



**Ernane Rosa Martins**  
**(Organizador)**

# **Princípios e Aplicações da Computação no Brasil 3**

**Atena**  
Editora  
Ano 2019

Ernane Rosa Martins  
(Organizador)

# Princípios e Aplicações da Computação no Brasil 3

Atena Editora  
2019

2019 by Atena Editora  
Copyright © Atena Editora  
Copyright do Texto © 2019 Os Autores  
Copyright da Edição © 2019 Atena Editora  
Editora Chefe: Profª Drª Antonella Carvalho de Oliveira  
Diagramação: Lorena Prestes  
Edição de Arte: Lorena Prestes  
Revisão: Os Autores



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição Creative Commons. Atribuição 4.0 Internacional (CC BY 4.0).

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores. Permitido o download da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

### **Conselho Editorial**

#### **Ciências Humanas e Sociais Aplicadas**

Profª Drª Adriana Demite Stephani – Universidade Federal do Tocantins  
Prof. Dr. Álvaro Augusto de Borba Barreto – Universidade Federal de Pelotas  
Prof. Dr. Alexandre Jose Schumacher – Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso  
Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná  
Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília  
Prof. Dr. Constantino Ribeiro de Oliveira Junior – Universidade Estadual de Ponta Grossa  
Profª Drª Cristina Gaio – Universidade de Lisboa  
Prof. Dr. Deyvison de Lima Oliveira – Universidade Federal de Rondônia  
Prof. Dr. Edvaldo Antunes de Faria – Universidade Estácio de Sá  
Prof. Dr. Eloi Martins Senhora – Universidade Federal de Roraima  
Prof. Dr. Fabiano Tadeu Grazioli – Universidade Regional Integrada do Alto Uruguai e das Missões  
Prof. Dr. Gilmei Fleck – Universidade Estadual do Oeste do Paraná  
Profª Drª Ivone Goulart Lopes – Istituto Internazionele delle Figlie di Maria Ausiliatrice  
Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense  
Profª Drª Keyla Christina Almeida Portela – Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso  
Profª Drª Lina Maria Gonçalves – Universidade Federal do Tocantins  
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Prof. Dr. Marcelo Pereira da Silva – Universidade Federal do Maranhão  
Profª Drª Miranilde Oliveira Neves – Instituto de Educação, Ciência e Tecnologia do Pará  
Profª Drª Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa  
Profª Drª Rita de Cássia da Silva Oliveira – Universidade Estadual de Ponta Grossa  
Profª Drª Sandra Regina Gardacho Pietrobon – Universidade Estadual do Centro-Oeste  
Profª Drª Sheila Marta Carregosa Rocha – Universidade do Estado da Bahia  
Prof. Dr. Rui Maia Diamantino – Universidade Salvador  
Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará  
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande  
Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

#### **Ciências Agrárias e Multidisciplinar**

Prof. Dr. Alan Mario Zuffo – Universidade Federal de Mato Grosso do Sul  
Prof. Dr. Alexandre Igor Azevedo Pereira – Instituto Federal Goiano  
Profª Drª Daiane Garabeli Trojan – Universidade Norte do Paraná  
Prof. Dr. Darllan Collins da Cunha e Silva – Universidade Estadual Paulista  
Profª Drª Diocléa Almeida Seabra Silva – Universidade Federal Rural da Amazônia  
Prof. Dr. Fábio Steiner – Universidade Estadual de Mato Grosso do Sul  
Profª Drª Girlene Santos de Souza – Universidade Federal do Recôncavo da Bahia  
Prof. Dr. Jorge González Aguilera – Universidade Federal de Mato Grosso do Sul  
Prof. Dr. Júlio César Ribeiro – Universidade Federal Rural do Rio de Janeiro  
Profª Drª Raissa Rachel Salustriano da Silva Matos – Universidade Federal do Maranhão  
Prof. Dr. Ronilson Freitas de Souza – Universidade do Estado do Pará  
Prof. Dr. Valdemar Antonio Paffaro Junior – Universidade Federal de Alfenas

### Ciências Biológicas e da Saúde

Prof. Dr. Benedito Rodrigues da Silva Neto – Universidade Federal de Goiás  
Prof. Dr. Edson da Silva – Universidade Federal dos Vales do Jequitinhonha e Mucuri  
Profª Drª Elane Schwinden Prudêncio – Universidade Federal de Santa Catarina  
Prof. Dr. Gianfábio Pimentel Franco – Universidade Federal de Santa Maria  
Prof. Dr. José Max Barbosa de Oliveira Junior – Universidade Federal do Oeste do Pará  
Profª Drª Magnólia de Araújo Campos – Universidade Federal de Campina Grande  
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Profª Drª Vanessa Lima Gonçalves – Universidade Estadual de Ponta Grossa  
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande

### Ciências Exatas e da Terra e Engenharias

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto  
Prof. Dr. Alexandre Leite dos Santos Silva – Universidade Federal do Piauí  
Profª Drª Carmen Lúcia Voigt – Universidade Norte do Paraná  
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná  
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará  
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande  
Profª Drª Neiva Maria de Almeida – Universidade Federal da Paraíba  
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista

<b>Dados Internacionais de Catalogação na Publicação (CIP) (eDOC BRASIL, Belo Horizonte/MG)</b>	
P957	<p>Princípios e aplicações da computação no Brasil 3 [recurso eletrônico] / Organizador Ernane Rosa Martins. – Ponta Grossa, PR: Atena Editora, 2019. – (Princípios e aplicações da computação no Brasil; v. 3)</p> <p>Formato: PDF Requisito de sistema: Adobe Acrobat Reader Modo de acesso: World Wide Web Inclui bibliografia ISBN 978-85-7247-702-4 DOI 10.22533/at.ed.024191510</p> <p>1. Computação. 2. Informática. 3. Programação de computador. I. Martins, Ernane Rosa. II. Série.</p> <p style="text-align: right;">CDD 004</p>
<b>Elaborado por Maurício Amormino Júnior – CRB6/2422</b>	

Atena Editora  
Ponta Grossa – Paraná - Brasil  
[www.atenaeditora.com.br](http://www.atenaeditora.com.br)  
contato@atenaeditora.com.br

## APRESENTAÇÃO

O volume 3 da referida obra “Princípios e Aplicações da Computação no Brasil” apresenta 15 novos capítulos, que abordam assuntos importantes sobre o panorama atual da computação no Brasil, tais como: desenvolvimento de aplicativos móveis, VoIP, modelagem computacional, simulação, recurso educacional aberto, projeto de banco de dados, mobile learning, robótica, avaliação automática de código-fonte e agrupamento difuso multivariado.

Deste modo, esta obra reúne debates e análises acerca de questões relevantes, tais como: desenvolvimento de uma aplicação móvel para realização de recargas, consultas e simulação de saldo na carteira de transporte público, usando o método Design Constructal para melhorar a configuração geométrica de um novo layout para TCSA com quatro dutos; análise dos benefícios da tecnologia VoIP como alternativa de baixo custo a um serviço de telefonia institucional gerenciado por minicomputadores Raspberry e adaptadores ATA; análise do comportamento mecânico de placas finas de materiais compósitos laminados reforçados por fibras submetidas a esforços de flexão, comparando-as com o de uma placa de aço, usando modelagem computacional utilizando o software Abaqus 2017 Student Version®; apresentação de uma arquitetura de sistemas de simulação para área de convergência; desenvolvimento de uma abordagem para auxiliar o docente na criação de atividades com REA's que quando executados pelos alunos gerem o registro de suas experiências; análise da gestão de recursos humanos na administração pública por meio de um projeto de banco de dados; análise dos recursos disponíveis no WhatsApp Messenger, que oferecem possibilidades de uso no ambiente escolar como recurso pedagógico que pode auxiliar o processo de ensino e aprendizagem; simulação de um agente robótico capaz de realizar busca e exploração visual num ambiente virtual; e o desenvolvimento de uma ferramenta de análise e avaliação automática de código-fonte como alternativa para a otimização do processo de ensino-aprendizagem de programação.

Nesse sentido, esta obra apresenta enorme potencial para contribuir com análises e discussões aprofundadas sobre assuntos relevantes da área da computação, podendo servir de referência para novas pesquisas e estudos nesta área. Agradecemos em especial aos autores dos capítulos apresentados, e desejamos aos leitores, inúmeras reflexões proveitosas sobre as temáticas abordadas nesta obra.

Ernane Rosa Martins

## SUMÁRIO

<b>CAPÍTULO 1</b> .....	<b>1</b>
WALLET BUS - TECNOLOGIA E TRANSPORTE PÚBLICO	
Carlos Adriano Carneiro Pestana Juan Castro Ribeiro Leonardo Carlos Balbino	
<b>DOI 10.22533/at.ed.0241915101</b>	
<b>CAPÍTULO 2</b> .....	<b>16</b>
UM PROJETO CONSTRUCTAL DE TROCADORES DE CALOR SOLO-AR COMPOSTOS POR QUATRO DUTOS	
Ruth da Silva Brum Liércio André Isoldi Jairo Valões de Alencar Ramalho Luiz Alberto Oliveira Rocha Michel Kepes Rodrigues Elizaldo Domingues dos Santos	
<b>DOI 10.22533/at.ed.0241915102</b>	
<b>CAPÍTULO 3</b> .....	<b>30</b>
TELEFONIA DIGITAL DE BAIXO CUSTO COM VOIP E RASPBERRY: UM ESTUDO DE CASO NO IFPI CAMPUS ANGICAL DO PIAUÍ	
Francisco Alysson da Silva Sousa Matheus Lopes Pereira	
<b>DOI 10.22533/at.ed.0241915103</b>	
<b>CAPÍTULO 4</b> .....	<b>40</b>
SIMULAÇÃO NUMÉRICA PARA ANÁLISE DE DEFLEXÕES E TENSÕES EM PLACAS FINAS DE AÇO E DE MATERIAI COMPÓSITOS SUBMETIDAS A ESFORÇOS DE FLEXÃO	
João Vítor de Azambuja Carvalho Daniel Helbig	
<b>DOI 10.22533/at.ed.0241915104</b>	
<b>CAPÍTULO 5</b> .....	<b>49</b>
SIMULAÇÃO DE ÁREA DE CONVERGÊNCIA NUVEM-HPC	
Alessandro Kraemer	
<b>DOI 10.22533/at.ed.0241915105</b>	
<b>CAPÍTULO 6</b> .....	<b>63</b>
RECREATE: RECOMENDAÇÃO DE REA'S POR MEIO DA DESCOBERTA DO ESTILO DE APRENDIZAGEM PELA EXPERIÊNCIA DO ESTUDANTE E EXPECTATIVA DO PROFESSOR	
Marco Antonio Colombo Da Silva Marilde Terezinha Prado Santos	
<b>DOI 10.22533/at.ed.0241915106</b>	
<b>CAPÍTULO 7</b> .....	<b>76</b>
PROJETO DE BANCO DE DADOS COM FOCO NA ALOCAÇÃO DEPARTAMENTAL DE SERVIDORES PÚBLICOS E SUAS ATRIBUIÇÕES FUNCIONAIS	
José Nazareno Alves Rodrigues	
<b>DOI 10.22533/at.ed.0241915107</b>	

<b>CAPÍTULO 8</b> .....	<b>87</b>
M-LEARNING ATRAVÉS DO WHATSAPP: UMA ANÁLISE DE RECURSOS	
Nilson Alves dos Santos Mayara Kaynne Fragoso Cabral Beatriz Francisco Coelho	
<b>DOI 10.22533/at.ed.0241915108</b>	
<b>CAPÍTULO 9</b> .....	<b>98</b>
EXPLORAÇÃO E BUSCA VISUAL ROBÓTICA EM AMBIENTE SIMULADO	
Felipe Duque Belfort de Oliveira Hansenclever de França Bassani Aluizio Fausto Ribeiro Araujo	
<b>DOI 10.22533/at.ed.0241915109</b>	
<b>CAPÍTULO 10</b> .....	<b>112</b>
ESTUDO SOBRE A ESPECTROSCOPIA NIR PARA DETECÇÃO DONÍVEL ALCOÓLICO NO SANGUE	
Luis Gustavo Longo da Silva Cláudio Luis D'Elia Machado Fabiano Sandrini Moraes Rafael Galli	
<b>DOI 10.22533/at.ed.02419151010</b>	
<b>CAPÍTULO 11</b> .....	<b>125</b>
ESTUDO COMPUTACIONAL DO COEFICIENTE DE IMPACTO VERTICAL EM LONGARINAS DE PONTES RODOVIÁRIAS BRASILEIRAS	
Vinícius Heidtmann Avila Mauro de Vasconcellos Real Márcio Wrague Moura	
<b>DOI 10.22533/at.ed.02419151011</b>	
<b>CAPÍTULO 12</b> .....	<b>140</b>
DESENVOLVIMENTO DE UMA ABORDAGEM DE CLASSIFICAÇÃO DE TRAÇOS DE PERSONALIDADE DO BIG-FIVE VIA ANÁLISE DE TEXTOS EM PORTUGUÊS DO BRASIL	
Angelo Travizan Neto Taís Borges Ferreira Márcia Aparecida Fernandes	
<b>DOI 10.22533/at.ed.02419151012</b>	
<b>CAPÍTULO 13</b> .....	<b>148</b>
CODETEACHER: UMA FERRAMENTA PARA CORREÇÃO AUTOMÁTICA DE TRABALHOS ACADÊMICOS DE PROGRAMAÇÃO EM JAVA	
Francisco Alan de Oliveira Santos Plácido Soares das Chagas Segundo Mardoqueu Sousa Telvina	
<b>DOI 10.22533/at.ed.02419151013</b>	
<b>CAPÍTULO 14</b> .....	<b>158</b>
CLASSIFICAÇÃO DE TENDÊNCIAS POLÍTICAS EM NOTÍCIAS VIA MINERAÇÃO DE TEXTO E REDES NEURAIS SEM PESO	
Rafael Dutra Cavalcanti	
<b>DOI 10.22533/at.ed.02419151014</b>	

<b>CAPÍTULO 15</b> .....	<b>169</b>
ANÁLISE DE DADOS DE EXPRESSÃO GÊNICA POR MEIO DE MÉTODOS DE AGRUPAMENTO DIFUSO MULTIVARIADO	
Bruno Almeida Pimentel	
Marcilio Carlos Pereira de Souto	
DOI 10.22533/at.ed.02419151015	
<b>SOBRE O ORGANIZADOR</b> .....	<b>182</b>
<b>ÍNDICE REMISSIVO</b> .....	<b>183</b>

## SIMULAÇÃO DE ÁREA DE CONVERGÊNCIA NUVEM-HPC

**Alessandro Kraemer**

Universidade Tecnológica Federal do Paraná  
Departamento de Computação  
Campo Mourão – Paraná

**RESUMO:** O estudo de sistemas de computação em larga escala frequentemente requer avaliação por meio de simulação. Entretanto, alguns mecanismos podem não estar disponíveis nas ferramentas de simulação. O desenvolvimento desses mecanismos pode ser uma tarefa complicada. O tema área de convergência nuvem-HPC foi recentemente apresentado à comunidade científica, e devido a isso seus recursos não estão implementados em ferramentas de simulação. No contexto de convergência, escalonadores de centrais de dados realizam previsão de tempo de resposta e usam fluxos de controle para migração de aplicação visando reduzir o número de violações. Neste artigo é apresentada uma arquitetura de sistemas de simulação para área de convergência. Apresentam-se também um simulador desenvolvido e os modelos de representação de plataforma e aplicação. Uma simulação é avaliada por meio de um experimento *in silico*.

**PALAVRAS-CHAVE:** área de convergência, centrais de dados, simulação.

**SIMULATION OF CLOUD-HPC  
CONVERGENCE AREA**

**ABSTRACT:** The study of large-scale computing systems often mandates evaluation by simulation. However, some mechanisms may not be available in simulation tools. The development of those mechanisms can be a complicated task. The topic of the cloud-HPC convergence area was presented recently for the scientific community, and due to it, their resources are not implemented in simulation tools. In the context of the convergence area, datacenters' schedulers predict response times for applications and use control flows for application migration to reduce the number of violations. This paper shows a simulation system architecture for the convergence area. It also shows a simulator developed and the models for platform and application representation. A simulation is evaluated by an *in silico* experiment.

**KEYWORDS:** convergence area, datacenter, simulation.

### 1 | INTRODUÇÃO

A Computação de Alto Desempenho (*HPC – High Performance Computing*) vem permitindo à ciência avançar rapidamente em muitas áreas do conhecimento. Por sua vez, as tecnologias de computação em nuvem

proporcionam o acesso a recursos computacionais a muitos usuários. Ambas as tecnologias proporcionam ambientes de computação de larga escala em centrais de dados, embora sejam usadas para propósitos distintos. Enquanto tecnologias de nuvem proporcionam computação diversa ao público como meio de negócio, tecnologias puramente HPC têm objetivo científico e seus usuários são especializados.

Em centrais de dados é frequente a implantação separada de plataformas HPC e de nuvem em um mesmo supercomputador. Em geral, cada plataforma gerencia uma fila de entrada de aplicações, tem seu próprio domínio de processadores, memória e rede de comunicação.

A fim de reduzir o número de violações de tempo de resposta de aplicações em centrais de dados, foi proposto em Kraemer et al. (2016) uma área de convergência nuvem-HPC. Para experimentos de simulação nesse contexto foi desenvolvido o CArS (*Convergence Area Scheduler* – Escalonador de Área de Convergência), que é apresentado neste artigo.

Apresentam-se a área de convergência na Seção 2, alguns programas simuladores na Seção 3, uma descrição geral do simulador CArS na Seção 4, modelos de definição de plataforma e aplicações na Seção 5, o gerenciamento de aplicações e o escalonamento realizados pelo simulador na Seção 6, seguidos de um experimento de avaliação onde visa-se reduzir o número de violações de tempo de resposta em um cenário de convergência nuvem-HPC.

## 2 | ÁREA DE CONVERGÊNCIA NUVEM-HPC

Define-se área de convergência nuvem-HPC em centrais de dados como um conjunto delimitado de processadores no qual está habilitada a migração de aplicações de plataformas de nuvem para plataformas HPC. Cada plataforma gerencia sua própria fila de entrada de aplicações e possui um escalonador. Uma área de convergência pode ser estática ou dinâmica em termos de número de processadores, podendo envolver processadores de partições de nuvem, partições de HPC e processadores fora dessas partições.

Os fluxos de controle de aplicação na Central de Dados determinam rotas para aplicações na área de convergência, sendo importantes à tomada de decisão de cada escalonador. Os escalonadores envolvidos possuem funções adicionais que permitem prever o tempo de resposta de aplicações e tomar decisões sobre migração considerando esses fluxos. O tempo de resposta é o principal critério em área de convergência. Os fluxos possíveis e dados necessários à tomada de decisão sobre aplicações são apresentados em Kraemer (2017).

### 3 | TRABALHOS RELACIONADOS

A área de convergência foi introduzida à comunidade científica por meio dos trabalhos Kraemet et al. (2016) e Kraemer et al. (2017). A comparação entre simuladores é uma tarefa difícil, devido a esse conhecimento ser recente e outros simuladores não implementarem características específicas de gerenciamento de tempo.

Considera-se que um programa simulador de área de convergência deve ter capacidade de previsão de tempo de resposta de aplicação no grid, por meio de escalonadores no Sistema de Gerenciamento de Tarefas e Recursos (RJMS – *Resource Job and Management System*). Neste sentido, buscou-se nesta Seção comparar o CARs em relação a outros simuladores, verificando principalmente o gerenciamento de tempo de resposta desses simuladores.

Casanova et al. (2014) apresentam várias abordagens de simuladores de aplicações e plataformas. Nesse estado da arte destacam-se os simuladores PSINS, LogGOPSim e BigSim, por apresentarem a trajetória de execução da aplicação. Essa representação torna possível a implementação de verificação de tempo de resposta. O BigSim, em especial, consegue prever desempenho de aplicação. Porém, esses simuladores não representam grids de computação, que os tornam limitados de representação de área de convergência. Outra limitação está relacionada com o tipo de aplicação. Esses simuladores representam aplicações MPI, enquanto na área de convergência não há tipo específico de arcabouço considerado.

Outros simuladores bem conhecidos na atualidade, tais como CloudSim e GridSim, não apresentam a trajetória da aplicação, impedindo dessa maneira o gerenciamento de tempo de resposta de cada aplicação. A tabela 1 apresenta um resumo das características necessárias à área de convergência e as compara com os recursos disponíveis em outros simuladores.

Simulador	Computação em grid	Trajétoria de execução da aplicação	Previsão de tempo de resposta	Verificação de violação de tempo	Representação nuvem-HPC
PSINS		X			
LogGOPSim		X			
BigSim		X			
CloudSim	X		X		
GridSim	X				
CARs	X	X	X	X	X

Tabela 1. Comparação de simuladores considerando características de área de convergência.

## 4 | DESCRIÇÃO GERAL DO SIMULADOR CARS

O simulador desenvolvido para representar uma área de convergência nuvem-HPC chama-se CARs (***Convergence Area Scheduler*** - Escalonador de Área de Convergência), disponível em <https://gforge.inria.fr/projects/cars/>. Esse programa é uma ramificação do RJMS OAR para escalonamento em lote e está sobre a licença do *Institut National de Recherche en Informatique et en Automatique* (INRIA). Sugere-se o trabalho Margery et al. (2014) para entendimento do OAR. Detalham-se a seguir a arquitetura do simulador CARs e suas principais características.

### 4.1 Arquitetura do Sistema

As plataformas de computação simuladas utilizadas pelo CARs são criadas por meio de um simulador base conhecido como Simgrid (*Grid Simulator* - Simulador de grids). Para prover a comunicação entre um escalonador e o Simgrid, pesquisadores do INRIA - MOAIS (*prograMming and scheduling design fOr Applications in Interactive Simulation* - Projeto de Escalonamento e Programação para Aplicações em Simulação Interativa) desenvolveram o BatSim (*Batch Scheduler Simulator* - Escalonador de Aplicações em Lote). Nessa arquitetura, o CARs é um escalonador incorporado ao BatSim.

### 4.2 Caracterização do Simgrid

O Simgrid surgiu em 1999 para simular escalonamentos offline de gráficos (CASANOVA et al., 2014). A classe de experimentação suportada por este simulador é *in silico*, caracterizada por utilizar dados reais e não apenas dados de simulações matemáticas. Uma característica de destaque nesse simulador é a capacidade de representar toda a trajetória temporal da aplicação na plataforma simulada. Atualmente, o tSimgrid tem suporte para simulação de grids, nuvens, HPC, P2P (*Peer-to-Peer* - Ponto-a-Ponto) e MPI (Message Passing Interface - Interface de

Passagem de Mensagem). As simulações são escritas em linguagem Java, C, Lua ou Ruby. A maior parte dos seus programas é distribuída com a licença L-GPL 2.1. (*Lesser General Public License* - Licença Pública Geral Menor).

### 4.3 Caracterização do Batsim

O BatSim (*Batch Scheduler Simulator* - Escalonador de Aplicações em Lote) é um simulador de RJMS (*Resource and Job Management System* - Sistema Gerenciador de Tarefas e Recursos) que habilita incorporação de um escalonador externo (DUTOT et al., 2016). O desenvolvimento do BatSim foi iniciado no INRIA em 2015 e está codificado em linguagem C++. Sua validação foi realizada em ambiente real, na central de dados Grid'5000 do governo francês.

O BatSim é orientado a eventos, sendo que existem somente dois eventos tratados no sistema: um para indicar que uma nova aplicação foi recebida na fila de espera e outro para indicar o término dela. Um escalonador incorporado considera esses eventos para tomadas de decisão. De um lado, o BatSim integra-se com o Simgrid para instanciar plataformas de computação; e de outro lado, ele abre um *socket unix* para comunicar-se com um escalonador incorporado, que passa a gerenciar as aplicações. O CARs é um exemplo de escalonador incorporável. Essa integração tem como objetivo tornar mais flexível, em termos de linguagem de programação e definição de aplicações, o desenvolvimento e a experimentação de novos escalonadores.

O protocolo de comunicação entre o BatSim e o escalonador incorporado define o formato de uma mensagem como `ILT | DATA: TIPO_OPERAÇÃO : OPERAÇÃO`, onde:

- **ILT** é o intervalo na linha de tempo (ou relógio do sistema), com especificação de início e fim;
- **DATA** é o instante exato para execução da operação;
- **TIPO\_OPERAÇÃO** é um parâmetro que depende de qual programa está enviando a mensagem. Caso seja o escalonador incorporado, têm-se os parâmetros: J (para alocar aplicação) ou R (para rejeitar aplicação). Caso seja uma mensagem do BatSim, tem-se: NOP (nenhuma operação sendo executada no momento), S (submissão de aplicação) ou C (aplicação completa/finalizada).

Para exemplificar uma mensagem, segue um caso onde o escalonador incorporado solicita a execução da aplicação de código 1 nos processadores {0,1,2} e a aplicação de código 2 no processador {3}. Assim, mais de uma aplicação pode ser informada no intervalo de tempo da mensagem entre 0 e 25.836709. As aplicações de código 1 e 2 serão executadas no instante 25.836709, como representado nesta mensagem:

#### 4.4 Caracterização do Cars

O CArS é um escalonador incorporado ao BatSim. Esse simulador visa escalonar aplicações de nuvem e de HPC, assim como prever tempo de resposta e verificar violações. O desenvolvimento do CArS foi iniciado em 2016 como projeto de tese de doutorado. Para uso de simulação de centrais de dados são também necessários os arcabouços BatSim e Simgrid. A integração desses sistemas está representada na Figura 1. Enquanto o Simgrid simula os recursos computacionais, tais como processador, memória principal, rede de comunicação e outros, o BatSim atua como um *middleware* do Simgrid e como um RJMS, habilitando integração de um novo escalonador. O CArS incorpora-se ao BatSim via *socket unix* e recebe dados de eventos do ambiente, tal como a mensagem apresentada na Seção 4.3.

Os experimentos executados nessa integração de arcabouços são reprodutíveis. Nesse contexto, as aplicações são sempre recebidas no seu tempo programado e utilizam-se constantes de tempo que impactam no tempo de processamento delas. Isso significa que os resultados são repetidos em novas simulações que utilizam a mesma carga de trabalho e a mesma configuração de ambiente.

## 5 | MODELOS DE SIMULAÇÃO

Para simulação de uma central de dados é necessária a definição de um arquivo de plataforma e a definição de um arquivo de carga de trabalho. A definição de plataforma é realizada em formato XML, tal como o exemplo de configuração apresentado na Figura 2. A carga de trabalho é definida em formato JSON, tal como apresentado na Figura 3.

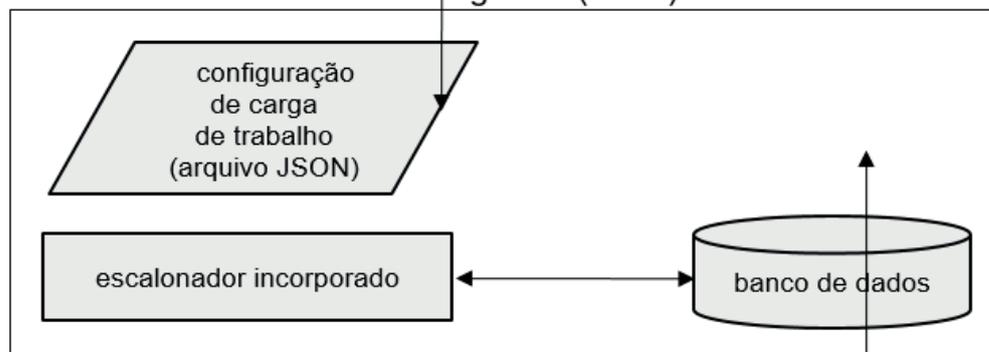
```

<?xml version='1.0'?>
<!DOCTYPE platform SYSTEM "http://simgrid.gforge.inria.fr/simgrid.dtd">
<platform version="3">
<AS id="AS0" routing="Full">
  <host id="master_host" power="98.095Mf"/>
  <host id="Jupiter" power="76.296Mf"/>
  ...
  <link id="6" bandwidth="41.279125Mbps" latency="59.904us"/>
  ...
  <route src="master_host" dst="master_host"><link_ctn id="loopback"/></route>
  <route src="Jupiter" dst="Jupiter"><link_ctn id="loopback"/></route>
  <route src="master_host" dst="Jupiter">
    <link_ctn id="6"/>
  </route>
  ...
</AS>
</platform>

```

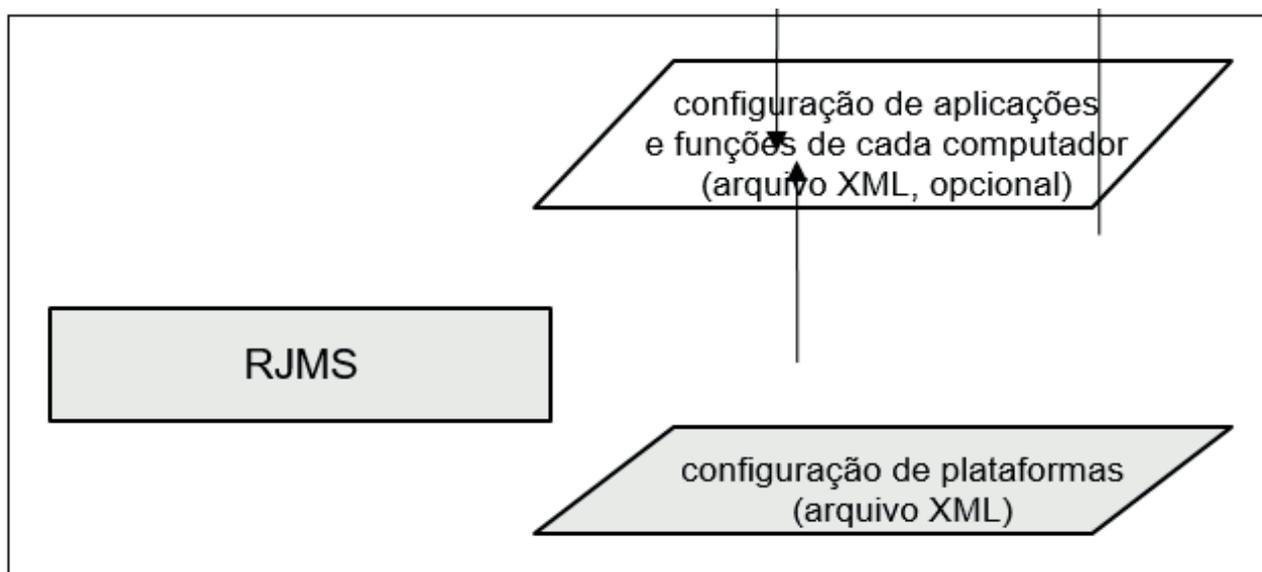
Figura 2. Exemplo de arquivo XML para configuração de uma plataforma no BatSim e no Simgrid. Neste exemplo constam as definições dos *hosts master* e *jupiter* em termos de *megaflop*, assim como a definição do enlace que os conecta e da largura de banda.

### Escalonador de área de convergência (CArS)

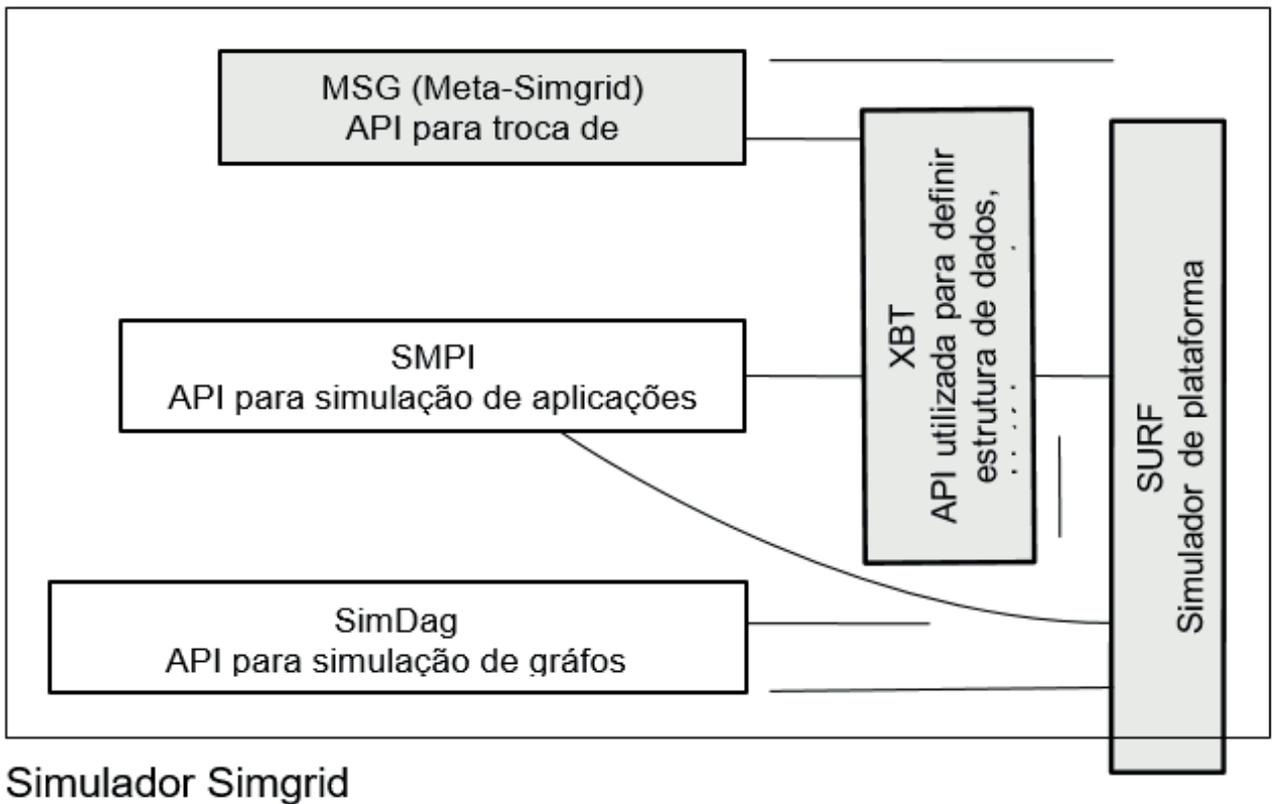


socket Unix  
(protocolo próprio)

### Simulador BatSim



importação de bibliotecas



### Simulador Simgrid

Figura 1. Integração de arcabouços para simulação de escalonamento de aplicações com verificação de violação de tempo de resposta.

```

...
{
  "nb_res": 4,
  "jobs": [
    { "id": 1, "subtime": 10, "walltime": 100, "max_rt_slo": 300, "res": 3, "profile": "2" },
    { "id": 2, "subtime": 20, "walltime": 100, "max_rt_slo": 300, "res": 1, "profile": "2" },
    { "id": 3, "subtime": 30, "walltime": 3, "max_rt_slo": 30, "res": 2, "profile": "2" },
    { "id": 4, "subtime": 32, "walltime": 100, "max_rt_slo": 400, "res": 4, "profile": "2" }
  ],
  "profiles": {
    "1": {
      "type": "msg_par",
      "cpu": [5e6, 5e6, 5e6, 5e6],
      "com": [5e6, 5e6, 5e6, 5e6, 5e6, 5e6, 5e6, 5e6]
    },
    "2": {
      "type": "msg_par_hg",
      "cpu": 10e6,
      "com": 1e6
    }
  }
}

```

*Número total de recursos que uma aplicação pode requisitar. Um recurso é representado aqui como um processador.*

*Perfil da aplicação, como definido abaixo.*

*Instante de submissão da aplicação na linha de tempo.*

*Objetivo de Nível de Serviço de máximo tempo de resposta.*

*Definição de perfil de arquitetura paralela com processadores heterogêneos. A unidade para CPU é expressa em Mbfllops e para a rede de comunicação em MBps.*

*Definição de perfil de arquitetura paralela com processadores homogêneos.*

*Número de recursos requisitados pela aplicação.*

Figura 3. Exemplo de arquivo JSON de configuração de carga de trabalho.

O formato padrão para especificação de cargas de trabalho em grids é o SWF (*Standard Workload Format* - Formato Padrão de Carga de Trabalho). Entretanto, o BatSim e o CArS adotam a especificação apresentada na Figura 3, em formato JSON. Uma razão para isto é que no formato JSON foram acrescentados mais campos sobre aplicações, tal como o limite de tempo de resposta e o perfil da aplicação (i.e. paralela, homogênea ou heterogênea). Assim, para simulação de área de convergência, o arquivo SWF deve ser convertido para JSON. Para realizar essa conversão, utiliza-se o *script swf2json.py*, disponível no CArS. Recomenda-se o repositório de cargas de trabalho SWF mantido por Feitelson (2019), por representar histórico de aplicações de diversos grids reais. Destaca-se que no CArS não há necessidade de especificação do tipo de arcabouço para execução da aplicação, tal como MPI. O arcabouço é, portanto, transparente na simulação.

Após a definição da carga de trabalho em JSON, demais aplicações podem ser inseridas na linha do tempo. As novas aplicações são inseridas em intervalos regulares, por meio do *script insert\_cloud\_job.py*. Dessa maneira, a carga de trabalho em JSON é atualizada, passando a representar aplicações submetidas de maneira esparsa, tal como ocorre em HPC, e também aplicações submetidas em frequência regular, tal como ocorre na nuvem Google (SHENG et al., 2012), por exemplo. Assim, tem-se a representação de aplicações para convergência nuvem-HPC.

Para inicialização de um experimento, basta acionar o BatSim e o CArS, passando como parâmetros o arquivo XML de definição de plataforma e o arquivo JSON de definição de carga de trabalho. Como resultado de um processo de simulação, são criados os seguintes arquivos:

- **out\_jobs.csv:** dados sobre o ciclo de vida da aplicação no sistema;
- **out\_jobs\_violated.csv:** dados sobre o número de aplicações violadas e os detalhes de cada violação;
- **out\_schedule.csv:** dados sobre o número de aplicações atendidas e detalhes do escalonador;
- **out\_violation\_prediction\_error.csv:** dados sobre todas as aplicações com erro de previsão, incluindo a previsão;
- **out\_violation\_prediction\_error\_g.csv:** dados sobre as aplicações violadas que não foram previstas;
- **out\_violation\_prediction\_error\_l.csv:** dados sobre as aplicações que foram previstas como violadas e que terminaram sem violação.

## 6 | GERENCIAMENTO DE APLICAÇÕES E ESCALONAMENTO

No CArS, a primeira etapa de convergência nuvem-HPC consiste na conciliação de propriedades de aplicação. Nesse processo, as propriedades de aplicações de nuvem são ajustadas para o formato aceito na plataforma HPC. No ambiente

simulador, a conciliação é realizada *offline*, durante a inserção de aplicações na carga de trabalho HPC. A carga de trabalho produzida é representada por um arquivo de formato JSON, contendo a configuração dos recursos requisitados, o máximo tempo de resposta SLO e o momento de submissão de cada aplicação, tal como apresentado na Figura 3. Com a inicialização do programa simulador, lê-se o arquivo XML de configuração de plataformas e a carga de trabalho JSON. Em seguida, a fila FCFS começa a receber as aplicações no seu tempo programado e a estratégia de escalonamento coleta e organiza essas aplicações nos processadores da plataforma.

Um ciclo de escalonamento inicia-se quando uma aplicação chega na fila FCFS (*First Come First Served* – Primeiro a chegar Primeiro a ser atendido) ou quando uma aplicação termina (esses são os eventos  $S$  e  $C$  discutidos na Seção 4.3.). Uma vez escalonada e iniciada a execução de uma aplicação, o término dela é antecipado, a fim de representar um cenário onde aplicações terminam mais cedo que o previsto. Essa técnica é usual em simulações de escalonamento realizadas para publicações científicas do grupo MOAIS. Ao término da aplicação, verifica-se a violação.

Todas as informações das aplicações escalonadas são armazenadas em um banco de dados e o núcleo do sistema CArS extrai um negativo das alocações, ou seja, espaços na linha de tempo onde não existe aplicação escalonada. Esses espaços são os *slots* de tempo ocioso. Com base nos slots disponíveis, o escalonador realiza o escalonamento de novas aplicações. A operacionalização do banco de dados de alocações é totalmente transparente e a determinação de escalonamento é realizada com base nos slots, em alto nível, via linguagem Python.

O Algoritmo 1 a seguir, em pseudo-código, mostra o escalonamento de uma aplicação  $J$  de índice  $j$  definida como  $J_j = (a_j, wt_j, h_j, SLO_{j,i})$ , onde  $a_j$  representa o momento de chegada da aplicação na Central de Dados  $wt_j$  o tempo de processamento requisitado para cada processador,  $h_j$  o número de processadores requisitado e  $SLO_{j,i}$  o  $i$  máximo tempo de resposta acordado para  $j$ . Para a determinação de um escalonamento, adiciona-se a propriedade para  $\sigma_j$  indicar o momento de início da aplicação e substitui-se a propriedade  $h_j$  por  $H_j$  para representar o subconjunto de processadores  $P$  que estão aptos para iniciar a aplicação em  $\sigma_j$ . O início simultâneo em  $\sigma_j$  representa a rigidez da aplicação. As propriedades  $a_j$  e  $SLO_{j,i}$  não são consideradas na determinação do escalonamento, pois todas as aplicações são escalonadas. No evento de término da aplicação computa-se  $(a_j + SLO_{j,i}) \leq C_j$  para determinar se houve violação, onde  $C_j$  é o momento de completude da aplicação.

- 1 coleta-se  $J_j = (a_j, wt_j, h_j, SLO_{j,i})$  da FCFS
- 2 obtem-se o conjunto  $S$  de *slots* de tempo ocioso
- 3 ordena-se  $S$  pelo menor momento de início primeiro
- 4 para cada  $S_z$  em  $S$ :
  - 5 *se*  $termino_z - inicio_z + 1 \gg wt_j$ :
  - 6 *se* tamanho de  $H_z \gg h_j$ :
    - 7  $H_j \leftarrow P \subseteq H_z$
    - 8  $\sigma_j \leftarrow inicio_z$
- 9 encerra-se o *loop*
- 10 escalona-se  $J_j \leftarrow (\sigma_j, wt_j, H_j, SLO_{j,i})$

Algoritmo 1. Escalonamento de uma aplicação.

## 7 | EXPERIMENTO DE AVALIAÇÃO

Como experimento de avaliação de uma simulação, definiram-se diversos cenários de área de convergência onde aplicações de nuvem são inseridas na fila de aplicações de uma plataforma HPC. Variou-se o tamanho da plataforma HPC considerando 10, 20, 40, 80, 160 e 320 processadores. Definiu-se que a área de convergência (em termos de número de processadores) abrange todos os processadores da plataforma HPC. Definiu-se uma carga de trabalho in silico com 11K aplicações HPC e 270K aplicações de nuvem. As aplicações HPC foram extraídas do ambiente de produção do PIK por meio do repositório de Feitelson (2019), contendo diversas configurações de aplicações paralelas e frequência de submissão esparsa. Enquanto isso, as aplicações de nuvem foram definidas conforme documentação publica do Google (REISS et al., 2012), com frequência regular de submissão de uma aplicação a cada 10 segundos. As aplicações de nuvem selecionadas para execução na área de convergência requisitam 4 processadores e 180 segundos de tempo de processamento. Essas aplicações são ditas candidatas à migração, de acordo com as especificações de área de convergência de Kraemer (2017). Variou-se a configuração do máximo tempo de resposta considerando um fator de folga composto de 1,25; 1,5; 2; 3; 4 até 15. Por exemplo, um fator de folga 2 significa que

o máximo tempo de resposta para caracterizar um processamento sem violação é de 360 segundos (ou seja,  $180 \times 2$ ). Como resultado da simulação, objetiva-se identificar o percentual de violações ocorridas nesse contexto de convergência nuvem-HPC.

## 8 | RESULTADOS E ANÁLISES

De acordo com Hemmat e Hafid (2016), a média percentual de aplicações violadas em sistemas de computação em larga escala é de 0,2%. Nos resultados apresentados na Figura 4, o pior cenário de violação foi observado quando a aplicação estava configurada com 1,25 fator de folga (ou seja, uma aplicação que requer execução urgente) e uma plataforma com 10 processadores disponíveis para o escalonador. Nesse cenário o percentual de violações foi de 0,0018%, que é bem abaixo do observado na literatura. Uma razão para isso é que as aplicações de nuvem migradas para HPC foram selecionadas de forma a aproveitar os espaços ociosos do escalonamento de aplicações HPC. Essa é uma característica da área de convergência. Observa-se também na figura, como esperado, que quanto maior o fator de folga e mais processadores disponíveis, menor é o percentual de violações. Um fator de folga 15 é suficiente para evitar todas as violações no cenário analisado.

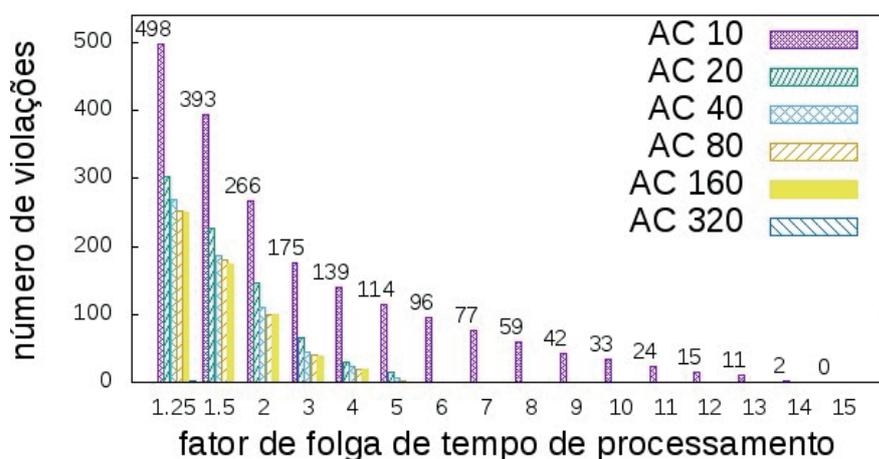


Figura 4. Resultado dos experimentos para cada tamanho de AC (Área de Convergência).

## 9 | CONCLUSÕES E TRABALHOS FUTUROS

Neste artigo foi apresentado o simulador CArS para área de convergência nuvem-HPC, sua arquitetura e integração com sistemas base. Também foram apresentados os modelos requisitados para simulação de plataformas e aplicações, assim como a estratégia de escalonamento de uma aplicação. Para avaliação via experimentos, foi considerado um cenário com 11K aplicações HPC e 270K aplicações de nuvem, representando uma simulação *in silico*, ou seja, com dados reais. O resultado dos experimentos mostra que a área de convergência nuvem-HPC tem potencial para

*diminuir o número de violações de tempo de resposta. O simulador CARs mostrou-se eficiente para esse contexto de simulação. Para trabalhos futuros, considera-se o escalonamento de aplicações moldáveis.*

## REFERÊNCIAS

- CASANOVA, H., GIERSCH, A., LEGRAND, A., QUINSON, M. e SUTER, F. **Versatile, scalable, and accurate simulation of distributed applications and platforms**. Journal of Parallel and Distributed Computing, 2014. 74(10):2899–2917.
- DUTOT, P., MERCIER, M., POQUET, M. e RICHARD, O. **BatSim: a realistic language-independent resources and jobs management systems simulator**. Em 20th Workshop on Job Scheduling Strategies for Parallel Processing, Chicago, United States, 2016.
- FEITELSON, D. **Logs of real parallel workloads from production systems**. <http://www.cs.huji.ac.il/labs/parallel/workload/logs.html>. Acessado em 25/08/2019.
- HEMMAT, R. A. & HAFID, A. **SLA violation prediction in cloud computing: A machine learning perspective**. CoRR, abs/1611.10338, 2016.
- KRAEMER, A., MAZIERO, C., RICHARD, O. e TRYSTRAM, D. **Reducing the number of response time SLO violations by a cloud-HPC convergence scheduler**. Em 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), 2016. p. 293–300.
- KRAEMER, A. **Proposta de uma área de convergência Nuvem-HPC para a redução do número de violações de tempo de resposta em centrais de dados: tese de doutorado**. Curitiba: UFPR, 2017. 128 p.
- KRAEMER, A., MAZIERO, C. A., RICHARD, O. e TRYSTRAM, D. **Reducing the number of response time service level objective violations by a cloud-HPC convergence scheduler**. Concurrency and Computation: Practice and Experience, 2017. p. e4352–n/a.
- MARGERY, D., MOREL, E., NUSSBAUM, L., RICHARD, O. e ROHR, C. **Resources Description, Selection, Reservation and Verification on a Large-Scale Testbed**. Springer International Publishing, Cham, 2014. p. 239–247.
- REISS, C., TUMANOV, A., GANGER, G. R., KATZ, R. H. e KOZUCH, M. A. **Towards understanding heterogeneous clouds at scale: Google trace analysis**. Relatório Técnico ISTC-CC-TR-12-101, Intel Science and Technology Center for Cloud Computing, Carnegie Mellon University, Pittsburgh, PA 15213-3890, 2012.
- SHENG, D., KONDO, D. e CIRNE, W. **Characterization and comparison of cloud versus grid workloads**. Em IEEE International Conference on Cluster Computing (CLUSTER), 2012. p. 230–238.

## ÍNDICE REMISSIVO

### A

Abaqus 40, 41, 43, 46, 48  
Agrupamento 81, 140, 169, 170, 175, 176  
Aplicação 5, 6, 28, 31, 33, 34, 37, 49, 50, 51, 52, 53, 58, 59, 60, 61, 68, 70, 80, 83, 89, 129, 135, 148, 150, 152, 157, 169, 174  
Aplicativo móvel 1, 7, 88  
Aprendizado online 98, 100  
Aprendizagem 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 74, 75, 87, 88, 89, 90, 92, 93, 94, 95, 96, 97, 100, 101, 140, 147, 148, 149, 150, 154, 156  
Aprendizagem colaborativa 89, 95, 140  
Área de convergência 49, 50, 51, 52, 58, 60, 61, 62  
Atenção visual 98, 101, 103, 106, 107, 110  
Avaliação automática 148, 149

### B

Banco de dados 8, 59, 66, 76, 77, 79, 81, 82, 84, 85, 86, 170, 182  
Busca visual 98, 99, 100, 101, 102, 107, 110

### C

Carteira estudantil 1  
Celular 87, 88, 90, 91, 96, 179  
Centrais de dados 49, 50, 54, 62  
Coeficiente de impacto 125, 130, 138  
Condicionamento de Ar 16  
Custos 30, 31, 32, 36, 37

### D

Design constructal 16, 18, 19, 28  
Desvio funcional 76  
Difuso 169

### E

Efeitos dinâmicos 125, 126, 127, 130, 136, 137  
Energias renováveis 16  
Ensino 2, 34, 38, 63, 64, 74, 87, 88, 90, 93, 94, 95, 96, 97, 148, 149, 156  
Espectroscopia 112, 114, 115, 116, 118, 121, 122  
Etilômetro 112, 113, 114, 116  
Exercícios de programação 148, 156  
Expressão gênica 169, 171, 174

## **G**

Gestão pública 76, 85

## **I**

Infravermelho 112, 114, 115, 116, 117, 118, 121

## **J**

Java 53, 148, 149, 150, 151, 156

## **L**

LIWC 140, 143, 144, 146

## **M**

Materiais compósitos 40, 41, 42, 43, 44, 46, 47

Mineração de textos 158, 166, 167

Mobile learning 87, 88, 89, 91, 93, 95, 96, 97

Modelagem computacional 16, 40, 42, 125, 138

Modelo de estilo de aprendizagem 63, 65, 69

Modelos de cargas móveis 125

Multivariado 169, 172, 176, 178

## **P**

Pontes rodoviárias 125, 126, 138, 139

Processamento de linguagem natural 140, 159

Processo ensino-aprendizagem 63

## **R**

Recomendação 63, 64, 67, 70, 71, 72, 73

Recurso educacional aberto 63, 64

Recursos humanos 76, 79, 86

Redes 7, 30, 31, 32, 33, 35, 37, 38, 39, 81, 97, 158, 161, 167

Redes neurais sem peso 158

Robótica 98, 99, 100, 101, 110

## **S**

Simulação 5, 13, 14, 40, 41, 43, 49, 50, 52, 54, 57, 58, 60, 61, 62, 68, 102

Simulação numérica 40, 41, 43

Sistemas de informação 76, 77, 79, 80, 86, 182

## **T**

Tecnologia 1, 30, 32, 33, 34, 38, 39, 63, 90, 97, 112, 176, 182

Telefonia 30, 31, 32, 33, 34, 35, 37, 38, 39

Transporte público 1, 2, 3, 5, 6, 7, 15

Trocadores de Calor Solo-Ar (TCSA) 16, 17

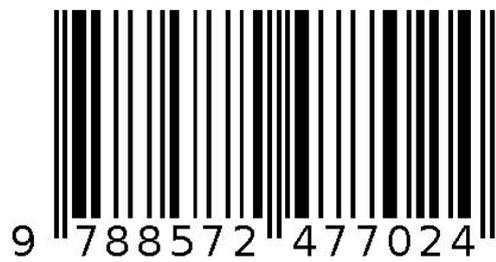
## **V**

VoIP 30, 32, 33, 34, 38, 39

## **W**

Whatsapp 87, 91, 96, 97

Agência Brasileira do ISBN  
ISBN 978-85-7247-702-4



9 788572 477024