

# IV Encontro Nacional LinuxChix Brasil

Balanceamento de carga em servidores WEB



**FUG-BR**  
Grupo Brasileiro de  
Usuários de FreeBSD  
[www.fug.com.br](http://www.fug.com.br)



Daniel Bristot de Oliveira  
[bristot@fug.com.br](mailto:bristot@fug.com.br)

# Introdução

A cada dia dependemos mais da *Web* para viver  
Processamento cada vez mais distribuído.

Transformação da Web  
De: Estática  
Para: Dinâmica

Um Servidor para vários clientes  
Processamento no lado do Servidor  
Ambiente Saturado.



# Um Novo Ambiente

## **Balanceamento de Carga entre Servidores Web**



# Vantagens/Desvantagens

- Menor processamento por nó
  - Modularidade do Sistema
  - Servidores para fins mais específicos
  - Servidores mais disponíveis
- 
- Maior latência de comunicação entre os servidores
  - Problemas com a comunicação
  - Inconsistência de dados.

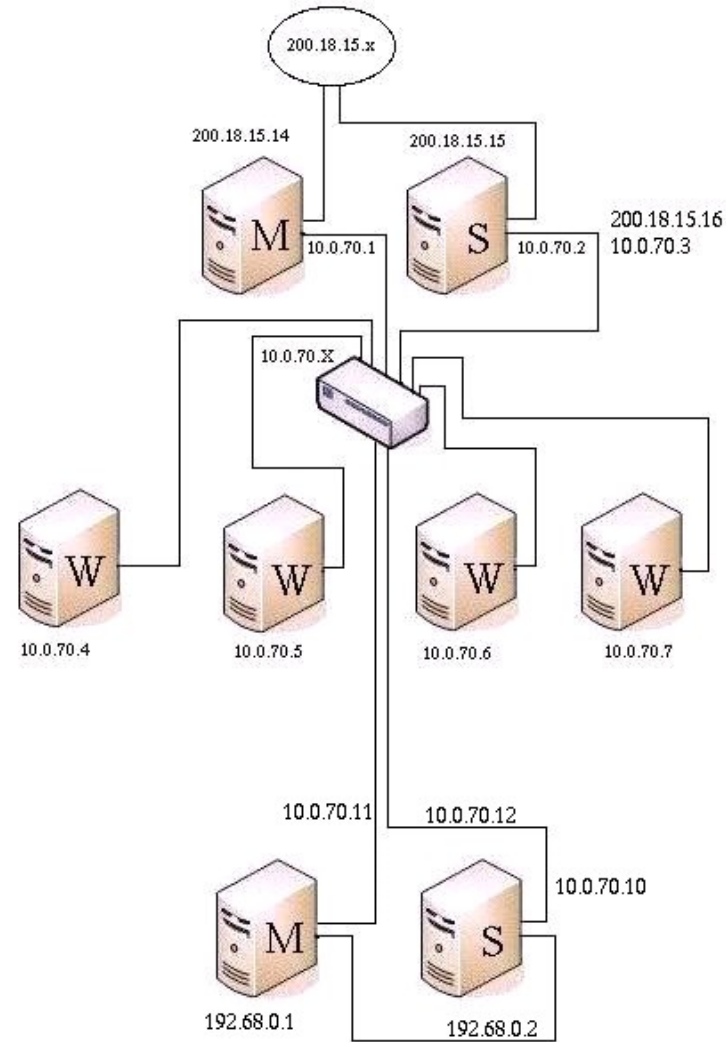


# O Ambiente

Roteadores

Servidores Web

Servidores de disco



# Os roteadores

Local: Entre a Internet e a rede local

Missão dos roteadores

Proteger a rede interna;

Receber as conexões destinadas aos servidores Web;

Testar a disponibilidade dos servidores Web;

Rotear, escalonar e distribuir os pacotes para os servidores Web;



# Ferramentas

**Hardware**, Duas interfaces de rede.

**Software:**

**Pen**, Balanceamento de carga.

**Carp**, Protocolo de redundância de endereço comum.

**Ifstated**, deamon de monitoramento de interfaces.



# O Pen

Um um *Balanceador de carga* para protocolos baseados em TCP;

Rastreamento de clientes;

*Persistência de conexões*, possibilitando o uso de aplicações baseadas em conexão;

Monitor de estado dos servidores Web;





# Como ele funciona?

Escuta a porta 80

Como um servidor Web, recebe os pacotes

Rastreia o cliente

Verifica a existência de uma conexão anterior

Escalonamento round-robin

Verifica o estado dos servidores

Persiste a conexão ou escalona

Estabelece a conexão



# Utilizando o Pen

## Sintax Básica:

```
# pen EndLocal:PortaLocal \  
EndereçoServidor1:Porta:LimiteDeConexões \  
EndereçoServidor2:Porta:LimiteDeConexões
```

## Exemplo:

```
# pen 200.18.15.16:80 \  
10.0.70.4:80:10 \  
10.0.70.5:80:20 \  
10.0.70.6:8080 \  
10.0.70.7
```



# Finalizando o Pen

Pronto temos o Pen configurado.

Balanceamento de carga

Tolerância a erros dos servidores Web

Um problema:

O roteador, se ele cair, todos ficarão como mortos para o resto do mundo.

**A solução?**



# CARP

*Carp, Protocolo de redundância de endereço comum*

Garantir a alta disponibilidade.

Protocolo de compartilhamento de endereços virtuais.

Grupo de roteadores.

Redundância de roteadores no mesmo segmento de rede.

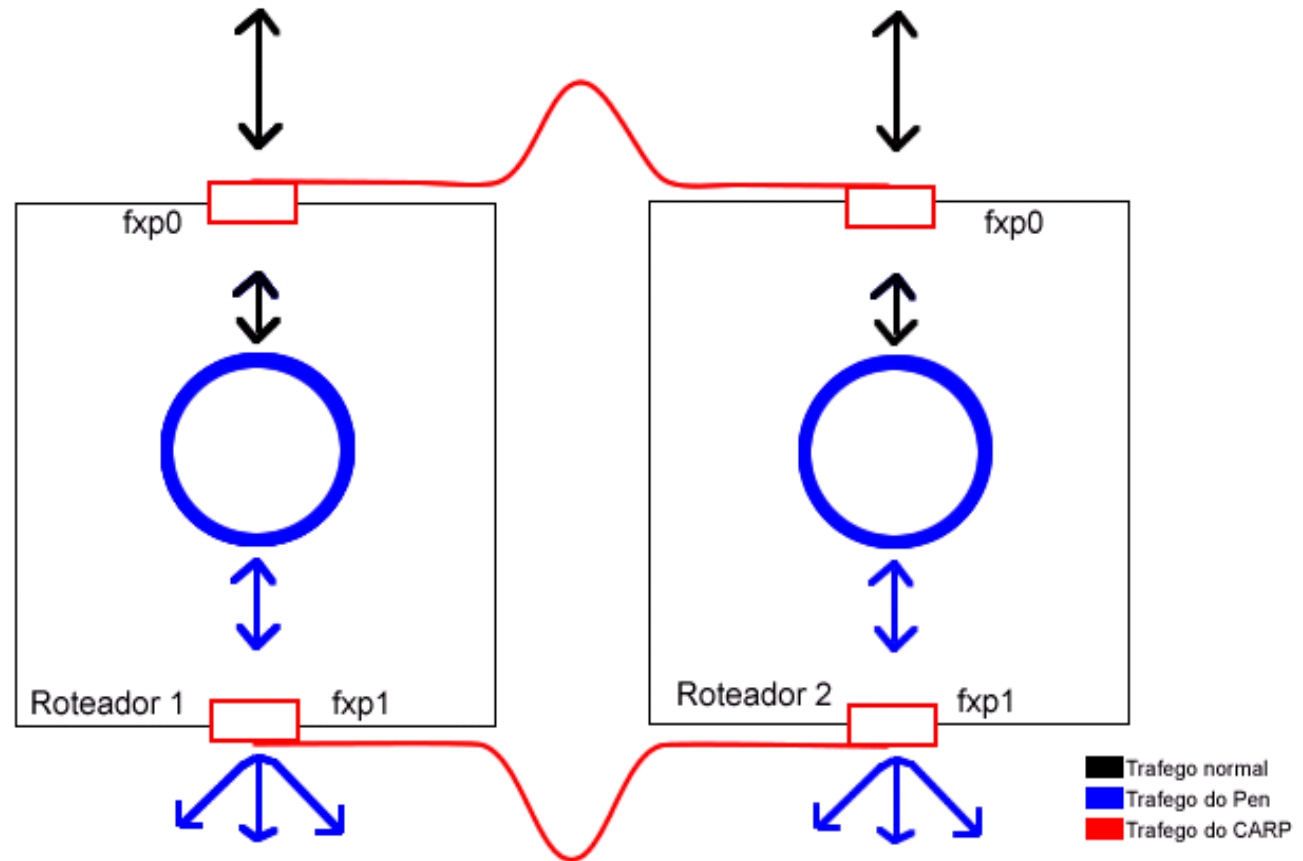


# O ambiente

**fxp0**

Real: 200.18.15.[14 | 15]

Virtual: 200.18.15.16



**fxp1**

Real: 10.0.70.[1 | 2]

Virtual: 10.0.70.3



# Como o carp funciona

Estados:

- Master .. 1
- Init .. Vários
- Backup .. Vários

Prioridades

Anúncios Broadcast

Mudança de estado

Preemptivismo



# Criando as interfaces

## Criando as interfaces:

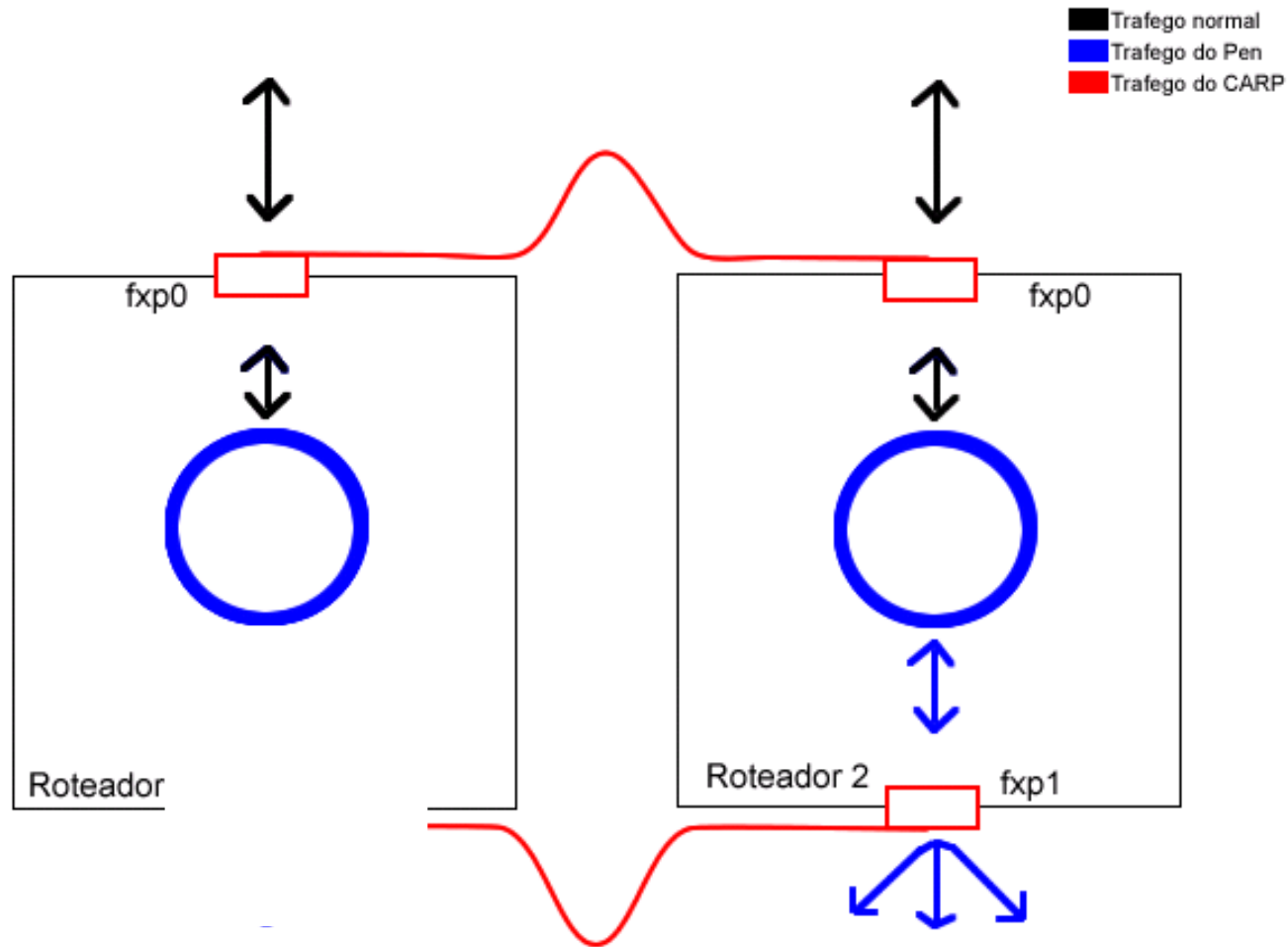
```
# ifconfig carp0 create  
# ifconfig carp1 create
```

## Configurando as interfaces:

```
# ifconfig carp0 vhid 1 advskew 0 pass senha1 200.18.15.16  
# ifconfig carp1 vhid 2 advskew 0 pass senha2 10.0.70.3  
# ifconfig  
carp0: flags=49<UP,LOOPBACK,RUNNING> mtu 1500  
    inet 200.18.25.16 netmask 0xffffffff00  
    carp: MASTER vhid 1 advbase 1 advskew 0  
carp1: flags=49<UP,LOOPBACK,RUNNING> mtu 1500  
    inet 10.0.70.3 netmask 0xff000000  
    carp: MASTER vhid 2 advbase 1 advskew 0
```



# Estágio atual





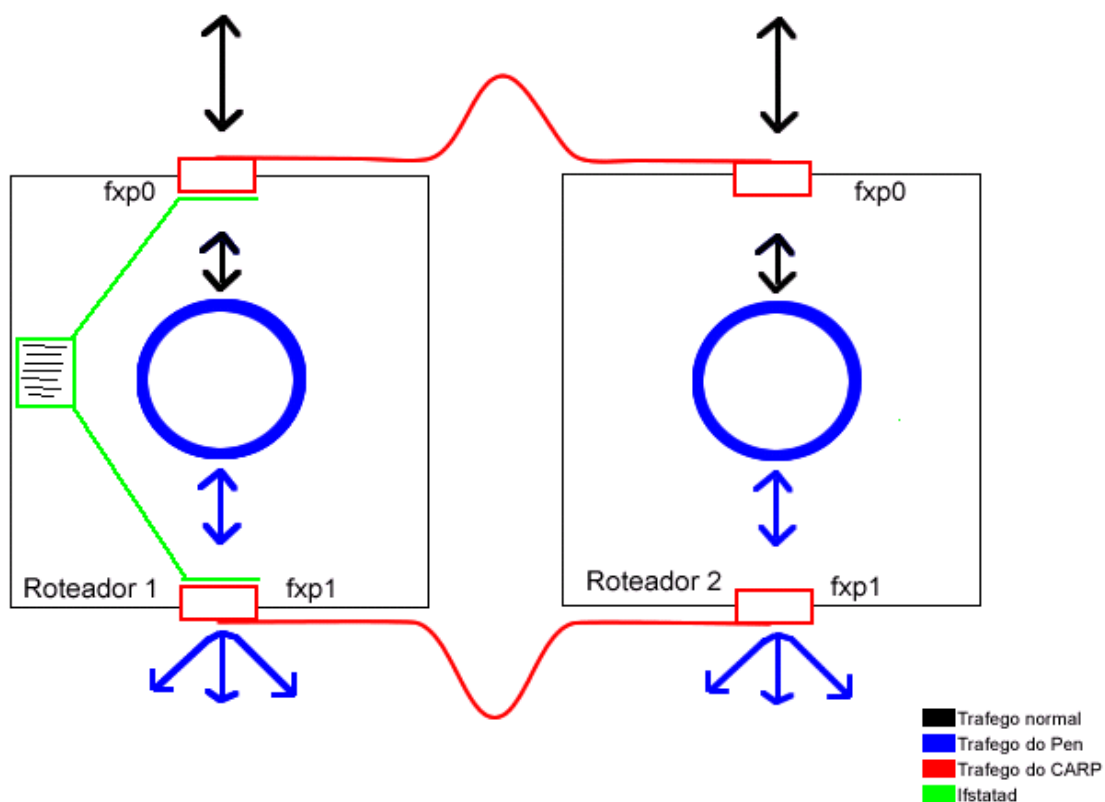
# O Ifstated(8)

*Daemon* de monitoramento de interfaces

Examinar as interfaces

Executa ações programadas em possíveis alterações de estado de interfaces

Irá fazer a ligação entre as interfaces.



# Exemplo

```
init-state one
```

```
if_up="fxp0.link.up && fxp1.link.up "
```

```
state one {  
    if ( ! $if_up ) {  
        run "ifconfig carp0 advskew 200"  
        run "ifconfig carp1 advskew 200"  
        set-state two  
    }  
}  
state two {  
    if ( $if_up ) {  
        run "ifconfig carp0 advskew 0"  
        run "ifconfig cap01 advskew 0"  
        set-state one  
    }  
}
```



# Observações

Muitas possibilidades, veja ifstaed.conf(5)

Habilitar somente primeiro roteador

Atenção as prioridades

Preempção do CARP habilitada:

```
# sysctl net.inet.carp.preempt=1
```



# Parte 2: Servidores Web

Parte principal do Sistema

Liga as três partes

Implementação mais fácil

Servidor Web

Lighttpd

Cliente NFS

Modularidade de hardware



# O Servidor Web Lighttpd

Compatível com sistemas Unix como os BSD, Linux...

Leve rápido e seguro

Virtual Hosting

Listagem virtual de diretórios

Re-escrita de URL, redirecionamento HTTP

Autenticação

FastCGI, CGI, SSI



# Discos no servidor Web

Simplex

Montados via NFS

Servidores de disco

*/etc/fstab*

```
10.0.70.10:/var/www/ /var/www nfs rw 0 0
```



# Parte 3: Servidores de disco

Última parte

Compartilhamento de dados entre os servidores Web.

Sistema de arquivos em rede, NFS.

Ponto único de falha.

Redundância

Espelhamento de dados entre os servidores de disco



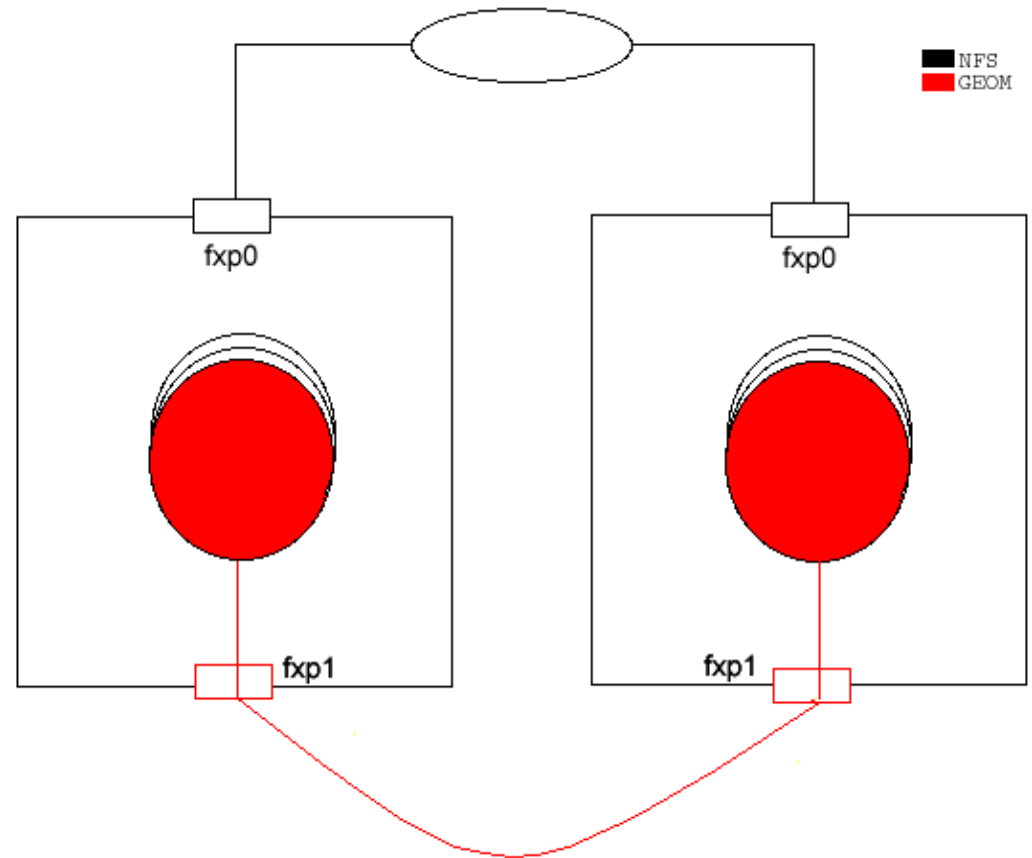
# Parte 3: Servidores de disco

Duas interfaces

Trafego da rede local

Crossover para espelhamento dos discos

Partição compartilhada do mesmo tamanho





# Ferramentas

Ferramentas:

Geom Gate: Exportação de dispositivos via Rede

ggated(8): Servidor

ggatec(8): Cliente

Geom Mirror espelhamento de disco

NFS: Network File System

Script de controle



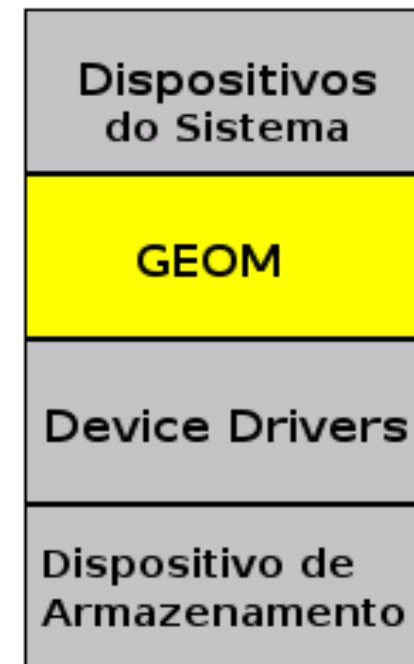
# O GEOM

Framework que possibilita fazer modificações no I/O dos disco.

Localizado entre o DevFS e os Device Drivers

Transparência nas transações

Várias possibilidades



# Classes do geom

Transformações são denominadas classes

Classe Mirror, espelhamento

Classe Gate, Importação/Exportação de Discos

Aplicativos para criar as classes

gmirror

ggatec/ggated

Um novo dispositivo “Virtual” e criado

/dev/ggate#

/dev/mirror/NOME



# Sincronização

Sincronização dos discos em tempo real.

Em um único Computador: RAID 1 ou espelhamento!

Em dois Computadores: RAID 1 em rede?

Isto é possível? ...



# Esquema

O servidor de backup exporta um disco para o mestre

O mestre importa do backup

O mestre cria um espelhamento do disco local no estrangeiro

RAID 1 em Rede! :)

Monta a partição

Exporta via NFS para os servidores Web



# Geom Gate

Exportação de dispositivos, `ggated(8)`

Importação de dispositivo, `ggatec(8)`

Por que não NFS?



# ggated

Exportará o disco do servidor em backup

Configurando:

```
/etc/gg.exports
```

```
192.68.0.0/24    RW /dev/ad0s1g
```

Iniciando serviço:

```
# ggated
# echo $?
0
```



# ggatec

Importará o disco do servidor de backup para o servidor ativo.

Para criar o dispositivo

```
# ggatec create 192.68.0.2 /dev/ad0s1g
```

Novo dispositivo criado...

```
/dev/ggate0
```





# Geom Mirror

Criar o espelhamento dos discos.

Dispositivo de dados:

`/dev/ad0s1g`

Espelhado para:

`/dev/ggate0`

Resultado: RAID 1 em rede e discos sincronizados em tempo real



# gmirror

gmirror na prática.

**Cria o mirror com o disco local**

```
# gmirror label -b prefer gm0 /dev/ad0s1g
```

**Novo dispositivo:**

```
/dev/mirror/gm0
```

```
mount /dev/mirror/gm0 /var/www
```

**Adiciona o Dispositivo estrangeiro ao mirror**

```
# gmirror insert -p 0 gm0 /dev/ggate0
```



# NFSD

Exportar para os servidores Web

Por que o NFS?

UDP.

Múltiplos acessos.

Transparência em uma queda.



# Configurando o NFSD

## O arquivo de configuração

```
10.0.0.0/8    /var/www
```

## Configuração do SO

*/etc/rc.conf*

```
nfs_server_enable= "YES"  
rpcbind_enable= "YES"
```



# Algoritmo

Ordem de execução.

**Disc1** = Servidor mestre

**Disc2** = Servidor escravo

1. *Disc1* - cria o mirror
  2. *Disc1* - Monta dispositivo
  3. *Disc1* - Lança NFSD
  4. *Disc1* - Assume o endereço Virtual
  5. *Disc2* - lança ggated
  6. *Disc1* - Importa disco
  7. *Disc1* - Adiciona ggate0 ao mirror
- Discos sincronizados...



# Algoritmo

Se o mestre cair...

1. *Disc2* - mata *ggated*
2. *Disc2* - Cria o *mirror*
3. *Disc2* - Checa a consistência do *mirror*
4. *Disc2* - Monta dispositivos
5. *Disc2* - Lança NFS
6. *Disc2* - Assume endereço Virtual
7. *Disc1* - Voltou...
8. *Disc1* - lança *ggated*
9. *Disc2* - Importa disco
10. *Disc2* - Adiciona *ggate0* ao *mirror*



# Final

Sistema configurado

Balancedores de carga, servidores Web e servidores de disco provendo alta disponibilidade, com algumas medidas de tolerância ao Erro

**PERGUNTAS?**



# Final

Documentação: <http://www.fug.com.br>

Meu Blog: <http://dbristot.info>

## **Agradecimentos:**

Universidade do Extremo Sul Catarinense

Amigos do FUG

Dirceu Tiegs: <http://dirceu.info>

