UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CIÊNCIA DA COMPUTAÇÃO

ISABEL SOUZA DE CARVALHO

ESTUDO DE FERRAMENTAS OPEN SOURCE PARA ANÁLISE DE DADOS EM BIG DATA

São Luís
2016

ISABEL SOUZA DE CARVALHO

ESTUDO DE FERRAMENTAS *OPEN SOURCE* PARA ANÁLISE DE DADOS EM *BIG DATA*.

Monografia apresentada ao Curso de Ciência da Computação, da Universidade Federal do Maranhão, **como parte dos requisitos necessários** para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profª. Drª. Simara Vieira da Rocha

São Luís
2016

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a). Núcleo Integrado de Bibliotecas/UFMA

CARVALHO, Isabel Souza de.

ESTUDO DE FERRAMENTAS OPEN SOURCE PARA ANÁLISE DE DADOS EM BIG DATA / Isabel Souza de CARVALHO. - 2016.

65 f.

Orientador(a): Simara Vieira da ROCHA. Monografia (Graduação) - Curso de Ciência da

Computação, Universidade Federal do Maranhão, São Luís/MA, 2016.

1. Análise de Dados. 2. Big Data. 3. Ferramentas Open Source de Big Data. I. ROCHA, Simara Vieira da. II. Título.

ESTUDO DE FERRAMENTAS *OPEN SOURCE* PARA ANÁLISE DE DADOS EM *BIG DATA*.

Monografia apresentada ao Curso de Ciência da Computação, da Universidade Federal do Maranhão, **como parte dos requisitos necessários** para obtenção do grau de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Profª Drª Simara Vieira da Rocha
(Orientadora)

Profº Me.
Carlos Eduardo Portela Serra de Castro
(Membro da Banca Examinadora)

Profº Me.
Maria Auxiliadora Freire
(Membro da Banca Examinadora)

Aprovado em: ___/___/_____
Nota: _____

AGRADECIMENTOS

A Deus, pelo dom da vida e pela força para enfrentar os obstáculos do dia-a-dia.

À minha família, pelo amor incondicional oferecido. Especialmente à minha mãe, por ser minha base ao longo da minha jornada acadêmica.

À minha orientadora, Profª Drª Simara Vieira da Rocha, pela proposta do tema, pela paciência durante o período de desenvolvimento do trabalho e pelas palavras de incentivo.

Aos professores do Curso de Ciência da Computação da UFMA, pela dedicação, atenção e conhecimento oferecido, visando sempre o crescimento profissional e pessoal dos seus alunos.

Aos amigos de turma, pelo companheirismo e amizades conquistadas.

Aos amigos do Programa de Educação Tutorial do Curso de Ciência da Computação (PetComp), pela companhia e pelas experiências únicas vividas ao longo das atividades do programa.

Ao Profº Drº Alexandre Cesar Muniz de Oliveira, meu tutor durante o período em que participei do PetComp, pela seriedade e responsabilidade com a qual sempre conduziu o programa.

Ao meu namorado e também melhor amigo, Agostinho Cardoso Nascimento Pereira, por todo carinho, apoio e incentivo nos diversos desafios que enfrentei.

À todas as pessoas, que direta ou indiretamente, contribuíram para a realização deste trabalho.

“Tenho posto minha confiança em Deus; não terei medo”
Salmos 56:4

RESUMO

É cada vez mais notável a influência dos dados no atual cenário dos negócios. Em algumas áreas de atuação, as coleções de dados ganham proporções de *exabytes*. O termo utilizado para descrever essas gigantescas coleções é o *Big Data*. O mais importante em um *Big Data* não é a quantidade de dados em si, mas o valor que as empresas conseguem extrair dele. Existem diversas ferramentas para analisar grandes conjuntos de dados, cada ferramenta possui características e propriedades distintas. Portanto, constitui um verdadeiro desafio para as empresas a escolha da ferramenta adequada. O presente trabalho realiza um estudo detalhado das principais ferramentas *open source* para análise de dados em *Big Data*, cujo objetivo é auxiliar o processo de escolha da ferramenta adequada conforme o perfil e as necessidades da organização.

Palavras-chave: *Big Data*, Análise de Dados e Ferramentas *Open Source* de *Big Data*.

ABSTRACT

It is increasingly noted the influence of the data in the current business scenario. In some areas, data collections gain exabytes proportions. The term used to describe these giant collections is Big Data. The most important in a Big Data is not the amount of data itself, but the value that companies can extract from it. There are several tools for analyzing large sets of data, each tool has different characteristics and properties. So is a real challenge for companies to choose the appropriate tool. This paper makes a detailed study of the major open source tools for Big Data in data analysis, which aims to assist the process of choosing the appropriate tool according to the profile and needs of the organization.

Keywords: Big Data, Data Analysis and Tools Open Source Big Data.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 Introdução

O tratamento de dados está presente na história humana revelando sua importância em diferentes áreas de aplicação. Se pudermos imaginar uma linha do tempo com os principais fatos ocorridos desde a invenção da escrita (aproximadamente 3000 a.C) até os dias de hoje, possivelmente veríamos o registro de conteúdos gerando inúmeras coleções de dados nas diversas épocas vividas (SAMPAIO, 2009).

A grande quantidade de dados gerados se torna mais visível nos dias atuais. Estima-se que no ano de 2007, foi possível armazenar $2{,}9 \times 10^{20}$ *bytes* e comunicar 2×10^{21} *bytes* de informações (HILBERT e LOPEZ, 2011). Segundo pesquisa realizada pela empresa Pingdom, que fornece soluções para o monitoramento de desempenho em aplicações *web*, a quantidade de dados circulando pela *internet* é expressiva. Dentre uma série de dados coletados, a pesquisa afirma existirem 634 milhões de *sites* na *web*, 144 bilhões de *emails* trafegando diariamente e 1,3 *exabytes* de dados trafegando mensalmente em redes móveis (PINGDOM, 2013).

Das placas de barro e argila na antiga Mesopotâmia aos *sites* hospedados na *web*, a quantidade de registros cresceu consideravelmente. Em meio ao cenário mais recente, de gigantescas coleções de dados, surge o conceito de *Big Data* (BD).

A International Data Corporation (IDC), empresa global fornecedora de serviços de consultoria em tecnologia da informação, define Big Data como uma nova geração de tecnologias e arquiteturas, concebidas para extrair valor a partir de grandes volumes de dados. Assim, tais tecnologias devem permitir alta velocidade de captura, descoberta e análise de uma ampla variedade de dados (GANTZ e REINSEL, 2011).

Para Lohr (2012), Big Data é também uma abreviação para impulsionar novas tendências em tecnologia, abrindo portas para novas abordagens na tomada de decisão e na compreensão do mundo. Não trata-se apenas de gerenciar mais fluxos de dados, mas também de gerenciar novos tipos de dados. Por exemplo, interligar sistemas e ferramentas organizacionais aos dados provenientes de sensores, de dispositivos móveis, de redes sociais, dentre outras aplicações.

Em geral, *Big Data* refere-se aos conjuntos de dados que não podem ser adquiridos, gerenciados e processados por ferramentas tradicionais de tecnologia de informação (CHEN, 2012).

Na tentativa de compreender o termo *Big Data* diversos conceitos que podem ser admitidos. Percebe-se a existência de um potencial a ser explorado pelas organizações, grandes coleções de dados podem se tornar informações valiosas no mercado.

Pesquisas recentes feitas pela IDC apontam que a tecnologia de Big Data e o mercado de serviços apresentam um rápido crescimento com múltiplas oportunidades em todo o mundo. A previsão da IDC mostra que o mercado de soluções em *Big Data* vai crescer a uma taxa anual composta de 26,4% de crescimento até 2018, ou seja, cerca de 6 vezes a taxa de crescimento do mercado global de tecnologia da informação (IDC, 2016).

No Brasil, o crescimento previsto é de 2,6% em comparação ao ano passado, a previsão é de que este mercado movimente US$ 811 milhões no Brasil em 2016 (IDC Brasil, 2016).

As soluções para tratamento de *Big Data* disponíveis no mercado estão ligadas a uma das etapas que compreendem o processo de criação de uma BD. De modo geral, as fases abrangem: geração, aquisição, armazenamento e análise de dados. Cada fase é fundamental para a efetividade no tratamento de um *Big Data*. Destaca-se a última fase, por ser responsável pela geração de valor de fato em um BD.

Para Mysore (*et al*, 2014) a escolha da solução apropriada para um *Big Data* é um desafio, pois é preciso considerar muitos fatores. Há diversas maneiras de adquirir, armazenar, processar e analisar um BD. Cada organização tem características diferentes, incluindo frequência, volume, velocidade, tipo e veracidade dos dados.

Portanto, antes de adoção de determinada solução para o tratamento de um conjunto grande de dados é fundamental alinhar as reais necessidades da organização junto às características da solução a ser escolhida.

1.1 Objetivos

1.1.1 Objetivo Geral

Apresentar um estudo detalhado das principais ferramentas *open source* para análise de dados em *Big Data.* Com base no presente estudo, promover a indicação das ferramentas adequadas para solucionar o problema exposto no estudo de caso.

1.1.2 Objetivos Específicos

- Pesquisar o conceito de *Big Data* e suas tecnologias;
- Identificar e avaliar as principais ferramentas *open source* aplicadas para análise de dados em *Big Data*;
- Apresentar um estudo de caso para exemplificar a teoria estudada, através do uso de uma metodologia que vise auxiliar a organização na escolha da ferramenta adequada para análise de dados.

1.2 Organização do Trabalho

O presente trabalho é composto por mais 3 capítulos.

O capítulo 2 apresenta a fundamentação teórica necessária para compressão deste trabalho. São abordados os seguintes tópicos: conceito, propriedades e etapas para criação de um *Big Data*, ferramentas *open source* para análise de BD, e em seguida as 7 ferramentas avaliadas: MapReduce, Apache Hadoop, Apache Storm, Apache Drill, RapidMiner, Apache Hortonworks e GridGain.

O capítulo 3 descreve o estudo de caso utilizado para demonstrar a aplicabilidade das características das ferramentas estudadas no capítulo 2, para

tanto propõem-se uma metodologia para auxiliar a organização no processo de escolha da ferramentas adequada.

Por fim no capítulo 4, são apresentadas as considerações finais a respeito deste trabalho, também discutidas sugestões de trabalhos futuros.

2 Fundamentação Teórica

Este capítulo descreve os temas que servirão de base para o desenvolvimento deste trabalho. Serão abordados: conceito e propriedades de *Big Data* (BD), etapas para criação de um BD e as ferramentas *open source* para análise de dados em *Big Data*. Dentre as ferramentas *open source* para Análise de BD serão abordadas as seguintes: MapReduce, Apache Hadoop, Apache Storm, Apache Drill, RapidMiner, Apache Hortonworks e GridGain. Após a explanação sobre cada ferramenta, será feita uma síntese das ferramentas *open source* em relação às características como: acesso, tipo, processamento e armazenamento de dados, análise em tempo real e segurança.

2.1 Conceito de *Big Data*

O termo *Big Data* é relativamente novo, aponta para uma definição conceitual com aplicações práticas. Para Laney (2001), coleções de dados identificadas como *Big Data* processam um grande volume, com complexa variedade e alta velocidade na geração dos dados.

São conjuntos grandes de dados, que desafiam a capacidade de ferramentas típicas de banco de dados para capturar, armazenar, gerenciar e analisar (MANYIKA, *et al*, 2011). De acordo com o referido autor, esta definição é subjetiva e não deixa claro o quão grande um conjunto de dados deve ser para ser considerado como *Big Data*. O autor ressalva ainda que, com os avanços tecnológicos ao longo do tempo, o tamanho das coleções de dados classificadas como *Big Data* irá aumentar, e que a definição de “conjuntos grandes de dados” pode variar de acordo com a área de atuação e das ferramentas de *software* que são comuns a tais setores.

Segundo Chen (*et al*, 2014), o termo *Big Data* é utilizado para caracterizar os conjuntos de dados e aplicações grandes e complexos que necessitam de *software* adequado para armazenamento, gerenciamento e visualização. Dados considerados

grandes possuem tamanho entre *terabytes* a *exabytes*, e são considerados complexos dados oriundos de diversas fontes como sensores e dados de mídia.

Com os dados provenientes de diversas fontes, eles possuem características distintas tornando complexo o tratamento dos mesmos. O *Big Data*, associado ao desenvolvimento da computação em nuvem e da memória de dispositivos, figura como uma solução para tornar a gestão de dados mais acessível (McCUE, 2007).

Portanto o *Big Data* trata-se de uma nova descoberta para processar uma vasta quantidade de informações, analisá-las instantaneamente e demonstrar conclusões claras sobre o objeto em questão.

2.2 Propriedades do *Big Data*

É comum abordar as propriedades do *Big Data* pela definição nomeada de 3V's (volume, velocidade e variedade), embora mais recentemente se fale de uma nova definição chamada 5V's (volume, velocidade, variedade, veracidade e valor).

A propriedade de volume é a dimensão que caracteriza o tamanho dos conjuntos dos dados. Não há um limite definido por uma quantidade especificada de *bytes* que devem ser atendidos para definir um *Big Data* (MANYIKA, *et al*, 2011). O crescimento do volume de dados manipulados pelas organizações por si só não garante vantagem competitiva, pois existem empresas que vêm as informações como um ativo tangível, descartando parte dessas informações sem nenhuma análise (LANEY, 2001).

A velocidade como característica de um *Big Data* diz respeito à celeridade com que os dados são gerados, armazenados e tratados. Além dessa abordagem, existe a definição para velocidade como a rapidez na qual os dados fluem e são tratados. Ou seja, em algumas situações as organizações devem ser capazes de analisar dados em tempo quase real (ZIKOPOULOS, et al, 2012).

O propósito da variedade é demonstrar que em um *Big Data* existem diversos tipos de dados, provenientes de variadas fontes. Essa diversidade de dados incluem dados estruturados, semi-estruturados e não estruturados. As organizações precisam integrar e analisar tais dados que surgem de dentro e fora da empresa

gerados de inúmeras formas, incluindo: texto, dados da *web*, dados de sensores, dentre outros (SCHROECK e SMART, 2012).

A veracidade é a qualidade de dados que correspondem com a verdade, ou seja, refere-se ao nível de confiabilidade associada aos dados. Portanto, manter dados com qualidade torna-se uma exigência para as organizações, e ao mesmo tempo um desafio. Há casos em que a imprevisibilidade é inerente, como o tempo, a economia ou as decisões do cliente, dessa forma, o gerenciamento de dados em meio às incertezas mostra-se complexo (SCHROECK e SMART, 2012).

O componente valor representa o diferencial competitivo incorporado às organizações que utilizaram o *Big Data* e foram bem sucedidas em suas áreas de atuação. Assim, o valor revela se o uso efetivo de *Big Data* tem o potencial de transformar as economias, fornecendo uma nova onda de crescimento da produtividade devido ao processamento de dados com um elevado volume, variedade e/ou velocidade (TIEFENBACHER e OLBRICH, 2015).

2.3 Etapas para criação de um *Big Data*

Para extrair valor de grandes volumes de dados, as empresas precisam executar ações específicas, descritas como uma cadeia de valor *Big Data*. Essas ações compreendem as etapas necessárias para converter dados em informações valiosas às organizações (MILLER e MORK, 2013). A Tabela 1 nos mostra uma proposta de cadeia de valor de dados, que pretende gerenciar dados desde a sua geração ao apoio na tomada de decisão.

Tabela 1: Cadeia de valor de dados. Fonte: (MILLER e MORK, 2013).

1 Descoberta de Dados			2 Integração de Dados	3 Exploração de Dados		
Coleta e Anotação	Preparação	Organização	Integração	Análise	Visualização	Decisão
Criar um inventário de fontes de dados e metadados.	Permitir o acesso às fontes e atualização de controle de regras.	Identificar sintaxe, estrutura e semântica para fonte de dados.	Estabelecer uma representação comum de dados.	Análise integrada de dados	Aplicações de apoio	Determina as ações conforme análise dos dados.

Segundo o referido autor, a primeira etapa de Descoberta de Dados deve incluir não apenas a descrição e enumeração dos dados, mas também a preparação e organização dos mesmos. Na fase da Coleta e Anotação as fontes de dados são descritas em termos de integridade, validade, consistência, atualidade e precisão. A fase seguinte de Preparação deve estabelecer acesso às fontes de dados por meio de um sistema compartilhado, além da criação de regras, isto é, restrições de segurança e privacidade para utilização de dados. Na fase de Organização são tomadas decisões sobre a sintaxe, estrutura e semântica dos dados.

Ainda conforme Miller e Mork (2013), com os dados devidamente organizados é chegada a etapa de Integração de Dados, responsável por criar uma representação comum aos dados. Essa etapa constitui um mapeamento que definirá quais representações comuns se referem às fontes de dados anteriormente catalogadas.

Após a Descoberta e Integração de dados a organização está apta para explorá-los. A etapa de Exploração de Dados é composta por três fases, a começar pela Análise, que consiste no processo intermediário entre entradas e resultados, reforçando a validade dos dados. Nessa fase são utilizados técnicas específicas como *MapReduce*. A seguir temos a fase de Visualização que envolve a apresentação dos resultados para os tomadores de decisão, o objetivo é fornecer aos principais interessados a informação em um formato adequado e conveniente. Dada a visualização dos resultados é oportuno determinar a decisão. Na última fase da cadeia de valor, os analistas podem usar o resultados visualizados para alterar um comportamento negativo ou recompensar um positivo, compreender os detalhes de um problema específico ou planejar ações futuras (MILLER e MORK, 2013).

Outra representação da cadeia de valor de *Big Data*, proposta por Chen (*et al*, 2014), nos mostra o processo de criação de valor a partir de grandes volumes de dados dividido em quatro fases: geração de dados, aquisição de dados, armazenamento de dados e análise de dados, conforme a Figura 1:

Figura 1: Cadeia de Valor de Big Data. Fonte: adaptado de (CHEN, *et al*, 2014).

A geração de dados é o primeiro elo da cadeia, corresponde ao processo de produção das informações. Os dados provêm de variadas fontes gerando conjuntos de dados em grande escala.

A segunda fase estabelece a aquisição de dados, que inclui a coleta, a transmissão e o pré-processamento de dados. Uma vez que coletados os dados brutos, utiliza-se a transmissão para enviá-los para um sistema de armazenamento adequado. As operações de pré-processamento de dados são indispensáveis para garantir o armazenamento de dados eficientes, diminuindo a redundância.

O armazenamento de dados é o processo de reunir e conservar os dados, que deve garantir a confiabilidade e disponibilidade para acesso aos dados.

Finalmente, ocorre a análise de dados, esse processo envolve métodos analíticos, arquiteturas e ferramentas para mineração e análise de conjuntos grandes de dados. O objetivo principal dessa fase é a extração de valores úteis às organizações, fornecendo sugestões ou decisões para os gestores das mais diversas áreas de atuação (CHEN, *et al*, 2014).

2.4 Ferramentas *open source* para análise de dados em *Big Data*

Conforme Chen (*et al*, 2014), a análise é a fase mais importante na cadeia de valor de *Big Data*, pois proporciona aos gestores a capacidade de tomar decisões baseados nas informações concernentes ao seu empreendimento.

Diretamente proporcional ao crescimento dos dados está o interesse por analisá-los, por essa razão a fase da análise de dados apresenta várias ferramentas e técnicas disponíveis (MILLER e MORK, 2013).

Dentre as ferramentas disponíveis para análise de dados focaremos nas alternativas *open source*. Para ser qualificado como *open source* o programa, ferramenta ou aplicação deve possuir uma licença permitindo sua livre distribuição e utilização, sem custos ao usuário (GARCIA, 2005).

Examinaremos as seguintes ferramentas: MapReduce, Apache Hadoop, Apache Storm, Apache Drill, RapidMiner, Apache Hortonworks e GridGain.

2.4.1 MapReduce

É um modelo de programação (geralmente associado à uma aplicação) para o processamento de conjuntos grandes de dados. O termo faz referência à duas tarefas distintas e separadas. A primeira é o *map*, que “mapeia” um conjunto de dados e converte-o em um outro conjunto, onde os elementos desse novo conjunto são agrupados em tuplas, ou seja, em pares de chave/valor. A segunda tarefa é o *reduce* que tem como entrada as saídas produzidas anteriormente pelo *map,* reduzindo-as em um conjunto menor de tuplas (DEAN e GHEMAWAT, 2008).

A implementação do MapReduce tem em vista a execução de muitos *terabytes* de dados sob *clusters* complexos de máquinas, entretanto, vejamos um exemplo simples sobre o funcionamento do MapReduce. Supondo que tenhamos um arquivo para ser processado, esse arquivo é composto por duas colunas: uma representa as cidades (chave) e a outra representa as temperaturas registradas naquela cidade (valor), conforme a Tabela 2:

Tabela 2: Cidades e Temperaturas

CIDADE	TEMPERATURA
Barão de Grajaú	37
Caxias	32
São Luís	30
Barão de Grajaú	38
São Luís	31
Barreirinhas	35

O objetivo é utilizar MapReduce para encontrar as temperaturas máximas em cada cidade. Utilizando a função *map* nesse exemplo, teríamos como saída o agrupamento dos dados da seguinte forma: (Barão de Grajaú, 37), (Caxias, 32), (São Luís, 30), (Barão de Grajaú, 38), (São Luís, 31) e (Barreirinhas, 35). Observe que alguns nomes de cidades ocorrem mais de uma vez no arquivo e por isso foram “mapeadas” várias vezes. A partir desse ponto entra a função *reduce,* que combina os resultados do *map* em um único valor para cada cidade, buscando a máxima temperatura correspondente. Os resultado obtidos pelo *reduce* seriam: (Barão de Grajaú, 38), (Caxias, 32), (São Luís, 31) e (Barreirinhas, 35).

É possível desenvolver várias implementações da interface MapReduce que sejam apropriadas ao ambiente de trabalho a ser utilizado, ou seja, uma versão pode ser adequada à uma máquina com pouca memória, já outras versões para grandes processadores. A implementação descrita no presente trabalho foi projetada pela Google (DEAN e GHEMAWAT, 2008).

O modelo de programação MapReduce é utilizado na Google para várias finalidades diferentes, dentre as quais citamos a geração dos dados para o serviço de busca na *web*, para classificação e mineração de dados, para a aprendizagem de máquina, e muitos outros sistemas (DEAN e GHEMAWAT, 2008).

A Figura 2 nos mostra uma visão geral da operação do *MapReduce*.

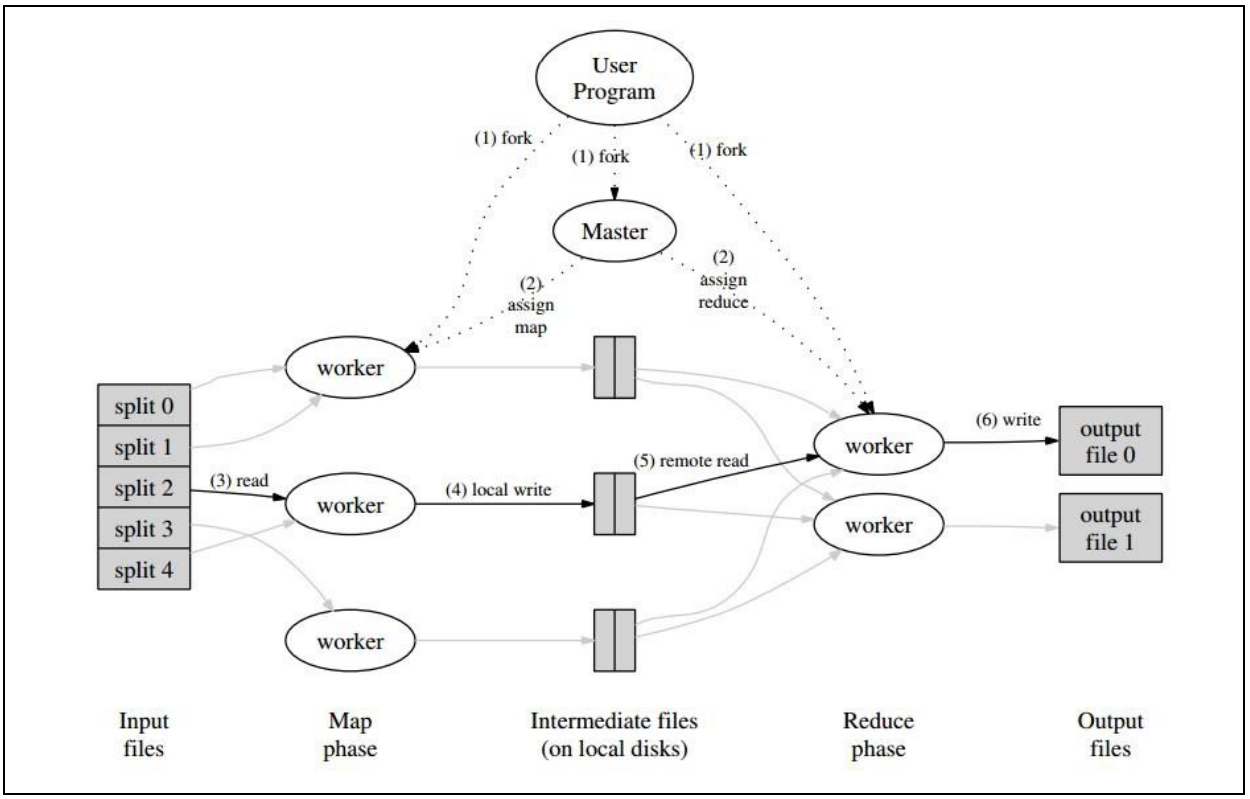

Figura 2: Visão geral do funcionamento do MapReduce. Fonte: DEAN e GHEMAWAT, 2008.

Quando o usuário faz referência à função MapReduce, ocorre uma sequência de ações conforme listado a seguir:

1. A função *MapReduce* inicia dividindo os arquivos de entrada (em um número de conjunto controláveis pelo usuário), em seguida, inicia uma série de cópias do programa nas máquinas que compõem o *cluster* computacional;
2. Uma das cópias do programa é escolhida como “mestre” e as demais serão “escravos”, os programas “mestres” atribuirão o trabalho aos “escravos”, que consiste em executar X funções de *Map* e Y de *Reduce*;

3. Determinado programa “escravo” (para o qual foi atribuído uma tarefa de mapeamento) deve ler o conteúdo correspondente à sua entrada, separando-as em tuplas de chave-valor. As tuplas serão armazenadas na memória;
4. Periodicamente, as tuplas armazenadas na memória são escritas em disco, particionadas em regiões pela função de particionamento. A localização das tuplas são passadas para o programa “mestre”, por sua vez, responsável por encaminhar esses locais para os “escravos” executarem o *Reduce*;
5. Quando um “escravo” que executará *Reduce* é notificado pelo “mestre”, ele utiliza Chamada de Procedimento Remoto para ler os dados dos “escravos” que executaram anteriormente o *Map*. Quando o “escravo” *Reduce* é executado, ele agrupa as tuplas de mesma ocorrência (repetidos) em conjuntos. Se a quantidade de dados gerados nessa fase for grande demais para a memória, um tipo de ordenação externa é usado;
6. O “escravo” *Reduce* deve enviar para a função *Reduce* os valores de uma determinada chave (que foi produzido pela função *Map* anteriormente);
7. Quando todas tarefas de *Map* e *Reduce* forem concluídas, o “mestre” acorda o programa do usuário e retorna o controle para ele.

Ao final da execução bem sucedida, são gerados arquivos disponíveis aos usuários para serem utilizados em aplicações específicas ou em outra chamada ao MapReduce (DEAN e GHEMAWAT, 2008).

Por se tratar de um modelo de programação, o MapReduce não compreende uma ferramenta em si, portanto exemplos de empresas que processam MapReduce estão associados à uma aplicação ou a uma das outras ferramentas para Análise de BD.

2.4.2 Apache Hadoop

É um projeto de *software* para soluções em sistemas distribuídos da organização Apache. Esse projeto inclui os módulos: Hadoop YARN (ambiente para programação e gestão de recursos), Hadoop MapReduce (sistema baseado no ambiente Hadoop YARN para processamento de grandes conjuntos de dados), Hadoop Distributed File System - HDFS (sistema distribuído de arquivo que fornece

acesso aos dados) e Haddop Commom (utilitários que dão suporte aos demais módulos) (APACHE HADOOP, 2015).

O Hadoop YARN compõe o projeto central Haddop, é a estrutura de gestão de recursos que permite o processamento de dados em múltiplas formas simultaneamente: lote, interativo e cargas de trabalho de dados em tempo real em um conjunto de dados compartilhado (APACHE HADOOP, 2015).

O YARN funciona também como pré-requisito para o Hadoop permitir uma grande variedade de métodos de acesso a dados. Por sua vez, o acesso aos dados no Hadoop pode ser feito a partir de várias ferramentas disponibilizadas pela Apache. A Tabela 3 mostra as principais ferramentas para o acesso aos dados.

Tabela 3: Principais ferramentas para acesso aos dados. Fonte: adaptado de (APACHE, 2015).

Ferramenta	Característica
Apache Hive	Tecnologia de acesso a dados mais amplamente adotada. Construído sobre a estrutura MapReduce, Hive é um depósito de dados que permite uma sumarização de dados e consultas via uma interface parecida com SQL para grandes conjuntos de dados armazenados em HDFS.
Apache Pig	Plataforma de *script* para processar e analisar grandes conjuntos de dados. Utiliza linguagem de alto nível chamada Pig Latin. Pig traduz o *script* Pig Latin em MapReduce para que ele possa ser executado no sistema de arquivo HDFS.
Apache HBase	Sistema que armazena dados NoSQL orientado por colunas, oferece acesso *read/write* aleatório em tempo real a *Big Data* para aplicativos de usuário.
Apache Accumulo	Ferramenta que promove acesso rápido a *Big Data*. Trabalha com controle de acesso em nível de célula.
Apache Tez	Estrutura para projetos de acesso em lote de alta performance e aplicações com processamento interativo de dados. Beneficia o funcionamento do *Map/Reduce* e torna mais rápido o tratamento de grandes conjuntos de dados.

Com base nas informações presentes na Tabela 3, percebe-se que o Hadoop trabalha com dados estruturados, acessados através das ferramentas Apache Hive, Pig, Accumulo e Tez; e dados não estruturados acessados pela Apache HBase. As ferramentas mostradas na Tabela 3 podem atuar em um conjunto e também associadas à outros mecanismos para acesso a dados.

O outro componente fundamental no projeto Hadoop é o Hadoop MapReduce. Trata-se do processamento do modelo de programação MapReduce em paralelo ao sistema de arquivo HDSF, conforme a Figura 3:

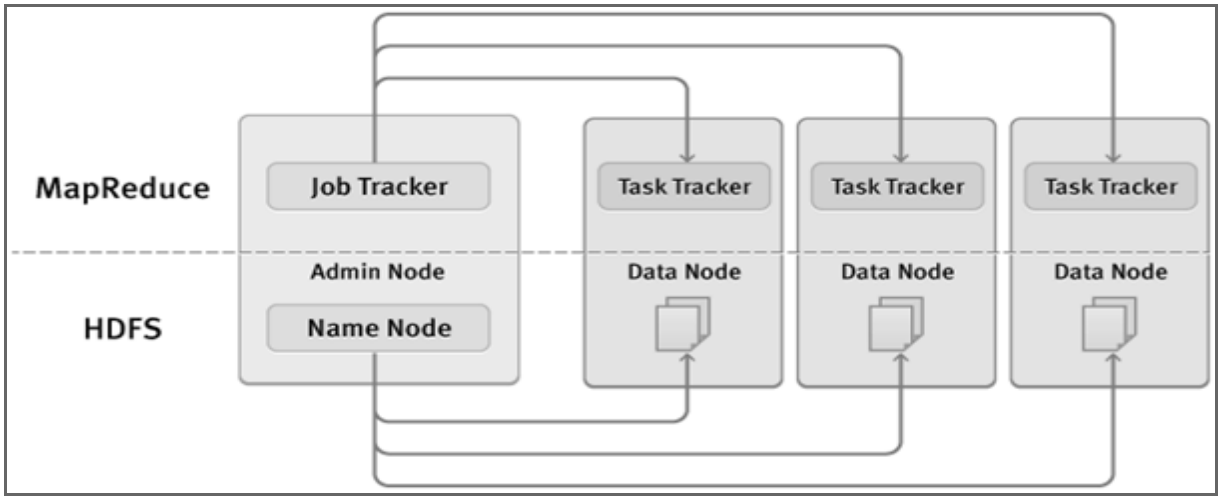

Figura 3: Componentes Apache Hadoop. Fonte: adaptado de (WHITE, 2012).

O *MapReduce* ocorre em duas fases, especificando as funções de mapeamento e redução, ambas ocorrem em paralelo. O armazenamento necessário para essa funcionalidade é promovido pelo HDFS.

Os principais componentes do *MapReduce* são: *Job Tracker* e *Task Tracker.* São nós que controlam o processo de execução. Os *Job Trackers* coordenam todo o trabalho do sistema por escalonamento das tarefas entre os *Task Trackers*. Por sua vez, os *Task Trackers* executam as tarefas e comunicam seus status de funcionamento aos *Job Trackers*. Se algum *Task Tracker* falhar, o *Job Tracker* pode reescalonar a tarefa a outro *Task Tracker* (WHITE, 2012).

Hadoop MapReduce é desenvolvido em linguagem de programação Java, porém pode ser trabalhado em outras linguagens, o Hadoop fornece um API para escrita das funcionalidades de *map* e *reduce* em outras linguagens de programação.

O projeto Hadoop inclui também o Hadoop Distributed File System (HDFS) que é um sistema de arquivo distribuído integrado ao Hadoop. É projetado para o armazenamento de arquivos grandes, com padrões de acesso instantâneo a dados, executados em *clusters* de computadores. Apresenta portabilidade entre sistemas operacionais. O HDFS armazena os arquivos em blocos de 64MB por unidade. Os arquivos possuem várias réplicas, armazenadas como unidades independentes, facilitando o processamento em paralelo (WHITE, 2012).

A Figura 4 mostra um modelo simplificado da arquitetura em um *cluster* HDFS:

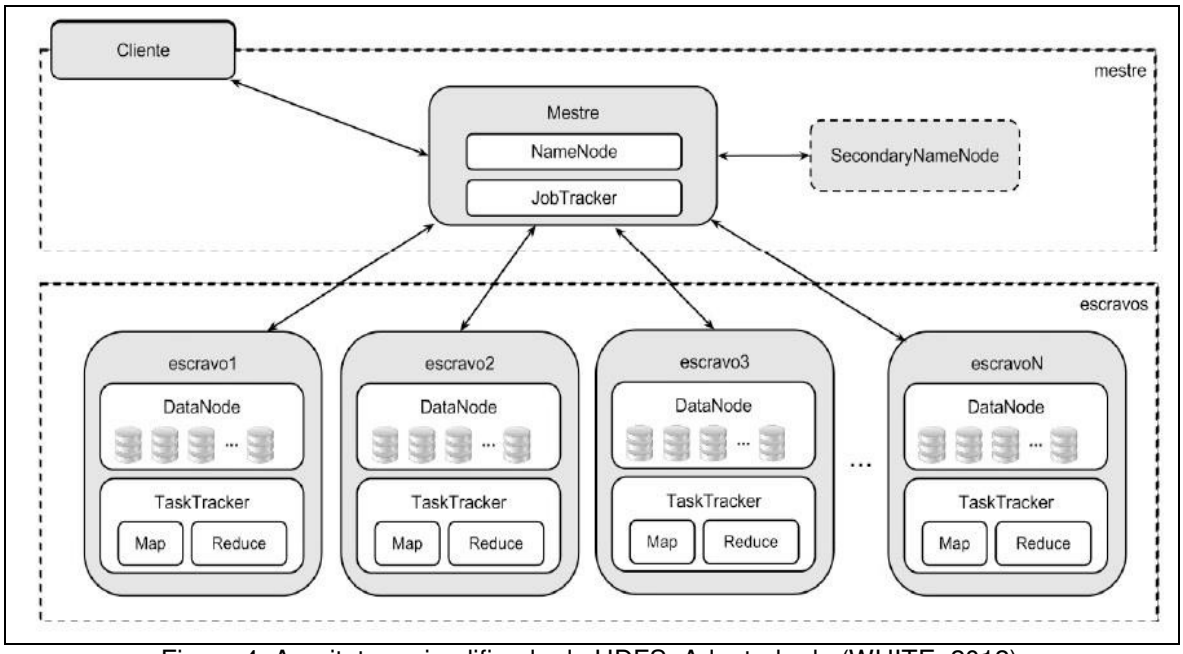

Figura 4: Arquitetura simplificado do HDFS. Adaptado de (WHITE, 2012).

O HDFS possui uma arquitetura do tipo mestre/escravo. Um cluster HDFS apresenta dois tipos de nós, são eles: um *namenode* (nó de nome) e múltiplos *datanodes* (nós de dados). O *namenode* gerencia o *namespace* do sistema de arquivo, administra todos os arquivos e diretórios, mapeia os arquivos e blocos que estão sendo utilizados. Um cliente acessa o sistema de arquivos através da comunicação com *namenodes* e *datanodes.* Os datanodes armazenam e recuperam os blocos quando eles são solicitados pelo usuário ou pelo *namenode*. Apresentam periodicamente um relatório ao *namenode* com as listas de blocos que estão armazenando dados.

Sem o *namenode* o sistema de arquivo não funcionará. Se os *namenodes* em execução forem apagados, todos os arquivos no sistema de arquivos estariam perdidos, pois não haveria outra maneira de acessar os *datanodes* senão pelos *namenodes.* Em razão disso, o Hadoop prover dois mecanismos de tolerância à falhas, sendo eles: o *back up* de estados persistentes do sistema de arquivos e a utilização de *namenodes* secundários (WHITE, 2012).

O Apache Hadoop mantém mecanismos próprios para autenticação e autorização de usuários.

O sistema de arquivos do Hadoop implementa Criptografia de Dados Transparente (*Transparent Data Encryption - TDE*), que possibilita que os arquivos

físicos, como os arquivos de *log*, de dados e de *backup*, sejam protegidos por uma chave que é utilizada para criptografar os dados junto com o certificado. Quando configurada da forma descrita anteriormente, a criptografia dos dados não requer alterações ao código do aplicativo do usuário. Essa criptografia é também *end-to-end*, significa que os dados só podem ser criptografados e descriptografados pelo cliente, o HDFS nunca armazena ou tem acesso às chaves de criptografia de dados. Tal característica satisfaz dois requisitos típicos para criptografia: criptografia em repouso (protege dados em mídia persistente, como um disco) e criptografia em trânsito (protege dados que trafegam pela rede) (APACHE HADOOP, 2015).

Com o módulo YARN sendo eixo central do projeto Apache Hadoop, novos pacotes de funcionalidades podem funcionar dentro da plataforma de dados HDFS.

Um exemplo de *engine* é o Apache Knox Gateway (ou Apache Knox) que promove segurança a um *cluster* Hadoop. O Knox oferece um ponto único de autenticação e acesso para serviços no Apache Hadoop, o objetivo é simplificar a segurança para os usuários que acessam os dados e executam trabalhos, e para operadores que controlam o acesso ao *cluster* (APACHE HADDOP, 2015).

Outra *engine* associada ao monitoramento de ações no Hadoop é o Apache Ambari, que é sistema de administração e monitoramento. Fornece ferramentas para simplificar o gerenciamento do *cluster* por meio de uma coleção de ferramentas e API's que mascaram a complexidade do Hadoop (APACHE HADOOP, 2015).

Várias empresas e organizações usam Hadoop tanto para pesquisa quanto para produção. Alguns exemplos são mostrados na Tabela 4.

Tabela 4: Exemplos de usuários Hadoop. Fonte: adaptado de (APACHE HADOOP, 2015).

Organização	**Forma de utilização**
Amazon	Construção de índices de pesquisa de produtos. Processamento de milhões de sessões diariamente para análise. Clusters que variam de 1 a 100 nós.
Facebook	Armazenamento de cópias de log interno e fontes de dados usados para relatórios, análise e aprendizagem de máquina.
Last.fm	Utilizado para o cálculo gráficos, registro de *royalties*, análise de *log* e fusão em bancos de dados.
Spotify	Empregado para geração de conteúdo, agregação de dados, relatórios e análises.
Yahoo	Usado para apoio em pesquisas Web. Mantêm *cluster* de até 4500 nós.

2.4.3 Apache Storm

É um sistema de computação distribuído para análise de dados em tempo real, pode ser utilizado em aplicações com aprendizagem de máquina, chamada de procedimento remoto, processos de ETL (Extração, Transformação e Leitura), entre outros. Todo o trabalho em Storm pode ser realizado em qualquer linguagem de programação, é compatível com os principais sistemas operacionais (APACHE STORM, 2015).

O Storm é acompanhado de módulos que estão incluídos na distribuição Apache Storm, eles não são necessários para o pleno funcionamento da ferramenta, mas são úteis para adicionar funcionalidades e extensões. Os principais módulos são mostrados na Tabela 5:

Tabela 5: Módulos agregáveis ao Storm. Fonte: adaptado de (APACHE STORM, 2015).

Módulos	Funcionalidades
HDFS	Sistema distribuído de arquivos integrado ao Hadoop
Apache HBase	Ferramenta para acesso aos dados não estruturados.
Apache Hive	Ferramenta para acesso aos dados estruturados.
Apache Solr	Plataforma para pesquisas de dados armazenados no HDFS. Permite a pesquisa de dados estruturados, de texto, de localização geográfica ou de dados de sensor.
Apache Kafka	É um sistema de inscrição e publicação de mensagens rápido e escalável, pode substituir serviços tradicionais de mensagens como JMS (Java Message Service) e AMQP (Advanced Message Queuing Protocol) pois promete melhor conexão, replicação e tolerância a erros.
Redis	Banco de dados NoSQL.

Conforme informações presentes na Tabela 5, o Storm pode trabalhar com o armazenamento de dados pelo HDFS. Os dados podem ser estruturados, acessados através das ferramentas Apache Hive, ou podem ser dados não estruturados acessados pela Apache HBase ou pelo Redis, a depender da aplicação.

O Storm manipula e transforma *streams* de dados, que podem conter objetos de qualquer tipo. *Stream* é a abstração fundamental para o Storm, é representada por uma sequência ilimitada de tuplas que são criadas e processadas em paralelo de forma distribuída. Por padrão, as tuplas contêm inteiros, *longs*, *shorts*, *bytes*, *strings*, *doubles*, *floats*, *booleans* e matrizes. Mas as tuplas podem conter objetos de qualquer tipo, os tipos de dados personalizados devem ser registrados no Serializador do Storm (APACHE STORM, 2015).

Em um *cluster Storm* operam 3 (três) tipos de nós: *Nimbus* (nó mestre), *Zookeeper* e *Supervisor*. Os nós são organizados em função de um nó mestre que trabalha continuamente. O nó mestre é responsável por distribuir o trabalho no *cluster*, delegar tarefas aos demais nós e monitorar a ocorrência de falhas. Os componentes básicos do *Storm* estão ilustrados na Figura 5:

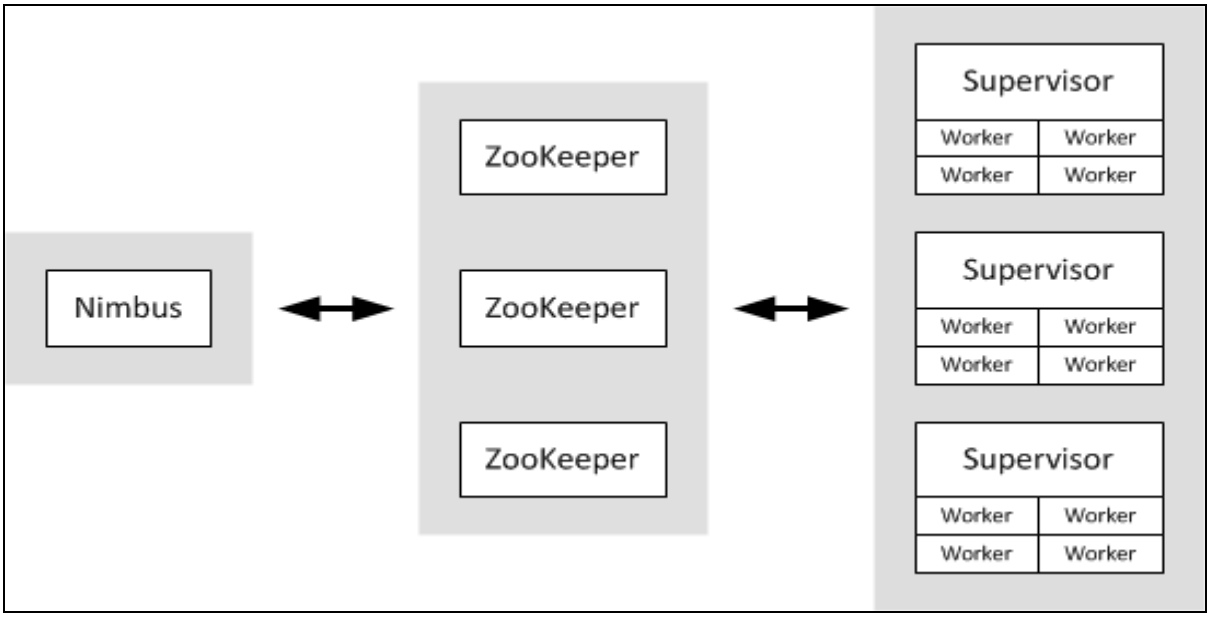

Figura 5: Componentes de um *cluster Storm*. Adaptado de (LEIBIUSKY *et al*, 2012).

Os nós do tipo *Zookeeper* coordenam o trabalho no cluster *Storm*. *Zookeeper* é uma implementação Apache que viabiliza coordenação entre processos distribuídos, fornece também serviços de sincronização e registro de nomes (*naming registry*) para sistemas distribuídos. Já os nós do tipo *Supervisor* são responsáveis por executar as tarefas, eles iniciam e encerram os processos conforme ordem emitida pelo *Nimbus*, a comunicação com o *Nimbus* é feita por intermédio do *Zookeeper* (APACHE STORM, 2015).

O projeto Storm é projetado para ser tolerante a falhas. Dentro de cada nó *Supervisor* existem entidades denominadas *Workers,* cada *worker* é responsável

pela execução dos processos no *cluster*. Quando nós do tipo *worker* morrem, o *Supervisor* reiniciará o *worker* automaticamente. Se ocorrer falha na inicialização, o Nimbus atribuirá outro *worker* como substituto.

As estruturas *Nimbus* e *Supervisors*, também são projetados para reagirem rapidamente em situações de falhas. Todos os estados são mantidos no Zookeeper ou no disco, se *Nimbus* ou *Supervisor* morrem, eles reiniciarão como se nada tivesse acontecido. Portanto, os processos de trabalho não são afetados pela morte de *Nimbus* ou *Supervisor*. Em contraste com o Hadoop, onde se um *JobTracker* morre, todas as tarefas em execução são perdidas (APACHE STORM, 2015).

Dentre as organizações que utilizam o Storm, alguns exemplos são citados na Tabela 6.

Tabela 6: Exemplos de usuários Apache Storm. Fonte: adaptado de (APACHE STORM, 2015).

Organização	**Forma de utilização**
Alibaba	Usa Storm para processar o *log* da aplicação e da mudança de dados no *database* para fornecer estatísticas em tempo real para aplicativos de dados.
Baidu	Usado para processar *logs* de busca e para fornecer estatísticas em tempo real.
Digital Sandbox	Utilizado para monitorar e para extrair dados de fontes estruturadas ou não. Viabiliza o sistema para recuperação de informação.
Spotify	Provê vários recursos de tempo real incluindo recomendação de música, monitoramento, análise e segmentação de anúncios.
Yahoo	Enquanto o Hadoop é a tecnologia para processamento em lote, o Storm habilita o processamento de *streams* de dados, de eventos do usuário e *log* de aplicativos.

2.4.4 Apache Drill

É uma ferramenta para pesquisa e exploração em *Big Data*, projetada para dá suporte em análises de alta performance em grandes conjuntos de dados, estruturados ou não (APACHE DRILL, 2014).

Drill suporta uma variedade de bancos de dados NoSQL e sistemas de arquivos, incluindo HBase, MongoDB, MapR-DB, HDFS, MAPR-FS, Amazon S3, Azure Blob Storage, Google Cloud Storage, Swift, NAS e arquivos locais. Uma

consulta pode reunir dados de diversas fontes. É compatível com sistemas Linux, MacOS e Windons (APACHE DRILL, 2014).

A conexão das fontes de dados ao Drill é feita por um *plugin* de armazenamento, que é um módulo de *software* interno ao projeto Drill. Tal *plugin* propõe a otimização na execução de consultas, oferece a localização dos dados e configuração da área de trabalho e dos arquivos para leitura de dados. As fontes de dados podem ser: um banco de dados, sistema de arquivo local ou distribuído (APACHE DRILL, 2014).

A Figura 6 mostra a camada do *plugin* conectando Drill às fontes de dados:

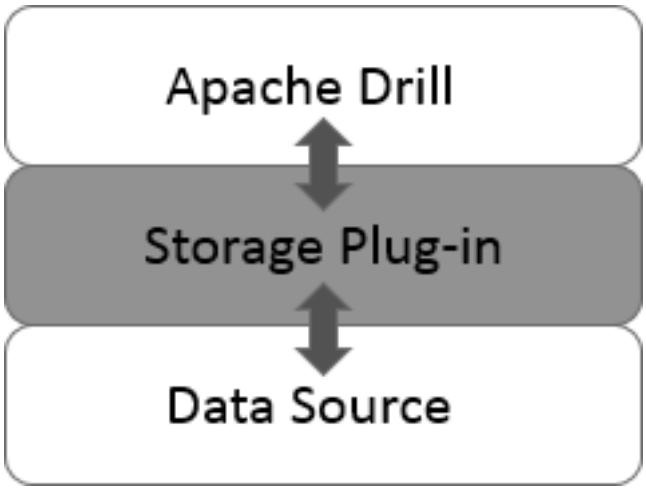

Figura 6: Plugin conectando Drill às fontes de dados. Fonte: (APACHE DRILL, 2014).

O projeto Drill disponibiliza *plugin* de armazenamento para HDFS, HBase, Hive, MongoDB, Amazon S3 e para RDBMS (Relational Database Management System) que permite a conexão do Drill com bancos de dados tradicionais.

Mecanismos de consulta tradicionais exigem intervenção significativa de alguns procedimentos antes que os dados podem ser consultados. Com Drill os usuários podem consultar apenas os dados brutos reduzindo a sobrecarga no sistema. Não é necessário criar e manter esquemas, ou carregar e transformar os dados antes que eles possam ser processados. No Drill, basta incluir na consulta o caminho para um diretório, por exemplo Hadoop, coleção MongoDB ou S3 (APACHE, 2014).

O projeto Apache Drill é desenvolvido em linguagem Java. Para consultar dados estruturados é utilizado o padrão SQL e para consultar arquivos de texto, entre outros formatos, são utilizados JSON (Notação de Objeto *JavaScript*) e Apache Parquet (formato de armazenamento colunar independente da escolha da

estrutura de processamento de dados, modelo de dados ou linguagem de programação).

Drill fornece extensões para SQL, seus usuários podem usar ferramentas de *Business Intelligence* e análise de dados, tais como Tableau, Qlik, MicroStrategy, Spotfire, SAS e Excel para interagir com armazenamentos de dados não-relacionais, utilizando o *driver* JDBC já incluso no Drill (APACHE DRILL, 2014).

A partir da notação JSON, o Drill permite consultas sobre dados complexos em aplicações e armazenamento de dados não-relacionais, conforme a Figura 7:

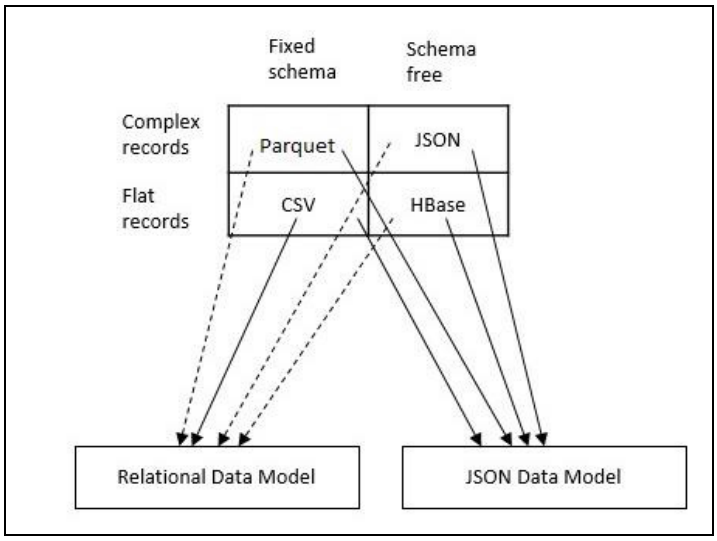

Figura 7: Formato de dados em Drill. Fonte: adaptado de (APACHE, 2014).

A Figura 7 mostra a flexibilidade na qual os dados são tratados: dados semi-estruturados podem ser transformados para serem tratados como modelo relacional e todos os dados (independentemente do tipo de estrutura) podem ser representados no formato JSON (APACHE DRILL, 2014).

A arquitetura alto nível Drill inclui um ambiente de execução distribuído para atender às demandas do processamento em larga escala. O componente central da arquitetura é denominado DrillBit, que é responsável por receber as requisições dos clientes, processar as consultas (*queries*) e entregar os resultados aos clientes (APACHE DRILL, 2014).

O serviço de Drillbit executado em cada nó de dados em um *cluster*, pode maximizar localidade de dados durante a execução da consulta, sem mover dados através da rede ou entre os nós. O ZooKeeper é utilizado para manter a coordenação entre os processos (APACHE DRILL, 2014).

A Figura 8 mostra o fluxo de uma consulta em Drill.

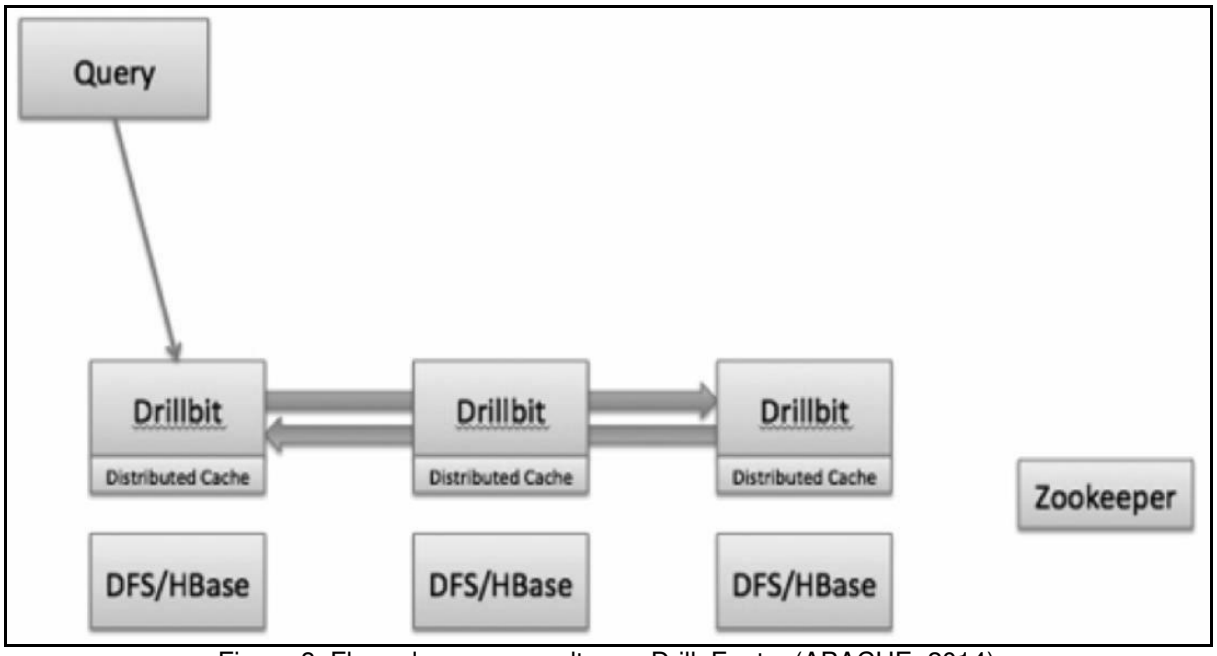

Figura 8: Fluxo de uma consulta em Drill. Fonte: (APACHE, 2014).

O fluxo de uma consulta em Drill segue alguns passos:

- O cliente Drill formula a consulta e faz a requisição utilizando de alguma ferramenta pelo *driver* JDBC ou interface por linha de comando. Um Drillbit pode aceitar a requisição do cliente, não há aplicação do conceito mestre-escravo;
- O Drillbit analisa a consulta e gera um plano de consulta distribuído que será otimizado para tornar a execução mais rápida e eficiente;
- O Drillbit que aceitar a consulta torna-se o nó condução para requisição. Ele recebe a lista de nós Drillbit disponíveis no *cluster* a partir do Zookeeper. O nó condução determina os nós apropriados para executar vários fragmentos do plano de consulta, a fim de maximizar a localidade de dados;
- O Drillbit agenda a execução de fragmentos de consulta em nós individuais de acordo com o plano de execução;
- Os nós individuais finalizam suas execuções e retornam os dados ao nó Drillbit de condução;
- O nó de condução encaminha o resultado da execução ao solicitante.

Os componentes internos de um Drillbit são mostrados na Figura 9:

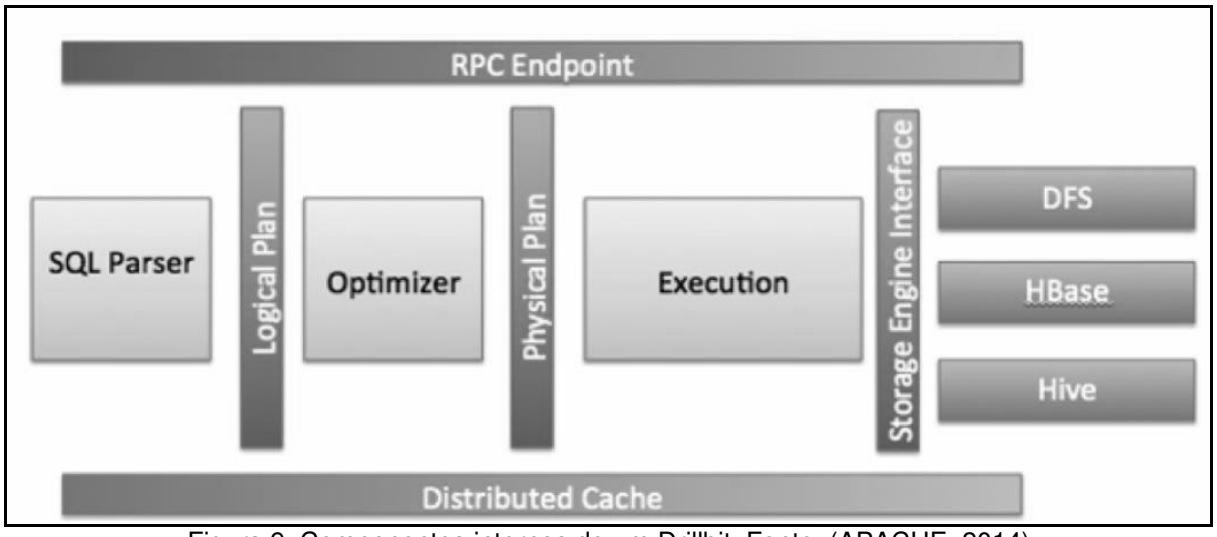

Figura 9: Componentes internos de um Drillbit. Fonte: (APACHE, 2014).

Cada componente de um Drillbit é encarregado de uma tarefa específica:

- RPC *Endpoint*: responsável por manter a comunicação com os clientes baseado em Chamada de Procedimento Remoto (*Remote Procedure Call* - RPC). Os clientes podem se comunicar com um Drillbit diretamente ou através do Zookeeper, para descobrir quais nós Drillbits estão disponíveis antes da submissão de consultas. É recomendado que a comunicação seja feita através do Zookeeper, pois protege os clientes de problemas com o gerenciamento do *cluster*, como exemplo a adição e remoção de nós;
- SQL Parser: o Drill utiliza uma ferramenta *open source*, Optiq, para analisar as consultas de entrada. A saída do Parser é um componente independente de linguagem, compatível com o Plano Lógico que representa as consultas;
- Optimizer: executa várias otimizações de bancos de dados padrão como regra de base/custo com base, bem como localização de dados e outras regras de otimização para reescrever e dividir a consulta. A saída do Otimizador é o Plano Físico de consulta, que representa o mais rápido e eficiente meio para executar as consultas através de diferentes nós no *cluster*;
- Execution Engine: o Drill executa Processamento Massivamente Paralelo (Massively Parallel Processing - MPP), único computador com muitos processadores, para executar consultas distribuídas através de vários nós no *cluster*;
- Storage Plugin Interfaces: o Apache Drill opera como uma camada de consulta em cima de várias fontes de dados e os *plugins* de armazenamneto

representam as abstrações que o Drill utiliza para interagir com as fontes de dados, como exemplo: Sistema Distribuído de Arquivos, HBase e Hive;

- Distributed Cache: Sistema de Cache Distribuído é usado para gerenciar metadados e configurar informações nos nós. Dentre as informações guardadas em cache estão: fragmentos do plano de consultas, estado intermediário de execução de consultas e dados estatísticos (APACHE DRILL, 2014).

O Apache Drill fornece ainda mecanismos para configuração e autenticação de usuários. A autenticação é baseada em nome de usuário e senha através do Módulo de Autenticação Conectável Linux (PAM). A opção de autenticação está disponível através de interface JDBC do Apache Drill (APACHE, 2014).

As principais formas de utilização do Apache Drill são mostrados por demanda dos usuários para solução de problemas específicos, conforme Tabela 7.

Tabela 7: Principais formas de utilização do Drill. Fonte: adaptado de (BANDUGULA, 2015).

Forma de utilização	Descrição
Aumento de dados em usuários Hadoop	Promove a redução de custos no tratamento de dados, otimizando cargas de trabalho de clientes que já utilizam Hadoop. Fornece aos usuários a capacidade de explorar de forma interativa dados armazenados no Hadoop.
Exploração de dados no HBase e MapR-DB	Permite o acesso a bancos HBase e MapR-DB (ambos bancos NoSQL) usando linguagem SQL.
Tratamento de dados brutos com facilidade	A partir da utilização da notação JSON é possível explorar dados de diversos formatos. A intenção é reduzir as atividades para preparação de dados.

2.4.5 RapidMiner

É uma plataforma utilizada para mineração de dados, aprendizagem de máquina e análise preditiva. Pode ser aplicada em qualquer processo de negócio, como uma solução para reduzir o tempo para descobrir oportunidades e riscos em um conjunto grande de dados (RAPIDMINER, 2015).

Trata-se de um composto de soluções em *software* que dispõem de vários produtos, entre eles: RapidMiner Studio, Server, Cloud e Radoop. A tecnologia RapidMiner inclui o processo de ETL (Extração, Transformação e Leitura), integração de dados, análise e geração de relatórios em uma única solução. Todo trabalho do RapidMiner pode ser executado nos principais sistemas operacionais.

O RapidMiner Studio é o ambiente de desenvolvimento, que capacita analistas para projetar a análise preditiva a partir da modelagem de implantação. Não é necessário programar, a modelagem é obtida através de processos simples de “arrastar e soltar”, embora os usuários podem incorporar códigos em R, Python e scripts SQL. Há ainda geração de gráficos automáticos (RAPIDMINER, 2015).

O RapidMiner Studio trabalha com grandes fontes de dados, operando com armazenamento em memória, em base de dados, em *streams*, em núvem e análise em Hadoop. O carregamento de dados pode ser feito por diversas fontes incluindo Excel, Access, Oracle, IBM DB2, Microsoft SQL, Netezza, Teradata, MySQL, Postgres, Salesforce.com entre outros. E ainda, é compatível com as principais plataformas e sistemas operacionais (RAPIDMINER, 2015).

Outra solução é o RapidMiner Server que atua como servidor para sustentação dos demais módulos do projeto. O servidor RapidMiner integra o RapidMiner Studio às fontes de dados em toda a empresa. Executa e monitora processos de análise remotamente, permite a execução da análise e obtenção de resultados em tempo real. Através de repositórios compartilhados e dos aplicativos interativos para visualização de resultados, o RapidMiner Server viabiliza um ambiente de colaboração entre os gestores (RAPIDMINER, 2015).

Outra solução para o armazenamento é o RapidMiner Cloud, um ambiente de computação de alto desempenho para as análises preditivas em larga escala. Permite a expansão das fontes de dados através de operadores que conectam os processos em utilização com o Amazon S3, Twitter, Salesforce, DropBox, entre outros (RAPIDMINER, 2015).

A capacidade disponível para o armazenamento dependerá do tipo de licença adquirida, que pode variar conforme Tabela 8.

Tabela 8: Características RapidMiner Cloud. Fonte: adaptado de (RAPIDMINER, 2015).

Característica	RapidMiner Studio Community	RapideMiner Studio Professional
Custo	Livre	Livre
Tamanho do Repositório	20MB	5GB
Tamanho do Hardware	8GB	16GB, 32GB e 64GB

Por último o RapidMiner Radoop, componente ofertado apenas na versão comercial. Trata-se da integração da tecnologia Apache Hadoop ao projeto RapidMiner. Por meio da inclusão da tecnologia Hadoop, outras ferramentas Apache podem ser acopladas como: Hive, Pig, MapReduce, Impala, Spark e Mahout. Apache Impala é um banco de dados SQL para aplicações com Hadoop, Apache Spark é uma *engine* para processamento de dados em larga escala, fornece um ambiente de programação compatível com linguagens como Java, Python e R, e Apache Mahout é uma biblioteca de algoritmos de aprendizado de máquina.

A proposta do Radoop, como componente central da plataforma RapidMiner, é de ocultar a complexidade no tratamento das possíveis ferramentas adicionadas ao Hadoop. Assim, o usuário concentra-se apenas nas análises dos dados. A Figura 10 mostra com mais detalhes o funcionamento do Radoop.

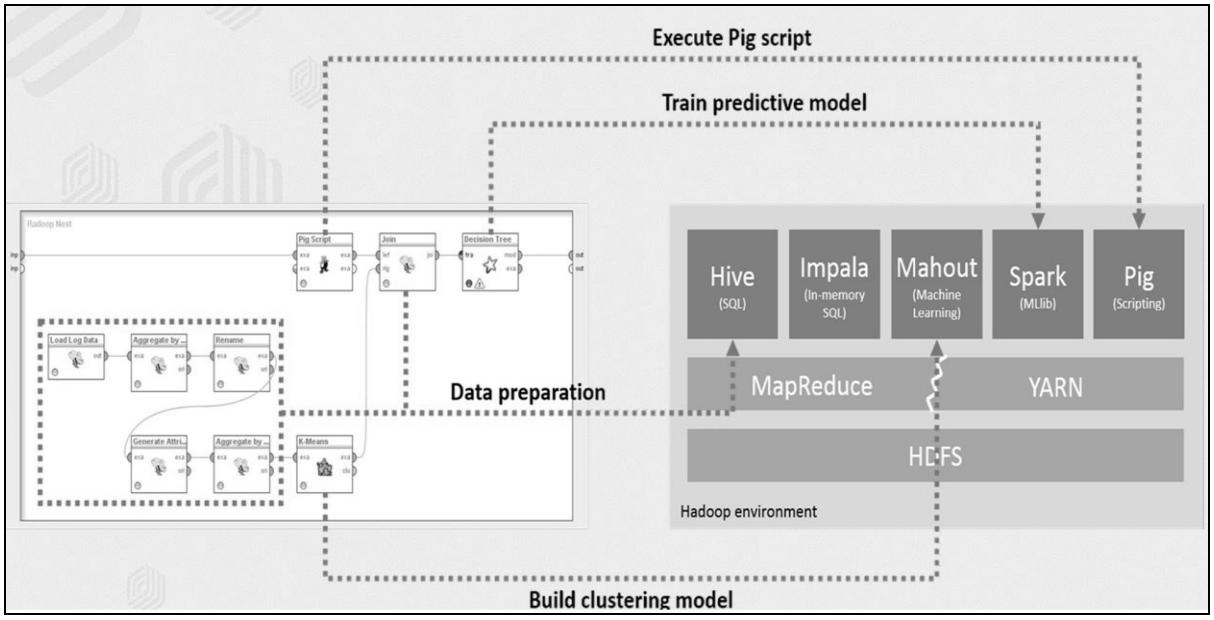

Figura 10: Fluxo de trabalho criado no RapidMiner Studio. Fonte: adaptado de (RAPIDMINER, 2015).

A Figura 10 mostra um exemplo de um fluxo de trabalho de análise preditiva criado em RapidMiner Studio. RapidMiner Radoop traduz cada etapa do fluxo de

trabalho em muitas línguas do Hadoop, executa *scripts* através do Pig, treina o modelo pelo Spark, prepara os dados pelo Hive e construe modelos de *cluster* como o Mahout.

RapidMiner Radoop fornece ainda autenticação de usuários por meio de integração ao protocolo Kerberos. A Figura 11 mostra como ocorre a autenticação de um usuário.

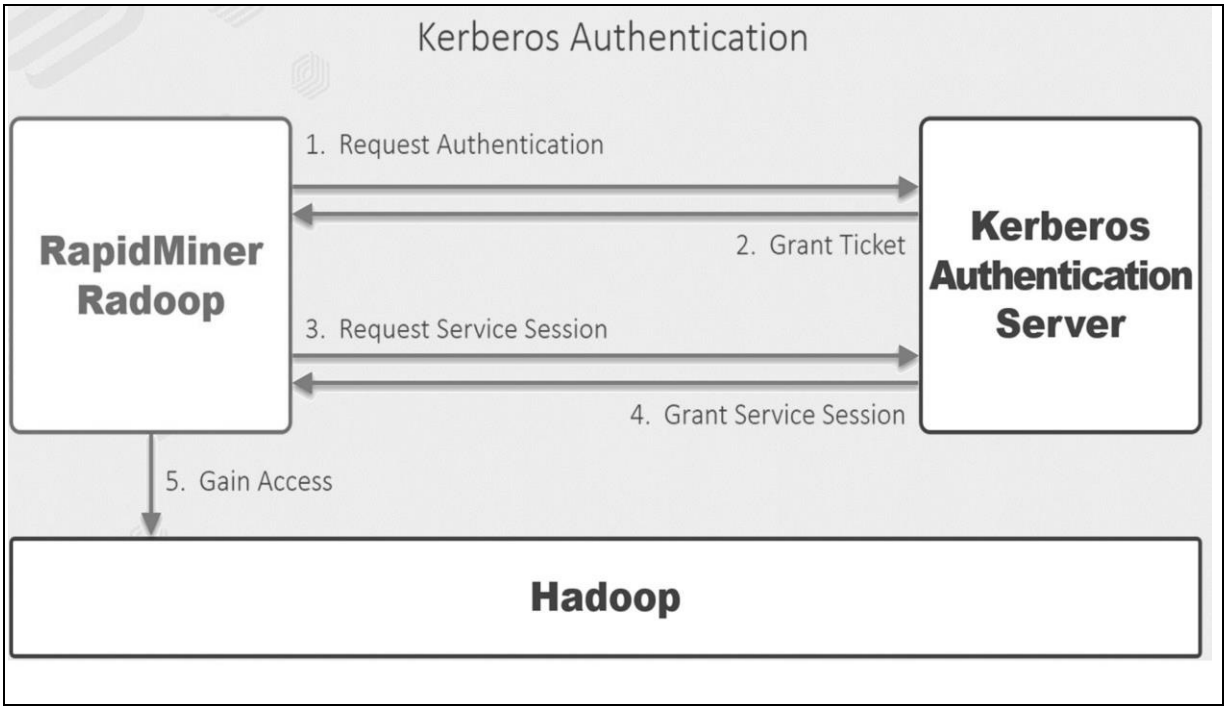

Figura 11: Autenticação de usuário no RapidMiner. Fonte: adaptado de (RAPIDMINER, 2015).

O RapidMiner Radoop solicita uma Requisição de Autenticação ao servidor Kerberos quando um usuário tenta acessar um *cluster* Hadoop. Quando o servidor Kerberos atende à requisição, é concedido um *Ticket*, que será usado como uma passagem para ter acesso ao RapidMiner Radoop. Para ter acesso aos serviços, é feita uma Requisição de Serviço ao servidor Kerberos, se concedido o acesso, o usuário poderá acessar serviços no Hadoop. Para confirmar as informações do usuário, Kerberos utiliza o protocolo LDAP (*Lightweight Directory Access Protocol*), protocolo de acesso aos diretórios, que permite acessar as informações sobre os usuários de uma rede através de protocolos TCP/IP.

Além de autenticação, o RapidMiner Radoop também suporta autorização de acesso a dados utilizando Apache Sentry e Apache Ranger (RAPIDMINER, 2015).

Existem mais de 250.000 usuários ativos de RapidMiner, dentre as organizações que utilizam essa ferramenta, citamos alguns exemplos na Tabela 9.

Tabela 9: Exemplos de usuários RapidMiner. Fonte: adaptado de (RAPIDMINER, 2015).

Organização	Forma de utilização
PayPal	Usa RapidMiner para análise de sentimento de comentários de clientes e feedback de todo o mundo.
SustainHub	Usado para processar mineração de dados e fazer análise de risco em cadeias de suprimentos
George Washington University	Utiliza RapidMiner para gestão de Políticas Públicas

2.4.6 Apache Hortonworks

Hortonworks Data Plataforma (HDP) é uma plataforma para gerenciamento de dados administrada pela Apache Software Foundation, que utiliza a implementação do Hadoop em seus processos. Desenvolvido em linguagem Java. O HDP apresenta um conjunto abrangente de recursos alinhados com as seguintes áreas funcionais: gestão de dados, acesso a dados, governança e integração de dados, segurança e operações (APACHE HORTONWORKS, 2014).

A Figura 12 ilustra o conjunto de recursos da plataforma Hortonworks:

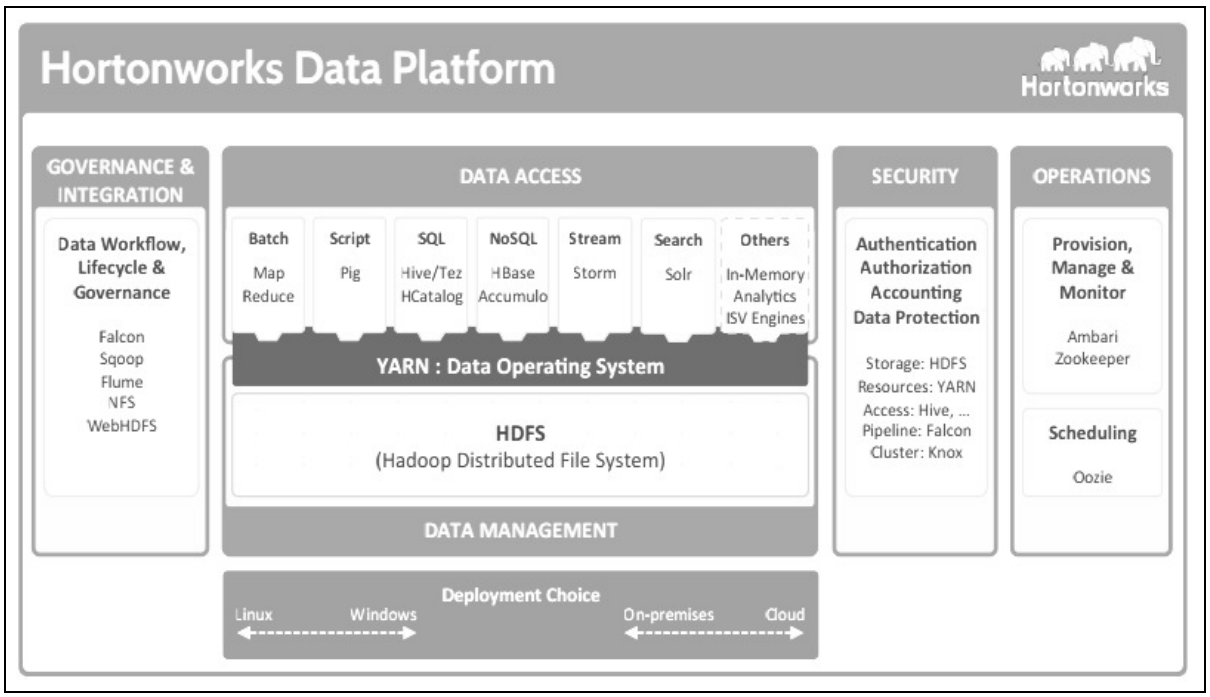

Figura 12: Plataforma Hortonworks. Fonte: (APACHE HORTONWORKS, 2014)

Para o Gerenciamento de Dados (*Data Management*) o HDP utiliza para armazenar e processar dados consecutivamente: o Sistema Hadoop Distributed File (HDFS) e Apache Hadoop YARN. O HDFS é o sistema de arquivos do Hadoop, projetado para computação distribuída apropriado para grandes conjuntos de dados. Já o YARN é um sub-projeto do projeto maior Apache Hadoop, oferece o gerenciamento de recursos e permitem que uma variedade de métodos de acesso a dados seja plugável ao HDP.

O Acesso aos Dados (*Data Access*) abrange uma variedade de motores para admissão e processamento de dados. Através do YARN é possível interagir com os dados provenientes de múltiplas fontes sem a necessidade de acessar individualmente cada *cluster*. Portanto, a arquitetura da plataforma Hortoworks baseada em YARN permite que o maior número possível de métodos de acesso possam coexistir dentro do mesmo *cluster* de dados reduzindo custos (APACHE HORTONWORKS, 2014).

A HDP oferece os tipos de acesso a dados, conforme Tabela 10:

Tabela 10: Tipos de Acesso aos Dados em HDP. Fonte: adaptado de (APACHE HORTONWORKS, 2014).

TIPO DE ACESSO	FERRAMENTA UTILIZADA
Batch	Ou arquivo de lote. É o modo de processamento de dados no qual são processados em grupos, ou lotes, por meio de uma rotina agendada. As funções Map e Reduce são utilizados.
Script	Apache Pig é uma linguagem de *script* para Hadoop que pode ser executado em MapReduce ou Apache Tez, permitindo agregar, juntar e classificar os dados.
SQL	Apache Hive é o padrão para interações SQL em escala *petabyte* dentro Hadoop, oferecendo consulta interativa e em lote através do um amplo conjunto de semântica SQL. Também são usados: Apache HCatalog e Apache Tez.
NoSQL	Apache HBase fornece acesso rápido aos dados como um formato de colunas, banco de dados NoSQL. Apache Accumulo também fornece armazenamento de alto desempenho e recuperação, mas com controle de acesso de granularidade fina ao dado.
Stream	Apache Storm processa fluxos de dados em tempo real e pode analisar e agir sobre os dados à medida que eles fluem no sistema HDFS.
Search	Apache Solr integrado à HDP fornece alta velocidade indexação e tempo reduzido para pesquisa em dados no HDFS.

Conforme informações presentes na Tabela 10, o HDP pode trabalhar com o armazenamento de dados pelo HDFS. Os dados podem ser estruturados, acessados através das ferramentas Apache Hive, HCatalog e Tez, ou podem ser dados não estruturados acessados pela Apache HBase e Accumulo, a depender da aplicação.

Ainda é possível a utilização de outros tipos de acesso aos dados por meio de ISV's (Independents Software Vendor) que são fornecedores independentes de *softwares* compatíveis em uma ou mais plataformas de *hardware* ou sistema operacional.

Para Governança de Dados e Integração (*Data Governance and Integration*), a plataforma Hortonworks amplia o acesso e gestão de dados com ferramentas para suporte. As principais ferramentas são: Apache Falcon que é um *framework* para simplificar gestão de dados e de processamento de *pipeline*; Apache Sqoop que transfere dados em massa entre Hadoop e os armazenamentos de dados estruturados tais como Teradata, Netezza, Oracle, MySQL, Postgres e HSQLDB; e Apache Flume é usado para transmitir dados de várias fontes em Hadoop para análise (APACHE HORTONWORKS, 2014).

Em relação à Segurança (*Security*), a HDP oferece uma abordagem centralizada para o gerenciamento o que permite criar e administrar uma política de segurança central. A plataforma Hortonworks fornece um conjunto abrangente de recursos para autenticação, autorização, auditoria e proteção de dados.

As Operações de Cluster (*Cluster Operations*) são mantidas como um conjunto de capacidades operacionais para dispor, gerenciar, monitorar e operar dados. Ferramentas são fornecidas para execução do trabalho no *cluster* Hadoop, são elas: Apache Ambari que é um *framework* para disposição, gerenciamento e monitoramento; Apache ZooKeeper que fornece serviços de configuração em aplicações distribuídas; e Apache Oozie com capacidade para organizar e agendar tarefas em todos os pontos de acesso de dados (APACHE HORTONWORKS, 2014).

Na Tabela 11 são mostrados algumas formas de utilização do HDP. Os exemplos são mostrados por área de aplicação e não por empresas que utilizam.

Tabela 11: Áreas de aplicação Hortonworks. Fonte: adaptado de (APACHE HORTONWORKS, 2014).

Áreas de aplicação	Descrição
Segmentação de Clientes	Descobrir uma segmentação natural de clientes em grupos de comportamento semelhante.
Recomendação de Produto	Prever a preferência dos clientes por um determinado produto e recomendar aos clientes produtos semelhantes.
Segurança da Informação	Detectar anomalias no tráfego da rede e identificar potenciais invasores.
Detecção de fraude	Identificar padrões fraudulentos em transações de crédito.
Dados de sensores	Solucionar os desafios do volume e da estrutura em diversas aplicações.

A Hortonworks em parceria com o Hadoop auxiliam muitas empresas no tratamento de grandes conjuntos de dados, existem aplicações em diversas áreas como arquitetura de dados, publicidade, serviços financeiros, aplicações industriais, entre outros.

2.4.7 GridGain

É uma solução em *middleware* para o processamento de conjuntos de dados de grande escala, mantida pela Apache Software Foundation. O projeto GridGain e suas tecnologias começou a ser desenvolvido em 2005 com o primeiro lançamento em 2007 pela empresa GridGain Systems. Desde 2014, o projeto GridGain é licenciado pela Apache, aceito no programa Apache Incubator sob o nome de “Apache Ignite”. O Apache Incubator é o caminho de entrada para os projetos e bases de código que pretendam aderir à Apache (GRIDGAIN, 2015).

A arquitetura GridGain é baseada em JVM (*Java Virtual Machine*), a topologia em um cluster GridGain não exige nós de servidor e cliente separados, todos os nós do cluster são iguais e pode desempenhar qualquer função lógica demandada pela aplicação. É compatível com os principais sistemas operacionais (GRIDGAIN, 2015).

O desenvolvimento do GridGain é baseado na tecnologia In-Memory Data Fabric (IMDF). Computação *In-Memory* é caracterizada por servir-se de alto desempenho utilizando sistemas de memória integrados e distribuídos para realizar

transações em conjuntos de dados de grande escala em tempo real. Portanto, o GridGain IMDF proporciona velocidade e escala ilimitado para o processamento de dados (GRIDGAIN, 2015).

A Figura 13, mostra uma visão geral do funcionamento do GridGain IMDF.

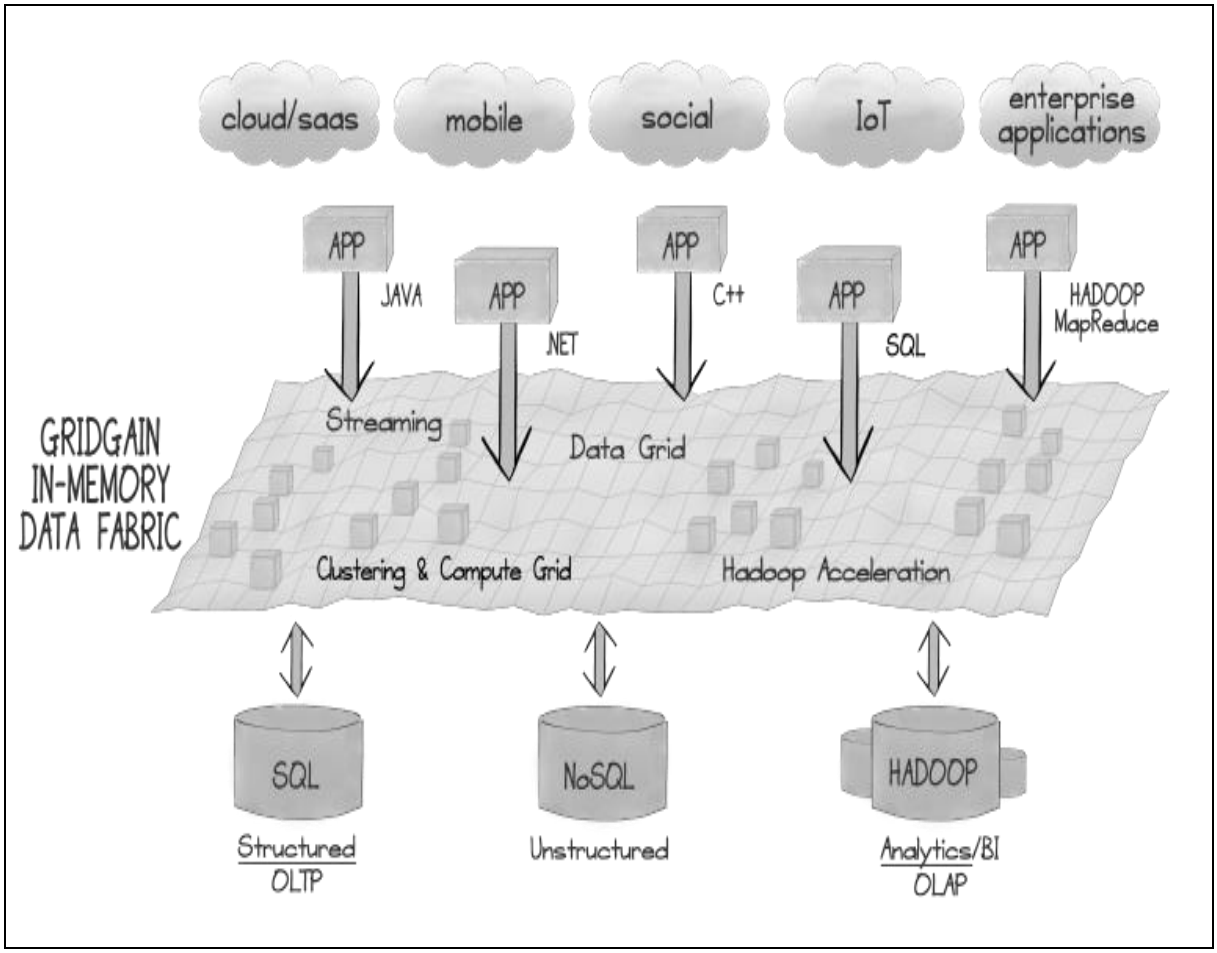

Figura 13: GridGain In-Memory Data Fabric. Fonte: (GRIDGAIN, 2015).

Conforme a Figura 13, dados distribuídos e baseados em nuvem podem ser acessados pelas as aplicações em uso. O GridGain IMDF fornece uma interface de programação unificada que faz a ligação entre os principais tipos de aplicações (Java, .NET, C ++, consultas SQL, Hadoop e MapReduce) com os vários meios para armazenamento de dados estruturados, semi-estruturados e não estruturados (SQL, NoSQL, Hadoop).

A Figura 13 mostra ainda que na camada central do GridGain IMDF estão presentes alguns elementos como: *Streaming*, *Data Grid*, *Compute Grid*, e *Hadoop Acceleration*. Tais elementos são características chave para a compreensão do funcionamento da tecnologia GridGain.

Streaming ou *In-Memory Streaming* é a capacidade de processar endereços de conjuntos de aplicações para as quais os métodos tradicionais de processamento e armazenamento (tais como bancos de dados baseados em disco ou sistemas de arquivos) ficam aquém. A partir do *Streaming* é possível consultar dados de entrada e utilizá-los para responder a perguntas como: "Quais são os 10 produtos mais populares por um determinado intervalo de tempo?", "Qual é a média de preço do produto em uma determinada categoria durante o dia anterior? ", entre outras (GRIDGAIN, 2015).

A Figura 14 mostra um dos objetivos da característica de *Streaming.*

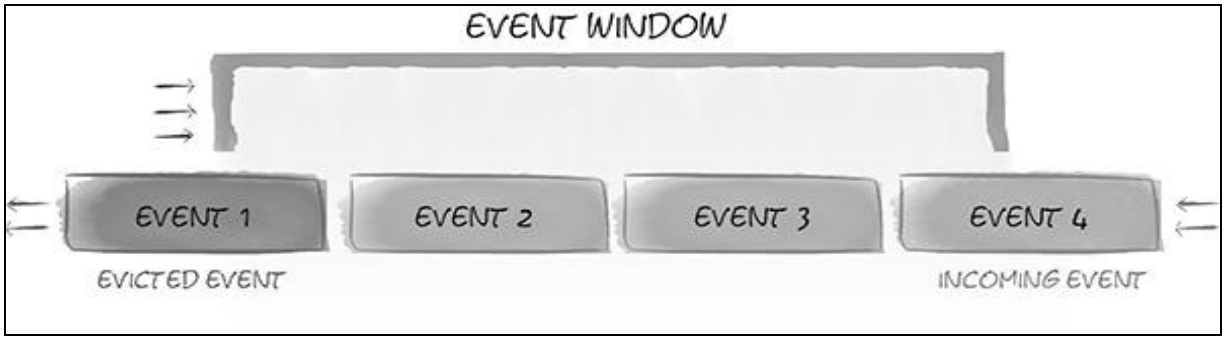

Figura 14: Utilização de In-Memory Streaming. Fonte: (APACHE IGNITE, 2015).

A utilização mais comum para *Streaming* é a capacidade de controlar o fluxo de eventos como um *pipeline*. Como os eventos estão chegando no sistema em taxas elevadas, o processamento de eventos deve ser dividido em múltiplas etapas e posteriormente encaminhados para o *cluster* de processamento (APACHE IGNITE, 2015).

Outra característica presente no processamento GridGain IMDF é o *Data Grid* ou *In-Memory Data Grid*. Trata-se de um dos principais recursos pois permite armazenar dados na memória dentro de *clusters* distribuídos. Trabalha na gestão dos dados distribuídos, coordena transações ACID, balanceamento de carga, suporte SQL, entre outras funções (APACHE IGNITE, 2015).

Além da capacidade de *Data Grid*, o GridGain também inclui *Compute Grid*, recurso que fornece meios para o processamento paralelo de CPU. *Compute Grid* fornece um conjunto de APIs que permitem execução de funções de forma distribuída e processamento de dados em vários computadores no cluster (APACHE IGNITE, 2015).

Outra característica presente no projeto GridGain é o *Hadoop Acceleration*, constituído para otimizar o processamento dos dados. Sua instalação e

funcionamento não requer mudança de código, através da tecnologia "*plug and play*" ("ligar e usar") as configurações são instaladas automaticamente (GRIDGAIN, 2015).

O *Hadoop Acceleration* melhora o funcionamento da tecnologia Hadoop existente por adicionar capacidade de análise em tempo real permitindo o processamento rápido de dados. O sistema de arquivos do GridGain (GridGain File System - GGFS) suporta dois modos de trabalho, como um sistema de arquivo autônomo no *cluster* Hadoop, ou em conjunto com HDFS, que serve como uma camada de cache, conforme a Figura 15:

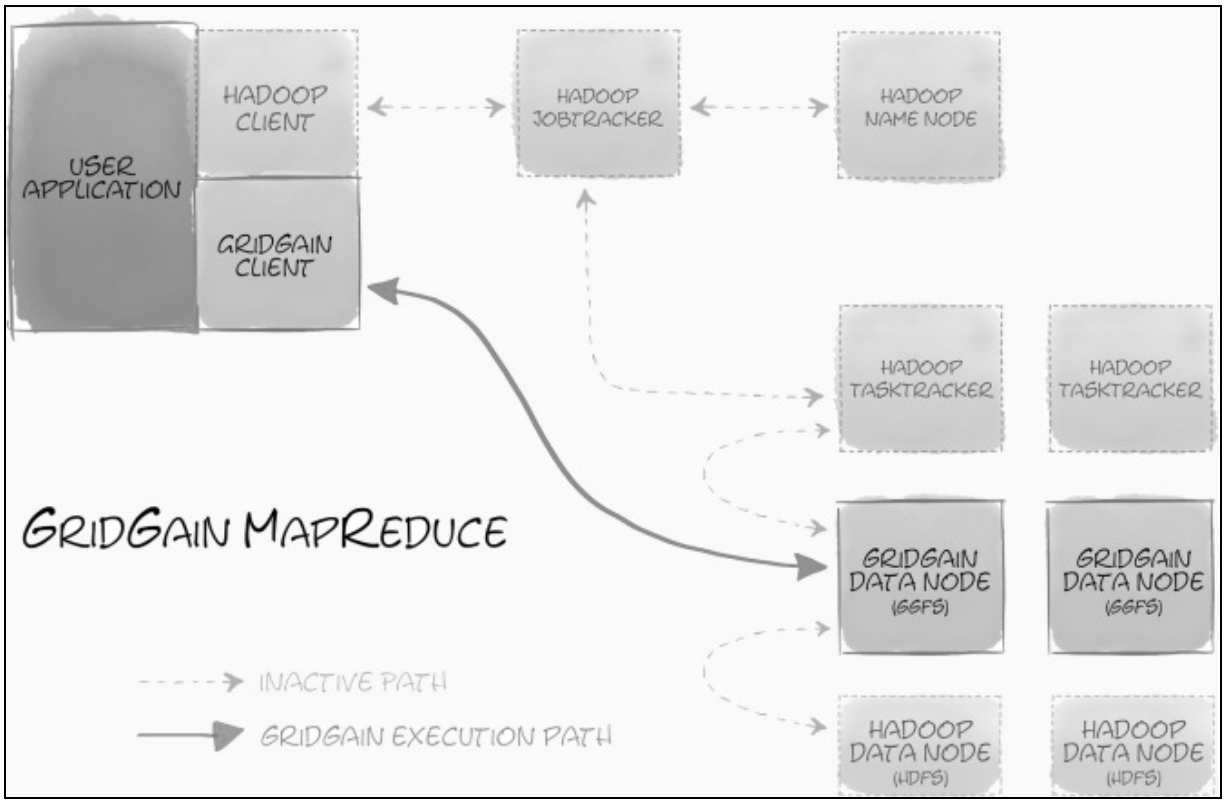

Figura 15: GridGain MapReduce. Fonte: (GRIDGAIN, 2015).

Como uma camada de *cache* fornece a lógica de leitura e escrita, os usuários podem selecionar quais arquivos ou diretórios a serem armazenados em *cache*. Por sua vez, a implementação do MapReduce no GridGain permite aos usuários paralelizar o processamento de dados armazenados na memória GGFS, além de eliminar a sobrecarga associada aos nós executores de tarefas, próprios da arquitetura padrão do Hadoop (GRIDGAIN, 2015).

O projeto GridGain prover funcionalidades de segurança, incluindo recursos de autenticação e autorização, disponível apenas na edição Enterprise. O suporte a autenticação e autorização de nós da rede e clientes depende do API GridSecurity.

Quando GridSecurity é ativado, os nós da rede e clientes remotos devem ser autenticados antes de entrar para o Grid. Para ativar a segurança da rede, credenciais de segurança e autenticação devem ser configuradas. Nós e os clientes remotos podem ser autorizados com permissões especificados para executar ou remover operações (GRIDGAIN, 2015).

Segundo o site do desenvolvedor, a tecnologia GridGain é usada em mais de 500 organizações em seus projetos, dentre alguns clientes que utilizam GridGain em suas aplicações e sistemas citamos: Apple, Canon, Embrapa, Sony, Stanford University, entre outros. Dentre os principais benefícios citados estão a redução do tempo de computação, processamento em tempo real, aumento da acurácia, confiabilidade e escalabilidade.

2.5 Síntese das principais ferramentas *open source* para Análise de Dados em *Big Data*

Diante da exposição das principais ferramentas *open source* para análise de dados em *Big Data,* percebe-se a presença de características comuns entre algumas ferramentas, mas existem alguns atributos que diferenciam as ferramentas em estudo.

As características selecionadas para demonstração das ferramentas são: acesso, tipo, processamento e armazenamento de dados, análise em tempo real e segurança. Cada característica será avaliada somente com base na descrição do desenvolvedor.

A Tabela 12 mostra a síntese das ferramentas em relação às 6 (seis) principais características. No campo onde houver o sinal tipográfico “ * ” (asterisco) significa que, as configurações da característica dependerá da aplicação a ser utilizada.

Tabela 12: Síntese das principais características das ferramentas *open source* para Análise de BD.

Características	Ferramentas						
	Map Reduce	**Apache Hadoop**	**Apache Storm**	**Apache Drill**	**RapidMiner**	**Horton Works**	**GridGain**
Acesso aos dados	*	Apache Hive, Pig, HBase, Accumulo, Tez, entre outros.	Apache Hive, HBase, entre outros.	Através de Sistemas Plugáveis localmente ou em cluster.	Carregamento por diversas fontes.	Múltiplas fontes admitidas através do YARN.	Acessa dados distribuídos e baseados em nuvem.
Tipos de dados	*	Flexível. Dados estruturados ou não. Diversos formatos.	Flexível. Dados estruturados ou não. Diversos formatos.	Flexível. Dados estruturados ou não. Diversos formatos.	Flexível. Dados estruturados ou não. Diversos formatos.	Flexível. Dados estruturados ou não. Diversos formatos.	Flexível. Dados estruturados ou não. Diversos formatos.
Processa-mento dos dados	Aplicação das funções Map e Reduce.	Inclui as funções Map/Reduce e Hadoop Módulo Hadoop YARN.	*Cluster* Storm e Apache Zookeeper.	*Cluster* Drill e Apache Zookeeper.	Tecnologia Hadoop.	Utiliza tecnologias Apache: Hadoop, Zookeeper, Storm entre outras.	Inclui as funções Map/Reduce e Hadoop.
Armazena-mento dos dados	*	Sistema Distribuído de Arquivo (HDFS).	Sistema Distribuído de Arquivo (HDFS), entre outros.	Sistema Distribuído de Arquivo (HDFS) e Sistema de Cache Distribuído	Em memória, base de dados, *stream*, nuvem, entre outros	Sistema Distribuído de Arquivo (HDFS)	Sistema de Arquivo GridGain File System (IMDG) e Sistema Distribuído de Arquivo (HDFS),
Análise em tempo real	*	*	Fornece suporte.	*	Fornece suporte.	Fornece suporte.	Fornece suporte.
Segurança	*	Suporte à autenticação, autorização, contabilidade e proteção de dados.	Não fornece suporte à autentica-ção de usuários.	Fornece configura-ções para autentica-ção de usuários.	Fornece autenticação, autorização e suporte à criptografia.	Política de Segurança Central. Possui recursos de autenticação, auditoria e proteção de dados.	Fornece recursos para autenticação e autorização de usuários

Neste capítulo foi apresentada a fundamentação teórica composta pelo estudo de Big Data e das principais ferramentas *open source* para análise de dados. No próximo capítulo será abordado o estudo de caso, que visa exemplificar a aplicabilidade da análise das ferramentas, feita no capitulo 2.

3 Estudo de Caso

Este capítulo apresenta o estudo de caso para demonstrar a aplicação dos conceitos estudados no presente trabalho, cujo objeto é mostrar, a partir das características e necessidades de uma organização, como pode ocorrer o processo de escolha de uma ferramenta apropriada para o tratamento dos dados, especificamente para efetuar a análise de dados em *Big Data*. As seções a seguir demonstram em detalhes os passos empregados na metodologia proposta.

3.1 Metodologia

A fim de exemplificar a aplicabilidade do estudo feito no capítulo 2, utilizou-se uma metodologia composta de 3 etapas. A primeira diz respeito a descrição do perfil da organização, em seguida tem-se a identificação do problema a ser solucionado, e na última ocorrerá a escolha da ferramenta adequada, conforme a Figura 17.

Figura 16: Metodologia utilizada

Nas próximas seções serão abordadas cada etapa detalhadamente. A primeira descreve o perfil da organização, delineia as principais características da organização em estudo, sua área de atuação, serviços prestados, estrutura organizacional, dentre outras características. Na segunda, um problema é identificado em meio à demanda da organização por encontrar novas formas para o tratamento de dados. A última, traça uma análise para escolha da ferramenta adequada ao problema identificado.

3.1.1 Descrição do Perfil da Organização

A organização selecionada para o presente estudo de caso é uma Secretaria Municipal de Educação (SME), órgão da Administração Pública Direta. Por questões relacionadas a confidencialidade de determinadas informações, será mantido em sigilo o nome do município ao qual pertence tal secretaria.

A SME em estudo está em atuação desde 1910, sendo instituída por lei em 1996. A secretaria é responsável pela gestão das políticas públicas voltadas para a área da educação do município. É atribuição da SME administrar os órgãos e as instituições oficiais do sistema municipal de ensino, integrando-os às políticas e planos educacionais da União e dos Estados.

Está sob responsabilidade da SME a educação nos níveis infantil, fundamental e educação de jovens e adultos (EJA). A rede de escolas possui 170 unidades de ensino, com o total de 93 mil alunos e 8 mil professores. Do total de unidades de ensino, 20% trabalha com algum tipo de atendimento especializado, ou seja, com a educação especial.

Em nível nacional, a educação especial teve seu marco histórico no final do século XIX, com a criação do Instituto dos Meninos Cegos, em 1854, e o Instituto dos Surdos-Mudos, em 1857. Atualmente, o Brasil conta com diversas legislações que asseguram e regulamentam a educação especial no país (MENDES, 2010).

No âmbito da SME em estudo, a educação especial é trabalhada nos níveis da educação infantil e fundamental, com a admissão de alunos com deficiência auditiva e deficiência intelectual. A admissão é feita de 2 formas: em salas especiais e em salas regulares através da inclusão. As salas especiais são multiseriadas, são trabalhadas simultaneamente séries que variam do 1º ao 4º, com alunos de diversas idades na mesma classe. A partir do 5º ano, os alunos deixam de frequentar as salas especiais e passam a frequentar as salas regulares na condição de alunos inclusos.

Ao longo dos anos, a SME tem desenvolvido novas competências para aprimorar os serviços prestados. Desde 2015, as escolas da rede tem adotado técnicas de ensino híbrido, passando a utilizar técnicas de ensino presencial em conjunto às técnicas de ensino *online.*

Ensino híbrido, do inglês *blended learning,* pode ser entendido como um método de ensino que integra experiências de aprendizagem de sala de aula face-a-face com experiências de aprendizagem *online*. Sua implementação representa um desafio frente às ilimitadas possibilidades de aplicação e contexto (GARRISON e KANUKA, 2004).

A forma de aplicação do ensino híbrido dependerá das capacidades e restrições da instituição de ensino. Há instituições que optam por trilhar um caminho mais brando, com alterações progressivas, mantendo o modelo curricular tradicional de disciplinas. Outras instituições buscam experiências mais inovadoras, com projeto pedagógico e metodologias baseadas em problemas e desafios (BACICH, *et al*, 2015).

Em relação às técnicas de ensino presencial, as escolas pertencentes à rede da SME seguem as Diretrizes Curriculares Nacionais para o Ensino Fundamental, propostas pelo Ministério da Educação. As escolas atuam conforme o modelo curricular predominante, com o ensino disciplinar das 9 áreas do conhecimento, a saber: Língua Portuguesa, Matemática, Ciências, Geografia, História, Língua Estrangeira, Educação Artística, Educação Física e Educação Religiosa. As principais técnicas utilizadas no ensino presencial são: aulas expositivas, leituras diversas, atividades e avaliações para verificação da aprendizagem.

Do 1º ao 5º ano as turmas são dirigidas por apenas um professor, professor titular, responsável por trabalhar todas as 9 áreas de conhecimento. Com exceção das salas de 5º ano onde há alunos surdos inclusos, nessas salas há também um professor intérprete de libras que auxilia o professor titular na comunicação com os alunos. Do 6º ao 9º ano, cada área de conhecimento é trabalhada por um professor específico, há também um professor intérprete nas turmas com alunos inclusos.

Já as técnicas de ensino híbrido praticadas pelas escolas da rede da SME funcionam como complemento às técnicas do ensino presencial. O público-alvo são alunos do ensino fundamental, do 3º ao 6º ano. Os conteúdos *online* são trabalhados na escola com o suporte de um *software* aplicativo e do laboratório de informática. Cada série possui um horário específico para a utilização do laboratório. Todo conteúdo *online* utilizado é composto por questões que contêm elementos de texto, áudio e vídeo.

3.1.1.1 *Softwares* existentes

A rede de escolas da SME utilizam 2 principais *softwares*: i-Educar e AprimoraEF.

O i-Educar é utilizado desde 2010 pelas escolas de rede da SME. É um sistema de gestão escolar, *open source*, distribuído pela licença GNU - *General Public License* (Licença Pública Geral), hospedado no Portal do Software Público Brasileiro. A plataforma é compatível com sistema operacional Linux e Windows. É escrito em linguagem PHP e utiliza o SGBD (Sistema Gerenciador de Banco de Dados) PostgreSQL.

As principais funcionalidades disponíveis no i-Educar são: cadastro único de alunos, gestão de matrículas, calendário letivo, quadro de horários, lançamento de notas e faltas, processamento de histórico escolar, cadastro de informações de servidores e o módulo biblioteca.

Por sua vez, o AprimoraEF é uma plataforma que disponibiliza questões de Língua Portuguesa e Matemática para todas as séries do ensino fundamental. Foi desenvolvido pela Positivo Informática, empresa do segmento de tecnologia educacional, que disponibiliza a plataforma para sistemas operacionais Windows e dispositivos móveis (Android e iOS), sem qualquer custo financeiro.

A plataforma é utilizada pelas escolas da rede da SME desde 2015, além de ofertar questões, o aplicativo apresenta outras funcionalidades como: cadastro de alunos e professores, gestão de turmas, acompanhamento de desempenho e relatórios.

Dentro do aplicativo as questões são separadas por níveis de dificuldade. As questões são baseadas nas diretrizes nacionais do SAEB (Sistema de Avaliação da Educação Básica) e da Prova Brasil. Cada aluno percorre as questões conforme a solicitação do professor (que indica qual conteúdo será trabalhado), posteriormente o aluno percorrerá conforme seu nível de conhecimento e ritmo de aprendizagem. O AprimoraEF possui elementos de gamificação, são usadas estratégias de jogos para motivar o aluno a aprender.

A plataforma possibilita acompanhamento de desempenho de alunos em tempo real. Ao final da realização das atividades, professores podem acompanhar o rendimento da turma de acordo com o número de acertos e erros.

3.1.2 Identificação do Problema

A SME pretende pôr em prática ações que viabilizem um ensino adaptativo na modalidade presencial.

O ensino adaptativo procura otimizar o processo de aprendizagem através do entendimento dos objetivos, das preferências e das limitações de cada aluno, auxiliando-o a encontrar o melhor caminho a percorrer rumo ao seu desenvolvimento (LOUREIRO, PAIVA, 2015).

A intenção é proporcionar a cada aluno um tratamento diferenciado conforme seu ritmo de aprendizagem. A princípio, o ensino adaptativo é praticado pelas escola da rede de forma elementar, através da utilização do AprimoraEF. A plataforma *online* utilizada identifica as habilidades e as dificuldades dos alunos e traça um caminho individualizado para cada aluno prosseguir. O plano é avançar, criando novas possibilidades no ensino presencial. Dessa forma, a SME deseja entregar a cada aluno o conteúdo adaptado: cronograma de atividades, leituras, provas, dentre outras coisas.

Para maior efetividade nesse novo desafio é fundamental o conhecimento do perfil de cada aluno. O perfil do aluno pode ser descrito sob 3 aspectos: dados pessoais (histórico do aluno), dados comportamentais percebidos pelos professores, e dados de desempenho do aluno no ensino *online*.

O primeiro aspecto refere-se aos dados pessoais e dados de histórico escolar do aluno. Tais dados são registrados no sistema de gestão escolar, i-Educar, nas funcionalidades de cadastro único de alunos, lançamento de notas e falta e processamento de histórico escolar. Compreendem dados como: data de nascimento, filiação, endereço, procedência escolar, relatórios de aprendizagem, notas, quantidade de faltas, números de reprovações, entre outros.

O segundo aspecto diz respeito aos assuntos comportamentais dos alunos. Os professores fazem registros sobre suas percepções em relação a cada aluno, no i-Educar. Na funcionalidade de lançamento de notas e faltas há um campo para incluir informações extra a respeito dos alunos, questões como: pontualidade, pro-atividade, participação, relacionamento com os colegas e relacionamento com os professores. Cada uma das questões mencionadas é avaliada segundo atribuição de um conceito que varia entre “ruim” a “excelente”.

Ressaltamos que os dados que compreendem os 2 primeiros aspectos descritos anteriormente são essenciais para a construção do perfil dos alunos e que apesar de serem coletados e registrados no *software* i-Educar, ainda não recebem nenhum tipo de tratamento que integre-os ou que torne sua visualização mais dinâmica aos gestores escolares.

O terceiro aspecto é registrado no AprimoraEF, na funcionalidade de acompanhamento de desempenho. O AprimoraEF armazena os dados referentes à quantidade de acertos e erros, gerando gráficos que mostram a performance de cada aluno. No entanto, o AprimoraEF não analisa algumas questões que poderiam ser tratadas para auxiliar o processo de construção do perfil do aluno. São questões como: qual o tempo para conclusão de determinada questão, em qual das duas disciplinas (Português ou Matemática) os alunos gastam mais tempo e qual tipo de questão com maior número de acertos e erros. Em relação ao tipo de questão, poderiam ser explorados dados como: quais as configurações da questão, se possui texto, áudio ou vídeo, trata-se de uma questão de múltipla-escola ou com alternativas em forma de *game.* Tais questões, que não são tratadas pelo AprimoraEF, são relevantes à SME por conter dados fundamentais sobre o aspecto de desempenho e aprendizagem de cada aluno.

Portanto, o desafio consiste em integrar os dados dos sistemas em uso, transformando-os em informações relevantes à SME para a identificação de perfis de alunos. Alguns dados encontram-se registrados nos *softwares* utilizados pela organização, outros ainda não foram coletados, caso de alguns dados provenientes do *software* AprimoraEF. Os dados já registrados, assim como os dados que ainda não foram coletados, que compreendem os 3 aspectos para a construção do perfil do aluno, são fundamentais para que os gestores das escolas da rede da SME tracem o perfil de cada aluno e a partir desse perfil seja possível praticar técnicas de ensino adaptativo com maior efetividade.

3.1.2.1 Identificação das Necessidades

A partir do perfil da organização em estudo e das demandas da mesma pela solução de um problema específico, é possível identificar quais ferramentas seriam mais adequadas para o tratamento do problema proposto.

A questão a ser solucionada pela organização tem como objetivo principal a identificação do perfil de cada aluno. Para a consecução do referido objetivo existem demandas que precisam ser satisfeitas. Cada uma delas foi descrita com base na observação do contexto vivido pelas escolas pertencentes à SME.

As demandas da organização e suas respectivas descrições estão apresentadas na Tabela 13.

Tabela 13: Demandas da organização e suas descrições.

Demanda	Descrição
Performance	Atender ao conjunto de dados da organização, alcançando o maior número possível de unidades de ensino atendidas.
Rapidez	Apresentar os resultados do processamento com custo de tempo reduzido. A organização pretende intensificar os trabalhos para prática do ensino adaptativo.
Tratamento de dados de diversos formatos	Certificar-se do efetivo processamento dos dados envolvidos no problema a ser solucionado. Tratar dos dados provenientes dos sistemas utilizados pela organização, assim como dos dados que ainda não são coletados pelos sistemas em uso.
Segurança dos dados	Garantir o correto funcionamento da ferramenta, evitando perdas de dados. Assegurar que o processamento dos dados pessoais dos alunos e de seus responsáveis sejam resguardados de exposições indevidas.
Usabilidade	Satisfazer as necessidades dos gestores de forma simples e eficiente, visto que a organização não dispõe de profissionais especialistas da área de TI.
Custo	Alcançar o atendimento das necessidades sem custo financeiro, considerando que a organização não dispõe de recursos para a aquisição de um *software* comercial.

Com base no conjunto de necessidades listadas na Tabela 13, é possível sugerir o uso de algumas soluções, a partir do estudo feito no capítulo 2. Convém ressaltar que, não serão mencionadas duas das ferramentas estudadas: MapReduce e RapidMiner. A primeira por não compreender uma ferramenta em si,

mas sim um modelo de programação. E a segunda, por contemplar funcionalidades que são ofertadas apenas na versão comercial.

3.1.2.1.1 Performance

Das 5 ferramentas disponíveis, todas demonstram a capacidade para tratar conjuntos grandes de dados, no entanto algumas ressalvas devem ser realizadas.

A arquitetura Hadoop utiliza único nó mestre, fato que poderá causar restrições em relação a escalabilidade, ainda que sejam fornecidos mecanismos de tolerância à falhas.

Frente às características da ferramenta Drill e a descrição de como ela é utilizada, a plataforma seria indicada para operar como um otimizador para Hadoop. Drill é focada em acesso, pesquisa e exploração de dados, servindo de suporte para outras ferramentas que trabalham com análise de dados. Vale lembrar que Drill reduz custos no tratamento de dados, permite acesso à dados NoSQL utilizando a linguagem SQL e o acesso à dados brutos.

As ferramentas Storm e Hortonworks também operam com a arquitetura do tipo mestre/escravo, no entanto elas utilizam uma alternativa para lidar com a dificuldade no gerenciamento dos nós. As ferramentas citadas anteriormente trabalham com o subprojeto Zookeeper, responsável pela configuração de nós e sincronização de processos distribuídos. Ressalta-se ainda que os nós do tipo escravo em Storm são ainda coordenados por uma estrutura do tipo *Supervisor* que é projetada para reagir rapidamente em situações de falhas, impedindo que os trabalhos sejam perdidos. Há ainda a possibilidade da arquitetura Hortonworks admitir o uso da ferramenta Hadoop e Storm em seus processos para acessar *stream* de dados.

Por fim, o projeto GridGain trabalha com *Hadoop Accelaration*, mecanismo que melhora o funcionamento do Hadoop, e com o sistema de arquivo GGFS, que pode funcionar como uma camada de cache, proporcionando mais flexibilidade no acesso aos dados e reduzindo a sobrecarga dos nós que executam tarefas.

3.1.2.1.2 Rapidez

Com relação ao requisito da rapidez, torna-se difícil determinar o quão rápido uma ferramenta é sem uma devida avaliação por comparação de valores estatísticos. Porém, é possível estabelecer um paralelo entre performance e rapidez. Portanto, a análise aqui realizada será feita com base no requisito performance.

Dispensando as influências exercidas pelas características de hardware e levando em consideração que o tempo gasto no processamento é influenciado por características internas de cada ferramenta, pode-se indicar que as ferramentas que apresentam maior capacidade de processamento serão também aquelas com menor custo de tempo em seus processos.

Dessa forma, as ferramentas mais indicadas para atender à necessidade em questão seriam: Storm, Hortonworks e GridGain. Além do Drill, que propõe-se a realizar análises de alta performance, porém é indicada para servir de suporte à outras ferramentas por ser projetada para consultar dados.

3.1.2.1.3 Tratamento de dados de diversos formatos

Conforme síntese das principais características das ferramentas *open source*, mostrada na Tabela 12, todas elas são flexíveis em relação ao tipo de dado que operam e todas são capazes de trabalhar com dados estruturas ou não. Assim, para o item em questão, qualquer das ferramentas estudadas podem atender à necessidade da organização para tratar dos dados provenientes dos sistemas em uso, bem como dos dados que ainda não são coletados.

3.1.2.1.4 Segurança dos dados

Ainda com base no estudo das ferramentas no Capítulo 2, todas oferecem algum mecanismo de segurança de dados. Entretanto, a organização almeja não

somente a confidencialidade dos dados, mas também o correto funcionamento da ferramenta, evitando perdas dos mesmos.

O Hadoop mantém mecanismos próprios para autenticação e autorização de usuários. Provê também 2 mecanismos de tolerância à falhas: o *back up* de estados persistentes do sistema de arquivo e a utilização de nó mestre secundário. Porém, na análise da performance, foi possível observar que o Hadoop, mesmo oferecendo os mecanismos citados anteriormente, apresenta vulnerabilidade mediante possíveis aumento de carga no sistema.

Drill também fornece mecanismos para configuração e autenticação de usuários.

Já o Storm, soluciona a questão do gerenciamento dos nós pelo uso do subprojeto Zookeeper, mas não oferece suporte à autenticação de usuários ou qualquer outro mecanismo próprio da ferramenta voltado para segurança.

Por sua vez, a plataforma Hortonworks também trata do gerenciamento de nós pelo uso do Zookeeper, evitando perdas de dados. Além disso, fornece um módulo específico para serviços de segurança, dentre os serviços destaca-se o uso do subprojeto Apache Knox, também utilizado para promover segurança em *cluster* Hadoop.

Para prevenir perdas de dados e falhas no processamento, a ferramenta GridGain pode utilizar seu sistema de arquivo (GGFS) em conjunto com o sistema de arquivo do Hadoop (HDFS), servindo como uma camada de cache. O GridGain também prover funcionalidades de segurança, incluindo recursos de autenticação e autorização, no entanto tais recursos estão disponíveis apenas na versão comercial.

3.1.2.1.5 Usabilidade

A organização em estudo busca por uma ferramenta capaz de satisfazer as necessidades dos gestores de forma simples e eficiente. Porém, fazer a avaliação da questão usabilidade não é uma tarefa simples, pois é difícil avaliar e comparar a usabilidade de várias ferramentas sem a aplicação de testes ou demonstrações de funcionamento. Contudo, serão usadas as características fornecidas pelos desenvolvedores de cada ferramenta.

As ferramentas apresentadas que pertencem a Apache (Hadoop, Storm, Drill e Hortonworks) possuem comunidades para desenvolvedores e colaboradores, além de fóruns para mediar discussões e sanar dúvidas. Destaca-se a ferramenta Hortonworks que promove treinamento para desenvolvedores, analistas e profissionais de TI. O projeto GridGain também fornece treinamento aos usuários através da ferramenta Apache Ignite.

3.1.2.1.6 Custo

Por se tratarem de ferramentas *open source*, a maioria atende ao anseio da organização por utilização de soluções sem custo financeiro. Porém, as plataformas RapidMiner e GridGain se configuram duas exceções, uma vez que somente disponibilizam determinadas funcionalidades nas versões comerciais.

3.1.3 Escolha da Ferramenta

A Tabela 14 ilustra as opções de ferramentas disponíveis para escolha de acordo com o atendimento às necessidades da organização. Serão usadas as seguintes abreviações: S (Satisfaz), SR (Satisfaz com Restrição) e NS (Não Satisfaz). Dessa forma, se determinada ferramenta satisfaz à uma demanda será usada a abreviação “S”, e assim sucessivamente.

Tabela 14: Atendimento às necessidades da organização

Ferramentas disponíveis	Demandas das organização					
	Performance	Rapidez	Tratamento de dados	Segurança de dados	Usabilidade	Custo
Hadoop	SR	SR	S	SR	SR	S
Drill	S	S	S	S	SR	S
Storm	S	S	S	NS	SR	S
Hortonworks	S	S	S	S	SR	S
GridGain	S	S	S	SR	SR	NS

De acordo com a Tabela 14 e as observações feitas sobre cada necessidade da organização, percebe-se que Hadoop, Storm e GridGain são as ferramentas que apresentam mais restrições. O Hadoop satisfaz com restrição 4 demandas: performance, rapidez, segurança de dados e usabilidade. Já o Storm não satisfaz a necessidade por segurança de dados e satisfaz com restrição à usabilidade. Por sua vez, o GridGain satisfaz com restrição a questão da segurança e da usabilidade, e não satisfaz a demanda por custo.

Portanto, as ferramentas Drill e Hortonworks seriam as mais indicadas, visto que ambas satisfazem à maioria da necessidades, com exceção da usabilidade que é satisfeita com restrição nos 2 casos. Como Drill é projetado para consultar dados, sua utilização seria mais indicada como um suporte à outras ferramentas. Dessa forma, Hortonworks seria a plataforma mais indicada, pois dispõe de um conjunto maior de recursos e funcionalidades, além de admitir o processamento de outras ferramentas como Storm (para acessar *stream* de dados) e o HDFS (sistema de arquivos do Hadoop).

Esse capítulo mostrou detalhadamente o estudo de caso de uma organização para exemplificar a aplicabilidade do estudo feito no capítulo 2 sobre as ferramentas *open source* para análise de dados. No próximo capítulo serão feitas algumas conclusões acerca da pesquisa desenvolvida nesta monografia, como também serão sugeridas propostas para trabalhos futuros.

4 Conclusão

Este trabalho teve como objetivo apresentar um estudo detalhado das principais ferramentas *open source* para análise de dados em *Big Data*.

A partir da pesquisa do conceito de *Big Data* foi possível compreender o valor que o correto tratamento de dados exerce sobre uma organização. A avaliação das principais ferramentas *open source* para análise de *Big Data* demonstrou que são inúmeras as possibilidades para tratamento de dados, cada ferramenta contendo características distintas. Portanto, como foi possível perceber, a escolha da ferramenta adequada não constitui uma tarefa simples frente às diversas opções.

Para demonstrar o propósito do estudo das ferramentas *open source* para análise de dados, elaborou-se o estudo de caso destinado exemplificar como pode ocorrer o processo de escolha da ferramenta adequada. A descrição do perfil da organização e a identificação do problema a ser resolvido foram fundamentais para descoberta das reais necessidades da organização. As plataformas para análise de *Big Data* estudadas foram avaliadas pela capacidade de atender às demandas identificadas. Ao final do estudo de caso, foi possível indicar qual a ferramenta mais apropriada à organização mediante seu perfil e suas necessidades.

Houve algumas limitações na produção da fundamentação teórica desta monografia. As principais fontes de pesquisa para descrição das ferramentas foram os sites dos seus respectivos desenvolvedores. Assim, as descrições contidas nas referidas fontes de pesquisa descreviam cada ferramenta exaltando suas supostas vantagens, ocultando possíveis inconveniências no uso. Mesmo nas descrições de empresas que utilizaram determinada ferramenta, não eram expostos detalhes da experiência percebida pela organização.

Outra dificuldade encontrada relaciona-se a elaboração do estudo de caso. Na última etapa do estudo, referente a análise das necessidades da organização junto às características das ferramentas, alguns itens não puderam ser avaliados com precisão. Caso da rapidez e da usabilidade, ambos seriam melhor avaliados se considerados valores estatísticos ou testes de funcionamento.

Em trabalhos futuros, pretende-se aplicar na organização em estudo, o processo completo para criação de um *Big Data.* Assim, deverá ser trabalhada não

somente a etapa de análise de dados, mas também as etapas para geração, aquisição e armazenamento, contemplando dados que ainda não são coletados e tratados. Após esse processo, pretende-se testar o funcionamento da ferramenta indicada no estudo de caso, para promover a análise de dados da organização em questão.

REFERENCIAS

Apache Drill (2015). Documentação. Disponível em: <http://drill.apache.org>. Acesso em: 05 de dezembro de 2015.

Apache Hadoop (2015). Documentação. Disponível em: <http://hadoop.apache.org>. Acesso em: 05 de outubro de 2015.

Apache Hortonworks (2014). Documentação. Disponível em: <http://br.hortonworks.com>. Acesso em: 20 de setembro de 2015.

Apache Storm (2014). Documentação. Disponível em: <http://storm.apache.org>. Acesso em: 10 de novembro de 2015.

Bandugula, N., **Insights from Apache Drill Beta – Part 1: Initial Drill Use Cases**, 2015. Disponível em: <https://www.mapr.com/blog/insights-apache-drill-beta-part-1-initial-drill-use-cases> Acesso em: 01 de março de 2016.

Bacich, L., Tanzi, A.N. e Trevisani, F.M, **Ensino Híbrido: personalização e tecnologia na educação**, Penso Editora Ltda, Porto Alegre, 2015.

Brasil, Ministério da Educação e Desporto**, Diretrizes Curriculares Nacionais para o Ensino Fundamental**, MEC, Brasília, 1998.

Chen, M., Mao, S., Zhang, Y. and Leung, V. **Big Data: Related Technologies, Challenges and Future Prospects**. Briefs in Computer Science. Springer International Publishing. 2014.

Chen, M., Mao, S. and Liu, Y. **Big Data: A Survey**, Mob New Appl 19, Springer Science + Business Media, New York, Pages 171-209, 2014.

Chen, H., Chiang, R., and Storey, V. **Business Intelligence and Analytics: From Big Data to Big Impact**, MIS Quarterly Executive, Special Issue: Business Intelligence Research, USA, Volume 36, Pages 1165-1188, 2012.

Dean, J. and Ghemawat, S. **MapReduce: Simplified Data Processing on Large Clusters**, Magazine Communications of the ACM – 50th anniversary issue: 1958 – 2008, Volume 51, Issu 1, New York, USA, 2008.

Gantz, J. and Reinsel, D. **Extra. Extracting Value from Chaos**, IDC iView, Jun, 2011. Disponível em: < https://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf > Acesso em: 25 de abril de 2016.

Garcia, M. **Informática Aplicada a Negócios**, 1ª edição, Editora Brasport, Rio de Janeiro, 2005.

Garrison, D.R. e Kanuka, H. **Blended learning: Uncovering its transformative potential in higher education**, Elsevier, Internet and Higher Education 7, p. 95-105, Canada, 2004.

GridGain (2015). Documentação. Disponível em: <http://gridgain.com>. Acesso em 25 de novembro de 2015.

Hilbert, M., and López, P. **The World's Technological Capacity to Store, Communicate, and Compute Information**, Science, Vol. 332, Issue 6025, p. 60-65, 2011. Disponível em:< http://science.sciencemag.org/content/332/6025/60> Acesso em: 15 de setembro de 2015.

IDC, "**Big Data Research**", 2016. Disponível em: < http://www.idc.com/prodserv/4Pillars/bigdata> Acesso em: 25 de abril de 2016.

IDC Brasil, "**Previsão da IDC para o mercado de TIC no Brasil em 2016 aponta crescimento de 2,6%**", 2016. Disponível em: <http://br.idclatin.com/releases/news.aspx?id=1970> Acesso em: 25 de abril de 2016.

Pingdom, **Internet 2012 in numbers**, 2013. Disponível em: <http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/> Acesso em: 13 de agosto de 2015.

Laney, D. **3d Data Management: Controlling Data Volume, Velocity and Variety**, Technical Report 949, META Group, 2001. Disponível em: <http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>. Acesso em: 30 de outubro de 2015.

Leibiusky, J., Eisbruch, G. and Simonassi, D. **Getting Started with Storm**. O'Reilly Media, USA, 2012.

Lohr, S. **The age of Big Data**. The New York Times. Feb, 2012. Disponível em:< http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html> Acesso em: 25 de abril de 2016.

Loureiro, A. e Paiva, A. **A personalização do ensino e aprendizagem**, Revista TecEduc, Editora Positivo, 2015. Disponível em: <http://www.positivoteceduc.com.br/palavra-do-especialista/personalizacao-do-ensino-e-aprendizagem/>. Acesso em: 25 de março de 2016.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A. **Big Data: The Next Frontier for Innovation, Competition and Productivity**, McKinsey & Company, 2011. Disponível em: <http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation >. Acesso em: 20 de outubro de 2015.

McCue, C. **Data Mining and Predictive Analysis: intelligence gathering and crime analysis**, 2ª edição, Editora Elsevier, USA, p. 368, 2007.

Mendes, E.G. **Breve histórico da educação especial no Brasil**, Revista Educación y Pedagogía, Volume 22, Número 57, 2010.

Miller, H. and Moker, P. **From Data to Decisions: A Value Chain for Big Data**, IT Professional, IEEE, Volume 15, Issue 1, Pages 57-59, 2013.

Mysore, D., Khupat, S. e Jain, S. **Arquitetura e Padrões de Big Data, Parte 1: Introdução à Classificação e à Arquitetura de Big Data**. IBM, 2014. Disponível em: < http://www.ibm.com/developerworks/br/library/bd-archpatterns1/> Acesso em: 25 de abril de 2016.

RapidMiner (2015). Documentação. Disponível em: <http://rapidminer.com>. Acessado em: 15 de dezembro de 2015.

Sampaio, A.F. **Letras e Memória - Uma Breve História da Escrita**, Editora: Ateliê Editorial, São Paulo, p. 302, 2009.

Schroeck, M.J., and Smart, J. **Analytics: The Real-World Use of Big Data**, IBM, New York, USA, 2012. Disponível em: <http://www-03.ibm.com/systems/hu/resources/the_real_word_use_of_big_data.pdf > Acesso em: 20 de outubro de 2015.

Tiefenbacher, K. and Olbrich, S. **Increasing the Value of Big Data Projects – Investigation of Industrial Success Stories**, 48th Hawaii International Conference on System Sciences (HICSS), Kauai, Hawaii, 2015.

White, T. **Hadoop: The Definitive Guide**, Third Edition, O'Reilly Media, USA, p. 660, 2012.

Zikopoulos, P., Eaton, C., Deroos, D., Deutsch, T., and Lapis, G. **Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data**, McGraw Hill, New York, USA, p.166, 2012.