Estudo de Soluções para implementação de Big Data com uso de Apache Hadoop e Spark

Gabriel V. De Souza[1], Lucas Bueno R. Barbosa[2],

Geraldo Henrique Neto[3], Anna Patricia Zakem China[4]

[1]Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

[1]gabriel.souza125@fatec.sp.gov.br,

[2]lucas.barbosa45@fatec.sp.gov.br,

[3]geraldo.henrique@fatec.sp.gov.br,

[4]anna.china@fatec.sp.gov.br

Resumo. *O grande volume de dados gerados principalmente por empresas e indivíduos, tem proporcionado a oportunidade de obter insights precisos. No entanto, esse volume de dados também apresenta vários desafios, um deles é manipular esses dados de forma eficiente e eficaz. Esse trabalho visa demonstrar uma solução para lidar com esse grande volume de dados utilizando ferramentas de Big Data. O uso combinado dessas ferramentas como, Hadoop e Spark potencializando a manipulação eficiente e eficaz dos dados coletados, obtendo uma solução completa para o processamento de Big Data. Dentre os benefícios obtidos no estudo de caso da Uber destacam-se a análise de dados em larga escala e com essas ferramentas torna-se possível analisar os dados em tempo real e utilizar essas informações para melhorar a experiência do usuário e otimizar seus serviços.*

Abstract. *The large volume of data generated mainly by companies and individuals has provided the opportunity to obtain accurate insights. However, this volume of data also presents several challenges, one of them is handling this data efficiently and effectively. This paper aims to demonstrate a solution to deal with this large volume of data by using Big Data tools. The combined use of these tools, such as Hadoop and Spark, leverages the efficient and effective manipulation of the collected data, obtaining a complete solution for processing Big Data. Large-scale data analysis stands out Among the benefits of the Uber case study. With these tools, it becomes possible to analyze data in real time and use this information to improve the user experience and optimize its services.*

1. Introdução

Big Data é uma área que tem ganhado paulatinamente mais destaque no mundo dos negócios e tecnologia. A grande quantidade de dados gerados diariamente por empresas e indivíduos torna-se uma oportunidade de gerar *insights* valiosos para o sucesso de uma organização. A utilização do *Big Data* permite a análise de dados em tempo real, a criação de modelos preditivos e a identificação de tendências e comportamentos do consumidor.

Dessa forma, as empresas conseguem tomar decisões mais assertivas e estratégicas para o negócio.

Ademais, o *Big Data* tem contribuído para o desenvolvimento de novas tecnologias e áreas de atuação. A inteligência artificial, por exemplo, tem sido amplamente utilizada em conjunto com o *Big Data* para a criação de soluções que antes eram inimagináveis. O *Big Data* é uma ferramenta que tem contribuído significativamente para a tomada de decisão e desenvolvimento de novas tecnologias e áreas de atuação.

Para a utilização efetiva de *Big Data*, é necessário utilizar ferramentas apropriadas. Essas ferramentas têm a finalidade de lidar com grandes volumes de dados, realizar análises complexas e extrair *insights* significativos. As mesmas são capazes de lidar com os desafios de armazenamento, processamento e análise de dados em larga escala. A integração de duas ferramentas como *Hadoop* e *Spark* pode contribuir com vários benefícios, como processamento e análise de dados, resultados em uma obtenção mais eficiente.

Neste sentido, esse trabalho tem por objetivo ilustrar como a combinação da capacidade de armazenamento escalável do *Hadoop* com o processamento em tempo real e flexibilidade de programação do *Spark*, torna-se possível processar grandes volumes de dados de forma eficiente, refinada e precisa. Obtendo informações relevantes para tomada de decisão. Conforme a experiência da Uber que utiliza a integração dessas ferramentas para o processamento de imagem dos seus mapas, com objetivo de melhorar a experiência dos seus usuários.

O trabalho é dividido em 5 seções, sendo a primeira seção a Introdução a qual discorre-se sobre o conceito de *Big Data* e suas soluções, bem como os benefícios de combinar duas ferramentas *Hadoop* e *Spark* para soluções de *Big Data* no estudo de caso da Uber. A segunda seção, o Referencial Teórico, é dividida em subseções e aborda as principais características dos 5Vs de *Big Data*, e o estudo de caso da Uber que tem como objetivo melhorar a experiência do usuário. Na terceira seção, Materiais e Métodos, são discorridas brevemente as histórias das ferramentas *Hadoop* e *Spark*. Em seguida, são abordados as etapas e os métodos de cada ferramenta. A quarta seção, Resultados, explica os benefícios obtidos no estudo de caso da Uber ao combinar as duas ferramentas, *Hadoop* e *Spark*. Finalizando o trabalho, na quinta seção, Considerações Finais, são explicados os benefícios do uso combinado do *Hadoop* e *Spark* para soluções de *Big Data*. A sexta seção, Referências, lista todas as fontes utilizadas para escrever esse trabalho.

2. Referencial Teórico

Como afirma Marquesone (2018, p.2): "*Big Data* ainda é um termo incipiente, gerando incertezas sobre sua definição, características, aplicabilidade e desafios".

No decorrer do tempo, torna-se mais crescente a conectividade da população e a geração constante de dados a partir de tudo que é feito na internet. Um dos fatores que contribui para esse crescimento exponencial de informações é a conectividade dos indivíduos – quanto mais tempo navegamos na internet, mais dados são gerados.

2.1. Big Data

Simplificando o termo *Big Data*, basicamente é um grande conjunto de dados. Uma das características da *Big Data* é o volume, muitos *softwares* tradicionais como sistemas de

gerenciamento de banco de dados (SGBD) não são capazes de gerenciar esses dados. Por fim, esse grande volume de dados pode ser utilizado para lidar com atividades de negócios, experiência do usuário até análise avançada (ORACLE BRASIL, [s.d]).

O termo *Big Data* geralmente evoca a ideia de um grande volume de dados, mas existem outras características que definem o *Big Data*, são elas variedade e velocidade. Essas características são conhecidas como os 3Vs de *Big Data* (MARQUESONE, 2018). Na Figura 1 é definido os 3Vs de *Big Data.*

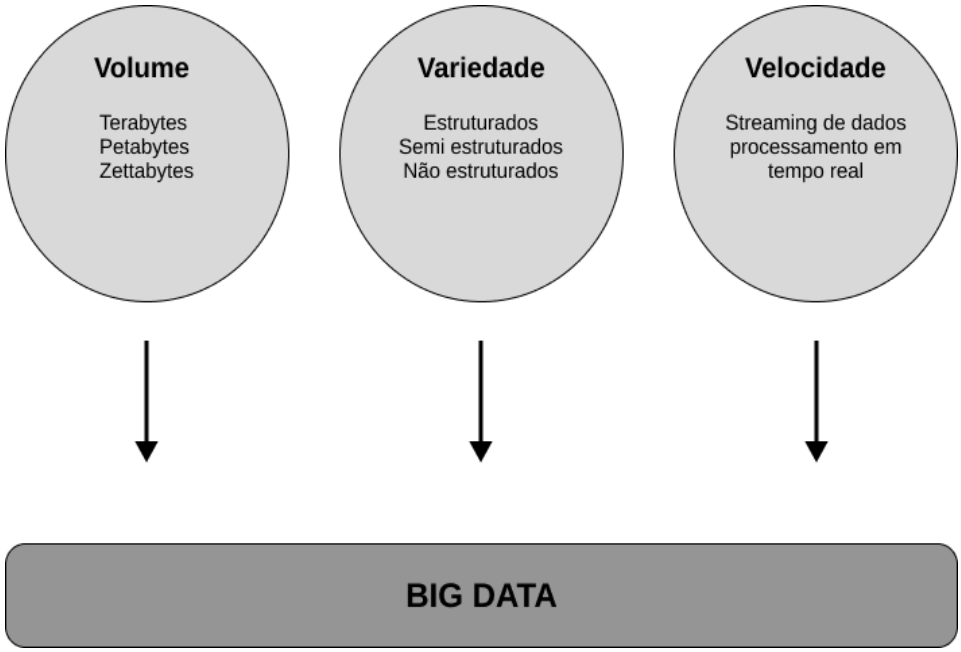

Figura 1. 3Vs de Big Data.
Fonte: (Marquesone, 2018, p.9)

2.1.1. Volume

O autor Marquesone (2018) explica volume como "O atributo volume é a característica mais significativa no conceito de *Big Data*. Ele faz referência à dimensão sem precedentes do volume de dados" (Marquesone, 2018, pag.9).

A seguir a origem desse grande volume de dados e algumas estatísticas que representam esse volume, conforme definido por Marquesone (2018, p. 9 e 10)

a) A cada segundo, cerca de 40.000 buscas são realizadas no Google.

b) A empresa Walmart manipula mais de 1 milhão de transações dos clientes por hora.

c) A rede social Facebook contabilizou em junho de 2016 uma média de 1,13 bilhões de usuários, 2,5 bilhões de compartilhamentos e 2,7 bilhões de "curtidas" diariamente.

d) A rede social de compartilhamento de fotos Instagram recebe atualmente cerca de 80 milhões de fotos por dia.

e) Em 2013, a plataforma de *blogs* Wordpress relatou a quantidade de 42 milhões de comentários por mês, entre os 3,6 bilhões de páginas existentes na plataforma.

De acordo com a Figura 2 a seguir, podemos visualizar o crescimento dos dados ao longo do tempo e uma projeção em médio prazo.

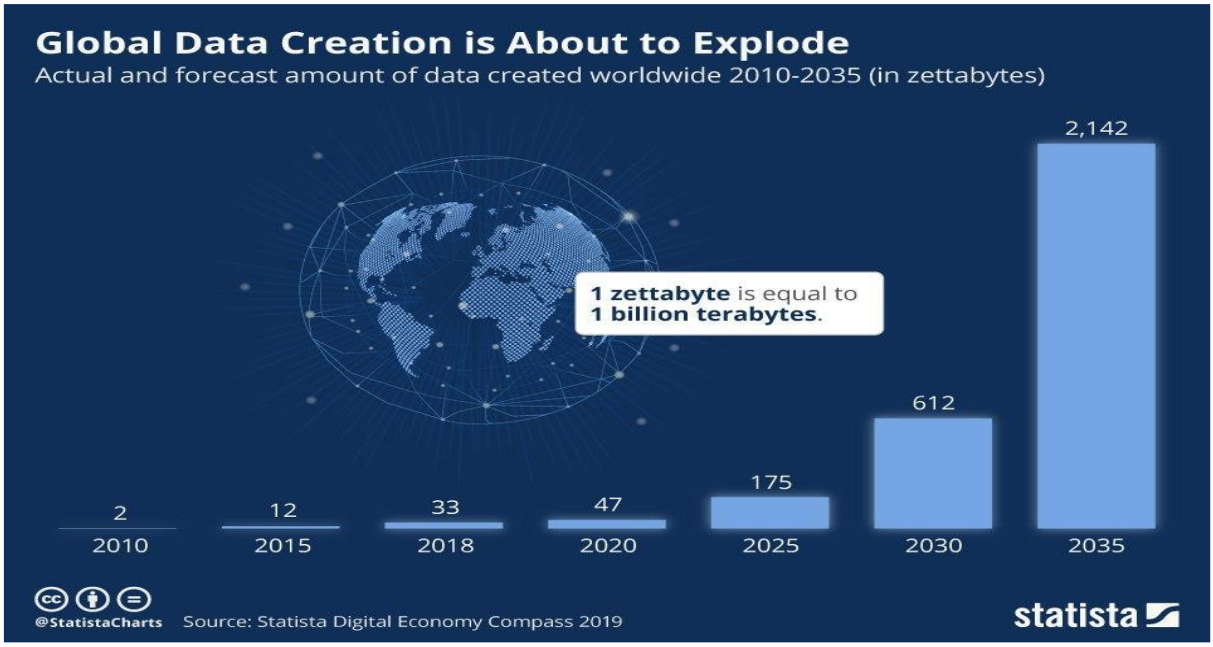

Figura 2. Crescimento dos dados ao longo dos anos
Fonte: (STATISTA, 2019)

Por fim, o atributo volume requer uma tecnologia de *Big Data,* quando as ferramentas tradicionais não são capazes de lidar com um determinado volume de dados.

2.1.2. Variedade

O atributo variedade refere-se aos vários tipos de dados. Os bancos de dados relacionais são projetados para armazenar dados em tabelas, tornando-se limitante para o *Big Data*. Pelo motivo de incluir vários tipos de dados, como semiestruturados e não estruturados (ORACLE BRASIL, [s.d]).

Dados semiestruturados possuem uma estrutura pré-definida e são normalmente apenas como um meio de marcação dos dados, como no caso dos arquivos *JSON* (*Javascript Object Notation*) e *XML* (*eXtensible Markup Language*). E dados não estruturados são basicamente vídeo e imagens (Marquesone, 2018).

2.1.3. Velocidade

Um exemplo do atributo velocidade em *Big Data* está relacionado a velocidade com que os dados estão sendo gerados. Estatísticas mostram que, em apenas um minuto, mais de 2 milhões de pesquisas são realizadas no Google, 6 milhões de páginas são visitadas no Facebook e 1,3 milhões de vídeos são vistos no Youtube. Para finalizar, vários aplicativos que mantem seus serviços em produção 24 horas por dia (Marquesone, 2018).

2.1.4. Valor e Veracidade

Além dos 3Vs de *Big Data* surgiram mais 2Vs nos últimos anos, eles são: valor e veracidade.

O valor conforme descrito no artigo Crawly (2021): “se uma empresa de *marketplace* quer saber mais sobre os produtos, fretes e condições de pagamento oferecidos pela concorrência, é por meio do *Big Data* que essas informações serão obtidas. Essas informações tornam-se dados acionáveis, ou seja, fazem a empresa reagir e auxiliam-na a tornar-se mais competitiva”.

Veracidade é basicamente a qualidade e a confiabilidade dos dados coletados, esses dados precisam estar completos e atualizados para ser extraído dados relevantes.

2.2. Aplicabilidade de Big Data

Sobre melhorar a experiência do usuário, grandes empresas utilizam *Big Data* com esse objetivo. Um estudo da Uber composto por uma equipe especialista em coleta de dados, capturam imagens de placas de rua com o objetivo de aprimorar a eficiência dos mapas e a qualidade dos dados. O resultado esperado é melhorar a experiência do usuário durante a viagem e dos motoristas parceiros. No caso da *Uber Eats*, a equipe tira fotos dos alimentos para utilização dentro do aplicativo. Após a coleta, as imagens e metadados são processados utilizando a ferramenta de *Big Data Apache Spark* e armazenamento *Hadoop Distributed File System* (*HDFS*). Essas coletas chegam a ultrapassar 8 bilhões de imagens (UBER ENGENEERING, 2019).

3. Materiais e Métodos

3.1. Ferramenta de Big Data: Apache Hadoop

O *Hadoop* foi uma das primeiras tecnologias de *Big Data* e ainda é amplamente utilizado em várias aplicações. Inicialmente, esse *framework open source* foi desenvolvido para um objetivo específico: ser uma *engine* de busca da *web* semelhante ao serviço da Google.

Desenvolvido por Doug Cutting e Mike Cafarella, o *Hadoop*, que antes fazia parte do projeto *Apache Nutch*, foi lançado oficialmente em 2006 com o nome de *Hadoop*. Esta tecnologia foi inspirada em duas soluções proprietárias da Google: o sistema de arquivos distribuídos *Google File System (GFS)* e o modelo de programação distribuída *MapReduce* (Marquesone, 2018).

Apesar de ter sido desenvolvido com um objetivo específico, várias empresas passaram a utilizar o *Hadoop* em diversas aplicações de *Big Data*. Uma das empresas foi o Yahoo, a qual até hoje é um dos principais utilizadores e colaboradores do *framework*, juntamente com outras empresas como o Twitter e o Facebook, que fizeram grandes contribuições (Marquesone, 2018).

3.2. Ferramentas de Big Data: Apache Spark

O *Spark* foi desenvolvido em 2009 por um grupo de pesquisadores da Universidade de Berkeley, utilizando a mesma tecnologia de processamento da Google, o *Google File System* e o algoritmo *MapReduce*, com uma diferença no processamento de dados em memória. Após um ano, tornou-se uma ferramenta *open source*, sendo incorporado pela *Apache Software Foundation* em 2013 (MEDIUM, 2021).

O objetivo do projeto *Spark* era melhorar o desempenho do processamento de dados em larga escala, superando as limitações do *framework* de processamento de dados distribuído anterior, o *Apache Hadoop*. Para o processamento de dados em lotes, o *Hadoop* utiliza o *MapReduce*, tornando o processamento em tempo real ou interativo mais difícil. Enquanto isso, o *Spark* tinha uma diferença em seu processamento de dados, projetado para uma abstração de processamento em memória, sendo mais rápido e eficiente (FORBES, 2015).

Em 2013, alguns criadores do *Spark* fundaram o *Databricks*, uma empresa que contribuiu para o seu desenvolvimento. Em 2014, o *Apache Spark* foi aceito como um projeto da *Apache Software Foundation*, tornando-se um dos projetos mais populares da *Apache* (DATABRICKS, [s.d]).

3.3. Análise da Aplicabilidade das Ferramentas pela Uber

No estudo de caso da Uber é descrito os formatos de arquivos que a equipe de dados do mapa utiliza para processar grandes volumes de imagens e metadados a fim de melhorar a experiência dos usuários (UBER ENGINEERING, 2019).

3.3.1. Processamento de dados com Spark e armazenamento no HDFS

Os dados são coletados por um aplicativo desenvolvido pela Uber para essa finalidade, e ao concluir a captura das imagens e metadados o aplicativo móvel faz o *upload* para o armazenamento em nuvem. Os dados são coletados do armazenamento e logo após são processados.

Na primeira etapa os dados brutos são coletados e organizados no centro de dados. Em seguida, várias etapas fazem a descompactação dos dados brutos e execução de processamento para refinar a veracidade dos dados para os consumidores de *downstream* (UBER ENGINEERING, 2019).

Na Figura 3 a seguir é apresentado as etapas de ingestão e processamento dos dados:

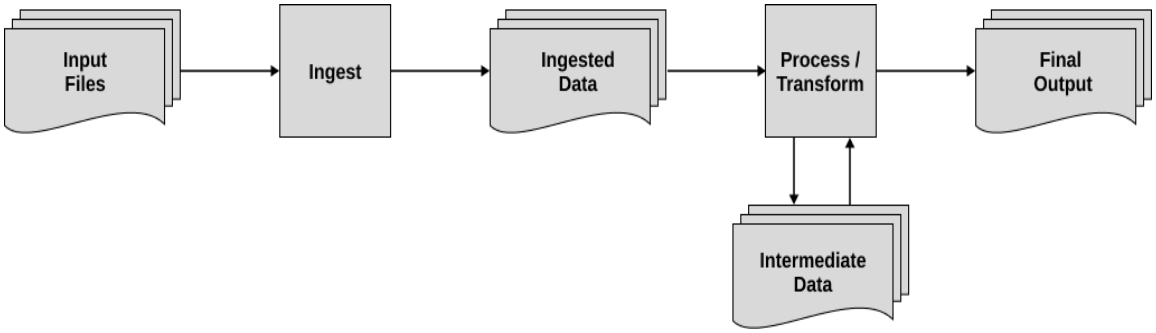

Figura 3. Os metadados de imagens e imagens são ingeridos nos centros de dados da Uber e, posteriormente, são processados utilizando o Apache Spark.
Fonte: (UBER ENGINEERING, 2019)

Formatos de arquivos utilizados para cada tipo de saída/resultado, como definido no estudo de caso da Uber (UBER ENGINEERING, 2019):

3.3.1.1. Dados ingeridos

Casos nos quais os dados são ingeridos eles são lidos de uma fonte externa e gravados no *HDFS*, para consumir menos recursos de memória no *NameNodes* os dados são gravados em arquivos grandes, sendo eficiente para os trabalhos *Spark.*

Os dados não são modificados durante a ingestão, mas podem ser processados posteriormente em paralelo por meio do *Spark*. Na etapa do processamento os dados são lidos por completo com pouca ou nenhuma filtragem.

Por fim cada instância de processamento salva os arquivos no formato (*SequenceFile*) do *HDFS*, resultando em arquivos grandes que é adequado para o *HDFS*, oferecendo o melhor desempenho de gravação e não sobrecarrega outros formatos de arquivos como o *Apache Avro* e *Apache Parquet*. Um *SequenceFile* é baseado em pares de chave/valor binários (UBER ENGINEERING, 2019).

3.3.1.2. Dados intermediários

Depois que os dados são gravados no *HDFS*, vários trabalhos *Spark* são executados para o processamento adicional dos dados. Cada trabalho grava dados intermediários.

Geralmente os dados gerados são lidos por completo e filtrados por coluna ou linhas. Para esse tipo de dados é usado o formato *Avro*.

No *Spark* o formato de arquivos padrão é o *Parquet,* esse formato oferece várias vantagens referente ao desempenho, mas essas vantagens têm um custo quando os arquivos são gravados.

O arquivo *Parquet* é baseado em colunas, resultando que todos os valores de uma coluna em todas as linhas armazenadas em um arquivo físico são reunidos antes de serem gravadas no disco. Sendo preciso armazenar todos os registos e realocar na memória antes que o arquivo seja gravado.

Por outro lado, o arquivo *Avro* é composto por registros permitindo que os registros sejam transmitidos para o disco de modo eficiente (UBER ENGINEERING, 2019).

Como definido no estudo de caso da Uber Engineering (2019) por ter uma sobrecarga menor de gravação, os *Avro* são utilizados para armazenar dados intermediários quando:

a) Os dados são gravados e lidos uma única vez.

b) O leitor não filtra os dados.
c) O leitor lê os dados sequencialmente.

3.3.1.3. Resultado/Saída Final

Nesse ponto os dados já estão processados e acessíveis para vários consumidores *downstream*. Cada um deles possui um requisito específico que são abordados ao consultar e filtrar os dados (UBER ENGINEERING, 2019).

No exemplo do estudo de caso, a Uber Engineering (2019), um consumidor pode executar uma consulta geoespacial em imagens de um tipo específico. O resultado/Saída final cai em três categorias:

a) Metadados de imagens

Diferente dos dados intermediários, os metadados são lidos várias vezes, filtrados e consultados. Esses dados são armazenados no *parquet* por ter um melhor desempenho, superando a sobrecarga adicional de gravação. A Uber realizou dois testes de desempenho dos formatos de arquivo *Avro* e *Parquet* em consultas de grande volume de dados, consultando aproximadamente cinco milhões de registros (UBER ENGINEERING, 2019).

O primeiro teste realizou uma consulta de caixa delimitadora simples na latitude e longitude das imagens que resulta aproximadamente 250.000 registros. Na Tabela 1 a seguir é apresentado os resultados do primeiro teste que demonstra a visão do consumidor *downstream*, a consulta *Parquet* é praticamente três vezes mais rápida de executar. O impacto está na infraestrutura subjacente (UBER ENGINEERING, 2019).

Resource	Avro	Parquet	Improvement
Wall Time (sec)	20.76	7.17	290%
Core Time (min)	24.80	1.28	1,94%
Reads (MB)	24,678.4	1,848.5	1,34%

Tabela 1. Resultados do primeiro teste
Fonte: (Uber Engineering 2019)

A maior melhoria é a *I/O* necessária para as consultas em que o *Parquet* consome 7,5% da *I/O* exigida pela consulta *Avro*. O formato de arquivo *Parquet* armazena estatísticas que reduzem significativamente a quantidade de dados lidos.

O segundo teste realizou uma comparação de *string*/sequência de caracteres que retorna aproximadamente 40.000 registros. Conforme resultados do segundo teste constantes na Tabela 2 o arquivo *Parquet* teve mais um desempenho superior ao arquivo *Avro*. A consulta *Parquet* é duas vezes mais rápido e requer 1,5% da *I/O*.

Resource	Avro	Parquet	Improvement
Wall Time (sec)	18.48	6.0	308%
Core Time (min)	1670.00	50.76	3,29%
Reads (MB)	24,678.4	376.6	6,55%

Tabela 2. Resultados do segundo teste
Fonte: (Uber Engineering 2019)

b) Imagens

As informações a seguir são referentes ao estudo de caso da Uber, 2019. O *Parquet* não é eficiente para armazenar grandes dados binários, como imagens, mas o *Avro* é adequado. Para suportar consultas, duas colunas são adicionadas aos arquivos *Parquet* de metadados de imagens para servir como chave estrangeira para as imagens. Essas colunas contêm o nome do arquivo de peça e o deslocamento do registro dentro do arquivo de peça, obtido por meio da *API* do *Avro* nativa.

Para gravar as imagens, o *Spark* divide os dados em partições e usa a API do *Avro* nativa para gravar um arquivo de peça individual que contenha todas as imagens contidas na partição. Os passos gerais são: ler o *SequenceFile* de ingestão, mapear cada partição, criar um *Avro Writer* padrão, iterar por meio de cada registro e gravá-los no arquivo *Avro*, chamar o *DataFileWriter.sync()* para liberar o registro para o disco e retornar o deslocamento do registro e, finalmente, salvar o formato *DataFrame* ou *RDD (Resilient Distributed Datasets)* no *Parquet* resultante.

Os resultados são um arquivo *Avro* com as imagens e um arquivo *Parquet* complementar que contém o caminho do arquivo *Avro* e o deslocamento de registro para consultar com eficiência um determinado registro de imagem.

O padrão geral para consultar e ler os registros de imagens é: consultar os arquivos *Parquet*, incluir o caminho do arquivo e o deslocamento nos resultados, opcionalmente reparticionar os resultados para ajustar o grau de paralelismo, mapear cada partição dos resultados da consulta e criar um leitor *Avro* padrão para o arquivo de peça *Avro* que contém o registro de imagem e chamar *DataFileReader.seek(long)* para ler o registro de imagem no deslocamento especificado.

c) Dados agregados

Para armazenar metadados agregados sobre um conjunto de imagens, a Uber utiliza arquivos *JSON*. Os metadados incluem a versão do *pipeline* usada para processar as imagens e a área geográfica em que foram coletadas. O uso de arquivos *JSON* permite fácil depuração, leitura eficiente e integridade referencial. Armazenar os arquivos *JSON*, *Avro* e *Parquet* em um diretório pai permite que todos os dados sejam arquivados com uma única operação *HDFS* atômica (UBER ENGINEERING, 2019).

4. Resultados

Dentre os benefícios obtidos como resultados pela Uber, destaca-se a análise de dados em larga escala. A Uber processa e armazena uma quantidade massiva de dados de seus motoristas, usuários e rotas.

O *Hadoop* é uma plataforma que permite o processamento de grandes quantidades de dados, enquanto o *Spark* é um *framework* para processamento de dados em larga escala em memória. Combinando essas tecnologias, a Uber pode analisar os dados de seus usuários e motoristas em tempo real e usar essas informações para melhorar a experiência do usuário, bem como para otimizar a logística da empresa. O uso do *Hadoop* e *Spark* pela Uber pode ter um impacto significativo também na experiência do usuário, permitindo que a empresa melhore a eficiência de seus serviços. A análise em tempo real dos dados de localização dos motoristas e dos usuários, por meio do *Spark Streaming*, pode permitir que a Uber monitore a disponibilidade de motoristas em tempo real. Portanto, a integração das tecnologias *Hadoop* e *Spark* proporciona dados que direcionam as tomadas de decisão no que tange a alocação de recursos de forma mais eficiente de modo a melhorar a experiência global dos usuários do aplicativo.

5. Considerações Finais

O *Big Data*, vem paulatinamente se tornado uma realidade cada vez mais presente nas organizações, impulsionado pela necessidade de soluções eficientes de processamento e análise de dados em larga escala. Nesse contexto, o *Hadoop* e o *Spark* surgem como duas das principais tecnologias que revolucionaram o campo do processamento distribuído e forneceram ferramentas poderosas para lidar com os desafios do *Big Data*.

Ao longo deste trabalho, exploramos as características e funcionalidades do *Hadoop* e do *Spark*, discutindo as diferenças, semelhanças e usos mais comuns. O *Hadoop*, com seu sistema de arquivos distribuído (HDFS) e o *framework MapReduce*, foi pioneiro na capacidade de processar grandes volumes de dados em *clusters* de computadores, permitindo a execução de tarefas paralelas de forma escalável e confiável. O *Spark*, por sua vez, surgiu como uma evolução do *Hadoop*, fornecendo um mecanismo de processamento em memória extremamente rápido e uma gama mais ampla de funcionalidades, incluindo suporte a SQL (*Structured Query Language*), *streaming* e aprendizado de máquina.

É importante ressaltar que o sucesso da implementação do *Hadoop* e do *Spark* em uma organização depende de vários fatores, sendo a principal uma infraestrutura adequada, com *clusters* de computadores dimensionados corretamente e configurados de forma otimizada. Além disso, é necessário ter profissionais capacitados e experientes em administrar, programar e otimizar essas tecnologias. Embora o *Hadoop* e o *Spark* sejam soluções maduras e amplamente adotadas, destaca-se que o campo do processamento distribuído e análise de *Big Data* está em constante evolução. Portanto, cabe ao profissional de TI manter-se atualizado sobre as tendências e inovações na área.

É possível deduzir que o *Hadoop* e o *Spark* têm desempenhado um papel fundamental na capacitação das organizações na era do *Big Data*. Devido as capacidades de processamento distribuído e análise em larga escala, permite às empresas tomar decisões mais precisas a partir de grandes volumes de dados, dessa maneira observa-se uma vantagem competitiva significativa. O futuro do processamento distribuído certamente reserva mais inovações e desafios, mas o *Hadoop* e o *Spark* estabeleceram um alicerce sólido para essa jornada, abrindo caminho para o avanço contínuo da análise de *Big Data* e do campo da inteligência artificial.

O objetivo de demonstrar a eficiência da integração das tecnologias foi

atingido, como exemplo fica o caso da Uber destacado neste trabalho, das dificuldades encontradas para realizar as pesquisas, destaca-se a escassez e dificuldade de encontrar material de estudos em língua portuguesa para equacionamento de informações. A solução neste caso foi utilizar materiais disponíveis em inglês.

Referências

Armstrong, M. (2019). *Global Data Creation is About to Explode*. Disponível em: <https://www.statista.com/chart/17727/global-data-creation-forecasts/>. Acesso em: 30 abr. 2023. Statista (Figura 2).

Crawly. (2021). Conheça os 5 Vs do *Big Data*. Disponível em: <https://www.crawly.com.br/blog/5-vs-do-big-data>. Acesso em: 13 abr. 2023.

Databricks [s.d]. Databricks é a empresa *lakehouse*. Disponível em: <https://www.databricks.com/company/about-us>. Acesso em: 30 abr. 2023.

Marquesone, R. F. (2018). *Big Data* técnicas e tecnologias para extração de valor dos dados. Casa do código.

Marr, B. (2015). "*Spark* ou *Hadoop* – Qual é a melhor estrutura de *Big Data*?". Disponível em: <https://www.forbes.com/sites/bernardmarr/2015/06/22/spark-or hadoop-which-is-the-best-big-data-framework/?sh=6f9ee9b9127e>. Acesso em: 30 abr. 2023.

Oracle Brasil [s.d]. *What is Big Data*. Disponível em: <https://www.oracle.com/br/big data/what-is-big-data/>. Acesso em: 12 abr. 2023.

Reis, L. (2021). "*Apache Spark*: Um *overview* do *framework*". Disponível em: <https://medium.com/engenharia-arquivei/apache-spark-um-overview-do-framework-1dad930dfb60>. Acesso em: 3 abr. 2023.

Short, S. (2019). *Introduction to the HDFS File Format in Apache Spark*. Traduzido por redação iMasters. (2019). Estudo de caso da Uber: escolhendo o formato de arquivo *HDFS* correto para seus trabalhos do *Apache Spark*. Disponível em: <https://imasters.com.br/desenvolvimento/estudo-de-caso-da-uber-escolhendo-o formato-de-arquivo-hdfs-correto-para-seus-trabalhos-apache-spark>. Acesso em: 5 abr. 2023.