IX SEGeT 2012
SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA
Tema: Gestão, Inovação e Tecnologia para a Sustentabilidade

Proposta de Método Preemptivo para Sistema de Tolerância a Falhas Supervisionar e Corrigir Registros de Aplicações Web em Servidores de Dns

Jucelino Dos Reis
jucelinodosreis@gmail.com
UNITAU

Francisco Carlos Parquet Bizarria
bizarriafcpb@iae.cta.br
UNITAU

José Walter Parquet Bizarria
jwpbiz@gmail.com
UNITAU

Resumo: Este trabalho apresenta uma proposta de método preemptivo para auxiliar sistemas de tolerância a falhas de aplicações WEB, que elimina a dependência existente entre os sistemas tradicionais de tolerância a falhas e os usuários utilizadores dos sites Web supervisionados. A proposta emprega a técnica de atualização dinâmica de DNS para minimizar as limitações dos sistemas tradicionais de tolerância a falhas, que são ineficientes quanto a garantia da integridade de registros em servidores de DNS relacionados aos sites das aplicações Web. A validação da eficácia do método apresentado neste trabalho é obtida por meio de testes em protótipo que adota os principais módulos previstos no sistema. Os resultados satisfatórios observados nesses testes indicam que a proposta apresentada é adequada para finalidade a que se destina.

Palavras Chave: Tolerância a falhas - Aplicação Web - Alta-disponibilidade - -

1. INTRODUÇÃO

Existem várias técnicas de distribuição de acesso de usuários para as aplicações Web. Nesse sentido, este trabalho aborda duas técnicas tradicionais utilizadas por sistemas de recuperação de falhas para aplicações Web (BIG-IP-GTM, 2010), (LVS, 2010), (CDD, 2010), (BIG-IP-GTM, 2010). Esses métodos são denominados como redirecionamento de acesso por DNS e redirecionamento de sessão HTTP (*HyperText Tranfer Protocol*). Ambas as técnicas são utilizadas por sistemas de tolerância a falhas de aplicações Web para distribuir o acesso de usuários nessas aplicações. Neste trabalho a aplicação Web é convencionada como um aplicativo que se fundamenta no modelo de arquitetura Web, na qual servidores Web permitem acesso as aplicações publicadas mediante algum software de gerenciamento de sites Web, como por exemplo: o I.I.S. (*Internet Information Services*) (IIS, 2009) ou o Apache (BOWEN, 2000). Essas aplicações são acessadas por navegadores Web, a partir de microcomputadores de usuários conectados a rede de comunicação de dados, assim como em redes de intranet corporativas (ANSWERS, 2011).

No exemplo da Figura 1(i) o sistema de tolerância a falhas usa a técnica de redirecionamento de acesso por DNS. Nesse método o resolvedor de nomes do computador pessoal (PC) do usuário "a" consulta "1" o servidor de DNS para obter o endereço IP (*Internet Protocol*) para acesso a aplicação Web. Na técnica de redirecionamento de acesso por DNS, o sistema de tolerância a falha é obrigado a responder pela tradução de nome para as aplicações Web que são supervisionadas pelo sistema. O servidor de DNS já existente na rede precisa consultar o sistema de tolerância a falhas "2", para obter qual é endereço que o usuário deve utilizar para acesso a aplicação Web. A limitação desse método está no sistema de tolerância a falhas ser obrigado a traduzir todas as solicitações de resolução de nomes "3" para os sites supervisionados pelo sistema. Somente após o sistema de tolerância a falhas informar ao servidor de DNS o endereço IP que deve ser usado para acesso a aplicação Web, o servidor de DNS repassa essa informação "4" para o resolvedor de nomes do PC do usuário. Conseqüentemente o usuário é redirecionado "5" para o servidor Web que hospeda o site da

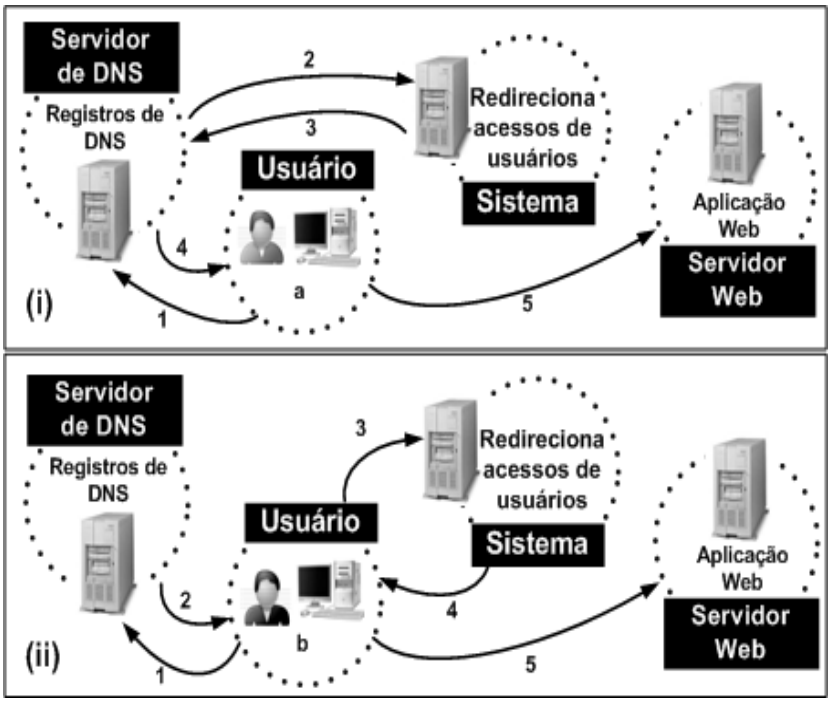

Figura 1: Métodos de distribuição de acesso por DNS (i) e sessão HTTP (ii).

aplicação. Esse procedimento é amplamente abordado por literatura técnica (HARVEY; SPEIER, 1997).

Sistemas tradicionais de tolerância a falhas baseados no redirecionamento por DNS carecem de funções capazes de interagir eficazmente com servidores de DNS já existentes na infra-estrutura. Os sistemas tradicionais de tolerância a falhas que utilizam essa técnica de redirecionamento herdam a mesma limitação de ter que resolver as consultas de DNS dos usuários na rede para as aplicações Web supervisionadas por esses sistemas (CITRIX, 2010), (CSS, 2010), (GSS, 2010).

No exemplo da Figura 1(ii) o sistema de tolerância a falhas usa a técnica de redirecionamento de sessão HTTP, nessa técnica o sistema de tolerância a falha responde pela as requisições HTTP para as aplicações Web supervisionadas pelo sistema. Nesse exemplo o resolvedor de nomes do PC do usuário “b” também consulta o servidor de DNS “1” para obter o endereço IP que permite o acesso a aplicação Web “2”. Nos registros do servidor de DNS o endereço do site corresponde ao IP do sistema de tolerância a falha “3”. Quando o sistema em questão recebe uma requisição HTTP para o site, o usuário é redirecionado pelo sistema “4” para acessar o servidor Web disponível “5”.

Nas duas técnicas de redirecionamento de acesso descritas anteriormente, os sistemas de tolerância a falhas estão aptos a detectar e corrigir falhas somente em componentes da infra-estrutura que estão sob seu domínio de supervisão. Os sistemas tradicionais de tolerância a falhas disponíveis no mercado não possuem a habilidade de detectar e corrigir registros incoerentes no servidor de DNS. As infra-estruturas de rede que usam esses sistemas de tolerância a falhas estão suscetíveis a existência de registros em servidores de DNS, ou que redirecionam usuários para acessar endereços de servidores inexistentes, ou que tornem completamente indisponível o acesso ao site da aplicação.

Os sistemas tradicionais de tolerância a falhas estabelecem uma dependência para todas as aplicações Web que são supervisionadas por esses sistemas. Esse vínculo está na condição de que cada novo acesso de usuário na rede para essas aplicações Web seja antes encaminhado para o sistema de tolerância a falhas. Somente após o sistema interceptar as requisições para os sites, os usuários são redirecionados para os servidores Web que atendem ao site da aplicação. Em ambos os métodos de redirecionamento, considerando a hipótese do sistema de tolerância a falhas apresentar indisponibilidade na operação, os usuários das aplicações Web não podem acessar o site dessas aplicações, enquanto o sistema de tolerância a falhas não estiver disponível novamente. A conseqüência dessa dependência sujeita os usuários de aplicações Web a não obterem acesso aos sites das aplicações, na condição do sistema de tolerância a falhas apresentar não conformidade.

2. OBJETIVOS DO TRABALHO

Propor um método preemptivo para sistema recuperar falhas em sites de aplicações Web com base no redirecionamento por DNS, sem a necessidade desse sistema prover a tradução de nomes para requisições de usuários, com a capacidade de identificar, corrigir e atualizar os registros na base de dados do servidor de DNS.

Apresentar os primeiros resultados obtidos nos testes realizados em protótipo que adota os principais módulos previstos no sistema apresentado neste trabalho.

3. ARQUITETURA DO SISTEMA PROPOSTO

A metodologia adotada neste trabalho sugere a interação do sistema de tolerância a falhas com os servidores de DNS existentes na infra-estrutura. A conseqüência dessa cooperação permite que os servidores de DNS sejam os únicos responsáveis pela tradução de

nomes de endereços na rede, enquanto o sistema de tolerância a falhas cumpre exclusivamente a função de reparar falhas, sem a necessidade de desempenhar uma tarefa que já é designada aos servidores de DNS na infra-estrutura de rede.

O método preemptivo utiliza funções que procuram por divergências nos registros do servidor de DNS comparando com as informações existentes na configuração do sistema de tolerância a falhas. Quando alguma informação referente ao site que está no servidor de DNS, mas não existe na configuração do sistema proposto, a informação que diverge da configuração é retirada dos registros do servidor de DNS. Com base nesse procedimento, somente as informações parametrizadas no sistema são válidas, enquanto modificações nos registros do DNS que não coincidem com a configuração do sistema são anuladas, ou seja, somente os servidores Web das aplicações Web pré-definidos na configuração do sistema proposto podem ser utilizados por usuários na rede. Essa metodologia, assim como nos demais sistemas clássicos de tolerância a falhas, é considerado um exemplo de gerenciamento centralizado (XU; ZHANG; ZHOU; KUANG, 2007), (ZHOU; ZHANG; XIE, 2006), (GOIS; BORELLI, 2010), (FERRAIOLO; CHANDRAMOULI; AHN; GAVRILA, 2003), pois todos os servidores Web capazes de prover serviços para essas aplicações devem ser previamente informados na configuração de cada sistema. A técnica de atualização dinâmica de DNS (RFC 2136), (RFC 2137), (RFC 2535), (RFC 3007) é usada pelo sistema proposto neste trabalho para adicionar ou remover registros no servidor de DNS de forma rápida e segura.

Em grandes ambientes de rede corporativos existem profissionais que freqüentemente modificam componentes na infra-estrutura para melhorar acesso, segurança e disponibilidade das informações na rede. A Figura 2 apresenta a condição na qual o analista de rede “c” inclui outro registro no servidor de DNS referente a aplicação Web, no entanto, por falta de atenção, o analista de rede “c” inclui registros de endereços IP para servidores Web inexistentes na rede “iv” para o “Site 1”. Nesse exemplo, representado na Figura 2, o sistema proposto faz a detecção e a correção da falha do servidor Web que atende o “Site 1”. Além de detectar e corrigir falhas dos servidores Web, o sistema proposto também identifica os registros

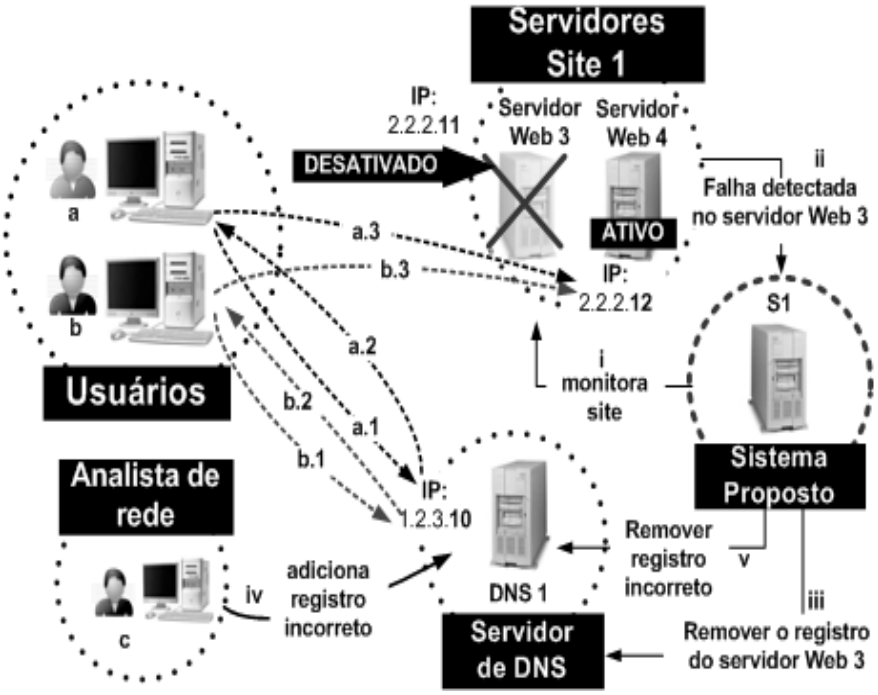

Figura 2: Arquitetura do protótipo com base no método preemptivo.

adicionados acidentalmente pelo analista de rede "c" ("iv – adiciona registro incorreto") e os remove da base de dados do servidor de DNS (" v – remover registro incorreto"). Na mesma figura, a tradução de nomes solicitada pelo resolvedor de nomes do usuário é sempre atendida pelo servidor de DNS existente na infra-estrutura. Quando o usuário "a" solicita acesso ao site no seu navegador Web para algum site supervisionado pelo sistema proposto, o resolvedor de nomes do PC do usuário consulta o servidor de DNS para obter qual é o endereço do site solicitado "a.1". O servidor de DNS "DNS 1" responde a requisição "a.2" informando ao resolvedor de nomes do usuário que o endereço IP a ser utilizado no acesso corresponde ao do servidor Web 4 "a.3".

Após o acesso satisfatório do usuário "a" ao site, o analista de rede "c" inclui registros incorretos no servidor de DNS. No entanto, o método preemptivo identifica que os registros adicionados pelo analista de rede não constam na configuração do sistema proposto. Para evitar divergências entre as informações existentes no servidor de DNS com os dados existentes na configuração do sistema, o método preemptivo remove "v" a informação adicionada pelo analista de rede "c", pois essa informação não existe na configuração do sistema.

Ainda na Figura 2 o usuário "b" acessa o mesmo site solicitado anteriormente pelo usuário "a", o resolvedor de nomes do usuário "b" consulta o servidor de DNS para obter qual é o endereço IP do servidor Web que pode ser utilizado no acesso "b.1". O mesmo servidor de DNS traduz a requisição "b.2" informando para o resolvedor de nomes do usuário "b" que o endereço correspondente a aplicação é o IP do servidor Web 4 "b.3". O acesso do usuário "b" só é possível porque o método preemptivo remove o registro no servidor de DNS que não coincide com a configuração do sistema, caso contrário, o usuário "b" é redirecionado para algum dos endereços IP adicionados acidentalmente pelo analista de rede "c". Nesse exemplo o sistema proposto neste trabalho monitora o funcionamento do site 1 ("i – monitora site") em ambos os servidores parametrizados na configuração. A falha do "servidor Web 3" para o "Site 1" é perceptível para o sistema ("ii – falha detectada no servidor Web 3") que remove o registro no servidor de DNS referente ao endereço IP do servidor com falha.

Toda alteração na base de dados do servidor de DNS é realizada por funções que enviam comandos para o servidor de DNS remover o registro relacionado ao endereço IP do servidor que apresenta falha ("iii – remover o registro do servidor Web 3"). Além de monitorar os servidores Web, o sistema proposto neste trabalho também monitora os Registros de Recursos (R.R.) existentes no servidor de DNS relativos aos sites parametrizados na configuração do sistema. Após a conclusão da análise do sistema, o servidor de DNS tem seus registros sincronizados com a configuração do sistema, como conseqüência o servidor de DNS direciona os usuários somente para o servidor Web 4 (ATIVO). Entretanto, isso ocorre somente após o sistema proposto validar o funcionamento adequado do servidor Web 4 para acesso a aplicação referente ao "Site 1".

Assim que o site é configurado para funcionar no modelo preemptivo, uma função específica do sistema considera mandatório que todos os servidores referentes a esse site sejam definidos previamente na configuração, cuja definição mantém compatibilidade com o modelo convencional de gerenciamento centralizado utilizado pela maioria dos sistemas de mercado. Essa metodologia exige que todos os servidores Web sejam adicionados ou removidos somente a partir da configuração do sistema de tolerância a falhas.

A Figura 3 ilustra a lógica de como o método preemptivo detecta e corrige registros incoerentes no servidor de DNS. Uma rotina do sistema apresentado neste trabalho separa todos os sites que estão pré-definidos na configuração para detecção preemptiva. Essa função consulta todas as entradas no servidor de DNS que constam para o site supervisionado pelo

sistema proposto. Após isso, o sistema compara os servidores configurados para o site com todas as entradas encontradas no servidor de DNS que possuem referência ao site da aplicação. Na Figura 3, durante a comparação a rotina detecta a entrada no registro do servidor de DNS para o hospedeiro (host) "k" que não conta na configuração do site apresentado na Figura 3(a). Uma outra rotina do sistema marca a entrada "k" para ser removida do servidor de DNS, pois a considera inválida para o site, conforme mostrado na Figura 3(b). Ainda na mesma comparação, a rotina também localiza a entrada "j" ausente nos registros do servidor de DNS para o site supervisionado, mostrado na Figura 3(c). Antes do próximo ciclo de testes do sistema, a rotina invoca a função que usa a técnica de atualização dinâmica de DNS passando como parâmetro a remoção do registro "k" inválido, Figura 3(b), e a inclusão do registro "j", correspondente ao endereço IP do servidor Web que está ausente no servidor de DNS, Figura 3(d). A atualização dinâmica de DNS (DDNS) é um método usado por clientes de DNS para enviar requisições ao servidor de nomes de domínio (servidores DNS) a fim de adicionar ou remover registros de uma zona; essa capacidade facilita a manutenção do DNS em grandes ambientes (DESMOND; RICHARDS; ALLEN; LOWE-NORRIS, 2008).

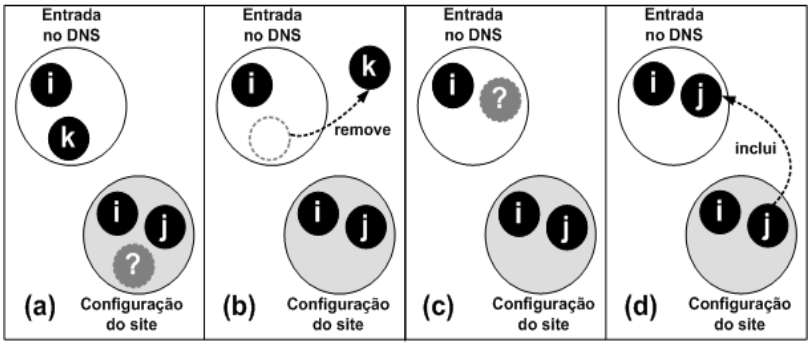

Figura 3: Lógica do método preemptivo.

Embora o sistema proposto neste trabalho não forneça o serviço de resolução de nomes, a técnica empregada para encaminhar as solicitações é o redirecionamento de acesso por DNS. Essa técnica também é utilizada por outros sistemas de recuperação de falhas desenvolvidos para o mesmo propósito (BIG-IP-GTM, 2010), (CDD, 2010), (LVS, 2010).

A técnica de atualização dinâmica de DNS é usada pelo sistema proposto neste trabalho para adicionar ou remover entradas no servidor de DNS. A metodologia usada pelo sistema proposto permite que os usuários das aplicações Web acessem o site dessas aplicações, independente do sistema estar disponível ou não na rede.

Para melhorar a disciplina de testes do sistema proposto para os sites supervisionados, todos os servidores Web parametrizados na configuração são testados simultaneamente. Essa atuação é feita com uma máquina multithread que processa instruções dentro de um laço finito, cujo ciclo ocorre a cada N período de tempo pré-definido no sistema. Essa máquina invoca threads específicas que validam ao mesmo tempo todos os servidores Web detectados na configuração. Os testes simultâneos são executados com base em threads POSIX (*Portable Operating System Interface for* UNIX) ou (pthreads) (PTHREAD-LIB, 2009). As threads POSIX são baseadas na especificação IEEE (*Institute of Electrical and Electronics Engineers*) POSIX 1003.1c, desenvolvida pelo IEEE para obter melhores resultados na programação com threads (BARNEY; LIVERMORE, 2009).

5. RESULTADOS

O protótipo apresentado neste trabalho foi testado para detectar e corrigir falhas que ocorrem em servidores Web predefinidos nos experimentos. Nesses testes o protótipo foi parametrizado com o tempo de 10 segundos, esse tempo é usado para determinar a freqüência em que o sistema valida a disponibilidade dos servidores Web e a consistência dos registros no servidor de DNS referentes aos sites supervisionados pelo sistema.

O sistema apresentado neste trabalho testa a disponibilidade dos servidores Web a cada ciclo de execução. Quando o tempo esgota, o sistema valida novamente todos os servidores detectados na configuração.

A Figura 4 mostra resumidamente os componentes usados nos experimentos. Para simular a infra-estrutura de rede normalmente usada em redes de intranet, são utilizados quatro microcomputadores com arquitetura Intel 32 bits (PC). Esses microcomputadores são conectados em modo Full Duplex a um switch Ethernet de 100 Mbps.

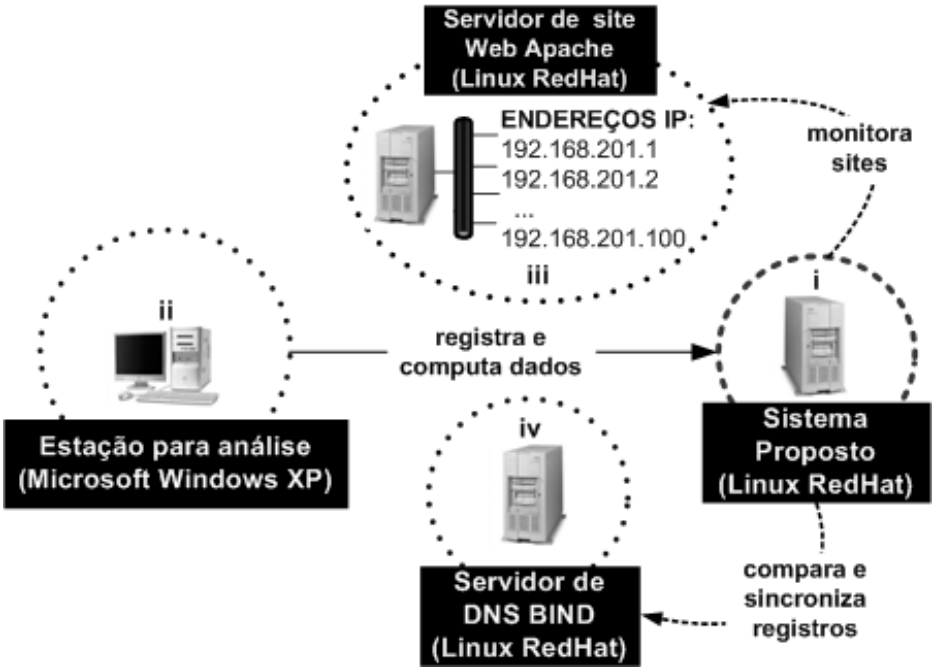

Figura 4: Componentes utilizados nos experimentos

Os microcomputadores utilizados nos testes possuem a configuração de 1.5 GHz de freqüência para a CPU (*Central Processing Unit*) e 512 MB de memória RAM (*Random Access Memory*).

No primeiro PC (“i”) do protótipo do sistema apresentado neste trabalho é instalado com o sistema operacional Linux Fedora 9 (FEDORA, 2010). Por meio desse microcomputador o protótipo monitora e recupera as aplicações Web previamente parametrizadas, assim que os servidores Web detectados na configuração apresentam falha na rede.

O segundo PC (“ii”) é usado para analisar os eventos do protótipo do sistema apresentado neste trabalho e processar o momento das falhas, que ocorrem simultaneamente em alguns servidores Web na rede; nesse PC são computados os tempos que o protótipo utiliza para recuperar os recursos afetados pelas falhas. Os programas utilizados nesse PC são: o sistema operacional Microsoft Windows XP™ e o aplicativo Microsoft Excel™ 2003.

O terceiro microcomputador (“iii”) simula todos os servidores pré-definidos em cada site de aplicação Web. Na configuração desse PC são definidos dez sites de aplicações Web distintos associados com dez endereços IP seqüenciais de servidores em cada site. Esses endereços de rede não se repetem e somam no total cem servidores Web. Esses endereços são configurados para aceitar requisições de acesso Web para os sites pré-definidos na configuração do protótipo. Os endereços IP dos servidores Web são associados às interfaces virtuais (PILLAY, 2009), que são definidas e carregadas no sistema operacional desse PC. Esse microcomputador (“iii”) utiliza o sistema operacional Linux Fedora 9, com o programa Apache (BOWEN, 2000), usado para suportar os sites Web das aplicações e aceitar as requisições HTTP.

O quarto PC (“iv”) executa o sistema operacional Linux Red Hat ES 3.0 (LRH, 2010), com o serviço de DNS BIND (MANSFIELD, 2003) ativado. Esse servidor é usado para resolver os nomes dos sites definidos nos experimentos com o protótipo do sistema apresentado neste trabalho. O sistema apresentado neste trabalho não exige que a tradução de nomes para os endereços IP referentes aos sites das aplicações Web seja fornecida pelo protótipo, desse modo é possível utilizar o servidor de DNS já existente na infra-estrutura. Na configuração do servidor de DNS existe uma chave embaralhada com o algoritmo de hash HMAC-MD5 (RFC 2104). Essa chave é usada pelo servidor de DNS para autorizar as alterações solicitadas pelo sistema proposto, tanto de inclusão como de exclusão de entradas nos registros de DNS.

A análise dos experimentos considera o tempo que protótipo utiliza para detectar e recuperar os servidores Web que apresentam falhas. Durante os experimentos são processados os momentos em que as falhas ocorrem e o tempo gasto pelo protótipo do sistema apresentado neste trabalho para recuperar as falhas, também chamado de MTTR (*Medium Time to Recovery* – Tempo Médio de Recuperação). Depois de registrado o momento da falha e o tempo de recuperação, o protótipo do sistema é reiniciado e o servidor anterior com falha é reativado na infra-estrutura. Esse servidor retorna ao seu funcionamento normal (sem falha) para prosseguir com os demais passos dos experimentos. Assim que o servidor Web é religado na infra-estrutura, o protótipo é reiniciado para reconhecer os servidores como funcionais e sem registros de falhas anteriores. Após isso, intencionalmente cada servidor Web é desligado simultaneamente com o próximo servidor da seqüência. Do mesmo modo, o momento da falha e o tempo para recuperar as aplicações associadas com os servidores com falha são computados em cada passo dos experimentos. Esse procedimento de interromper o sistema, religar o servidor que retorna falha na rede e simular falhas simultâneas de cada servidor junto com o próximo servidor da seqüência, ocorre até que todos os servidores definidos na configuração do protótipo sejam validados. A Figura 5 apresenta o gráfico gerado com os dados coletados a partir do experimento com protótipo do sistema apresentado neste trabalho.

No eixo vertical (y) do gráfico é esboçado o MTTR, ou seja, o tempo que protótipo do sistema leva para corrigir as falhas simuladas. O eixo horizontal (x) representa o número de servidores concomitantemente com falha que devem ser recuperados na infra-estrutura de rede. Esses servidores, quando funcionam adequadamente, fornecem acesso para as aplicações Web parametrizadas no protótipo do sistema. Ainda no gráfico da Figura 5, o sistema proposto neste trabalho leva pouco tempo para detectar e recuperar falhas dos servidores monitorados. O sistema proposto leva pouco menos que nove segundos para detectar e recuperar a falha de apenas um servidor; com a duração de até 11 segundos para detectar, revalidar e recuperar 100 servidores simultaneamente com falha.

A interação do método preemptivo com atualização dinâmica de DNS, associado com testes concorrentes apresenta resultados favoráveis.

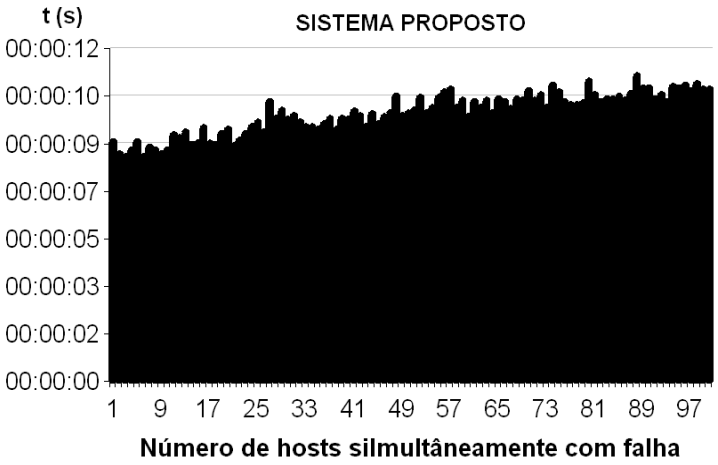

Figura 5: Informações coletadas a partir dos experimentos com o protótipo

5. CONCLUSÕES

Resultados satisfatórios obtidos nos experimentos com o protótipo do sistema apresentado neste trabalho sugerem que o método preemptivo é adequado para uso em sistemas de tolerância a falhas de aplicações Web. O protótipo atende as expectativas, porque associa o uso de funções que tomam decisões sobre corrigir registros incoerentes no servidor de DNS, além de possuir a capacidade suplementar de eliminar a dependência que existe entre os sistemas tradicionais de tolerância a falhas e os recursos supervisionados por esses sistemas.

6. REFERÊNCIAS

ANSWERS. Guide: Intranet. Disponível em: <http://www.answers.com/topic/intranet> Acesso em: 18 de Dezembro 2011.

BARNEY, B.; LIVERMORE, L. POSIX Threads Programming. National Laboratory. Disponível em: <https://computing.llnl.gov/tutorials/pthreads/> Acesso em: 05 de Julho de 2009.

BIG-IP-GTM. Global Traffic Manager. Disponível em: <http://www.f5.com/products/big-ip/global-traf fic-manager.html> Acesso em: 29 de Outubro 2010.

BIG-IP-LTM. Local Traffic Manager. Disponível em: <http://www.f5.com/products/big-ip/local-traffic-manager.html> Acesso em: 17 de Outubro de 2010.

BOWEN, R. Apache Guide: Setting Up Virtual Hosts, July 17, 2000. Disponível em: <http://www.server watch.com/tutorials/article.php/1127571/Apache-Guide-Setting-Up-Virtual-hosts.htm> Acesso em: 02 de Março 2009.

CDD. Cisco Distributed Director Configuration Example Overview. Disponível em: <http://www. cisco.com/en/US/products/hw/contnetw/ps813/products_tech_note09186a00801fa9dd.shtml> Acesso em: 10 de Agosto de 2010.

CITRIX. NetScaler. Disponível em: <http://www.citrix.com.br/products/netscaler.php> Acesso em: 26 de Maio de 2010.

CSS. Cisco 11500 Series Content Services Switches. Disponível em: <http://www.cisco. com/en/US/products/ hw/contnetw/ps792/index.html> Acesso em: 14 de Abril de 2010.

DESMOND, B.; RICHARDS, J.; ALLEN, R.; LOWE-NORRIS, A. G. Active Directory. 4th Edition, O'Reilly Media, November 26, 2008, pp118-120.

FEDORA. Linux Fedora 9. Disponível em: <http://fedoraproject.org> Acesso em: 10 de Outubro de 2010.

FERRAIOLO, D.; CHANDRAMOULI, R.; AHN, G.; GAVRILA, S. The role control center: features and case studies, SACMAT '03: Proceedings of the eighth ACM symposium on Access control models and technologies, June 2003.

GOIS, L.; BORELLI, W. Optimization of Procedures for Discovery and Information of Idle Resources in Distributed Systems. SIGOPS Operating Systems Review, Volume 44 Issue 1, March 2010.

GSS. Cisco ACE 4400 Series Global Site Selector Appliances. Disponível em: <http://www.cisco.com/ en/US/products/hw/contnetw/ps4162/index.html> Acesso em: 11 de Março de 2010.

HARVEY, M. G.; SPEIER, C. Intranets and Organizational Learning. SIGCPR '97 Proceedings of the 1997 ACM SIGCPR conference on Computer personnel research.

IIS. Servidor Web. Disponível em: <http://technet.microsoft.com/pt-br/library/cc753433(WS.10). aspx> Acesso em: 17 de Novembro de 2009.

LRH. Linux Red Hat ES 3.0. Disponível em: <http://www.br.redhat.com/products/rhel/details/enterprise linux3> Acesso em: 22 de novembro de 2010.

LVS. Linux Virtual Server. Disponível em: <http://www.linuxvirtualserver.org> Acesso em 12 de Setembro de 2010.

MANSFIELD, N. Practical TCP/IP: Designing, using, and troubleshooting TCP/IP networks on Linux and Windows, Pearson Education, 1st Edition, 2003, pp270-273.

PILLAY, H. Setting up IP Aliasing on A Linux Machine Mini-HOWTO. Disponível em: <http://www.faqs. org/docs/Linux-mini/IP-Alias.html> Acesso em: 15 de Junho de 2009.

PTHREAD-LIB. POSIX thread (pthread) libraries. Disponível em: <http://www.yolinux.com/ TUTORIALS/ LinuxTutorialPosixThreads.html#BASICS> Acesso em: 11 de Agosto de 2009.

RFC 2104, 1997, HMAC: Keyed-Hashing for Message Authentication, Disponível em: <http://www.ietf.org/ rfc/rfc2104.txt> Acesso em: 12 de Outubro de 2009.

RFC 2136, 1997, Dynamic Updates in the Domain Name System (DNS UPDATE). Disponível em: http://www.ietf.org/rfc/rfc2136.txt. Acesso em: 06 de Setembro de 2009.

RFC 2137, 1997, Secure Dynamic Update. Disponível em: <http://www.ietf.org/rfc/rfc2137.txt> Acesso em: 17 de Agosto de 2009.

RFC 2535, 1999, Domain Name System Security Extensions. Disponível em: <http://www.ietf.org/rfc/ rfc3007 .txt> Acesso em: 22 de Junho de 2009.

RFC 3007, 2000, Secure Domain Name System (DNS) Dynamic Update. Disponível em: <http://www.ietf.org/ rfc/rfc3007.txt> Acesso em: 11 de Julho de 2009.

XU, G.; ZHANG, Y.; ZHOU, Y.; KUANG, W. An Architecture for on-demand Desktop Computing in a Network Environment. 2007, International Conference on Convergence Information Technology.

ZHOU, Y.; ZHANG, Y.; XIE, Y. Virtual disk based centralized management for enterprise networks. Proceedings of the 2006 SIGCOMM workshop on Internet network management.