

Análise de Desempenho de Soluções para Customização da *url* de Acesso às Páginas do Sítio do CEFET-MG

Bruno Gradela Robazza, Marco Túlio Alves Rodrigues, Paulo Eduardo Maciel de Almeida

*Departamento de Recursos em Informática
Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 5253, Belo Horizonte, MG, Brasil.*

Email: robazza@dri.cefetmg.br, tulio@dri.cefetmg.br, pema@dri.cefetmg.br

Resumo – Um objetivo importante para as organizações que possuem websites é a disponibilização da informação de modo ágil e dinâmico. Uma opção viável para atingir esse objetivo é utilizar o Sistema de Gerenciamento de Conteúdo OpenCms. Um problema do OpenCms é que, após sua instalação, a url de um website – que inicialmente é www.site.com – passará a ser www.site.com:8080/opencms/opencms. Este artigo analisa o desempenho de duas formas de se remover o [:8080/opencms/opencms](http://www.site.com:8080/opencms/opencms) da url de um site que utiliza o OpenCms 7.0.5 junto com o Servidor HTTP Apache 2.2 e o Servidor de Aplicações Apache Tomcat 5.5 e as compara com o desempenho da instalação padrão do OpenCms.

Termos de Indexação – OpenCms, performance, mod_jk, mod_proxy.

1. Introdução

Um objetivo importante para as organizações que possuem *website* é a disponibilização e atualização da informação de forma ágil e dinâmica. Uma das formas de se fazer isso é utilizando o *OpenCms*, um sistema gerenciador de conteúdo de *websites* que auxilia a criação e a manutenção de *websites*, portais e *intranets* [1] [2].

Um problema do *OpenCms* é que, após sua instalação, a *url* de um *website* – que inicialmente é www.site.com – passará a ser www.site.com:8080/opencms/opencms [1]. É, então, necessário retirar o incômodo “[:8080/opencms/opencms](http://www.site.com:8080/opencms/opencms)” do caminho do *website*. Este artigo analisa o desempenho de 2 maneiras diferentes de se remover o “[:8080/opencms/opencms](http://www.site.com:8080/opencms/opencms)” da *url* do *website*.

2. APACHE e TOMCAT

O *OpenCms* é baseado na tecnologia Java e pode ser disponibilizado utilizando-se o Servidor HTTP Apache e o container de *servlets* Apache Tomcat [3]. A integração do

Servidor Apache junto com o Apache Tomcat é explicada em [4]. [5] mostra como instalar e configurar o *OpenCms*.

Foram encontradas duas maneiras básicas de se retirar o prefixo “:8080/opencms/opencms”. Uma delas utiliza o módulo *mod_proxy* do servidor Apache e é descrita em [6]. A outra, utilizando o módulo *mod_jk*, pode ser encontrada em [7].

3. Metodologia de Avaliação

Para a realização dos testes de performance foram utilizados 2 micro-computadores: um atuando como servidor e um como cliente. O servidor é um Core 2 Duo E8400 com 2 Giga bytes de RAM e 2 discos rígidos SATA de 160 gigabytes da Western Digital em RAID 1 rodando Ubuntu 8.10 com *kernel* 2.6.27-11. O cliente é um Intel Pentium 4 CPU de 2.66GHz com 256 MB de RAM rodando Debian GNU Linux 3.1 com *kernel* 2.6.8. A conexão entre eles foi feita utilizando uma conexão Ethernet 10/100 ponto a ponto.

O servidor possui 3 instalações do Servidor Apache 2.2.9, do Servidor de Aplicações Tomcat 5.5, do banco de dados MySql 5.0.67 e do OpenCms 7.0.5. Uma das instalações do *OpenCms* utiliza o *mod_jk*, uma o *mod_proxy* e outra não remove o prefixo :8080/opencms/opencms. Desse modo pode-se testar o desempenho de cada forma de se remover o prefixo e saber se houve perda de desempenho em relação a instalação padrão do OpenCms.

O cliente utiliza a ferramenta de avaliação de performance ab[8], que permite configurar o número total de requisições HTTP e o número de conexões simultâneas (*threads*) que serão abertas ao mesmo tempo com o servidor.

Foram feitos acessos em 3 caminhos distintos: */index.jsp*, */performance/test/inside/folder/index.jsp* e */performance/test/inside/folder/index.txt*. Os arquivos *index.jsp* são idênticos, possuem 2.408 bytes e produzem uma saída de 6911 bytes. O arquivo *index.txt* possui 6.911 bytes.

Em todos os testes foram feitas 3500 requisições ao servidor variando o número de conexões simultâneas entre 1 e 21. Para a obtenção de resultados mais confiáveis foi calculada a media de 20 amostras independentes.

4. Resultados

Os gráficos apresentados nas figuras 1, 2 e 3 mostram o tempo médio necessário para se atender as 3500 requisições feitas ao servidor em função do numero de conexões

simultâneas acessando, respectivamente, o arquivo */index.jsp*, */performance/test/inside/folder/index.jsp* e */performance/test/inside/folder/index.txt*.

Nota-se, nas figuras 1 e 2, que o tempo necessário para a realização das requisições sofre pouca alteração quando se alterna entre o *mod_jk*, o *mod_proxy* e a instalação padrão. Os gráficos das figuras 1 e 2 mostram o *mod_proxy* e o *mod_jk* com apenas uma pequena perda – menos de 7% – de desempenho em relação a instalação padrão. No entanto, não se nota diferença no desempenho do módulos *mod_jk* e *mod_proxy*.

Por outro lado, o gráfico da Figura 3 (*/performance/test/inside/folder/index.txt*) mostra que o tempo necessário para responder as requisições não se altera quando há alternância entre os módulos e a instalação padrão. Isso já era esperado, pois os arquivos de configuração do *OpenCms* definem que alguns tipos de arquivo sejam respondidos pelo servidor HTTP (Servidor Apache) e não pelo servidor de aplicação (Apache Tomcat). Os arquivos .txt são um dos tipos desses arquivos.

A figura 4 mostra o tempo necessário para que o servidor, utilizando o *mod_jk*, atenda as requisições aos 3 caminhos de teste. Nota-se, neste gráfico, que o servidor consegue atender as requisições ao arquivo .txt com um desempenho superior (quase 50% mais rápido) ao dos arquivos .jsp.

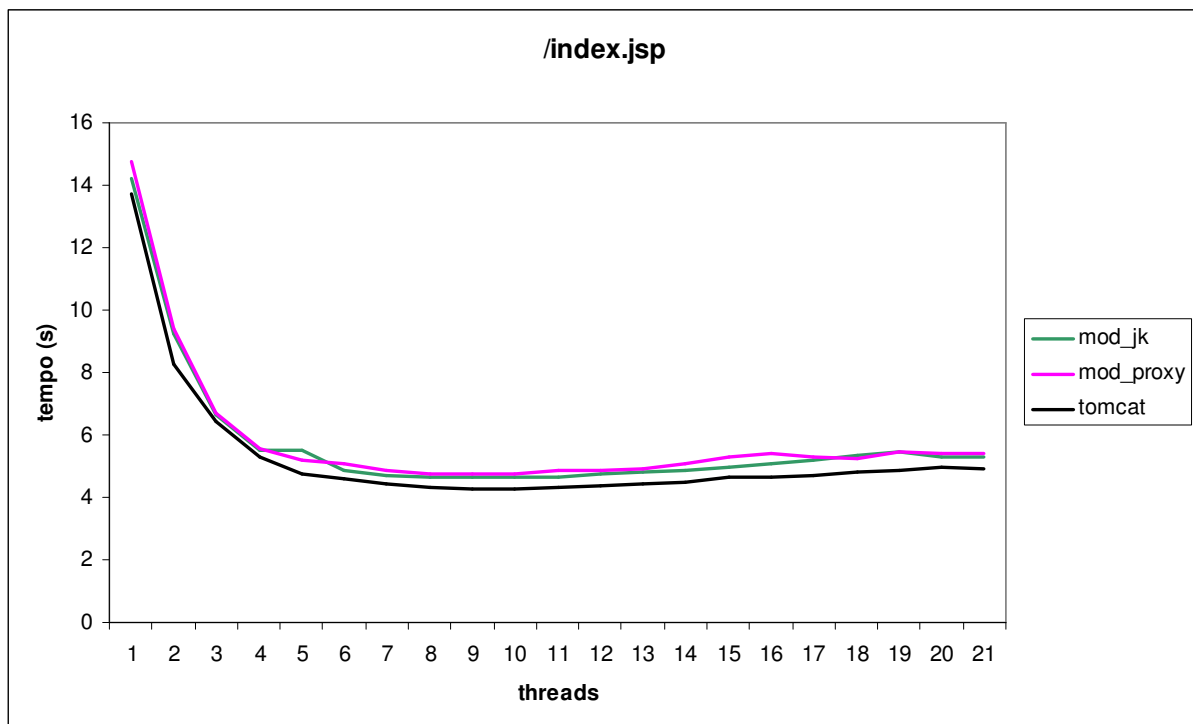


Figura 1 – Tempos de acesso ao arquivo */index.jsp*

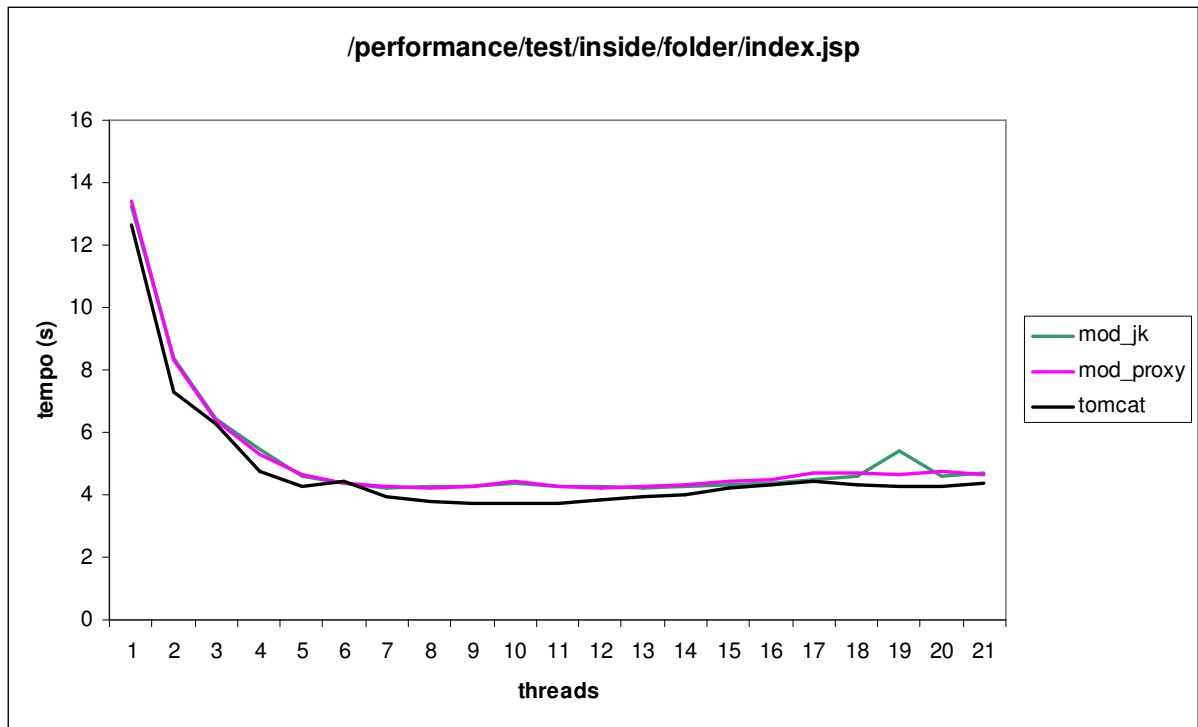


Figura 2 – Tempos de acesso ao arquivo /performance/test/inside/folder/index.jsp

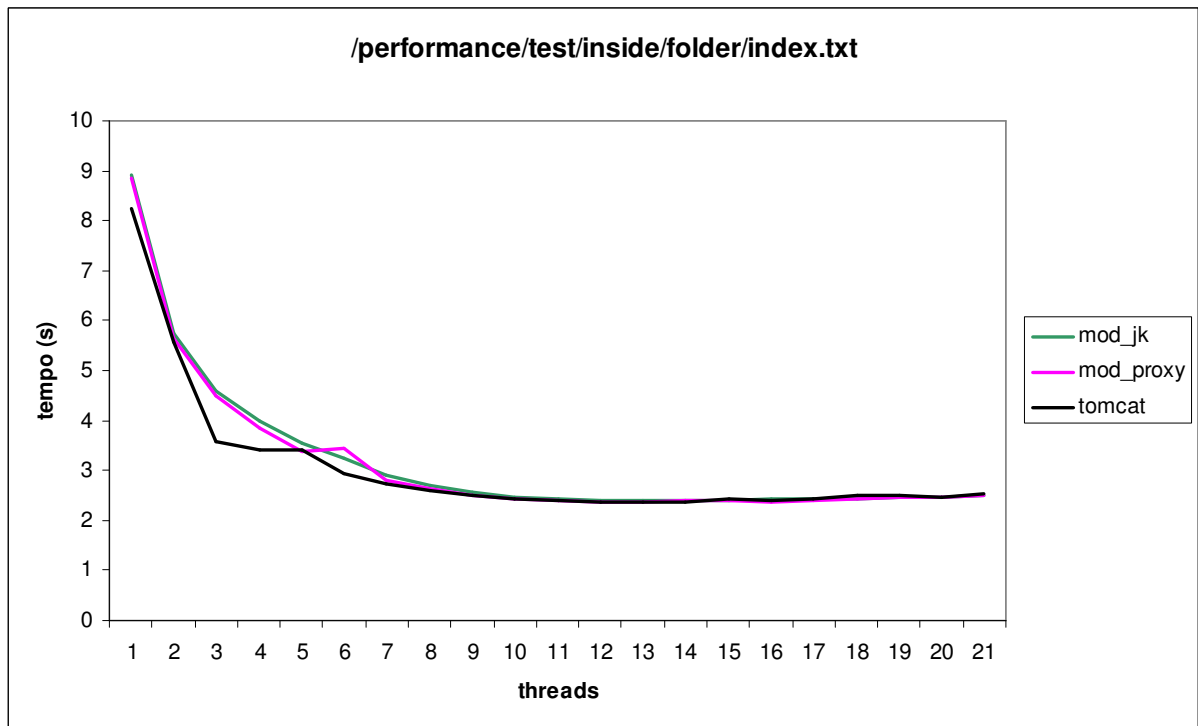


Figura 3 – Tempos de acesso ao arquivo /performance/test/inside/folder/index.txt

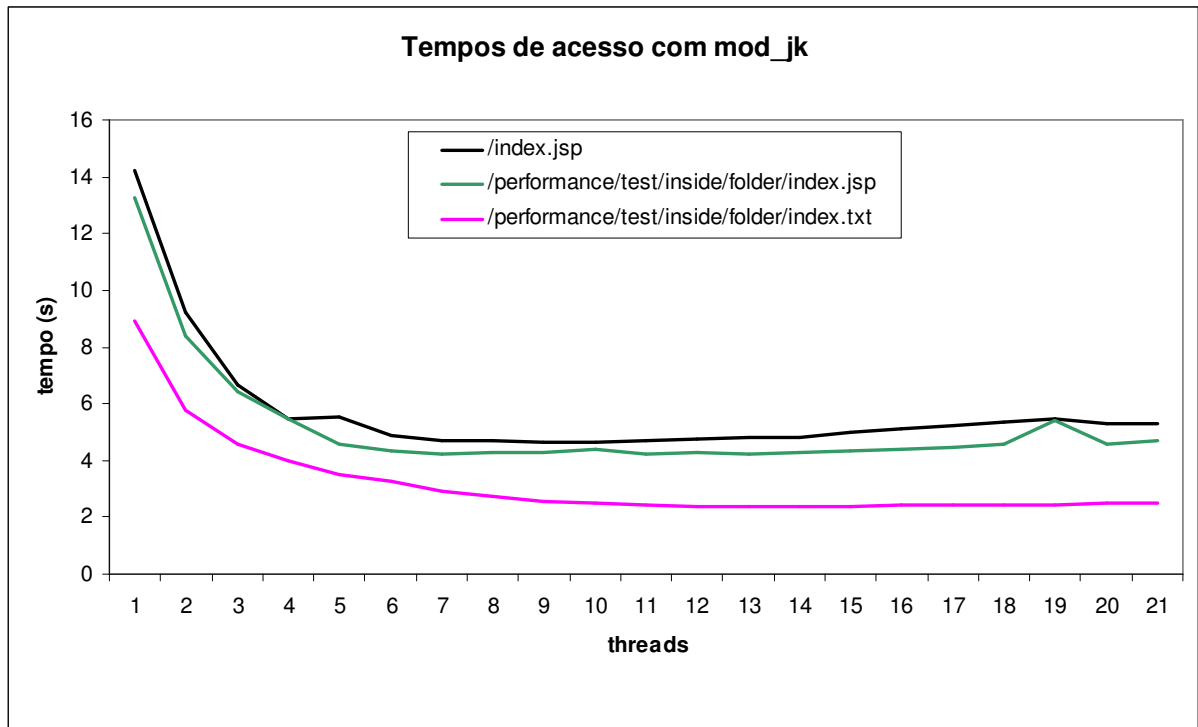


Figura 4 – Tempos de acesso aos 3 arquivos de teste utilizando mod_jk

5. Conclusão

A escolha pela utilização de um dos módulos para se remover o “:8080/opencms/opencms” da url do site não será dada pelos seus desempenhos, uma vez que eles foram iguais, e sim pela utilidade que esses módulos podem ter para o administrador do website. Assim, caso o servidor já utilize o *mod_proxy* para alguma atividade, é uma boa opção utilizá-lo, também, para a remoção do sufixo da url. O mesmo é válido para o *mod_jk*.

Caso nenhum dos módulos possua outra utilidade, recomenda-se a utilização do *mod_proxy*, que é solução sugerida pelo fabricante do *OpenCms*. No entanto, mais importante que a escolha do módulo utilizado para a modificação da url, é a correta configuração dos arquivos que serão atendidos diretamente pelo Servidor HTTP (como, por exemplo, o .txt).

6. Referências

- [1] OpenCms, the Open Source Content Management System / CMS.
“<http://opencms.org/en>”, Acesso em 20/março/2009.
- [2] BUTCHER, M., “Building Web Sites with OpenCms”, 1ª ed., Packt Publishing, 2004.
- [3] BUTCHER, M., “Managing and Customizing OpenCms 6”, 1ª ed., Packt Publishing, 2004.

- [4] LAURIE, B., LAURIE, P. "Apache: The Definitive Guide", 3ª ed., O' Reilly, 2003
- [5] OpenCms. "<http://www.opencms.org/en/development/installation/server.html>". Acesso em 20/março/2009.
- [6] Alkacon OpenCms 7 Howto: Setting up httpd, Tomcat and mod_proxy.
"http://www.opencms.org/en/modules/downloads/dl_dochow_httpd_7.0.0.html".
Acesso em 20/março/2009.
- [7] OpenCms Fórum – View Thread – Remove /opencms/opencms w.o. installing as ROOT w. multisite. "<http://www.opencms-forum.de/opencms-forum/viewthread?thread=713>".
Acesso em 20/março/2009.
- [8] ab - Apache HTTP Server benchmarking tool - Apache HTTP Server.
"<http://httpd.apache.org/docs/2.0/programs/ab.html>". Acesso em 20/março/2009.