



## DESENVOLVIMENTO DE UM CLUSTER DE BALANCEAMENTO DE CARGA PARA SERVIDORES WEB

Rodrigo Everton Soares Oliveira, Gabriela Santos Silva, Michael Jonnes Lopes Souza, Hugo Luciano de Souza Leal,  
Raimundo Sebastião Teodoro Santana

### Introdução

O balanceamento de carga consiste na divisão da quantidade de trabalho realizado entre dois ou mais computadores, possibilitando assim a realização da mesma quantidade de trabalho com a mesma quantidade de tempo e também todos os utilizadores sejam servidos de forma mais rápida.

Este projeto consiste na contextualização da implementação do balanceamento de carga utilizando o servidor Apache. Através desta implementação foi possível a análise de conceitos como clusters, balanceamento de carga e alta disponibilidade, conceitos estes necessários para o entendimento do presente trabalho, assim propondo a realização de um ambiente de alta disponibilidade para máquinas virtuais, utilizando o servidor Apache.

### Materiais e métodos

#### A. Sistemas Distribuídos

Segundo Tanenbaun [3], sistemas distribuídos é uma “coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente”. Segundo Colouris [1], os sistemas distribuídos podem ser conceituados como sistemas onde os seus componentes localizados em computadores interligados se comunicam e se relacionam através da troca de mensagens.

A Fig.1 apresenta um cenário da aplicação de sistemas distribuídos: a internet. Pode-se observar através desta figura, a relação entre computadores pessoais, os servidores de redes, os provedores e os links de comunicação.

#### B. Cluster

Cluster é o nome dado a um sistema que relaciona dois ou mais computadores para que estes trabalhem de maneira conjunta no intuito de processar uma tarefa. Cada computador participante do cluster recebe um nome de nó (ou node). Teoricamente, não há limite máximo de nós, mas independentemente da quantidade de máquinas que o compõe, o cluster deve ser “transparente”, ou seja, ser visto pelo usuário ou por outro sistema que necessita deste processamento como um único computador.

Em clusters de balanceamento de carga, as tarefas de processamento são distribuídas o mais uniformemente possível entre os nós. O foco é fazer com que cada computador receba e atenda a uma requisição e não que divida uma tarefa com outras máquinas.

### Resultados

Para a criação do ambiente que possibilitaria a realização deste projeto, foram utilizadas as seguintes ferramentas, *Linux Ubuntu*: O Ubuntu é um sistema operacional baseado em Linux desenvolvido pela comunidade podendo ser utilizado em notebooks, desktops e servidores; *Server Linux*: A plataforma líder para a computação *scale-out*, o Ubuntu Server ajuda a tirar o máximo partido da sua infraestrutura. *VMware*: software/máquina virtual que permite a instalação e utilização de um sistema operacional dentro de outro dando suporte real a software de outros sistemas operativos.

Foram criadas três máquinas virtuais Ubuntu, sendo que duas utilizam o sistema operacional convencional e outra utilizando o Ubuntu Server, descrito anteriormente.

Para o desenvolvimento, foi necessária a instalação do servidor Apache nas duas máquinas utilizando o Ubuntu. O servidor Apache foi criado em 1995 por Rob McCool, é considerada a principal tecnologia da Apache Software Foundation, empresa responsável por diversos projetos envolvendo tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos utilizando a API do software.

Através do comando “*1 - apt-get install apache2 php5*”, foi possibilitada a instalação do servidor Apache nas máquinas que receberam o servidor web.



Para fazer o balanceamento de carga foi utilizado o Pound, que configura as máquinas em um cluster e trabalha com conexões de entrada HTTP. Assim quando a conexão é aceita ele manda para uma das máquinas do cluster para que seja atendida. Para a instalação do Pound utiliza-se o comando “*1 - apt-get install pound*”.

Após a sua instalação, o Pound foi configurado criando e editando o arquivo `/etc/pound.cfg`. Para a realização de teste da configuração original, foi alterada a linha “*Alive 2*”, esta linha determina de quanto enquanto tempo, em segundos, o balanceador de cargas deve verificar os servidores balanceados.

```
ListenHTTP
  Address 192.168.1.109
  Port 80
End
Service
  BackEnd
    Address 192.168.2.105
    Port 80
  End
  BackEnd
    Address 192.168.1.106
    Port 80
  End
Session
  Type Cookie
  ID "loadbal_session"
  TTL 300
End
End
```

No bloco apresentado, o balanceador vai ouvir as requisições na porta 80 e em seu IP. Na série BackEnd foi especificado quais foram os servidores web e em qual porta atenderão, no caso a porta 80. Em Session descreve que o controle de sessão será feito por cookie, nomeado por `loadbal_session` e o tempo de vida será 300 segundos. Este procedimento é apresentado nas Fig.1 e Fig.2.

Foi editado o arquivo `/etc/default/pound` e alterada a linha “*Startup=1*”. Esse procedimento é apresentado na figura que segue.

Originalmente a linha vem com o valor zero, assim ele pode indicar que o Pound tem uma nova configuração.

Para que o serviço seja inicializado, utiliza-se o comando “*1 - /etc/init.d/pound start*”.

Após a configuração do Pound, foi criado um site simples apresentado na Fig.5, foram realizados testes que demonstraram e comprovam na prática o balanceamento de carga realizado com sucesso.

## Conclusão

Inicialmente, no projeto seria utilizado a ferramenta hyper-v que é uma tecnologia de virtualização baseada em hypervisor. O hypervisor é a plataforma de processamento de virtualização que permite que múltiplos Sistemas Operacionais compartilhem uma única plataforma de hardware. Porém foram encontrados diversos problemas com a utilização do hyper v, como a difícil inicialização de uma máquina virtual, impossibilitando assim, a sua utilização no presente projeto.

Através da implementação deste ambiente de alta disponibilidade, foi possível colocar em prática os conceitos aprendidos na disciplina de sistemas distribuídos II de forma simples e rápida, como por exemplo, cluster, balanceamento de carga, alta disponibilidade, dentre outros. Foi criado um site simples onde se alterava apenas o nome da máquina para demonstrar a possibilidade de efetuação de cópias do site em dois servidores, e servidores de máquina virtual e que ele funcionaria. Assim, foi possível observar diversos resultados importantes para um sistema, como por exemplo, o balanceamento de carga permitindo a divisão de carga de processamento.

## Referências

- [1] COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Sistemas Distribuídos - Conceitos e Projeto**. 4ª Edição. São Paulo: Bookman, 2007.
- [2] F. B. **Criando um cluster de balanceamento de carga para servidores web**. 2014 disponível em: <[http://www.securso.com.br/index.php?option=com\\_content&view=article&id=337%3Acra-do-um-cluster-de-balanceamento-de-carga-para-servidores-web&catid=912%3Alinux-&Itemid=72](http://www.securso.com.br/index.php?option=com_content&view=article&id=337%3Acra-do-um-cluster-de-balanceamento-de-carga-para-servidores-web&catid=912%3Alinux-&Itemid=72)> Acesso em: dez 2014.



[3] TANENBAUM, A. S. **Sistemas Operacionais Modernos**. Rio de Janeiro: Prentice Hall do Brasil, 1995. 62 p.

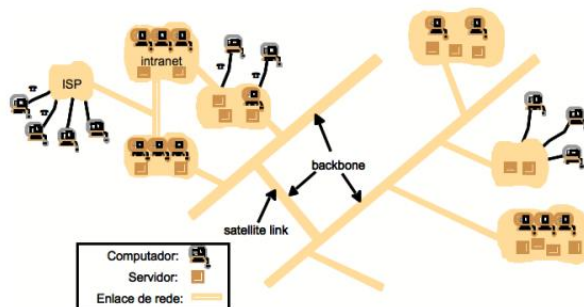


Figura 1: Um exemplo de sistemas distribuídos **Fonte:** [http://btd.egc.ufsc.br/wp-content/uploads/2008/03/Douglas\\_Macedo.pdf](http://btd.egc.ufsc.br/wp-content/uploads/2008/03/Douglas_Macedo.pdf)

```

GNU nano 2.2.6      New Buffer      Modified
# check backend every X secs:
Allow 2
## use hardware-acceleration card supported by openssl(1):
##SSL Engine "ChaCha"
# poundctl control socket
Control "/usr/sbin/poundctl.sock"

#####
## listen, redirect and ... to:
## redirect all requests on port 8080 ("ListenHTTP") to the local webserver (see "Service" below):
ListenHTTP
Address 192.168.1.105
Port 80
## allow PUT and DELETE also (by default only GET, POST and HEAD):
## HTTP
0

Service
Backend
Address 192.168.1.106
Port 80
End
Backend
Address 192.168.1.104
Port 80
End
End

```

Figura 2: Configuração do Pound. **Fonte:** Própria

```

GNU nano 2.2.6      New Buffer      Modified
Port 80
## allow PUT and DELETE also (by default only GET, POST and HEAD):
## HTTP
0

Service
Backend
Address 192.168.1.106
Port 80
End
Backend
Address 192.168.1.104
Port 80
End
Session
Type Cookie
ID "hashba1_session"
TTL 300
End
End

```

Figura 3: Configuração do Pound, **Fonte:** Própria

```

GNU nano 2.2.6      New Buffer      Modified
# Defaults for pound initscript
# sourced by /etc/init.d/pound
# installed at /etc/default/pound by the maintainer scripts
# present startup with default configuration
# set the below variable to 1 in order to allow pound to start
startup_1

```

Figura 4: Configuração do Pound. **Fonte:** Própria



Figura 5: Site utilizado para a realização de testes. **Fonte:** Própria.