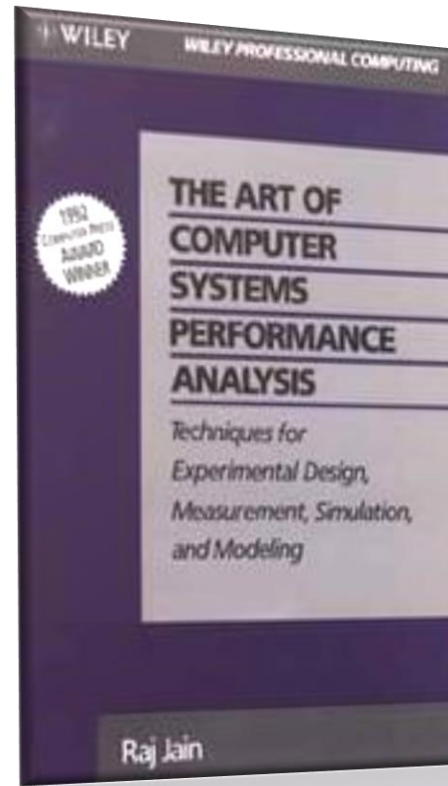
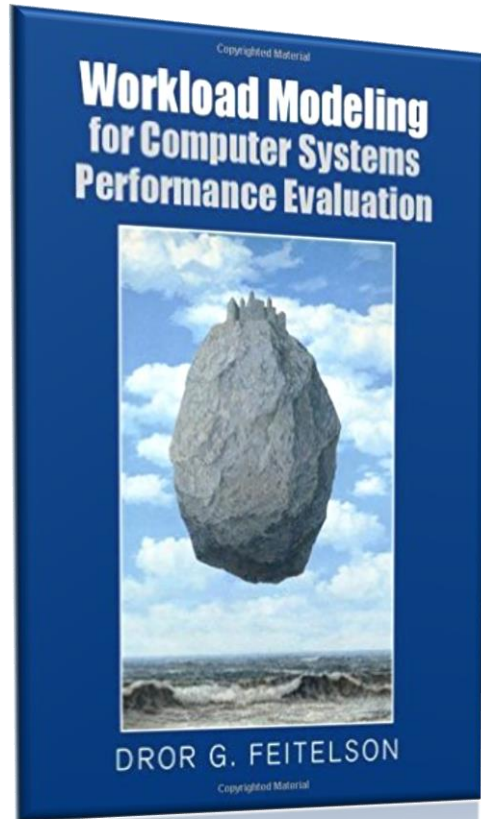




Geração e Modelagem de Carga de Trabalho (Workload)

Aline Oliveira aso2@cin.ufpe.br
Camila Araujo cga2@cin.ufpe.br
Iure Fé isf2@cin.ufpe.br

Referência



Teste de Desempenho

- ✓ O **Teste Desempenho** em sua mais pura definição é o tipo de teste realizado para se verificar o tempo de resposta de uma aplicação, determinando assim a sua escalabilidade e confiança levando-se em consideração uma carga.



Teste de Desempenho

- ✓ Avaliar a infraestrutura;
- ✓ Avaliar a capacidade atual;
- ✓ Determinar a estabilidade;
- ✓ Avaliar a release;
- ✓ Avaliar o tempo de resposta;
- ✓ Verificar se aplicação suportar o workload pretendida;
- ✓ Verificar se aplicação possui estabilidade para aguentar o crescimento do workload.





Teste de Desempenho - Propósito

- ✓ Determinar a probabilidade que o sistema irá atender aos **SLA's** (*Service Level Agreement*) acordados;
- ✓ Não mitiga o risco diretamente, mas identifica e quantifica o risco;
- ✓ Determinar tempos de resposta em *throughputs*;
- ✓ Determinar gargalos (bottlenecks) no sistema;

Teste de Desempenho – Benefícios

- ✓ Melhoria da qualidade do ponto de vista do usuário;
- ✓ Redução do custo de mudanças;
- ✓ Redução dos custos de sistema;
- ✓ Identificação antecipada dos defeitos mais críticos da aplicação como arquitetura do sistema.



Workload

- ✓ Conjunto de todas as requisições de serviço (processamento) submetidas a um sistema durante um período de tempo.
- ✓ Capaz de mostrar, capturar e reproduzir um comportamento carga de trabalho real e suas funcionalidades mais importantes.





Workload

- ✓ A carga de trabalho utilizada para comparar dois sistemas devem ser representativas em relação ao uso dos sistemas em campo.

Exemplo:

Um sistema real, os pacotes de uma rede são geralmente de dois tamanhos, um curto e outro longo, a carga de trabalho utilizada para aferir este sistema dever conter as mesmas características do sistema real.

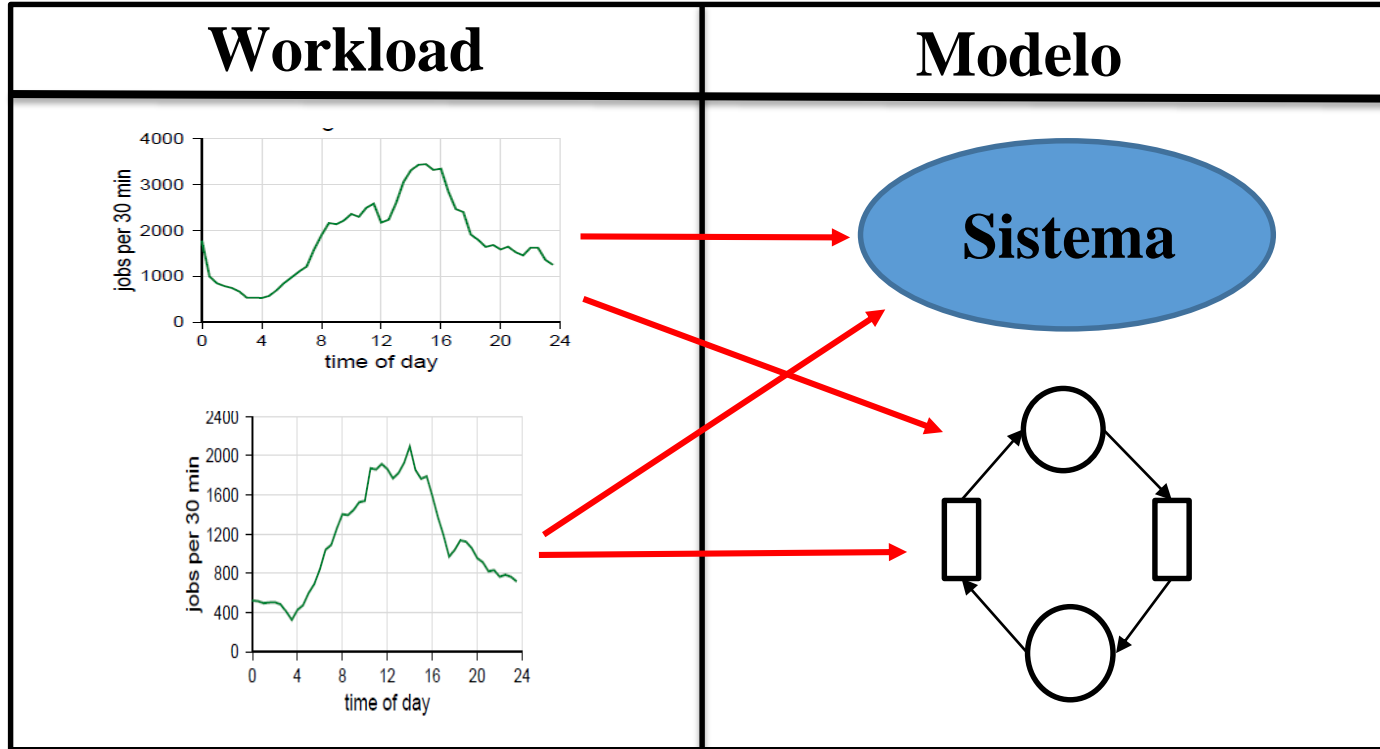


Workload Real e Sintética

Real	Sintética
Operação normal	Operação não normal
Base de dados grande	Base de dados menor
Dados sensíveis	Dados não sensíveis
Não pode ser modificado	Facilmente modificada sem impactar sua operação
Não suporta portabilidade	Suporta portabilidade
Sem recursos internos de medição	Com recursos internos de medição
	Pode ser aplicado várias vezes de maneira controlada



Workload Real e Sintética





Tipos de Workloads

Addition Instruction

Instruction Mixes

Kernel

Synthetic Programs

Application Benchmarks

Addition Instruction

- ✓ Processadores eram os componentes mais caros e mais usados do sistema
- ✓ Máquinas mais simples;
- ✓ Conjunto de instruções reduzidas;
- ✓ Instruções de adição era a mais frequente;
- ✓ Tempo de adição métrica definida.



Tipos de Workloads

Instruction Mixes

- ✓ Crescimento do processamento;
- ✓ Instruções de adição não era suficiente;
- ✓ Instruction mix = Instruction + Frequência de uso;
- ✓ Média das instruções para comparar velocidade de processadores;
- ✓ Desenvolvido o Mix de Gibson.

Métricas de Performance

- ✓ MIPS = Millions of Instructions Per Second;
- ✓ MFLOPS = Millions of Floating Point Operations Per Second.



Tipos de Workloads

1. Load and Store	31.2
2. Fixed-Point Add and Subtract	6.1
3. Compares	3.8
4. Branches	16.6
5. Floating Add and Subtract	6.9
6. Floating Multiply	3.8
7. Floating Divide	1.5
8. Fixed-Point Multiply	0.6
9. Fixed-Point Divide	0.2
10. Shifting	4.4
11. Logical, And, Or	1.6
12. Instructions not using registers	5.3
13. Indexing	<u>18.0</u>
	100.0



Tipos de Workloads

Instruction Mixes - Desvantagem

Com a inovação da tecnologia de computador, as instruções de hoje são mais complexas e essas alterações não são refletidas na *Instruction Mixes*:

- ✓ Modos de endereçamento;
- ✓ Interferência de outros dispositivos durante os ciclos de acesso ao processador / memória;
- ✓ Número de vezes que um ramo condicional é utilizado.

Tipos de Workloads

Kernel

- ✓ Devido à introdução de novos mecanismos, como vários esquemas de endereçamento de cache e pipeline, a execução em tempo de instrução é muito variável;
- ✓ Tempo de execução de uma função;
- ✓ Kernels utiliza: Sieve, Puzzle, Pesquisa em árvore, função de Ackerman, inversão de matriz e ordenação;

Desvantagens: Não faz uso de dispositivos de entrada/saída.

Synthetic Programs

- ✓ Para medir o desempenho de I/O, os analistas utilizam loops exercitadores;
- ✓ O primeiro loop exercitador foi criado por Buchholz (1969) que o chamou de programa sintético.



Tipos de Workloads

Vantagens:

- ✓ Não há dados reais de arquivos;
- ✓ Facilmente modificados e transportados para diferentes sistemas;
- ✓ O processo de medição é automatizado.

Desvantagens:

- ✓ Mecanismos de falhas de página e cache de disco podem não se adequadamente exercitados;
- ✓ A sobrecarga da CPU-I/O pode não ser representativa.



Tipos de Workloads - Benchmarks



Benchmarks



Weicker (2002), o que torna um programa um benchmark é o fato de que esse programa é padronizado (contendo especificações detalhadas), tendo sido projetado ou selecionado para executar em diferentes sistemas computacionais com o objetivo de realizar comparações justas do desempenho desses sistemas.



Os benchmarks são programas utilizados para avaliar hardware, software ou sistemas computacionais completos. Esses programas definem cargas de trabalho que representam as características essenciais de um domínio de aplicação.

Tipos de Workloads - Benchmarks

- ✓ Benchmark = Carga de Trabalho Sintética;
- ✓ Alguns autores: Benchmark = São conjuntos de programas que geram cargas de trabalho sintéticas semelhantes as reais.
- ✓ Os benchmarks de aplicação oferecem melhores medidas para a performance no "mundo real" para alguns sistemas.



Tipos de Workloads

Linpack

- ✓ É constituído por um certo número de programas que resolvem sistemas densos de equações lineares usando o pacote de sub-rotina de descompactação.
- ✓ Os programas podem ser caracterizados como tendo uma elevada percentagem de ponto flutuante de adições e multiplicações.
- ✓ São utilizadas para avaliar desempenho de aplicações científicas e de engenharias.

Tipos de Benchmarks

Lawrence Livermore Loops ou LLLoops

- ✓ É constituído de 24 testes;
- ✓ Grandes cálculos científicos;
- ✓ Resultados complexos dados em MFLOPS;
- ✓ Aplicação em larga escala (Laboratórios de Química e Física).
- ✓ Os resultados deste conjunto de benchmarks são bastante complexos de interpretar, porque não há nenhuma tentativa de reduzir.





Tipos de Workloads

Whetstones: Benchmark programa que testa o desempenho de um sistema de números fracionários (de ponto flutuante). O teste de CPU é limitado e não realiza chamadas de E / S.

Tipos de Benchmarks

```
#####  
  
Assembler CPUID and RDTSC  
CPU GenuineIntel, Features Code BFEBFBFF, Model Code  
00040651  
Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz  
Measured - Minimum 2394 MHz, Maximum 2394 MHz  
Linux Functions  
get_nprocs() - CPUs 4, Configured CPUs 4  
get_phys_pages() and size - RAM Size 3.77 GB, Page Size  
4096 Bytes  
uname() - Linux, warley-Vostro, 3.13.0-32-generic  
#57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014, x86_64  
  
#####
```

```
Whetstone Single Precision C Benchmark No Opt 64 Bit, Sun Nov  
29 16:11:58 2015
```

Loop content	Result	MFLOPS
MOPS Seconds		
N1 floating point	-1.12475013732910156	567.969
0.073		
N2 floating point	-1.12274742126464844	543.642
0.533		
N3 if then else	1.00000000000000000	
1301.018 0.171		
N4 fixed point	12.00000000000000000	
1154.565 0.588		
N5 sin,cos etc.	0.49911010265350342	
93.751 1.912		
N6 floating point	0.99999982118606567	296.263
3.922		
N7 assignments	3.00000000000000000	
384.239 1.036		
N8 exp,sqrt etc.	0.75110864639282227	
44.347 1.807		
MWIPS		2145.308
10.041		



Conheça outros Benchmarks

- ✓ U.S. Steel
- ✓ Doduc
- ✓ TOP
- ✓ Lawrence Livermore Loops
- ✓ Digital Review Labs
- ✓ Abingdon Cross Image-Processing
- ✓ Benchmark
- ✓ Access time
- ✓ Average seek time
- ✓ FLOPS
- ✓ FPS
- ✓ Random access/read speed
- ✓ WebTP
- ✓ NpBench
- ✓ Sieve kernel
- ✓ Ackermann's function
- ✓ Lawrence Livermore loops
- ✓ Debit-Credit Benchmark
- ✓ NAS Parallel Benchmarks
- ✓ PARKBENCH
- ✓ SPLASH
- ✓ COMSS
- ✓ Bit to the User
- ✓ TinyBench
- ✓ Express
- ✓ Parellel Virtual Machine
- ✓ GENESIS
- ✓ SPEC

http://www.benchmarkhq.ru/english.html?/be_net.html



Seleção da Carga de Trabalho

- A coleta de dados da carga de trabalho é crucial para a modelagem de desempenho
- Escolha correta da carga de trabalho requer análise.
- A fim de executar medições significativas, a carga de trabalho deve ser cuidadosamente selecionada.



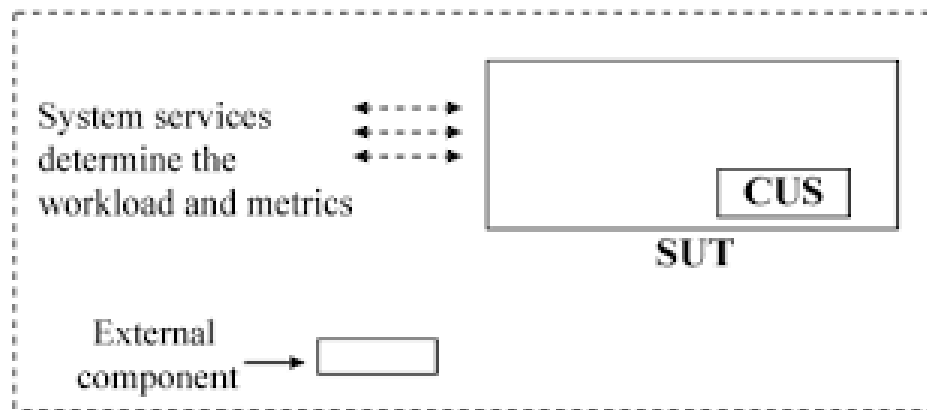


- Listar os Serviços
- SUT = Sistema em teste
- CUS = Componente em estudo.
- Não confundir SUT com CUS

Exemplo:

Uma equipe de projeto pretende verificar o impacto da Unidade Lógica Aritmética (ULA) na Unidade Central de Processamento (CPU).

Neste caso, a CPU é o SUT e a ULA é o CUS.





SERVICES
EXERCISED

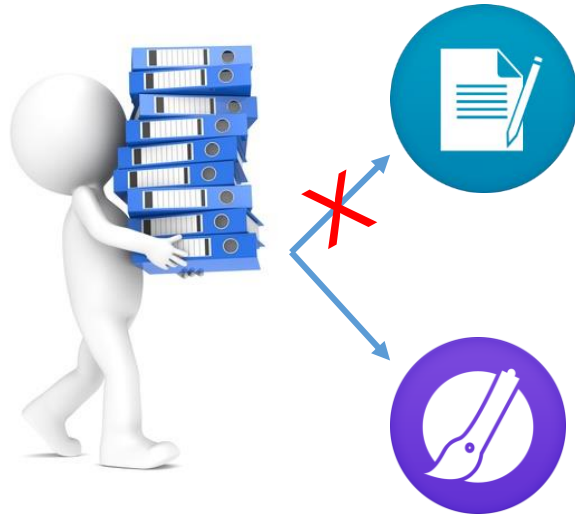
LEVEL OF DETAIL

REPRESENTATIVE-
NESS

TIMELINESS

- Carga de trabalho: depende do sistema;
- Deve-se considerar a finalidade do estudo

Exemplos:



A carga de trabalho de um Editor de Texto não é apropriado se o objetivo do estudo é avaliar um Editor Gráfico.



SERVICES EXERCISED

LEVEL OF DETAIL

REPRESENTATIVE-NESS

TIMELINESS

APLICAÇÃO

Serviços: Aplicações do Usuário

Workload: especificar a frequência de vários tipos de aplicações e suas características

APRESENTAÇÃO

Serviços: Compressão e segurança dos dados

Workload: especificar a frequência de vários tipos de solicitações de segurança e compressão de dados

SESSÃO

Serviços: Analisar o diálogo completo das mensagens

Workload: incluir a frequência e duração de vários tipos de sessões

TRANSPORTE

Serviços: Segmentação e remontagem das mensagens

Workload: especificar frequência, tamanhos e outras características das várias mensagens.

REDE

Serviços: Analisar as características dos pacotes transmitidos

Workload: especificar a matriz de origem-destino e as características dos pacotes transmitidos

ENLACE

Serviços: Analisar as características dos quadros transmitidos e as taxas

Workload: especificar características de pacotes transmitidos, comprimentos e taxas de chegada

FÍSICA

Serviços: Combinar frequências de vários símbolos ou padrões de bits

Workload: frequência de vários símbolos ou padrões de bit observadas em redes reais



SERVICES
EXERCISED

LEVEL OF DETAIL

REPRESENTATIVE-
NESS

TIMELINESS

➤ **Requisição mais frequentes;**

Selecionar o serviço mais solicitado e usá-lo como a carga de trabalho.

➤ **Frequência de Tipos de Requisições;**

Listar vários serviços, suas características e frequência

➤ **Requisições do Tipo Time-stamped;**

Registrar o tempo de pedidos de um sistema real e usá-lo como workload

➤ **Demanda Média de Recursos;**

Agrupar serviços similares

➤ **Distribuição das Demandas de Recursos.**

Usado se a distribuição impactar no desempenho



SERVICES
EXERCISED

LEVEL OF DETAIL

REPRESENTATIVE-
NESS

TIMELINESS

- **Taxa de entrada:** A taxa de entrada dos pedidos deve ser igual ou proporcional ao da aplicação;
- **Demandas de recursos:** As demandas totais de cada um dos recursos fundamentais devem ser proporcionais ou as mesmas da aplicação;
- **Perfil de uso dos recursos:** Refere-se a sequência e a quantidade na qual cada recurso é utilizado na aplicação.



SERVICES
EXERCISED

LEVEL OF DETAIL

REPRESENTATIVE-
NESS

TIMELINESS

- As cargas de trabalho devem seguir as mudanças no padrão de uso, em tempo hábil;
- Os usuários são um objeto em movimento;
- Os usuários tendem a otimizar a demanda;
- Novos sistemas ⇒ novas cargas de trabalho;
- É importante monitorar o comportamento do usuário em uma base contínua.



Seleção da Carga de Trabalho

- ✓ Os **serviços exercidos** determinam a carga de trabalho
- ✓ O **nível de detalhe** da carga de trabalho deve estar de acordo com aquele do modelo em uso
- ✓ A carga de trabalho deve ser suficientemente **representativa** quanto ao uso real do sistema
- ✓ A carga de trabalho deve variar de acordo com o **comportamento do usuário**



Seleção da Carga de Trabalho

“Se você não sabe onde quer ir, qualquer caminho serve”

Lewis Carroll



Teste de Carga em aplicações Web

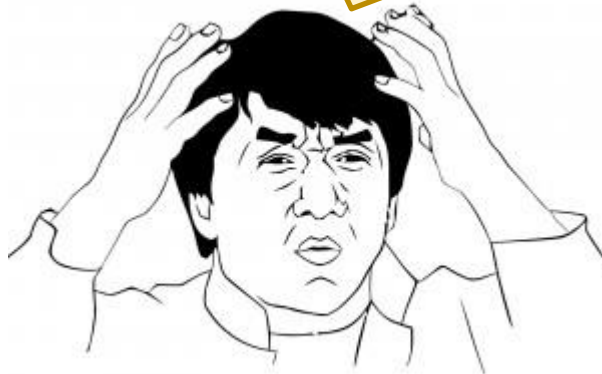
Em pesquisas realizadas pela Amazon, 100 milissegundos a mais no tempo carregamento de uma página, resulta em uma perda de 1% nas vendas. Nesse caso, uma loja virtual que passa a demorar 1 segundo a mais para carregar a página perderia quanto em dinheiro?

100 milissegundos = -1% de vendas.

1 Segundo = 1.000 milissegundos.

1 Segundo = pode significar até – 10% de vendas.

*perdemos 10% de
potenciais vendas a cada
segundo que o site ficou
mais lento!*





Teste de Carga em aplicações Web

americanas.com
a maior loja, os menores preços.

Ops, o site está muuuito cheio.
Nada de F5 ou atualizar esta página! Não se preocupe, você está na fila e, em breve, vai conseguir aproveitar nossas ofertas. :)

aguarde um pouquinho

FAST

GoldenFriday
Procure o selo dourado

Ops, tem muita gente por aqui!

Não precisa dar F5 ou atualizar a página. Você já garantiu seu lugar na fila e, logo, logo, vai poder agarrar nossas ofertas :D

Parabéns @kabumcombr site fora do ar nos primeiros minutos da Black Friday! Preparação 100%!

10:09 PM - 26 Nov 2015

Request Timeout

www.pontofrio.com.br

Request Timeout

The server timed out while waiting for the browser's request.

Reference #2.349262be.1448589024.0

livraria cultura

Nosso site está momentaneamente em manutenção.

Pedimos desculpas pelo transtorno.

Aguarde alguns instantes e tente novamente.

BLACKFRIDAY

OPSI! A casa ficou cheia. Aguarde um pouquinho.

Não feche esta janela. Você será redirecionado em alguns instantes.



Ex: Teste de Carga em aplicações Web

Google PageSpeed Insights

Mede o desempenho de uma página para dispositivos móveis e computadores. Analisa o conteúdo de uma página e gera sugestões de melhorias para otimizar o carregamento

<https://developers.google.com/speed/pagespeed/insights/>





Mobile



Desktop

55 / 100 Velocidade

! Você deve corrigir:

Eliminar JavaScript e CSS de bloqueio de renderização no conteúdo acima da borda

› [Mostrar como corrigir](#)

Aproveitar cache do navegador

› [Mostrar como corrigir](#)

Ativar compactação

› [Mostrar como corrigir](#)

Otimizar imagens

› [Mostrar como corrigir](#)

! Considere corrigir:

Reduzir o tempo de resposta do servidor

› [Mostrar como corrigir](#)

Compactar JavaScript

› [Mostrar como corrigir](#)

Compactar CSS

› [Mostrar como corrigir](#)

Compactar HTML

› [Mostrar como corrigir](#)

✓ Aprovado em 2 regras

› [Exibir detalhes](#)

Faça o download de [recursos de imagem](#), [JavaScript](#) e [CSS](#) otimizados para esta página.

A pontuação varia de 0 a 100 pontos;

85 ou superior indica bom desempenho.





Mobile



Desktop

85 / 100 Experiência do usuário

! Você deve corrigir:

Tamanho apropriado dos pontos de toque

► [Mostrar como corrigir](#)

! Considere corrigir:

Dimensionar o conteúdo para se ajustar à janela de visualização

► [Mostrar como corrigir](#)

✓ Aprovado em 4 regras

► [Exibir detalhes](#)

**Os resultados são armazenados em cache por 30 s. Se tiver feito alterações na sua página, aguarde por 30 s antes de executar o teste novamente.*

A pontuação varia de 0 a 100 pontos;

85 ou superior indica bom desempenho.





Ex: Teste de Carga em aplicações Web

BlazeMeter

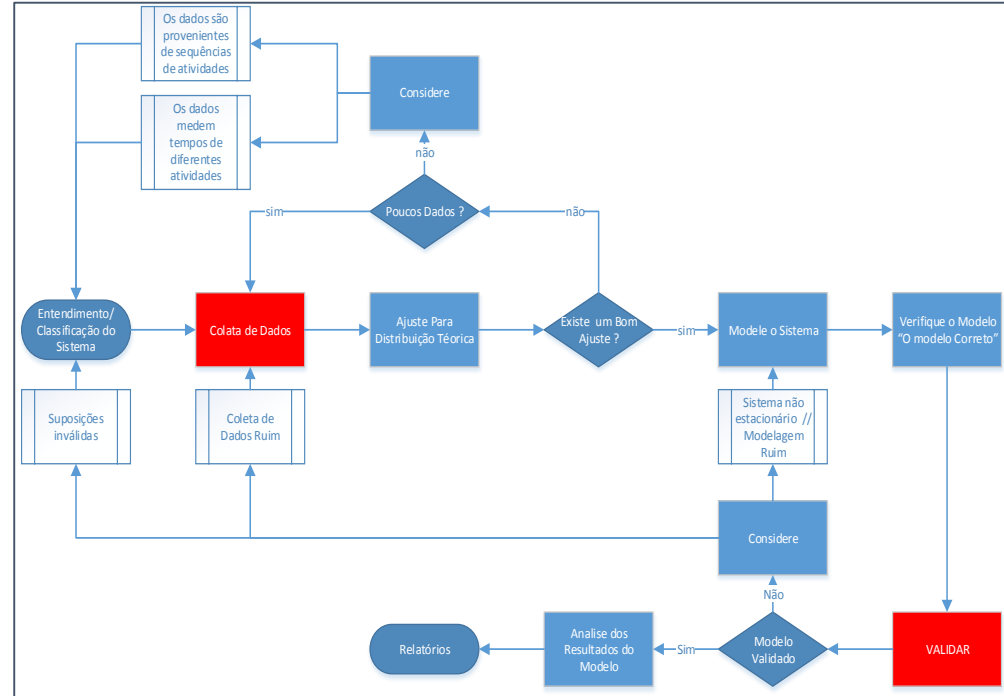
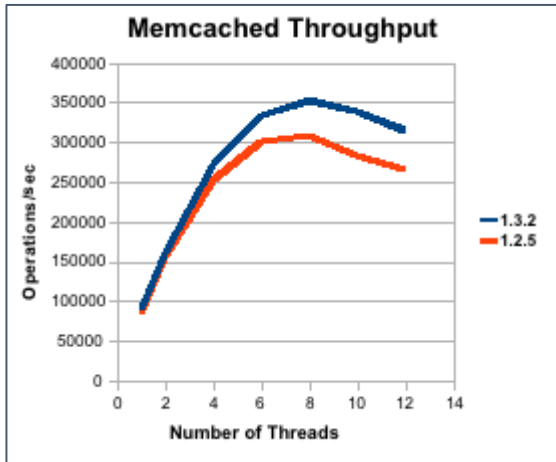
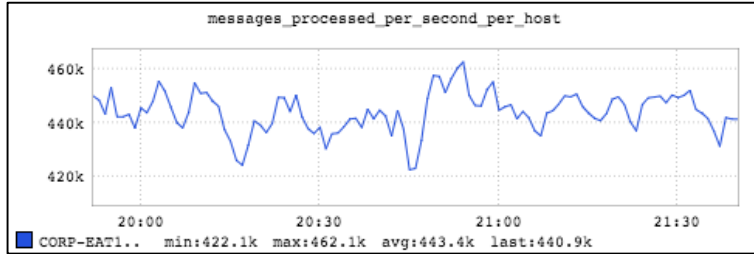
Plataforma como um serviço (PaaS) para realização de teste de carga. Além de testar o desempenho de aplicações Web, é possível criar/enviar um script jmeter para ser executado em um ambiente virtualizado de workload.

<https://blazemeter.com/>





Técnicas de Caracterização de Workload





Técnicas de Caracterização de Workload

- ✓ O **modelo** de carga de trabalho deve ser repetível – conceito fundamental;
- ✓ O efeito deve ser controlado mudando parâmetros no modelo de workload;
- ✓ É interessante separar cada **Workload Component**, usuários diferentes;



Parâmetros de Workload

- ✓ Deve depender da carga, não do sistema,
 - ✓ Tempo de resposta não é um bom parâmetro;
 - ✓ Email->Tamanho da mensagem;
- ✓ Modelagem de Workload
 - ✓ Descritiva;
 - ✓ Gerador;

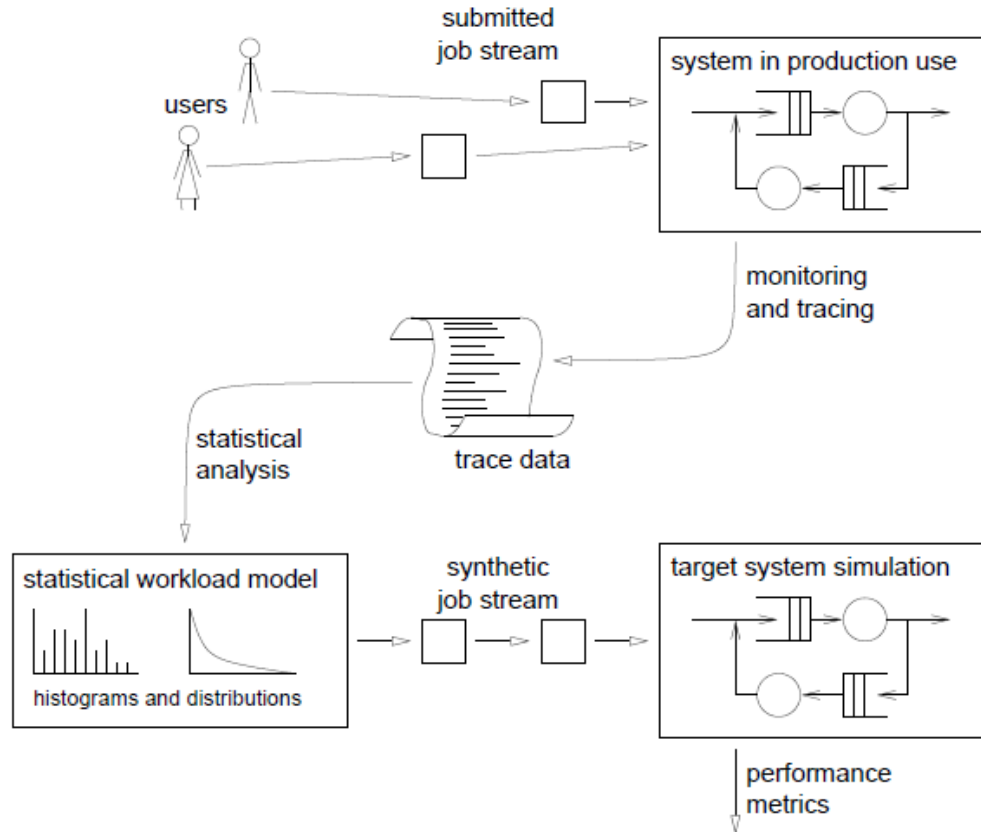


Técnicas de Caracterização de Workload

✓ Modelagem Descritiva;

✓ A carga é gerada de acordo com o modelo;

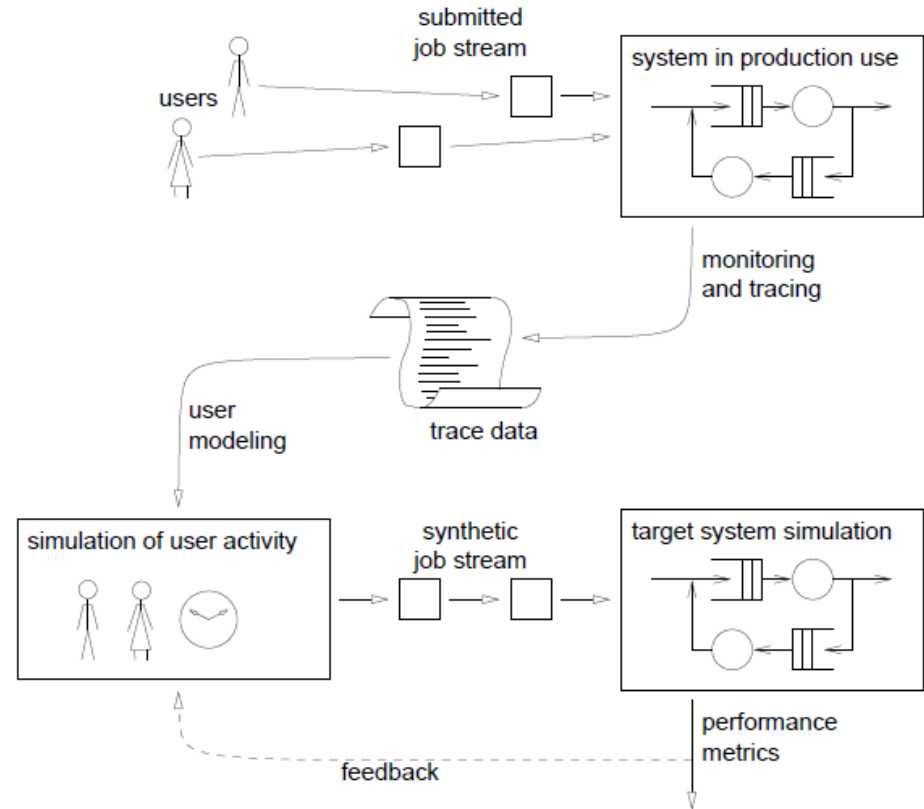
✓ Repete diretamente modelo gerado pelo trace;





Técnicas de Caracterização de Workload

- ✓ **Modelagem geradora;**
- ✓ Não repete diretamente o trace;
- ✓ Descreve como os usuários usam o sistema;
- ✓ Mais fácil de manipular o workload;



Técnicas de Caracterização de Workload

- Quais parâmetros nos traces;
- Deve incluir apenas os parâmetros mais relevantes;
- Algumas vezes desejamos apenas algumas classes;





Técnicas de Caracterização de Workload

- ✓ Média, mediana, moda;
- ✓ Variância, Desvio padrão, Coeficiente de variação = C.O.V;

TABLE 6.1 Workload Characterization Using Average Values

Data	Average	Coefficient of Variation
CPU time (VAX-11/780)	2.19 seconds	40.23
Number of direct writes	8.20	53.59
Direct-write bytes	10.21 kbytes	82.41
Number of direct reads	22.64	25.65
Direct-read bytes	49.70 kbytes	21.01

TABLE 6.2 Characteristics of an Average Editing Session

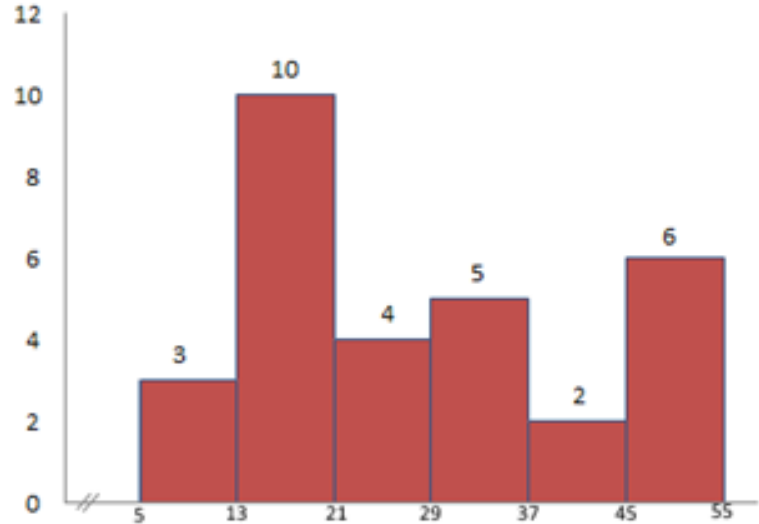
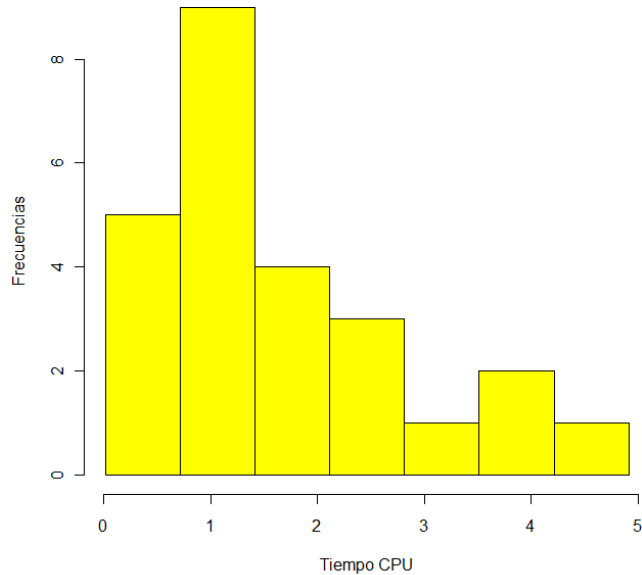
Data	Average	Coefficient of Variation
CPU time (VAX-11/780)	2.57 seconds	3.54
Number of direct writes	19.74	4.33
Direct-write bytes	13.46 kbytes	3.87
Number of direct reads	37.77	3.73
Direct-read bytes	36.93 kbytes	3.16

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$



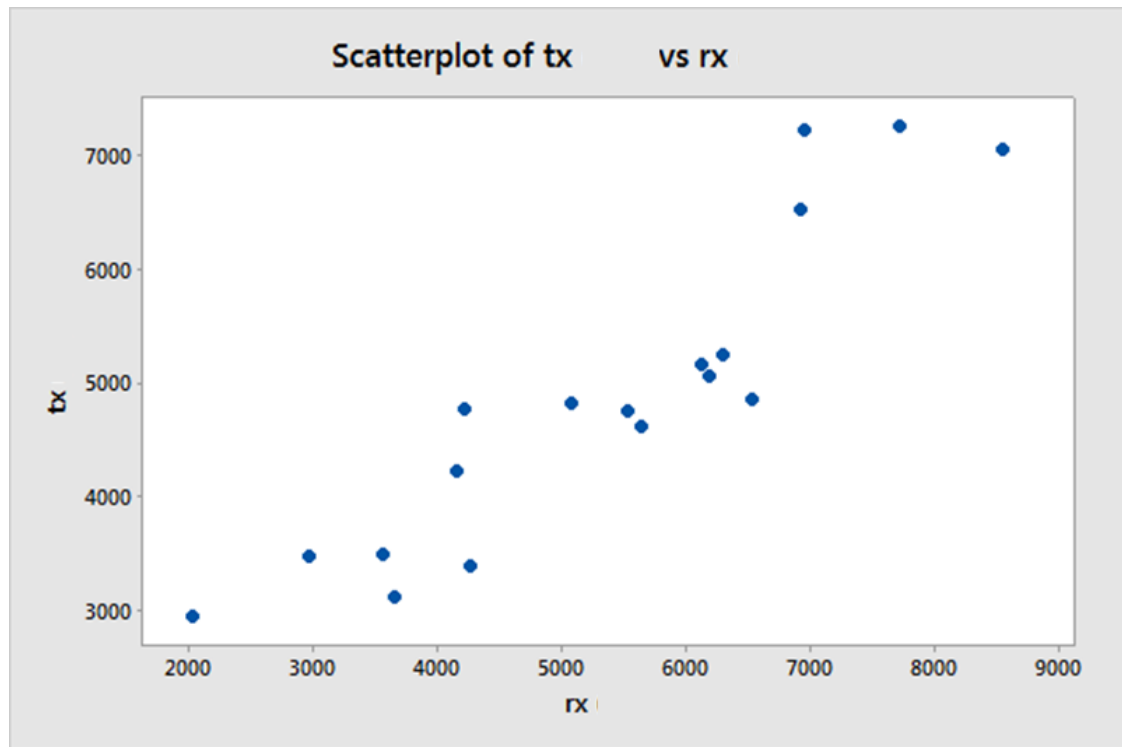
Histograma de um único parâmetro

Histograma de Datos CPU





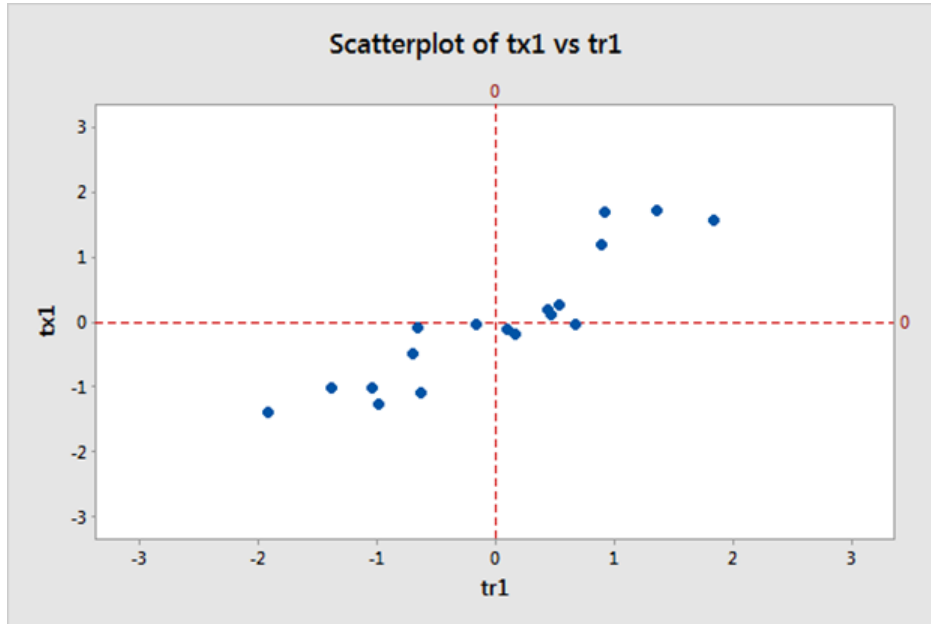
Histograma de múltiplos parâmetros





Analise de componente principal - PCA

- Identificar **Componentes** mais significantes

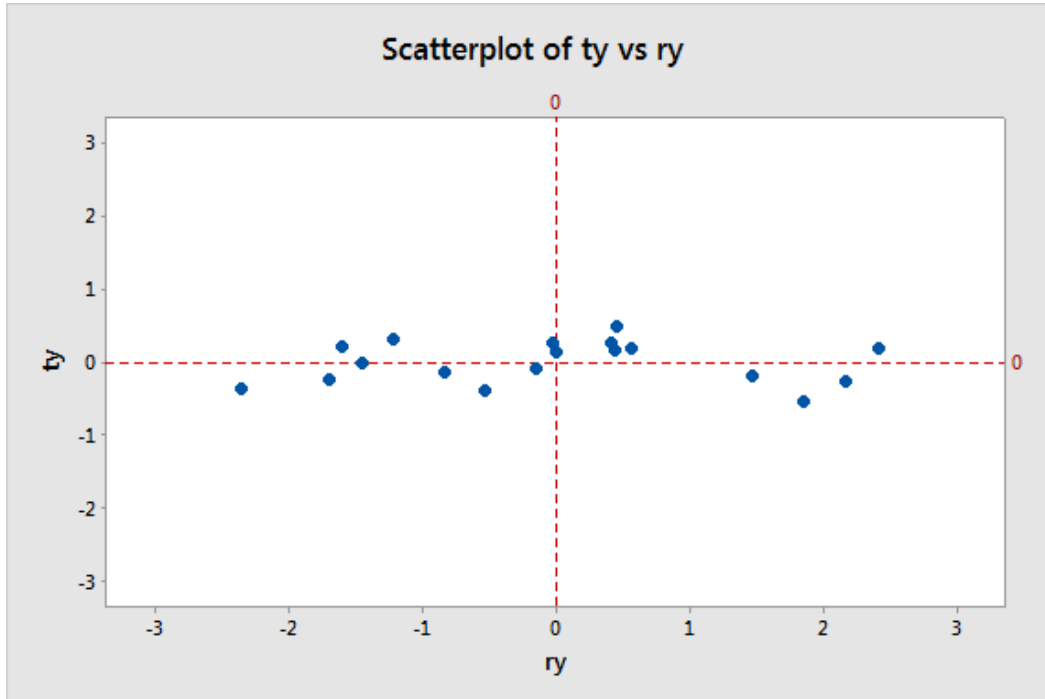


$$x'_s = \frac{x_s - \bar{x}_s}{s_{x_s}}$$

$$x'_r = \frac{x_r - \bar{x}_r}{s_{x_r}}$$



Análise de componente principal - PCA



$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{x_s - 5352}{1741} \\ \frac{x_r - 4889}{1380} \end{bmatrix}$$

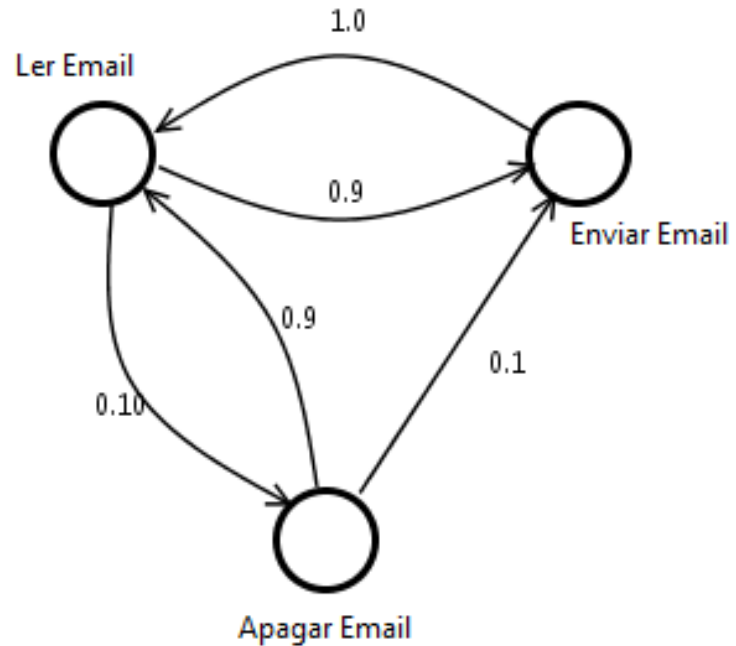
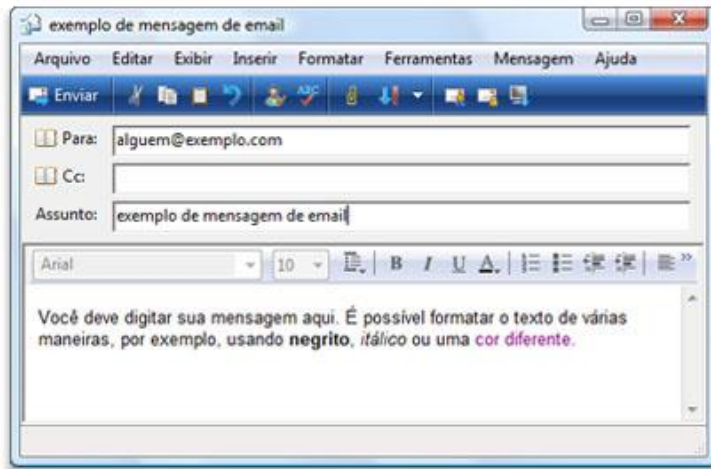


PCA - Minitab

1. Calculo dos componentes normalizados:
 1. $(\text{'rx'} - \text{MEAN}(\text{'rx'})) / \text{STDEV}(\text{'rx'})$
 2. $(\text{'tx'} - \text{MEAN}(\text{'tx'})) / \text{STDEV}(\text{'tx'})$
2. Calcular os autovetores:
 1. Stat -> multivariate -> principal Components;
3. Calcular os Componentes Principais:
 1. $C16(1) * C14 + C16(2) * C15$
 2. $C17(1) * C14 + C17(2) * C15$

Cadeia de Markov

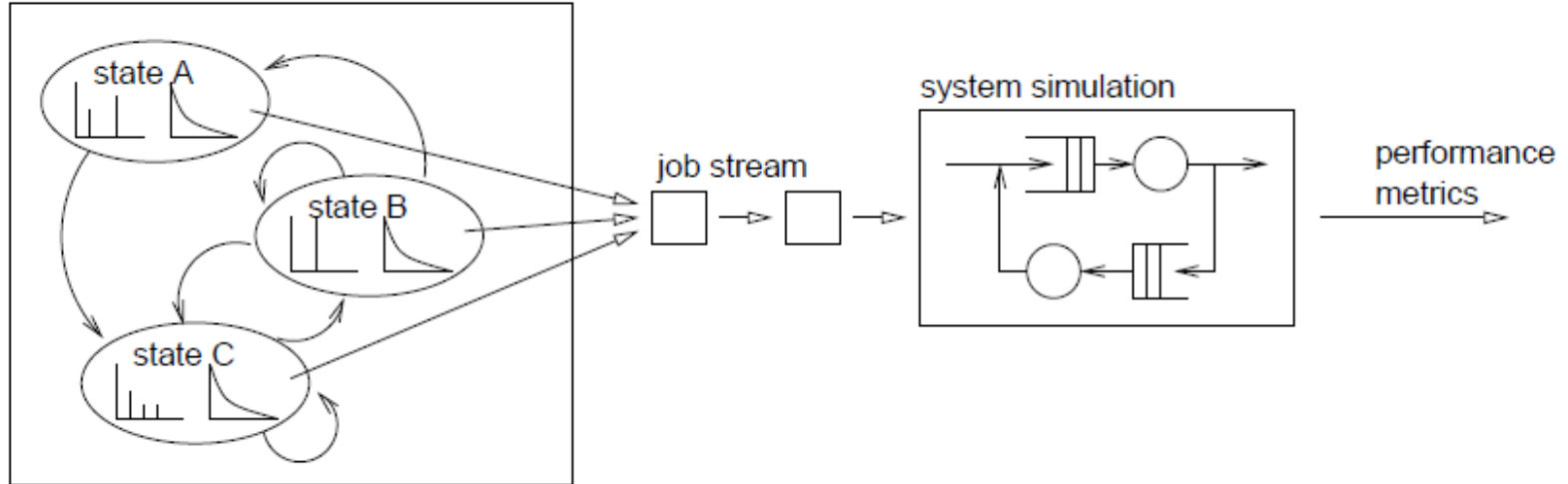
Comportamento de cargas diferentes.





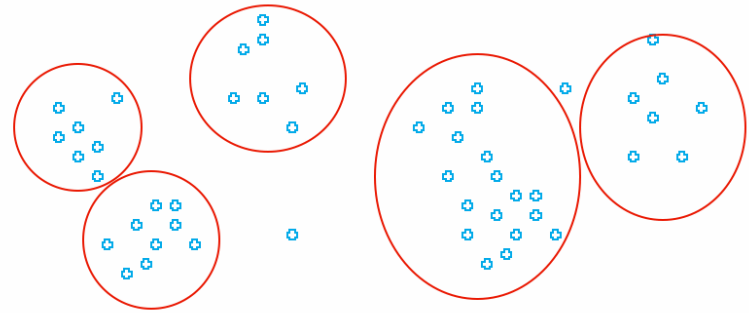
Cadeia de Markov

HMM-based workload model



Clustering

- Separar cargas em grupos;
- Selecionar um subconjunto de cargas;
- Pode-se descartar populações pequenas;
- Pode-se descartar componentes que usam poucos recursos do sistema;





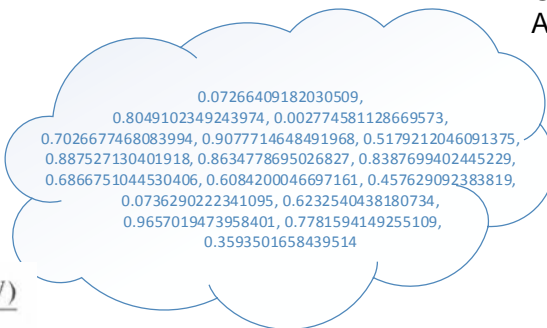
Números Aleatórios

Em qualquer simulação uma das questões principais é gerar valores aleatórios seguindo distribuições específicas;

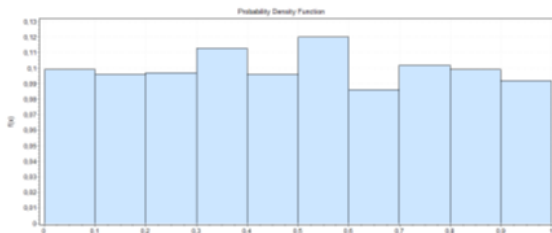
Geração de Número Aleatório Uniforme



Geração de Variável Aleatória



$$X = -\frac{\ln(U)}{\lambda}$$



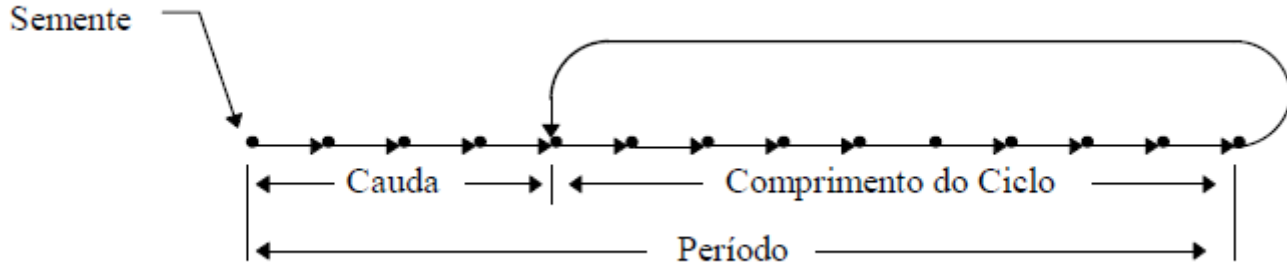


Geração de Números Aleatórios

- Normalmente utiliza uma função recursiva que gera números entre 0 e 1;

$$x_n = (5 * x_{n-1} + 1) \bmod 16$$

- $X_0=5 \Rightarrow 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5$
 $5, 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5.$





Geração de Números Aleatórios

- Devem ser computacionalmente eficientes;
- Período Longo;
- Sucessivos valores devem ser independentes e uniformemente distribuídos;
- Métodos:
 - Método congruente linear;
 - Método congruente linear multiplicativo;
 - Geradores Tausworthe;
 - Geradores de Fibonacci estendidos;
 - Combinação de Geradores.



Geração de variáveis aleatórias

- Método da transformação inversa:
 - Baseado no Inverso da CDF $F(x)$; $u=F(x)$;
 - Sendo u uniformemente distribuído entre 0 e 1;

The pdf $f(x) = \lambda e^{-\lambda x}$.

The CDF $F(x) = 1 - e^{-\lambda x} = u$ or $x = -\frac{1}{\lambda} \ln(1 - u)$.

$$x = -\frac{1}{\lambda} \ln(u)$$



Geração de variáveis aleatórias-Java

```
public double generateExponential(double lambda, double seed){  
  
    Random r= new java.util.Random();  
    r.setSeed(100);  
  
    double expRandom = -Math.Log(r.nextDouble())/lambda;  
  
    return expRandom;  
}
```

```
public double generateExponential(double lambda){  
    //gera número de 0 a 1  
    double uniform = Math.random();  
  
    double expRandom = -Math.Log(Math.random())/lambda;  
  
    return expRandom;  
}
```



Apache JMeter

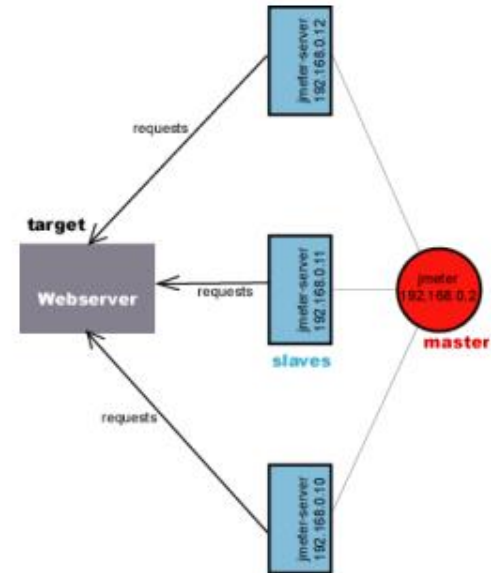


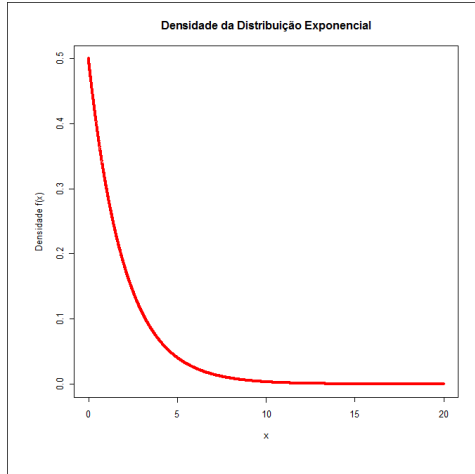
- Uma aplicação para teste de carga e mensuração de performance
- Protocolos testáveis
 - Web - HTTP, HTTPS
 - SOAP / REST
 - FTP
 - Database via JDBC
 - LDAP
 - Message-oriented middleware (MOM) via JMS
 - Mail - SMTP(S), POP3(S) e IMAP(S)
 - MongoDB (NoSQL)
 - shell scripts
 - TCP



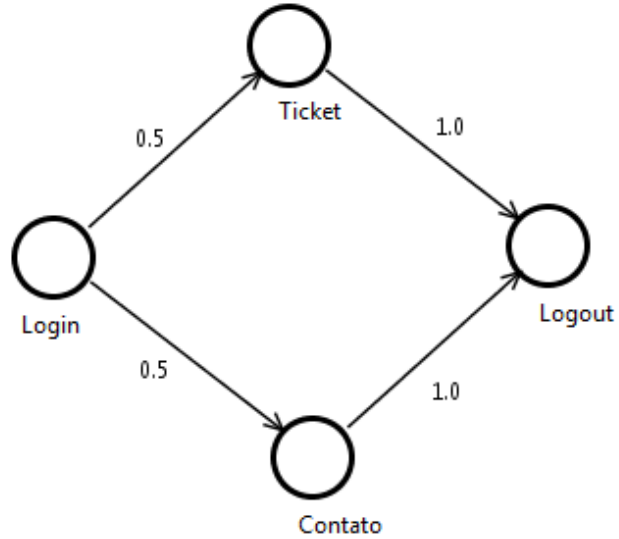
Apache JMeter

- Extensível - linguagens (BeanShell, BSF-compatible, JSR223);
- Dados - podem vir de arquivos
- Compatível com Junit;
- Diversos plugins;
- Relatórios;
- Aceita Testes Distribuídos.



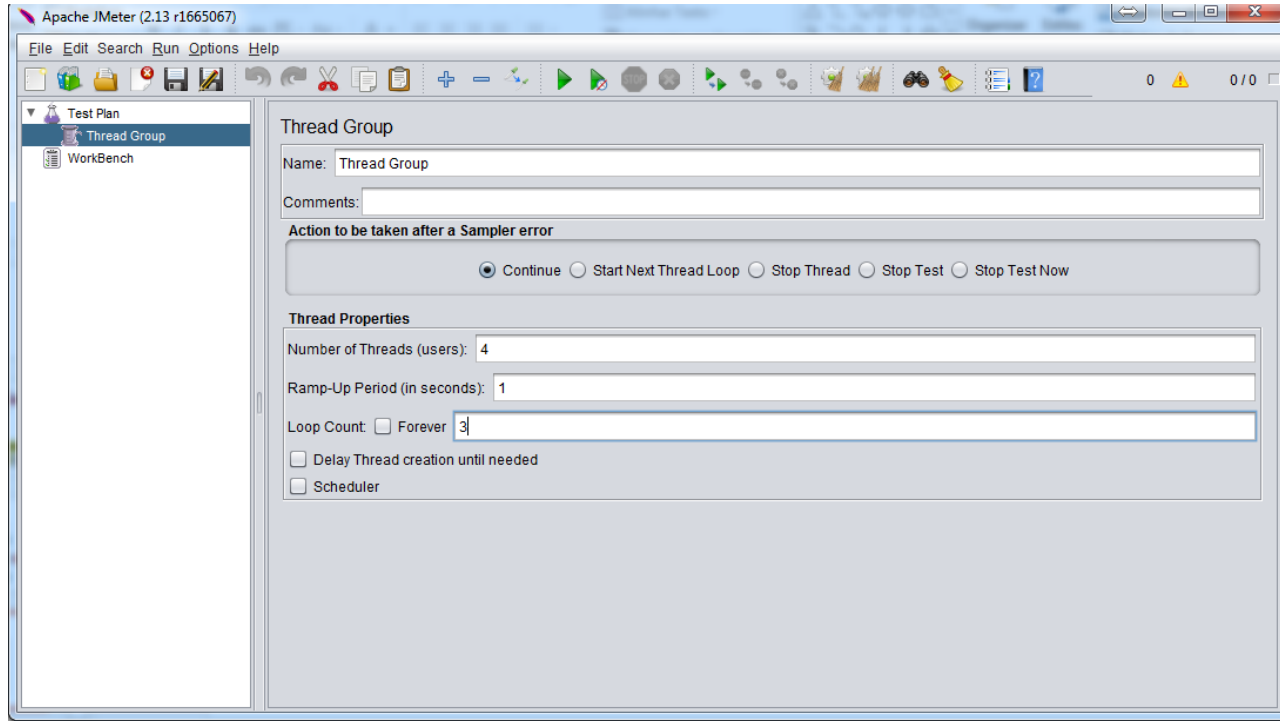


0.0001

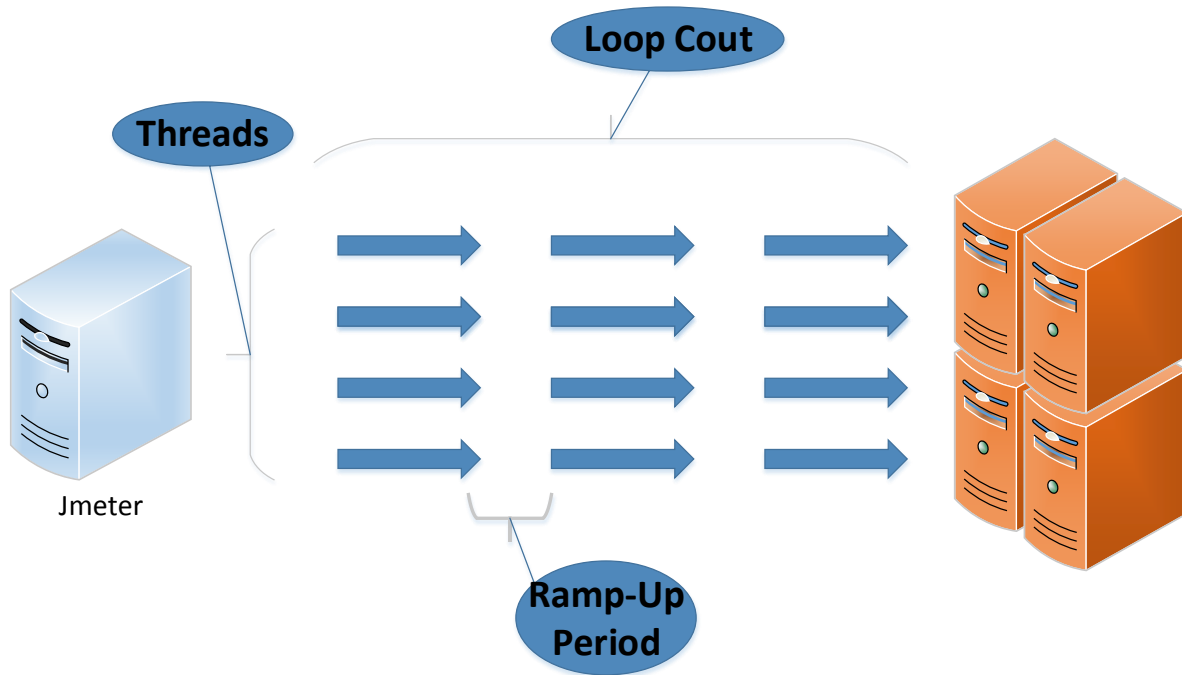


0.01 segundos

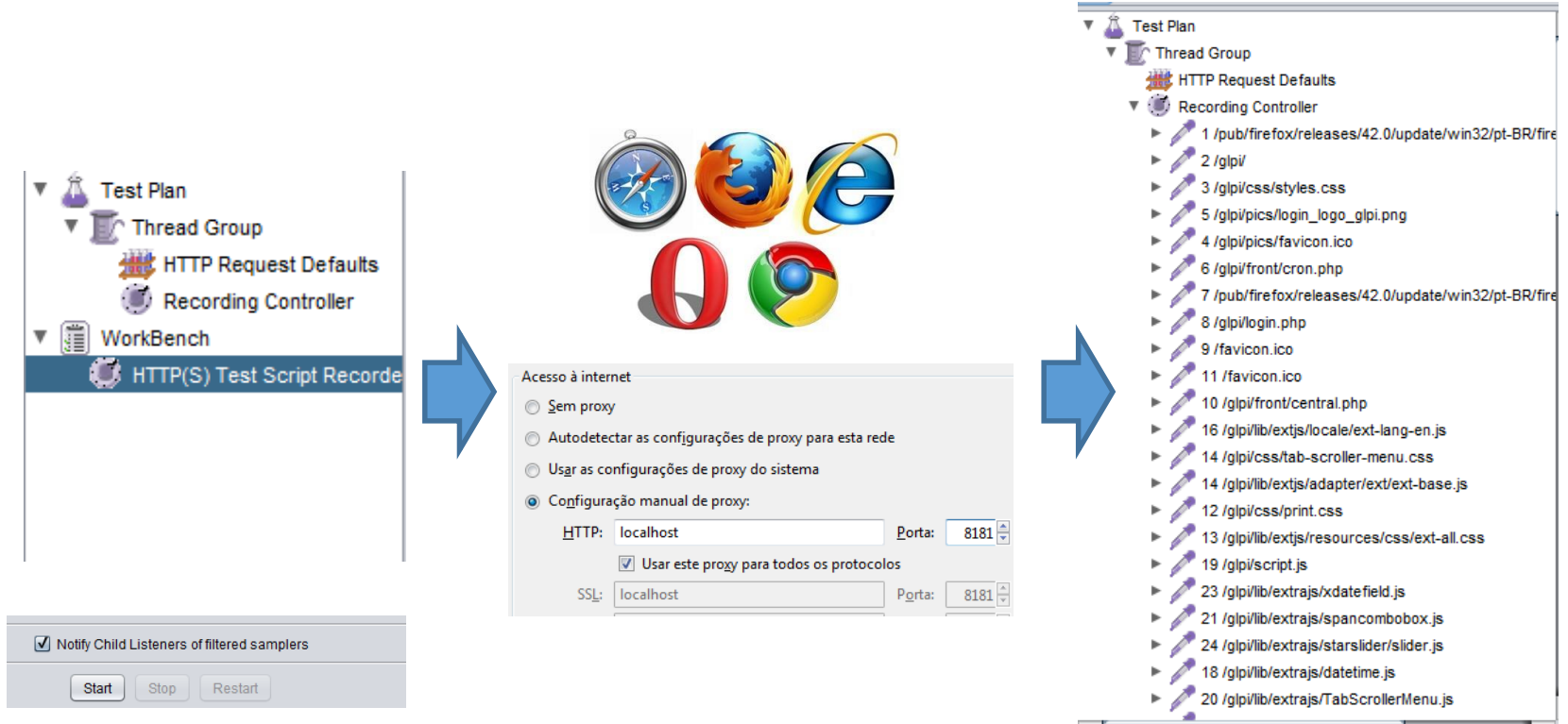
JMeter-Thread Group



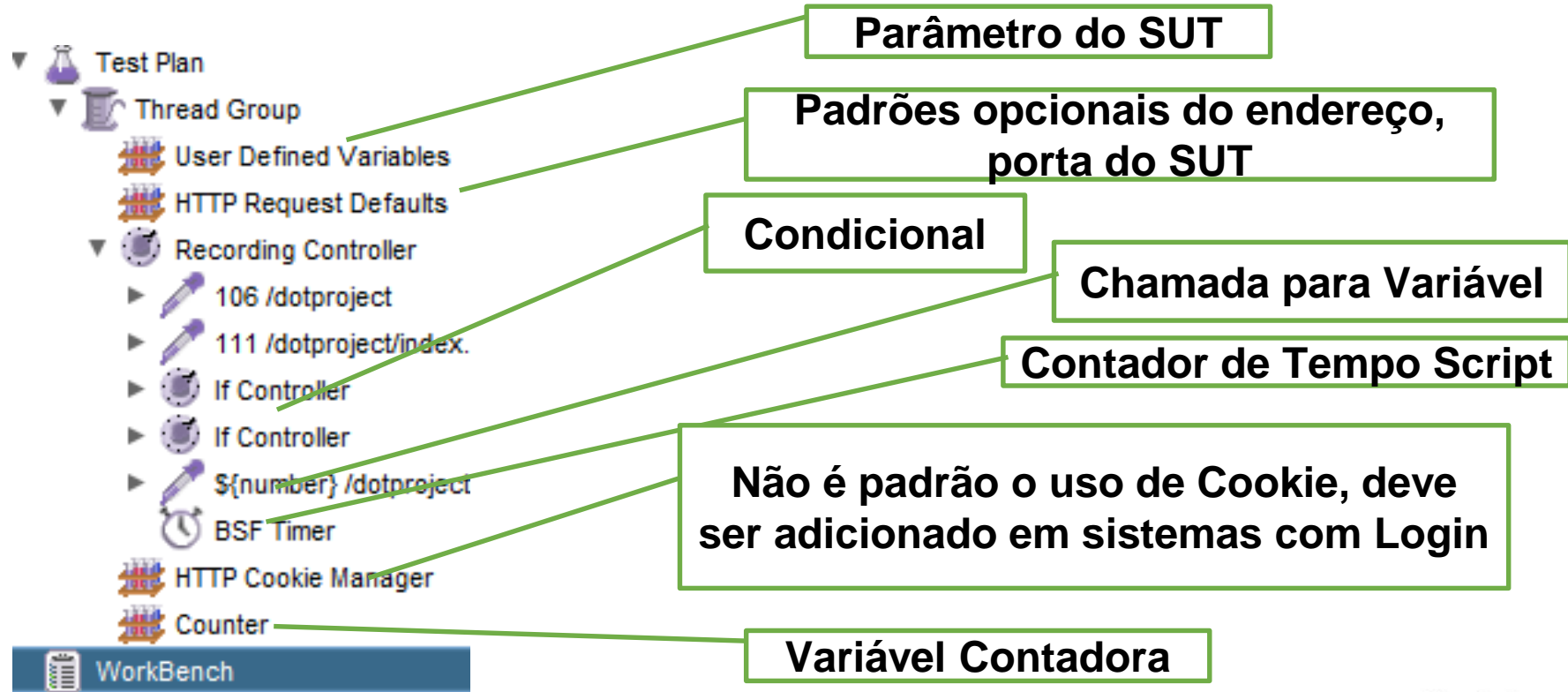
JMeter-Thread Group



JMeter - Record



JMeter - Estrutura



JMeter - Estrutura

- Counter: `number`
- Condicional : `(${number} % 2)==0`

- Time:

```
//gera número de 0 a 1 tempo em milisegundos  
double lambda = 0.001;
```

```
double uniform = Math.random();  
double expRandom = -Math.log(Math.random())/lambda;
```

```
log.info("*****sleep "+(int) expRandom+"*****");
```

```
return ((int) expRandom);
```

- É possível mudar o tamanho máximo de memória, Jmeter.sh ou Jmeter.bat
 - set HEAP=-Xms2048m -Xmx2048m
 - set NEW=-XX:NewSize=1024m -XX:MaxNewSize=1024m
 - JVM_ARGS="-Xms512m -Xmx512m" jmeter.sh
- É possível executar por linha de comando;

- `jmeter -n -t teste.jmx -l log.jtl -j logjmeter.log`

Log erros JMeter

Resultado de table, deve ser aberto dentro do Jmeter

- <http://jmeter-plugins.org/> - UltimateThreadGroup In.ufpe.br

Ultimate Thread Group

Name:

Comments:


Action to be taken after a Sampler error

Continue Stop Thread Stop Test Stop Test Now

Threads Schedule

Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown Time
100	0	10	60	5
100	15	30	10	10
100	60	30	60	45
100	120	30	60	10
100	240	60	60	10

Expected parallel users count



http://ape.ig