SAPOTI: Servidores de Aplicações cOnfiáveis Tcp/Ip

Egon Hilgenstieler, Roverli Pereira Ziwich, Emerson Fabiano Fontana Carara, Luis Carlos Erpen De Bona, Elias Procópio Duarte Jr.

Universidade Federal do Paraná, Departamento de Informática
Caixa Postal 19081 - 81531-990, Curitiba PR Brasil

{egon,roverli,emerson,bona,elias}@inf.ufpr.br

Resumo. *Este trabalho apresenta o SAPOTI (*Servidores de APlicações cOnfiáveis Tcp/Ip*), uma ferramenta distribuída que garante a alta disponibilidade de servidores de aplicações TCP/IP aplicada em particular para a implementação de um servidor Web tolerante a falhas. A ferramenta é baseada no protocolo SNMP e é executada em um conjunto de máquinas que são monitoradas por uma ferramenta de gerência confiável e distribuída baseada no algoritmo de diagnóstico hierárquico Hi-ADSD with Timestamps. Através da identificação de falhas em um servidor Web, o serviço é recuperado na máquina sem-falha de maior prioridade da rede, ficando disponível mesmo quando apenas uma máquina está sem-falha. Experimentos são descritos nos quais em configurações de rede com máquinas onde ocorreram até 210 falhas distribuídas entre as máquinas, a disponibilidade foi de no nínimo 97,3%. Em outra configuração de rede, onde ocorreram 27 falhas, a disponibilidade foi de 99,5% no tempo do experimento.*

Abstract. *This work presents SAPOTI (*Servidores de APlicações cOnfiáveis Tcp/Ip - *Dependable TCP/IP Application Servers), a distributed tool that guarantees the availability of TCP/IP application servers applied particularly to the implementation of a fault-tolerant Web server. The tool is based on the SNMP framework and is executed in a set of machines that are monitored by a dependable distributed management tool based on the hierarchical diagnosis algorithm Hi-ADSD with Timestamps. After the Web server becomes faulty and this event is diagnosed, the service is recovered by the fault-free machine with the highest priority, so that the service is available when only one machine is fault-free. Experiments are described in which with network configurations with machines where occurred until 210 faults distributed among the machines, the availability was at least 97,3%. In another network configuration, where 27 faults occurred, the availability was 99,5% in the experiment time.*

1. Introdução

As organizações dependem cada vez mais do bom funcionamento de suas redes. Em muitos casos é necessário garantir a alta disponibilidade do sistema. Os servidores de aplicações TCP/IP (*Transfer Control Protocol/Internet Protocol*) [1] estão entre os componentes críticos para o bom funcionamento da rede de uma organização. Para alcançar este objetivo é necessário um sistema de gerenciamento de redes para monitoração e controle da rede.

Este trabalho apresenta o SAPOTI (*Servidores de APlicações cOnfiáveis Tcp/Ip*), uma ferramenta para implementação de servidores de aplicações confiáveis TCP/IP, em específico de um servidor Web. Estes servidores confiáveis são implementados sobre a ferramenta de gerência confiável e distribuída baseada em diagnóstico hierárquico [2] que implementa o algoritmo *Hi-ADSD with Timestamps* (*Hierarchical Distributed System-Level*

Diagnosis with Timestamps) [3], um algoritmo de diagnóstico em nível de sistema. Desta maneira é possível identificar falhas em um servidor Web e recuperar o serviço iniciando-o em outra máquina sem-falha que está sendo monitorada pela ferramenta de gerência. O servidor Web é confiável no sentido em que mesmo quando apenas uma máquina está sem-falha o serviço continua sendo disponibilizado.

O objetivo do diagnóstico em nível de sistema é identificar quais unidades do sistema estão falhas e quais estão sem-falhas [4]. Quando o diagnóstico é distribuído [4] cada nodo tem a habilidade de realizar o diagnóstico de todo o sistema. Quando o diagnóstico é adaptativo, os testes que cada nodo realiza são baseados em rodadas e cada rodada é realizada com base nos resultados da rodada anterior. Quando o diagnóstico é hierárquico, os nodos são divididos em clusters e a cada rodada de testes o tamanho destes clusters aumenta. O algoritmo *Hi-ADSD with Timestamps* [3] é um exemplo de um algoritmo hierárquico, distribuído e adaptativo de diagnóstico em nível de sistema.

A ferramenta de gerência confiável é implementada utilizando o protocolo SNMP (*Simple Network Management Protocol*) [5] que é o padrão da Internet para gerência de redes. Um sistema de gerência SNMP é composto de entidades de gerência que se comunicam utilizando o protocolo de gerência. A arquitetura também define uma Base de Informações de Gerência ou MIB (*Managent Information Base*) [5] como uma coleção de objetos de gerência correlatos que são mantidos com o objetivo de permitir às aplicações de gerência monitorar e controlar os nodos gerenciados.

A ferramenta de gerência confiável e distribuída também permite especificar e executar de maneira confiável testes concebidos especificamente para entidades testadas. A ferramenta SAPOTI configura e utiliza em cada máquina um teste para verificar a disponibilidade do servidor Web. Através do resultado destes testes e de informações de diagnóstico disponibilizadas pela ferramenta de gerência, é possível identificar falhas em um servidor Web e recuperar o serviço em outra máquina sem-falha de maior prioridade da rede. Para que o atendimento às requisições destinadas ao servidor Web seja transparente, um endereço IP virtual único é utilizado associado ao servidor Web. Desta forma, quando uma máquina está disponibilizando o servidor Web, este endereço IP virtual está configurado na sua inteface de rede.

Uma interface Web foi implementada para visualizar as informações de diagnóstico geradas pela ferramenta de gerência. A partir de qualquer máquina da rede, a interface permite a visualização dos resultados de todos os testes da MIB utilizada pela ferramenta de gerência, inclusive a disponibilidade do servidor Web. Experimentos foram realizados e são descritos. Como resultado foi constatado que em configurações de rede com seis máquinas onde ocorreram até 210 falhas distribuídas entre todas as máquinas, a disponibilidade do servidor Web foi de no mínimo 97,3%. Em outra configuração de rede com cinco máquinas, onde algumas máquinas juntas falharam 27 vezes, a disponibilidade do servidor Web foi de 99,5%.

O restante deste trabalho está organizado como segue. A seção 2 apresenta uma visão geral da ferramenta de gerência confiável e distribuída baseada em diagnóstico hierárquico. A seção 3 apresenta a ferramenta SAPOTI. Resultados experimentais são descritos na seção 4 e a seção 5 conclui o trabalho.

2. Sistema de Gerência Confiável e Distribuída

A ferramenta SAPOTI obtém informações de diagnóstico de uma ferramenta de gerência confiável e distribuída baseada em diagnóstico hierárquico [2]. Esta ferramenta implementa o algoritmo *Hi-ADSD with Timestamps* [3] que é um algoritmo distribuído adaptativo e

hierárquico em nível de sistema. Um algoritmo de diagnóstico distribuído em nível de sistema permite que os nodos sem-falha determinem o estado de todos os outros nodos do sistema [3,4]. Um algoritmo de diagnóstico é chamado de adaptativo se os nodos determinam os próximos testes a serem executados com base nos resultados dos testes anteriores. Quando o diagnóstico é hierárquico, os nodos são divididos em clusters e a cada rodada de testes o tamanho destes clusters aumenta. Inicialmente é testado um cluster de tamanho *1*. O tamanho do cluster é dobrado a cada rodada até chegar a *n*/2, sendo *n* o número de nodos do sistema. A cada rodada, cada nodo realiza o teste de um cluster inteiro. Assume-se que os nodos sem-falha são capazes de executar e informar os testes de maneira confiável e que existe um enlace entre cada par de nodos.

A ferramenta de gerência confiável é implementada utilizando o protocolo SNMP. As entidades SNMP são tradicionalmente chamadas de gerentes e agentes. Os agentes disponibilizam informações de gerência através de uma MIB. Os gerentes são coleções de aplicações em nível de usuário, que podem ser voltadas para tarefas como avaliação de desempenho, diagnóstico de falhas, entre outras aplicações.

O sistema de gerência consiste em um conjunto de agentes, onde cada agente, também chamado de testador, mantém a Test-MIB, que contém informações sobre os testes executados pelo agente, assim como o estado de outros testadores e componentes do sistema. Os testadores são identificados pela ferramenta de gerência através de identificadores sequenciais. A Test-MIB permite a especificação de procedimentos de testes secundários concebidos especificamente para cada entidade testada. O sistema assume que um procedimento de teste permite ao testador sem-falhas determinar corretamente o estado da entidade testada.

Os procedimentos de testes podem ser configurados para serviços ou dispositivos de uma rede. Um serviço é qualquer processo executando em um testador, um dispositivo é qualquer recurso acessível da rede. Quando um teste secundário é do tipo dispositivo e o seu testador falha, o algoritmo determina um outro testador para executar este teste. Se o teste secundário for do tipo serviço o algoritmo não atribui outro testador para assumir o teste. Considera-se que um teste do tipo serviço somente pode ser realizado por um determinado testador. Neste caso o estado do serviço testado é considerado como desconhecido.

De acordo com o algoritmo, os testes são executados em intervalos de testes. A Test-MIB permite a configuração deste intervalo para cada teste. Cada MIB mantém informações sobre o número de intervalos de testes e o número de testes executados pelo testador. A MIB também mantém estatísticas que podem ser obtidas a partir da monitoração distribuída do sistema, tais como o tempo médio que um recurso monitorado permanece sem-falha e o tempo médio necessário para um recurso monitorado ser reparado.

3. Arquitetura da Ferramenta SAPOTI

A ferramenta SAPOTI pode ser utilizada em uma rede TCP/IP com máquinas que utilizam o sistema operacional GNU/Linux [6] para garantir a alta disponibilidade de servidores de aplicações TCP/IP, em particular para um servidor Web. O servidor Web utilizado foi o Apache [7]. O servidor Web deve estar instalado em um grupo de máquinas da rede que podem disponibilizar o serviço confiável.

A implementação da ferramenta SAPOTI foi feita na linguagem *bash script* [8], e atua de forma distribuída em uma rede de computadores monitorados pela ferramenta de gerência confiável e distribuída que implementa o algoritmo *Hi-ADSD with Timestamps*. A ferramenta SAPOTI utiliza a identificação sequencial atribuída às máquinas pela ferramenta

de gerência como ordem sequencial de prioridade para disponibilizar o servidor Web, sendo que o identificador 1 é o de maior prioridade.

A ferramenta SAPOTI configura em cada máquina um teste secundário na ferramenta de gerência para verificar a disponibilidade do servidor Web nesta máquina. Através do resultado deste teste e de informações de diagnóstico da rede sobre as máquinas obtidas da Test-MIB local da ferramenta de gerência, a ferramenta SAPOTI verifica qual a máquina de maior prioridade está disponibilizando o servidor Web.

Como toda máquina executa a ferramenta SAPOTI e verifica qual a máquina de maior prioridade está disponibilizando o servidor Web, se uma máquina *a* identificar que não existe nenhuma máquina com maior prioridade que está disponibilizando o servidor Web, esta máquina *a* passa a disponibiliza-lo, se ainda não o estiver fazendo. Da mesma forma, se uma máquina *a* que disponibiliza o servidor Web identificar que existe uma outra máquina com maior prioridade que também o está disponibilizando, esta máquina *a* deixa de disponibilizar o servidor Web. Todas estas informações são obtidas única e exclusivamente através da Test-MIB da máquina local. Como a Test-MIB é local, a ferramenta SAPOTI não executa nenhuma requisição pela rede. Assim, como cada máquina descobre independentemente se deve disponibilizar ou não o servidor Web, o algoritmo da ferramenta SAPOTI disponibiliza o servidor Web na máquina sem-falha de maior prioridade da rede.

Para que o atendimento às requisições destinadas ao servidor Web seja transparente, é utilizado um endereço IP virtual único que acompanha o servidor Web. Então para a disponibilização do servidor Web em uma máquina, é iniciado o Apache e a máquina recebe este endereço IP virtual além de seu próprio endereço IP. Um segundo endereço IP, no caso o endereço IP virtual, pode ser configurado em uma máquina através da técnica de IP *aliasing* [9] do sistema operacional Linux.

A figura 1 mostra um grupo de quatro máquinas rodando a ferramenta SAPOTI para implementar um servidor Web tolerante a falhas. Na figura as máquinas com identificadores 1 e 2 estão falhas. Como a máquina com identificador 3 é a máquina sem-falha de maior prioridade, ela assume a disponibilização do servidor Web e recebe o IP virtual associado.

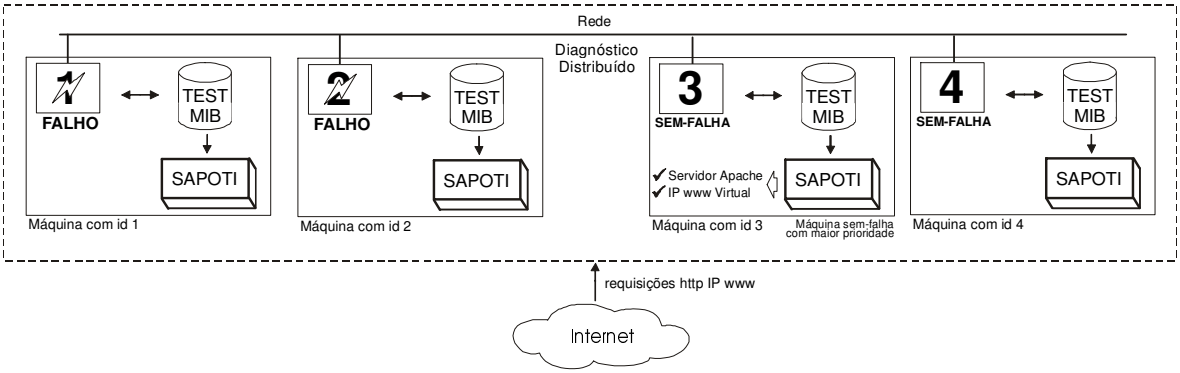

Figura 1: Um grupo de quatro máquinas rodando a ferramenta SAPOTI.

Existe a necessidade de que as máquinas mantenham atualizado o conjunto de arquivos acessados pelo servidor Web. A atualização deste conteúdo foi feita através do aplicativo RSYNC [10] que sincroniza entre todas as máquinas da rede qualquer alteração neste conteúdo. O aplicativo RSYNC permite que somente as diferenças entre os arquivos sejam atualizadas e transferidas pela rede.

Abaixo o algoritmo em alto nível da ferramenta SAPOTI é mostrado em fonte destacada.

```
meu_id := identificador da máquina local;
repetir para sempre
    servidor_id := id da máquina de menor identificador que disponibiliza o servidor Web;
    se (servidor_id > meu_id) ou (servidor_id é nulo)
        se não estou com a interface do IP Virtual
        então inicializa a interface do IP Virtual;
        se estou com o servidor Web fora ar
        então inicializa a disponibilização do servidor Web;
    senão
        se estou com a interface do IP Virtual
        então retira a interface do IP Virtual;
        se estou com o servidor Web no ar
        então retira o servidor Web do ar;
    fim_se;
    aguardar 10 segundos;
fim_repetir;
```

3.1 Interface e Funcionalidade

Uma interface Web foi implementada em PHP *(PHP: Hypertext Preprocessor)* [11] e *bash script* para visualizar as informações de diagnóstico da rede monitorada pela ferramenta de gerência. A figura 2 mostra um exemplo desta interface. Como se trata de uma interface Web, qualquer máquina pode ter a interface instalada, para tal, a máquina deve ter um servidor Web instalado. A cada acesso a esta página, as informações são extraídas da Test-MIB de um dos testadores do sistema de gerência.

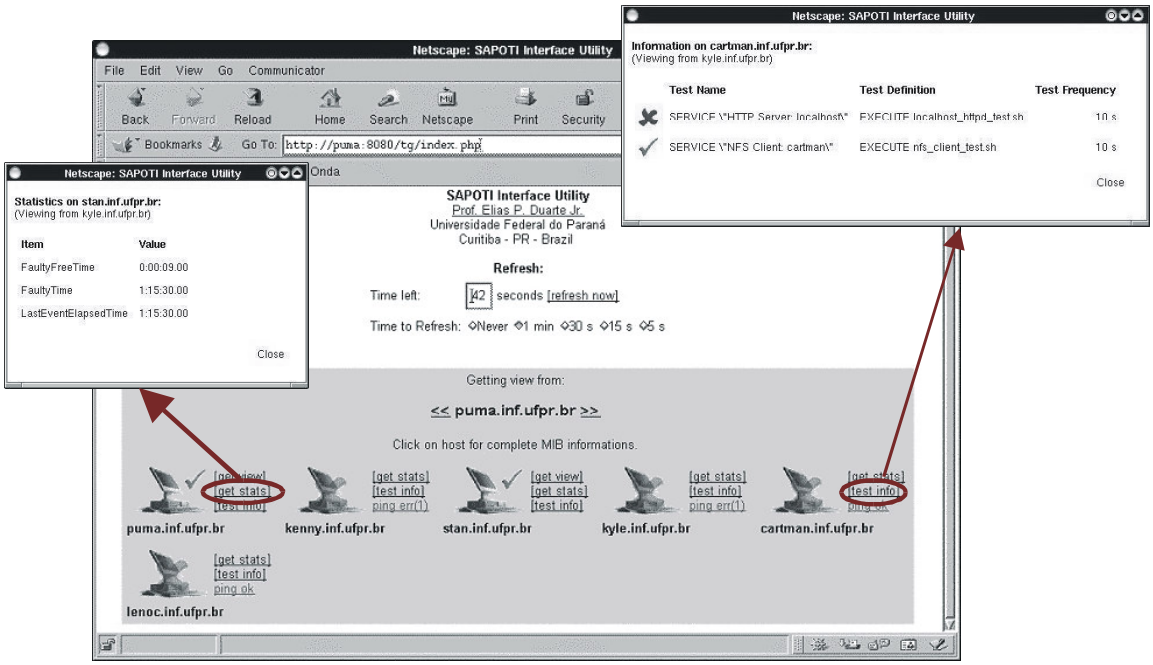

Figura 2: Interface da ferramenta.

Para acessar a interface é necessário conhecer o nome ou endereço IP de pelo menos uma máquina pertencente ao grupo de máquinas monitoradas. Escolhida a máquina, a interface exibe então as informações de diagnóstico. Além disso, a MIB da ferramenta de gerência fornece também o resultado da execução do comando *ping* caso a máquina esteja sem falha, o resultado dos testes configuráveis e estatísticas sobre o estado das máquinas.

4. Resultados Experimentais

São descritos dois experimentos que visam medir a latência de recuperação e a disponibilidade do servidor Web. No primeiro experimento foram usadas seis máquinas e foram realizadas medições em um intervalo de 12 horas e 23 minutos. A tabela 1 mostra as máquinas usadas no experimento, seu identificador e como foi configurada a situação de

falhas. A probabilidade de uma máquina estar sem-falha em um dado momento variou entre 33% e 86%. Neste experimento, ocasionalmente todas as máquinas podem estar falhas ocasionando a indisponibilidade do servidor Web.

No segundo experimento foram usadas cinco máquinas e foram realizadas medições em um intervalo de 12 horas e 40 minutos. A tabela 1 também mostra as máquinas usadas no experimento, seu identificador e como foi configurada a situação de falhas. Neste experimento sempre existe uma máquina sem-falha e não existe mais de uma máquina ficando falha ou se recuperando no mesmo instante.

Primeiro Experimento		
Máquina	***ID***	***Configuração das Falhas***
Puma	1	5 minutos no ar / 10 minutos fora do ar
Kenny	2	8 minutos no ar / 9 minutos fora do ar
Stan	3	13 minutos no ar / 7 minutos fora do ar
Kyle	4	17 minutos no ar / 6 minutos fora do ar
Cartman	5	21 minutos no ar / 5 minutos fora do ar
Lenoc	6	25 minutos no ar / 4 minutos fora do ar

Segundo Experimento		
Máquina	***ID***	***Configuração das Falhas***
Puma	1	20 minutos no ar / 40 minutos fora do ar
Kenny	2	40 minutos no ar / 20 minutos fora do ar
Stan	3	sempre no ar
Kyle	4	sempre no ar
Cartman	5	sempre no ar

Tabela 1: Nome, ID e configuração da situação de falha das máquinas no dois experimentos.

Nos dois experimentos, as falhas das máquinas foram simuladas através do encerramento do processo do agente SNMP. As máquinas utilizadas estavam em uma rede Ethernet 100 Mbps com NFS. Todas as máquinas rodavam o sistema operacional Linux Debian [12]. Em todas as máquinas estava instalada a versão 4.2.1 do pacote NET-SNMP [5]. Qualquer uma das máquinas estava apta a disponibilizar o servidor Web. O servidor Web usado no experimento foi o Apache. Durante a monitoração, os dados eram coletados de 5 em 5 segundos.

A tabela 2 mostra o número de vezes que cada máquina utilizada nos experimentos falhou e a porcentagem do tempo de cada experimento em que as máquinas ficaram falhas. No primeiro experimento ocorreram 210 falhas distribuídas entre as máquinas e no segundo experimento ocorreram 27 falhas.

Primeiro Experimento		
Máquina	***Falhas***	***Tempo de Falha em %***
Puma	50	66,4
Kenny	44	46,6
Stan	37	35,2
Kyle	32	26,3
Cartman	29	19,2
Lenoc	26	13,7

Segundo Experimento		
Máquina	***Falhas***	***Tempo de Falha em %***
Puma	14	68,5
Kenny	13	51,4
Stan	0	0
Kyle	0	0
Cartman	0	0

Tabela 2: Número de falhas de cada máquina e a porcentagem do tempo em que as máquinas ficaram falhas nos dois experimentos.

Durante todo o primeiro experimento, o servidor Web ficou indisponível e se recuperou 72 vezes. A figura 3 mostra a latência de recuperação do servidor Web neste experimento. O melhor caso de recuperação do servidor Web após uma falha foi de 6 segundos. O pior caso de recuperação deu-se em 53 segundos. A média ponderada do tempo de recuperação do servidor Web foi de 16,4 segundos. A figura 4 mostra a latência acumulada do tempo de recuperação do servidor Web neste experimento. Durante o experimento, 54% das vezes que o servidor Web caiu, o tempo de recuperação foi menor que 10 segundos e 90% das vezes que o servidor Web caiu o tempo de recuperação foi menor que 30 segundos.

Durante todo o segundo experimento o servidor Web ficou indisponível e se recuperou 41 vezes. A figura 3 também mostra a latência de recuperação do servidor Web neste experimento. O melhor caso de recuperação do servidor Web após uma falha foi de 7

segundos. O pior caso de recuperação deu-se em 29 segundos. A média ponderada do tempo de recuperação do servidor Web foi de 14,0 segundos. A figura 4 também mostra a latência acumulada do tempo de recuperação do servidor Web neste segundo experimento. Durante o experimento, 64% das vezes que o servidor Web caiu o tempo de recuperação foi menor que 10 segundos e 100% das vezes que o servidor Web caiu o tempo de recuperação foi menor que 30 segundos.

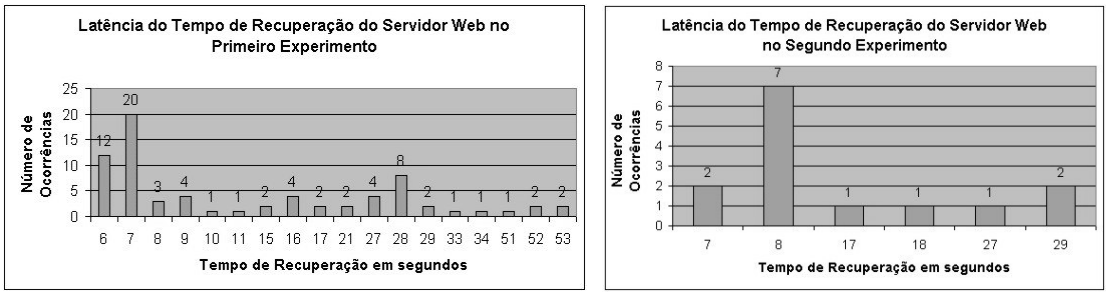

Figura 3: Latência do tempo de recuperação do servidor Web nos experimentos.

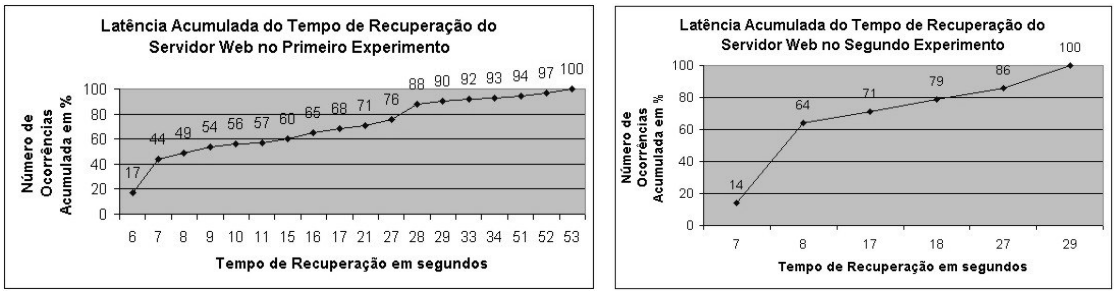

Figura 4: Latência acumulada do tempo de recuperação do servidor Web nos experimentos.

Deve ser considerado que o intervalo de testes da ferramenta de gerência foi e o intervalo de testes da ferramenta SAPOTI foi de 10 segundos. Naturalmente é possível melhorar os resultados obtidos reduzindo estes intervalos.

A figura 5 mostra o número de vezes que cada máquina iniciou o servidor Web nos dois experimentos. No primeiro experimento, mesmo com um ambiente com 210 falhas distribuídas entre todas as máquinas, onde o servidor Web trocou 177 vezes de máquinas, a disponibilidade do servidor Web foi de 97,35% do tempo do experimento. Portanto, durante as 12 horas e 23 minutos do experimento a indisponibilidade do servidor Web foi de 2,65%, ou seja, não passou de 20 minutos. Deve ser frisado que durante parte deste tempo todas as máquinas estavam falhas, o que não ocorre no segundo experimento.

No segundo experimento, mesmo com 27 falhas, onde o servidor Web trocou 41 vezes de máquinas, a disponibilidade do servidor Web foi de 99,58% do tempo do experimento. Portanto, durante as 12 horas e 40 minutos do experimento a indisponibilidade do servidor Web foi de 0,42%, ou seja, não passou de 4 minutos.

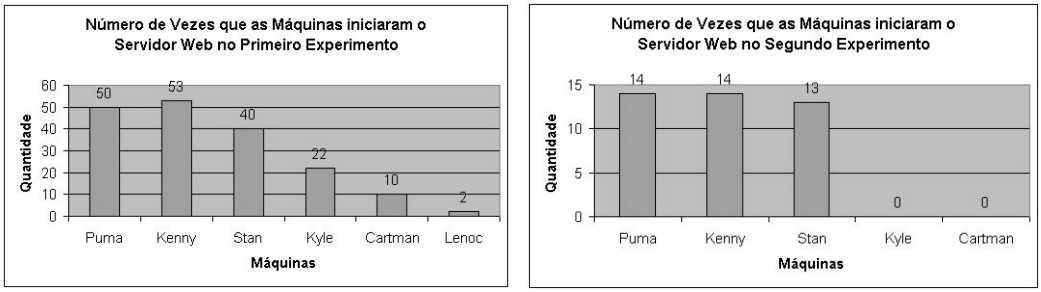

Figura 5: Número de vezes que cada máquina iniciou o servidor Web nos dois experimentos.

5. Conclusão

Neste trabalho foi apresentado o SAPOTI, uma ferramenta distribuída que garante a alta disponibilidade de servidores de aplicações TCP/IP aplicada especificamente para a implementação de um servidor Web tolerante a falhas. A ferramenta SAPOTI atua de forma distribuída em uma rede de computadores monitorados pela ferramenta de gerência confiável e distribuída baseada em diagnóstico hierárquico que implementa o algoritmo *Hi-ADSD with Timestamps.* Através das informações de diagnóstico geradas por esta ferramenta, o SAPOTI disponibiliza o servidor Web em uma máquina da rede, caso exista alguma máquina sem-falha que esteja sendo monitorada nesta rede.

Uma interface Web foi implementada para visualizar as informações de diagnóstico. Experimentos com injeção de falhas foram realizados em um intervalo de observação de cerca de 12 horas, em máquinas monitoradas pela ferramenta de gerência confiável e distribuída e que executavam a aplicação apresentada neste trabalho. Como resultado foi constatado que em configurações de rede com seis máquinas onde ocorreram 210 falhas distribuídas entre todas as máquinas, a disponibilidade do servidor Web foi de 97,3%. Em outra configuração de rede com cinco máquinas, onde algumas máquinas juntas falharam 27 vezes, a disponibilidade do servidor Web foi de 99,5%.

Trabalhos futuros incluem a aplicação da ferramenta SAPOTI para outros serviços TCP/IP, além da Web.

6. Referências

[1] D. E. Comer, Internetworking with TCP/IP – Principles, Protocolos, and Architectures, Prentice Hall, 4a ed., Vol. 1, 1995.

[2] E. P. Duarte Jr., and L. C. E. Bona, "A Dependable SNMP-based Tool for Distributed Network Management", IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'2002), 2002. IEEE Computer Society Press, 2002. pp.279-284.

[3] E. P. Duarte Jr., A. Brawerman, and L. C. P. Albini, "An Algorithm for Distributed Hierarquical Diagnosis of Dynamic Fault and Repair Events", Proceedings of the IEEE International Conference on Parallel and Distributed Systems 2000.

[4] G. Masson, D. Blough, and G. Sullivan, "System Diagnosis in Fault-Tolerant Comuter System Design", ed. D. K. Pradhan, Prentice-Hall, 1996.

[5] *The NET-SNMP Project Home Page*, http://www.net-snmp.org. Acesso em 21/01/2003.

[6] *The Linux Home Page at Linux on Line*, http://www.linux.org. Acesso em 21/01/2003.

[7] *The Apache Software Foundation*, http://www.apache.org. Acesso em 21/01/2003.

[8] *Bash*, http://www.gnu.org/software/bash/bash.html. Acesso em 21/01/2003.

[9] *IP-Alias*, http://www.ibiblio.org/pub/Linux/docs/HOWTO/mini/IP-Alias. Acesso em 21/01/2003.

[10] *RSYNC*, http://www.rsync.org. Acesso em 21/01/2003.

[11] *PHP Hypertext Preprocessor*, http://www.php.net. Acesso em 21/01/2003.

[12] *Debian GNU/Linux, The Universal Operating System*, http://www.debian.org. Acesso em 21/01/2003.