UNICEUB - FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS

CLUSTERS

Aline Sayuri Hashimura
Igor Gomes
Samuel Costa Carvalho
Sandro Vieira

RESUMO

Os clusters são um conjunto de computadores onde dois ou mais computadores trabalham de forma conjunta e executam tarefas complexas como se fossem um único sistema. Todo sistema de cluster é utiliza o conceito de paralelismo como principal ferramenta, onde os nós trabalham interconectados e cooperando uns com os outros. Hoje em dia existe uma grande variedade de clusters com finalidades diferentes e pré-requisitos específicos. São alguns exemplos os clusters de alta disponibilidade, clusters de alto desempenho e clusters de balanceamento de carga. As duas implementações mais conhecidas são o beowulf que explora a computação paralela para o processamento de grandes cálculos, e mosix, adota que uma abordagem simétrica.

PALAVRAS-CHAVE

Clusters, paralelismo, alta disponibilidade, alto desempenho, balanceamento de carga, beowulf, mosix.

ABSTRACT

Clusters are a set of computers where two or more computers work together and perform complex tasks as if they were a single system. Every cluster system uses the concept of parallelism as the main tool, where nodes work interconnected and cooperating with one another. Nowadays there is a great variety of clusters with different purposes and specific prerequisites. Examples include high-availability clusters, high-performance clusters, and load-balancing clusters. The two most well-known implementations are the beowulf that exploits parallel computing for processing large calculations, and mosix, adopts a symmetric approach.

KEY WORKS

Clusters, parallelism, high availability, high performance, load balancing, beowulf, mosix.

1. INTRODUÇÃO

Atualmente no ano de 2018 em relação a computação em vários âmbitos, como unidade de pesquisa e indústria, onde há a necessidade de execução de muitas tarefas com um grande volume de informações e de trabalho ao mesmo tempo, assim causando a demora excessiva da finalização das tarefas, e em alguns casos podendo acarretar o travamento e a danificação do sistema.

Com a finalidade de ajudar no processamento do grande volume de tarefas complexas foi desenvolvido os clusters, uma estrutura na qual possibilita um melhor funcionamento e agilidade na execução dos processos de alta complexibilidade.

Neste artigo iremos explicar o que são clusters, alguns tipos de clusters que existem e suas características, as implementações mais conhecidas como o beowulf e mosix, o impacto de

sua utilização, as tecnologias em evidência e a descrição de uma implementação para a simulação de um dos tipos cluster que serão apresentados.

2. ENTENDENDO O QUE SÃO CLUSTERS

Clusters são um conjunto de computadores, podendo ser denominados nós, onde dois ou mais computadores trabalham de forma conjunta e executam tarefas como se fossem um único sistema, as tarefas são divididas entre si para que sejam processados e executados de forma simultânea. Para a interconexão dos nós é preferivelmente que seja feita por uma tecnologia na qual a rede seja conhecida como a ethernet, com o intuito de facilitar a manutenção e controle de custos.

Os clusters surgiram na década de sessenta pela IBM (International Business Machines) com o pensamento de interligar os mainframes, onde permitia que o hardware, sistema operacional, middleware e softwares de gerenciamento de sistemas promovessem uma notável melhora na performance e custo. Mas foi apenas em meados da década de oitenta, onde surgiu os microprocessadores com alto desempenho, que os clusters começaram a ganhar popularidade.

3. ABSTRAINDO A ARQUITETURA DOS CLUSTERS

Basicamente todo o sistema de cluster utiliza o conceito de paralelismo como principal ferramenta, onde os nós trabalham interconectados e cooperando uns com os outros, assim formando um recurso computacional único e integrado, deixando os usuários que vejam o sistema como um único sistema.

Podemos exemplificar a arquitetura utilizando uma máquina como o nó mestre, este é responsável por receber as requisições e realizar os comandos direcionados ao tipo de clusterização que esteja sendo utilizado, assim o nó mestre é responsável por gerenciar a utilização dos outros nós conectados ao clusters, chamados de nós escravos.

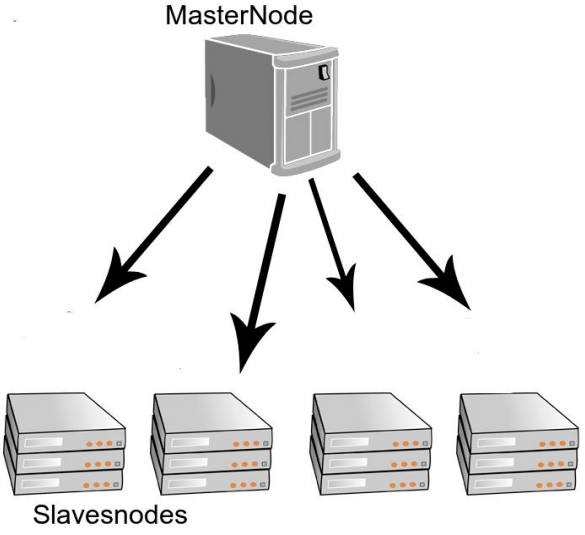

Figura 1: Abstração da arquitetura de clusters

4. TIPOS DE CLUSTERS

Hoje em dia existe uma grande variedade de clusters com finalidades diferentes e pré-requisitos específicos. São alguns exemplos:

4.1. CLUSTERS DE ALTA DISPONIBILIDADE

Também conhecidos como clusters de *Failover*, tem o objetivo de aumentar a disponibilidade dos serviços que sustentam. Para isso, se utilizam da redundância, ou seja, em caso de falha de algum dos nós do cluster, os demais nós podem assumir o serviço e manter seu pleno funcionamento.

A utilização deste modelo permite ao sistema uma maior tolerância a falhas. Em geral, a operação desses clusters requer ferramentas de monitoração e sistemas de *Nobreak*.

4.2. CLUSTERS DE ALTO DESEMPENHO

Em geral, são utilizados para a execução de tarefas que requerem alto poder computacional, por exemplo grandes cálculos em pesquisas científicas, aplicação de modelos matemáticos na previsão meteorológica, dentre outras aplicações. O nó mestre deve realizar distribuição de tarefas entre os nós, de forma a se obter o maior desempenho possível de cada um.

4.3. CLUSTERS DE BALANCEAMENTO DE CARGA

O principal objetivo deste modelo é distribuir as tarefas de maneira homogênea entre os nós, de forma a evitar a sobrecarga. É possível que os nós sejam heterogêneos, isto é, possuem configurações diferentes entre si.

Cabe ao nó mestre a monitoração da carga sobre cada nó, e realizar a distribuição de tarefas entre estes levando em consideração sua configuração. É possível realizar o balanceamento de carga entre as máquinas com maior e menor poder de processamento para executar tarefas distintas.

5. TIPOS DE IMPLEMENTAÇÃO

As duas implementações mais conhecidas são o **Beowulf** e **Mosix**. Os dois modelos são analisados a seguir mostrando suas principais diferenças, vantagens e desvantagens.

5.1. BEOWULF

O projeto **beowulf** foi desenvolvido pela NASA, com o objetivo de explorar a computação paralela para o processamento de grandes cálculos. O cluster utiliza os microcomputadores pessoais não especializados, permitindo grande escalabilidade e boa relação custo-benefício. Hoje, os clusters **beowulf** são amplamente difundidos.

Outras vantagens oferecidas são a flexibilidade, pois é possível incorporar nós com configurações diferentes ao cluster, desde que executem o sistema operacional padrão. Tolerante a falhas, por permitir fácil substituição dos nós e por ser um software aberto, a comunidade pode rapidamente identificar falhas e providenciar as correções.

O modelo é baseado no sistema operacional linux, em uma estrutura hierárquica. O nó mestre executa o *middleware* e o gerenciamento do cluster. Aos nós escravos, basta que executem o sistema operacional padrão.

5.2. MOSIX

Em contraposição ao modelo hierárquico, há o sistema **mosix**, que adota uma abordagem simétrica. De acordo com Tanenbaum, “esse sistema tenta prover a imagem de um sistema único de um cluster, o que significa que, para um processo, um computador de cluster oferece a transparência de distribuição definitiva porque parece ser um único computador”.

O alto grau de transparência é conseguido através da permissão para que cada nó tenha o conhecimento sobre a disponibilidade de recursos dos demais nós, e possa realizar a transferência de processos. Desta forma, o usuário pode iniciar um processo em algum dos nós e o próprio cluster realizar a transferência desse processo, de maneira transparente para o usuário.

O cluster **mosix** oferece, como principais benefícios, a escalabilidade, isto é, a capacidade de facilmente aumentar o poder computacional do cluster e a flexibilidade, pois é possível agregar nos clusters nós com diferentes arquiteturas.

6. VANTAGENS E DESVANTAGENS

A utilização de clusters de computadores tem inúmeras vantagens. Abaixo segue as principais vantagens no uso dos clusters:

- **Expansibilidade**: A utilização de clusters de computadores deixa o sistema computacional facilmente expansível, uma vez que, para aumentar o poder de processamento, basta apenas incluir um novo nó ao cluster.

- **Alta disponibilidade:** Um nó que está desativado não prejudica o sistema como um todo, levando em consideração que também, para a manutenção não é necessário tirar o cluster inteiro do funcionamento, apenas o nó com problemas.

- **Balanceamento de carga:** Cluster de computadores também podem ser formados de forma heterogêneas (com máquinas de configurações diferentes), sendo assim é possível realizar o balanceamento de carga para as máquinas com maior e menor processamento para executar tarefas distintas.

- **Baixo custo:** Como os clusters de computadores podem ser configurados e utilizados por computadores convencionais, o custo não fica limitado a apenas um único fornecedor, dessa forma clusters de computadores são bem mais econômicos que sistemas específicos.

Como toda tecnologia, clusters de computadores também possuem desvantagens. Abaixo segue as principais desvantagens no uso dos clusters:

- **Gargalos de troca de informações:** Como a comunicação de clusters de computadores ocorrerem por uma tecnologia de rede, a troca de informação se transforma no principal gargalo, uma vez que a transmissão de rede é bem lenta se comparada à troca de informação com um barramento de um sistema de memória compartilhada, entretanto, é possível realizar ajustes de granulosidade para diminuir esse problema.

- **Manutenção de equipamento:** Por o cluster ser facilmente expansível, o sistema computacional pode se tornar muito grande, e a manutenção do sistema pode se tornar uma tarefa imensamente grande, pois cada máquina em um clusters devem ter todos os seus componentes em perfeito estado de funcionamento.

- **Monitoração dos nós:** Monitorar as informações trocadas em cada nó pode ser um problema dependendo como foi configurado o cluster, levando em consideração a expansibilidade do cluster.

7. TECNOLOGIAS EM EVIDÊNCIA

7.1. ROCKS

Um dos principais softwares para clustering é o Rocks, ele é um dos softwares mais utilizados pelos usuários do TOP500 (2018), que é um ranking dos 500 supercomputadores mais poderosos do mundo.

O Rocks é uma software de cluster, open-source, baseado em Linux, que permite que os usuários finais criem facilmente clusters computacionais.

Rocks foi inicialmente baseado na distribuição Red Hat Linux , no entanto versões modernas do Rocks foram baseadas no CentOS , com um instalador Anaconda modificado. Isso simplifica a instalação em massa em muitos computadores. O Rocks inclui muitas ferramentas (como MPI) que não fazem parte do CentOS, mas são componentes integrais que transformam um grupo de computadores em um cluster.

Desde maio de 2000, o grupo Rocks vem lidando com as dificuldades de implantar clusters gerenciáveis. O objetivo principal deles é facilitar o uso dos clusters, ou seja, tornar os clusters mais fáceis de implantar, gerenciar, atualizar e dimensiona.

Em outubro de 2010, o Rocks era usado por organizações acadêmicas, governamentais e comerciais, empregadas em 1.376 clusters, em todos os continentes, exceto na Antártida. O maior agrupamento acadêmico registrado, com 8632 CPUs, é o GridKa , operado pelo Instituto de Tecnología de Karlsruhe em Karlsruhe, Alemanha. Há também vários clusters com menos de 10 CPUs, representando os estágios iniciais na construção de sistemas maiores, além de serem usados em cursos de design de cluster. Essa escalabilidade fácil foi um dos principais objetivos no desenvolvimento do Rocks, tanto para os pesquisadores envolvidos.

7.2. SLURM

Um outro software muito utilizado para clustering é o Slurm Workload Manager ou Slurm. O Slurm é um agendador de tarefas gratuito e open-source para Linux e kernels similares ao Unix, usado por muitos dos supercomputadores e clusters de computadores do mundo. Ele fornece três funções principais. Primeiro, ele atribui acesso exclusivo e/ou não exclusivo a recursos (nós de computador) aos usuários por algum tempo para que eles possam executar o trabalho. Em segundo lugar, ele fornece uma estrutura para iniciar, executar e monitorar o trabalho (normalmente, um trabalho paralelo, como o MPI) em um conjunto de nós alocados. Finalmente, ele arbitra a contenção de recursos gerenciando uma fila de tarefas pendentes.

Slurm é o gerente de carga de trabalho em cerca de 60% dos supercomputadores TOP500, incluindo o Tianhe-2 que, até 2016, era o computador mais rápido do mundo. O Slurm usa um algoritmo de melhor ajuste baseado no agendamento da curva de Hilbert ou na topologia da rede de árvore fatiada para otimizar a localidade das atribuições de tarefas em computadores paralelos.

O design do Slurm é muito modular com cerca de 100 plugins opcionais. Em sua configuração mais simples, ele pode ser instalado e configurado em alguns minutos. Configurações mais

sofisticadas fornece integração de banco de dados para contabilidade, gerenciamento de limites de recursos e priorização de carga de trabalho.

8. IMPLEMENTAÇÃO

Com base em todos os conhecimentos adquiridos até aqui, este experimento busca expor um modelo de clusters com foco no balanceamento de carga, expondo assim uma estrutura construída sobre containers utilizando a tecnologia *docker* e sua ferramenta *docker-compose* na criação desta estrutura, utilizando o servidor *nginx* para compor o papel do nó mestre direcionando requisições aos seus nós escravos representados por containers nesta estrutura. Todo código necessário para realizar a reprodução deste experimento está disponível no site oficial do github através do diretório ***sancozta/dockercomposer*** , basta seguir as instruções do arquivo readme.md utilizando a ferramenta *docker-compose*.

Descrevendo o procedimento, utilizaremos o docker para criar 4 containers, sendo que um deles utilizaremos como nó mestre, todas as requisições passaram primeiro por este container, este container irá direcionar a requisição para um dos outros 3 containers restantes, desta forma distribuído a carga das requisições e assim consequentemente o processamento. No experimento ainda colocamos um quinto containers que fará o papel de um banco de dados, caso nosso processamento necessite interações com o banco de dados. A representação do experimento pode ser visualizada na imagem abaixo.

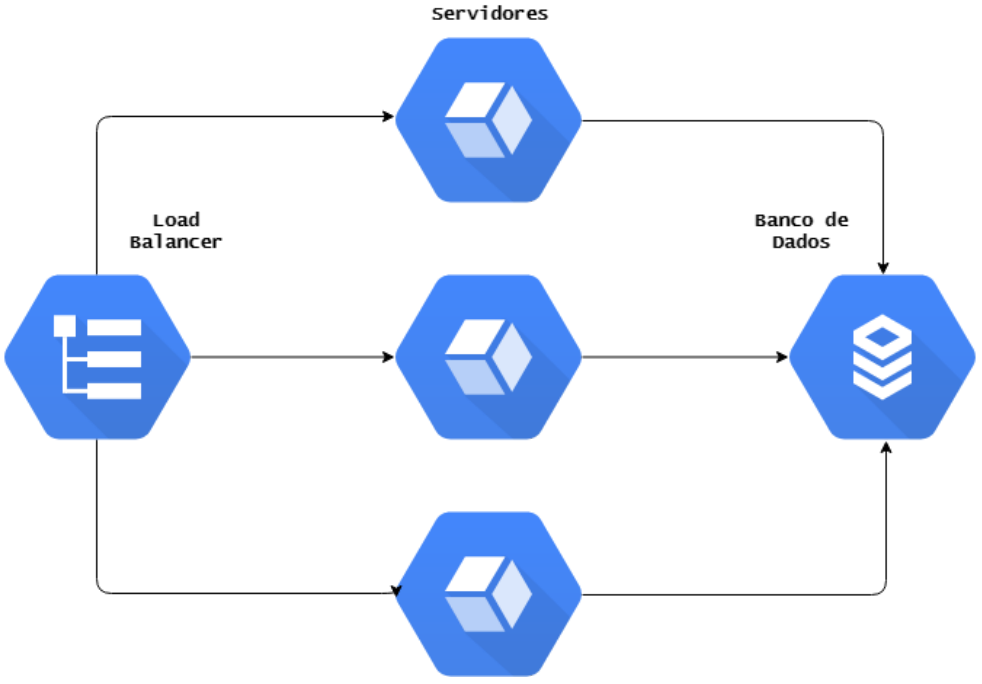

Figura 2: Componentes da arquitetura implementada no experimento

Para realizar a simulação, o experimento utiliza o *nginx* no container mestre, utilizando as configurações do *nginx*, realizamos os procedimentos necessários para implementar o balanceamento de carga. Os containers que irão manter a aplicação terão o *nodejs* como base para poder executar nosso código. O container que fará o papel do banco de dados, terá o *mongoDB*. Podemos ver isso com clareza na figura abaixo.

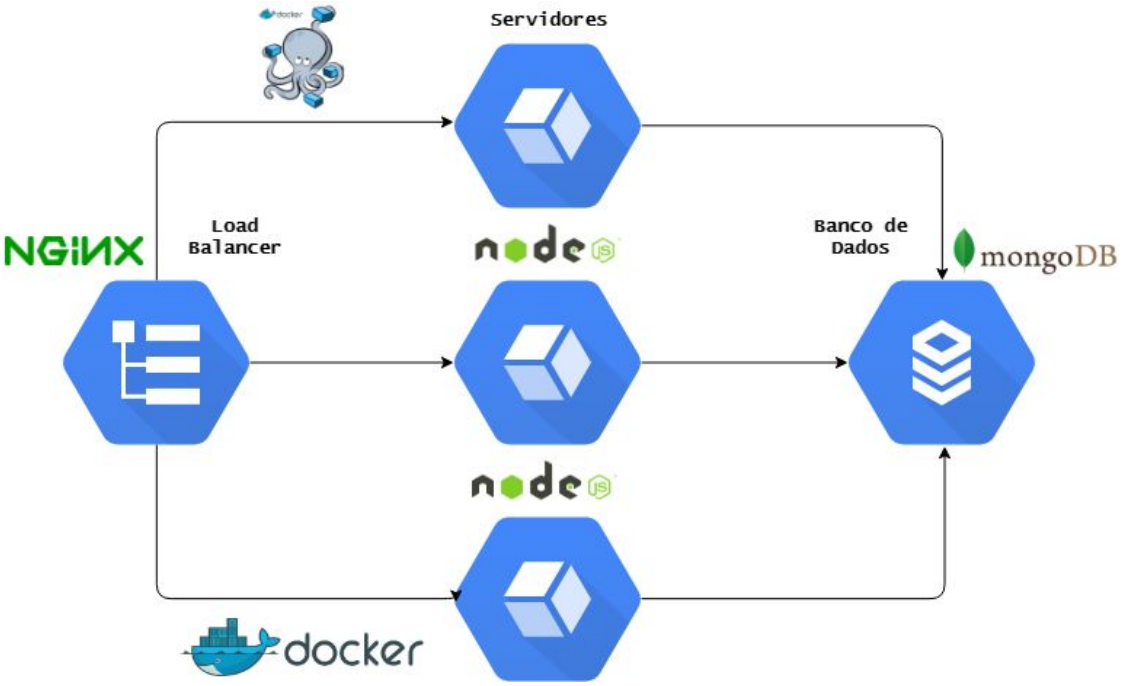

Figura 3: Ferramentas utilizadas no experimento

Baixando o código e executando os comandos *docker-compose build* e *docker-compose* up poderemos visualizar esta estrutura sendo utilizada, ao realizar uma requisição no navegador pelo endereço *localhost* poderemos identificar que diferentes containers retornam a requisição, isso é em consequência da ação do balanceamento de carga, isso também poderá ser visualizado no terminal onde foi executado o comando *docker-compose up*.

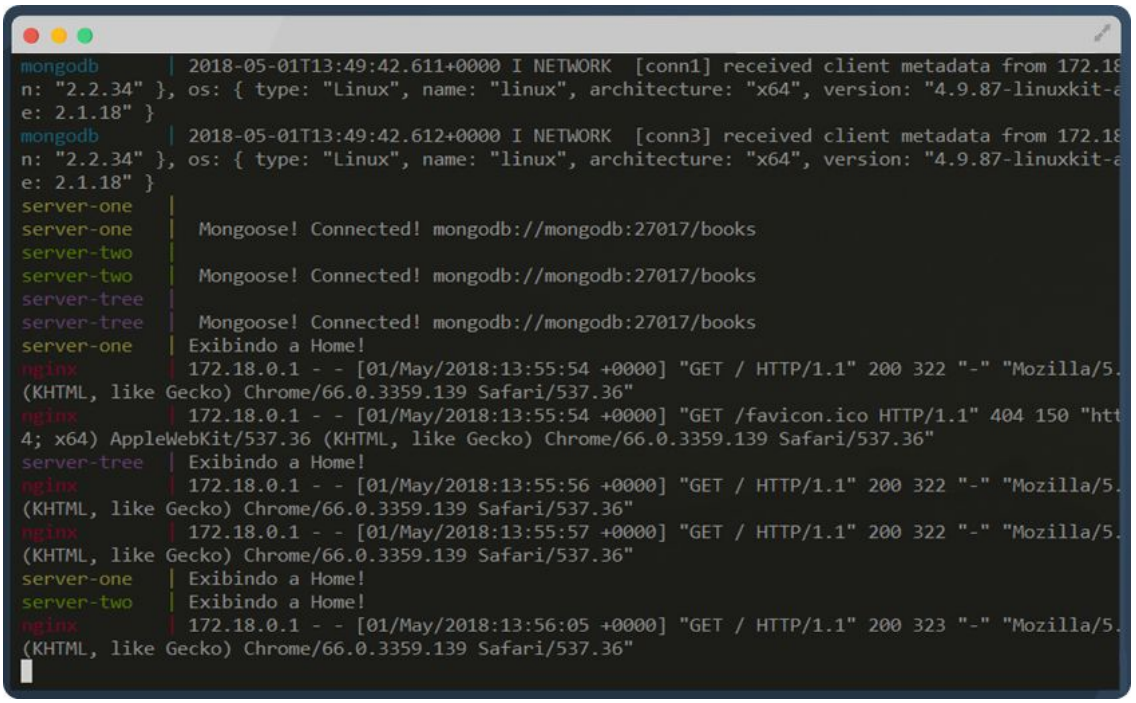

Figura 4: Visualização do terminal após executar docker-compose up

Acessando localhost no navegador podemos visualizar a distribuição das requisições sendo realizada, mostrando o id do container que está retornando a resposta para cada requisição.

Essa distribuição pode ser visualizada com mais detalhes pelo terminal onde foi executado o comando *docker-compose up .*

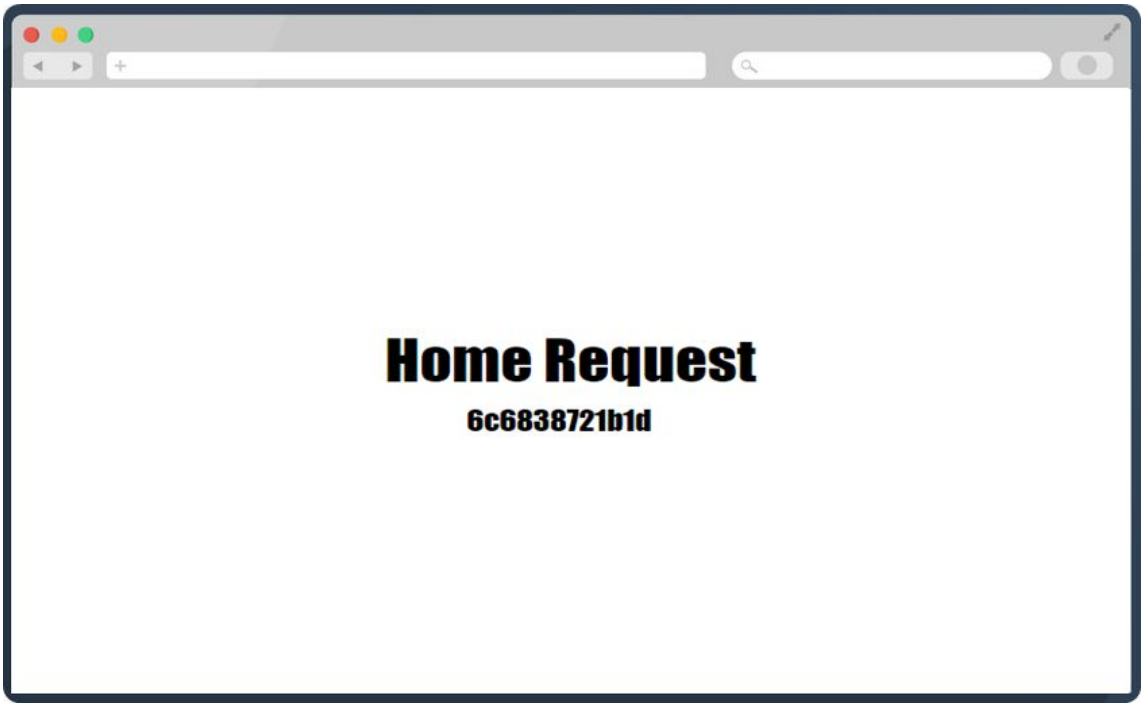

Figura 5: Visualização do navegador retornando a requisição feita a localhost

9. CONCLUSÃO

Neste trabalho pudemos conhecer e entender os principais conceitos envolvimento a clusterização e os resultados de sua aplicabilidade em nossa realidade, onde uma grande capacidade de processamento é necessária para trabalhar com o gigantesco volume de dados existentes. Entendemos os tipos de clusters e em qual necessidade cada um desses tipos foca seus esforços, desta forma podemos separar um deles para realizar um experimento e aplicar os conceitos estudados tendo como resultado a construção de um cluster focado no balanceamento de carga utilizando containers. Conhecemos também muito sobre dois modelos de implementação bastante populares atualmente que são os beowulf e mosix como também dois softwares muito utilizados para implementação de um sistema de clusters sendo eles o rocks e o slurm utilizados pelos principais computadores do mundo.

10. REFERÊNCIAS

- [1] TANENBAUM, A.S e STEEN, M.V. Sistemas distribuídos - princípios e paradigmas. 2ª ed Pearson Prentice Hall. São Paulo, 2007.

- [2] BACELLAR, H.V. Cluster - Computação de alto desempenho. Disponível em: <http://www.ic.unicamp.br/~ducatte/mo401/1s2010/T2/107077-t2.pdf>. Acesso em 08 de abril de 2017.

- [3] BARROS, A.B.P. Computação em cluster. Disponível em: <http://www.ice.edu.br/TNX/encontrocomputacao/artigos-internos/prof_andersown_computacao_em_cluster.pdf>. Acesso em 08 de abril de 2017.

- [4] CANALTI. Cluster - O que é, história e tipos de cluster. Disponível em: <https://www.canalti.com.br/computacao/cluster-o-que-e-historia-tipos-de-cluster-exemplo/>. Acesso em 08 de abril de 2017.

- [5] ROCKSCLUSTERS. Open-Source Toolkit for Real and Virtual Clusters. Disponível em: <http://www.rocksclusters.org/>. Acesso em 08 de abril de 2017.

- [6] SCHED MD. Slurm Overview. Disponível em: <https://slurm.schedmd.com/overview.html>. Acesso em 08 de abril de 2017.

11. AUTORES

Aline Sayuri
Estudante do 7º Semestre de Ciência da Computação
Uniceub - Centro Universitário de Brasília

Igor Gomes
Estudante do 7º Semestre de Ciência da Computação
Uniceub - Centro Universitário de Brasília

Sandro Vieira
Estudante do 7º Semestre de Ciência da Computação
Uniceub - Centro Universitário de Brasília

Samuel Costa
Estudante do 7º Semestre de Ciência da Computação
Uniceub - Centro Universitário de Brasília