



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## **Escalonador GPU Aware para a Plataforma de Nuvens Federadas BioNimbuZ**

Francisco Anderson Bezerra Rodrigues

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientador

Prof.a Dr.a Aletéia Patrícia Favacho de Araújo

Brasília  
2018



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## **Escalonador GPU Aware para a Plataforma de Nuvens Federadas BioNimbuZ**

Francisco Anderson Bezerra Rodrigues

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Prof.a Dr.a Aletéia Patrícia Favacho de Araújo (Orientador)  
CIC/UnB

Prof. Dr. Goku     Dr. Bruce Wayne  
Planeta Vegeta     Wayne Enterprises

Prof. Dr. Rodrigo Bonifácio de Almeida  
Coordenador do Bacharelado em Ciência da Computação

Brasília, 32 de Onzembro de 2018

# Dedicatória

Na *dedicatória* o autor presta homenagem a alguma pessoa (ou grupo de pessoas) que têm significado especial na vida pessoal ou profissional. Por exemplo (e citando o poeta):  
*Eu dedico essa música a primeira garota que tá sentada ali na fila. Brigado!*

# Agradecimentos

Nos *agradecimentos*, o autor se dirige a pessoas ou instituições que contribuíram para elaboração do trabalho apresentado. Por exemplo: *Agradeço aos gigantes cujos ombros me permitiram enxergar mais longe. E a Google e Wikipédia.*

# Resumo

O *resumo* é um texto inaugural para quem quer conhecer o trabalho, deve conter uma breve descrição de todo o trabalho (apenas um parágrafo). Portanto, só deve ser escrito após o texto estar pronto. Não é uma coletânea de frases recortadas do trabalho, mas uma apresentação concisa dos pontos relevantes, de modo que o leitor tenha uma ideia completa do que lhe espera. Uma sugestão é que seja composto por quatro pontos: 1) o que está sendo proposto, 2) qual o mérito da proposta, 3) como a proposta foi avaliada/validada, 4) quais as possibilidades para trabalhos futuros. É seguido de (geralmente) três palavras-chave que devem indicar claramente a que se refere o seu trabalho. Por exemplo: *Este trabalho apresenta informações úteis a produção de trabalhos científicos para descrever e exemplificar como utilizar a classe L<sup>A</sup>T<sub>E</sub>X do Departamento de Ciência da Computação da Universidade de Brasília para gerar documentos. A classe UnB-CIC define um padrão de formato para textos do CIC, facilitando a geração de textos e permitindo que os autores foquem apenas no conteúdo. O formato foi aprovado pelos professores do Departamento e utilizado para gerar este documento. Melhorias futuras incluem manutenção contínua da classe e aprimoramento do texto explicativo.*

**Palavras-chave:** Computação em nuvem, Federação de nuvens, escalonamento, GPGPU, BioNimbuZ

# Abstract

O *abstract* é o resumo feito na língua Inglesa. Embora o conteúdo apresentado deva ser o mesmo, este texto não deve ser a tradução literal de cada palavra ou frase do resumo, muito menos feito em um tradutor automático. É uma língua diferente e o texto deveria ser escrito de acordo com suas nuances (aproveite para ler [http://dx.doi.org/10.6061/2Fclinics%2F2014\(03\)01](http://dx.doi.org/10.6061/2Fclinics%2F2014(03)01)). Por exemplo: *This work presents useful information on how to create a scientific text to describe and provide examples of how to use the Computer Science Department's L<sup>A</sup>T<sub>E</sub>X class. The UnB-CIC class defines a standard format for texts, simplifying the process of generating CIC documents and enabling authors to focus only on content. The standard was approved by the Department's professors and used to create this document. Future work includes continued support for the class and improvements on the explanatory text.*

**Keywords:** Cloud Computing, Cloud Federation, schedulling, GPGPU, BioNimbuZ

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
1.2	Estrutura do Trabalho . . . . .	2
<b>2</b>	<b>Escalonamento</b>	<b>3</b>
2.1	Algoritmo proposto . . . . .	3
	<b>Referências</b>	<b>6</b>

# Lista de Figuras

2.1 Diagrama de Funcionamento do Algoritmo de Escalonamento. . . . .	4
----------------------------------------------------------------------	---



# Lista de Abreviaturas e Siglas

**CFS** *Completely Fair Scheduler.*

**CPU** Unidade de Processamento Central, do inglês *Central Processing Unit.*

**GPU** Unidade de Processamento Gráfico, do inglês *Graphics Processing Unit.*

**LABID** Lboratório de Informática e Dados.

**REST** *REpresentational State Transfer.*

# Capítulo 1

## Introdução

A tecnologia tem se tornado cada vez mais ubíqua na sociedade. Com o advento da Internet, a interação dos seres humanos com a tecnologia explodiu, gerando um tráfego imenso de dados, e com isso a necessidade de processamento em larga escala. Nesse cenário, surgiu o conceito de nuvem computacional, um paradigma que permite processamento em larga escala sem ser necessário que o usuário tenha em mãos hardware com tamanha capacidade computacional.

Assim, grandes empresas da área de Tecnologia da Informação, como o *Google* e a *Microsoft*, possuem vários *datacenters* com uma imensa quantidade de computadores interligados via rede, os quais disponibilizam esses recursos de forma virtualizada a usuários que necessitem de processamento e armazenamento em larga escala. A disponibilidade dessa capacidade de computação tem gerado uma revolução na forma como os serviços computacionais são disponibilizados na Internet. Dessa forma, pequenas empresas agora conseguem prover serviços em larga escala sem necessitarem de um grande investimento em infraestrutura computacional e grandes empresas conseguem reduzir custos com equipamentos.

Como existem vários provedores de nuvem e cada um deles tem seus pontos fortes e fracos, surgiu então a ideia de criar uma plataforma que utilize serviços de vários provedores de nuvem, podendo explorar o ponto forte de cada um deles. Assim, emergiu o conceito de Federação de Nuvens, que são plataformas nas quais os usuários conseguem o máximo de flexibilidade provido pela combinação de funcionalidades que os distintos provedores de nuvem disponibilizam a seus usuários.

Todavia, como toda nova tecnologia, ela possui seus próprios desafios, pois desenvolver uma plataforma com tamanha flexibilidade requer uma arquitetura muito bem projetada, implementada e capaz de ser eficiente concomitantemente em que sua interface seja agradável ao usuário. Uma plataforma de Federação de Nuvens que tem sido continuamente desenvolvida é o BioNimbuZ, desenvolvido no Laboratório de Informática e Dados (LA-

BID) da Universidade de Brasília (UnB)[1] [2] [3] [4] [5] [6] [7] [8] por alunos de graduação e pós-graduação.

## 1.1 Objetivos

Este trabalho tem como objetivos:

- Implementar um escalonador capaz de escalonar tarefas para arquiteturas heterogêneas.
- Integrar o escalonador para funcionar no BioNimbuZ, adaptando o BioNimbuZ se necessário.
- Testar o ganho de desempenho obtido ao se utilizar o escalonador desenvolvido na nuvem.

## 1.2 Estrutura do Trabalho

Esse projeto contém mais três capítulos e um apêndice. O segundo capítulo aborda a atividade de escalonamento e propõe o algoritmo que será implementado. O capítulo três descreve a plataforma de nuvens Federadas BioNimbuZ. E o quarto e último capítulo fala como foi o processo de implementação do escalonador.

# Capítulo 2

## Escalonamento

A atividade de escalonamento pode ser otimizada para vários objetivos, entre os quais podemos citar:

- Maximizar quantidade de trabalho realizada por unidade de tempo;
- Minimizar tempo no qual trabalhos ficam esperando para serem executados;
- Minimizar tempo entre um conjunto de trabalhos estarem prontos para serem executados até o fim da execução do conjunto (latência ou tempo de resposta);
- Distribuir de forma justa o tempo que cada um dos trabalhos terão de uso de um recurso escasso.

Esses objetivos são, às vezes, contraditórios. Na prática prioriza-se um conjunto de métricas como base para otimização. Por exemplo o GNU/Linux utiliza o *Completely Fair Scheduler* (CFS), que se baseia no algoritmo *Fair queuing*. Como o nome já diz, o foco desse escalonador está em ser justo. Internamente utiliza-se uma árvore rubro-negra indexada pelo tempo gasto no processador. Para ser justo, o tempo máximo de cada processo fica em execução interrompida é o quociente do tempo que o processo ficou aguardando para ser executado pelo número total de processos.

### 2.1 Algoritmo proposto

O escalonador proposto para implementação segue a ideia básica de escalonamento de listas. Haverão três listas: lista de tarefas a serem feitas, lista de CPUs disponíveis e lista de GPUs disponíveis. A lista de tarefas é ordenada por tempo previsto de execução, que é dado por uma estimativa a partir do programa a ser rodado e do arquivo de entrada. Essa ordenação será em ordem decrescente de tempo previsto. A lista de CPUs disponíveis é ordenada com base na frequência e no número de núcleos. A lista de GPUs tem sua

ordem determinada pela quantidade de operações em pontos flutuante consegue realizar por segundo.

O escalonamento ocorre da seguinte forma, como ilustrado na figura 2.1 :

1. Existem tarefas que podem ser executadas nos recursos disponíveis?
2. Se sim, obtenha o processo que está no topo da lista.
  - (a) É capaz de rodar em GPU?
  - (b) Se sim:
    - i. O tempo prevista para rodá-lo na melhor GPU disponível é melhor que o tempo previsto para rodá-lo na melhor CPU disponível?
    - ii. Se sim, escalone-o para nessa GPU.
    - iii. Se não, escalone-o para na melhor CPU disponível.
  - (c) Se não, escalone-o na melhor CPU disponível.
  - (d) Remova a tarefa e o recurso alocado de suas respectivas listas.
3. Se não, encerre o escalonamento.
4. Volte para a regra 1.

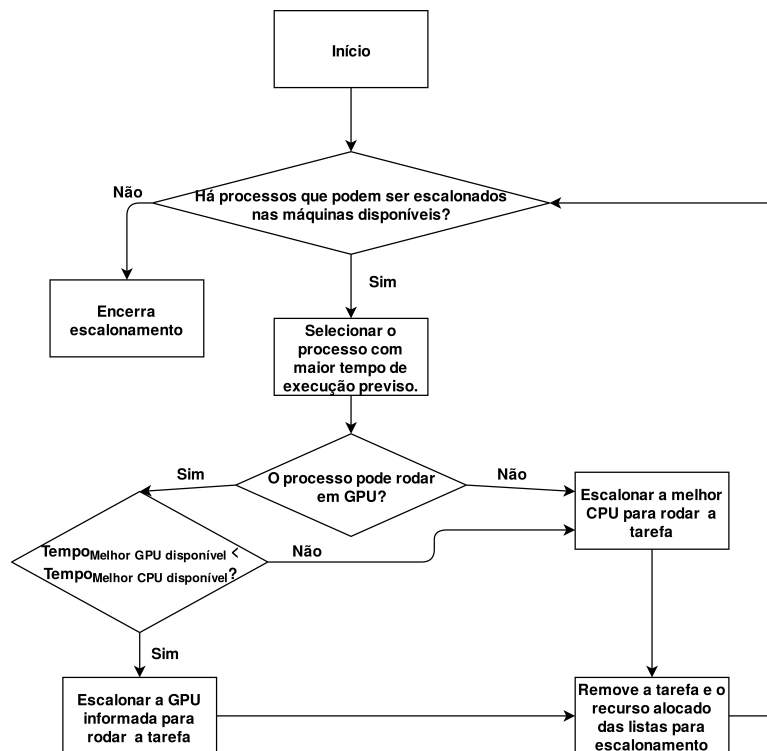


Figura 2.1: Diagrama de Funcionamento do Algoritmo de Escalonamento.

Observa-se que, para o algoritmo supracitado seja válido, pressupõe-se, que toda tarefa do algoritmo é capaz de rodar em CPU. O que é correto, pois atualmente todas as tarefas do BioNimbuZ rodam em CPU. O pressuposto simplifica o primeiro passo do algoritmo, pois se a lista de CPUs não estiver vazia, essa condição é automaticamente satisfeita. Futuramente pode ser necessário adaptá-lo para ser capaz de lidar com tarefas que só são capazes de serem executadas em GPU.

Como trabalho futuro, se o BioNimbuZ obtiver suporte execução de uma mesma tarefa de forma distribuída, uma pequena modificação que pode ser feita no algoritmo supracitado é: ao invés de remover as tarefas escalonadas da lista de tarefas, colocá-las no fim dessa mesma lista. Fará com que o escalonamento seja interrompido apenas quando todos os recursos disponíveis forem alocados, pois sempre haverá tarefas para serem alocadas.

# Referências

- [1] Bacelar, Breno Rodrigues Moura; Deric Lima: *Política para armazenamento de arquivos no zoonimbus*. Biblioteca de monografias da UnB, 2013. Monografia (Licenciatura em Ciência da Computação). 2
- [2] Hugo Saldanha, Edward de Oliveira Ribeiro, Maristela Holanda Aleteia PF Araujo Genaína Nunes Rodrigues Maria Emilia Telles Walter Jo btxfnamespacelong ao Carlos Setubal Alberto MR Dávila: *A cloud architecture for bioinformatics workflows*. CLOSER, 11:477, 2011. 2
- [3] Lima, D., B. Moura, G. Oliveira, E. Ribeiro, A. Araujo, M. Holanda, R. Togawa e M. E. Walter: *A storage policy for a hybrid federated cloud platform: A case study for bioinformatics*. Em *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, páginas 738–747, May 2014. 2
- [4] Saldanha, Hugo, Edward Ribeiro, Carlos Borges, Aletéia Araújo, Ricardo Gallon, Maristela Holanda, Maria Emília Walter, Roberto Togawa e Jo btxfnamespacelong ao Carlos Setubal: *Towards a hybrid federated cloud platform to efficiently execute bioinformatics workflows*. Em Pérez-Sánchez, Horacio (editor): *Bioinformatics*, capítulo 05. InTech, Rijeka, 2012. <http://dx.doi.org/10.5772/50289>. 2
- [5] Oliveira, G. S. S. de, E. Ribeiro, D. A. Ferreira, A. P. F. Araújo, M. T. Holanda e M. E. M. T. Walter: *Acosched: A scheduling algorithm in a federated cloud infrastructure for bioinformatics applications*. Em *2013 IEEE International Conference on Bioinformatics and Biomedicine*, páginas 8–14, Dec 2013. 2
- [6] Oliveira Barreiros Júnior, Willian de: *Escalonador de tarefas para o plataforma de nuvens federadas bionimbuz usando beam search iterativo multiobjetivo*. Biblioteca de monografias da UnB, 2016. Monografia (Bacharelado em Engenharia da Computação). 2
- [7] Borges, C. A. L., H. V. Saldanha, E. Ribeiro, M. T. Holanda, A. P. F. Araujo e M. E. M. T. Walter: *Task scheduling in a federated cloud infrastructure for bioinformatics applications*. Em *Proceedings of the 2nd International Conference on Cloud Computing and Services Science - Volume 1: CLOSER*,., páginas 114–120. INSTICC, SciTePress, 2012, ISBN 978-989-8565-05-1. 2
- [8] Saldanha, Hugo Vasconcelos: *Bionimbus: uma arquitetura de federação de nuvens computacionais híbrida para a execução de workflows de bioinformática*. Tese de

Mestrado, Universidade de Brasília, 2013. <http://repositorio.unb.br/handle/10482/12046>. 2

- [9] Vergara, Guilherme Fay: *Arquitetura de um controlador de elasticidade para nuvens federadas*. 2017.
- [10] Foundation, The Apache Software: *Apache Zookeeper - Home*. <https://zookeeper.apache.org/>, 2017. [Online; acessado em 24 de Novembro de 2017].
- [11] Foundation, The Apache Software: *Welcome to Apache Software Foundation!* <https://apache.org/>, 2017. [Online; acessado em 15 de Dezembro de 2017].
- [12] Foundation, The Apache Software: *Welcome to Apache Avro!* <https://avro.apache.org/>, 2017. [Online; acessado em 24 de Novembro de 2017].