



CLUSTER DE ALTA DISPONIBILIDADE DE BAIXO CUSTO PARA SERVIDORES WEB NO PODER LEGISLATIVO MUNICIPAL

Danilo M. FUGI¹, Augusto M. S. JÚNIOR²

RESUMO

Este artigo descreve a implementação de um conjunto de computadores interligados através de uma rede, conhecido como *Cluster*, para solucionar o problema de disponibilidade do servidor *web* da Câmara Municipal de Areado - MG. Este *Cluster* de Alta Disponibilidade possui dois computadores convencionais, sistema operacional *Linux* e *softwares Open Source*, que atingiu resultados esperados, atingiu Alta Disponibilidade, e manteve o servidor *web* em funcionamento.

INTRODUÇÃO

As redes de computadores se tornaram, atualmente, um dos principais meios de comunicação e acesso a informações, são formadas, a grosso modo, por componentes chamados clientes e servidores. Os Servidores, segundo Tanenbaum (2003), tem a finalidade de disponibilizar inúmeros serviços. Notoriamente os dispositivos eletrônicos que compõem estes servidores estão sujeitos a falhas, tanto físicas quanto lógicas. Estas falhas fazem que o servidor fique indisponível, que traz enormes prejuízos e perda de confiabilidade.

A partir da publicação da Lei Federal 12.527³, os órgãos públicos receberam prazos que os obriga a disponibilizar informações aos cidadãos. Manter os Servidores

¹ Laboratório de redes e sistemas distribuídos – (LARS) - Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais - Campus Muzambinho - Muzambinho – MG – Brasil - danilofugi@gmail.com

² Laboratório de redes e sistemas distribuídos – (LARS) - Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais - Campus Muzambinho - Muzambinho – MG – Brasil - augusto.junior@muz.ifsuldeminas.edu.br

³ http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm

web da Câmara Municipal de Areado – MG em funcionamento, para atender esta exigência, se torna um desafio.

O cidadão ao acessar as informações no servidor da Câmara Municipal deve encontrar o que busca. Este serviço estava hospedado em um servidor terceirizado, que não possuía recursos de réplica e backup e muitas vezes se encontrava indisponível por ter ocorrido alguma falha. Este servidor ao estar indisponível pode fazer com que as informações essenciais não possam ser acessadas por pessoas interessadas.

Ao utilizar um plano de gerenciamento de falhas é possível manter o servidor ativo por mais tempo. Este gerenciamento, como explicado por Zem (2005), utiliza uma gama de recursos para reduzir as falhas, como equipamentos especiais de redundância e tecnologia.

Uma solução é o uso de *mainframes* terceirizados, que garantem alta disponibilidade e dominam as técnicas de gerenciamento de falhas, de acordo com Weber (2002), possibilita disponibilizar um serviço de qualidade e que possa se recuperar de forma eficiente, mas possuem custos elevados e chegam a ser inviáveis.

Uma outra maneira de gerenciar as falhas e reduzir os gastos é a utilização de softwares livres e computadores comuns na construção de um modelo chamado *Cluster* de Alta Disponibilidade, conceituado por Ferreira, Santos e Antunes (2005), é formado por um conjunto de computadores interligados através de uma rede, que se apresentam ao usuário como sendo um único e grande computador.

Zem (2005) citou o *Cluster* de **Alta Disponibilidade**: (*High Availability – HA*) que são caracterizados por se manter em pleno funcionamento por um longo período de tempo e disponibiliza o mesmo recurso em todas as máquinas da rede, onde existe um servidor ativo e os outros ociosos.

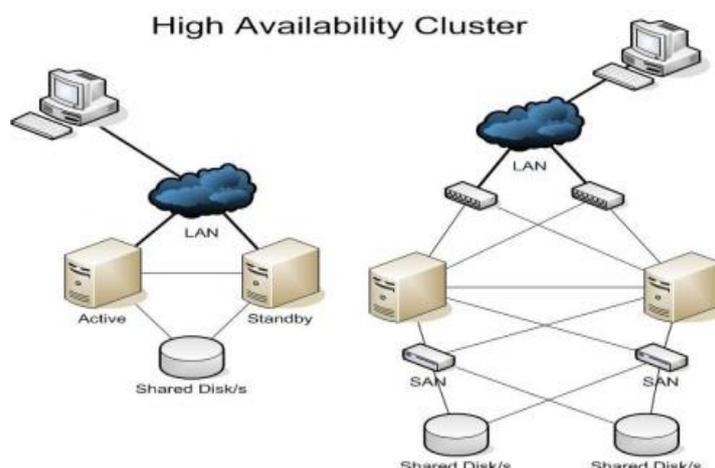


Figure 1: *Cluster* de Alta Disponibilidade (HA)

Fonte: <http://web.science.mq.edu.au>

Lopes Filho (2008) nos mostrou que Alta Disponibilidade é a classificação de um servidor que fica em funcionamento 99,99% (noventa e nove virgula noventa e nove por cento) do tempo durante um ano.

Diante disso, empregar o *Cluster* de Alta Disponibilidade formado com computadores convencionais juntamente com softwares livres, para solucionar o problema de disponibilidade do Servidor *Web* da Câmara Municipal de Areado viável, pois garante que o servidor esteja disponível para acesso 99,99% (noventa e nove virgula noventa e nove por cento) do tempo. Praticamente com custo zero (muito bem visto no orçamento público) e máxima eficiência de disponibilidade de informações.

MATERIAL E MÉTODOS

Os materiais utilizados na implementação do *Cluster* de Alta Disponibilidade foi a infraestrutura de rede já disponível e computadores comuns do patrimônio da Câmara Municipal, que se encontravam desligados e são dois: **computador01**: 1GB de Memória *RAM*, 80GB de *HD*, Processador Pentium 4 e **computador02**: 2GB de Memória *RAM*, 160GB de *HD*, Processador Celeron D. Depois de limpos, foi feita a instalação do Sistema Operacional *Linux*, distribuição *Debian*, versão 7.1, e reservada uma partição do *HD* para ser replicada entre os computadores através de *software*.

Bacellar (2008) explica que o *Cluster* pode conter em sua formação computadores não necessariamente idênticos e se apresenta de forma transparente ao cliente, utilizando um *Middleware*, que foi conceituado por Tanenbaum (2003) como sendo um software que permite ao sistema distribuído conseguir uniformidade na presença de diferentes hardwares, assim a diferença entre os computadores foi contornada de forma eficiente.

Antes da instalação dos *softwares* do *Cluster* procedeu-se a configuração da rede que obedeceu a faixa de endereços já configurada nos outros computadores, logo após foi aplicado aos dois computadores do *Cluster* o *software* de SSH (*Secure Shell*) que garante comunicação segura entre os nós e possibilita o acesso remoto.

Explicado por Dake, Caulfield e Beekhof (2008), o **Corosync Cluster Engine** foi formalizado na conferência Ottawa Linux Symposium e foi uma redução do *OpenAIS*, que é um sistema de comunicação de grupo para a implementação de alta

disponibilidade dentro de servidores Linux, é considerado a evolução do *Software Heartbeat*, que possui recursos adicionais e mais suporte. Este software instalado e configurado cria o *Cluster* preparado para ter várias funcionalidades

A API (*Application Programming Interface*) interna do *Corosync* fornece os serviços através de uma implementação de um *Totem - Single Ring Ordering and Membership Protocol* que fornece o modelo virtual para troca de mensagens e membros do *Cluster*, recurso confiável para a criação de réplicas de estado, reinício da aplicação em caso de falhas e sistema de quórum, que notifica sempre que a aplicação ou máquina venha a falhar.

O software *Pacemaker*, segundo Haas (2012) é um gerenciador de recursos de *Cluster* ou *Cluster Resource Manager* (CRM), sua função é oferecer uma interface para que o administrador possa gerenciar o *Cluster*. Possui muitas funcionalidades como capacidade de replicar as informações do *Cluster* entre os nós de maneira transparente e eficiente, todas as tarefas de configuração precisam ser executadas em um nó e já são replicadas para todo o *Cluster* automaticamente.

O *Cluster* possui regras, nós de **failover**, é a capacidade de determinado serviço ser migrado automaticamente para um outro servidor, **failback**, é o processo de restauração de um serviço que se encontra em um estado de **failover**, monitoramento, migração e muitos outros, recursos de *Open Cluster Framework* (OCF).

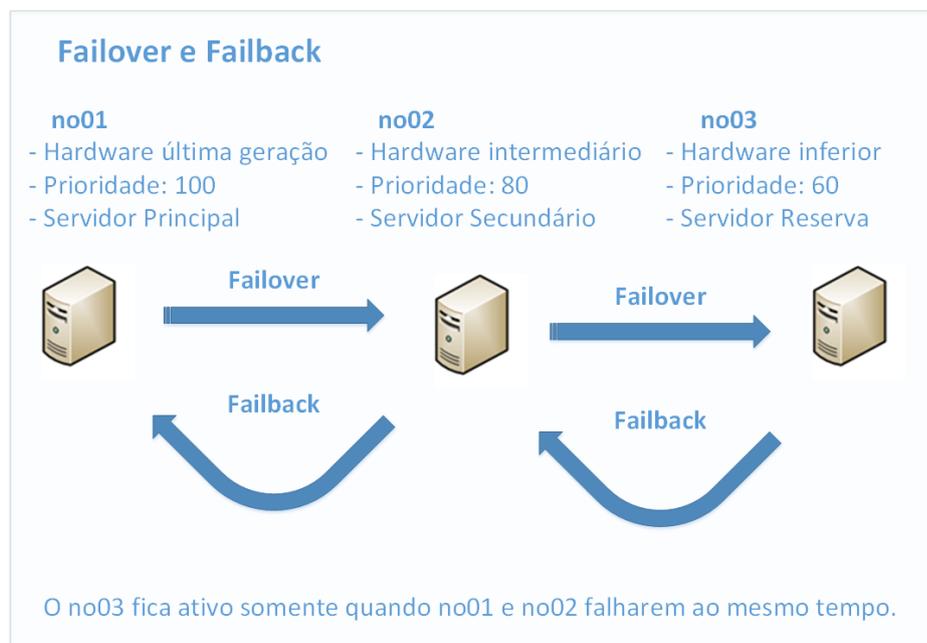


Figura 02 – Failover e Failback

Distributed Replicated Block Device (DRBD), segundo Pla (2006), nada mais é que um módulo para espelhamento de armazenamento distribuídos, geralmente utilizado em *Clusters* de alta disponibilidade - *HA*, isto é feito espelhando as partições do disco rígido via rede. O *DRBD* funciona como um sistema *RAID* baseado em rede, que cria em todos os nós um dispositivo virtual inacessível diretamente e toda a escrita é realizada no nó primário, que irá transferir os dados para a partição e propagá-los para os restantes dos nós secundários.

Os testes de disponibilidade realizados com as próprias ferramentas do *Cluster*, através do *software Pacemaker* juntamente com o *Software LCMC - Linux Cluster Management Console* que é uma interface gráfica, construída em *JAVA*, para gerenciar, representar e monitorar o estado e relações entre os serviços de um *Cluster*, identificando seus nós e recursos, ativos do sistema. Este software foi utilizado para monitoramento, onde é possível acessar o *Cluster* remotamente e gerenciar os recursos e prioridades de servidores.

RESULTADOS E DISCUSSÃO

Aplicar o *Cluster* construído como *Servidor Web* na sede da Câmara Municipal de Areado, para reduzir os custos com servidores terceirizados e aumentar a eficiência. De forma preliminar foi possível constatar que houve melhoras em disponibilidade, pois as falhas ocorridas durante um mês (30 dias) de funcionamento foram apenas 9 segundos, isto é, o servidor falhou 3 vezes e o servidor 2 que estava ocioso, assumiu os recursos após 3 segundos em cada falha, através deste monitoramento e testes do sistema como um todo pode-se perceber que o sistema superou os limites de Alta Disponibilidade (99,99% do tempo) e permaneceu disponível mesmo ocorrendo falhas em um dos servidores.

Estes testes preliminares é notório que a solução é satisfatória. E nas averiguações foi constatado que as falhas do servidor 1 se deram por superaquecimento e desligamento automático, com o tratamento destes pontos críticos de falhas é possível saná-los.

CONCLUSÕES

O emprego do *Cluster* de Alta Disponibilidade (*HA*) para *Servidores Web* da Câmara Municipal de Areado é uma das soluções que tem a melhor relação custo-benefício, pois os computadores utilizados são comuns e já disponíveis e, mesmo com

resultados preliminares, apresentou frutos muito além dos esperados, que atingiu os níveis exigidos para classificação de Alta Disponibilidade que satisfaz os objetivos do trabalho. Ao ocorrer as falhas, o sistema se recuperou quando transferiu, automaticamente, o recurso do computador com falha para outro servidor, que manteve as informações disponíveis aos computadores clientes de forma transparente e imperceptível. A vantagem da utilização desta tecnologia de *Cluster* é o corte com gastos de terceirização do serviço e a manutenção programada, que mantém o recurso web disponível mesmo com o desligamento de algumas máquinas.

REFERÊNCIAS BIBLIOGRÁFICAS

BACELLAR, Hilário Viana. *Cluster: Computação de Alto Desempenho*. Campinas: Instituto de Computação, 2010. Disponível em <http://www.ic.unicamp.br/~ducatte/mo401/1s2010/T2/107077-t2.pdf>, 21 de fevereiro de 2015.

DAKE, Steven C.; CAULFIELD, Christine; BEEKHOF, Andrew. The corosync *Cluster* engine. In: Linux Symposium. 2008. Disponível em <http://landley.net/kdocs/mirror/ols2008v1.pdf>, 22 de fevereiro de 2015.

FERREIRA, Filipa; SANTOS, Nélia; ANTUNES, Mário. *Clusters de alta disponibilidade—abordagem OpenSource*. Relatório desenvolvido na disciplina de Projecto I, 2005. Disponível em http://www.faceca.br/bsi/documentos/cluster_open_source.pdf, 20 de fevereiro de 2015.

HAAS, Florian. Ahead of the pack: the pacemaker high-availability stack. *Linux Journal*, v. 2012, n. 216, p. 4, 2012. Disponível em <http://dl.acm.org/citation.cfm?id=2208863>, 22 de fevereiro de 2015.

LOPES FILHO, Edmo. Arquitetura de alta disponibilidade para firewall e IPS baseada em SCTP. 2008. Tese de Doutorado. Universidade Federal de Uberlândia. Disponível em [http://repositorio.ufu.br/bitstream/123456789/632/1/ArquiteturaAlta Disponibilidade.pdf](http://repositorio.ufu.br/bitstream/123456789/632/1/ArquiteturaAlta%20Disponibilidade.pdf), 22 de fevereiro de 2015.

PLA, Pedro. Drbd in a heartbeat. *Linux Journal*, v. 2006, n. 149, p. 3, 2006. Disponível em <http://dl.acm.org/citation.cfm?id=1152902>, 22 de fevereiro de 2015

TANENBAUM, A. S. *Sistemas Operacionais Modernos*. 3. ed. Pearson Education do Brasil Ltda, 2003.

WEBER, Taisy Silva. Um roteiro para exploração dos conceitos básicos de tolerância a falhas. Relatório técnico, Instituto de Informática UFRGS, 2002. Disponível em <http://www-usr.inf.ufsm.br/~ceretta/papers/TaisyDependabilidade.pdf>, 22 de fevereiro de 2015.

ZEM, J. Luís. Um *Cluster* de computadores de uso geral. Universidade Metodista de Piracicaba, 2005. Disponível em http://www.unimep.br/~jlzem/lecc/trabalhos/artigo_sucesu_2004.doc, 22 de fevereiro de 2015.