Análise Integrada de Desempenho e Consumo de Energia em Sistemas de Armazenamento de Dados Distribuídos

Juccelino Barros[1], **Gustavo Callou**[1], **Glauco Gonçalves** [1]

[1]Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco (UFRPE) – Recife – PE – Brasil

{juccelino.abarros,gustavo.callou,glauco.goncalves}@ufrpe.br

Abstract. *Performing large amount of data storage is a challenge, since the demand to manage the massive volume of data is growing. However, traditional data storage systems, such as relational databases, were not designed to meet this demand. Therefore, new data storage systems have been emerging, for instance, non-relational databases NoSQL. The main factor for choosing a data storage system is the performance in typical operations. However, energy consumption also represents a major factor for both environment and cost. This paper performs an integrated analysis of performance and energy consumption of a distributed data storage system, represented by Cassandra, in the data insertion process. A case study was conducted taking into account measurement techniques and the YCSB benchmark to generate load in several scenarios with different numbers of nodes in the Cassandra cluster. The achieved results were significant, observing in several situations that the improvement of the computational resources lead, as a matter of priority, the increase of the energy consumption, without having a representative increase in the performance.*

Resumo. *Realizar o armazenamento de grande quantidade de dados é um desafio, visto que a demanda para gerenciar o volume massivo de dados vem crescendo. Contudo, os sistemas tradicionais de armazenamento de dados, como os bancos de dados relacionais, não foram projetados para atender à essa demanda. Consequentemente surgiram novos sistemas de armazenamento de dados, entre eles os bancos de dados não relacionais NoSQL. O principal fator para a escolha de um sistema de armazenamento de dados é o desempenho em operações típicas. No entanto, o consumo de energia também é considerado um fator importante principalmente para o meio ambiente e para os investimentos. Este artigo realiza uma análise integrada de desempenho e consumo de energia, através da medição, de um sistema de armazenamento de dados distribuídos, representado pelo Cassandra no processo de inserção de dados. Um estudo de caso foi realizado fazendo uso de técnicas de medição e do benchmark YCSB para gerar carga em diversos cenários com quantidade diferente de nós no cluster do Cassandra. Os resultados obtidos foram significativos, observando-se em diversas situações que o aumento dos recursos computacionais acarretam, prioritariamente, o aumento do consumo de energia, sem ter um incremento representativo no desempenho.*

1. Introdução

A demanda para gerenciar grandes quantidades de dados em *datacenters* vem crescendo ao longo dos anos. Além disso, os tradicionais sistemas de armazenamento de dados,

como os bancos de dados relacionais, não foram projetados para atender esta demanda [Lóscio et al. 2011] e, consequentemente, novos sistemas estão surgindo.

Paralelo ao aumento desta demanda está o rápido crescimento no consumo de energia nesses ambientes [Koomey 2011][Venkatraman 2012]. Segundo [NRDC 2015], o consumo de energia dos *datacenters* nos Estados Unidos foi de 91 bilhões *KiloWatts* (KW) em 2013, custando USD 9 bilhões. Estima-se que este consumo atingirá 140 bilhões de KW em 2020, custando em torno de USD 13,7 bilhões. Alguns dos principais motivos para este crescimento foram o aumento do número de *datacenters*, a demanda de dados e o barateamento do custo de aquisição de novos equipamentos, como servidores e demais componentes.

Entre os novos sistemas de armazenamento de dados estão os bancos de dados não relacionais, conhecidos como NoSQL (*Not Only SQL*), os quais são facilmente escaláveis entre várias máquinas e que têm o foco no desempenho em operações típicas como inserção e consulta. Os bancos de dados NoSQL apresentam características diferentes dos bancos de dados relacionais, pois focam na disponibilidade dos dados ao invés da consistência e não apresentam os padrões rígidos encontrados no modelo relacional [McCreary and Kelly 2014]. Por serem facilmente escaláveis, bancos de dados NoSQL permitem maior tolerância a falhas. O principal fator para a escolha de um sistema de armazenamento de dados é o desempenho em operações típicas. Este fator é bastante investigado na academia [Cooper et al. 2010][Li and Manoharan 2013][Abubakar et al. 2014]. Contudo, o consumo de energia também é considerado um fator importante principalmente para o meio ambiente e para os investimentos. Porém, quais são as melhores formas de utilização desses sistemas levando em consideração o desempenho e o consumo de energia?

Este artigo realiza uma análise integrada de desempenho e consumo de energia, através da medição de um sistema de armazenamento de dados distribuídos, representado pelo Cassandra, no processo de inserção de dados. Para isso, foi utilizada a ferramenta de *benchmark Yahoo! Cloud Serving Benchmark* (YCSB) [Cooper et al. 2010] para gerar carga em diversos cenários variando a quantidade de operações de inserção e a quantidade de nós no *cluster* do Cassandra. Sendo assim, o objetivo desse trabalho é fornecer insumos para ajudar os projetistas de ambientes de armazenamento de grande quantidade de dados a escolher uma configuração mais apropriada do seu ambiente de armazenamento, levando em consideração esses dois requisitos conflitantes, energia e desempenho.

O artigo está dividido conforme descrito a seguir. A Seção 2 apresenta as características de um sistema de armazenamento de dados distribuídos, abordando alguns aspectos do Cassandra e uma descrição da ferramenta geradora de carga YCSB. A Seção 3 mostra os trabalhos relacionados. Em seguida, a Seção 4 apresenta a metodologia utilizada nessa pesquisa. A Seção 5 apresenta um estudo de caso que realiza uma análise de desempenho e consumo de energia nos cenários propostos. Por último, a conclusão do artigo e futuros direcionamentos desse trabalho são apresentados.

2. Fundamentação Teórica e Ferramentas

Esta seção apresenta as principais características de um sistema de armazenamento de dados distribuídos, abordando alguns aspectos dos bancos de dados não relacionais e do Cassandra. Também é oferecida uma descrição do *Yahoo! Cloud Serving Benchmark*.

2.1. Características dos Sistemas de Armazenamento de Dados Distribuídos

Com o surgimento de novos meios de acesso à Internet, o número de dados trafegados está em constante crescimento. Com o objetivo de atender a grande demanda de dados, os sistemas de armazenamento estão cada vez mais robustos. Estes sistemas de armazenamento apresentam um conjunto de características que atendem a grande demanda de dados. Entre as principais características estão: escalabilidade, elasticidade e alta disponibilidade.

A escalabilidade pode ser dividida em 2 formas: vertical e horizontal. A vertical consiste em aumentar o número de recursos computacionais (memória, processador e espaço em disco) no servidor com o sistema de armazenamento. Já a horizontal consiste em aumentar o número de servidores, com cópias ou partes do sistema de armazenamento [Lóscio et al. 2011]. A elasticidade compreende em adicionar mais servidores com o sistema de armazenamento de dados em execução e sem afetar os demais componentes negativamente, podendo até distribuir os dados entre as instâncias do sistema de armazenamento seguindo critérios previamente definidos. A alta disponibilidade consiste em fazer com que o sistema de armazenamento de dados seja tolerante a falhas, visto que em um cenário com vários servidores é comum acontecer uma falha. Logo estes sistemas possuem técnicas e estratégias para prover alta disponibilidade mesmo quando os servidores apresentam problemas [De Diana and Gerosa 2010].

Um dos grandes desafios é aplicar essas 3 características nos sistemas de armazenamento tradicionais [Cooper et al. 2010], como os bancos de dados relacionais, pois eles não foram projetados com tais finalidades. Consequentemente, novos bancos de dados foram surgindo com essas características. Porém, para atender as 3 principais características, as consultas complexas e transações sofisticadas foram sacrificadas em alguns desses novos sistemas de armazenamento de dados. Entre os sistemas de armazenamento de dados distribuídos que têm essas características estão os bancos de dados NoSQL, que são pertencentes a um novo paradigma chamado de Não Relacional.

2.2. Banco de Dados Não Relacional - NoSQL

Os bancos de dados NoSQL (*Not Only SQL*) não utilizam a linguagem SQL para realizar suas transações. Eles apresentam um conjunto de conceitos que permitem o processamento de dados de forma rápida e eficiente com o foco em desempenho [McCreary and Kelly 2014]. Representam uma alternativa para modelar dados sem se preocupar com os padrões rígidos propostos pelo modelo relacional. O NoSQL tem uma estrutura distribuída e tolerante a falhas que se baseia na redundância de dados em vários servidores.

Existem vários modelos diferentes desses bancos de dados. Os principais modelos são chave-valor, orientado a coluna, orientado a documentos e orientado a grafos [Carniel et al. 2012]. Cada modelo possui vantagens e desvantagens. Para manipulação de dados estatísticos, frequentemente escritos mas raramente lidos, pode ser usado um banco de dados do tipo chave e valor ou orientado a documentos. Para uma aplicação com alta disponibilidade, onde a minimização da inatividade é fundamental, é recomendado utilizar um banco de dados orientado a coluna. Para alto desempenho de consultas mais complexas, recomenda-se o modelo orientado a grafos [Lóscio et al. 2011].

Cassandra. É um banco de dados orientado a coluna, criado para armazenar grandes quantidades de dados espalhados por vários servidores e, mesmo assim, oferece alta viabilidade de acesso a dados consistentes [Lakshman and Malik 2010]. Este banco de dados foi baseado na arquitetura do *Dynamo* da Amazon[1] e o no modelo de dados do *Bigtable* da Google[2]. Possui uma linguagem própria chamada CQL (*Cassandra Query Language*), muito semelhante ao SQL.

Uma das principais características do Cassandra é a sua escalabilidade horizontal, onde vários servidores possuem a mesma instância do banco de dados. A forma de distribuição é do tipo anel, não existe a abordagem do tipo "*master-slave*" como também não existe a distribuição de tarefas exclusivas para um nó específico, ou seja, todos os nós armazenam dados e podem executar as instruções requisitadas pela aplicação cliente. A grande vantagem de utilizar esta arquitetura é que o Cassandra não apresenta um ponto comum de falha, caracterizando a alta disponibilidade dos dados. Outra vantagem desta forma de distribuição é a possibilidade de adicionar mais nós no *cluster* em tempo de execução, caracterizando a elasticidade.

O nó responsável pela distribuição das requisições é chamado de *Coordinator Node*. Normalmente, o nó que apresenta este papel é o que a aplicação cliente se conecta. É importante lembrar que qualquer nó pode ser um *Coordinator Node*. As requisições da aplicação cliente também poderão ser executadas no próprio *Coordinator node*, visto que este nó também armazena dados.

A escrita no Cassandra passa por algumas etapas. Quando uma escrita ocorre, o Cassandra armazena os dados na memória RAM, em uma estrutura chamada *memtable*, enquanto a instrução de escrita é gravada em outra estrutura no disco chamada *commitlog*. O *commitlog* é importante para que, em caso de falha no *hardware*, não percam os registros que serão inseridos. Quando a *memtable* fica cheia acontece o processo de *flush*, em que os dados são descarregados para outra estrutura em disco chamada de *SSTable*. Após o *flush*, o *commitlog* é apagado. Para aumentar a tolerância a falhas, é necessário diminuir a memória da *memtable*, pois assim os dados são escritos mais rapidamente na *SSTable*, mas isto diminui o desempenho da consulta. A Figura 1 ilustra a estrutura da escrita e leitura do banco.

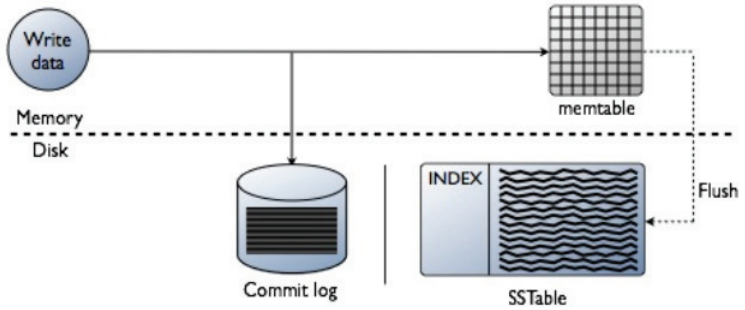


Figura 1. Estrutura da escrita do Cassandra [Enterprise 2017].

[1]http://aws.amazon.com/pt/documentation/dynamodb/ (Acessado em 27/04/2017)
[2]https://cloud.google.com/bigtable/ (Acessado em 27/04/2017)

2.3. Yahoo! Cloud Serving Benchmark (YCSB)

A ferramenta YCSB foi desenvolvida pela *Yahoo!* em conjunto com vários membros dos mais variados sistemas de armazenamento de dados (entre eles o Cassandra). Os objetivos dessa ferramenta são: criar uma ferramenta padrão para auxiliar a escolha do sistema de armazenamento de dados, facilitar a comparação de desempenho provendo as principais métricas e automatizar o ambiente de medição. É uma ferramenta de código aberto[3], desenvolvida em Java e com suporte para vários bancos de dados. Contudo, a principal característica do YCSB é a sua extensibilidade. Por meio desta ferramenta é possível criar cargas de trabalho através de arquivos de configuração e experimentar diferentes bancos de dados escrevendo novas classes aproveitando os métodos e as interfaces já desenvolvidos [Cooper et al. 2010].

O YCSB possui duas fases de execução. A primeira fase é denominada de *load*, onde os dados são inseridos. Nesta fase é definida a quantidade de dados inseridos no banco de dados. A outra fase é a *transaction*, que é executada através do parâmetro *run*, onde os dados são lidos e alterados. Normalmente, nesta fase são realizadas consultas e alterações numa quantidade menor que a inserida na fase anterior. Em ambas as fases é possível determinar a quantidade de usuários simultâneos, a carga de trabalho, o banco de dados e também definir uma vazão fixa. Após a execução de cada fase, o YCSB computa diversas métricas, entre elas: o tempo de execução, a vazão média, a latência das requisições e a quantidade de operações bem sucedidas e que falharam.

3. Trabalhos Relacionados

A avaliação de desempenho de bancos de dados é uma área de pesquisa ativa e que tem se intensificado recentemente devido ao aparecimento de modelos de bancos de dados alternativos ao relacional. Contudo, poucos trabalhos têm focado em analisar o consumo de energia desses sistemas.

Em [Abubakar et al. 2014] foi feita uma análise de desempenho em operações de inserção, leitura e atualização comparando 4 sistemas de armazenamento em um ambiente com recursos limitados, utilizando a ferramenta de *benchmark* YCSB. Em [Li and Manoharan 2013] também foi realizada uma análise comparativa de desempenho de sistemas de armazenamento de dados , com o foco em comparar os paradigmas relacional e não relacional. No total foram 6 bancos de dados não relacionais de diferentes modelos e arquiteturas e um banco de dados relacional. Para realizar os experimentos foi desenvolvido um *framework* específico para aquele ambiente. Embora tenham uma boa cobertura com diferentes bancos de dados, estes trabalhos não avaliam cenários distribuídos, que são essenciais para um melhor desempenho dos bancos analisados.

Os autores em [Cooper et al. 2010] propõem a ferramenta de *benchmark Yahoo! Cloud Serving Benchmark* (YCSB), descrita na seção 2.3. Uma análise de desempenho foi realizada em 4 sistemas de armazenamento de dados que estavam na forma distribuída em 6 servidores. Em [Maciel et al. 2014] foi realizada uma análise de desempenho e escalabilidade no *Sheepdog*, que é um sistema de armazenamento de dados distribuídos que fornece alta disponibilidade utilizando máquinas comuns. Embora tenham feito análises com bancos distribuídos, os trabalhos limitaram-se à avaliação de desempenho e não avaliaram aspectos de consumo de energia.

[3]https://github.com/brianfrankcooper/YCSB/wiki (Acessado em 27/04/2017)

Com foco apenas no consumo de energia, [Li et al. 2014] propõem uma nova estratégia de redução do consumo de energia através da diminuição da *Waiting Energy Consumption* (WEC). Este tipo de energia é desperdiçada quando alguns nós do *cluster* estão à "espera" de alguma atividade, devido a isto, não podem ser desligados. Outros trabalhos realizaram uma análise tanto de desempenho quanto do consumo de energia. Em [Gomes et al. 2016] foi feita uma análise comparativa de desempenho e consumo de energia de 3 sistemas de armazenamento de dados, porém desconsiderando cenários distribuídos. Em [Niemann 2015] e [Niemann 2016] foram propostos modelos computacionais utilizando *Queued Petri Nets* (QPN) para modelar o desempenho e o consumo de energia de um sistema de armazenamento de dados distribuídos, representado pelo Cassandra. Esses trabalhos abordaram cenários onde o *cluster* do Cassandra já estava povoado e eram realizadas operações de leitura e escrita sobre os dados já existentes. A Tabela 1 apresenta uma visão geral de cada trabalho relacionado.

Tabela 1. Visão geral dos trabalhos relacionados.

Trabalho	Desempenho	Energia	Dados Distribuídos
[Cooper et al. 2010]	Sim	Não	Sim
[Li and Manoharan 2013]	Sim	Não	Não
[Li et al. 2014]	Não	Sim	Sim
[Abubakar et al. 2014]	Sim	Não	Não
[Maciel et al. 2014]	Sim	Não	Sim
[Niemann 2015]	Sim	Sim	Sim
[Gomes et al. 2016]	Sim	Sim	Não
[Niemann 2016]	Sim	Sim	Sim

Diferente dos trabalhos anteriores, esta pesquisa realiza uma análise integrada de desempenho de consumo de energia de sistemas de armazenamento de dados distribuídos em vários servidores durante a inserção de pequenas e grandes quantidades de dados. O sistema de armazenamento escolhido foi o Cassandra. Esta escolha foi feita pelo fato do Cassandra estar entre os 10 bancos de dados mais utilizado, conforme o *DBEngines*[4] e por ele ter sido projetado para ser distribuído em várias máquinas. Foram abordados diversos cenários variando o número de nós no *cluster* do Cassandra como também a quantidade de operações de inserção.

4. Metodologia

Para realizar a análise de desempenho e consumo de energia de sistemas de armazenamento de dados distribuídos no processo de inserção de dados, são analisados dois fatores: a quantidade de nós no *cluster* e a quantidade de operações de inserção. Os fatores são variados em níveis. A combinação dos níveis dos fatores indicam a quantidade de cenários que devem ser analisados.

A primeira etapa é a montagem de um ambiente isolado, sem interferências de fatores externos. Em seguida, realiza-se os experimentos conforme os fatores e os níveis escolhidos. Nesta etapa, é recomendável utilizar ferramentas de *benchmarks*, que executam tarefas bem definidas no sistema e fornecem várias métricas. A ferramenta de

[4]http://db-engines.com/en/ranking (Acessado em 27/04/2017)

benchmark YCSB é uma das mais utilizadas para analisar o comportamento de sistemas de armazenamento de dados, fornecendo as principais métricas. Também é responsável por gerar carga de trabalho. Já para obter métricas de consumo de energia, pode-se utilizar dispositivos de medição, entre eles, o *Wattsup Meter*[5]. Este dispositivo registra métricas de consumo de energia, inserindo os valores em arquivos textos. Esses valores são capturados em intervalo de tempo ajustável. Quanto menor o intervalo de tempo, mais preciso são os valores do consumo de energia. O *Wattsup Meter* é conectado entre o estabilizador e a rede elétrica, também servindo como condutor de energia ao sistema. A Figura 2 ilustra a distribuição dos componentes no ambiente de medição.

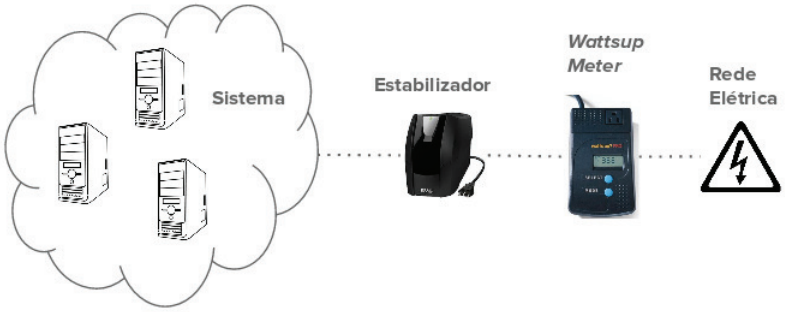


Figura 2. Visão geral dos componentes do ambiente de medição.

4.1. Processo de Obtenção das Amostras

O primeiro passo da metodologia consiste na preparação dos nós do *cluster* do Cassandra, onde todos os *logs*, *commitlogs* e *caches* gerados pelo *cluster* são apagados. Em seguida, realiza-se a remoção dos registros gerados no *Wattsup Meter*, que armazena, sem interrupções, as informações do consumo de energia a cada 1 segundo, por exemplo.

Após os procedimentos de preparo dos componentes, se executa a fase *load* da ferramenta YCSB, ou seja, inicia-se a inserção dos dados no *cluster* do Cassandra. Ao término da inserção de todos os dados, os registros gerados no *Wattsup Meter* são armazenados e para cada amostra é gerada uma tabela em um arquivo texto. A partir dessas informações pode-se computar o consumo de energia da amostra. Em seguida, registra-se as métricas obtidas no YCSB: tempo de execução, vazão e latência média. Este procedimento é repetido ao menos 30 vezes para que se possa ter confiabilidade nos resultados obtidos. Por último, realiza-se a análise de resultado fornecendo suporte para as conclusões dos experimentos. A Figura 3 ilustra a sequência de passos do processo de obtenção das amostras.

[5]https://www.wattsupmeters.com/secure/index.php (Acessado em 27/04/2017)

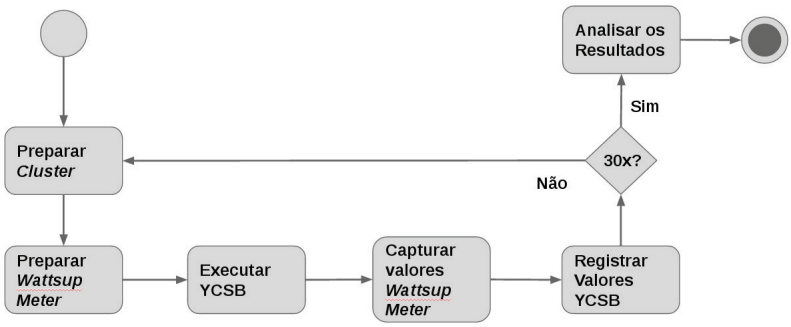


Figura 3. Metodologia utilizada na obtenção das métricas de desempenho e consumo de energia.

5. Estudo de Caso

Esta seção apresenta um conjunto de experimentos realizados com o propósito de analisar o desempenho e o consumo de energia do banco de dados Cassandra no processo de inserção de dados. O fator quantidade de nós no *cluster* tem 4 níveis: 1, 2, 4 e 6 nós. Já o fator quantidade de operações de inserção apresenta 3 níveis: 10.000 (10K), 100.000 (100K) e 1.000.000 (1M) operações. Os registros gerados pelo YCSB eram do tipo texto e inseridos em uma tabela com 10 colunas. Cada registro tinha 1KB de informação. Logo foram inseridos 10MB, 100MB e 1GB de dados para os cenários 10K, 100K e 1M, respectivamente. Foram utilizados 10 usuários simultâneos.

As métricas de desempenho, obtidas pelo YCSB, são: o tempo de execução para inserir o todos os registros (segundos), a vazão do sistema (operação por segundo) e a latência média (milissegundos). O consumo de energia foi obtido ao final do tempo de execução do experimento e a unidade de medida foi *joules*. Para cada combinação dos níveis dos fatores (12 combinações) foram coletadas 33 amostras, as 3 primeiras amostras eram descartadas, pois foi observado através do tempo de execução que elas atingiam valores bem acima das demais, sendo consideradas como período de *warm-up* do sistema de armazenamento. O intervalo de confiança para a média, considerando 95% de confiança, foi calculado e, na maioria dos casos, os intervalos não se sobrepõem, mostrando que os valores das médias são estatisticamente diferentes. Por isso, os intervalos de confiança não serão mostrados nos gráficos, mas os casos em que não foi possível definir a diferença estatística serão pontuados no texto.

5.1. Ambiente

O ambiente montado em laboratório foi composto por quatro computadores, cada um com processador Intel core i5, 8GB de memória RAM, 500GB de espaço em disco e com sistema operacional Ubuntu 14.04 LTS. Todos conectados através de uma rede local *Gigabit Ethernet*. A ferramenta de *benchmark* YCSB foi executada em um computador dedicado. Já os nós do *cluster* do Cassandra eram máquinas virtuais (VM), todas idênticas com 2GB de memória RAM, 30GB de espaço em disco e com sistema operacional Ubuntu 14.04 LTS. Nos três computadores restantes eram hospedados até dois nós (2 VMs) de acordo com o cenário. Logo, para os experimentos com 2, 4 e 6 nós foram utilizados 1, 2 e 3 computadores, respectivamente. Apenas as máquinas físicas necessárias para cada *cluster* eram ligadas, por exemplo, para o *cluster* com 2 nós, apenas 1 máquina física (com as 2 VMs hospedadas) era utilizada, as demais permaneciam desligadas.

Foram utilizados 2 estabilizadores, um alimentava o computador com a aplicação cliente (YCSB) junto com os demais componentes: monitor e *switch*. Já os demais computadores estavam ligados ao segundo estabilizador que, por sua vez, se conectava ao *Wattsup Meter* o qual estava ligado à rede elétrica. Esta abordagem foi feita com o objetivo de isolar todo o sistema de armazenamento para obtenção do consumo de energia. A Figura 4 ilustra uma visão geral dos componentes do ambiente de medição.

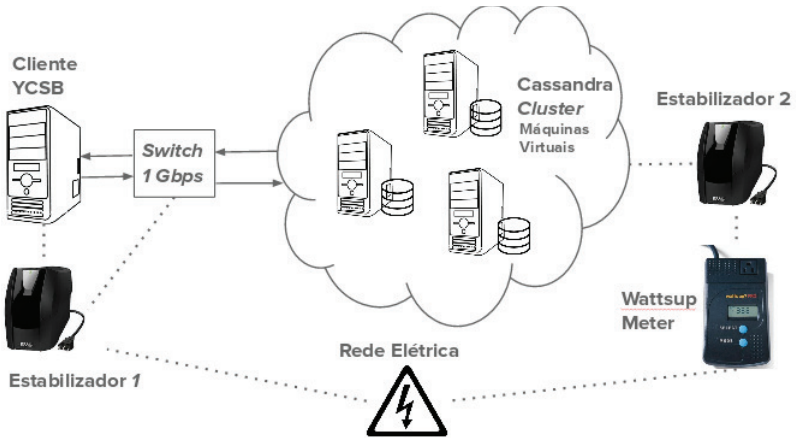


Figura 4. Visão geral do ambiente de experimentação.

5.2. Análise de Desempenho

Foram calculados valores estatísticos como a média, desvio padrão e intervalo de confiança para cada cenário. Além disso, em algumas situações foi realizado Teste-t emparelhado com o objetivo de verificar se as amostras são estatisticamente equivalentes. Cada métrica será abordada separadamente.

Tempo de Execução. A Figura 5 mostra o tempo de execução de acordo com a quantidade de registros inseridos. Para 10K operações de inserção, é possível observar que o tempo permaneceu constante mesmo com o aumento da quantidade de nós. Já para 100K, é possível observar que o tempo de execução para 1 e 2 nós são estatisticamente iguais. Isto pode ser observado através do teste-t emparelhado cujo resultado é mostrado na Tabela 2. Dessa forma, não podemos rejeitar a hipótese nula, a qual indica a igualdade dos tempos de execução. Contudo, a partir de 4 nós, é possível observar uma queda no tempo de execução. Já para 1M de operações, o tempo de execução para 2 nós é um pouco maior do que para 1 nó. Também foi realizado teste-t (Tabela 2) nesta situação, que indicou que os valores são diferentes, visto que o valor-p é menor que o α. Este comportamento pode ser justificado pelo compartilhamento de recursos entre as VMs, em razão de utilizar um único computador para hospedar os 2 nós e com isso os recursos como a placa de rede e disco são compartilhados entre as duas VMs.

Tabela 2. Teste-T no tempo de execução entre *clusters* com 1 e 2 nós.

Qntd Operações	Valor-p	α	Diferença entre as médias
100K	0,1397	0,05	0,54 seg
1M	6,147E-008	0,05	7,35 segs

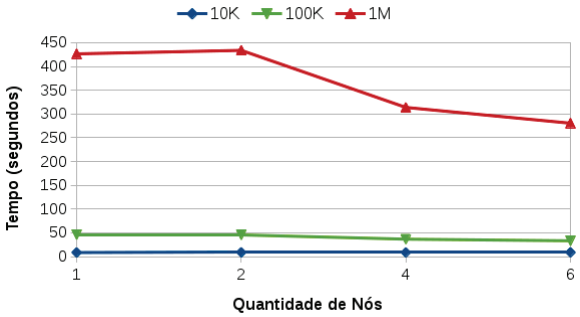


Figura 5. Tempos de execução de acordo com o número de registros inseridos.

Vazão. A Figura 6 mostra o comportamento da vazão em relação à quantidade de operações de inserção no *cluster*. É possível observar que para 10K operações, a vazão apresentou um comportamento uniforme, no entanto para as demais situações foi constatado um crescimento considerável a partir de 4 nós. Em todos os cenários, a vazão permaneceu similar para 1 e 2 nós. Para as inserções de 100K e 1M, houve um aumento significativo, principalmente para 1 milhão de registros, este último apresentou um crescimento de mais de 50% em relação à um nó. Este crescimento pode ser comprovado na Figura 7, que indica a proporção da vazão para 2, 4 e 6 nós em relação a 1 nó. Pode-se concluir que o aumento do *cluster* de 1 para 2 nós, não foi uma estratégia interessante em razão da vazão não apresentar nenhum aumento significativo em nenhuma das situações analisadas. Além de consumir mais recursos computacionais.

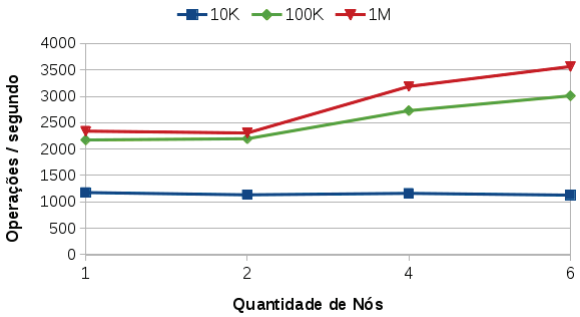


Figura 6. Comportamento da Vazão.

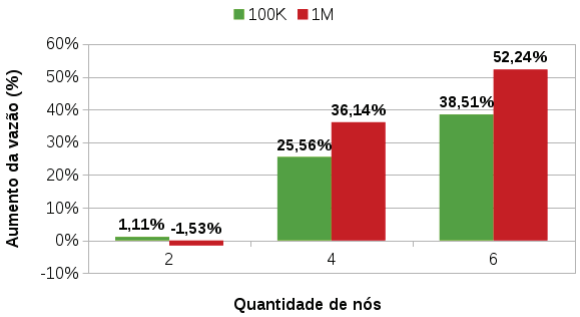


Figura 7. Comparação da vazão em relação a 1 nó para 100K e 1M de operações.

Latência. A latência é o intervalo de tempo entre o envio da requisição para inserir o registro e a chegada da resposta do sistema de armazenamento para a aplicação cliente. A Figura 8 ilustra o comportamento da latência. É possível observar que para 10K operações de inserção, a latência apresentou um comportamento uniforme, onde os tempos tinham diferença média de 2%. No entanto, quando o *cluster* era composto por 2 nós, a latência teve um aumento de 5% em relação a 1 nó. Para 100K e 1M operações de inserção, é constatada uma queda a partir de 4 nós. Esta queda representou uma diminuição de aproximadamente 26,8% quando o *cluster* era composto com 4 nós e 35% para 6 nós, em relação a 1 nó. É possível observar que para as duas maiores quantidades de operações de inserção (100K e 1M), os valores médios quantificados ficaram sobrepostos, indicando que, para os cenários analisados, um *cluster* Cassandra com 4 nós oferece a melhor razão entre latência e uso de recursos.

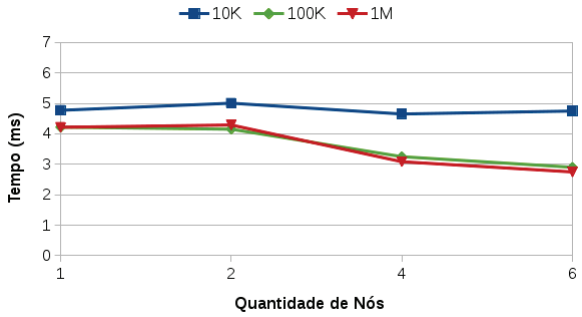
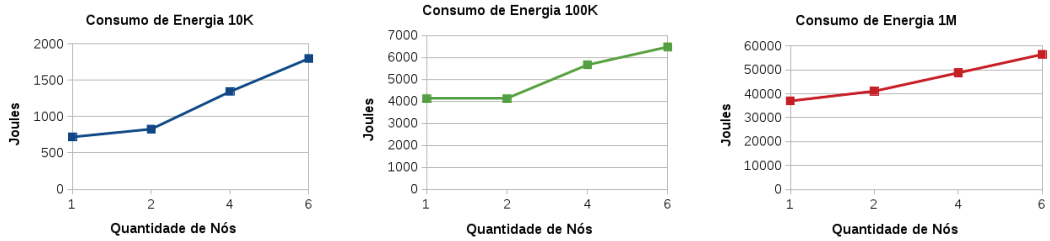


Figura 8. Análise Latência.

5.3. Análise do Consumo de Energia

As Figuras 9(a), 9(b), 9(c) ilustram o consumo de energia de acordo com o aumento do número de nós no *cluster* para, respectivamente, 10K, 100K e 1M operações de inserção. É possível obsevar que para 10K (Figura 9(a)), houve um aumento significativo do consumo de energia para 4 e 6 nós. Para 100K (Figura 9(b)), o consumo de energia para 1 e 2 nós foi similar. Mesmo utilizando 2 nós hospedados em um único computador, o consumo de energia não sofreu alterações. Contudo para 4 e 6 nós houve um aumento significativo, visto que a quantidade de computadores aumentou para 2 e 3 respectivamente.



(a) 10K (b) 100K (c) 1M

Figura 9. Consumo de Energia para 10K (a), 100K (b) e 1M (c)

A Figura 10 mostra o aumento do consumo de energia em relação a 1 nó. É constatado um aumento significativo de 87% e 150% no consumo de energia para realizar 10K

operações de inserção em 4 e 6 nós respectivamente, devido a adição de mais computadores no ambiente (2 computadores para 4 nós e 3 computadores para 6 nós). Já para 100K e 1M, os aumentos foram próximos. Para 4 nós o aumento foi 37% e 32% para 100K e 1M, respectivamente. Enquanto que para 6 nós o aumento foi de 57% e 53% nas mesmas condições.

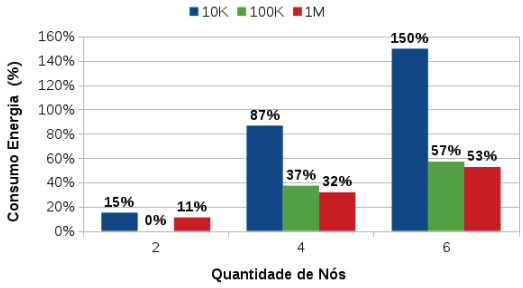


Figura 10. Aumento do consumo de energia em relação a 1 nó.

5.4. Discussão

Analisando de forma conjunta o desempenho e o consumo de energia dos cenários propostos é possível observar que o aumento de 1 para 2 nós no *cluster* não apresentou melhorias significativas para o desempenho. Um dos fatores para este comportamento é o compartilhamento de recursos, como placa de rede e disco, já que as 2 máquinas virtuais estavam hospedadas em um único computador. Observa-se que a introdução de novos nós virtuais sob um mesmo computador físico, pode não apresentar bons resultados. Isso decorre do fato de que os nós virtuais compartilham recursos físicos de entrada e saída que são críticos em sistemas virtualizados. Por outro lado, quando utilizavam-se 2 nós, o aumento do consumo de energia em relação a 1 nó foi de 15% para realizar 10K operações de inserção e 11% para inserir 1M de registros.

Quando o *cluster* era composto por 4 e 6 nós, foi constatada uma melhoria significativa no desempenho em relação a 1 nó, principalmente para realizar inserção dos dois maiores conjuntos de dados, 100K e 1M. Entre as métricas de desempenho, pode-se destacar a vazão, que obteve um aumento de mais de 50% quando inseridos 1M de registros em um *cluster* composto por 6 nós. Entretanto, houve um aumento de 150% no consumo de energia quando inseridos 10K registros. Além disso, ocorreu um aumento, em relação a 1 nó, de 57% e 53% no consumo de energia para inserir, respectivamente, 100K e 1M de registros. A Tabela 3 mostra uma análise na eficiência do desempenho e do consumo de energia para 100K e 1M em um *cluster* composto por 4 e 6 nós em relação a 1 nó. Não foi pontuado 10K operações nas métricas de desempenho, pois não houve alterações significativas, entretanto foi adicionado na métrica de consumo de energia. Os valores negativos indicam uma redução em relação ao valor encontrado em 1 nó.

Baseados nos valores da Tabela 3, pode-se chegar a conclusão da quantidade ideal de nós de acordo com o número de operações de inserção realizadas. O número ideal de nós para inserir 10K registros no *cluster* é 1, pois o desempenho não apresentou grandes melhorias para *clusters* maiores e o consumo de energia nesta situação foi o menor entre os cenários avaliados. Já para 100K e 1M, o número de nós ideal foi 4, pois foi notado um aumento considerável no desempenho enquanto que o consumo de energia atingiu valores

Tabela 3. Análise das métricas para 4 e 6 nós em relação a 1 nó.

Métricas	Qtd Operações	4 nós	6 nós
Tempo de Execução	100K	-20,42%	-27,88%
	1M	-26,42%	-34,20%
Vazão	100K	25,56%	38,51%
	1M	36,14%	52,24%
Latência	100K	-22,95%	-31,24%
	1M	-26,80%	-34,81%
Consumo de Energia	10K	86,67%	150,00%
	100K	37,21%	56,98%
	1M	31,77%	52,56%

abaixo do *cluster* com 6 nós. O *cluster* com 6 nós obteve o melhor desempenho, contudo, houve um aumento significativo do consumo de energia, em todos os casos o aumento foi superior a 50%. Então, se o objetivo é apenas melhorar o desempenho o *cluster* ideal é com 6 nós. No entanto, para avaliação integrada de consumo de energia e desempenho, o melhor seria com 4 nós devido ao aumento registrado com 6 nós em relação a 4 nós no consumo de energia.

6. Conclusão

Devido ao grande aumento de dados gerados, a necessidade por um sistema de armazenamento de grande desempenho e que impacta pouco o meio ambiente está aumentando. Neste trabalho foi realizada uma análise integrada de desempenho e consumo de energia. Como resultado, mostrou-se que para algumas situações não é necessário aumentar os recursos computacionais, pois o desempenho não apresenta melhorias significativas enquanto que o consumo de energia do sistema aumenta consideravelmente.

Como trabalhos futuros, pretende-se desenvolver modelos formais de desempenho e consumo de energia desses ambientes. Com esses modelos validados através dos resultados das medições, é possível obter métricas para cenários não analisados em laboratório, economizando tempo e otimizando o planejamento de recursos.

Agradecimentos

Os autores gostariam de agradecer à FACEPE e ao CNPq pelo suporte financeiro a esta pesquisa.

Referências

Abubakar, Y., Adeyi, T. S., and Auta, I. G. (2014). Performance evaluation of NoSQL systems using YCSB in a resource austere environment. In *Performance Evaluation*, volume 7.

Carniel, A. C., de Aguiar Sá, A., Brisighello, V. H. P., Xavier, M., Ribeiro, R. B., Ciferri, R. R., and de Aguiar Ciferri, C. D. (2012). Query processing over data warehouse using relational databases and nosql. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pages 1–9.

Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. (2010). Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 143–154. ACM.

De Diana, M. and Gerosa, M. A. (2010). Nosql na web 2.0: Um estudo comparativo de bancos não-relacionais para armazenamento de dados na web 2.0. In *IX Workshop de Teses e Dissertações em Banco de Dados*, volume 9.

Enterprise, D. (2017). A brief introduction to apache cassandra. Disponível em https://academy.datastax.com/resources/brief-introduction-apache-cassandra (Acessado em 27/04/2017).

Gomes, C., Tavares, E., and Junior, M. N. d. O. (2016). Energy consumption evaluation of NoSQL DBMSs.

Koomey, J. (2011). Growth in data center electricity use 2005 to 2010. *A report by Analytical Press, completed at the request of The New York Times*, 9.

Lakshman, A. and Malik, P. (2010). Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, 44(2):35–40.

Li, T., Yu, G., Liu, X., and Song, J. (2014). Analyzing the waiting energy consumption of NoSQL databases. pages 277–282. IEEE.

Li, Y. and Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. In *Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on*, pages 15–19. IEEE.

Lóscio, B. F., OLIVEIRA, H. R. d., and PONTES, J. C. d. S. (2011). NoSQL no desenvolvimento de aplicações web colaborativas. In *VIII SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, Paraty, RJ: SBC*.

Maciel, P., Matos, R., Callou, G., Silva, B., Barreto, D., Araujo, J., Araujo, J., Alves, V., and Worth, S. (2014). Performance evaluation of sheepdog distributed storage system. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 3370–3375. IEEE.

McCreary, D. and Kelly, A. (2014). Making sense of NoSQL. *Shelter Island: Manning*, pages 19–20.

Niemann, R. (2015). Evaluating the performance and energy consumption of distributed data management systems. In *Global Software Engineering Workshops (ICGSEW), 2015 IEEE 10th International Conference on*, pages 27–34. IEEE.

Niemann, R. (2016). Towards the prediction of the performance and energy efficiency of distributed data management systems. In *Companion Publication for ACM/SPEC on International Conference on Performance Engineering*, pages 23–28. ACM.

NRDC (2015). America's data centers consuming and wasting growing amounts of energy. Disponível em https://www.nrdc.org/resources/americas-data-centers-consuming-and-wasting-growing-amounts-energy (Acessado em 27/04/2017).

Venkatraman, A. (2012). Global census shows datacentre power demand grew 63% in 2012. Disponível em http://www.computerweekly.com/news/2240164589/Datacentre-power-demand-grew-63-in-2012-Global-datacentre-census (Acessado em 27/04/2017).