

Alta disponibilidade em Ambiente EAD TelEduc

CLODONIL HONÓRIO TRIGO¹

¹Centro Universitário Adventista de São Paulo - CEP 05858-001 São Paulo (SP)
clodonil@nisled.org

Resumo: *O ensino a distância ganhou força com o surgimento da Internet, os cursos por e-mail deram lugar a ferramentas especializadas, como o TelEduc. Porém, as indisponibilidades dessas ferramentas causam prejuízos irreparáveis para os cursos. Este artigo apresenta e analisa a estrutura de um cluster de alta disponibilidade para a ferramenta TelEduc.*

Palavras-Chave: *TelEduc, EAD, Alta disponibilidade, Réplica de Dados.*

1 Introdução

Educação a distância denota-se como a característica básica de estabelecimento de uma comunicação de via dupla, em que o professor e o aluno não estão no mesmo ambiente físico, necessitando, portanto, de um meio de comunicação entre ambos (NUNES, 1994). A história da educação a distância é bem antiga, remontando-se às cartas de Platão e as epístolas de São Paulo (OEIRAS *et al.*, 2001).

Na versão moderna, a Suécia registra a primeira experiência nesse campo de ensino em 1793. Em 1840, tem-se notícia do ensino a distância (EAD) na Inglaterra. Na Alemanha, foi implementado em 1856 e, nos Estados Unidos, notou-se o ensino por correspondência em 1874. No Brasil, a data mais provável do seu início foi em 1904 (ALVES, 1994).

Atualmente, a Internet vem impulsionando o ensino a distância, provocando ao longo dos últimos cinco anos, uma discussão de um novo modelo de educação, onde a sociedade está interconectada em redes de tecnologia digital (ALVES; NOVA, 2003). A Internet tem a capacidade de disseminar o conhecimento de forma globalizada, e através desse meio (Internet) o ensino a distância ganhou uma nova roupagem e novo fôlego.

Com isso, surgiram vários tipos de cursos a distância pela Internet no Brasil, desde aqueles ministrados por lista de *e-mail* até os por *software* de gerenciamento de cursos. O importante é que os clientes (alunos) foram criando um nível de exigência, tais como uma maior interação entre professor e o aluno, o que fez diminuir os cursos oferecidos sem uma estrutura básica.

Com a exigência cada vez maior dos clientes (alunos), as entidades que oferecem cursos procuram manter sempre os seus serviços ativos. Caso isso não aconteça, a entidade perde credibilidade frente aos clientes. Um *site* “fora do ar” significa um serviço falho, independente da causa.

Por esse motivo, este artigo procura apresentar como implementar um *cluster* de Alta Disponibilidade para ferramentas (aplicativos) de ensino a distância. Com o sistema de

alta disponibilidade, procura-se chegar na casa de 99,999% de disponibilidade, o que significa que durante um ano o sistema pode ficar indisponível durante 5 minutos.

Este trabalho encontra-se organizado da seguinte maneira: na seção 2 é apresentada a estrutura do ambiente virtual e as razões da escolha do TelEduc; na seção 3 são apresentados os conceitos sobre alta disponibilidade e sistemas de tolerância a falhas; na seção 4 é apresentado o procedimento para implementar o Heartbeat e os conceitos envolvidos neste processo; na seção 5 é apresentado a implementação do programa Mon; na seção 6 é apresentado a implementação e discussão sobre a réplica de dados utilizando banco de dados MySQL, DRBD e Rsync; na seção 7 são apresentadas as conclusões deste trabalho.

2 Ambiente Virtual

Existe hoje um enorme leque de ferramentas para ensino à distância, também conhecido como ambientes virtuais para o ensino, que são usados por empresas e Universidades para gerenciar os cursos. Algumas das ferramentas são de uso livre, outras não. Ambientes virtuais, de acordo com (DIAS, 2003), podem ser definidos como um conjunto de ferramentas disponibilizadas via Internet que ofereçam aos usuários as mesmas características de uma sala de aula.

No ambiente virtual, o aluno pode realizar os estudos e desenvolver tópicos de acordo com o seu tempo e interesse, e o professor (tutor), terá ferramentas para oferecer suporte e acompanhamento do estudo que está sendo realizado. O *site* da Associação Brasileira de Educação a Distância¹ contém uma lista de 100 ferramentas. Uma comparação de ambientes virtuais distribuídos como *software* livre pode ser encontrada em (PIRES, 2005).

Como mencionado anteriormente, existem inúmeras ferramentas para implementação do ensino à distância pela Internet, desenvolvida em todo o mundo. A ferramenta escolhida, neste trabalho para implementação e para testes de alta disponibilidade foi o TelEduc, apresentado na Figura 1. A motivação para a escolha dessa ferramenta foi a experiência do autor no seu uso e a utilização do TelEduc pelas Universidades em que o autor trabalha.

O TelEduc é um ambiente para a criação, participação e administração de cursos pela Internet (OEIRAS *et al.*, 2001). Ele foi desenvolvido pela Unicamp, e tem como principal característica uma metodologia seqüencial. Seu desenvolvimento teve início no ano de 1996 e, conforme (ROCHA; ROMANI; SILVA, 2000), teve como motivação os seguintes itens:

- escassez de recursos humanos para formação de professores em Informática Educativa; e
- ambientes existentes não atendiam as necessidades em termos metodológicos.

O desenvolvimento do TelEduc foi colaborativo, tendo como principal foco os professores. Durante o desenvolvimento, os usuários tiveram participação direta para a construção de um ambiente de acordo com as necessidades definidas por eles (ROCHA, 2002). Como descrito em (ROCHA *et al.*, 2002) e no site oficial do projeto², o TelEduc

¹Associação Brasileira de Educação a Distância - ABED: <http://www.abed.org.br>.

²TelEduc: <http://hera.nied.unicamp.br/teleduc/>

Figura 1: Ambiente Virtual TelEduc

foi concebido tendo como elemento central a ferramenta que disponibiliza Atividades. Isso possibilita a ação onde o aprendizado de conceitos em qualquer domínio do conhecimento é feito a partir da resolução de problemas, com o subsídio de diferentes materiais didáticos como textos, *software*, referências na Internet, entre outros.

A intensa comunicação entre os participantes do curso e ampla visibilidade dos trabalhos desenvolvidos também são pontos importantes. Por isso, foi desenvolvido um conjunto de ferramentas de comunicação como o Correio Eletrônico, Grupos de Discussão, Mural, Portfólio, Diário de Bordo, Bate-Papo etc., além de ferramentas de consulta às informações geradas em um curso como a ferramenta Intermap, Acessos, etc. Entre os usuários deste sistema estão: APAE, CAEM-IME-USP, UNASP, Marinha Brasileira e outros. O TelEduc é um *software* livre, de acordo com o termo da *General Public License*³.

3 Alta disponibilidade

Conforme (WEBER, 2001), sistemas totalmente infalíveis são impossíveis de ser praticados, porém através de aplicações de técnicas de tolerância a falhas é possível fazer o valor da disponibilidade de um serviço chegar bem perto dos 100%. As falhas em sistemas computacionais são inevitáveis, essas falhas podem ser de *software* ou *hardware*, porém a perda de dados e a paralisação do sistema podem ser evitadas pelo uso adequado de técnicas viáveis e de fácil compreensão. Essas técnicas possibilitam o administrador implementar sistemas tolerantes à falhas.

Toda vez que é falado em tolerância à falha é quase automático pensar na palavra redundância. Pode ser verificado em (CREVELING, 1956) que a redundância é tão antiga como a história da computação. É importante implementar um bom mecanismo de redundância para garantir a confiabilidade do sistema de tolerância à falha. A redundância para sistema de tolerância à falha pode aparecer de várias formas (WEBER, 2001):

³General Public License (GPL): <http://www.gnu.org/licenses/gpl.html>.

- redundância de *hardware*;
- redundância de *software*;
- redundância de informações (dados); e
- redundância de tempo.

Para um sistema de tolerância à falha ser eficiente, é necessário implementar a redundância de *hardware* e *software* e a técnica de detecção de erros. A partir desse momento, este texto adotará “sistema de tolerância à falha” e “alta disponibilidade” como sinônimos.

Alta disponibilidade dos sistemas tornou-se uma questão vital de sobrevivência empresarial, ou seja, se o sistema por algum motivo fica indisponível, a empresa pode ter prejuízos incalculáveis. Por outro lado, existe disponibilidade de *software* em Linux para implementar os níveis de alta disponibilidade, o que é ilustrado neste trabalho. Para implementar a redundância de *hardware*, foram tomados como base os equipamentos apresentados na Tabela 1.

Tabela 1: Configuração dos Computadores

Computador Primário	Computador Secundário
Pentium II	Pentium II
128 MB RAM	64 MB RAM
Partição / (4GB)	Partição / (2GB)
Partição /home (5GB)	Partição /home (5GB)

No computador Primário, foram instalados todos os programas necessários, inclusive o TelEduc, e o servidor Secundário deve ser uma imagem do servidor primário. Na Figura 2, encontra-se representada como sugestão a estrutura dos servidores. É importante observar que os dois servidores têm acesso externo independente e também existe um acesso interno entre os computadores. Esse acesso interno é fundamental para o *software* de detecção de erros. Nas seções seguintes, serão discutidos os aplicativos que realizarão a redundância e a detecção de erro.

4 Heartbeat

O programa Heartbeat é o responsável pela detecção de erro e pela monitoração dos sinais de vida dos nós das extremidades conectados ao *gateway* no qual está ativa. O termo “Heartbeat” significa batimento cardíaco. Este termo é usado para definir os pulsos enviados entre duas máquinas, que indicam que estão “vivas”. O programa Heartbeat está disponível no site <http://www.linux-ha.org/> e sua instalação segue o procedimento normal dos aplicativos Linux. A instalação deve ser feita nos dois computadores envolvidos no *cluster*. Em seguida, é necessário configurar o Heartbeat, que consiste em alterar os seguintes arquivos:

ha.cf: Nesse arquivo são configurados os computadores envolvidos e qual o meio de comunicação utilizado entre eles;

haresources: Nesse arquivo são configurados o número IP, que o servidor *backup* vai assumir, e quais serão os serviços que serão ativados;

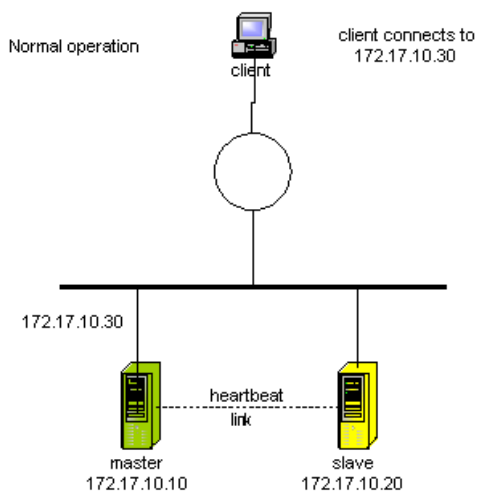


Figura 2: Alta Disponibilidade

authkeys: Contém a autenticação necessária entre os dois *hosts*.

A configuração desses arquivos deve ser idêntica tanto no servidor primário como no secundário, exceto o arquivo *haresource*, que não precisa ser configurado no primário. A Figura 3 apresenta as principais linhas de configuração do *ha.cf*. Como pode ser percebido, a configuração é extremamente simples, observando-se que:

1. Os nomes dos computadores envolvidos devem ser resolvidos através do arquivo */etc/hosts*, devendo ser colocado o nome que aparece na saída do comando `uname -a`;
2. A ligação entre os computadores pode ser serial, porém foi utilizado um cabo de rede normal para facilitar o envio dos arquivos de redundância;
3. Não deve ser diminuído muito a variável do *keepalive* por causa dos falsos positivos.

```
debugfile /var/log/ha-debug
keepalive 2
deadtime 30
ucast eth0 172.17.10.10
auto_failback off
node primario
node secundario
```

Figura 3: Arquivo *ha.cf*

O próximo arquivo a ser alterado é o *authkeys* (Figura 4), responsável pela segurança dos pacotes do Heartbeat. Se a rede envolvida for segura, pode ser utilizada a criptografia demonstrada no exemplo. Caso contrário é aconselhável utilizar SHA1 ou MD5, como também está no exemplo, porém comentadas.

```
auth 1
1 crc # 2 sha1 senha # 3 md5 senha
```

Figura 4: Arquivo authkeys

O último arquivo de configuração do Heartbeat, `haresource` (Figura 5), deve ser configurado apenas no servidor secundário. Nesse arquivo deve estar o número IP do servidor primário e os aplicativos que irão funcionar caso o primário pare.

```
primario 172.17.10.10 cluster.sh
```

Figura 5: Arquivo haresource

No caso da Figura 5, tem-se que `cluster.sh` é um *script* que inicia os aplicativos (Apache, MySQL e Sendmail). Esse *script* tem que estar dentro do diretório `resource.d` no diretório do Heartbeat. Além disso, o Heartbeat deve ser iniciado primeiramente no servidor primário.

5 Mon

O programa Mon é fundamental para o funcionamento do *cluster* de alta disponibilidade. A função desse programa é monitorar aplicações e conexões de rede e executar um alerta de acordo com o evento. Caso uma aplicação ou a conexão com a internet pare, o Mon aciona o programa Heartbeat. A sua configuração é bem simples, como é possível ver no exemplo apresentado na Figura 6. Mais detalhes podem ser verificados em seu *site*⁴.

```
hostgroup apache 172.17.10.10
hostgroup routers 172.17.10.1
watch apache
service http
interval 30s
monitor http.monitor -p 80 -u /index.html
allow_empty_group
period wd {Mon-Sun}
alert heartbeat.alert -S "O HeartBeat esta sendo alertado"
alertafter 2 30m
```

Figura 6: Arquivo mon.cf

6 Replicação de Dados

Como comentado anteriormente, o sucesso da Alta Disponibilidade para ensino a distância está diretamente relacionado aos serviços de réplica. Não tem valor algum o aplicativo funcionar no servidor secundário sem os dados. Portanto, serão analisados alguns meios de replicação de dados.

⁴Mon: <http://www.kernel.org/software/mon/>

Para a réplica dos dados ser realizado é importante conhecer o *software* que gera e utiliza esses dados, que neste estudo é o TelEduc. O TelEduc utiliza o MySQL para gravar os dados dos usuários e de gerenciamentos dos cursos. Além disso, também utiliza o disco para gravar arquivos, como os inseridos na “Parada Obrigatória”.

Neste projeto, foram analisados três formas de redundância dos dados, a primeira é sobre o MySQL. Pode ser utilizado o próprio banco de dados para fazer a redundância, ou pode ser enviado o banco de dados inteiro via réplica de disco. E na réplica de disco, a grande questão sobre o *software* a ser utilizado: alguns trabalhos apontam para o DRBD (REISNER, 2004) outros para RSYNC (SOBTI, 2004). Para resolver esses questionamentos, foram analisados os pontos positivos e negativos de cada aplicação, no contexto do TelEduc.

6.1 Replicação com MySQL

A réplica do banco de dados é feito alterando algumas configurações básicas do MySQL. Tudo que é relacionado ao banco é automaticamente replicado para outro banco. A configuração consiste na alteração do servidor primário e secundário. A configuração do MySQL do servidor primário basicamente consiste em alterar o arquivo `my.cf` e criar o usuário que vai ser responsável pela réplica. As alterações nos arquivos `my.cf` do servidores primário é apresentado na Figura 7.

```
[mysqld]
log-bin
server-id=1
```

Figura 7: Alterações no Arquivo `my.cf` do Servidor Primário

No arquivo `my.cf`, é necessário inserir a linha `log-bin` para ser gerado os registros de alteração do banco, para o servidor secundário saber que houve alteração. Além de alterar o `my.cf` é necessário criar um usuário para a réplica. Tendo sido criado um usuário, a atribuição de privilégios para réplica pode ser feita utilizando-se o comando apresentado na Figura 8.

```
mysql> GRANT REPLICATION SLAVE ON *.* TO repl@'%'
IDENTIFIED BY 'senha';
```

Figura 8: Configuração do Usuário `repl`

Na parte do servidor secundário que vai receber as alterações, é necessário alterar também o arquivo `my.cf` e configurar o número ip do servidor primário. No arquivo `my.cf` é definido o número de identificação do servidor secundário. Dentro do MySQL, podem ser executados comandos como os apresentados na Figura 9 para se definido o número ip do servidor primário. Essa é uma forma de fazer réplica do MySQL. Alternativa a essa opção é enviar os dados do banco via rede e fazer a réplica pelo disco. Nas próximas seções serão tratados esses casos.

```
mysql> CHANGE MASTER TO
      MASTER_HOST='192.168.100.1',
      MASTER_USER='repl',
      MASTER_PASSWORD='senha',
      MASTER_LOG_FILE='primario-bin.000001',
      MASTER_LOG_POS=412;
mysql> START SLAVE;
```

Figura 9: Apontando Servidor Primário no MySQL

6.2 DRBD

Como mencionando anteriormente, o DRBD (*Distributed Replicated Block Device*) é um programa de réplica de dados via disco. Ele trabalha como módulo do kernel, estando disponível nas principais distribuições Linux. O processo de instalação está descrito em seu *site*⁵. A utilização do DRBD pode ser feita através de linha de comando. Esse método é interessante para testes, porém, na prática é utilizado o arquivo de configuração `drbd.conf` apresentado na Figura 10.

```
resource drbd0 {
    protocol=C
    disk {
        do-panic
        disk-size=5004216
    }
    on server1 {
        device=/dev/nb0
        disk=/dev/hdd1
        address=172.17.10.10
        port=7789
    }
    on server2 {
        device=/dev/nb0
        disk=/dev/hdd1
        address=172.17.10.20
        port=7789
    }
}
```

Figura 10: Arquivo `drbd.conf`

Para iniciar o DRBD, deve-se começar pelo servidor primário, uma vez que no momento da iniciação, o servidor envia um broadcast na rede a procura de outro servidor. Caso encontre, será perguntado na hora da iniciação do programa se o DRBD é primário ou secundário. Na utilização do DRBD foram encontrados pontos positivos e negativos na sua utilização. A sincronização é feita *bit a bit*, o que nem sempre é interessante. Caso o HD seja muito grande, será transferido todo o HD, não importando o espaço que os dados estão utilizando. A Figura 11 apresenta a latência durante o sincronismo. Na Figura 12

⁵DRBD: <http://www.drbd.org>.

mostra o uso da memória e da CPU durante o mesmo. O programa utilizado para fazer os gráficos foi o MRTG.

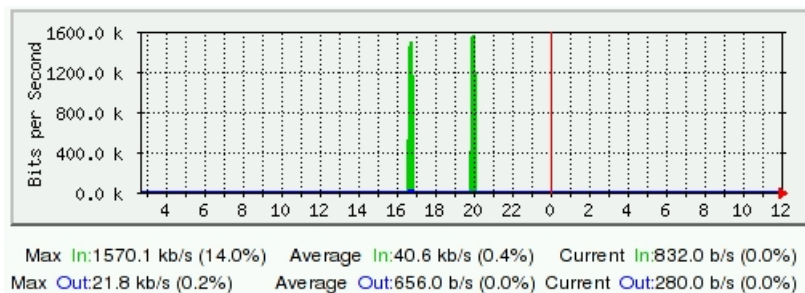


Figura 11: Latência – DRBD

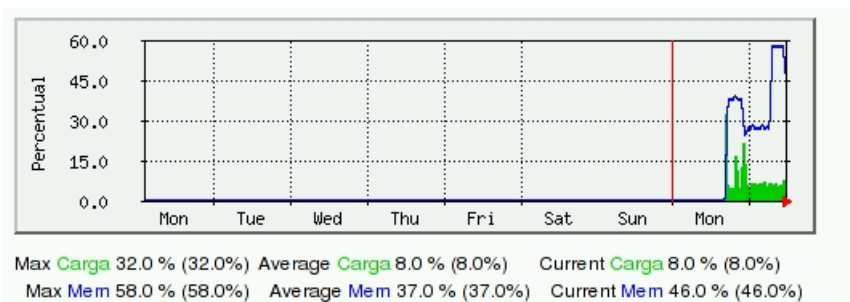


Figura 12: Consumo de Memória e CPU – DRBD

6.3 Rsync

Rsync é um *software* livre utilizado para transferência de dados via rede. Ele trabalha sobre o SSH, portanto a transferência é segura. O Rsync foi criado como substituição ao *rcp*, com algoritmos próprios. Neste trabalho o Rsync, foi instalado para comparação com o DRBD. O processo de instalação do Rsync é bem documentada, o que normalmente não traz trabalhos. Para o Rsync funcionar como um replicador de dados, confiável e viável para o ensino a distância, foi criado um *script* para automatizar seu uso, apresentado na Figura 13.

O *script* apresentado faz a réplica do diretório `/home` do servidor primário para o mesmo destino do servidor secundário. Ele fica em um *loop* infinito, porém, não é toda vez que é enviado arquivo para o servidor secundário. Só é enviado quando existir arquivos diferentes, ou quando é alterado algum arquivo. Isso diminuiu a latência e o consumo de memória e CPU. Na Figura 14, é apresentada a latência gerada pelo Rsync durante a instalação. Depois da instalação, o sincronismo foi bem menor e o consumo também

```
#!/bin/bash
DIR=/usr/local/rsync && touch $DIR/new.home && touch $DIR/old.home

while [ 1 ]
do
  find /home -cmin 1 {\textgreater} $DIR/new.home
  if [ ! -z {\textquotedbl}diff $DIR/new.home $DIR/old.home {\textquotedbl} ]; then
    if [ ! -z {\textquotedbl}cat $DIR/new.home {\textquotedbl} ]; then
      rsync -razpotg --delete /home/ root@server2:/home
    fi
  fi
  mv $DIR/new.home $DIR/old.home
done
```

Figura 13: Script para Automação do Rsync

abaixou. O consumo de CPU e memória durante a instalação do TelEduc é mostrado na Figura 15. Após a instalação, estes valores caíam, e tendendo para zero.

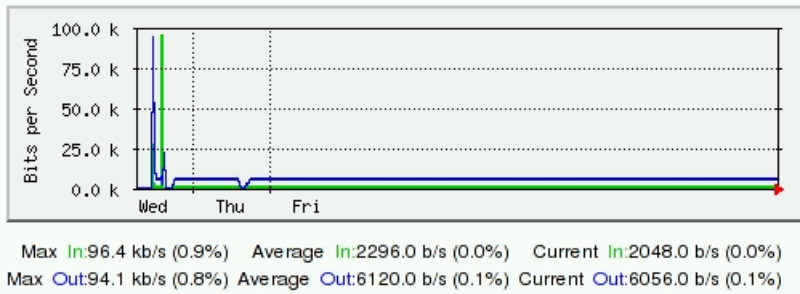


Figura 14: Latência – Rsync

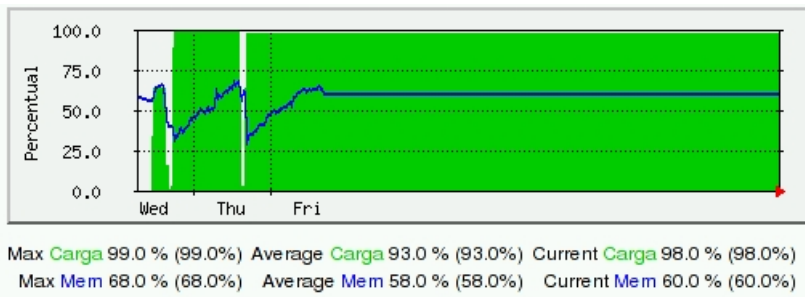


Figura 15: Consumo de Memória e CPU – Rsync

A restauração dos dados do servidor secundário para o servidor primário utilizando o Rsync é semelhante ao processo do envio dos arquivos do primário para o secundário. O

que muda é unicamente o sentido que os dados estão indo, e o número ip do servidor no script apresentado.

6.4 Análise e Discussão

O ponto positivo da réplica utilizando o MySQL é a velocidade que os dados são transferidos via rede. A latência da alteração é muito pequena. Na tentativa de medir a latência, foi utilizando o MRTG, porém ele não conseguiu capturar diferença na rede com o tráfego.

O ponto negativo é a volta dos dados. Em um cenário onde o primário falha e o secundário assume, e os dados são alterados no secundário. Esses dados precisam ser passados para o primário para o mesmo voltar a funcionar com as configurações do secundário. Esse processo de volta dos dados do secundário para o primário é complicado. Toda vez que o servidor pára é gerado um número de identificação que deve estar no servidor secundário. Além disso, só é possível fazer replica dos dados que estão no banco de dados.

Na utilização do DRBD, a principal vantagem é a boa velocidade na transferência dos dados. Outra vantagem é a possibilidade de definir a largura de banda que o *software* pode utilizar. A principal desvantagem é que a partição será replicada e a mesma é feita *bit a bit*, o que pode causar uso demasiado alto da largura de banda e processamento durante a transferência dos dados. Além disso, não há confirmação, se realmente os dados estão no servidor secundário. Se durante o sincronismo a comunicação cair, na maioria das vezes os dados ficam inconsistentes.

No Rsync, a principal vantagem é a flexibilidade na transferência dos dados, independente do sistema de arquivo ou diretório. Ele possibilita fazer a réplica de um único diretório e é de fácil instalação. Além disso, o Rsync possibilita confirmar se os dados foram gravados e tem a conversão mais simples de secundário para primário. A principal desvantagem do Rsync é largura de banda utilizada para a transferência, além de média de consumo de CPU e memória maior, em comparação ao DRBD.

Como análise geral, tomando como base os testes e os resultados discutidos anteriormente, o autor aponta que o Rsync mostrou ser a ferramenta que melhor se adapta no uso de replica para o ensino à distância (TelEduc). Os pontos que levaram a essa conclusão foram: desempenho satisfatório, flexibilidade e simplicidade de implantação.

7 Conclusões

Com a implementação do *cluster* de alta disponibilidade para o ensino à distância, foi possível detectar alguns pontos positivos nessa implementação. Entre os pontos positivos cita-se que a implementação é considerável fácil e rápida, consistindo na junção de 3 programas (Heartbeat, Mon, Rsync).

O programa HearBeat mostrou-se altamente confiável e funcionou corretamente nos momentos em que o servidor primário apresentava problema, automaticamente o secundário assumia. O tempo gasto para o secundário assumir levou em média 30 segundos, o que possibilitou a continuação da conexão. O programa Rsync mostrou-se ser uma solução melhor que o DRBD e a réplica via MySQL, devido às análises mencionadas. O programa Mon é um simples e fundamental programa para detectar erro.

O TelEduc aceitou muito bem a integração com alta disponibilidade. Nos testes realizados em nenhum momento deu problema. Quando o controle passava para o servidor secundário, o máximo que o usuário tinha que fazer era atualizar a página. Apenas durante a utilização do *chat*, que ele teve que sair e entrar outra vez, o que não gerou grandes problemas. Portanto é uma solução viável e segura e, além disso, é de baixo custo.

Referências

- ALVES, J. R. M. *Educação a Distância e as Novas Tecnologias de Informação e Aprendizagem*. 1994. WWW. Last Access: 25/07/2005. Disponível em: <<http://www.engenheiro2001.org.br/programas/980201a1.htm>>.
- ALVES, L. R. G.; NOVA, C. Educação à distância: limites e possibilidades. In: ALVES, L. R. G.; NOVA, C. (Ed.). *Educação à distância: uma nova concepção de aprendizado e interatividade*. São Paulo: Futura, 2003. p. 5–27. Disponível em: <<http://www.lynn.pro.br/livros.htm>>.
- CREVELING, C. J. Increasing the reliability of electronic equipment by the use of redundant circuits. In: *Proc. IRE*. New York: IEEE, 1956. v. 44, n. 4, p. 509–515.
- DIAS, S. P. *E-duc@re: Proposta de Ambiente de Aprendizagem Suportado pela Web para Cursos de Nível Superior Oferecidos a Distância*. Monografia (Monografia de Graduação) — DCC/UFLA, Lavras, 2003. Disponível em: <<http://www.comp.ufla.br/monografias/ano2003/ano2003.htm>>.
- NUNES, I. B. Noções de Educação a Distância. *Revista Educação a Distância*, n. 4/5, p. 7–25, 1994.
- OEIRAS, J. Y. Y.; ROCHA, H. V. da; FREIRE, F. M. P.; ROMANI, L. A. S. Contribuições de conceitos de comunicação mediada por computadores e visualização de informação para o desenvolvimento de ambientes de aprendizagem colaborativa. In: *Anais do XII Simposio Brasileiro de Informática na Educação*. Vitória: UFES/SBC, 2001. p. 127–135.
- PIRES, H. M. S. *Ambiente Virtual de Aprendizagem para Ensino a Distância - VLE*. Monografia (Monografia do curso de Especialização “Administração em Redes Linux”) — DCC/UFLA, Lavras, 2005.
- REISNER, P. *Rapid resynchronization for replicated storage Activity-logging for DRBD*. 2004. Disponível em: <http://www.drbd.org/fileadmin/drbd/publications/drbd-activity-logging_v6.pdf>.
- ROCHA, H. V. da. O ambiente TelEduc para educação a distância baseada na web: Princípios, funcionalidades e perspectivas de desenvolvimento. In: MORAES, M. C. (Org.). *Educação a distância: Fundamentos e práticas*. Campinas: Unicamp/NIED, 2002. cap. 11, p. 197–212.
- ROCHA, H. V. da; KROPIWIEC, D. D.; FUKAYA, S. M.; NETO, J. C.; GAYARD, L. A.; SILVA, C. G. da; FREIRE, F. M. P.; OEIRAS, J. Y. Y.; OTSUKA, J. L.; JÚNIOR, J. C. V.; ROMANI, L. A. S.; NETO, M. de S.; LACHI, R. L.; FERREIRA, T. B. *Projeto TelEduc: Pesquisa e Desenvolvimento de Tecnologia para Educação a Distância*. IX Congresso Internacional de Educação a Distância da ABED (Associação Brasileira de Educação a Distância), set. 2002.
- ROCHA, H. V. da; ROMANI, L. A. S.; SILVA, C. G. a. da. Slides de Palestra apresentada no Centro de Computação (CCUEC), Unicamp, *O Ambiente para Ensino a Distância TelEduc*. dez. 2000. WWW. Disponível em: <<http://teleduc.nied.unicamp.br/teleduc/>>.
- SOBTI, S. *Network-Embedded Programmable Storage and Its Applications*. 2004. Disponível em: <<http://www.cs.princeton.edu/~rywang/papers/networking04/>>.
- WEBER, T. S. *Tolerância a falhas: conceitos e exemplos*. Porto Alegre, 2001. Disponível em: <<http://www.inf.ufrgs.br/taisy/disciplinas/textos/>>.