# Comparação de Desempenho entre Ambientes Distribuídos Virtualizados na Mineração de Dados

**Joelson Antônio dos Santos**[1], **Murilo Coelho Naldi** [2]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - USP
São Carlos – SP – Brasil

[2]Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa (UFV)
Rio Paranaíba – MG – Brasil

joelsonn.santos@gmail.com, murilocn@ufv.br

***Abstract.*** *Nowadays, big amounts of data are challenging and cause the need for distribution and management of huge data sets in separate repositories. New distributed systems have been designed to scale up from a single server to thousands of machines. Systems like Apache Hadoop and Apache Mahout are flexible and reliable, supporting Data Mining techniques. Therefore, Virtualization became an important tool to contribute in the development of cheap and stable systems to support the analysis of large amounts of data. Nowadays, there are several consolidated virtualization tools on the market, like VMware, VirtualBox and Xen, among others. However, it may be difficult to determine which tool has the best performance for a given scenario of application. Therefore, computational performance evaluation techniques became important to assess accurately the advantages and disadvantages of each virtualization software. The main objective of this work is compare the performance of different distributed and virtualized environments on VirtualBox, VMware Player and Xen to support data mining tasks executed in the Apache Hadoop and Apache Mahout platforms. The performance of each environment is compared in order to evaluate the advantages of the use of Virtualization in the Data Mining context.*
***key-words:*** *Virtualization, Data Mining, Apache Hadoop, Apache Mahout, Big Data.*

***Resumo.*** *Atualmente, grandes quantidades de dados são um desafio e causam a necessidade de distribuição e gerenciamento de grandes conjuntos de dados em repositórios separados. Novos sistemas distribuídos foram desenvolvidos para escalonar de um único servidor para centenas de máquinas. Sistemas como o Apache Hadoop e Apache Mahout são flexíveis e confiáveis, possibilitando o suporte à técnicas de Mineração de Dados. Aliada à esses sistemas, a Virtualização é um mecanismo importante para o desenvolvimento de sistemas estáveis e econômicos para que sejam passíveis de analise de grandes quantidades de dados. Atualmente, existem diversos softwares de Virtualização consolidados no mercado como VMware, Virtualbox e Xen, dentre outros. Entretanto, é preciso escolher qual software de Virtualização atende com maior eficiência*

*as necessidades de cenários de aplicações reais ou simuladas. Técnicas de avaliação de desempenho são importantes para avaliar de forma mais precisa as vantagens e desvantagens de cada software de Virtualização. O principal objetivo deste trabalho consiste em desenvolver ambientes virtuais e distribuídos sobre os virtualizadores Virtualbox, VMware Player e Xen que sejam capazes de suportar as plataformas Apache Hadoop e Apache Mahout. O desempenho de cada ambiente desenvolvido é comparado por meio de técnicas de avaliação de desempenho computacional, a fim de buscar vantagens na utilização da Virtualização em tarefas de Mineração de Dados.*
***Palavras-chaves:*** *Big Data, Virtualização, Mineração de Dados, Apache Hadoop, Apache Mahout.*

## 1. Introdução

A evolução tecnológica e a necessidade de comunicação entre as pessoas são fatores para a produção contínua e armazenamento de grandes quantidades de dados. Publicações de fotos e mensagens em redes sociais, compras pela internet, ou até mesmo a quantidade de cliques e as buscas feitas por usuários por um determinado conteúdo na internet contribuem para a aumento da produção de dados que é conhecida como o *Big Data*. Além do volume de dados produzidos continuamente, o *Big Data* possui mais três dimensões: a variedade, a velocidade e a veracidade. A variedade representa os diferentes formatos de dados, planilhas, textos, músicas, vídeos, entre outros, todos esses formatos recebem o nome de dados não estruturados. A velocidade em que os dados são gerados e analisados é uma dimensão que atribui ao *Big Data* a possibilidade de analisar dados em tempo real, o objetivo é retornar um predição dos dados analisados em poucos segundos. Por fim, a veracidade lida com incerteza dos dados, uma vez nem sempre é possível encontrar certos padrões nos dados analisados mesmo com as técnicas mais sofisticadas, fatores climáticos e economia são alguns exemplos práticos [Schroeck et al. 2012].

Devido as características supracitadas, analisar manualmente os dados obtidos pode ser uma tarefa árdua ou até impossível. Portanto, é interessante o uso de técnicas de descoberta do conhecimento que sejam automáticas, como as tarefas de Mineração de Dados [TAN et al. 2009]. Dentre as tarefas de Mineração de Dados, podem ser listadas a Classificação, a Associação, o Agrupamento de Dados, dentre outros. Entretanto, os algoritmos utilizados neste trabalho tem como objetivo resolver o problema de Agrupamento de Dados. O problema de Agrupamento de Dados consiste em técnicas de aprendizagem não-supervisionada, ou seja, técnicas que não necessitam da intervenção ou conhecimento externo ao próprio conjunto de dados. Portanto, a tarefa de Agrupamento de Dados é ideal para dados novos ou sem conhecimento prévio. O Agrupamento de Dados pode ser representado pela organização dos dados em grupos de uma forma que possa facilitar o seu entendimento. Vários algoritmos foram propostos para resolver esse problema, dentre eles está o algoritmo $k$-médias, que consiste em separar um conjunto com $n$ objetos em $k$ grupos [Larose 2006, TAN et al. 2009]. Através de informações extraídas por essas técnicas e algoritmos, empresas podem encontrar padrões nos perfis dos clientes, tais como: preferências de compra, características pessoais e outras tendências [Faceli et al. 2011]. Adicionalmente, a combinação de técnicas de Mineração de Dados e as tecnologias de sistemas distribuídos proporciona um aumento na capacidade de análise de quantidades massivas de dados que não são suportadas pela limitada capacidade de

processamento e armazenamento de um simples computador [Sosinsky 2010].

Dado que técnicas de Mineração de Dados e a tecnologia de sistemas distribuídos são fatores importantes para análise de *Big Data*, softwares especialistas foram desenvolvidos para lidar com as dificuldades impostas pelo *Big Data*. Dentre eles, pode-se citar o *Apache Hadoop* e o *Apache Mahout*, ferramentas fundamentais para o desenvolvimento deste trabalho e de outros relacionados. O *Apache Hadoop* é uma plataforma distribuída de código aberto que tem como característica a capacidade de processar uma grande quantidade de dados de forma distribuída [White 2012]. Ao contrário de sistemas distribuídos convencionais, o *Apache Hadoop* dá aos desenvolvedores a flexibilidade, robustez, escalabilidade e simplicidade para o desenvolvimento de novas aplicações que são executadas na plataforma [Lam 2011]. Já o *Apache Mahout* é uma biblioteca de algoritmos de Mineração de Dados e *Aprendizado de Máquina*, desenvolvida sobre o modelo *MapReduce*. As aplicações desta biblioteca se encaixam no processamento de dados em larga escala [OWEN et al. 2012].

Apesar do suporte oferecido pelas ferramentas citadas, o *Big Data* ainda apresenta alguns desafios a serem solucionados. Dentre eles, podem ser citadas a instabilidade de sistemas físicos onde são instalados e configurados as ferramentas de análises e uma potencial ociosidade de recursos computacionais durante períodos de baixo uso. A instabilidade de sistemas físicos é uma grande inimiga do tempo de resposta necessário para a eficiência da análise dos dados. Essa instabilidade pode ser causada por uma vasta gama de razões: erros de instalação ou configuração dos softwares; uso de versões depreciadas de softwares; falhas de hardware e comunicação de rede; erros de usuários relacionados com a falta de conhecimento prévio ou mau uso as API's do *Apache Hadoop* e *Apache Mahout*; dentre outros [Rabkin and Katz 2013]. A ociosidade de recursos computacionais tornou-se um grande dilema devido ao crescimento desenfreado das tecnologias [VERAS 2011]. O aumento da velocidade de uma Unidade Central de Processamento (CPU), o aumento na capacidade de endereçamento de memória RAM, são fatores importantes para um bom desempenho computacional. Entretanto, podem ocasionar em desperdícios de recursos, tais como, alto consumo de energia elétrica, a necessidade de grandes espaços físicos para suportar os equipamentos e a dificuldade na recuperação de sistemas com falhas. Pesquisas do IDC dizem que, apenas 15% do total de recursos computacionais disponíveis, são utilizados em uma organização [VERAS 2011]. Logo, a Virtualização pode ser o mecanismo chave para conter partes desses problemas observados.

A Virtualização é capaz de lidar com os problemas de instabilidade e gerenciamento de recursos. Através dela é possível ter flexibilidade, rapidez na recuperação de sistemas em casos de falhas ocorridas e isolamento entre os sistemas virtualizados (máquinas virtuais) e sistemas hospedeiros. Adicionalmente, a Virtualização possui a capacidade de moldar o espaço de disco de acordo com a necessidade [Laureano 2006].

Para garantir maior precisão na escolha de um sistema computacional que possa ser mais eficiente em relação ao seu desempenho computacional, ou a melhor configuração de determinado sistema que possa contribuir com menor gasto de recursos financeiros ou energéticos, faz-se necessário a utilização de técnicas de avaliação de desempenho computacional. A *experimentação* ou *aferição* é uma dessas técnicas. A *experimentação* é utilizada para avaliar sistemas computacionais existentes, ou seja,

aplicações reais. O propósito da avaliação de desempenho computacional é medir de forma quantitativa e qualitativa a capacidade ou eficiência de determinado sistema, exemplos disso são: a medição de pacotes enviados por uma rede de computadores, a velocidade de transações de um determinado sistema gerenciador de banco de dados ou até mesmo a satisfação de usuários por determinado software [Johnson 2011]. Portanto, tais técnicas serão avaliadas neste trabalho.

O principal objetivo deste trabalho consiste em desenvolver ambientes virtualizados e distribuídos sobre os virtualizadores *Virtualbox, VMware Player* e *Xen* que sejam capazes de suportar as plataformas *Apache Hadoop* e *Apache Mahout*. Como objetivo específico tem-se, a instalação e a configuração de ambas plataformas em diferentes máquinas virtuais. A proposta é realizar uma comparação de desempenho computacional dos diferentes ambientes virtualizados a partir de experimentações feitas com o algoritmo *KMeans Mahout* implementado sobre a plataforma *Apache Mahout*, e o algoritmo *Multiple Parallel MapReduce k-means (MRMKMmeans)* que é uma versão paralela e distribuída para múltiplas execuções do algoritmo *k*-médias [Dearo Garcia and Coelho Naldi 2014]. Tal comparação será feita através de técnicas de avaliação de desempenho computacional tais como: análise de intervalos de confiança e modelos de regressão simples, utilizando como variável de resposta o tempo de execução dos algoritmos *KMeans Mahout* e *MRMKMeans* [Jain 1991, Galdámez 2002, Johnson 2011], a fim de indicar vantagens da Virtualização no contexto de Mineração de Dados.

O presente trabalho está dividido da seguinte forma: Na Seção 2 são apresentados os trabalhos relacionados a este. Na Seção 3 são apresentados os métodos de desenvolvimento e materiais utilizados. Na Seção 4 são apresentados os resultados obtidos, e na Seção 5 são apresentadas as conclusões e considerações finais deste trabalho.

## 2. Trabalhos Relacionados

O trabalho intitulado como *Virtualized Hadoop Performance with Vmware vSphere 5.1* [VMware 2013b], foi desenvolvido no ano de 2013 pela empresa *VMware*. Ele teve como objeto de estudo a avaliação de desempenho computacional entre dois ambientes distribuídos com 32 nós (cada) e configurados com a plataforma *Apache Hadoop*. Um desses ambientes foi desenvolvido sobre máquinas virtuais criadas sobre o software de virtualização *Vmware vSphere 5.1* e o outro ambiente sobre o sistema operacional nativo. As aplicações de *benchmarks* utilizadas para coletar dados foram: *TeraSort*, *TeraGen* e *TeraValidate*, também são chamadas coletivamente de *TeraSort suite*. Essas aplicações criam, classificam e validam registros de tamanhos de 100 *Bytes*, no qual, se tornam mecanismos que influenciam as operações de I/O, tráfego de rede e processamento que são atividades comuns processadas pelo *Apache Hadoop*. O tempo decorrido de execução entre os *benchmarks* escolhidos, a utilização da CPU, a taxa de transferência de dados pela rede e eficiência de hardware representaram as variáveis de resposta nos experimentos. De acordo com os resultados obtidos, a virtualização do *Apache Hadoop* funciona bem sobre o *Vmware vSphere 5.1* desde que o ambiente virtualizado seja configurado corretamente. Diferentemente do trabalho feito em [VMware 2013b], a comparação feita neste artigo estende-se para avaliação de desempenho entre outros softwares de virtualização quando aplicados no contexto de Mineração de Dados.

[Ivanov et al. 2014] realizou uma investigação sobre o desempenho computacional do *Apache Hadoop* em um *cluster* virtualizado. Sobre o software de virtualização *VMware vSphere 5.1*, foram construídas máquinas virtuais, onde uma foi denominada por *Master* e outras por *Workers*. Essas máquinas virtuais foram experimentadas com diferentes configurações em apenas uma máquina física. O objetivo principal do trabalho foi utilizar e avaliar o desempenho computacional de dois tipos de ambientes configurados com o *Apache Hadoop*, um com configurações padrão (*Standard Hadoop*) e outro chamado de *Data- Compute Hadoop*, ambos avaliados sobre o software *HiBench Benchmark*. Esse software proporciona uma carga de trabalho suficiente para as experimentações sobre algumas aplicações implementadas sobre a plataforma *Apache Hadoop*. Os fatores utilizados nesse trabalho foram: *WordCount* e *Enhanced DFSIO*, já a variável de resposta para os experimentos foi o tempo de execução de cada aplicação. Os resultados obtidos neste trabalho mostraram que as configurações do ambiente *Standard Hadoop* pode alcançar melhores resultados em relação ao *Data-Compute Hadoop*, entretanto, o último citado obteve resultados bons quando submetido a cargas de trabalho maiores. Diferentemente desse trabalho, avaliamos o desempenho computacional de uma variedade de ambientes virtualizados distribuídos sobre diferentes softwares de virtualização.

## 3. Metodologia

Nesta seção são apresentados todos os materiais utilizados para o desenvolvimento deste trabalho. Na Seção 3.1 são apresentadas as configurações do ambiente de desenvolvimento. E na Seção 3.2 é descrito o planejamento dos experimentos.

### 3.1. Ambiente de Desenvolvimento

Para o desenvolvimento dos ambientes virtualizados experimentalmente comparados foram utilizados três computadores conectados em rede. O computador mestre possui 1 TB de armazenamento, 32 GB de memória RAM, sistema operacional hospedeiro ubuntu 13.10 e processador AMD FX. Os outros dois computadores que foram utilizados como escravos possuem 500 GB de armazenamento, 15 GB de memória RAM, sistema operacional hospedeiro ubuntu 13.10 e processadores AMD FX.

Existem várias maneiras de utilizar os recursos computacionais por meio da Virtualização. Por exemplo, pode-se configurar a quantidade de máquinas virtuais por computador. Entretanto, neste trabalho foi implementada apenas uma máquina virtual por computador, a fim de garantir o máximo desempenho de cada máquina virtual. Basicamente, cada máquina virtual possui o *Apache Hadoop* e o *Apache Mahout* instalados e configurados de forma que seja possível executar experimentos de forma distribuída no ambiente virtualizado. A máquina virtual *VM-Master* possui o nó mestre do *Apache Hadoop*, ou seja, esse nó está configurado para ter o total controle sobre os outros nós do *Apache Hadoop* configurados nas máquinas virtuais *VM-1 e VM-2*. Cada máquina virtual implementada nos virtualizadores possuem o sistema operacional ubuntu 10.04 server, 4 núcleos de processador, 12 GB de memória RAM e 300 GB de armazenamento. A Figura 1 ilustra a rede local em alto nível utilizada, e a forma com que as máquinas virtuais estão implementadas em cada computador.

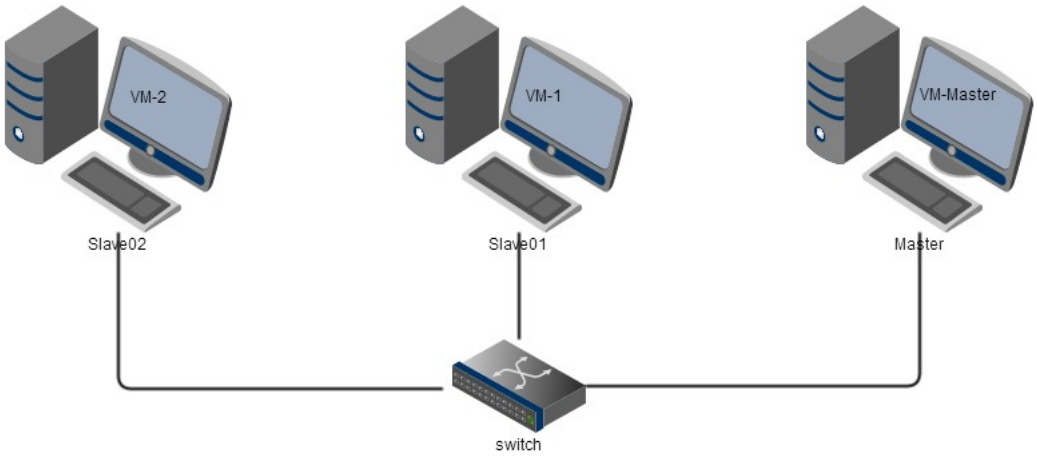



**Figura 1. Rede local**

Neste trabalho foram comparados os virtualizadores *Xen, VMware Player e Virtualbox*. A escolha destes virtualizadores se deu por serem alternativas de acesso gratuito e por serem produtos consolidados no mercado. O *Virtualbox* e o *VMware Player* são baseados na técnica de *virtualização total*, essa técnica consiste em virtualizar todos os recursos do *hardware* hospedeiro, a fim de aumentar a portabilidade entre arquiteturas distintas [Romero 2010, VMware 2013a, Portnoy 2012]. Já o *Xen* é baseado na técnica de *paravirtualização*, nessa técnica o sistema virtualizado é modificado para que o mesmo tenha permissão de acesso direto ao *hardware*, tal função melhora o desempenho computacional do virtualizador [Barham et al. 2003, Portnoy 2012].

### 3.2. Planejamento dos Experimentos

O planejamento de experimentos é uma das fases mais importantes da avaliação de desempenho. É nessa fase que são definidos os objetivos e características dos experimentos a serem realizados. Para isso, é necessário identificar alguns pontos que auxiliam na melhoria de processos de um determinado sistema ou produto. Esses pontos são: os *fatores*, *variáveis de resposta*, *níveis* e *carga de trabalho* [Galdámez 2002]. Os *fatores* indicam os objetos de estudo dos experimentos, por exemplo: algoritmo de ordenação, já os *níveis* representam as configurações ou tipos desses algoritmos, *quick sort* e *merge sort* são exemplos desses tipos. As *váriáveis de resposta* indicam as métricas que serão avaliadas, o tempo de execução de um determinado algoritmo, taxa de transferência de dados em rede são tidas como exemplos de utilização. E por fim, a *carga de trabalho* que representa o nível de estresse que um sistema tem que sofrer no momento em que é experimentado [Jain 1991, Johnson 2011].

A técnica de *experimentação* é utilizada neste trabalho pois, com ela, é possível obter resultados reais que podem ser reproduzidos. Portanto, essa técnica é indicada para sistemas completamente desenvolvidos. Por outro lado, é interessante ressaltar que a *experimentação* só é válida por meio da disponibilidade dos sistemas a serem testados. Consequentemente os resultados de experimentos que utilizam os mesmos *fa-*

*tores* e *variáveis de resposta* podem ter algumas diferenças dependendo da execução [Jain 1991, Johnson 2011].

No cenário de execução dos experimentos utilizado foram escolhidos três fatores, dois deles com 3 níveis cada e um com 2 níveis:

- *Base de dados*: são conjuntos de dados artificiais produzidos por meio de misturas de distribuições gaussianas [Melnykov et al. 2012]. As bases compostas por 10 atributos cada e, possuem 1 milhão, 5 milhões e 10 milhões de objetos, totalizando três níveis. Apesar de ser possível questionar se o volume de dados utilizado corresponde a *Big Data*, os conjuntos escolhidos se mostraram suficientes para avaliar a carga de trabalho do sistema experimental.
- *Algoritmo de Agrupamento de Dados*: foram utilizados dois algoritmos de agrupamento de dados, sendo eles o *KMeans Mahout* e *MRMKMeans*. O *Algoritmo kmeans Mahout* é uma versão do clássico algoritmo $k$-médias implementada sobre o modelo de programação *MapReduce* [OWEN et al. 2012]. Por sua vez, o *Algoritmo MRMKMeans* é uma versão múltipla e paralela do clássico algoritmo $k$-médias [Dearo Garcia and Coelho Naldi 2014]. O interesse de utilizar esse algoritmo nos experimentos se dá pela sua característica de executar vários algoritmos ao mesmo tempo de forma paralela e distribuída na plataforma *Apache Hadoop*, o que necessita ao máximo de seus recurso computacionais. Tal necessidade garante um esforço considerável dos ambientes virtualizados.
- *Virtualizadores: Xen, Virtualbox e VMware Player*.

As definições básicas para realização da avaliação de desempenho computacional deste trabalho são estruturadas por *variáveis de resposta*, *fatores* e *carga de trabalho*. O tempo de execução dos algoritmos: *MRMKmeans* e *KMeans Mahout* é a *variável de resposta* observada. A base de dados, os algoritmos de agrupamento e os virtualizadores constituem os *fatores* dos experimentos. Por último, a quantidade de iterações dos algoritmos *MRMKmeans* e *KMeans Mahout* são a *carga de trabalho*.

É importante ressaltar alguns pontos sobre a avaliação de desempenho realizada. O objetivo do trabalho não é mostrar qual virtualizador é melhor em absoluto, pois tal comparação extrapola o escopo deste trabalho, mas observar o comportamento de cada virtualizador submetido a cada combinação dos fatores do experimento, a fim de extrair vantagens na utilização de virtualização no contexto de Mineração de Dados. Para a utilização dos algoritmos *KMeans Mahout* e *MRMKMeans* foram necessárias algumas adaptações, uma vez que o algoritmo *MRMKMeans* varia seu valor de número de grupos ($k$) automaticamente a cada execução, ao contrário do outro algoritmo que avalia um valor de $k$ por vez. Então, foram delimitados os valores de $k$ entre 2 e 10 para os dois algoritmos. E para cada execução do algoritmo *MRMKMeans* corresponde a nove execuções do algoritmo *KMeans Mahout* com $k$ variando de 2 a 10.

## 4. Resultados Obtidos

Nas Figuras 2 e 3 são apresentados os intervalos de confiança das médias dos tempos de execuções dos experimentos descritos na Seção 3.2. Na Subseção 4.1 são apresentados detalhes sobre os resultados obtidos em relação aos tempos de execução através da análise dos intervalos de confiança das médias. Na Subseção 4.2 são apresentados modelos de regressão simples que demonstram a influência em porcentagem de cada fator nos experimentos.

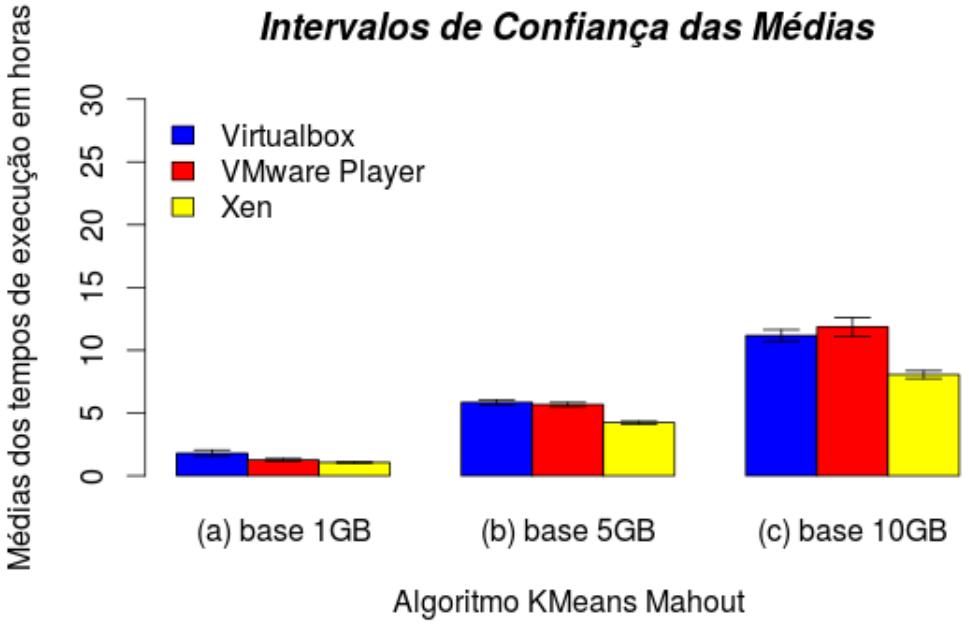



**Figura 2. Algoritmo KMeans Mahout - Intervalos de Confiança das Médias - Execuções sobre a base de dados de 1GB, 5GB e 10GB, respectivamente.**

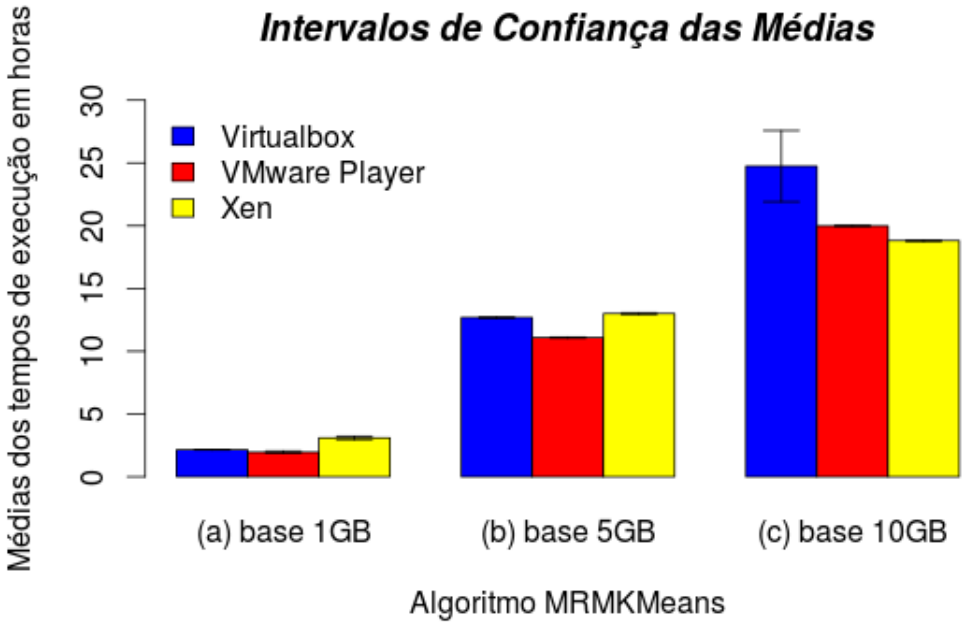



**Figura 3. Algoritmo MRMKMeans - Intervalos de Confiança das Médias - Execuções sobre a base de dados de 1GB, 5GB e 10GB, respectivamente.**

### 4.1. Intervalos de Confiança das Médias

Através da análise de intervalos de confiança é possível comparar o desempenho computacional de diferentes sistemas ao realizar experimentos entre eles para um determinado

cenário [Johnson 2011]. Logo, nota-se que nas Figuras 2 (a), 2 (b) e 2 (c) em que são apresentados os resultados dos experimentos realizados com o algoritmo *KMeans Mahout*, o virtualizador *Xen* obteve melhor desempenho sobre os outros virtualizadores, uma vez que o limite superior do intervalo de confiança das médias foi menor que os limites inferiores dos intervalos de confiança das médias do *Virtualbox* e *VMare Player*. Adicionalmente, os limites centrais dos intervalos de confiança dos três virtualizadores indicados na Figura 2 (a), indicam que o limite central do intervalo de confiança das médias do *Xen* obteve o menor intervalo entre o limite inferior e o limite superior, 1,021 e 1,096 respectivamente. Isso mostra que esses experimentos sofreram menores variações nos tempos de execução dos experimentos feitos neste virtualizador. O que também ocorre com os demais intervalos de confiança das médias mostrados nas Figuras 2 (b) e 2 (c). Portanto, nos experimentos feitos com o algoritmo *KMeans Mahout* sobre todas as bases de dados avaliadas, o *Xen* obteve melhores resultados tanto com menores médias de tempo de execução, quanto em termos de menores intervalos de confiança das médias em relação aos outros virtualizadores.

Na Figura 3 (a), o *VMware Player* obteve menor média de 1,954 com limite inferior igual a 1,850 e limite superior igual a 2,058, o *Virtualbox* obteve média de 2,156 com limite inferior igual a 2,130 e limite superior igual a 2,181 e, o *Xen* obteve maior média de 3,100 com limite inferior igual a 2,966 e limite superior igual a 3,234. Neste resultado, é relativamente trivial dizer que o *Xen* obteve o pior resultado em relação aos outros virtualizadores pois seu limite inferior é maior que os limites superiores dos outros virtualizadores. Por outro lado, não é possível indicar entre *Virtualbox* e *VMware Player* qual obteve melhor resultado, pois, os intervalos de confiança das médias, ou seja, os limites inferiores e superiores obtidos por ambos virtualizadores mostram que eles estão sobrepostos. Na Figura 3 (b) o virtualizador que obteve melhores resultados foi o *VMware Player*, entretanto os experimentos executados no *Virtualbox* obtveram menor intervalo entre o limite inferior e o limite superior, 2,130 e 2,181 respectivamente. Por fim, na Figura 3 (c) o virtualizador *Xen* obteve a menor média dos tempos de execução e menor intervalo de confiança das médias. Portanto, o *Xen* mostrou melhor desempenho em relação aos outros virtualizadores.

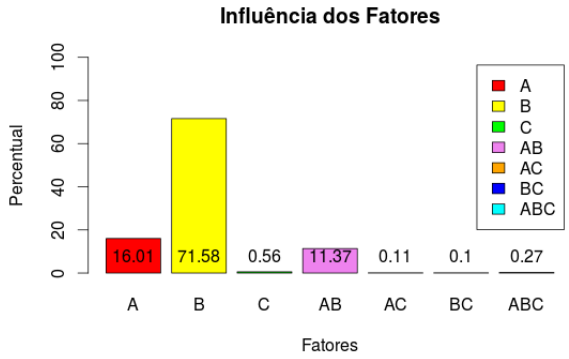
### 4.2. Influência de Fatores

A proposta do planejamento fatorial é definir as fatores importantes de um experimento, a fim de estudá-los em relação a influência que eles exercem sobre as variáveis de respostas escolhidas [Galdámez 2002]. O planejamento realizado neste trabalho foi o fatorial $2^k$, pois este método possibilita a análise de $k$ fatores com 2 níveis cada. Porém, neste trabalho, existem 2 fatores com 3 níveis cada, são eles: *base de dados virtualizadores*. Logo, houve a necessidade efetuar 9 combinações diferentes demonstradas na Tabela 1 para descobrir a influência de cada fator nos experimentos.

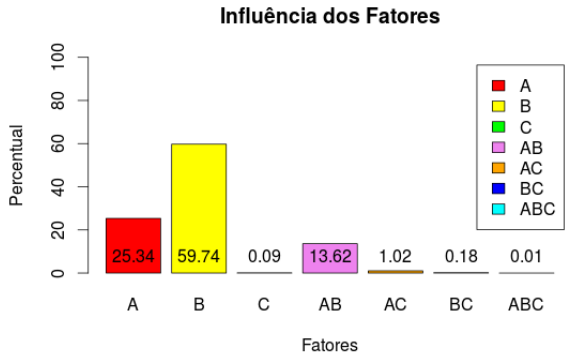
## Tabela 1. Combinações de testes para todos os fatores

| Experimento 1º Combinação | Fator A Algoritmo | Fator B Tamanho Base de Dados (GB) | Fator C Virtualizador | Tempo Médio de Execução (horas) | Experimento 2º Combinação | Fator A Algoritmo | Fator B Tamanho Base de Dados (GB) | Fator C Virtualizador | Tempo Médio de Execução (horas) | Experimento 3º Combinação | Fator A Algoritmo | Fator B Tamanho Base de Dados (GB) | Fator C Virtualizador | Tempo Médio de Execução (horas) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | KMeans Mahout | 1 | Virtualbox | 1,794 | 1 | KMeans Mahout | 1 | Virtualbox | 1,794 | 1 | KMeans Mahout | 1 | VMware Player | 1,269 |
| 2 | KMeans Mahout | 1 | VMware Player | 1,269 | 2 | KMeans Mahout | 1 | Xen | 1,058 | 2 | KMeans Mahout | 1 | Xen | 1,058 |
| 3 | KMeans Mahout | 5 | Virtualbox | 5,848 | 3 | KMeans Mahout | 5 | Virtualbox | 5,848 | 3 | KMeans Mahout | 5 | VMware Player | 5,674 |
| 4 | KMeans Mahout | 5 | VMware Player | 5,674 | 4 | KMeans Mahout | 5 | Xen | 4,231 | 4 | KMeans Mahout | 5 | Xen | 4,231 |
| 5 | MRMKMeans | 1 | Virtualbox | 2,156 | 5 | MRMKMeans | 1 | Virtualbox | 2,156 | 5 | MRMKMeans | 1 | VMware Player | 1,954 |
| 6 | MRMKMeans | 1 | VMware Player | 1,954 | 6 | MRMKMeans | 1 | Xen | 3,100 | 6 | MRMKMeans | 1 | Xen | 3,100 |
| 7 | MRMKMeans | 5 | Virtualbox | 12,685 | 7 | MRMKMeans | 5 | Virtualbox | 12,685 | 7 | MRMKMeans | 5 | VMware Player | 11,093 |
| 8 | MRMKMeans | 5 | VMware Player | 11,093 | 8 | MRMKMeans | 5 | Xen | 13,005 | 8 | MRMKMeans | 5 | Xen | 13,005 |
| **Experimento 4º Combinação** | **Fator A Algoritmo** | **Fator B Tamanho Base de Dados (GB)** | **Fator C Virtualizador** | **Tempo Médio de Execução (horas)** | **Experimento 5º Combinação** | **Fator A Algoritmo** | **Fator B Tamanho Base de Dados (GB)** | **Fator C Virtualizador** | **Tempo Médio de Execução (horas)** | **Experimento 6º Combinação** | **Fator A Algoritmo** | **Fator B Tamanho Base de Dados (GB)** | **Fator C Virtualizador** | **Tempo Médio de Execução (horas)** |
| 1 | KMeans Mahout | 5 | Virtualbox | 5,848 | 1 | KMeans Mahout | 5 | Virtualbox | 5,848 | 1 | KMeans Mahout | 5 | VMware Player | 5,674 |
| 2 | KMeans Mahout | 5 | VMware Player | 5,674 | 2 | KMeans Mahout | 5 | Xen | 4,231 | 2 | KMeans Mahout | 5 | Xen | 4,231 |
| 3 | KMeans Mahout | 10 | Virtualbox | 11,169 | 3 | KMeans Mahout | 10 | Virtualbox | 11,169 | 3 | KMeans Mahout | 10 | VMware Player | 11,858 |
| 4 | KMeans Mahout | 10 | VMware Player | 11,858 | 4 | KMeans Mahout | 10 | Xen | 8,054 | 4 | KMeans Mahout | 10 | Xen | 8,054 |
| 5 | MRMKMeans | 5 | Virtualbox | 12,685 | 5 | MRMKMeans | 5 | Virtualbox | 12,685 | 5 | MRMKMeans | 5 | VMware Player | 11,093 |
| 6 | MRMKMeans | 5 | VMware Player | 11,093 | 6 | MRMKMeans | 5 | Xen | 13,005 | 6 | MRMKMeans | 5 | Xen | 13,005 |
| 7 | MRMKMeans | 10 | Virtualbox | 24,740 | 7 | MRMKMeans | 10 | Virtualbox | 24,740 | 7 | MRMKMeans | 10 | VMware Player | 19,979 |
| 8 | MRMKMeans | 10 | VMware Player | 19,979 | 8 | MRMKMeans | 10 | Xen | 18,811 | 8 | MRMKMeans | 10 | Xen | 18,811 |
| **Experimento 7º Combinação** | **Fator A Algoritmo** | **Fator B Tamanho Base de Dados (GB)** | **Fator C Virtualizador** | **Tempo Médio de Execução (horas)** | **Experimento 8º Combinação** | **Fator A Algoritmo** | **Fator B Tamanho Base de Dados (GB)** | **Fator C Virtualizador** | **Tempo Médio de Execução (horas)** | **Experimento 9º Combinação** | **Fator A Algoritmo** | **Fator B Tamanho Base de Dados (GB)** | **Fator C Virtualizador** | **Tempo Médio de Execução (horas)** |
| 1 | KMeans Mahout | 1 | Virtualbox | 1,794 | 1 | KMeans Mahout | 1 | Virtualbox | 1,794 | 1 | KMeans Mahout | 1 | VMware Player | 1,269 |
| 2 | KMeans Mahout | 1 | VMware Player | 1,269 | 2 | KMeans Mahout | 1 | Xen | 1,058 | 2 | KMeans Mahout | 1 | Xen | 1,058 |
| 3 | KMeans Mahout | 10 | Virtualbox | 11,169 | 3 | KMeans Mahout | 10 | Virtualbox | 11,169 | 3 | KMeans Mahout | 10 | VMware Player | 11,858 |
| 4 | KMeans Mahout | 10 | VMware Player | 11,858 | 4 | KMeans Mahout | 10 | Xen | 8,054 | 4 | KMeans Mahout | 10 | Xen | 8,054 |
| 5 | MRMKMeans | 1 | Virtualbox | 2,156 | 5 | MRMKMeans | 1 | Virtualbox | 2,156 | 5 | MRMKMeans | 1 | VMware Player | 1,954 |
| 6 | MRMKMeans | 1 | VMware Player | 1,954 | 6 | MRMKMeans | 1 | Xen | 3,100 | 6 | MRMKMeans | 1 | Xen | 3,100 |
| 7 | MRMKMeans | 10 | Virtualbox | 24,740 | 7 | MRMKMeans | 10 | Virtualbox | 24,740 | 7 | MRMKMeans | 10 | VMware Player | 19,979 |
| 8 | MRMKMeans | 10 | VMware Player | 19,979 | 8 | MRMKMeans | 10 | Xen | 18,811 | 8 | MRMKMeans | 10 | Xen | 18,811 |

Dado todas as combinações de testes sobre os fatores apresentadas na Tabela 1, nas Figuras 4 a 8 são mostrados os percentuais de influência de cada fator dos experimentos.

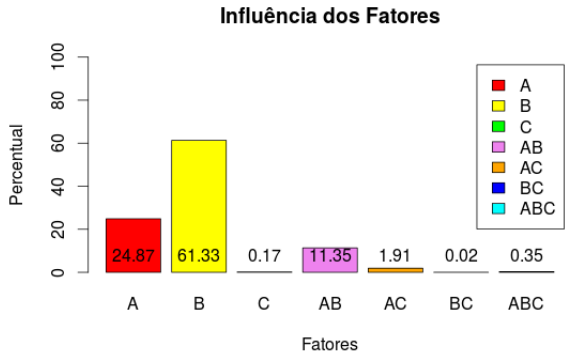



**(a)** Influência dos Fatores - 1º Combinação

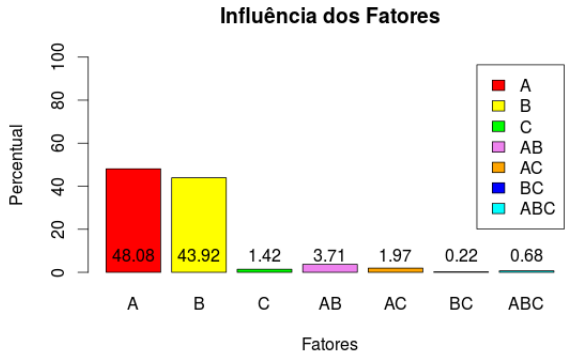



**(b)** Influência dos Fatores - 2º Combinação

**Figura 4. Influência dos Fatores - 1º e 2º Combinação**

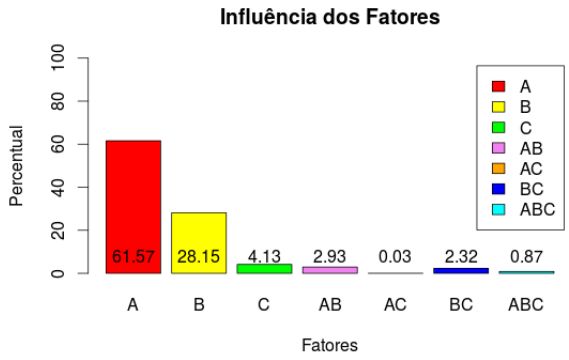



**(a)** Influência dos Fatores - 3º Combinação

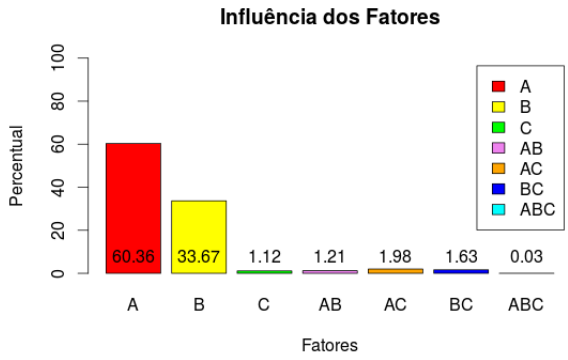



**(b)** Influência dos Fatores - 4º Combinação

**Figura 5. Influência dos Fatores - 3º e 4º Combinação**

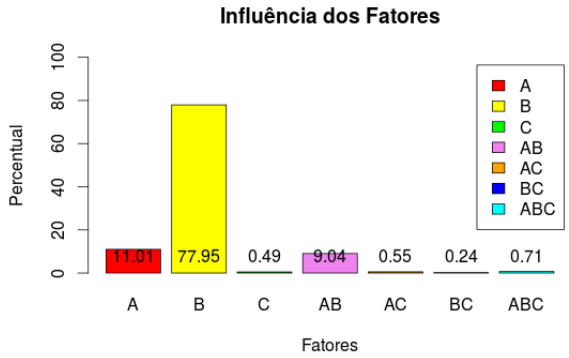



**(a)** Influência dos Fatores - 5º Combinação

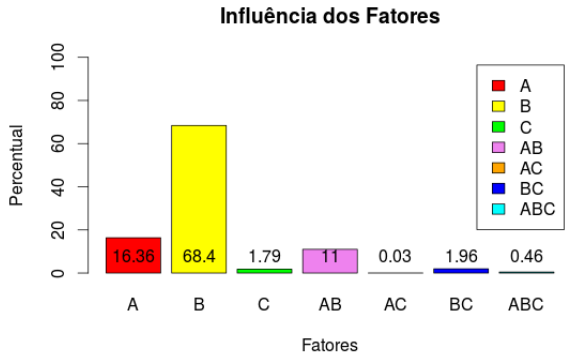



**(b)** Influência dos Fatores - 6º Combinação

**Figura 6. Influência dos Fatores - 5º e 6º Combinação**

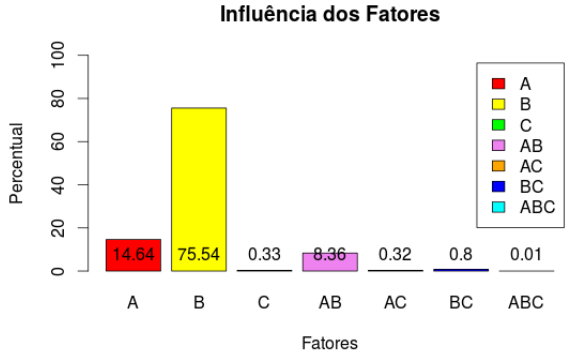



**(a)** Influência dos Fatores - 7º Combinação




**(b)** Influência dos Fatores - 8º Combinação

**Figura 7. Influência dos Fatores - 7º e 8º Combinação**




**(a)** Influência dos Fatores - 9º Combinação

**Figura 8. Influência dos Fatores - 9º Combinação**

Através da figuras apresentadas, nota-se que na maioria das combinações de teste feitas, o fator que mais influenciou nos experimentos foi a *base de dados*, isso pode ser explicado pelo o aumento gradativo da quantidade de objetos contidos nas *base de dados*. O segundo fator mais influente nos experimentos foi o *algoritmo de agrupamento* como pode ser visto nas Figuras 5b, 6a e 6b, as execuções sobre o algoritmo *MRMKMeans* fez com que as máquinas virtuais despendessem maior esfoço em relação as execuções sobre o algoritmo *KMeans Mahout*. Houve também, uma breve influência entre o *algoritmo de agrupamento* e a *base de dados* em algumas das combinações de teste, isso indica que essas combinações resultaram em médias de tempos de execução bem distintas. Já o fator *virtualizador* sofreu pouca influência de acordo com o percentual de influência indicado pelas figuras representadas na Seção 4.2, pois na maioria dos percentuais de influência dos virtualizadores apresentaram poucas diferenças entre elas. Uma vez que as instruções dos algoritmos de agrupamento de dados não são modificadas a cada execução dos experimentos nos diferentes virtualizadores, e sim submetidos a cargas de trabalhos diferentes, pressupõe-se que as arquiteturas de construção dos softwares de virtualização não sofreram impactos significantes em termos de operações de I/O entre as máquinas virtuais e o sistema hospedeiro. Portanto, a escolha do software de virtualização foi o fator que possui a menor influência no tempo computacional quando aplicado em tarefas

de agrupamento de dados distribuídos em plataformas escaláveis.

## 5. Conclusão

Com o desenvolvimento deste trabalho, observamos que todos os objetivos foram alcançados, tanto no desenvolvimento dos ambientes virtualizados quanto na avaliação de desempenho dos mesmos. A construção de diferentes ambientes virtualizados e distribuídos mostrou alguns aspectos importantes sobre utilização de Virtualização para a tarefa de Mineração de Dados Distribuídos. A flexibilidade e a facilidade para recuperar e gerenciar os ambientes distribuídos durante a realização dos experimentos propostos neste trabalho denotam esses aspectos.

Adicionalmente, a utilização de técnicas de avaliação de desempenho computacional foi possível identificar de forma mais precisa a influência de cada fator nos experimentos realizados neste trabalho. Notou-se a partir dos resultados, que a escolha do software de virtualização não influenciou tanto quanto o número de objetos de cada *base de dados* ou o *algoritmo de agrupamento de dados* utilizado. Logo, tal característica dos resultados mostraram que a Virtualização independente dos softwares utilizados pode trazer benefícios no contexto de Mineração de Dados.

### 5.1. Trabalhos Futuros

Como trabalhos futuros tem-se o estudo e avaliação de desempenho entre ambientes distribuídos configurados com as plataformas *Apache Hadoop* e *Apache Mahout* sem Virtualização e os ambientes virtuais desenvolvidos neste trabalho. Adicionalmente, será proposto uma avaliação de desempenho com diferentes quantidades de máquinas virtuais por computador.

## Referências


Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003). Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37(5):164–177.

Dearo Garcia, K. and Coelho Naldi, M. (2014). Multiple parallel mapreduce k-means clustering with validation and selection. In *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*, pages 432–437.

Faceli, K., Gama, J., Carvalho, A. C. P. L. d., and Lorena, A. C. (2011). *Inteligência Artificial, Uma Abordagem de Aprendizado de Máquina*. GEN.

Galdámez, E. V. C. (2002). Aplicação das Técnicas de Planejamento e Análise de Experimentos na Melhoria da Qualidade de um Processo de Fabricação de Produtos Plásticos. *Dissertação de Mestrado*.

Ivanov, T., Zicari, R. V., Izberovic, S., and Tolle, K. (2014). Performance evaluation of virtualized hadoop clusters. *CoRR*, abs/1411.3811.

Jain, R. (1991). The art of computer system performance analysis: techniques for experimental design, measurement, simulation and modeling. *New York: John Willey*.

Johnson, T. (2011). *Avaliação de Desempenho de Sistemas Computacionais*. Gen.

Lam, C. (2011). *Hadoop in Action*. Manning.

Larose, D. T. (2006). *Data mining methods & models*. John Wiley & Sons.

Laureano, M. (2006). *Máquinas Virtuais e Emuladores, Conceitos, Técnicas e Aplicações*. Novatec.

Melnykov, V., Chen, W.-C., and Maitra, R. (2012). Mixsim: An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):131–158.

OWEN, S., Anil, R., Dunning, T., and Friedman, E. (2012). *Mahout in Action*. Manning Publications (October 17, 2011).

Portnoy, M. (2012). *Virtualization Essentials*. Wiley / Sybex.

Rabkin, A. and Katz, R. (2013). How hadoop clusters break. *Software, IEEE*, 30(4):88–94.

Romero, A. V. (2010). *Virtualbox 3.1 - Deploy and Manage a cost-effective virtual environment using Virtualbox - Beginner's Guide*. PACKT.

Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., and Tufano, P. (2012). Analytics : The real-world use of big data - How innovative enterprises extract value from uncertain data. Technical report, IBM Global Services, Route 100 Somers, NY 10589 U.S.A.

Sosinsky, B. (2010). *Defining Cloud Computing*, pages 1–22. Wiley Publishing, Inc.

TAN, P.-N., STEINBACH, M., and KUMAR, V. (2009). *Introdução ao Data Mining, Mineração de Dados*. CIÊNCIA MODERNA.

VERAS, M. (2011). *Virtualização, Componente Central do Datacenter*. Brasport.

VMware (2013a). Getting started with vmware player - vmware player 6. `http://www.vmware.com/pdf/desktop/vmware_player60.pdf`. Acessado em 02/05/2014.

VMware (2013b). Virtualized hadoop performance with vmware vsphere® 5.1 - performance study - technical white paper. `http://www.vmware.com/files/pdf/vmware-virtualizing-apache-hadoop.pdf`. Acessado em 28/11/2014.

White, T. (2012). *Hadoop The Definitive Guide*. O'REILLY, 3º edition.