



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. Dr. Aletéia Patrícia Favacho de Araújo

Brasília  
2014



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

Prof. Dr. Donald Knuth    Dr. Leslie Lamport  
Stanford University    Microsoft Research

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

Brasília, 24 de dezembro de 2014

# 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	O problema do escalonamento . . . . .	1
1.2	Unidade de processamento gráfico . . . . .	2
1.3	Computação em Nuvem . . . . .	2
	<b>Referências</b>	<b>4</b>

# Lista de Abreviaturas e Siglas

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

**GPGPU** Unidade de Processamento Gráfico de Propósito Geral, do inglês *General Purpose Graphics Processing Unit*.

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



# Capítulo 1

## Introdução

### 1.1 O problema do escalonamento

O escalonamento é o método pelo qual trabalho, definido por algum conjunto de características, é atribuído à recursos que capazes de completá-lo. O Problema do Escalonamento, que é a busca do escalonamento no qual o tempo de execução do conjunto de trabalhos é mínimo, é NP-completo[1]. De acordo com Sipser[2], um problema A é NP-completo se:

1. A pertence à classe de problemas NP, e
2. Todo problema  $B \in NP$  é redutível em tempo polinomial a A.

Um problema pertence à NP se a for possível verificar a validade de uma solução em tempo polinomial.[2][3].

Por mais que exista a dificuldade teórica supracitada, isso não impediu a evolução dos Sistemas Operacionais, os quais foram capazes de fazer escalonamento de vários processos mesmo na época de Unidade de Processamento Central, do inglês *Central Processing Unit* (CPU)s tinham apenas um núcleo. Popularizando dessa forma os computadores pessoais na década de 80.

Atualmente, as CPUs possuem vários núcleos, e são capazes de ter mais de um contexto carregado por núcleo, vide Ryzen<sup>TM</sup> Threadripper<sup>TM</sup>[4]. O que faz com que seja necessário que o processo de escalonamento leve em consideração como o mesmo será distribuído(ou não) entre os núcleos.

Além do escalonamento de processos para serem executados na CPU, o problema do escalonamento também aparece em várias outras situações na computação:

- Escalonamento de processos que farão swapping.

- Escalonamento de requisições que farão acesso ao disco.
- Escalonamento de pacotes que serão enviados pela rede.
- Escalonamento de tarefas para serem executadas em máquinas virtuais numa nuvem.

Esta monografia focará no último tópico citado. O qual recentemente presenciou a ascensão do uso de unidades de processamento gráfico (Unidade de Processamento Gráfico, do inglês *Graphics Processing Unit* (GPU)) para processamento de propósito geral (Unidade de Processamento Gráfico de Propósito Geral, do inglês *General Purpose Graphics Processing Unit* (GPGPU)) [5][6].

## 1.2 Unidade de processamento gráfico

As GPUs surgiram inicialmente como hardware dedicado embutido para acelerar a renderização de imagens em jogos eletrônicos por volta dos anos 80[7].

Com o passar das décadas e a evolução da tecnologia, as GPUs ficaram cada vez mais poderosas e complexas, principalmente para atender à rápida evolução dos jogos eletrônicos. Quando o computador pessoal popularizou surgiu as GPUs como as conhecemos, como placas acopláveis à placa mãe focadas em processamento gráfico.

Na última década, as unidades de processamento gráfico (GPUs) ascenderam como opção para processamento paralelo. Em especial operações que envolvam matrizes conseguem ter um ganho de performance considerável[8], como processamento de imagens e vídeos[9].

Atualmente provedores de nuvem estão provendo máquinas virtuais com GPUs, como a *Amazon Web Services*[10], o *Google Cloud Platform*[11], *Microsoft Azure*[12] e a *IBM Cloud*[13].

## 1.3 Computação em Nuvem

A Computação em Nuvem é um sistema distribuído que disponibiliza serviços de computação ao usuário abstraindo informações sobre como esse serviço é provido. Existem muitas características que ajudam a definir a Computação em Nuvem, mas as consideradas principais são:

- Os recursos são virtualizados;
- Fácil acesso aos recursos;
- Alta escalabilidade, com capacidade de redimensionamento durante a execução; e,

- Geralmente possui um modelo de pagamento *pay-per-use*.

# Referências

- [1] Ullman, J.D.: *Np-complete scheduling problems*. Journal of Computer and System Sciences, 10(3):384 – 393, 1975, ISSN 0022-0000. <http://www.sciencedirect.com/science/article/pii/S0022000075800080>. 1
- [2] Sipser, Michael: *Introdução à Teoria da Computação*. Cengage Learning, 2007. 1
- [3] Cook, Stephen A.: *The complexity of theorem-proving procedures*. Em *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, páginas 151–158, New York, NY, USA, 1971. ACM. <http://doi.acm.org/10.1145/800157.805047>. 1
- [4] AMD: *Ryzen™ Threadripper™ Processors* . <http://www.amd.com/en/products/ryzen-threadripper>, 2017. [Online; acessado em 21 de Novembro de 2017]. 1
- [5] Dimitrov, Martin, Mike Mantor e Huiyang Zhou: *Understanding software approaches for gpgpu reliability*. Em *Proceedings of 2Nd Workshop on General Purpose Processing on Graphics Processing Units*, GPGPU-2, páginas 94–104, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-517-8. <http://doi.acm.org/10.1145/1513895.1513907>. 2
- [6] Yang, Yi, Ping Xiang, Jingfei Kong e Huiyang Zhou: *A gpgpu compiler for memory optimization and parallelism management*. SIGPLAN Not., 45(6):86–97, junho 2010, ISSN 0362-1340. <http://doi.acm.org/10.1145/1809028.1806606>. 2
- [7] Oguchi, T., M. Higuchi, T. Uno, M. Kamaya e M. Suzuki: *A single-chip graphic display controller*. Em *1981 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, volume XXIV, páginas 170–171, Feb 1981. 2
- [8] Dalton, Steven, Luke Olson e Nathan Bell: *Optimizing sparse matrix-matrix multiplication for the gpu*. ACM Trans. Math. Softw., 41(4):25:1–25:20, outubro 2015, ISSN 0098-3500. <http://doi.acm.org/10.1145/2699470>. 2
- [9] Mulligan, Jeffrey B.: *A gpu-accelerated software eye tracking system*. Em *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, páginas 265–268, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1221-9. <http://doi.acm.org/10.1145/2168556.2168612>. 2
- [10] Services, Amazon Web: *Amazon EC2 Instance Types - Amazon Web Services(AWS)*. <https://aws.amazon.com/ec2/instance-types/>, 2017. [Online; acessado em 21 de Novembro de 2017]. 2

- [11] Plataform, Google Cloud: *Google Compute Engine Pricing / Compute Engine Documentation / Google Cloud Plataform*. <https://cloud.google.com/compute/pricing>, 2017. [Online; acessado em 21 de Novembro de 2017]. 2
- [12] Azure, Microsoft: *Azure Linux VM sizes - GPU*. <https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes-gpu>, 2017. [Online; acessado em 21 de Novembro de 2017]. 2
- [13] Cloud, IBM: *GPU Computing - IBM Cloud* . <https://www.ibm.com/cloud/bare-metal-servers/gpu>, 2017. [Online; acessado em 21 de Novembro de 2017]. 2