



Centro Federal de Educação Tecnológica de Minas Gerais  
Departamento de Computação  
Engenharia de Computação

# COMPARAÇÃO DE ALGORITMOS PARALELOS DE ORDENAÇÃO EM MAPREDUCE

Mariane Raquel Silva Gonçalves  
Orientadora: Cristina Duarte Murta

Belo Horizonte  
18 de março de 2012

# SUMÁRIO

<b>1</b>	<b>PROPOSTA DE PROJETO .....</b>	<b>2</b>
1.1	Tema do projeto .....	2
1.2	Relevância .....	3
1.3	Objetivos .....	4
1.4	Resultados esperados .....	5
1.5	Metodologia .....	5
1.6	Infraestrutura necessária .....	5
1.7	Cronograma de trabalho .....	6
	<b>REFERÊNCIAS .....</b>	<b>8</b>

# 1 PROPOSTA DE PROJETO

Este texto apresenta uma proposta de projeto para a disciplina Trabalho de Conclusão de Curso, como parte dos requisitos exigido para obtenção de créditos e aprovação do projeto.

## 1.1 Tema do projeto

As tendências atuais em design de microprocessadores estão mudando fundamentalmente a maneira que o desempenho é obtido a partir de sistemas computacionais. A indústria declarou que seu futuro está em computação paralela, com o aumento crescente do número de núcleos dos processadores [Asanovic et al. 2009]. O modelo de programação *single core* (sequencial) está sendo substituído rapidamente pelo modelo *multi-core* (paralelo), e com isso surge a necessidade de escrever software para sistemas com multiprocessadores e memória compartilhada [Ernst et al. 2009].

Arquiteturas *multi-core* podem oferecer um aumento significativo de desempenho sobre as arquiteturas de núcleo único, de acordo com as tarefas paralelizadas. No entanto, muitas vezes isto exige novos paradigmas de programação para utilizar eficientemente a arquitetura envolvida [Prinslow 2011]. A ordenação é um exemplo de tarefa que pode ter seu desempenho aumentado com o uso de paralelismo. A ordenação paralela é o processo de reorganizar uma sequência de entrada e produzir uma saída ordenada de acordo com um atributo através de múltiplas unidades de processamento, que ordenam em conjunto a sequência de entrada.

Uso crescente de computação paralela em sistemas computacionais gera a necessidade de algoritmos de ordenação inovadores, desenvolvidos para dar suporte a essas aplicações.

## 1.2 Relevância

O limite físico de aumento na frequência de operação dos processadores levou a indústria de hardware a substituir o processador de núcleo único por vários processadores eficientes em um mesmo *chip*, os processadores *multi-core*. É preciso, então, criar aplicações que utilizem efetivamente o crescente número de núcleos dos processadores [Asanovic et al. 2009].

Um grande número de aplicações paralelas possui uma fase de computação intensa, na qual uma lista de elementos deve ser ordenada com base em algum de seus atributos. Um exemplo é o algoritmo de Page Rank [Page et al. 1999] da Google: as páginas de resultado de uma consulta são rankeadas de acordo com sua relevância, e então precisam ser ordenadas de maneira eficiente [Kale e Solomonik 2010].

Na criação de algoritmos de ordenação paralela, é ponto fundamental ordenar coletivamente os dados de cada processo individual, de forma a utilizar todas as unidades de processamento e minimizar os custos de redistribuição de chaves entre os processadores. Fatores como movimentação de dados, balanço de carga, latência de comunicação e distribuição inicial das chaves são considerados ingredientes chave para o bom desempenho da ordenação paralela, e variam de acordo com o algoritmo escolhido como solução [Kale e Solomonik 2010]. No exemplo do Page Rank, o número de páginas a serem ordenadas é enorme, e elas são recolhidas de diversos servidores da Google; é uma questão fundamental escolher algoritmo paralelo com o melhor desempenho dentre as soluções possíveis.

O MapReduce [Dean e Ghemawat 2008] é um modelo de programação paralela criado pela Google para processamento de grandes volumes de dados em *clusters*. Esse modelo propõe simplificar a computação paralela e ser de fácil uso, abstraindo conceitos complexos da paralelização - como tolerância a falhas, distribuição de dados e balanço de carga - e utilizando duas funções principais: *map* e *reduce*. A complexidade do algoritmo paralelo não é vista pelo desenvolvedor, que pode se ocupar em desenvolver a solução proposta. O Hadoop [White 2009] é uma das implementações do MapReduce, um *framework open source* desenvolvido por Doug Cutting em 2005 que provê o gerenciamento de computação distribuída.

*Não é fácil medir o volume total de dados armazenados digitalmente, mas uma estimativa da IDC colocou o tamanho do "universo digital" em 0,18 zettabytes*

*em 2006, e previa um crescimento dez vezes até 2011 (para 1,8 zettabytes). The New York Stock Exchange gera cerca de um terabyte de novos dados comerciais por dia. The Internet Archive armazena aproximadamente petabytes de dados, com aumento de 20 terabytes por mês [White 2009]. Estima-se que dados não estruturados são a maior porção e de a mais rápido crescimento dentro das empresas, o que torna o processamento de tal volume de dados muitas vezes inviável.*

MapReduce e sua implementação open source Hadoop representam uma alternativa economicamente atraente oferecendo uma plataforma eficiente de computação distribuída para lidar com grandes volumes de dados e mineração de petabytes de informações não estruturadas [Cherkasova 2011].

Devido ao grande número de algoritmos de ordenação paralela e variadas arquiteturas paralelas, estudos experimentais assumem uma importância crescente na avaliação e seleção de algoritmos apropriados para multiprocessadores.

## 1.3 Objetivos

- Estudar a programação paralela aplicada à algoritmos de ordenação
- Implementar uma ou mais soluções no modelo MapReduce, com o software Hadoop
- Comparar duas ou mais implementações de algoritmos paralelos de ordenação

O trabalho desenvolvido por [Pinhão 2011] apresentou um estudo sobre a computação paralela e algoritmos de ordenação no modelo MapReduce, através da implementação do algoritmo de Ordenação por Amostragem feita em ambiente Hadoop.

Este projeto busca continuar o estudo sobre ordenação paralela feito no trabalho citado, através da análise de desempenho de dois ou mais algoritmos de ordenação - sendo um deles o algoritmo de ordenação por amostragem. A análise busca compará-los com relação à quantidade de dados a serem ordenados, variabilidade dos dados de entrada e número máquinas utilizadas.

## 1.4 Resultados esperados

Com a realização do trabalho, busca-se ampliar e consolidar conhecimentos adquiridos na área de computação paralela, assim como a capacidade de análise e desenvolvimento de algoritmos paralelos no modelo MapReduce.

Ao final do trabalho, espera-se obter a implementação de algoritmos de ordenação paralela em ambiente Hadoop e uma análise comparativa de desempenho entre tais algoritmos.

A comparação entre os resultados poderá mostrar como o uso de um determinado algoritmo em uma situação específica pode influenciar no tempo de processamento e no desempenho geral do sistema.

## 1.5 Metodologia

O início do projeto será destinado ao estudo mais detalhado da computação paralela, em especial os algoritmos de ordenação paralela, dos fatores que influenciam o desempenho de tais algoritmos, o modelo MapReduce e a plataforma Hadoop. O passo seguinte é conhecer detalhadamente o algoritmo paralelo a ser implementado e definir as estratégias para sua implantação ambiente Hadoop. O algoritmo implementado deve ser cuidadosamente avaliado para verificar um funcionamento adequado com diferentes entradas e número de máquinas.

Em seguida, serão realizados experimentos para testes de desempenho dos algoritmos com relação à quantidade de máquinas, quantidade de dados e conjunto de dados. Os resultados obtidos serão analisados e permitirão comparar a desempenho dos algoritmos em cada situação.

## 1.6 Infraestrutura necessária

A infra estrutura necessária ao desenvolvimento do projeto será fornecida pelo Laboratório de Redes e Sistemas (LABORES) do Departamento de Computação (DECOM). Esse laboratório possui um *cluster* formado por cinco máquinas Dell Optiplex 380, que serão utilizadas na realização dos testes dos algoritmos. Os algoritmos serão desenvolvidos em linguagem Java, de acordo com o modelo MapReduce,

no ambiente Hadoop.

O *cluster* a ser utilizado apresenta as seguintes características:

- 5 nodos
- Processador Intel Core 2 Duo de 3.0 GHz
- Disco rígido SATA de 500 GB 7200 RPM
- Memória RAM de 4 GB
- Placa de rede Gigabit Ethernet
- Sistema operacional Linux Ubunbu 10.04 32 bits (kernel 2.6.XX)
- Sun Java JDK 1.6.0 19.0-b09
- Hadoop 0.20.2

## 1.7 Cronograma de trabalho

Na Tabela 1.1 está descrito o cronograma esperado para o desenvolvimento do projeto. Cada atividade foi descrita para se adequar da melhor maneira ao tempo disponível, mas é possível que o cronograma seja refinado posteriormente, com a inclusão de novas atividades ou redistribuição das tarefas existentes.

Atividade	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
1	•	•								
2		•	•							
3			•							
4			•	•						
5				•	•					
6						•	•			
7								•		
8									•	
9										•

**Tabela 1.1:** Cronograma proposto para o projeto

1. Pesquisa bibliográfica sobre o tema do projeto e escrita da proposta

2. Estudo mais detalhado dos algoritmos de ordenação paralela, modelo MapReduce e Hadoop.
3. Configuração do ambiente Hadoop no laboratório.
4. Implementação e testes.
5. Escrita, revisão e entrega do relatório.
6. Implementação e testes.
7. Análise comparativa entre os resultados.
8. Escrita e revisão do projeto final.
9. Entrega e apresentação.



## REFERÊNCIAS

- [Asanovic et al. 2009]ASANOVIC, K. et al. A view of the parallel computing landscape. *Commun. ACM*, ACM, New York, NY, USA, v. 52, n. 10, p. 56–67, out. 2009. ISSN 0001-0782.
- [Cherkasova 2011]CHERKASOVA, L. Performance modeling in mapreduce environments: challenges and opportunities. In: *ICPE'11*. [S.l.: s.n.], 2011. p. 5–6.
- [Dean e Ghemawat 2008]DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, ACM, New York, NY, USA, v. 51, n. 1, p. 107–113, jan. 2008. ISSN 0001-0782.
- [Ernst et al. 2009]ERNST, D. et al. Preparing students for ubiquitous parallelism. In: *SIGCSE*. [S.l.: s.n.], 2009. p. 136–137.
- [Kale e Solomonik 2010]KALE, V.; SOLOMONIK, E. Parallel sorting pattern. In: *Proceedings of the 2010 Workshop on Parallel Programming Patterns*. New York, NY, USA: ACM, 2010. (ParaPLoP '10), p. 10:1–10:12. ISBN 978-1-4503-0127-5.
- [Page et al. 1999]PAGE, L. et al. *The PageRank Citation Ranking: Bringing Order to the Web*. 1999.
- [Pinhão 2011]PINHÃO, P. de M. *Ordenação Paralela no Ambiente Hadoop*. 2011.
- [Prinslow 2011]PRINSLOW, G. *Overview of Performance Measurement and Analytical Modeling Techniques for Multi-core Processors*. 2011.
- [White 2009]WHITE, T. *Hadoop: The Definitive Guide*. first edition. O'Reilly, 2009. Disponível em: <<http://oreilly.com/catalog/9780596521981>>.