

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

*Método para a Avaliação de Servidores WWW
no Ambiente Corporativo*

por

LEANDRO CÔRTE

**Trabalho de conclusão submetido à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Informática**

Prof. Rômulo Silva de Oliveira
Orientador

Porto Alegre, 31 de Janeiro de 2002.

Sumário

Lista de Abreviaturas	4
Lista de Figuras	5
Lista de Tabelas	6
Resumo	7
Abstract	8
1. Introdução	9
1.1 Objetivos	11
1.2 Estrutura do Texto.....	11
2. World Wide Web (WWW)	13
2.1 Cliente	13
2.2 Servidor.....	13
2.3 HTTP - <i>HyperText Transfer Protocol</i>	14
2.4 Aplicações Web	16
2.4.1 CGI.....	16
2.4.2 ISAPI / NSAPI.....	17
2.4.3 ASP / PHP	19
2.5 Conclusão.....	21
3. Ferramentas, <i>Benchmarks</i> e Métricas para Avaliação de Servidores WWW	22
3.1 Ferramentas de Avaliação do Serviço WWW	22
3.1.1 SPECweb99	22
3.1.2 WebStone	24
3.1.3 WebBench	26
3.2 <i>Benchmarks</i>	27
3.3 Conclusão.....	31
4. Método para Avaliação de Servidores WWW.....	32
4.1 Descrição do Método Proposto	32
4.1.1 Definição dos Objetivos	33
4.1.2 Seleção das Métricas	33
4.1.3 Levantamento de Informações e Definição dos Parâmetros	33
4.1.4 Seleção e Caracterização do <i>Workload</i>	34
4.1.5 Montagem do Ambiente de Teste e Medições	34
4.1.6 Apresentação dos Resultados	35
4.1.7 Análise e Interpretação dos Dados	35
4.2 Proposta de Critérios de Avaliação	35

4.3 Métricas Propostas	37
4.4 Conclusão.....	41
5. Aplicação no Contexto da Procempa.....	42
5.1 Definição dos Objetivos	42
5.2 Seleção das Métricas	43
5.3 Levantamento de Informações e Definição dos Parâmetros	44
5.4 Seleção e Caracterização do <i>Workload</i>	46
5.5 Montagem do Ambiente de Teste e Medições	48
5.6 Apresentação dos Resultados	51
5.7 Análise e Interpretação dos Dados	63
5.8 Conclusão.....	65
6. Conclusões	66
Anexo 1 Levantamento de Informações do Servidor Web.....	68
Anexo 2 Arquivos de Configuração WebStone.....	77
Bibliografia	83

Lista de Abreviaturas

ASP	Active Server Pages
CGI	Common Gateway Interface
DLL	Dynamic Linked Libraries
GNU	GNU's not Unix
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IIS	Internet Information Services
ISAPI	Internet Server Application Program Interface
MTBF	Mean Time Between Failures
MTTF	Mean Time to Failure
NSAPI	Netscape Application Program Interface
PHP	PHP Hypertext Preprocessor
URL	Uniform Resource Locator
WWW	World Wide Web

Lista de Figuras

FIGURA 2.1 – Diferença entre ISA e CGI	17
FIGURA 2.2 – Funcionamento de aplicação ASP	20
FIGURA 3.1 – Desempenho conteúdo dinâmico	28
FIGURA 3.2 – Desempenho aplicações E-Commerce	28
FIGURA 3.3 – Desempenho conteúdo estático.....	29
FIGURA 4.1 – Possibilidades de saídas em uma requisição de serviço.....	36
FIGURA 5.1 – Ambiente de teste.....	50
FIGURA 5.2 – Comparativo desempenho temporal (vazão)....	57
FIGURA 5.3 – Comparativo desempenho temporal (tempo de resposta)	57
FIGURA A1.1 – Monitor de desempenho: Memória	72
FIGURA A1.2 – Monitor de desempenho: Processador	74
FIGURA A1.3 – Monitor de desempenho: Serviço web.....	76

Lista de Tabelas

TABELA 2.1 – Métodos utilizados no HTTP	15
TABELA 2.2 – Comparativo CGI e ISAPI	18
TABELA 2.3 – Vantagens e desvantagens de CGI e ISAPI ...	18
TABELA 4.1 – Métricas para avaliação do serviço WWW	39
TABELA 5.1 – Características do servidor web Procempa	43
TABELA 5.2 – Distribuição páginas estáticas e dinâmicas	45
TABELA 5.3 – Informações de acesso a páginas no servidor .	45
TABELA 5.4 – Distribuição dos tamanhos de páginas estáticas	46
TABELA 5.5 – Características dos equipamentos do ambiente de teste	48
TABELA 5.6 – Relatório final apresentado pelo WebStone ...	53
TABELA 5.7 – Resultados desempenho temporal IIS	54
TABELA 5.8 – Resultados desempenho temporal Apache	56
TABELA 5.9 – Métricas para avaliação do serviço IIS	58
TABELA 5.10 – Métricas para avaliação do serviço Apache...	61
TABELA A1.1 – Métricas referentes à memória do servidor web	69
TABELA A1.2 – Métricas referentes ao processador do servidor web	73
TABELA A1.3 – Métricas referentes ao serviço web do servidor	75

Resumo

O principal objetivo deste trabalho é apresentar um método e métricas para a avaliação do serviço Internet mais amplamente utilizado: a *World Wide Web*.

As características básicas e funcionamento do serviço, bem como algumas ferramentas de avaliação de desempenho, serão descritas. Estes capítulos servirão de base para os demais, onde serão apresentados o método para avaliação do serviço web e métricas usadas para análise de desempenho, disponibilidade, confiabilidade, facilidades de administração e recursos.

Por fim, o método e métricas serão aplicados na Procempa – Companhia de Processamento de Dados do Município de Porto Alegre, onde será possível verificá-los na prática. Além disto, dados importantes sobre a infra-estrutura web da Procempa serão fornecidos, os quais permitem uma análise do ambiente web atual e futuro da empresa.

Palavras-Chave: Métricas, *Benchmarks*, Internet, Sistemas Operacionais.

Abstract

The main goal of this work is to show a method and metrics used for the evaluation of the most used Internet service - the World Wide Web.

The basic features and operation of this service, as well as some performance evaluation tools will be described. These chapters will be the base for the other ones that present a method to evaluate the web service and the metrics used for the analysis of performance, availability, dependability, management facilities and resources.

Finally, the method and the metrics will be applied at Procempa – Porto Alegre Data Processing Company, where we shall be able to verify both in practice. In addition, some important data about the Procempa web service infra-structure will be supplied. This will enable an analysis of the company's web environment of today and its future.

Keywords: Metrics, Benchmarks, Internet, Operating Systems.

1. Introdução

A Internet tem experimentado um crescimento exponencial desde seu surgimento. O número de servidores Internet pode ser contado em dezenas de milhões e, de acordo com várias pesquisas, cresce a uma taxa muito alta em todo mundo. A *World Wide Web* (WWW) tem crescido a passos largos. O número de *sites* web dobram em menos de 6 meses, aplicações como bibliotecas digitais, comércio eletrônico, vídeo sob demanda e ensino a distância aumentam o tráfego Internet e web a taxas ainda maiores. Lojas virtuais permitem a compra de carros, livros, computadores e muitos outros produtos. Muitos órgãos governamentais estão usando servidores web para disseminar documentos e formulários para a população em geral [MEN 98].

Com este aumento de importância, principalmente com o advento do comércio eletrônico e aplicações web, a infra-estrutura das empresas de um modo geral (sistema operacional, serviços, linhas de comunicação) assume um papel cada vez mais crítico.

Algumas páginas web populares recebem milhões de acessos por dia e não é incomum encontrarmos tempos de resposta insatisfatórios ou *sites* inacessíveis, sendo causa de frustração entre os usuários destes serviços [MEN 98].

A fidelidade dos usuários não é um fator comum quando tratamos de Internet, por isso diz-se usualmente que o concorrente está a um clique de mouse. Atrasos no acesso, serviços não disponíveis, perda de transações, podem ser decisivos para que o usuário busque outro *site* que atenda suas exigências.

A perda de dinheiro nestes casos é facilmente notada e contabilizada. A Dell Computer serve como exemplo disto. Os dados do início do ano de 1999 apontam que seu faturamento diário via web *site* foi de \$ 10 milhões. Uma hora parada implica em uma perda média de \$ 416.000 [JAE 99]. Os montantes vêm crescendo ano após ano.

Estes fatores apontam para a relevância de se ter uma estrutura bem dimensionada e preparada para receber as solicitações dos usuários. Um importante mecanismo para alcançar este objetivo é o emprego de um método para avaliação de servidores web que pode ser aplicado em casos como:

- Um período de avaliação prévio à montagem de uma estrutura web e servidores de aplicação;
- Avaliar alternativas de sistemas operacionais e serviços oferecidos no mercado;
- Verificar a estrutura já existente e como ela está atendendo as suas funções.

A partir do estudo feito no trabalho individual [COR 00], percebe-se que existem no mercado ferramentas desenvolvidas para avaliação de servidores web

relacionadas com o aspecto desempenho [MIN 99] [STA 00] [ZDN 99]. Têm-se artigos, como em [RAD 99] [CHE 96] que avaliam servidores web sob aspectos específicos como hardware e sistema operacional. Entretanto, falta um detalhamento maior do que exatamente está se medindo. Os trabalhos publicados normalmente apresentam um resultado final e geral, ou seja, não se sabe o que exatamente foi levado em consideração do sistema operacional, do serviço testado e dos demais fatores, como, por exemplo, limitações de rede.

Além das disparidades com respeito às medidas relacionadas com desempenho, também existe uma carência de métricas mais abrangentes que envolvam disponibilidade, administração e recursos, levando sempre em consideração a necessidade de *benchmark* específico por aplicação, seguindo uma metodologia adequada [SEL 99]. Esta é uma área onde existe muito espaço para evolução considerando-se as práticas atuais no mercado. Em particular, a definição de um conjunto coerente de métricas que cubra as várias dimensões (desempenho, disponibilidade, administração, recursos, etc) do termo “Qualidade de Serviço” neste contexto.

O presente trabalho tem como objetivo apresentar um método para avaliação de servidores WWW, bem como, definir métricas que permitam avaliar alguns aspectos da estrutura de servidores web: desempenho, disponibilidade, administração e recursos. Estas métricas foram obtidas a partir de um levantamento de *benchmarks* e ferramentas existentes no mercado, além de consulta bibliográfica sobre o assunto, em revistas técnicas e na Internet.

Após o estudo do funcionamento do serviço e posterior definição do método e das métricas a serem usadas, passou-se para a etapa de aplicação em ambiente de teste na Procempa, a fim de avaliar os sistemas operacionais GNU-Linux e o Windows 2000 executando seus serviços específicos web o Apache e o *Internet Information Services* (IIS) [MAN 98] respectivamente. Com isto, foi possível, em primeiro lugar, validar o conjunto de métricas estabelecidas, permitindo a verificação das escolhas e método utilizado e, posteriormente, avaliar o ambiente atual de infraestrutura de web utilizada, servindo como base para alterações nesta estrutura e planejamento para futuras implementações.

A Procempa está engajada no projeto de estudo e aplicação de softwares livres que vem sendo coordenada por diversos órgãos estaduais e municipais, universidades e empresas privadas. Busca-se alcançar uma série de benefícios com a utilização de software livre como redução de custos, flexibilidade, independência no desenvolvimento, aprimoramento de aplicações e independência de fornecedor. No entanto, não se quer abrir mão de qualidade e funcionalidade já existentes.

A avaliação do serviço web utilizando-se GNU-Linux (dentro do contexto de software livre, a principal ferramenta empregada) e Windows 2000 pode esclarecer e embasar decisões, fugindo de informações subjetivas do tipo “Sistema operacional ‘x’ é mais estável que o ‘y’ ” ou “A empresa ‘z’ diz que o software ‘a’ executa funções mais rapidamente que ‘b’ “.

Desta forma, o trabalho desenvolvido nesta dissertação de mestrado pode ser de grande valia não somente para empresas em geral que mantêm servidores web, mas principalmente para aquelas empresas considerando a possibilidade de alterações fundamentais em sua infra-estrutura computacional de suporte aos serviços Internet, como empresas migrando de software proprietário para software livre.

1.1 Objetivos

O objetivo principal do presente trabalho é definir um método para avaliação de servidores WWW, bem como, estabelecer métricas abrangendo os diversos aspectos relevantes a administração e funcionamento do serviço, sendo os principais elementos: desempenho temporal, disponibilidade, confiabilidade, administração e recursos.

O método foi aplicado em um ambiente de teste na Procempa - Companhia de Processamento de Dados do Município de Porto Alegre, avaliando os sistemas operacionais GNU-Linux e Windows 2000 com seus serviços, Apache e IIS respectivamente.

Os objetivos parciais que tiveram de ser alcançados para se chegar ao resultado final foram:

- Estudar ferramentas existentes no mercado e avaliar sua utilização como fonte para obtenção de métricas;
- Estudar os sistemas operacionais Windows 2000 e Linux, enfocando os serviços web IIS (*Internet Information Services*) e Apache;
- Definir o método a ser utilizado;
- Estabelecer métricas para avaliação de servidores web;
- Aplicar o método em ambiente de teste;
- Avaliar e analisar os resultados obtidos;
- Elaborar a dissertação de mestrado que detalha o método, as métricas e explica os testes feitos, resultados obtidos e análise final.

1.2 Estrutura do Texto

A apresentação deste trabalho está segmentada em seis capítulos, sendo o primeiro esta introdução.

O segundo capítulo apresenta de forma sucinta o serviço *World Wide Web*, trazendo os principais conceitos da estrutura web, tanto do lado do cliente, quanto

do servidor, incluindo a conceituação do protocolo *HTTP - HyperText Transfer Protocol*. Em seguida são descritas algumas técnicas de desenvolvimento de aplicações para o ambiente web como CGI, ISAPI, NSAPI, ASP e PHP.

O terceiro capítulo, primeiramente, apresenta três das principais ferramentas de avaliação do serviço WWW, sendo elas: SPECweb99, WebStone e WebBench. São descritos, então, alguns *benchmarks* de serviço web realizados por empresas e organizações em geral, com o objetivo de verificar métodos e métricas utilizadas.

No quarto capítulo é proposto um método de avaliação do serviço WWW, sendo detalhadas sete etapas: Definição dos objetivos, seleção das métricas, levantamento de informações e definição dos parâmetros, seleção e caracterização do *workload*, montagem do ambiente de teste e medições, apresentação dos resultados e análise e interpretação dos dados. Por fim, são descritas as métricas, buscando fornecer indicadores que permitam mensurar o serviço no que tange ao desempenho temporal, disponibilidade e confiabilidade, administração e recursos.

No quinto capítulo a teoria colocada nas etapas anteriores é verificada na prática. O método e conjunto de métricas são aplicadas em ambiente de teste, usando como base informações da Procempa.

No sexto e último capítulo são discutidas as conclusões obtidas com a realização do trabalho, incluindo as propostas do método e métricas, bem como a aplicação prática dos mesmos no ambiente corporativo.

2. World Wide Web (WWW)

Desde sua criação em 1989 pelo CERN, centro de pesquisa nuclear europeu, a World Wide Web, ou simplesmente web, tem evoluído em muitos aspectos e se tornou o serviço mais popular e de maior importância da Internet. Hoje ela é usada não somente para buscar informações de centros de pesquisa ou universidades, mas também como meio de se fazer negócios e trocar dados dos mais variados possíveis.

Serão descritos, em seguida, os dois lados da estrutura web, o lado cliente e o lado servidor. Também será abordado o protocolo utilizado, o HTTP (*HyperText Transfer Protocol*), nas suas versões 1.0 e 1.1, além das formas de desenvolver aplicações utilizando esta tecnologia.

2.1 Cliente

O software responsável pelo lado cliente de uma conexão web é o navegador. Sua função principal é receber e tratar páginas web, contendo textos e imagens formatadas no padrão HTML (*Hypertext Markup Language*) [NIE 99].

Com os avanços que vem ocorrendo, a importância e as funções do navegador aumentaram muito. Linguagens de programação foram agregadas e executam no cliente, sendo as mais utilizadas o Java, JavaScript e Visual Basic Script, recursos de multimedia, como áudio e vídeo devem ser tratados, novas funcionalidades que se agregam ao HTML surgiram, como *Dynamic HTML* (DHTML), *Cascading Style Sheets* e XML (*Extensible Markup Language*), sem falar na questão de segurança, onde o navegador viabiliza conexões seguras, através de protocolos seguros como o SSL (*Secure Sockets Layer*).

Enfim, o navegador deixou de ser o programa simples que tratava apenas de texto e imagens no formato gif e jpg, para se tornar um software mais complexo e que está se consolidando como interface da Internet e de aplicações em geral.

2.2 Servidor

A forma tradicional de uma comunicação entre o cliente e o servidor inicia através de uma solicitação do navegador a uma determinada página. Um processo servidor deve estar executando, normalmente “escutando” a porta 80, para atender a

requisição do usuário e enviar via protocolo HTTP a página solicitada.

Detalhando melhor este processo, temos as seguintes etapas:

1. O navegador determina o URL (*Uniform Resource Locator*), verificando qual foi selecionado
2. O navegador pergunta ao DNS (serviço responsável pela resolução de nomes) qual o endereço IP do site em questão
3. O DNS responde ao navegador com o IP
4. O navegador estabelece uma conexão TCP com a porta 80 (porta padrão do serviço HTTP) do servidor
5. Em seguida, o navegador envia um comando GET, o qual requisita uma determinada página
6. O servidor envia a página solicitada
7. A conexão TCP é liberada
8. O navegador apresenta a página
9. O navegador busca e apresenta (seguindo os mesmos passos anteriores) as imagens da página

Uma observação importante é que ao enviar o comando GET, também é negociada a versão do protocolo a ser usado: HTTP 1.0 ou HTTP 1.1.

Como os clientes, os servidores também evoluíram muito em suas funções. Agregam outros serviços como servidor de indexação e de certificação, suportam formas de autenticações diferenciadas, implementam protocolos seguros, tratam e personalizam mensagens de erros, entre outros.

Mas a principal evolução dos servidores web é o suporte ao desenvolvimento de aplicações. O servidor está deixando de ter a simples função de devolver páginas estáticas e partindo para tratar o que se denomina de páginas dinâmicas.

O servidor executa programas definidos em variados tipos de padrão como CGI's, API's e linguagens de script, podendo acessar bases de dados ou processar lógicas mais complexas.

2.3 HTTP - HyperText Transfer Protocol

O HTTP é o protocolo de transferência padrão da web. Cada interação consiste em uma solicitação ASCII, seguida de uma resposta RFC 822 do tipo fornecido pelo MIME.

O HTTP está em constante evolução. Há várias versões em uso e outras tantas em desenvolvimento. As versões que atualmente são usadas são a versão 1.0 e

1.1.

As novas versões do protocolo aceitam dois tipos de solicitações: simples e completas. Uma solicitação simples é apenas uma linha GET que identifica a página desejada, sem a versão do protocolo. A resposta é formada apenas pela página, sem cabeçalhos, sem MIME e sem códigos.

As solicitações completas, hoje as mais utilizadas, são indicadas pela presença do protocolo na linha de solicitação GET. As solicitações podem consistir em várias linhas, seguidas de uma linha em branco para indicar o fim da solicitação.

Abaixo é apresentada uma tabela contendo os principais comandos, ou métodos, utilizados em uma conexão web.

TABELA 2.1 - Métodos utilizados no HTTP

Método	Descrição
GET	Solicita a leitura de uma página da Web
HEAD	Solicita a leitura de um cabeçalho da página
PUT	Solicita o armazenamento de uma página
POST	Transporta uma URL, acrescentando a um recurso
DELETE	Remove a página
LINK	Conecta dois recursos existentes
UNLINK	Desfaz uma conexão entre dois recursos

Fonte: TANENBAUM. Redes de Computadores. p. 788

Cada solicitação obtém uma resposta que consiste em uma linha de status e informações adicionais (como a página ou parte dela). Pode ser apresentado o código de OK (200), ou qualquer um dentre uma variedade de códigos de erro, por exemplo solicitação inválida (400) ou acesso proibido (403).

As novas versões de clientes e servidores suportam o protocolo HTTP 1.1. Algumas de suas características são:

- *Pipelining*: Permite ao cliente enviar várias requisições antes de receber uma resposta do servidor. Ocorre um ganho considerável de desempenho com o uso deste recurso.
- Conexões persistentes (*Keep-Alives*): Quando o navegador conecta ao servidor e requisita uma página, a conexão é estabelecida com o servidor. Estabelecer e liberar conexões a cada requisição é um processo que dispende recursos do cliente, do servidor e da rede. Com o uso de conexões persistentes, um cliente pode usar uma única conexão, ou reduzir o seu número, quando fizer requisições

- múltiplas.
- HTTP PUT e DELETE: Com os métodos PUT e DELETE os usuários podem enviar ou deletar arquivos do servidor.
- Utilização de *cache*: Usa um mecanismo de expiração que evita que sejam enviadas páginas que não sofreram alterações.

2.4 Aplicações Web

2.4.1 CGI (Common Gateway Interface)

CGI é uma forma de interligação entre aplicativos externos com servidores de informação, do tipo servidores HTTP ou servidores web. Uma página HTML (*Hipertext Markup Language*) comum é composta por informações estáticas, ou seja, um texto que não pode ser mudado. Um programa CGI permite a montagem de uma página HTML em tempo real, com informações mais específicas ao interesse do usuário.

Por exemplo, digamos que se queira disponibilizar informações contidas em uma base de dados para qualquer usuário de um servidor de informações. Para isso, tem-se um formulário inicial, onde o usuário preenche determinadas informações e ao final, aciona um botão na tela, submetendo o pedido. Essas informações são passadas ao programa CGI, que processa essas informações, consulta uma base de dados e retorna as informações solicitadas através de uma página específica.

Um programa CGI pode ser escrito em qualquer linguagem que permita ser executado no sistema, como: C/C++, Fortran, PERL, TCL, Visual Basic, entre outras.

Contudo, esses programas necessitam residir em diretórios especiais (geralmente nomeado *cgi-bin*), por questões de segurança e também para diferenciar de páginas comuns. Assim, o servidor sabe que esses programas devem ser executados e não simplesmente mostrados. Estes diretórios estão sob controle direto do responsável pela administração web, proibindo usuários comuns de habilitarem programas CGI.

Cada vez que o cliente pede o URL (*Uniform Resource Locator*) correspondente ao programa CGI, este é executado em tempo real, através da criação de um processo específico no servidor.

2.4.2 ISAPI (Internet Server API) / NSAPI (Netscape Server API)

ISAPI (*Internet Server API*) é a interface proposta pela Microsoft como uma alternativa mais rápida e mais eficiente que o CGI. ISAPI é parte integrante do *Microsoft Internet Information Services (MS-IIS)*.

Da mesma forma que em programas CGI, um programa ISAPI permite que usuários remotos executem um programa no servidor, procurando informações em uma base de dados ou interagindo com outros programas.

Programas escritos utilizando tecnologia ISAPI são compilados como DLL (*Dynamic-Linked Libraries*), que são lidas pelo servidor e mantidas em memória, no caso de estarem em constante uso. Estas DLL baseadas em ISAPI são chamadas de *Internet Server Applications (ISA)*.

As DLL ISA, por padrão, são executadas no mesmo processo do servidor HTTP. Isto significa dizer que todos os recursos disponíveis ao servidor HTTP também estão disponíveis às DLL ISA. Conseqüentemente, uma DLL ISA executa mais rapidamente que um programa CGI convencional.

A figura seguinte, ilustra a diferença entre as arquiteturas ISA e CGI.

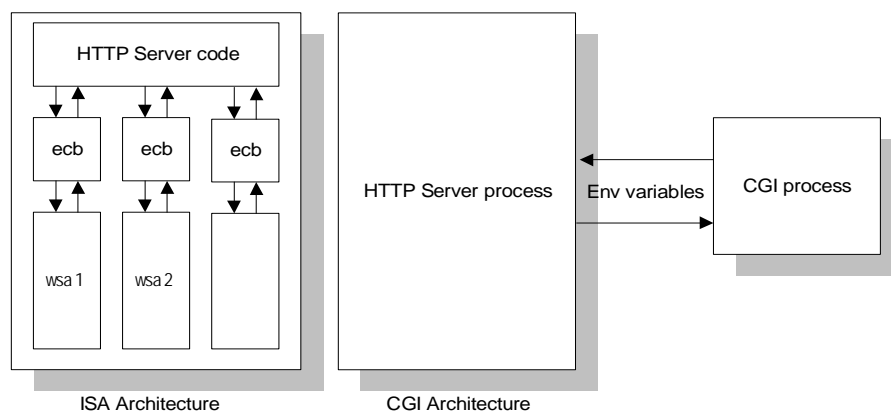


FIGURA 2.1 – Diferença entre ISA e CGI

Como descrito anteriormente, múltiplas DLL ISA podem coexistir no mesmo processo do servidor, enquanto que aplicações CGI convencionais rodam em processos diferentes.

A interação entre o servidor HTTP e uma DLL ISA é feita através dos ECB (*Extension Control Blocks*), ou seja, endereço de uma estrutura com os dados de entrada.

Uma ISA DLL deve suportar múltiplas *threads*, pois múltiplos pedidos

serão recebidos simultaneamente.

É claro que existe um preço a ser pago pelo melhor desempenho, pois na arquitetura ISA, um problema na DLL ISA pode causar um problema no servidor.

A tabela abaixo mostra as principais diferenças de uma implementação CGI convencional e uma aplicação ISAPI.

TABELA 2.2 – Comparativo CGI e ISAPI

	CGI	ISAPI
Programa	.EXE	.DLL
Web Server	IIS, Apache, Netscape, ...	IIS
Processo	Cada requisição, um processo diferente	Todos no mesmo processo do servidor

A tabela abaixo demonstra as principais vantagens/desvantagens comparativamente entre as arquiteturas CGI convencional e ISAPI.

TABELA 2.3 – Vantagens e Desvantagens de CGI e ISAPI

	CGI	ISAPI
Vantagens	<ul style="list-style-type: none"> - facilidade de implementação - portabilidade do aplicativo - possibilidade de uso de uma grande gama de linguagens - outras possibilidades de servidores web 	<ul style="list-style-type: none"> - mais rápido - menor <i>overhead</i> - necessita hardware de menor capacidade
Desvantagens	<ul style="list-style-type: none"> - necessita hardware de maior capacidade - mais lento 	<ul style="list-style-type: none"> - Um problema em uma DLL específica pode ocasionar problemas no servidor web - Específico para Windows IIS

Além da ISAPI, recurso proprietário da Microsoft, outras formas de desenvolver aplicações via API são empregadas, como NSAPI.

A NSAPI é a API proprietária dos servidores Netscape e, como o ISAPI, se constitui em uma forma de estender as funcionalidades do servidor web específico.

Através desta API é possível criar módulos executáveis para adicionar ou substituir partes do próprio servidor web como autenticação e autorização de usuários, históricos de erro e geração de conteúdo.

Em duas situações específicas, a utilização de NSAPI é extremamente útil:

1. Códigos que devem ser executados em quase todos os acessos feitos ao servidor, por exemplo segurança.
2. Rotinas com processamento pesado. NSAPI, como ISAPI, provê um aumento substancial de desempenho quando comparado com CGI.

2.4.3 ASP (*Active Server Pages*) / PHP (*PHP: Hypertext Preprocessor*)

A tecnologia ASP (*Active Server Pages*) foi introduzida com o servidor web da Microsoft, o IIS, na sua versão 3.0 e visa a simplificar a programação de modo que seja fácil criar páginas com conteúdo dinâmico e aplicações web.

Os scripts ASP tem a mesma funcionalidade que as CGI e aplicações ISAPI, mas são mais fáceis de desenvolver e modificar.

No núcleo desta tecnologia está uma extensão ISAPI, chamada Asp.dll, que é responsável por compilar e armazenar os arquivos .asp na memória em tempo de execução, sendo este um interpretador de scripts. Por exigir que as páginas ASP sejam interpretadas e compiladas antes de executá-las, scripts complexos podem ser quatro vezes mais lentos do que páginas HTML comuns e até três vezes mais lentos que aplicações ISAPI. Para amenizar esta perda, é mantida na memória do servidor uma cópia compilada, fazendo com que requisições subseqüentes sejam mais rápidas.

A tecnologia não é limitada a uma linguagem em particular. Linguagens como VBScripts, Jscripts e PerlScripts são suportadas.

As instruções ASP aparecem juntamente com comandos HTML (pode-se criar uma página ASP, simplesmente renomeando um .htm para .asp). Para diferenciar entre HTML e o script que rodará no servidor usa-se os delimitadores <% e %>.

A execução de um script ASP está representada na figura a seguir:

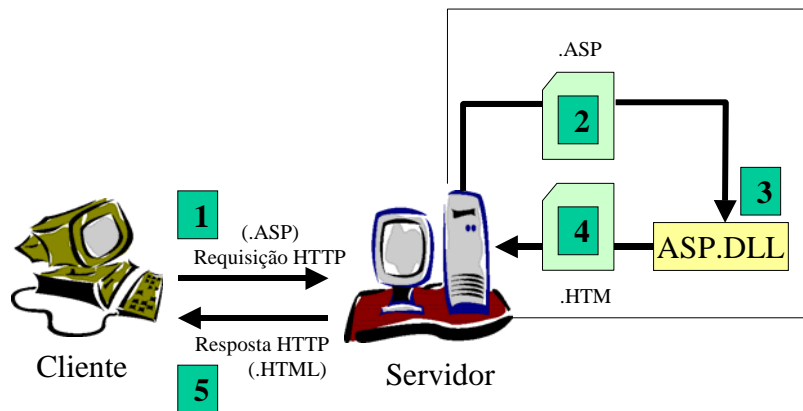


FIGURA 2.2 – Funcionamento de aplicação ASP

1. O cliente requisita uma página ASP, enviando uma requisição HTTP para o servidor;
2. Como a página tem terminação .asp, o servidor (IIS) a reconhece como um arquivo que tem de ser mapeado e envia o arquivo à ISAPI apropriada (neste caso Asp.dll) para execução;
3. O script é executado e o texto dinâmico é incorporado a página que será retornada ao usuário;
4. O servidor envia a página HTML resultante deste processo na forma de uma requisição HTTP comum;
5. A página é exibida ao cliente.

A tecnologia PHP (*Hypertext Preprocessor*) tem características bem similares, no aspecto de funcionalidade, ao ASP.

O PHP é uma linguagem de script para criação de páginas dinâmicas que trabalha em conjunto com o HTML. Quando um visitante abre uma página, o servidor processa os comandos PHP e envia o resultado desta execução ao navegador do usuário.

Uma importante diferença entre o PHP e ASP é que o PHP é uma tecnologia aberta, ou seja, é um software livre e que também executa em diversas plataformas, como Windows e diversas versões de Unix, incluindo Linux.

Pode-se trabalhar com o PHP de duas formas: como um módulo do Apache ou como código binário, funcionando como uma CGI. Quando usado como um módulo do Apache, ele se caracteriza por ser leve e rápido, sem *overhead* de criação de

processos. Quando usado o código binário, ganha-se em flexibilidade na execução com outros servidores.

O código PHP é incluído diretamente nas páginas web, sem necessitar ambientes específicos de programação. Um bloco é iniciado com os símbolos `<?php` e finalizado com `?>`. Pode-se configurá-lo para trabalhar com o estilo de ASP, usando `<%` e `%>` ou `<SCRIPT LANGUAGE="php">` `</SCRIPT>`.

A sintaxe da linguagem é similar ao C e Perl. Não há necessidade de declarar variáveis antes de usá-las e é fácil a definição de arrays. Algumas características rudimentares de orientação a objetos, também estão definidas, permitindo organizar e encapsular o código.

2.5 Conclusão

Foram apresentados os principais conceitos envolvidos na tecnologia *World Wide Web*, incluindo as definições básicas dos elementos no lado cliente e servidor. Em seguida, o protocolo HTTP foi descrito, em sua funcionalidade básica e em sua mais recente versão 1.1.

O desenvolvimento de aplicações constitui a evolução maior da tecnologia e é também o campo em maior ampliação. Diversos conceitos, estruturas e linguagens de programação são utilizados, sendo que para o presente trabalho as tecnologias CGI, ISAPI, ASP e PHP foram abordadas em decorrência de sua utilização no ambiente corporativo da Procempa.

O número de tecnologias associadas ao WWW aumenta constantemente e uma descrição completa e abrangente não é viável e de interesse para este texto. As descrições contidas neste capítulo contemplam o essencial para a execução do trabalho.

3. Ferramentas, Benchmarks e Métricas para Avaliação de Servidores WWW

Neste capítulo serão descritas três das principais ferramentas de avaliação do serviço WWW: SPECweb99, WebStone e WebBench. Existem várias outras disponíveis no mercado, sendo que a grande maioria segue os formatos expostos nestes exemplos.

Ao final, serão apresentados alguns *benchmarks* realizados por empresas e organizações em geral, onde o que se busca é basicamente verificar métodos e métricas utilizadas.

3.1 Ferramentas de Avaliação do Serviço WWW

Existem no mercado algumas ferramentas para avaliação de servidores web, principalmente relacionadas com a questão desempenho temporal. A seguir são descritas três destas ferramentas, sendo que a importância em analisá-las está centrada principalmente em conhecer o funcionamento e facilidades oferecidas.

3.1.1 SPECweb99

SPECweb99 (<http://www.spec.org/osg/web99/>) é um software de benchmark desenvolvido pelo SPEC (*Standard Performance Evaluation Corporation*) e foi projetado para mensurar desempenho através da habilidade de um sistema agir como um servidor web, atendendo requisições a páginas estáticas e dinâmicas.

A ferramenta executa um gerador de carga *multi-thread* HTTP 1.0 / 1.1 em uma quantidade específica de clientes que farão uma série de requisições tanto de páginas estáticas, quanto dinâmicas em um sistema que está sob teste, ou SUT (*System Under Test*).

O principal resultado do SPECweb99 é o número máximo de conexões simultâneas suportadas por um servidor web. Estas conexões requisitam uma carga predefinida que o servidor web deve ser capaz de suportar enquanto mantém uma vazão específica e uma taxa de erros exigida. As conexões são feitas e mantidas a um bit rate especificado e com tamanho máximo de segmento ajustado ao que é usado na Internet.

A carga gerada pelo SPECweb99 reproduz páginas web de diferentes organizações, variando em tamanho, indo desde pequenas páginas com ícones, até documentos grandes com imagens. Como no mundo real, são reproduzidas páginas mais populares que outras e comandos GET dinâmicos simulam as propagandas rotativas, comuns nos *sites* existentes. Também são simuladas, através de comandos POST, entradas de usuários através de formulários e armazenamento em arquivos.

Nesta versão do SPECweb foi adicionada a capacidade de atender a requisições dinâmicas. A distribuição atual é: 16% POST, 41,5% GET, 42% GET com *cookies* e 0,5% CGI através de GET. As requisições dinâmicas compreendem 30% do total da carga.

Existe também uma distribuição como esta para classes de tamanho de páginas estáticas, indo de páginas pequenas (1 KB) até páginas com tamanho acima da média (1 MB).

Outra característica importante é de que 70% das requisições usam o recurso de *Keep-alive header* do HTTP 1.0 ou o recurso de conexões permanentes do HTTP 1.1. Os outros 30% são requisições HTTP 1.0 normais, ou seja, são encerradas a cada conexão.

A métrica de desempenho usada no software também é chamada de SPECweb99 e se caracteriza por mensurar o número máximo de conexões simultâneas, como colocado anteriormente.

O software usa um ou mais clientes para criar a carga descrita anteriormente. Cada cliente envia requisições HTTP ao servidor e valida as respostas recebidas. Ao final da execução do *benchmark*, os dados de todos os clientes são coletados por um cliente central. Este cliente trabalha com os dados para calcular o bit rate do teste e determinar o número de conexões simultâneas que estejam em conformidade com os limites de bit rate estabelecidos.

Ao final da execução do SPECweb99 é gerado um relatório contendo os resultados alcançados, os quais são computados a partir da execução do teste três vezes consecutivas.

Os dados reportados são divididos em sete blocos:

1. Hardware do Servidor: Engloba dados como o nome do fabricante, taxas de *clock* e tipo do processador, número de processadores, tamanho da memória principal, tamanhos e tipos de *cache*, controladoras de disco e *drivers* e tipo do sistema de arquivos.
2. Software do Servidor: Servidor HTTP (Web) usado, sistema operacional e versão, valores do MSL (*Maximum Segment Life*) e *TIME-WAIT*, qualquer outro software usado durante o processo de *benchmark*, número de *daemons* rodando, tamanho do *cache* do servidor web e se está configurado para *disk striping*.
3. Configuração de Rede: Número, tipo e modelo das controladoras de

rede, tipos de configurações de rede utilizadas, velocidade base da rede, características dos componentes externos da rede (*hubs*, roteadores e *switchs*) e como os clientes do teste estão conectados a rede.

4. Clientes: Número de clientes, características das máquinas clientes, sistema operacional usado e versão e compilador usado para gerar o programa de *benchmark*.
5. Datas de disponibilidade dos produtos: São informadas as datas em que o hardware, software HTTP e sistema operacional estavam ou estarão à disposição.
6. Quem realizou o teste: Contém a data de realização do teste, qual organização o realizou e número de licença (visto que o SPECweb é um produto comercial).
7. Observações: Se o sistema é mono ou multi-usuário, parâmetros do sistema e processos diferentes do padrão, tamanho do MTU usado, carga do sistema em background e observações sobre os softwares e hardware utilizado.

No endereço Internet:

<http://www.specbench.org/osg/web99/results/res99q4/web99-19991213-00011.html>

é apresentado o resultado das interações feitas com o SPECweb99 pela Dell Computer, testando o *Internet Information Services 5* da Microsoft, executando sobre o sistema operacional Windows 2000.

3.1.2 WebStone

O *benchmark* WebStone (<http://www.mindcraft.com/webstone>) foi desenvolvido originalmente pela Silicon Graphics com a finalidade de mensurar o desempenho de servidores web. A empresa Mindcraft adquiriu os direitos sobre o produto e desenvolveu uma nova versão, WebStone 2.5, buscando aumentar a confiabilidade e portabilidade. Também disponibiliza os executáveis e fontes através de sua página web.

O seu funcionamento, de um modo geral, é semelhante ao SPECweb99. É criada uma carga de arquivos que é recuperada através da simulação de diversos clientes acessando um servidor web. Pode-se ter a simulação de até 200 clientes web em um mesmo computador e/ou ter ainda a possibilidade de distribuir os clientes por vários equipamentos separados.

O programa que faz todo o controle do teste chama-se Webmaster e pode rodar em um dos clientes que participam do teste ou em um equipamento separado. O Webmaster é responsável por distribuir os softwares que rodam no cliente e os arquivos de teste para os participantes da simulação. Também é ele quem inicia o processo e recebe os dados de desempenho medidos. Ao final, ele condensa os resultados de todos

os clientes web em um relatório consolidado.

Os testes realizados com o WebStone, visam a avaliar, além do servidor web que está atendendo as requisições, o sistema operacional, o processador e a velocidade de rede envolvidos no processo.

São utilizados três métodos de acesso diferentes:

1. HTML: Este teste retorna páginas no formato padrão HTML e se constitui no principal método utilizado.
2. CGI: O servidor web executa outro programa, usando CGI (*Common Gateway Interface*) para passar os parâmetros especificados na requisição. Este método é normalmente utilizado para processar formulários.
3. API: Este método é, na maioria das vezes, mais eficiente do que as CGI ao responder requisições que envolvem alguma programação. Um dos testes executados recebe o nome de um arquivo específico e retorna seu conteúdo, utilizando uma determinada API (*Application Programming Interface*). As API suportadas pela versão corrente são a NSAPI (*Netscape API*) e ISAPI (*Microsoft Internet Information Services API*).

Os seguintes dados são reportados no relatório final:

1. Servidor testado: Inclui informações como número e tipo do processador, quantidade de memória, controladoras de disco, número de discos, qual disco contém o sistema operacional e o servidor web, controladoras de rede utilizadas, tipo e versão do sistema operacional, configurações feitas no TCP/IP.
2. Computador Cliente: Mesmos dados informados para servidor devem ser colocados nesta categoria.
3. Servidor Web: O nome do servidor web, versão, configurações feitas, número de *threads* e tamanho do *cache*.
4. Clientes Web: Número de clientes que participaram do teste e como foram distribuídos entre os computadores clientes.

As principais métricas fornecidas pelo WebStone são a vazão (bytes / segundo) e latência (tempo para completar uma requisição). Também são reportados número de páginas por minuto e média de taxas de conexão.

Os tipos de vazão mensurados são: o agregado e por cliente. A vazão agregada é simplesmente o total de bytes (corpo mais cabeçalho) transferidos dividido pelo tempo total de teste. A vazão por cliente divide o resultado do agregado pelo número de clientes.

A latência também é dividida em dois tipos: latência de conexão e latência de requisição. Para cada métrica o tempo médio é fornecido, bem como o desvio padrão e tempos mínimos e máximos. A latência de conexão reflete o tempo que

se leva para estabelecer a conexão, enquanto que a latência de requisição indica o tempo para completar a transferência dos dados, depois da conexão estabelecida.

A latência percebida pelo usuário será o somatório dos dois tipos descritos acima mais a latência da rede, decorrentes das conexões WAN, roteadores e modems.

Outra métrica informada pelo WebStone é denominada Little's Ls e reflete quanto tempo é gasto pelo servidor para processar as requisições. O Ls também indica indiretamente o número médio de conexões que o servidor web abriu em um determinado instante. Este número deve ser muito próximo ao número de clientes, ou então algum cliente está tendo o acesso negado ao servidor em um determinado instante.

3.1.3 WebBench

WebBench (<http://www.zdnet.com/zdbop>) é desenvolvido pela *Ziff-Davis Benchmark Operation* e está atualmente na versão 3.0. É um software gratuito, mas não permite acesso aos fontes.

O produto é projetado para “estressar” o servidor web sob testes até o limite de sua capacidade. Ao instalar o produto, um conjunto de arquivos é copiado para o diretório raiz do servidor. Este conjunto é composto por páginas html, imagens (gif) e arquivos executáveis.

Os clientes requisitam os arquivos através de HTTP, seguindo uma distribuição onde um maior número de pedidos são feitos a páginas estáticas e imagens e um número inferior a executáveis.

Cada cliente mantém armazenado o número de bytes transferidos, número de requisições bem-sucedidas e quantas falharam durante a fase de conexão ou durante as transferências das páginas.

Ao final da execução o WebBench, executando em um equipamento denominado controlador, recolhe os resultados dos clientes, consolida-os e produz dois resultados principais: requisições por segundo e vazão, medido em bytes por segundo.

Requisições por segundo é uma métrica básica em uma interação cliente / servidor, onde os clientes consideram apenas as requisições completas. Pegando-se o resultado de requisições por segundo atendidas pelo servidor, tem-se uma idéia do número de *hits* que o mesmo pode atender por dia. Como o WebBench é um teste que busca chegar ao limite do servidor, os clientes fazem o máximo de requisições possíveis e, como resultado, o escore de requisições por segundo gerado no teste tende a ser maior que a carga típica da maioria dos servidores web em operação.

A vazão informa quantos bytes por segundo o servidor está enviando aos clientes, sem considerar cabeçalhos de HTTP ou TCP/IP. O valor pode variar de acordo com as configurações informadas. Caso tenha-se optado por demandar um número maior de páginas grandes, a vazão será maior e o número de requisições menor e vice-versa.

Outras informações importantes que são colocadas nos relatórios de resultados:

- Informações do teste, como horário de início e fim e comentários, como características da máquina ou software envolvido;
- Número de clientes que participaram do teste;
- Número de conexões e transferência com problema;
- Número de transações bem-sucedidas distribuídas por tipo de requisição (HTTP 1.0, conexões persistentes, *pipeline*, SSL). É importante notar que o WebBench realiza testes com protocolos seguros;
- Média, valores máximos, mínimos e desvio padrão dos escores obtidos;
- Total e média de bytes transferidos, número de conexões por segundo, tempo médio de conexão, tempo médio de transferência e tempo médio de latência.

3.2 Benchmarks

Neste tópico serão apresentados alguns resultados de benchmarks executados por empresas e organizações relativos a avaliação do serviço WWW.

O objetivo aqui não é ratificar ou retificar os resultados apontados e sim verificar a forma de como eles foram obtidos, quais os métodos, que ferramentas usaram e quais as métricas consideradas.

Uma referência interessante que apresenta resultados comparativos entre Windows NT e Linux foi apresentado pela Microsoft (<http://www.microsoft.com/brasil/ntserver/linux.htm>), baseando suas informações em uma reportagem da revista PC Magazine de junho de 1999.

O comparativo foi feito entre o servidor web IIS da Microsoft e o servidor Apache, utilizando-se para isto as ferramentas WebBench 3.0 e SPECWeb, apresentadas anteriormente. Foram considerados três blocos para a avaliação:

1. Desempenho com conteúdo dinâmico: Foi considerado o número de requisições por segundo ao executar API e CGI no servidor web. No

caso do IIS a API utilizada foi a ISAPI e o Linux não apresentou dados nesta sessão.

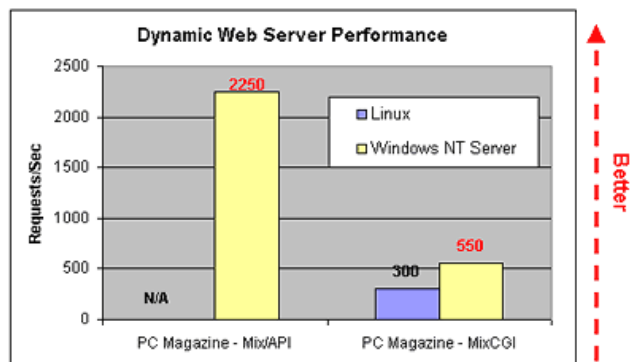


FIGURA 3.1 – Desempenho conteúdo dinâmico

- Desempenho com aplicações E-Commerce (SSL): Com o crescimento do comércio eletrônico, comparativos de servidores web executando protocolos seguros como o SSL se tornam de grande importância. Também foram realizados testes com ISAPI e CGI.

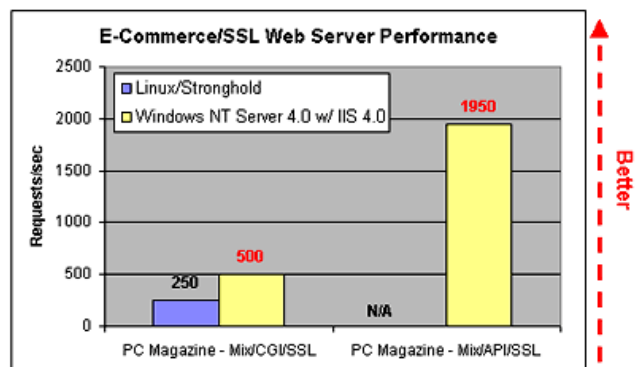


FIGURA 3.2 – Desempenho aplicações E-Commerce

- Desempenho com conteúdo estático: Obteve-se o número de conexões por segundo através da métrica SPECWeb.

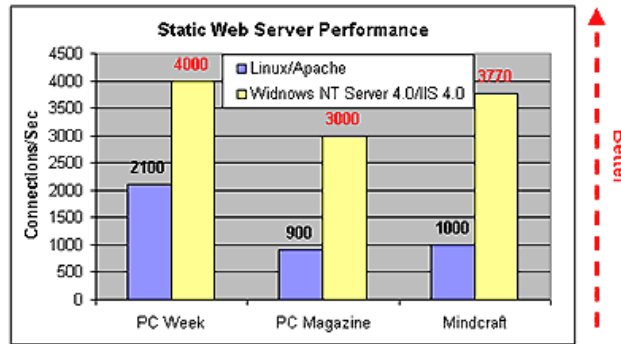


FIGURA 3.3 – Desempenho conteúdo estático

Além destes resultados de desempenho, uma série de outras métricas foram utilizadas para a avaliação dos servidores web. Abaixo são apresentadas as mais relevantes para este trabalho:

- Custos do sistema operacional e do servidor web.
- Relação preço / desempenho: É a medida de custo total do sistema por unidade de desempenho e serve para avaliar a eficiência dos sistemas operacionais.
- Confiabilidade: Funcionamento permanente garantido.
- Escalabilidade: Habilidade de crescer para dar suporte a mais usuários e a maiores demandas de carga de trabalho.
- Segurança: Mecanismos para garantir a segurança de acesso aos dados.
- Custo total de propriedade: Custos de implantação e manutenção.
- Disponibilidade de aplicativos: Número de aplicativos disponíveis e integrados ao serviço.
- Suporte: Auxílio na resolução de problemas.
- Facilidade de uso: Tempo gasto para aprender, instalar e gerenciar o serviço.

Em outra comparação, realizada pela empresa Mindcraft, chamada “Open Benchmark: Windows NT Server 4.0 and Linux”, foram utilizadas as ferramentas WebStone e WebBench [WEI 99].

Foram consideradas as métricas de desempenho temporal “número de requisições por segundo que o servidor pode atender”, além do “número de bytes por segundo” que foram enviados aos clientes durante o período de testes. Duas informações importantes são colocadas no relatório, com relação aos resultados obtidos nos gráficos:

- Pico de desempenho: Indica o número máximo de requisições por segundo atendidas pelo servidor web e o pico de vazão que foi gerado.

- Formato da curva de desempenho: O tempo que o servidor web demora para atingir o pico de desempenho depende basicamente de hardware e do sistema de testes, podendo ser ignorada. Já o formato da curva após ter se chegado no período de pico mostra como o servidor web executa com carga elevada. Se o desempenho cai de forma brusca durante um período, os usuários podem ter tempos de resposta lentos e imprevisíveis. Por outro lado, se o desempenho se degrada de forma gradual depois do pico o desempenho será mais previsível quando o servidor for submetido a cargas pesadas.

Em outro teste, realizado pela PC Magazine em Maio de 1999, fez-se a comparação entre diversos sistemas operacionais com variados servidores web. A ferramenta usada foi novamente o WebBench.

Foram usados 56 clientes em quatro segmentos de rede isolados de 100 Mbps para “estressar” os sistemas sob teste. O *benchmark* contou com a participação dos vendedores para que acertos específicos fossem feitos.

Utilizou-se apenas dois servidores, um Compaq e uma UltraSPARC, para tentar minimizar os efeitos de diferença de hardware. Em cada teste, somente um disco foi utilizado e configurou-se o *cache* dos servidores web para valores elevados para que toda a carga gerada ficasse em memória.

Os testes apresentados anteriormente tiveram como principal enfoque o desempenho. Outro aspecto fundamental para análise de servidores web é a disponibilidade.

Em um relatório apresentado por Jürgen Schmidt denominado “*To be or not to be up – Analysis of Web Server Downtimes*” esta questão foi explorada e testada [SCH 00]. Uma das primeiras constatações feitas é de suma importância. É afirmado que existe uma série de preceitos que se assume, como por exemplo de que Linux com Apache é mais estável que Windows NT executando IIS, e que são raramente apresentados com testes e dados concretos.

Com o objetivo de apresentar resultados devidamente comprovados, foram realizados testes com 100 servidores que recebem acessos freqüentes, indo de um milhão por mês até o mesmo número de acessos por dia. Concorrentes de competidores deste porte estão a um clique de distância, ou seja, problemas de disponibilidade são críticos para garantir seu sucesso.

Dentro destes 100 sites verificou-se através do serviço Netcraft (www.netcraft.com) quais servidores web estavam sendo utilizados. Uma constatação interessante apontada é de que apenas 10 sites executavam Windows NT, enquanto 58 SUN Solaris, 29 Linux e 3 com outras soluções.

O equipamento utilizado para os testes foi uma máquina com Linux e a duração foi de 32 dias corridos, sendo que a cada 10 minutos era disparado um script em Perl que verificava a disponibilidade de cada site considerado.

Este script enviava uma requisição HTTP GET para a página em teste e se não houvesse resposta por 3 minutos era contabilizada uma falha. Para eliminar problemas de rede que estivessem influenciando nos resultados, como, por exemplo, o *site* estar disponível, mas a conexão falha, outro script executava regularmente verificando o último ponto de acesso antes de chegar ao servidor web em questão. Se este script apontasse algum problema, a falha não era registrada, pois caracterizava problema de rede.

Os resultados apresentam o período em que os sistemas não estavam respondendo as requisições e a divisão básica feita foi por sistema operacional. Durante o período de teste de 32 dias, os servidores rodando Windows NT ficaram indisponíveis por 15 horas ou 1,9% do tempo. Já nos sistemas rodando Linux a média foi de 4 horas ou 0,5%. O Solaris apresentou um resultado ainda melhor, com 2,5 horas ou 0,3%.

Outra métrica resultante do teste foi o tempo médio de duração das falhas. O Linux em média ficava indisponível por 23 minutos, o Solaris por 25 minutos e o NT por 53 minutos. Além destas, o número de falhas também foi registrado.

O Windows 2000 fez parte dos testes e apresentou um resultado muito melhor do que o NT podendo competir diretamente com o Unix e Linux. Estes resultados não foram considerados, pelo número escasso de servidores e também para evitar distorções (como, por exemplo, o servidor estar em constante monitoramento, visto que o sistema operacional era recente na época do teste).

3.3 Conclusão

O capítulo 3 apresentou três ferramentas de avaliação do serviço WWW: SPECweb99, WebStone e WebBench. Estas representam, de forma geral, as diversas opções existentes no mercado, sendo que muitas são ferramentas comerciais. Os próprios produtores de servidores web, como Microsoft, através de uma ferramenta denominada WAS (*Web Stress Tool*) e Apache possuem opções para avaliação de desempenho do serviço HTTP.

Também foram apresentados alguns *benchmarks*, com a finalidade de verificar métodos e métricas utilizadas.

4. Método para Avaliação de Servidores WWW

Este capítulo está dividido em dois tópicos, a descrição de um método e passos a serem realizados para a avaliação de servidores WWW e a proposta de conjuntos de métricas abrangendo os diversos aspectos do serviço web, incluindo desempenho temporal, disponibilidade e confiabilidade, administração e recursos.

Existem descritas algumas propostas de métodos para a avaliação de desempenho que englobam de forma genérica qualquer medição e técnica que se pretenda usar, como avaliação de um processador, sistema operacional ou banco de dados.

O método descrito em [JAI 91] apresenta dez passos que devem ser seguidos para uma avaliação de desempenho que evite os erros comuns quando se propõem trabalhos desta natureza. Incurrer nestas falhas pode comprometer a avaliação, sendo que algumas recomendações são indicadas para obter bons resultados:

- Definir com clareza os objetivos das medições;
- Evitar objetivos tendenciosos, visando a favorecer determinado sistema ou produto;
- Pleno entendimento do problema a ser tratado;
- Uso de métricas relevantes ao que se quer verificar;
- Definição de um *workload* adequado ao problema;
- Uso de técnica apropriada;
- Entendimento da lista de parâmetros que afeta os resultados do sistema;
- Projeto do ambiente de avaliação adequado;
- Realização de análise e interpretação dos dados medidos;
- Clareza na apresentação dos resultados e na análise.

Baseado nos passos definidos em [JAI 91] para um método genérico, esta dissertação propõe um método com sete etapas para a realização de uma avaliação de servidores web, sendo que a utilização e correto planejamento da avaliação é fundamental para atingir as metas propostas e evitar os erros comuns a estes processos e atentar as recomendações, como as apontadas anteriormente.

4.1 Descrição do Método Proposto

O restante desta seção descreve o método proposto neste trabalho, visando especificamente à avaliação de servidores WWW.

4.1.1 Definição dos Objetivos

O primeiro passo em qualquer projeto de avaliação de desempenho é o estabelecimento dos objetivos do estudo e a definição dos limites do sistema a ser analisado.

Tomando-se o mesmo conjunto de hardware e software, a definição do sistema pode variar de acordo com as metas estabelecidas para o estudo e o escopo do trabalho. Tomando como exemplo a avaliação do serviço web, pode-se analisar o impacto de uso de *cache* agregado ao servidor ou o impacto de tempos de resposta de uma determinada tecnologia de rede sobre o serviço.

As métricas, bem como a carga de trabalho usada para comparar os sistemas, dependem destas definições.

4.1.2 Seleção das Métricas

O próximo passo é selecionar os critérios de comparação. Estes critérios são as chamadas métricas e em geral estão relacionadas com desempenho temporal, disponibilidade e confiabilidade. No caso de uma avaliação de rede, por exemplo, os indicadores normalmente usados são a vazão e atraso (desempenho temporal), a taxa de erros (confiabilidade) e a disponibilidade no envio de pacotes.

Para a comparação de sistemas pode-se também estabelecer medidas relacionadas com as funcionalidades oferecidas. Normalmente envolvem o conhecimento das características do que está sendo avaliado e pode ser fator primordial no momento da tomada de decisão. Pode-se, por exemplo, optar por um serviço que na avaliação não é o de melhor desempenho, mas que apresente as maiores facilidades de utilização ou tenha uma maior escalabilidade.

No caso de serviço web, dependendo do escopo definido, tem-se uma série de métricas possíveis de serem usadas. Na seção 4.3 é proposto um conjunto de indicadores envolvendo os aspectos de desempenho temporal, disponibilidade, confiabilidade, administração e recursos.

4.1.3 Levantamento de Informações e Definição dos Parâmetros

Após a definição das métricas que serão empregadas, o entendimento e a

formalização dos parâmetros que interferem na avaliação é fundamental. As definições de hardware e software constituem basicamente os parâmetros de sistema, enquanto que os parâmetros do *workload* estão relacionados com as interferências na medição. Pode-se ter como resultado deste passo uma lista indicando a relação dos resultados com a capacidade de processamento do servidor ou clientes que estão sob avaliação, a interferência da velocidade da rede ou a execução de processos paralelos durante a medição.

Faz parte desta etapa um levantamento de informações do ambiente real que servirá de base para a avaliação. É importante este entendimento para poder identificar as interferências e definir a forma com que serão conduzidas as medições. Trazendo para o objeto deste trabalho, definições como o número de requisições em um período de tempo, número máximo de clientes, natureza das requisições, entre outras, são pré-requisitos ao estabelecimento do ambiente para a avaliação e serão a base para a caracterização do *workload*.

4.1.4 Seleção e Caracterização do *Workload*

O *workload* consiste em uma lista de requisições de serviço ao sistema. Para exemplificar, o *workload* para a comparação de sistemas gerenciadores de banco de dados consiste em um conjunto de acessos a base de dados. Dependendo da técnica de avaliação escolhida, o *workload* poderá ser expresso de formas diferentes. No caso de *benchmark*, técnica proposta neste trabalho, a carga de trabalho deve ser bem conhecida, devendo ter sido projetada para exercitar aquelas características que se deseja medir.

Deve ocorrer, com base no passo anterior, uma caracterização do *workload*, ou seja, o estabelecimento de parâmetros que reproduzam de alguma forma o que se quer medir, o mais próximo possível do ambiente real.

4.1.5 Montagem do Ambiente de Teste e Medições

Depois de clarear os objetivos, ter definidas as métricas e escolhida a ferramenta para reproduzir a carga de trabalho levantada no ambiente real, pode-se passar para a etapa de montagem do ambiente de teste.

Neste ambiente serão realizados os testes e medições necessárias para a obtenção dos valores relativos aos indicadores escolhidos. O número de execuções deve ser suficiente para evitar distorções nos resultados. Distorções ocorrem normalmente com um número baixo de execuções ou quando os tempos estabelecidos são reduzidos.

4.1.6 Apresentação dos Resultados

O próximo passo da avaliação é a apresentação dos resultados, sendo importante que seja realizado de forma clara e o mais concisa possível. Isto implica normalmente na utilização de gráficos, tabelas resumidas e que se evite jargões demasiadamente técnicos.

4.1.7 Análise e Interpretação dos Dados

Nesta etapa os resultados são coletados e agregados de forma a possibilitar a análise dos mesmos. A interpretação dos dados de uma análise produz resultados, mas não as conclusões. Os resultados servirão de base para os tomadores de decisão chegarem as conclusões.

Também é importante ressaltar que os resultados das medições podem ser diferentes em cada execução e podem variar quando mais de uma categoria de informações estão sendo consideradas, como no caso proposto para servidores web. Por isso é importante que o tratamento dos dados resultantes da avaliação, tenha sido realizado adequadamente na etapa anterior.

4.2 Proposta de Critérios de Avaliação

Segundo [JAI 91], para cada estudo de avaliação de serviços, um conjunto de critérios ou métricas deve ser escolhido.

Uma forma de preparar este conjunto é listar os serviços oferecidos pelo sistema, sendo que para cada requisição de serviço feita pelo sistema há várias possibilidades de saída. Geralmente as saídas podem ser classificadas em três categorias, como apresentado na figura 4.1, onde o sistema pode executar o serviço corretamente, incorretamente, ou não executar.

Se o serviço executa corretamente, o desempenho é medido pelo tempo levado para executá-lo, pela taxa em que é executado e pelos recursos consumidos durante seu processamento. Estes três itens relacionam-se com as seguintes métricas, respectivamente: tempo de resposta, produtividade (vazão) e utilização.

Se o sistema executa o serviço incorretamente, um erro ocorre. É interessante classificar os erros e determinar a probabilidade de ocorrência de cada classe de erros considerada.

Caso o sistema simplesmente não execute o serviço, este pode estar em uma situação de queda, falha ou indisponível.

As métricas relacionadas com as três saídas apontadas anteriormente, serviço executado com sucesso, erro e indisponibilidade são também denominadas velocidade (desempenho temporal), confiabilidade e disponibilidade.

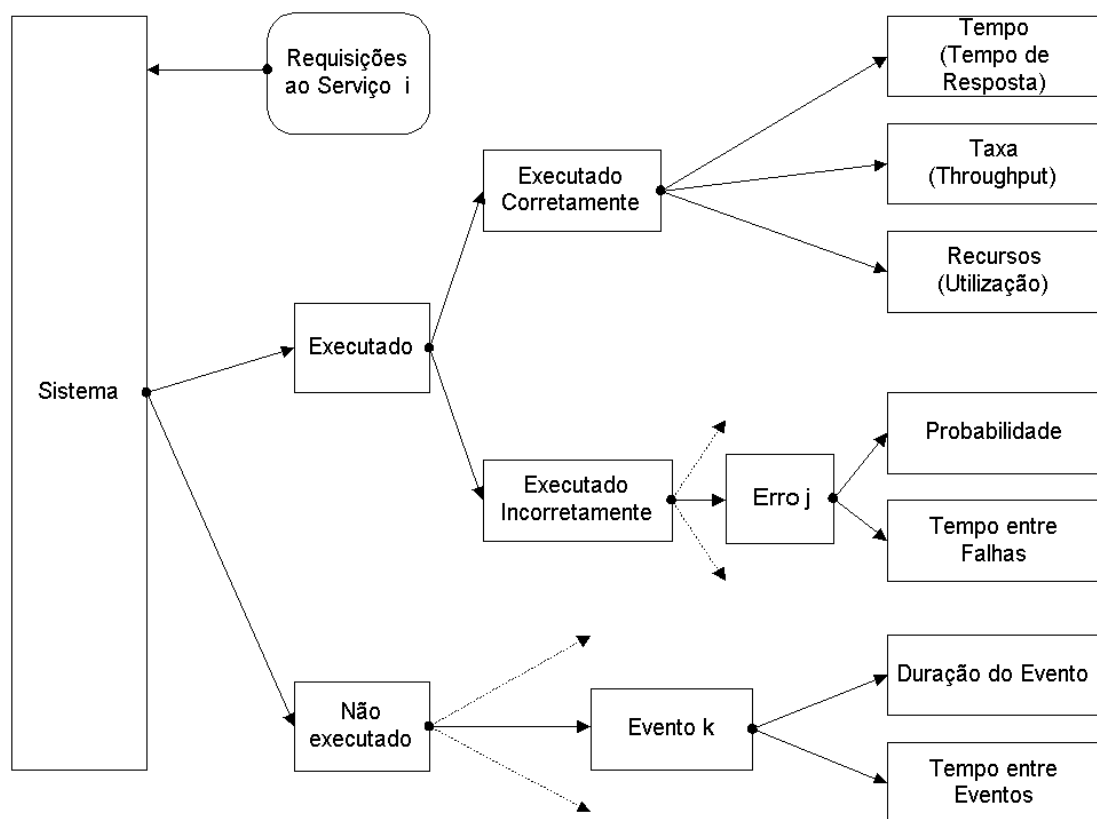


FIGURA 4.1 – Possibilidades de saídas em uma requisição de serviço
 Fonte: *The Art of Computer System Performance Analysis*. Pág. 33

Dentro deste contexto genérico, quando tratamos de desempenho temporal em ambiente web, as duas métricas mais importantes são: vazão e latência [MEN 98].

A taxa com que as requisições HTTP são atendidas representa a vazão de conexão. É comumente expressa em termos de operações HTTP por segundo, mas devido a grande variedade de tipos e tamanhos de objetos web, a vazão também é

medida em bits por segundo (bps). O tempo necessário para que a requisição seja completa determina a latência ou tempo de resposta.

O tempo de resposta percebido pelo cliente inclui a latência do servidor, ou seja, o tempo que o servidor leva para processar a requisição, somada ao tempo gasto para a comunicação via rede e mais o tempo utilizado pela máquina cliente para processar a resposta recebida.

Então, o desempenho depende de uma série de fatores: a capacidade do servidor, a carga de rede, a banda disponível, sistema operacional e serviço www, bem como o poder de processamento da máquina cliente.

Outra métrica bastante utilizada é o número de *hits*, o qual representa qualquer conexão, incluindo erros e solicitações não atendidas. Um *hit* que retorna um documento com sucesso é denominado uma requisição. Uma série de requisições de um usuário em um período limitado de tempo é conhecida como visita. O número de *hits*, como também o número de visitas tem sido utilizado para expressar a popularidade de um *site*.

Quando tratamos de disponibilidade, ou seja, a medida de sucesso com a qual o sistema atende a sua especificação em um período de tempo considerado [PRA 96], outras métricas devem ser consideradas. As principais são o tempo médio para a ocorrência de falhas (*Mean Time To Failure – MTTF*) e a disponibilidade calculada como percentual do tempo em que o sistema esteve em funcionamento durante o período considerado.

A confiabilidade de um sistema define a continuidade do serviço ou medida de sucesso com o qual este atende continuamente a sua especificação [PRA 96]. Os indicadores mais utilizados são o tempo médio entre falhas (*Mean Time Between Failures – MTBF*) e o percentual de ocorrência de erros.

4.3 Métricas Propostas

Abaixo são descritas as métricas que estão sendo propostas neste trabalho para servir como base para a avaliação de um servidor web em seus diferentes aspectos:

Tempo de resposta: É definida como o intervalo entre a requisição do usuário e a resposta do sistema. Esta definição é genérica e pode ser desmembrada em outras duas: intervalo entre o final da submissão de uma requisição e o início da resposta correspondente do sistema ou o intervalo entre o fim da submissão de uma requisição e o fim da resposta correspondente do sistema. Ambas definições são aceitáveis, mas a segunda é preferível se o tempo entre o início e o fim da resposta é alto.

É uma métrica que permite avaliar a qualidade do servidor WWW. O tempo de resposta também pode ser subdividido em tempo de resposta do servidor, da rede e do equipamento do cliente.

Vazão: É definido como a taxa (requisições por unidade de tempo) em que as requisições podem ser atendidas pelo sistema. Para sistemas batch, a vazão é medida em *jobs* por segundo. Para sistemas interativos a medição é em requisições por segundo. No caso de redes de computadores, a vazão é dada em pacotes por segundo (pps) ou bits por segundo (bps). Em processamento por transações a medida é transações por segundo (tps).

Como exposto anteriormente, vazão é uma das principais métricas para avaliação de servidores web. Permite determinar a velocidade com que as requisições são atendidas e, portanto, como o servidor está tratando as diversas solicitações de serviço.

A vazão pode ser dividida em vazão total e a vazão de rede. Como o que se busca neste trabalho é avaliar a qualidade do serviço WWW, pode-se considerar a vazão total como sendo o resultado final que o usuário observa.

Taxa de Custo / Desempenho: É uma métrica interessante para comparar dois sistemas. O custo inclui hardware, software, instalação e manutenção em um número de anos. O desempenho é medido em termos de sua vazão, considerando uma dada restrição de tempo de resposta.

Confiabilidade: É medida pela probabilidade de erros ou o intervalo de tempo entre erros (MTBF – *Mean Time Between Failures*), também com frequência especificada como segundos livres de erros.

Através deste indicador pode-se comparar serviços diferentes e indicar o que está mais apto a atender requisições conforme sua especificação durante o maior tempo possível.

Disponibilidade: É definido como a fração de tempo que o sistema está disponível para atender as requisições do usuário. O tempo em que o sistema não está disponível é chamado *downtime* e o tempo em que está ativo *uptime*. Uma medida que se usa com frequência é denominada MTTF (*Mean Time To Failure*).

Esta métrica é importante para determinar a prontidão para uso. Servidores indisponíveis podem deixar usuários sem informações importantes e acarretar prejuízos enormes em sites de comércio eletrônico ou na imagem de empresas.

Administração e Recursos: Grupo de métricas que tem como objetivo caracterizar os serviços web analisados nos aspectos de administração e facilidades oferecidas aos responsáveis pelo funcionamento do serviço e os recursos que o servidor web oferece, bem como, características inerentes ao produto.

A seguir, são propostos itens concentrados em grupos para nortear uma

avaliação de servidores web.

A divisão proposta inclui 5 grupos principais: Informações Gerais, Desempenho Temporal, Confiabilidade / Disponibilidade, Administração e Recursos. Estão detalhados na tabela a seguir.

TABELA 4.1 – Métricas para avaliação do serviço WWW

Métricas	
Informações Gerais	
Produto	Nome do servidor web a ser avaliado
Fabricante	Fabricante / Desenvolvedor do produto
Plataformas Suportadas	Hardware e Sistemas operacionais suportados pelo servidor web
Preço	Preço de venda
Desempenho Temporal	
Vazão do servidor	Detalhado anteriormente. Dado em bits por segundo e Requisições por segundo.
Tempo de resposta ou Latência	Tempo gasto pelo servidor para atender as requisições
Taxa de Custo / Desempenho	Relação entre o custo total e a vazão
Confiabilidade / Disponibilidade	
Duração do teste	Tempo em dias de duração dos testes
Período em que o sistema está operacional (em horas)	Base para cálculo do percentual de disponibilidade
Percentual de disponibilidade	Relação entre o período em que o sistema está operacional (em horas) pelo tempo de duração do teste.
Tempo médio de duração das falhas	Relação entre o somatório dos tempos de duração das falhas e o número total de falhas.
MTBF – <i>Mean Time Between Failures</i>	Tempo médio entre a ocorrência de falhas

Probabilidade de ocorrência de erros	Relação entre número de transações com problema e o número total de transações
Administração	
Possuir ferramenta gráfica de administração (sim/não)	Facilitar as atividades de administração
Possibilidade de administrar via web (sim/não)	Permitir administração remota
Administração via linha de comandos (sim/não)	Possibilitar a execução de scripts de administração
Formatos e possibilidades de logs (descritivo)	Os logs permitem a detecção de problemas e verificação de eventos, bem como, gerar estatísticas de utilização
Permite o estabelecimento de classificação do <i>site</i> , como nudez, violência, sexo, e linguagem ofensiva (sim/não)	Permitir a seleção de <i>sites</i> pelo browser de acordo com o conteúdo
Permite a criação de <i>headers</i> HTTP customizados (sim/não)	Possibilitar a personalização de <i>web sites</i>
Permite administrar expiração de conteúdo (sim/não)	Permitir flexibilidade com serviços de <i>cache</i>
Permite limitar o número de conexões (sim/não)	Limitar o número de conexões e evitar sobrecargas no servidor
Customização de mensagens de erro (sim/não)	Personalizar mensagens de erro e traduzí-las para o português
Recursos	
Sistemas Operacionais Suportados (descritivo)	Verificar se o serviço é multiplataforma
Suporte a CGI, API e linguagem de script (descritivo)	Verificar quais interfaces com aplicações são oferecidas
Suporte a SSL (sim/não)	Permitir conexões seguras através de criptografia via SSL
Suporte a HTTP 1.1 (sim/não)	Oferecer inovações introduzidas com o protocolo HTTP 1.1
Integração com outros serviços – FTP, SMTP, Monitor transações (descritivo)	Oferecer uma gama de serviços acessórios aos usuários e desenvolvedores
Integração com ferramenta de pesquisa (sim/não)	Permitir pesquisas nas páginas catalogadas

Isolamento de processos (sim/não)	Verificar se é possível isolar aplicações web em espaço de memória isolados
Suporte a múltiplos <i>web sites</i> (sim/não)	Possibilitar a hospedagem de diversos domínios em um mesmo servidor
Controle de uso de banda e processador (descritivo)	Verificar a possibilidade de estabelecer banda e tempo de processador para os diversos <i>sites</i> hospedados
Métodos de autenticação e controles de acesso suportados (descritivo)	Autenticar usuários, permitindo acesso autorizado a <i>sites</i> e diretórios / arquivos
Forma de execução - processos múltiplos ou <i>threads</i> (descritivo)	Permitir identificar como o sistema operacional trata o serviço web

4.4 Conclusão

Este capítulo apresentou uma proposta de método para avaliação de servidores web e os passos necessários para a sua execução. Em seguida foram discutidos os critérios de avaliação e formas de analisar um sistema que está sendo avaliado, com a finalidade de estabelecer métricas relevantes. Por fim, foram estabelecidos cinco grupos de métricas para a avaliação de servidores web, sendo eles: Informações Gerais, Desempenho Temporal, Confiabilidade / Disponibilidade, Administração e Recursos.

O método proposto foi inspirado em [JAI 91], mas as especificidades foram definidas em função da literatura e da experiência do mestrando, enquanto administrador de serviços web na Procempa.

5. Aplicação no Contexto da Procempa

Neste capítulo o método e métricas descritos anteriormente serão inseridos no ambiente corporativo, mais especificamente na Companhia de Processamento de Dados do Município de Porto Alegre - Procempa. Os dados da empresa são utilizados como base para a definição do objetivo e escopo, como também para as medições realizadas.

A estrutura do capítulo 5 segue a seqüência de passos proposta no método descrito no capítulo anterior.

5.1 Definição dos Objetivos

A Companhia de Processamento de Dados do Município de Porto Alegre – PROCEMPA é a empresa responsável por atender as demandas de tecnologia de informação da Prefeitura de Porto Alegre, abrangendo as diversas secretarias, órgãos e empresas municipais. O serviço oferecido vai desde o desenvolvimento de sistemas até o planejamento e manutenção da rede de computadores que conta hoje com mais de 3.500 pontos.

Dentre os serviços oferecidos estão o desenvolvimento de aplicações web e criação / manutenção do portal da Prefeitura.

Em 1995 foi criado um novo serviço, com enfoque diferente do atendimento mencionado à Prefeitura, de provimento de acesso Internet. Este serviço é chamado PortoWeb e atende hoje a mais de 10.000 clientes, dentre acesso discado, via *cablemodem*, ADSL, linha dedicada, conexão por fibra e hospedagem de páginas web.

A infra-estrutura web da empresa é composta por servidores com sistema operacional Windows NT, que constitui o principal ambiente, e servidores Linux para demandas específicas.

A estrutura atual centraliza em um servidor de porte a grande maioria das páginas hospedadas pelo provedor, incluindo o portal da Prefeitura, as páginas e aplicações do PortoWeb e o serviço de hospedagem de *sites* de terceiros.

Abaixo a descrição de hardware, software e rede deste servidor específico.

TABELA 5.1 - Características do servidor web Procempa

Característica	Valor
Nome	POA
Fabricante / Modelo	Itautec Infoserver 3030
Processador	Pentium III 700 Mhz
Memória	512 Mbytes
Sistema Operacional	Windows NT 4.0 (<i>Service Pack 6a</i>)
Serviços	Internet Information Services, Serviços Web (FTP, <i>Index Server</i> , <i>Chat</i> , SMTP, MTS) Macromedia BackStage 2, Perl, PHP
Número de domínios virtuais	504
Rede	Placa <i>Ethernet</i> 100 Mbps. Executa em um segmento próprio de rede, isolado por <i>Switch</i>

O objetivo principal da aplicação do método no contexto da Procempa é avaliar de forma genérica os sistemas operacionais GNU-Linux e Windows 2000 com seus respectivos serviços web, Apache e IIS respectivamente. O método foi aplicado em um ambiente de teste, sendo que a carga de trabalho aplicada neste ambiente foi baseada nas informações reais de requisições web realizadas ao servidor da Procempa.

É importante ressaltar que não se pretende fazer ajustes ou instalações específicas nos sistemas operacionais e serviços sob teste, devendo-se utilizar a forma básica do serviço e instalações típicas, sem nenhuma compilação ou acerto específico. O trabalho de *tuning* [CUR 01] [KIL 98] de serviço web pode se constituir em um trabalho a parte.

A técnica empregada para a avaliação de desempenho temporal escolhida é o *benchmark*, a qual constitui-se na aplicação de uma carga de trabalho (*workload*) sobre os sistemas. Esta carga de trabalho deve ser bem conhecida, ou seja, deve ter sido projetada para exercitar aquelas características que se deseja medir [NET 01].

5.2 Seleção das Métricas

A apresentação de um conjunto de métricas foi proposta no capítulo 4.3 e são estes os indicadores que basearam a avaliação no ambiente de teste.

Os indicadores de disponibilidade / confiabilidade não foram objeto de medição desta dissertação. Sua aplicação exige estudos específicos e implicam em um trabalho a parte, envolvendo tempos de medições bastante elevados, dependendo da forma escolhida para a coleta de dados. Uma proposta simples e interessante pode ser encontrada em [SCH 00].

5.3 Levantamento de Informações e Definição dos Parâmetros

Tomando como base o servidor descrito na tabela 5.1, buscou-se informações relativas a estrutura utilizada, bem como levantamentos de páginas hospedadas e dados de desempenho deste servidor. Estas informações basearam a caracterização da carga utilizada nos testes de desempenho temporal.

Para este levantamento, foram utilizadas basicamente três ferramentas: Medidor de Desempenho nativo do sistema operacional Windows NT / Windows 2000, relatórios de utilização de páginas web gerados pelo software *Site Server* da Microsoft e os dados do próprio sistema de arquivos do sistema operacional.

Estes dados mostram a situação do servidor em um período de tempo, visto que o número de páginas, tamanhos e até mesmo desempenho são alterados com o crescimento do número de *sites* hospedados e com o emprego de técnicas e softwares diferenciados no decorrer do tempo.

O levantamento foi realizado entre Julho e Outubro de 2001. Destaca-se nestas informações o alto índice de utilização de três domínios específicos, sendo o site do PortoWeb o mais acessado, seguido pelo portal da Prefeitura e da CEEE (Companhia de Energia Elétrica do Estado) .

A tabela abaixo mostra o número de páginas estáticas e dinâmicas do servidor, destacando o número de aplicações desenvolvidas com a tecnologia CGI, ASP e ISAPI.

TABELA 5.2 - Distribuição páginas estáticas e dinâmicas

	Páginas HTML	Páginas ASP	Aplicações CGI	Aplicações ISAPI
PortoWeb	4.625	271	2	1
Prefeitura	4.216	582	3	9
CEEE	288	290	0	0
Outros Clientes	25.300	2.805	8	0
TOTAL	34.429	3.948	13	10

Através dos dados coletados nos logs de acesso, com auxílio da ferramenta de extração de dados e geração de relatórios *Site Server*, obteve-se as informações da tabela 5.3. A distribuição dos três principais sites também fica evidenciada, onde se obtém 58,41% do número total de requisições.

Os dados de acesso representam três meses – Julho, Agosto e Setembro de 2001, sendo que os seguintes conceitos são empregados:

- Hit: Qualquer conexão ao *site*, incluindo requisições de imagens e retorno de erro.
- Requisições: Um hit com sucesso. Não incluem imagens e retorno de erro.
- Usuário: Qualquer pessoa que visitou o *site* no mínimo uma vez.

TABELA 5.3 - Informações de acesso a páginas no servidor

	Média de Hits / mês	Média de Requisições / mês	% Requisições	Média Visitas / mês	Média usuários / dia
PortoWeb	3.969.101	776.771	28,63	141.931	1.733
Prefeitura	3.347.239	703.051	25,91	163.928	2.250
CEEE	900.333	105.050	3,87	14.823	375
Outros Clientes	5.773.090	1.128.727	41,59	205.090	2.520

Através de um levantamento no sistema de arquivos do Windows NT, obteve-se os dados a seguir quanto ao tamanho das páginas estáticas.

TABELA 5.4 - Distribuição dos tamanhos de páginas estáticas

	0 – 5 Kbytes	6 - 10 Kbytes	11 – 100 Kbytes	101 – 1000 Kbytes	+ 1001 Kbytes
PortoWeb	2.995	462	944	224	0
Prefeitura	2.378	839	909	90	0
CEEE	244	21	23	0	0
Outros Sites	16.508	4.005	4.556	231	0
TOTAL	22.125	5.327	6.432	545	0

O anexo 1 apresenta o levantamento de informações completo do servidor principal da Procempa.

Quanto aos principais fatores ou parâmetros que influenciam a avaliação do serviço WWW, principalmente no aspecto desempenho temporal, podemos destacar:

- O desempenho do servidor;
- O desempenho das estações clientes;
- Velocidade e retransmissões da rede local;
- Tempo entre as requisições;
- Natureza das requisições (páginas estáticas ou dinâmicas);
- Número e tamanho das requisições e respostas;
- Cargas adicionais nos equipamentos envolvidos (outros serviços, não relacionados a web).

5.4 Seleção e Caracterização do *Workload*

Com base nas métricas propostas no item 4.3, buscou-se a identificação de uma ferramenta que permitisse alcançar os resultados que se desejava, ou parte deles, e que fosse viável de ser empregado no ambiente de teste disponível.

Optou-se pela ferramenta WebStone [MIN 99], visto que a mesma responde as métricas de desempenho temporal básicos como vazão e latência e ainda permite alguma análise com relação a erros encontrados durante o período de medições.

Outro fator decisivo para sua utilização é a natureza de código aberto e gratuito da ferramenta, apesar de não ter sido necessário alterar o código fonte obtido, com exceção da aplicação de uma correção exigida para a correta compilação e

aplicação no ambiente Linux de teste.

Para a obtenção das métricas de administração e recursos, foram utilizadas consultas aos manuais dos produtos, bibliografia disponível e experimentação direta na interface do serviço web analisado.

Fonte de informações do *Internet Information Services* (IIS) são [LIU 94] [CUR 01] e o Technet, conjunto de informações disponíveis em CD ou Internet que engloba questões técnicas dos produtos da Microsoft.

Já o Apache possui bibliografia específica [LAU 99] e o manual do produto que acompanha a instalação [APA 01] oferece dados essenciais para a identificação das métricas de administração e recursos.

Como colocado anteriormente, os indicadores de disponibilidade não foram objeto de medição desta avaliação.

Com base no levantamento de informações, pode-se definir a carga empregada nos testes de desempenho temporal.

Utilizou-se principalmente as informações obtidas através do levantamento dos tamanhos dos arquivos (páginas) do sistema de arquivos do Windows NT e dos indicadores do serviço web obtidas através de diversas medições de desempenho, utilizando-se a ferramenta monitor de desempenho.

O anexo 1 apresenta tabelas com os resultados de um levantamento de desempenho do servidor em questão, baseados nas recomendações presentes em [CUR 01]. Foram feitas quatro coletas de dados semanais envolvendo três grupos de métricas: memória, processador e serviço Web. Também são mostradas imagens do software utilizado, monitor de desempenho do Windows 2000, com a finalidade de ilustrar a ferramenta.

Os dados apresentados refletem o comportamento deste servidor e podem auxiliar na identificação de gargalos, sendo que este não é o objetivo do presente trabalho, mas são informações que servem de base para a caracterização da carga a ser empregada no ambiente de teste.

Algumas informações que merecem destaque são o número máximo de conexões, cujo valor apontado no levantamento foi de 592 e a média de conexões, a qual apresentou o resultado de 177. Esta média pode ser utilizada como ponto de partida para o estabelecimento do número de clientes em uma medição.

O peso das páginas estáticas (HTML) e páginas dinâmicas (CGI, ISAPI e ASP) também são relevantes nesta etapa. Utilizando-se o valor médio do número de requisições por segundo obteve-se um percentual de 89,66 % de páginas HTML, 0,03 % de aplicações CGI e 10,31% de aplicações ISAPI, considerando que as requisições ASP fazem uso de ISAPI em sua estrutura básica.

Da mesma forma, as médias de tamanho de páginas são informações de extrema importância na fase de caracterização de *workload*. Utilizando-se os dados da fase de levantamento de informações, pode-se, de forma concisa, chegar nos seguintes percentuais de utilização:

Arquivos entre 0 e 5 Kbytes	– 64,26 %
Arquivos entre 6 e 10 Kbytes	– 15,47 %
Arquivos entre 11 e 100 Kbytes	– 18,68%
Maiores que 101 Kbytes	– 1,58 %

5.5 Montagem do Ambiente de Teste e Medições

Conforme os objetivos propostos e a técnica escolhida, foi necessário estabelecer um ambiente de teste para instalar os sistemas operacionais e serviços web definidos e executar a ferramentas de medição e verificar funcionalidades.

Foram utilizados quatro computadores ao todo no ambiente e um *switch* para interconectá-los em rede local. Um dos equipamentos foi definido como servidor e três como clientes. A tabela abaixo relaciona as características de cada um deles.

TABELA 5.5 - Características dos equipamentos do ambiente de teste

Designação	Função	Hardware	Software
CORTE	Servidor Web	Pentium III 800 Mhz 128 Mbytes de Memória 20 Gbytes de Disco Placa de rede <i>Ethernet</i> 10 Mbps	Partição 1 - Windows 2000 com IIS (versão 5.0) Partição 2 - Linux RedHat 7.1 com Apache (versão 1.3.19)
WSCLI1	Cliente Web	Pentium 200 Mhz 64 Mbytes de Memória 8 Gbytes de Disco Placa de rede <i>Ethernet</i> 10 Mbps	Linux distribuição Conectiva 6.0
WSCLI2	Cliente Web	2 x Pentium 133 Mhz 64 Mbytes de Memória 10 Gbytes de Disco Placa de rede <i>Ethernet</i> 10 Mbps	Linux distribuição RedHat 7.1

WSCLI3	Cliente Web Webmas ter	Pentium III 800 Mhz 128 Mbytes de Memória 20 Gbytes de Disco Placa de rede <i>Ethernet</i> 10 Mbps	Linux distribuição RedHat 7.1
--------	---------------------------------	--	----------------------------------

Com a finalidade de eliminar parâmetros que poderiam influenciar nas medições, foi montada uma rede Ethernet própria, executando a 10 Mbits por segundo e ocorreu o monitoramento dos equipamentos durante o período de teste para evitar que o desempenho de algum cliente interferisse nos resultados obtidos.

Utilizou-se o mesmo equipamento como servidor, com mesmas características de hardware, para avaliar o serviço web do Windows 2000 e Linux, *Internet Information Services* e Apache respectivamente. Foram estabelecidas duas partições, sendo cada uma com um sistema operacional e, com isto, evitou-se variações de hardware que pudessem interferir na medição.

A distribuição utilizada no servidor Linux foi o RedHat 7.1 e o serviço web o Apache 1.3.19. A opção por esta distribuição decorre da mesma ser adotada nas instalações de servidores pela Procempa, sendo que a versão do Apache é a que acompanha o RedHat.

Outra medida que foi adotada com a finalidade de eliminar ou diminuir um dos principais fatores que influenciam a avaliação do serviço, conforme apresentado no item 5.3, foi a utilização de um *switch*, fazendo com que cada equipamento possuísse um segmento próprio com 10 Mbps.

Verificou-se nos primeiros experimentos que a adoção de um *hub* ocasionava uma taxa de colisão muito elevada, o que poderia comprometer os resultados da avaliação.

Não se fez ajustes ou alterações em parâmetros do sistema operacional ou serviço web, utilizando-se, portanto, um ambiente web padrão. Esta é a prática na maioria das instalações que são feitas no ambiente corporativo, apesar de ser desaconselhada devido as perdas de desempenho e, principalmente, riscos de segurança decorrentes desta alternativa. O mais indicado é a realização de um trabalho de *tuning* [CUR 01] [KIL 98] antes de colocar o ambiente web em produção, onde são realizados ajustes nos parâmetros do sistema operacional e serviço web para adequá-los a realidade das demandas que o servidor será submetido.

A figura 5.1 mostra o ambiente de teste montado para esta avaliação.

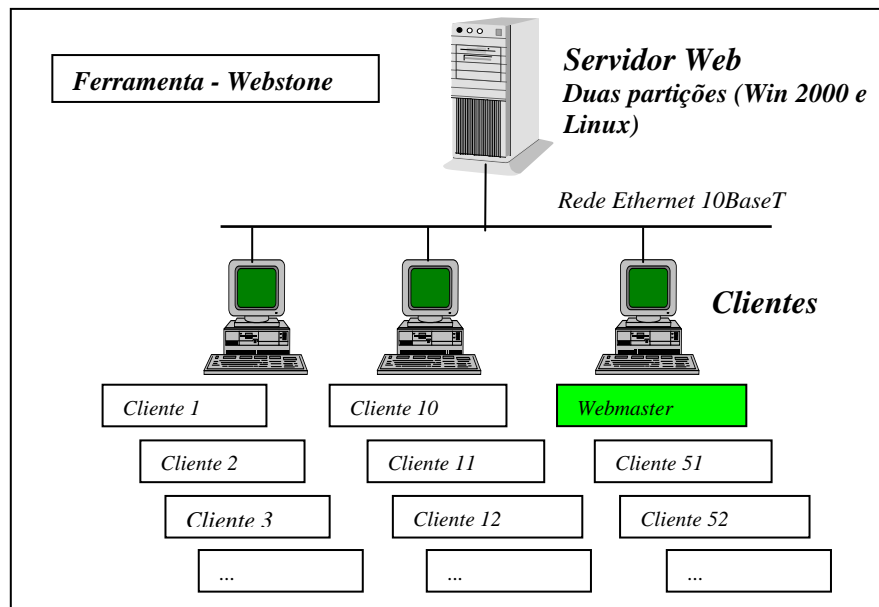


FIGURA 5.1 – Ambiente de teste

Como colocado anteriormente, utilizou-se a ferramenta WebStone para as medições referentes a desempenho temporal.

Os principais parâmetros de configuração da ferramenta são o número inicial de clientes, número máximo de clientes, incremento de cada execução, o número de interações do sistema e o tempo de cada execução.

O serviço é iniciado na estação cliente caracterizada como WebMaster, no caso da aplicação deste trabalho, a máquina denominada WSCLI3. O WebStone divide então o número de clientes estabelecido como parâmetro pelo número de máquinas envolvidas no teste. Como o experimento utilizou três máquinas clientes, incluindo o WebMaster, e o número inicial de clientes foi definido em 180, de acordo com a caracterização de carga, a divisão resultou em 60, número este que estabelece o número de processos que serão executados em cada máquina.

Os processos em questão iniciam as chamadas GET, utilizando HTTP 1.0, no servidor web e permanecem fazendo requisições durante o tempo estabelecido no arquivo de configuração para cada execução.

O arquivo LISTFILE contém todos os parâmetros de tamanho de páginas a serem utilizados, bem como o peso de aplicações estáticas (.html) e dinâmicas (CGI e/ou ISAPI).

No anexo 2 estão colocados os arquivos de configuração utilizados. Os parâmetros do arquivo LISTFILE que estabelecem o tamanho e a distribuição das aplicações baseou-se nos dados extraídos na fase de caracterização da carga de trabalho,

sendo que as bases para atribuição dos valores foram o levantamento dos tamanhos de páginas utilizados no servidor da Procempa, o número de páginas estáticas e dinâmicas do servidor, os dados do medidor de desempenho do serviço web, principalmente requisições a CGI, ISAPI e páginas estáticas.

Para os parâmetros do arquivo TESTBED, o qual determina o número de clientes mínimo e máximo, os dados referentes ao número de conexões foram a base para determinar os valores, sendo que o número máximo de conexões e número corrente de conexões foram os principais parâmetros considerados.

Ao terminar o tempo estabelecido para as requisições, o WebMaster busca os resultados de cada um dos clientes e consolida em um diretório chamado RUNS, onde os mesmos são armazenados a cada execução. Com o programa WSCOLLECT obtém-se um relatório com os resultados consolidados.

Para a execução seguinte, o número de clientes é acrescido do valor contido no parâmetro incremento e isto se repete a cada rodada. Ocorre novamente a divisão pelo número de máquinas, o novo número de processos é estabelecido e ocorre a inicialização de processos em cada máquina cliente.

Este processo de incremento ocorre até ser alcançado o número máximo de clientes estabelecido no arquivo de configuração.

Todo este processo corresponde a uma interação, sendo que o parâmetro número de interações determina quantas vezes este ciclo irá se repetir. No caso deste experimento foram estabelecidas três interações.

Além da utilização da ferramenta de *Benchmark* WebStone, o ambiente de teste serviu para verificação das funcionalidades dos serviços web no que tange a administração e recursos. Utilizou-se como fonte principal para obtenção dos resultados destas métricas a pesquisa em manuais e bibliografia, além da experiência do autor deste trabalho enquanto analista responsável pela administração de servidores web. Agregado a isto, a utilização do laboratório foi importante para verificar muitas das funcionalidades apontadas.

5.6 Apresentação dos Resultados

A apresentação dos resultados retrata os valores obtidos nas medições de desempenho temporal, obtidos com a ferramenta WebStone, e o levantamento das características dos servidores web avaliados quanto a administração e recursos.

Antes de serem apresentados os resultados propriamente ditos, estão descritos abaixo os indicadores fornecidos pelo WebStone:

- Taxa de conexões por segundo: Representa o número de conexões por segundo livre de erros que o servidor web conseguiu atender durante o período de testes. Este indicador também é comumente referenciado como operações http por segundo ou *hits* por segundo. Uma operação típica http envolve o estabelecimento de conexão (SYN, SYN ACK), a requisição do cliente (GET /index.html HTTP/1.0...), a resposta do servidor cotendo o *header* (HTTP/1.0 200 OK...), a resposta do servidor com os dados e a finalização da conexão (FIN, FIN ACK).
- Taxa de erros: Conforme o número de clientes aumenta, o servidor pode começar a recusar requisições. Isto significa que o servidor está ocupado atendendo as solicitações anteriores e os clientes devem parar de enviar novas requisições.
- Latência: Representa, conforme detalhado anteriormente, o tempo de acesso percebido pelo usuário. Deve ser confrontada com o número de erros gerados, pois pode ocorrer de apenas poucas conexões serem atendidas, devido a uma alta taxa de erros, com uma latência mais adequada do que um servidor que tem uma latência mais alta, mas apresenta pouca ou nenhuma taxa de erros.
- Fator de carga Little's: A lei de Little's é baseada na teoria de filas. Representa a taxa de tempo gasto na comunicação com o servidor web em comparação com o tempo que o servidor leva para responder. Em um teste utilizando o WebStone, o fator de carga Little é restrito aos processos clientes, sendo que quanto mais alto melhor. Quanto mais este indicador se aproximar do número de clientes estabelecidos, melhor é o resultado obtido.
- Vazão: Conforme detalhado anteriormente, é definido como a taxa em que as requisições podem ser atendidas pelo sistema. O WebStone não inclui os *headers* do TCP/IP, reportando o *header* http e os dados efetivamente enviados.

O resultado de uma execução típica está apresentado abaixo:

```

WebMaster name = wscli
Client 0: 192.168.1.2 .....# Processes: 107
Webserver: 192.168.1.1
Webmaster: 192.168.1.2:4091
Client 1: 192.168.1.3 .....# Processes: 107
Webserver: 192.168.1.1
Webmaster: 192.168.1.2:4091
Client 2: 192.168.1.4 .....# Processes: 106
Webserver: 192.168.1.1
Webmaster: 192.168.1.2:4091

Waiting for READY from 320 clients
All READYs received
Sending GO to all clients
All clients started at Sun Dec  2 21:00:20 2001

Waiting for clients completion...

```

Reading results				
.....				
.....				
All clients ended at Sun Dec 2 21:31:32 2001				
=====				
WEBSTONE 2.5b3 results:				
Total number of clients:	320			
Test time:	30 minutes			
Server connection rate:	54.50 connections/sec			
Server error rate:	0.00 err/sec			
Server thruput:	7.78 Mbit/sec			
Little's Load Factor:	317.63			
Average response time:	5.828 sec			
Error Level:	0.00 %			
Average client thruput:	0.02 Mbit/sec			
Sum of client resp. times:	571729.56 sec			
Total num. of pages read:	98095			
98095 connection(s) to server, 0 errors				
	Average	Std Dev	Minimum	Maximum
Connect time (sec)	1.251292	4.495606	0.000343	549.157205
Response time (sec)	5.828325	10.140654	0.012296	553.935103
Response size (bytes)	17842	64736	251	512337
Body size (bytes)	17528	64730	35	512000

Ao final de todas as interações, os relatórios individuais são consolidados em uma tabela como a que segue.

TABELA 5.6 – Relatório final apresentado pelo WebStone

Timestamp	Total Number of Clients	Server Connection rate	Little's Load Factor	Average Response Time	Error Level	Average Client Thruput
011202_2059	320	54.50	317.63	5.8280	0.0000	7.78

Os resultados obtidos com as medições feitas utilizando o WebStone estão sintetizados na tabela 5.7 e 5.8, onde estão colocados os levantamentos feitos com o *Internet Information Services* e Apache, discriminados por número de clientes fazendo acessos simultâneos. Estão separados em três conjuntos de resultados, sendo o primeiro utilizando apenas páginas estáticas como carga. No seguinte, os dois serviços web foram submetidos a uma carga de páginas estáticas, mescladas com páginas dinâmicas utilizando CGI. No último quadro, o serviço IIS foi submetido a cargas

envolvendo ISAPI, já que a tecnologia é exclusiva da Microsoft e não pôde ser aplicada no servidor executando Apache.

Os dados representam as médias de todas as execuções, em um número total de quatro, sendo cada uma composta de três interações. Obteve-se, portanto, doze execuções completas.

TABELA 5.7 – Resultados desempenho temporal IIS

Servidor Executando Internet Information Services (Windows)					
Páginas Estáticas					
Número Total de Clientes	Taxa de conexão ao servidor	Fator de carga Little's	Tempo de Resposta Médio	Taxa de erro	Vazão Média do Cliente
180	57,11	179,41	3,14	0,0000	8,14
210	57,13	208,94	3,65	0,0000	8,12
240	57,53	239,06	4,15	0,0000	8,14
270	57,23	268,68	4,69	0,0000	8,14
300	57,22	298,64	5,21	0,0000	8,15
330	57,16	326,74	5,71	0,0000	8,13
360	57,18	357,59	6,25	0,0000	8,11
390	57,76	387,46	6,70	0,0000	8,11
420	58,31	416,93	7,15	0,0100	8,10
450	58,26	447,48	7,68	0,0100	8,14
480	57,94	476,69	8,22	0,0100	8,10
510	58,03	504,06	8,68	0,0300	8,02
540	58,02	533,47	9,19	0,0300	7,95
570	57,45	564,09	9,81	0,0500	7,89
600	56,62	594,04	10,49	0,0500	7,77

Páginas Dinâmicas – CGI					
Número Total de Clientes	Taxa de conexão ao servidor	Fator de carga Little's	Tempo de Resposta Médio	Taxa de erro	Vazão Média do Cliente
180	57,19	179,49	3,13	0,0000	8,15
210	57,06	208,50	3,65	0,0000	8,13
240	57,62	239,05	4,15	0,0000	8,16
270	57,28	268,90	4,69	0,0000	8,14
300	57,19	298,62	5,22	0,0000	8,15
330	57,12	328,33	5,74	0,0000	8,13
360	57,31	355,03	6,22	0,0000	8,12
390	57,70	387,18	6,71	0,0000	8,13
420	58,14	416,03	7,16	0,0200	8,08
450	58,18	447,05	7,68	0,0200	8,10
480	57,89	473,33	8,19	0,0200	8,08
510	57,86	504,65	8,72	0,0200	7,99
540	57,99	534,86	9,22	0,0200	7,96
570	57,58	565,12	9,81	0,0500	7,87
600	56,64	594,27	10,49	0,0500	7,78
Páginas Dinâmicas – CGI e ISAPI					
Número Total de Clientes	Taxa de conexão ao servidor	Fator de carga Little's	Tempo de Resposta Médio	Taxa de erro	Vazão Média do Cliente
180	50,22	179,56	3,58	0,0000	8,25
210	50,44	203,80	4,04	0,0000	8,22
240	50,58	234,72	4,64	0,0000	8,23
270	50,05	268,64	5,37	0,0000	8,21
300	49,82	295,01	5,92	0,0000	8,22
330	50,40	325,43	6,46	0,0000	8,21
360	50,60	354,31	7,00	0,0000	8,20
390	50,95	385,56	7,57	0,0000	8,21
420	50,74	414,67	8,17	0,0100	8,19
450	51,00	439,28	8,61	0,0200	8,17
480	51,08	475,55	9,31	0,0100	8,14
510	50,97	503,76	9,88	0,0100	8,06
540	50,65	532,68	10,52	0,0100	8,00
570	49,91	557,64	11,17	0,0300	7,90
600	49,49	585,53	11,83	0,0300	7,83

TABELA 5.8 – Resultados desempenho temporal Apache

Servidor Executando Apache (Linux)					
Páginas Estáticas					
Número Total de Clientes	Taxa de conexão ao servidor	Fator de carga Little's	Tempo de Resposta Médio	Taxa de erro	Vazão Média do Cliente
180	44,28	177,64	4,01	0,00	7,40
210	44,13	206,83	4,69	0,00	7,39
240	44,20	235,43	5,33	0,00	7,39
270	44,22	264,09	5,97	0,00	7,36
300	44,20	292,45	6,62	0,00	7,35
330	44,89	321,31	7,16	0,00	7,34
360	44,94	348,64	7,76	0,00	7,32
390	45,09	376,69	8,35	0,00	7,31
420	45,20	404,93	8,96	0,00	7,32
450	45,38	434,88	9,58	0,00	7,33
480	45,28	464,52	10,26	0,00	7,34
510	45,49	490,08	10,77	0,00	7,31
540	45,56	524,25	11,51	0,02	7,35
570	45,37	551,02	12,15	0,02	7,35
600	45,22	581,05	12,85	0,05	7,36
Páginas Dinâmicas – CGI					
Número Total de Clientes	Taxa de conexão ao servidor	Fator de carga Little's	Tempo de Resposta Médio	Taxa de erro	Vazão Média do Cliente
180	51,46	176,41	3,42	0,0000	7,30
210	52,03	208,35	4,00	0,0000	7,32
240	52,04	236,66	4,54	0,0000	7,29
270	51,53	265,63	5,15	0,0000	7,25
300	51,20	293,65	5,73	0,0000	7,24
330	51,66	323,15	6,25	0,0000	7,25
360	52,43	349,64	6,66	0,0000	7,21
390	52,11	379,21	7,27	0,0000	7,23
420	52,83	407,39	7,71	0,0000	7,23
450	53,36	436,50	8,18	0,0000	7,22
480	52,77	459,30	8,70	0,0000	7,14
510	53,16	457,32	8,60	0,0000	7,16
540	52,96	519,42	9,80	0,0000	7,18
570	53,36	549,30	10,29	0,0200	7,16
600	53,43	583,46	10,90	0,0200	7,22

Algumas comparações são estabelecidas nas figuras 5.2 e 5.3. Estes gráficos servirão de base para a análise e interpretação dos dados.

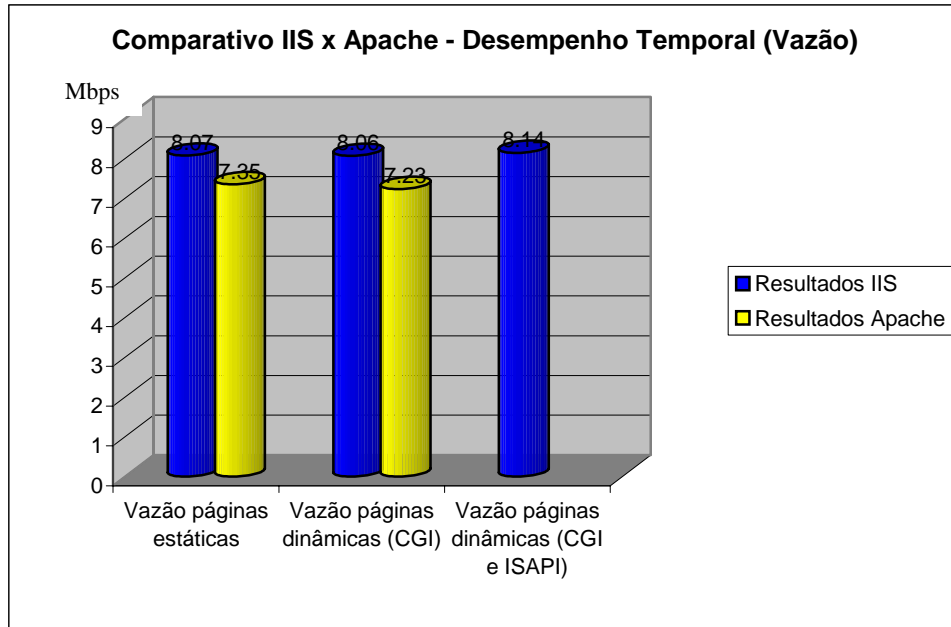


FIGURA 5.2 – Comparativo desempenho temporal (vazão)

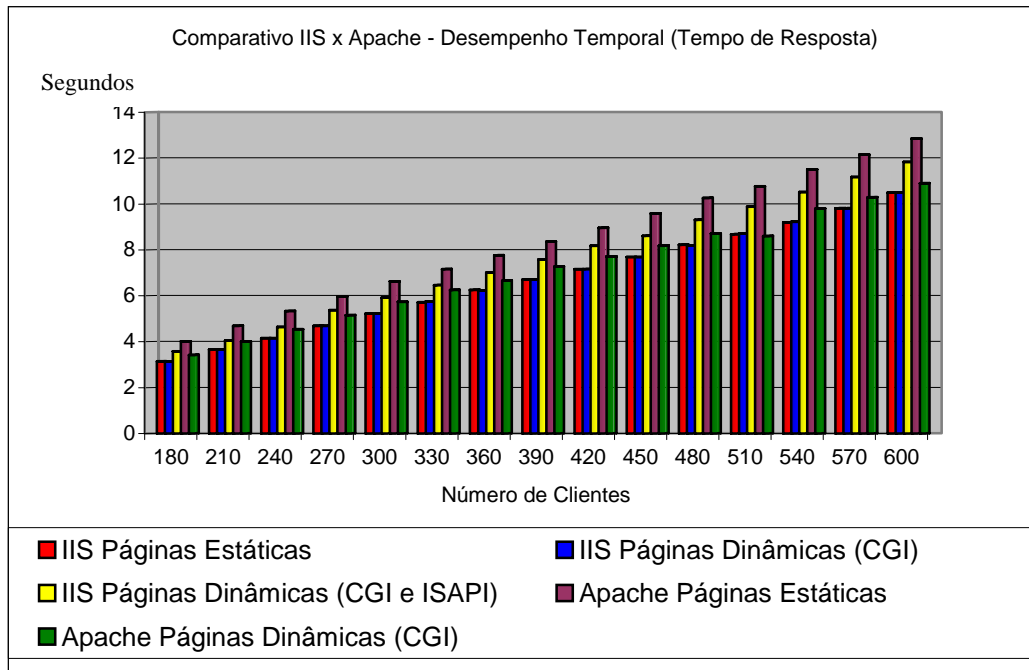


FIGURA 5.3 – Comparativo desempenho temporal (tempo de resposta)

Os resultados sintetizando estes indicadores de desempenho, com suas médias calculadas com base nos dados apontados anteriormente, e as métricas envolvendo administração e recursos estão apontados nas tabelas 5.9, apresentando os dados da avaliação do IIS executando em Windows 2000 e tabela 5.10, apontando os resultados do Apache em um sistema operacional Linux.

Uma observação importante que ajuda a explicar alguns resultados obtidos nas métricas de administração e recursos é a de que o Apache se constitui em um servidor modular, fazendo com que somente as mais básicas funcionalidades sejam disponibilizadas pelo núcleo do sistema. Características adicionais são obtidas pela instalação de módulos compatíveis no momento da compilação ou pela utilização dos chamados módulos de carga dinâmica que exigem apenas a compilação dos módulos a serem usados. Mais recentemente, o Apache incorporou uma característica chamada DSO (*Dynamic Shared Object*), o qual permite a carga dos módulos somente no momento de suas execuções.

A métrica relacionando taxa de custo e desempenho, bem como os indicadores de disponibilidade e confiabilidade, não foi apurada. Considerando a realidade da Procempa, os valores de treinamentos e salários pagos aos funcionários se equivalem tanto para o ambiente Windows, quanto Linux. Não existe divisão alguma de profissionais para um ambiente ou outro. Existe uma variação de hardware exigido para a execução do Apache, em comparação com o Windows, o qual exige uma máquina com configurações de processador e memória superiores. A questão relativa as horas necessárias para administração dos ambientes é outra variável, sendo esta de mais difícil mensuração.

TABELA 5.9 – Métricas para avaliação do serviço IIS

Métricas	
Informações Gerais	
Produto	Internet Information Services 5.0.
Fabricante	Microsoft Corporation.
Plataformas Suportadas	Windows 2000.
Preço	Acompanha o sistema operacional. Preço Windows 2000 (licenças para 15 usuários) = U\$ 740,00.

Desempenho Temporal	
Vazão do servidor no teste (Mbps)	Páginas Estáticas: 8,07. Páginas Estáticas e CGI: 8,06. Páginas Estáticas, CGI e ISAPI: 8,14.
Tempo de resposta (segundos)	Páginas Estáticas: 3,14 à 10,49. Páginas Estáticas e CGI: 3,13 à 10,49. Páginas Estáticas, CGI e ISAPI: 3,58 à 11,83.
Taxa de Custo / Desempenho	Não foi avaliado. Observações feitas anteriormente.
Confiabilidade / Disponibilidade	
Duração do teste	Não foi avaliado.
Período em que o sistema está operacional (em horas)	Não foi avaliado.
Percentual de disponibilidade	Não foi avaliado.
Tempo médio de duração das falhas	Não foi avaliado.
MTBF – <i>Mean Time Between Failures</i>	Não foi avaliado.
Probabilidade de ocorrência de erros	Não foi avaliado.
Administração	
Possuir ferramenta gráfica de administração (sim/não)	Sim, através do recurso MMC (<i>Microsoft Management Console</i>).
Possibilidade de administrar via web (sim/não)	Sim, através do IISAdmin.
Administração via linha de comandos (sim/não)	Sim. Possui API para os principais comandos do servidor web.
Formatos e possibilidades de <i>logs</i> (descritivo)	- <i>Logs</i> de erros e acessos; - Permite gerar arquivos de <i>logs</i> isolados para cada <i>web site</i> ; - Formatos suportados: W3C, NCSA e formato proprietário.
Permite o estabelecimento de classificação do site, como nudez, violência, sexo, e linguagem ofensiva (sim/não)	Sim. Suporte à PICS (<i>Platform for Internet Content Selection</i>).
Permite a criação de <i>headers</i> HTTP customizados (sim/não)	Sim, configurado na própria interface.

Permite administrar expiração de conteúdo (sim/não)	Sim, configurado na própria interface.
Permite limitar o número de conexões (sim/não)	Sim, configurado na própria interface.
Customização de mensagens de erro (sim/não)	Sim, configurado na própria interface.
Recursos	
Sistemas Operacionais Suportados (descritivo)	Windows 2000.
Suporte a CGI, API e linguagem de script (descritivo)	Suporte nativo a CGI, ISAPI e ASP. Permite utilização de outras alternativas como PHP e Java.
Suporte a SSL (sim/não)	Sim. Versões 2 e 3.
Suporte a HTTP 1.1 (sim/não)	Sim.
Integração com outros serviços – FTP, SMTP, Monitor transações (descritivo)	Por padrão, os serviços de FTP, índice e SMTP (para utilização como gateway) são disponibilizados. Inclui também um monitor de transações denominado MTS (<i>Microsoft Transaction Server</i>).
Integração com ferramenta de pesquisa (sim/não)	Sim, chamada <i>Index Server</i> .
Isolamento de processos (sim/não)	Sim. É possível executar processos na mesma área do serviço web ou isolá-lo em um processo a parte.
Suporte a múltiplos <i>web sites</i> (sim/não)	Sim.
Controle de uso de banda e processador (descritivo)	É possível determinar banda e quantidade do processador que cada <i>web site</i> pode consumir.
Métodos de autenticação e controle de acessos suportados (descritivo)	<ul style="list-style-type: none"> - Controle de acesso baseado em nome de máquinas origem ou endereços IP; - Suporte a acesso via usuário anônimo (usuário IUSR_nomedamáquina); - Autenticação básica (sem criptografia); - Windows NTLM.
Forma de execução - processos múltiplos ou <i>threads</i> (descritivo)	Cada instância do serviço web utiliza <i>threads</i> específicas. Quando especificado explicitamente, pode-se ter processos separados.

TABELA 5.10 – Métricas para avaliação do serviço Apache

Métricas	
Informações Gerais	
Produto	Apache HTTP Server.
Fabricante	Apache Software Foundation.
Plataformas Suportadas	Linux, Windows e diversos tipos de Unix (AIX, Solaris, HPUX, entre outros).
Preço	Software Livre e gratuito.
Desempenho Temporal	
Vazão do servidor (Mbps)	Páginas Estáticas: 7,35. Páginas Estáticas e CGI: 7,23.
Tempo de resposta (segundos)	Páginas Estáticas: 4,01 à 12,85. Páginas Estáticas e CGI: 3,42 à 10,90.
Taxa de Custo / Desempenho	Não foi avaliado. Observações feitas anteriormente.
Confiabilidade / Disponibilidade	
Duração do teste	Não foi avaliado.
Período em que o sistema está operacional (em horas)	Não foi avaliado.
Percentual de disponibilidade	Não foi avaliado.
Tempo médio de duração das falhas	Não foi avaliado.
MTBF – <i>Mean Time Between Failures</i>	Não foi avaliado.
Probabilidade de ocorrência de erros	Não foi avaliado.
Administração	
Possuir ferramenta gráfica de administração (sim/não)	Sim, através de ferramentas de terceiros ou incorporadas em algumas distribuições de Linux.
Possibilidade de administrar via web (sim/não)	Sim, através de ferramentas de terceiros.
Administração via linha de comandos (sim/não)	Sim. Permite iniciar, parar, administrar usuários e alterar arquivo de configuração.

Formatos e possibilidades de <i>logs</i> (descritivo)	- <i>Logs</i> de erros e acessos; - Permite gerar arquivos de <i>logs</i> isolados para cada <i>web site</i> ; - Formatos suportados: CERN, NCSA e permite a utilização de um método flexível em que o administrador pode estabelecer o formato do <i>log</i> de acesso.
Permite o estabelecimento de classificação do site, como nudez, violência, sexo, e linguagem ofensiva (sim/não)	Sim. Suporte à PICS (Plataform for Internet Content Selection) através de produtos de terceiros.
Permite a criação de headers HTTP customizados (sim/não)	Sim, através de módulo próprio.
Permite administrar expiração de conteúdo (sim/não)	Sim, através de módulo próprio.
Permite limitar o número de conexões (sim/não)	Sim, através de parâmetro de configuração.
Customização de mensagens de erro (sim/não)	Sim, através de parâmetro de configuração.
Recursos	
Sistemas Operacionais Suportados (descritivo)	Linux, Windows e diversos tipos de Unix (AIX, Solaris, HPUX, entre outros).
Suporte a CGI, API e linguagem de script (descritivo)	Suporte a CGI e PHP para todos os sistemas operacionais e ISAPI para ambiente Windows, através de módulos. Suporta também ASP, JSP e Java através de produtos de terceiros.
Suporte a SSL (sim/não)	Sim. Versões 2 e 3, através de produtos de terceiros.
Suporte a HTTP 1.1 (sim/não)	Sim.
Integração com outros serviços – FTP, SMTP, Monitor transações (descritivo)	Não possui os serviços agregados. Por filosofia o Apache se propõe a ser exclusivamente um servidor http, podendo se integrar com outros serviços de terceiros ou módulos agregados.
Integração com ferramenta de pesquisa (sim/não)	Sim, através de ferramentas de terceiros.
Isolamento de processos (sim/não)	Sim. Cada processo filho atende as requisições de um cliente conectado. Também permite a utilização de módulos dinâmicos, carregados no momento de sua execução.

Suporte a múltiplos <i>web sites</i> (sim/não)	Sim, implementado em arquivo de configuração.
Controle de uso de banda e processador (descritivo)	Necessita desenvolvimento de programa ou <i>script</i> .
Métodos de autenticação suportados (descritivo)	- Controle de acesso baseado em nome de máquinas origem ou endereços IP; - Suporte a acesso via usuário anônimo; - Autenticação através de arquivo texto; - Autenticação através de arquivo DBM ou DB.
Forma de execução - processos múltiplos ou <i>threads</i> (descritivo)	Um processo “pai” exerce o papel de criar processos filhos, os quais serão responsáveis pelo atendimento das requisições http. Possui parâmetros no arquivo de configuração para definir o número de processos filhos utilizados ao iniciar o servidor web.

5.7 Análise e Interpretação dos Dados

Antes de analisar os dados apresentados na seção anterior, é importante que sejam feitas algumas considerações acerca dos produtos avaliados.

Uma primeira constatação é a característica reportada na própria documentação do servidor web Apache. É colocado que o Apache é um software projetado com a preocupação principal de executar corretamente, para depois se preocupar com desempenho, já que o gargalo normalmente ocorre na rede. Em uma rede de 10 Mbps, um servidor Pentium com características básicas, executando Apache, pode explorar a banda de forma adequada. O foco de desenvolvimento do produto é a correção e facilidades de configuração [GAU 01].

De qualquer forma, as últimas versões do Apache vêm trazendo alterações significativas em parâmetros e ajustes, visando melhorias de desempenho temporal.

Como colocado nos objetivos gerais desta avaliação, não se pretendeu fazer ajustes ou instalações específicas nos sistemas operacionais e serviços sob teste. Utilizou-se a forma básica do serviço e instalações típicas, sem nenhuma compilação ou acerto específico. Um trabalho de *tuning* [CUR 01] [KIL 98] do serviço web deve refletir em alterações significativas nos resultados obtidos.

Na verdade, este é um trabalho essencial de um administrador de serviços

Internet. Ocorre com muita freqüência de simplesmente ser ativado o servidor e serviço, sem ajustes nos parâmetro de configuração. Profissionais que conheçam o sistema operacional a ser empregado são fundamentais para garantir que o servidor atenderá a demanda com os padrões de qualidade exigidos, tanto no que tange a desempenho, disponibilidade, quanto em quesitos de segurança.

O *Internet Information Services* (IIS) apresentou melhores resultados na avaliação de desempenho temporal, tanto nos comparativos entre páginas estáticas, quanto entre páginas dinâmicas, vide gráficos comparativos 5.2 e 5.3. A utilização de uma carga de trabalho mais próxima do real, envolvendo aplicação CGI e ISAPI, não foi possível em ambas as plataformas, já que o ambiente Apache em Linux não suporta programas desenvolvidos utilizando tecnologia ISAPI. De qualquer forma, as medições foram feitas no ambiente Windows, sendo possível verificar os resultados e compará-los com os outros obtidos através da aplicação de cargas de trabalho com páginas estáticas e aplicações CGI.

Os indicadores relativos à administração e recursos mostraram-se com resultados semelhantes quanto ao suporte e a existência de alternativas, apesar da diferença de proposta entre os dois produtos. O IIS tem uma preocupação e foco na facilidade de uso e na disponibilização de recursos de forma mais direta, ou seja, o produto fornece uma variada gama de serviços adicionais e prontos para o uso. Já o Apache possui uma filosofia diferente, onde pretende focar a realização da atividade de servidor HTTP. Possui interfaces e possibilidades de integração com serviços de terceiros. Além disto, possui uma estrutura modular, permitindo a ativação de componentes a medida em que são necessários.

Esta diferença de posicionamento reflete em vantagens e desvantagens para ambos os lados. A facilidade de utilização e a disponibilização automática de recursos oferecida pelo IIS, gera problemas e dificuldades relativos à segurança.

O mesmo pode ser dito do Apache. Os ganhos de segurança oferecidos por uma estrutura enxuta, geram dificuldades e tempos maiores para a instalação e configuração dos módulos e produtos externos.

As desvantagens apontadas podem ser minimizadas com uma administração correta.

Como colocado e justificado anteriormente, as métricas de disponibilidade e confiabilidade não foram apuradas. São áreas onde tradicionalmente os sistemas operacionais Unix e derivados dele, com seus serviços, possuem avaliações melhores que os sistemas baseados em Windows, sendo que o Windows 2000 traz melhorias nestas áreas.

Os dados de utilização apontados pela Netcraft (www.netcraft.com) a qual disponibiliza informações de utilização de servidores web, mostram que o Apache é o produto de maior utilização e é empregado em uma série de empresas de porte. Já o IIS apresenta um crescimento significativo em sua utilização. São indicadores subjetivos, mas que mostram a viabilidade de uso de ambos os produtos para atender

demandas corporativas.

A Procempa utiliza como ambiente web principal o produto da Microsoft, o IIS, mas também possui servidores com o Apache para demandas mais específicas. Existe uma expectativa muito grande com a evolução dos projetos de software livre e de toda as idéias de desenvolvimento de tecnologia própria e liberdade em distribuir conhecimento livremente, além das reduções de custo provenientes desta opção.

Apesar dos resultados superiores do IIS nos indicadores de desempenho temporal e uma relativa equivalência nas avaliações das alternativas de administração e recursos, não seria adequado indicar um produto em detrimento do outro. Além da ausência da avaliação completa das métricas propostas, ajustes nos serviços representam ganhos significativos e alteram os resultados. Ambos os produtos representam o que há de melhor neste mercado e pode-se afirmar que com uma administração correta, ambos produtos podem atender as demandas exigidas pelo provedor de acesso PortoWeb e pelas demandas de páginas e aplicações da Prefeitura.

5.8 Conclusão

O capítulo 5 buscou colocar em prática o método, passos e métricas abordados no capítulo anterior. A aplicação foi feita utilizando-se as informações do servidor web corporativo da Procempa e através da montagem de um ambiente de teste, reproduzindo as cargas encontradas no levantamento de informações, chegou-se aos resultados das métricas relativas ao desempenho temporal.

O mesmo ambiente, somado à pesquisa na bibliografia e documentação dos produtos, permitiram a obtenção dos resultados das métricas de administração e recursos dos servidores web.

Os produtos analisados foram o Apache executando em sistema operacional Linux e o IIS (*Internet Information Services*) no ambiente Windows 2000.

Ao final, foram realizadas interpretações e análise dos resultados obtidos, além de algumas observações relevantes relativas a itens não analisados e características específicas dos produtos.

6. Conclusões

A cada dia que passa a Internet vem assumindo um papel mais importante para as organizações e as pessoas de um modo geral. Ela vem revolucionando a forma como as pessoas se comunicam, obtém informações e fazem negócios, sendo que o serviço essencial e mais utilizado neste processo é o *World Wide Web*, baseado no protocolo HTTP [TAN 94].

A demanda e a exigência de qualidade estão aumentando drasticamente. Por um lado, os usuários querem serviços rápidos e sem ocorrência de erros, por outro as empresas devem ter estruturas técnicas bem dimensionadas e estar preparadas para atender as requisições feitas aos diversos serviços por elas oferecidos.

Para que os provedores de serviços Internet tenham critérios técnicos para estabelecer a sua estrutura de hardware e software, são utilizadas métricas apropriadas ao contexto em questão. Tais métricas permitem a avaliação de itens como desempenho, custo, usabilidade, disponibilidade, facilidade de administração, possibilitando comparações entre plataformas, sistemas operacionais e softwares aplicativos.

O presente trabalho apresentou um método de avaliação de servidores web, estabelecendo um conjunto de métricas para a avaliação do serviço WWW - *World Wide Web*, abrangendo quatro aspectos básicos: Desempenho [BLA 99] [SLO 96], disponibilidade [SCH 00], facilidades de administração e recursos. Estas métricas foram estabelecidas a partir de um levantamento na literatura existente, seguido por uma etapa de seleção baseada na análise e experimentação das mesmas.

A aplicação de um método de avaliação de servidores WWW, baseado em métricas bem definidas, pode desempenhar um papel de suma importância para um dimensionamento e escolha corretos das ferramentas a serem empregadas.

Depois de estabelecido, o método foi aplicado no ambiente corporativo da Companhia de Processamento de Dados do Município de Porto Alegre – Procempa para validação e testes dos servidores WWW da empresa. Foram dois os objetivos de sua aplicação na Procempa:

- Servir como ambiente de teste para validar o método, e neste caso, apontar falhas, implicando em alterações no conjunto inicial de métricas e na forma de sua aplicação;
- Fornecer dados importantes para a empresa, com respeito ao seu planejamento de infra-estrutura computacional para Web, permitindo avaliação do ambiente atual e futuro dentro deste contexto.

Ambos objetivos foram atingidos. O método e métricas propostas foram

aplicadas e um levantamento importante do servidor web foi feito, com uma posterior medição dos serviços web IIS e Apache em ambiente de teste. Tanto o método, quanto as métricas utilizadas para desempenho, recursos e administração, se mostraram adequadas, sendo que se faz necessária uma constante atualização das mesmas em decorrência dos avanços tecnológicos.

Apesar do teste realizado não ter sido completo, faltando as métricas relativas à disponibilidade e confiabilidade, os resultados permitiram uma avaliação parcial dos serviços. O IIS apresentou indicadores melhores de desempenho temporal e foi constatada uma relativa equivalência nas avaliações de administração e recursos.

O objetivo deste trabalho não era apresentar uma conclusão definitiva, indicando qual o melhor produto e sim, apontar características, pontos positivos e negativos das soluções analisadas. Não seria adequado indicar um produto em detrimento do outro. Ambos os produtos são os mais utilizados e representam o que há de melhor neste mercado, podendo-se afirmar que com uma administração correta, podem atender as demandas exigidas pelo provedor de acesso PortoWeb e pelas demandas de páginas e aplicações da Prefeitura.

O resultado do trabalho pôde ser utilizado na Procempa, tanto a questão de levantamento de informações feito no ambiente principal da empresa, o qual permitiu conhecer a estrutura e comprovar a adequação do servidor, baseado nos indicadores apurados, quanto na indicação da viabilidade de uso dos serviços avaliados e de um método para utilização em futuras avaliações.

Algumas adequações, melhorias e trabalhos futuros devem ser feitos nesta proposta de método para a avaliação de servidores WWW no ambiente corporativo. A utilização das métricas de disponibilidade e confiabilidade, além de uma estrutura mais detalhada para a apuração destes indicadores se faz necessário.

Outro aspecto importante é a identificação e destaque para novas métricas específicas para a avaliação dos aspectos de segurança. Cada vez mais, as garantias de segurança têm assumido um papel importante quando tratamos de servidores web, visto que os mesmos se constituem em uma porta de entrada para invasores em potencial e sempre são considerados em análises de risco.

Anexo 1 Levantamento de Informações do Servidor Web

Este anexo apresenta as informações relativas ao levantamento de informações feito no servidor web principal da Procempa, apresentando dados referentes ao processador, memória e serviço web [CUR 01].

O monitor de desempenho, software utilizado para obter estas informações, foi configurado para gerar dados a cada 5 minutos, sendo o tempo total de cada levantamento de uma semana.

Ao todo foram quatro períodos, portanto quatro semanas, de captura de dados e o valor médio e máximo apresentados foram apurados neste tempo, sendo que a cada cinco minutos o monitor de desempenho gerava uma média e ao final de uma semana obteve-se a média do período. Este valor apurado foi somado as outras três medições para obter o valor final apresentado nas tabelas.

Os campos mostrados nas tabelas são o nome da medida, o processo que se refere, uma descrição do que significa a medida e seus valores médio e máximo.

O campo processo define quando o indicador reflete todos os processos da máquina ou algum processo específico. O processo comumente encontrado é o Inetinfo, o qual representa o serviço web em servidores Windows executando *Internet Information Services*.

TABELA A1.1 - Métricas referentes à memória do servidor Web

Nome	Processo	Descrição	Valor médio	Valor máximo
<i>Working Set</i>	Inetinfo	O <i>Working Set</i> é o conjunto de páginas de memória acessadas recentemente pelas <i>threads</i> do processo, representados em bytes. Se o computador possuir memória livre acima de um determinado número, as páginas são deixadas no <i>Working Set</i> de um processo mesmo se não estiver em uso. Quando a memória livre atingir o limite, as páginas são tiradas do <i>Working Set</i> . Se estas forem necessárias será gerada uma <i>soft-faulted</i> para trazer de volta ao <i>Working Set</i> .	157.305.854	157.833.216
<i>Virtual Bytes</i>	Inetinfo	<i>Virtual Bytes</i> é o tamanho em bytes de espaço de endereçamento virtual que o processo está utilizando.	631.155.564	637.427.712
<i>% Usage (Paging File)</i>	Total	Percentual de uso do arquivo de paginação.	19,179	19,2755
<i>Transition Fault / sec</i>	Total	<i>Transition Faults/sec</i> representa o número de <i>page faults</i> resolvidas pela recuperação de páginas que estavam na lista de páginas modificadas, na lista de espera, ou sendo escrita no disco no momento do <i>page fault</i> .	304,058	691,556

<i>Pool Paged Bytes</i>	Total	<i>Pool Paged Bytes</i> é o número de bytes no <i>paged pool</i> , uma área de memória do sistema (memória física usada pelo sistema operacional) para objetos que podem ser gravados no disco quando não estiverem em uso.	44.869.109	45.004.800
<i>Pool Nonpaged Bytes</i>	Total	<i>Pool Nonpaged Bytes</i> é o número de bytes no <i>nonpaged pool</i> , uma área de memória do sistema para objetos que não podem ser escritos para disco, mas devem permanecer na memória física, enquanto estiver alocada.	34.874.925	34.947.072
<i>Paged Inputs / Sec</i>	Total	Pages Input/sec representa o número de páginas lidas do disco para resolver <i>hard page faults</i> . (<i>Hard page faults</i> ocorrem quando um processo solicita dados que não estão no <i>working set</i> ou memória física e devem ser buscadas do disco).	319,692	529,027
<i>Page Reads / Sec</i>	Total	<i>Page Reads/sec</i> indica o número de vezes que o disco foi lido para resolver <i>hard page faults</i> .	56,343	163,561

<i>Page Faults / Sec</i>	Total	<i>Page Faults/sec</i> representa o número total de <i>pages fault</i> tratado pelo processador. Este indicador inclui tanto <i>hard faults</i> (que exigem acesso ao disco) quanto <i>soft faults</i> (quando a página é encontrada na memória física). Os processadores podem tratar grandes números de <i>soft faults</i> sem maiores conseqüências, entretando, <i>hard faults</i> podem ocasionar atrasos significativos.	1.024,138	5.269,544
<i>Cache Bytes</i>	Total	<i>Cache Bytes</i> é o somatório de bytes residentes no <i>cache</i> de sistema, bytes residentes de código do sistema e bytes residentes no <i>Pool Paged</i> . É o espaço utilizado para armazenar em <i>cache</i> as páginas estáticas.	47.792.694	57.016.320
<i>Available Bytes</i>	Total	Este indicador representa o número em bytes de memória física disponível aos processos que estejam executando.	166.012.076	210.518.016
<i>% Cache Hits (IIS)</i>	Total	<i>% Cache Hits</i> é a taxa de sucessos em relação ao número total de requisições ao <i>cache</i> . Para <i>sites</i> exclusivamente com páginas estáticas este indicador deve ficar entre 80% a 90%.	66,639	66,641

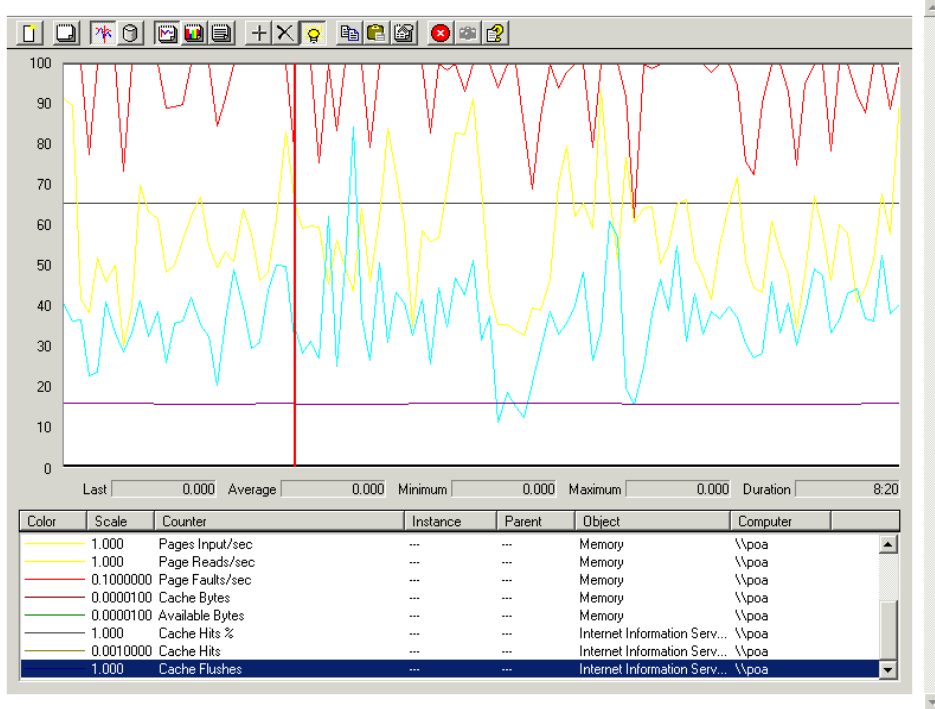


FIGURA A1.1 – Monitor de Desempenho: Memória

TABELA A1.2 - Métricas referentes ao processados do servidor web

Nome	Proc.	Descrição	Valor médio	Valor máximo
<i>Processor Queue Length</i>	Total	Representa o número de <i>threads</i> na fila do processador, considerando apenas aquelas prontas para execução. Uma fila constantemente com mais de duas <i>threads</i> , normalmente representa um congestionamento no processador.	0,340	6,000
<i>Interrupts / Sec</i>	Total	<i>Interrupts/sec</i> é a média de interrupções de hardware que o processador está recebendo e servindo a cada segundo. Não inclui DPCs, o qual é representado separadamente. Este valor é um indicador indireto da atividade dos dispositivos que geram interrupções, como o <i>clock</i> , mouse, dispositivos de disco, interfaces de rede e outros periféricos. Estes dispositivos normalmente interrompem o processador quando finalizam uma tarefa ou requisitam o processador, sendo que a execução de uma <i>thread</i> normal fica suspensa durante a interrupção. Um aumento drástico neste indicador, sem aumento correspondente na atividade do sistema, aponta para um problema de hardware.	268,327	491,385
<i>% Processor Time</i>	Total	Representa o percentual de tempo que o processador está ocupado (executando uma <i>thread</i> diferente de <i>Idle</i>). Valores constantemente maiores que 80% podem indicar limitações de processador.	30,998	57,321
<i>% DPC Time</i>	Total	<i>% DPC Time</i> é o percentual de tempo que o processador utiliza recebendo e servindo <i>deferred procedure calls</i> (DPCs). DPCs são interrupções que executam em uma prioridade mais baixa que uma interrupção padrão.	0,062	1,094

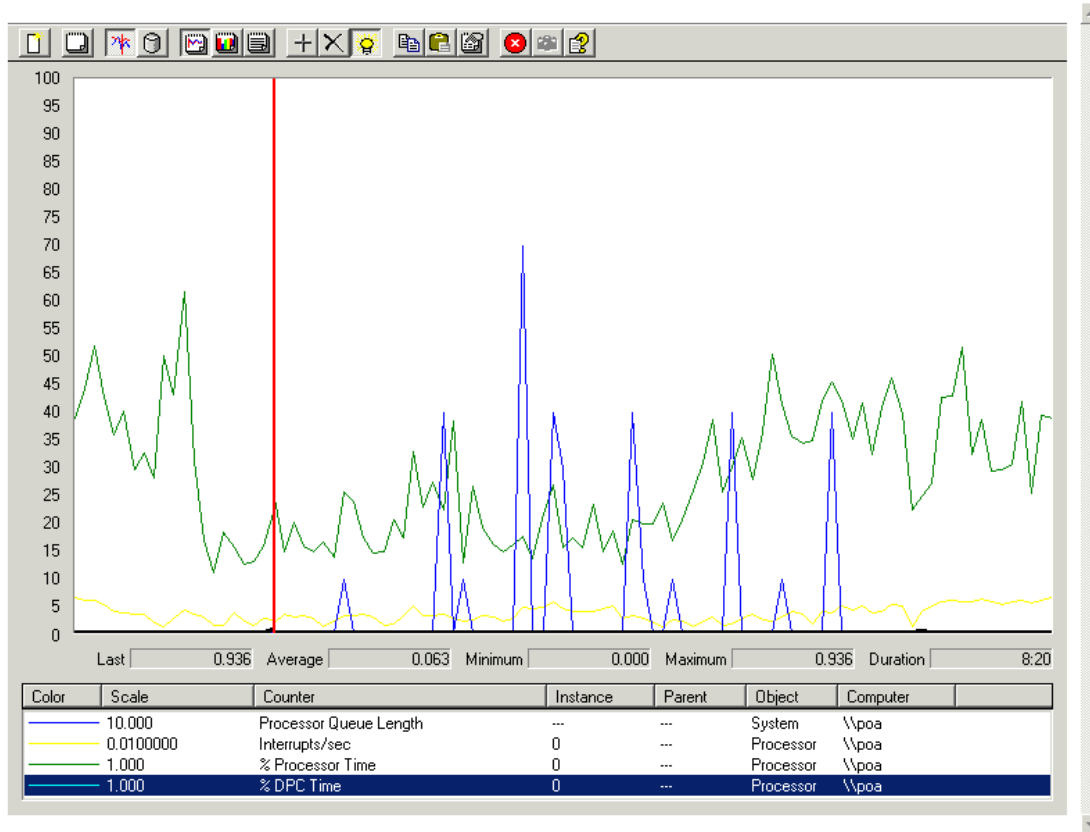


FIGURA A1.2 – Monitor de desempenho: Processador

TABELA A1.3 - Métricas referentes ao serviço web do servidor

Nome	Processo	Descrição	Valor médio	Valor máximo
<i>Post Request / sec</i>	Web Service	Representa a taxa de requisições HTTP usando o método POST (geralmente utilizado através de formulários e <i>gateways</i>).	0,087	0,410
<i>Maximum Connections</i>	Web Service	Indica o número máximo de conexões simultâneas estabelecidas com o serviço web.	–	592
<i>ISAPI Extentions Request / Sec</i>	Web Service	É a taxa de requisições a extensões ISAPI que estão sendo processadas simultaneamente pelo servidor WWW.	0,323	1,699
<i>Get Request / Sec</i>	Web Service	É a taxa de requisições HTTP utilizando o método GET (geralmente empregado em recuperação básica de arquivos e imagens para serem utilizados por formulários).	4,424	20,470
Current Connections	Web Service	Representa o número corrente de conexões estabelecidas com o serviço web.	177	185
Connection Attempts / Sec	Web Service	Indica a taxa de tentativas de conexões usando o serviço web.	1,478	6,133
CGI Request / Sec	Web Service	Representa a taxa de requisições CGI que estão sendo simultaneamente processadas pelo servidor WWW.	0,142	0,803
Request / Sec (ASP)	Web Service	É o número de requisições a páginas ASP por segundo sendo executadas.	0,240	1,302

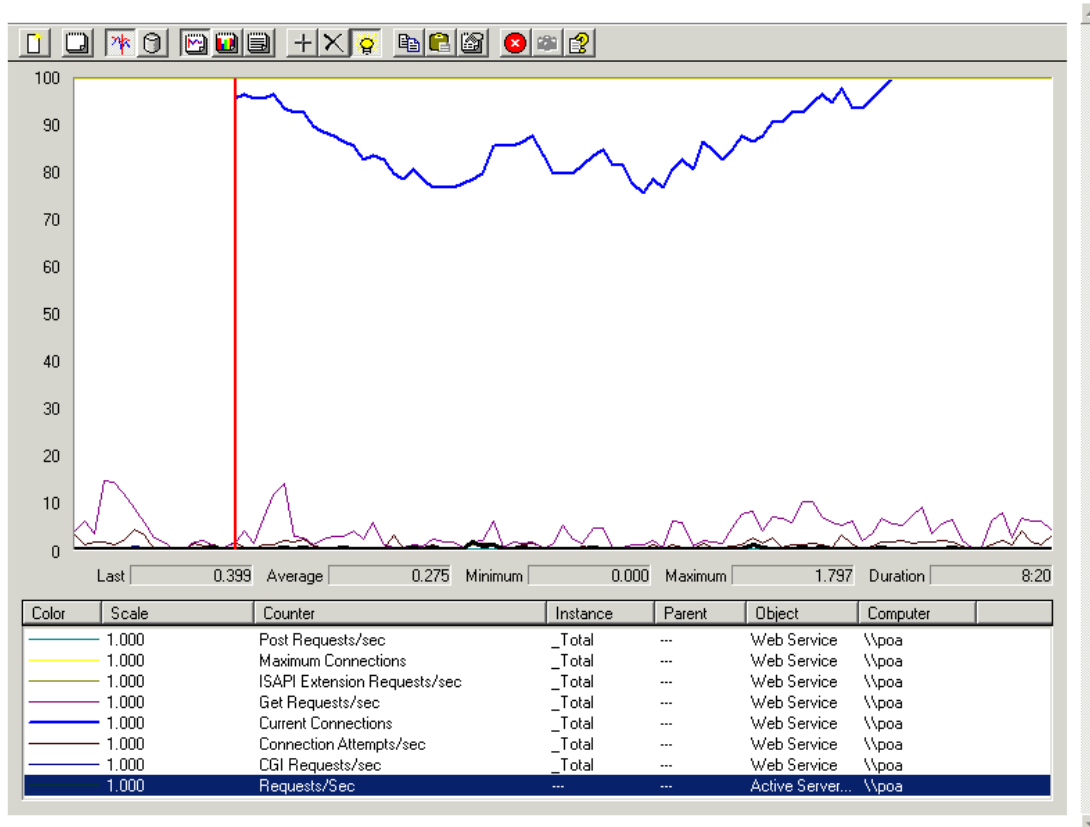


FIGURA A1.3 – Monitor de desempenho: Serviço web

Anexo 2 Arquivos de Configuração WebStone

O anexo 2 apresenta os arquivos de configuração da ferramenta WebStone utilizados no ambiente de teste. São apresentados os dois arquivos principais. O primeiro chamado TESTBED, onde são definidos o número inicial de clientes, número máximo de clientes, incremento de cada execução, o número de interações do sistema e o tempo de cada execução.

O segundo é denominado LISTFILE e contém o nome dos arquivos que serão requisitados pelos clientes web. O WebStone oferece arquivos padrão com tamanhos variados, sendo que a denominação reflete o tamanho de cada um. Como exemplo, file3k.html, resultará na chamada a um arquivo com 3 Kbytes de informação. Também são disponibilizadas uma aplicação CGI denominada ws25_cgi.exe e uma aplicação ISAPI com o nome de ws25_iis.dll.

Outro parâmetro importante deste arquivo é o peso que cada uma das requisições terá durante o período de teste. Definiu-se um valor total de 10.000 e o peso de cada uma das linhas estabelecidas de acordo com o levantamento feito na caracterização da carga de trabalho.

```

#####
#
# Arquivo de Configuração TESTBED
# -----
#
#
# Licença:
# -----
#
# The contents of this file are subject to the WebStone Public License
# Version 1.0 (the "License"); you may not use this file except in
# compliance with the License. You may obtain a copy of the License
# at http://www.mindcraft.com/webstone/license10.html
#
# Software distributed under the License is distributed on an "AS IS"
# basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See
# the License for the specific language governing rights and limitations
# under the License.
#
# The Original Code is WebStone 2.5.
#
# The Initial Developer of the Original Code is Silicon Graphics, Inc.
# and Mindcraft, Inc.. Portions created by Silicon Graphics. and
# Mindcraft. are Copyright (C) 1995#1998 Silicon Graphics, Inc. and
# Mindcraft, Inc. All Rights Reserved.
#
#####
### Parâmetros do BENCHMARK – Alterados por Leandro Côte
# Webstone irá iniciar com o número de processos definidos em MINCLIENTS
# Ele irá executar por TIMEPERRUN minutos. Quando a execução acabar o número
# de clientes será incrementado de CLIENTINCR e outro teste será feito. Este
# processo continuará até que o número alcance número de clientes MAXCLIENTS
# Este conjunto de passos será executado pelo número de ciclos definidos em
# ITERATIONS
ITERATIONS="3"
MINCLIENTS="180"
MAXCLIENTS="600"
CLIENTINCR="60"
TIMEPERRUN="30"

### Parâmetros do Servidor
PROXYSERVER=

# Abaixo o endereço IP ou nome do servidor web
SERVER="192.168.1.1"

# A porta 80 é a porta padrão dos servidores web
PORTNO=80

# RCP é um comando usado para copiar arquivos entre os sistemas clientes e
# entre os clientes e o servidor. O RCP e RSH são usados para copiar os arquivos

```

```

# de configuração e distribuí-lo entre os clientes web. Se forem deixados em
branco # o WebStone não fará a distribuição automaticamente, devendo a mesma
ser feita
# manualmente
#RCP=rcp
#RSH=rsh
RCP=
RSH=

# Quando o WebStone inicia a execução, ele tentará conectar ao servidor web e
# buscar os arquivos listados em SERVERINFO, OSTUNINGFILES, e
# WEBSERVERTUNINGFILES. As informações serão armazenadas no diretório de
# log
SERVERINFO=
OSTUNINGFILES=
WEBSERVERTUNINGFILES=

# Parâmetro não utilizado
WEBSERVERDIR=

# Caminho absoluto para o diretório de páginas web no servidor
WEBDOCDIR=/

# Lista de máquinas clientes separadas por vírgulas. Pode ser utilizado o endereço
# IP ou nome da máquina. Será executado o comando rexec em cada um destes
# sistemas para iniciar o programa webclient
CLIENTS="192.168.1.2 192.168.1.3 192.168.1.4"

# Nome do usuário e senha válidos nas máquinas clientes. O programa webmaster
# irá executar um rexec a cada sistema cliente com a finalidade de iniciar o
# programa webclient
CLIENTACCOUNT=$LOGNAME
CLIENTPASSWORD=xxxxxx

# Quando este parâmetro é setado para "true", será utilizada a mesma semente
# randômica durante cada teste. Com isto, os testes serão mais facilmente
# reproduzíveis
FIXED_RANDOM_SEED=true

# Caso o parâmetro RSH (acima) foi definido, o WebStone executar o comando
# definido em CLIENTINFO. A saída será gravada no diretório de log. Como
# exemplo, pode se obter a configuração do sistema com o comando uname
CLIENTINFO="uname -a"

# Diretório temporário nos sistemas clientes
TMPDIR=/tmp

# Caminho para o programa webclient nos sistemas clientes
CLIENTPROGFILE=/home/corte/WebStone2.5/bin/webclient

# Quando setado para 1 habilita a opção de debug.
DEBUG=0

```

```
#####  
#  
# Arquivo de Configuração LISTFILE  
# -----  
#  
#  
# Descrição:  
# -----  
# Arquivo de carga baseado nos dados do servidor principal da Procempa  
# Inclui paginas estáticas  
#  
# autor: Leandro Corte  
#  
#####
```

```
/file3k.html          6427  
/file8k.html          1547  
/file50k.html         1868  
/file500k.html        158
```

TOTAL = 10000


```
#####  
#  
# Arquivo de Configuração LISTFILE  
# -----  
#  
#  
# Descrição:  
# -----  
# Arquivo de carga baseado nos dados do servidor principal da Procempa  
# Inclui paginas estáticas e cgi  
#  
# autor: Leandro Corte  
#  
#####
```

```
/file3k.html 5894  
/cgi-bin/ws25_cgi.exe?file=/file3k.html 533  
/file8k.html 1014  
/cgi-bin/ws25_cgi.exe?file=/file8k.html 533  
/file50k.html 1336  
/cgi-bin/ws25_cgi.exe?file=/file50k.html 532  
/file500k.html 158
```

TOTAL = 10000

```
#####  
#  
# Arquivo de Configuração LISTFILE #  
# ----- #  
# # #  
# # #  
# Descrição: #  
# ----- #  
# Arquivo de carga baseado nos dados do servidor principal da Procempa #  
# Inclui paginas estáticas, cgi e ISAPI #  
# # #  
# autor: Leandro Corte #  
# # #  
#####
```

```
/file3k.html 5894  
/cgi-win/ws25_cgi.exe?file=/file3k.html 107  
/cgi-win/ws25_iis.dll?file=c:\inetpub\wwwroot\file3k.html 426  
/file8k.html 1014  
/cgi-win/ws25_cgi.exe?file=/file8k.html 107  
/cgi-win/ws25_iis.dll?file=c:\inetpub\wwwroot\file8k.html 426  
/file50k.html 1336  
/cgi-win/ws25_cgi.exe?file=/file50k.html 107  
/cgi-win/ws25_iis.dll?file=c:\inetpub\wwwroot\file50k.html 425  
/file500k.html 158
```

TOTAL = 10000

Bibliografia

- [APA 01] APACHE HTTP SERVER DOCUMENTATION PROJECT. **Apache HTTP Server Version 1.3** . 2001. Disponível por WWW em <http://httpd.apache.org/docs>. (19 Dez. 2001).
- [BLA 99] BLAKELEY, Michael. **Web Server Performance Measurement FAQ**. 1999. Disponível por WWW em http://www.brakeley.com/resources/web_server_perf_faq.html. (26 Jan. 2000).
- [CHE 96] CHEN, B.; ENDO, Y; CHAN, K.; MAZIERES, D.; DIAS, A.; SELTZER, M; SMITH, M. **The Measured Performance of Personal Computer Operating Systems**. ACM Transactions on Computer Systems, Jan. 1996.
- [COR 00] CÔRTE, Leandro. **Métricas para Avaliação dos Serviços Internet Básico – WWW e Correio Eletrônico**. Trabalho individual, Jul. 2000.
- [CUR 01] CURRY, Benjamim; REILLY, George; KALDESTAD Hallvard. **The Art and Science of Web Server Tuning with Internet Information Services 5.0. Microsoft Technical Information**, Redmond, Jan. 2001.
- [GAU 01] GAUDET, Dean. **Apache Performance Notes**. 2001. Disponível por WWW em <http://httpd.apache.org/docs/misc/perf-tuning.html>. (08 Jan. 2002).
- [JAE 99] JAEGER, Bill. Minimize Risk with Proactive Performance Testing. **E-Business Advisor**, San Diego, v.17, n.4, p.32-35, Abr. 1999.
- [JAI 91] JAIN, Raj. **The Art of Computer Systems Performance Analysis**. Littleton: John Wiley & Sons Inc, 1991. 685p.
- [KIL 98] KILLELEA, Patrick. **Web Performance Tuning**. Sebastopol: O'Reilly & Associates Inc, 1998. 349p.
- [LAU 99] LAURIE, Ben; LAURIE, Peter. **Apache: The Definitive Guide**. Sebastopol: O'Reilly & Associates Inc, 1999. 385p.

- [LIU 94] LIU, Cricket; PEEK, Jerry; JONES, Russ; BUUS, Bryan; NYE, Adrian. **Managing Internet Information Services**. Sebastopol: O'Reilly & Associates Inc, 1994. 630p.
- [MAN 98] MANHEIM, Seth; MOREY, Jim. **Microsoft Internet Information Services Resource Kit**. Redmon: Microsoft Press, 1998. 587p.
- [MEN 98] MENASCÉ, Daniel A.; ALMEIDA, Virgilio A.F. **Capacity Planning for Web Performance Metrics, Models, & Methods**. Upper Saddle River: Prentice-Hall Inc, 1998. 321p.
- [MIN 99] MINDCRAFT WEBSTONE. **WebStone 2.5**. 1999. Disponível por WWW em <http://www.mindcraft.com/webstone>. (15 Fev. 2000).
- [NET 01] NETTO, João C.; CECHIN, Sérgio L. **Ferramentas de Avaliação de Desempenho de Sistemas Computacionais**. 1^a. Escola Regional de Alto Desempenho, 2001. 258p.
- [NIE 99] NIEDERST, Jennifer. **HTML Pocket Reference**. Sebastopol: O'Reilly & Associates Inc, 1999. 95p.
- [PRA 96] PRADHAM, D. **Fault-tolerant Computer Design**. Prentice-Hall Inc, 1996.
- [RAD 99] RADHAKRISHNAN, Ramesh; RAWSON, Freeman L. **Characterizing the Behavior of Windows NT Web Server Workloads Using Processor Performance Counters**. 1999.
- [SCH 00] SCHMIDT, Jürgen. **Analysis of Web Server Downtimes**. 2000. Disponível por WWW em <http://www.heise.de/ct/english/00/08/174>. (29 Abr. 2000).
- [SEL 99] SELTZER, Margo; KRINSKY, David; SMITH, Keith; ZHANG, Xiolan. **The Case for Application-Specific Benchmarking**. Harvard University. 1999.
- [SLO 96] SLOTHOUBER, Louis P. **A Model of Web Server Performance**. 1996. Disponível por WWW em <http://www.starnine.com/websar/overview.html>. (26 Jan. 2000).

- [STA 00] STANDARD PERFORMANCE EVALUATION CORPORATION. **SPECWeb99**. 2000. Disponível por WWW em <http://www.spec.org>. (28 Jan. 2000).
- [TAN 94] TANENBAUM, Andrew S. **Redes de Computadores**. Rio de Janeiro: Campos, 1994. 923p.
- [ZDN 99] ZDNET BENCHMARK INSIDER. **WebBench 3.0**. 1999. Disponível por WWW em <http://www.zdnet.com/zdlabs/filters/benchmarkinsider>. (10 Fev. 2000).
- [WEI 99] WEINER, Bruce. **Open Benchmark Windows NT Server 4 and Linux**. 1999. Disponível por WWW em <http://www.mindcraft.com/whitepapers/openbench1.html>. (18 Abr. 2000).