MONTAGEM DE CLUSTER BEOWULF

Autor: Danilo Menzanoti Fugi <danilofugi at gmail.com>
Data: 13/03/2015

O QUE É CLUSTER

Um *Cluster* é formado por um conjunto de computadores interligados através de uma rede, as máquinas membros deste cluster são denominadas nó ou node. É importante utilizar uma infraestrutura de rede que facilite a inclusão, alteração e exclusão de máquinas.

Na maioria das vezes o cluster é formado por computadores convencionais e se apresenta de forma transparente ao usuário, como sendo um único computador de grande porte. É válido frisar que é possível a utilização de máquinas mais robustas para construção de clusters.

Não é necessário que as máquinas sejam idênticas, mas sim o Sistema Operacional, para que os softwares que gerenciam as trocas de mensagens e sincronismo dos dados funcionem de forma correta.

De acordo com Zem (2005), existem hoje alguns tipos de cluster, mas alguns se destacam pela aplicação e custo benefício:

- **`Cluster de Alto Desempenho`** : denominado, também, de Alta Performance (High Performance Computing - HPC), sua característica é o grande volume de processamento de dados em computadores convencionais, que garante baixo custo na construção, e com processamento na ordem de gigaflops. Os servidores deste cluster trabalham com a tecnologia de paralelismo, dividindo o processamento com as outras máquinas, buscando a otimização e desempenho de um supercomputador.
- **`Cluster de Alta Disponibilidade`** (High Availability - HA): são caracterizados por se manterem em pleno funcionamento por um longo período de tempo, utilizando redundância para manter um serviço ativo e se proteger de falhas, geralmente são computadores convencionais que disponibilizam o mesmo recurso em todas as máquinas da rede, configuradas com prioridades diferentes, onde existe um servidor ativo e os outros ociosos.
- **`Cluster de Balanceamento de Carga`** (Horizontal Scaling - HS): são caracterizados por dividirem, de forma equilibrada, as tarefas entre os membros do cluster, onde cada nó atenda a uma requisição e não, necessariamente, que divida uma tarefa com outras máquinas.

É importante salientar que é possível a combinação de mais de uma metodologia de construção de clusters, onde uma implementação de Alta Disponibilidade para garantir acesso aos serviços de vendas online, possa ser incrementada com a utilização de um cluster de Balanceamento de Carga para atender o aumento nos acessos ao serviço.

As características marcantes dos clusters, são a facilidade de gerenciamento dos nós, onde podemos adicionar, dar manutenção e remover um nó do cluster sem que afete seu funcionamento, recuperação de falhas de forma otimizada.

Podemos obter resultados tão satisfatórios no uso de um cluster, quanto em servidores sofisticados com um custo muito menor. A implementação pode ser utilizada para aplicações sofisticadas e, também, para aplicações domésticas.

MONTAGEM DO CLUSTER

A implementação do cluster foi realizada, inicialmente, em máquinas virtuais, para testes, visando fácil locomoção, ambiente físico reduzido e clonagem dos nós é feita de forma rápida e confiável.

A máquina física hospedeira das virtuais utiliza processador Intel i3, com 4 GB de memória RAM e HD de 1 TB, cada máquina virtual possui 512 MB de memória RAM, HD de 8 GB e 02 placas de rede, as configurações das máquinas são mostradas abaixo:

- Máquina 01: Debian 7.1, hostname Servidor, IP eth0 10.0.2.15/24 (recebido via DHCP da máquina hospedeira que faz o NAT para acesso a internet) IP eth1192.168.0.1/24 (rede interna), domínio: dominio.com.br.
- Máquina 02: Debian 7.1, hostname bw2, IP eth0 10.0.2.15/24 (recebido via DHCP da máquina hospedeira que faz o NAT para acesso a internet) IP eth1192.168.0.2/24 (rede interna), domínio: dominio.com.br.
- Máquina 03: Debian 7.1, hostname bw3, IP eth0 10.0.2.15/24 (recebido via DHCP da máquina hospedeira que faz o NAT para acesso a internet) IP eth1192.168.0.3/24 (rede interna), domínio: dominio.com.br.
- Máquina 04: Debian 7.1, hostname bw4, IP eth0 10.0.2.15/24 (recebido via DHCP da máquina hospedeira que faz o NAT para acesso a internet) IP eth1192.168.0.3/24 (rede interna), domínio: dominio.com.br.

Os ajustes realizados nos nós foram feitos em apenas um nó e copiados via SSH para os outros nós, garantindo a não ocorrência erros de configurações diferentes. Atualização de repositórios de pacotes no arquivos "sources.list" e atualização com o comando:

```
# aptitude update && aptitude -y dist-upgrade
```

Nome de máquina: servidor, bw2, bw3 e bw4.

Abaixo, segue os "sources.list" utilizados nos sistemas. São arquivos básicos, uma vez que somente o servidor está com a parte gráfica e os nós estão sem a parte gráfica.

Arquivo "sources.list" servidor:

```
#
# deb cdrom:[Debian GNU/Linux (//www.vivaolinux.com.br/linux/) 7.1.0 _Wheezy_ - Official amd64 CD Binary-1 20130615-23:06]/ wheezy main
#deb cdrom:[Debian GNU/Linux 7.1.0 _Wheezy_ - Official amd64 CD Binary-1 20130615-23:06]/ wheezy main

deb http://ftp.br.debian.org/debian/ wheezy main contrib non-free
deb-src http://ftp.br.debian.org/debian/ wheezy main contrib non-free

deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-free

# wheezy-updates, previously known as 'volatile'
deb http://ftp.br.debian.org/debian/ wheezy-updates main contrib non-free
deb-src http://ftp.br.debian.org/debian/ wheezy-updates main contrib non-free

deb http://mirrors.kernel.org/debian/ wheezy-updates main contrib non-free
deb-src http://mirrors.kernel.org/debian/ wheezy-updates main contrib non-free

deb http://ftp.debian.org/debian/ wheezy-updates main contrib non-free
deb-src http://ftp.debian.org/debian/ wheezy-updates main contrib non-free
```

CONFIGURAÇÃO

1. Preparando o sistema (servidor e nós):

```
# apt-get update
# aptitude safe-upgrade
```

2. Instalando e atualizando pacotes (servidor e nós) - (na dúvida, faça cada pacote separado):

```
# aptitude install build-essential module-init-tools kernel-package initramfs-tools
# aptitude install autoconf libaal-dev wget liblzo2-dev gzip libncurses5 libncurses5-dev dpatch udev
# aptitude install openjdk-7-jre # Somente se for trabalhar com Java - servidor e nós
```

4. Instalando mais pacotes necessários (servidor e nós):

```
# apt-get update
# apt-get install -y gfortran-*
```

* Tem que ser o **apt-get** , serão 500mb aproximadamente.

Reiniciar:

```
# shutdown -r now
```

5. Alterando os arquivos necessários (servidor e nós):

```
# ifconfig
# nano /etc/network/interfaces
```

```
allow hotplug eth1
iface eth1 inet static
address 192.168.0.1
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

O arquivo interfaces acima deve ser configurado em todos os nós, a configuração acima é do Servidor. O Arquivo "hosts" deve ser igual em todos os nós:

```
# nano /etc/hosts
```

```
127.0.0.1   localhost
192.168.0.1   servidor.dominio.com.br      servidor
192.168.0.2   bw2.dominio.com.br      bw2
192.168.0.3   bw3.dominio.com.br      bw3
192.168.0.4   bw4.dominio.com.br      bw4
```

O arquivo "hosts.equiv" deve ser igual em todos os nós:

 # nano /etc/hosts.equiv

```
servidor
bw2
bw3
bw4
```

O arquivo "/home/.rhosts" deve ser igual em todos os nós:

 # nano /home/.rhosts

```
servidor
bw2
bw3
bw4
```

O arquivo */root/.rhosts*, deve ser o mesmo em todas as máquinas do cluster:

 # nano /root/.rhosts

```
servidor
bw2
bw3
bw4
```

 # nano /etc/securetty

No arquivo securetty, somente acrescente as linhas:

```
console
rsh
ssh
```

O arquivo */opt/hostfile* deve ser o mesmo em todas as máquinas do cluster:

```
# nano /opt/hostfile
```

```
servidor
bw2
bw3
bw4
```

SERVIDOR SSH

Servidor e nós:

```
# aptitude install -y ssh openssh-server
```

No servidor, gerando a chave com 1024 bits:

```
# ssh-keygen -b 1024 -t rsa
```

Copiando a chave, comando (troque os IPs dos nós):

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub root@192.168.0.X
```

Para testar, na primeira vez, deverá pedir a senha, digite a senha:

```
# ssh 192.168.0.2 -n 'echo $SHELL'
```

Deverá aparecer: `/bin/bash`

Teste todos os IPs do cluster. Em caso de erro, primeiro reinicie a máquina e apague todo o conteúdo do arquivo:

```
# nano /root/.ssh/known_hosts
```

Depois, ao acessar um nó:

```
# ssh no01
```

Aparece uma mensagem: `Are you sure you want to continue...`

Digite "yes". Irá pedir a senha, digite e, na próxima vez, não pedirá mais senha.

INSTALAÇÕES DE PACOTES, MONTAGEM E MONITORAMENTO

Instalando PVFS2 (servidor e nós)

Parallel Virtual File System, é um sistema de arquivos concebido para proporcionar alto desempenho para aplicações paralelas. Nos nós e no servidor, instalando as bibliotecas:

```
# apt-get install libdb5.1 libdb5.1-dev
```

Após isso, faça os passos para instalar o PVFS:

```
# cd /usr/src
# wget ftp://ftp.parl.clemson.edu/pub/pvfs2/pvfs-2.8.2.tar.gz
```
(ftp://ftp.parl.clemson.edu/pub/pvfs2/pvfs-2.8.2.tar.gz)

```
# mkdir /opt/mpich
# mkdir /opt/pvfs2
# tar -xzvf pvfs-2.8.2.tar.gz
# cd pvfs-2.8.2
# ./configure
```

Deve terminar sem erros, com a linha: `PVFS2 version string: 2.8.2`

```
# make
```

Deverá terminar com a última linha: `GENCONFIG examples/fs.conf`

```
# make install
```

Acrescentar no arquivo (servidor e nós):

 # nano /etc/fstab

```
tcp://servidor(ou no0X):3334/pvfs2-fs /mnt/pvfs2 pvfs2 defaults,noauto 0 0
```

 # mkdir /mnt/pvfs2

Nos nós, remover MTA para não atrasar a inicialização:

 # update-rc.d -f exim4 remove

Configurando a variável de ambiente, entrar no arquivo:

 # nano ~/.bashcrc

E acrescentar no final:

```
LD_LIBRARY_PATH=/opt/pvfs2/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

Salvar e sair. Reiniciar:

 # reboot

Executar no servidor:

 # /opt/pvfs2/bin/pvfs2-genconfig /etc/pvfs2-fs.conf

Criando um novo storage e preparando para iniciar pela primeira vez:

 # /opt/pvfs2/sbin/pvfs2-server /etc/pvfs2-fs.conf -f

Deve retornar:

```
/opt/pvfs2/sbin/pvfs2-server /etc/pvfs2-fs.conf -f
[S 09/15 15:50] PVFS2 Server on node servidor version 2.8.2 starting...
[D 09/13 15:50] PVFS2 Server: storage space created. Exiting.
```

Iniciando:

```
# /opt/pvfs2/sbin/pvfs2-server /etc/pvfs2-fs.conf
[S 09/09 15:55] PVFS2 Server on node servidor version 2.8.2 starting...
```

Copiando para os nós "pvfs2-fs.conf" para os nós:

```
# scp /etc/pvfs2-fs.conf 192.168.1.X:/etc/
```

Ainda no servidor:

```
# /opt/pvfs2/sbin/pvfs2-server /etc/pvfs2-fs.conf
```

Testando:

```
# /opt/pvfs2/bin/pvfs2-ping -m /mnt/pvfs2
```

SERVIDOR NFS

O NFS faz o compartilhamento e sincronização de diretórios e arquivos no cluster. Iniciando a instalação do NFS:

```
# cd /home/Usuario
```

No servidor:

```
# apt-get install portmap
# apt-get install nfs-common
# apt-get install nfs-kernel-server
# apt-get install nfs-user-server
```

Entre no arquivo:

```
# nano /etc/exports
```

Coloque o conteúdo no final:

```
/home/Usuario 192.168.1.0/24(rw,all_squash,subtree_check,anonuid=150,anongid=100)
/opt 192.168.1.0/24(rw,all_squash,subtree_check)
/usr/local 192.168.1.0/24(rw,all_squash,subtree_check)
```

Atualizando o kernel com as mudanças no arquivo */etc/exports*:

```
# exportfs -a
```

Reinicie o serviço:

```
# service nfs-kernel-server restart
```

Nos nós, entre no arquivo */etc/fstab*:

```
# nano /etc/fstab
```

Adicione no final:

```
192.168.1.6:/home/Usuario   /home/Usuario  nfs   defaults   0   0
192.168.1.6:/opt            /opt           nfs   defaults   0   0
192.168.1.6:/usr/local      /usr/local     nfs   defaults   0   0
```

MPICH

Message Passing Interface é uma Interface de Passagem de Mensagens. Essa é a biblioteca que transforma um conjunto de máquinas em um Cluster. Servidor e nós (ORANGEFS já está trás tudo embutido como PVFS, MPICH2 e MPI-IO (ROMIO)):

```
# cd /usr/src
# wget http://orangefs.org/downloads/LATEST/source/orangefs-2.9.1.tar.gz
```
(http://orangefs.org/downloads/LATEST/source/orangefs-2.9.1.tar.gz)

```
# tar -xzvf orangefs-2.9.1.tar.gz
# cd orangefs-2.9.1.tar.gz
# ./configure
```

Se não tiver erros, terminará com: `Configuration completed`

```
# make
# make install
# wget http://ftp.de.debian.org/debian/pool/main/l/lam/lam_7.1.4.orig.tar.gz
```
(http://ftp.de.debian.org/debian/pool/main/l/lam/lam_7.1.4.orig.tar.gz)

```
# tar -xzvf lam_7.1.4.orig.tar.gz
```

```
# cd lam-7.1.4
# ./configure
```

Se não tiver erros, terminará com: `Configuration completed`

```
# make
# make install
# wget http://www.mpich.org/static/downloads/3.0.4/mpich-3.0.4.tar.gz
```
(http://www.mpich.org/static/downloads/3.0.4/mpich-3.0.4.tar.gz)

```
# tar -xzvf mpich-3.0.4.tar.gz
# cd mpich-3.0.4
# ./configure
```

Se não tiver erros, terminará com: `Configuration completed`

```
# make
# make install
```

Configurando as variáveis:

```
# nano ~/.bashrc
```

```
PATH=/opt/mpich/bin:$PATH
export PATH

LD_LIBRARY_PATH=/opt/mpich/lib:/opt/pvfs2/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
/opt/mpich/lib
```

O arquivo "bashrc" deve ser copiado para todos os nós. Reinicie:

```
# reboot
```

Ver as informações:

```
# mpiexec -info
```

Vamos testar nossa instalação, compilando e executando:

```
# cd /usr/src/mpich-3.0.4/examples
# mpicc -o cpi cpi.c
```

Compilando o arquivo em todas as máquinas. Utilizar o comando:

```
# mpicc -o -hostfile /opt/hostfile
/usr/src/mpich-3.0.4/examples/cpi.c cpi
```

Onde: **cpi.c** - Calcula o valor de PI.

Testando:

```
# mpirun -hostfile /opt/hostfile -n 7
/usr/src/mpich-3.0.4/examples/cpi
```

Deverá aparecer o processo de calculo dividindo o processo em todos os nós, com resultado e tempo gasto no cálculo. Neste momento. Nosso cluster HP já está funcionando!

INSTALANDO O GANGLIA (MONITOR GRÁFICO)

Nos nós:

```
# apt-get update
# apt-get install ganglia-monitor
```

No servidor:

Onde: **invoke-rc.d gdm3 start** - habilita interface gráfica, caso tenha desabilitado.

```
# apt-get update
# apt-get install apache2
```

Abra o navegador e digite: **localhost**
Deverá aparecer: `It works` (assim o apache está funcionando)

```
# apt-get install php5 libapache2-mod-php5
```

Crie o arquivo:

```
# nano /var/www/info.php
```

Coloque dentro dele:

```
<?php phpinfo(); ?>
```

Salve e saia. Reinicie o Apache:

```
# service apache2 restart
```

Abra o navegador e digite: **http://localhost/info.php**

Deverá aparecer a página de informações do PHP.

Instalando o Ganglia:

```
# apt-get install ganglia-webfrontend ganglia-monitor
```

Vamos copiar o arquivo necessário:

```
# cp /etc/ganglia-webfrontend/apache.conf /etc/apache2/sites-enabled/ganglia.conf
```

Alterar o arquivo */etc/ganglia/gmod.conf* no servidor e copiar para os nós.

A parte que nos interessa alterar, está na imagem acima. Comente as linhas onde tem o IP 239.2.11.71

```
name= "kluster"
owner= "kluster"
/* mcast join 239.2.11.71 */
host = 192.168.0.1
/* mcast join 239.2.11.71 */
/* bind = 239.2.11.71 */
```

Ou similar. E configure o host com o IP do servidor.

Altere o arquivo */etc/ganglia/gmetad.conf* no servidor, deixando como na imagem acima, não sendo necessário copiar para os nós.

```
data_source "kluster" 15 localhost 192.168.0.1
gridname = "kluster"
authority "http://localhost/ganglia/"
```

Para reiniciar o serviço nos nós:

```
# service ganglia-monitor restart
```

Reiniciar o Apache e o Ganglia no servidor:

```
# service apache2 restart
# service gmetad restart
```

A partir daí, é só digitar no navegador: **`http://localhost/ganglia`**

Danilo M. Fugi - Ciência da Computação - 7º Período
danilofugi@gmail.com
IF Sul de Minas - Muzambinho

Voltar (verArtigo.php?codigo=15145)