October 7-10 • Ceará • Brazil

34th Brazilian Symposium on DATABASES

SBBD|2019

PROCEEDINGS COMPANION

October 7-10 • Ceará • Brazil

34th Brazilian Symposium on DATABASES

PROCEEDINGS COMPANION

Promotion
Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization
Departamento de Computação, Universidade Federal de Ceará– UFC

Steering Committee Chair
Bernadete Farias Lóscio (UFPE, Brazil)

Local Chair
José Maria da Silva Monteiro Filho (UFC, Brazil)

Program Committee Chairs
Full Paper: Carina F. Dorneles (UFSC, Brazil)
Short Paper: Fábio Porto (LNCC, Brazil)
Demos and Applications Chair: Robson L. F. Cordeiro (ICMC-USP, Brazil)
Thesis and Dissertation Workshop Chair: Jonice Oliveira (UFRJ, Brazil)
Tutorials Chair: Altigran Soares da Silva (UFAM, Brazil)
Short course Chair: Maria Cláudia Cavalcanti (IME, Brazil)
Workshop Chair: José Antônio Macedo (UFC, Brazil)
Thesis and Dissertation Contest Chair: Caetano Traina Jr. (USP, Brazil)
Graduation Student Workshop Chair: Ticiana Linhares (UFC, Brazil)

B839
Brazilian Symposium on Databases (SBBD 2019) (25.: 2019 october 07-10, 2019 –Fortaleza, CE)
Proceedings of 34nd Brazilan Symposium on Databases - SBBD 2019 [recurso eletrônico] / Organização: José Maria da Silva Monteiro Filho, Robson Leonardo Ferreira Cordeiro, Jonice de Oliveira Sampaio, Ticiana Linhares Coelho da Silva, Caetano Traina Junior - Fortaleza: SBC, 2019.

447p. v.2

Modo de acesso: http://sbbd.org.br/2019/

ISSN: 2016-5170

1. Computação - Congressos. 2. Bases de Dados– Congressos. I. José Maria da Silva Monteiro Filho. II. Sociedade Brasileira de Computação. III. Título.

CDD: 005

Elaborada pela Biblioteca da Universidade Federal do Ceará Campus Crateús (BCCR).

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Workshop on Thesis and Dissertations in Databases

PROCEEDINGS

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

Program Chair

Jonice Oliveira (UFRJ, Brazil)

Table of Contents (Thesis and Dissertation Workshop)

Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um *data warehouse* de imagens em ambientes paralelos e distribuídos

Guilherme Muzzi da Rocha[1],
Profa. Dra. Cristina Dutra de Aguiar Ciferri[1]

[1]Pós-Graduação em Ciências de Computação e Matemática Computacional
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos, SP, Brasil

guilherme.muzzi.rocha@usp.br, cdac@icmc.usp.br

Nível: Mestrado
Ano de ingresso no programa: 2018
Exame de qualificação: Abril de 2019
Época esperada de conclusão e defesa: Abril de 2020
Etapas Concluídas: Créditos em disciplinas; Definição do Problema; Proposta de solução; Testes de desempenho preliminares
Publicações: [Rocha and Ciferri 2018, Traina et al. 2019]

Abstract. *Analytical queries in conventional data warehousing environments have a high computational cost, as they run over voluminous data warehouses and require the processing of expensive star join operations. This cost is even greater in image data warehousing environments. First, image data warehouses are more voluminous. Second, analytical queries are extended with a similarity search predicate, which requires the processing of costly operations that calculate the distance between images. In the literature, the use of parallel and distributed computing environments has become an attractive alternative to individually minimize the cost of the star join processing over conventional data and the cost of calculating the distances between images. In our master's research, we fill a gap in the literature by jointly investigating the processing of analytical queries extended with a similarity search predicate over image data warehouses using the framework Spark. In this paper, we describe the methods that we are developing to this end. We consider the context of medical images, due to the importance of the analytical decision-making over these images and their impact on society*[1].

Palavras-Chave. *Data warehouse de imagens, consultas analíticas estendidas com predicado de similaridade, Spark, imagens médicas.*

[1]Trabalho sendo desenvolvido com recursos financeiros da FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), processos 2018/10607-3 e 2018/22277-8, e do CNPq.

1. Introdução

Um ambiente de *data warehousing* engloba técnicas e ferramentas voltadas à extração, tradução, filtragem e integração de dados de provedores autônomos, heterogêneos e distribuídos (processo ETL - *extract, transform, load*). Esses dados são armazenados no *data warehouse* (DW), que é um banco de dados caracterizado por ser integrado, não-volátil, orientado a assunto e histórico. Sobre o DW incidem as consultas analíticas (processo OLAP - *on-line analytical processing*) [Kimball and Ross 2002]. Um DW convencional contém apenas dados convencionais, como dados do tipo numérico, alfanumérico e data.

Um ambiente de *data warehousing* de imagens estende o *data warehousing* convencional para também manipular imagens representadas por seus vetores de características e atributos para pesquisa por similaridade [Teixeira et al. 2015]. O processo ETL é estendido para também extrair as características intrínsecas das imagens, o DW de imagens estende o DW convencional para armazenar essas características e o processo OLAP também oferece suporte para consultas analíticas estendidas com predicado de similaridade de imagens. Como resultado, uma nova gama de consultas analíticas pode ser realizada. Por exemplo, em uma aplicação médica, pode-se determinar *"Quantas imagens de câncer pulmonar são similares a uma determinada imagem e pertencem a pacientes com idade maior do que 40 anos no estado de São Paulo nos últimos 3 anos"*.

Um DW convencional é muito volumoso. Ele é frequentemente povoado, além de conter dados históricos [Kimball and Ross 2002]. Consultas OLAP são caras porque a junção-estrela tem custo computacional alto por lidar com tabelas de fatos muito volumosas em implementações relacionais. Um DW de imagens é ainda mais volumoso porque contém dados convencionais e características intrínsecas das imagens. A frequência de povoamento também é alta. Por exemplo, em uma aplicação de uma rede de hospitais de uma determinada região, os provedores podem ser numerosos e as atividades da área médica muito frequentes [Sebaa et al. 2018]. Consultas OLAP estendidas com predicado de similaridade de imagens são muito mais caras porque envolvem onerosos cálculos de distância entre imagens [Traina et al. 2007] em adição à operação de junção-estrela.

Um *data warehousing* de imagens pode se beneficiar de ambientes computacionais paralelos e distribuídos. O uso desses ambientes tem se tornado uma alternativa atrativa para minimizar individualmente o custo do processamento da junção-estrela sobre dados convencionais e o custo do cálculo das operações de distância entre imagens (seção 3). Esses ambientes também proveem disponibilidade dos dados, tolerância a falhas e estratégias para replicação. Na área médica, eles facilitam o compartilhamento de dados e possibilitam que o suporte à tomada de decisão seja mais robusto [Sebaa et al. 2018].

O objetivo deste projeto de mestrado é propor métodos voltados ao processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens utilizando o *framework* de processamento paralelo e distribuído Spark [Zaharia et al. 2010]. Embora o trabalho a ser desenvolvido seja genérico, ele tem como motivação a manipulação de imagens médicas, devido à importância da tomada de decisão analítica considerando essas imagens e seu impacto para a sociedade. Portanto, esse contexto é considerado ao longo de todo o artigo.

Este artigo está estruturado da seguinte forma. A fundamentação teórica é descrita na seção 2 e a revisão sistemática é resumida na seção 3. O estado atual do desenvolvi-

mento do projeto e os resultados preliminares são detalhados na seção 4. As considerações finais e as próximas atividades a serem desenvolvidas são listadas na seção 5.

2. Fundamentação Teórica

2.1. *Data warehouse* de imagens

Na Figura 1 é ilustrado um DW de imagens da área médica. Em implementações relacionais, os dados são armazenados segundo o esquema-estrela, com uma tabela de fatos (*Exame*) que se relaciona com várias tabelas de dimensão convencionais (*Paciente, DescriçãoExame, Hospital, DataExame*). Também existem tabelas de dimensão projetadas para armazenar as características intrínsecas das imagens [Teixeira et al. 2015]. A tabela *Vetor Características* armazena os vetores de características de todas as imagens do DW. Esses vetores representam as características das imagens de acordo com as camadas perceptuais definidas no predicado de similaridade. Para cada camada perceptual há um vetor de características obtido por um extrator de características (ex: cor, textura e forma).

A partir dos vetores de características e de uma função de distância, que mede o grau de dissimilaridade entre as imagens, define-se o espaço métrico [Traina et al. 2007]. Esse espaço possibilita a execução de operações de similaridade, como a operação *range query*, a qual, a partir de um centro de consulta, retorna todas as imagens que estão a uma distância menor ou igual ao raio dado. Para diminuir a quantidade de cálculos de distância nas operações de similaridade, a técnica Omni [Traina et al. 2007] define elementos representativos que criam uma região de interesse no espaço métrico, sendo que somente as distâncias às imagens dentro dessa região sejam calculadas. Cada tabela de dimensão $CamadaPerceptual_i$ contém as distâncias entre cada imagem do DW e cada elemento representativo relativo à camada perceptual i.

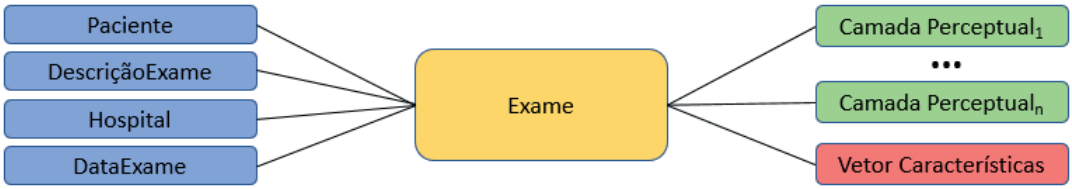

Figura 1. Exemplo de um *data warehouse* de imagens da área médica.

A organização do DW segundo o esquema-estrela requer a realização de operações de junção-estrela no processamento de consultas OLAP. Essas operações realizam junções entre a tabela de fatos e cada uma das tabelas de dimensão envolvidas na consulta, bem como resolvem condições de seleção e agrupamento.

2.2. Ambientes computacionais paralelos e distribuídos

Ambientes computacionais paralelos e distribuídos são baseados no sistema de arquivos distribuído HDFS [Shvachko et al. 2010], que divide o arquivo de dados em blocos, distribuindo e replicando esses blocos em nós do *cluster*. Para abstrair a complexidade inerente ao paralelismo, surgem os *frameworks* MapReduce [Dean and Ghemawat 2008] e Spark [Zaharia et al. 2010]. MapReduce usa um modelo de programação genérico composto de funções *map* e *reduce*, e Spark baseia-se em computação em memória e na abstração de RDD (*resilient distributed dataset*), sendo disponíveis como Apache Hadoop MapReduce (https://hadoop.apache.org/) e Apache Spark (https://spark.apache.org/).

Nesses ambientes, duas técnicas têm sido usadas para melhorar o processamento da junção-estrela [Brito et al. 2016]. A técnica de *broadcast join* assume que as tabelas de dimensão são suficientemente pequenas para serem enviadas para todos os nós durante o processamento da consulta, e realiza todas as junções, em paralelo, localmente em cada nó. Já a técnica de *bloom filter cascade join* usa uma estrutura de dados probabilística para diminuir o tamanho da tabela de fatos antes de realizar a junção-estrela em cascata.

3. Revisão Sistemática

A revisão sistemática analisa o estado-da-arte, identifica lacunas a serem exploradas e detecta tecnologias dentro do contexto da pesquisa [Kitchenham and Charters 2007]. A revisão realizada visou identificar artigos que atendessem às questões: (i) Como realizar junção-estrela em Hadoop? (ii) Como realizar operações de similaridade de imagens em Hadoop? (iii) Como se caracteriza um DW da área médica em Hadoop? (iv) Como realizar operações de junção-estrela e de similaridade sobre um DW em Hadoop? Hadoop foi definido porque mais artigos relacionados à proposta deste projeto foram retornados usando-se essa palavra-chave ao invés de "processamento paralelo e distribuído". Foram consideradas as fontes de busca IEEEXplore Digital Library (https://ieeexplore.ieee.org), Springer (https://www.springer.com/ComputerScience), ACM Digital Library (https://dl.acm.org) e Elsevier (https://www.elsevier.com/physical-sciences/computer-science). Foram analisados os últimos 5 anos, ou seja, de 2014 até o momento.

Na Figura 2 é ilustrada a revisão sistemática realizada, com as palavras-chave, as fontes de busca e as quantidades de artigos retornados. Os idiomas selecionados foram Português e Inglês, sendo as palavras-chave em Inglês omitidas por falta de espaço. Dos 103 artigos retornados, 87 foram excluídos na seleção inicial por meio da leitura do título e resumo, e mais 5 foram excluídos na seleção final por meio da leitura do artigo completo. Os 11 artigos remanescentes foram agrupados de acordo com as questões definidas.

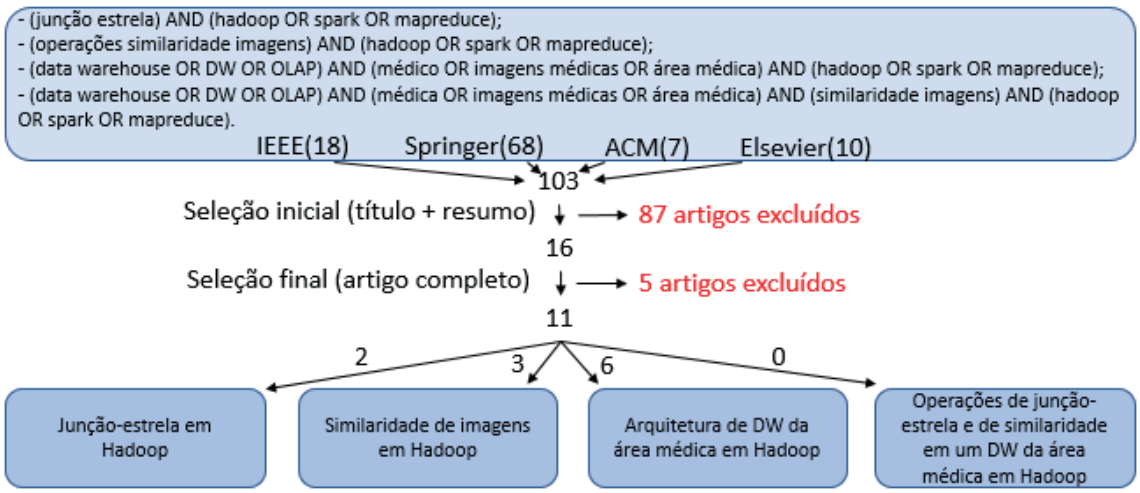

Figura 2. Processo de revisão sistemática que analisou o estado-da-arte.

Os trabalhos de [Guoliang and Guilan 2015, Brito et al. 2016] processam a *junção-estrela* usando MapReduce e Spark. Desses trabalhos, os métodos SBFCJ (*Spark Bloom Filter Cascade Join*) e SBJ (*Spark Broadcast Join*) [Brito et al. 2016] representam

o estado-da-arte, e empregam, em Spark, as técnicas de *bloom filter cascade join* e *broadcast join*, respectivamente. Entretanto, trabalhos desse grupo não manipulam imagens.

Os trabalhos de [Li et al. 2017, Nguyen et al. 2016, Nguyen and Huh 2017] otimizam *operações de similaridade de imagens* em MapReduce e Spark definindo funções *hash* e o uso do método de acesso métrico VP-tree. Limitações incluem a complexidade de se definir funções *hash* apropriadas e o fato de que a Omni tem melhor desempenho que a VP-tree [Traina et al. 2007]. Esses trabalhos não otimizam operações de junção-estrela.

Na proposta de *arquitetura de DW da área médica*, [Istephan and Siadat 2015, Istephan and Siadat 2016, Kuo et al. 2015, Raja and Sivasankar 2014, Sebaa et al. 2017, Sebaa et al. 2018] evidenciaram as vantagens de se usar processamento paralelo e distribuído em aplicações médicas e destacaram as tecnologias Hive e HBase. Porém, esses trabalhos não processam consultas analíticas estendidas com predicado de similaridade.

No melhor do conhecimento dos autores deste artigo, não existem trabalhos que realizem *operações de junção-estrela e de similaridade sobre um DW da área médica* em Hadoop. O projeto de mestrado visa preencher essa lacuna, considerando Spark.

4. Proposta e Estágio Atual de Desenvolvimento

4.1. Descrição da Proposta

Dado o objetivo do projeto de mestrado, estão sendo desenvolvidos dois métodos para o processamento de consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens da área médica em Spark. A descrição a seguir é baseada na Figura 1.

As consultas alvo possuem dois tipos de predicado. O *predicado convencional* é composto por condições de seleção, sendo cada condição definida sobre um atributo de uma tabela de dimensão convencional. O *predicado de similaridade* é composto por uma operação de similaridade e pelas camadas perceptuais consideradas. Na consulta *"Liste a quantidade de imagens similares a uma dada imagem, para pacientes do sexo feminino com câncer de mama e as camadas perceptuais de cor e textura"*, tem-se: (i) condições de seleção: pacientes do sexo feminino e câncer de mama; (ii) operação *range query* para resolver a similaridade; e (iii) camadas perceptuais: cor e textura.

O primeiro método integra as técnicas *broadcast join* e Omni da seguinte forma. Cada k tabela de dimensão convencional envolvida na consulta é filtrada de acordo com as condições de seleção correspondentes, sendo o resultado armazenado na estrutura $HashMapConvencional_k$. Cada i tabela $CamadaPerceptual_i$ envolvida na consulta é filtrada pelo predicado de similaridade usando a Omni, gerando imagens candidatas armazenadas na estrutura $HashMapFiltragem_i$. Por *broadcast*, as estruturas $HashMapFiltragem_i$ são transmitidas e processadas sobre a tabela *Vetor Características* para calcular a distância entre cada imagem candidata e os elementos representativos da camada perceptual correspondente, sendo o resultado armazenado na estrutura $HashMapRefinamento$. Por *broadcast*, as estruturas $HashMapConvencional_k$ e $HashMapRefinamento$ são transmitidas e processadas sobre a tabela de fatos para a realização da junção-estrela. Esse primeiro método é adequado para ambientes nos quais os nós possuem memória primária suficiente para armazenar as tabelas de dimensão, ou seja, eles possuem memória suficiente para processar todas as estruturas *hash map* usadas. Detalhes sobre esse método são descritos em [Rocha and Ciferri 2019].

O segundo método integra as técnicas *bloom filter cascade join* e Omni de forma similar ao primeiro método. Porém, ele usa a estrutura *bloom filter* ao invés da estrutura *hash map* quanto à aplicação dos filtros convencionais e de similaridade, gerando as estruturas $BloomFilterConvencional_k$ e $BloomFilterFiltragem_i$. As estruturas $BloomFilterFiltragem_i$ são processadas sobre a tabela *Vetor Características* gerando a estrutura $BloomFilterRefinamento$. As estruturas $BloomFilterConvencional_k$ e $BloomFilterRefinamento$ atuam sobre a tabela de fatos como filtro, diminuindo o tamanho dessa tabela para o processamento das operações de junção-estrela em cascata. Esse segundo método é adequado para ambientes nos quais as memórias primárias dos nós sejam insuficientes para processar as estruturas completamente.

4.2. Validação da Proposta

Testes de desempenho compararam o primeiro método proposto com o trabalho mais próximo na literatura: SBJ [Brito et al. 2016] (seção 3). O esquema-estrela da Figura 1 foi povoado com dados gerados pela ferramenta ImgDW [Rocha and Ciferri 2018], que já representa uma contribuição do trabalho de mestrado. Foram definidas consultas analíticas com diferentes condições de seleção e aplicada a operação *range query* (raio de abrangência de 20% do diâmetro do conjunto de dados) para calcular a similaridade, considerando as camadas perceptuais *Haralick Variância* (baixa dimensionalidade: 4 dimensões) e *Histograma de Cores* (alta dimensionalidade: 256 dimensões). Foi usado um *cluster* com 5 nós, cada qual com no mínimo 3GB de RAM. Os resultados mostraram que, para consultas contendo pelo menos a camada perceptual de alta dimensionalidade, o método proposto proveu ganhos de desempenho que variaram de 60,01% a 65,49%. Para consultas contendo apenas a camada perceptual de baixa dimensionalidade, o método proposto empatou ou proveu ganhos de desempenho de até 10,50%. A variação do desempenho em termos da dimensionalidade das camadas perceptuais é uma característica herdada da técnica Omni [Traina et al. 2007]. Detalhes sobre os resultados obtidos são descritos em [Rocha and Ciferri 2019].

5. Conclusão

No projeto de mestrado estão sendo desenvolvidos dois métodos para o processamento de consultas analíticas com predicado de similaridade sobre um DW de imagens da área médica em Spark. O primeiro método integra as técnicas de *broadcast join* e Omni e o segundo método integra as técnicas de *bloom filter cascade join* e Omni. As próximas atividades referem-se à continuidade do processo de validação do primeiro método, com a definição de um ambiente de teste mais robusto considerando novas consultas e testes de escalabilidade, bem como a implementação e validação do segundo método proposto.

Referências

Brito, J. J., Mosqueiro, T., Ciferri, R. R., and Ciferri, C. D. A. (2016). Faster cloud star joins with reduced disk spill and network communication. In *ICCS 2016*, volume 80 of *Procedia Computer Science*, pages 74–85.

Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Guoliang, Z. and Guilan, W. (2015). GBFSJ: Bloom filter star join algorithms on GPUs. In *FSKD 2015*, pages 2427–2431.

Istephan, S. and Siadat, M.-R. (2015). Extensible query framework for unstructured medical data – a big data approach. In *IEEE ICDMW 2015*, pages 455–462.

Istephan, S. and Siadat, M.-R. (2016). Unstructured medical image query using big data – an epilepsy case study. *Journal of Biomedical Informatics*, 59:218–226.

Kimball, R. and Ross, M. (2002). *The data warehouse toolkit: the complete guide to dimensional modeling, 2nd Edition*. Wiley.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Kuo, M., Chrimes, D., Moa, B., and Hu, W. (2015). Design and construction of a big data analytics framework for health applications. In *IEEE SmartCity 2015*, pages 631–636.

Li, D., Zhang, W., Shen, S., and Zhang, Y. (2017). SES-LSH: Shuffle-efficient locality sensitive hashing for distributed similarity search. In *ICWS 2017*, pages 822–827.

Nguyen, D.-T., Yong, C. H., Pham, X.-Q., Nguyen, H.-Q., Loan, T. T. K., and Huh, E.-N. (2016). An index scheme for similarity search on cloud computing using MapReduce over docker container. In *IMCOM 2016*, pages 60:1–60:6.

Nguyen, T. D. T. and Huh, E.-N. (2017). An efficient similar image search framework for large-scale data on cloud. In *IMCOM 2017*, pages 65:1–65:8.

Raja, P. V. and Sivasankar, E. (2014). Modern framework for distributed healthcare data analytics based on hadoop. In *ICT-EurAsia 2014*, pages 348–355.

Rocha, G. M. and Ciferri, C. D. A. (2018). ImgDW generator: a tool for generating data for medical image data warehouses. In *SBBD 2018 Proc. Companion*, pages 23–28.

Rocha, G. M. and Ciferri, C. D. A. (2019). Processamento eficiente de consultas analíticas estendidas com predicado de similaridade em Spark. In *SBBD 2019*, pages 1–6.

Sebaa, A., Chikh, F., Nouicer, A., and Tari, A. (2018). Medical big data warehouse: Architecture and system design, a case study: Improving healthcare resources distribution. *Journal of Medical Systems*, 42(4):59.

Sebaa, A., Nouicer, A., Chikh, F., and Tari, A. (2017). Big data technologies to improve medical data warehousing. In *BDCA 2017*, pages 21:1–21:5.

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The Hadoop distributed file system. In *IEEE MSST 2010*, pages 1–10.

Teixeira, J. W., Annibal, L. P., Felipe, J. C., Ciferri, R. R., and Ciferri, C. D. A. (2015). A similarity-based data warehousing environment for medical images. *Computers in Biology and Medicine*, 66:190 – 208.

Traina, C., Filho, R. F. S., Traina, A. J. M., Vieira, M. R., and Faloutsos, C. (2007). The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. *The VLDB Journal*, 16(4):483–505.

Traina, C., Moriyama, A., Rocha, G. M., Cordeiro, R., Ciferri, C. D. A., and Traina, A. J. M. (2019). The SimilarQL framework: similarity queries in plain SQL. In *SAC 2019*, pages 1–4.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. In *USENIX HotCloud 2010*.