

# **Homologação de ZFS Data Storage Servers: Testes funcionais e não funcionais**

## Sumário

1 Testes de Homologação de ZFS Data Storage Servers.....	5
1.1 Objetivos.....	5
1.2 Preparação.....	5
1.3 Cronograma de execução:.....	6
1.3.1 Dia 1 .....	6
1.3.1.1 Apresentação dos recursos funcionais e não funcionais do ZFS Data Storage Server.....	6
1.3.1.2 Apresentação do plano de testes a ser praticado.....	6
1.3.1.3 Apresentação da documentação exigida por alguns requisitos não funcionais. ....	7
1.3.1.4 Descrição da massa de dados de e-mail a ser utilizada em cada teste. ....	7
1.3.1.5 Montagem da infra-estrutura e configuração inicial das máquinas de carga. ....	7
1.3.1.6 Primeiros testes rápidos para verificações preliminares de funcionamento. ....	7
1.3.1.7 Início dos testes funcionais.....	7
1.3.2 Dia 2 .....	7
1.3.2.1 Testes funcionais.....	7
1.3.3 Dia 3 .....	7
1.3.3.1 Testes funcionais. ....	7
1.3.4 Dia 4 .....	7
1.3.4.1 Testes não funcionais.....	7
1.3.4.2 Testes não funcionais desejáveis.....	7
1.3.5 Dia 5.....	8
1.3.5.1 Elaboração de relatório dos testes.....	8
1.3.5.1.1 Consolidação dos resultados. ....	8
1.3.5.1.2 Esclarecimentos sobre interpretações. ....	8
1.3.5.1.3 Apresentação de eventual cronograma de correção de problemas críticos.....	8
1.3.5.1.4 Nova execução de algum teste em que houve dúvida ou falha que já tenha recebido correção. ....	8
1.3.6 Elaboração do relatório de testes.....	8
1.4 Procedimentos de testes funcionais.....	8
1.4.1 Duração dos testes.....	9
1.4.2 Ferramentas de gerenciamento e verificação de desempenho .....	9
1.4.2.1 Comandos úteis para monitoração .....	10
1.4.3 Kernel tuning.....	10
1.4.3.1 Tuning de memória compartilhada no kernel, tcp/ip, max file descriptors, max open files .....	10
1.4.3.1.1 Adicionar no /etc/sysctl.conf dos backends .....	11
1.4.3.1.2 incluir no /etc/security/limits.conf de todas máquinas .....	11
1.4.3.1.3 incluir no /etc/pam.d/common-account .....	11
1.4.3.1.4 incluir no /etc/profile .....	12
1.4.3.2 Tuning para leitura de arquivos pequenos.....	12
1.4.3.2.1 Acrescentar no /etc/sysfs.conf .....	12
1.4.3.2.2 Acrescentar no /etc/sysctl.conf .....	13
1.4.4 Configurando o multipath .....	13
1.4.4.1 Descobrir qual placa HBA está instalada .....	13
1.4.4.2 Configurar APT para incluir drivers de rede e HBA QLogic.....	13
1.4.4.3 Configurar APT para incluir drivers de rede e HBA Emulex Light Pulse Fiber Channel .....	13
1.4.4.4 Descobrir o wwid para o /etc/multipath.conf .....	14
1.4.4.5 /etc/multipath.conf .....	14
1.4.4.6 Remover volumes e mapeamentos inválidos .....	15
1.4.4.7 Precisa configurar o /etc/lvm/lvm.conf corretamente. ....	15

1.4.4.8 Como descobrir o wwpn que o storage enxerga .....	16
1.4.4.9 Como descobrir o id da interface do storage .....	17
1.4.5 Configurando o filesystem.....	17
1.4.5.1 /etc/fstab .....	18
1.4.6 lozone .....	19
1.4.6.1 Open Indiana, Open Solaris, Illumos kernel .....	19
1.4.6.2 Debian GNU/Linux .....	19
1.4.6.2.1 Multithreaded: .....	19
1.4.6.2.1.1 Todos testes. ....	19
1.4.6.2.1.2 Testes de escrita. ....	19
1.4.7 Bonnie++ .....	20
1.4.7.1 Single threaded. ....	20
1.4.7.2 MULTIPLE PROCESSES .....	21
1.4.7.3 Multithreaded.....	21
1.4.7.3.1 Saída html:.....	21
1.4.7.3.2 Saída txt:.....	21
1.4.8 FIO over NFSv4 share.....	23
1.4.8.1 Multi threaded. ....	23
1.4.8.1.1 etc passwd .....	23
1.4.8.1.2 modificações /etc/hosts .....	23
1.4.8.1.3 /etc/hostname .....	23
1.4.8.1.4 server /etc/default/nfs-common .....	24
1.4.8.1.5 server etc hosts.allow .....	24
1.4.8.1.6 server /etc/idmapd.conf .....	24
1.4.8.1.7 server /etc/default/rpcbind .....	24
1.4.8.1.8 server /etc/exports .....	25
1.4.8.1.9 cliente modificações /etc/hosts .....	25
1.4.8.1.10 cliente /etc/hostname .....	25
1.4.8.1.11 client /etc/default/nfs-common .....	26
1.4.8.1.12 client etc hosts.allow .....	26
1.4.8.1.13 client /etc/idmapd.conf .....	26
1.4.8.1.14 client /etc/fstab/ .....	26
1.4.8.1.15 fio tests /home/andremachado/fio_randomrw_1kthreads_nfsv4.txt .....	27
1.4.8.1.16 run the test .....	28
1.4.9 Imaptest .....	28
1.4.10 Obter código fonte e compilar pacote do Imaptest .....	28
1.4.10.1 preparação para teste, instalação do cyrus imap .....	29
1.4.10.1.1 modificações no /etc/imapd.conf para o teste de carga .....	30
1.4.10.1.2 modificações no /etc/cyrus.conf .....	33
1.4.10.1.3 modificações no /etc/default/saslauthd .....	33
1.4.10.1.4 modificações no /etc/default/cyrus-imapd .....	33
1.4.10.1.5 exibir arquivos sem comentários nem linhas em branco .....	33
1.4.10.1.6 retorno a defaults dos databases do cyrus .....	35
1.4.10.2 preparação para teste, backport dos pacotes de Dovecot .....	35
1.4.10.2.1 /home/andremachado/usersimaptest.txt .....	36
1.4.10.3 execução teste com dataset menor da Dovecot .....	38
1.4.10.4 preparação para teste, dataset maior da Dovecot .....	39
1.4.10.5 preparação para teste, dataset da Enron .....	39
1.4.10.6 Monitoração de desempenho .....	41
1.5 Procedimentos de testes não funcionais.....	42
1.5.1 Failover automático e failback de unidades HEAD gerenciadoras. ....	42
1.5.1.1 Fork bomb:.....	42
1.5.1.2 Usar Dtrace:.....	43
1.5.1.3 Causar Non Maskable Interrupt NMI.....	43

1.5.1.4 Medições.....	43
1.5.2 Tolerância e recuperação de falhas individuais e múltiplas de dispositivos de blocos. .	44
1.5.2.1 Tolerância e recuperação de falhas transientes de rede. ....	44
1.5.2.2 Tolerância e recuperação de falhas transientes na alimentação. ....	44
1.5.2.3 Operações de snapshots, remoção de snapshots, replicação, backup e seus impactos no desempenho. ....	44
1.5.2.4 Migrações de dados entre servidores de armazenamento. ....	44
1.5.2.5 Tempos para recuperação de desastres (replicação e ciclo backup com restore). ....	45
1.5.2.6 Compatibilidade padrão open source (livremente incorporada no kernel oficial) com Debian GNU/Linux 7.x e superiores.....	45
1.5.2.7 Formato de sistema de arquivos ZFS em licença livre.....	45
1.5.2.8 Compatibilidade com virtualizadores XenServer e VmWare. ....	45
1.5.2.9 Desempenho sustentado sob carga contínua. ....	45
1.5.3 Requisitos não funcionais desejáveis:.....	46
1.5.3.1 Replicação rede local LAN e remota WAN.....	46
1.5.3.2 Expansão de capacidade de armazenamento.....	46
1.5.3.3 Flexibilidade na reconfiguração de recursos de LUNs e compartilhamentos de sistemas de arquivos.....	46
1.5.3.4 Interface aggregation.....	46
1.5.3.5 Acesso ao sistema .....	46
1.5.3.6 Compatibilidade com infraestrutura existente.....	46
1.5.3.7 Monitoração e relatórios.....	47
1.5.3.8 Armazenamento segregado.....	47
1.5.3.9 Auto-call .....	47
1.5.3.10 Manutenção e suporte técnico .....	47
1.5.3.11 Migração de Dados para outras soluções .....	47
1.6 Relatório dos testes.....	47
1.7 Atualizações deste documento:.....	48
1.8 Bibliografia .....	48
1.8.1 ZFS technology presentation, vendor independent: .....	49
1.8.2 E-mail sample data for loads .....	49
1.8.3 benchmarking software for imap .....	50
1.8.4 tips for storage benchmarking .....	51
1.8.5 NFSv4 tips .....	51
1.8.6 Como causar kernel panic no Solaris, Linux, FreeBSD.....	52
1.8.7 PostgreSQL benchmarking.....	52
1.8.8 Data sets.....	53

## **1 Testes de Homologação de ZFS Data Storage Servers**

O presente documento possui o objetivo de descrever o procedimento de testes para simular a utilização de um Data Storage Server no ambiente produtivo do Expresso.

### **1.1 Objetivos**

Testar a adequação dos equipamentos, desempenho e coletar resultados finais dos recursos tecnológicos frente aos requisitos funcionais e não funcionais de armazenamento do projeto Expresso BR para Governo Federal através de testes sintéticos práticos modelados em escala significativa à implantação final de armazenamento.

Será testada a adequação mínima para funcionamento para utilização em um dos serviços individuais usados pelo Expresso: Cyrus IMAP.

Verificaremos o grau da compatibilidade de hardware e software com a infraestrutura de armazenamento existente nos ambientes da empresa listados no edital: SAN, Ethernet, conforme detalhado nos demais itens.

### **1.2 Preparação**

O fornecedor previamente terá preparado o ambiente de testes conforme esta documentação e eventualmente outras documentações livres visando minimizar o tempo de ajustes e configurações para a execução.

Avançará o máximo possível nas configurações já indicadas nesta documentação, visando minimizar o prazo para início dos testes.

Idealmente, já terá executado todo o roteiro de testes independentemente e corrigido eventuais falhas, para que durante o evento de testes de prova de conceito com acompanhamento da nossa empresa tudo ocorra nos prazos e resultados esperados.

Necessitaremos máquinas para provocar carga sobre o ZFS Data Storage Server, compatíveis com Debian GNU/Linux 7.x (ou a mais recente estável), XenServer 6.1 (ou a mais recente estável) e VmWare 5.5 (ou a mais recente estável), ambos com máquinas virtuais Debian GNU/Linux 7.x (ou a mais recente estável).

Os servidores para carga serão equivalentes a máquinas com 40 núcleos mais HT, 512 GB RAM e disco local.

Deverá haver rede SAN compatível com os SAN FC Director, SAN FC Switch, SAN FC Backbone já utilizados na empresa.

Deverá haver rede Ethernet 10 Gb/s, com um segmento exclusivo para uso NFSv4 e iSCSI, preferencialmente em placas de interface dedicadas ao segmento.

As redes SAN FC e Ethernet deverão ser dimensionadas de tal forma a não limitar os resultados dos testes pela interconexão de equipamentos. Deverão haver ao menos 4 interfaces Ethernet 10 Gb/s e ou 4 interfaces FC-HBA 8Gb/s em cada head node, conforme os testes.

Deverá haver acesso aos repositórios oficiais do Projeto Debian.

Os softwares e versões serão os dos repositórios oficiais versão Estável do Projeto Debian, ou as especificadas nos procedimentos de teste deste documento, como as do wheezy-

backports ou em backport dos repositórios Testing ou Unstable oficiais, tal como PostgreSQL 9.2.x ou superior.

Deverão haver desktops multiboot, todos brasileiro português, Ubuntu 12.04 LTS, Debian GNU/Linux 7.x e Windows 7.x para acesso aos servidores e ambiente de testes. Todos atualizados com as mais recentes correções de segurança.

Deverão haver dois ZFS Data Storage Servers, para testes de migração de dados, com capacidade instalada mínima de 10 TB de espaço útil total líquido para dados finais.

Massas de dados de e-mail para testes (alguns com 100 GB e 1 TB) já deverão estar disponíveis e preparadas tanto quanto possível.

Uma alternativa é usar os e-mails de listas públicas, como os da Apache Foundation armazenados na Amazon <http://aws.amazon.com/datasets> ou alguma outra lista pública.

Para informação, o tamanho médio de nossos e-mails é 20 KB, variando de 1 KB a 50 MB, com muitas mensagens com anexos de documentos LibreOffice e MS-Word, arquivos zip e imagens jpeg e png.

Sugerimos a disponibilização de equipe adicional para auxiliar na coleta e formatação dos dados (registros, logs, tabelas, gráficos) das máquinas visando acelerar a elaboração de relatório final de testes.

Deverá ser utilizada compressão máxima de dados armazenados.

O sistema de arquivos raiz será ext3. As montagens onde estarão os dados serão em sistema de arquivos XFS criadas e montadas conforme instruções ou em compartilhamentos NFSv4 montados conforme instruções.

### **1.3 Cronograma de execução:**

Sugestão de cronograma de execução. Pode ser alterado conforme constatações de durações de testes e conveniência de recursos.

#### **1.3.1 Dia 1**

##### **1.3.1.1 Apresentação dos recursos funcionais e não funcionais do ZFS Data Storage Server.**

##### **1.3.1.2 Apresentação do plano de testes a ser praticado.**

Especialmente nos requisitos que dependem da implementação particular proposta de cada fornecedor e que não foram detalhados neste documento.

#### **1.3.1.3 Apresentação da documentação exigida por alguns requisitos não funcionais.**

#### **1.3.1.4 Descrição da massa de dados de e-mail a ser utilizada em cada teste.**

#### **1.3.1.5 Montagem da infra-estrutura e configuração inicial das máquinas de carga.**

#### **1.3.1.6 Primeiros testes rápidos para verificações preliminares de funcionamento.**

#### **1.3.1.7 Início dos testes funcionais.**

Alguns testes funcionais que requeiram muitas horas para conclusão podem ser iniciados já no fim do primeiro dia. Também dependem da quantidade de ZFS Data Storage Servers disponíveis e de máquinas geradoras de carga.

### **1.3.2 Dia 2**

#### **1.3.2.1 Testes funcionais.**

Testes funcionais sobre máquinas físicas com Debian GNU/Linux.

Planejar o início de testes de longa duração para o fim do expediente, de forma a coletar resultados no início do dia seguinte.

### **1.3.3 Dia 3**

#### **1.3.3.1 Testes funcionais.**

Testes funcionais em máquinas virtuais sobre XenServer e VmWare.

Havendo possibilidade, adiantar o início dos testes não funcionais.

Planejar o início de testes de longa duração para o fim do expediente, de forma a coletar resultados no início do dia seguinte.

### **1.3.4 Dia 4**

#### **1.3.4.1 Testes não funcionais.**

Alguns testes não funcionais que requeiram muitas horas para conclusão podem ser iniciados ao final do expediente ainda no prazo dos testes funcionais.

Planejar o início de testes de longa duração para o fim do expediente, de forma a coletar resultados no início do dia seguinte.

#### **1.3.4.2 Testes não funcionais desejáveis.**

Se houver disponibilidade de tempo.



### **1.3.5 Dia 5**

#### **1.3.5.1 Elaboração de relatório dos testes.**

##### **1.3.5.1.1 Consolidação dos resultados.**

##### **1.3.5.1.2 Esclarecimentos sobre interpretações.**

##### **1.3.5.1.3 Apresentação de eventual cronograma de correção de problemas críticos.**

##### **1.3.5.1.4 Nova execução de algum teste em que houve dúvida ou falha que já tenha recebido correção.**

#### **1.3.6 Elaboração do relatório de testes.**

Utilizaremos o LibreOffice com formato ODT e pacote em português brasileiro na redação.

### **1.4 Procedimentos de testes funcionais**

Os testes descritos abaixo visam avaliar o comportamento de data storage servers no ambiente produtivo do Expresso, simulando a forma como é feita a leitura/escrita dos dados pelos serviços do Expresso. Os parâmetros descritos precisam ser reajustados para o *hardware* disponível, uma vez que os parâmetros de tuning citados estão configurados para a infraestrutura atual do Expresso BR.

Para a execução dos testes, serão necessárias máquinas geradoras de carga cuja memória RAM seja aproximadamente 2x (duas vezes) maior que o cache do data storage server, possibilitando a avaliação do desempenho do equipamento em situações onde as operações solicitadas excedam o tamanho do cache do data storage server, como em carga sustentada.

A ferramenta em que pudemos modelar, com a maior fidelidade possível, o comportamento do Cyrus IMAP sob alta carga é o *fio*. Mesmo assim, essa ferramenta e o *imaptest*, explicado a seguir neste documento, dificilmente conseguirão exaurir os caches de poucas máquinas geradores de carga e de um grande data storage server. Para exaurir os caches das máquinas e do data storage server, testes com *bonnie++* e *iozone* serão mais adequados.

Os outros testes são mais conhecidos e servem para denunciarem problemas específicos, que precisam ser resolvidos antes de prosseguir, e que até podem eliminar propostas de soluções de armazenamento, elementos da cadeia de hardware e software ou mesmo tunings. Sugerimos até realizar os testes na mesma sequência do artigo: *iozone*, *bonnie*, *fio*, *imaptest*.

Os erros gerados durante a execução dos testes serão tratados como defeitos críticos e impeditivos no processo de aquisição.

O teste funcional com *imaptest* consegue homologar contra erros de protocolo e fazer um benchmark do servidor imap com toda a pilha de hardware e software envolvida.

O teste com *imaptest*, devido à sua concepção, provavelmente não medirá os ganhos de



desempenho com indexação noturna de emails antigos feitos pelo squatter do cyrus imap. As mensagens serão enviadas, lidas e removidas durante o teste, não havendo oportunidade para a indexação noturna que ocorre em caixas postais de produção.

Isso reduzirá a atividade IOPS nos volumes de metadados (índices na meta-partition), diferente do que acontece em uso real, que medimos como variando entre 70% a 80% da atividade total, e oscilando entre 80% e 90% de proporção de escritas de arquivos pequenos típicos de metadados.

#### 1.4.1 Duração dos testes

Os testes terão duração mínima de 1 hora para coleta de resultados ao relatório final. Todavia, testes com duração mínima de 8 horas serão executados, avaliando o desempenho sustentado sob carga contínua.

Esses testes mais longos poderão ser planejados para iniciarem ao final de um dia de testes, e continuar sua execução durante a noite, permitindo a coleta das informações na manhã seguinte.

#### 1.4.2 Ferramentas de gerenciamento e verificação de desempenho

Os pacotes abaixo estão disponíveis nos repositórios oficiais da distribuição Debian GNU/Linux 7.x. Essas ferramentas serão utilizadas durante a execução dos testes para gerar as atividades no disco e monitorar o comportamento do data storage server sob carga.

```
# apt-get install jnettop iotop htop atop sysstat bonnie++ fio iozone3 \
tiobench iftop nload pktstat tcptrack psmisc pslist procs procinfo ips \
conntack iptables-persistent dnstop postmark tcpdump arping iwatch iperf \
rcconf chkconfig sysv-rc-conf wajig tshark apachetop ntp zabbix-agent \
xfsprogs xfsdump sudo openssh-server openssh-client openssh-blacklist \
inotify-tools irqbalance sysfsutils util-linux debsums dstat \
ncdu attr acl quota binutils stress haveged screen nethogs snoopy collectl
```

Habilitar sar no /etc/default/sysstat .

Se tiver postgresql, instalar visualizador de desempenho também

```
# apt-get install ptop pg-activity pgfouine check-postgres
```

As mais utilizadas nesse teste poderão ser:

- *jnettop*
- *iotop*
- *atop*
- *sysstat*
- *bonnie++*

- *fio*
- *iozone3*
- *tiobench*
- *stress*
- *nload*

#### 1.4.2.1 Comandos úteis para monitoração

```
iostat -dmht 1 100000
iostat -dmhtx 1 100000
sar -dp
sar -f /var/log/sysstat/sa06 -d
sar -f /var/log/sysstat/sa06 -q
iotop
vmstat -d
vmstat -D
atop
top
htop
```

#### 1.4.3 Kernel tuning

Para melhorar a resposta ao usuário e utilizar da melhor forma os equipamentos disponíveis, são necessário a intervenção técnica em parâmetros de funcionamento do *kernel* do Debian. Os parâmetros listados a seguir são utilizados na infraestrutura atual do Expresso e devem ser revisados na realização da Prova de Conceito para novos Data Storage Servers.

O objetivo das alterações a seguir é obter a menor latência possível e o maior número de operações por segundo no data storage server, penalizando, para isso, o desempenho throughput (taxa de transferência) da máquina cliente.

Talvez, apenas a mudança de escalonador (*scheduler*) de *io*, descrita abaixo, seja necessária para saturar o equipamento a ser testado, mas será necessário avaliar esse comportamento.

```
echo deadline > /sys/block/sda/queue/scheduler
```

Até mesmo o kernel pode ser necessário usar o 3.12 ou maior do repositório backports.

Será necessária a alteração de outros parâmetros, como o número máximo de descritores de arquivos, limite memória compartilhada, entre outros.

##### 1.4.3.1 Tuning de memória compartilhada no kernel, tcp/ip, max file descriptors, max open files

Para suportar a carga, precisamos ajustar parâmetros de kernel, de pilha TCP/IP e de número máximo de descritores de arquivos e de arquivos abertos.

#### 1.4.3.1.1 Adicionar no /etc/sysctl.conf dos backends

```
# maximum number of file-handles that the Linux kernel will allocate
# 256 for every 4M de RAM
#AFM 20140404 hard coded limit
fs.file-max = 1048576
# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736
# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296
net.ipv4.ip_local_port_range = 15000 61000
net.ipv4.tcp_fin_timeout = 10
#net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_low_latency=1
```

Para aplicar as alterações descritas no arquivo, basta executar o comando:

```
sysctl -p
```

#### 1.4.3.1.2 incluir no /etc/security/limits.conf de todas máquinas

```
@adm soft nfile 1024000
@adm hard nfile 1024000
* soft nfile 1024000
* hard nfile 1024000
```

#### 1.4.3.1.3 incluir no /etc/pam.d/common-account

Não é necessário no /etc/pam.d/common-session pois os outros scripts de serviços sempre carregam o /etc/pam.d/common-account.

```
#AFM 20110915
session required pam_limits.so
```

#### 1.4.3.1.4 incluir no /etc/profile

Não adianta incluir no /etc/bashrc ou /etc/bash.bashrc pois estes são para shell interativo. Precisamos para serviços iniciados por shell não interativo.

É preciso fazer as alterações nos outros arquivos para que o usuário possa alterar os próprios limites, até o valor máximo configurado para todo sistema. E /etc/profile vale para todos usuários, inclusive os dos serviços.

```
#AFM 20110916
ulimit -n 1024000
```

#### 1.4.3.2 Tuning para leitura de arquivos pequenos

Uma vez que o Cyrus IMAP cria um arquivo para cada mensagem recebida pelo usuário, é necessário verificar a latência da leitura/escrita de arquivos pequenos, normalmente com tamanhos inferiores a 4KB, tamanho padrão de um bloco no disco.

Para isso, é necessário instalar o pacote sysfsutils, que é responsável por verificar possíveis alterações de parâmetros em relação ao discos em utilização na máquina. Segue as alterações realizadas no /etc/sysfs.conf. No caso do arquivo abaixo, o disco xvdb é utilizado para as mensagens das caixas postais e o xvdc é utilizado como meta-partição, ou seja, possui os índices de cada caixa existente no Cyrus IMAP.

##### 1.4.3.2.1 Acrescentar no /etc/sysfs.conf

Adaptar ao seu mapeamento de blocos.

Recomendável iniciar apenas com a mudança de scheduler, e talvez em seguida o vm.dirty\_expire\_centisecs, e deixar os tunings mais agressivos para o caso de não ser possível saturar o I/O de outra forma.

```
#AFM 20120523
#block/xvdb/queue/nr_requests = 4
block/xvdb/queue/scheduler = deadline
#block/xvdb/queue/iosched/front_merges = 1
#block/xvdb/queue/iosched/fifo_batch = 1

#AFM 20120523
#block/xvdc/queue/nr_requests = 4
block/xvdc/queue/scheduler = deadline
#block/xvdc/queue/iosched/front_merges = 1
#block/xvdc/queue/iosched/fifo_batch = 1
```

#### 1.4.3.2.2 Acrescentar no /etc/sysctl.conf

```
#AFM 20120523
vm.swappiness = 10
vm.dirty_background_ratio = 1
vm.dirty_expire_centisecs = 500
vm.dirty_ratio = 15
vm.dirty_writeback_centisecs = 100
```

Aplicar modificações:

```
sysctl -p
```

#### 1.4.4 Configurando o multipath

Para evitar que problemas na fibra afetem a comunicação com o Data Storage Server, é utilizado o multipath. Nesse caso, são configuradas “n” conexões entre as máquinas e o Storage Server, possibilitando que, em caso de falhas de uma das fibras, o tráfego seja direcionado para outra fibra que esteja funcional. Para isso, é necessário configurar o serviço multipathd, é necessário primeiramente realizar a configuração das placas HBA's no Sistema Operacional nativo e depois configurar o multipathd.

##### 1.4.4.1 Descobrir qual placa HBA está instalada

```
lspci |grep QLogic
lspci |grep Emulex
```

##### 1.4.4.2 Configurar APT para incluir drivers de rede e HBA QLogic

É preciso seguir minuciosamente os passos em [[Configurar APT no Debian ]]? e somente instalar esses drivers se forem as placas instaladas. Se forem outras, irá travar o kernel na sequência de boot.

```
# apt-get install multipath-tools-boot multipath-tools firmware-qlogic firmware-
bnx2 sysfsutils scsiboot
# reboot
```

##### 1.4.4.3 Configurar APT para incluir drivers de rede e HBA Emulex Light Pulse Fiber Channel

O driver Emulex Light Pulse Fiber Channel (lpfc) já está incorporado no kernel Debian.

É preciso habilitar a seção non-free do repositório oficial Debian e somente instalar esses drivers se forem as placas de rede e HBA instaladas. Se forem outras, irá travar o kernel na sequência de boot.

```
# apt-get install multipath-tools-boot multipath-tools firmware-bnx2 sysfsutils  
scsitools  
# reboot
```

Sem reboot, não irá reconhecer corretamente as placas nem remapear corretamente o device-mapper. Sem a instalação dos pacotes corretos, mesmo se forem da seção non-free, não atualizará kernel corretamente no futuro.

Mesmo que durante a instalação por DVD tenham sido detetadas as placas, só o kernel instalado foi configurado mas não serão possíveis atualizações de segurança que gerem novo initrd por falta dos pacotes instalados.

#### 1.4.4.4 Descobrir o wwid para o /etc/multipath.conf

Como ainda não está configurado corretamente, o comando abaixo irá retornar "undef" para alguns paths, como no exemplo abaixo. O que precisamos agora é encontrar o wwid, que fica entre parênteses.

```
multipath -l
```

#### 1.4.4.5 /etc/multipath.conf

adaptar ao cenário local. Algumas vezes pode ficar como /dev/mapper/mpath0

```
multipaths {  
    multipath {  
        wwid 360a98000572d4f61614a5a7235576575  
# alias disk1_4TB  
        path_grouping_policy failover  
        path_selector "round-robin 0"  
    }  
  
    multipath {
```

```
wwid 360024e805520f9001270b63307b03f4e
# alias disk0_135GB
path_grouping_policy failover
path_selector "round-robin 0"
}
}

devnode_blacklist {
    devnode "*"
}

blacklist {
devnode "sda"
}
```

#### 1.4.4.6 Remover volumes e mapeamentos inválidos

Se não remover os volumes lvm, o multipath -F não vai liberar os /dev/mapper/mpath\*

O comando abaixo remove os mapeamentos de lvm QUE NÃO EXISTAM MAIS e estejam inativos, para liberar o multipath -F e permitir remapeamento.

Também precisa mover para outro lugar o /var/lib/multipath/bindings

```
# mv /var/lib/multipath/bindings /var/lib/multipath/bindings.bak
```

Senão voltará a ler sempre os mesmos parâmetros.

Ou edite-o e copie-o para as outras máquinas que montarão o storage. Depois, limpar os lvm fantasmas.

```
# dmsetup remove grupol-labpostfix--exp--02
```

#### 1.4.4.7 Precisa configurar o /etc/lvm/lvm.conf corretamente.

Procure o trecho no lvm.conf similar ao trecho abaixo e configure-o similar ao exemplo, mas adequado ao seu servidor que pode ter mais unidades LVM mapeadas em discos locais comuns, fora do storage.

```
# This is an example configuration file for the LVM2 system.
```



```
# By default we accept every block device:
# AFM 16out2009 trying to filter duplicate volumes allowing only dm
#filter = [ "a/.*/" ]
# filter = [ "a|/dev/dm(\-[0-9])|", "r/sd.*/" , "r|.*/" ]
# AFM 22out2009 use correct multipath devices instead of dm-X
filter = [ "a|^/dev/mapper/mpath.*|" , "r|.*/" ]
```

```
invoke-rc.d multipath-tools stop
multipath -F
reboot
```

Exemplos de comandos úteis para remover mapeamentos inválidos. Estude as man pages. Você só deverá usá-los se houverem mapeamentos inválidos, e com extremo cuidado.

```
# vgreduce --test --removemissing vg01
# dmsetup ls --tree
# dmsetup info grupol-lvstripe14
# dmsetup info mpath0
# dmsetup targets
# dmsetup table --target multipath mpath0
# dmsetup table --target striped grupol-lvstripe14
```

Em seguida, depois de gerar nova imagem initrd de boot

```
update-initramfs -u
```

TEM DE FAZER REBOOT

#### 1.4.4.8 Como descobrir o wwpn que o storage enxerga

Para configuração no storage, é preciso saber o identificado que a HBA expõe.

```
cat /sys/class/fc_host/host*/port_name
0x2100001b328f8e2c
0x2101001b32af8e2c
0x2100001b32932bff
```

#### 1.4.4.9 Como descobrir o id da interface do storage

```
cat /sys/class/fc_remote_ports/rport-*/port_name
0x500a098389db4661
0x500a098489db4661
0x500a098399db4661
0x500a098499db4661
0x500a098189db4661
0x500a098289db4661
0x500a098199db4661
0x500a098299db4661
```

#### 1.4.5 Configurando o filesystem

Nos discos utilizados diretamente pelo Cyrus IMAP, como o disco em que estão os índices e o disco em que estão os dados, é utilizado o XFS, uma vez que esse tipo de sistema de arquivo se mostrou eficiente para vários acessos simultâneos, como é o caso do Cyrus IMAP.

```
apt-get install xfsdump xfsprogs attr lvm2
```

```
pvccreate /dev/mapper/mpath0
pvdisplay
vgcreate --autobackup y grupo01 /dev/mapper/mpath0
lvcreate --autobackup y --readahead auto --extents 100%FREE --name lv01 grupo01
```

O internal log de metadados pode ser expandido até preencher um Allocation Group.

Isso PODERÁ garantir um melhor comportamento sobre bursts de gravação e reduzir a latência do sistema de arquivos.

No exemplo abaixo, para um AG de 256 MB, usamos um log de metadados de 255 MB, o máximo que foi aceito NESSE CASO.

O Allocation Group deve ser dimensionado para a quota do usuário, para melhor desempenho e menor fragmentação.

Hoje faremos com 10 GB em vez do exemplo abaixo de 256 MB.

```
mkfs -t xfs -l internal,lazy-count=1,size=255m -d agsize=256m -b size=4096
-f /dev/grupo01/lv01
```

```
meta-data=/dev/grupo01/lv01      isize=256    agcount=2, agsize=65536 blks
      =                               sectsz=512    attr=2, projid32bit=0
data      =                               bsize=4096    blocks=131072, imaxpct=25
      =                               sunit=0      swidth=0 blks
naming    =version 2                bsize=4096    ascii-ci=0
log       =internal log             bsize=4096    blocks=65280, version=2
      =                               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                     extsz=4096    blocks=0, rtextents=0
```

```
mount \
-o norelatime,noatime,nodiratime,attr2,nobarrier,logbufs=8,logbsize=256k,\
osyncisdsync /dev/grupo01/lv01 /data

df -h
```

#### 1.4.5.1 /etc/fstab

Atenção que há quatro quebras de linha apenas para exibição na tela.

```
# /etc/fstab: static file system information.
#AFM 22oct2010
# <file system> <mount point>    <type>  <options>          <dump>  <pass>
proc                /proc                proc     defaults           0        0
/dev/disk/by-id/scsi-360026b9038e754001270c9de079bca7c-part3
ext3      errors=remount-ro 0      1
/dev/disk/by-id/scsi-360026b9038e754001270c9de079bca7c-part1
ext3      defaults          0      2
/dev/disk/by-id/scsi-360026b9038e754001270c9de079bca7c-part2
swap      sw                0      0
/dev/scd0        /media/cdrom0      udf,iso9660 user,noauto        0        0
/dev/grupo1/lv01 /data      xfs
norelatime,noatime,nodiratime,attr2,nobarrier,logbufs=8,logbsize=256k,
osyncisdsync      0        0

/dev/grupo2/lv01 /indicesimap      xfs
norelatime,noatime,nodiratime,attr2,nobarrier,logbufs=8,logbsize=256k,
osyncisdsync      0        0
```

### 1.4.6 Iozone

As opções escolhidas farão executar no modo automático todos os testes de tamanho de gravação de 4KB a 16MB em arquivos de 64KB a 512MB, forçando testar gravações pequenas para todos os arquivos, com saída em operações por segundo, usando DIRECT\_IO onde possível, acessos aleatórios, escritas síncronas O\_SYNC, com tamanho máximo de arquivos em 2GB (você PRECISA ajustar esse tamanho para ficar maior que a memória RAM disponível), gravando saída em arquivo especificado.

Atenção nas opções de teste multithreaded descritas, limitadas ao número de núcleos e especificando os testes a executar pois não aceitou modo automático -a.

#### 1.4.6.1 Open Indiana, Open Solaris, Illumos kernel

Demora algumas horas rodar todos os testes com opção -a.

Single threaded por default mas pode ser configurado para multi threaded.

<http://www.it-sudparis.eu/s2ia/user/procacci/Doc/NFS/nfs014.html>

```
/usr/benchmarks/iozone/iozone -a -z -O -I -K -o -g 2G -b \  
/root/iozoneopenindianazfstestallsync2blockz2gbfile.xls
```

#### 1.4.6.2 Debian GNU/Linux

```
iozone -a -z -O -I -K -o -g 2G -b /tmp/iozonetmp.xls
```

##### 1.4.6.2.1 Multithreaded:

###### 1.4.6.2.1.1 Todos testes.

64M pelo tamanho de anexos, mas relacionar com quantidades de threads e memória ram.

-t quantidade de núcleos da máquina geradora de carga.

Se sobrar ram, aumentar tamanho do arquivo, não threads.

```
iozone -i 0 -i 1 -i 2 -i 3 -i 4 -i 5 -i 6 -i 7 -i 8 -i 9 -i 10 -i 11 -i 12 \  
-z -O -I -t 4 -Q -K -o -r 4 -s 64M -b /home/andremachado/iozonenfsimports.xls
```

###### 1.4.6.2.1.2 Testes de escrita.

64M pelo tamanho de anexos.

-t deve ter o número de núcleos da máquina geradora de carga:

```
iozone -i 0 -i 2 -i 9 -i 11 -z -O -I -t 4 -Q -K -o -r 4 -s 64M -b  
/home/andremachado/iozonenfsimports.xls
```

#### 1.4.7 Bonnie++

Preferência pelo teste multithreaded com opção -c.

Usaremos escritas não bufferizadas, execução como usuário root (0), neste caso com apenas 4 threads para 4 núcleos, e tamanho de 3 GB que precisa ser ajustado pelo tamanho da RAM.

O parâmetro de número de arquivos -n 100:200K:10K:128:4096 é entendido como: 100 \* 1024 arquivos, variando entre 200KB tamanho máximo e 10KB tamanho mínimo, distribuídos em 128 diretórios, escritos em porções de 4096 bytes. Esse parâmetro também precisa ser ajustado para simular uma carga de backend imap, o que pode fazer o teste durar horas.

Sugestão -n **10000:150K:1K:1024:4096**.

Atenção, pois isso criará **10 milhões** de arquivos.

Os diretórios e arquivos são criados quando alcança a etapa *start 'em*:

```
Using uid:0, gid:0.  
Writing a byte at a time...done  
Writing intelligently...done  
Rewriting...done  
Reading a byte at a time...done  
Reading intelligently...done  
start 'em...done...done...done...done...done...
```

##### 1.4.7.1 Single threaded.

O tamanho dos arquivos de teste deve ser maior que o dobro da memória RAM do servidor gerador de carga. O bonnie nem aceita rodar com menos.

O tamanho e as latências da cadeia de armazenamento podem impactar muito na duração de teste. O bonnie pode demorar várias horas.

Para melhor simular condições de servidores de email ou database, usar opção -b.

```
time /usr/sbin/bonnie++ -b -d /mnt/storage -s 32G -n 100:200K:10K:128:4096 -x 1 |  
bon_csv2html
```

Poderia ser testado em multi threaded se:

#### 1.4.7.2 MULTIPLE PROCESSES

Run the following commands to run three copies of Bonnie++ simultaneously:

```
bonnie++ -p3
bonnie++ -y > out1 &
bonnie++ -y > out2 &
bonnie++ -y > out3 &
```

#### 1.4.7.3 Multithreaded

A opção -c dispara o número de threads especificadas para escreverem e lerem no disco em várias das etapas de teste. Ocupa os núcleos do processador. Versões antigas podem não suportar a opção -c.

##### 1.4.7.3.1 Saída html:

```
time /usr/sbin/bonnie++ -b -d /mnt/nfsimports -c 4 -s 3G -n 100:200K:10K:128:4096
-x 1 | bon_csv2html > /home/andremachado/bonnie_nfsimports.html
```

##### 1.4.7.3.2 Saída txt:

```
sudo time /usr/sbin/bonnie++ -b -u 0 -d /mnt/nfsimports -c 4 -s 3G -n
10000:150K:1K:1024:4096 -x 1 | bon_csv2txt >
/home/andremachado/bonnie_nfsimports_20131220.txt
```

Analisar o conteúdo do arquivo de saída, no caso o  
/home/andremachado/bonnie\_nfsimports.html, abrindo em navegador web.

Analisar o conteúdo dos arquivos relatórios de latência, como  
/mnt/nfsimports/Child\_2\_randwol.dat em busca de problemas. Os resultados abaixo são apenas ilustrativos de formato, pois obtidos numa máquina desktop.

Offset in Kbytes	Latency in microseconds	Transfer size in bytes
264	548	4096
708	2332	4096
520	529	4096
508	461	4096
224	701	4096
932	4409	4096
420	6481	4096

20	1726	4096
652	954	4096
700	549	4096
288	561	4096
628	461	4096
236	450	4096
148	750	4096
936	871	4096
408	660	4096
704	477	4096
516	384	4096

Exemplo de formato de saída do bonnie++. Os resultados abaixo são apenas ilustrativos de formato, pois obtidos numa máquina desktop.

Version	1.96	-----Sequential Output-----						--Sequential Input-				--Random-	
		-Per Chr-		--Block--		-Rewrite-		-Per Chr-		--Block--		--Seeks--	
Machine	Size	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	/sec	%CP
debian64bits	3G	726	96	16322	82	4836	91	2391	89	9811	98	561.5	189
Latency		160ms		5641ms		3732ms		57890us		144ms		480ms	
		-----Sequential Create-----						-----Random Create-----					
		-Create--		--Read---		-Delete--		-Create--		--Read---		-Delete--	
files:max:min		/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP
debi 200:102400:100		230	32	43	10	563	16	191	35	37	11	425	18
Latency		22838ms		2024ms		8076ms		19737ms		932ms		7313ms	

```
echo deadline > /sys/block/sda/queue/scheduler
cat /sys/block/sda/queue/scheduler
noop [deadline] cfq
```

Version	1.96	-----Sequential Output-----						--Sequential Input-				--Random-	
		-Per Chr-		--Block--		-Rewrite-		-Per Chr-		--Block--		--Seeks--	
Machine	Size	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	/sec	%CP
debiantesting64	2G	520	99	99693	26	70418	44	3254	90	267854	97	3490	473
Latency		37405us		423ms		379ms		82940us		17621us		165ms	
		-----Sequential Create-----						-----Random Create-----					
		-Create--		--Read---		-Delete--		-Create--		--Read---		-Delete--	
files:max:min		/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP
debi 200:102400:100		208	16	52	1	294	11	203	15	49	1	256	10



Latency	1347ms	2603ms	3796ms	1661ms	950ms	4664ms
---------	--------	--------	--------	--------	-------	--------

#### 1.4.8 FIO over NFSv4 share

O *fio* é um pacote Debian que simula operações no sistema de arquivos e acordo com os parâmetros descritos no arquivo de configuração.

Para simular o comportamento do Cyrus IMAP, o *fio* será configurado para realizadas diversas escritas aleatórias síncronas paralelas de arquivos pequenos no sistema de arquivos. Além disso, o arquivo de parâmetros descritos abaixo deve ser configurado para exaurir a memória RAM da máquina geradora de carga e do data storage server, alterando o parâmetro *size*.

Os testes a serem realizados serão executados durante 1 hora, no primeiro momento, e depois durante 8 horas, para verificar o comportamento dos equipamentos envolvidos. Nesse caso, os *tunings* descritos anteriormente devem estar em funcionamento no ambiente.

##### 1.4.8.1 Multi threaded.

We will use example for NFSv4 tests. You may change to other filesystem types mountpoints.

Para dispensar Kerberos, os usuarios e grupos DEVERÃO estar sincronizados:

##### 1.4.8.1.1 etc passwd

```
cyrus:x:117:8:Cyrus Mailsystem User,,,:/var/spool/cyrus:/bin/sh
```

Você pode alterar propriedade dos arquivos com o comando abaixo. Adapte para o SEU caso, e no cliente e ou no servidor:

```
find / -type d -user 107 -exec chown -R 117 {} \;
```

##### 1.4.8.1.2 modificações /etc/hosts

```
127.0.0.1 localhost
```

##### 1.4.8.1.3 /etc/hostname

```
localhost
```

executar para verificar:

```
hostname --fqdn
```

#### 1.4.8.1.4 server /etc/default/nfs-common

add the following

```
#AFM 20130326  
NEED_IDMAPD=YES
```

#### 1.4.8.1.5 server etc hosts.allow

```
#AFM 20131223  
portmap: 192.168.
```

#### 1.4.8.1.6 server /etc/idmapd.conf

```
[General]  
  
Verbosity = 0  
Pipefs-Directory = /var/lib/nfs/rpc_pipefs  
# set your own domain here, if id differs from FQDN minus hostname  
# AFM 20131226 fqdn  
Domain = localhost  
  
[Mapping]  
  
Nobody-User = nobody  
Nobody-Group = nogroup
```

#### 1.4.8.1.7 server /etc/default/rpcbind

May work without this at some scenarios.

```
#AFM 20131223 for nfsv4
#OPTIONS= " "
```

#### 1.4.8.1.8 server /etc/exports

Add the following

```
#AFM20130326
/exports
192.168.0.0/255.255.255.0(sync,rw,no_root_squash,no_subtree_check,crossmnt,fsid=0)
```

ou

```
#AFM 20131226
/exports/
192.168.56.0/255.255.255.0(sync,rw,no_root_squash,no_subtree_check,crossmnt,fsid=0)
/exports/data01
192.168.56.0/255.255.255.0(sync,rw,no_root_squash,no_subtree_check,nohide)
/exports/indicesimap01
192.168.56.0/255.255.255.0(sync,rw,no_root_squash,no_subtree_check,nohide)
```

```
invoke-rc.d rpcbind restart
invoke-rc.d nfs-common restart
invoke-rc.d nfs-kernel-server restart
exportfs -r
```

#### 1.4.8.1.9 cliente modificações /etc/hosts

```
127.0.0.1    localhost
```

#### 1.4.8.1.10 cliente /etc/hostname

```
localhost
```

executar para verificar:

```
hostname --fqdn
```

#### 1.4.8.1.11 client /etc/default/nfs-common

```
#AFM 20131226
NEED_IDMAPD=YES
```

#### 1.4.8.1.12 client etc hosts.allow

```
#AFM 20131223
portmap: 192.168.
```

#### 1.4.8.1.13 client /etc/idmapd.conf

```
# AFM 20131226 fqdn
Domain = localhost
```

```
mkdir -p /mnt/nfsimports
mount -t nfs4 -o timeo=900,retrans=3,intr,proto=tcp,lock,vers=4,actimeo=10 \
192.168.0.110:/ /mnt/nfsimports

mkdir -p /data
mkdir -p /indicesimap
mount -t nfs4 -o timeo=900,retrans=3,intr,proto=tcp,lock,vers=4,actimeo=10 \
192.168.56.108:/indicesimap01 /indicesimap
mount -t nfs4 -o timeo=900,retrans=3,intr,proto=tcp,lock,vers=4,actimeo=10 \
192.168.56.108:/data01 /data
df -h
```

#### 1.4.8.1.14 client /etc/fstab/

```
192.168.56.108:/data01 /data      nfs4
_netdev,auto,timeo=900,retrans=3,intr,proto=tcp,lock,vers=4,actimeo=10  0      0
192.168.56.108:/indicesimap01 /indicesimap      nfs4
_netdev,auto,timeo=900,retrans=3,intr,proto=tcp,lock,vers=4,actimeo=10  0      0
```

***Only for tests, without security !!!!***

```
chmod -R 777 /mnt/nfsimports
```

#### **1.4.8.1.15 fio tests /home/andremachado/fio\_randomrw\_1kthreads\_nfsv4.txt**

Ajustar size para esgotar RAM do gerador de carga e do data storage server, senão medirá desempenho apenas de caches. Ajustar numjobs para pelo menos o número de núcleos do gerador de carga. Ajustar o tempo para 1 hora no mínimo. Durante homologação ajustar para 8 horas.

O gerador de carga deverá ter filesystem e kernel tunados para mínima latência conforme documentação.

```
; random read write of data

[random-rw]
iodepth=64
rw=randrw
#rw=randwrites
rwmixwrite=90 #90% writes
invalidate=1
#buffered=1
direct=1 #zfsfuse segfault
fsync_on_close=1
size=4G #test dataset size ceiling when filesize range OR sum of
        #file jobs when filesize undefined
directory=/mnt/nfsimports
numjobs=50
filesize=1k,75k
runtime=1800
time_based
verify=md5
verify_fatal=1
verify_async=10
#write_lat_log

lockfile=readwrite
fallocate=0 #cyrus, nfs use this?
```

```
fadvise_hint=0 #cyrus, nfs use this?
##scramble_buffers=false #anti dedupe. not implemented in debian squeeze
#ioengine=mmap #cyrus imap uses writev, write, fdatsync, mmap/memcpy, fclose
ioengine=sync
#ioengine=vsync #takes hours for finish, despite a 3 min time limit
fdatsync=1 #cyrus number of writes before force fdatsync syscall
#iomem=mmap #cyrus reads from mmap files
do_verify=1
nrfiles=1
openfiles=1
file_service_type=random
nice=19
prio=7 #highest 0, lowest 7
prioclass=2 #best effort priority
unlink=1 #unlink files at end of job
```

#### 1.4.8.1.16 run the test

Para executar o pacote, pode ser executado o seguinte comando:

```
$ nice fio -output=/tmp/fio_output_nfs.log \
/root/fio_randomrw_1kthreads_nfsv4.txt
```

#### 1.4.9 Imaptest

O *imaptest* é uma ferramenta que emula o comportamento de vários usuários utilizando o IMAP concorrentemente. O ImapTest pode ser baixado no endereço <http://www.imapwiki.org/ImapTest> e compilado conforme documentação.

Para sua execução, é necessário possuir um serviço IMAP já configurado e funcional. Vale ressaltar que, para os casos de testes utilizando ImapTest, serão definidas LUNs de médio desempenho de 2TB para as mensagens dos usuários e LUNs de 300 GB de alto desempenho para os índices das caixas postais.

#### 1.4.10 Obter código fonte e compilar pacote do Imaptest

Single threaded mas provoca carga multithreaded sobre o servidor imap.

O Imap Test precisa ser baixado do site

<http://www.imapwiki.org/ImapTest>

e compilado com as bibliotecas iguais ou mais novas Dovecot 2.2.x. Ocorre que isto está no

Debian Instável e precisa ser feito backport. Você também pode baixar o pacote Dovecot do Debian Backports. Ou você instala toda a máquina Debian Instável.

<http://packages.debian.org/source/sid/dovecot>

Optamos por fazer backport para o Debian Wheezy e compilar o imapttest.

Geramos pacote grosseiro do imapttest usando o programa checkinstall.

```
sudo apt-get install checkinstall
```

Para o cyradm guardar histórico de comandos e aceitar backspace e setas, precisa instalar mais um pacote:

```
sudo apt-get install libterm-readline-gnu-perl
```

#### 1.4.10.1 preparação para teste, instalação do cyrus imap

Faremos várias modificações nas configurações do cyrus imap visando aproximar às configurações usadas em produção, sob os aspectos que envolvem o acesso a discos.

Serão necessárias várias LUNS configuradas para diferentes desempenhos. Uma de 2TB de armazenamento principal de dados em médio desempenho, que será montada em /data, e outra de 300 GB (ou 15% do /data) configurada para o mais alto desempenho, que será montada em /indicesimap.

A LUN para montagem da máquina virtual poderá ser configurada em médio desempenho.

```
sudo apt-get update

sudo apt-get install db4.7-util gawk libhesiod0 db5.1-util libdb5.1 \
  libpcre3 libperl5.14 libsasl2-2 libsasl2-modules libsasl2-modules-ldap \
  libsensors4 libsnmp15 libsnmp-base libsysfs2 libzephyr4 perl perl-base \
  perl-modules sasl2-bin ucf libwrap0 libpam0g libssl1.0.0 \
  libcomerr2 ldap-utils libdb4.8 db4.8-util gawk libauthen-sasl-perl \
  postfix postfix-ldap postfix-pcre openssl-blacklist openssl-blacklist \
  openssl-server openssl-client libsasl2-modules-gssapi-heimdal \
  libterm-readline-gnu-perl sysfsutils util-linux libauthen-sasl-cyrus-perl
```



```
sudo apt-get install cyrus-admin-2.4 cyrus-clients-2.4 cyrus-common \
cyrus-common-2.4 cyrus-imapd-2.4 cyrus-imapd libcyrus-imap-perl24

sudo invoke-rc.d cyrus-imapd stop

cp /etc/cyrus.conf /etc/cyrus.conf.original
cp /etc/imapd.conf /etc/imapd.conf.original

nano /etc/imapd.conf
```

#### 1.4.10.1.1 modificações no /etc/imapd.conf para o teste de carga

```
# Configuration directory
#AFM 20110906
configdirectory: /indicesimap/var/lib/cyrus

#AFM 20110906 metapartition names are defined following these definitions below
partition-default: /data/var/spool/cyrus/mail

#AFM 20ago2010 se usar diferentes particoes e
# Para permitir a movimentacao entre backends
allowusermoves: yes

#AFM 20ago2010
# Colocado para compatibilizacao com Clientes para subscrever em caixas
# pertencentes a diferentes backends
allowallsubscribe: 1

#AFM 20ago2010
# Eliminar mensagens duplicadas
duplicatesuppression: 1

#AFM 20ago2010
# Habilitar que as mensagens nao sejam deletadas
# imediatamente e possam ser recuperadas.
expunge_mode: delayed
```

```
# News setup
#AFM 20110906
partition-news: /data/var/spool/cyrus/news
newsspool: /data/var/spool/news

#sasl_pwcheck_method: saslauthd auxprop
#AFM 02set2010 tentando evitar pedir senha qdo especifica servidor ao creatembox
sasl_pwcheck_method: alwaystrue

#AFM 25ago2010
skiplist_always_checkpoint: 1

#AFM 25ago2010
singleinstancestore: 1

#AFM 20110908
sievedir: /data/var/spool/sieve

#AFM 20110908 debian cyrus 2.2 defaults proved to have best performance
# cat /usr/lib/cyrus/cyrus-db-types.active
# debian cyrus 2.4 is at
# cat /usr/lib/cyrus/cyrus-db-types.txt

#annotation_db: skiplist
#duplicate_db: berkeley-nosync
#mboxkey_db: skiplist
#mboxlist_db: skiplist
#ptscache_db: berkeley
#quota_db: quotalegacy
#seenstate_db: skiplist
#statuscache_db: berkeley-nosync
#subscription_db: skiplist
#tlscache_db: berkeley-nosync
#userdeny_db: skiplist

#berkeley_locks_max: 50000
#berkeley_txns_max: 800

#AFM 20131226
```

```
metapartition_files: header index cache expunge squat
metapartition-default: /indicesimap/var/cyrus/metapartition/default
statuscache: 1
temp_path: /indicesimap/var/cyrus/tmp
deletedprefix: DELETED
delete_mode: delayed
expunge_days: 1
foolstupidclients: 0
#imapidlepoll: 60
#imapmagicplus: 0
improved_mboxlist_sort: 1
#internaldate_heuristic: standard
#suppress_capabilities: <none>

#AFM 20131220
unixhierarchysep: yes

#AFM 20131211
admins: cyrus

#AFM 20131220
proc_path: /indicesimap/var/run/cyrus/proc
mboxname_lockpath: /indicesimap/var/run/cyrus/lock

##
## KEEP THESE IN SYNC WITH cyrus.conf
##
# Unix domain socket that lmtpd listens on.
#AFM 20131220
lmtpsocket: /indicesimap/var/run/cyrus/socket/lmtp

# Unix domain socket that idled listens on.
#AFM 20131220
idlesocket: /indicesimap/var/run/cyrus/socket/idle

# Unix domain socket that the new mail notification daemon listens on.
#AFM 20131220
notifysocket: /indicesimap/var/run/cyrus/socket/notify
```

#### 1.4.10.1.2 modificações no /etc/cyrus.conf

nano /etc/cyrus.conf

```
#AFM 20131220
    lmtplib          cmd="lmtplib"
listen="/indicesimap/var/run/cyrus/socket/lmtplib" prefork=0 maxchild=20

#AFM 10131220
    notify          cmd="notifyd"
listen="/indicesimap/var/run/cyrus/socket/notify" proto="udp" prefork=1
```

#### 1.4.10.1.3 modificações no /etc/default/saslauthd

nano /etc/default/saslauthd

```
#MECHANISMS="pam"
#AFM 07out2010
MECHANISMS="sasldb"

#AFM 20131211 verbose
OPTIONS="-c -m /var/run/saslauthd -V"
```

#### 1.4.10.1.4 modificações no /etc/default/cyrus-imapd

nano /etc/default/cyrus-imapd

```
#AFM 20131220
CYRUS_VERBOSE=1
```

#### 1.4.10.1.5 exibir arquivos sem comentários nem linhas em branco

```
egrep -v "^[[:space:]]*#|^$|^#|^[[:space:]]*$" /etc/cyrus.conf
egrep -v "^[[:space:]]*#|^$|^#|^[[:space:]]*$" /etc/imapd.conf
```

```
mkdir -p /run/cyrus/lock
```

```
mkdir -p /indicesimap/var/lib/cyrus
mkdir -p /data/var/spool/cyrus/mail
mkdir -p /data/var/spool/cyrus/news
mkdir -p /data/var/spool/sieve
mkdir -p /data/var/spool/news
mkdir -p /indicesimap/var/cyrus/metapartition/default
mkdir -p /indicesimap/var/cyrus/tmp
chown cyrus.mail /run/cyrus/lock
mkdir -p /indicesimap/var/run/cyrus/socket/
chown -R cyrus.mail /data/var/spool/sieve
chown -R cyrus.mail /data/var/spool/news
chown -R cyrus.mail /data/var/spool/cyrus
mkdir -p /indicesimap/var/run/cyrus/socket/lmtp
mkdir -p /indicesimap/var/run/cyrus/socket/idle
mkdir -p /indicesimap/var/run/cyrus/socket/notify
chown -R cyrus.mail /indicesimap/var/run/cyrus/socket

mkdir -p /indicesimap/var/run/cyrus/proc
mkdir -p /indicesimap/var/run/cyrus/lock
chown -R cyrus.mail /indicesimap/var/run/cyrus/proc
chown -R cyrus.mail /indicesimap/var/run/cyrus/lock

touch /indicesimap/var/lib/cyrus/tls_sessions.txt
cvt_cyrusdb /indicesimap/var/lib/cyrus/tls_sessions.txt flat
/indicesimap/var/lib/cyrus/tls_sessions.db berkeley-nosync
touch /indicesimap/var/lib/cyrus/user_deny.txt
cvt_cyrusdb /indicesimap/var/lib/cyrus/user_deny.txt flat
/indicesimap/var/lib/cyrus/user_deny.db skiplist

chown -R cyrus:mail /indicesimap/var/lib/cyrus
chown -R cyrus:mail /indicesimap/var/run/cyrus
chown -R cyrus:mail /indicesimap/var/cyrus
chmod o-rwx /indicesimap/var/run/cyrus/socket/
cyrus-makedirs
usermod -a -G ssl-cert cyrus
sudo -u cyrus ctl_cyrusdb -r

mkdir -p /indicesimap/var/spool/postfix

cyrus-makedirs
```

```
sudo invoke-rc.d cyrus-imapd restart
```

Examine os arquivos de logs em busca de erros ou anomalias.

#### 1.4.10.1.6 retorno a defaults dos databases do cyrus

Podem ocorrer dificuldades na alteração dos tipos de databases internos do cyrus. Então você pode restabelecer os formatos default comentando fora as alterações dos tipos e executando os comandos numa etapa só:

```
invoke-rc.d cyrus-imapd stop
mv /indicesimap/var/lib/cyrus/statuscache.* /tmp/
mv /indicesimap/var/lib/cyrus/user_deny.db* /tmp/
mv /indicesimap/var/lib/cyrus/tls_sessions.db* /tmp/
mv /indicesimap/var/lib/cyrus/deliver* /tmp/
mv /indicesimap/var/lib/cyrus/annot* /tmp/

cvt_cyrusdb /indicesimap/var/lib/cyrus/tls_sessions.txt flat
/indicesimap/var/lib/cyrus/tls_sessions.db skiplist
cp /indicesimap/var/lib/cyrus/user_deny.txt
/indicesimap/var/lib/cyrus/user_deny.db
chown cyrus.mail /indicesimap/var/lib/cyrus/user_deny.db

invoke-rc.d cyrus-imapd start
```

#### 1.4.10.2 preparação para teste, backport dos pacotes de Dovecot

```
sudo apt-get install devscripts

sudo apt-get install pkg-config libssl-dev libpam0g-dev libldap2-dev libpq-dev \
libmysqlclient-dev libsqlite3-dev libsasl2-dev zlib1g-dev libkrb5-dev drac-dev \
libbz2-dev libdb-dev libcurl4-gnutls-dev libexpat-dev libwrap0-dev dh-systemd
checkinstall

mkdir -p ~/projetos/dovecot/
cd ~/projetos/dovecot/
```

Baixar os arquivos-fonte do pacote dovecot para criar backport.

<http://packages.debian.org/source/sid/dovecot>

```
dpkg-source -x dovecot_2.2.9-1.dsc
```

```
cd dovecot-2.2.9/  
nice debuild -uc -us  
cd ~/projetos/dovecot/  
  
sudo dpkg -i dovecot-dev_2.2.9-1_amd64.deb dovecot-core_2.2.9-1_amd64.deb  
  
mkdir -p ~/projetos/imaptest  
wget -c http://dovecot.org/nightly/imaptest/imaptest-latest.tar.gz  
cd ~/projetos/imaptest  
tar -xzvf imaptest-latest.tar.gz  
cd ~/projetos/imaptest/imaptest-20131206/  
  
#### ./configure --with-dovecot=/usr/include/dovecot  
  
./configure --with-dovecot=/usr/lib/dovecot  
nice make  
checkinstall
```

O pacote checkinstall criará um pacote debian do imaptest de forma bruta, sem verificações minuciosas. Ainda assim será útil para eventual remoção de programas e upgrades.

#### 1.4.10.2.1 /home/andremachado/usersimaptest.txt

Formato usuario:senha\_texto\_claro

Os usuários e senhas criados previamente no servidor imap.

Adotamos o padrão hardcoded do imaptest para nome de usuário numerados de 1 a 100, viabilizando testes diferentes se necessário. Os usuários no exemplo serão criados em base local do sasldb. Para ldap, siga procedimentos apropriados.

```
user1:123  
user2:123  
user3:123  
user4:123  
user5:123  
user6:123  
user7:123  
user8:123
```



```
user9:123
user10:123
```

```
sudo saslpasswd2 -c user2

sudo sasldblistusers2

andremachado@debiantesting64:~$ sudo saslpasswd2 -c user3
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user4
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user5
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user6
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user7
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user8
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user9
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo saslpasswd2 -c user10
Password:
Again (for verification):
andremachado@debiantesting64:~$ sudo sasldblistusers2
andremachado@debiantesting64: userPassword
andremachado@localhost: userPassword
mupdateuser@debiantesting64: userPassword
user2@debiantesting64: userPassword
user4@debiantesting64: userPassword
user6@debiantesting64: userPassword
user8@debiantesting64: userPassword
cyrus@debiantesting64: userPassword
```

```
cyrus@localhost: userPassword
user1@debiantesting64: userPassword
user10@debiantesting64: userPassword
user3@debiantesting64: userPassword
user5@debiantesting64: userPassword
user7@debiantesting64: userPassword
user9@debiantesting64: userPassword
andremachado@debiantesting64:~$

andremachado@debiantesting64:~$ cyradm --user cyrus localhost
Password:
localhost>
localhost> cm user.user2
localhost> sam user.user2 user2 all
localhost> cm user.user3
localhost> sam user.user3 user3 all
localhost> cm user.user4
localhost> sam user.user4 user4 all
localhost> cm user.user5
localhost> sam user.user5 user5 all
localhost> cm user.user6
localhost> sam user.user6 user6 all
localhost> cm user.user7
localhost> sam user.user7 user7 all
localhost> cm user.user8
localhost> sam user.user8 user8 all
localhost> cm user.user9
localhost> sam user.user9 user9 all
localhost> cm user.user10
localhost> sam user.user10 user10 all
localhost> quit
localhost> exit
andremachado@debiantesting64:~$
```

É possível criar um arquivo texto puro com os dados das contas e importar para o cyrus imap server. Mas o formato do arquivo texto é crítico e contém caracteres não imprimíveis que precisam estar todos corretos. Experimente criar para os 10 primeiros manualmente, exportar, editar um novo ampliando os usuários e então importar. Ainda restarão os usuários de sasldb.

#### 1.4.10.3 execução teste com dataset menor da Dovecot

O dataset menor de 10MB da Dovecot é bom para testes rápidos.

Foi obtido dos arquivos públicos de lista de discussão e já está devidamente limpo de caracteres e no formato que as bibliotecas dovecot usadas pelo imaptest compreendem.

```
sudo ls -lh /var/spool/cyrus/mail/u/user/user6
cd ~/Downloads
wget -c http://www.dovecot.org/tmp/dovecot-crlf

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/dovecot ; \
imaptest host=localhost port=143 mbox=/home/andremachado/Downloads/dovecot-
crlf.mbox \
secs=60 seed=1234 clients=4 disconnect_quit
userfile=/home/andremachado/usersimaptest.txt msgs=100
```

#### 1.4.10.4 preparação para teste, dataset maior da Dovecot

O dataset de 268 MB da Dovecot foi obtido dos arquivos públicos de lista de discussão e já está devidamente limpo de caracteres e no formato que as bibliotecas dovecot usadas pelo imaptest compreendem.

```
cd ~/Downloads
wget -c http://dovecot.org/archives/dovecot.mbox

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/dovecot ; \
imaptest host=localhost port=143 mbox=/home/andremachado/Downloads/dovecot.mbox \
secs=60 seed=1234 clients=4 disconnect_quit \
userfile=/home/andremachado/usersimaptest.txt msgs=100
```

#### 1.4.10.5 preparação para teste, dataset da Enron

O dataset de 1.4GB da Enron é disponibilizado em formato Maildir e o imaptest precisa de formato mbox. São dados de contas reais de uma empresa falida, divulgados pelo governo dos EUA. Um dos locais para obtê-lo é <https://www.cs.cmu.edu/~enron/>

O dataset da Enron contém muitos anexos grandes, deixando o teste bem mais lento que o dataset da Dovecot. Servidores de listas geralmente descartam anexos. Mas é um teste interessante para reproduzir melhor um ambiente de usuários reais.

Instale o pacote procmail que contém o utilitário formail.

```
sudo apt-get update
sudo apt-get install procmail
```

```
cd /home/andremachado/temp/enron_mail_20110402/maildir

for file in `find . -type f` ; do formail -a Date: <"$file" >> ../enron.mbox ;
done
```

O dataset da Enron tem caracteres de controle windows nos headers e precisa ser limpo. Também iremos consolidar as várias caixas postais num único mailbox para este teste. Limparemos alguns erros de edição escapado no cabeçalho "From " do formato mailbox. E recriar os message-id para evitar erros na caixa consolidada.

```
cd /home/andremachado/temp/enron_mail_20110402/maildir
#rm /home/andremachado/temp/enron_mail_20110402/enron_unix.mbox
#rm /home/andremachado/temp/enron_mail_20110402/enron.mbox
mv ../*.mbox /tmp/

for file in `find . -type f` ; do formail -a 'From ' <"$file" >>
../enron.mbox ; done

tr -cd '[:print:]\n' \
< /home/andremachado/temp/enron_mail_20110402/enron.mbox \
> /home/andremachado/temp/enron_mail_20110402/enron_unix.mbox

sed -i 's/^>From /> From /g'
/home/andremachado/temp/enron_mail_20110402/enron_unix.mbox

nice formail -i Message-ID: -a Message-ID: -b -ds \
</home/andremachado/temp/enron_mail_20110402/enron_unix.mbox \
>>/home/andremachado/temp/enron_mail_20110402/enron_filtered.mbox

sed -i 's/^>From /From /g'
/home/andremachado/temp/enron_mail_20110402/enron_filtered.mbox

less /home/andremachado/temp/enron_mail_20110402/enron_filtered.mbox

formail -a 'From ' -bds \
</home/andremachado/temp/enron_mail_20110402/enron_filtered.mbox \
```

```
>>/home/andremachado/temp/enron_mail_20110402/enron_filtered2.mbox

less /home/andremachado/temp/enron_mail_20110402/enron_filtered2.mbox

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/dovecot ; \
imaptest host=localhost port=143
mbox=/home/andremachado/temp/enron_mail_20110402/enron_filtered2.mbox \
secs=60 seed=1234 clients=4 disconnect_quit \
userfile=/home/andremachado/usersimaptest.txt msgs=100
```

Vai gerar saída num formato similar ao abaixo, com números apropriados para servidores de alto desempenho claro:

Logi	List	Stat	Sele	Fetc	Fet2	Stor	Dele	Expu	Appe	Logo
100%	50%	50%	100%	100%	100%	50%	100%	100%	100%	100%
					30%				5%	
7	1	1	6	6	7	1	4	5	6	6 4/ 4
4	4	3	4	4	5	2	4	4	4	4 4/ 4
5	1	1	6	6	7	2	4	4	5	5 4/ 4
5	2	2	5	5	7	0	3	7	5	5 4/ 4
6	4	3	6	6	9	1	7	7	6	6 4/ 4
5	1	2	5	5	7	0	4	5	5	5 4/ 4
5	1	4	5	5	8	0	3	4	5	5 4/ 4
4	3	3	4	4	7	1	5	4	4	4 4/ 4
4	1	1	4	4	5	1	4	4	4	4 4/ 4
11	36	245	259	7	66	45	115	223	565	0 ms/cmd avg
Totals:										
Logi	List	Stat	Sele	Fetc	Fet2	Stor	Dele	Expu	Appe	Logo
100%	50%	50%	100%	100%	100%	50%	100%	100%	100%	100%
					30%				5%	
2904	1470	1398	2904	2863	4126	468	2297	2863	2903	2905

Procure por indicações de erro na saída e nos logs.

#### 1.4.10.6 Monitoração de desempenho

Acompanhando a própria saída do imaptest é um ótimo indicador, mas você pode acompanhar também o desempenho do NFSv4.

Poucas ferramentas estão adequadas para o protocolo v4. Nfswatch, nmon nesta data ainda não suportam bem NFSv4.

Usaremos ferramentas do pacote sysstat e collectl (opcionalmente o collectl-utils).

```
sudo apt-get update
sudo apt-get install sysstat collectl collectl-utils nfsiostat
```

No cliente Linux:

```
nfsiostat 1
watch -n 1 nfsstat -c -o all -4
collectl -sjmf -oT
```

No servidor linux:

```
watch -n 1 nfsstat -o all -4
```

## 1.5 Procedimentos de testes não funcionais

Entre os requisitos não funcionais, serão verificadas a eficácia e eficiência de recursos de disponibilidade, recuperação de desastres e a compatibilidade com ambiente e aplicações atualmente utilizadas na produção do Expresso no SERPRO. Além disso, serão avaliados o impacto no desempenho da utilização de tais funcionalidades.

### 1.5.1 Failover automático e failback de unidades HEAD gerenciadoras.

Será necessário simular um *freeze* ou *kernel panic* e também um extreme load nas unidades gerenciadoras dos ZFS data storage servers.

É preciso encontrar os comandos equivalentes aos usados para realizar essas operações no Debian GNU/Linux, onde é usado o seguinte comando:

```
$ echo c > /proc/sysrq-trigger
```

Ou como no FreeBSD:

```
sysctl debug.kdb.panic=1
```

#### 1.5.1.1 Fork bomb:

Nos kernels Illumos/Solaris, uma possibilidade é causar um fork bomb para causar extrema carga através de comandos de bash shell:

```
:() { :|:& } ;:
```

mas se o sistema tiver conservadores limites de segurança para processos e memória ou de deadman timer (watchdog) pode não ser suficiente. Ao menos deverá deixar os serviços muito lentos durante um período ou entrar em swap.

Um fork bomb pode ser bem interessante para simular efeitos de uma sobrecarga massiva lenta o suficiente para poder confundir o estado das conexões e do failover.

#### 1.5.1.2 Usar Dtrace:

Uma possibilidade simples é usar Dtrace para gerar coredump e rebotar, loggado como **REAL** root.

```
dtrace -wn 'BEGIN { panic(); }'
hald -d
reboot -d
uadmin 5 1
```

Alguns sistemas podem iniciar o procedimento de savecore dump imediatamente no primeiro comando e em seguida reboot.

Ao entrar no GRUB, pode ser necessário intervenção manual para pausar o reboot e aguardar o failover.

#### 1.5.1.3 Causar Non Maskable Interrupt NMI

Outra possibilidade é carregar o kmdb no boot através de opção de boot -k da linha de initrd no GRUB, usando os recursos de edição e boot.

E no **/etc/system** também acrescentar:

```
set pcplusmp:apic_panic_on_nmi=1
set pcplusmp:apic_kmdb_on_nmi=1
```

Depois usar ipmitool para disparar uma NMI

```
ipmitool -I lanplus -H somebox -U root chassis power diag
```

Os serviços devem parar e no console entrará em debug.

Com a queda de serviços, a outra unidade HEAD deverá assumir o controle do ZFS Data Storage Server, concluindo o failover.

#### 1.5.1.4 Medições

Nesse quesito, serão avaliados o tempo necessário para o chaveamento entre unidades gerenciadoras, bem como impacto para os clientes conectados. Além disso, serão verificados a integridade dos sistemas de arquivos e LUNs afetadas.

### **1.5.2 Tolerância e recuperação de falhas individuais e múltiplas de dispositivos de blocos.**

Será necessário verificar a capacidade do equipamento testado de continuar a operação em caso de falhas nos dispositivos de blocos, inutilizando-os para o uso. Nesse caso, seria necessário simular uma perda de conexão com um dispositivo aleatório.

Deverão ser observados as atuações de hot spares e o processo de substituição de unidades por novas.

Nesse caso, serão avaliados o impacto no desempenho do sistema, a variação na latência das operações, entre outros detalhes.

Verificar falha de um ou mais dispositivos do servidor (disco de ZIL, disco de L2ARC e discos de dados), tempo de recuperação, eventuais perdas de performance em SCRUB, possibilidade de troca a quente dos discos. Sugere-se, inclusive, a remoção de um dispositivo a quente no ambiente para verificar alguma falha de escrita e problema de recuperação do ambiente ao perder um dispositivo sem qualquer evento anterior.

#### **1.5.2.1 Tolerância e recuperação de falhas transientes de rede.**

Será necessário verificar a capacidade do equipamento de suportar falhas de comunicação na rede Ethernet e rede SAN com simulação de falha de fibra e porta FC, no lado do Host e do ZFS Data Storage Servers (back-end e front-end). Nesse caso, serão inseridos *delays* e *perdas de pacotes* aleatórios ao sistema, para verificar o comportamento do equipamento.

Nesse caso, serão avaliados o impacto no desempenho do sistema, a variação na latência das operações, entre outros detalhes.

#### **1.5.2.2 Tolerância e recuperação de falhas transientes na alimentação.**

Será necessário verificar a capacidade do equipamento de suportar falhas na alimentação (ex: desligamento de um head node do conjunto ZFS) de rede elétrica (simulando pane inteira do equipamento). Nesse caso, serão desligados, em momentos distintos e em conjunto, cada um dos sistemas alimentadores do equipamento.

Nesse caso, serão avaliados tempo de recuperação do dispositivo 'secundário' de forma a voltar a responder a requisições de I/O dos clientes, o impacto no desempenho do sistema, a variação na latência das operações, entre outros detalhes.

#### **1.5.2.3 Operações de snapshots, remoção de snapshots, replicação, backup e seus impactos no desempenho.**

Será necessário avaliar o impacto das operações sobre snapshots e outras operações (verificar a operação de execução automática) no desempenho do sistema. Dessa forma, serão executadas cada uma das operações sobre snapshots (criação, remoção) e a utilização de replicação síncrona e assíncrona e backup, de forma a avaliar o impacto no desempenho, a variação na latência nas operações, entre outros detalhes.

#### **1.5.2.4 Migrações de dados entre servidores de armazenamento.**

Será necessário mostrar a migração de dados de LUNs entre 100GB e 1TB, simulando as operações a serem realizadas em equipamentos no fim de vida útil e o comissionamento de novos equipamentos.

Efetuar a migração para dados em LUN e em compartilhamento NFSv4.

Documentar os efeitos nos equipamentos ZFS Data Storage Server e clientes. Medir



variações de latência, IOPS e velocidade.

#### **1.5.2.5 Tempos para recuperação de desastres (replicação e ciclo backup com restore).**

Simular o retorno a operação de um equipamento principal de alto desempenho (LUN e NFSv4) a partir de 100 GB e 1 TB de dados de e-mail armazenados em um outro equipamento replicado, de desempenho igual ou inferior, e de um outro equipamento onde são armazenados dados de backup, unidade de fita ou outro equipamento ZFS Data Storage Server de desempenho adequado a backup.

Documentar o tempo até disponibilidade do serviço normalizado e o perfil dos equipamentos envolvidos na operação.

#### **1.5.2.6 Compatibilidade padrão open source (livremente incorporada no kernel oficial) com Debian GNU/Linux 7.x e superiores.**

Todo software utilizado nos clientes para acessarem recursos do ZFS Data Storage Server, como o sistema de arquivos ZFS na versão utilizada para os testes ou a interface de gerenciamento, deverá ter demonstrada a disponibilidade pública de código fonte em licença aberta e livre disponível oficialmente nos repositórios Debian ou algum outro, que garanta as 4 liberdades do código fonte.

[http://pt.wikipedia.org/wiki/Licen%C3%A7a\\_de\\_software\\_livre](http://pt.wikipedia.org/wiki/Licen%C3%A7a_de_software_livre)

<http://opensource.org/licenses/category>

Falhas por incompatibilidades serão relatadas como bugs críticos desqualificadores.

#### **1.5.2.7 Formato de sistema de arquivos ZFS em licença livre.**

Deverá ser demonstrada a disponibilidade pública de código fonte em licença aberta e livre disponível oficialmente ligada a partir do site <http://www.open-zfs.org> do sistema de arquivos ZFS na versão utilizada para os testes de recursos.

Falhas de incompatibilidades serão relatadas como bugs críticos desqualificadores.

#### **1.5.2.8 Compatibilidade com virtualizadores XenServer e VmWare.**

Deverá ser demonstrada a disponibilidade pública de documentação listando como software homologado para operação nas versões requisitadas de XenServer e VmWare.

Erros durante os testes decorrentes de incompatibilidades serão relatadas como bugs críticos desqualificadores.

#### **1.5.2.9 Desempenho sustentado sob carga contínua.**

Deverão ser apresentados para inclusão no relatório final dos testes os gráficos, logs e registros de saídas de comandos do ZFS Data Storage Server e dos clientes que demonstrem a sustentação do desempenho sob carga contínua e homogênea e sob uma rede SAN como a utilizada na empresa.

### **1.5.3 Requisitos não funcionais desejáveis:**

#### **1.5.3.1 Replicação rede local LAN e remota WAN.**

Implantar uma replicação síncrona e uma assíncrona numa rede LAN e/ou SAN e numa que simule as características de uma WAN na falta de uma disponível real para testes. Especialmente a latência, largura de banda, taxa de erros e retransmissões.

Documentar os efeitos nos equipamentos ZFS Data Storage Server e clientes. Medir variações de latência, IOPS e velocidade do Data Storage Server.

#### **1.5.3.2 Expansão de capacidade de armazenamento.**

Inserir novos dispositivos de blocos físicos no ZFS Data Storage Server para expansão de capacidade de armazenamento e os incluir funcionalmente e de forma não disruptiva.

Documentar os efeitos nos equipamentos ZFS Data Storage Server e clientes. Medir variações de latência, IOPS e velocidade.

#### **1.5.3.3 Flexibilidade na reconfiguração de recursos de LUNs e compartilhamentos de sistemas de arquivos.**

Criar, remover, redimensionar e mover LUNs e compartilhamentos NFSv4.

Documentar os efeitos nos equipamentos ZFS Data Storage Server e clientes. Medir variações de latência, IOPS e velocidade.

#### **1.5.3.4 Interface aggregation.**

Especialmente caso a interface de rede atinja saturação antes de atingir o limite do equipamento, agregar interfaces Ethernet no ZFS Data Storage Server.

Tanto quanto possível, fazer bonding ou link aggregation nos clientes Debian, XenServer, VmWare.

Documentar os efeitos nos equipamentos ZFS Data Storage Server e clientes. Medir variações de latência, IOPS e velocidade.

#### **1.5.3.5 Acesso ao sistema**

Apresentar sobre forma de perfis de segurança de acesso que garantam a operação e administração para níveis e equipes distintas (segregação de papéis), de preferência utilizando-se de autenticação LDAP.

#### **1.5.3.6 Compatibilidade com infraestrutura existente**

Verificar o grau da compatibilidade de hardware e software com a infraestrutura de armazenamento existente nos ambientes da empresa: SAN, Ethernet, conforme detalhado nos demais itens.

#### **1.5.3.7 Monitoração e relatórios**

A solução deve prover interface de gerenciamento que possibilite a monitoração e o acompanhamento dos seus componentes, bem como fornecer informações que possibilitem a análise e gerenciamento do desempenho e da capacidade, assim como a geração de alertas automáticos baseados em “thresholds” e a disponibilização de estatísticas e desejáveis relatórios de utilização de recursos.

Verificar o grau de compatibilidade com algum padrão de mercado, como: SNMP v3 (Simple Network Management Protocol) e/ou SNIA SMI-S (Storage Networking Industry Association - Storage Management Initiative – Specification).

#### **1.5.3.8 Armazenamento segregado**

Verificar se é possível configurar áreas de armazenamento de forma segregada por ambiente. Isto facilitaria, por exemplo, a definição de ambientes distintos por cliente ou por finalidade (desenvolvimento, testes, homologação e produção).

#### **1.5.3.9 Auto-call**

Verificar que a solução realize monitoração e o acionamento automático do fornecedor em caso de falha de algum de seus componentes, de forma a permitir a atuação imediata, principalmente nas falhas de hardware.

#### **1.5.3.10 Manutenção e suporte técnico**

Para a manutenção do hardware, verificar se o fornecedor possui suprimento de peças no Brasil, de preferência na região onde será produzido o serviço.

Para o suporte técnico, verificar se o fornecedor possui atendimento de primeiro nível no Brasil, com recorrência ao laboratório do fabricante.

#### **1.5.3.11 Migração de Dados para outras soluções**

Verificar como seria a migração de dados para outra solução, para o caso de haver troca de solução ao final do contrato ou por quaisquer outros motivos.

### **1.6 Relatório dos testes**

Deverão ser apresentados dados e registros das máquinas geradoras de carga que demonstrem o desempenho e sustentação de desempenho durante os testes.

Deverão ser apresentados dados (registros e ou gráficos) do ZFS Data Storage Server que demonstrem o desempenho global, a manutenção do desempenho e consumo de recursos proporcionalmente à carga e volume de dados utilizados frente ao dimensionamento nominal do equipamento.

Deverão ser apresentados dados da utilização de rede Ethernet e SAN FC que demonstrem o grau de saturação das interfaces durante os testes. O objetivo é demonstrar que os testes

não foram limitados pela capacidade de comunicação dos equipamentos.

### 1.7 Atualizações deste documento:

Este documento foi elaborado pela equipe CEAGO/COCOE/COTSC e novas atualizações devem ser consultadas no momento da utilização.

### 1.8 Bibliografia

<http://lists.andrew.cmu.edu/pipermail/info-cyrus/2013-March/036886.html> submitted Cyrus Imap filesystem stress load simulator \*\*\*\*\*

<https://confluence.terena.org/display/Storage/Measuring+storage+performance> \*\*\*\*\*

[http://www.terena.org/activities/tf-storage/ws3/IV\\_StorageBenchmarkingCookbook-StijnEeckhaut-final.pdf](http://www.terena.org/activities/tf-storage/ws3/IV_StorageBenchmarkingCookbook-StijnEeckhaut-final.pdf)

<http://www.storageperformance.org> \*\*\*\*\*

[http://www.storageperformance.org/specs/SPC-1\\_SPC-1E\\_v1.14.pdf](http://www.storageperformance.org/specs/SPC-1_SPC-1E_v1.14.pdf)

[http://www.storageperformance.org/results/benchmark\\_results\\_spc1](http://www.storageperformance.org/results/benchmark_results_spc1)

[http://info.nexenta.com/rs/nexenta/images/data\\_sheet\\_auto\\_tiered\\_storage.pdf](http://info.nexenta.com/rs/nexenta/images/data_sheet_auto_tiered_storage.pdf)

<http://recoverymonkey.org/2012/07/26/an-explanation-of-iops-and-latency/>

<https://communities.netapp.com/community/netapp-blogs/efficiency/blog/2011/02/08/flash-cache-doesnt-cache-writes--why>

<http://recoverymonkey.org/2012/06/20/netapp-posts-great-cluster-mode-spc-1-result/>

<http://recoverymonkey.org/2011/11/01/netapp-posts-world-record-spec-sfs2008-nfs-benchmark-result/> but sfs2008 is only nfsv3, and we NEED nfsv4.

[http://www.fujitsu.com/global/services/computing/storage/eternus/newsroom/disk-201004\\_faq.html](http://www.fujitsu.com/global/services/computing/storage/eternus/newsroom/disk-201004_faq.html)

<http://storagebuddhist.wordpress.com/>

[https://www.ibm.com/developerworks/community/blogs/InsideSystemStorage/entry/spc\\_benchmarks\\_for\\_disk\\_system?lang=en](https://www.ibm.com/developerworks/community/blogs/InsideSystemStorage/entry/spc_benchmarks_for_disk_system?lang=en) \*\*

<http://www.thefreelibrary.com/The+workload+driving+the+first+industry-standard+storage+benchmark.-a0110262678> \*\*\*

<http://www.openldap.org/lists/openldap-software/200605/msg00091.html> problems with txn\_checkpoint directive

[http://www.detran.pr.gov.br/arquivos/File/coordenadoria/coad/2013\\_PUB\\_N\\_11.pdf](http://www.detran.pr.gov.br/arquivos/File/coordenadoria/coad/2013_PUB_N_11.pdf)

<http://www.spec.org/sfs2008/> mas apenas nos interessam NFSv4

<http://www.spec.org/sfs2008/results/>

<http://www.phoronix-test-suite.com> tiobench e postmark

<http://archive09.linux.com/feature/138453> fraco...

[http://en.wikipedia.org/wiki/High\\_availability](http://en.wikipedia.org/wiki/High_availability)

<http://public.dhe.ibm.com/storage/software/virtualization/Taneja-IBM-SVC-4-3-Product-Profile.pdf>

<http://phoronix.com/forums/showthread.php?19066-test-suite-tests-disk>

<http://packages.debian.org/wheezy/tiobench>

<http://sourceforge.net/projects/tiobench/files/tiobench/>

<http://fsbench.filesystems.org/> \*\*\*\*\*

<http://www.bluestop.org/fio/HOWTO.txt> \*\*\*\*\*

database, but somewhat useful:

<http://www.davidklee.net/2013/07/29/benchmark-your-sql-server-instance-with-dvdstore/>

<http://linux.dell.com/dvdstore/>

<http://en.community.dell.com/techcenter/extras/w/wiki/dvd-store.aspx>

<http://jkshah.blogspot.com.br/2012/09/pgopen-2012-dvdstore-benchmark-and.html>

<http://www.slideshare.net/jkshah/sfpug-dvd-store>

[https://blogs.oracle.com/roch/entry/decoding\\_bonnie](https://blogs.oracle.com/roch/entry/decoding_bonnie) \*\*\*\*\*

[http://info.nexenta.com/rs/nexenta/images/tech\\_brief\\_nexenta\\_performance.pdf](http://info.nexenta.com/rs/nexenta/images/tech_brief_nexenta_performance.pdf) \*\*\*\*\*

<http://www.adcapnet.com/blog/cisco-nexenta-zfs-storage-appliance-configuration-and-benchmarking/> \*\*\*\*\*

<http://www.bussink.ch/?p=759> \*\*\*\*\*

<http://everythingshouldbevirtual.com/nexenta-performance-testing-no-ssdssd> \*\*\*\*\*

<http://harryd71.blogspot.com.br/2010/05/freenas-vs-nexentastor.html>

<http://nfsv4.bullopenSource.org/tools/tests/page20.php>

<http://nfsv4.bullopenSource.org/tools/tests/page21.php>

<http://forums.servethehome.com/index.php?threads/encryption-on-openindiana-and-zfs-v-28.592/>

### 1.8.1 ZFS technology presentation, vendor independent:

<http://assiste.serpro.gov.br/cisl/zfs.html>

<http://www.slideshare.net/NexentaWebinarSeries/nexenta-at-vmworld-handson-lab>

[http://www.techforce.com.br/news/linux\\_blog](http://www.techforce.com.br/news/linux_blog)

### 1.8.2 E-mail sample data for loads

<http://www.dovecot.org/tmp/dovecot-crlf> 10MB size

<http://dovecot.org/archives/dovecot.mbox> 268MB size

<https://www.cs.cmu.edu/~enron/> 423MB size

[http://en.wikipedia.org/wiki/Enron\\_Corpus](http://en.wikipedia.org/wiki/Enron_Corpus)  
<https://foundationdb.com/documentation/beta2/enron.html>  
<http://www.edrm.net/resources/data-sets/edrm-enron-email-data-set> The Enron PST Data Set Cleansed of PII by Nuix and EDRM 18GB size  
<http://info.nuix.com/Enron.html>  
<http://sociograph.blogspot.com.br/2011/04/communication-networks-part-1-enron-e.html>  
<http://sociograph.blogspot.com.br/2011/04/communication-networks-part-2-mit.html>  
<http://sociograph.blogspot.com.br/2011/05/communication-networks-part-3.html>  
<http://smashtech.net/2008/11/19/convert-maildir-to-mbox-with-1-line/>  
<https://www.benjaminwiedmann.net/convert-maildir-to-mbox-format-on-linux-bsd-or-macosx.html>  
<https://gist.github.com/nyergler/1709069>  
<http://stackoverflow.com/questions/2501182/convert-maildir-to-mbox>  
<http://www.pageofguh.org/technicality/524>  
<http://aws.amazon.com/datasets/> Apache Foundation List Archives 200GB  
<http://aws.amazon.com/datasets/7791434387204566>  
<http://aws.amazon.com/pt/publicdatasets/>  
<http://www.apache.org/foundation/maillinglists.html>  
<http://www.edrm.net/projects/dataset> Enron raw and filtered data set  
<http://www.isi.edu/~adibi/Enron/Enron.htm>  
<http://www.cs.cmu.edu/~einat/datasets.html> some newsgroup and Enron sub datasets  
[http://mail-archives.apache.org/mod\\_mbox/](http://mail-archives.apache.org/mod_mbox/)  
<http://www.mail-archive.com/>  
<http://www.mail-archive.com/faq.html#download>  
<ftp://ftp.ist.utl.pt/apache/foundation/public-archives.html>  
<http://free-downloadz.net/download/apache%20software%20foundation%20public%20mail%20archives%20download&t=1>  
<http://www.psych.ualberta.ca/~westburylab/downloads/usenetcorpus.download.html>  
USENET corpus 36GB  
<http://aws.amazon.com/datasets/1679761938200766> USENET corpus 36GB  
<http://httpd.apache.org/mail/> apache public archives  
[http://mail-archives.apache.org/mod\\_mbox/](http://mail-archives.apache.org/mod_mbox/) apache public archives

### 1.8.3 benchmarking software for imap

<http://www.imapwiki.org/ImapTest>  
<http://www.imapwiki.org/ImapTest/Examples>



<http://www.imapwiki.org/ImapTest/Running>  
<http://stackoverflow.com/questions/9094111/how-to-test-a-mail-server-using-jmeter>  
<http://sourceforge.net/p/mstone/code/HEAD/tree/trunk/mstone/>  
<http://mstone.sourceforge.net/>  
<http://archiveopteryx.org/clients/imaptest>  
<http://www.pushtotest.com/testmaker-open-source-testing>  
<http://dovecot.org/nightly/imaptest/>  
<https://packages.debian.org/wheezy-backports/dovecot-core>  
<https://packages.debian.org/wheezy-backports/dovecot-dev>

#### **1.8.4 tips for storage benchmarking**

<http://www.cognizant.com/perspectives/benchmarking-data-storage-operations>  
<http://www.cognizant.com/InsightsWhitepapers/Solving-Storage-Headaches-Assessing-and-Benchmarking-for-Best%20Practices.pdf>  
<http://searchdatacenter.techtarget.com/guides/Server-performance-and-benchmark-testing-guide>  
<http://www.it-sudparis.eu/s2ia/user/procacci/Doc/NFS/nfs014.html> \*\*\*\*\*

#### **1.8.5 NFSv4 tips**

<https://wiki.archlinux.org/index.php/NFS#Exports>  
<http://nixcraft.com/showthread.php/18145-NFS-server-mount-nfs-access-denied-by-server-while-mounting-x-y-z-w-shared-folder>  
<http://ubuntuforums.org/showthread.php?t=1635989>  
[http://wiki.linux-nfs.org/wiki/index.php/Nfsv4\\_configuration](http://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration)  
<https://lists.fedoraproject.org/pipermail/users/2012-April/417364.html>  
<http://www.math.ucla.edu/~jimc/documents/nfsv4-0601.html>  
<https://wiki.gentoo.org/wiki/NFSv4>  
[http://www.centos.org/docs/5/html/5.1/Deployment\\_Guide/s3-nfs-server-config-exportfs-nfsv4.html](http://www.centos.org/docs/5/html/5.1/Deployment_Guide/s3-nfs-server-config-exportfs-nfsv4.html)  
<http://zfsguru.com/forum/zfsgurudevelopment/516>  
<https://help.ubuntu.com/community/NFSv4Howto> \*\*  
<https://communities.netapp.com/thread/15097>  
[http://dfusion.com.au/wiki/tiki-index.php?page=Why+NFSv4+UID+mapping+breaks+with+AUTH\\_UNIX](http://dfusion.com.au/wiki/tiki-index.php?page=Why+NFSv4+UID+mapping+breaks+with+AUTH_UNIX)  
<http://askubuntu.com/questions/206843/nfs4-idmap-incorrect-user-name-mapping>  
<http://www.crazysquirrel.com/computing/debian/servers/setting-up-nfs4.aspx>  
[http://www.citi.umich.edu/projects/nfsv4/crossrealm/ASC\\_NFSv4\\_WKSHX\\_DOMAIN\\_N2I](http://www.citi.umich.edu/projects/nfsv4/crossrealm/ASC_NFSv4_WKSHX_DOMAIN_N2I)

[D.pdf](#) \*

<http://unix.stackexchange.com/questions/36812/how-can-i-do-nfsv4-uid-mapping-across-systems-with-uid-mismatches> \*\*\*

<https://bugs.launchpad.net/ubuntu/+source/nfs-utils/+bug/966734> \*

<http://www.iljacoolen.nl/2011/02/nfsv4-user-and-group-mappings/>

<https://bbs.archlinux.org/viewtopic.php?pid=895402> \*

<http://reyansh-netapp.blogspot.com.br/2013/05/nfsv4-mounts-show-file-owners-as-root.html>

<http://www.cyberciti.biz/faq/nfs4-server-debian-ubuntu-linux/>

<http://www.novell.com/support/kb/doc.php?id=7005060>

<http://manned.org/nfsidmap/076ccdbd>

<http://snipplr.com/view/14971/> Change ownership of all files owned by user1 to user2

<http://collectl.sourceforge.net/>

<http://www.vclouds.nl/building-my-nexenta-vm-using-nfs-best-practices/>

#### **1.8.6 Como causar kernel panic no Solaris, Linux, FreeBSD**

<http://www.cuddletech.com/blog/pivot/entry.php?id=1044>

<http://www.oracle.com/technetwork/server-storage/solaris10/manage-core-dump-138834.html>

<http://unix.stackexchange.com/questions/66197/how-to-cause-kernel-panic-with-a-single-command>

<http://lakeness.blogspot.com.br/2011/04/howto-generate-coredump-on-solaris.html>

<http://www.oracle.com/technetwork/server-storage/solaris10/dtrace-tutorial-142317.html>

<http://www.tablespace.net/quicksheet/dtrace-quickstart.html>

<http://blog.bignerdranch.com/1907-hooked-on-dtrace-part-1/>

<http://greg.porter.name/wordpress/?p=646>

[https://blogs.oracle.com/observatory/entry/making\\_yourself\\_indispensible\\_with\\_dtrace](https://blogs.oracle.com/observatory/entry/making_yourself_indispensible_with_dtrace)

#### **1.8.7 PostgreSQL benchmarking**

<http://www.postgresql.org/docs/9.3/static/pgbench.html>

<http://hammerora.sourceforge.net>

<http://oltpbenchmark.com>

<https://wiki.postgresql.org/wiki/Pgbench>

<http://www.tpc.org/tpcc>



### **1.8.8 Data sets**

<http://www.briandunning.com/sample-data/>

<http://docs.basho.com/riak/latest/references/appendices/Sample-Data/>

<http://www.datawrangling.com/some-datasets-available-on-the-web>

<http://www.webresourcesdepot.com/test-sample-data-generators/>

<http://www.generatedata.com/>

<https://gist.github.com/mrflip/3307566>