos com manutenção e modificação (*manut*) dependem das despesas com tempo gasto (*dTempo*) em manutenções gerais e em modificações feitas para implementação de serviços $\left(C^{manut}_{dTempo}\right)$. O cálculo das despesas de tempo para uma respectiva tarefa de manutenção é baseada na fórmula do tipo de custo *dEst*: $\left(c^{manut}_{dTempo} = \sum p^{dEst}_{dTempo,m} * a^{dEst}_{dTempo,m}\right)$.

Custos totais de uma falha de sistema precisam ser declarados para cada empresa individualmente. Os possíveis fatores de custo são, por exemplo, perda de tempo de trabalho produtivo, penalidades contratuais por atrasos ou danos à reputação da empresa, que são difíceis de mensurar. Assim, apenas destaca-se uma fórmula geral que representa a perda

por período *i*:

$$C_{perda}^{falha} = \sum_{i=1}^{n} a_{perda,i}^{falha} * p_{perda,i}^{falha} \tag{13}$$

O processo de descarte, ou *backsourcing*, de um sistema envolve despesas de tempo (*dTempo*) e processo de portabilidade (*pPort*). No entanto, os custos com a portabilidade dos dados entre nuvens, ou para um sistema diferente, fazem parte do TCO de um novo serviço e não do TCO do sistema no qual os dados estão sendo retirados. Os custos podem ser determinados da mesma maneira que os custos do processo de portabilidade dos dados para a nuvem $\left(C_{pPort}^{bs} = a_{pPort}^{bs} * p_{pPort}^{bs}\right)$ e também dependem do gasto de tempo necessário para a decisão estratégica necessária:

$$C_{dTempo}^{bs} = \sum p_{dTempo,m}^{bs} * a_{dTempo,m}^{bs} \tag{14}$$

3.3 Custo Global

Em um ambiente de nuvem pública do tipo IaaS, o custo total de propriedade de um serviço de computação em nuvem é igual à soma de todos os tipos de custos envolvidos e pode ser definido como:

$$TCO_{Nuvem} = \sum C^t \text{ onde } t \in T \tag{15}$$

O valor total de um tipo de custo t é igual à soma de todos os fatores de custo f envolvidos, conforme segue: $C^t = \sum C_f^t$ onde $t \in T, f \in F$.

Neste modelo de TCO é considerado o período total de tempo no qual os serviços de nuvem foram ou serão utilizados. Este período é subdividido em vários períodos menores i, com duração de um mês, geralmente, predefinido pelo provedor de serviços. Assim, o período total é composto por n períodos menores, de acordo com $C_f^t = \sum_i^n C_{f,i}^t$ onde $i = \{1, \ldots, n\}, t \in T, f \in F$. As variáveis $a_{f,i}^t$ e $p_{f,i}^t$ são utilizadas, respectivamente, para representar a quantidade consumida ou necessária no período i e caracterizar os custos ou preços unitários na fórmula $C_{f,i}^t = a_{f,i}^t * p_{f,i}^t$.

Ao contrário de serviços entregues por meio de nuvens públicas, nas estruturas de nuvens privadas os usuários e provedores fazem parte da mesma organização ou os serviços são prestados, de modo exclusivo, por terceiros. No primeiro cenário, os custos envolvidos incluem, por exemplo, licenciamento de *softwares* implementados bem como a infraestrutura de TI que deve ser fornecida pela organização. Já no caso de entrega de serviços por provedor exclusivo, o fornecimento é semelhante a uma nuvem pública IaaS, no modo em que o usuário obtém os recursos de um provedor. Apesar disso, o provedor não gerencia dados em uma estrutura pública, mas, sim, em uma nuvem privada exclusiva.

Finalmente, em soluções de nuvens híbridas, que agregam as características de nuvens

públicas e privadas, as despesas totais são iguais aos custos totais, ou pelo menos proporcionais, envolvidos em cada solução individual. Além disso, despesas com o processo de agregação de soluções (públicas e privadas) devem ser consideradas na composição dos custos e investimentos.

4 TRABALHOS RELACIONADOS E ESTADO DA ARTE

Neste capítulo é apresentada a Revisão Sistemática da Literatura (RSL), juntamente com os trabalhos relacionados ao tema deste estudo. A Seção 4.2 sintetiza os resultados da RSL, respondendo as questões de pesquisa a partir dos trabalhos restantes do processo de revisão. Desta síntese, aponta-se oportunidades de pesquisa na Seção 4.3. A Seção 4.4 encerra o capítulo apresentando uma discussão sobre os trabalhos relacionados selecionados a partir da RSL realizada. Esta RSL encontra-se publicada na Revista de Informática Teórica e Aplicada (RITA), no ano de 2020 (SANTOS; CAVALHEIRO, 2020).

4.1 Revisão Sistemática

No processo de pesquisa e seleção dos trabalhos relacionados ao tema deste estudo, foi realizada uma Revisão Sistemática da Literatura, desenvolvendo seus três grandes estágios: planejamento, condução e relatório da revisão (XIAO; WATSON, 2017). Na fase de planejamento, é identificada a necessidade de uma revisão, com a especificação de questões de pesquisa e desenvolvimento de um protocolo de revisão. Na condução da revisão, são identificados e selecionados os estudos primários, extraídos os dados, analisados e sintetizados. Por fim, no relatório da revisão sistemática, são divulgadas as descobertas e discutidos os trabalhos restantes da pesquisa.

4.1.1 Planejamento da RSL

Para o desenvolvimento da RSL no tema proposto, durante o estágio de planejamento, foram definidas algumas Questões de Pesquisa (QPs) com o objetivo de nortear e fundamentar o estudo. As Questões de Pesquisa são:

QP 1: Qual o impacto econômico nas decisões de implantação de aplicações de alto desempenho em nuvens computacionais?

Esta QP busca identificar como instituições (acadêmicas ou comerciais) podem se beneficiar do uso de nuvens computacionais para execução de aplicações que demandem alto poder de processamento. Esta questão considera os custos financeiros envolvidos.

QP 2: Como os usuários podem identificar qual a melhor configuração de nuvem,

seja pública ou privada, para executar suas aplicações?

Nesta questão, o objetivo é prover subsídios para que os usuários de nuvem possam determinar qual opção de configuração, oferecida pelos provedores de serviço, mais se adapta às necessidades de suas demandas de processamento. Neste aspecto deve ser levado em conta que existe perda financeira à medida que houver super ou subdimensionamento de recursos.

QP 3: Quais modelos de custo financeiro são empregados na execução de aplicações de alto desempenho em nuvens?

Com esta questão, busca-se identificar possíveis modelos de custo utilizados por instituições para definir as demandas de infraestruturas de nuvem para execuções de aplicações com demanda de processamento de alto desempenho.

QP 4: Como identificar eventuais perdas financeiras em função de imprecisão no dimensionamento de infraestruturas de nuvens para aplicações com demandas de HPC?

O objetivo desta questão é identificar se haverá perda financeira em função do dimensionamento inadequado da infraestrutura para a demanda do usuário. Neste caso, pode ocorrer superdimensionamento, sendo dispendidos mais recursos na infraestrutura que o necessário para atender à demanda, ou subdimensionamento, quando a execução da aplicação pode atrasar devido à insuficiência de recursos computacionais para atender à demanda. No primeiro caso, a perda está relacionada ao passivo em recursos instalados, e as consequentes despesas em manutenção. No segundo, a perda está relacionada ao baixo índice de produção e a consequente perda de lucros.

De posse destas Questões de Pesquisa, foi realizada uma busca exploratória à procura de trabalhos que abordassem os principais temas contidos nas perguntas. Com isso foi possível identificar como a comunidade científica faz referência aos temas e também extrair palavras-chave utilizadas para composição da *string* de busca utilizada na presente RSL, conforme documentado na sequência (Seção 4.1.2).

4.1.2 Condução da RSL

Para a condução desta RSL foram acessadas diversas bases de indexação de trabalhos, amplamente utilizadas por pesquisadores. Ao todo foram selecionadas cinco bases: *i*) ACM Digital Library; *ii*) IEEE Digital Library; *iii*) Science@Direct; *iv*) Scopus; e *v*) Web of Science. Tais bases foram escolhidas devido a sua importância e por serem repositórios digitais que oferecem acesso eletrônico à maioria dos periódicos e artigos de conferências publicados na área da Ciência da Computação. (OKOLI; SCHABRAM, 2010).

Em seguida, um conjunto de termos de pesquisa foi identificado para compor a *string* de busca com a qual será possível extrair da literatura trabalhos relacionados com o tema abordado. Os termos selecionados visam identificar, na literatura, trabalhos envolvendo ambientes de nuvens (*(cloud OR "cloud computing")*) e computação de alto desempenho

(*(hpc OR "high performance computing")*). A esta *string* foram associados termos referentes a possíveis modelos de custo e preços praticados em situações que contemplem a execução de aplicações de HPC em ambientes de nuvem, analisando sua viabilidade financeira (*("cost model" OR "cost efficiency" OR "cost analysis" OR "economic analysis" OR "monetary cost" OR "billing model" OR "price efficiency" OR investment OR pricing OR price)*). A *string* de busca concebida é:

> ***((cloud OR "cloud computing") AND (hpc OR "high performance computing") AND ("cost model" OR "cost efficiency" OR "cost analysis" OR "economic analysis" OR "monetary cost" OR "billing model" OR "price efficiency" OR investment OR pricing OR price))***

De posse da *string* de busca, foram realizadas pesquisas nas bases selecionadas. Uma das características disponíveis nas bases de indexação utilizadas é a possibilidade de exportação dos resultados para o formato *BibTeX*[1]. Os resultados alcançados por meio da execução de buscas nas bases foram extraídos para arquivos (*.bib*) e, posteriormente, importados na ferramenta Parsifal (PARSIFAL, 2014) que auxilia na análise dos resultados, dando prosseguimento ao processo de revisão.

O processo de busca por artigos desta RSL se desenvolveu em quatro fases, cada uma com critérios de exclusão associados. A primeira fase consiste na aplicação da *string* de busca nas bases de indexação. Na sequência, em cada uma das demais fases, serão aplicados filtros nos resultados em conformidade com os critérios de exclusão da Tabela 6.

Tabela 6 – Critérios de exclusão

Fase	ID	Critério de exclusão
Fase 1	-	Aplicação da *string* de busca nas bases de indexação.
Fase 2	2.1	Trabalhos anteriores ao ano de 2010.
Fase 3	3.1	Trabalhos que não contêm a *string* de busca no título, resumo ou palavras-chave.
Fase 4	4.1	Trabalhos duplicados.
	4.2	Trabalhos que não estão publicados em conferências ou periódicos.
	4.3	Trabalhos nos quais título e resumo não abordam o tema de estudo.
	4.4	Trabalhos que não apresentem uma avaliação de custos de infraestrutura.
	4.5	Trabalhos que não relacionem nuvens e execução de aplicações HPC.
	4.6	Trabalhos sem acesso ao texto completo.
	4.7	Trabalhos não classificados pela avaliação de qualidade.
	4.8	Trabalhos que apresentem pequenas modificações de estudos do mesmo grupo.

Na segunda fase é aplicado o critério 2.1, na qual fica explícito que trabalhos anteriores ao ano de 2010 encontram-se desatualizados ou, caso tenham sido continuados, novos resultados devem estar contemplados em publicações mais atuais.

[1] Ferramenta para formatação de bibliografias utilizada em documentos LaTeX.

Na terceira fase da revisão foi aplicado o critério de exclusão 3.1, retirando os trabalhos que não possuem os termos da *string* de busca no título, resumo ou palavras-chave. Este critério visa reduzir o número de resultados falso-positivos da pesquisa realizada na busca inicial (fase 1).

Já na quarta e última fase, os demais critérios de exclusão são aplicados. Esta fase requer uma análise mais detalhada dos textos, com base nos critérios restantes. O objetivo da fase quatro é selecionar somente os trabalhos que abordam o tema de pesquisa relacionado, apresentando avaliações de custos de infraestruturas locais e de nuvem, bem como relacionando a execução de aplicações com demandas de alto desempenho em ambientes de nuvem.

Na fase 4, o critério 4.1 realiza a busca por trabalhos duplicados, removendo-os da seleção, enquanto o critério 4.2 procura por estudos que não estão publicados em conferências ou periódicos. Para o critério de exclusão 4.3, foi realizada a leitura dos títulos e resumos a fim de retirar artigos que não abordam o tema de pesquisa objeto deste estudo.

Neste ponto, para aplicação dos critérios 4.4 e 4.5, que buscam, respectivamente, por trabalhos que não apresentam uma avaliação de custos de infraestrutura e não relacionam nuvens com a execução de aplicações HPC, foi necessária uma leitura completa dos trabalhos. Durante o processo de leitura, não foi possível obter acesso aos textos completos de alguns trabalhos que, por consequência, foram excluídos pelo critério 4.6.

Para avaliação dos trabalhos selecionados até este momento da RSL, são aplicadas algumas questões de qualidade (critério de exclusão 4.7), conforme segue:

- Os objetivos da pesquisa estão claramente especificados?
- O trabalho considera a satisfação do usuário?
- O modelo de custo financeiro considera o desempenho das aplicações?
- O trabalho considera a satisfação do provedor de serviços?
- O desempenho da execução de aplicações é considerado?
- O trabalho apresenta resultados com relação aos custos?

Cada questão de qualidade possui três opções de respostas: “sim”, “parcialmente” ou “não”, com valores atribuídos, respectivamente, “1”, “0.5” e “0”. Os trabalhos podem ser pontuados com, no máximo, seis pontos e no mínimo, zero. Foi definida a pontuação “3.5” como ponto de corte, ou seja, pontos mínimos para ser considerado aceito.

Ao final da fase 4, o último critério de exclusão, 4.8, foi aplicado aos textos desta RSL com o objetivo de identificar trabalhos que tragam pequenas modificações de trabalhos anteriores de um mesmo grupo de pesquisa.

A Tabela 7 apresenta, de modo geral, os quantitativos de trabalhos suprimidos em cada fase da revisão, de acordo com os critérios de exclusão definidos. Conforme os valores demonstrados na tabela, percebe-se que o critério que mais excluiu trabalhos pertence à fase 3, sendo o 3.1, o qual exclui trabalhos que não contêm a *string* de busca no título, *abstract* ou palavras-chave dos materiais analisados, totalizando 4593 trabalhos.

Tabela 7 – Quantitativo de trabalhos rejeitados em cada critério de exclusão

Base	**Critérios de exclusão**										
	Fase 1	**Fase 2**	**Fase 3**	**Fase 4**							
	-	**2.1**	**3.1**	**4.1**	**4.2**	**4.3**	**4.4**	**4.5**	**4.6**	**4.7**	**4.8**
ACM Digital Library	134	3	62	47	9	8	0	1	1	3	0
IEEE Digital Library	509	11	367	48	2	62	6	9	0	2	0
Science@Direct	328	0	323	2	0	1	1	0	0	1	0
Scopus	3918	6	3761	23	19	58	14	4	4	12	7
Web of Science	168	1	80	76	1	8	2	0	0	0	0
Total	5057	21	4593	196	31	137	23	14	5	18	7
Trabalhos restantes	5057	5036	443	247	216	79	56	42	37	19	12

Por fim, esta RSL selecionou 12 trabalhos que versam sobre a análise de investimentos em infraestruturas de nuvem, levando em consideração modelos de custo para execução de aplicações HPC. A Seção 4.1.3 trará um apanhado sobre os artigos, apresentando-os com maiores detalhes.

Tabela 8 – Trabalhos selecionados na RSL

ID	**Título**	**Citação**
T01	High Performance Computing in the cloud: Deployment, performance and cost efficiency	(ROLOFF et al., 2012)
T02	A comparative study of high-performance computing on the cloud	(MARATHE et al., 2013)
T03	A performance/cost model for a CUDA drug discovery application on physical and public cloud infrastructures	(GUERRERO et al., 2014)
T04	Cost-Optimized Resource Provision for Cloud Applications	(SHEN et al., 2014)
T05	Evaluating and Improving the Performance and Scheduling of HPC Applications in Cloud	(GUPTA et al., 2016)
T06	Price efficiency in High Performance Computing on Amazon Elastic Compute Cloud provider in Compute Optimize packages	(PRUKKANTRAGORN; TIENTANOPAJAI, 2016)
T07	Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources	(ARABNEJAD; BUBENDORFER; NG, 2017)
T08	Cost Analysis Comparing HPC Public Versus Private Cloud Computing	(DREHER et al., 2017)
T09	HPC Application Performance and Cost Efficiency in the Cloud	(ROLOFF et al., 2017)
T10	Understanding the Performance and Potential of Cloud Computing for Scientific Applications	(SADOOGHI et al., 2017)
T11	Amazon Elastic Compute Cloud (EC2) versus In-House HPC Platform: A Cost Analysis	(EMERAS et al., 2019)
T12	Exploring Instance Heterogeneity in Public Cloud Providers for HPC Applications	(ROLOFF et al., 2019)

4.1.3 Relatório da RSL

O último estágio desta RSL traz a etapa de extração de dados dos trabalhos selecionados pela revisão. Tal processo se dará com a análise dos artigos, buscando recuperar os conteúdos referentes ao problema abordado, solução proposta, método utilizado para validação da proposta e trabalhos futuros. Na sequência, cada trabalho será apresentado em uma seção específica.

4.1.3.1 High Performance Computing in the cloud: Deployment, performance and cost efficiency (ROLOFF et al., 2012)

Problema: A execução de aplicações de alto desempenho (HPC) em nuvem alcançou um lugar de destaque e se tornou um importante tópico de pesquisa científica e para a indústria. No entanto, o foco dos provedores de serviços em nuvem é a entrega de conteúdo e não HPC. Por esse motivo, faltam pesquisas direcionadas à implantação de aplicações HPC em infraestruturas de nuvem.

Proposta: Realizar uma avaliação abrangente de três aspectos importantes da execução de aplicações HPC em nuvem: implantação, desempenho e eficiência de custos. Para a avaliação foram utilizados conjuntos de *benchmarks* conhecidos, como o *NAS Parallel Benchmarks* (NPB), e as execuções ocorreram em três provedores de nuvem diferentes: Amazon Elastic Cloud Compute (EC2), Microsoft Windows Azure e Rackspace.

Validação: Os resultados das avaliações foram comparados com um *cluster* real, com características semelhantes às instâncias das nuvens utilizadas, para analisar diferenças de eficácia de custos e desempenho e, também, prover argumentos para discussões sobre quando aplicações de alto desempenho fazem sentido em ambientes de nuvem e para quais casos de uso.

Trabalhos Futuros: Migrar aplicações HPC completas para ambientes de nuvem e estender a métrica de custo e eficiência para cobrir mais fatores e ser mais flexível.

4.1.3.2 A comparative study of high-performance computing on the cloud (MARATHE et al., 2013)

Problema: A popularidade da plataforma em nuvem EC2 da Amazon aumentou nos últimos anos. No entanto, muitos usuários de computação de alto desempenho (HPC) consideram que os *clusters* dedicados de alto desempenho, normalmente encontrados em grandes centros de computação, são muito superiores ao EC2, devido à significativa sobrecarga de comunicação deste último.

Proposta: Examinar, de forma inovadora, diferenças ao comparar os *clusters* Amazon EC2 de ponta com os *clusters* HPC tradicionais, mas com um esquema de avaliação mais geral. Primeiro, comparar o EC2 a cinco *clusters* do *Lawrence Livermore National Laboratory* (LLNL) com base no tempo total de resposta para um conjunto típico de *benchmarks* de HPC em diferentes escalas; para o tempo de espera na fila nos *clusters* HPC, usou-se uma distribuição desenvolvida a partir de simulações com rastreamentos reais. Segundo, para permitir uma comparação do custo total de execução, foi desenvolvido um modelo econômico para precificar *clusters* do LLNL, supondo que eles sejam oferecidos como recursos de nuvem a preços por hora do nó.

Validação: Para estimar o custo foi desenvolvido um modelo de preços, relativo aos preços por hora por nó do EC2, para que fosse possível estimar os mesmos valores para o *cluster* do LLNL.

Trabalhos Futuros: Utilizar tempo de resposta e custo de para desenvolver ferramentas e técnicas que direcionam os usuários de diversos conjuntos de aplicativos, dadas restrições específicas, ao *cluster* mais apropriado.

4.1.3.3 A performance/cost model for a CUDA drug discovery application on physical and public cloud infrastructures (GUERRERO et al., 2014)

Problema: Na pesquisa clínica, é crucial determinar a segurança e a eficácia dos medicamentos atuais e acelerar os achados na pesquisa básica, como a descoberta de novos compostos ativos, em resultados significativos para a saúde. Ambos os objetivos requerem o processamento de grandes conjuntos de dados de estruturas de proteínas disponíveis em bancos de dados biológicos.

Proposta: É proposto um modelo de desempenho/custo para a aplicação BINDSURF, permitindo que o usuário decida qual infraestrutura, local ou de um conhecido provedor de nuvem pública, é ideal para um determinado tipo e tamanho de problema. A execução de uma aplicação com uso intensivo de GPU, como o BINDSURF, pode sobrecarregar o orçamento de uma instituição ao processar grandes quantidades de dados. Quanto maior o número de recursos físicos computacionais ou maior o tempo de execução para esses recursos, mais o custo total é aumentado, mesmo para infraestruturas locais não utilizadas.

Validação: Criação de um modelo de custo e desempenho para bioinformática utilizando o algoritmo BINDSURF, para obter o melhor desempenho de execução durante o tempo e otimizar os custos.

Trabalhos Futuros: Portar o BINDSURF para OpenCL, permitindo que ele seja executado em uma variedade maior de sistemas computacionais heterogêneos, como CPUs com vários núcleos. Isso permitirá que um número maior e mais barato de tipos de instâncias de provedores de nuvem pública sejam usados para um modelo mais abrangente de desempenho e custo.

4.1.3.4 Cost-Optimized Resource Provision for Cloud Applications (SHEN et al., 2014)

Problema: Com um número crescente de provedores de serviços em nuvem oferecendo locação de recursos virtuais, os provedores de aplicações têm mais opções quando precisam de recursos. Mas alcançar a solução de recursos mais otimizada em custo ainda é um desafio.

Proposta: Uma abordagem para auxiliar os usuários a calcular a quantidade otimizada de recursos virtuais com base na carga de trabalho prevista e resolver soluções de fornecimento de recursos, o que inclui instâncias de máquinas virtuais de diferentes tipos e preços. Os SLAs publicados pelos provedores são atendidos o máximo possível.

Validação: Para estimar a relação entre a carga de trabalho prevista e a quantidade de máquinas virtuais, foram realizados experimentos de projeção, simulando as instâncias do tipo micro do Amazon EC2. Utilizando as políticas de preços do EC2, foi obtida uma solução de provisão de economia de custos para fornecedores de aplicações.

Trabalhos Futuros: Implementar a abordagem proposta como uma estrutura de tempo de execução dinâmico para aplicações em nuvem. Isso envolverá mais modelos de previsão e técnicas de aprendizado de máquina. Além disso, aprimorar o algoritmo de provisionamento para suportar políticas de preços de mais provedores de nuvem existentes.

4.1.3.5 Evaluating and Improving the Performance and Scheduling of HPC Applications in Cloud (GUPTA et al., 2016)

Problema: A computação em nuvem está surgindo como uma alternativa promissora aos supercomputadores para algumas aplicações de computação de alto desempenho (HPC). Com a nuvem como uma opção de implantação adicional, os usuários e fornecedores de HPC enfrentam os desafios de lidar com recursos altamente heterogêneos, onde a variabilidade se estende por uma ampla gama de configurações de processadores, interconexões, ambientes de virtualização e modelos de preços.

Proposta: Uma avaliação detalhada e abrangente de desempenho e custo de execução de um conjunto de aplicações HPC em uma variedade de plataformas, variando desde supercomputadores às nuvens computacionais. Este estudo permite uma visão holística que busca responder ao questionamento por que e quem deve escolher uma nuvem para execução de aplicações HPC e quais aplicações e como as nuvens devem ser utilizadas para HPC. Investigar os aspectos econômicos da execução em nuvem e discutir por que é desafiador ou gratificante para os provedores de nuvem operar negócios para a HPC em comparação com as aplicações em nuvem tradicionais.

Validação: Avaliação de desempenho e gargalos de aplicações HPC em estruturas de supercomputadores, *clusters* e nuvens (privadas e públicas). Com o uso de *benchmarks* executados no mesmo *hardware*, com e sem uso de *hypervisors*, foi possível uma análise detalhada do impacto do uso de virtualização para HPC. Para a tarefa de investigar a coexistência de várias plataformas foi utilizado o simulador CloudSim.

Trabalhos Futuros: Considerar outros fatores no escalonamento de várias plataformas: qualidade de serviço (QoS), prazos, prioridades e segurança. Além disso, pesquisas futuras são necessárias no que diz respeito ao preço das nuvens em ambientes de várias plataformas. Outro tema promissor é a avaliação e caracterização de aplicações com paralelismo irregular e conjuntos de dados dinâmicos.

4.1.3.6 Price efficiency in High Performance Computing on Amazon Elastic Compute Cloud provider in Compute Optimize packages (PRUKKANTRAGORN; TIENTANOPAJAI, 2016)

Problema: Atualmente, a computação de alto desempenho (HPC) é usada em muitas pesquisas e trabalha para calcular ou processar dados. Neste sentido, em relação à computação de alto desempenho (HPC), o trabalho pretende dar subsídios aos usuários para que possa ser determinado qual o pacote otimizado de serviços, oferecido pelo provedor, é o mais adequado para uma dada aplicação HPC.

Proposta: Investigar a eficiência de preços que pode beneficiar o cliente na escolha do pacote de otimizações oferecido pelo fornecedor de serviços de nuvem. Analisar a relação entre preços, tempos de execução e tamanhos de problemas ou cargas de trabalho do HPL no pacote otimizado para computação do Amazon EC2.

Validação: Avaliação de todas as instâncias do pacote otimizado para computação do Amazon EC2, usando o *benchmark* HPL, com base na carga de trabalho ou no tamanho do problema de entrada. Para eficiência de preço, a relação de tamanho do

problema, tempo em HPL e preço foi analisada para obter uma instância adequada para o uso da computação de alto desempenho.

Trabalhos Futuros: Não apresenta trabalhos futuros.

4.1.3.7 Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources

(ARABNEJAD; BUBENDORFER; NG, 2017)

Problema: Uma nuvem permite que pesquisadores e instituições provisionem recursos de computação apenas quando necessário e escalem conforme necessário. No entanto, ainda existem obstáculos técnicos significativos associados à obtenção de desempenho de execução suficiente e limitação do custo financeiro. Os esforços se concentram no problema de agendamento de cargas de trabalho científicas com restrições de prazo em recursos de nuvem provisionados dinamicamente, enquanto reduz o custo da computação.

Proposta: São apresentados dois algoritmos, o *Proportional Deadline Constrained* (PDC) e o *Deadline Constrained Critical Path* (DCCP) que abordam o problema de agendamento de fluxo de trabalho nos recursos de nuvem provisionados dinamicamente. Esses algoritmos são adicionalmente estendidos para refinar sua operação na priorização de tarefas e preenchimento, respectivamente.

Validação: Os algoritmos foram avaliados, por meio de simulação, com o uso do CloudSim, que apresenta recursos de nuvem provisionados dinamicamente e um modelo de pagamento por uso derivado do modelo de precificação EC2 da Amazon. As simulações foram realizadas usando cinco fluxos de trabalho científicos: Montage, SIPHT, LIGO, Cybershake e Epigenomics. Cada fluxo consistiu em 1000 tarefas e foram obtidos a partir do gerador de fluxos de trabalho Pegasus.

Trabalhos Futuros: Investigar o impacto da estrutura de fluxo de trabalho, procurando uma medida de simetria a fim de considerar como isso pode ser incorporado nas decisões de escalonamento.

4.1.3.8 Cost Analysis Comparing HPC Public Versus Private Cloud Computing (DREHER et al., 2017)

Problema: Nos últimos anos, houve um rápido aumento no número e tipo de configurações de *hardware* de computação em nuvem pública e opções de preços oferecidas aos clientes. Além disso, os provedores de nuvem pública também expandiram o número e o tipo de opções de armazenamento e estabeleceram preços incrementais

para armazenamento e transmissão em rede de dados de saída da instalação em nuvem. Tal cenário prejudica a análise para determinar a opção mais econômica em uma migração de aplicações de uso geral para a nuvem.

Proposta: Investigar se a análise econômica para mover aplicações de uso geral para uma nuvem pública pode ser estendida para execuções do tipo HPC com uso intensivo de computação.

Validação: Uma comparação de custos com uma determinada configuração de *hardware* HPC é estabelecida para determinar sob quais condições uma nuvem pública e não uma nuvem privada será mais econômica em cálculos, armazenamento e transferências de dados de rede para aplicativos do tipo HPC.

Trabalhos Futuros: Não apresenta trabalhos futuros.

4.1.3.9 HPC Application Performance and Cost Efficiency in the Cloud (ROLOFF et al., 2017)

Problema: Nos últimos anos, várias abordagens foram introduzidas para o uso eficiente de nuvens para a execução de aplicações HPC. No entanto, faltam pesquisas sobre oportunidades e desvantagens do uso de nuvens públicas como ambiente eficiente para HPC.

Proposta: Identificar as instâncias de máquina virtual em nuvens públicas disponíveis que possam ser adequadas para aplicações HPC, tanto em termos de desempenho quanto de eficiência de custos. Também avaliar que tipo de aplicação pode se beneficiar da execução na nuvem. Para isso, é preciso analisar as características das instâncias, levando em consideração os aspectos relevantes para HPC. Também é necessária uma análise da eficiência de custos usando os *benchmarks* tradicionais de HPC.

Validação: Foi realizada uma extensa avaliação dos dois maiores provedores de computação em nuvem, Amazon EC2 e Microsoft Azure, considerando comunicação de rede, processamento e desempenho de memória.

Trabalhos Futuros: Adicionar métricas de desempenho para dispositivos de entrada e saída à avaliação, pois tem sido uma área com bastante desenvolvimento na nuvem nos últimos anos. Também avaliar ambientes de nuvem que possuem aceleradores, como GPUs.

4.1.3.10 *Understanding the Performance and Potential of Cloud Computing for Scientific Applications (SADOOGHI et al., 2017)*

Problema: As aplicações científicas geralmente exigem recursos significativos, no entanto, nem todos os cientistas têm acesso a sistemas de computação de ponta suficientes. A computação em nuvem chamou a atenção dos cientistas como um recurso competitivo para execução de aplicações HPC, a um custo potencialmente mais baixo. Mas, como uma infraestrutura diferente, não está claro se as nuvens são capazes de executar aplicações científicas com um desempenho razoável por dinheiro gasto.

Proposta: Avaliar a capacidade de uma nuvem em ter um bom desempenho, bem como avaliar o custo da nuvem em termos de desempenho bruto e desempenho de aplicações científicas. Além disso, são avaliados outros serviços, incluindo S3, EBS e DynamoDB, a fim de avaliar as habilidades daqueles a serem utilizados por aplicações e estruturas científicas. Também são avaliadas aplicações de computação científica reais por meio do sistema de *scripts* paralelos em escala Swift.

Validação: Foi verificado o desempenho bruto do EC2 com a execução de micro *benchmarks* para medir o desempenho bruto de diferentes tipos de instância, em comparação com o pico de desempenho teórico reivindicado pelo provedor de recursos. Também se comparou o desempenho real com um sistema não virtualizado típico para entender melhor o efeito da virtualização.

Trabalhos Futuros: Não apresenta trabalhos futuros.

4.1.3.11 *Amazon Elastic Compute Cloud (EC2) versus In-House HPC Platform: A Cost Analysis (EMERAS et al., 2019)*

Problema: Embora os centros de computação de alto desempenho (HPC) evoluam continuamente para fornecer mais poder de computação a seus usuários, observa-se um desejo de convergência entre plataformas de computação em nuvem (CC) e computação de alto desempenho (HPC). Excluindo-se o ponto de vista do desempenho, em que muitos estudos destacam uma sobrecarga induzida pela camada de virtualização no núcleo dos *middlewares* em nuvem ao executar uma carga de trabalho HPC, a relação custo-benefício real costuma ser deixada de lado com o desejo de que as instâncias oferecidas pelos provedores de nuvem sejam competitivas do ponto de vista dos custos.

Proposta: Analisar os elementos que compõe o custo total de propriedade (TCO) de uma instalação interna de HPC, operada desde 2007. A partir do modelo de TCO,

comparar os custos necessários para executar a mesma plataforma (e a mesma carga de trabalho) em uma oferta competitiva de nuvem do tipo IaaS. Uma abordagem tripla para comparação de preços é utilizada. Primeiro, é proposto um modelo de preço-desempenho teórico baseado no estudo das instâncias da Amazon EC2. Em seguida, com base na análise de custo total de propriedade da plataforma HPC, é feita uma comparação horária de preços entre o *cluster* interno e as instâncias equivalentes da EC2. Por fim, com base no *benchmarking* experimental no *cluster* local e nas instâncias da nuvem, foi proposta uma atualização do antigo modelo teórico de preços para refletir o desempenho real do sistema.

Validação: Comparação entre as instâncias EC2 e os nós do *cluster* local. A partir dessa comparação, o modelo de custo é refinado, integrando a pontuação de referência real na equação do modelo. Para esse fim, foi utilizado o *benchmark High Performance Conjugate Gradients* (HPCG).

Trabalhos Futuros: Estender a análise sobre instâncias do tipo *spot* que permitem fazer lances pelo preço dos recursos. Isso oferece uma chance de uma melhor economia de custos nos preços das taxas de instância e, portanto, pode indicar outras classes de recursos HPC para os quais a opção de aluguel faz sentido. Integrar o custo real de uma nova sala para servidores HPC à análise TCO, a partir do monitoramento dos custos com construção e implementação. Isso também permitirá atualizar o modelo com relação ao custo das tecnologias de ponta para HPC, como, resfriamento direto líquido e interconexões Infiniband.

4.1.3.12 Exploring Instance Heterogeneity in Public Cloud Providers for HPC Applications (ROLOFF et al., 2019)

Problema: A execução de grandes aplicações paralelas (como as de HPC) tornou-se um aspecto importante da computação em nuvem nos últimos anos. Com estas execuções, os usuários podem se beneficiar de custos iniciais mais baixos, maior flexibilidade e atualizações de *hardware* mais rápidas em comparação com os *clusters* tradicionais. No entanto, o desempenho bruto e a eficiência de custos para uso a longo prazo podem ser uma desvantagem.

Proposta: Analisar três provedores de nuvem diferentes (Microsoft Azure, Amazon AWS e Google Cloud), por meio da aplicação paralela ImbBench, em termos de adequação a uma execução tão heterogênea de aplicações paralelas grandes. O ImbBench é um aplicativo baseado em MPI que pode criar diferentes padrões de desequilíbrio no uso da CPU e da memória que imita o comportamento de aplicações HPC do mundo real.

Validação: Execução do *benchmark* ImbBench em infraestruturas compostas por *clusters* com 32 núcleos, formados por quatro instâncias com oito *cores* cada. As instâncias foram criadas nos três provedores escolhidos para este trabalho. Para a métrica de eficiência de custos foi utilizada a metodologia descrita em trabalhos anteriores do autor.

Trabalhos Futuros: Estender o *benchmark* ImbBench, adicionando suporte para operações de entrada e saída e comunicação de rede a fim de avaliar esses aspectos em termos de heterogeneidade. Também acrescentar suporte para combinar diferentes tipos de operações para melhor representar aplicações do mundo real. Além disso, fornecer uma maneira automática de combinação de instâncias da nuvem para um comportamento de aplicação específico.

4.1.4 Considerações

Com a conclusão da Revisão Sistemática da Literatura foi possível identificar doze trabalhos que abordam temas relacionados à análise de investimentos em infraestruturas de nuvem. Os trabalhos levam em consideração a execução de aplicações com demandas de processamento de alto desempenho em ambientes de nuvem, bem como a possibilidade de migração destas aplicações, a partir de estruturas clássicas de HPC, para ambientes de nuvem e os custos financeiros envolvidos. Na Seção 4.4 estes artigos são discutidos como trabalhos relacionados ao tema da pesquisa.

4.2 Síntese

Nesta seção, o estudo decorrente da leitura dos trabalhos selecionados é sintetizado de forma a responder as questões de pesquisa que motivaram a realização da presente RSL. Estas questões são, portanto, retomadas e respondidas tendo como base o conhecimento absorvido.

QP 1: Qual o impacto econômico nas decisões de implantação de aplicações de alto desempenho em nuvens computacionais?

Esta QP tem o objetivo de identificar, considerando custos financeiros, como instituições podem se beneficiar da adoção de nuvens. Os trabalhos (ROLOFF et al., 2012; SADOOGHI et al., 2017) exemplificam estes ganhos caracterizando as aplicações e investigando aspectos econômicos das execuções HPC em ambientes de nuvem. Também são elencados argumentos para determinar quando aplicações de alto desempenho podem, realmente, obter vantagens de nuvens computacionais em detrimento de infraestruturas locais dedicadas.

QP 2: Como os usuários podem identificar qual a melhor configuração de nuvem, seja pública ou privada, para executar suas aplicações?

A migração para um ambiente de nuvem computacional requer, da parte de instituição, investimentos na nova infraestrutura. Como caracterizado em (DREHER et al., 2017; GUERRERO et al., 2014), a análise de custos deve tanto identificar o tipo de nuvem a ser implantado, privada ou pública, não desconsiderando a hipótese de uma implementação híbrida (GUPTA et al., 2016; SADOOGHI et al., 2017). Em alguns casos, a análise dos custos utiliza dados de desempenho obtidos por simulação (SHEN et al., 2014; GUPTA et al., 2016; ARABNEJAD; BUBENDORFER; NG, 2017) ou por experimentos envolvendo o uso de uma infraestrutura existente, explorando a execução de benchmarks (ROLOFF et al., 2012; MARATHE et al., 2013; GUPTA et al., 2016; PRUKKANTRAGORN; TIENTANOPAJAI, 2016; ROLOFF et al., 2017; SADOOGHI et al., 2017; EMERAS et al., 2019; ROLOFF et al., 2019) ou analisando o comportamento da execução (MARATHE et al., 2013; ARABNEJAD; BUBENDORFER; NG, 2017). Este tipo de estudo requer grande envolvimento de pessoal, seja na elaboração do processo de simulação ou de coleta de dados de execuções, seja na análise e interpretação dos resultados. A literatura também apresenta, como em (ROLOFF et al., 2012; MARATHE et al., 2013; GUERRERO et al., 2014; SHEN et al., 2014; PRUKKANTRAGORN; TIENTANOPAJAI, 2016; DREHER et al., 2017; SADOOGHI et al., 2017; EMERAS et al., 2019), modelos de custos analíticos, cujo esforço de aplicação é, quando comparado aos anteriormente citados, menor. O aspecto relevante a ser considerado, neste caso, é identificar o grau de precisão do modelo a ser utilizado.

QP 3: Quais modelos de custo financeiro são empregados na execução de aplicações de alto desempenho em nuvens?

Os trabalhos identificados na RSL empregam informações sobre o desempenho das aplicações na análise do impacto do investimento realizado. Nestes trabalhos (ROLOFF et al., 2012; MARATHE et al., 2013; GUPTA et al., 2016; ROLOFF et al., 2017; SADOOGHI et al., 2017; PRUKKANTRAGORN; TIENTANOPAJAI, 2016; EMERAS et al., 2019), a métrica sobre desempenho pode ser entendida como principal componente do modelo de custo. Destaca-se que, para análise do investimento e do seu impacto no desempenho da aplicação, o modelo de análise de custo TCO foi o mais utilizado. Também foi identificado que, à exceção do trabalho T11 (EMERAS et al., 2019), os modelos de custo apresentados estão voltados para responder às necessidades dos usuários sobre a análise dos custos. Os trabalhos (SHEN et al., 2014) e (GUPTA et al., 2016) apresentaram uma análise de custo sobre a ótica do provedor.

QP 4: Como identificar eventuais perdas financeiras em função de imprecisão no dimensionamento de infraestruturas de nuvens para aplicações com demandas de HPC?

Nos trabalhos identificados na RSL, os casos de estudo relatados nos trabalhos selecionados apresentam a instanciação de uma aplicação na nuvem e avaliação de seu comportamento. A perda financeira é analisada pela avaliação da estimativa do desempenho das aplicações sobre um conjunto de recursos alocados. O modelo de decisão baseado

em custos, TCO, é o mais utilizado para apoiar esta análise.

4.3 Oportunidades de Pesquisa

A consolidação das tecnologias de computação em nuvem promoveu um grande crescimento no interesse por ambientes capazes de suportar a execução de aplicações que necessitam de alto desempenho (HPC). Conforme apresentado neste trabalho, percebe-se que a migração das aplicações, a partir de estruturas locais dedicadas para nuvens, não é uma tarefa fácil e traz alguns desafios, principalmente no que diz respeito aos custos financeiros envolvidos no processo. Neste estudo, uma revisão sistemática da literatura buscou temas relacionados à análise de investimentos em infraestruturas de nuvens computacionais e quais aspectos devem ser levados em consideração frente aos novos desafios impostos pela migração de aplicações HPC para ambientes de nuvem.

No entanto, na adoção de uma infraestrutura de nuvem, eventuais perdas financeiras não são resultado apenas do sub ou superdimensionamento dos recursos alocados. Outros aspectos podem ser relevantes no contexto, como a questão de privacidade das informações (RAMGOVIND; ELOFF; SMITH, 2010; BHAVANI; JYOTHI, 2017) e também o suporte à aplicações de missão crítica (FICCO; AMATO; VENTICINQUE, 2018) que limitam o horizonte de opções de implantação da infraestrutura de suporte.

Especificamente sobre custos relacionados à infraestrutura, observa-se que não emergiram considerações sobre os custos de comunicação associados às transferências de dados nem à adoção de soluções de nuvens híbridas. Estes aspectos se mostram como oportunidades de pesquisa em aberto. Outra consideração a ser feita é que os trabalhos selecionados, embora focados em aplicações HPC, não consideram características específicas a determinadas categorias de instituições, como industrial, comercial, acadêmica ou de pesquisa, na análise dos investimentos. Entende-se que a natureza das instituições deva impactar na análise dos resultados financeiros, pois é possível que, como pode ser o caso em instituições acadêmicas e de pesquisa, a análise em termos do resultado financeiro imediato pode não ser suficiente. Um estudo aprofundado sobre o uso de infraestruturas de nuvem em ambientes acadêmicos e de pesquisa se apresenta, assim, como um tema a ser investigado.

4.4 Trabalhos Relacionados

Esta seção apresenta os trabalhos relacionados ao tema abordado neste estudo, destacando os pontos relevantes com relação à análise de investimentos para infraestruturas de nuvem.

No mercado de computação em nuvem, o valor cobrado por cada solução varia muito. Levar em conta somente o desempenho na comparação de fornecedores pode não ser

suficiente. Em seu trabalho, (ROLOFF et al., 2012) (T01, conforme Tabela 8) define uma métrica de eficiência de custos que se propõe a realizar uma comparação mais justa em relação ao que é disponibilizado ao usuário, escalando o valor do desempenho com o preço por hora. Suas conclusões mostram que nuvens podem fornecer uma plataforma viável para a execução de aplicações HPC, mesmo que com algumas desvantagens na implantação, como criação e personalização de instâncias virtuais, problemas de conexão e gerenciamento e tempo de inicialização. Em *benchmarks*, os provedores de nuvem obtiveram desempenho e eficiência de custos melhores que o *cluster* local. Além disso, é necessário analisar o comportamento das aplicações de destino, bem como características dos provedores, a fim de escolher o mais adequado para uma determinada aplicação.

Em (MARATHE et al., 2013) (T02), são comparadas instâncias da Amazon EC2 com *clusters* locais na execução de um conjunto de *benchmarks*. O índice considerado é o tempo de resposta. Com relação aos tempos de espera em filas, no modelo, foram utilizados traços de execuções reais. Para que fosse possível uma comparação dos custos totais das execuções, foi desenvolvido um modelo econômico com o objetivo de precificar os *clusters*, supondo que estes fossem oferecidos como recursos de nuvem. Por fim, os autores concluem que *clusters* HPC de ponta são superiores em desempenho e que, a partir da perspectiva de usuário, há várias considerações na escolha de uma plataforma, como tempo de espera e custo real.

O trabalho de (GUERRERO et al., 2014) (T03) apresenta um modelo de desempenho/-custo que permite ao usuário decidir qual infraestrutura, local ou em nuvem pública, é ideal para um determinado tipo e tamanho de problema. Quanto maior o número de recursos computacionais ou maior o tempo de execução, maior o custo total, mesmo para estruturas locais não utilizadas. Um exemplo são aplicações que fazem uso intensivo de aceleradores baseados em GPUs. Tal condição pode sobrecarregar o orçamento de uma instituição ao processar grandes quantidades de dados em um ambiente de nuvem ou gerar desperdícios financeiros devido à subutilização em uma infraestrutura local. A principal conclusão é que o uso de máquinas locais, por ano, deve ser bastante alto, algo entre 50% e 100%, para ser rentável. Do contrário, a computação em nuvem é uma alternativa mais econômica que a computação local se o uso de recursos estiver abaixo desses valores.

Shen e seu grupo, (SHEN et al., 2014) (T04), propuseram uma abordagem para provisionamento de recursos, baseada em preços, capaz de atingir uma meta de economia de custos para provedores de serviços de nuvem. A abordagem proposta fornece um conjunto de algoritmos de previsão junto com um modelo auto-regressivo padrão para facilitar a necessidade de previsão de cargas de trabalhos. Para estimar a relação entre carga de trabalho prevista e quantidade de máquinas virtuais, foram realizadas simulações de instâncias da Amazon EC2. Utilizando as políticas de preços da Amazon, foi obtida uma solução de economia de custos para fornecedores de aplicativos. Os resultados do experimento demonstram que esta abordagem é mais econômica em comparação com

outras soluções de provisionamento.

Uma avaliação detalhada e abrangente de desempenho e custo de execução de um conjunto de aplicações HPC em uma diversidade de plataformas, variando desde supercomputadores às nuvens computacionais foi apresentada por (GUPTA et al., 2016) (T05). Este estudo oportuniza uma visão global que busca responder ao questionamento *por que* e *quem* deve escolher uma nuvem para execução de aplicações HPC e *quais* aplicações e *como* as nuvens devem ser utilizadas para HPC. Também são investigados os aspectos econômicos da execução em nuvem e discute por que é desafiador ou gratificante para os provedores de nuvem operar negócios para a HPC em comparação com as aplicações em nuvem tradicionais. Deste estudo, algumas lições podem ser observadas: *i*) nuvens podem complementar, com sucesso, supercomputadores, porém substituí-los totalmente ainda é inviável. *Cloud bursting* é uma solução promissora; *ii*) para uma execução de alto desempenho eficiente em nuvem, as aplicações HPC precisam estar cientes do ambiente de nuvem e a nuvem, por sua vez, deve estar preparada para executar aplicações com demandas de alto desempenho; e *iii*) os benefícios econômicos são substanciais, porém, as análises de custo/desempenho para aplicações HPC não são uma tarefa trivial.

(PRUKKANTRAGORN; TIENTANOPAJAI, 2016) (T06), investigaram a eficiência de valores cobrados que podem beneficiar o cliente de serviços em nuvem no processo de escolha dos pacotes de otimizações oferecidos pelos provedores. Foram estudados os valores praticados pelo provedor de serviços Amazon para a execução de cargas de trabalho de computação de alto desempenho. Ao final, este trabalho apresenta a instância de tamanho adequado para o uso de HPC em diferentes cargas de trabalho e proporções entre o valor cobrado e o tempo de execução reduzido, destacando que a decisão na escolha de um pacote depende da satisfação e uso dos clientes.

No trabalho de (ARABNEJAD; BUBENDORFER; NG, 2017) (T07) são apresentados dois algoritmos, o *Proportional Deadline Constrained* (PDC) e o *Deadline Constrained Critical Path* (DCCP) que abordam o problema de escalonamento de fluxos de trabalho nos recursos de nuvem provisionados dinamicamente. Em termos de desempenho de custo, em geral, os algoritmos PDC e DCCP retornaram o menor custo de computação, em todos os fluxos de trabalho e configurações de instância. No geral, ambos os algoritmos são capazes de obter altas taxas de sucesso, enquanto na maioria dos casos apresentam o menor custo geral por uso.

Os últimos anos conduziram a um rápido aumento no número de tipos de configurações de *hardware* de computação em nuvens públicas e opções de valores oferecidos aos usuários, conforme nos demonstra (DREHER et al., 2017) (T08) em seu estudo. Tal movimentação dificulta a análise de qual opção se torna mais vantajosa, economicamente, em uma migração de aplicações de uso geral para a nuvem. Com isso, o autor investiga se esta mesma análise pode ser estendida para execuções de HPC. Com o uso de uma configuração de *hardware* clássica para HPC, foi realizada uma comparação do custo total

das operações de vários provedores de nuvem pública e privada de HPC. A análise mostrou sob quais condições operacionais a opção de nuvem pública pode ser uma alternativa mais econômica para aplicações do tipo HPC.

Em (ROLOFF et al., 2017) (T09), os autores buscaram identificar as instâncias de máquina virtual, em nuvens públicas disponíveis, que possam ser adequadas para aplicações HPC, tanto em termos de desempenho quanto de eficiência de custos. Também foram avaliados quais tipos de aplicações podem se beneficiar da execução na nuvem. Para isso, é preciso analisar as características das instâncias, levando em consideração os aspectos relevantes para HPC. Também é necessária uma análise da eficiência de custos usando os *benchmarks* tradicionais de HPC. Os resultados mostraram que o desempenho de rede continua sendo um gargalo significativo para o desempenho das aplicações. Além disso, pagar por uma nuvem mais poderosa nem sempre garante melhorias e pode até levar a reduções de desempenho. Isso deve ao fato de que as aplicações HPC podem ter características de escalonamento não suportadas pelo ambiente de nuvem, além de sofrer com os efeitos de limitações próprias de ambientes virtualizados, comumente utilizados em nuvens computacionais.

Com foco em aplicações científicas, os autores de (SADOOGHI et al., 2017) (T10) realizaram uma avaliação das instâncias da Amazon EC2, com o objetivo de executar aplicações com desempenho satisfatório e com custo potencialmente mais baixo. Em comparação das instâncias de nuvem pública com as de uma nuvem privada, foi constatado que a eficiência e o desempenho das duas estruturas eram bastante similares. Quanto aos custos, as instâncias virtuais otimizadas para computação são as que apresentaram melhor eficiência financeira. O maior gargalo detectado foi com relação à comunicação de rede que pode afetar, diretamente, a execução satisfatória de aplicações HPC.

No seu trabalho, (EMERAS et al., 2019) (T11) propôs analisar os elementos que compõe o custo total de propriedade (TCO) de uma estrutura interna de HPC, operando desde 2007 e, de posse das informações, comparar os custos necessários para executar a mesma carga de trabalho em uma estrutura de nuvem pública. Os resultados obtidos mostram que a migração de cargas de trabalho HPC para nuvem não é apenas um problema de adaptabilidade do desempenho da nuvem às necessidades das aplicações HPC, mas também um problema ao determinar corretamente quais tipos de trabalho são bons candidatos a serem executados na nuvem para evitar sobrecarga de custos.

Em seu estudo, (ROLOFF et al., 2019) (T12) analisou três provedores de nuvens públicas: Microsoft Azure, Amazon AWS e Google Cloud. Com uso de uma aplicação paralela ImbBench, avaliou a adequação de uma execução heterogênea de aplicações paralelas grandes. A avaliação de eficiência de custo foi realizada por meio da metodologia apresentada em (ROLOFF et al., 2012). Os resultados destacam que a execução heterogênea é mais benéfica na plataforma Azure, com uma eficiência de custos de até 50% em comparação com a execução em instâncias homogêneas, mantendo o mesmo desempenho. Os

outros dois provedores são menos adequados, pois o tipo de instância mais barato também é o mais rápido, para o caso da Amazon AWS, ou o provedor oferece apenas instâncias que variam no tamanho da memória, mas não no desempenho, como é o caso do provedor Google Cloud.

A Tabela 9 sumariza os trabalhos relacionados nesta seção. São apresentadas características identificadas nos trabalhos para um melhor entendimento do posicionamento de cada um deles em relação aos demais. A primeira coluna desta tabela identifica os trabalhos que tratam diretamente de assuntos relacionados às execuções HPC. Nas colunas "Simulação" e "Infra. Real" são marcados os trabalhos de acordo com a técnica utilizada para validação das propostas. Na sequência, as colunas "Traço" e "*Benchmark*" indicam os tipos de origem dos dados usados para a validações.

Nas colunas seguintes, "Nuvem Privada", "Nuvem Pública" e "*Cluster* Local" são referenciados os ambientes de testes utilizados para execução dos experimentos apresentados nos trabalhos. A coluna "Análise de Custo" indica os trabalhos que apresentam análises de custo que considerem aplicações HPC suas execuções em ambientes de nuvem. Por fim, as colunas "Satisf. Usuário" e "Satisf. Provedor" caracterizam se o trabalho considera a satisfação do usuário, do provedor de nuvem ou de ambos.

Tabela 9 – Abordagens adotadas pelos trabalhos relacionados

	HPC	**Simulação**	**Infra. real**	**Traço**	***Benchmark***	**Nuvem privada**	**Nuvem pública**	***Cluster* local**	**Análise de custo**	**Satisf. usuário**	**Satisf. provedor**	***Cloud bursting***
(ROLOFF et al., 2012)	✓		✓		✓		✓	✓	✓	✓		
(MARATHE et al., 2013)	✓		✓	✓	✓		✓	✓	✓	✓		
(GUERRERO et al., 2014)	✓		✓				✓	✓	✓	✓		
(SHEN et al., 2014)		✓				✓	✓		✓	✓	✓	
(GUPTA et al., 2016)	✓	✓	✓		✓	✓	✓	✓		✓	✓	
(PRUKKANTRAGORN; TIENTANOPAJAI, 2016)	✓		✓		✓		✓		✓	✓		
(ARABNEJAD; BUBENDORFER; NG, 2017)		✓		✓						✓		
(DREHER et al., 2017)	✓					✓	✓	✓	✓	✓		
(ROLOFF et al., 2017)	✓		✓		✓		✓	✓		✓		
(SADOOGHI et al., 2017)	✓		✓		✓	✓	✓		✓	✓		
(EMERAS et al., 2019)	✓				✓		✓	✓	✓			
(ROLOFF et al., 2019)	✓		✓		✓		✓			✓		
Proposta deste Trabalho	✓	✓			✓	✓	✓		✓		✓	✓

Em comparação com os trabalhos relacionados (Tabela 9), este estudo busca avaliar a utilização de abordagens que envolvam técnicas de *cloud bursting* no desenvolvimento e aplicação de estratégias de escalonamento. Tais estratégias podem ser empregadas em diferentes aplicações sobre estruturas diferenciadas, em termos de capacidade de processamento, em ambientes de nuvem computacional.

5 WCSIM – *WORKFLOW CLOUD SIMULATOR*

Este capítulo apresenta o WCSim, *Workflow Cloud Simulator*, um simulador voltado para analisar o desempenho de aplicações em nuvens computacionais quando submetidas a diferentes estratégias de escalonamento. Este capítulo inicia, na Seção 5.1, apresentando as premissas adotadas para construção do simulador e identifica seus principais atores e elementos. A Seção 5.2 segue posicionando o WCSim em relação a outras ferramentas para simulação de nuvens computacionais. A organização geral do simulador construído é apresentado na Seção 5.3 e a Seção 5.4 apresenta a entrada de dados requisitada para realização de um estudo de caso e a 5.5 a saída de dados produzida. A ilustração de como introduzir novas estratégias de escalonamento é apresentada na Seção 5.6.

5.1 Premissas Básicas

Em um ambiente de nuvem computacional, *usuários* submetem suas *aplicações* sobre a *infraestrutura* disponibilizada, utilizando uma camada de *virtualização de recursos* e explorando estratégias de *escalonamento* para distribuição da carga computacional.

5.1.1 A infraestrutura

Os elementos de *hardware* básicos para construção de uma nuvem são os servidores de processamento, o conceito de sítio, ou nó, e a rede de interconexão entre servidores de processamento e sítios. A modelagem destes três elementos no contexto do WCSim é caracterizada na sequência.

5.1.1.1 O servidor de processamento

O modelo de arquitetura para uma nuvem computacional assumido na simulação tem como elemento básico um servidor de processamento, composto por um *bare metal*. Este *bare metal* é descrito em termos de capacidade de processamento e armazenamento, conforme a seguinte lista de atributos:

- Id: identificação única do nó;
- Name: Nome simbólico para o nó;

- Start: Horário de ingresso do nó na nuvem;
- Cores: Número de *cores* disponíveis;
- MIPS: Velocidade (em milhões de instruções por segundo) de cada *core*;
- RAM: Quantidade de RAM (por *cores*) disponível, informada em gigabytes (Gb);
- Storage: Quantidade de armazenamento em disco disponível, informada em gigabytes (Gb);
- GPU: Quantidade de placas de processamento gráfico disponíveis;
- Log: Registro de atividades sobre o recurso.

O atributo Id é utilizado para identificar o servidor na nuvem instalada. O atributo *Name* foi incluído para permitir maior facilidade na interpretação dos resultados e não é utilizado durante o processo de simulação. Já o atributo *Start* indica a data que o recurso estará disponível na nuvem. Toda submissão a este servidor com data anterior a sua incorporação à nuvem é descartada[1]. Os demais atributos, Cores, MIPS, RAM, Storage e GPU informam a capacidade de processamento oferecida. Na atual implementação do simulador, são oferecidas quatro famílias de *bare metal*, oferecendo quatro capacidades de processamento distintas para os servidores: *default*, *tiny*, *fat* e *huge*. Enquanto as três primeiras famílias são voltadas para construção dos nós federados da nuvem, a configuração huge foi concebida para representar a nuvem pública que eventualmente será alocada para suportar *bursting*. As capacidades de processamento destas diferentes famílias de *bare metal* são apresentadas junto à apresentação dos experimentos (Capítulo 6, página 98). Outras famílias de *bare metal* podem ser introduzidas a partir das existentes, alterando os valores associados a seus atributos.

O atributo *Memory* possui o registro da utilização dos recursos de um determinado servidor. A Seção 5.1.6 descreve as informações coletadas para registro de uso dos recursos.

Os servidores que compõem a infraestrutura estão interligados em uma rede de comunicação. É previsto que todos os nós possam comunicar entre si em canais bidirecionais. A velocidade de comunicação em ambas direções é a mesma. Na implementação realizada, os tempos de comunicação são constantes e não é prevista a ocorrência de falhas.

Em curso de execução, os servidores suportam a execução de máquinas virtuais (ver Seção 5.1.2) e possuem atributos que caracterizam seu estado. Estes atributos são:

- nVM: número de máquinas virtuais em execução sobre o servidor;
- *status*: Estado do servidor, que pode ser:

[1] A atual implementação do simulador não contempla a operação de saída de nó da nuvem (*shutdown*).

  - *alive*: o servidor está ativo e executando ou apto a executar máquinas virtuais;
  - *suspended*: o servidor está suspenso e as máquinas virtuais nele hospedadas também encontram-se com sua execução suspensa.

- *occupedCores*: número de *cores* virtuais alocados por máquinas virtuais;
- *occupedRAM*: quantidade, em Gb, de RAM alocada por máquinas virtuais;
- *occupedStorage*: quantidade de armazenamento, em Gb, em disco alocada por máquinas virtuais;
- *occupedGPU*: quantidade de GPUs alocada por máquinas virtuais.

De forma derivada aos atributos armazenados, outras informações podem ser obtidas, como a taxa de utilização (*utilizationRate*), que corresponde ao grau de degradação do desempenho do *bare metal*, em termos de queda na entrega da quantidade de MIPS nominal do *core*. Esta degradação ocorre quando a demanda de máquinas virtuais por *cores* virtuais suplantar a oferta de *cores* reais. Assim, quando o número de *cores* virtuais alocados sobre o servidor for acima da quantidade de *cores* reais, a quantidade de MIPS entregue é degradada pela razão entre o número de *cores* reais pelo número de *cores* virtuais[2]. No entanto, deve ser observado que alocar máquinas virtuais sobre servidores é uma decisão de escalonamento. Assim, a decisão de alocar máquinas virtuais em um servidor de forma a degradar o desempenho do servidor é uma decisão da política de escalonamento adotada.

5.1.1.2 O sítio

Um *sítio* é composto por um conjunto de servidores de processamento caracterizando um nó de processamento na nuvem. Cada usuário possui um sítio hospedeiro, ou seja, o nó sobre o qual suas aplicações são inicialmente lançadas e para o qual os resultados são repatriados.

A arquitetura de um sítio pode ser heterogênea, com seus servidores de processamento possuindo diferentes capacidades de processamento. Um usuário, ao submeter uma aplicação, realiza esta submissão a qualquer um dos servidores de processamento do seu sítio hospedeiro, sendo a política de escalonamento responsável por determinar em qual servidor deste sítio as execuções serão realizadas.

Os servidores de processamento em um sítio estão interligados entre si por uma rede de comunicação.

[2] A atual versão do simulador não trata degradação de desempenho quando a memória RAM demandada pelos máquinas virtuais suplanta a disponível no nó.

5.1.1.3 A rede de comunicação

Em uma arquitetura de nuvem, a rede de comunicação tem papel determinante no compartilhamento de recursos. Na atual implementação do simulador, o modelo de rede prevê canais de comunicação bidirecionais entre servidores de processamento. Este modelo não contempla o tratamento de ocorrência de falhas de comunicação, mas permite individualizar cada canal de comunicação com uma velocidade própria. A velocidade de comunicação dos canais entre dois servidores é informada em megabits por segundo (Mbps).

Embora espere-se que a rede de comunicação da nuvem tenha um bom desempenho, em termos de largura de banda e tempos de latência, é previsto a ocorrência de atrasos no tempo total de execução de aplicações devido a migração de trabalho entre os servidores de processamento. De forma geral, é esperado que os atrasos nas comunicações internas a um sítio, ou seja, entre servidores de processamento pertencentes a um mesmo nó, sejam inferiores aos atrasos que possam ser observados nas comunicações entre servidores pertencentes a sítios distintos.

No modelo de rede adotado, a velocidade dos canais de comunicação entre servidores define a composição dos diferentes nós que compõem a nuvem. Assim, servidores conectados por canais de comunicação que possuem uma velocidade acima de um determinado limiar são considerados pertencentes a um mesmo nó.

5.1.2 A camada de virtualização

Na camada de virtualização, uma instância consiste no elemento de simulação sobre o qual são alocadas as tarefas submetidas pelo usuário. A instâncias, conceitualmente, possuem uma estrutura bastante semelhante a um *bare metal*, sendo suas características as seguintes:

- Id: identificação única da instância;
- *Status*: Estado da máquina virtual, que pode ser:
  - *alive*: a máquina virtual está ativa e executando ou apta a executar tarefas e as tarefas nela hospedadas também encontram-se com sua execução suspensa;
  - *migrating*: a máquina virtual, além de se encontrar suspended, encontra-se em processo de migração entre dois servidores.
- vCores: Número de *cores* virtuais disponíveis;
- vMIPS: Velocidade (em milhões de instruções por segundo) de cada *core* virtual;
- vRAM: Quantidade de RAM virtual (por *cores*) disponível, informada em gigabytes (Gb);

- vStorage: Quantidade virtual de armazenamento em disco disponível, informada em gigabytes (Gb);
- vGPU: Quantidade de placas de processamento gráfico virtuais disponíveis;
- *Log*: Registro das ações envolvendo as diferentes instâncias.

O atributo Id é utilizado como identificador único para cada instância. Os atributos, vCores, vMPIS, vRAM, vStorage e vGPU informam a capacidade de processamento oferecida pela instância. Na atual implementação do simulador, são oferecidas três famílias de instâncias, oferecendo três capacidades de processamento distintas para máquinas virtuais: *default*, *tiny* e *fat*. As capacidades de processamento destas diferentes famílias de instâncias são apresentadas junto à caracterização dos experimentos. Os nomes utilizados para as famílias de máquinas virtuais são os mesmos adotados para nomear as famílias de *bare metals* (cf. Seção 5.1.1). No entanto, cabem duas observações. Primeiro, as capacidades de processamento oferecidas pelas famílias de instâncias são diferentes daquelas oferecidas pelas famílias de *bare metals*, havendo apenas uma relação de proporcionalidade entre elas. Segundo, não é oferecida uma família denominada huge para máquinas virtuais. Outras famílias para máquinas virtuais podem ser introduzidas, introduzindo novos valores para os atributos de processamento.

5.1.3 O usuário

O usuário é uma entidade no processo de simulação que submete demandas computacionais, na forma de aplicações, à nuvem. Os usuários são individualmente identificados e seus atributos são:

- Id: Identificador único para o usuário;
- *Login*: A data de ingresso (login) do usuário na nuvem;
- *Home*: O sítio na nuvem sobre o qual o usuário está logado;
- Grupo: Indica o nível de acesso aos recursos da nuvem pelo usuário;
- Prioridade: Prioridade de execução das tarefas deste usuário sobre os recursos que possui acesso;
- *Wallet*: Registro de seus créditos e consumos de recursos da nuvem;
- *Log*: Registro das ações do usuário sobre a nuvem.

A data de *Login* informa o momento a partir do qual as aplicações submetidas pelo usuário serão aceitas e poderão ser executadas. Todas as submissões de aplicações de

um determinado usuário com data anterior à data de Login são descartadas[3]. O atributo *Home* indica sobre qual servidor o usuário efetua seu *Login* e sobre o qual todas as suas aplicações serão lançadas. Caso a tentativa de *login* do usuário se dê sobre um servidor não iniciado, é falha a tentativa de *login* e, por consequência, suas aplicações não serão lançadas.

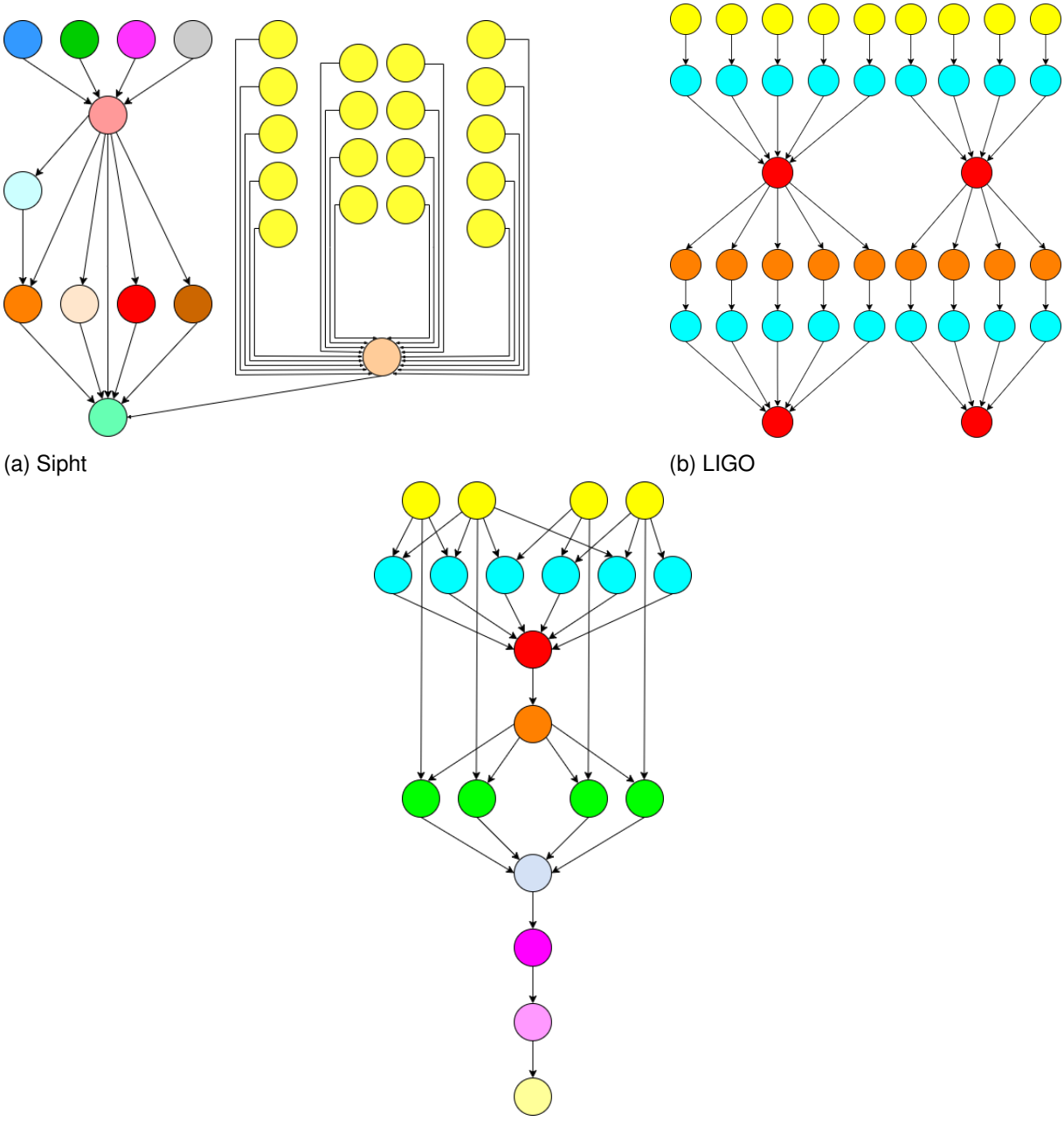
O atributo *Wallet* possui o registro dos recursos (créditos) concedidos ao usuário e do histórico de seus consumos.

5.1.4 Aplicações

A literatura indica que grande parte das aplicações submetidas a grades e nuvens computacionais é composta de aplicações do tipo *bag-of-tasks* (BoT). A literatura também indica que muitas aplicações submetidas a este tipo de arquitetura são descrita na forma de *workflows* (JUVE et al., 2013), sendo vários *workflows* de aplicações científicas documentados e analisados no contexto do projeto Pegasus[4]. A diferença fundamental entre estes dois tipos de aplicações encontra-se nas relações de dependência entre as tarefas. Enquanto nas aplicações BoT as tarefas são independentes entre si, nas aplicações *workflow* existe uma relação de ordem entre as tarefas, refletindo produção e consumo de dados. No entanto, diversas aplicações *workflow* podem ser representadas na forma de um grafo de dependências, ou um grafo dirigido acíclico (DAG, *Direct Acyclic Graph*), no qual os nós do grafo representam um conjunto de tarefas que compõem a aplicação *workflow*, as quais podem ser executadas, entre si, segundo a política de um BoT, e as arestas as dependências entre estes BoTs. A Figura 4 ilustra três *workflows* conforme caracterizados no projeto Pegasus, Sipht, LIGO e Galactic.

[3] A atual implementação do simulador não contempla a operação de logoff.

[4] `https://pegasus.isi.edu`, acesso em 6 de outubro de 2022.



(a) Sipht

(b) LIGO

(c) Galactic

Figura 4 – Workflows caracterizados no projeto Pegasus: Sipht, Ligo e Galactic.

O fluxo de trabalho Sipht, Figura 4.a, do projeto de bioinformática da Universidade de Harvard, é usado para automatizar a busca por RNAs não traduzidos (sRNAs) em *replicons* bacterianos no banco de dados do NCBI[5] (*National Center for Biotechnology Information*).

Por sua vez, o fluxo de trabalho LIGO, Figura 4.b, analisa dados provenientes da coalescência de sistemas binários compactos, como estrelas de nêutrons binárias e buracos negros. Este fluxo de trabalho é muito complexo e é composto por vários subfluxos de trabalho.

Já o fluxo de trabalho Galactic, Figura 4.c, utiliza o motor do mosaico de imagens

[5]O Centro Nacional de Informações Biotecnológicas (NCBI) da Biblioteca Nacional de Medicina dos Estados Unidos mantém bases de dados com informações para acesso a informações biomédicas e genômicas.

Montage[6] para transformar todas as imagens, em 17 *surveys* do céu, para uma escala de pixel comum de 1 segundo de arco, na qual todos os *pixels* são co-registrados no céu e representados em coordenadas galácticas e projeção cartesiana.

Neste trabalho, considerando que um *workflow* estende a especificação de um BoT, este será o modelo de aplicação adotado. Por questões de terminologia, neste trabalho, a expressão *grafo dirigido de BoTs*, ou **DoB** para *DAG of BoTs* será utilizada. A Figura 5 ilustra um fluxo de trabalho, que tem por base o LIGO, com as tarefas dos BoTs que serão executados em cada tarefa do *workflow*. Neste exemplo, cada tarefa de cada BoT é independente e possui um custo computacional de 100.000 MIPS.

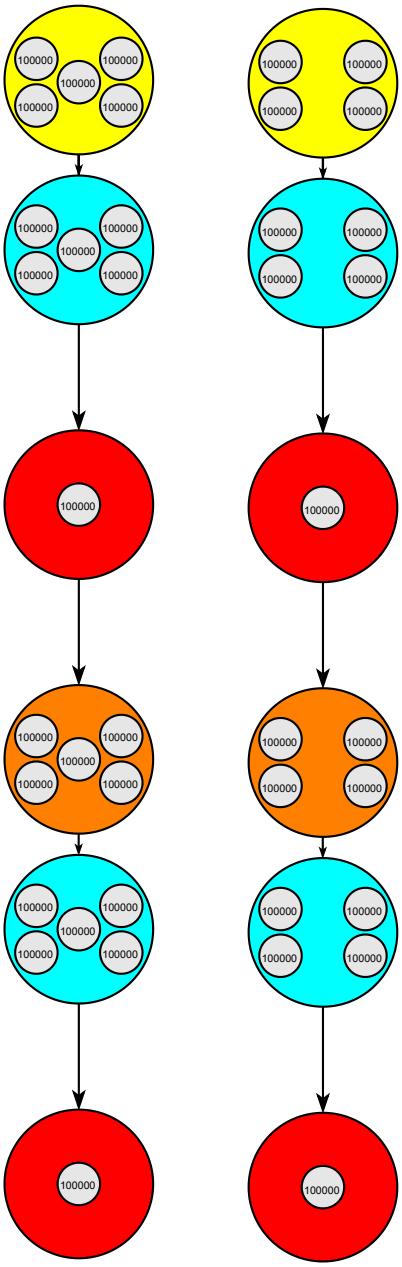


Figura 5 – Fluxo de trabalho baseado no LIGO, com seus BoTs para execução.

[6] A aplicação Montage, criada pela NASA, junta várias imagens de entrada para criar mosaicos personalizados do céu.

Um DoB consiste em uma conjunto de BoTs, existindo a possibilidade de que existam relações de precedência entre os BoTs. Cada BoT descreve um conjunto de $n \geq 1$ tarefas idênticas e possui o seguinte conjunto de atributos:

- Id: Identificação única para o BoT;
- *Owner*: Identificação do usuário responsável pelo lançamento do BoT;
- nSucc: Número de BoTs que estarão habilitados a executar quando o BoT terminar;
- listaSucc: Caso nSucc $\geq$ 1, este atributo é uma lista contendo os nSucc identificadores dos BoTs sucessores;
- nPred: Número de dependências que devem ser atendidas antes que o BoT esteja apto à executar;
- listPred: Caso nPred $\geq$ 1, este atributo é uma lista contendo os nSucc identificadores dos BoTs sucessores;
- *Arrival*: Caso nPred = 0 (o BoT não possui dependências), corresponde à data de lançamento do BoT;
- nTasks: Número de tarefas contidas no BoT;
- nbInstructions: Número, em milhões, de instruções que cada tarefa do BoT deve executar;
- *Memory*: Quantidade de memória, informada em Gb, necessária para executar cada tarefa;
- *Storage*: Quantidade de memória secundária, informada em Gb, necessária para executar cada tarefa;
- nGPUs: Número de GPUs necessárias para execução de cada tarefa;
- *Log*: Registro das ações decorrentes da manipulação do BoT sobre a nuvem.

Um DoB representa uma aplicação submetida por um usuário à nuvem. Um BoT pertencente a um DoB é submetido ao escalonamento quando sua data de lançamento (Arrival) for alcançada ou, se for o caso, quando todas os seus BoTs predecessores tiverem sido concluídos. Um BoT é considerado concluído quando todas as suas tarefas foram concluídas com sucesso. Na atual implementação do simulador, a única situação de falha de execução de uma tarefa tratada é o caso em que a tarefa possui uma demanda de processamento maior do que aquela provida pelo servidor de processamento sobre o qual foi submetida. Outra situação de exceção associada à execução de uma tarefa é a não

disponibilidade de créditos (conforme Seção 5.1.6) pelo usuário para execução de novas tarefas. Nestes casos, a tarefa associada falha e o BoT ao qual pertence não irá concluir sua execução.

5.1.5 O escalonamento

O mecanismo de escalonamento, no simulador, é responsável pela alocação de recursos na nuvem para atender as demandas computacionais geradas pelo lançamento de BoTs. O escalonamento é ativado para realizar diferentes operações:

- Mapeamento de tarefas: diz respeito a seleção da máquina virtual responsável por executar uma tarefa;
- Compartilhamento dos recursos da máquina virtual: distribuição dos recursos computacionais de uma máquina virtual entre as tarefas sobre ela mapeadas;
- Mapeamento de máquinas virtuais: diz respeito a seleção do servidor de processamento responsável por executar uma máquina virtual;
- Compartilhamento dos recursos dos servidores de processamento: distribuição dos recursos computacionais de um servidor de processamento entre as tarefas sobre ele mapeadas.

O escalonamento é dito *correto* quando for capaz de executar, em tempo finito, toda a demanda computacional recebida. A restrição de atendimento a um determinada demanda é de que esta demanda deve ter dimensões compatíveis de serem atendidas pelo provedor do recurso. Caso não seja, a execução da demanda falha. Nos níveis de escalonamento apresentados, como exemplo, uma tarefa ou uma máquina virtual não poderão ser lançadas caso demandem mais memória que aquela oferecida pelas máquinas virtuais do usuário ou dos servidores de processamento disponíveis, respectivamente.

O mecanismo de escalonamento pode ser dotado de uma política para alocação das demandas sobre os recursos computacionais disponíveis. Quando dotado, a política usualmente visa otimizar algum índice de desempenho na execução das cargas computacionais submetidas. Este índice pode estar relacionado ao tempo de execução das aplicações, ao consumo energético ou ainda ao custo financeiro da alocação dos recursos.

5.1.5.1 Política de escalonamento

No modelo de simulação implementado, os níveis de escalonamento que realizam mapeamento tanto de tarefas sobre máquinas virtuais como de máquinas virtuais sobre servidores de processamento podem incluir uma política de alocação de demandas. Os níveis previstos para compartilhamento de recursos, tanto de máquinas virtuais por tarefas como de servidores de processamento entre máquinas virtuais, implementam uma política de round-robin sem prioridade para atender as demandas ativas.

O presente trabalho apresenta WCSim como uma ferramenta de apoio à identificação dos benefícios financeiros do uso de uma nuvem federada entre instituições de ensino. A análise contempla uma arquitetura de nuvem híbrida, com parte das demandas computacionais atendidas por *bursting* de processamento sobre recursos computacionais abrigados em nuvens públicas. Como métrica de apoio à discussão de resultados, assume-se que as políticas de escalonamento adotadas visam a redução do tempo total de execução das aplicações. Estas políticas de escalonamento são introduzidas nos níveis de mapeamento de tarefas a máquinas virtuais e de mapeamento de máquinas virtuais a servidores de processamento.

5.1.5.2 Ativação do escalonamento

O escalonamento no processo de simulação é reativo à evolução das aplicações que são submetidas à execução e a eventuais alterações na topologia da nuvem. Nestes casos, o simulador contempla duas classes de operações de escalonamento, uma aplicada ao mapeamento de tarefas sobre máquinas virtuais, outra aplicada ao mapeamento de máquinas virtuais sobre os servidores de processamento. O escalonamento é ativado sempre que:

- Um BoT torna-se apto a executar, seja por ter chegado a data de seu lançamento, seja por terem sido atendidas as suas dependências;
- Uma tarefa deve ser lançada;
- Uma tarefa termina sua execução;
- Uma máquina virtual é lançada;
- Um novo nó ingressa na nuvem;
- Um nó deixa a nuvem[7].

Um BoT, ao estar apto a executar, tem suas tarefas acolhidas pelo escalonador em nível de tarefa. O escalonamento assume que uma tarefa, ao ser lançada, irá executar sobre a máquina virtual ao qual foi alocada até ser concluída. Este nível de escalonamento, portanto, não aceita preempção nem migração de tarefas entre máquinas virtuais. As operações de escalonamento de tarefas sobre máquinas virtuais são:

- Lançar: selecionar a máquina virtual sobre a qual uma tarefa deverá ser executada e lançar sua execução;
- Finalizar: desocupar os recursos de processamentos alocados a uma tarefa ao seu término;

[7] A atual versão do simulador não trata a situação de falha ou desconexão de um nó da nuvem.

- Pospor: retardar o lançamento de uma tarefa por um tempo determinado pela política de escalonamento.

O escalonador de tarefas restringe a seleção de máquinas virtuais para uma determinada tarefa às máquinas virtuais pertencentes ao usuário responsável pela submissão do BoT. Qualquer máquina virtual pertencente ao usuário pode ser selecionada para qualquer tarefa, independente do sítio onde esta máquina virtual estiver ativa. Localmente a uma máquina virtual, as tarefas são executadas em regime de *round-robin*, sem prioridade. O escalonador de tarefas é ativado e dois momentos: quando uma nova tarefa for submetida a execução e quando uma tarefa termina sua execução.

- Ao receber uma tarefa, o escalonador de tarefas realizar três ações: descartar a tarefa por insuficiência de créditos do usuário, incluir a tarefa em uma lista de tarefas prontas para execução ou então selecionar, segundo algum critério, uma das máquinas virtuais pertencentes ao usuário para executar a nova tarefa. Para tomada de decisão da ação a ser realizada, estão disponíveis informações sobre a taxa de utilização das máquinas virtuais, os atributos da tarefa e do BoT a qual ela pertence, incluindo número de BoTs sucessores, e também os atributos do usuário.

- Ao terminar uma tarefa, o escalonador seleciona, caso exista, alguma tarefa da lista de tarefas prontas e então realiza o procedimento de recebimento de tarefa, descrito no item acima.

Na arquitetura de nuvem concebida para o simulador, cada usuário possui um conjunto de máquinas virtuais a sua disposição. A ação de *login* de um usuário na nuvem, reflete no lançamento de suas máquinas virtuais sobre o sítio de acesso deste usuário. As máquinas virtuais somente são desativadas quando o usuário realizar *logoff*[8] A execução das máquinas virtuais, no entanto, pode sobre operações de preempção, visando suspender momentaneamente sua execução sobre um servidor de processamento ou realizar sua migração entre servidores. As operações de escalonamento de máquinas virtuais sobre servidores de processamento são:

- Lançar: iniciar a execução de uma máquina virtual;
- Derrubar: finalizar a execução de uma máquina virtual;
- Balancear: redistribuir as máquinas virtuais entre os servidores de processamento
- Preemptar: suspender a execução da máquina virtual;
- Retomar: retomar a execução de uma máquina virtual previamente suspensa;

[8] A operação *logoff* não está contemplada na atual versão do simulador.

- Migrar: alterar o servidor sobre o qual a máquina virtual encontra-se processando.

Todos os servidores da nuvem, incluindo os servidores da nuvem pública, caso exista, são candidatos a abrigar qualquer máquina virtual lançada. Atributos relacionados ao usuário, notadamente seu *grupo* ou sua disponibilidade de créditos, podem ser observadas para determinar o comportamento da política de escalonamento em relação à migração de suas máquinas virtuais.

5.1.6 Contabilidade e utilização

O modelo de nuvem adotado considera que o usuário dispõe de um conjunto de créditos a serem utilizados para consumo dos recursos de processamento da nuvem. O valor em créditos disponível é contabilizado atributo de usuário *Wallet*. Na implementação realizada do simulador cada usuário possui uma dotação inicial em créditos, não sendo prevista renovação de créditos. A medida em que as aplicações de um determinado usuário são executadas, seus os créditos são consumidos em função da quantidade de instruções já executadas. É possível, portanto, que uma tarefa lançada consuma todo o saldo restante de créditos do usuário antes de sua conclusão. Esta situação não gera o colapso da tarefa e é permitido que ela conclua com sucesso. À título de registro, um eventual saldo negativo de créditos de um usuário é mantido.

O registro da utilização dos recursos se dá no nível do servidor de processamento e na camada de rede. Em termos de servidor de processamento, considera-se que existem oito faixas de carga de operação. A primeira faixa indica ociosidade do servidor, a última faixa indica sobreutilização do servidor com uma carga acima de 200% da sua capacidade. As demais faixas correspondem aos seguintes intervalos de carga: (faixa 2) > 0 e $< 25\%$, (faixa 3) $\geq 25\%$ e $< 50\%$, (faixa 4) $\geq 50\%$ e $< 75\%$ ... (faixa 7) 175% e $< 200\%$. O registro do uso da rede é realizado no nível do canal de comunicação, onde sua utilização é informa o volume total de dados transmitidos sobre ele.

A carga de operação indica a proporção do número de *cores* virtuais efetivamente ativos, ou seja, os que possuem ao menos uma tarefa em execução, sobre o número de *cores* físicos disponíveis. Se a taxa de utilização é de 50%, então o número de *cores* virtuais executando sobre a máquina representa a metade do número de *cores* físicos disponíveis. Uma taxa de utilização $\geq$ 200% significa que, sobre o servidor, o número de *cores* virtuais executando ao menos uma tarefa, é o dobro ou mais do número de *cores* físicos disponíveis.

Além das informações sobre o saldo em créditos e da utilização dos recursos, o atributo Log, presente em todos componentes da nuvem[9], mantém o registro histórico (log) das ações ocorridas na nuvem. Cada entrada do atributo log anota, ao menos, as seguintes

[9]Os componentes da nuvem são aqueles componentes que, no modelo de simulador adotado, geram eventos que alteram o estado de qualquer elemento envolvido na simulação. Estes componentes processadores de armazenamento, rede de comunicação, máquinas virtuais, usuários, BoTs e tarefas, sendo este aspecto discutido na Seção 5.3.

informações:

- Data: Informa a data que ocorreu a ação;
- Componente Ativador: Identificador único do componente, usuário ou servidor, que
- Componente Ativado: Identificador único do componente envolvido na ação realizada. No caso do componente ativador ser um usuário, este componente pode indicar um servidor de processamento, um BoT, uma tarefa ou ainda o identificador de um usuário ou de um servidor de processamento;
- Ação: Descrição textual da ação realizada, como login de usuário ou ingresso de um servidor na nuvem, lançamento de um BoT, finalização de uma tarefa, término de créditos, recebimento de máquina virtual por migração.

A combinação dos dados de contabilidade, expressos pelo consumo de créditos dos usuários e pela efetiva exploração dos servidores, com os dados do histórico de eventos, expressos pelas informações de registro de eventos na nuvem, permitem analisar a utilização efetiva da nuvem, avaliando seu desempenho no atendimento às demandas apresentadas.

5.2 Simuladores na literatura

A literatura apresenta diversos simuladores voltados para a computação em nuvem. Em (MANSOURI; GHAFARI; ZADE, 2020), onde é feita uma comparação entre diversos simuladores para computação em nuvens já apresentados, são listados 33. Este mesmo trabalho indica que o uso de simuladores é essencial para desenvolver, configurar e avaliar o desempenho de diferentes configurações de arquiteturas, mas que, ainda assim, nenhum dos simuladores é completo e/ou ideal para todo e qualquer tipo de análise. Entre as ferramentas mais populares para simulação de sistemas computacionais em nuvem, encontram-se o SimGrid (CASANOVA et al., 2014) e CloudSim (CALHEIROS et al., 2009), as quais são consideradas neste trabalho. Além da popularidade destas ferramentas, também foi levado em conta na seleção, o fato que membros das comunidades envolvidos no desenvolvimento destas ferramentas possuem relação com nosso grupo de trabalho.

SimGrid[10] trata-se de uma ferramenta genérica, desenvolvida em C, para simulação de ambientes distribuídos. A comunidade envolvida é bastante ativa, havendo facilidades de instalação de pacotes em ambiente GNU-Linux via mecanismos de pacotes compatíveis com a distribuição Debian. A sua atual interface permite o desenvolvimento de casos de simulação em C++, Java e Python, adicionando as devidas extensões, além de sua própria interface nativa em C.

O núcleo de escalonamento de SimGrid é baseado em eventos discretos, fato que pode estar associado à baixa escalabilidade observada no seu desempenho, a medida

[10] `https://simgrid.org`, acesso em 25 de outubro de 2022.

em que a complexidade do modelo de simulação cresce (DEPOORTER et al., 2008). O modelo de simulação inclui o conceito de *actor*. Por meio meio de *actors*, o usuário pode informar o comportamento de simulação no que diz respeito às computações de tarefas, comunicações, utilização de disco, entre outras atividades. O núcleo de simulação prevê o tempo necessário para execução de cada atividade e orquestra a ativação destes actors conforme suas atividades são completadas.

Embora apresentado como um ambiente de simulação genérico para ambientes distribuídos, sua vocação para avaliações de aplicações paralelas, especialmente aplicações desenvolvidas em MPI, é aparente nas diversas publicações, como (RAMAMONJISOA et al., 2014), (PHAM; JÉRON; QUINSON, 2017) e (FANFAKH, 2019) e ainda em (CASANOVA et al., 2018), no qual esta ferramenta foi utilizada como apoio didático ao ensino de MPI. Em outra publicação, (DAOUDI et al., 2020), sua utilização permitiu analisar o comportamento de aplicações no estilo de tarefas OpenMP em arquiteturas NUMA (DAOUDI et al., 2020). Uma funcionalidade do SimGrid é a possibilidade de especificar uma grafo dirigido de tarefas (DAG).

O projeto CloudSim também possui uma comunidade bastante envolvida em seu desenvolvimento. Neste trabalho, é considerado o *fork* do projeto intitulado CloudSim Plus[11], a qual traz diversas melhorias de desempenho a versão original (SILVA FILHO et al., 2017). Por sua vez, o próprio CloudSim foi apresentado como um projeto derivado do GridSim (BUYYA; MURSHED, 2002). CloudSim Plus, a exemplo de seus predecessores, é escrito em Java. O uso do CloudSim Plus é disponibilizado via repositório de *software* e prevê instalação manual, com utilização direta dos fontes disponibilizados na forma de pacotes. O núcleo básico de simulação do CloudSim Plus estende as funcionalidades do pacote SimJava, e prevê um modelo de simulação discreta.

Sendo o projeto do CloudSim mais recente, em relação ao projeto do SimGrid, os conceitos de nuvem estão mais presentes, provavelmente por estarem mais estabelecidos na comunidade envolvente. Conceitos de máquina virtual e *cloudlets*, estes últimos representando uma carga de trabalho, são explorados. O uso de deste simulador tem se destacado na avaliação de estratégias de escalonamento para nuvens nos níveis de escalonamento de carga de trabalho entre máquinas virtuais e de máquinas virtuais entre máquinas hospedeiras, como em (BENDECHACHE et al., 2019), (NARANG; GOSWAMI; JAIN, 2019) e (NARANG; GOSWAMI; JAIN, 2021). Outros trabalhos exploram o uso do CloudSim em outros modelos de computação, derivados da Computação em Nuvem, como Computação na Névoa ((HATTI; SUTAGUNDAR, 2022), (MECHALIKH; TAKTAK; MOUSSA, 2019)) e na Borda ((LI et al., 2020),(LI et al., 2021)).

Em relação a uma comparação entre SimGrid e CloudSim, ressalta-se que ambos possuem em comum uma grande comunidade envolvida em cada um dos projetos e a existência de diversos projetos deles derivados, sendo, provavelmente, como atestado em

[11] `https://cloudsimplus.org`, acesso em 25 de outubro de 2022.

(MANSOURI; GHAFARI; ZADE, 2020), CloudSim o mais popular. É importante também ter em mente que, embora ambos possam ser utilizados para simulação de modelos sobre nuvens computacionais, SimGrid é considerado por (MANSOURI; GHAFARI; ZADE, 2020) um simulador para grades computacionais, com forte vocação para simulação de aplicações com granularidade de tarefa fina (MASTENBROEK et al., 2021), e CloudSim proposto já oferecendo da abstração de *máquina virtual* e a possibilidade de escalonamento de máquinas virtuais – uma extensão à SimGrid ((HIROFUCHI; LÈBRE; POUILLOUX, 2016)) também oferece recursos de manipulação de máquinas virtuais.

Ambos, SimGrid e CloudSim, permitem construir *workflows* de tarefas, o primeiro utilizando uma extensão própria para tal fim, o SimDAG ((MOHAMMED; ELELIEMY; CIORBA, 2017)). Em CloudSim, a construção de um *workflow* é possível adicionando ao evento *término de um cloudlet*, uma ação que cria um, ou eventualmente diversos, *cloudlets* que o sucederão. A bibliografia destas duas ferramentas também contempla seu uso para avaliação de estratégias de escalonamento que consideram, como um dos parâmetros de escalonamento, o custo de uso da nuvem. Exemplos são os artigos (ARABNEJAD H. BARBOSA, 2014), (BUYYA et al., 2005) e (MEHTA; KANUNGO; CHANDWANI, 2011) desenvolvidos tendo como base o SimGrid e, como base o CloudSim, (ARABNEJAD; BUBENDORFER; NG, 2018), (NASR et al., 2019) e (GUPTA et al., 2021).

Outro aspecto observado nestas duas ferramentas, relevantes neste trabalho, é que ambas não proveem, nativamente, uma abstração para entidade representando um usuário com acesso à nuvem. Tal entidade é vista como responsável pelas submissões de trabalho e detentor de direitos sobre máquinas virtuais, tal como introduzido nas premissas básicas do simulador projetado.

5.3 Organização Geral

O WCSim é um simulador de nuvens computacionais de uso geral, implementando um modelo de simulação discreto, capaz de prover recursos de processamento em nuvens computacionais compostas por *data centers* heterogêneos federados, interligados por uma rede de interconexão igualmente heterogênea. O modelo de aplicação é caracterizado como um *workflow* de pacotes de tarefas independentes.

5.3.1 Núcleo de simulação

O *núcleo de simulação* é organizado sobre uma *lista de eventos futuros* e um *relógio global*. O relógio global é representado por um valor inteiro sem sinal, o qual indica a data corrente. Para compatibilizar a medição do tempo com os indicadores de desempenho dos *cores* e dos canais de comunicação, aferidos em MIPS e Mbps, respectivamente, o tempo é apresentado em segundos. A lista de eventos futuros é mantida ordenada pela data de ocorrência do próximo evento. O evento é uma estrutura que contém, além da data do

evento, a identificação do tipo de evento, para que o núcleo de simulação opere a devida ação, e a identificação do componente do ambiente simulado que gerou o evento.

A lista de eventos futuros é mantida ordenada segundo a data em que o evento deverá ocorrer. Havendo dois ou mais eventos programados para a mesma data, a prioridade é dada para eventos envolvendo a infraestrutura da nuvem (servidores de processamento, sítios e canais de comunicação, não havendo prioridade entre estes), então para eventos sobre a camada de virtualização e então do usuário e por fim, envolvendo as aplicações. Os eventos, portanto, são reagrupados em quatro níveis de prioridade.

Em um ciclo contínuo, enquanto o critério de término da simulação não é atingindo, o núcleo de simulação realiza as seguintes operações. O relógio global da simulação avança para a data do primeiro evento na lista de eventos futuro e mantém esta data inalterada até o término do tratamento deste evento. O evento é retirado da lista de eventos futuros e é ativada sua execução. A medida em que um evento é executado, novos eventos podem ser criados e adicionados a lista de eventos futuros. Novos eventos podem ser programados para receberem tratamento na mesma data do evento que o originou ou em uma data futura.

5.3.2 Componente

Um *componente* é uma representação no modelo de simulação de uma entidade dinâmica do modelo real que evolui no domínio do tempo (WAINER; MOSTERMAN, 2018). Um componente é um elemento ativo, que possui um estado associado e que pode promover a realização de atividades associadas ao seu fim[12]. Componentes são reativos a eventos que ocorrem durante o processo de simulação e, por sua vez, eles próprios podem ocasionar novos eventos, os quais podem causar reação em si próprios ou em outros componentes.

São componentes no modelo de simulação: todos os elementos de infraestrutura (sítios, servidores de processamento e canais de comunicação), as máquinas virtuais, o usuário, os BoTs que compõem a aplicação do usuário e as tarefas que compõem os BoTs.

5.3.3 Eventos

Um *evento*, no WCSim, possui a data em que estará habilitado a ser executado, a identificação do tipo do evento e a identificação do componente da simulação que deve sofrer a ação durante o tratamento do evento. Os eventos podem ser exógenos ou endógenos.

Um *evento exógeno* é aquele evento decorrente de uma ação externa a nuvem. Nestes casos, o tempo (data) que o evento se manifesta está associado ao relógio externo a simulação. O tratamento deste evento pode implicar em uma modificação nos elementos da simulação, podendo interferir nas ações já programadas. Exemplos de eventos exógenos: ativação de um novo servidor ou sítio na nuvem, login de um novo usuário, submissão de uma nova aplicação.

[12]Neste texto, o significado do termo componente está contido na terminologia adotada em simulação discreta de eventos e não do contexto associado à disciplina de Engenharia de Software.

Um *evento endógeno* é um evento gerado durante o processo de simulação, cuja data de ativação é dependente das condições de execução na nuvem simulada. Estes eventos obedecem a datação do relógio global, podendo ser programados para uma data futura ou ser tratados imediatamente como consequência do tratamento de um evento em curso. Exemplos de eventos endógenos com tratamento imediato: suspender a execução de uma máquina virtual por decisão do mecanismo de escalonamento – provavelmente em função da detecção de uma situação de desbalanceamento de carga – ou alterar o status de um BoT para pronto para executar assim que todas suas dependências forem satisfeitas. Exemplo de evento endógeno cujo tratamento é programado: reativação de uma máquina virtual sobre um servidor em um sítio remoto após cumprir os custos decorrentes da comunicação.

5.3.4 Diagrama de classes

O simulador WCSim foi implementado em C++ e seu diagrama de classes, estilo UML, é apresentado nas Figuras 6, com as classes relacionadas ao núcleo de simulação, e 7, com as classes relacionadas aos componentes do modelo de simulação. Entre estes dois segmentos do diagrama de classes, a classe `Component` se apresenta como a conexão. Esta classe oferece as funcionalidades para que cada componente do modelo de simulação de nuvem possa manipular a lista de eventos futuros, adicionando novos eventos. Os componentes do modelo de simulação da nuvem herdam estas funcionalidades por herança simples (Figura 7).

Na sequência, estas classes são apresentadas, tendo como apoio a Tabela 10. Nesta tabela são apresentados os principais métodos de cada classe com seus parâmetros de entrada e saída, bem como uma breve descrição de sua funcionalidade.

Tabela 10 – Classes e métodos do WCSim.

Classe	Método	Entrada	Saída	Descrição
Bare metal	get[INFO]	-	int	Métodos de acesso que retornam informações relacionadas ao atributo Info informado, como cores e memória disponível RAM disponível
	miDelivered	-	int	Retorna a velocidade atual de cada *core* em função da taxa de utilização do servidor de processamento.
	pinCore	-	-	Ativa um *core* no servidor, aumentando sua taxa de utilização.
	releaseCore	-	-	Libera um *core* no servidor, diminuindo sua taxa de utilização.
	place	Instance	-	Instancia uma máquina virtual sobre o servidor de processamento.
	unplace	Instance	-	Remove uma máquina virtual do servidor de processamento.
	getVMList	-	list<Instance*>	Retorna a lista de máquinas virtuais abrigadas no servidor de processamento.
BoT	get[INFO]			Retorna o atributo Info solicitado.
	setSucessor	BoT*	-	Vincula o BoT recebido como parâmetro à lista de BoT sucessores.
	readBoTFile	string	-	Faz a leitura de um arquivo .dob.

Tabela 10 – Classes e métodos do WCSim.

Classe	Método	Entrada	Saída	Descrição
	buildBoT	string	-	A partir de cada linha do arquivo .dob, gera um novo BoT.
	aTaskCompleted	-	-	Contabiliza as tarefas completadas do BoT, invocando dependenceSatisfied sobre todos os BoTs sucessores.
	dependenceSatisfied	-	-	Contabiliza as dependências satisfeitas, tornando o BoT pronto para executar ao ter todas as dependências satisfeitas.
Component	pushEvent	Event	-	Insere um novo evento na lista de eventos.
	popEvent	-	-	Retira o primeiro evento da lista de eventos.
	inCurseEvent	-	Event	Retorna o primeiro evento da lista de eventos, sem retirá-lo da lista.
	setName	string	-	Nome oferecido para o componente, para registro em log.
	getName	-	string	Retorna nome oferecido para o evento.
Event	getDate	-	int	Data associada ao evento.
	getPriority	-	int	Prioridade associada ao evento em função do componente.
	execute	-	-	Método virtual que deverá ser especializado com o tratamento do evento.
	setName	string	-	Nome oferecido para o evento, para registro em log.
	getName	-	string	Retorna nome oferecido para o evento.
FutureEventList	push	Event	-	Insere um novo evento na lista de eventos.
	pop	-	-	Retira o primeiro evento da lista de eventos.
	front	-	Event	Retorna o primeiro evento da lista de eventos, sem retirá-lo da lista.
GlobalClock	get	-	int	Retorna o tempo atual.
	set	int	-	Avança o relógio para o novo tempo informado.
Instance	get[INFO]	-	int	Métodos de acesso que retornam informações relacionadas ao atributo Info informado, como *vcores* e memória disponível vRAM disponível.
	getSourceNode	-	int	Retorna o identificador do servidor de processamento em que foi lançada.
	getRunningNode	-	int	Retorna o identificador do servidor de processamento em que encontra-se em execução.
	getTaskList	-	list<Task*>	Retorna a lista de tarefas sob a responsabilidade da máquina virtual.
	place	Task	-	Vincula e execução de uma tarefa sobre a máquina virtual.
	unplace	Task	-	Informa término da execução de uma tarefa.
	migrate	int	-	Suspende a execução da máquina virtual, retomando sua execução no servidor de processamento indicado.
	suspend	-	-	Suspende a execução da máquina virtual.
	resume	-	-	Suspende a execução da máquina virtual.
	getStatus	-	int	Retorna informação de status da máquina virtual (executando, suspensa, migrando).
Log	insert	Event	-	Insere o registro de um novo evento no log.
	list	-	-	Apresenta todo o log de todos os eventos na saída padrão.
	listByEvent	string	-	Apresenta todo o log na saída padrão de todos eventos associado a um determinado nome de evento.
	listByComponent	string	-	Apresenta todo o log na saída padrão de todos eventos associado a um determinado nome de componente.

Tabela 10 – Classes e métodos do WCSim.

Classe	Método	Entrada	Saída	Descrição
Kernel	run	-	-	Executa a simulação.
Cloud	readCloudFile	string	-	Faz a leitura do arquivo .cld, com a descrição dos servidores de processamento que compõem a nuvem.
	setLinkSpeeds	int, vector<int>	-	Informa a velocidade de conexão entre servidor de processamento informado e todos os servidores cujo identificador seja maior que o seu.
	getLinkSpeed	int, int	int	Retorna a velocidade do link entre dois servidores de processamento.
	pushHost	string, Host*	-	Insere um novo servidor de processamento em um sítio da nuvem.
	getHost	int	Host*	Retorna um ponteiro para o servidor de processamento identificado.
Task	get[INFO]	-	int	Métodos de acesso que retornam informações relacionadas ao atributo Info informado.
	hup	int	-	Avança a execução da tarefa pelo tempo informado.
	suspend	-	-	Suspende a execução da tarefa.
	resume	-	-	Retoma a execução da tarefa.
User	readUserFile	string	-	Faz a leitura do arquivo .pwd com a identificação dos usuários que utilizarão a nuvem.
	get[INFO]	-	diversos	Retorna o atributo do usuário solicitada em INFO, sendo o tipo de retorno dependente do tipo do atributo solicitado.
	billing	int, int	-	Adiciona na conta do usuário o uso de um servidor de processamento por um determinado número de instruções.
	getInvoice	-	map<int,int>&	Retorna pares informando, para cada servidor de processamento, quantas instruções o usuário executou sobre ele.

No núcleo de execução (Figura 6), as classes `Kernel`, `GlobalClock` e `FutureEventList` geram *singletons*, sendo os dois últimos acessíveis em todo o sistema. A funcionalidade destas duas classes também é clara em função de seus nomes, permitindo, respectivamente, acessar e alterar a data atual do relógio global de simulação e manipular a lista de eventos futuros, a qual mantém os eventos ordenados. A classe `Kernel` representa o motor de execução do simulador, gerindo a ativação das ações que realizam as atividades previstas pelos eventos.

A classe virtual *Event* oferece, em sua interface, métodos para informar a data em que o evento está previsto para ocorrer, seu nível de prioridade e também os operadores $<$ e $\leq$ para permitir comparação entre os eventos e, então, manter a lista de eventos futuros ordenado. Outra funcionalidade oferecida é a redefinição do operador $<<$ (fluxo de saída), o qual permite gerar informações de log. A operação de construção de um objeto *Event* implica na ativação do componente envolvido no evento e sua consequente inserção na lista de eventos. A invocação ao destrutor deste objeto também ativa o componente envolvido para ciência da ação de término. Os métodos virtuais oferecidos são *execute* e *eventName*. As diferentes classes que vierem a implementar eventos de simulação devem especializar

o comportamento destes métodos com as ações associadas a realização das atividades previstas pelo evento e informar um nome para o evento (para fins de registro em log).

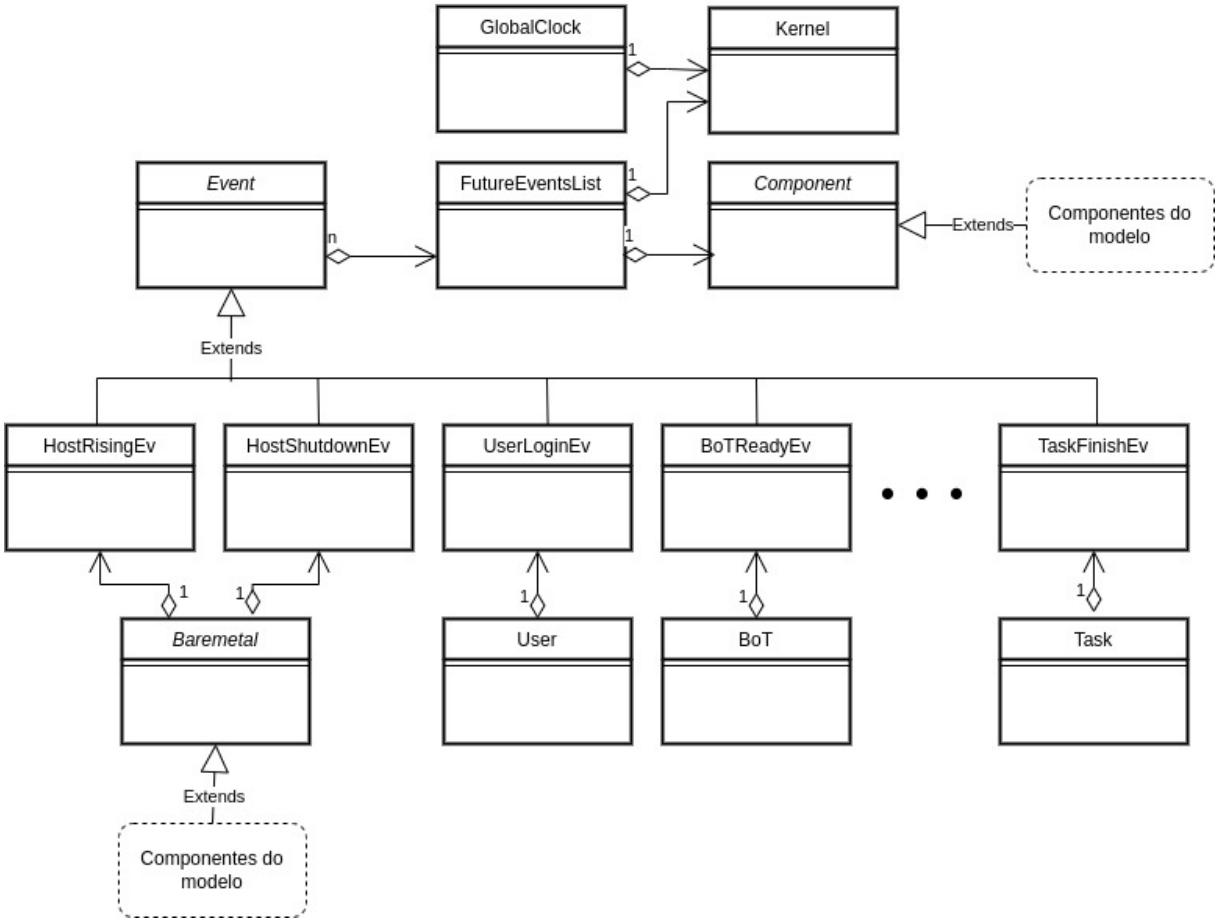


Figura 6 – Representação UML das classes modelando o núcleo de simulação.

Na Figura 7, além de estar reproduzida a associação entre as classe *Component* e FutureEventList, é apresentada a classe Log. Esta classe, possui métodos para adição de registros e apresentação destes na saída padrão. Neste segmento do diagrama de classes, são apresentadas os componentes do modelo de simulação, as classes User, BoT, Task, *Baremetal*, Node, Network e *Instance*.

As classes *Baremetal* é uma classe virtual que abriga uma série de atributos de um servidor de processamento, como quantidade de *cores*, velocidade dos *cores*, quantidade de RAM e armazenamento oferecida, e também de utilização destes recursos, como taxa de utilização dos *cores*, velocidade real de atendimento das demandas em função da taxa de utilização, quantidade de memória RAM e de armazenamento secundário alocadas. Esta classe também oferece os métodos de alocação e liberação de *cores* por máquinas virtuais. Esta classe deve ser especializada por classes que especifiquem os atributos conforme as características desejadas para o servidor de processamento. Como métodos virtuais, a classe *Baremetal* requer que a especificação de uma família de *bare metal* também implemente um métodos para receber a demanda da instanciação de uma nova máquina

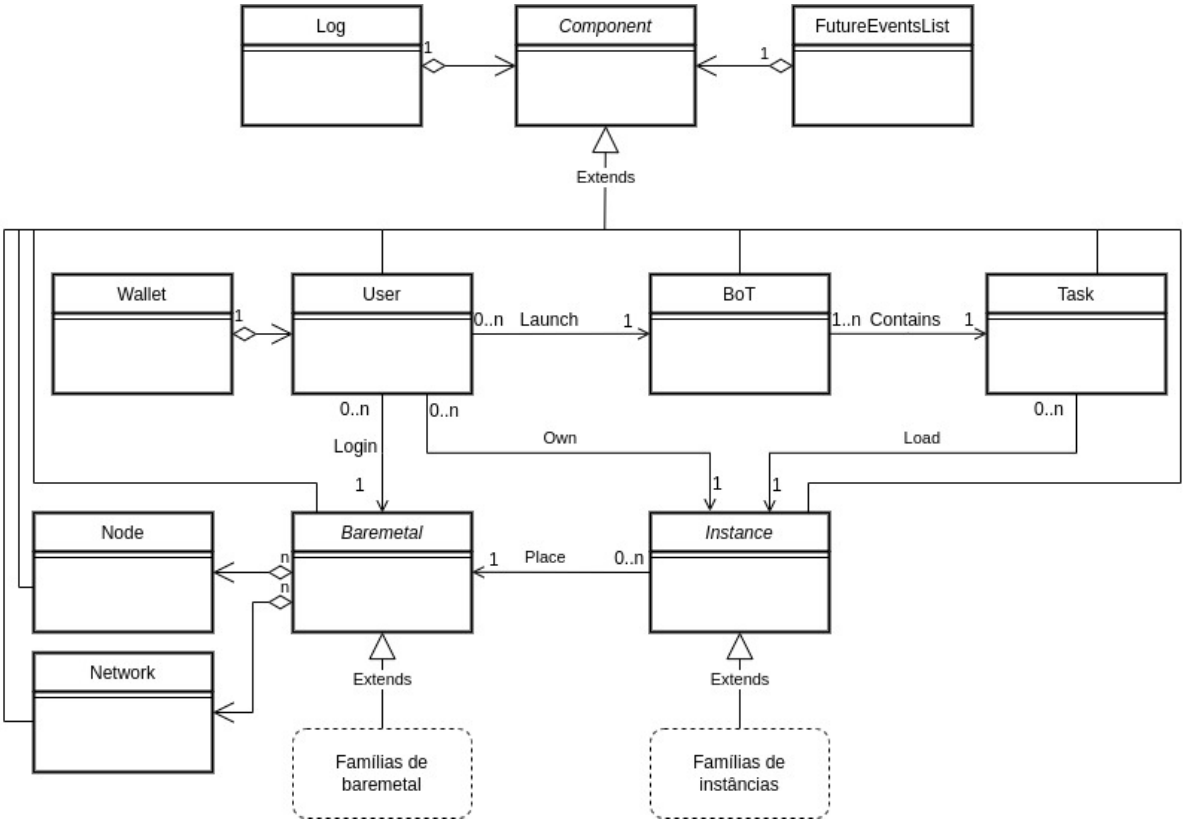


Figura 7 – Representação UML das classes modelando os componentes do modelo de simulação.

virtual e outro para desligar uma máquina virtual.

A classe `Instance` também é uma classe virtual e possui um conjunto de atributos semelhantes aos utilizados para descrever um *bare metal*: como quantidade de *cores*, velocidade dos *cores*, quantidade de RAM e armazenamento virtuais oferecida. Também possui um conjunto de métodos equivalentes aos oferecidos pela classe descritora de servidores de processamento para acesso aos atributos da instância. Em adição, esta classe define um conjuntos de métodos para fazer a alocação de novas tarefas e indicar término de tarefas, além de oferecer funcionalidades para permitir a migração de máquinas virtuais entre servidores de processamento.

As classe `BoT` e `Task` descrevem as aplicações submetidas. A classe `BoT` consiste em um contêiner de tarefas (instâncias da classe `Task`). Instancias desta classe descrevem o conjunto de atributos da aplicação submetida, como custo individual das tarefas, em termos de necessidades de processamento expressa em milhões de instruções, memória, armazenamento e uso de GPU, data de lançamento e relações de precedência no *workflow* do usuário. As instancias dos BoTs também mantêm os identificadores do usuário responsável pelo seu lançamento e tanto do servidor de processamento sobre o qual foi lançado como daquele em que está executando. Além de métodos para acessar informações sobre o BoT, esta classe também prevê métodos para realizar a manutenção das precedências no *workflow*. As instâncias da classe `Task` possuem a identificação do BoT a qual pertence e, a partir desta informação, acesso ao usuário proprietário e às informações sobre os servidores

de processamento. Localmente, cada tarefa mantém informações sobre seus atributos e contabiliza a execução de sua carga computacional. Para realizar esta contabilização, cada tarefa armazena um atributo, denominado `lastDataStamp` que indica a última data em que a tarefa foi manipulada pelo escalonamento.

A classe `User` descreve o modelo adotado para o usuário, responsável pela submissão de aplicações, à nuvem. O usuário possui como principais atributos seus dados de identificação dados pelo seu nome (string) e um identificador único. Outros atributos indicam seu status atual, logado ou não, o número e a família de máquinas virtuais que tem a seu uso, além de oferecer acesso a lista contendo suas máquinas virtuais já instanciadas. Instâncias desta classe também realizam registro do uso da nuvem, para fins de faturamento. No modelo de simulação, instâncias da classe `User` permitem combinar todas as informações entre as tarefas criadas por um usuário, as máquinas virtuais responsáveis pela sua execução, os servidores de processamento ocupados, além de identificar o sítio de origem das aplicações.

Outra funcionalidade oferecida por todas as classes, explorada pelas estratégias de escalonamento, é a possibilidade de iterar sobre todas as instâncias de um determinado tipo. Assim, no momento em que uma atividade de escalonamento é realizada, ela tem acesso a todos sítios, servidores de processamento, máquinas virtuais, BoTs, tarefas e usuários ativos no sistema.

5.3.5 Classes para escalonamento

As classes responsáveis por implementar as estratégias de escalonamento consistem em classes estáticas, não representadas no diagrama de classes do simulador discutidas na Subseção 5.3.4. Estas classes são `Scheduler`, `LoadEvaluator`, `VMSelection`, `HostSelection`, `MigrationVMSelection`, `MigrationHostSelection`. Os métodos da classe `Scheduler` são utilizados pelo kernel de simulação para ativar as diferentes operações de escalonamento na nuvem, utilizando as informações de carga oferecidos pelos métodos da classe `LoadEvaluator`. A implementação das políticas de seleção utilizadas pelo escalonamento é realizada nos métodos das demais classes. Desta forma, a classe `Scheduler` se caracteriza como uma interface, a qual deve ser modificada, combinando o uso dos métodos oferecidos pelas demais classes, para oferecer a política de escalonamento desejada.

Tabela 11 – Métodos da classe `Scheduler`

Classe	Método	Entrada	Saída	Descrição
Scheduler	localSchedule	-	-	Realiza o compartilhamento do uso dos servidores de processamento entre as máquinas virtuais e do poder de processamento das máquinas virtuais entre as tarefas.
	nodeSchedule	-	-	Visita todos os sítios, promovendo o balanceamento de carga entre seus servidores de processamento pela migração de máquinas virtuais.
	cloudSchedule	-	-	Visita todos os sítios federados, promovendo o balanceamento de carga entre servidores de processamento de todos os sítios pela migração de máquinas virtuais.
	burstSchedule	-	-	Visita todos os sítios, promovendo o envio de carga computacional para a nuvem pública.
	vmSelection	User*, Task*	VM*	Seleciona, dentre as máquinas virtuais de um usuário, qual deverá receber a tarefa indicada.
	vmMigrationSelection	Host*	VM*	Seleciona, em um determinado servidor de processamento, qual máquina deve ser migrada.
	hostSelection	Node*, VM*	Host*	Seleciona, em um determinado sítio, qual servidor de processamento deve receber uma máquina virtual.
	receiverNodeSelection	Host*	Host*	Seleciona o servidor de processamento que deve receber uma máquina virtual por migração, limitando a seleção entre servidores de processamento do mesmo sítio.
	receiverCloudSelection	Host*	Host*	Seleciona o servidor de processamento que deve receber uma máquina virtual por migração, permitindo a seleção entre servidores de processamento de sítio diferente da origem.

A classe `Scheduler` tem seus métodos apresentados na Tabela 11. Estes métodos são implementados combinando os métodos oferecidos pelas demais classes associadas ao escalonamento de forma a oferecer a estratégia de escalonamento desejada. Esta estratégia de implementação visa simplificar a adição de novos escalonadores sem a necessidade de manipular diretamente o código do simulador. Os métodos que possuem no nome o termo *Schedule* dizem respeito a implementação do escalonamento nos diferentes níveis de processamento oferecidos pela nuvem. Os métodos que possuem o termo *Selection* oferecem as políticas de seleção de máquinas virtuais e servidores de processamento em caso de alocação de tarefas ou máquinas virtuais ou de migração de máquinas virtuais. Um detalhamento dos métodos de escalonamento desta classe é apresentado na sequência.

O método `localScheduling` implementa a estratégia de escalonamento interna a cada servidor de processamento. Este método é aplicado a cada um dos servidores de processamento ativos, realizando o compartilhamento de seus *cores* físicos entre as máquinas virtuais nele abrigada e dos *cores* virtuais das máquinas virtuais entre as tarefas em execução local. O escalonamento local é ativado a cada passo da simulação, garantindo a evolução das tarefas em execução.

O escalonamento de máquinas virtuais entre os servidores de processamento de um sítio é realizada pelo método `nodeScheduling`. Sua implementação visa promover a migração de máquinas virtuais entre servidores de processamento de um sítio em prol do balanceamento da carga computacional. Diferente do escalonamento local, o escalonamento neste nível não é obrigatório, pois o escalonamento adotado pode não realizar balanceamento de carga entre os servidores de processamento de um sítio. Sendo este nível de escalonamento contemplado, ele pode ser disparado em intervalos de tempo, de duração fixa ou adaptativa, ou na detecção de uma situação em que um servidor encontra-se sobre ou subcarregado (estratégias *sender* ou *receiver initiated*, respectivamente).

O escalonamento no nível da nuvem federada é suportado pelo método `cloudScheduling`. Este nível de escalonamento possui uma semântica operacional bastante semelhante com o escalonamento interno a um sítio e igualmente sua implementação não é obrigatória e disparada em observação ao atendimento a uma condição: intervalo de tempo decorrido ou detecção de sobrecarga ou subutilização de algum servidor de processamento ou sítio. Como observação relevante, salienta-se que, caso exista algum sítio na nuvem pertencente a uma nuvem pública, ou seja, um sítio não federado, o modelo de simulação prevê que este não participa deste nível de escalonamento. No entanto, alguma estratégia *receiver initiated* pode contemplar repatriar a execução de máquinas virtuais sobre os servidores da nuvem pública para servidores de processamento dos sítios federados.

O método `burstScheduling` implementa a estratégia de escalonamento que decide pelo uso de recursos de processamento de uma nuvem federada. Também disparado para atender alguma condição observada durante a simulação, só é aplicado para realizar o envio de carga computacional (representada por máquinas virtuais) para a nuvem pública, quando a nuvem federada estiver sem condições de atender a demanda gerada.

Os métodos para as classes `VMSelection`, `HostSelection`, `MigrationVMSelection` e `MigrationHostSelection` não possuem uma interface rígidas. Estas classes são apresentadas para auxiliar na organização das políticas de escalonamento a medida em que novas estratégias são desenvolvidas. Novas políticas dependem apenas da introdução de métodos que as implementem, passando a estar disponíveis para compor novas estratégias na implementação dos métodos da classe `Scheduler`.

A classe `LoadEvaluator` oferece serviços para detecção da carga computacional nos servidores de processamento, apoiando as decisões das políticas de escalonamento. Esta

classe possui, atualmente, apenas dois métodos: `overLoaded` e `underLoaded`. Ambos métodos recebem como parâmetro um sítio e retornam uma lista, ordenada do maior para o menor, de servidores de processamento que estejam sobrecarregados ou subutilizados, respectivamente, segundo algum critério. A exemplo da classe `Schedule`, um conjunto de classes acessórias pode ser implementada oferecendo diferentes critérios de detecção de carga. A implementação atual estima a carga computacional em função da taxa de utilização dos *cores* físicos.

5.4 Entradas para o Simulador

As entradas para o simulador são em número de três: as descrições dos *workflows*, as informações dos usuários da nuvem e a descrição da infraestruturas. Estas entradas são apresentadas na forma de arquivos `.dob`, `.pas` e `.inf`, respectivamente.

O arquivo `.dob` (DAG of BoTs) descreve o conjunto de aplicações *workflow* a serem avaliadas em uma simulação. Neste arquivo, cada linha descreve um BoT pertencente a um determinado usuário. Os BoTs são identificados individualmente neste arquivo, sendo as precedências entre BoTs, descrevendo o *workflow*, descritas em termos deste identificador. Um fragmento de um arquivo `.dob` é apresentado no Código 5.1. Em um arquivo `.dob`, o caractere # marca início de comentário e a primeira linha identifica o conteúdo de cada coluna neste arquivo. Ao final da descrição de cada BoT foi, neste arquivo, adicionado um comentário caracterizando a operação executada pelo BoT.

Código 5.1 – Fragmento de arquivo `.dob` (DAG of BoTs).

```
#idBoT ownerId nDep arrival ntasks nbInstrucoes memory [id_dep] # Comments
0 0 0 0 4 216000 50           # Blast ...
1 0 1 0 1 432000 50 0         # SRNA
2 0 1 0 1 432000 50 1         # FFN_parse
3 0 1 0 3 216000 50 1         # Blast_candidate ...
4 0 1 0 1 432000 50 2         # Blast_syntese
5 0 3 0 1 216000 50 4 3 8     # SRNA_annotation
6 0 1 0 1 432000 50 5         # Sendemail
7 0 0 0 19 108000 60          # Patser
8 0 1 0 1 432000 50 7         # Patser_con
9 1 0 3600 4 216000 50        # Blast ...
10 1 1 3600 1 432000 50 9     # SRNA
11 1 1 3600 1 432000 50 10    # FFN_parse
12 1 1 3600 3 216000 50 10    # Blast_candidate ...
13 1 1 3600 1 432000 50 11    # Blast_syntese
...
```

A primeira coluna em cada linha que descreve um BoT apresenta seu identificador. Este identificador pode ser utilizado na descrição dos demais BoTs para indicar relações de precedência. Não é necessário que o identificador dos BoTs seja sequencial, mas é

necessário que cada BoT possua um identificador único, para o correto estabelecimento das relações de precedências e, consequentemente, criação do *workflow* do usuário. A segunda coluna de descrição de um BoT identifica o usuário responsável pela submissão do BoT. O identificador apresentado é relacionado com a identificação do usuário, caracterizada no arquivo `.pas`.

A terceira e a quarta coluna identificam, respectivamente, o número de BoTs do qual o BoT descrito depende e a data da chegada, em tempo de simulação, do BoT. Caso o número de dependências seja igual a zero (0), ou seja, o BoT não possui dependências, o tempo de submissão é considerado. Este é o caso das operações `Blast` e `Patser` no fragmento apresentado como exemplo. Caso contrário, o BoT possui ao menos uma dependência, como as operações `SRNA` e `SRNA_syntese`, o valor da data de submissão é desprezado.

As colunas 5, 6 e 7 caracterizam o BoT em termos do número de tarefas e número de instruções e requerimentos de memória de cada tarefa, respectivamente. A partir da coluna oitava, caso existam, são listados os identificadores dos BoTs precedentes, caso existam, do BoT descrito.

Um arquivo `.pas` (password) contém a lista de usuários considerados em uma simulação. Um exemplo deste arquivo é apresentado no Código 5.2. Cada linha corresponde a descrição de um usuário, contendo, nas colunas iniciais, dois strings correspondendo ao nome do usuário e ao nome do sítio ao qual ele se encontra filiado e a data, em tempo de simulação, de seu ingresso no sistema. As duas últimas colunas informam o número de máquinas virtuais que o usuário possui a sua disposição e a família a qual pertencem as instâncias de suas máquinas virtuais.

Código 5.2 – Fragmento de arquivo `.pas` (password).

```
user00 UFPel 0 4 0
user01 UFSM 0 2 0
user02 IFSul 0 4 0
user03 UFPel 0 2 0
user04 UFSM 0 1 0
user05 IFSul 2000 3 0
user06 UFPel 3000 2 0
user07 UFSM 4000 4 0
user08 IFSul 1000000 4 0
```

Em um arquivo `.pas`, a ordem de inserção dos usuários é relevante, pois está relacionada ao identificador associado ao usuário no processo de simulação. O usuário nomeado `user00`, por ser o primeiro, recebe o identificador `0` (zero), `user01` recebe o identificador `1` (um) e assim por diante. Este identificador é relevante para caracterizar o proprietário de cada BoT no arquivo `.dob`.

A infraestrutura geral da nuvem é apresentada em um arquivo `.inf` (infraestrutura) conforme exemplo no Código 5.3. Cada linha descreve um servidor de processamento. A

primeira coluna identifica o sítio ao qual o servidor pertence. No exemplo do código, são descritos três sítios, cada um com quatro servidores de processamento. Internamente, gera identificadores únicos para os servidores com valores inteiros sequenciais, refletindo a ordem de ocorrência de cada servidor no arquivo. Na segunda coluna do arquivo `.inf` é informada data, em tempo de simulação, na qual o servidor entra em operação e na terceira coluna a família de *bare metal* que compõe a arquitetura do servidor. No exemplo apresentado, todos os servidores estão ativo no tempo 0 (zero) de simulação e todos possuem a mesma configuração de arquitetura.

As colunas subsequentes que descrevem cada servidor de processamento indicam as velocidades dos links de comunicação entre os servidores. Esta informação é apresentada em termos de gigabits por segundo (Gbps). Como o modelo de simulação prevê canais de comunicação bidirecionais com mesma largura nos dois sentidos, a apresentação das velocidades dos canais é apresentada em apenas um sentido. Assim, o primeiro valor de velocidade de comunicação apresentado corresponde à velocidade do link de comunicação entre o servidor sendo descrito e o próximo na lista de servidores. O segundo valor corresponde a velocidade do link entre o servidor sendo descrito e o segundo, a partir dele, descrito na lista de servidores. Ao final, o último servidor descrito não possui nenhuma velocidade de comunicação explicitada, uma vez que a velocidade dos links deste servidor para com os demais já foi informada nas linhas anteriores.

Código 5.3 – Arquivo `.inf` (infraestrutura) para uma infraestrutura com três sítios.

```
UFPel 0 0 10 10 10  1  1  1  1  1  1  1  1
UFPel 0 0 10 10  1  1  1  1  1  1  1  1
UFPel 0 0 10  1  1  1  1  1  1  1  1
UFPel 0 0  1  1  1  1  1  1  1  1
UFSM  0 0 10 10 10  1  1  1  1
UFSM  0 0 10 10  1  1  1  1
UFSM  0 0 10  1  1  1  1
UFSM  0 0  1  1  1  1
IFSul 0 0 10 10 10
IFSul 0 0 10 10
IFSul 0 0 10
IFSul 0 0
```

No exemplo apresentado no Código 5.3, a comunicação entre servidores de um mesmo sítio é dada a uma velocidade de 10Gbps (10000Mbps) e, entre servidores de sítios distintos, de 1Gbps (1000Mbps).

5.5 Saídas Oferecidas

As saídas do simulador são em formato de arquivos CSV[13], de forma a facilitar a manipulação dos dados e geração de visualizações alternativas, como gráficos e tabelas, utilizando outras ferramentas. São três arquivos produzidos como saída: `performance`, `trace` e `wallet`. A performance de execução dos *workflows* de cada usuário (Código 5.4) é apresentada em termos de data da execução do último BoT de cada usuário. O arquivo apresenta uma linha para cada usuário, contendo o nome do usuário, o seu identificador e a data, em tempo de simulação, em que todos os BoTs do usuário foram concluídos.

Código 5.4 – Fragmento do arquivo `performance.csv`.

```
#User,UserId,ExecutionTime
user00,0,2594
user01,1,2593
user02,2,2593
user03,3,2593
user04,4,0
...
```

O registro de uso dos recursos computacionais (Código 5.5) identifica a parcela do tempo total da simulação em que cada servidor de processamento se manteve em uma determinada faixa de utilização. O arquivo de saída apresenta uma linha por servidor de processamento. As primeiras duas colunas correspondem ao nome e ao identificador do servidor. As colunas subsequentes, em número de oito, indicam a quantidade de tempo em que o referido servidor de processamento se manteve: ocioso, com carga maior que 0%, mas inferior a 25%, então, nas colunas subsequentes, incremento de 25% nas faixas de uso (ou seja, [25%;50%), [50%;75%)...) até que a oitava coluna de carga indica quanto tempo o servidor permaneceu ocupado a uma taxa de ocupação acima de 200%.

Código 5.5 – Fragmento do arquivo `trace.csv`.

```
#Host,HostId,Idle,25%,50%,75%,100%,125%,150%,175%,>=200%
UFPelH0,0,0,0,464,648,324,157,136,0
UFPelH1,1,230,321,1217,1,0,0,137,190
UFPelH2,2,731,0,1027,161,324,0,325,0
UFPelH3,3,1406,810,729,0,0,0,649,190
UFSMH4,4,2731,808,730,0,0,0,325,190
...
```

Código 5.6 – Fragmento do arquivo `wallet.csv`.

```
#User,UserId,Host,HostId,nbInst
user00,0,UFPel,0,3079728
user00,0,UFPel,1,5726270
```

[13] CSV: *Comma Separated Values* – Valores separados por vírgulas.

```
user00,0,UFPel,2,5022350
user00,0,UFPel,3,3998172
user01,1,UFSM,4,4863310
user01,1,UFSM,5,4105350
user01,1,UFSM,6,3350820
user01,1,UFSM,7,5508702
user02,2,UFPel,0,4428528
user02,2,UFPel,1,3349488
...
```

5.6 Estratégias de Escalonamento

A simulação de execução de aplicações *workflow* em uma nuvem federada considera que o escalonamento deve realizar o mapeamento de tarefas sobre máquinas virtuais, de máquinas virtuais sobre servidores de processamento e ainda garantir que os recursos das máquinas virtuais sejam compartilhados entre as tarefas, bem como os recursos dos servidores compartilhados entre as máquinas virtuais. Neste contexto, tarefas são unidades geradoras de custo computacional no ponto de vista das máquinas virtuais, assim como as máquinas virtuais são vistas como geradoras de custo computacional para os servidores. A estratégia de escalonamento deve prever, então, mecanismos que permitam distribuir os custos computacionais entre os provedores de recursos, buscando balancear o custo computacional entre os provedores, na expectativa de melhorar os índices de desempenho. As funcionalidades de escalonamento modeladas são as seguintes:

1. *Mapeamento de máquinas virtuais sobre servidores de processamento*.
   As máquinas virtuais são lançadas sobre os servidores de processamento do sítio sobre o qual o usuário possui acesso à nuvem. As políticas desenvolvidas são as seguintes:

   (a) ***Static***: O usuário identifica o servidor sobre o qual a máquina virtual deverá ser lançada.

   (b) ***Circular***: Os servidores são organizados em um anel e a seleção se dá de forma ordenada neste anel, de forma circular.

   (c) ***Random***: O servidor é selecionado de forma randômica.

   (d) ***Best Fit***: Aloca a máquina virtual no servidor que deixará o menor número de *cores* livres no servidor após sua alocação.

   (e) ***Worst Fit***: Aloca a máquina virtual no servidor que deixará o maior número de *cores* livres no servidor após sua alocação.

   (f) ***Best Rate***: Aloca a máquina virtual no servidor que encontra-se oferecendo a maior velocidade por *core*.

Nos casos em que existe um critério de seleção, havendo empate no valor do critério entre servidores, é selecionado o servidor com menor identificador.

2. *Escalonamento dos* cores *virtuais das máquinas virtuais sobre os* cores *físicos de um servidor de processamento*.
   Independente do usuário, todas as máquinas virtuais possuem a mesma prioridade de execução, não havendo, portanto, distinção entre os *cores* virtuais em termos de acesso aos recursos de processamento.

   (a) ***Round Robin***: Compartilhamento dos recursos em fila simples, sem prioridade.

   Um fator a observar é relacionado à relação entre o número de *cores* virtuais efetivamente em uso e o número de *cores* físicos disponíveis. Enquanto o número de *cores* virtuais efetivamente em uso não ultrapassar o número de *cores* físicos da máquina hospedeira, a vazão de processamento (velocidade em vMIPS) dos *cores* virtuais é aquela prevista ao seu lançamento. A medida em que o número de *cores* virtuais efetivamente em uso for maior, a queda de desempenho das máquinas virtuais decresce proporcionalmente.[14]

3. *Mapeamento de tarefas sobre máquinas virtuais*.
   As tarefas possuem os mesmos atributos de prioridade e são escalonadas de forma independente. As estratégias buscam distribuir a carga de processamento entre as máquinas virtuais, seja distribuindo as tarefas segundo alguma política associada ao número de tarefas por máquina virtual, seja considerando a velocidade de processamento oferecida.

   (a) ***Static***: O usuário identifica a máquina virtual sobre o qual a tarefa deverá ser lançada.

   (b) ***Circular***: As máquinas virtuais são organizados em um anel e a seleção se dá de forma ordenada neste anel, de forma circular.

   (c) ***Random***: A máquina virtual destino é selecionada de forma randômica.

   (d) ***Balance***: Aloca a tarefa na máquina virtual que possui o menor número de tarefas alocadas.

   (e) ***Best Rate***: Aloca a tarefa na máquina virtual que encontra-se oferecendo a maior velocidade por *core*.

   Em todos os casos, não são considerados os custos de comunicações para deslocamento da tarefa até o servidor de processamento para seleção da máquina virtual. No

[14] O número de *cores* virtuais *efetivamente em uso* de uma máquina virtual depende do número de tarefas sobre ela alocada. Caso este número seja igual ou superior ao número de *cores* virtuais disponíveis, todos os *cores* são considerados efetivamente em uso. Caso seja menor, o número de *cores* efetivamente em uso restringe-se ao número de tarefas alocadas sobre a máquina virtual.

entanto, tendo sido selecionada uma máquina virtual hospedada em outro sítio que não o de origem do usuário, o tempo de latência de comunicação da tarefa entre um servidor de processamento do sítio domínio do usuário (é considerado o servidor com menor identificador) até o servidor destino. Uma vez sobre a máquina virtual destino, a tarefa é considerada pronta e iniciada sua execução.

4. *Escalonamento de tarefas sobre os* cores *virtuais de uma máquina virtual*.
   Independente do usuário, todas as tarefas possuem a mesma prioridade, não havendo, portanto, distinção entre elas em termos de acesso aos recursos virtualizados de processamento.

   (a) ***Round Robin***: Compartilhamento dos recursos em fila simples, sem prioridade.

5. *Balanceamento de carga entre servidores de processamento*.
   Consiste na distribuição da carga computacional entre servidores de processamento de um mesmo sítio.

   (a) ***Chrono***: Disparada após a passagem de um intervalo de tempo, fixo ou determinado de forma adaptativa.

   (b) ***Sender Initiated***: É iniciada por um servidor de processamento sempre que o número de *cores* virtuais suportados for acima de um patamar pré-definido.[15] O servidor destino é selecionado de forma randômica.

   (c) ***Receiver Initiated***: É iniciada por um servidor de processamento sempre que for verificado que nenhum *core* virtual encontra-se alocado. O servidor fonte de trabalho é selecionado de forma randômica.

   Nos casos em que o servidor de processamento selecionado esteja ele também sobre carregado ou não possua carga de trabalho para migrar, a operação de balanceamento falha. Caso alguma política de balanceamento de carga entre sítios ou de *cloud bursting* esteja ativa, uma ação neste nível pode ser disparada.

6. *Balanceamento de carga entre sítios*.
   Consiste na distribuição da carga computacional entre servidores de processamento de diferentes sítios. Caso esteja ativa, este nível de balanceamento é ativado quando a tentativa de balanceamento interna ao sítio falhou.

   (a) ***Chrono***: Disparada após a passagem de um intervalo de tempo, fixo ou determinado de forma adaptativa.

   (b) ***Sender Initiated***: O servidor destino é selecionado de forma randômica dentre os servidores dos diferentes sítios.

[15] Na atual implementação, um servidor de processamento é considerado sobre-carregado quando o número de *cores* virtuais alocados for igual ou superior à 150% do número de *cores* físicos.

(c) ***Receiver Initiated***: O servidor fonte de trabalho é selecionado de forma randômica dentre os servidores dos diferentes sítios.

Nos casos em que o servidor de processamento selecionado esteja ele também sobre carregado ou não possua carga de trabalho para migrar, a operação de balanceamento falha.

7. *Cloud Burst*.
   Os recursos em uma nuvem pública são tratados como um caso particular de sítio na nuvem. As políticas de escalonamento refletem os princípios de utilização de tais recursos.

   (a) ***Chrono***: Disparada após a passagem de um intervalo de tempo, fixo ou determinado de forma adaptativa.

   (b) ***Sender Initiated***: A nuvem recebe uma máquina virtual para processamento.

   (c) ***Receiver Initiated***: Uma máquina virtual, caso disponível, é randomicamente selecionada e repatriada a um servidor da nuvem federada.

   Internamente à nuvem pública, todas as políticas de compartilhamento de recursos são as mesmas das estabelecidas para os servidores de processamento da nuvem federada, tanto ao nível interno aos servidores, quanto no balanceamento de carga interno ao sítio. No entanto, não é permitido aos servidores na nuvem pública disparar operações de escalonamento entre sítios.

No início do processo de simulação, uma infraestrutura básica, composta de sítios contendo servidores de processamento é instanciada. Os servidores se encontram ociosos e as atividade de escalonamento se iniciam quando os usuários da nuvem realizam entrada no sistema (login), instanciando suas máquinas virtuais. A primeira ação de escalonamento é realizar o *mapeamento de máquinas virtuais sobre os servidores de processamento* disponíveis. O modelo prevê que um usuário possui autorização para lançar máquinas virtuais apenas sobre o sítio sobre o qual seu login foi realizado. A atual implementação conta com seis políticas de mapeamento de máquinas virtuais.

Uma vez lançada uma máquina virtual, esta máquina possui o mesmo atributo de prioridade que qualquer outra máquina virtual em execução no sistema. Desta forma, o *escalonamento dos* cores *virtuais das máquinas virtuais sobre os* cores *físicos de um servidor de processamento* se dá em uma política simples de compartilhamento. O simulador conta com recursos para simular a redução de prioridade pela redução temporária da velocidade dos *cores* virtuais, ou mesmo suspender e programar a retomada de máquinas virtuais, mas não foram concebidas políticas de escalonamento explorando tais funcionalidades.‘

Na atual implementação, todas as tarefas submetidas por um usuário, independente de qual BoT pertençam, possuem igual prioridade. A política de *escalonamento de tarefas*

sobre os cores *virtuais* segue, assim, um regime de *round-robin*, sem prioridade. O mecanismo implementado não prevê preempção e retomadas de tarefas, ainda que mecanismos para tais operações estejam previstos no modelo, sendo possível mesmo realizar a migração de tarefas entre máquinas virtuais.

O *balanceamento de carga entre servidores de processamento* tem como objetivo distribuir a carga computacional entre os recursos computacionais de um sítio. São oferecidas duas políticas básicas (WILLEBEEK-LEMAIR; REEVES, 1993), iniciada pelo servidor que for identificado sobre-carregado (*sender initiated*) ou iniciada por um servidor ocioso (*receiver initiated*). A tentativa de balanceamento pode ser bem sucedida ou fracassar. Caso seja bem sucedida, uma máquina virtual, randomicamente selecionada, é migrada, havendo os devidos atrasos na execução das tarefas que abriga em função das latências de comunicação. A implementação prevê que apenas uma tentativa de migração será realizada no atual passo de simulação. Em caso de fracasso, caso esteja habilitado o balanceamento entre sítios ou realização de *cloud bursting*, este escalonamento poderá ser realizado.

No caso do balanceamento de carga entre servidores de processamento falhar, o *balanceamento de carga entre servidores de sítio* pode ser disparado. Neste caso políticas equivalentes, *sender* e *receiver initiated*, são disparadas e a negociação se dá de forma equivalente ao balanceamento interno ao sítio. O *cloud burst* é outra alternativa ao balanceamento de carga entre servidores de processamento. Neste caso, a negociação se dá para migração da carga computacional do servidor de processamento para a nuvem (*sender initiated*) ou para repatriar para a nuvem federada, representada pelo servidor de processamento ocioso, a responsabilidade pela máquina virtual (*receiver initated*).

5.7 Conclusão do Capítulo

Neste capítulo foi apresentada uma importante contribuição deste trabalho, o WCSim, *Workflow Cloud Simulator*. Este simulador foi desenvolvido para suprir a necessidade de avaliar o comportamento de cargas computacionais representadas por *workflows* em nuvens computacionais. O desenvolvimento deste simulador foi motivado pela constatação, documentada na Seção 5.2, de que as duas mais populares ferramentas de simulação de nuvens computacionais, CloudSim e SimGrid, requerem um esforço de programação para modelar aplicações representado *workflows*, estendendo e/ou introduzindo código que represente as aplicações em pontos específicos dos frameworks de simulação oferecidos. Considerou-se necessário possuir um modelo de simulação que permitisse introduzir aplicações de forma descritiva, similar ao proposto em (BARIKA et al., 2019) simplificando tanto a simulação de diferentes padrões de *workflows* como a execução conjunta de diversas aplicações simultaneamente.

Outro aspecto que motivou o desenvolvimento de WCSim foi a constatação da oportunidade de estender os resultados documentados em (SANTOS, 2016) e (SANTOS; DU BOIS;

CAVALHEIRO, 2017), em que foi desenvolvido um simulador para escalonamento de aplicações do tipo *bag-of-tasks* em ambiente de nuvem. Vários aspectos do modelo de simulação anterior foram contemplados, como a consolidação de tarefas em máquinas virtuais e compartilhamento de recursos dos hospedeiros físicos entre os recursos virtualizados. No entanto, destaca-se que o novo simulador oferece abstração para a entidade *Usuário*, que representa o responsável pela submissão de uma aplicação, e permite a construção de *workflows* representando aplicações reais – quando que o primeiro simulador desenvolvido utilizava modelos estatísticos (EPEMA; IOSUP, 2011) descrevendo aplicações recorrentes na nuvem para geração de programas *bag-of-tasks* sintéticos.

O alto grau de adaptação do modelo simulado, em termos de *workflows*, usuários e configuração da infraestrutura da nuvem, sem a necessidade de modificações no código do simulador é um diferencial importante da contribuição dada. Outro ponto forte é também a possibilidade de introduzir estratégias de escalonamento em cinco níveis distintos: (*i*) de tarefas sobre máquinas virtuais, (*ii*) de máquinas virtuais sobre servidores de processamento, (*iii*) entre servidores de processamento pertencentes a um mesmo sítio, (*iv*) entre servidores de processamento pertencentes a sítios distintos e, ainda, (*v*) entre os recursos de processamento de uma nuvem federada e uma nuvem pública. A introdução de novas estratégias de escalonamento se dá de forma programática, reimplementando métodos em uma interface específica.

6 ESTUDO DE CASOS

Neste capítulo, WCSim é utilizado para avaliar o comportamento da execuções de aplicações *workflow* sobre uma nuvem federada, utilizando técnicas de *cloud bursting*. Os resultados obtidos são analisados em função do desempenho em termos de tempo total de execução das aplicações submetidas e custo final da operação. Estes custos são apresentados em função do CAPEX (despesas iniciais de capital) e do OPEX (despesas operacionais) resultante das configurações de infraestrutura de nuvem adotadas.

Além de permitir facilmente alterar as configurações de infraestrutura e a carga de processamento submetida, o simulador proposto ainda permite a introdução de uma vasta gama de estratégias de escalonamento. O horizonte de possibilidades de casos a serem estudados se torna bastante ampla. Neste trabalho, o objetivo é observar o impacto do uso de técnicas de *cloud bursting* no desempenho e no custo de uma rede acadêmica federada. Os cenários propostos para este estudo de casos buscam reduzir o número de variáveis observadas para destacar apenas as consequências da adoção de *cloud bursting* em uma nuvem federada.

6.1 Infraestruturas Modeladas

O modelo de infraestrutura utilizado distingue as famílias de servidores de processamento em termos de número de *cores* físicos disponíveis e as famílias de máquinas virtuais em termos de número de *cores* virtuais oferecidos. Em ambos os casos, modelo de servidor de processamento e máquina virtual, a quantidade de memória RAM e de armazenamento é considerada infinita e não é oferecida a opção de uso de GPU. A Tabela 12 apresenta as configurações de infraestrutura utilizadas e a Tabela 13 as famílias de máquinas virtuais utilizadas.

A informação de custo por *core* (US$ por *core*) na Tabela 12 indica o custo de uso durante uma hora de cada *core* fornecido. Para o caso da configuração *Huge*, utilizada na nuvem pública, o valor representa uma média dos valores cobrados por provedores de serviços IaaS para configurações equivalentes[1]. Para as demais configurações, utilizadas

[1]Os provedores IaaS considerados foram a AWS (Amazon Web Services) e o GCP (Google Cloud Platform) e a tomada de valores realizada via seus sites oficiais.

na implantação da nuvem federada, este custo foi calculado considerando o valor, em dólar, de máquinas possuindo configurações semelhantes às apresentadas[2]. O valor por hora do uso do *core* foi calculado conforme a Equação 16, onde $Investimento$ corresponde ao custo de aquisição de uma máquina na referida configuração, 8760 o número de horas do ano[3] e $0,20$ corresponde a taxa de depreciação anual do equipamento adquirido[4]. Os valores apresentados, portanto, correspondem ao uso dos recursos no seu primeiro ano.

$$Custo\ p/core\ =\ \frac{Investimento \times 0,20}{8760} \tag{16}$$

Tabela 12 – Famílias de servidores de processamento dos estudo de casos

Famílias de Servidores de Processamento					
Código	**Nome**	***Cores***	**MIPS**	**US$ p/ *core***	**US$ total (*cores)***
0	Thin	4	100000	0,03	0,12
1	Medium	12	100000	0,07	0,84
2	Large	24	100000	0,08	1,92
100	Huge	48	100000	0,03	1,44

Tabela 13 – Famílias de máquinas virtuais dos estudo de casos

Famílias de Máquinas Virtuais			
Código	**Nome**	***vCores***	**vMIPS**
0	Thin	1	100
1	Medium	4	100
2	Large	8	100

6.2 Aplicações Modeladas

Nas simulações realizadas, os usuários submetem aplicações Ligo, Sipth e Galactic, conforme definidos por Pegasus (Figura 4, página 68) e modelados na forma de DoBs. São considerados três tamanhos de cargas submetidas, *short*, *medium* e *large*, especificadas como AppS, AppM e AppL na Tabela 14. Cada usuário submete uma aplicação, sendo 1/3 de cada submissão correspondendo a cada uma das aplicações consideradas. O número de DoBs lançados na execução é apresentado na coluna Usuários/DoBs e o número de

[2]Os valores representam uma média simples do preço de venda de servidores em distribuidores nacionais, tendo sido a coleta de preços realizada por consulta aos sites destes distribuidores no início de janeiro de 2023.

[3]Os provedores IaaS, para efeitos de orçamento, consideram que um mês possui 730 horas.

[4]Considera que a taxa anual de depreciação permitida para bens referentes a computadores é de 20% ao ano, no prazo de cinco anos de acordo a legislação brasileira (RIR/1999, art.310).

BoTs gerados em cada caso é apresentado na coluna BoTs. Nos estudos de caso, as cargas de trabalho individuais de cada aplicação são mantidas constantes. A variação está no número de aplicações submetidas à execução. No exercício realizado, tanto os usuários como as aplicações são igualmente distribuídos entre os sítios da nuvem federada, de forma a não existir heterogeneidade nas demandas.

Tabela 14 – Aplicações modeladas para execução no simulador.

Aplicação	**Usuários/DoBs**	**BoTs**
AppS	12	180
AppM	48	720
AppL	192	2880

6.3 Metodologia de Avaliação

São considerados nove cenários de estudo de caso, cada cenário combinando um contexto de demanda de utilização e uma configuração de infraestrutura de nuvem para os sítios federados. São três os contextos de demanda de utilização, denominados AppS, AppM e AppL (Tabela 14) caracterizando diferentes níveis de taxas de utilização, baixa, média e alta demanda. Também em número de três as configurações de infraestrutura: *thin*, *medium* e *large*, caracterizando diferentes capacidades de processamento, de menor à maior quantidade de recursos (Tabela 12). Soma-se ainda à infraestrutura a possibilidade de adicionar um sítio representando uma nuvem pública. Este sítio público foi configurado para possuir uma capacidade de processamento maior que as providas individualmente pelos sítios federados, implantando uma infraestrutura denominada *huge*.

O objetivo dos casos de estudo é identificar os custos de utilização da nuvem federada, considerando os casos em que uma nuvem pública é utilizada para suportar *cloud bursting*. Para efeitos de análise dos estudos de caso, ao final da execução de cada experimento é contabilizado o custo de uso da infraestrutura instanciada considerando os valores na Tabela 12. Seis diferentes estratégias de escalonamento foram utilizadas para permitir avaliar os resultados:

1. **Escalonador 1: Escalonamento Fixo**. Este escalonamento oferece o pior caso em termos de tempo de execução e foi implementado para oferecer uma base de comparação com os demais escalonamentos. Neste escalonamento, todas as máquinas virtuais lançadas em um sítio são executadas sobre o servidor de processamento com identificador 0 (zero) deste sítio e todas as tarefas lançadas por um usuário são executados na sua máquina virtual de identificador também 0 (zero). Não são realizadas operações de migração ou balanceamento de carga.

2. **Escalonador 2: Escalonamento Randômico**: Neste escalonamento também não há

operações de migração de máquinas virtuais ou balanceamento de carga. No entanto, a seleção do servidor de processamento que irá executar uma máquina virtual e a seleção da máquina virtual que irá executar uma tarefa são realizadas de forma aleatória (randômica).

3. **Escalonador 3: Escalonamento Randômico com Balanceamento Intra-Sítio**: Este escalonamento opera as seleções de servidor de processamento e máquina virtual como no Escalonador 2, mas permite o balanceamento de carga, pela migração de máquinas virtuais, entre servidores de processamento de um mesmo sítio.

4. **Escalonador 4: Escalonamento Randômico com Balanceamento Entre-Sítios**: Este escalonamento opera como o Escalonador 3, mas permite que servidores de processamento de sítios distintos sejam considerados para o balanceamento de carga, pela migração de máquinas virtuais.

5. **Escalonador 5: Escalonamento Randômico com Balanceamento Entre-Sítios e Cloud Bursting**: Este escalonamento estende o Escalonador 4, ao permitir que máquinas virtuais sejam migradas para uma nuvem pública. A estratégia primeiro realiza o balanceamento de carga entre sítios para depois avaliar a necessidade de realizar bursting. Deve ser observado que os servidores de processamento do sítio representando a nuvem pública são habilitados a apenas realizar balanceamento de carga intra-sítio, não podendo retornar máquinas virtuais para sítios da nuvem federada.

6. **Escalonador 6: Escalonamento Randômico com Balanceamento Intra-Sítios e *Cloud Bursting***: Este escalonamento estende o Escalonador 3, permitindo a realização de *cloud bursting*, mas não realizando escalonamento entre-sítios. A estratégia primeiro realiza o balanceamento de carga entre servidores de processamento do sítio para depois avaliar a necessidade de realizar *bursting*.

Nas estratégias de escalonamento, a carga considerada é a taxa de ocupação dos servidores de processamento, expressa pela relação entre o número de *cores* virtuais demandados pelas máquinas virtuais hospedadas e o número de *cores* físicos disponíveis. A quantidade de memória disponível, tanto RAM como de armazenamento, foram consideradas infinitas, ou seja, não afetam os escalonamentos implementados. Também importante salientar que todas as estratégias que implicam em balanceamento de carga são do tipo *sender initiated*: um servidor de processamento dispara a realização de uma operação de escalonamento quando sua taxa de ocupação suplantar um determinado patamar, fixado, nos experimentos, em uma taxa de utilização superior à 150% de sua capacidade nominal.

Considerando as três configurações de infraestrutura (Tabela 12), os três cenários de demandas de utilização (Tabela 14 e os seis escalonadores, o número total de estudos

de caso é 54. Os resultados são apresentados em termos de tempo de execução e custo de utilização da infraestrutura. Por tempo de execução entende-se a data em que foi executado o último BoT pelo simulador. Por custo de utilização da infraestrutura entende-se o somatório de dois termos, o primeiro indicando o custo da execução de cada milhão de instrução sobre a infraestrutura federada e o segundo do custo por milhão de instruções executado na nuvem pública.

6.4 Baixa Demanda Computacional

O caso que considera baixa demanda computacional é caracterizado pelo cenário onde quatro usuários em cada sítio submetem, respectivamente, uma aplicação Sipht, Ligo e Galactic. Um total de 12 aplicações são submetidas neste cenário. As avaliações foram realizadas nuvens federadas com infraestruturas oferecendo baixa, média e alta capacidade de processamento.

6.4.1 Infraestrutura de baixa capacidade

A infraestrutura de baixa capacidade neste cenário é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de quatro *cores* físicos rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem baixa capacidade computacional, em infraestruturas ofertando baixa quantidade de recursos de processamento, encontram-se na Tabela 15 e nos gráficos nas figuras 8 e 9. Na Tabela 15 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 8 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 9 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo à demanda e, acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 15 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da

tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 15 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de baixa demanda computacional em infraestrutura de baixa capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	225524	90,21	-	90,21
Caso 2: Randômico	136091	54,55	-	54,44
Caso 3: Randômico, Balanceamento intra-sítio	76197	30,48	-	30,48
Caso 4: Randômico, Balanceamento intra/entre sítios	86792	34,70	-	34,70
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	38010	15,20	60,82	76,02
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	39179	15,67	62,69	78,36
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	38010	15,20	26,75	41,95
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	39179	15,67	25,73	41,40

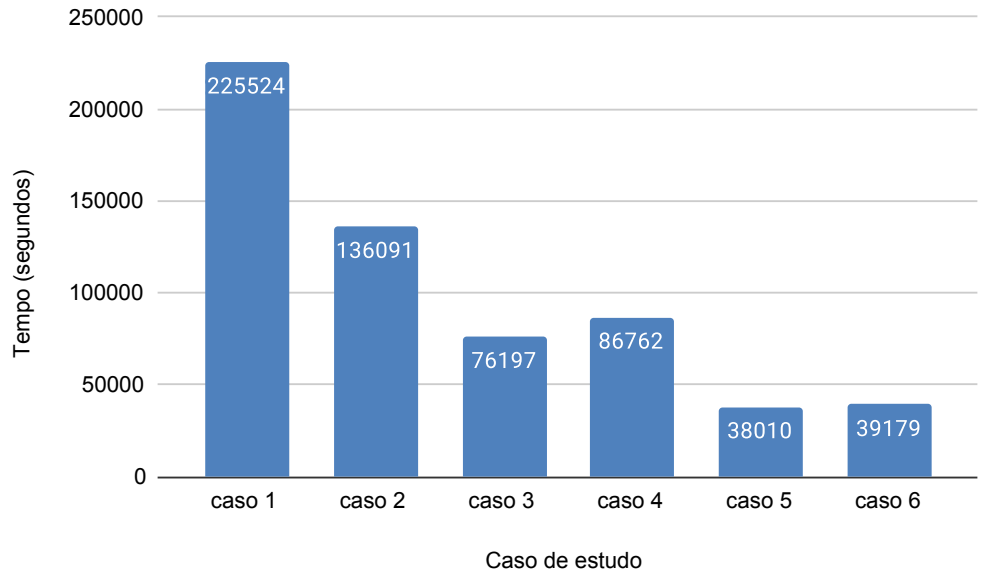
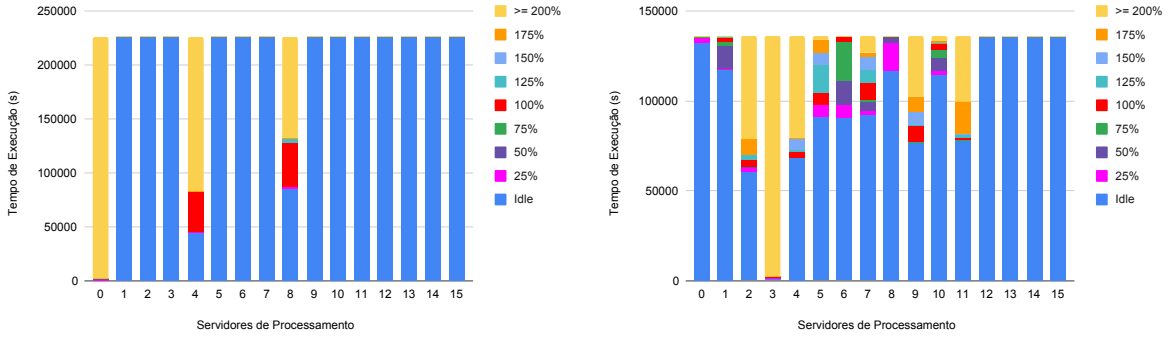
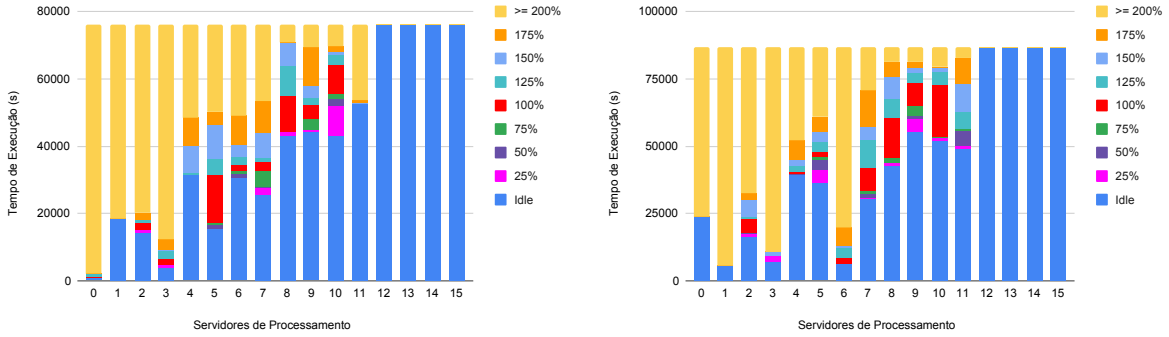


Figura 8 – Tempo de execução de aplicações de baixa demanda computacional em infraestrutura de baixa capacidade.



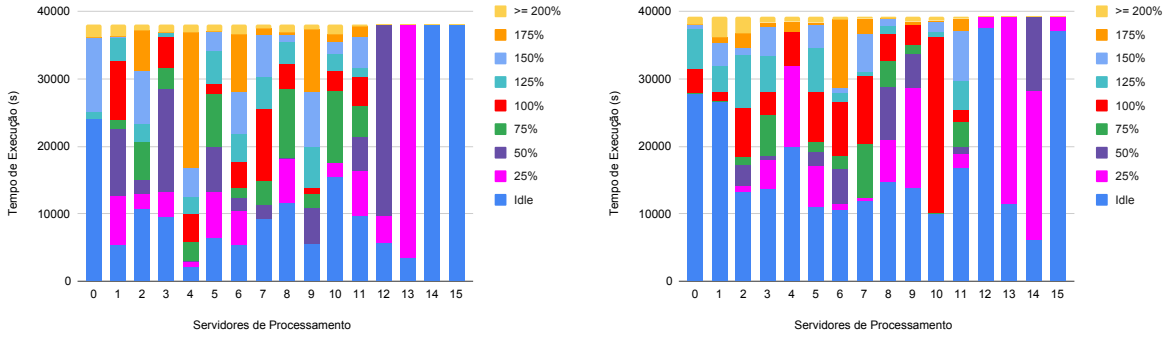
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.

(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.

(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.

(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 9 – Execução de aplicações de baixa demanda computacional em infraestrutura de baixa capacidade.

Os dados apresentados permitem concluir que o cenário de uma infraestrutura pobre em recursos de processamento suportando a execução de uma aplicação com baixa demanda computacional apresenta melhor desempenho quando recursos adicionais à nuvem federada são introduzidos por *cloud bursting*: o melhor tempo de execução foi obtido com o Caso 5. Sem utilização de *cloud bursting*, o melhor escalonamento foi obtido no Caso 3, com balanceamento de carga restrito ao sítio, aos servidores pertencentes ao mesmo sítio em que a aplicação foi submetida. Observa-se também que nos escalonamentos que realizam balanceamento de carga entre sítios, casos 4 e 5, dados os tempos de comunicação maiores entre servidores de processamento em sítios distintos, o tempo de execução também foi maior.

Em relação aos custos, os casos 3 e 4, nesta ordem, explorando apenas recursos da nuvem federada, foram os que apresentaram os menores valores. Explorando *cloud bursting*, a opção de pagamento por *spot* é a que se mostrou mais econômica. Observa-se que os Caso 5 e 6 com *spot* apresentaram um custo cerca de 35% superior ao Caso 3. Já em termos de desempenho, o Caso 6 resolveu o problema requerendo, praticamente, 50% do tempo do Caso 3.

6.4.2 Infraestrutura de média capacidade

A infraestrutura de média capacidade neste cenário é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de doze *cores* físicos, rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem baixa capacidade computacional, em infraestruturas ofertando média quantidade de recursos de processamento, encontram-se na Tabela 16 e nos gráficos nas figuras 10 e 11. Na Tabela 16 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 10 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 11 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 16 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam

cloud bursting e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 16 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de baixa demanda computacional em infraestrutura de média capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	88229	247,04	-	247,04
Caso 2: Randômico	45333	126,93	-	126,93
Caso 3: Randômico, Balanceamento intra-sítio	45333	126,93	-	126,93
Caso 4: Randômico, Balanceamento intra/entre sítios	45333	126,93	-	126,93
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	45333	126,93	72,53	199,46
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	45333	126,93	72,53	199,46
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	45333	126,93	-	126,93
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	45333	126,93	-	126,93

Os dados apresentados permitem concluir que o cenário de uma infraestrutura oferecendo uma média, na escala utilizada neste trabalho, quantidade em recursos de processamento para suporte à execução de uma aplicação com baixa demanda computacional não possui nenhum ganho pela introdução de *cloud bursting*. Os gráficos (e) e (f) na Figura 11 explicitam que os servidores 12 a 15 permaneceram ociosos. A melhor execução, em termos de desempenho e custo, foi obtida com a simples distribuição das demandas submetidas em um sítio entre os Servidores de Processamento do próprio sítio (Caso 2).

Para registro, em termos de custo, faz-se notar que a opção de contrato de nuvem pública por *spot* não onera o custo final, a exemplo de um contrato regular.

6.4.3 Infraestrutura de alta capacidade

A infraestrutura de alta capacidade é composta por uma nuvem federada de três sítios, cada sítio com quatro Servidores de Processamento de 24 *cores* físicos, cada *core* rodando a uma taxa de 100.000 MIPS. Quando aplicada, a nuvem pública possui quatro servidores

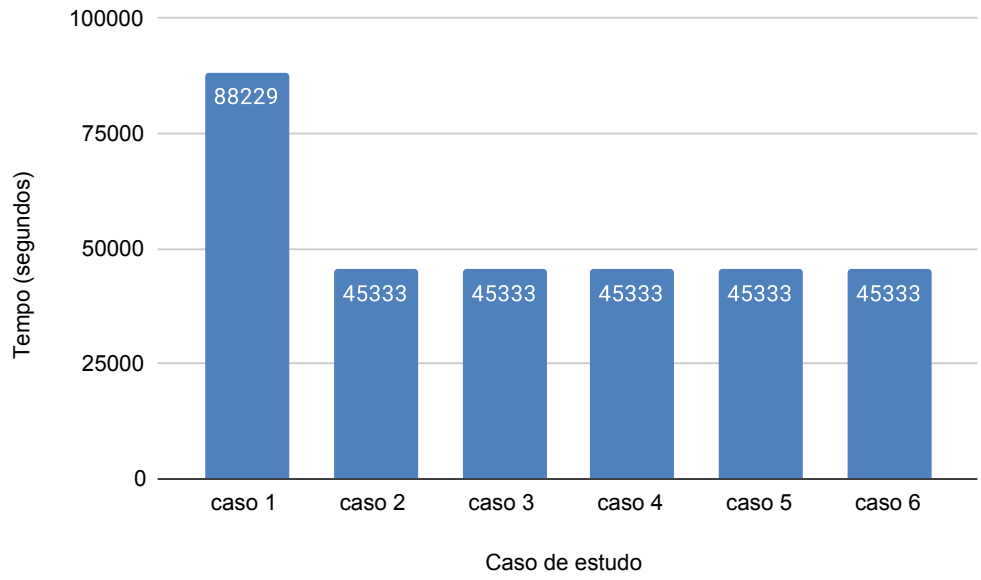
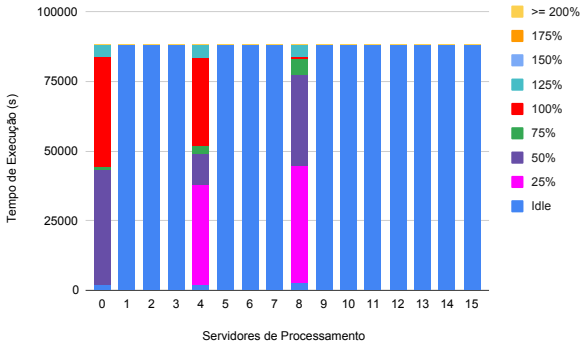


Figura 10 – Tempo de execução de aplicações de baixa demanda computacional em infraestrutura de média capacidade.

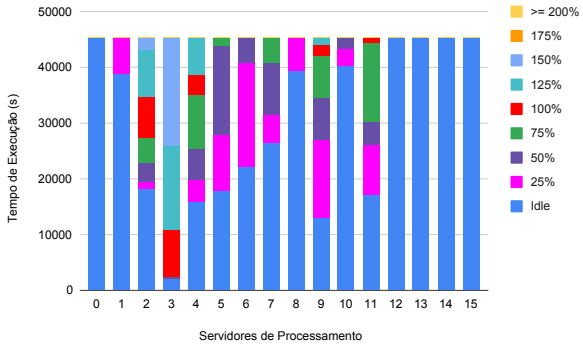
de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*. Na sequência desta subseção, os resultados são discutidos. Antecipa-se que as observações apresentadas na subseção anterior (aplicação com baixa demanda em infraestrutura com média capacidade de processamento) são ressaltadas.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem baixa capacidade computacional, em infraestruturas ofertando alta quantidade de recursos de processamento, encontram-se na Tabela 17 e nos gráficos nas figuras 12 e 13. Na Tabela 17 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 12 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 13 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

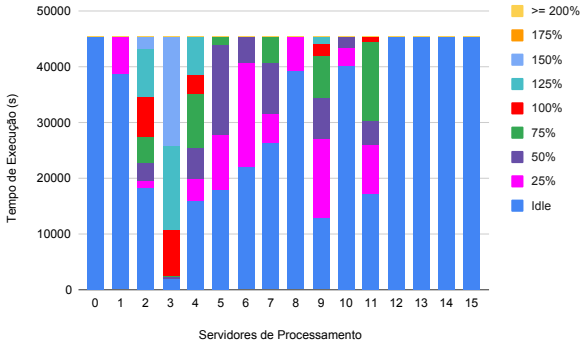
Na Tabela 17 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da



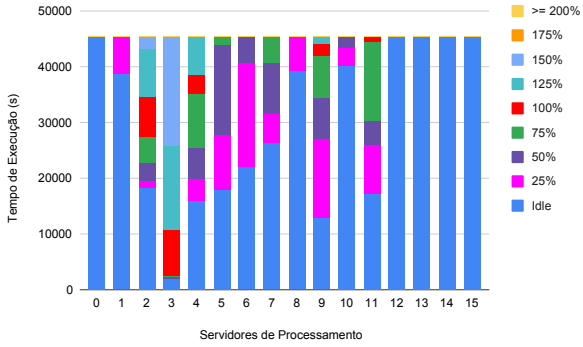
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



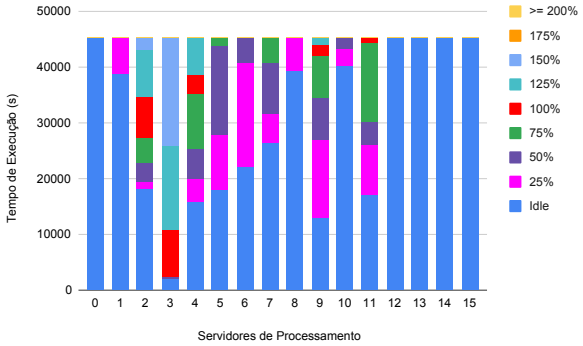
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



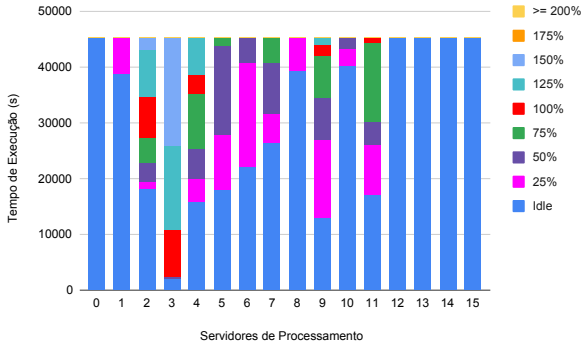
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 11 – Execução de aplicações de baixa demanda computacional em infraestrutura de média capacidade.

tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 17 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de baixa demanda computacional em infraestrutura de alta capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	86272	552,14	-	552,14
Caso 2: Randômico	34071	218,05	-	218,05
Caso 3: Randômico, Balanceamento intra-sítio	34071	218,05	-	218,05
Caso 4: Randômico, Balanceamento intra/entre sítios	34071	218,05	-	218,05
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	34071	218,05	54,51	272,57
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	34071	218,05	54,51	272,57
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	34071	218,05	-	218,05
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	34071	218,05	-	218,05

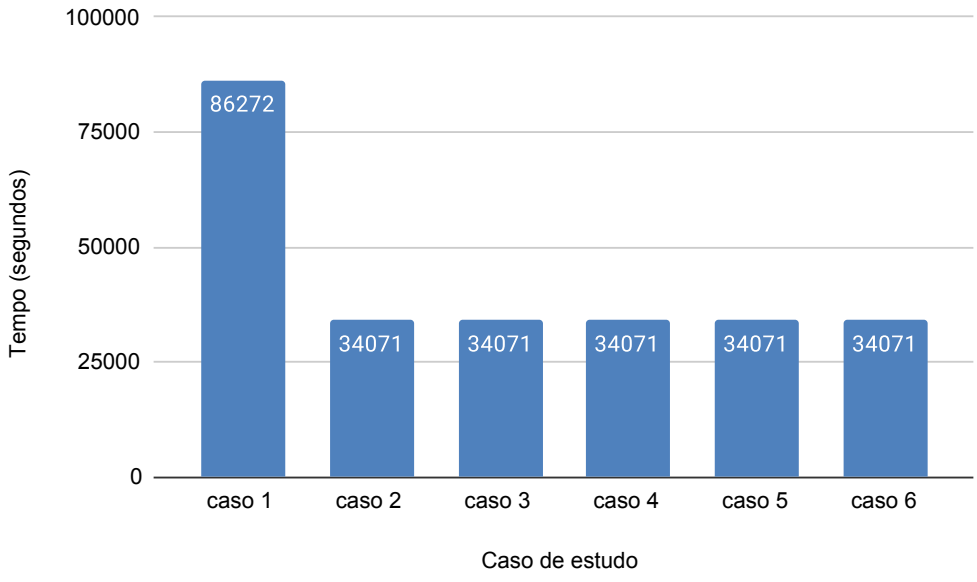
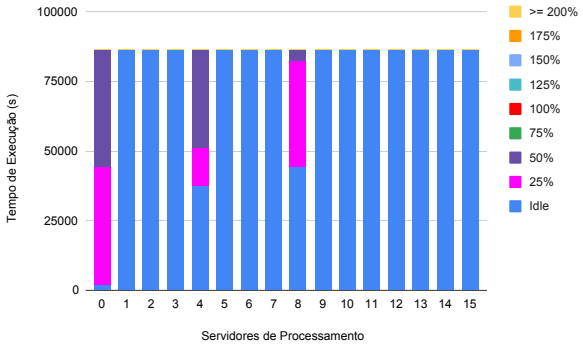
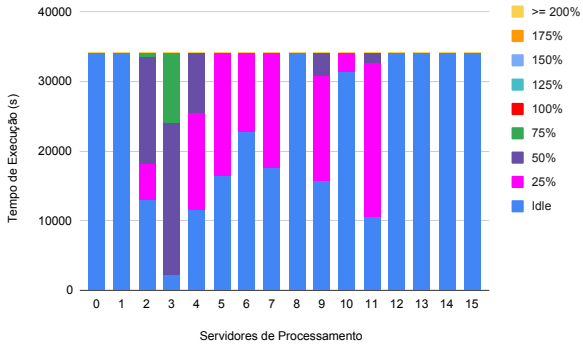


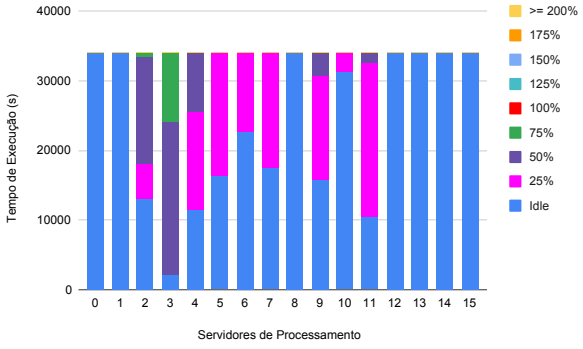
Figura 12 – Tempo de execução de aplicações de baixa demanda computacional em infraestrutura de alta capacidade.



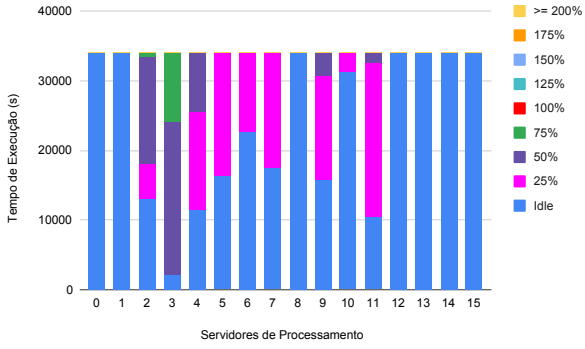
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



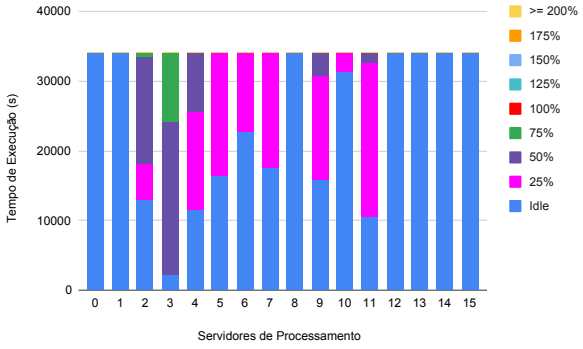
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



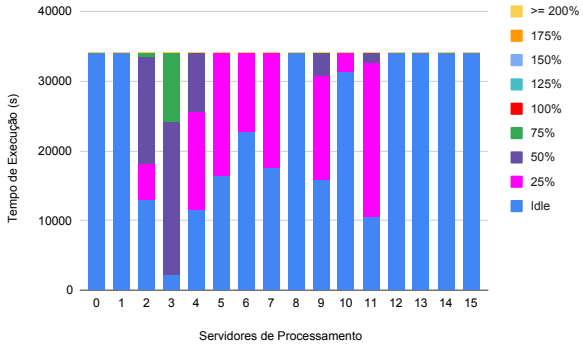
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 13 – Execução de aplicações de baixa demanda computacional em infraestrutura de alta capacidade.

Os dados apresentados permitem concluir que o cenário de uma infraestrutura oferecendo alta quantidade de recursos de processamento para suporte à execução de uma aplicação com baixa demanda computacional não possui nenhum ganho pela introdução de *cloud bursting*. Os gráficos (e) e (f) na Figura 13 explicitam que os servidores 12 a 15 permaneceram ociosos. A melhor execução, em termos de desempenho e custo, foi obtida com a simples distribuição das demandas submetidas em um sítio entre os Servidores de Processamento do próprio sítio (Caso 2).

Para registro, em termos de custo, faz-se notar que a opção de contrato de nuvem pública por *spot* não onera o custo final, a exemplo de um contrato regular.

6.4.4 Comparação: Execução de aplicações com Baixa Demanda

Quando comparadas as ofertas de capacidade de processamento baixa, média e alta para uma situação em que a demanda de processamento das aplicações é baixa, é possível constatar que, apenas no cenário em que a infraestrutura de nuvem federada possui baixa capacidade, obteve-se um ganho de desempenho pelo o uso de *cloud bursting* sobre uma nuvem pública. Este ganho de desempenho foi considerável, cerca de 50%, em relação ao melhor desempenho obtido executando apenas recursos da nuvem federada, com acréscimo de 35% no custo, considerando o uso de *cloud bursting* por *spot* (cf. Tabela 15). Considerando apenas o uso da nuvem federada, o aumento da capacidade de processamento da infraestrutura diminuiu o tempo total de execução. Com a infraestrutura de média capacidade, o tempo obtido foi cerca de 60% menor e com a de maior capacidade, 45%.

Em relação ao custo, a infraestrutura de baixa capacidade utilizando *cloud bursting* com *spot* (Caso 5), tem um custo que corresponde à, aproximadamente, 1/3 do valor da execução da aplicação na infraestrutura federada com média capacidade, sendo que seu desempenho foi superior. Estes dados permitem considerar a opção em utilizar uma infraestrutura oferecendo baixa capacidade de processamento, compensada pelo uso de *cloud bursting*.

Note-se ainda que uma infraestrutura para a nuvem federada com alta capacidade de processamento ofereceu um ganho de, no melhor caso, cerca de 10% em relação ao Caso 5 na infraestrutura de baixa capacidade, mas com valor final cerca de 5 vezes maior.

6.5 Média Demanda Computacional

O estudo documentado nesta seção considera a submissão de aplicações de média demanda computacional. O cenário é descrito por 48 usuários, 16 por sítio, submetendo 48 aplicações Sipht, Ligo e Galactic. As aplicações são distribuídas uniformemente entre os sítios da nuvem federada. As avaliações foram realizadas nuvens federadas com infraestruturas oferecendo baixa, média e alta capacidade de processamento.

6.5.1 Infraestrutura de baixa capacidade

A infraestrutura de baixa capacidade neste cenário é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de quatro *cores* físicos, rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem média capacidade computacional, em infraestruturas ofertando baixa capacidade de processamento, encontram-se na Tabela 18 e nos gráficos nas figuras 14 e 15. Na Tabela 18 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 14 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 15 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 18 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 18 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de média demanda computacional em infraestrutura de baixa capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	910061	364,02	-	364,02
Caso 2: Randômico	277796	111,12	-	111,12
Caso 3: Randômico, Balanceamento intra-sítio	1633978	653,59	-	653,59
Caso 4: Randômico, Balanceamento intra/entre sítios	1866681	746,67	-	746,67
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	255776	102,31	409,24	511,55
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	162825	65,13	260,52	325,65
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	255776	102,31	341,39	443,70
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	162825	65,13	228,13	293,26

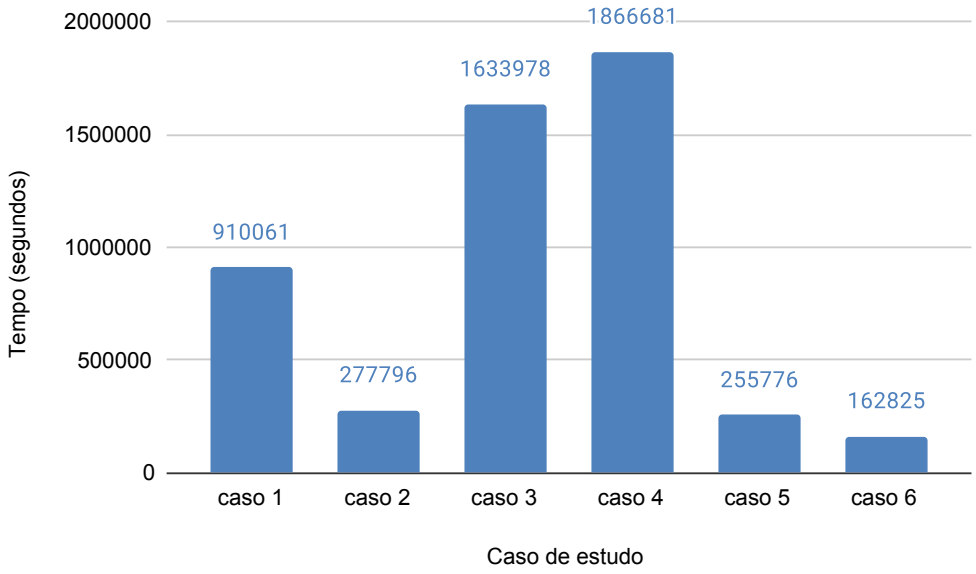
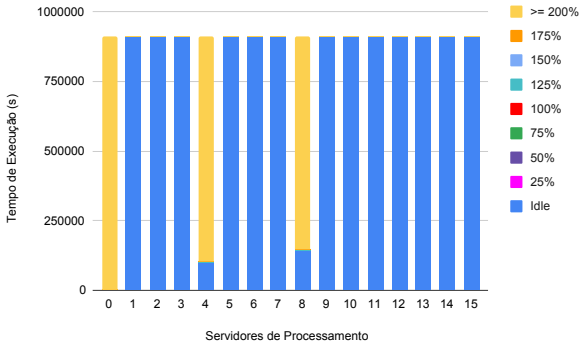
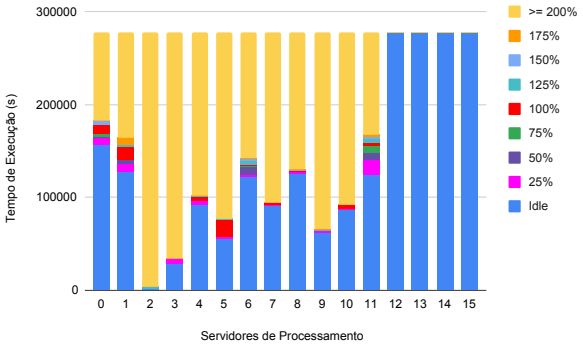


Figura 14 – Tempo de execução de aplicações de média demanda computacional em infraestrutura de baixa capacidade.

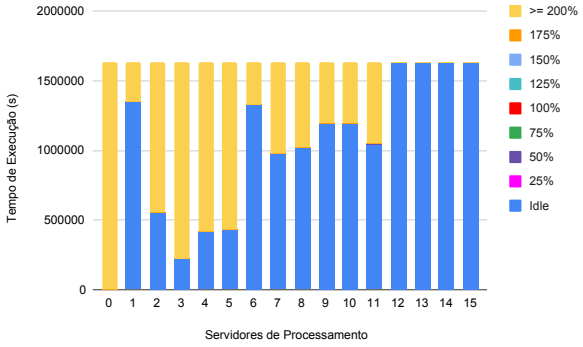
Os dados apresentados permitem concluir que o cenário de uma infraestrutura com capacidade média de processamento é capaz de suportar localmente a submissão de aplicações com baixa demanda computacional. Os resultados mostraram que a opção de realizar escalonamento randômico, Caso 3, apenas no sítio em que as aplicações são submetidas já oferece os melhores resultados de desempenho e custo. Qualquer introdução de migração, mesmo intra-sítio (como o Caso 4) adiciona sobrecustos à execução.



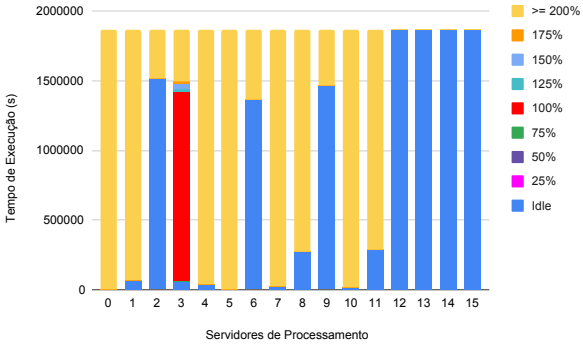
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



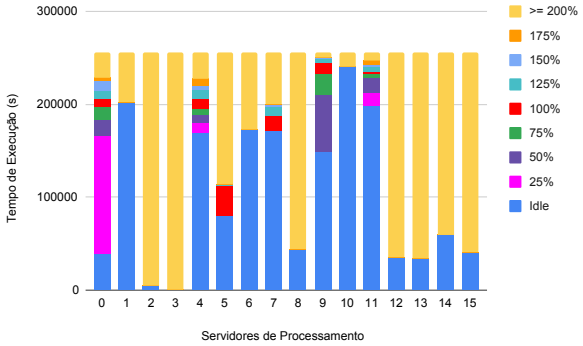
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



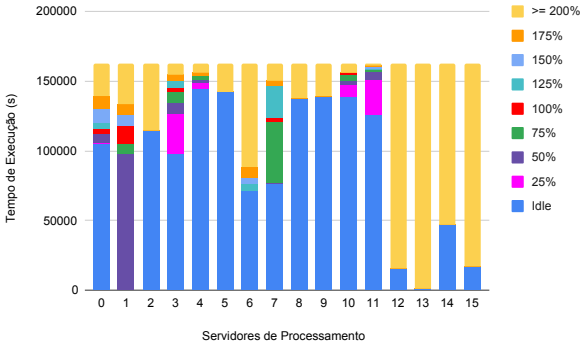
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 15 – Execução de aplicações de média demanda computacional em infraestrutura de baixa capacidade.

6.5.2 Infraestrutura de média capacidade

A infraestrutura de média capacidade é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de 12 *cores* físicos rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem média capacidade computacional, em infraestruturas ofertando média quantidade de recursos de processamento, encontram-se na Tabela 19 e nos gráficos nas Figuras 16 e 17. Na Tabela 19 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 16 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 17 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 19 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 19 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de média demanda computacional em infraestrutura de média capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	256828	719,12	-	719,12
Caso 2: Randômico	91816	257,08	-	257,08
Caso 3: Randômico, Balanceamento intra-sítio	141409	395,95	-	395,95
Caso 4: Randômico, Balanceamento intra/entre sítios	164675	461,09	-	461,09
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	94660	265,05	151,46	416,50
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	64032	179,29	102,45	281,74
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	94660	265,05	133,00	398,04
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	64032	179,29	90,40	269,69

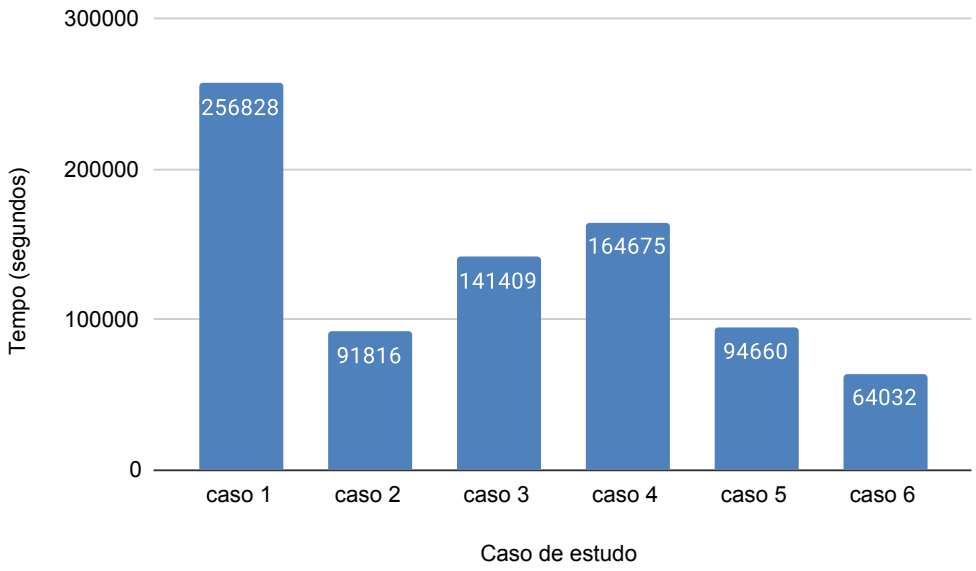
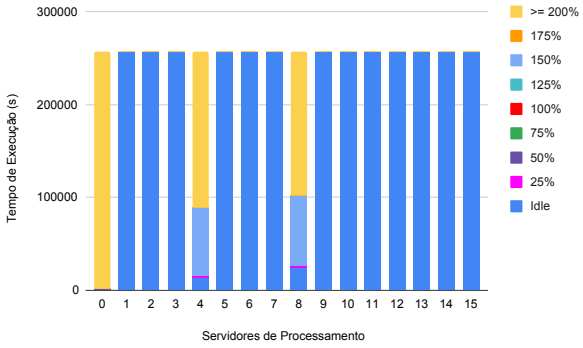
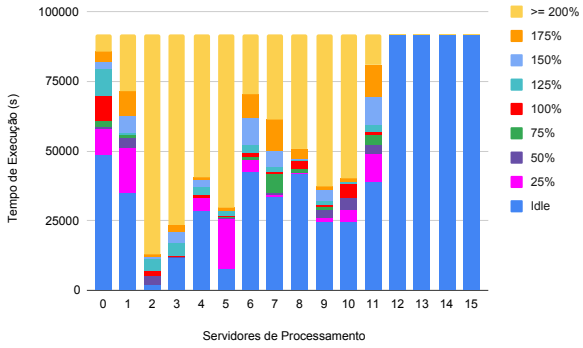


Figura 16 – Tempo de execução de aplicações de média demanda computacional em infraestrutura de média capacidade.

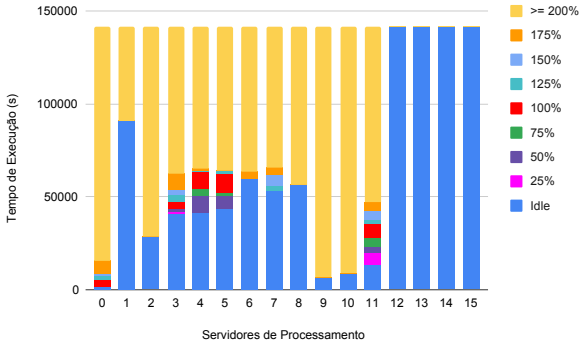
Os resultados obtidos na simulação de uma carga de médio tamanho, sobre uma infraestrutura também de média capacidade mostra que o uso de *cloud bursting* pode oferecer melhores índices de desempenho, em termos de tempo de processamento, com um custo ligeiramente superior. Comparando os resultados apresentados pelo Caso 2 e pelo Caso 6 (com *spot*): o *cloud bursting* ofereceu um desempenho superior, reduzindo o tempo em cerca de 40%, ao passo que a majoração no custo foi na ordem de 10%.



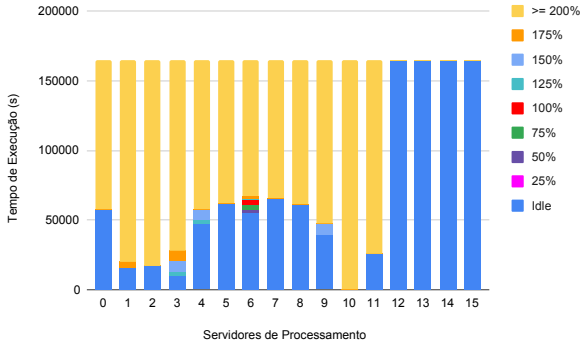
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



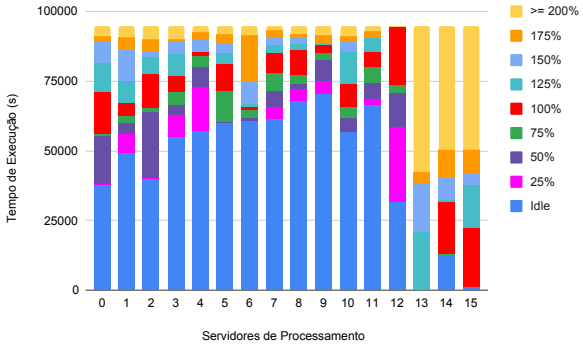
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



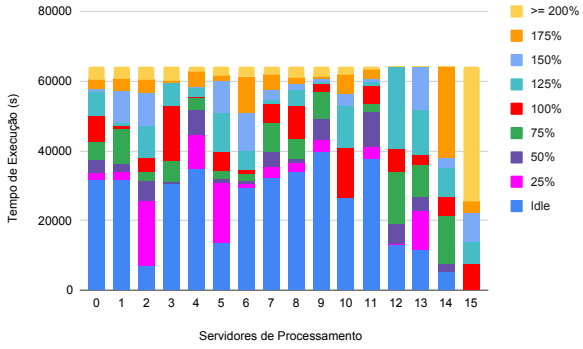
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 17 – Execução de aplicações de média demanda computacional em infraestrutura de média capacidade.

6.5.3 Infraestrutura de alta capacidade

A infraestrutura de alta capacidade é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de 24 *cores* físicos rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem média capacidade computacional, em infraestruturas ofertando alta quantidade de recursos de processamento, encontram-se na Tabela 20 e nos gráficos nas figuras 18 e 19. Na Tabela 20 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 18 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 19 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 20 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 20 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de média demanda computacional em infraestrutura de alta capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	127641	816,90	-	816,90
Caso 2: Randômico	54145	346,33	-	346,33
Caso 3: Randômico, Balanceamento intra-sítio	52059	333,18	-	333,18
Caso 4: Randômico, Balanceamento intra/entre sítios	53519	342,52	-	342,52
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	53519	342,52	85,63	428,15
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	53050	339,52	84,88	424,40
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	53519	342,52	-	342,52
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	53050	339,52	-	339,52

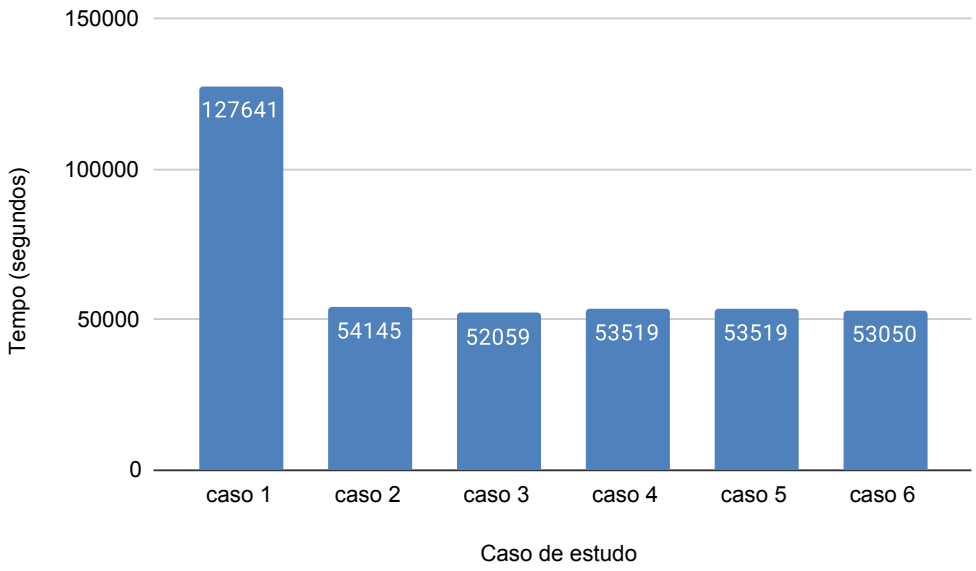
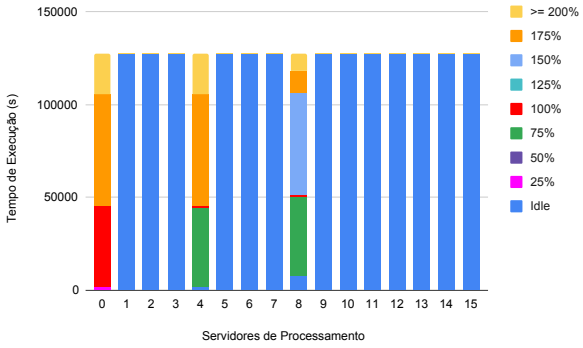
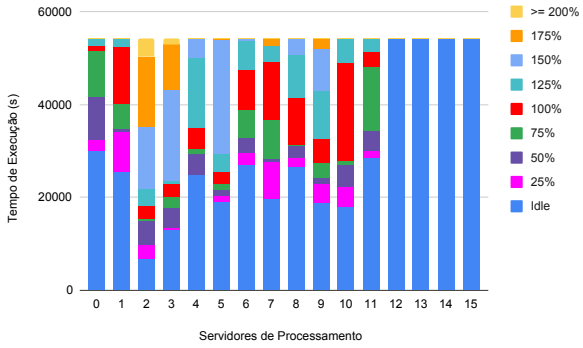


Figura 18 – Tempo de execução de aplicações de média demanda computacional em infraestrutura de alta capacidade.

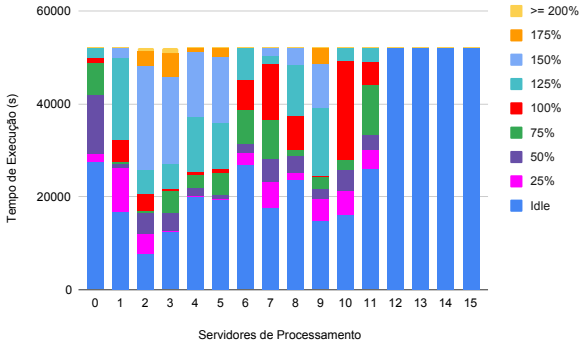
Os resultados coletados neste cenário mostram que todos os escalonamentos são equivalentes, entre si, em termos de desempenho e custo. Marginalmente, o Caso 3, onde todo o escalonamento é realizado localmente ao sítio onde as aplicações são submetidas, apresenta um melhor desempenho. O uso de *cloud bursting* não é capaz de melhorar este desempenho, tão pouco afetar significativamente o custo quando o contrato se dá por *spot*. Por outro lado, o contrato regular de recursos em uma nuvem pública onera



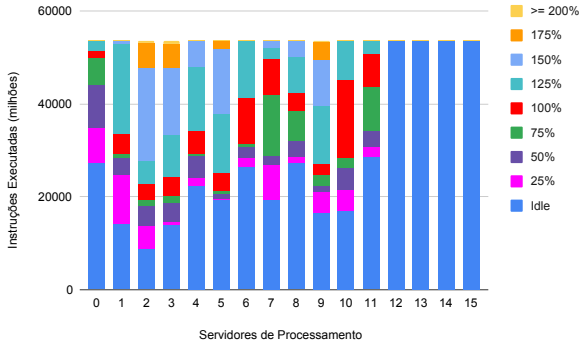
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



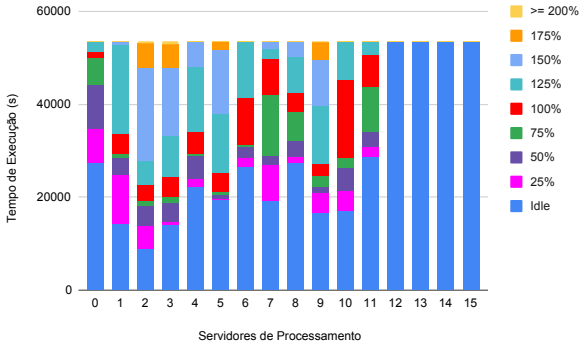
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



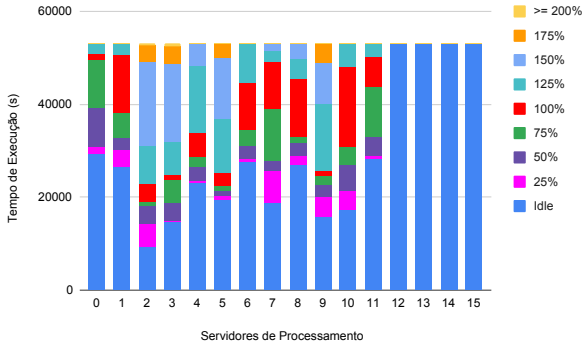
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 19 – Execução de aplicações de média demanda computacional em infraestrutura de alta capacidade.

desnecessariamente a execução.

6.5.4 Comparação: Execução de aplicações com Média Demanda

Quando comparadas as ofertas de capacidade de processamento baixa, média e alta para uma situação em que a demanda de processamento das aplicações é média, é possível constatar que, apenas no cenário em que a infraestrutura de nuvem federada possui baixa capacidade, obteve-se um ganho de desempenho pelo o uso de *cloud bursting* sobre uma nuvem pública. Este ganho de desempenho foi considerável, cerca de 40%, em relação ao melhor desempenho obtido executando apenas recursos da nuvem federada, havendo, no entanto, um acréscimo de pouco mais de 100% no custo operacional, mesmo considerando o uso de *cloud bursting* por *spot* (cf. Tabela 18).

Um resultado interessante observado é quando da submissão do conjunto de aplicações sobre uma infraestrutura oferecendo média capacidade de processamento (Tabela 19). Neste caso, a opção por executar a aplicação utilizando *cloud bursting* (Caso 6) ofereceu cerca de 30% a mais de desempenho por um custo operacional equivalente ao melhor caso utilizando apenas a nuvem federada (Caso 2), com custo equivalente.

Note-se ainda que em uma infraestrutura para a nuvem federada com alta capacidade de processamento, as aplicações com média demanda computacional não acionaram o uso da nuvem pública.

6.6 Alta Demanda Computacional

O último caso de estudo considera a execução de um elevado número de submissões à infraestrutura de nuvem. Neste cenário, 192 usuários, 64 por sítio, submetem 192 aplicações Sipht, Ligo e Galactic. As aplicações são distribuídas uniformemente entre os sítios da nuvem federada. As avaliações foram realizadas em nuvens federadas com infraestruturas oferecendo baixa, média e alta capacidade de processamento.

6.6.1 Infraestrutura de baixa capacidade

A infraestrutura de baixa capacidade neste cenário é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de quatro *cores* físicos rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

As informações sobre os dados coletados no experimento realizado com aplicações que requerem baixa capacidade computacional, em infraestruturas ofertando média capacidade de processamento, encontram-se na Tabela 21 e nos gráficos nas figuras 20 e 21. Na Tabela 21 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 20 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 21 o tempo total

de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 21 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 21 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de alta demanda computacional em infraestrutura de baixa capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	1927779	771,11	-	771,11
Caso 2: Randômico	1206387	482,55	-	482,55
Caso 3: Randômico, Balanceamento intra-sítio	2200032	880,01	-	880,01
Caso 4: Randômico, Balanceamento intra/entre sítios	2200020	880,01	-	880,01
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	1850000	740,00	2.960,00	3.700,00
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	1242330	496,93	1.987,73	2.484,66
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	1850000	740,00	2.949,99	3.696,98
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	1242330	496,93	1.957,04	2.453,97

Os resultados coletados neste experimento mostram que o melhor desempenho foi atingido utilizando os recursos locais ao sítio onde as aplicações foram submetidas, sem

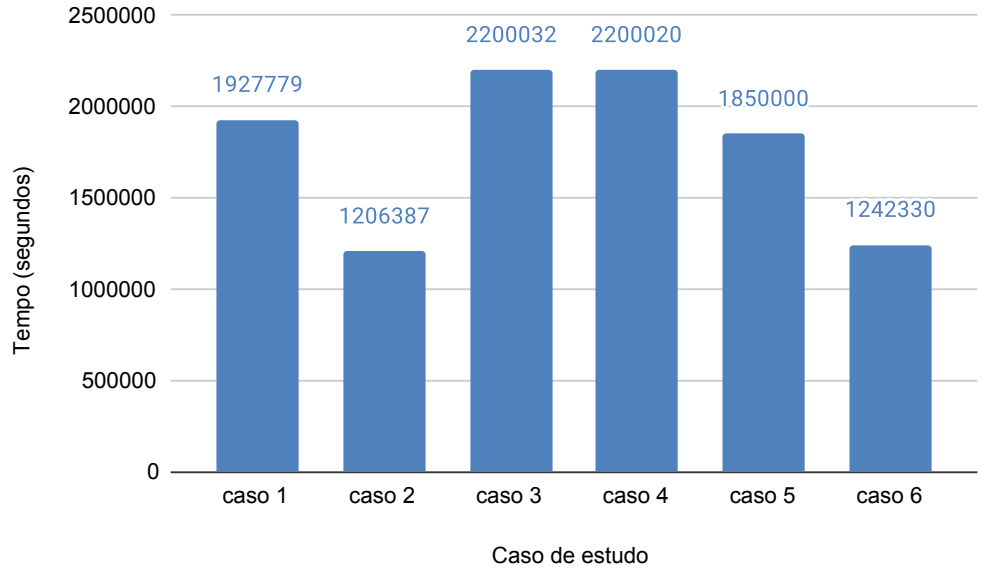


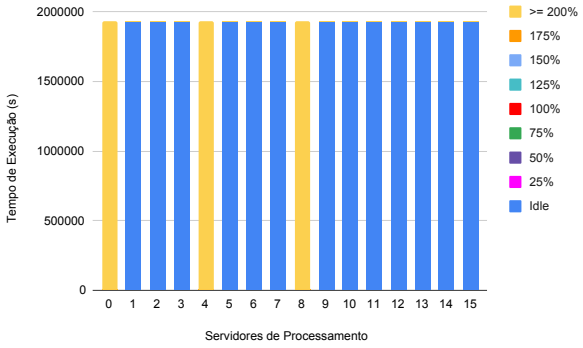
Figura 20 – Tempo de execução de aplicações de alta demanda computacional em infraestrutura de baixa capacidade.

introdução de nenhum mecanismo de balanceamento de carga. Em particular, o Caso 2 foi o que apresentou o melhor desempenho, tanto em relação ao tempo total de processamento quanto ao custo. O uso de *cloud bursting* eleva consideravelmente o custo total da execução, passando de US$ 482,55 (Caso 2) para US$ 2.453,97 (Caso 6, com *spot*), com ainda um pequeno decréscimo de 3% no tempo de execução. Observando o traço de utilização dos servidores nos gráficos da Figura 21, verifica-se que grande parte do processamento, quando recursos de uma nuvem pública estão disponíveis (Servidores de Processamento 12 a 15 nos gráficos, (e) Caso 5 e (f) Caso 6), migra e permanece em execução sobre os equipamentos alugados.

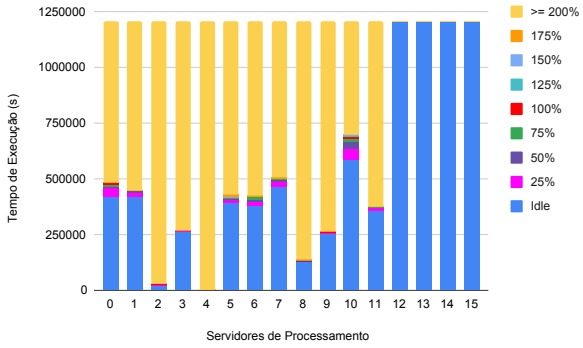
6.6.2 Infraestrutura de média capacidade

A infraestrutura de média capacidade neste cenário é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de 12 *cores* físicos rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente à 100.000 MIPS por *core*.

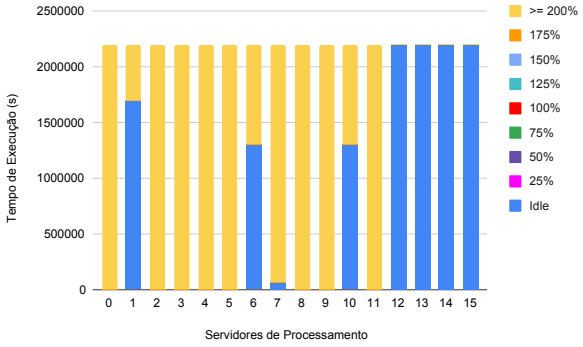
As informações sobre os dados coletados no experimento realizado com aplicações que requerem baixa capacidade computacional, em infraestruturas ofertando média capacidade de processamento, encontram-se na Tabela 22 e nos gráficos nas figuras 22 e 23. Na Tabela 22 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 22 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 23 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização,



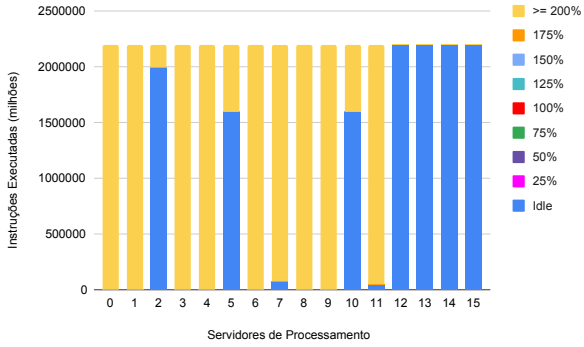
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



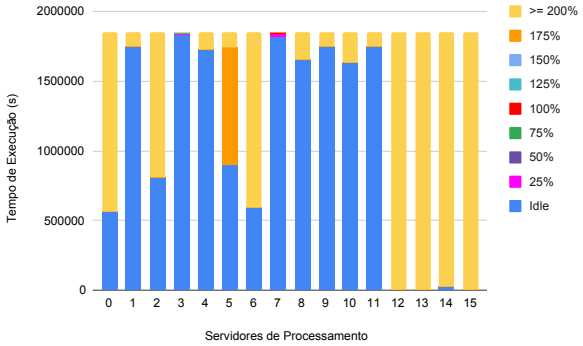
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



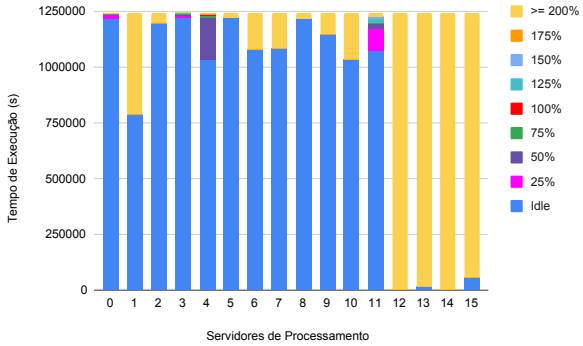
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 21 – Execução de aplicações de alta demanda computacional em infraestrutura de baixa capacidade.

sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 22 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 22 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de alta demanda computacional em infraestrutura de média capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	1378704	3.860,37	-	3.860,37
Caso 2: Randômico	327892	918,10	-	918,10
Caso 3: Randômico, Balanceamento intra-sítio	1900026	5.320,07	-	5.320,07
Caso 4: Randômico, Balanceamento intra/entre sítios	2200026	6.160,07	-	6.160,07
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	1439985	4.031,96	2.303,98	6.335,93
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	1180829	3.306,32	1.889,33	5.195,65
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	1439985	4.031,96	2.142,64	6.174,60
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	1180829	3.306,32	1.766,38	5.072,70

O comportamento geral da execução de uma grande demanda sobre uma configuração de média capacidade de processamento apresentou um comportamento similar ao apresentado por uma infraestrutura de baixa capacidade. Grande parte do processamento foi executado sobre recursos de processamento na nuvem pública, quando disponíveis,

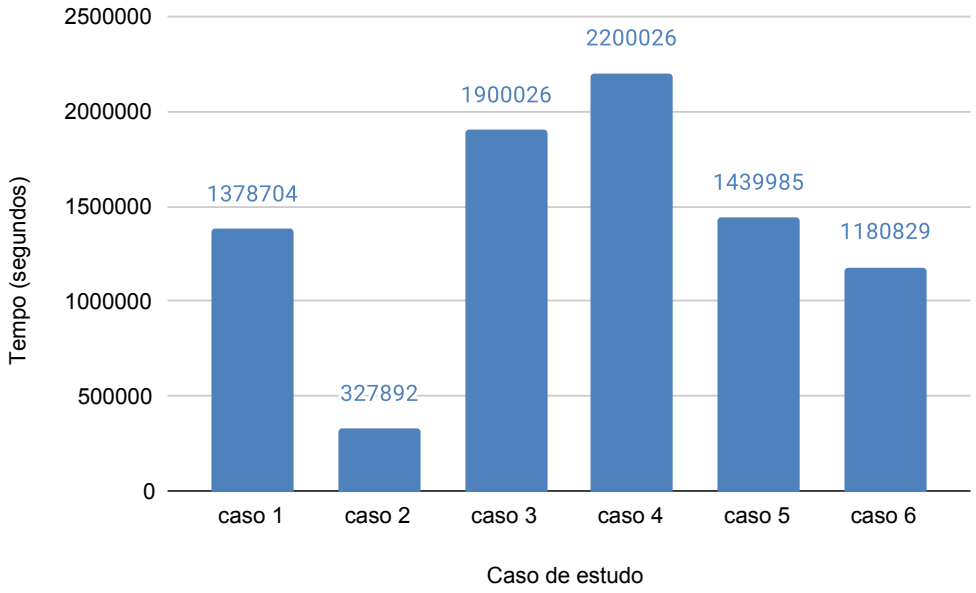


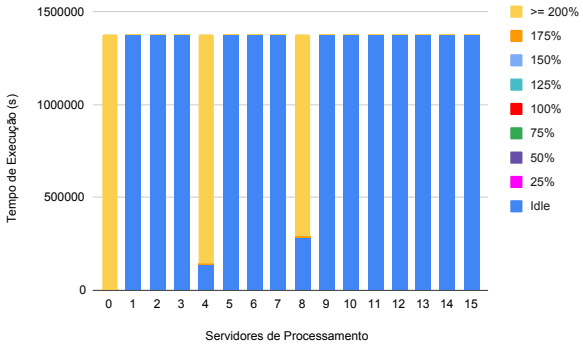
Figura 22 – Tempo de execução de aplicações de alta demanda computacional em infraestrutura de média capacidade.

conforme visualiza-se nos Caso 5 e Caso 6 presentes nos gráficos (e) e (f) na Figura 23. Em termos de desempenho e custo, em função da capacidade presente em cada sítio, o escalonamento realizado no Caso 2 foi o melhor. Nenhuma estratégia de balanceamento de carga melhorou o desempenho da simples distribuição da carga computacional no lançamento das aplicações.

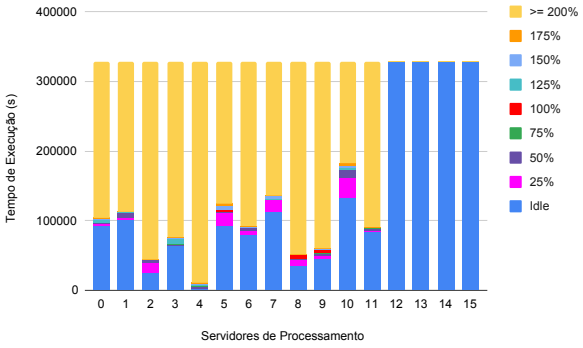
6.6.3 Infraestrutura de alta capacidade

A infraestrutura de alta capacidade neste cenário é composta por uma nuvem federada de três sítios, cada sítio oferecendo quatro Servidores de Processamento dotados de 24 *cores* físicos rodando a uma taxa de 100.000 MIPS cada *core*. Quando aplicada, a nuvem pública possui quatro servidores de 48 *cores*, rodando igualmente a 100.000 MIPS por *core*.

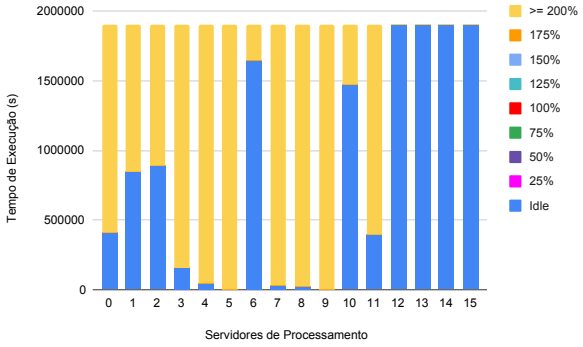
As informações sobre os dados coletados no experimento realizado com aplicações que requerem alta capacidade computacional, em infraestruturas ofertando alta capacidade de processamento, encontram-se na Tabela 23 e nos gráficos nas figuras 24 e 25. Na Tabela 23 são apresentados o tempo de execução observado para a execução das submissões das aplicações e seus respectivos custos, sendo que na Figura 24 é apresentada uma visualização dos tempos de execução em cada caso. Nos gráficos da Figura 25 o tempo total de execução das aplicações permite individualizar, para cada servidor de processamento, a proporção do tempo total de execução que passou com uma determinada taxa de utilização, sendo até 100% considerado atendendo a demanda e acima deste valor, sobrecarregado. As informações referentes aos servidores na nuvem pública (servidores 12 a 15) são relevantes apenas nos casos 5 e 6. Em todos os casos, deve ser considerado que existem três sítios, cada sítio com quatro servidores (servidores 0 a 3 no primeiro sítio, 4 a 7 no



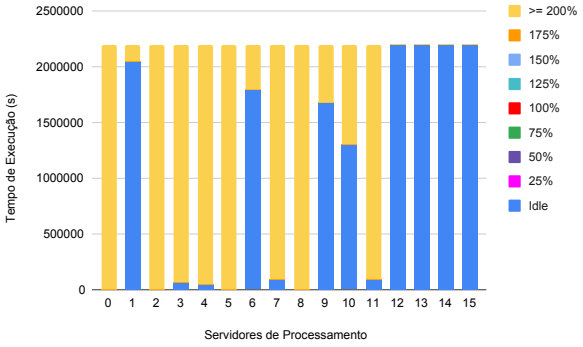
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



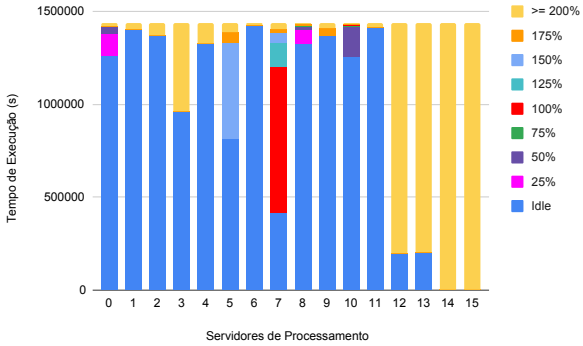
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



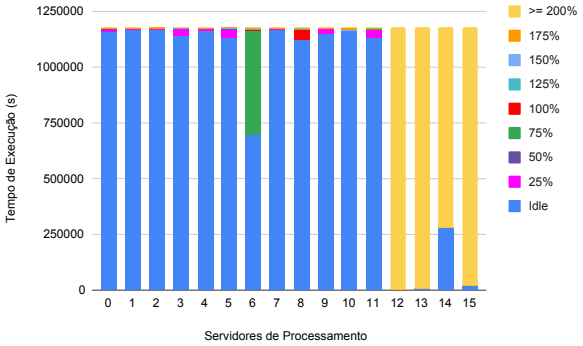
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 23 – Execução de aplicações de alta demanda computacional em infraestrutura de média capacidade.

segundo e 8 a 11 no terceiro). É importante observar ainda que o Caso 1 apresenta o desempenho no caso extremo, em que apenas um servidor por sítio suporta a execução de todas as aplicações submetidas naquele sítio.

Na Tabela 23 o tempo é apresentado em segundos e corresponde ao tempo registrado para o final da última execução de tarefa. Os casos de escalonamento de 1 a 4 não utilizam *cloud bursting* e os custos de utilização, informados em dólar, correspondem a tarifação de uso da infraestrutura da nuvem federada (*NF*) unicamente. As duas seções seguintes da tabela apresentam custos de utilização da nuvem pública (**NP**) segundo dois modelos de contrato de locação: considerando a locação fixa dos recursos e utilizando pagamento por uso, referenciado como ***spot***. Nos casos em que é realizado *cloud bursting*, os custos são apresentados de forma decomposta, informando separadamente o valor da tarifação da nuvem federada, da nuvem pública e o somatório final.

Tabela 23 – Tempo total, em segundos, e custos, em US$ para a execução de aplicações de alta demanda computacional em infraestrutura de alta capacidade. **NF**: Nuvem Federada, **NP**: Nuvem Pública.

		Custos		
Escalonador	**Duração (s)**	**NF**	**NP**	**Total**
Caso 1: Fixo	558517	3.574,51	-	3.574,51
Caso 2: Randômico	159964	1.023,77	-	1.023,77
Caso 3: Randômico, Balanceamento intra-sítio	948082	6.067,72	-	6.067,72
Caso 4: Randômico, Balanceamento intra/entre sítios	1899994	12.159,96	-	12.159,96
Regular				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	1170824	7.493,27	1.873,32	9.366,59
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	785599	5.027,83	1.256,96	6.284,79
Spot				
Caso 5: Randômico, Balanceamento intra/entre sítios, *cloud bursting*	1170824	7.493,27	1.640,18	9.133,45
Caso 6: Randômico, Balanceamento intra-sítio, *cloud bursting*	785599	5.027,83	1.164,51	6.192,34

Neste cenário, de alto investimento no poder de processamento de cada sítio, o melhor escalonamento foi atingido pela simples distribuição da carga computacional entre os Servidores de Processamento (Caso 2), sem a realização de qualquer operação de balanceamento de carga. Também foi observada a migração para os recursos de processamento na nuvem pública de boa parte da carga computacional gerada pelas aplicações, não refletindo esta estratégia de escalonamento nem em melhor desempenho nem em melhor custo operacional.

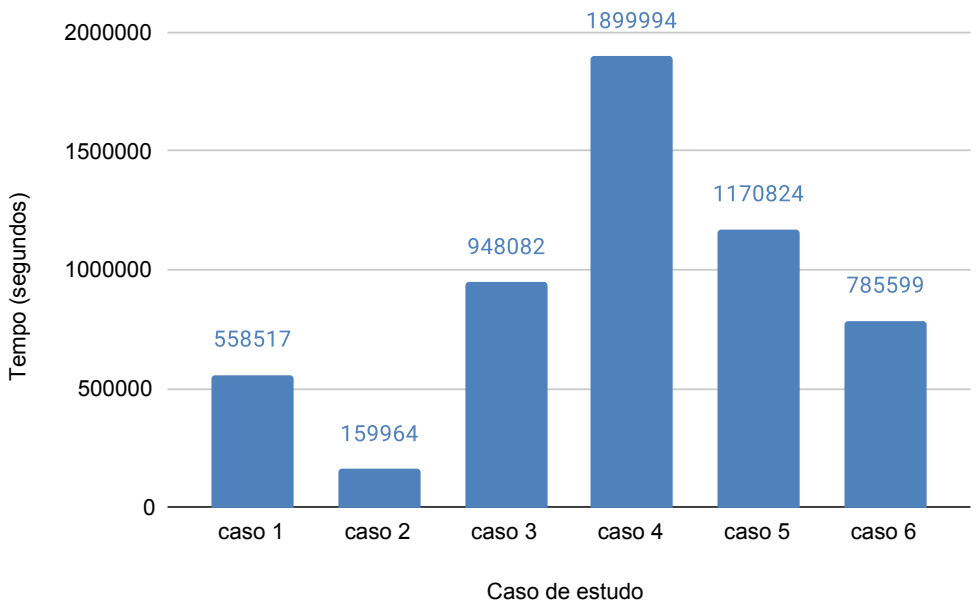


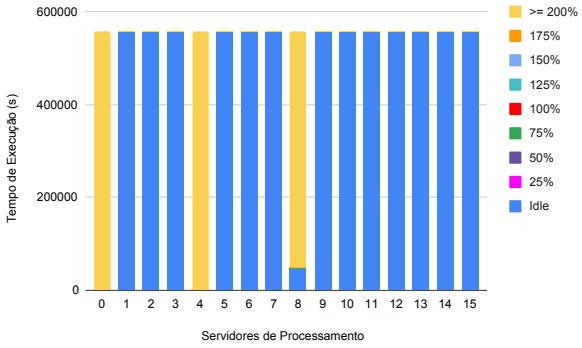
Figura 24 – Tempo de execução de aplicações de alta demanda computacional em infraestrutura de alta capacidade.

6.6.4 Comparação: Execução de Aplicações com Alta Demanda

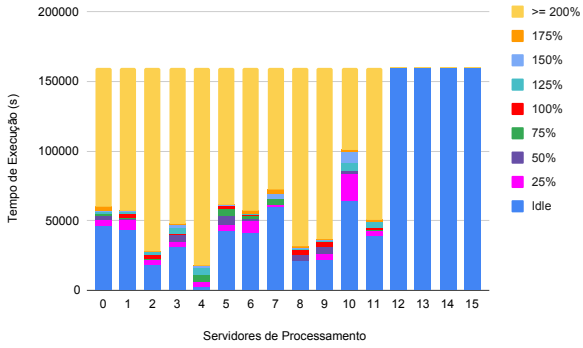
Em um cenário no qual a alta demanda de recursos computacionais é expressa em termos de número de tarefas e não a uma maior carga de processamento por tarefas, observa-se que o uso de *cloud bursting* não foi capaz de melhorar os índices de desempenho. Na verdade, os melhores desempenhos foram obtidos pela simples alocação randômica das aplicações sobre os servidores do próprio sítio de lançamento (Caso 2). O segundo melhor desempenho foi apresentado pelo escalonamento realizado pelo Caso 6, ainda que apresentando tempos de execução e custos bastante superiores.

6.7 Discussão

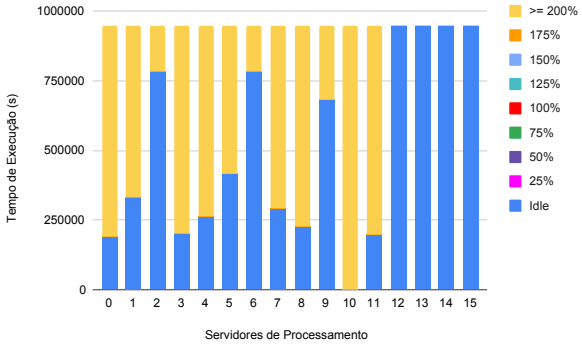
Os casos de estudo apresentados neste capítulo representam cenários possíveis de utilização de nuvens em ambiente acadêmico. Foram selecionadas aplicações científicas descritas por *workflows*, no contexto do projeto Pegasus, na expectativa de representar demandas de processamento no meio acadêmico. As infraestruturas apresentadas foram concebidas a partir de configurações realizáveis a partir das opções de mercado, tanto no que diz respeito a precificação da infraestrutura federada (equipamentos adquiridos) quanto locados em nuvem pública. Os algoritmos de escalonamento utilizados não são inovadores. Pelo contrário, a simplicidade das estratégias de escalonamento visa facilitar a interpretação dos resultados numéricos, indicando o interesse em adotar estratégias de implantação de nuvens em meio acadêmico utilizando em complementação à infraestrutura física construída nas instituições federadas, um suporte computacional em nuvens públicas.



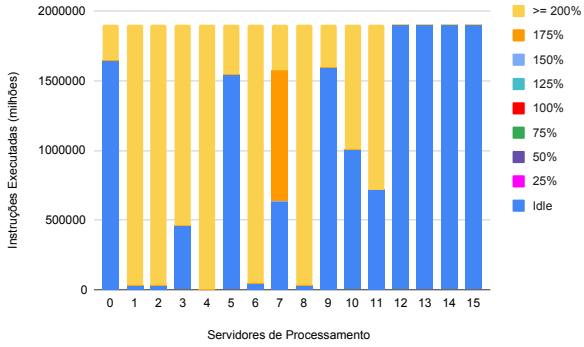
(a) Caso 1: Escalonamento Fixo - Toda execução submetida em um sítio é suportada em um único Servidor de Processamento.



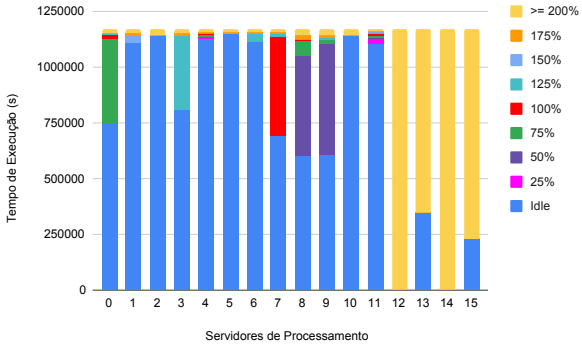
(b) Caso 2: Escalonamento Randômico - Seleção randômica de servidor no sítio.



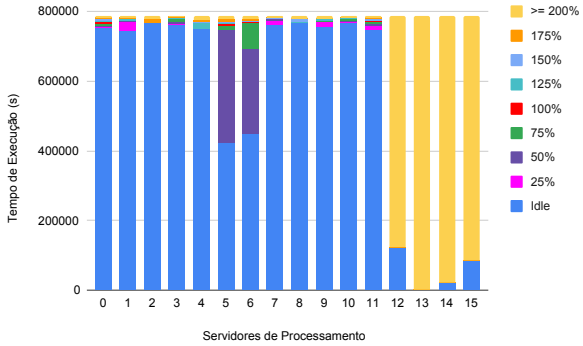
(c) Caso 3: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio.



(d) Caso 4: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios.



(e) Caso 5: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra e entre sítios com *cloud bursting*.



(f) Caso 6: Escalonamento Randômico - Seleção randômica de servidor no sítio e balanceamento de carga intra-sítio com *cloud bursting*.

Figura 25 – Execução de aplicações de alta demanda computacional em infraestrutura de alta capacidade.

A principal constatação é que a modalidade de precificação de *spot* deve ser considerada, caso ofertada pelo provedor de recursos em nuvem. Foi observado, nos experimentos, custos geralmente inferiores realizando pagamento por *spot* ao obtido com um contrato regular de reserva. Ainda assim, o uso de nuvem pública por *spot* justificou-se apenas nos casos em que a demanda computacional foi baixa ou média, em geral, considerando as diferentes configurações de infraestrutura da nuvem federada.

A observação sobre as estratégias de escalonamento também permite verificar que o balanceamento de carga intra e entre sítios é um recurso interessante a ser explorado, havendo potencial para obtenção de ganhos de desempenho. Uma modelagem mais realista dos *thresholds* aplicados para disparo das operações de escalonamento e dos custos de comunicação poderá oferecer resultados mais precisos. No entanto, os dados coletados permitem realizar uma análise sobre a escalabilidade das infraestruturas propostas em relação ao crescimento da demanda computacional das cargas submetidas.

Na Tabela 24 os dados apresentados informam, para diferentes infraestruturas da nuvem federada, *Thin*, *Medium* e *Large*, o tempo médio obtido para execução de cada uma das aplicações Sipht, LIGO e Galactic submetidas e os custos da execução de toda a carga submetida, explicitando os custos na nuvem federada e na nuvem pública. A informação de tempo médio de execução por aplicação submetida indica a expectativa de tempo de espera para que cada usuário obtenha o resultado da computação submetida. A tabela ainda estratifica as submissões em termos da demanda computacional gerada, Baixa, Média ou Alta. Nos dados apresentados, são considerados apenas os casos 4 e 5 de escalonamento, que se referem ao emprego do balanceamento de carga intra e entre sítios (Caso 4) e ao uso do balanceamento de carga intra e entre sítios, utilizando *cloud bursting* em uma nuvem pública como apoio (Caso 5).

A partir dos dados apresentados nesta tabela, considerando somente o uso dos recursos da nuvem federada, escalonamento Caso 4, é possível constatar que, para aplicações com pouca demanda computacional, o incremento da infraestrutura, passando de uma infraestrutura *Thin* para uma *Medium*, resulta em uma melhora considerável de desempenho. Porém, o custo de execução passa a ser bastante expressivo, pois, ao passo que o tempo de execução médio das submissões cai para valores próximos à metade do tempo inicial, os custos de execução aumentam na ordem de, aproximadamente, quatro vezes. No entanto, com o uso de uma infraestrutura de nuvem pública, por meio da estratégia de escalonamento Caso 5, os tempos médios de execução são bastante atraentes, com um custo de execução para o conjunto de submissões ligeiramente superior ao custo obtido utilizando apenas a infraestrutura federada.

Tabela 24 – Tempos médios de execução dos *workflows* e seus custos. **CF**: Custo da Infraestrutura Federada, **CN**: Custo da Nuvem Pública.

		Tempo médio (s)					
Infra.	**Esc.**	**Sipht**	**LIGO**	**Galactic**	**CF**	**CN**	**Custo Total**
Baixa demanda computacional							
Thin	Caso 4	73386	51772	15437	34,70	-	34,70
Medium	Caso 4	39102	25622	9906	126,93	-	126,93
Thin	Caso 5	34222	32042	12384	15,20	26,75	41,95
Média demanda computacional							
Thin	Caso 4	1610784	1096387	385192	746,67	-	746,67
Medium	Caso 4	128353	102779	44195	461,09	-	461,09
Thin	Caso 5	165145	149767	54769	102,31	341,39	433,70
Medium	Caso 5	66547	36619	14103	265,05	133,00	398,04
Large	Caso 4	40487	31099	13152	342,52	-	342,52
Alta demanda computacional							
Medium	Caso 4	1954715	2072887	740944	6.160,07	-	6.160,07
Medium	Caso 5	988912	648929	78300	4.031,96	2.142,64	6.164,60
Large	Caso 4	1310498	1031711	379766	12.159,96	-	12.159,96
Large	Caso 5	712883	361170	41562	7.493,27	1.640,18	9.133,45

No caso em que a demanda de processamento é considerada média, o escalonamento Caso 4 indica ganhos de desempenho em ordens de grandeza bastante importantes à medida em que aumenta a capacidade da infraestrutura oferecida. Os custos computacionais para o conjunto das submissões realizadas também decresce, embora em uma relação inferior à percebida nos ganhos de desempenho. A exploração de uma nuvem pública, escalonamento Caso 5, em apoio às infraestruturas *Thin* e *Medium*, se mostra como alternativa a, respectivamente, incrementar uma infraestrutura *Thin* para porte *Medium* e incrementar uma infraestrutura *Medium* para *Large*. Na primeira situação, a execução sob a política Caso 5 em uma infraestrutura *Thin* apresentou tempos de execução médios para submissões bem mais interessantes que a execução exclusiva sobre os recursos federados (Caso 4), e ainda com custo inferior para execução do total da carga submetida. A mesma análise pode ser aplicada comparando as execuções sobre a infraestrutura federada *Medium* sob as políticas de escalonamento realizadas pelos casos 4 e 5.

Observando a situação na qual a demanda de processamento é muito alta, percebe-se que o uso de *bursting*, tanto para a infraestrutura *Medium* quanto para a *Large*, oferece um importante incremento no desempenho. Considerando um médio investimento na infraestrutura, observa-se este incremento em termos de desempenho, sendo que o custo permanece praticamente o mesmo utilizando (Caso 5) ou não (Caso 4) *bursting*. No entanto, ao adotar uma infraestrutura *Large* para a infraestrutura federada, o uso de *bursting* permitiu

uma importante redução, em torno de 25%, no custo final de execução.

Tabela 25 – Estimativa de custos para alguns Casos de Estudo.

					Estimativa de custo anual			
Infra.	**Esc.**	**Tempo Total (h)**	**Infra Fed.**	**Custo Fed. Est.**	**100%**	**Custo Nuv. Est.**	**Total**	**Diferença**
Baixa demanda computacional								
Thin	Caso 4	24,66	14.400,00	12.324,30	12.614,40	-	12.324,30	290,10
Medium	Caso 4	12,59	34.800,00	88.299,13	29.433,60	-	88.299,13	-58.865,33
Thin	Caso 5	10,56	14.400,00	12.611,08	12.614,40	22.193,84	34.804,93	3,32
Média demanda computacional								
Thin	Caso 4	518,52	14.400,00	12.614,36	12.614,60	-	12.614,36	0,04
Medium	Caso 4	45,74	34.800,00	88.300,80	29.433,60	-	88.300,80	-58.867,20
Thin	Caso 5	71,05	14.400,00	12.614,35	12.614,40	42.091,81	54.706,16	0,05
Medium	Caso 5	26,29	34.800,00	88.301,47	29.433,60	44.308,98	132.610,45	-58.867,87
Large	Caso 4	14,87	40.800,00	201.829,46	33.638,40	-	201.829,46	-168.191,06
Alta demanda computacional								
Medium	Caso 4	611,12	34.800,00	88.300,76	29.433,60	-	88.300,76	-58.867,16
Medium	Caso 5	400,00	34.800,00	88.300,84	29.433,60	46.924,30	135.225,15	-58.867,27
Large	Caso 4	527,78	40.800,00	201.830,37	33.638,40	-	201.830,37	-168.191,97
Large	Caso 5	325,23	40.800,00	201.830,30	33.638,40	44.178,05	246.008,35	-168.191,90

Os dados apresentados na Tabela 24, e todas as demais informações de tempo de execução apresentadas anteriormente neste capítulo, corroboram o senso comum de que o aumento da capacidade de uma infraestrutura física de processamento melhora o desempenho, em termos de tempo de execução da carga de trabalho submetida, considerando o uso de uma estratégia de escalonamento adequada. A questão que se coloca é a identificação do momento em que um investimento em incremento da infraestrutura se faz necessário. Uma análise neste sentido é realizada considerando os dados na Tabela 25. Nesta tabela é apresentada a estimativa de custo de uso de uma infraestrutura de nuvem durante 12 meses, nos quais a submissão de cargas computacionais se dá nos mesmos padrões daqueles apresentados na Tabela 24.

Na Tabela 25 é apresentado o tempo, em horas, para executar o conjunto de submissões relacionados a uma determinada carga computacional baixa, média ou alta, em uma dada infraestrutura de nuvem *Thin*, *Medium* ou *Large*, considerando o emprego do balanceamento de carga intra e entre sítios (Caso 4) e o uso do balanceamento de carga intra e entre sítios utilizando *cloud bursting* em uma nuvem pública como apoio (Caso 5). A tabela também apresenta informações sobre custos de implantação da infraestrutura física (conforme valores indicados na Tabela 12) e também o que representa este investimento em termos de oferta de processamento no período considerado. A coluna “100%” indica o custo da utilização plena desta infraestrutura durante os 12 meses para quais a estimativa é levantada. Sobre estes valores, a estimativa considera uma submissão constante da carga caracterizada no cenário durante os 12 meses, atendendo ao mesmos requisitos de desempenho (expressos pelo tempo de execução aferido). A coluna “Total” resulta

dos somatórios dos custos computacionais observados pelo consumo de recursos de processamento da nuvem federada e da nuvem pública. A coluna “Diferença” indica se a infraestrutura federada atende ou não a demanda computacional submetida, apresentando valores positivos ou negativos, respectivamente.

Na análise da Tabela 25 para o cenário em que a carga de trabalho submetida é baixa, o entendimento é realizado em um ciclo de 24,66 horas. Caso esta carga seja constantemente submetida, o atendimento das submissões no tempo mensurado se dá pela infraestrutura federada, havendo uma pequena margem de ociosidade, na ordem de US$ 290,10. O incremento da infraestrutura federada para equipamentos de média capacidade de processamento demanda um investimento de US$ 88.299,13, significativamente superior ao realizado na infraestrutura de baixa capacidade (US$ 12.324,20). No entanto, o tempo para execução do cenário considerado caiu para cerca de 50% do tempo, executando um conjunto de submissões em 12,59 horas, praticamente a metade do tempo observado na execução do mesmo cenário em uma infraestrutura *Thin*. Nesta situação, caso a demanda permaneça constante, o custo computacional requerido ultrapassa, com larga margem (US$ 58.865,33) a capacidade nominal da infraestrutura federada (US$ 29.433,60). Nesta situação, o uso de *cloud bursting* justifica-se por permitir o aumento da capacidade de processamento global oferecida (nesta situação, cada cenário é executado em 10,56 horas) e o custo de locação de recursos de processamento na nuvem é de US$ 22.193,84 inferior ao custo de atualização da infraestrutura federada.

Considerando o cenário em que a demanda computacional é considerada média, é possível observar que, quando apenas recursos da nuvem federada são utilizados, o incremento da capacidade oferecida reverte-se em um considerável ganho de desempenho na execução das submissões. No entanto, o investimento realizado não garante este desempenho caso a demanda se mantenha constante no cenário considerado. Os dados informam que uma infraestrutura *Thin* é capaz de suportar ciclos de submissões a cada 518,52 horas, embora no limite, havendo uma ociosidade de apenas US$ 0,04 de recursos de processamento. Aumentando a infraestrutura para *Medium* e *Large*, os ciclos de execução reduzem para 45,74 e 14,87 horas, mas o valor total de processamento requerido ultrapassa o oferecido na infraestrutura, em USS$ 58.865,22 e US$ 168.191,06, respectivamente.

No cenário de alta demanda computacional, observa-se que em todas as situações consideradas, nem a infraestrutura *Medium* nem a infraestrutura *Large* são capazes de atender a demanda computacional em um ciclo contínuo de 12 meses. Ainda assim, pode-se ter como conclusão que o uso de uma infraestrutura *Medium* com adoção de *cloud bursting* é mais interessante, financeiramente, que a atualização dos recursos de processamento da nuvem federada.

Tabela 26 – Tempos de execução e simulação das aplicações.

Infra.	Esc.	Tempo de Execução (h)	Tempo de Simulação (s)
Baixa demanda computacional			
Thin	Caso 4	24,66	1,18
Medium	Caso 4	12,59	0,57
Thin	Caso 5	10,55	0,58
Média demanda computacional			
Thin	Caso 4	518,52	87,61
Medium	Caso 4	45,74	9,20
Thin	Caso 5	71,04	10,31
Medium	Caso 5	26,29	4,20
Large	Caso 4	14,86	2,70
Alta demanda computacional			
Medium	Caso 4	611,11	543,38
Medium	Caso 5	399,99	242,36
Large	Caso 4	527,77	339,45
Large	Caso 5	325,22	171,12

Por fim, a Tabela 26 apresenta dados que corroboram a viabilidade de uso do simulador WCSim. Por meio destes dados, é possível verificar o bom desempenho do simulador, uma vez que, para execuções de aplicações submetidas ao WCSim, com tempo de execução na ordem de 24,66 horas (aplicação com baixa demanda computacional, infraestrutura *Thin* e escalonamento Caso 4), o tempo de simulação foi de 1,18 segundos. Para aplicações maiores e com alta demanda computacional, com tempos de execução de mais de 300 horas (infraestrutura *Medium* com escalonamento Caso 5, infraestrutura *Large* com escalonamento Casos 4 e 5), os tempos também se mostraram muito interessantes, girando em torno de, aproximadamente, 200 a 300 segundos de tempo de simulação. Desse modo, o WCSim se posiciona como uma promissora ferramenta para a análise do comportamento de aplicações modeladas a partir de fluxos de trabalho.

6.8 Conclusão do Capítulo

Este capítulo apresentou uma série de estudo de casos da aplicação do simulador WCSim, buscando identificar situações em que o uso de *cloud bursting* sobre uma nuvem acadêmica de instituições federadas se mostra satisfatória. O estudo permitiu identificar a consistência nos resultados apresentados pelo simulador. Para cada caso estudado foram apresentados os dados referentes à execução do pior escalonamento, o Caso 1, no qual todas as submissões sobre um sítio são executadas sobre o mesmo Servidor de Processamento. Os dados de desempenho desta execução, notadamente o seu custo computacional em termos de número de instruções executadas, foi igual ao observado

pelo somatório do número de instruções executadas pelos Servidores de Processamento individualmente nos demais casos.

Durante a etapa de modelagem dos estudos de caso, chamou a atenção o grande número de cenários que poderiam ser concebidos para realização dos experimentos. De fato, o problema pode ser analisado sob diferentes óticas e o simulador construído oferece vários graus de liberdade para construir estes diferentes cenários, mas a opção recaiu em delimitar os caso de estudos realizando a análise em apenas três dimensões: demanda computacional dos usuários, oferta de poder de processamento na nuvem federada e estratégias de escalonamento.

Em relação ao desempenho do próprio simulador, este mostrou-se bastante eficiente na realização das simulações. Os tempos totais de simulação variaram conforme a complexidade da aplicação e da infraestrutura utilizada nos cenários simulados. Os tempos de simulação, em segundos, e seus respectivos desvios padrões, estão na Tabela 27. São apresentadas as médias dos tempos de simulação nos diferentes cenários de demanda computacional, independente da infraestrutura utilizada[5], em um computador com processador Intel Core i7-10875H, 2.30GHz, 8 *cores*, 64GB RAM.

Tabela 27 – Tempos de simulação dos estudos de caso.

Cenário de Simulação	**Tempo de Simulação (s)**	**Desvio Padrão**
Baixa demanda	0,85	0,61
Média demanda	15,58	24,11
Alta demanda	297,94	180,26

[5]Os resultados representam a média do tempo colhido das execuções das simulações da demanda computacional informada na linha correspondente da tabela, com as três configurações de infraestrutura, e as seis estratégias de escalonamento. Portanto, a média é uma média aritmética simples de 18 valores, cada um identificando uma combinação de propriedades distintas. Assim, esta informação de tempos médios de execução é apenas ilustrativa.

7 CONCLUSÃO

Com a consolidação das infraestruturas de nuvem como suporte à execução de aplicações sob demanda, o meio acadêmico demonstra interesse nesta utilização para a submissão de trabalhos científicos que exigem uma vasta gama de opções de configuração e disponibilidade de recursos. Não raro, grupos de pesquisa necessitam de estruturas computacionais com capacidades específicas, em função dos fluxos de trabalho submetidos à execução e o gerenciamento destas estruturas pode se tornar complexo à medida que os recursos computacionais tornam-se escassos e os investimentos financeiros são insuficientes.

Para tratar do problema de gerenciamento de recursos computacionais e financeiros, o ambiente de nuvem se mostra muito bem projetado, sendo capaz de lidar com a disponibilidade de acessos aos recursos computacionais, bem como permitir um melhor aproveitamento dos recursos financeiros disponíveis.

Com relação aos custos financeiros e sua contabilização, a precificação em ambientes de nuvem trata do processo pelo qual se determina quanto um provedor de serviços receberá de um cliente final pelos serviços prestados. O processo de precificação pode ser regular, quando o cliente paga sempre o mesmo valor durante todo o período de uso ou na modalidade *spot*, pagando apenas pelo que for realmente utilizado. Assim, neste trabalho, as modalidades de precificação em nuvem foram consideradas para as análises a serem realizados na execução dos experimentos.

Os estudos realizados nesta Tese buscaram atingir os objetivos de modelar diferentes configurações para soluções de nuvem, sejam elas privadas, públicas ou híbridas, a fim de atender um conjunto de instituições de ensino e pesquisa e identificar custos associados à implementação de diferentes estratégias de nuvem, verificando a demanda de recursos computacionais requeridos pelas pesquisas de diferentes áreas. Isto foi realizado a partir de uma das contribuições científicas apresentada por este trabalho, no caso, o simulador WCSim. Com o WCSim foi possível simular a execução de vários cenários nos quais aplicações, de baixa, média e alta demanda computacional, tiveram seus comportamentos analisados. Tais análises também levaram em conta estruturas de nuvens federadas e estratégias de escalonamento, além de *cloud bursting*.

Como parte do esforço em compreender o estado da arte dos estudos relacionados ao uso de nuvens computacionais para a execução de aplicações com diferentes demandas por recursos computacionais, foi elaborada uma Revisão Sistemática da Literatura (RSL) na área do tema deste trabalho. A RSL teve como ponto de partida algumas Questões de Pesquisa que foram respondidas a partir dos achados resultantes do processo de revisão. Com a conclusão da Revisão Sistemática da Literatura foi possível identificar trabalhos que abordam temas relacionados à análise de investimentos em infraestruturas de nuvem. Os trabalhos levam em consideração a execução de aplicações com demandas de processamento de alto desempenho em ambientes de nuvem, bem como a possibilidade de migração destas aplicações, a partir de estruturas clássicas de HPC ou nuvens privadas, para ambientes de nuvem pública e os custos financeiros envolvidos. Ainda, a partir dos resultados da RSL, é possível constatar que a migração de aplicações de suas estruturas locais dedicadas para nuvens públicas, não é uma tarefa fácil, principalmente no que diz respeito aos aspectos financeiros envolvidos no processo. Especificamente sobre custos relacionados à infraestrutura, na RSL observa-se que não emergiram considerações sobre os custos de comunicação associados às transferências de dados nem à adoção de soluções de nuvens híbridas. Estes aspectos se mostram como oportunidades de pesquisa em aberto.

Uma das contribuições científicas desta Tese é o WCSim (*Workflow Cloud Simulator*). Ele é um simulador voltado para a análise de desempenho de aplicações em nuvens computacionais, considerando diferentes estratégias de escalonamento e a opção de *cloud bursting*. O WCSim foi desenvolvido para suprir a necessidade de avaliação do comportamento de cargas computacionais modeladas em *workflows*, a partir de tarefas relacionadas em *bag-of-tasks*. Outro ponto importante para o desenvolvimento do WCSim foi a possibilidade de estender resultados documentados em trabalhos anteriores.

O simulador WCSim possui alto grau de adaptação a diferentes modelos de *workflows*, usuários e configurações de infraestrutura de nuvem, sem a necessidade de alterações em seu código fonte. Tais caraterísticas são um diferencial em comparação com outros simuladores encontrados na literatura. Soma-se a isso o fato de o simulador também permitir a introdução de novas estratégias de escalonamento por meio de interfaces de programação específicas.

Por fim, os estudos de caso apresentados neste trabalho representam os possíveis cenários de utilização de nuvens computacionais em ambientes acadêmicos. Uma importante constatação extraída dos resultados dos experimentos é a de que precisa-se levar em consideração o processo de precificação na modalidade *spot*, principalmente se ofertada por algum eventual provedor de serviços de nuvem, durante processos de contratação. A modalidade *spot* oferece serviços a preços geralmente inferiores à modalidade regular de contratação. Outra observação, agora sobre estratégias de escalonamento, permitiu verificar que o balanceamento intra e inter sítio é um recurso com grande potencial de exploração,

podendo contribuir para o aumento dos ganhos de desempenho da infraestrutura de nuvem federada.

7.1 Trabalhos Futuros

O simulador WCSim desenvolvido no contexto do presente trabalho mostrou-se, durante a realização dos experimentos, uma ferramenta bastante robusta. Sua vocação para simulação de *workflows* e sua simplicidade para introduzir casos de estudo a torna atrativa frente aos simuladores para computação em nuvem disponíveis atualmente. Nesta Tese, foi atestada a operacionalidade do simulador. Deve-se seguir sua validação frente aos demais simuladores disponíveis, comparando e contrastando os resultados obtidos.

Na exploração do simulador existem diversas outras frentes que podem ser seguidas. A primeira diz respeito ao aprimoramento do modelo de custos para execução de aplicações em nuvens federadas com apoio de *cloud bursting*. Os modelos apresentados nesta tese limitaram-se a apontar as relações de custo e desempenho na exploração desta infraestrutura de execução. Novos modelos devem ser mais realistas no que diz respeito aos custos de comunicação e mais detalhistas em relação às técnicas de escalonamento empregadas. Em particular, a observação de alguns resultados mostrou que, em algumas situações, grande parte do processamento se deu em recursos provisionados na nuvem pública quando havia algum recurso de processamento disponível na nuvem federada. Estratégias de escalonamento que contemplem repatriar máquinas virtuais que estejam na nuvem pública para a nuvem federada são abordagens que merecem ser avaliadas, assim como técnicas de escalonamento que considerem o custo de execução, combinando com desempenho esperado, como um fator decisório de alocação e migração de trabalho entre as máquinas.

REFERÊNCIAS

ACETO, G.; BOTTA, A.; de Donato, W.; PESCAPÈ, A. Cloud Monitoring: A Survey. **Computer Networks**, Napoli, Italy, v.57, n.9, p.2093–2115, June 2013.

Al-Roomi, M.; Al-Ebrahim, S.; BUQRAIS, S.; AHMAD, I. Cloud Computing Pricing Models: A Survey. **International Journal of Grid and Distributed Computing**, Hobart, Australia, v.6, n.5, p.93–106, Oct. 2013.

ALADYSHEV, O. S. et al. Variants of deployment the high performance computing in clouds. In: IEEE Conference OF Russian Young Researchers IN Electrical AND Electronic Engineering (EIConRus), 2018., 2018, Moscow. **Anais...** IEEE, 2018. p.1453–1457.

ARABNEJAD H. BARBOSA, J. A Budget Constrained Scheduling Algorithm for Workflow Applications. **J Grid Computing**, Porto, Portugal, v.12, p.665–679, 2014.

ARABNEJAD, V.; BUBENDORFER, K.; NG, B. Scheduling Deadline Constrained Scientific Workflows on Dynamically Provisioned Cloud Resources. **Future Generation Computer Systems**, Wellington, New Zealand, v.75, p.348–364, Oct. 2017.

ARABNEJAD, V.; BUBENDORFER, K.; NG, B. Budget and deadline aware e-science workflow scheduling in clouds. **IEEE Transactions on Parallel and Distributed systems**, Los Alamitos, CA, USA, v.30, n.1, p.29–44, 2018.

BARIKA, M. et al. IoTSim-Stream: Modelling stream graph application in cloud simulation. **Future Generation Computer Systems**, Australia, v.99, p.86–105, 2019.

BENDECHACHE, M. et al. Modelling and simulation of ElasticSearch using CloudSim. In: IEEE/ACM 23RD INTERNATIONAL SYMPOSIUM ON DISTRIBUTED SIMULATION AND REAL TIME APPLICATIONS (DS-RT), 2019., 2019, Cosenza, Italy. **Anais...** IEEE, 2019. p.1–8.

BHAVANI, P.; JYOTHI, C. Investigation on Security Challenges over a Cloud Computing. **International Journal of Scientific Research in Science and Technology**, Gujarat, India, v.3, p.1374–1380, 2017.

BUYYA, R.; MURSHED, M. M. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. **Concurr. Comput. Pract. Exp.**, Melbourne, Australia, v.14, n.13-15, p.1175–1220, 2002.

BUYYA, R.; MURSHED, M. M.; ABRAMSON, D.; VENUGOPAL, S. Scheduling parameter sweep applications on global Grids: a deadline and budget constrained cost–time optimization algorithm. **Software: Practice and Experience**, Melbourne, Australia, v.35, 2005.

CALHEIROS, R. N.; RANJAN, R.; ROSE, C. A. F. D.; BUYYA, R. CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services. **CoRR**, Melbourne, Australia, v.abs/0903.2525, 2009.

CASANOVA, H.; LEGRAND, A.; QUINSON, M.; SUTER, F. SMPI Courseware: Teaching Distributed-Memory Computing with MPI in Simulation. In: IEEE/ACM WORKSHOP ON EDUCATION FOR HIGH-PERFORMANCE COMPUTING (EDUHPC), 2018., 2018, Dallas, USA. **Anais...** IEEE, 2018. p.21–30.

CASANOVA, H. et al. Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms. **Journal of Parallel and Distributed Computing**, Los Angeles, USA, v.74, n.10, p.2899–2917, June 2014.

CIRNE, W.; BRASILEIRO, F.; SAUVÉ, J. Grid computing for bag of tasks applications. **In Proc. of the 3rd IFIP ...**, São Paulo, Brasil, 2003.

CUI, Y.; INGALZ, C.; GAO, T.; HEYDARI, A. Total Cost of Ownership Model for Data Center Technology Evaluation. In: IEEE Intersociety Conference ON Thermal AND Thermomechanical Phenomena IN Electronic Systems (ITherm), 2017., 2017, Orlando, FL. **Anais...** IEEE, 2017. p.936–942.

DAOUDI, I. et al. sOMP: Simulating OpenMP Task-Based Applications with NUMA Effects. In: 2020, Lyon, France. **Anais...** INRIA, 2020. p.197–211.

DEELMAN, E. et al. Pegasus in the Cloud: Science Automation through Workflow Technologies. **IEEE Internet Computing**, Chipre, v.20, n.1, p.70–76, Jan. 2016.

DEPOORTER, W.; DE MOOR, N.; VANMECHELEN, K.; BROECKHOVE, J. Scalability of Grid Simulators: An Evaluation. In: EURO-PAR 2008 – PARALLEL PROCESSING, 2008, Berlin, Heidelberg. **Anais...** Springer Berlin Heidelberg, 2008. p.544–553.

DREHER, P.; NAIR, D.; SILLS, E.; VOUK, M. Cost Analysis Comparing HPC Public Versus Private Cloud Computing. In: HELFERT, M.; FERGUSON, D.; MÉNDEZ MUÑOZ, V.; CARDOSO, J. (Ed.). **Cloud Computing and Services Science**. Cham: Springer International Publishing, 2017. v.740, p.294–316.

EMERAS, J.; VARRETTE, S.; PLUGARU, V.; BOUVRY, P. Amazon Elastic Compute Cloud (EC2) versus In-House HPC Platform: A Cost Analysis. **IEEE Transactions on Cloud Computing**, San Francisco, CA, USA, v.7, n.2, p.456–468, Apr. 2019.

EPEMA, D.; IOSUP, A. Grid Computing Workloads. **IEEE Internet Computing**, Los Alamitos, CA, USA, v.15, n.02, p.19–26, mar 2011.

FALCÃO, I. W. S. et al. Modelagem de Custo Total de Propriedade (TCO) de uma Infraestrutura Computacional em Nuvem. In: Anais do Seminário Integrado de Software e Hardware (SEMISH), 2019. **Anais...** Sociedade Brasileira de Computação - SBC, 2019. p.57–68.

FANFAKH, A. B. M. Predicting the Performance of MPI Applications over Different Grid Architectures. **Journal of University of Babylon for Pure and Applied Sciences**, Babylon, Iraq, v.27, n.1, p.468–477, 2019.

FICCO, M.; AMATO, A.; VENTICINQUE, S. Hosting Mission-Critical Applications on Cloud: Technical Issues and Challenges. In: LAMBOGLIA, R.; CARDONI, A.; DAMERI, R. P.; MANCINI, D. (Ed.). **Network, Smart and Open**. Cham: Springer International Publishing, 2018. v.24, p.179–191. Series Title: Lecture Notes in Information Systems and Organisation.

FILIOPOULOU, E. et al. Integrating Cost Analysis in the Cloud: A SoS Approach. In: International Conference ON Innovations IN Information Technology (IIT), 2015., 2015, Dubai, United Arab Emirates. **Anais...** IEEE, 2015. p.278–283.

GANTIKOW, H.; REICH, C.; KNAHL, M.; CLARKE, N. A Taxonomy for HPC-aware Cloud Computing. In: 2015, Konstanz, Germany. **Anais...** Hochschule Konstanz, 2015. p.57 – 62. (2nd Baden-Württemberg Center of Applied Research Symposium on Information and Communication Systems SInCom 2015, Konstanz).

GUERRERO, G. D. et al. A Performance/Cost Model for a CUDA Drug Discovery Application on Physical and Public Cloud Infrastructures. **Concurrency and Computation: Practice and Experience**, USA, v.26, n.10, p.1787–1798, July 2014.

GUPTA, A. et al. Evaluating and Improving the Performance and Scheduling of HPC Applications in Cloud. **IEEE Transactions on Cloud Computing**, New York, USA, v.4, n.3, p.307–321, July 2016.

GUPTA, S.; SINGH, R. S.; VASANT, U. D.; SAXENA, V. User defined weight based budget and deadline constrained workflow scheduling in cloud. **Concurrency and Computation: Practice and Experience**, USA, v.33, n.24, p.e6454, 2021.

HATTI, D. I.; SUTAGUNDAR, A. V. Resource Provisioning in Fog-Based IoT. In: **Inventive Computation and Information Technologies**. Coimbatore, India: Springer, 2022. p.433–447.

HIROFUCHI, T.; LèBRE, A.; POUILLOUX, L. SimGrid VM: Virtual Machine Support for a Simulation Framework of Distributed Systems. **IEEE Transactions on Cloud Computing**, New York, USA, v.PP, 01 2016.

JIGSAW. **What is Cloud Bursting? All You Need To Know in 2021**. Disponivel em: <https://www.jigsawacademy.com/blogs/cloud-computing/what-is-cloud-bursting/>. Accesso em: 19 janeiro 2022.

JUVE, G. et al. Characterizing and Profiling Scientific Workflows. **Future Generation Computer Systems**, Marina del Rey, USA, v.29, n.3, p.682–692, Mar. 2013.

KEHRER, S.; BLOCHINGER, W. A survey on cloud migration strategies for high performance computing. In: SYMPOSIUM AND SUMMER SCHOOL ON SERVICE-ORIENTED COMPUTING (SUMMERSOC19), 13., 2019, Reutlingen, Germany. **Proceedings...** IBM, 2019. p.57–69.

LEE, Y. C.; LIAN, B. Cloud Bursting Scheduler for Cost Efficiency. In: IEEE 10TH International Conference ON Cloud Computing (CLOUD), 2017., 2017, Honolulu, CA, USA. **Anais...** IEEE, 2017. p.774–777.

LI, D.; ASIKABURU, C.; SHANG, J.; WANG, N. Numerical and simulation verification for optimal server allocation in edge computing. In: IEEE INTERNATIONAL IOT, ELECTRONICS AND MECHATRONICS CONFERENCE (IEMTRONICS), 2021., 2021, Toronto, ON, Canada. **Anais...** IEEE, 2021. p.1–7.

LI, D. et al. Towards optimal system deployment for edge computing: a preliminary study. In: INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS AND NETWORKS (ICCCN), 2020., 2020, Honolulu, HI, USA. **Anais...** IEEE, 2020. p.1–6.

MANSOURI, N.; GHAFARI, R.; ZADE, B. M. H. Cloud computing simulators: A comprehensive review. **Simulation Modelling Practice and Theory**, Thessaloniki, Greece, v.104, p.102144, 2020.

MANSOURI, Y.; PROKHORENKO, V.; BABAR, M. A. An Automated Implementation of Hybrid Cloud for Performance Evaluation of Distributed Databases. **arXiv:2006.02833 [cs]**, Adelaide, Australia, June 2020. arXiv: 2006.02833.

MARATHE, A. et al. A Comparative Study of High-Performance Computing on the Cloud. In: INTERNATIONAL SYMPOSIUM ON High-PERFORMANCE PARALLEL AND DISTRIBUTED COMPUTING - HPDC '13, 22., 2013, New York, New York, USA. **Proceedings...** ACM Press, 2013. p.239.

MASTENBROEK, F. et al. OpenDC 2.0: Convenient Modeling and Simulation of Emerging Technologies in Cloud Datacenters. In: IEEE/ACM 21ST INTERNATIONAL SYMPOSIUM

ON CLUSTER, CLOUD AND INTERNET COMPUTING (CCGRID), 2021., 2021, Melbourne, Australia. **Anais...** IEEE, 2021. p.455–464.

MECHALIKH, C.; TAKTAK, H.; MOUSSA, F. PureEdgeSim: A simulation toolkit for performance evaluation of cloud, fog, and pure edge computing environments. In: INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING & SIMULATION (HPCS), 2019., 2019, Dublin, Ireland. **Anais...** IEEE, 2019. p.700–707.

MEHTA, H.; KANUNGO, P.; CHANDWANI, M. EcoGrid: a dynamically configurable object oriented simulation environment for economy-based grid scheduling algorithms. In: FOURTH ANNUAL ACM BANGALORE CONFERENCE, 2011, Bangalore, India. **Proceedings...** Association for Computing Machinery, 2011. p.1–8.

MELL, P.; GRANCE, T. **The NIST definition of cloud computing**. Gaithersburg, MD: National Institute of Standards and Technology, 2011.

MIERITZ, L.; KIRWIN, B. Defining Gartner Total Cost of Ownership. , Stamford, USA, p.11, 2005.

MOHAMMED, A.; ELELIEMY, A.; CIORBA, F. M. Towards the Reproduction of Selected Dynamic Loop Scheduling Experiments Using SimGrid-SimDag. In: IEEE 19TH INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS; IEEE 15TH INTERNATIONAL CONFERENCE ON SMART CITY; IEEE 3RD INTERNATIONAL CONFERENCE ON DATA SCIENCE AND SYSTEMS (HPCC/SMARTCITY/DSS), 2017., 2017, Bangkok, Thailand. **Anais...** IEEE, 2017. p.623–626.

NARANG, S.; GOSWAMI, P.; JAIN, A. Statistical Analysis of Cloud Based Scheduling Heuristics. In: INTERNATIONAL CONFERENCE ON INFORMATION, COMMUNICATION AND COMPUTING TECHNOLOGY, 2019, Singapore. **Anais...** Springer Singapore, 2019. p.98–112.

NARANG, S.; GOSWAMI, P.; JAIN, A. A Comprehensive Review of Load Balancing Techniques in Cloud Computing and Their Simulation with CloudSim Plus. **Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)**, Rome, Italy, v.14, n.6, p.1684–1694, 2021.

NASR, A. A.; EL-BAHNASAWY, N. A.; ATTIYA, G.; EL-SAYED, A. Cost-effective algorithm for workflow scheduling in cloud computing under deadline constraint. **Arabian Journal for Science and Engineering**, Dhahran, Saudi Arabia, v.44, n.4, p.3765–3780, 2019.

NETAPP. **Cloud Bursting: Choosing the Best Approach for You**. Disponivel em: <https://cloud.netapp.com/blog/cvo-blg-cloud-bursting-choosing-the-best-approach-for-you#H_H4>. Accesso em: 19 janeiro 2022.

NETTO, M. A. S. et al. HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges. **ACM Computing Surveys**, New York, USA, v.51, n.1, p.1–29, Apr. 2018. arXiv: 1710.08731.

OKOLI, C.; SCHABRAM, K. A Guide to Conducting a Systematic Literature Review of Information Systems Research. **SSRN Electronic Journal**, Montreal, Canada, 2010.

PARASHAR, M.; ABDELBAKY, M.; RODERO, I.; DEVARAKONDA, A. Cloud Paradigms and Practices for Computational and Data-Enabled Science and Engineering. **Computing in Science & Engineering**, Washington, DC, USA, v.15, n.4, p.10–18, July 2013.

PARSIFAL. **Systematic Literature Review Tool**. Disponivel em: <https://parsif.al/>. Accesso em: 15 novembro 2019.

PEGASUS. **Pegasus WMS - Automate, recover and debug scientific computations**. Disponivel em: <https://pegasus.isi.edu/>. Accesso em: 26 junho 2022.

PHAM, A.; JÉRON, T.; QUINSON, M. Verifying MPI applications with SimGridMC. In: FIRST INTERNATIONAL WORKSHOP ON SOFTWARE CORRECTNESS FOR HPC APPLICATIONS, 2017, Denver, CO, USA. **Proceedings...** Association for Computing Machinery, 2017. p.28–33.

PRUKKANTRAGORN, P.; TIENTANOPAJAI, K. Price Efficiency in High Performance Computing on Amazon Elastic Compute Cloud Provider in Compute Optimize Packages. In: International Computer Science AND Engineering Conference (ICSEC), 2016., 2016, Chiang Mai, Thailand. **Anais...** IEEE, 2016. p.1–6.

RAMAMONJISOA, C. E. et al. Simulation of asynchronous iterative algorithms using simgrid. In: IEEE INTL CONF ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS, 2014 IEEE 6TH INTL SYMP ON CYBERSPACE SAFETY AND SECURITY, 2014 IEEE 11TH INTL CONF ON EMBEDDED SOFTWARE AND SYST (HPCC, CSS, ICESS), 2014., 2014, Paris, France. **Anais...** IEEE, 2014. p.890–895.

RAMGOVIND, S.; ELOFF, M. M.; SMITH, E. The management of security in Cloud computing. In: Information Security FOR South Africa, 2010., 2010, Johannesburg, South Africa. **Anais...** IEEE, 2010. p.1–7.

ROLOFF, E.; DIENER, M.; CARISSIMI, A.; NAVAUX, P. O. A. High Performance Computing in the Cloud: Deployment, Performance and Cost Efficiency. In: IEEE International Conference ON Cloud Computing Technology AND Science Proceedings, 4., 2012, Taipei, Taiwan. **Anais...** IEEE, 2012. p.371–378.

ROLOFF, E.; DIENER, M.; GASPARY, L.; NAVAUX, P. Exploring Instance Heterogeneity in Public Cloud Providers for HPC Applications:. In: International Conference ON Cloud

Computing AND Services Science, 9., 2019, Heraklion, Crete, Greece. **Proceedings...** SCITEPRESS - Science and Technology Publications, 2019. p.210–222.

ROLOFF, E.; DIENER, M.; GASPARY, L. P.; NAVAUX, P. O. A. HPC Application Performance and Cost Efficiency in the Cloud. In: Euromicro International Conference ON Parallel, Distributed AND Network-BASED Processing (PDP), 2017., 2017, St. Petersburg, Russia. **Anais...** IEEE, 2017. p.473–477.

SADOOGHI, I. et al. Understanding the Performance and Potential of Cloud Computing for Scientific Applications. **IEEE Transactions on Cloud Computing**, New York, USA, v.5, n.2, p.358–371, Apr. 2017.

SANTOS, M. A. d.; CAVALHEIRO, G. G. H. Cloud infrastructure for HPC investment analysis. **Revista de Informática Teórica e Aplicada**, Porto Alegre, Brasil, v.27, n.4, p.45–62, Dec. 2020.

SANTOS, M. A. dos. **Modelo de Escalonamento Aplicativo para Bag of Tasks em Ambientes de Nuvem Computacional**. Pelotas, Brasil: UFPel, 2016.

SANTOS, M. A. dos; DU BOIS, A. R.; CAVALHEIRO, G. G. H. A User-Level Scheduling Framework for BoT Applications on Private Clouds. In: INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE AND HIGH PERFORMANCE COMPUTING (SBAC-PAD), 2017., 2017, Campinas, Brazil. **Anais...** IEEE, 2017. p.81–88.

SHEN, Y. et al. Cost-Optimized Resource Provision for Cloud Applications. In: IEEE Intl Conf ON High Performance Computing AND Communications, 2014 IEEE 6TH Intl Symp ON Cyberspace Safety AND Security, 2014 IEEE 11TH Intl Conf ON Embedded Software AND Syst (HPCC,CSS,ICESS), 2014., 2014, Paris, France. **Anais...** IEEE, 2014. p.1060–1067.

SILVA FILHO, M. C. et al. CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In: IFIP/IEEE SYMPOSIUM ON INTEGRATED NETWORK AND SERVICE MANAGEMENT (IM), 2017., 2017, Lisbon, Portugal. **Anais...** IEEE, 2017. p.400–406.

STREBEL, J.; STAGE, A. An Economic Decision Model for Business Software Application Deployment on Hybrid Cloud Environments. **IT Performance Management**, Germany, p.13, 2010.

THAKUR, A.; GORAYA, M. S. RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment. **Simulation Modelling Practice and Theory**, Longowal, India, v.116, p.102485, 2022.

WAINER, G.; MOSTERMAN, P. **Discrete-Event Modeling and Simulation**: Theory and Applications. [S.l.]: CRC Press, 2018. (Computational Analysis, Synthesis, and Design of Dynamic Systems).

WALTERBUSCH, M.; MARTENS, B.; TEUTEBERG, F. Evaluating Cloud Computing Services from a Total Cost of Ownership Perspective. **Management Research Review**, Osnabrueck, Germany, v.36, n.6, p.613–638, May 2013.

WILLEBEEK-LEMAIR, M.; REEVES, A. Strategies for dynamic load balancing on highly parallel computers. **IEEE Transactions on Parallel and Distributed Systems**, Illinois, USA, v.4, n.9, p.979–993, 1993.

XIAO, Y.; WATSON, M. Guidance on Conducting a Systematic Literature Review. **Journal of Planning Education and Research**, Texas, USA, v.39, n.1, p.93–112, Mar. 2017.

YELICK, K. et al. The Magellan Report on Cloud Computing for Science. **U.S. Department of Energy - Office of Science - Office of Advanced Scientific Computing Research (ASCR)**, Oak Ridge, TN, p.170, Dec. 2011.

ZHAO, H.; LI, X. Designing Flexible Resource Rental Models for Implementing HPC-as-a-Service in Cloud. In: IEEE 26TH International Parallel AND Distributed Processing Symposium Workshops & PhD Forum, 2012., 2012, Shanghai, China. **Anais...** IEEE, 2012. p.2550–2553.