

# Cluster Beowulf com FreeBSD

Autoria de Daniel Bristot de Oliveira  
13/09/2006  
Última Atualização 15/09/2006

Clusters Beowulf são formados por computadores pessoais, normalmente utilizando arquitetura Intel x86 e sistema operacional livre, neste caso o FreeBSD, interligados em uma rede privada de alta velocidade. Os nós do cluster são dedicados à execução de tarefas paralelas e não possuem um terminal acoplado ao sistema. Normalmente estão conectados ao resto do mundo por um único nó do, chamado de Nó Mestre.

Neste artigo mostrarei como criar um Cluster Beowulf, criando um ambiente homogêneo de software, um requisito para a criação deste cluster, a partir de boot via rede, e ligarei estes hosts utilizando duas bibliotecas de programação paralela, a MPI e a PVM.

A temos a figura do ambiente utilizado.

Para esta modalidade de cluster, é exigido uma configuração homogênea de software, isto é, o mesmo conjunto de software, e compartilhamento de disco entre todos os nós do cluster para a entrada/saída de dados. A partir deste contexto, uma solução para estes dois paradigmas é a utilização de um sistema operacional em rede compartilhado por todos os nós do cluster. O sistema operacional escolhido dá suporte a operação sem disco, isto é, ele permite a sua inicialização e execução em um computador sem uma unidade de disco rígido, para isto ele obtém seus arquivos e configurações de um computador através de uma conexão de rede. Esta forma centralizada de armazenamento de software e configuração possibilita manter todo o sistema homogêneo com a administração de apenas um nó da rede. Inicializando um sistema sem disco

Algumas placas de rede possuem uma propriedade chama PXE - pre-boot eXecution Enviroment, que é uma memória ROM que da suporte a inicialização via rede. Quando este módulo é inicializado, ele procura por um servidor DHCP - Dynamic Host Configuration Protocol na rede. Ao encontrar este servidor, o PXE diz para o servidor qual o seu endereço MAC - Media Access Protocol, e se o servidor for responsável por este endereço, ele irá responder um endereço de IP para o PXE. Após o PXE receber o endereço, ele pergunta para o mesmo servidor DHCP se ele possui uma imagem de inicialização, e o servidor responde o endereço de onde está a imagem, esta imagem é o carregador de boot do sistema operacional, ela é uma pequena imagem, que o PXE descarrega através de um servidor TFTP - Trivial File Transfer Protocol que o DHCP informa, quando esta imagem é executada, ela pega o endereço do diretório raiz, via DHCP e inicializa o kernel do sistema operacional. Após o kernel estar inicializado, ele irá iniciar a execução dos scripts de inicialização, como o PXE não se comunica diretamente com o kernel, o kernel solicita novamente ao servidor DHCP seu endereço de IP, após isto, as partições de disco são montadas via rede, isso é possível através do NFS - Network File System, após a montagem dos sistemas de arquivos o sistema é inicializado normalmente.

Para melhorar o entendimento, esta configuração será dividida em dois segmentos:

Configuração do nó mestre, que será responsável pela distribuição de todo o software, juntamente com as unidades de disco, para todos os nós escravos do cluster. É a parte mais importante do cluster, pois toda a configuração será feita nele. Basicamente o nó mestre irá disponibilizar três serviços de rede:

- DHCP
- TFTP
- NFS

Estes três serviços irão disponibilizar tudo o que os nós escravos necessitam para inicializar.

Configuração dos nós escravos será feita no nó mestre, a configuração basicamente é a configuração do sistema raiz, onde estão os principais arquivos do sistema, configuração de um kernel reduzido e com suporte ao inicialização sem disco, configuração do /etc e scripts de inicialização. Instalação do Sistema Operacional

A instalação será feita no mesmo computador duas vezes. Instalação do Nó-mestre

O nó mestre será o carro chefe desta operação, ele será responsável pelo armazenamento do sistema, e pelos meios de compartilhamento. Primeiramente o nó mestre deve ter o sistema operacional FreeBSD instalado, atualmente estamos utilizando a versão 6.1-RELEASE. Está fora do escopo deste artigo a instalação do sistema operacional, porém, abaixo

segue algumas informações essenciais sobre a instalação sistema operacional no nó mestre. Ponto de montagem

Tamanho	Descrição	
/	256 Mb	Sistema de arquivos raiz do sistema
/usr	10Gb	Aplicativos do sistema
/home	Indefinido	Compartilhamento de arquivos
/barra	256Mb	Sistema de arquivos raiz para os nós escravos
Swap	Variável	Memória virtual

Tabela 1 &ndash; Particionamento do nó mestre:

Um número mínimo de pacotes são necessários para a configuração do nó mestre, em uma configuração customizada, eles são os seguintes:

- Base
- Kernel -> GENERIC
- Ports
- src

Demais pacotes podem ser instalados, porém estes são o mínimo necessários para a instalação. Instalação do Nó-escravo

O nó escravo também será instalado no nó mestre, o que se deve levar em conta é que o nó escravo terá uma partição raiz separada da partição raiz do nó mestre, pois a inicialização dos nós escravos é diferente da inicialização do nó mestre. Os nós escravos terão um kernel e arquivos e de configuração customizados para esta operação. A instalação pode ser feita de diversas formas, porém utilizamos a forma convencional de instalação, ela será feita como a instalação do mestre, a partir do instalador do FreeBSD. A diferença é: O único sistema de arquivos montado, o /barra da outra instalação e será montado como o sistema de arquivos raiz (/).

O pacotes instalados serão:

- Kernel: Generic
- Base

Não habilite nenhum daemon do sistema, e não adicione usuários e não configure a rede. A instalação do Sistema está pronta. Na inicialização Real do sistema, somente inicialize o sistema operacional do nó mestre. Configurando o Nó-Mestre

A configuração do nó mestre é basicamente a instalação dos daemons de rede que o mestre irá disponibilizar. Para a instalação e configuração iremos seguir a ordem em que os daemons são solicitados pelos nós escravos durante a inicialização, esta ordem é:

- DHCP
  - TFTP
  - NFS
- Instalando e configuração do DHCP

O servidor DHCP utilizado neste artigo o DHCP da ISC - Internet Systems Consortium, ele está disponível para instalação de duas formas, via package ou via ports, aqui instalaremos via ports. O caminho do ports é: /usr/ports/net/isc-dhcp3-server/, para executar a instalação execute, como root, os seguintes comandos. # cd /usr/ports/net/isc-dhcp3-server/ # make install

Para nossos propósitos, não é necessário nenhum dos módulos mostrados nesta tela de configuração, por isto desabilite todos.

Se tudo ocorrer bem, o instalador irá terminar a compilação do DHCP sem mais perguntas. Configurando o DHCP

O arquivo de configuração do DHCP normalmente está localizado em /usr/local/etc/ com o nome dhcpd.conf, abaixo veremos um exemplo comentado de parte do arquivo de configuração, palavras em verde são comentários. #

Abaixo opções padrões

```
default-lease-time 600;
max-lease-time 7200;
authoritative;
```

```
option domain-name "cluster.unesc.net";      ## Dominio do cluster
option domain-name-servers 10.0.70.1;        ## Servidor DNS, Não necessário
option routers 10.0.70.1; Roteador,          ## Não necessário
```

# Abaixo, Informações sobre endereços de rede, neste caso a rede 10.0.70

```

subnet 10.0.70.0 netmask 255.255.255.0 {
    use-host-decl-names on;           ## Atribui o valor de host como o nome da maquina
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.0.70.255;

    host s2.cluster.unesc.net {       ## Informações sobre o nó 02
        hardware ethernet 00:04:23:04:4a:72;    ## Endereço de hardware
        fixed-address 10.0.70.2;               ## Endereço a de IP do host
        next-server 10.0.70.1;                ## Endereço do servidor TFTP
        filename "pxeboot";                   ## Imagem de inicialização do sistema operacional
        option root-path "10.0.70.1:/barra";   ## Define o sistema de arquivo raiz na notação usual do NFS
    }
}

```

A figura abaixo exemplifica o laboratório onde que foram feitos os testes, e logo abaixo é mostrado o arquivo de configuração utilizado no teste.

Seguindo a nossa Figura 1, a configuração do DHCP será assim:

```

Arquivo de configuração do dhcp.conf    default-lease-time 600;
max-lease-time 7200;
authoritative;
ddns-update-style ad-hoc;
option domain-name "unesc.net";
option routers 10.0.70.1

```

```

subnet 10.0.70.0 netmask 255.255.255.0 {
    use-host-decl-names on;
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.0.70.255;

```

```

    host s2.cluster.unesc.net {
        hardware ethernet 00:0a:5e:31:24:06;
        fixed-address 10.0.70.2;
        next-server 10.0.70.1;
        filename "pxeboot";
        option root-path "10.0.70.1:/barra/";
    }
}

```

```

    host s3.cluster.unesc.net {
        hardware ethernet 00:0a:5e:31:2c:a0;
        fixed-address 10.0.70.3;
        next-server 10.0.70.1;
        filename "pxeboot";
        option root-path "10.0.70.1:/barra/";
    }
}

```

```

    host s4.cluster.unesc.net {
        hardware ethernet 00:0a:5e:31:23:70;
        fixed-address 10.0.70.4;
        next-server 10.0.70.1;
        filename "pxeboot";
        option root-path "10.0.70.1:/barra/";
    }
}

```

```

    host s5.cluster.unesc.net {
        hardware ethernet 00:0a:5e:31:23:b0;
        fixed-address 10.0.70.5;
        next-server 10.0.70.1;
        filename "pxeboot";
        option root-path "10.0.70.1:/barra/";
    }
}

```

```

    host s6.cluster.unesc.net {

```

```

hardware ethernet 00:0a:5e:31:24:f3;
fixed-address 10.0.70.6;
next-server 10.0.70.1;
filename "/pxeboot";
option root-path "10.0.70.1:/barra/";
}
host s7.cluster.unesc.net {
hardware ethernet 00:0a:5e:31:23:54;
fixed-address 10.0.70.7;
next-server 10.0.70.1;
filename "/pxeboot";
option root-path "10.0.70.1:/barra/";
}

host s8.cluster.unesc.net {
hardware ethernet 00:0a:5e:31:23:50;
fixed-address 10.0.70.7;
next-server 10.0.70.1;
filename "/pxeboot";
option root-path "10.0.70.1:/barra/";
}
}

```

Para que o DHCP seja inicializado juntamente com o sistema operacional adicione a seguinte linha ao arquivo `/etc/rc.conf`: `dhcpcd_enable="YES"`

#### O DHCP está configurado. Configurando o TFTP

O TFTP é um servidor de arquivos semelhante ao ftp, porém ele não trata autenticação, por isso somente arquivos em que o público tenham permissão de leitura poderão ser lidos, e nenhum arquivo novo poderá ser criado. Este serviço vai ser gerenciado pelo `inetd(8)`, então as permissões do arquivo `/etc/hosts.allow` terão efeito sobre quem poderá acessar este serviço, em nosso caso somente os componentes do cluster, veja abaixo o exemplo do arquivo `/etc/hosts.allow`:  
`tftpd: 10.0.70.0/255.255.255.0 :allow`

O `tftpd(8)` vem instalado na base do sistema operacional, por isso nenhuma instalação adicional é requerida.

Sua inicialização e configuração é feita através do `inetd`, para isto somente precisamos descomentar a linha referente ao `tftpd` no arquivo `/etc/inetd.conf`. Em nosso caso a seguinte linha: `tftp dgram udp wait root /usr/libexec/tftpd`  
`tftpd -l -s /tftpboot`

Preste atenção no último argumento, ele é o diretório base do `tftpd`, os arquivos que serão exportados deverão estar dentro deste diretório, caso este diretório não exista, crie-o.

Para que o TFTP seja inicializado juntamente com o sistema operacional adicione a seguinte linha ao arquivo `/etc/rc.conf`: `inetd_enable="YES"`

Agora que o TFTP está configurado, copie o arquivo `pxeboot` do diretório `/boot` para o diretório raiz do TFTP, neste caso, `/tftpboot`. `# cp /boot/pxeboot /tftpboot`

#### O arquivo pxeboot é a imagem de inicialização do sistema operacional. Configurando o NFS.

O NFS irá exportar os sistemas de arquivos para os outros nós do cluster. Veja na tabela abaixo os sistemas de arquivos exportados e suas propriedades.

ponto de montagem	propriedades	descrição
/barra	Somente Leitura	Sistema de arquivos raiz dos nós escravos
/usr	Somente Leitura	/usr dos nós escravos
/home	Leitura e escrita	/home, Diretório de usuários compartilhado

O arquivo de configuração do NFS está em /etc/exports, veja abaixo o exemplo que foi utilizado em nossos testes. /barra  
 -alldirs -maproot=0:0 -ro -network 10.0.70 -mask 255.255.255.0  
 /usr -alldirs -maproot=0:0 -ro -network 10.0.70 -mask 255.255.255.0  
 /home -alldirs -maproot=0:0 -network 10.0.70 -mask 255.255.255.0

Para que o NFSD seja inicializado juntamente com o sistema operacional, adicione as seguintes linhas ao arquivo /etc/rc.conf: nfs\_server\_enable="YES"  
 rpcbind\_enable="YES"

A configuração do serviço está pronta. Configurando os nós escravos

A Configuração dos nós escravos serão feitas no nó mestre, pois somente ele terá acesso total ao sistema de arquivo raiz dos nós escravos. Seguindo a tabela 1 o sistema de arquivos raiz dos nós escravos está montado sobre o diretório /barra no nó mestre. Construindo um kernel com suporte a inicialização em rede

Um novo kernel deve ser compilado com suporte ao boot na rede. Os arquivos de configuração do kernel estão localizados em /usr/src/sys/i386/conf/ e o arquivo padrão de configuração é nomeado com GENERIC. Copie o arquivo GENERIC para outro arquivo com o nome DISKLESS, e adicione as seguintes linhas. options BOOTP  
 options BOOTP\_NFSROOT  
 options BOOTP\_NFSV3  
 options BOOTP\_COMPAT

O arquivo genérico de configuração do kernel do FreeBSD trás vários módulos compilados estaticamente com o kernel, em nosso sistema apenas deixaremos habilitados o necessário para o sistema sem disco, foram retiradas várias opções, entre elas, suporte a disco, a placas de redes não utilizadas, dispositivos ATA, SCSI, entre outros, veja baixo o arquivo de configuração utilizado em nossos testes.

```
DISKLESS arquivo de configuração. machine      i386
cpu      I686_CPU
ident    DISKLESS

options  SCHED_4BSD          # 4BSD scheduler
options  PREEMPTION          # Enable kernel thread preemption
options  INET                 # InterNETworking
options  FFS                  # Berkeley Fast Filesystem
options  MD_ROOT              # MD is a potential root device
options  NFSCLIENT            # Network Filesystem Client
options  NFS_ROOT             # NFS usable as /, requires NFSCLIENT
options  PSEUDOFHS            # Pseudo-filesystem framework
options  GEOM_GPT              # GUID Partition Tables.
options  COMPAT_43            # Compatible with BSD 4.3 [KEEP THIS!]
options  COMPAT_FREEBSD4      # Compatible with FreeBSD4
options  COMPAT_FREEBSD5      # Compatible with FreeBSD5
options  KTRACE                # ktrace(1) support
options  SYSVSHM               # SYSV-style shared memory
options  SYSVMSG               # SYSV-style message queues
options  SYSVSEM               # SYSV-style semaphores
options  _KPOSIX_PRIORITY_SCHEDULING # POSIX P1003_1B real-time extensions
options  KBD_INSTALL_CDEV      # install a CDEV entry in /dev
options  AHC_REG_PRETTY_PRINT  # Print register bitfields in debug
options  AHD_REG_PRETTY_PRINT  # Print register bitfields in debug
options  ADAPTIVE_GIANT        # Giant mutex is adaptive.
device   apic                  # I/O APIC
options  BOOTP                 #OPÇÕES DE BOOT
options  BOOTP_NFSROOT         #PELA REDE
options  BOOTP_NFSV3
options  BOOTP_COMPAT
device   eisa
device   pci
# atkbd0 controls both the keyboard and the PS/2 mouse
device   atkbd                # AT keyboard controller
```

```

device    atkbd        # AT keyboard
device    vga          # VGA video card driver
device    sc
device    agp          # support several AGP chipsets
device    pmtimer
# Network devices.
device    miibus       # MII bus support
device    fxp          # Intel EtherExpress PRO/100B (82557, 82558)
device    xl
# Pseudo devices.
device    loop         # Network loopback
device    random       # Entropy device
device    ether        # Ethernet support
device    tun          # Packet tunnel.
device    pty          # Pseudo-ttys (telnet etc)
device    md           # Memory "disks" UTILIZADO EM FS DE MEMÓRIA

```

Salve o arquivo de configuração do kernel com o nome DISKLESS e o compile com o seguintes comandos: # cd /usr/src/sys/i386/conf/  
# config DISKLESS  
# cd ../compile/DISKLESS  
# make cleandepend; make depend  
# make all  
# make DESTDIR=/barra install

O kernel está instalado. Arquivos de configuração

O diretório /etc é substituído durante a inicialização do sistema sem disco, ele é substituído por um diretório dentro de /conf/\$IP/etc, onde \$IP é o endereço de IP do nó escravo, caso este diretório não existir, ele será substituído por /conf/default/etc. Em nossos exemplos utilizaremos somente o diretório /conf/default/etc. Copie todos os arquivos dentro do /etc para /conf/default/etc, relativos ao endereço raiz dos nós escravos, neste caso /barra/etc/ e /barra/conf/default/etc/.  
# cp -rf /barra/etc/ /barra/conf/default/etc/

Agora todos os arquivos necessários para a inicialização estão criados, agora devemos editar alguns, como o /barra/conf/default/fstab e /barra/conf/default/etc/rc.confstab

O fstab terá que ser alterado, para montar os sistemas de arquivo a partir do nó mestre, abaixo veremos o fstab de nosso exemplo. 10.0.70.1:/barra/ / nfs ro 0 0  
10.0.70.1:/usr/ /usr nfs ro 0 0  
10.0.70.1:/home/ /home nfs rw 0 0  
md /var mfs rw,-s32m 0 0

As 3 primeiras linhas representam os sistemas de arquivos importados do nó mestre, a última linha é um detalhe importante; como as partições são exportadas como somente leitura, então os arquivos de log, sockets, arquivos de pid, que são necessários para o sistema não poderão serem escritos. A solução para isto é o md, o md é um pseudo sistema de arquivos de memória, ele aceita leitura e escrita, seus dados são gravados em memória e serão perdidos após o sistema desligar, como os arquivos que serão gravados sobre /var tem estado temporários para este caso, o md é perfeito para isto.rc.conf

O rc.conf é o arquivo de configuração de inicialização do sistema, por isso são necessários alguns ajustes nele.

Alguns serviços vem habilitados por padrão junto com a inicialização do FreeBSD, porém alguns deles não são necessários, vamos desabilita-los. Somente um serviço, fora do padrão, deve ser habilitado, o inetd, veremos a configuração dele mais a frente, por enquanto apenas iremos inicializa-lo. Veja abaixo o o exemplo de nosso /barra/conf/default/etc/rc.conf sendmail\_enable=NONE  
sendmail\_submit\_enable=NO  
sendmail\_msp\_queue\_enable=NO  
inetd\_enable=YES

Os usuários do sistema deverão ser os mesmos do nó mestre, existem varias formas dinâmicas de fazermos isto, porém em nosso caso, como os usuários serão poucos, a melhor solução é apenas replicar os arquivos de usuário do nó

mestre para os nós escravos. Para isto execute o seguinte comando: # cd /etc  
# cp master.passwd passwd pwd.db spwd.db /barra/conf/default/etc/

Agora vou apenas exemplificar a instalação e configuração de duas das principais bibliotecas de programação paralela, a MPICH2 e a PVM, para maiores informações leia as páginas dos manuais das bibliotecas. Integrando os nós via MPICH

A primeira parte já está pronta, já temos um sistema homogêneo e com discos compartilhados, porém isto ainda não é um cluster. Nosso cluster, seguindo a definição de cluster beowulf, será dedicado a execução de processos paralelos. Para isto utilizaremos a MPI - Message Passing Interface, ela é uma biblioteca de troca de mensagem entre processos paralelos. Cada processo trabalha com um conjunto de dados, e eles utilizam ela para acessar e modificar os dados de outros processos, assim criando um ambiente de compartilhamento de memória. Iremos utilizar a MPICH, que é uma implementação livre da MPI. Ela está disponível para instalação no FreeBSD via ports. Instalado a MPICH

O ports da MPICH está no diretório /usr/ports/net/mpich, para iniciar a instalação, execute os seguintes comandos: # cd /usr/ports/net/mpich  
# make install

Se nada anormal acontecer, a MPICH estará instalada. Ela é instalada sobre o diretório /usr/local/mpich, e seus executáveis ficarão dentro do diretório /usr/local/mpich/bin, vamos adicionar este diretório a variável PATH do ambiente, para isto execute o seguinte comando. # setenv PATH \$PATH:/usr/local/mpich/bin

A MPICH utiliza uma shell remota para a execução de processos, atualmente ela pode utilizar o RSH ou o SSH, como estamos em um ambiente seguro iremos utilizar o RSH pois o SSH utiliza criptografia, o que adiciona uma carga maior de processamento. Configurando RSH

O rsh é um serviço de login remoto, ele será utilizado pela MPICH para a comunicação entre processos. Para habilitá-lo precisamos apenas descomentar duas linhas do arquivo /barra/conf/default/etc/inetd.conf, as referentes ao rshd e logind, veja abaixo as linhas que devem ser descomentadas: shell stream tcp nowait root /usr/libexec/rshd  
rshd  
login stream tcp nowait root /usr/libexec/rlogind rlogind

Agora precisamos liberar o acesso sem senha a shell remota de todos os nós do cluster, isto é feito adicionando os endereços de ip, um por linha, de todos os nós do cluster, ao arquivo, /barra/conf/default/etc/hosts.equiv, em nosso exemplo. Ele ficou assim: 10.0.70.1

10.0.70.2  
10.0.70.3  
10.0.70.4  
10.0.70.5  
10.0.70.6  
10.0.70.7

10.0.60.8 Arquivos de tradução de nomes em IP

A MPICH comunica-se através de nomes de hosts não por endereços de ip, como estamos construindo um sistema pequeno, utilizaremos o arquivo /etc/hosts. A sintaxe do arquivo é: endereçoDoIP NomeDoHost Em nosso caso ele ficou assim: 10.0.70.1 mestre.cluster.unesc.net mestre

10.0.70.2 s2.cluster.unesc.net s2  
10.0.70.3 s3.cluster.unesc.net s3  
10.0.70.4 s4.cluster.unesc.net s4  
10.0.70.5 s5.cluster.unesc.net s5  
10.0.70.6 s6.cluster.unesc.net s6  
10.0.70.7 s7.cluster.unesc.net s7  
10.0.70.8 s8.cluster.unesc.net s8

Agora os programas que utilizam a MPICH já podem ser executados em nosso sistema. PVM

PVM - Parallel Virtual Machine fornece um conjunto integrado de bibliotecas e de ferramentas de software, que emula um ambiente de programação paralela que permite agregar, de maneira transparente, arquiteturas homogêneas ou heterogêneas. Instalando a PVM

No FreeBSD a PVM está disponível para instalação via ports, cuja localização é /usr/ports/net/pvm. Para instalá-la, é

necessário executar somente um comando, `make install`. Após a instalação, é preciso declarar a variável de ambiente `PVM_ROOT` com o diretório `/usr/local/lib/pvm`. Para isso, adiciona-se a seguinte linha de comando ao arquivo `/etc/csh.cshrc`: `setenv PVM_ROOT /usr/local/lib/pvm`.

A configuração da PVM é realizada através de um prompt. O comando `pvm` inicia o daemon de `pvm3d`, que é o daemon de monitoramento do cluster, e após abre o prompt. Porém, antes de iniciar o cluster, é preciso configurar o RSH, como para a MPI e o arquivo de hosts.

A configuração necessita da mesma configuração do `/etc/inetd`, `/etc/hosts.equiv` e `/etc/hosts` que a MPICH precisa.

Agora que temos tudo configurado, vamos inicializar o nosso cluster com o comando `pvm`, e depois adicionar os demais nós com o comando `add` do prompt do `pvm`. `$ pvm`

```
pvm> add s2
pvm> add s3
pvm> add s4
pvm> add s5
pvm> add s6
pvm> add s7
pvm> add s8
pvm> quit
```

Agora o nosso Cluster está funcionando. Teste com PVM e PovRay

Nos testes realizados em laboratório foram utilizados 8 computadores. Um nó mestre, Intel Pentium4 1.8 GHz, com 256 MB de memória principal e sete nós escravos, com processadores AMD Athlon 2800 (2079 MHz), MB de memória principal. Todos interligados por um Switch Fast Ethernet.

Como o objetivo principal é a paralelização de tarefas para o processamento de imagens, utilizou-se o PovRay ([www.povray.org](http://www.povray.org)). O PovRay um programa que gera gráficos 3D a partir de um vetor, o que demanda grande capacidade de processamento.

O PovRay contém tanto uma versão convencional (para ambiente monoprocessado) como uma versão paralela que utiliza a PVM para a comunicação, conhecida como `pvm pov`. As duas versões estão disponíveis para o FreeBSD a partir do ports.

Para ver o cluster funcionando, fiz processamento de uma imagem, denominada Cornell, com a versão paralela do `povray`. Primeiro fiz o processamento só no nó mestre, posteriormente foi adicionado gradativamente os outros nós do cluster.

A imagem processada.

Obteve os seguintes resultados.

A relação é de tempo em segundos em relação ao número de nós, e o primeiro gráfico é da imagem em 1024x768 e a outra é 2048x1536. A queda grande em relação a um nó e a dois nós, é que o nó mestre serve arquivos e o X para que tudo aconteça, então um maior processamento é feito nela, além do processamento de imagem, como no caso dos nós escravos do clusterXPVM

O XPVM é um aplicativo gráfico que mostra o estado do cluster em tempo de processamento, veja um exemplo a baixo do cluster com todos os nós em funcionamento.

Finalizando

Mais um artigo Pronto! espero que seja de utilidade para alguém.

A melhor fonte de documentação sempre são as páginas do manual e dos desenvolvedores, ninguém sabe mais sobre um produto que seu produtor.

<http://freebsd.org/doc/>



<http://freebsd.org/cgi/man.cgi>

Escutando Cake - She'll Come Back To Me, <http://cakemusic.com>

Um abraço a todos e até na próxima! No Fantástico Mundo De Bob!

Retirado de: <http://dbristot.info/wiki/index.php?title=Cluster>