Como analisar um Big Data com a Tecnologia Hadoop: Aplicação no Mercado Financeiro

Sabrina Bojikian Rissi[1] **Luís Alexandre da Silva**[1]

[1]Centro Paula Souza - Faculdade de Tecnologia de Bauru (FATEC)
Rua Manoel Bento da Cruz, 3-30 – Bauru – SP – Brazil

sabrinabrissi@gmail.com luis.silva51@fatec.sp.gov.br

Abstract. *Data storage technologies has been suffering several changes to keep up with the exponential increase in data volume, that has been growing in recent years.Alternative solutions are being created to supply the necessities. The objective of this study is to understand what is Big Data Hadoop and how it can be used in the purchase and sale of stock market, by solving the most important targets for Big Data technology, which is data storage and time processing. This paper is done by comparing the model of a relational database (MySQL) with the model of a non-relational database (Hadoop). The results show that the use of Big Data Hadoop is an efficient and satisfactory solution for database with a large volume of information.*

Resumo. *As tecnologias de armazenamento de dados tem sofrido várias mudanças para conseguir acompanhar o aumento exponencial de volume de dados, que vem crescendo nos últimos anos. Soluções alternativas estão sendo criadas para suprir essas necessidades. O objetivo deste trabalho é entender o que é o Big Data Hadoop e como ele pode ser utilizado no mercado de compra e venda de ações, solucionando os objetivos mais importantes para a tecnologia de Big Data, que é o armazenamento de dados e o tempo de processamento. Esse estudo é feito comparando o modelo de um banco de dados relacional (MySQL) com o modelo de um banco de dados não relacional (Hadoop). Os resultados apontam que o uso de um Big Data Hadoop é uma solução eficiente e satisfatório para banco de dados com grande volume de informação.*

1. Introdução

Nos dias de hoje, com a evolução crescente da tecnologia e com o aumento da população, as empresas e sistemas de internet estão gerando enormes quantidades de dados, da ordem de terabytes e até mesmo petabytes. Esses dados estão sendo gerados em enorme volume e grande velocidade, de diversos formatos e diferentes fontes, como imagem, video, blog, textos, dados de sensores, entre outros. Existe uma demanda muito grande para armazenar de forma eficiente esses dados, e mais do que só armazenar, há também a necessidade de processar e analisar essa grande quantidade de informações para torná-la utilizável. Em um mundo cheio de informações, aprender a lidar com grandes quantidades de dados tornou-se um ingrediente primordial para o sucesso nos negócios. O gerenciamento de um Big Data vem com vários desafios, incluindo o armazenamento de baixo custo e consultas através de

dados estruturados, semi-estruturados e não estruturados. Em janeiro de 2015 a revista Exame (2015) publicou uma reportagem mostrando um estudo feito pela *The Economist Intelligent Unit*, onde a maioria dos presidentes de empresas concordam sobre a importância do Big Data, mas a grande parte não tem amplo conhecimento em suas aplicações.

Um dos mercados que trabalha com grande volume de informação é o mercado financeiro. Segundo Sandroni (1999), Mercado Financeiro é um conjunto formado pelo mercado monetário e pelo mercado de capitais. Um bom investimento vem precedido de estudos em cima de balanço gráfico, notícias, histórico da empresa, especulações em internet, entre outros. O Hadoop é um banco de dados que pode processar grandes quantidades de informações de uma maneira eficiente que vai ajudar o investidor a fazer uma análise melhor e ter uma previsão de como o mercado vai agir.

O objetivo deste trabalho é entender o que é o Banco de Dados Hadoop e como ele pode ser utilizado no mercado de compra e venda de ações, solucionando os objetivos mais importantes para a tecnologia de Big Data, o armazenamento e o tempo de processamento. Mostrar como o Hadoop pode ser mais eficiente que um banco de dados relacional, se tratando de grande quantidade de dados. Para isso será feito um comparativo com o banco de dados relacional MySQL.

O mercado de Big Data vem crescendo cada vez mais nos dias atuais, aprender a lidar como essa ferramente é um grande diferencial. O mercado Financeiro é um bom lugar para se aplicar o conhecimento desse banco, já que as análises são feitas através de diferentes tipos de dados. No próprio site Bm&Bovespa (2015) essas informações podem ser obtidas de uma maneira simples.

Este artigo está estruturado da seguinte forma: na Seção 2 descreve-se a definição do que é um Big Data; na Seção 3 apresenta-se o banco de dados Hadoop; na Seção 4 uma introdução ao que se refere Mercado Financeiro; na Seção 5 uma descrição de como os experimentos foram feitos; na Seção 6 encontra-se os resultados obtidos; e na Seção 7 a conclusão.

2. Big Data

Big Data é o termo usado para conjunto de dados que são grandes e complexos para ser manipulados por métodos habituais. De acordo com a pesquisa feita pelo Instituto McKinsey Manyika et al. (2011) o grande volume de dados gerados, armazenados e minado para ideias tornou-se economicamente relevantes para as empresas, governo e consumidores. Para as empresas não basta apenas ter a capacidade de se obter informação. O passo seguinte, a análise de dados, é de grande valia. A importância desta área aumentou porque dá uma melhor visão dos dados estruturados e não estruturados, levando a análise potencialmente mais precisas, o que pode levar a tomada de decisão mais confiantes. Em 2001, um analista da Gartner, empresa de consultoria fundada por Gartner (1979), caracterizou o conceito de Big Data como os três "Vs": Volume, velocidade e variedade.

2.1. Volume

O volume apresenta o desafio mais imediato às estruturas de TI convencionais. É armazenamento escalável, e tem uma abordagem distribuída para consulta. Muitas empresas têm grandes quantidades de dados arquivados, mas não a capacidade de processá-los.

Partindo do princípio de que os volumes de dados são maiores do que a infra-estruturas do banco de dados relacionais convencionais pode lidar, opções de processamento que quebram amplamente em uma escolha entre arquiteturas de processamento massivamente paralelo, como data warehouses ou alguns bancos de dados. Esta escolha é frequentemente informada pelo grau em que um dos outro "Vs- variedade - entra em jogo. Normalmente, as abordagens de data warehouse envolvem esquemas pré-determinado, adequando um conjunto de dados regular e evoluindo lentamente. No Hadoop, por outro lado, não há condições de colocar sobre a estrutura dos dados que serão processados.

2.2. Velocidade

Velocidade para dar conta de determinados problemas, o tratamento dos dados (obtenção, gravação, atualização) deve ser feito em tempo hábil - muitas vezes em tempo real. Se o tamanho do banco de dados for um fator limitante, o negócio pode ser prejudicado. Como exemplo pode-se citar uma operadora de crédito, onde a demora de uma transação qualquer pode causar um grande transtorno. Essa demora é causada pelo fato de o sistema de segurança não conseguir analisar rapidamente todos os dados que podem indicar uma fraude.

2.3. Variedade

Variedade, outro aspecto importante. os volumes de dados que tempos hoje são consequência também da diversidade de informações. Temos dados em formato estruturados, isto é, armazenados em banco de dados como PostgreeSQL, MySQL, Oracle e outros. Temos os dados não estruturados, oriundos de inúmeras fontes, como documentos, imagens, áudios, vídeos e assim por diante. É necessário saber tratar a variedade como parte de um todos, um tipo de dados pode ser inútil se não for associado a outro.

2.4. Veracidade

Veracidade também tem que ser considerada. É necessário que haja processos que garantam o máximo possível a consistência dos dados. Voltando ao exemplo da operadora de cartão de crédito, imagine o problema que a empresa teria se o seu sistema bloqueasse uma transação genuína por analisar dados não condizentes com a realidade.

3. Hadoop

Um estudo feito por de Souza Issa (2011) mostra que para armazenar uma grande quantidade de informação, é necessário um banco de dados que além de possuir grande capacidade de armazenamento, consiga desempenhar suas funções sem perda de desempenho.

Foi escolhido o Banco de Dados Hadoop porque ele é um framework de código aberto. Foi desenvolvido pelo Apache na linguagem Java. É utilizado e conta a colaboração de grandes empresas como Yahoo, Facebook, IBM e Google, diz Patil (2014). Lida com volumes extremamente grandes de dados, dos mais variados tipos, suporta aplicações com volumes de dados que crescem substancialmente em pouco tempo. Hadoop conta por padrão com recursos de tolerância a falhas, como replicação de dados. Além de ser escalável, ou seja, quando necessário acrescentar computadores devido ao aumento de quantidade de dados, não é necessário reconfigurações complexas no sistema.

Segundo Apache (2015) a biblioteca de Sftware Hadoop é um framework que permite o processamento distribuído de grandes conjuntos de dados através de clusters de computadores que utilizam modelos de programação simples. Ele é projetado para garantir a alta escalabilidade a partir de um único servidor até um cluster com milhares de máquinas, cada uma oferecendo capacidade de armazenamento e computação local. Ao invés de confiar em Hardware para proporcionar maior disponibilidade, a própria biblioteca foi desenvolvida para detectar e tratar falhar na camada de aplicação, fornecendo um serviço com alta disponibilidade de um grid de computadores.

3.1. Componentes do Hadoop

O framework do Hadoop é formado por quatro componentes:

- Hadoop Common Utilitários comuns que suportam os outros módulos Hadoop.
- Hadoop Distributed File System (HDFS) Um sistema de arquivos distribuídos que oferece alto rendimento no acesso aos dados de aplicação.
- Hadoo YARN um framework de programação e gestão de recursos de cluster.
- Hadoop MapReduce Um sistema baseado no YARN para processamento paralelo de grandes conjuntos de dados.

Grandes empresas fazem uso desse Banco de Dados. Empresas como Facebook, Yahoo, Amazon, IBM, Twitter, eBay, LinkedIN, RackSapce são citadas por da Costa (2011) como usuárias do Hadoop.

3.2. Tecnologias Utilizadas no Hadoop

O quadro a seguir faz uma comparação de 3 tecnologias utilizadas no Hadoop.

Tabela 1. Comparação das Tecnologias

			
Destaque	MapReduce	Pig	Hive
Linguagem	Algoritmos da Função de MapReduce	Linguagem de Script	Parecido com SQL
Tipos de Esquemas	Não	Sim - Implícito	Não - Explícito
Partição	Não	Não	Sim
Servidor	Não	Não	Opcional
Linhas de Código	Muitas linhas	Poucas Linhas	Menos que MapReduce e Pig, devido a natureza do SQL
Tempo de Desenvolvimento	Muito Desenvolvimento	Rápido Desenvolvimento	Rápido Desenvolvimento
Abstração	Baixo Nível	Alto Nível	Alto Nível
Joins	Difícil de aplicar	Fácil	Fácil
Estruturado - Semiestruturado - não Estruturado	Pode trabalhar com todos esses tipos de dados	Pode trabalhar com todos esses tipos de dados	Melhor com estruturado e semi-estruturado
Complexidade de Lógica de Negócio	Bastante controle	Pouco Controle	Pouco Controle
Performance	Mais rápido que Pig e Hive	Mais lento do que programa MapReduce totalmente sintonizada, mas mais rápido do que o código mal escrito MapReduce	Mais lento do que programa MapReduce totalmente sintonizada, mas mais rápido do código mal escrito em MapReduce

A tabela acima mostra que, quando se usa o MapReduce, é necessário escrever uma lógica de negócios complexos para lidar com os joins. Pensar em como mapear e reduzir a perspectiva e em qual trecho em particular do código entrará no mapa e qual deles irá entrar no reduzir. É necessário um desenvolvimento grande para decidir como mapear e reduzir os joins. Todos os esquemas e esforços devem ser manuseados de forma programática.

Quando usado o Pig, não é possível particionar os dados, que pode ser utilizado para exemplos de processamento a partir de um subconjunto de dados por um símbolo em específico de determinada ação, uma data ou mês específico. Ele também não oferece a facilidade de mapear os dados em um esquema explícito.

O Hive oferece um modelo de programação familiar para as pessoas que conhecem SQL. Se aplicado o Hive para analisar os dados de estoque, os dados podem ser gerenciados em um esquema particular. Hive também tem seus servidores, pelo qual pode apresentar consultas Hive de qualquer lugar para o servidor Hive. Consultas HQL são convertidas em trabalho de MapReduce pelo compilador Hive, livrando os programadores de pensar em uma programação complexa e oferece uma oportunidade para se concentrar no problema do negócio.

3.3. Banco de Dados Hadoop vs Banco de Dados MySQL

O MySQL é um dos bancos de dados mais usados no mundo, por ser um Sistema Gerenciador de Banco de Dados Relacional de código fonte aberto, de baixo custo e de fácil uso, diz Duarte (2006). O MySQL foi feito sob a licença GNU e escrito em C e em C++, testado em uma ampla gama de diferentes compiladores, e não possuí vazamento de memória e funciona em várias plataformas.

A principal diferença entre o Hadoop e um banco de dados MySQL é o mecanismo de armazenamento, eles são fundamentalmente diferentes. O Banco de Dados MySQL armazena as informações em tabelas definidas por esquemas enquanto o Hadoop usa pares de valores como unidade fundamental. Outra diferença é a forma como os dados são consultados. No MySQL é usado comandos SQL para especificar consultas. Já o Hadoop usa programas de MapReduce que também pode ser iniciado através de comandos SQL para especificar quais os dados a serem recuperados e como eles serão recuperados. Uma terceira grande diferença entres esses bancos é a escalabilidade. O Banco de dados MySQL normalmente se adapta adicionando Hardware mais potentes em um conjunto único ou pequeno de servidores. O Banco de dados Hadoop se adapta adicionando muito mais Hardware, porém com menos poder, fazendo as máquinas trabalharem em paralelo.

O MySQL é um Banco de Dados Relacional e segue as formas tradicionais ACID (Atomicidade, Consistência, Isolamento e Durabilidade) que são propriedades fundamentais nesse tipo de banco, como diz no site Oracle (2015). Já o Banco de Dados Hadoop é um sistema de arquivos que é construído em redundância e paralelismo.

4. Mercado Financeiro

Mercado Financeiro na economia é um mecanismo que permite a compra e venda de valores mobiliários, mercadorias, câmbio e outros bens. No Brasil temos a BO-

VESPA, que segundo o próprio site Bm&Bovespa (2015) é a companhia que administra mercados organizados de títulos, valores mobiliários e contratos derivativos, além de prestar serviços de registro, compensação e liquidação, atuando, principalmente, como contraparte central garantidora da liquidação financeira das operações realizadas em seus ambientes.

Independente do perfil de um investidor do mercado financeiro, o seu objetivo de uma forma geral, é sempre fazer com o que seu dinheiro cresça. Não é necessário ser um grande gênio da economia, como Warren Buffet, ou George Soros, para saber que esse objetivo é alcançado com a compra de uma ação por um valor baixo e venda por um preço mais alto. Para isso é necessário estudo e análises. Segundo Chaves (2004) as operações do mercado financeiro faz referência à utilização das duas mais importantes ferramentas para análise de ativos: Análise Técnica e Análise Fundamentalista.

A análise fundamentalista consiste no estudo dos fundamentos econômicos de uma empresa. Neste tipo de análise, o investidor estuda os balanços e demonstrativos financeiros da empresa, analisa indicadores, considera o valor da marca, o crescimento passado, a gestão da empresa, entre diversos outros pontos.

A análise técnica, também conhecida como análise gráfica, é quando o investidor tenta identificar padrões e tendências a partir da análise de gráficos de cotações das ações, e assim ganhar na variação dos preços, comprando e vendendo nos momentos certos, tanto no curto prazo quanto no longo prazo. Para isto, são utilizadas diversas ferramentas e conceitos matemáticos, tentando prever os preços futuros de ações e títulos.

Independente do tipo de análise, ambas são feitas em cima de diversos tipos de informação. Com o Banco de Dados Hadoop os comerciantes do mercado financeiro e os acionistas podem processar grandes quantidades de dados não estruturados para identificar quais as melhores empresas para se investir.

Informações publicas, como notícia de empresa, análise de produtos, dados de fornecedores e mudança de preço podem ser processados em massa com o Hadoop, produzindo modelos matemáticos que ajudam na tomada de decisão, se determinada ação deve ser comprada ou vendida.

5. Experimentos

Todos os experimentos foram realizados em um computador com Sistema Operacional GNU/Linux Lubuntu, versão de Kernel 3.13.0-53-generic. A etapa inicial consiste na instalação dos Bancos de Dados. As versões do Hadoop e MySQL utilizadas foram 2.7.0 e 14.14, respectivamente. Os parâmetros de inicialização utilizados foram os padrões para ambos os bancos de dados.

Depois de analisar as principais ferramentas do Hadoop foi decidido utilizar o Hive. Ele foi devidamente instalado no Hadoop e a mesma consulta SQL utilizada no MySQL pode ser usada no Hive também.

Para os testes foi criada uma base de dados de tamanho pequeno (200MB a 500MB), uma base de dados de tamanho médio (500MB a 2GB) e uma base de dados de tamanho grande (2GB a 5 GB). Assim a quantidade de registros na tabela

variam de 5000 a 200000. O conjunto de dados foi um arquivo CSV, baixado do site (Yahoo, 2015). Onde esse arquivo contém informações das ações escolhidas em um determinado período de tempo. O conteúdo é separado por virgula contendo as cotações diárias, preços de abertura e fechamento, qual o maior preço do dia, o menor preço do dia. Na Figura 1 a modelagem da tabela.

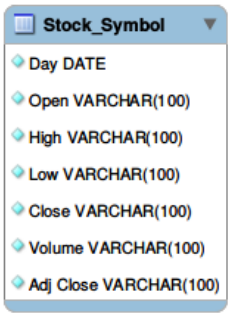

Figura 1. Modelagem Tabela

A tabela foi criada no MySQL através do seguinte comando SQL:

```
CREATE TABLE IF NOT EXISTS ‘Stock_Symbol‘ (
  ‘Day‘ date NOT NULL,
  ‘Open‘ varchar(100) NOT NULL,
  ‘High‘ varchar(100) NOT NULL,
  ‘Low‘ varchar(100) NOT NULL,
  ‘Close‘ varchar(100) NOT NULL,
  ‘Volume‘ varchar(100) NOT NULL,
  ‘Adj Close‘ varchar(100) NOT NULL
);
```

A tabela foi criada no Hive através do comando HQL:

```
CREATE TABLE Stock_Symbol (
Day date,
Open String,
High String,
Low String,
Close String,
Volume String,
Adj_Close String)
ROW format delimited fields terminated BY ’,’;
```

Após a criação da tabela, o script para carregar o arquivo no banco de dados MySQL:

```
load data local infile '<caminhoArquivo>' into table <nomeTabela>
columns terminated by ',' optionally enclosed by '"'
escaped by '"' lines terminated by '\n';
```

O script para carregar o arquivo no banco de dados Hive:

```
load data local inpath '<caminhoArquivo' into table <nomeTabela>;
```

Na Tabela abaixo o tamanho aproximado do banco de dados de acordo com o número de registros.

Tabela 2. Tamanho Aproximado do Banco

Registros	Tamanho Aproximado
5000	200MB
10000	400MB
25000	800MB
50000	1GB
100000	2GB
150000	3GB
200000	5GB

6. Resultado

Os resultados apresentados neste artigo avaliam apenas o tempo de execução das instruções SQL no banco de Dados MySQL e Hadoop em um contexto específico. Os resultados foram obtidos a partir da comparação de quantidade de dados em cada banco de dados e na comparação do resultado de tempo de acordo com a quantidade de dados obtido nos bancos.

Na tabela a seguir mostra a relação entre o tempo da consulta versus o tamanho do banco de dados.

Tabela 3. Resultado do Teste de Seleção

	MySQL	Hadoop
5000	4.20s	535.1s
10000	13.83s	543.64s
25000	85.42s	548.45s
50000	392.42s	553.44s
100000	1518.18s	557.51s
150000	1390.25s	581.5s
200000	2367.81s	582.7s

Para uma melhor visualização segue o gráfico representando o resultado dos testes realizados.

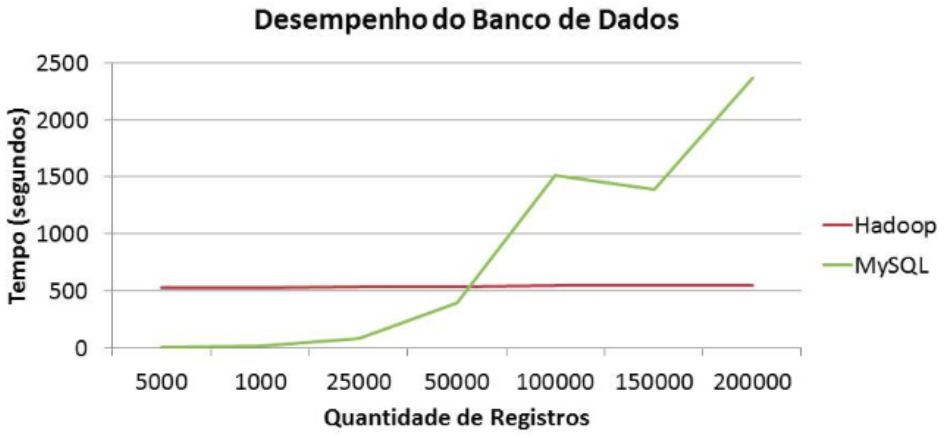

Figura 2. Gráfico Desempenho Hadoop e MySQL

Os resultados foram obtidos através de consulta idêntica em ambos os bancos de dados. Como o Hive utiliza o SQL foi utilizado o mesmo query para os dois Banco de Dados. A instrução de consulta executada foi um calculo para encontrar a covariância entre duas ações. Segue a consulta:

```
select  month(a.Date),
(AVG(a.High*b.High) - (AVG(a.High)*AVG(b.High)))
from ac_aa a join ac_aapl b on
a.Date=b.Date where  year(a.Date)=2008
Group by month(a.Date);
```

Esta consulta HQL irá desencadear o trabalho MapReduce como abaixo:

```
hive> select  month(a.Day),
    > (AVG(a.High*b.High) - (AVG(a.High)*AVG(b.High)))
    > from ac_aa a join ac_aapl b on a.Day=b.Day
    > where  year(a.Day)=2008
    > Group by month(a.Day);
Query ID = hduser_20151116174335_e3106ae5-d702-4a75-8c1c-8c558ba649fa
Total jobs = 1
15/11/16 17:43:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Execution log at: /tmp/hduser/hduser_20151116174335_e3106ae5-d702-4a75-8c1c-8c558ba649fa.log
2015-11-16 17:43:47     Starting to launch local task to process map join;      maximum memory = 477364224
2015-11-16 17:43:51     Dump the side-table for tag: 1 with group count: 253 into file: file:/tmp/hduser/61964070-c9ae-4828-8ce4-e6f86481c1cc/hive_
2015-11-16_17-43-35_703_4116875716168285418-1/-local-10004/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
2015-11-16 17:43:51     Uploaded 1 File to: file:/tmp/hduser/61964070-c9ae-4828-8ce4-e6f86481c1cc/hive_2015-11-16_17-43-35_703_4116875716168285418-
1/-local-10004/HashTable-Stage-2/MapJoin-mapfile11--.hashtable (20254 bytes)
2015-11-16 17:43:51     End of local task; Time Taken: 3.572 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2015-11-16 17:43:53,973 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local1195726518_0002
MapReduce Jobs Launched:
Stage-Stage-2:  HDFS Read: 12843456 HDFS Write: 11190776 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
```

Figura 3. Hadoop Map Reduce

Os resultados da análise de covariância:

Figura 4. Resultado Consulta

Ribeiro (2014) define variância como uma medida de dispersão que indica a regularidade de um conjunto de dados em função da medida aritmética. Com a variância, os investidores têm a oportunidade de buscar diferentes opções de investimento com base em seu respectivo perfil de risco, explica Bm&Bovespa (2015). É uma medida estatística de como um investimento se move em relação ao outro.

A covariância foi calculada entre duas ações diferentes para cada mês em uma data específica para o ano disponível. A partir dos resultados de covariância, corretores da bolsa ou gestores de fundos podem fornecer as recomendações abaixo:

- A variância positiva significa que a ação tem a tendencia de continuar na mesma direção que está indo, ou seja, se o preço está subindo, tende a continuar subindo, se o valor está caindo tende a continuar caindo.
- Uma variância negativa significa que os retornos tendem a mover inversamente. Se um instrumento de investimento está subindo ele tende a cair, e vice e versa.

7. Conclusão

A partir da estrutura montada para os bancos de dados, as consultas executadas resultaram que quando a quantidade de registros é inferior a cem mil, o uso de um banco de dados relacional, no caso MySQL, pode ser mais eficiente. Mas se tratando de grandes quantidades de informações o uso do Hadoop é melhor, pois demonstra um desempenho superior. Com esse estudo conclui-se que o Hadoop resolveu dois importantes objetivos da tecnologia de um Big Data. O armazenamento de dados e o processamento deles.

- Armazenamento: Ao armazenar os dados em enormes quantidade no Hadoop, a solução fornecida é muito mais robusta, econômica e escalável. Sempre que o tamanho dos dados aumentar, você pode simplesmente adicionar mais alguns nós e configurar o Hadoop.
- Processamento: Uma vez que o esquema Hive é criado em um banco de dados, tem-se a vantagem de executar consultas HQL em conjunto enormes de dados, e também são capazes de processar GBs ou TBs de dados com consultas simples SQL.

Referências

Apache. Apache hadoop, Junho 2015. URL `https://hadoop.apache.org/`.

Bm&Bovespa. Bm&bovespa a nova bolsa, 2015. URL `http://www.bmfbovespa.com.br/`.

D. A. T. Chaves. Analise tecnica e fundamentalista: Divergencias, similaridades e complementariedades. 2004. URL http://www.ead.fea.usp.br/TCC/trabalhos/TCC-DanielChaves-2004.pdf.

P. A. R. S. da Costa. Hadoop mapreduce tolerante a faltas bizantinas. page 1, 2011. URL http://docs.di.fc.ul.pt/bitstream/10451/8695/1/ulfc104210_tm_Pedro_Costa.pdf.

F. G. de Souza Issa. Estudo comparativo entre banco de dados relacionais e banco de dados nosql na utilizacao por aplicacoes de bussiness intelligence. 2011. URL http://fatecsjc.edu.br/trabalhos-de-graduacao/wp-content/uploads/2012/03/Trabalho-de-Gradua%C3%A7%C3%A3o-Felipe-G.-S.-Issa.pdf.

E. Duarte. Mysql. *SQL Magazine*, 3(37):22–27, 11 2006.

Exame. Empresas ainda nao sabem como usar o big data, diz estudo. 2015. URL http://exame.abril.com.br/negocios/noticias/empresas-ainda-nao-sabem-como-usar-o-big-data-diz-estudo.

Gartner. About gartner, 1979. URL http://www.gartner.com/technology/about.jsp.

J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. 2011. URL http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation.

Oracle. Mysql, 2015. URL https://dev.mysql.com/doc/refman/5.6/en/mysql-acid.html.

Y. K. Patil. Design and implementation of k-means and hierarchical document clustering on hadoop. 2014. URL http://ijsr.net/archive/v3i10/T0NUMTQ1MjY%3D.pdf.

A. G. Ribeiro. Variancia e desvio padrao. 2014. URL http://www.mundoeducacao.com/matematica/variancia-desvio-padrao.htm.

P. Sandroni. *Novissimo Dicionario de Economia*. Editora Best Seller, 1999.

Yahoo, Maio 2015. URL http://finance.yahoo.com.