UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE CIÊNCIAS EXATAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARIA TERESA KRAVETZ ANDRIOLI

TITULO

CURITIBA

2021

MARIA TERESA KRAVETZ ANDRIOLI

TITULO

Trabalho apresentado como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação no curso de Ciência da Computação, Setor de Ciências Exatas da Universidade Federal do Paraná.

Orientador: Dr. Luiz Carlos P. Albini, Prof.

CURITIBA

2021

TERMO DE APROVAÇÃO

MARIA TERESA KRAVETZ ANDRIOLI

TITULO

Trabalho apresentado como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação no curso de Ciência da Computação, Setor de Ciências Exatas da Universidade Federal do Paraná, pela seguinte banca examinadora:

Dr. Luiz Carlos P. Albini, Prof.
Orientador

Professora
UFPR

Professora
ENSEADE

Professora
TIT

Curitiba, 09 de Dezembro de 2018.

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

AGRADECIMENTOS

Os agradecimentos principais são direcionados à Gerald Weber, Miguel Frasson, Leslie H. Watter, Bruno Parente Lima, Flávio de Vasconcellos Corrêa, Otavio Real Salvador, Renato Machnievscz[1] e todos aqueles que contribuíram para que a produção de trabalhos acadêmicos conforme as normas ABNT com LaTeX fosse possível.

Agradecimentos especiais são direcionados ao Centro de Pesquisa em Arquitetura da Informação[2] da Universidade de Brasília (CPAI), ao grupo de usuários *latex-br*[3] e aos novos voluntários do grupo *abnTeX2*[4] que contribuíram e que ainda contribuirão para a evolução do abnTeX2.

Os agradecimentos principais são direcionados à Gerald Weber, Miguel Frasson, Leslie H. Watter, Bruno Parente Lima, Flávio de Vasconcellos Corrêa, Otavio Real Salvador, Renato Machnievscz[5] e todos aqueles que contribuíram para que a produção de trabalhos acadêmicos conforme as normas ABNT com LaTeX fosse possível.

[1] Os nomes dos integrantes do primeiro projeto abnTeX foram extraídos de http://codigolivre.org.br/projects/abntex/

[2] http://www.cpai.unb.br/

[3] http://groups.google.com/group/latex-br

[4] http://groups.google.com/group/abntex2 e http://abntex2.googlecode.com/

[5] Os nomes dos integrantes do primeiro projeto abnTeX foram extraídos de http://codigolivre.org.br/projects/abntex/

RESUMO

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.Ter no máximo 500 palavras!!! As palavras chave são separadas por ponto e vírgula.

Palavras-chaves: latex; abntex; editoração de texto.

ABSTRACT

This is the english abstract.

Key-words: latex. abntex. text editoration.

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTO

O uso, armazenamento e controle de dados é um tema muito discutido na área de computação desde seus primórdios até os dias de hoje. Por causa disso, muitos métodos e algoritmos e termos surgiram ao longo do tempo com o objetivo de gerenciar de forma eficiente esses dados. O surgimento dessas novas ferramentas computacionais e métodos de armazenamento foi importantíssimo para a evolução da área.

Atualmente, os métodos mais comuns são os bancos de dados relacionais e *datas warehouses* usando computação em nuvem (KUO; KUSIAK, 2019). Além disso, pesquisas nos campos de mineração de dados e aprendizagem de máquina cresceram bastante recentemente de modo a prover técnicas que permitissem analisar dados complexos e variados entre si (BELCASTRO et al., 2022). Um grande desafio é o fato de algoritmos sequenciais não serem otimizados o suficiente para lidar com dados em grande quantidade. Por causa disso, computadores de alta performance, com múltiplos *cores*, sistemas na nuvem e algoritmos paralelos e distribuídos são usados para lidar com esses empecilhos de *Big Data* (BELCASTRO et al., 2022).

Big Data refere-se a grandes conglomerados de dados complexos sobre os quais não é possível aplicar ferramentas tradicionais de processamento, armazenamento ou análise (KHALEEL; AL-RAWESHIDY, 2018). Estima-se que em 2025. os dados atuais criados, capturados ou replicados atinjam 175 Zettabytes, ou seja 175,000,000,000 Gigabytes (RYDNING, 2018).

A fim de lidar com essa enorme quantidade de dados, foi desenvolvido pelo Google o MapReduce, que é um modelo e implementação feito para processar e gerar grandes conglomerados de dados. Esse modelo é inspirado nos conceitos de mapear e reduzir, ou seja, aplicar uma operação que conecta cada item da base de dados a um determinado par de chaves e valores e então aplicar uma operação de reduzir, que junta os valores que compartilham chaves (DEAN; GHEMAWAT, 2004). Com essas operações é possível paralelizar dados em grandes quantidades e utilizar mecanismos de reutilização para facilitar a busca e manipulação destes.

Um dos *frameworks* mais populares que utiliza o MapReduce é o Hadoop, desenvolvido pela Apache que é capaz de armazenar e processar de giga a petabytes de dados eficientemente. Essa ferramenta é capaz de fazer isso optando por usar múltiplos computadores (*clusters*) em paralelo (WHITE, 2015).

[FALAR AQUI SOBRE O OBJETIVO DESTE TRABALHO, A MOTIVAÇÃO, METODOLOGIA(?) E COMO ELE ESTÁ ESTRUTURADO]

2 REFERENCIAL TEÓRICO

Esse capítulo tem como objetivo apresentar detalhadamente os conceitos técnicos que serão utilizados ao longo do trabalho. A Seção 2.1 apresenta o MapReduce, o modelo de manipulação de dados feito pelo Google e a Seção 2.2 trata do Hadoop, o *framework* desenvolvido pela Apache.

2.1 MAPREDUCE

MapReduce é um modelo de programação associado a uma implementação que tem como objetivo processar, manipular e gerar grandes *datasets* de modo eficiente, escalável e com aplicações no mundo real. As computações acontecem de acordo com funções de mapeamento e redução e o sistema do MapReduce paraleliza essa computações entre grandes *clusters*, lidando com possíveis falhas, escalonamentos e uso eficiente de rede e discos (DEAN; GHEMAWAT, 2008).

As operações de mapeamento e redução são baseadas em conceitos presentes em linguagens funcionais e fazem com que seja possível fazer diversas reutilizações, assim lidando com tolerância de falhas (DEAN; GHEMAWAT, 2008).

2.1.1 Modelo de programação

A computação recebe um conjunto de pares (VALOR, CHAVE) e produz um conjunto de pares de (VALOR, CHAVE). O usuário cria as funções *Map* e *Reduce* de acordo com seu caso de uso. *Map* recebe um único par (VALOR, CHAVE) e produz um conjunto intermediário de pares. Em seguida, a biblioteca *MapReduce* agrupa os valores com a mesma chave e esses valores servirão de entrada para a função *Reduce*. A função *Reduce* então junta os valores com a mesma chave de modo a criar um conjunto menor, sendo possível dessa forma lidar com listas muito grandes para a memória disponível (DEAN; GHEMAWAT, 2004).

Como exemplo, considere o problem de contar quantas vezes determinada palavra aparece em um documento. Nesse problema, as funções *Map* e *Reduce* seriam similares aos seguintes pseudocódigos (DEAN; GHEMAWAT, 2008):

```
map(String chave, String valor):
// chave: nome do documento
// valor: conteudo do documento

  para cada palavra W em valor:
    criaIntermediario(W, '1');
```

Código 1: Exemplo de função Map em pseudocódigo adaptado de (DEAN; GHEMAWAT, 2008)

```
reduce(String chave, Iterador valores):
// chave: uma palavra
// valores: lista de contagens

  int resultado = 0;
  para cada V em valores:
    resultado = resultado + converteEmInt(V);
  cria(resultado);
```

Código 2: Exemplo de função Reduce em pseudocódigo adaptado de (DEAN; GHEMAWAT, 2008)

A função *Map* gera um objeto intermediário de cada palavra associada a uma lista do seu número de occorrências no texto e a função *Reduce* soma os valores até ter o total de ocorrências por palavra. Além disso, o usuário cria um configuração de *MapReduce* com os parâmetros de entrada e saída e eventuais parâmetros de *tuning*.

REFERÊNCIAS

BELCASTRO, L.; CANTINI, R.; MAROZZO, F.; ORSINO, A.; TALIA, D.; TRUNFIO, P. Programming big data analysis: principles and solutions. **Journal of Big Data**, SpringerOpen, v. 9, n. 1, p. 1–50, 2022. Citado 2 vez na página 8.

DEAN, J.; GHEMAWAT, S. MapReduce: Simplified data processing on large clusters, 2004. Citado 2 vezes nas páginas 8, 9.

_____. MapReduce: simplified data processing on large clusters. **Communications of the ACM**, ACM New York, NY, USA, v. 51, n. 1, p. 107–113, 2008. Citado 7 vezes nas páginas 9, 10.

KHALEEL, A.; AL-RAWESHIDY, H. Optimization of computing and networking resources of a Hadoop cluster based on software defined network. **IEEE Access**, IEEE, v. 6, p. 61351–61365, 2018. Citado 1 vez na página 8.

KUO, Y.-H.; KUSIAK, A. From data to big data in production research: the past and future trends. **International Journal of Production Research**, Taylor & Francis, v. 57, n. 15-16, p. 4828–4853, 2019. DOI: 10.1080/00207543.2018.1443230. eprint: https://doi.org/10.1080/00207543.2018.1443230. Disponível em: https://doi.org/10.1080/00207543.2018.1443230. Citado 1 vez na página 8.

RYDNING, D. R.-J. G.-J. The digitization of the world from edge to core. **Framingham: International Data Corporation**, p. 16, 2018. Citado 1 vez na página 8.

WHITE, T. **Hadoop: The definitive guide**. [S.l.]: "O'Reilly Media, Inc.", 2015. Citado 1 vez na página 8.