

Configuração - Cluster Rápido Beowulf

Etapas

Esta configuração utiliza como base 4 etapas:

1. Identificação dos computadores em ambiente de rede.
2. Compartilhamento dos arquivos que irão processar em cluster – NFS.
3. Configuração do acesso remoto – SSH.
4. Configuração da biblioteca de passagem de mensagem - MPICH.

Etapas complementares:

- Teste com exemplo cpi fornecido com a biblioteca de passagem de mensagem – MPICH
- Instalação e utilização da aplicação de computação gráfica POV RAY
- Instalação da ferramenta de monitoramento de cluster - SCMS

Para configuração a partir dos CD's de instalação.

Serão utilizados os CD's 1, 2, 3 e 4 do kernel 2.6.8.2 sarge da distribuição Debian.

Pré-configuração

Atualizar o arquivo source.list, que se encontra no diretório /etc/apt.

Digite os comandos:

```
apt-cdrom add
```

Insira o CD 1

```
apt-cdrom add
```

Insira o CD 2

```
apt-cdrom add
```

Insira o CD 3

```
apt-cdrom add
```

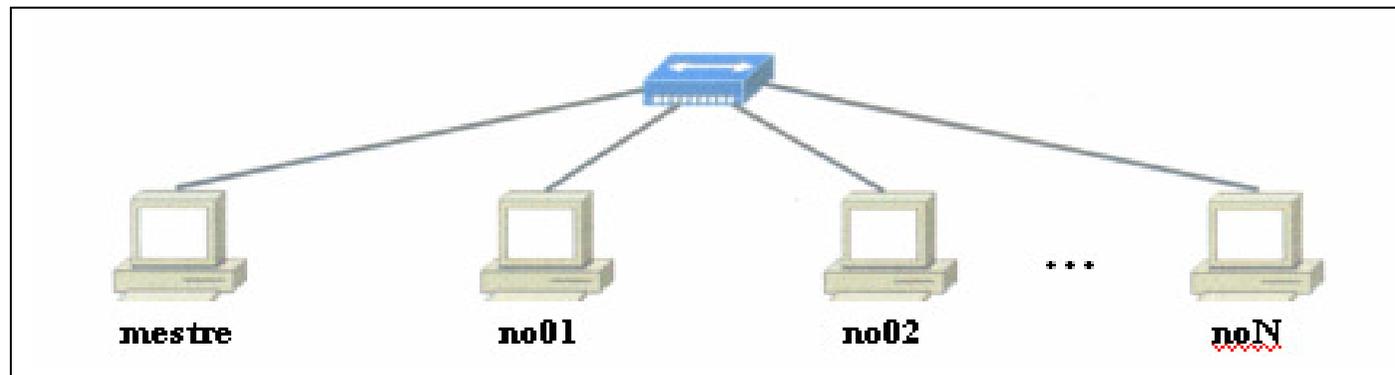
Insira o CD 4

Instale os seguintes pacotes:

Pacote	Comando
Patch	apt-get install patch
Make	apt-get install make
xlibs-dev	apt-get install xlibs-dev
libpng-dev	apt-get install libpng-dev
Gcc	apt-get install gcc
g++	apt-get install g++
libstdc++2.10-dev	apt-get install libstdc++2.10-dev

1. IDENTIFICAÇÃO DOS COMPUTADORES EM AMBIENTE DE REDE

Durante a configuração da rede será necessário definir as configurações básicas da rede propriamente dita, o nome do computador e um conversor de IP's em nomes. Esta configuração será formada por quatro computadores, sendo um mestre e os outros três nós participantes do cluster, podendo ser ampliada para o número de computadores disponíveis.



As tabelas a seguir mostram as atividades que deve ser realizada em cada computador (mestre, no01, no02,...)

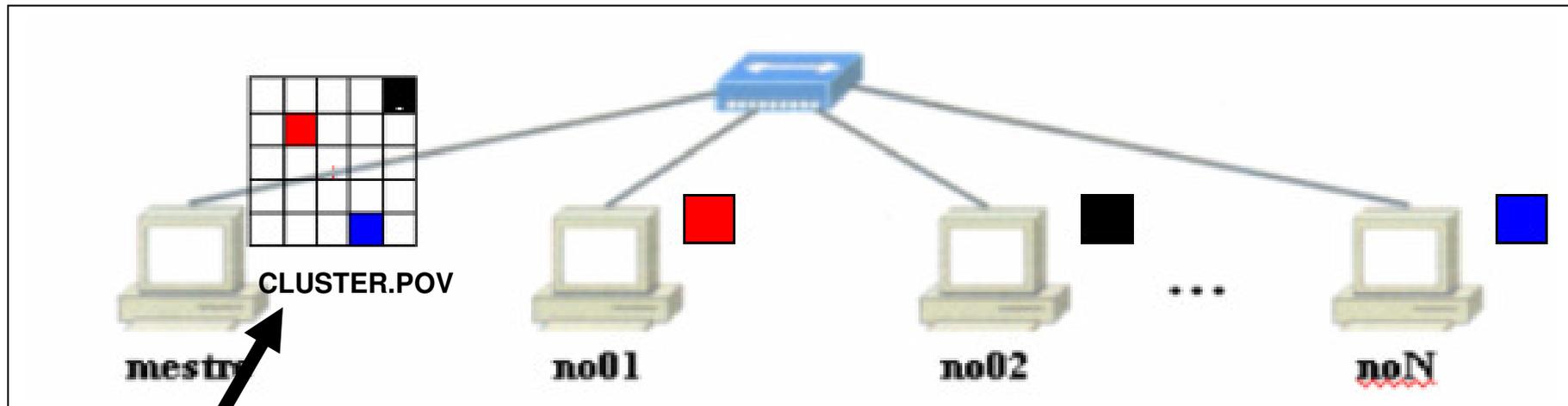
Atividade	Mestre	No01	No02
Configuração da placa de rede Editar o arquivo <code>/etc/network/interfaces</code> <code>vi /etc/network/interfaces</code>	<code>auto eth0</code> <code>iface eth0 inet static</code> <code>address 192.168.0.10</code> <code>netmask 255.255.255.0</code> <code>network 192.168.0.0</code> <code>broadcast 192.168.0.255</code> <code>gateway 192.168.0.1</code>	<code>auto eth0</code> <code>iface eth0 inet static</code> <code>address 192.168.0.11</code> <code>netmask 255.255.255.0</code> <code>network 192.168.0.0</code> <code>broadcast 192.168.0.255</code> <code>gateway 192.168.0.1</code>	<code>auto eth0</code> <code>iface eth0 inet static</code> <code>address 192.168.0.12</code> <code>netmask 255.255.255.0</code> <code>network 192.168.0.0</code> <code>broadcast 192.168.0.255</code> <code>gateway 192.168.0.1</code>
Configuração do nome do computador Editar o arquivo <code>/etc/hostname</code> <code>vi /etc/hostname</code>	<code>mestre</code>	<code>no01</code>	<code>no02</code>
Conversão de um IP em um nome Editar o arquivo <code>/etc/hosts</code> <code>vi /etc/hosts</code> Obs: Este passo não é necessário, serve apenas para facilitar o gerenciamento do cluster.	<code>192.168.0.10 mestre</code> <code>192.168.0.11 no01</code> <code>192.168.0.12 no02</code>	<code>192.168.0.10 mestre</code> <code>192.168.0.11 no01</code> <code>192.168.0.12 no02</code>	<code>192.168.0.10 mestre</code> <code>192.168.0.11 no01</code> <code>192.168.0.12 no02</code>
Reinicializar os computadores através	<code>shutdown -r now</code>	<code>shutdown -r now</code>	<code>shutdown -r now</code>

<p>do comando.</p> <p>shutdown -r now</p> <p>Objetivo: atualizar o nome dos computadores, configurados no arquivo hostname.</p>			
<p>Caso os computadores já possuam os seus respectivos nomes, a rede pode ser reiniciada através dos comandos</p>	<p>cd /etc/init.d</p> <p>./networking restart</p>	<p>cd /etc/init.d</p> <p>./networking restart</p>	<p>cd /etc/init.d</p> <p>./networking restart</p>

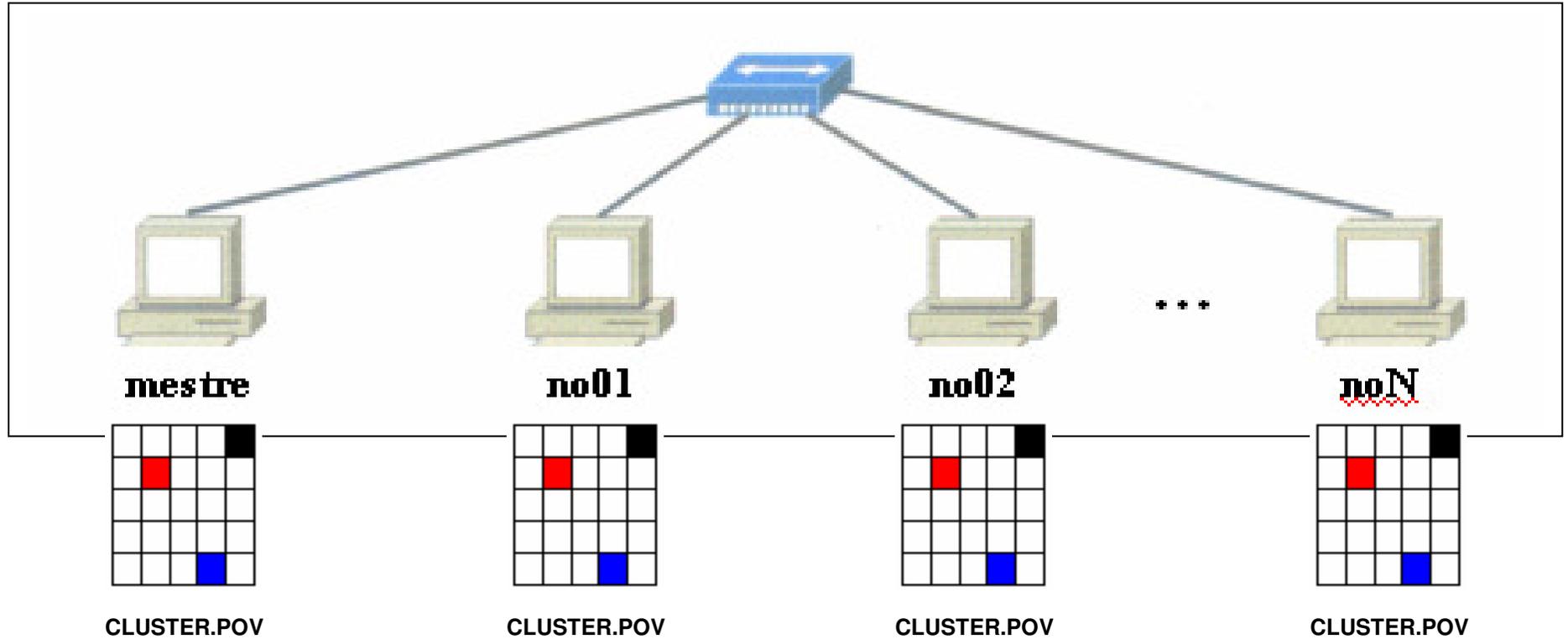
Nota 1: O título Atividade se refere a atividade que deve ser realizada em cada uma das máquinas; através da edição de arquivos ou execução de comandos; as configurações em laranja dependem do ambiente utilizado para a construção do cluster, em um ambiente doméstico não são necessárias.

Nota 2: O arquivo hosts fará o papel de um servidor de DNS convertendo os números IP em nomes.

2. COMPARTILHAMENTO DOS ARQUIVOS QUE IRÃO PROCESSAR NO CLUSTER



A quadro da figura acima representa uma aplicação – CLUSTER.POV, que necessita de uma grande capacidade computacional para renderizar uma animação feita com o aplicativo POV RAY. Para processar em um cluster Beowulf a aplicação deverá ser capaz de se dividir, portanto, cada quadradinho do quadro representa um pedaço da aplicação que deverá ser enviado para os nós para processamento, o servidor só deve gerenciar o processo de envio dos pacotes e a recuperação do resultado do processamento dos nós. Cada um dos computadores participantes do cluster possui o seu sistema de gerenciamento de arquivos local, o que não permite uma visão integral da aplicação como um todo. Todos os computadores devem enxergar a aplicação de forma total, para isso será necessário compartilhar a aplicação no ambiente de rede. O NFS permite o compartilhamento da aplicação através da rede.



Atividade a ser realizada	Mestre	No01	No02
Instalação do NFS Pacotes utilizados: nfs-kernel-server nfs-common	apt-get install nfs-kernel-server	apt-get install nfs-kernel-server	apt-get install nfs-kernel-server
Configuração do NFS Editar o arquivo /etc/exports vi /etc/exports	Acrescente as linhas /usr *(rw,sync,no_root_squash) /lib *(rw,sync,no_root_squash)		
Inicializar ou reinicializar o serviço NFS.	cd /etc/init.d ./nfs-kernel-server restart ./nfs-common restart		
Montar o NFS		mount mestre:/usr /usr -t nfs mount mestre:/lib /lib -t	mount mestre:/usr /usr -t nfs mount mestre:/lib /lib -t

		nfs	nfs
Configurar a montagem automática NFS.		Editar o arquivo /etc/fstab vi /etc/fstab	Editar o arquivo /etc/fstab vi /etc/fstab
Acrescente as linhas.		mestre:/usr /usr nfs defaults 0 0	mestre:/usr /usr nfs defaults 0 0
		mestre:/lib /lib nfs defaults 0 0	mestre:/lib /lib nfs defaults 0 0

Nota 1: No arquivo exports se definiu, que os diretórios /usr e /lib do mestre serão compartilhados para leitura e escrita, adotando a permissão do usuário local das máquinas.

Nota 2: Durante a inicialização ou reinicialização do NFS o conteúdo do arquivo exports será verificado.

Nota 3: A montagem do NFS com os comandos; `mount mestre:/usr /usr -t nfs` e `mount mestre:/lib /lib -t nfs` vão gerar o compartilhamento do conteúdo do exports entre as máquinas.

Nota 4: A montagem automática, evita a montagem do NFS toda vez que se for utilizar o cluster.

Linhas de comando para os nós:

```
mestre:/usr /usr nfs defaults 0 0
mestre:/lib /lib nfs defaults 0 0
```

3. CONFIGURAÇÃO DO ACESSO REMOTO

Atividade	Mestre	No01	No02
Instalação do SSH	<code>apt-get install ssh</code>	<code>apt-get install ssh</code>	<code>apt-get install ssh</code>
Gerar chave pública	<code>ssh-keygen -b 1024 -t rsa</code>		
Copiar a chave pública para os nós e para o próprio mestre.	<code>cd /root/.ssh</code> <code>scp /root/.ssh/id_rsa.pub mestre:/root/.ssh/authorized_keys</code> <code>scp /root/.ssh/id_rsa.pub no01:/root/.ssh/authorized_keys</code> <code>scp /root/.ssh/id_rsa.pub no02:/root/.ssh/authorized_keys</code> ...		

Nota: O acesso a partir do computador mestre aos nós com a utilização do ssh, não deve pedir senha.

Exemplos:

ssh no01

ssh no02

...

Não deve haver pedido de senha.

4. CONFIGURAÇÃO DA BIBLIOTECA DE PASSAGEM DE MENSAGEM

Atividade	Mestre	No01	No02
Instalação do mpich	<code>apt-get install mpich</code>		
<p>Configurar o arquivo <code>machines.LINUX</code></p> <p>Este arquivo é responsável pelas máquinas que farão parte do cluster.</p> <p>Acrescente as linhas <code>no01</code>, <code>no02</code> e <code>no03</code>, o mestre não precisa ser incluído.</p> <p>Remova a linha <code>localhost</code></p>	<pre>cd /usr/lib/mpich/share vi machines.LINUX no01 no02 ...</pre>		
<p>Alterar o arquivo <code>/root/.bashrc</code></p> <p>Acrescente as linhas:</p>	<pre>PATH=\$PATH:/usr/lib/mpich/bin export PATH</pre>		

5. TESTE DE FUNCIONAMENTO

Atividade	Mestre	No01	No02
Entre no diretório	<code>cd /usr/share/doc/libmpich1.0-dev/examples</code>		
Compilar o arquivo <code>cpi.c</code> com o compilador <code>mpicc</code>	<code>mpicc cpi.c -o cpi</code>		
Executar o cluster com o comando <code>mpirun</code>	<code>mpirun -np 3 cpi</code>		

6. INSTALAÇÃO DO POV RAY

Os arquivos exemplo, utilizados para teste que compõem o pacote do mpich são muito simplistas para se ter um noção real de desempenho do cluster. O PovRay pode exigir um pouco mais de desempenho do cluster.

- Fazer o download dos arquivos.

Arquivos do PovRay versão 3.1g

ftp://ftp.povray.org/pub/povray/Old-Versions/Official-3.1g/Unix/povuni_s.tgz

ftp://ftp.povray.org/pub/povray/Old-Versions/Official-3.1g/Unix/povuni_d.tgz

Patch MPI-Povray

<http://www.verrall.demon.co.uk/mpipov/mpi-povray-1.0.patch.gz>

- Copiar os arquivos para um diretório padrão.

```
cp povuni_s.tgz /usr/local/src
```

```
cp povuni_d.tgz /usr/local/src
```

```
cp mpi-povray-1.0.patch.gz /usr/local/src
```

- Descompactar os arquivos do PovRay.

```
cd /usr/local/src
```

```
tar -zxvf povuni_s.tgz
```

```
tar -zxvf povuni_d.tgz
```

- Mover o patch para o diretório do PovRay

```
mv mpi-povray-1.0.patch.gz /usr/local/src/povray31
```

- Aplicar o patch.

```
cd /usr/local/src/povray31  
gunzip mpi-povray-1.0.patch.gz  
cat mpi-povray-1.0.patch | patch -p1
```

- Compilar a aplicação.

```
cd /usr/local/src/povray31/source/mpi-unix  
make
```

- Copiar o arquivo mpi-x-povray para /usr/local/bin

```
cp /usr/local/src/povray31/source/mpi-unix/mpi-x-povray /usr/local/bin
```

- Testar o funcionamento do programa

```
cd /usr/local/src/povray31/scenes/animate
```

Nota: O diretório /animate possui uma série de sub-diretórios com exemplos de animações.

```
cd ambient
```

```
mpirun -np 4 /usr/local/bin/mpi-x-povray ambient.ini +L/usr/local/src/povray31/include -w800 -h600
```

7. Instalação da ferramenta de gerenciamento - SCMS (Smile Cluster)

Apenas no Mestre:

1) Arquivos necessários

Entrar no site: <http://downloads.opensce.org/pub/sce> fazer o download dos seguintes arquivos:

- libhal-2.5.8.tar.gz
- scebase-1.1.2.tar.gz
- Pmw.1.2.tar.gz
- scms-2.3.7.tar.gz

2) Instalar o Python

```
apt-get install python pytho-tk python-dev
```

3) Instalar o libhal

```
tar -zxvf libhal-2.5.8.tar.gz
cd libhal-2.5.8
./configure
make
make install
```

4) Instalar o scebase

```
tar -zxvf scebase-1.1.2.tar.gz
cd scebase-1.1.2
./configure
make
make install
```

5) Instalar o python-imaging

```
apt-get install python-imaging python-imaging-tk
```

6) Instalar o pacote libltdl

```
apt-get install libltdl3 libltdl3-dev
```

7) Descompactar o Pmw dentro do diretório site-packages do Python

```
cd /usr/lib/python2.3/site-packages  
tar -zxvf /CAMINHO_DO_Pmw/Pmw.1.2.tar.gz
```

8) Instalar o SCMS

```
tar -zxvf scms-2.3.7.tar.gz  
cd scms-2.3.7  
./configure  
make  
make install
```

9) Editar o arquivo /usr/local/etc/sce/sce.conf

```
0=mestre  
1=no01  
2=no02  
3=no03
```

```
host_num=4
```

Obs: Verificar se o plugin_path aponta para /usr/local/lib

10) Inicializar o daemon rms

O comando abaixo, executado no mestre, inicializa o rms em todas as máquinas do cluster.

```
Pexec -a -suf-forceall /usr/local/sbin/rms
```

Nota: Digite em todos os nós: rms

11) Rodar o SCMS

```
scms
```