UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Maurício Matter Donato

ARQUITETURA DINÂMICA PARA O GERENCIAMENTO DE MEMÓRIA EM APLICAÇÕES COM REUSO DE DADOS NO APACHE SPARK

Santa Maria, RS
2019

Maurício Matter Donato

ARQUITETURA DINÂMICA PARA O GERENCIAMENTO DE MEMÓRIA EM APLICAÇÕES COM REUSO DE DADOS NO APACHE SPARK

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

ORIENTADORA: Prof.ª Patrícia Pitthan de Araujo Barcelos

Santa Maria, RS
2019

Maurício Matter Donato

ARQUITETURA DINÂMICA PARA O GERENCIAMENTO DE MEMÓRIA EM APLICAÇÕES COM REUSO DE DADOS NO APACHE SPARK

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

Aprovado em 31 de dezembro de 2019:

_____

Patrícia Pitthan de Araujo Barcelos, Dra. (UFSM)
(Presidenta/Orientadora)

_____

Indefinido, Dra. (UFSM)

_____

Indefinido, Dr. (UFSM)

Santa Maria, RS
2019

DEDICATÓRIA

AGRADECIMENTOS

()

RESUMO

ARQUITETURA DINÂMICA PARA O GERENCIAMENTO DE MEMÓRIA EM APLICAÇÕES COM REUSO DE DADOS NO APACHE SPARK

AUTOR: Maurício Matter Donato
ORIENTADORA: Patrícia Pitthan de Araujo Barcelos

Palavras-chave:

ABSTRACT

DYNAMIC ARCHITECTURE FOR MEMORY MANAGEMENT ON APPLICATIONS WITH DATA REUSE ON APACHE SPARK

AUTHOR: Maurício Matter Donato
ADVISOR: Patrícia Pitthan de Araujo Barcelos

Keywords:

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RDD	Resilient Distributed Datasets

SUMÁRIO

1 INTRODUÇÃO

Os avanças nas tecnologias de informação permitiram o armazenamento de grandes e variados conjuntos de dados. Com o advento da *internet*, interações *online* são cada vez mais presentes no cotidiano, possibilitando a comunicação entre pessoas e empresas de maneira fácil e descomplicada. Como consequência dessa interatividade, a quantidade de dados tem crescido a uma taxa significativa durante os últimos anos, de acordo com Goldschimidt e Bezerra (2015).

Estes dados, os quais podem ir desde manifestações em redes sociais até movimentações financeiras, são gerados e disponibilizados em diferentes tamanhos e formatos. De acordo com McAfee (2012), através desses dados gerentes podem medir e conhecer radicalmente seus negócios e consequentemente, traduzir esse conhecimento em decisões melhores para seus negócios.

Porém, a realização das etapas necessárias para extrair conhecimento a partir desses dados implica em altos custos econômicos e computacionais, uma vez que o armazenamento e processamento dos mesmos não são tarefas triviais. De acordo com Oussous et al. (2018), essas dificuldades afetam a captura de dados, armazenamento, pesquisa, compartilhamento, análise, gerência e visualização dessas informações. Além disso, a segurança e provacidade são problemas em aplicações guiadas a dados.

O processamento dessas informações requer ferramentas capazes de recorrer ao processamento paralelo e distribuído entre um conjunto de máquinas (*cluster*), além de suportar altas variações no volume de dados utilizado. *Frameworks* baseados no paradigma *MapReduce*, como o Apache Hadoop, têm sido amplamente utilizados para o processamento de grandes volumes de dados. De forma geral, esses *frameworks* oferecem operações de processamento de alto nível e abstrações para acesso aos recursos do *cluster* com o objetivo de facilitar o desenvolvimento de aplicações pelos usuários.

Entretanto, segundo Zaharia et al. (2012), esses *frameworks* falham em oferecer abstrações para acesso à memória distribuída tornando-os ineficientes no processamento de algoritmos de reuso, como aqueles utilizados em Mineração de Dados e Aprendizado de Máquina. Nesse sentido, o Apache Spark[1] surge como um *framework* capaz de processar de grandes quantidades de dados de maneira paralela e distribuída estendendo o modelo *MapReduce*, já consolidado pelo Apache Hadoop, de modo a facilitar o desenvolvimento de aplicações com estas características.

O Spark foi pensado e projetado para implementar um mecanismo de execução multiestágio em memória principal, onde juntamente com sua principal abstração, o *Resilient Distributed Datasets* Zaharia et al. (2012) (RDD), permite que diversas computações sejam realizadas em memória, dispensando a escrita de dados intermediários em disco.

[1] Disponível em: https://spark.apache.org/

Dessa forma, o Spark consegue alcançar um desempenho superior quando comparado ao mecanismo baseado em disco utilizado pelo Hadoop.

Um RDD consiste em uma coleção imutável de objetos, os quais podem ser operados e processados de forma paralela e distribuída entre os nós do *cluster*. Uma vez realizado o processamento de um determinado RDD, este pode ser mantido em *cache* para que seja possível reutilizá-lo em futuras computações sem necessidade de realizar a sua recomputação. Por padrão, o nível de armazenamento utilizado pelo Spark para o *caching* de dados é apenas a memória principal, podendo ser alterado para armazenamento estável ou uma combinação de ambos.

Conforme novos RDDs são armazenados e processados, a memória disponível tende a ficar esgotada e, portanto, políticas de gerenciamento de memória devem ser utilizadas. Assim, em situações de sobrecarga no uso do espaço disponível, o Spark remove partições mantidas em *cache* de acordo com o algoritmo LRU (*Least Recently Used*) Luu Hien. (2018). Desta forma, é possível gerar situações onde apenas uma fração do RDD permanece armazenado em *cache*.

REFERÊNCIAS BIBLIOGRÁFICAS

GOLDSCHMIDT, R.; BEZERRA, E.; PASSOS, E. Data mining: conceitos, técnicas, algoritmos, orientações e aplicações. **Rio de Janeiro-RJ: Elsevier**, p. 56–60, 2015.

LUU, H. **Beginning Apache Spark 2: with resilient distributed datasets, Spark SQL, structured streaming and Spark machine learning library**. [S.l.]: Apress, 2018.

MCAFEE, A. et al. Big data: the management revolution. **Harvard business review**, v. 90, n. 10, p. 60–68, 2012.

OUSSOUS, A. et al. Big data technologies: A survey. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, v. 30, n. 4, p. 431–448, 2018.

ZAHARIA, M. et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: USENIX ASSOCIATION. **Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation**. [S.l.], 2012.