



INTER
FACES
CIENTÍFICAS

EXATAS E TECNOLÓGICAS

ISSN IMPRESSO - 2359-4934

ISSN ELETRÔNICO - 2359-4942

DOI - 10.17564/2359-4934.2015v1n3p23-30

TESTE DE PERFORMANCE EM APLICAÇÕES WEB: GARANTINDO O DESEMPENHO DE UMA APLICAÇÃO

Thiago de Almeida Correia¹

Thiciane Couto²

RESUMO

O crescimento exponencial da Internet criou uma lacuna para algumas empresas responsáveis pelo desenvolvimento de aplicações Web. Haja vista pelas falhas que muitas dessas aplicações apresentam e a mais comum é o desempenho que não atende as necessidades reais de seus usuários. As empresas estão criando novas aplicações Web para atender os mais diferentes setores econômicos. Essas aplicações em determinadas situações, tais como, o mercado financeiro, requer como requisito mínimo uma aplicação estável e que principalmente tenha um desempenho garantido, pois caso contrário seus usuários poderão perder receitas, porque sua aplicação encontra-se com lentidão ou com conexões excedidas. As aplicações para o mercado financeiro

é apenas uma pequena parcela quando se comparado a tantas outras soluções existentes no mercado que requerem os mesmos requisitos. Visando uma aplicação que atenda as exigências do mercado com planos de testes bem definidos, essa pesquisa visa apresentar alguns métodos de performance que podem ser integrados no projeto de desenvolvimentos das aplicações Web, garantindo assim um melhor desempenho e performance.

PALAVRAS-CHAVE

Teste de performance; desempenho; conexões simultâneas; aplicação lenta.

ABSTRACT

The exponential growth of the Internet has created a gap for some companies responsible for developing Web applications. Considering the failures that many of these applications present and the most common is the performance that does not meet the real needs of its members. Companies are creating new Web applications to meet the most different economic sectors. These applications in certain situations, such as the financial market requires as a minimum a stable application and that mainly has an assured performance, otherwise your users may lose income, because your application is slowly or exceeded connections. Applications for financial market is only a small portion

when compared to many other solutions on the market that require the same requirements. Seeking an application that meets the demands of the market with well-defined test plans, this research aims to present some methods of performance that can be integrated into the development of Web applications design, thus ensuring better performance and performance.

KEYWORDS

Performance Test. Performance. Simultaneous Connections. Slow Application.

RESUMEN

El crecimiento exponencial de Internet ha creado un espacio para algunas empresas responsables del desarrollo de aplicaciones Web. Habiendo visto los defectos que muchas de estas aplicaciones presentan y el más común es el rendimiento que no cumple con las necesidades reales de sus usuarios, las empresas están creando nuevas aplicaciones web para responder a los más diferentes sectores económicos. Estas aplicaciones en determinadas situaciones, como el mercado financiero, requieren como mínimo una aplicación estable y que tenga principalmente un funcionamiento seguro. De lo contrario, sus usuarios pueden perder ingresos por su aplicación ser lenta o conexiones superadas. Las aplicaciones para el mer-

cado financiero es solamente una pequeña parte en comparación con muchas otras soluciones en el mercado que requieren los mismos requisitos. Buscando una aplicación que cumpla con las demandas del mercado, con planes de prueba bien definidos, esta investigación tiene como objetivo presentar algunos métodos de actuación que se pueden integrar en el proyecto de desarrollo de las aplicaciones Web, garantizando así un mejor rendimiento y desempeño.

PALABRAS CLAVE

Prueba de rendimiento; actuación; conexiones simultáneas; retrasos de aplicación.

1 INTRODUÇÃO

A evolução dos sistemas e a velocidade em que o seu desenvolvimento tem se dado atualmente, gerou modelos de sistemas lentos. A utilização da Internet, Intranet e extranet quebrou barreiras, mas criou problemas com o desempenho. Sistemas lentos tornam o ambiente de trabalho impraticável, criando problemas de custos à empresa, visto que o tempo dos funcionários implica na produtividade e pode estar sendo despendido em esperas desnecessárias. Para evitar isso, é necessário que os sistemas sejam testados, analisados e corrigidos, antes de sua utilização. Desta forma, a avaliação de performance de um sistema antes de seu uso é de extrema importância, visto que muitos sistemas são produzidos para uma determinada quantidade de usuários, mas no momento de uso não atendem os requisitos, causando paradas e travamento, perda de dados, lentidão no processamento ou em todo o sistema.

Normalmente o sistema é planejado para um ambiente, e este deve servir para o uso em um determinado período. O mesmo acontece com o crescimento do número de usuários, a exemplo tem-se um sistema que foi planejado para ter 10 usuários simultâneos, mas sabe-se que é esperado ter 100 usuários. O teste de performance serve não só para saber se o sistema atende aos requisitos, mas, também, até onde o sistema possui suporte a usuários e o mínimo de recursos aceitável.

Assim, esta pesquisa buscou os paradigmas atuais de teste de performance, bem como suas aplicações e a criação de métodos de utilização.

2 TESTE DE PERFORMANCE

Segundo Bastos e outros autores (2007, p. 50), o teste de desempenho “garante a velocidade de processamento de informação”. Os testes de desempenho, de acordo com Pfleeger (2004), dividem-se em teste de estresse, que tem como objetivo sobrecarregar o sistema de forma que ele não consiga mais

processá-lo. Esse teste também é conhecido como teste de volume, o teste de tempo, e visa verificar a taxa de transmissão de dados e o tempo de resposta em um teste de sobrecarga, verificando, assim, se o programa corresponde bem aos estímulos do teste. No decorrer do crescimento de acessos por clientes e funcionários aos sistemas empresariais, se faz cada vez mais importante uma aplicação com um bom desempenho que possa ser simulado por meio dos testes acima especificados.

O teste de **performance** busca possíveis falhas no sistema, que são importantes para o processo de construção do mesmo. A falha é um comportamento inesperado e pode ocorrer pelo mau entendimento do projeto, da funcionalidade que está sendo implementada, da comunicação, além de outros fatores que fazem o sistema não obter um comportamento da forma desejada. Os testes são essenciais para que o sistema aja da forma como foi proposta e, uma vez testado, está pronto para seu uso, garantindo, assim, a qualidade do **software**.

Segundo Molinari (2009, p. 50) o “Teste de **performance** tem como objetivo principal/geral demonstrar se numa carga qualquer de informação o desempenho (tempo) do aplicativo atende às metas desejadas”. Para que um teste de performance seja bem aplicado, deve-se elaborar um plano de testes, contendo o cenário sob qual o **software**/ sistema foi desenvolvido a fim de verificar se a sua meta foi atingida, ou seja, se ele se comportou da forma esperada com a quantidade de acessos impostas em seus requisitos mínimos e qual foi o limite de usuários conectados simultaneamente.

3 O JMETER

O JMeter é uma aplicação **desktop** de código aberto, 100% feita em Java desenvolvida para executar testes funcionais e medir o desempenho de aplicações. Ele foi originalmente projetado para testar aplicações **Web**, mas seu uso se expandiu para outras funções de teste. O JMeter pode ser usado para testar os seguintes tipos de servidores:

- Web – HTTP, HTTPS
- SOAP
- Database via JDBC
- LDAP
- JMS
- Mail – SMTP(S), POP3(S) and IMAP(S)
- Native commands or shell scripts

O objetivo da estratégia do JMeter é prover cenários de testes mais reais à forma de uso dos sistemas testados. Portanto, os testes de carga devem simular o mais próximo possível da realidade de utilização, cenários realistas ajudam a minimizar os efeitos da subestimação ou superestimação dos tempos de resposta das aplicações.

4 DEFINIÇÕES

Para que isso possa ser viável a ferramenta de teste de carga deve garantir os seguintes comportamentos.

1. Tempo para pensar: tempo para que o usuário pare (pense) durante suas interações com o sistema. Faz os usuários virtuais se comportarem como se fossem usuários reais. Por meio deste tempo, são injetados atrasos e paradas variáveis nos testes para simular cargas de utilização mais realistas.

2. Cache de navegador: permite que requisições estáticas como imagens, folhas de estilos, scripts entre outros sejam realizadas apenas uma vez por usuário, sabendo que o comportamento padrão dos navegadores de mercado faz cache de arquivos que não são alterados com frequência.

3. Concorrência: significa vários usuários, usando a aplicação em diversas funcionalidades no mesmo período.

4. Requisições derivadas: requisições feitas a partir de uma requisição HTTP principal, são elas as requisições para imagens, folhas de estilo e arquivos de script Javascript.

5. Número de iterações: recomenda-se que haja mais de uma iteração para cada usuário virtual, preferencialmente que o teste seja feito por um período de tempo que permita a todos os usuários

estar utilizando o sistema concorrentemente. A repetição dos testes por várias iterações permite ao sistema se adaptar ao número de usuários concorrentes, tornando os tempos de resposta mais próximos de um cenário real.

6. Temporizador de Sincronização: o temporizador de sincronização deve ser utilizado em casos que se deseje garantir que uma requisição seja executada por todos os usuários virtuais ao mesmo tempo. Normalmente a requisição a ser sincronizada encontra-se entre outras requisições que podem tirar a sincronização dos usuários virtuais, como por exemplo, o login no sistema, sendo que neste normalmente a funcionalidade que se deseja testar esta após a entrada dos usuários no sistema.

O JMeter é uma ferramenta poderosa para automatizar testes, essencialmente aqueles voltados para o mundo **Web**. Deve-se atentar para que os testes reflitam ao máximo a execução real do **software**, tanto no que diz respeito às páginas navegadas, como no tempo de intervalo de acesso de uma página para outra e também ao número de usuários simultâneos.

5 MÉTODO DE TESTE DE PERFORMANCE

Independente do sistema a ser testado, as fases do projeto são as mesmas, o que se alterará são as atividades do produto, porém todas possuem entradas e saídas a serem analisadas, sendo dividido, segundo Molinari (2009, p. 65) em: “Definição do processo, planejamento de testes e cenários, construção de cenários, executando cenários e analisando cenários”.

Para um teste de **performance** ser considerado válido deve ser elaborado um cenário com base nos requisitos pleiteados no desenvolvimento da aplicação, de modo que venha a utilizar ao máximo os recursos necessários e a avaliação da quantidade de acessos simultâneos suportadas na aplicação.

Quando executado um procedimento de teste deve ser considerado o tempo de resposta da aplicação em diversos níveis de acessos simultâneos.

6 FÓRMULAS A SEREM ANALISADAS

Como seria viável obter-se 1000 usuários para fazer o teste simultaneamente, é necessário, então, utilizar-se sistemas que possuem suporte a usuários virtuais, possibilitando a utilização de 10, 20 ou mais usuários, testando o sistema simultaneamente. Para saber se a rede de entrada e saída do computador suporta 100 usuários no computador, pode-se utilizar a fórmula apresentada na Tabela 1:

Tabela 1 – Quantidade de usuários virtuais

$$VT=(VR-20\%)/UV$$

Onde:

VT = Velocidade de teste

VR = Velocidade rede

UV = Quantidade de usuários virtuais por maquina.

Com isto é possível saber qual a velocidade que cada usuário utilizaria realmente na rede. Sobre o funcionamento, inicialmente, verifica-se a velocidade da rede atual em kbps, mbps etc, seguido por subtrair 20% que é o valor que o sistema Windows subtrai automaticamente da velocidade da rede, e divide-se pela quantidade de usuários virtuais a serem aplicadas. Mas, como avaliar a velocidade de um sistema básico que trafega pela Internet? Apresenta-se a fórmula da Tabela 2:

Tabela 2 – Velocidade que trafega pela Internet

$$TPW=(\sum(N \text{ Objetos}) + tchttp)$$

Onde:

TPW = tamanho da pagina web

N Objetos = objetos embutidos na pagina web

Tchttp = tamanho cabeçalho HTTP

Com os valores obtidos, é possível saber a velocidade da rede no teste, e se é possível criar uma simulação real do ambiente.

Após a realização de cada etapa obtém-se como resultado os dados ou gráficos, que devem ser adicionados ao relatório antes de continuar os próximos testes, a fim de obter resultados comparativos e as demonstrações para o cliente do ambiente do teste. Esta etapa conclui a etapa de testes, passando-se para a próxima, de pós-teste.

7 ANÁLISE DE RESULTADOS E CRIAÇÃO DE RELATÓRIOS

O que é preciso nesta etapa é reunir os resultados de cada etapa dos testes realizados, traçando um comparativo de performance. Para a criação do relatório é importante uma boa noção dos conceitos estatísticos para que a análise resulte em dados mais precisos. A análise deve considerar os dados de tempo gasto pelos usuários virtuais na realização dos testes e o tempo da execução do fato, tempo de paralisação.

Com base nos resultados dos testes, procede-se a média dos testes, que aponta para a tendência de comportamento dos dados, conforme exemplo da Tabela 3:

Tabela 3 – Média dos testes

$$\text{Média}=(\text{Teste1} + \text{Teste2} + \text{Teste3} + \dots + \text{TesteN})/N=\sum X/n$$

O relatório deve possuir a média de vazamento, o vazamento real de cada teste, o tempo e valor de vazamento, a mediana e os cálculos de desvio padrão.

O desvio padrão é o valor que representa a dispersão dos eventos sob distribuição normal. A fórmula para o cálculo está na Tabela 4:

Tabela 4 – Desvio padrão

$= \sqrt{(\sum(X_i - X)^2) / (n - 1)}$ <p>Onde: (sigma) = desvio padrão Xi = valor de cada dado coletado (Teste1, Teste2 ... TesteN) X = média aritmética dos valores Xi (Feito com a fórmula de média) N = quantidade de dados da amostra</p>
--

Com base nas fórmulas apresentadas, deve-se realizar a criação dos relatórios de análise dos resultados.

8 CONCLUSÃO

Os testes fazem com que se obtenha um melhor retorno de investimento, pois é possível prever o suporte futuro do sistema, atendendo-se, assim, as necessidades futuras de desenvolvimento do **software**.

O teste de performance garantirá, então, a qualidade no quesito **performance** do sistema. O paradigma aqui apresentado possui facilidade de implementação e utilização de poucos recursos técnicos e de **software**, sendo que o ponto de destaque para o sucesso na implementação do modelo de teste de performance é a qualificação do testador e do analista, os quais devem possuir a capacidade analítica e conhecimentos consistentes sobre **hardware**, teoria das filas, redes, protocolos, SOAP dentre outros temas, a fim de fazer que o **software** corresponda ao que estabelecido como expectativa, garantindo qualidade de **software**.

REFERÊNCIAS

BECK, Kent. **TDD Desenvolvimento guiado por testes**. Porto Alegre: Bookman, 2010.

MOLINARI, Leonardo. **Teste de Performance**. São Paulo: Visual Books, 2010

PFLEEGER; Shari Lwarence. **Engenharia de Software: Teoria e Prática**. 2.ed. São Paulo: Prentice Hall, 2004.

SOMMERVILLE, Ian. **Engenharia de Software**. 6.ed. São Paulo: Prentice Hall, 2003.

USO DO JMETER PARA TESTES DE DESEMPENHO NA WEB. Laboratório I Móbilis Computação Móvel. 27 de agosto de 2012. Disponível em: <<http://www.decom.ufop.br/imobilis/?p=1146>>. Acesso em: 29 ago. 2013.

Recebido em: 12 de Abril 2015
Avaliado em: 18 de Abril 2015
Aceito em: 4 de Maio de 2015

1. Especialista em Redes, Professor Faculdade Impacta de Tecnologia – São Paulo/SP, Membro do GPITIC. E-mail: thiago@thiagocorreia.com.br
2. Graduanda em Ciências da Computação Unit/e membro do GPITIC – Aracaju/SE. E-mail: thicianecouto@gmail.com

