

Virtualização e Alta Disponibilidade: Estudo de caso de Sistema para Diagnóstico Radiológico de Pneumonias

Paulo Júnior do Nascimento Lima¹, Iwens G. Sene Junior¹, Leandro Luís G. de Oliveira¹, Renato de Freitas Bulcão Neto¹

¹Instituto de Informática – Universidade Federal de Goiás(UFG)
Caixa Postal 15.064 – 91.501-970 –Goiânia– GO – Brasil

{paulolima, iwens, leandro, renato}@inf.ufg.br

Abstract. *The propose of this paper is to describe the state of the art in virtualization and high availability. As a result, we present an open source based software solution designed for the configuration of a diagnoses server of pneumonia, which in turn implements virtualization and high availability for the Linux environment.*

Resumo. *O objetivo deste artigo é apresentar o estado da arte acerca da virtualização e alta disponibilidade. Como principal resultado desse levantamento é apresentada uma solução baseada em software livre para configuração de um servidor para diagnóstico radiológico de pneumonias, sendo sua implementação com virtualização e alta disponibilidade para ambiente Linux.*

1. Apresentação

A virtualização, aliada com a alta disponibilidade, está bastante presente no mercado tecnológico atual. Isto se deve ao desenvolvimento e uso cada vez mais frequente das aplicações que utilizam serviços de *cloud computing*. Hoje em dia a virtualização tem focado em diminuir os recursos de *hardware* adquiridos, consumo de energia elétrica e gastos com manutenção e funcionários. Dentre as soluções existentes no mercado tecnológico, as *open source* tornam-se muito atraentes, na maioria das vezes, no quesito custos, e pelo fato de fornecerem um código-fonte aberto, sendo possível também melhorar ou personalizar tais soluções. Geralmente, essas aplicações ficam hospedadas em *clusters*(computacionais), tornando-as altamente disponíveis (FERREIRA, 2005). Essas máquinas físicas que abrigam os sistemas virtualizados ficam disponíveis quase em tempo integral, mesmo quando há manutenções a serem realizadas.

De uma maneira geral, os *clusters* podem ser formados por computadores dedicados ou agrupados fisicamente em um ambiente. Outra maneira de formação é através de uma conexão virtual de computadores espalhados em uma rede. Neste método, *softwares* apropriados auxiliam as estações distribuídas ao longo da rede para que possam servir à solicitações de terceiros com seus recursos computacionais (DANTAS, 2006). Assim, aplicativos agrupados foram desenvolvidos como forma de gerenciar uma carga maior de dados.

Nesse contexto, este artigo propõe o uso de ferramentas de *software* livre para aumentar a disponibilidade e segurança para hospedar o Sistema do servidor do projeto diagnóstico radiológico de pneumonias – DRP, juntamente com sua base de dados (OLIVEIRA, 2008). Uma das principais características do projeto é a tolerância a falhas,

pois o servidor deve sempre estar disponível para que as requisições cheguem e sejam atendidas.

2. Virtualização e Cluster

Atualmente o uso em ambientes computacionais de servidores virtualizados favorece em desempenho e ganho de produtividade. Esse aumento é possível em função de alguns componentes computacionais terem seus custos barateados, o que proporciona uma melhora no *hardware* o que contribuem diretamente no *software*. A seguir é apresentado com um maior detalhamento sobre virtualização e *clusters*.

2.1 – Virtualização

Em TI, virtualização é a capacidade de executar, em um único servidor físico, vários sistemas(máquinas virtuais) (CARISSIMI, 2008). Foi originada da técnica de particionamento, que divide um único servidor físico em múltiplos servidores lógicos. Depois da divisão do servidor físico, cada servidor lógico pode executar um sistema operacional e aplicativos de maneira independente, como ilustrado na Figura 1.

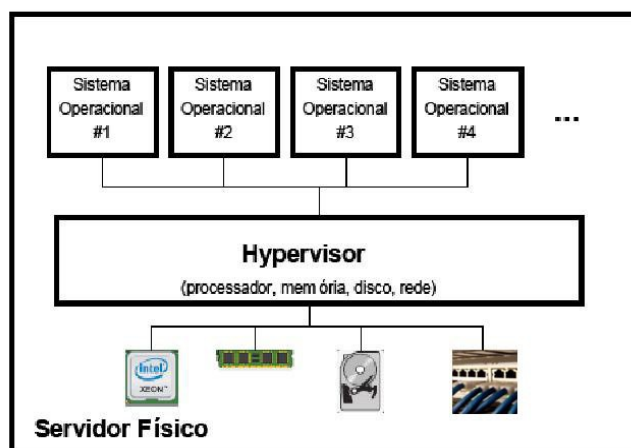


Figura 1 – Virtualização utilizando servidor físico.

Entre tantas soluções de virtualização, proprietárias e não proprietárias, as mais adotadas são dos fornecedores VMware, Citrix, Microsoft, IBM, Xen.org e KVM.org. Basicamente, as soluções para as plataformas x86 e RISC são muito semelhantes com relação as funcionalidades, administração e gerenciamento. As principais diferenças estão na aderência ao *hardware* e conseqüentemente às melhores utilizações dos recursos do *hardware*, ou seja; soluções com mais tempo em pesquisa, desenvolvimento, manutenção e atualização tem demonstrado maior eficiência.

2.2 – Cluster

Os *clusters* de computadores processam as tarefas paralelas de forma transparente, ou seja, aparentando ser um único sistema para o usuário. Para a implementação de um *cluster*, deve-se levar em consideração a aplicação que será executada, haja vista que, em problemas de granulosidade fina, o *cluster* não é tão eficiente quanto a um supercomputador, entretanto, para os problemas de granulosidade grossa, o *cluster* se mostra tão eficiente quanto um supercomputador (JAQUIE, 2011).

Assim, podemos dividir os *clusters* em duas categorias básicas: Alta Disponibilidade (HA – *High Availability*) e Alta Performance de Computação (HPC – *High Performance Computing*) (PITANGA, 2003).

Quando se fala de disponibilidade, pensa-se no tempo em que determinado sistema permanece ativo e em condições de uso. A alta disponibilidade se refere aos sistemas que praticamente não param de funcionar. Usados em aplicações de missão crítica, eles costumam ter meios eficientes de proteção e de detecção de falhas. Já alta *performance* refere-se à distribuição equilibrada de processamento aos nós (computadores) do *cluster*. É muito usado na internet, em servidores *Web* e de e-mail, comércio eletrônico e em bancos de dados distribuídos.

Nesse contexto, os *clusters* podem ser usados em uma infinidade de aplicações, basicamente em qualquer uma que exija grandes capacidades de processamento, solucionando problemas antigos da supercomputação, como o altíssimo custo de aquisição e manutenção dos equipamentos, uso de *softwares* proprietários e caros, total dependência dos fornecedores, bem como dificuldades de atualizações (LAGAR-CAVILLA, 2009).

3. Diagnóstico Radiológico de Pneumonia

O ambiente virtualizado, objetivo dessa proposta é parte do projeto de pesquisa “Inteligência Artificial na Medicina: Aplicação do Raciocínio Baseado em Caso no Auxílio ao Diagnóstico Radiológico de Pneumonias na Infância” da linha de pesquisa de “Computação Aplicada à Saúde” do LAVICOM (Laboratório de Visão Computacional) do INF-UFG (OLIVEIRA, 2008).

Esse Sistema contará com uma base de dados de 15.000 imagens radiológicas de crianças suspeita clínica de pneumonia. Essas imagens foram laudadas por dois radiologistas treinados de acordo com as diretrizes preconizadas pela Organização Mundial da Saúde – OMS. Essa base representa casos de diagnóstico positivo, negativo e indeterminados de pneumonia na infância e servirá como base de conhecimento para o sistema auxiliar nos diagnósticos de pneumonias.

O Sistema será implementando em linguagem Java™ com uso da API JAI (*Java Advanced Image*) e utilizará um Sistema Gerenciador de Banco de Dados de código aberto.

4. Alta Disponibilidade no Servidor DPM

Uma das tendências que fizeram com o conceito de virtualização retornasse fortemente foi a linguagem de programação Java (MAZIEIRO, 2008). Essa linguagem tornou-se multiplataforma e fornece um código altamente portátil. Atualmente, a virtualização vem se focando em diminuir os recursos de *hardware* adquiridos, consumo de energia elétrica, gasto com manutenção e funcionários. Os benefícios relacionados a virtualização são:

- a) Seleção de ambiente por tipo de aplicação, colocando aplicações similares e que executam de forma semelhante juntas;
- b) Balanceamento de carga entre as máquinas virtuais;
- c) Maior facilidade no gerenciamento dos servidores virtualizados;

- d) Redução dos custos de manutenção de *hardware* e consequente com energia elétrica.

Um sistema computacional altamente disponível está presente, principalmente, nos centro de dados das empresas líderes no ramo da computação. Esses hospedam os mais variados serviços, sendo que milhões de usuários podem fazer acessos diários. A alta disponibilidade está relacionada ao mascaramento à ocorrência de falhas, o usuário que usa um sistema altamente disponível mal sabe que o sistema pode estar parcialmente inativo por uma falha de qualquer de *hardware* ou recebendo manutenções, tais como:

- a) Checagem de disco;
- b) Atualização do sistema operacional e dos serviços que ele dispõe;
- c) Manutenção do *hardware* utilizado;
- d) Eventual manutenção do local que abriga essas máquinas.

Em um *cluster*, outra máquina física pode assumir o papel da que ficou inativa de forma transparente para o usuário, aumentando a disponibilidade dos serviços. É importante que falhas sejam corrigidas e manutenções levem o menor tempo possível para manter um bom nível de disponibilidade do serviço prestado por essas máquinas e diminuir a carga que cada máquina física assume quando uma outra está inoperante. Assim, um conjunto de computadores agindo como um só e cada um monitorando o outro a fim de assumir a função de outra máquina assim que essa parar de prestar tal serviço (*failover*) até que essa máquina fique ativa novamente (*failback*) (CARISSIMI, 2008).

A porcentagem da Disponibilidade pode ser calculada a partir da seguinte equação: $\text{Disponibilidade} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$ (FILHO, 2002)

Onde MTBF é o tempo médio entre falhas e MTTR é o tempo médio que esse componente levou para ser reparado. Um dos conceitos que sustentam a Alta Disponibilidade é Tolerância à Falhas. Esse eixo é responsável por tratar uma falha enquanto essa falha ainda não se tornou um erro.

As soluções que se aproximam de 100% de disponibilidade exigem a redundância da estrutura para evitar a existência de pontos únicos de falha, conhecidos também pela sigla em inglês SPOF (*Single Point Of Failure*). Um ponto único de falha é simplesmente um recurso do sistema em que, caso falhe, provoca a indisponibilidade de todo o sistema. Óbvio e infelizmente, reduzir os pontos únicos de falha exige custos e adiciona um considerável grau de complexidade na infra-estrutura.

A redundância da estrutura é, portanto, um dos requisitos para se conseguir alta disponibilidade, e deve ser combinada com uma camada de *software* capaz de monitorar e assumir os serviços de um servidor em produção quando nele ocorrer uma falha. Exatamente essa camada de *software* é preenchida pelo *Heartbeat* e o DRBD.

O *Heartbeat* pode ser considerado o núcleo do ambiente de alta disponibilidade. Porque é sua a responsabilidade de monitorar os servidores em produção e, em caso de falha, realizar automaticamente os procedimentos para preservar o funcionamento do sistema como um todo.

Através de um meio de comunicação, que pode ser *Ethernet* ou *Serial*, um servidor redundante verifica a disponibilidade do servidor em produção enviando-lhe

uma mensagem e exigindo a resposta. Essa checagem é feita entre as duas instâncias do *Heartbeat* instaladas nos dois servidores. Se por algum motivo o servidor em produção não responder, ele será considerado indisponível, e então o *Heartbeat* do servidor redundante automaticamente providencia a configuração e inicialização dos serviços locais, além de outros recursos, como o endereço IP, partições de disco, etc.

Conforme a Figura 2, um diagrama envolvendo dois servidores: *webserver1* (produção) e *webserver1-bkp* (redundante). Ambos configurados com o *Heartbeat* para prover alta disponibilidade de um serviço Web (Apache):

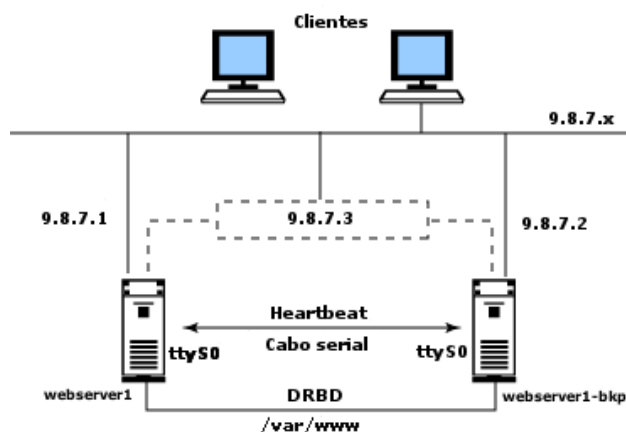


Figura 2- Detalhamento da solução utilizando *Heartbeat* e DRBD.

Um ponto importante de observação é que o próprio *Heartbeat* controla a inicialização de determinados serviços e recursos dos servidores, como o Apache e o endereço IP virtual. Logo, eles não devem ser configurados para inicializar durante o processo de *bootstrap* do sistema operacional, apenas no *Heartbeat*.

Assumindo o controle total dos serviços e recursos do ambiente de alta disponibilidade, o *Heartbeat* evita qualquer tipo de conflito que possa afetar o correto funcionamento do sistema. Entretanto, não é seu objetivo garantir a sincronia e a integridade dos dados entre os servidores. Para conseguir isso, o *Heartbeat* tem que agir em conjunto com algum *software* que se encarregue de manter os mesmos arquivos do servidor em produção também no servidor redundante. Essa será justamente a função do DRBD.

O DRBD (*Distributed Replicated Block Device*) consiste em um módulo para o *kernel* Linux que faz o espelhamento dos dados de um dispositivo de bloco (partições de disco) entre diferentes servidores, interligados geralmente através de uma rede Ethernet. Para quem já é familiarizado com o conceito de RAID, o DRBD pode ser muito bem explicado como sendo um RAID-1 via rede.

Cada dispositivo de bloco envolvido na configuração do DRBD tem um estado, que pode ser primário ou secundário. Todas as operações de escrita feitas no primário são replicadas para o secundário, sendo que o protocolo padrão de replicação garante a sincronia e a integridade dos dados replicados. Quanto as operações de leitura, elas são sempre realizadas localmente.

6. Conclusão

Essa tecnologia facilita bastante a administração e o gerenciamento. Isso permite a criação de ambientes para a execução, pesquisas e até mesmo produção, de forma simples e rápida.

Além disso, sua adoção permite uma grande diminuição com custos de obtenção de equipamentos, diminuindo o “lixo tecnológico” e consumo elétrico e com a diminuição na emissão do carbono (carbon footprint) produzido na manufatura dos servidores (TRAVASSOS, 2010).

Em meio a algumas soluções proprietárias para construir um ambiente de alta disponibilidade, o *Heartbeat* e o DRBD conseguem juntos excelentes resultados para sistemas Linux, e sem absolutamente nenhum custo por se tratarem de *softwares* livres. A maturidade desses *softwares* atingiu um certo nível que venceu determinadas desconfianças presentes no rigoroso mercado corporativo (FERRAZ, 2002).

Referências

- Carissimi, A. Virtualização: da teoria a soluções. 26º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Março de 2008.
- Laureano, M. A. P.; Mazieiro, C. A. Virtualização: Conceitos e Aplicações em Segurança 8º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2008, Minicurso 1.
- Lagar-Cavilla, H.A.; Whitney, J.A.; Scannell, A.M.; Patchin, P.; Rumble, S.M.; de Lara, E.; Brudno, M. & Satyanarayanan, M., SnowFlock: rapid virtual machine cloning for cloud computing, EuroSys '09: Proceedings of the fourth ACM European conference on computer systems. ACM, 2009, pp. 1-12.
- Shaikh, Hidayatullah (12-10-2004). High-availability middleware on Linux, *IBM DeveloperWorks*. Acessado em 04-02-2011.
- Dantas, M. Ambientes Distribuídos de Alto Desempenho: Clusters e Grades Computacionais. [S.l.:Portal Brasileiro sobre computação de alto desempenho, 2006.
- Pitanga, M. Computação em cluster: o estado da arte da computação. Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2003. 322p.
- Jaquie, K. R. L. Extensão da Ferramenta de Apoio à Programação Paralela (F.A.P.P.) para ambientes paralelos virtuais. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08022001-095456/>>, visitado em: 23 de fevereiro. 2011.
- Oliveira, L. L.G.; Silva, S. A.; Ribeiro, L. H. V.; Oliveira, R. M.; Coelho, C. J.; Andrade, A. L. S. S. . Computer-aided diagnosis in chest radiography for detection of childhood pneumonia. *International Journal of Medical Informatics*, v. 77, p. 555-564, 2008.
- Ferreira, F. S. F. ; dos Santos, N. C. G. G. Clusters de Alta Disponibilidade abordagem OpenSource. Disponível em <<http://mosel.estg.ipleiria.pt/files/Artigo.pdf>>, visitado em 08 de outubro de 2011.
- Filho, N. A. P. Linux, Clusters e Alta Disponibilidade. Disponível em <<http://www.ime.usp.br/~nelio/publications/>>, visitado em 09 de outubro de 2011.

Travassos, V. Sistemas Virtualizados. Disponível em <www.ic.unicamp.br/~ducatte/mo401/1s2010/T2/078272-t2.pdf>, visitado em 09 de outubro de 2011.

Ferraz, N. C. T. Vantagens Estratégicas do Software Livre para o Ambiente Corporativo. Disponível em <softwarelivre.org/cooperativas/veslac-vantagens-estrategicas-do-software-livre-para-o-ambiente-corporativo.pdf>, visitado em 09 de outubro de 2011.