

CLUSTER DE ALTA DISPONIBILIDADE COM ARQUITETURA HEARTBEAT

Jamilson Ramalho Dantas

Bacharelado do curso de Sistemas de Informação da Faculdade Sete de Setembro – FASETE

RESUMO

Este artigo descreve a construção de um servidor Web de Alta Disponibilidade, utilizando apenas software livre. Serão mostradas as funcionalidades de um sistema Alta Disponibilidade com arquitetura heartbeat, além do seu funcionamento em um modelo de dois nós. A pesquisa baseou-se em um estudo bibliográfico, sendo abordados os conceitos para a compreensão da proposta e orientação na montagem de um laboratório para o servidor web. Como resultados pode-se avaliar a viabilidade do serviço de alta disponibilidade para o uso das informações por parte das empresas, garantido confiabilidade e disponibilidade do serviço prestado.

Palavras-chave: Alta Disponibilidade, Heartbeat, Linux

INTRODUÇÃO

A base para a construção de uma empresa está diretamente ligada ao meio tecnológico, que necessita de componentes específicos para auxiliar o crescimento da mesma. Podendo ser destacado, servidores e unidades de armazenamentos de dados. Acoplados a isso estão os usuários que fazem uso destes elementos através da rede.

Os equipamentos tecnológicos são constituídos de componentes eletrônicos, componentes esses que possuem vida útil limitada, necessitando de manutenção e substituição com o passar do tempo. Quando algum desses componentes apresenta algum defeito pode ocorrer a parada de todo o sistema, deixando o usuário sem acesso às informações necessárias.

Com isso, há a necessidade da criação de um sistema de Alta Disponibilidade, que é capaz de detectar, recuperar e realizar o mascaramento de falhas, visando manter o funcionamento dos serviços, durante o máximo de tempo possível, inclusive, no processo de manutenções programadas. Um sistema com essa característica traz benefícios para todos da área de tecnologia, já que os custos associados são o menor possível.

Os funcionários da área de TI também ganham com a alta disponibilidade dos servidores implementadas, pois podem aproveitar melhor os recursos graças ao baixo custo de manutenção dos servidores reduzidos para implementar novos serviços e estudar novas tecnologias para agregar valor aos serviços já prestados pela empresa. (LINUX, 2009, p. 1)

Com isso é possível a criação e manutenção de um sistema de Alta Disponibilidade com um custo relativamente baixo, que podem reduzir gastos, aumentar lucros e conseqüentemente ter perdas menores com sistemas disponíveis.

1 DISPONIBILIDADE

A importância de um sistema disponível é notável em grande parte das tarefas executadas em nosso dia a dia, em que dispomos de sistemas informáticos, como por exemplo, na compra de um bilhete de passagem, compras em lojas virtuais e até mesmo em transações financeiras em agências bancárias. Essa disponibilidade torna-se tão importante a ponto de se tornar um fator determinante, se a empresa terá prosperidade em seus negócios ou chegará ao fracasso ou até mesmo à falência.

A disponibilidade de um serviço é calculada na percentagem que quantifica a probabilidade de encontrar o serviço operacional em determinado momento (FERREIRA e SANTOS, 2005).

Com isso, pode-se calcular o tempo em que um servidor estará disponível (*uptime*), ou o tempo em que o mesmo estará indisponível (*downtime*) (FERREIRA e SANTOS, 2005). Para o cálculo da percentagem, é utilizada a fórmula descrita abaixo, sendo que a mesma pode ser aplicada para as três classes de disponibilidades existentes descritas posteriormente.

$$\text{Disponibilidade} = (\text{Unidade de tempo total} - \text{downtime}) / \text{Unidade de tempo total}$$

Como exemplo, pode ser citado um servidor *Web* que está operando em um período de uma semana, considerando que o mesmo apresentou problemas ficando indisponível (*downtime*) durante 1 hora.

Unidade de tempo total = 1 semana (168 horas)

Downtime = 1 hora

$$\text{Disponibilidade} = (168 - 1) / 168 = 99,4047 \%$$

Esta percentagem é normalmente associada ao termo “número de noves de disponibilidade”, onde uma solução de “5 noves” possui um *uptime* de 99.999% (FERREIRA e SANTOS, 2005, p. 7).

Na tabela 1, o site da Microsoft *TechNet*, traduz o significado dos noves em um sistema disponível, enquanto mais nove tiver, menor será o *downtime* (Microsoft, 2009).

%	<i>Downtime</i> (por ano)
100	Nenhum <i>downtime</i>
99,999 (5 noves)	Menos de 5,26 min.
99,99 (4 noves)	De 5,26 a 52 min.
99,9 (3 noves)	De 52 min. a 8 horas e 45 min.
99 (2 noves)	De 8 horas e 45 min. a 87 horas e 56 min
90,0 - 98,9 (1 nove)	87 horas e 56 min a 875 horas 54 min

Tabela 1 – O significado dos noves

Fonte: Microsoft *TechNet*

Como podemos perceber, a quantidade de noves em um sistema de Alta Disponibilidade é modificado, conforme o tempo de *downtime* por ano, com isso houve a necessidade de classificar a disponibilidade de um sistema, conforme a quantidade de noves existentes sendo possível dividi-la em três classes: disponibilidade básica, alta disponibilidade e disponibilidade contínua.

A disponibilidade básica pode ser encontrada em todos os computadores, desde os mais simples até os mais avançados, sem a necessidade do uso de *software* ou *hardware* específico para o efeito. Assim sendo, um sistema de disponibilidade básica será capaz de atender ao cliente no tempo necessário, respondendo às expectativas do mesmo.

Com um sistema desse nível, pode-se realizar tarefas, desde as mais simples, até as mais complexas, ressaltando, então, que se ocorrerem falhas ou manutenções planejadas, o sistema ficará indisponível.

Segundo Guindani (2008, p. 66), um sistema com esse tipo de disponibilidade pode chegar a 99% a 99,9% de disponibilidade. Dizendo ele que, em um ano de operação, a máquina pode ficar indisponível por um período de 9 horas a quatro dias, o tempo de indisponibilidade pode ser visto na tabela 1.

A Alta Disponibilidade só é possível, por intermédio de dispositivos de *software* ou *hardware* que sejam capazes de detectar e recuperar *hosts* que obtiveram falhas. Segundo Guindani (2008, p. 66), nessa classe, as máquinas, normalmente, apresentam disponibilidade na faixa de 99,99% a 99,999%, podendo ficar indisponíveis por período de pouco mais de cinco minutos até uma hora em um ano de operação.

A Disponibilidade Contínua visa manter um sistema totalmente disponível, sem nenhuma falha, o que implica em 24 horas por dia e sete dias por semana. Segundo Ferreira e Santos (2005, p. 10), “a disponibilidade contínua (*continuous availability*) combina as características da operação contínua e da alta disponibilidade, representando um estado ideal”.

Com esse tipo de sistema, é feita a junção das outras duas características acima citadas, podendo ser criados *data centers* em pontos estratégicos de uma cidade, por exemplo, quando o *data Center* principal falhar o segundo assume os serviços, fazendo com que os dados trafeguem ininterruptamente.

2 HEARTBEAT

Heartbeat é um daemon que fornece infra-estrutura de cluster de comunicação e participação dos serviços aos seus clientes. Isso permite que os clientes saibam sobre a presença ou desaparecimento dos processos de outras máquinas, através de troca de mensagens com as mesmas (LINUX-HA, 2009).

Praticamente, existem duas versões principais do *heartbeat*, a versão 1.2.3, limitada à configuração de apenas dois nós em *cluster*, e a versão 2, que permite a configuração de até 16 nós em *cluster* (SAMBA, 2007).

O *HeartBeat* é responsável por monitorar os servidores do *cluster*, verificando se os mesmos estão ativos, em um *cluster* de dois nós, o *heartbeat*, apenas, informa ao servidor secundário que o principal está inoperante e que ele tem que assumir naquele instante. Com isso, o *heartbeat* do servidor secundário inicia os serviços configurados e passa a dispor informação necessária, deixando o sistema disponível.

Para realizar a instalação e configuração do *HeartBeat*, recomenda-se a utilização de servidores idênticos, tanto em software quanto em hardware. Desta forma, tanto o processo de instalação e configuração quanto o de manutenção serão facilitados.

Após a instalação e configuração do sistema operacional, é necessário realizar a instalação e configuração do *Heartbeat*, que está descrita na próxima seção.

2.1. INSTALAÇÃO E CONFIGURAÇÃO

Em sistemas operacionais baseados no Debian Gnu Linux, o processo de instalação pode ser realizado, a partir do terminal utilizando o comando *apt-get*:

```
apt-get install heartbeat
```

```
apt-get install apache2
```

Com isso, serão baixados e instalados a versão mais recente do *Heartbeat* e do *apache*. Em seguida, é necessário descompactar alguns arquivos e copiar os mesmos para uma pasta do sistema, em que o *heartbeat* buscará os arquivos de configuração, as linhas de comando abaixo efetua essa tarefa.

```
cd /usr/share/doc/heartbeat
```

```
sudo cp authkeys /etc/ha.d
```

```
sudo gzip -d ha.cf.gz
```

```
sudo cp ha.cf /etc/ha.d
```

```
sudo gzip -d haresources.gz
```

```
sudo cp haresources /etc/ha.d
```

Note que, apenas serão configurados três arquivos, *authkeys*, arquivo de criptografia, *ha.cf*, arquivo de configuração geral, e *haresources*, arquivo de inicialização de serviços que trabalharão em conjunto com o *heartbeat*.

authkeys

```
auth 1  
1 sha1 senhaparacriptografia
```

ha.cf

```
debugfile /var/log/ha-  
debug  
logfile /var/log/ha-log  
keepalive 1  
deadtime 10  
warntime 5  
initdead 30  
udpport 694  
bcast eth1  
auto_failback on  
node server01  
node server02
```

haresources

Server01 192.168.0.20 apache2

No arquivo *authkeys* é definido apenas uma senha que será entendida apenas pelo *heartbeat* e os servidores em questão. No *ha.cf* são definidos os *logs4* de erros, tempos de respostas e definição do nó morto, porta de comunicação, placa de rede usada pelo *heartbeat* e o nome dos nós que participarão do *cluster*. E, por fim, no arquivo *haresources* está definido qual é o servidor primário, o ip virtual que será distribuído e a inicialização do servidor *apache*, necessariamente nessa ordem, caso o servidor vier a ficar inoperante ele realiza a operação inversa, parando o *apache* e liberando o ip.

Essas configurações devem ser as mesmas para os dois nós configurados, bastando apenas, a configuração de um servidor e posteriormente a copia dos arquivos para o outro servidor.

3 IMPLEMENTAÇÃO DO CENÁRIO

Foi criado um cenário para implementação e testes do *heartbeat*, através da utilização de uma rede de computadores com a topologia representada na **Figura 1**.

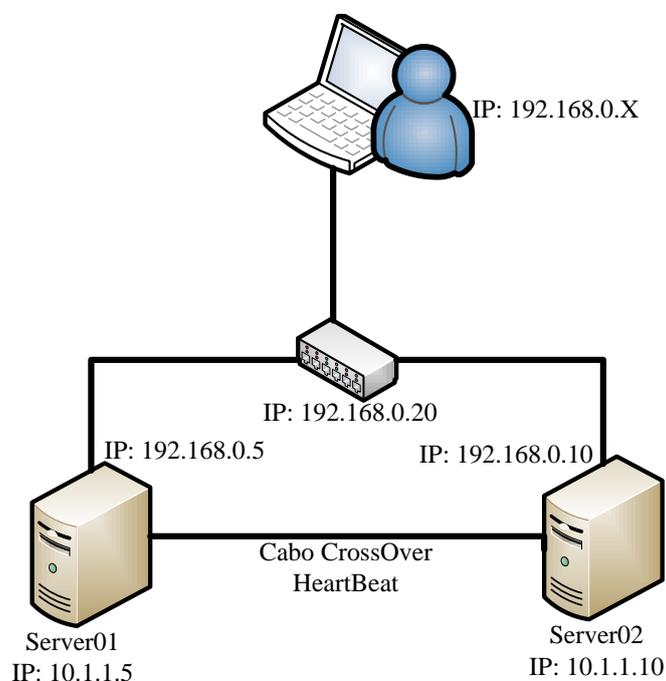


Figura 1 – Cluster de dois nós
Fonte: Dantas 2009

4 Arquivo gerado para descrever possíveis eventos, falhas, erros e até mesmo o comportamento de um determinado sistema.

Os dois servidores (Server01 e Server02) possuem as mesmas configurações e duas placas de rede em cada um deles, sendo uma para a comunicação do *heartbeat* e será atribuída uma faixa de ip apenas para realização dessa tarefa e outra para ser compartilhada entre os clientes através de um *switch* onde receberão outra faixa de *ip*.

O cliente acessará os serviços, através do *browser* utilizando o *ip* virtual *http://192.168.0.20*, configurado no arquivo *haresources*, esse *ip* irá direcionar o cliente para a página *web* associado ao servidor que estiver operando naquele instante, *server01*. Caso o *server01* vier a falhar ficando inoperante o *heartbeat* do *server02* irá verificar a indisponibilidade do mesmo e ele próprio irá iniciar suas atividades direcionando todo o acesso para si (figura 2).

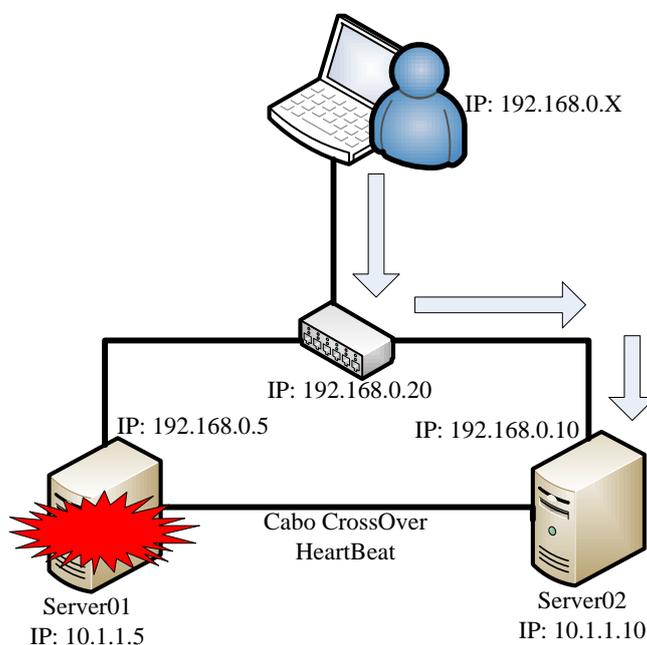


Figura 2 – Falha no cluster principal

Fonte: Dantas 2009

Nos testes realizados essa tarefa foi executada com sucesso, mas se ocorrerem falhas no cabo *crossover* responsável pela comunicação dos dois nós, ou mesmo se o cabo for desconectado, os dois computadores iniciarão os serviços do *heartbeat*, já que não haverá comunicação um com outro, ficando os dois com os serviços ativos, isso é inaceitável para serviços de alta disponibilidade, sendo necessária a implementação de soluções para o problema.

CONSIDERAÇÕES FINAIS

Quando se fala em alta disponibilidade, pensa-se logo em soluções proprietárias e custos elevados para instalação e manutenção do mesmo. Este artigo demonstra que é possível a construção desse tipo de sistema fazendo uso, apenas, de software de código aberto.

O *cluster* implementado aqui demonstra, de forma simples e objetiva, maneiras de obter alta disponibilidade em um servidor *web*, sendo assim, pequenas empresas podem adotar tal medida como solução de disponibilidade de seus serviços, tendo o menor custo possível.

REFERÊNCIAS

LINUX. **Your intelligence in Linux. Cluster e Alta Disponibilidade.** Disponível em: <<http://www.4linux.com.br/consultoria/cluster-alta-disponibilidade.html>> Acesso em: 22 Ago. 2009.

FERREIRA, Felipa Silva; SANTOS, Nélia Catarina Gaspar Gil dos. **Cluster de Alta Disponibilidade Abordagem *OpenSource*.** Portugal: 2005.

MICROSOFT TechNet. **Conceitos sobre Disponibilidade.** Disponível em: <<http://technet.microsoft.com/pt-br/library/cc668492.aspx>> Acesso em: 8 Set. 2009.

GUINDANI, Alexandre. **Gestão da Continuidade dos Negócios.** Disponível em: <http://www.upis.br/posgraduacao/revista_integracao/gestao_continuidade.pdf> Acesso em: 10 Set. 2009.

LINUX-HA. **Projeto High-Availability Linux.** Disponível em: <http://www.linux-ha.org/pt_BR/HomePage_pt_BR> Acesso em: 29 Ago. 2009.

SAMBA. **Heartbeat HA Configuration.** Modificado em 12 de março de 2007. Disponível em: <http://wiki.samba.org/index.php/5.0:_Heartbeat_HA_Configuration> Acesso em: 27 Set. 2009.

DANTAS, Jamilson Ramalho. **Cluster de Alta Disponibilidade Com Arquitetura *Heartbeat*, Um Projeto *Linux-Ha* Para *Software Livres*.** Paulo Afonso, BA, 2009.