



Ministério da  
Ciência e Tecnologia



INPE-15184-PUD/191

## CONFIGURANDO UM CLUSTER NO FEDORA CORE 4 COM MPICH2

Fernando Jaques Ruiz Simões Junior

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m17@80/2008/01.18.16.34>>

INPE  
São José dos Campos  
2008

## **PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: [pubtc@sid.inpe.br](mailto:pubtc@sid.inpe.br)

## **CONSELHO DE EDITORAÇÃO:**

### **Presidente:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

### **Membros:**

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

## **BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)

Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)

## **REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

## **EDITORAÇÃO ELETRÔNICA:**

Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)

## RESUMO

A utilização de conjuntos de computadores (clusters) tornou-se uma poderosa ferramenta para solução de problemas complexos. Com a popularização da informática, os computadores domésticos (PCs) tornaram-se muito baratos, de forma que a utilização de vários PCs trabalhando em paralelo surgiu como uma alternativa para utilização de processamento de alto desempenho. Neste guia vamos realizar detalhadamente todos os passos para configurar um conjunto de micros de modo à obter-se um cluster para processamento paralelo. A idéia deste trabalho é oferecer os passos operacionais do processo de configuração, portanto não vamos entrar em discussões profundas dos detalhes de cada procedimento.

# SUMÁRIO

1- Preparando o Sistema .....	4
2- Exportando diretórios para os outros micros do cluster.....	6
3- Instalando o MPICH-2.....	7
4- Configurações do usuário.....	8
Apêndice A – Comandos básicos de vi .....	10
Apêndice B – SSH sem senha (Fedora).....	12
REFERÊNCIAS BIBLIOGRÁFICAS.....	13

# 1- Preparando o Sistema

Vamos partir do princípio que todos os micros que integrarão o cluster já estão com o sistema operacional GNU/LINUX Fedora instalado e configurado. Antes de começarmos as configurações dos micros, devemos verificar se é possível realizar o comando rsh sem a utilização de senhas, isto é necessário para que os micros do cluster executem comandos uns nos outros sem a solicitação de senha.

## 1-Verificando se o rsh está instalado

```
chkconfig --list | grep rsh
ou
rpm -qa | grep rsh-server
```

Se não estiver instalado, devemos instalar o rsh. Como administrador (root) devemos realizar o comando em um terminal de comandos.

```
$ yum install rsh-server
```

## 2- Após instalado o rsh-server devemos editar os seguintes arquivos:

```
/etc/xinetd.d/rexec com DISABLE = NO
/etc/xinetd.d/rsh com DISABLE = NO
/etc/xinetd.d/rlogin com DISABLE = NO
```

## 3- Devemos editar o arquivo /etc/securetty e adicionar os comandos:

```
rsh
rexec
rlogin
```

## 4- Editar os seguintes arquivos PAM para login remoto;

### i- Editar o arquivo /etc/pam.d/login e comentar a seguinte linha com (#):

```
# auth required pam_securetty.so
```

### ii- Editar o arquivo /etc/pam.d/rsh alterar a linha

```
auth required pam_rhosts_auth.so
```

para

```
auth sufficient pam_rhosts_auth.so
```

no mesmo arquivo, comentar a linha:

```
# auth required pam_securetty.so
```

## 5- Se o cluster estiver alocado em uma rede com "Firewall" podemos

desabilita o iptables para facilitar as configurações e liberar as portas de rsh. Devemos lembrar que este procedimento diminui a segurança do sistema. As portas de rsh podem ser liberadas direto no iptables mas não entraremos em detalhes neste guia.

```
$ /sbin/services iptables stop
```

para desabilita a inicialização do iptables no boot e iniciar os serviços de rsh, devemos executar o comando abaixo como root:

```
$/usr/sbin/ntsysv
```

desmarcar a opção iptables e marcar para inicialização klogin, krb5-telnet e kshell.

6- Liberar o rsh e rlogin do kerberos (krb5):

É aconselhável fazer uma cópia de segurança dos arquivos /etc/profile.d/krb\*

```
$tar czvf krb5.tar.gz /etc/profile.d/krb*
```

remover os arquivos

```
$rm /etc/profile.d/krb*
```

7- Criar um arquivo chamado .rhost no ambiente do usuário que terá autorização para executar o rsh do micro1 para micro2 e vice versa, ex:

```
#configuração dos hosts que terão acesso por rsh
micro1.dominio.br
micro2.dominio.br
# final da configuração
```

Colocar este arquivo em todos os ambientes do usuário em todos os hosts que farão rsh.

8- Reiniciar os serviços xinetd;

```
$ /etc/init.d/xinetd restart
```

9- Se tudo estiver correto, a partir do micro1 podemos executar qualquer comando no micro2. Por exemplo, como usuário digite o comando abaixo,

```
$rsh micro2 date
```

o resultado será:

```
Ter Jan 12 11:05:19 UTC 2008
```

O próximo passo, será fazer com que o micro1 exporte os diretórios compartilhados em todos os micros que fazem parte do cluster.

## 2- Exportando diretórios para os outros micros do cluster.

Como o objetivo de configurar um cluster é a utilização de programação paralela, uma solução simples é a utilização de diretórios em comum em todos os micros do cluster, este procedimento facilita o acesso dos processadores as informações dos programas e subrotinas dos códigos.

1 – O primeiro passo é adicionar o usuário em todas as máquinas do cluster, como root:

```
$useradd usuário
```

```
$passwd usuário
```

OBS: todos os micros que farão parte do cluster devem ter o mesmo usuário com mesmo id e mesmo grupo

2- Criar no micro escolhido como servidor (micro1), o diretório que será exportado em todas as máquinas integrantes do cluster.

3- Configurar, no servidor, o arquivo /etc/exports, incluir a linha abaixo no arquivo exports

```
/home/usuario/diretorioexportado Ip-do-cliente(sync,rw,no_root_squash)
```

iniciar os serviço nfs com o comando /usr/sbin/ntsysv

4- configurar os clientes (outros micros do cluster) para que montem o diretório exportado, editar o arquivo /etc/fstab incluindo a linha

```
Ip-do-servidor:/home/usuario/diretorioexportado  
/home/usuario/diretorioexportado nfs rw,bg 0 0
```

5- Reiniciar os serviços de rede em todos os computadores.

```
$service xinetd restart
```

### 3- Instalando o MPICH-2

Para instalar o MPICH-2, devemos seguir as configurações indicadas pelos desenvolvedores do software [MPICH2, 2007], devemos obter os arquivos fontes no endereço de internet dos desenvolvedores, seção downloads e copiar o arquivo para um diretório.

1- Como root, devemos entrar no diretório onde os arquivos fontes foram copiados e extraídos, vamos executar

```
./configure  
make  
make install
```

2- Exportar as variáveis de ambiente do mpich2 para os usuários, vamos criar o script abaixo chamado mpiconf.sh e salvar em /etc/profile.d/.

```
$vi mpiconf.sh  
  
#!/bin/bash  
# PATH ENVIRONMENT TO MPICH2 #  
PATH=$PATH:/local-de-instalação-do-mpi/bin  
export PATH  
# END ENVIRONMENT TO MPICH2 #
```

Para que o script possa ser utilizado por todos os usuários, devemos dar permissões de execução a todos. Vamos alterar as permissões do arquivo mpiconf.sh

```
$chmod og+x mpiconf.sh
```

Após a instalação e configuração do mipch2 todos os usuários poderão utilizar as rotinas do mpi no cluster.

## 4- Configurações do usuário

Cada usuário do cluster necessita fazer algumas configurações para que seja possível a execução dos comandos de mpi. O mpi utiliza o rsh para a comunicação entre as máquinas, por este motivo cada usuário necessita realizar algumas configurações para acessar o rsh sem senha.

1- Devemos criar um arquivo chamado `.rhost` em cada micro do cluster (permite a comunicação de comandos via rsh sem senha), como escrito abaixo (comandos de vi no Apêndice A)

```
$vi .rhost

#hosts com acesso por rsh
monica.plasma.inpe.br
cebolinha.plasma.inpe.br
cascao.plasma.inpe.br
#end
```

o arquivo pode ser copiado a partir do `micro1` para os outros micros com o comando `security copy scp:`

```
$scp .rhost usuario@micro2.dominio:~/
```

2- Criar um arquivo chamado `.mpd.conf` que será utilizado para alocar a senha usada pelo mpd:

```
$vi .mpd.conf
```

este arquivo deve conter uma única linha:

```
secretword=senha
```

onde `senha`, é a senha utilizada pelo mpd e deve ser diferente da senha do usuário do sistema.

Alterar as permissões do arquivo `.mpd.conf` em todos os micros, é obrigatório para a utilização do mpiboot e evita que outros usuários possam acessar sua senha mpd.

```
$chmod 600 .mpd.conf
```

3- Criar o arquivo `mpd.hosts` em todas as máquinas do cluster, este arquivo informa ao mpi quais máquinas do cluster estarão acessíveis ao mpi

```
$vi mpd.hosts

#hosts usados no mpi
micro1
micro2
```

```
micro3
```

copiar este arquivo para as outras máquinas do cluster, podemos utilizar o comando scp:

```
$scp mpd.hosts usuario@micro2.dominio:~/
```

4- Para iniciar a comunicação, a partir do servidor, executar o comando

```
$mpdboot -r rsh -n numerodehosts
```

onde numerodehosts é o número de hosts que foram alocados em mpd.hosts

5- Testar a comunicação entre os micros com os comandos

```
$mpdtrace -l
```

o resultado deve ser algo semelhante com:

```
micro1.dominio_XXXXX (número de ip)  
micro2.dominio_XXXXX (número de ip)
```

onde XXXXX é a porta de comunicação entre os micros. Se o resultado estiver correto os micros estão configurados corretamente, se o resultado não estiver semelhante ao acima, deve-se rever os arquivos mpd.hosts .rhosts

6- Testar a comunicação entre os micros calculando pi em paralelo, o exemplo está no diretório de instalação do MPICH2, podemos executar o exemplo com o comando:

```
$mpiexec -n 4 /usr/local/mpich2-1.0.3/examples/cpi
```

O resultado deve ser algo parecido com:

```
Process 0 of 4 is on micro1.dominio  
Process 1 of 4 is on micro2.dominio  
Process 2 of 4 is on micro3.dominio  
Process 3 of 4 is on micro1.dominio  
pi is approximately 3.1415926544231239, error is 0.0000000008333307  
wall clock time = 0.002839
```

Se o resultado acima for obtido, significa que todas as configurações estão corretas e o cluster está trabalhando em paralelo. Muitas informações podem ser obtidas em Gropp et al. [1999] e Gropp et al. [1999].

## Apêndice A – Comandos básicos de vi

A sintaxe para executar o vi é a seguinte:

```
$ vi nome_do_arquivo
```

Veja abaixo uma lista dos principais comandos: Observação: Para passar para o modo de comandos pressione **ESC**.

### Comandos básicos de inserção de texto:

i	Inserir texto antes do cursor
a	Inserir texto depois do cursor
r	Inserir texto no início da linha onde se encontra o cursor
A	Inserir texto no final da linha onde se encontra o cursor
o	Adiciona linha abaixo da linha atual
O	Adiciona linha acima da linha atual
Ctrl + h	Apaga o último caractere

### Comandos básicos de movimentação:

Ctrl+f	Mover o cursor para a próxima tela
Ctrl+b	Mover o cursor para a tela anterior
H	Mover o cursor para a primeira linha da tela
M	Mover o cursor para o meio da tela
L	Mover o cursor para a última linha da tela
h	Mover o cursor um caractere para a esquerda
j	Mover o cursor para a próxima linha
k	Mover o cursor para a linha anterior
l	Mover o cursor um caractere para a direita
w	Mover o cursor para o início da próxima palavra (Ignora a pontuação)
W	Mover o cursor para o início da próxima palavra (Não ignora a pontuação)
b	Mover o cursor para o início da palavra anterior (Ignora a pontuação)
B	Mover o cursor para o início da palavra anterior (Não ignora a pontuação)
0	Mover o cursor para o início da linha atual
^	Mover o cursor para o primeiro caractere não branco da linha atual
\$	Mover o cursor para o final da linha atual
nG	Mover o cursor para a linha n

G Move o cursor para a ultima linha do arquivo

### Comandos básicos para localizar texto:

/palavra Busca pela palavra ou caracter em todo o texto  
?palavra Move o cursor para a ocorrência anterior da palavra  
n Repete o ultimo comando / ou ?  
N Repete o ultimo comando / ou ? , na direção reversa  
Ctrl+g Mostra o nome do arquivo, o numero da linha corrente e o total de linhas

### Comandos básicos para alteração de texto:

x Deleta o caracter que esta sob o cursor  
dw Deleta a palavra, da posição atual do cursor ate o final  
dd Deleta a linha atual  
D Deleta a linha a partir da posição atual do cursor ate o final  
rx Substitui o caracter sob o cursor pelo especificado em x (é opcional indicar o caracter)  
Rx Substitui a palavra sob o cursor pela palavra indicada em x  
u Desfaz a ultima modificação  
U Desfaz todas as modificações feitas na linha atual  
J Une a linha corrente a próxima  
s:/palavra1/palavra2 Substitui a primeira ocorrência de "palavra1" por "palavra2"

### Comandos para salvar o texto:

:wq Salva o arquivo e sai do editor  
:w nome\_do\_arquivo Salva o arquivo corrente com o nome especificado  
:w! nome\_do\_arquivo Salva o arquivo corrente no arquivo especificado  
:q Sai do editor  
:q! Sai do editor sem salvar as alterações realizadas

## Apêndice B – SSH sem senha (Fedora)

Vamos gerar um par de chaves (pública e privada) para o usuário que terá permissão de efetuar ssh sem a digitação de senha.

Para tal, usaremos o comando `ssh-keygen`, conforme mostrado abaixo. Quando for solicitada a digitação de senha, apenas pressione a tecla ENTER.

```
$cd /home/usuario
```

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

Observe que na pasta `/home/usuario/.ssh` surgiram dois arquivos: **id\_rsa** (chave privada) e **id\_rsa.pub** (chave pública).

Após a geração das chaves, é preciso copiar a chave pública para as outras máquinas.

De posse da chave pública, a máquina não solicitará senha para aquele usuário durante o ssh.

```
$scp /home/usuario/.ssh/id_rsa.pub <destino>:/home/usuario/.ssh/
```

Agora, efetue um ssh para a máquina de destino. Certamente, ainda será solicitada uma senha neste momento.

Então, execute o comando a seguir:

```
$cat /home/usuario/.ssh/id_rsa.pub >> /home/usuario/.ssh/authorized_keys
```

Isso fará com que a chave pública do usuário gerada anteriormente seja adicionada ao arquivo `authorized_keys`.

Agora, altere as permissões sobre o arquivo **authorized\_keys**, conforme mostrado abaixo:

```
$chmod 600 /home/usuario/.ssh/authorized_keys
```

Digite `exit` para voltar à máquina anterior e efetue novamente um ssh.

Se tudo estiver correto, não será solicitada a digitação de senha.

## REFERÊNCIAS BIBLIOGRÁFICAS

Gropp, W., Lusk, E., Skejellum, A. **Using MPI: portable parallel programming with the message-passing interface**, 2ed., Local: The MIT Press, 1999

Gropp, W., Lusk, E., Skejellum, A. **Using MPI-2: Advanced features of the message-passing interface**, 2ed., Local: The MIT Press, 1999

MPICH2, <http://www.mcs.anl.gov/research/projects/mpich2/index.php> acesso em 18-01-2008

## **PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE**

### **Teses e Dissertações (TDI)**

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### **Manuais Técnicos (MAN)**

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### **Notas Técnico-Científicas (NTC)**

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### **Relatórios de Pesquisa (RPQ)**

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### **Propostas e Relatórios de Projetos (PRP)**

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### **Publicações Didáticas (PUD)**

Incluem apostilas, notas de aula e manuais didáticos.

### **Publicações Seriadas**

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### **Programas de Computador (PDC)**

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

### **Pré-publicações (PRE)**

Todos os artigos publicados em periódicos, anais e como capítulos de livros.