



Home Textos Notícias Fórum Vídeo Download Login/Entrar Cadastro

Procurar
search...

Novos Textos

Concepção de Negócio Competitivo e Lucrativo

Preços e a Proposição de Valor

Os dez Pontos Chaves de Decisão em Vendas – PCDV

Muito cuidado com Negociação de Preço

Mitos sobre os Preços

A Importância do Tempo de Vida do Cliente (TVC) para a Empresa

Os Mitos e as Realidades sobre a Inovação

Como criar novo Espaço no Mercado para Ganhar Dinheiro

A Evolução dos Mercados e a Dinâmica da Concorrência

Mitos e Realidade do Empreendedorismo

Hierarquia ou Pirâmide do Empreendedorismo

Estrutura Organizacional e Forma de Atuação

Como utilizar as sete (7) Características Empreendedoras

Como diminuir os Riscos na Criação de Empresas pelo Planejamento e Acompanhamento do Plano

Teste de Software

Processos, Técnicas, Ferramentas Prep p/ Onde a sua empresa mais gasta? Obtenha a resposta - Teste Grátis

Software Financeiro Web

Anúncios Google

TUTORIAL DE INSTALAÇÃO E CONFIGURAÇÃO DO JMETER

Após leitura, ajude a classificar este artigo.

Nível de qualidade: / 1

Ruim Ótimo

Postado por: Márcio Santos

Friday, 28 November 2008

Os Testes de Performance, tem como finalidade garantir a integridade e assegurar a qualidade de um sistema ou site, tendo como objetivos principais encontrar erros e corrigi-los. Servem também para obter medida de requisitos não funcionais do software, tais como confiabilidade ou desempenho, usando-se técnicas estatísticas apropriadas. A realização de testes tem por objetivo determinar se o desempenho do sistema integrado é adequado, de acordo com os requisitos do sistema. Testes de performances podem ser realizados ao longo do desenvolvimento, mas somente quando todos os componentes são finalmente integrados é que se pode ter uma medida real do seu desempenho, mostrar a relação entre risco de teste versus requisito de qualidade em aplicações Web, ou seja, quais os requisitos de qualidade que mais comprometem o correto funcionamento destes sistemas caso não sejam corretamente testados. Segundo Molinari, o desempenho é o requisito de qualidade que deveria ter mais cuidado em aplicações Web. Esta importância é ainda mais crítica em aplicações Web tendo em vista o grande volume de dados gerenciados.

JMETER

JMETER é uma ferramenta do grupo apache, para a realização de testes de performance, carga e stress. Apesar de ser este o foco do JMETER (testes de performance, carga e stress) ele também pode ser utilizado para realizar testes em web services, banco de dados e também automatizar alguns teste funcionais, seu uso, alias, é bastante amplo. Lógico que existem ferramentas que desempenham o mesmo papel que o JMETER como o caso WAST (Web Application Stress Test), WebLoad mas vamos abordar JMETER neste momento. O JMETER em sua operação mais básica e nos posts decorrentes irei aprofundando na sua aplicação. Quem vê o JMETER pela primeira vez não deve se sentir muito confortável devido a sua relativa falta de intuitividade, mas isso logo passa depois que você entende sua estrutura, que pelo que vocês irão ver, que é bem simples. A melhor forma de entender a estrutura de elementos da árvore do JMETER é mostrando logo de início um script do JMETER todo pronto, pois assim fica mais prático. Irei portanto, usar como exemplo um script que escrevi para alguns testes de performance e stress, são scripts simples que mostram como é fácil usar esta ferramenta. Antes, irei mostrar onde se encontram os elementos que serão utilizados no exemplo que irei dar, veja na imagem abaixo como chegar neles. O JMETER permite diversas requisições tais como

PRE-REQUISITOS

A versão mais recente do JMeter, usada na época da elaboração deste artigo, é a 2.3.2. Para sua instalação temos os seguintes pré-requisitos:

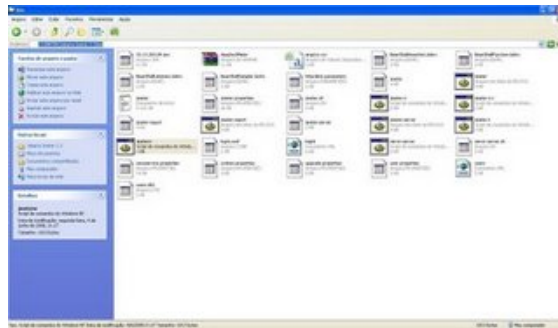
1. JDK 1.4 ou superior corretamente instalado. Você poderá baixar o ambiente Java diretamente na Sun Microsystems pelo endereço <http://java.sun.com/j2se/1.5.0/download.jsp>.
2. Um sistema operacional que execute uma JVM compatível com o ambiente descrito acima. Exemplo: Microsoft Windows (XP, 2003), Solaris, AIX, Linux, etc.
3. Dependendo da carga que será testada, um ou mais equipamentos com boa capacidade de processamento e memória.

PASSO A PASSO

O JMeter vem empacotado no formato zip ou tgz (tar). Optamos por fazer o download no formato zip por ser mais conhecido pela maioria dos usuários no site http://jakarta.apache.org/site/downloads/downloads_jmeter.cgi. O segundo passo na instalação do JMeter é descompactar o arquivo em uma pasta. Para ilustrar esse procedimento, optou-se por descompactar o JMeter no diretório raiz do disco C em um ambiente Microsoft e executar a partir C:\JMeter\Jakarta Jmeter 2.3\bin\jmeter.bat:

Comentários

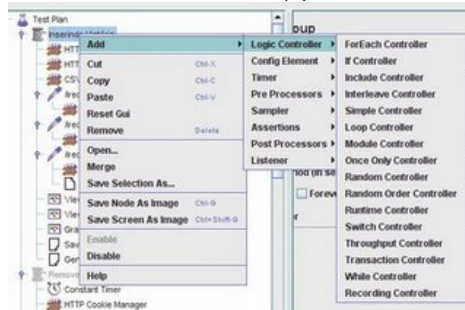
- Gliard
como eu faço para rodar o windows no pen drive
Inês
Amigo Filósofo! A função da Filosofia é qu...
Sidney Souza
Você não terminou essa dica, faz tudo mas n...
GILBER CARVAJAL
gostaria de conhecer melhor todos vcs e com...
Joel Wittes Narciso
Pessoal! Era o que estava faltando na área...
...
Wilde Ramon Rosa
Fico muito agradecido por esta iniciativa d...
jiloccha
muito boa!
jsilva
Ola! Este produto é relativamente nov o e ai...
Leomar Gonçalves Ara...
Primeiro quando se fala em gerenciar riscos...
Ronaldo
Amigo, existe alguma loja aqui no Brasil q ...
Nelson Rodrigo
Ola!! preciso enrolar um transformador p...
Jorge Toledo
Dou meus parabens pela iniciativa, se todos...
Eduardo Schubert
gostei da materia, e gostaria de colaborar ...
Renato Oliveira Vil...
Seria dificil encontrar este tester no...
Ermane
Muito Obrigado pelo material e Parabéns pel...



Bem agora vamos para o script com os elementos utilizados

Vamos na ordem:

1. Inserindo História - Thread Group (**Add > New > Thread Group**)



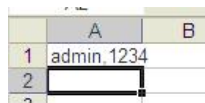
Este é o começo de qualquer script (ou plano de teste, se preferir) no JMETER, este elemento inicia seu script, todos os outros elementos que serão interpretados pelo JMETER devem ficar obrigatoriamente abaixo deste thread group, você pode, se preferir, organizar seu script em vários thread groups sem problemas. É aqui no Thread Group que você irá indicar número de conexões simultâneas, Ramp-up e Tempo de Execução.



2. HTTP Request Defaults (Thread Group > Add > Config Element > HTTP Request Defaults) Aqui você define o padrão de seus requests HTTP, este elemento basicamente gerencia os HTTP requests que essa thread está prestes a fazer. O que basicamente deve estar preenchido neste elemento é o campo **Server Name or IP**.

3. HTTP Cookie Manager (Thread Group > Add > Config Element > HTTP Cookie Manager) Suporte a Cookies, você pode remover este elemento, mas recomendo deixar. Este elemento garante que cada Thread vai receber seu próprio cookie.

4. CSV Data Set Config (Thread Group > Add > Config Element > CSV Data Set Config)



Esta aqui é uma facilidade do JMETER, em telas de login ou qualquer tela que você tenha que passar dados para o browser, você pode usar este elemento, ele funciona da seguinte forma. Você preenche arquivo .csv com os parâmetros que você vai querer passar, conforme imagem abaixo. Salve e depois volte ao elemento CSV Data Set Config e informe respectivamente nome para os valores acima *admin* e *1234*, conforme imagem abaixo

Variable Names (comma-delimited): USER,PASS

Neste momento, ao rodar o script o JMETER vai vincular *USER=admin* e *PASS=1234*, e você pode chamar estes valores em qualquer lugar do script assim: *\${USER}* e *\${PASS}*. No caso deste script irei utilizar estes valores no campo abaixo. **5.** ./redes2/login.do

Aqui é onde começa a interação com o browser, ou seja, onde as requisições são efetivamente realizadas, existe uma forma bem fácil de conseguir estas informações, usando uma funcionalidade chamada **HTTP Proxy Server**.



Com o **HTTP Proxy Server** você pode gravar ações no browser e o JMETER irá gravando tudo pra você, basta que você informe nas configurações de seu browser a porta 8080 (você pode alterar para qualquer porta disponível no seu pc) e pressionar PLAY



Pronto, feito isso, o JMETER deverá lhe poupar de realizar manualmente estes passos. Como mencionado no tópico 5) irei usar os valores *\${USER}* e *\${PASS}* aqui.

Name:	Value
login	\${USER}
password	\${PASS}
status	1

6. Browser-derived headers

Estes controles o **HTTP Proxy Server** adiciona automaticamente, não se preocupe com eles.

7. ./redes2/viewHome.do

Mais um request gravado pelo **HTTP Proxy Server**

8. Browser-derived headers

Estes controles o **HTTP Proxy Server** adiciona automaticamente, não se preocupe com eles.

9. ./iredes2/confirmCreateNarrative.do

Mais um request gravado pelo **HTTP Proxy Server**

10. Browser-derived headers

Estes controles o **HTTP Proxy Server** adiciona automaticamente, não se preocupe com eles.

11. Counter - (Thread Group > Add > Pre Processors > Counter)

Esta aqui é outra facilidade do JMETER, quando existe a necessidade de se incluir vários registros seqüencialmente no banco, a melhor forma de se fazer isso é via estes Counters, ele vai basicamente incrementar em +1 (isto no entanto é configurável)

Counter

Name: Counter

Comments:Incrementa a História +1

Start 1

Increment 1

Maximum 100

Number format

Reference Name H

☐ Track counter independently for each user

No caso acima ele vai começar com 1 incrementar de 1 em 1 até 100 e o nome de referência é H, ou seja, onde você quiser utilizar este counter, basta chamar `$(H)` onde queres incrementar, veja imagem abaixo.

selectedContacts	
selectedContactGroups	
name	História \$(H)
narrativeDescription	task \$(H) - robot running...
visibilityLevel	0
tags	

12. View Results in Table - (Thread Group > Add > Listener > View Results in Table) Bem aqui é a parte dos relatórios, este relatório exibirá os seguintes valores.

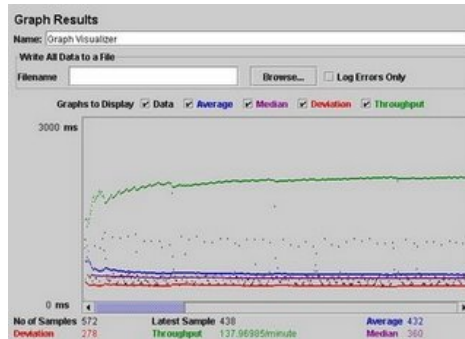
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes
----------	------------	-------------	-------	-----------------	--------	-------

13. View Results Tree - (Thread Group > Add > Listener > View Results Tree) Os resultados estão vazios, mas ele lista aqui todos os requests e alem disso aqui você também pode renderizar o HTML e ver exatamente o que este request fez, ótimo para debugar caso ocorra algum problema.

Filename

Sampler result

14. Graph Results - (Thread Group > Add > Listener > Graph Results)



15. Save Responses to a file - (Thread Group > Add > Post Processors > Save Responses to a file) Salva requisições em um arquivo. No **Graphs to Display** esta identificado abaixo o que cada medidor significa:

DATA - Traça os valores de dados reais.

AVERAGE - Traça a media

MEDIAN - Traça o mediano (o valor intermediário)

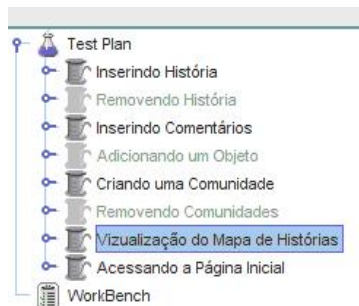
DEVIATION - Traça o desvio padrão (uma medida da variação)

THROUGHPUT - Traça o número de amostras por a unidade de tempo. (definição sobre o que é

"Throughput" no site <http://br.answers.yahoo.com/question/index?qid=20060731114758AAFAaRX>)

As Indicadores individuais na parte inferior da exposição são os valores atuais. " **Latest Sample**" o mais atrasado; O tempo decorrido atual da amostra é mostrado no gráfico como o " **Data**".

16. Generate Summary Results - (Thread Group > Add > Post Processors > Generate Summary Results). Salva um sumário em um arquivo. Você pode organizar tudo em várias threads desta forma.



E antes de finalizar esta parte que é a mais importante

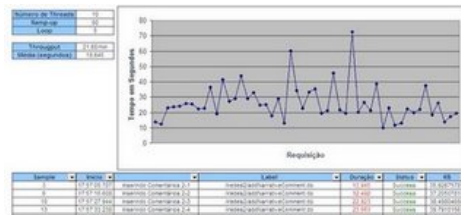
Estas são as propriedades de uma Thread, você deve informar nela o que você vai fazer:

Number of Threads - Quantidade de Threads que será realizada

Ram-UP - Tempo que o JMETER para executar as threads informada acima

Loop - Quantas vezes ele irá realizar esta atividade

No caso acima 10 requisições em 60 segundos repetindo 5 vezes



No final consolidado estes resultados em uma planilha

Ferramentas alternativas para testes de performance no Linux utilizado geralmente em servidor Apache.

Uma alternativa é a ferramenta **OpenSTA**, distribuída de forma popular, software de fonte aberta que realiza o teste de regressão, de carga e de esforço. Também conhecida como alternativa do pobre homem a LoadRunner & a seda. Utilizar emulação de vinho, não existe nenhuma evidência documentada de não ter funcionado com sucesso.

LDTP- Projeto de teste do Desktop Linux, criado para produzir a estrutura de automatização de testes de alta qualidade, e com ferramentas pioneiras, que são utilizadas para testar desktops com Linux e melhorá-los. Ele utiliza bibliotecas de acessibilidade, e realizar suas funções através de interfaces do utilizador de suas aplicações. Possui ferramentas para gravar test-cases que se baseia em usuário-seleção na aplicação.

Espere do NIST - está ferramenta utilizada para automatizar aplicações interativas, exemplo: telnet, FTP, passwd, fsck, rlogin, ponta, ssh, etc. é utilizado também para testar estas mesmas aplicações, e adicionando o TK também se pode envolver aplicações interativas em X11 GUIs. Esta ferramenta faz facilmente as sortes das tarefas que são classificadas proibitivas difíceis com qualquer outra coisa, é uma ferramenta inestimável, com ela o usuário poderá automatizar as tarefas mais difíceis de uma forma muito fácil. O Espere também realiza testes de Unix, e foi escrito por Don Libes, para aplicações interativas.

DejaGNU - é uma estrutura para testar outros programas. Tem como finalidade fornecer uma única parte frontal para todos os testes. É como uma biblioteca feita sob encomenda dos procedimentos do Tcl crafted para suportar a escrita de um chicote de fios de teste. Um chicote de fios de teste nada mais é que a infra-estrutura do teste que é criada para suportar um programa ou uma ferramenta específica. Cada programa pode ter os test suites múltiplos, suportados toda por um único chicote de fios de teste. DejaGnu é escrito dentro espera, que usa por sua vez o Tcl - utilize ferramentas a língua de comando. Baseou-se na estrutura da automatização do teste da regressão e de carga.

As alternativas comerciais são:

ProxySniffer - a ferramenta trabalha a partir do proxy mede o período de resposta e a estabilidade das aplicações web sob condições de carga reais - simulando centenas ou milhares de usuários de correia fotoreceptora.

PureLoad - é uma ferramenta de teste da carga que simula uma quantidade grande de usuários que executam pedidos de encontro às aplicações baseadas usuário. Utilizado para verificar se a aplicação encontrará os critérios de desempenho previstos. Ele emite relatórios informando problemas de qualidade e de desempenho e também estatísticas detalhadas extraídas durante o teste de carga. Realiza também sustentação extensiva para facilitar a gravação e teste de aplicações baseadas nas seqüências de fotoreceptora.

CONCLUSÃO

Os testes de performance são de bastantes importância, pois analisa um determinado trafego e um melhor desempenho da rede, assim como detecta falhas para que seja projetadas novas soluções que melhorem o desempenho da aplicação executada. O JMeter é uma ferramenta que executa este teste, sendo de fácil implementação, há diversas no mercado mais complexas mais o JMeter desempenha muito bem seu papel.

COMENTÁRIOS

[ADICIONAR NOVO](#)
[BUSCA](#)

Sidney Souza - **Assunto incompleto**


2009-04-14 19:14:16

|200.247.49.xxx |

Você não terminou essa dica, faz tudo mas não conclui ai fica difícil. depois que faço todas essas etapas, vou pra onde! aperto aonde?

Responder

ESCREVER UM COMENTÁRIO

Nome:	
E-mail:	
Título:	
<input type="button" value="Enviar"/>	
Por favor coloque o código anti-spam que você lê na imagem.	
Powered by !JoomlaComment 3.26	

[< Prev](#) [Next >](#)

TECNOCIÊNCIA