UNIVERSIDADE FEEVALE

MARCELO CORREIA FLORES

COMPARAÇÃO DE DESEMPENHO ENTRE ALGORITMOS DE ORDENAÇÃO UTILIZANDO O MODELO DE PROGRAMAÇÃO MAPREDUCE E SEQUENCIAIS

(Título Provisório)

Anteprojeto de Trabalho de Conclusão

Novo Hamburgo
2015

MARCELO CORREIA FLORES

COMPARAÇÃO DE DESEMPENHO ENTRE ALGORITMOS DE ORDENAÇÃO UTILIZANDO O MODELO DE PROGRAMAÇÃO MAPREDUCE E SEQUENCIAIS

(Título Provisório)

Anteprojeto de Trabalho de Conclusão de Curso, apresentado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação pela Universidade Feevale

Orientador: Juliano Varella de Carvalho

Novo Hamburgo
2015

RESUMO

Manipular grandes massas de informações ainda é um desafio para as ferramentas atuais de manipulação de dados. As gigantes IBM e SAP estão envolvidas neste segmento de mercado com ferramentas robustas que prometem auxiliar a manipulação deste grande volume de dados denominado de Big Data. Nascido na Google, o MapReduce é um modelo de programação simples que promete auxiliar o segmento de Big Data. O Apache Hadoop é uma ferramenta que implementa este modelo de programação. Nele, é possível desenvolver algoritmos que executam em ambientes distribuídos. Desta forma, este trabalho visa comparar os tradicionais algoritmos de ordenação, levando em consideração o seu desempenho em um ambiente convencional (sequencial), em relação ao desempenho destes em um ambiente MapReduce com Apache Hadoop.

Palavras-chave: Big Data. MapReduce. Apache Hadoop. Algoritmos de ordenação.

SUMÁRIO

MOTIVAÇÃO

A informação se faz necessária para a sociedade atual. Não só no meio corporativo se produz massas de informações, mas também no meio pessoal, por exemplo, indivíduos em suas rotinas diárias através de seus dispositivos eletrônicos. “Estima-se que até 2020 serão em torno de 30 bilhões de dispositivos móveis conectados a internet” (VELLOSO, 2014). Com o aumento constante da produção de dados, torna-se cada vez mais desafiador obter determinadas informações com eficácia.

Desta forma, depara-se com uma quantidade muito grande de dados na qual nem sempre é possível manipular a informação pelos meios convencionais. Tal quantidade de dados leva ao conceito de Big Data. “Em suma, o termo Big Data aplica-se a informações que não podem ser processadas ou analisadas por meio de processos ou ferramentas tradicionais” (ZIKOPOULOS; EATON; DEROOS, 2012). A mecânica do Big Data é o armazenamento de grandes massas de dados e recuperação em alta velocidade, basicamente. Para conceituar Big Data, alguns autores utilizam a abordagem dos 3Vs (velocidade, variedade e volume) e outros os 5Vs (velocidade, volume, valor, variedade e veracidade) (VELLOSO, 2014).

> Big Data é a simples constatação prática que o imenso volume de dados gerados a cada dia excede a capacidade das tecnologias atuais de os tratarem adequadamente. [...] Big Data = volume + variedade + velocidade. Hoje adiciono mais dois “V”s: veracidade e valor. (TAURION, 2012)

Com a crescente demanda de ferramentas de manipulação de Big Data, algumas companhias já estão investindo em produtos para suprir esta necessidade, a exemplo de grandes empresas como IBM[1] e SAP[2] que possuem um portfólio de ferramentas desta linha. A Cloudera[3] investe em uma ferramenta baseada no *framework open source* Apache Hadoop, que possibilita uma interface administrativa para acesso de escalabilidade e flexibilidade para diferentes tipos de dados. (RUSSOM; IBM, 2011)

Para ajudar na manipulação de Big Data, foi desenvolvido pela Google um modelo de programação denominado MapReduce, que inicialmente estava implementado para análise de pesquisas web. “MapReduce é um modelo de programação associado ao processamento e geração de grandes conjuntos de dados” (DEAN; GHEMAWAT, 2008). Atualmente grandes companhias como Facebook, Yahoo, Amazon, IBM, entre outras, vem utilizando este modelo

[1] http://www-01.ibm.com/software/data/bigdata
[2] http://go.sap.com/solution/big-data.html
[3] http://www.cloudera.com/content/cloudera/en/about/hadoop-and-big-data.html

como ferramenta para manipulação de dados em implementações sobre *Cloud Computing* (ANJOS, 2012).

Alguns algoritmos já conhecidos, estão sendo testados com implementações MapReduce. Um exemplo é o algoritmo Apriori, que "é um dos algoritmos mais representativos na mineração de padrões frequentes e regras de associação" (BOLINA, 2013). No artigo publicado por BOLINA et al. (2013), foi proposto o desenvolvimento do algoritmo Apriori implementando MapReduce, nomeado pelo autor de MRA (MapReduce Apriori) e compará-lo ao já existente *Distributed Multithread* Apriori (DMTA), que é uma implementação paralelizada de Apriori, porém sem MapReduce. Os resultados de ambos foram similares em sua proposta, sendo o ganho do novo método proposto observado no caso de "bases de dados que envolvam grande número de transações com pequena média de itens, normalmente encontrado em logs de acesso a páginas *Web*" (BOLINA, 2013, p.10).

Em outro trabalho, Um Algoritmo Paralelo para Cálculo de Centralidade em Grafos Grandes (NASCIMENTO, 2011, p. 56), cita alguns algoritmos que utilizam MapReduce, tais como o HADI, Pegasus e Pregel. O HADI (*Hadoop based Diameter estimator*) que é utilizado para processar o diâmetro de grafos, em dimensões na casa dos *Petabytes*. O Pegasus tem a mesma funcionalidade, além de realizar operações de mineração de grafos, como classificação de páginas, calcular proximidade de nós em um grafo e busca por componentes conectados. Há também o Pregel, desenvolvido para o processamento distribuído de grandes grafos.

Embora surgido na Google, foi desenvolvido pela *Apache Software Foundation* uma implementação de MapReduce chamada Hadoop. Trata-se de um software de código aberto, projetado para permitir o processamento de grandes volumes de dados em *clusters*. Ele também possui uma camada de aplicação capaz de detectar e tratar falhas. Baseada no GFS (*Google File System*), o sistema de gerenciamento de arquivos do Apache Hadoop, chamado HDFS (*Hadoop Distributed File System*) oferece acesso de alto desempenho para os dados aplicados, gerenciando-os sob demanda no *cluster*. O *framework* também oferece o módulo YARN, responsável por escalonar tarefas e recursos do *cluster* (APACHE, 2015).

Apesar da evolução constante da tecnologia, manipular Big Data ainda é um desafio. Devido à complexidade da distribuição de dados e do tamanho, que gira em torno dos *petabytes* (NESELLO, P.; FACHINELLI, 2014). Apesar dos diferentes métodos que ajudam a indexar os dados e da evolução de hardware, até mesmo as grandes corporações enfrentam dificuldades para manipular Big Data. "A IBM investiu, nos últimos cinco anos, mais de 14

bilhões de dólares na compra de 24 companhias para reforçar as capacidades analíticas de suas tecnologias" (OLIVEIRA, 2012).

Algoritmos de ordenação, ainda são considerados um dos problemas fundamentais da computação. Sendo assim, existem desempenhos distintos entre os algoritmos conhecidos. O objetivo da ordenação, é facilitar o acesso aos dados, organizando-os de forma sequencial, geralmente em ordem alfabética (OLIVEIRA; SOUZA, 2008).

Existem diversos algoritmos para ordenação, suas complexidades variam em $O(n^2)$ e O(n log n). Alguns exemplos de algoritmos com complexidades $O(n^2)$: Bubble sort, Quick sort e Selecion sort. Com complexidade O(n log n) existem: Heapsort sort, Merge sort e Quick sort, entre outros. Os algoritmos citados têm suas complexidades iguais no seus melhores e piores casos, exceto o Quick sort que possui no melhor caso O(n log n) e no pior $O(n^2)$ (BARBOSA; TOSCANI; RIBEIRO, 2000).

Alguns algoritmos de ordenação são propícios de serem paralelizados, é o que trata o trabalho Algoritmos Paralelos de Ordenação (OLIVEIRA, 2008), que visa melhorar a performance dos já existentes, que rodam sequencialmente. Neste mesmo contexto, o trabalho: Implementação e Avaliação de Algoritmos de Ordenação Paralela em MapReduce (MURTA; GONÇALVES; DE MORAIS PINHÃO, 2013), visa comparar o algoritmo, Quick Sort ao de Ordenação Por Amostragem, ambos usando o MapReduce em um ambiente Hadoop. Este trabalho conclui que o algoritmo de Ordenação por Amostragem apresenta melhores resultados e executa um melhor balanceamento de carga.

Visando oferecer um demonstrativo de desempenho na manipulação de grandes conjuntos de dados, este trabalho propõe o desenvolvimento de algoritmos tradicionais de ordenação, para posterior comparação com os desempenhos dos mesmos algoritmos em um ambiente multiprocessado, usando o *framework* Apache Hadoop. Desta forma, será possível medir em quais situações, os algoritmos MapReduce são ou não mais eficientes que os tradicionais.

OBJETIVOS

Objetivo geral:

Comparar o desempenho de algoritmos tradicionais de métodos de ordenação com algoritmos desenvolvidos no modelo de programação MapReduce, executados em um ambiente multiprocessado, usando o framework Apache Hadoop.

Objetivos específicos:

- Pesquisar sobre algoritmos de ordenação e suas diferentes complexidades.
- Pesquisar sobre o framework Apache Hadoop.
- Pesquisar sobre Big Data.
- Implementar algoritmos já conhecidos de ordenação.
- Aplicar o modelo de programação MapReduce para implementar algoritmos de ordenação.
- Aplicar os algoritmos de ordenação em grandes massas de dados em ambientes tradicionais e multiprocessados, usando Apache Hadoop.
- Avaliar o desempenho dos algoritmos de ordenação utilizados.

METODOLOGIA

A metodologia para elaboração deste trabalho, consistirá em uma primeira fase de pesquisa bibliográfica conceituando os assuntos propostos, uma segunda fase prática para o desenvolvimento de algoritmos que darão ênfase técnica ao argumentos citados, e por último também terá uma fase de experimentação.

Segundo Prodanov e Freitas (2013), é de natureza aplicada natureza aplicada e de método cientifico dedutivo. O objetivo de estudo será explicativa, aprofundando o conhecimento. O trabalho terá também, utilizará uma abordagem quantitativa, “requer o uso de recursos e técnicas de estatística, procurando traduzir em números os conhecimentos gerados pelo pesquisador” (PRODANOV; FREITAS, 2013).

A pesquisa bibliográfica, contextualizará os argumentos necessários para respaldar o desenvolvimento da aplicação prática. Expondo primeiramente o problema de manipulação de Big Data, buscando conceitos que provam as dificuldades de manipulação de grandes massas de dados. O estudo buscará também informações sobre o MapReduce, que será o modelo de programação utilizado no decorrer do trabalho, assim como o software Apache Hadoop, que implementa o MapReduce e será o *framework* utilizado para desenvolver a parte prática.

Como a proposta do trabalho é comparar métodos de ordenação, este assunto também será estudado e conceituado. Trabalhos similares usando processamento paralelo nos algoritmos serão pesquisados e analisados seus resultados e conclusões. Focando ainda mais em MapReduce, o trabalho trará as análises desses algoritmos utilizando este modelo de programação e também iniciativas de outros algoritmos utilizando MapReduce. Sendo assim, serão exibidos os resultados práticos destes trabalhos.

Os algoritmos de ordenação serão aplicados em bases de dados oriundas dos da iniciativa Open Data. Que consiste em reunir dados reais, normalmente governamentais, para gerar bases de dados com o propósito de pesquisas e manipulação de informações em geral (Open Data Handbook, 2012). O trabalho visa aplicar os experimentos em bases de dados na casa dos gigabytes, neste caso, o segmento dos dados coletados não tem uma importância vital no experimento final, entretanto, será utilizado um modelo consistente de dados de um mesmo assunto e diferentes regiões para aumentar a quantidade de dados.

A parte prática do trabalho elaborará uma análise de desempenho dos algoritmos desenvolvidos em um ambiente convencional e em um ambiente Apache Hadoop. Os algoritmos serão desenvolvidos em Java e serão comparados com os desempenhos do mesmo

método de ordenação com e sem a utilização do MapReduce, será usado uma ferramenta para medir a performance. Resultando assim em uma avaliação de desempenho dos algoritmos de ordenação mais utilizados na computação.

CRONOGRAMA

Trabalho de Conclusão I

Etapa	Meses			
	Mar	Abr	Mai	Jun
Realizar pesquisa bibliográfica				
Elaborar e entregar o anteprojeto				
Aprofundar e desenvolver a pesquisa bibliográfica				
Investigar e analisar estudos relacionados				
Pesquisar massa de dados grande para utilização nos experimentos				
Elaborar e entregar o TC I				

Trabalho de Conclusão II

Etapa	Meses			
	Ago	Set	Out	Nov
Iniciar o desenvolvimento do TC II				
Iniciar desenvolvimento dos algoritmos de ordenação				
Implementar algoritmos em MapReduce				
Analisar o desempenho dos experimentos				
Relatar conclusões sobre os desempenhos dos algoritmos				
Entregar e apresentar o TC II				

BIBLIOGRAFIA

ANJOS, J. C. S. DOS. **Adequação da Computação Intensiva em Dados para Ambientes Desktop Grid com uso de MapReduce**. Universidade Federal Do Rio Grande Do Sul, n. Porto Alegre,2012, p. 113, 2012.

APACHE. **Welcome to Apache Hadoop**. Disponível em <http://hadoop.apache.org/>. Acesso em: 10/03/2015.

BARBOSA, M. A.; TOSCANI, L.; RIBEIRO, L. **Ferramenta para a Automatização da Análise da Complexidade de Algoritmos**. SBIE2000–XI Simpósio Brasileiro de Informática na Educação/SBC. Anais..., n. Maceió, 2000.

BOLINA, A. C. et al. **MapReduce Apriori (MRA)** : Uma Proposta de Implementação do Algoritmo Apriori Usando o Framework MapReduce. Universidade Federal de Lavras (UFLA), n. Lavras, 2013.

BOLINA, C. **Avaliação do Framework Mapreduce para Paralelização do Algoritmo Apriori**. Universidade Federal de Lavras, Lavras, p. 70, 2013.

DEAN, J.; GHEMAWAT, S. **MapReduce**: Simplified Data Processing on Large Clusters. Communications of the ACM, SIGMOD '07. v. 51, n. 1, p. 1–13, 2008.

DÉBORAH OLIVEIRA. **Big data**: o desafio de garimpar informações. Computerworld, p. 22, 2012.

MURTA, Cristina Duarte; GONÇALVES, Mariane Raquel Silva; DE MORAIS PINHÃO, Paula. **Implementação e Avaliação de Algoritmos de Ordenação Paralela em MapReduce**. WSCAD-SSC 2013 - XIV Simpósio em Sistema Computacionais, n. Belo Horizonte, MG, Brasil, p. 18, 2013.

NASCIMENTO, J. P. B. **Um Algoritmo Paralelo para Cálculo de Centralidade em Grafos Grandes**. Centro Federal De Educação Tecnológica De Minas Gerais, n. Belo Horizonte, MG, Brasil, p. 112, 2011.

NESELLO, P.; FACHINELLI, A. C. **Big Data**: O Novo Desafio para Gestão. Revista Inteligência Competitiva, 2014.

OLIVEIRA, A. L. F. DE; SOUZA, U. DOS S. **Algoritmos Paralelos De Ordenação**. Universidade Federal do Rio de Janeiro, n. Niterói, RJ, Brasil, p. 70, 2008.

OPEN DATA HANDBOOK. Open Data Handbook Documentation. Open Knowledge Foundation, 2012. Disponível em: http://opendatahandbook.org/pdf/OpenDataHandbook.pdf. Acessado em: 01/04/2015.

PRODANOV, Cleber C.; FREITAS, Ernani C. de. **METODOLOGIA DO TRABALHO CIENTÍFICO:** Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico. 2ª ed. Novo Hamburgo: FEEVALE, 2013.

RUSSOM, P.; IBM. **Big data analytics**. TDWI Best Practices Report, Fourth Quarter, p. 38, 2011.

SANTOS, Fabrício; ORTEGA, José Miguel. **Bioinformática aplicada a genômica**. Disponível em <http://www.icb.ufmg.br/lbem/aulas/grad/tge/bioinfo/bioinfogenomica.pdf>. Acesso em 23 mar. 2014.

SETUBAL, Carlos; MEIDANIS, João. **Introduction to Computational Molecular Biology**. Boston: PWS, 1997.

TAURION, Cezar. **Você realmente sabe o que é Big Data?.** Disponível em <https://www.ibm.com/developerworks/mydeveloperworks/blogs/ctaurion/entry/voce_realmente_sabe_o_que_e_big_data?lang=en>. Acesso em: 10/03/2015

VELLOSO, Fernando. Informática: **Conceitos Básicos**. [S.I], Campus Elsevier, 2014.

ZIKOPOULOS, P.; EATON, C.; DEROOS, D. Understanding big data. [s.l], McGraw Hill. p. 166, 2012.