

SSC0643 e SSC5872

Avaliação de Desempenho de Sistemas Computacionais

4ª Aula – Técnicas de Avaliação de Desempenho

Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

Material baseado nos slides de

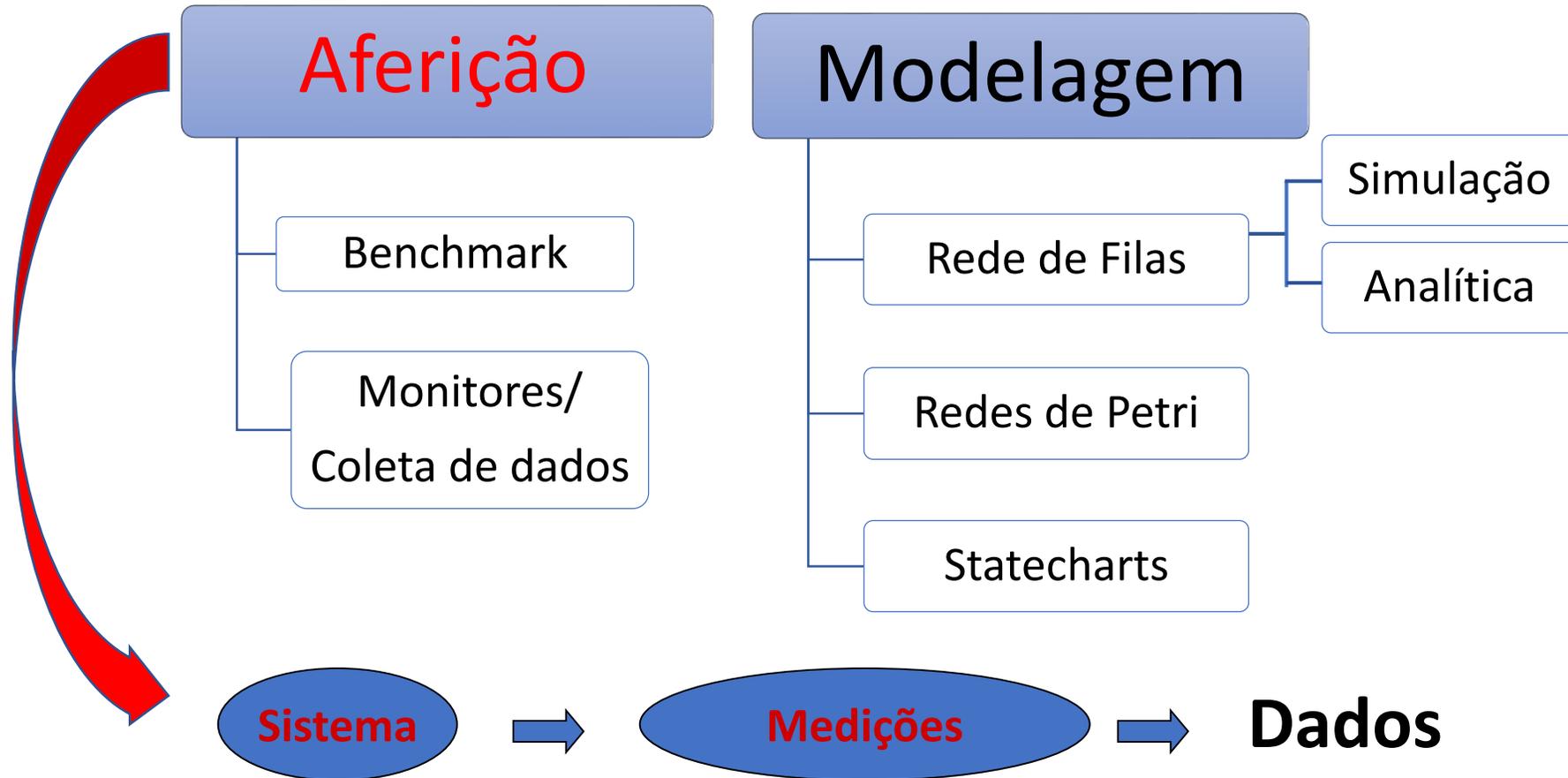
Marcos José Santana

Regina H. C. Santana

Conteúdo

1. Introdução
2. Planejamento de Experimentos
3. Técnicas para Avaliação de Desempenho
 - Apresentação das técnicas
 - Técnicas de Aferição:
 - Protótipos, Benchmarks e Monitores
 - Técnicas de Modelagem:
 - Solução Analítica e por Simulação
4. Análise de resultado

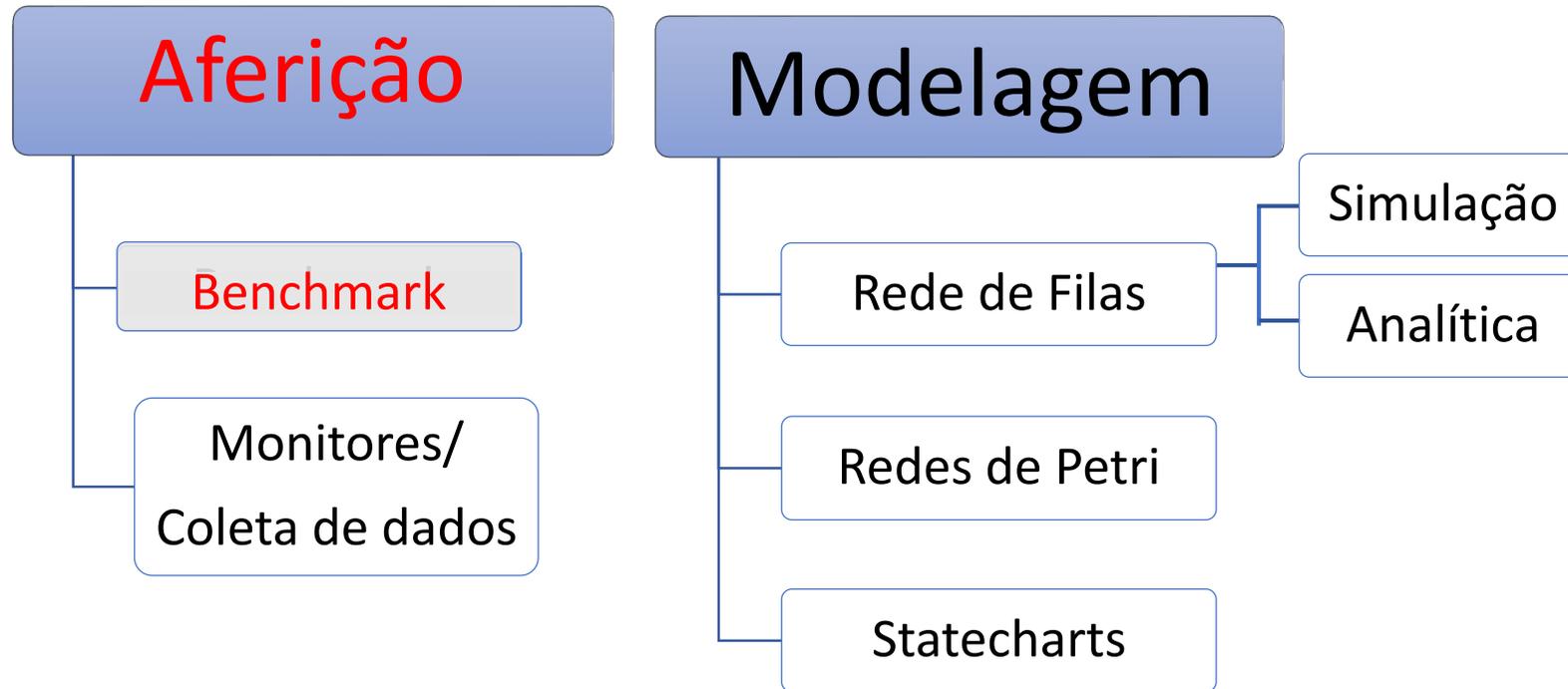
Técnicas de Avaliação de Desempenho



Técnicas de Avaliação de Desempenho

- Aferição
 - Medidas no próprio sistema
 - Sistema deve existir e estar disponível
 - Experimentação restrita
 - Muito cuidado com aquisição dos dados

Técnicas de Avaliação de Desempenho



Benchmarks

- Instrumento fixo, que permite comparar uma medida (mark - marca) a um padrão preestabelecido
- Deve-se ter um ponto de observação (bench - banco)
- Ponto fixo ou referência para comparações



Técnicas de Aferição

- Benchmarks
 - Programa escrito em linguagem de alto nível, representativo de uma classe de aplicações, utilizado para medir o desempenho de um dado sistema ou para comparar diferentes sistemas

Benchmarks

- Abordagem muito utilizada para a avaliação de desempenho por aferição

- Exemplo

- Qual a diferença entre um i5 e um i7?

- Qual a influência no desempenho?

i5	i7
2 ou 4 núcleos	4 ou 6 núcleos
Não possui Hyper-threading	Possui Hyper-threading (2 núcleos lógicos para cada físico)
DMI – Direct Media Interface (taxa de transferência ~2Gb/s)	QPI – Quick Path Interconnect (taxa de transferência > 4,8 Gb/s)
Quantidade de canais para acesso a memória: 2 (acessa 2 pentes ao mesmo tempo)	Quantidade de canais para acesso a memória: 3 (acessa 3 pentes ao mesmo tempo)

Benchmarks

- <http://www.cpubenchmark.net/>
- PassMark Performance Test

Processador	Benchmark (2015)	Benchmark (2017)	Preço (\$) (2015)	Preço (\$) (2017)
Intel Core i7-4960X @ 3.60GHz	13939	13863	1029,99	1220,76
Intel Core i7-4930K @ 3.40GHz	13079	13058	599,99	529,90
Intel Core i7-3970X @ 3.50GHz	12715	12651	1110,63	369,99
Intel Core i5-4690K @ 3.50GHz	7733	7765	239,99	259,01
Intel Core i5-4670K @ 3.40GHz	7647	7411	274,99	NA
Intel Core i7-3610QM @ 2.30GHz	7476	7465	399,99	399,99

Benchmarks

- Uso:
 - Comparar desempenho de máquinas diferentes
 - Re projetar hardware e software
 - Decidir sobre aquisição de sistemas
 - Ajudar na otimização de programas
 - Previsão de desempenho de aplicações em computadores específicos

Benchmarks

- Permitem avaliar o sistema como um todo, incluindo aspectos relacionados com:
 - Arquitetura do sistema
 - Eficiência do compilador
 - Sobrecarga do sistema operacional
 - Sobrecarga de comunicação
 - Identificam pontos fortes e fracos de um sistema em particular ou de aplicações

Benchmarks

- Como escolher um benchmark?
 - Ideal: aplicação do usuário
 - O ideal pode ser inviável quando os sistemas são de propósito geral
 - Necessita-se de algo mais amplo e representativo

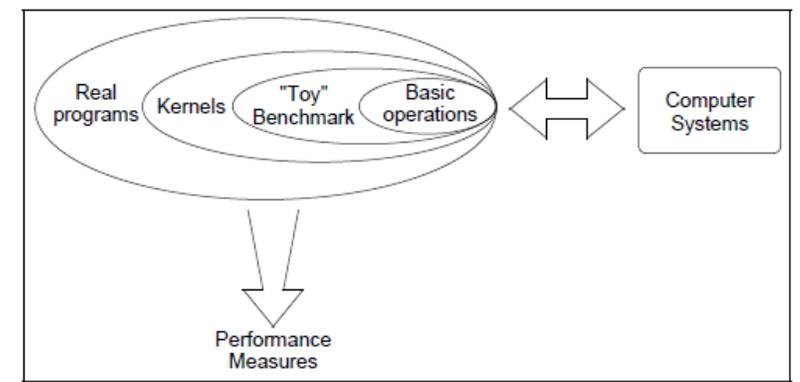
Benchmarks

- Programa escrito em linguagem de alto nível
- Representativo de alguma categoria de programação
- Que possa ser avaliado facilmente
- Que possua larga distribuição

Benchmarks

- Divididos em:
 - Benchmark de componente: analisa um componente específico;
 - Benchmark de sistema: analisa o sistema como um todo;
 - Benchmark de aplicação: utilizam programas representativos
 - Benchmark sintético: imitam ambientes de execução de programas reais. Geram dados estatísticos

Benchmarks



Capacity Planning for Web Services: Metrics, Models, and Methods
Daniel A. Menasce & Virgílio A. F. Almeida

- Hierarquia:

- *Basic Operations*: benchmarks sintéticos que realizam operações básicas, como adição e multiplicação. Exemplo: Dhrystone
- *“Toy” Benchmarks*: pequenos programas que implementam problemas clássicos tais como o Crivo de Eratosthenes e as Torres de Hanoi
- *Kernels*: trechos de códigos extraídos de programas reais, os quais representam os trechos com a maior quantidade de computação. Normalmente medem o desempenho do processador. Exemplos: Livermore Loops e Linpack
- *Real Programs*: execução de programas reais, e normalmente estão dentro de conjuntos de benchmarks. Exemplos: SPEC e TPC

Benchmarks

- Classificação quanto a arquitetura:
 - Sequenciais;
 - Paralelas;
 - Comitê Parkbenck (*Parallel Kernels and Benchmarks*)

Áreas de Aplicação e Objetivos

- Três grandes áreas:
 - Predição
 - não aplicável
 - Monitoração e Análise
 - aplicável, porém insuficiente
 - Comparação
 - técnica ideal

Aspectos Relacionados aos Benchmarks

- Quando os resultados dos Benchmarks forem interpretados, deve-se conhecer:
 - arquitetura do processador;
 - processadores;
 - clock;
 - barramento;
 - memória e caches;
 - disco rígido;
 - sistema operacional;
 - compilador
 - etc.

Aspectos Relacionados aos Benchmarks

- Problemas...
 - Sistemas com configurações diferentes geram medidas de desempenho diferentes
 - Otimização do compilador: influencia diretamente no desempenho medido

Exemplos de Benchmarks

- Benchmarks mais comuns
 - Whetstone, Linpack, Dhrystone
- Outros programas de Benchmarks
 - Stanford Small Programs Benchmark Set
 - EDN Benchmarks
 - Sieve of Eratosthenes
 - Livermore Fortran Kernels
 - Perfect Club Benchmarks
 - SPEC Benchmarks
 - EuroBen Benchmarks
 - TPC Benchmarks

Exemplos de Benchmarks

- Whetstone
 - Benchmark sintético
 - Primeiro grande programa da literatura escrito para Benchmarking
 - Elaborado para análise de programação numérica de ponto flutuante intensivo
 - Resultado: número de loops por segundo

Exemplos de Benchmarks

- Características do Whetstone
 - Possui alto percentual de dados e operações de ponto flutuante
 - Alto percentual de tempo de execução é gasto em funções matemáticas
 - Ao invés de variáveis locais, Whetstone utiliza muitos dados globais
- Ranking das melhores máquinas
 - Mwips: Million Whetstones Instructions Per Second
 - Whetstone Benchmark Detailed Results On PCs
 - <http://www.roylongbottom.org.uk/whetstone.ht>

Exemplos de Benchmarks

- Livermore Fortran Kernels
 - Este benchmark foi criado em 1970 por F.H. McMahon
 - Consiste de 24 “kernels” ou loops de processamento numérico
 - Analisa desempenho de computadores e compiladores em operações com ponto flutuante

Exemplos de Benchmarks

- SPEC Benchmarks
 - SPEC (*System Performance Evaluation Cooperative ou Standard Performance Evaluation Corporation*) fundada em Outubro de 1988 por Apollo, Hewlett-Packard, MIPS e Sun Microsystems.
 - Possui como meta testar situações reais, fornecendo medidas de sistemas computacionais avançados em relação a aspectos de desempenho.

Exemplos de Benchmarks

- SPEC Benchmarks
 - CPU
 - Aplicações Gráficas
 - Aplicações de Alto Desempenho
 - MPI e OpenMP
 - Cliente/Servidor Java
 - Projetado para medir o desempenho de servidores rodando aplicações Java
 - Mail Servers
 - Mede desempenho de servidores de e-mail
 - Network File Systems
 - Web Servers
 - Simula usuários enviando requisições

Exemplos de Benchmarks

- Benchmarks para Arquiteturas Paralelas e Sistemas Distribuídos:
 - problemas com MIPS e MFLOPs: não basta somá-los:
 - comunicação;
 - balanceamento;
 - etc.
 - para cada tipo de arquitetura deve haver um benchmark específico;
 - Exemplos:
 - PING e PING-PONG:
 - utilizados na avaliação de comunicação;
 - TTCP:
 - avaliação de comunicação usando TCP ou UDP.

Exemplos de Benchmarks

- Para utilização específica:
 - Servidores Web
 - Redes
 - HD
 - Servidores de e-mail
 - Virtualização
 - SOA
 - Servidores de arquivos
 - Processamento gráfico
 - Etc

Exemplos de Bechmarks

WebStone

- Desenvolvido pela Silicon Graphics
- Objetivo: medir desempenho de software e hardware de servidores Web
- Versão 2.0.1
- Testa o servidor Web, incluindo sistema operacional, CPU e velocidade de acesso a rede.
- Não testa influência da velocidade de acesso ao disco
- Principais medidas
 - Throughput máximo
 - Tempo médio de resposta

Exemplos de Bechmarks

Surge

- Simula uma coleção de usuários acessando um servidor Web
- Arquitetura
 - Conjunto de clientes conectados a um servidor Web
 - Cliente
 - Conjunto de threads, especificado pelo usuário
 - Requisições a documentos no servidor
 - Estados ON e OFF
 - ON -> transferência de arquivos
 - OFF -> Ativo e Inativo

Exemplos de Bechmarks

httperf

- Ferramenta para medir o desempenho de servidores Web
- Gera várias cargas HTTP
- Arquitetura
 - Maior independência do SO
 - Única thread
 - Mecanismo próprio de Timeout
 - Três módulos
 - HTTP
 - Gerenciamento de conexões
 - Geração de requisições http
 - Geração de carga
 - Regras para a geração de requisições
 - Coleta de estatísticas

Exemplos de Bechmarks

httperf

- Carga
 - Conexões persistentes e requisições em cascata
 - Geradores de carga
 - Geração de requisições
 - Requisições a uma taxa fixa. Padrão: 1 req/con
 - Sessões a uma taxa fixa. 1 sessão: várias rajadas de requisições. Think time: 0.
 - Geração de URLs
 - Sequência de URLs que devem ser acessadas no servidor.

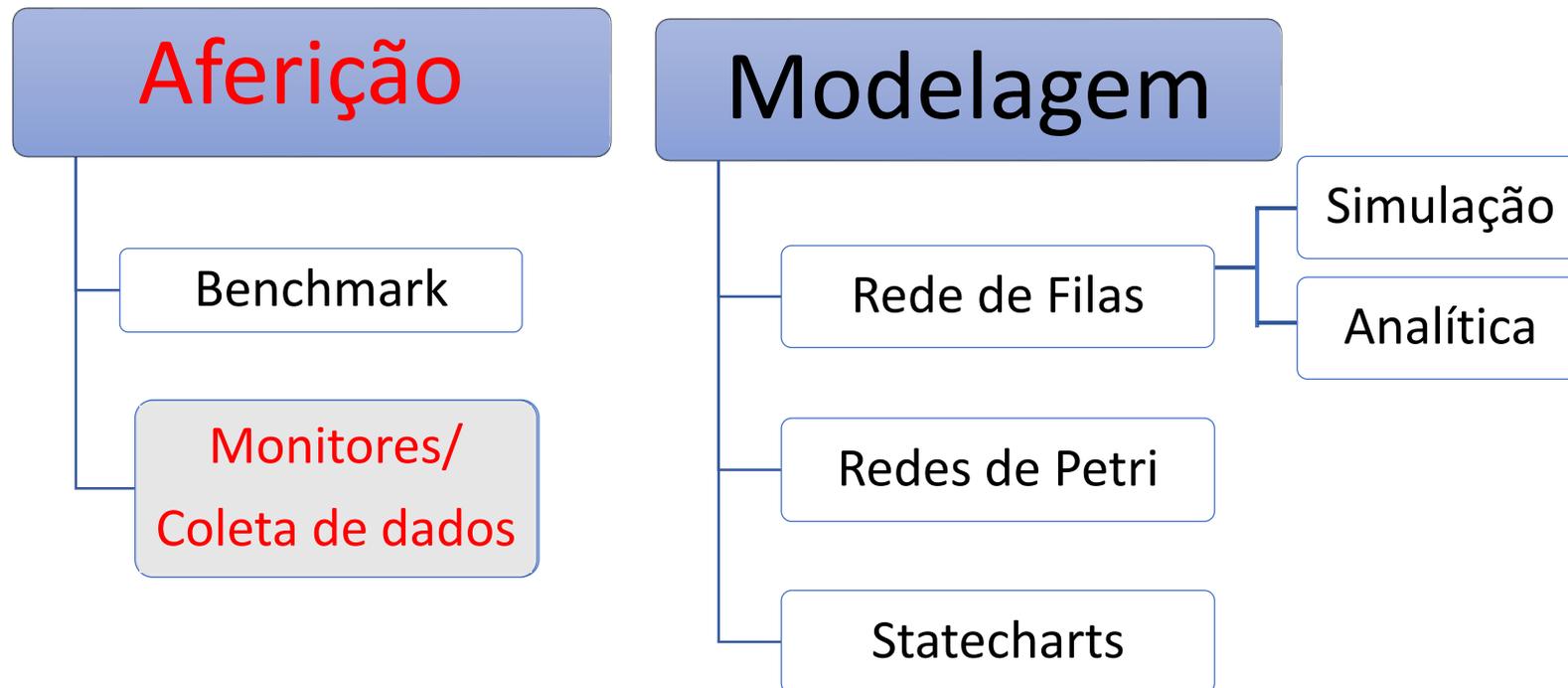
Concluindo....

- Benchmarks podem ser utilizados para verificar diversos tipos de sistemas ...
 - Servidores Web,
 - Banco de dados,
 - Processadores,
 - Redes de comunicação
- ... sendo utilizados com diferentes objetivos...
 - Codificação de vídeo e edição de imagens,
 - Jogos,
 - Processamento intensivo,
 - Processamento de textos, etc.
- ... querendo avaliar diferentes características...
 - Produtividade
 - Desempenho
 - Confiabilidade, etc.

O importante é...

- Escolher o Benchmark adequado
- Aplicar o Benchmark de forma adequada
- Analisar os resultados obtidos com critério

Técnicas de Avaliação de Desempenho



Técnicas de Aferição

- Coleta de dados
 - Ferramenta para observar as atividades de um sistema coletando as características relevantes para a análise do sistema
 - Ferramenta = Monitor

Monitores

- Avaliar o desempenho e identificar pontos críticos
- Objetivos:
 - Determinar partes mais utilizadas
 - Determinar gargalos
 - Ajustar Parâmetros
 - Caracterizar Carga de Trabalho
 - Determinar Parâmetros para modelos

Monitores

- Técnica que oferece os melhores resultados;
- Problema principal: interfere com o sistema e o sistema TEM de existir!
- Dois tipos básicos de abordagem:
 - Monitores de Software e de Hardware.

Componentes de um Monitor

Observador
Coletor
Analizador
Apresentação
Interpretação
Console
Gerenciamento

Componentes de um Monitor

1. Observador

- Obtém os dados do sistema
- Espião – observa as atividades do sistema
 - Ex. Escuta em uma rede
- Instrumentação - incorpora pontos de prova, traces ou contadores
 - Sobrecarga no sistema
 - Ex. Determinar tempo para enviar mensagem
- Teste – inclui usuário para monitorar sistema
 - Ex. Envia mensagens para monitorar filas
- Sistema pode ter um ou mais observadores

Componentes de um Monitor

2. Coletor

- Centraliza e filtra os dados coletados
- Armazena dados obtidos e histórico dos dados
- Sistema pode ter um ou mais coletores: problemas de sincronismo e comunicação de dados

3. Analisador

- Analisa os dados coletados
- Considera análise estatística

Componentes de um Monitor

4. Apresentação dos dados

- Produz Gráficos, Relatórios, Alarmes, etc.
- Oferece interface Homem-Máquina

5. Interpretação

- Analisa o significado dos dados
- Análise inteligente

Componentes de um Monitor

6. Console

- Controla os parâmetros e os estados do monitor ou do sistema
- Nem sempre faz parte do Monitor

7. Gerenciamento

- Toma decisão sobre o que deve ser alterado no sistema ou no monitor

Monitores

- Monitores podem ser classificados seguindo diferentes enfoques:
 1. Forma de Ativação
 - Mecanismo de Disparo
 2. Forma de Obtenção dos dados
 - Mecanismo de Coleta dos Dados
 - Mecanismo Visualização dos Dados
 - Definição da Carga de Trabalho
 3. Forma de Implementação
 - Hardware, Software, Firmware e Híbrido

Monitores

Forma de Ativação

- **Mecanismo de Disparo**

- Define quando o monitor será ativado para coletar e gravar as informações

1. Baseado em Evento

- Mudança de estado no sistema ativa o Monitor

- Exemplos: Chaveamento de processos, Acesso a disco, Acesso a rede

2. Baseado em Amostragem

- O Monitor é ativado em intervalos de Tempo Fixos

- Exemplo: a cada 1 seg o monitor verifica se a rede está ocupada

Monitores

Forma de Ativação

- **Mecanismo de disparo**

1. Baseado em Evento

- Assíncrono
- Adequado para eventos raros – causa menor sobrecarga

2. Baseado em Amostragem

- Síncrono
- Adequado para eventos frequentes
- Pode perder eventos
- Período de amostragem deve ser adequadamente definido
 - Longo – perda de eventos
 - Curto – aumento de sobrecarga

Monitores

Forma de Obtenção de Dados

- **Mecanismo de Coleta dos Dados**

- Define como os dados serão coletados

1. Coleta Direta

- Métrica é retirada diretamente do sistema
 - Exemplos: Tempo de resposta de um disco

2. Coleta Indireta

- Métrica é obtida através de relações de dependência com outras métricas retiradas do sistema
 - Exemplo: Cálculo da Utilização de um Processador

Monitores

Forma de Obtenção de Dados

- **Mecanismo Visualização dos Dados**

1. On-line

- Dados analisados e apresentados ao avaliador ao mesmo tempo em que são coletados
 - Exemplos: Avaliação do tráfego de uma rede

2. Batch

- Dados são coletados, armazenados e depois de terminada a coleta são analisados
 - Exemplo: Tempo médio de resposta ou utilização de um processador

Monitores

Forma de Obtenção de Dados

- **Mecanismo Visualização dos Dados**

1. On-line

- Número pequeno de eventos
- Avaliação comportamental do sistema

2. Batch

- Grande número de eventos
- Avaliação de Desempenho, Confiabilidade ou da Segurança de um Sistema

Monitores

Forma de Obtenção de Dados

- **Definição da Carga de Trabalho**

1. Passiva

- Coleta realizada com a operação normal do Sistema

2. Ativa

- Coleta realizada com cargas controladas
- Carga é gerada e injetada no sistema

Monitores

Forma de Obtenção de Dados

- **Definição da Carga de Trabalho**

1. Passiva

- Resultados não podem ser reproduzidos

2. Ativa

- Carga Controlada

- Mais limitada que a passiva
- Resultados não pode ser totalmente reproduzidos
- Testes podem ser repetidos de forma similar

- Carga Sintética

- Não real
- Resultados podem facilmente ser reproduzidos

Monitores

Forma de Implementação

- Define o nível em que o monitor será implementado
 1. Hardware
 2. Software
 3. Firmware
 4. Híbrido

Monitores

Forma de Implementação

- **Hardware**

- Monitor de hardware que é conectado com o sistema (observador silencioso)
- Não interfere no funcionamento normal do sistema medido
- Captura eventos rápidos
- Apresenta dificuldades em fazer medidas em nível de software
- Técnica cara

Monitores

Forma de Implementação

- **Software**

- Utilização de programas que fazem medições
- Vantagens:
 - Generalidade
 - Flexibilidade
 - Para medidas em nível de programas
- Desvantagens:
 - Ele pode interferir com o funcionamento normal do sistema
 - Não captura eventos que ocorrem rapidamente

Monitores

Forma de Implementação

- Exemplos
 - Software
 - Rotina inserida nos protocolos de comunicação para medir o tempo gasto em uma transação em arquivos
 - Hardware
 - Hardware adicionado ao sistema para espionar e contabilizar o tempo gasto em uma transação em arquivos

Monitores

Forma de Implementação

- **Firmware**

- Implementado através da alteração de micro-código do processador
- Implementação em hardware, de rotinas de software através de MEF (Máquinas de Estado Finito)
- Similar ao monitor de Software

Monitores

Forma de Implementação

- **Híbrido**

- Vantagens:

- Esboça as vantagens de ambas técnicas de monitoramento (hardware e software)
 - Flexível

- Desvantagens:

- Sincronização pode causar interferência
 - Difícil obtenção de programas ou medidas em nível de S.O.
 - Caro

Comparação entre monitores de Software e Hardware

Critério	Hardware	Software
Domínio	Eventos de Hardware	Eventos de SO e Software
Taxa de Entrada	Alta (10^5 / Seg)	Depende do proc.
Resolução	Nanosegundos	Milisegundos
Conhecimento Necessário	Hardware	Software
Capacidade de Armazenamento	Limitada pelo armazenamento disp.	Limitada pela sobrecarga
Largura de Entrada	Obtém vários dados simultâneos	Único processador – um evento

Comparação entre monitores de Software e Hardware

Critério	Hardware	Software
Sobrecarga	Nenhuma	Variável - <5%
Erros	Mais fácil de ocorrer	Raro
Custo	Alto	Baixo
Disponibilidade	Grande— mesmo com crash	Para durante crash
Flexibilidade	Baixa	Alta

Resumo

- Monitores de Software:
 - Gerais e Flexíveis
 - Produzem interferência no sistema
 - Informações possíveis de serem obtidas:
 - Nível de aplicação
 - Sistema operacional
- Monitores de Hardware:
 - Eficientes
 - Menos invasivos
 - Problemas: custo e complexidade

Monitores - Exemplo

- Ganglia
 - Monitor para clusters e grids
 - Métricas e forma de coleta configuráveis
 - Pode ser baseado em evento ou amostragem
 - Em uso por mais de 500 clusters
 - Possui um núcleo + ferramentas auxiliares
 - <http://ganglia.sourceforge.net/>

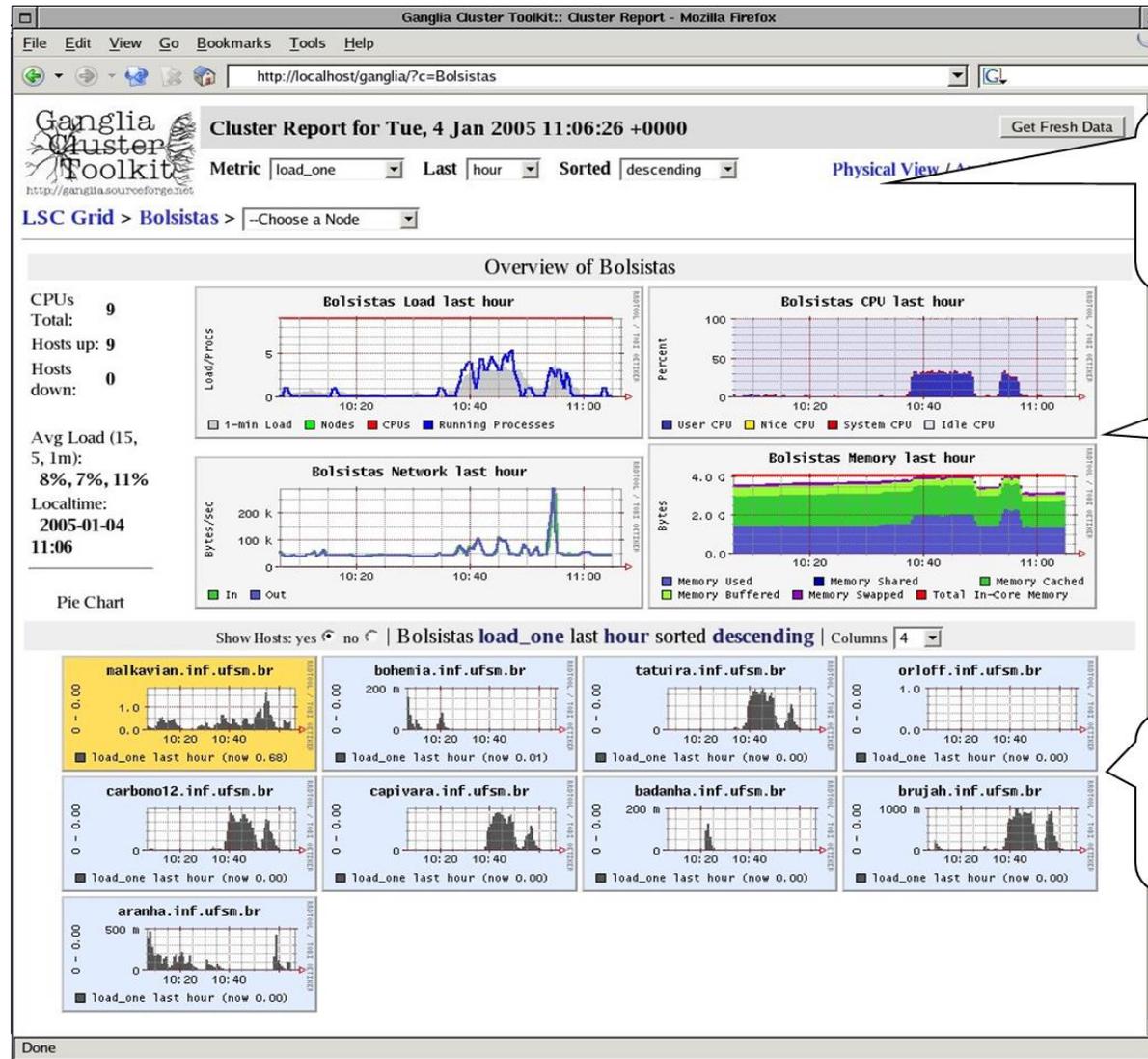
Monitores - Ganglia

- Núcleo:
 - daemon que deve estar em todos os nós do cluster
 - responsável por coletar informações dos nós
- Ferramentas:
 - Gmetric – permite adicionar métricas durante monitoração
 - Gmetad – armazenar infos coletadas
 - RRDtool – armazenamento e visualização dos dados

Monitores - Ganglia

- Propagação da informação coletada é feita por multicast
- Informações enviadas em um documento XML
- Informações armazenadas em um banco de dados RRD (*Round Robin Database*)
- Utiliza XDR (eXternal Data Representation) para transporte dos dados

Monitores - Ganglia



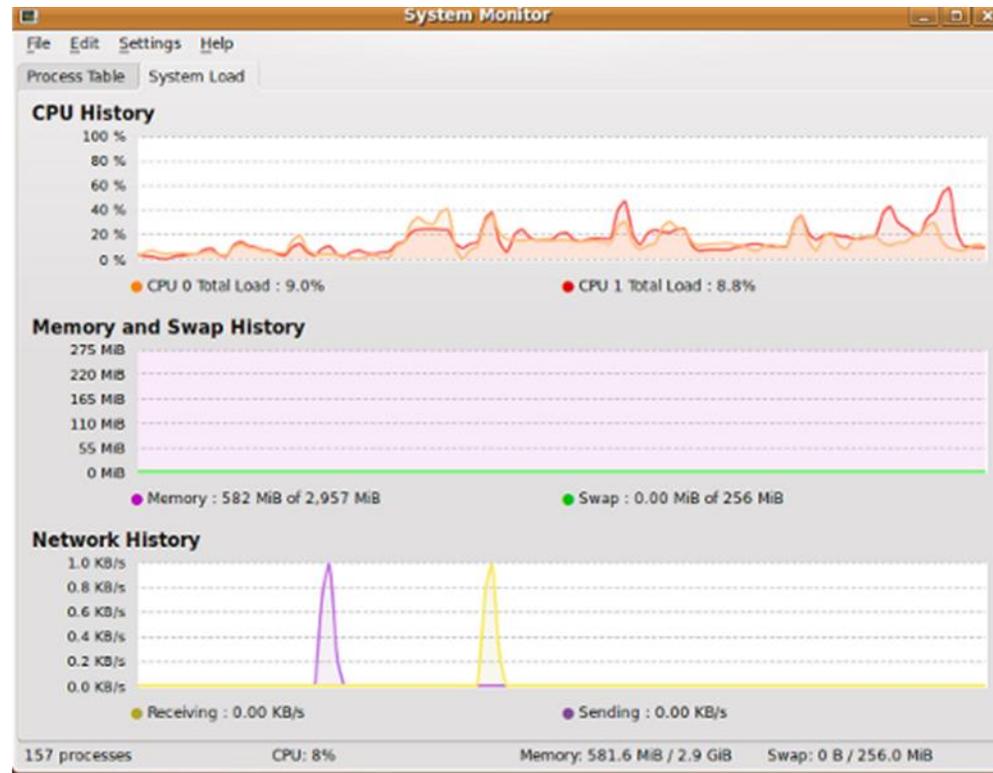
selecionar a métrica e a granularidade de tempo

métricas globais de um cluster

visualização dos estados de cada nó

Monitores - Linux

- Ubuntu



Monitores - Linux

- Linux - /proc
 - Coleta de Informações no diretório /proc
 - Vários sub-diretórios

```
william@escrivao:~$ cat /proc/stat
CPU 220 236 245 161347 1018 0 8
cpu0 220 236 245 161347 1018 0 8 0
intr 415799 407667 10 0 0 0 0 3 0 4 1 0 0 110 0 40 0 5033 2931
...
ctxt 140868
btime 1133036288
processes 4601
procs_running 1
procs_blocked 0
```

Figura 4.1 - Conteúdo do arquivo /proc/stat.

Monitores - Linux

```
william@escrivao:~$ cat /proc/meminfo
MemTotal:      514232 kB
MemFree:      426016 kB
Buffers:      4660 kB
Cached:       42800 kB
SwapCached:   0 kB
Active:       50680 kB
Inactive:     25136 kB
HighTotal:       0 kB
HighFree:        0 kB
LowTotal:        514232 kB
LowFree:         426016 kB
SwapTotal:    1084348 kB
SwapFree:     1084348 kB
Dirty:           128 kB
Writeback:       0 kB
Mapped:          44780 kB
Slab:            8356 kB
CommitLimit:    1341464 kB
Committed_AS:   81100 kB
PageTables:      660 kB
VmallocTotal:   507896 kB
VmallocUsed:     2320 kB
VmallocChunk:   505512 kB
```

Figura 4.2 - Conteúdo do arquivo /proc/meminfo

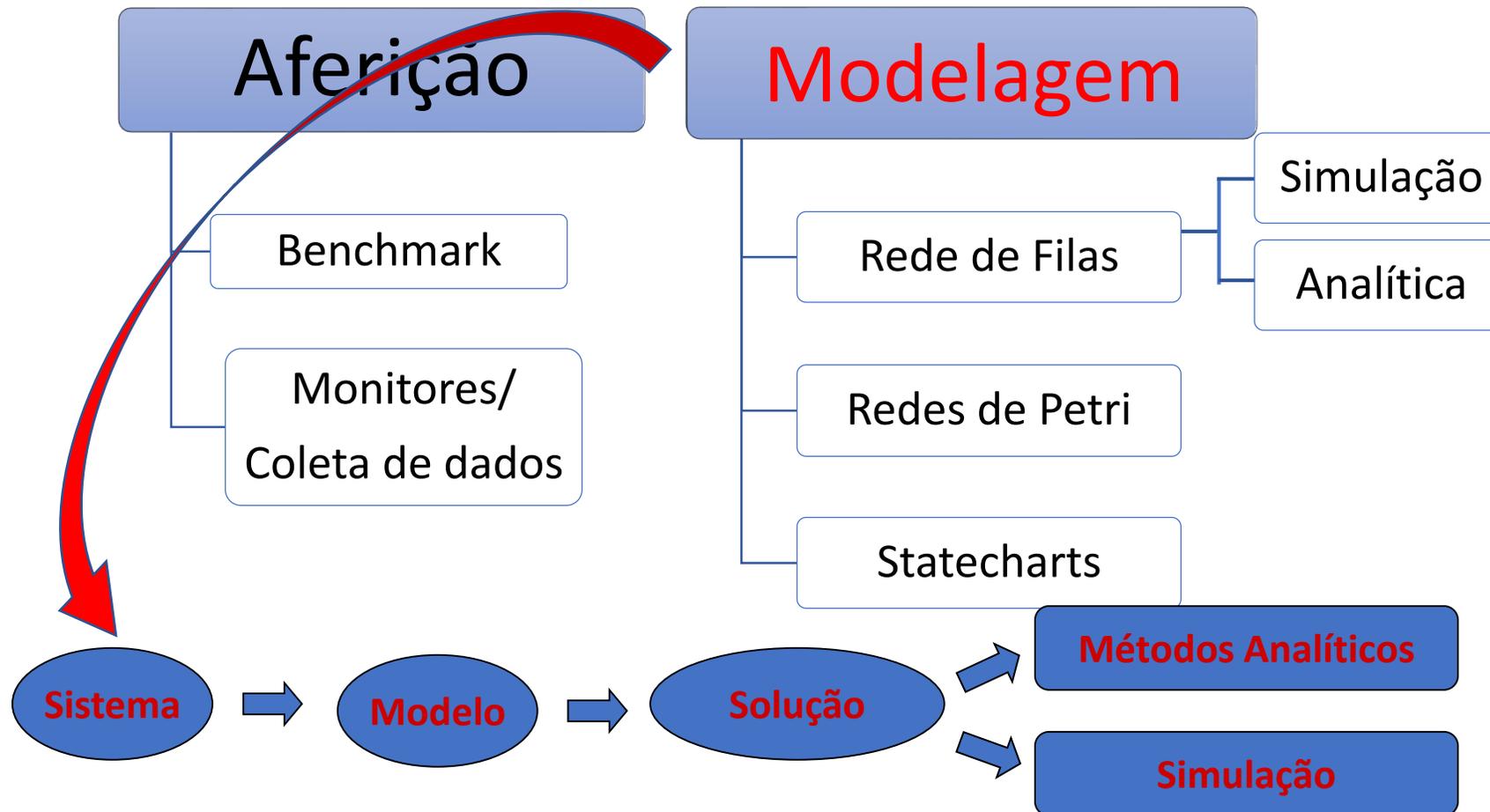
Monitores

- Concluindo....
 - Podem gerar resultados bastante confiáveis.
 - O sistema deve existir e estar disponível.
 - Cuidado com a interferência do Monitor nos resultados
 - Dois tipos básicos de abordagens:
 - Monitores de Software e de Hardware.

Técnicas de Aferição

- Benchmarks
 - Comparação entre Sistemas
 - Avaliar partes específicas de um Sistema
- Monitores ou Coleta de Dados
 - Avaliar um Sistema ou partes dele
 - Avaliação de sistemas existentes – real ou protótipo

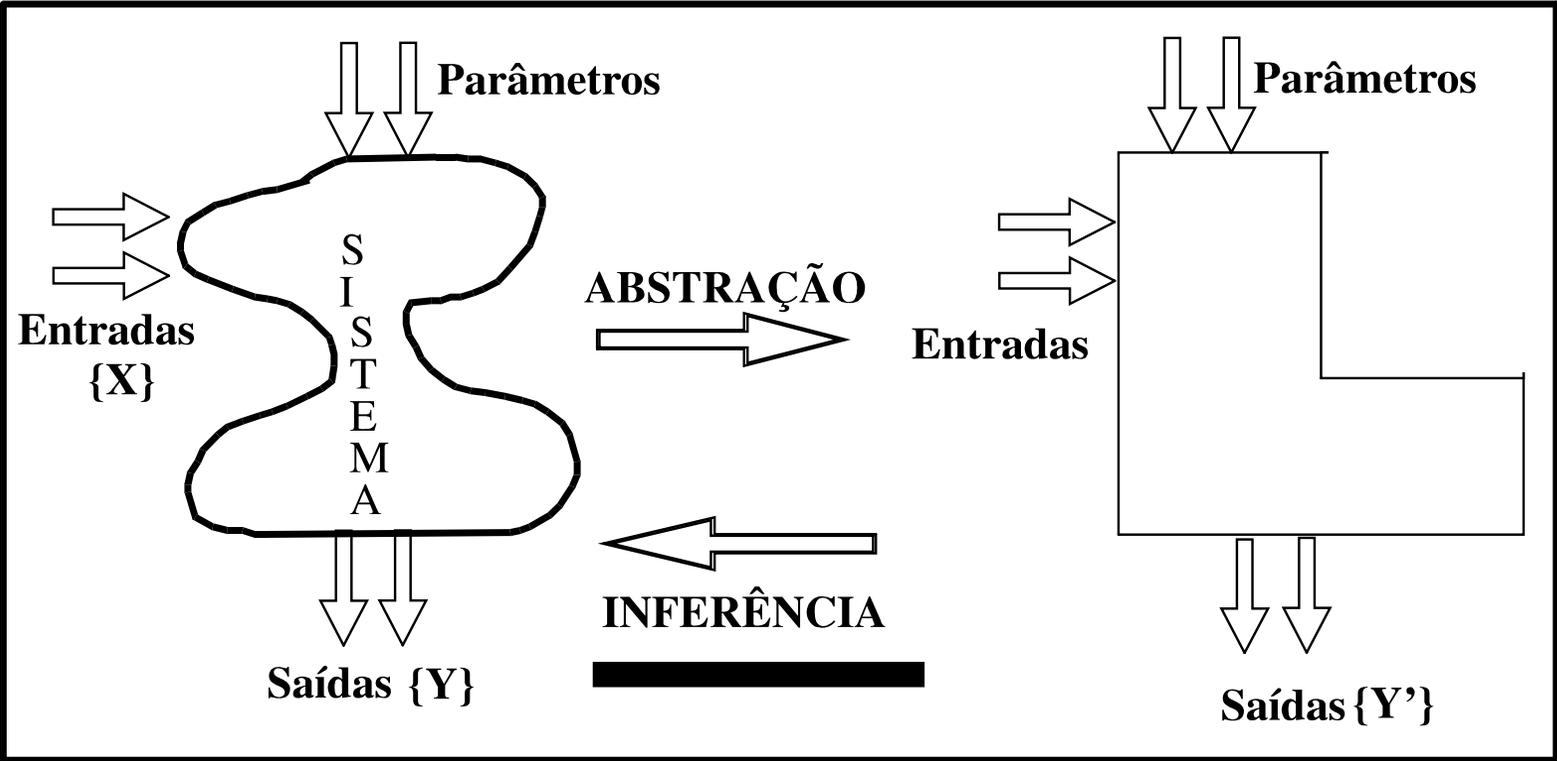
Técnicas de Avaliação de Desempenho



Técnica de Modelagem

- Técnica baseada na construção e análise de um modelo
 - Abstração que contempla as características essenciais de um sistema real
 - Aproximação de como o sistema se comporta
 - Depende dos objetivos da avaliação
 - Sofrem alterações no decorrer do tempo, pois as variáveis de estado de um sistema são dependentes do tempo.

Técnica de Modelagem



Técnica de Modelagem

- Vantagens:
 - Flexibilidade
 - Pode ser utilizada para sistema existente ou não
 - Custo X Precisão
- Dificuldades:
 - Descrição das características principais
 - Validação

Classificação dos modelos

- Quanto ao **comportamento** de suas variáveis de estado em relação ao tempo
 - **Modelo de mudança discreta** (ou *modelo discreto*): variáveis dependentes variam discretamente em pontos específicos do tempo simulado
 - **Modelo de mudança contínuo**: variáveis dependentes podem variar continuamente ao longo do tempo simulado

Sistemas computacionais utilizam-se modelos discretos, uma vez que a alteração de estado em computadores ocorre a intervalos discretos de tempo

Classificação dos Modelos

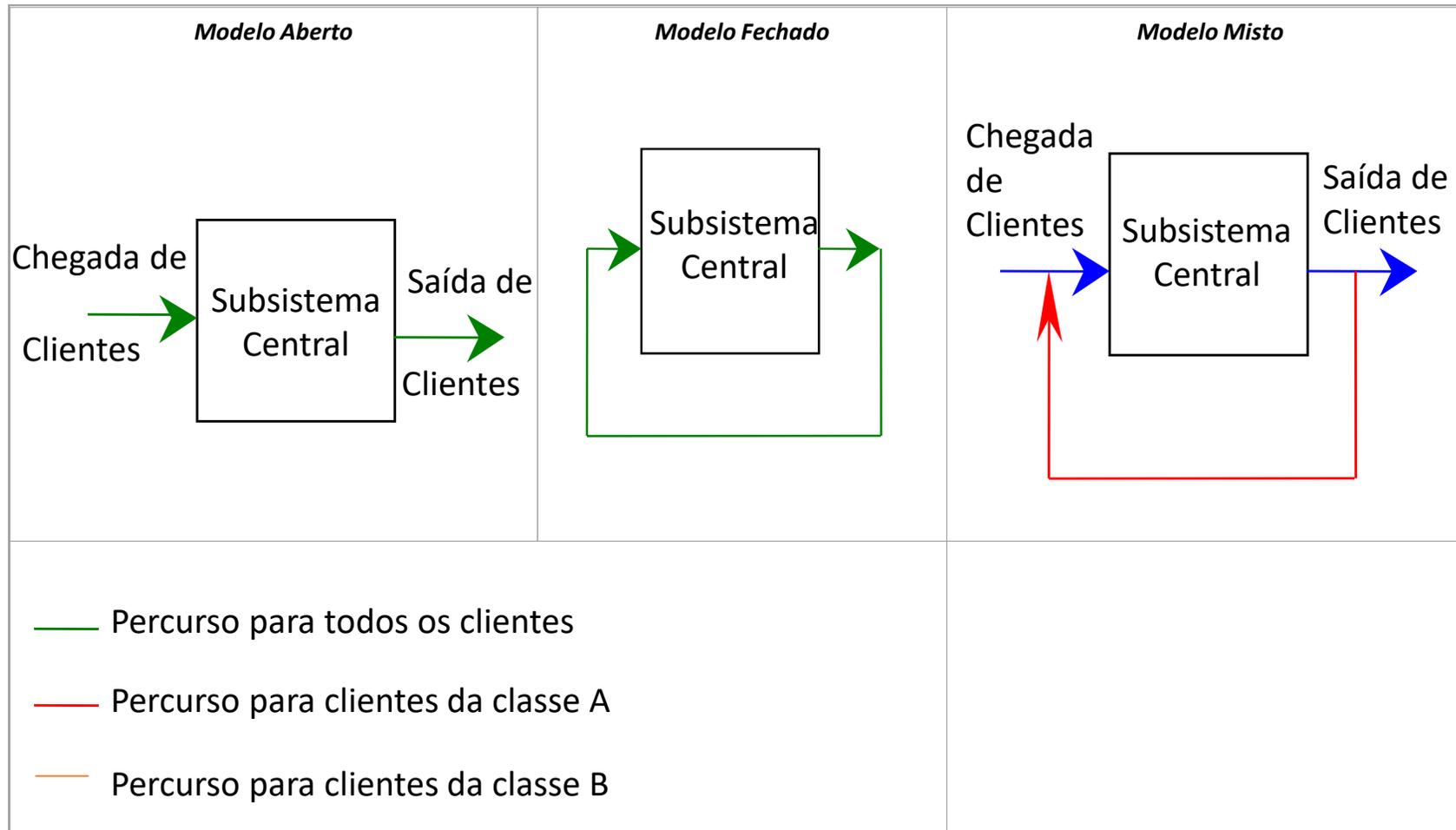
- **Sistemas Abertos**

- Número desconhecido e ilimitado de clientes no sistema
- Carga no sistema – depende da taxa de chegada

- **Sistemas Fechados**

- Número limitado e conhecido de clientes no sistema
- Carga do sistema depende do número de clientes

Classificação dos Modelos



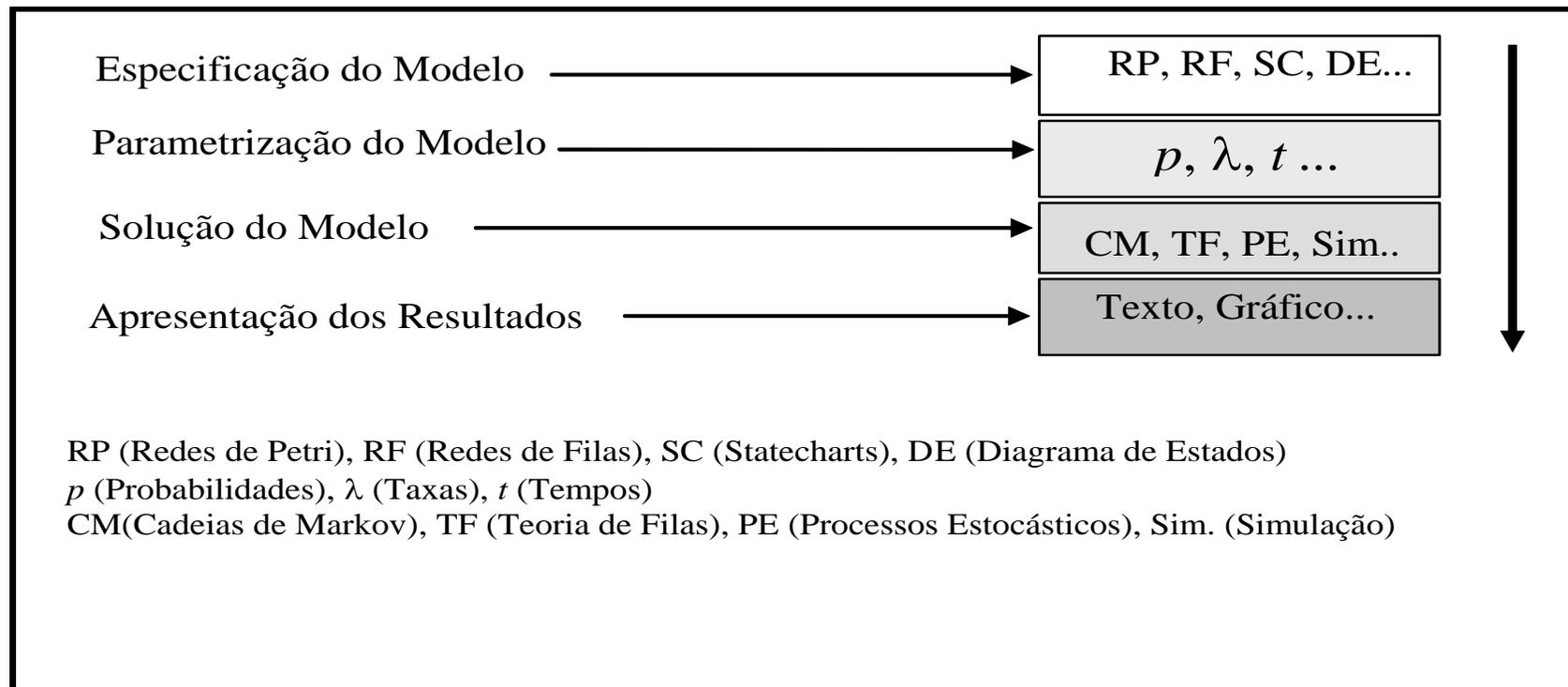
Classificação dos Modelos

- **Classes de Clientes**

- Um modelo pode possuir uma ou mais classes de clientes
- Quando um modelo possui mais de uma classe de clientes, pode-se estabelecer prioridades entre as mesmas, trajetórias distintas para cada delas, etc.

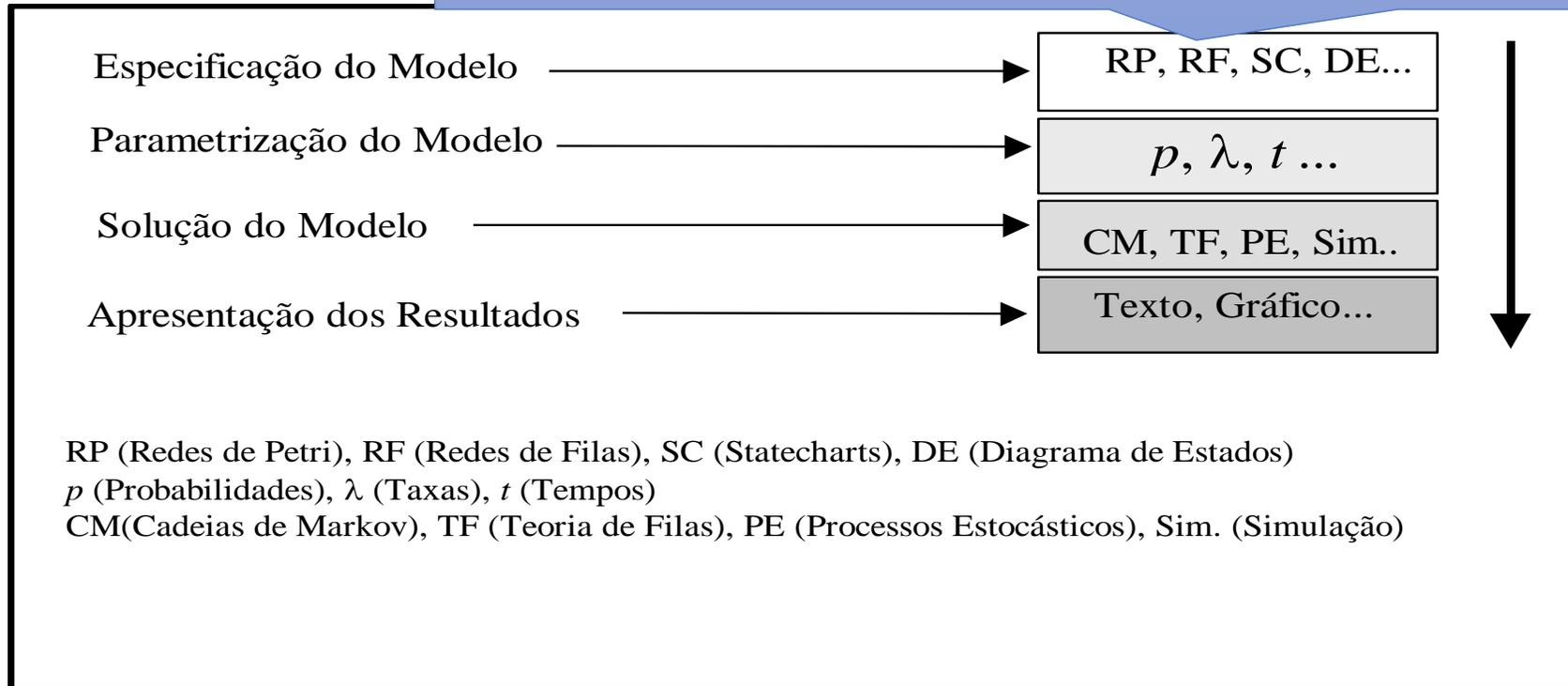
Técnica de Modelagem

- Conjunto de etapas independentes, mas inter-relacionadas

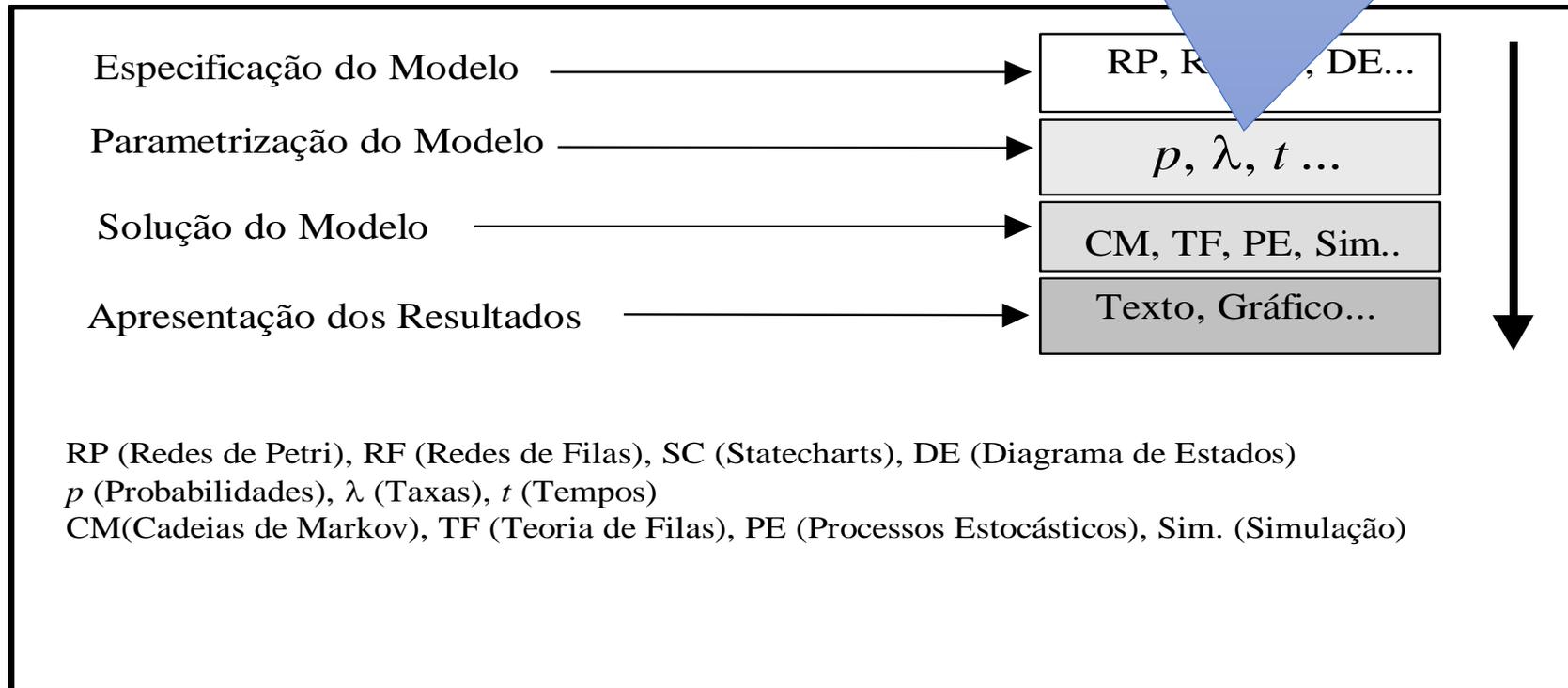


Etapa 1: Criar uma *especificação* condizente com o sistema real

- Componentes do sistema relevantes à avaliação
- Relacionamento entre eles

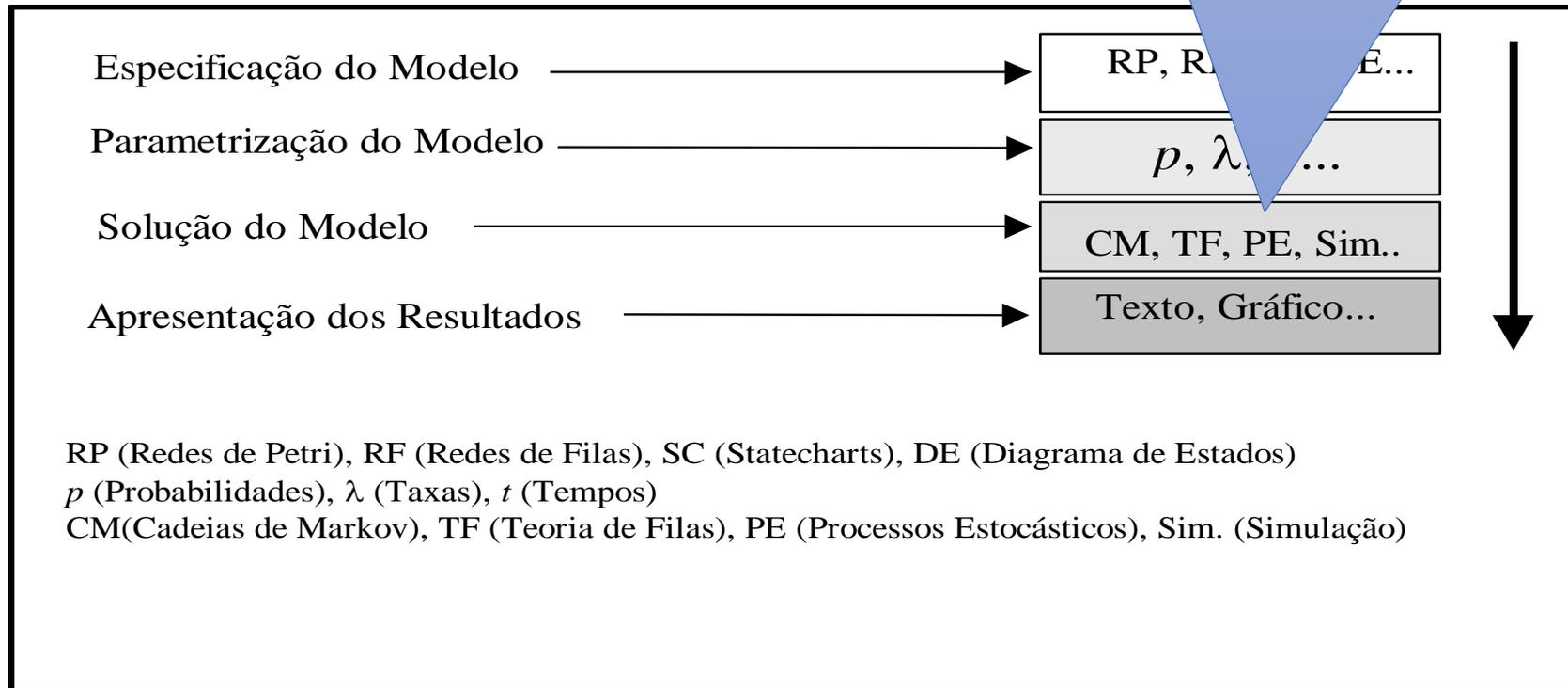


Etapa 2: Parametrizar o modelo com elementos que serão dados de entrada para a próxima fase (solução)

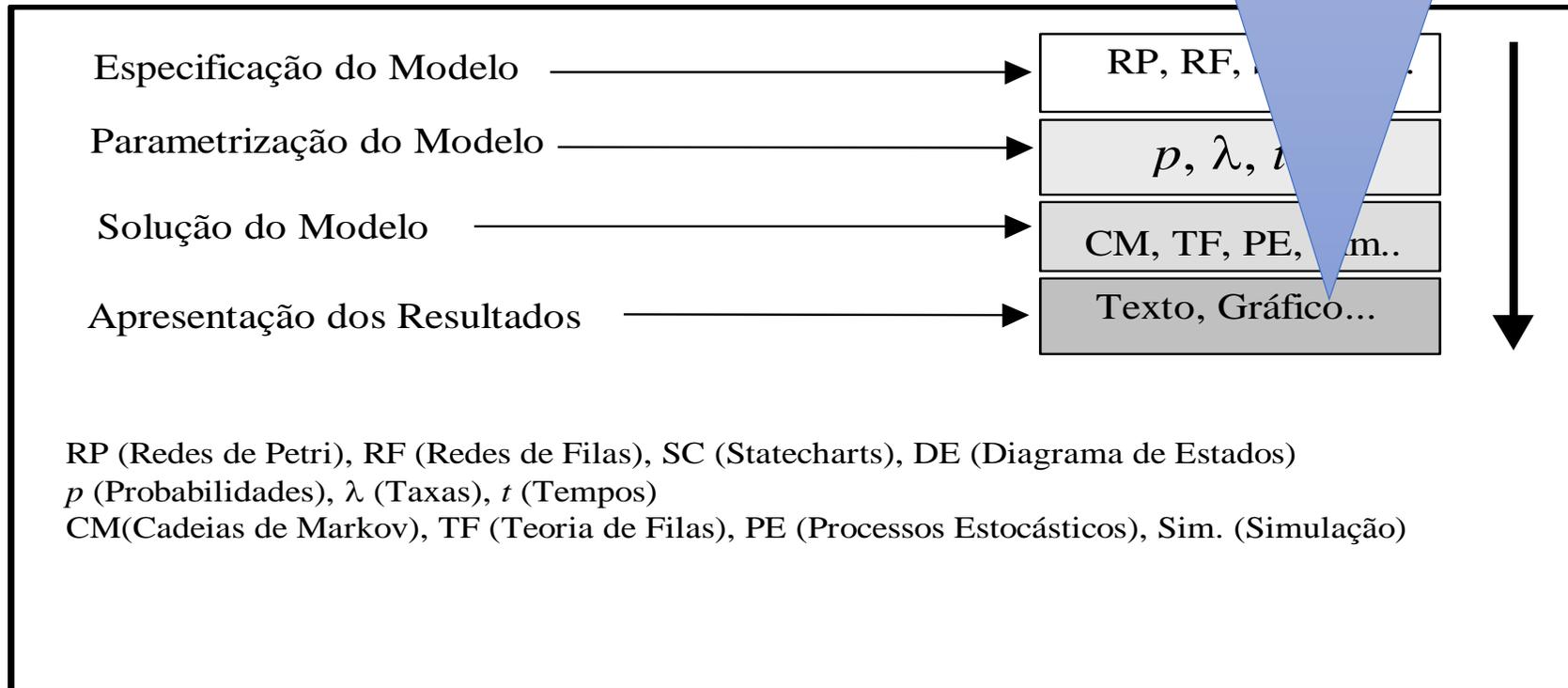


Etapa 3: Solução do modelo

- Aplicação de um método matemático (estocástico), automatizado ou não, para adquirir medidas de desempenho a partir das entradas.



Etapa 4: Apresentar os resultados através de uma maneira conveniente: gráficos, arquivos-texto, etc.

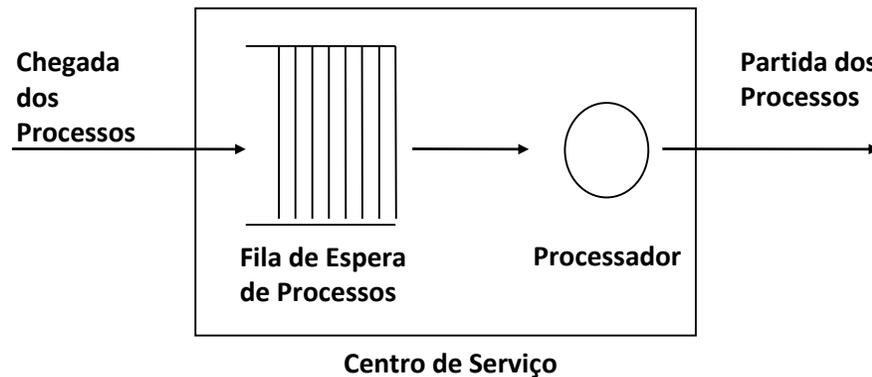


Técnicas de Modelagem

- **Etapa 1:**
 - Criar uma especificação condizente com o sistema real
 - Componentes do sistema relevantes à avaliação
 - Relacionamento entre eles
 - Como representar o modelo:
 - Redes de Filas;
 - Redes de Petri;
 - Statecharts;
 - etc.

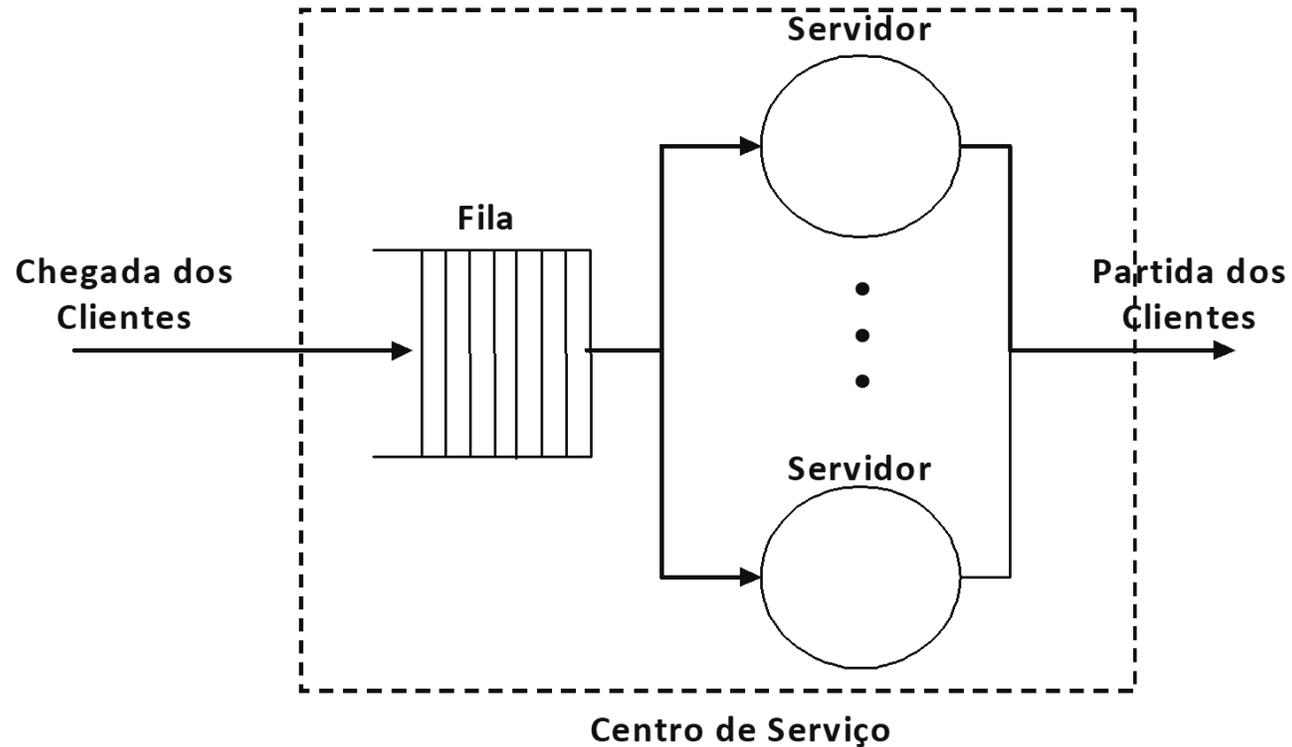
Redes de Filas

- Ramo da probabilidade que estuda o fenômeno da formação de filas de solicitantes de serviços, que são providos por um determinado recurso



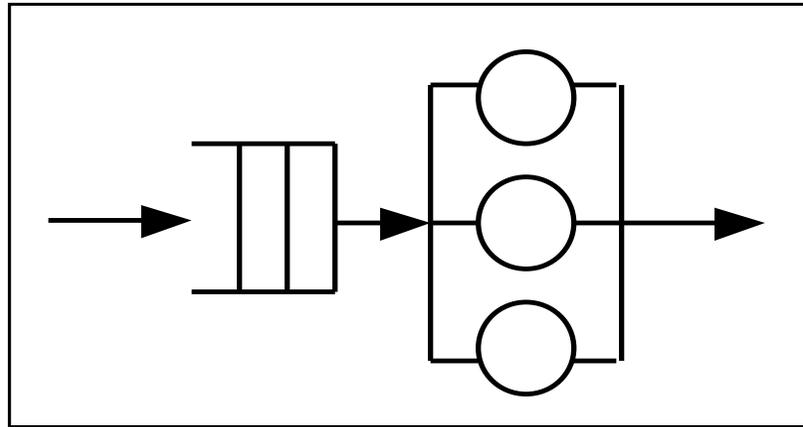
Redes de Filas

- Um centro de serviços pode ter um ou mais servidores e uma ou mais filas

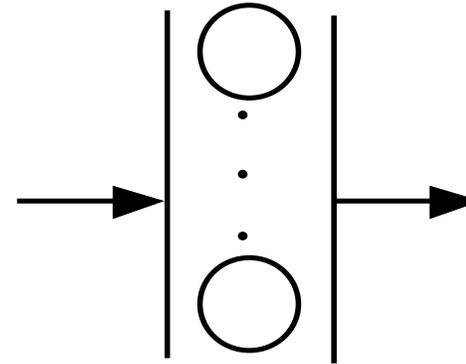


Redes de Filas

- Os centros de serviços podem ser de **capacidade fixa** ou **centros de delay**



Centro de Capacidade Fixa



Centro *Delay*

- Nos **centro delay** não há competição, pois existe um número muito grande de servidores disponíveis, sendo que o serviço é infinitamente disponível

Redes de Filas

- Exemplos:
 - Cinema com uma bilheteria
 - Banco com quatro caixas
 - Correio tipo 1 – Agência de Água Vermelha, com:
 - 1 caixa
 - Dois tipos de cliente: comum e preferencial

Redes de Filas

- Exemplos:
 - Correio tipo 2 – Agência de São Carlos com:
 - Dois tipos de serviço: comum (5 caixas) e sedex (2 caixas)
 - Dois tipos de cliente: comum e preferencial
 - Incluir reserva de recurso para cliente preferencial

Use fila única sempre que possível!!!

Redes de Filas

- Exemplos:
 - Correio tipo 3 – Agência de São Paulo, com:
 - Três tipos de serviço: comum (10 caixas), sedex (2 caixas), encomendas (2 caixas)
 - Dois tipos de clientes: comum e preferencial
 - Parte dos clientes passam pelo balcão para fechar envelope (eu possui infinitos tubos de cola) antes de ir para as filas dos caixas

Notação para Sistemas de Filas

- **A/B/c/K/m/Z**
 - A -> tempo entre chegadas;
 - B -> distribuição tempo de serviço;
 - c -> número de servidores;
 - K -> capacidade do sistema;
 - m -> número de clientes na fonte;
 - Z -> disciplina da fila.

Notação para Sistemas de Filas

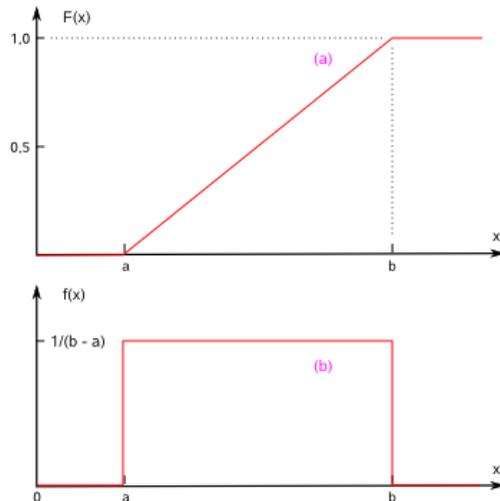
- Normalmente usa-se apenas $A/B/c$:
 - não há limite para o tamanho da fila;
 - fonte de clientes é infinita;
 - disciplina da fila é FCFS
- $M/M/1$
 - taxas de chegadas entre clientes - distribuição exponencial
 - tempo de serviço - distribuição exponencial
 - um único servidor

Notação para Sistemas de Filas

- Parâmetros A e B são normalmente representados por Distribuições de Probabilidade
 - Uniforme
 - Exponencial
 - Erlang
 - Triangular

Notação para Sistemas de Filas

- Distribuição uniforme
 - a probabilidade de se gerar qualquer ponto em um intervalo contido no espaço amostral é proporcional ao tamanho do intervalo



A função de distribuição acumulada:

$$F(x) = 0 \quad \text{para } x < a$$

$$F(x) = (x - a) / (b - a) \quad \text{para } a \leq x < b$$

$$F(x) = 1 \quad \text{para } x \geq b$$

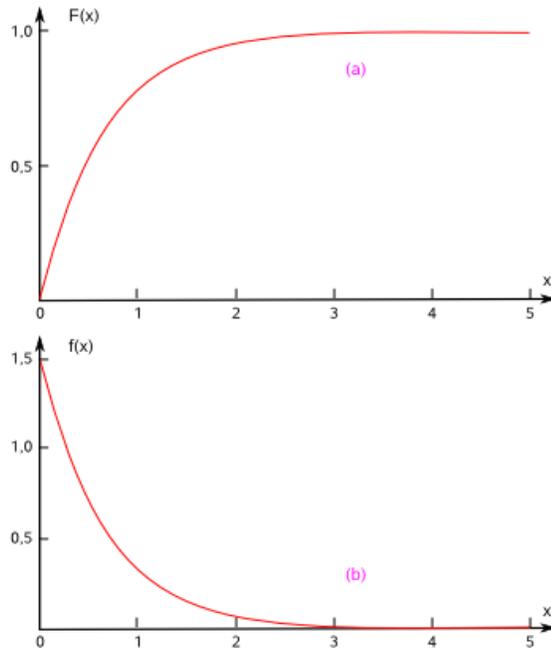
Função densidade de probabilidade:

$$f(x) = 1/(b-a), \quad \text{para } a \leq x \leq b$$

$$f(x) = 0, \quad \text{cc}$$

Notação para Sistemas de Filas

- Distribuição Exponencial



A função de distribuição acumulada:
 $F(x) = P(X \leq x) = 1 - e^{-\lambda x}$ para $x \geq 0$

Média = $1 / \lambda$

Função densidade de probabilidade:

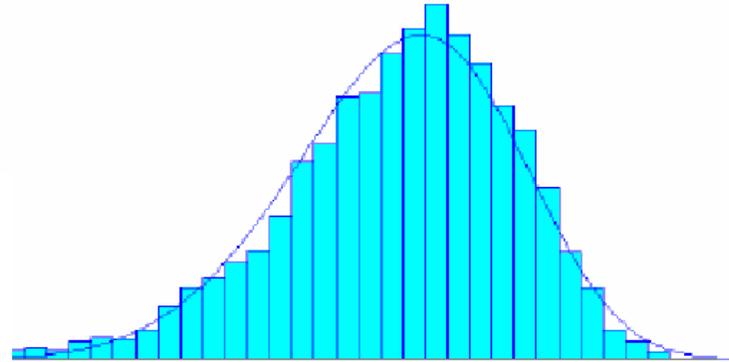
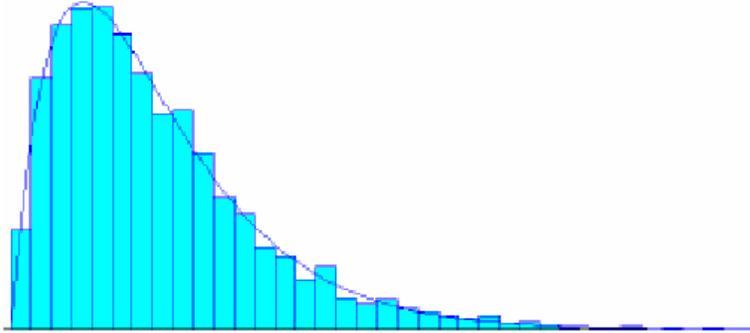
$f(x) = \lambda e^{-\lambda x}$ para $x \geq 0$ e

$f(x) = 0$ para $x < 0$

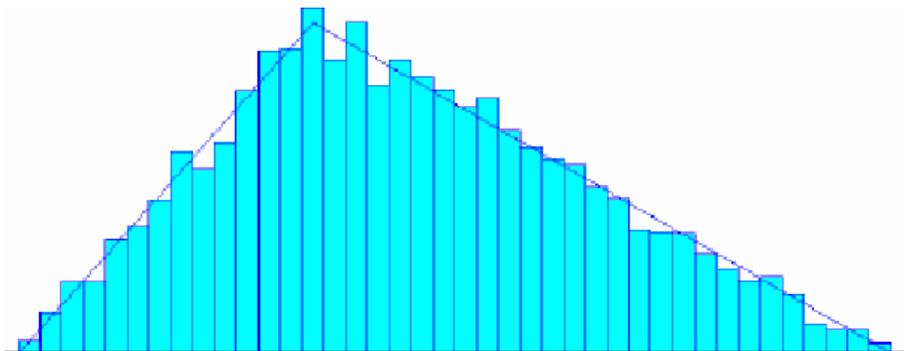
- Exemplo: O tempo de espera para se obter um arquivo em um servidor, segue uma distribuição exponencial e, em média, leva 80mseg para ser atendido. Qual a probabilidade de levar mais de 100mseg?
 - $P(X > 100) = 1 - P(X \leq 100) = 1 - F(100) = 1 - 1 + e^{-(1/80) 100} \approx 0,2865$

Notação para Sistemas de Filas

Erlang



Triangular

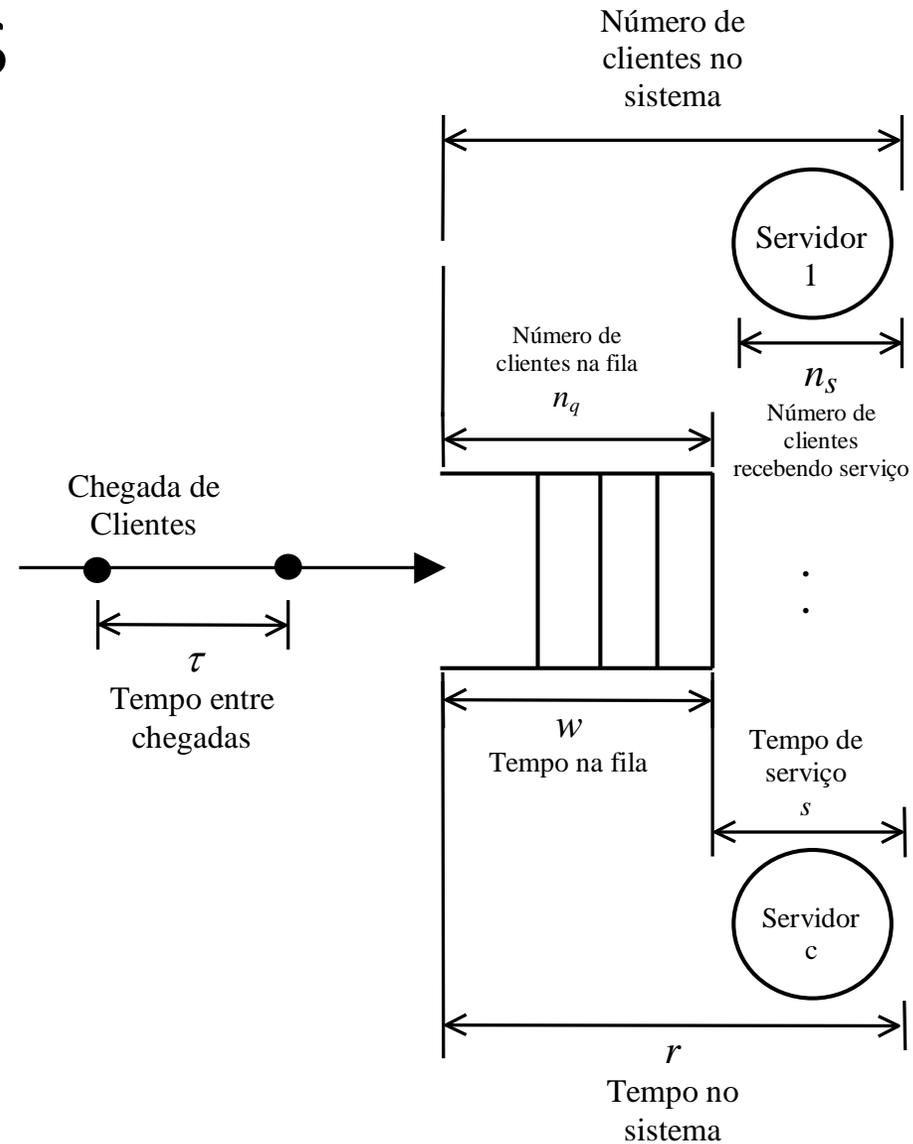


Notação para Sistemas de Filas

- Disciplinas de Atendimento de Clientes
 - FCFS - First Come First Served;
 - LCFS - Last Come First Served;
 - RR - Round Robin (Circular);
 - Prioridades e Preempção.

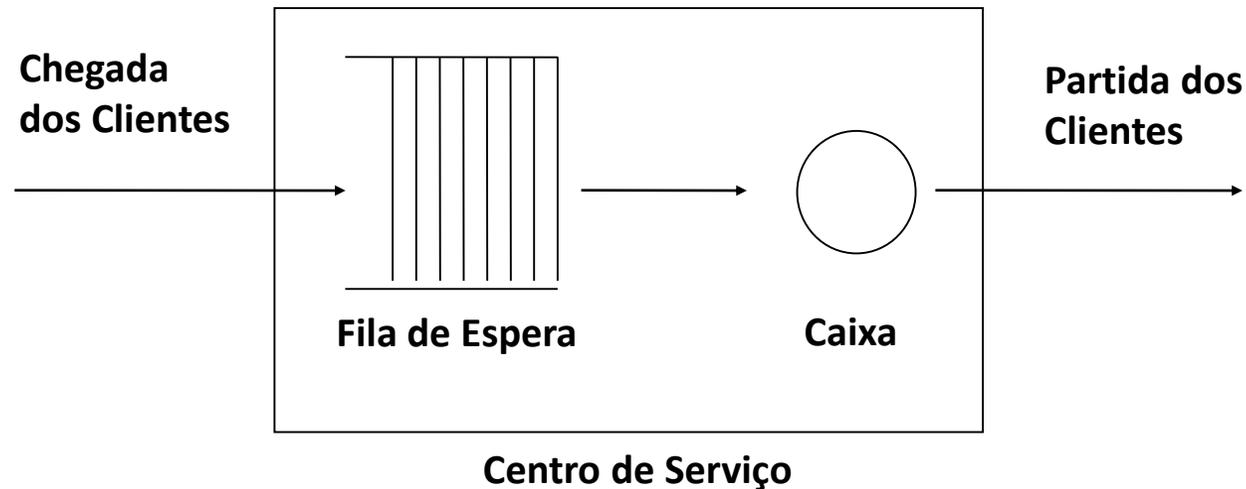
Variáveis Aleatórias em um Sistema de Filas

Sistema descrito
por uma combinação
de variáveis aleatórias e
respectivas distribuições



Exemplos – Redes de Filas

- Cinema com uma bilheteria

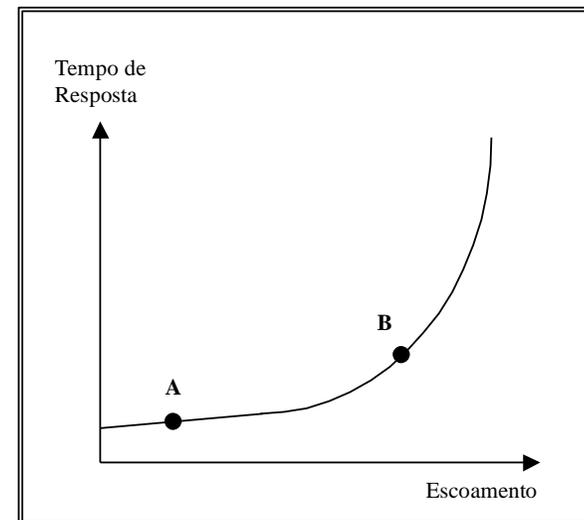
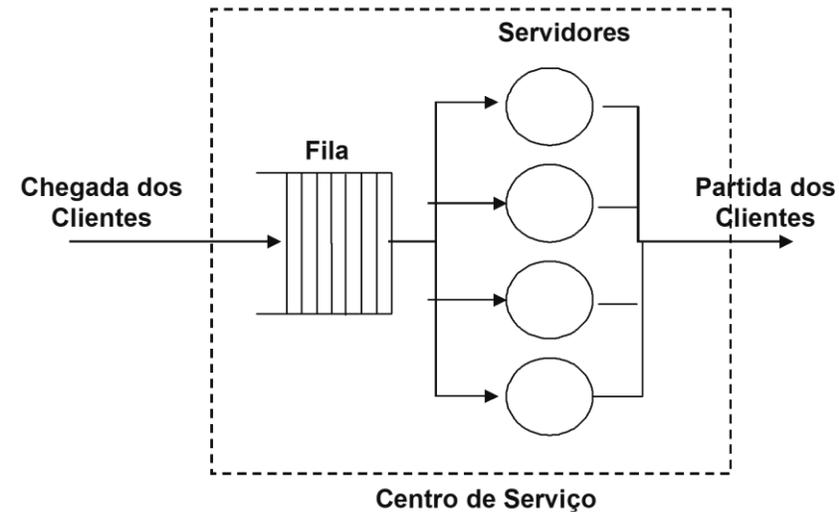


- Quantos clientes são atendidos por unidade de tempo?
- Qual o número médio de clientes na fila?
- Qual o tempo médio de espera para compra do ingresso?
- Qual o tempo médio necessário para comprar o ingresso?

Exemplos – Redes de Filas

Banco com quatro caixas

- Qual o tempo médio de espera na fila?
- Qual o tempo de atendimento total?
- Tempo de atendimento/na fila encontra-se em um patamar desejado? Ponto A da figura
- Qual a consequência em se diminuir um caixa?
- A fila e o tempo de atendimento diminuirá substancialmente aumentando um caixa? Compensa aumentar um caixa?
- Quantos caixas devem ser abertos para trazer o sistema para próximo ao ponto A?



Exemplos - Redes de Filas

- Sistema de arquivos com um processador e quatro unidades de disco. Algumas requisições (30%) não necessitam acessar os discos para serem executadas.
- Sistema de arquivos onde um servidor central recebe as requisições e redireciona para outros quatro servidores.

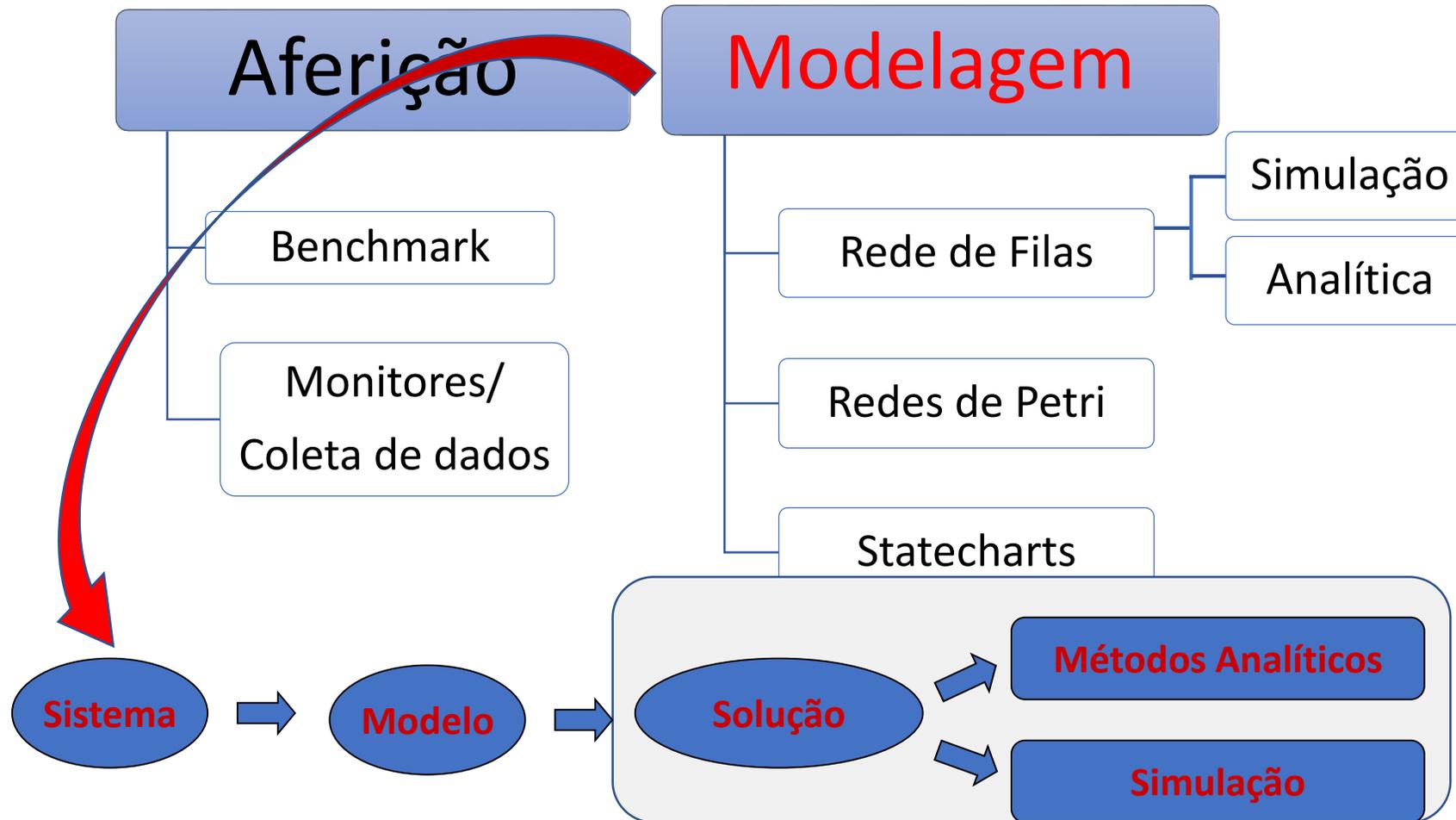
Redes de Filas

- **Restrições das redes de filas:**
 - Representação gráfica oferece apenas os elementos fila e servidor
 - Não permite posse simultânea de recursos
 - Necessidade de representação mais minuciosa de filas e servidores
 - Isso leva a uma perda da realidade quando representando sistemas reais.

Redes de Filas

- Por outro lado...
 - a representação do caminho linear que os clientes traçam através do sistema é descrita com bastante propriedade, noção que na maioria das técnicas é perdida com facilidade
 - possuem uma base matemática bastante solidificada

Técnicas de Avaliação de Desempenho



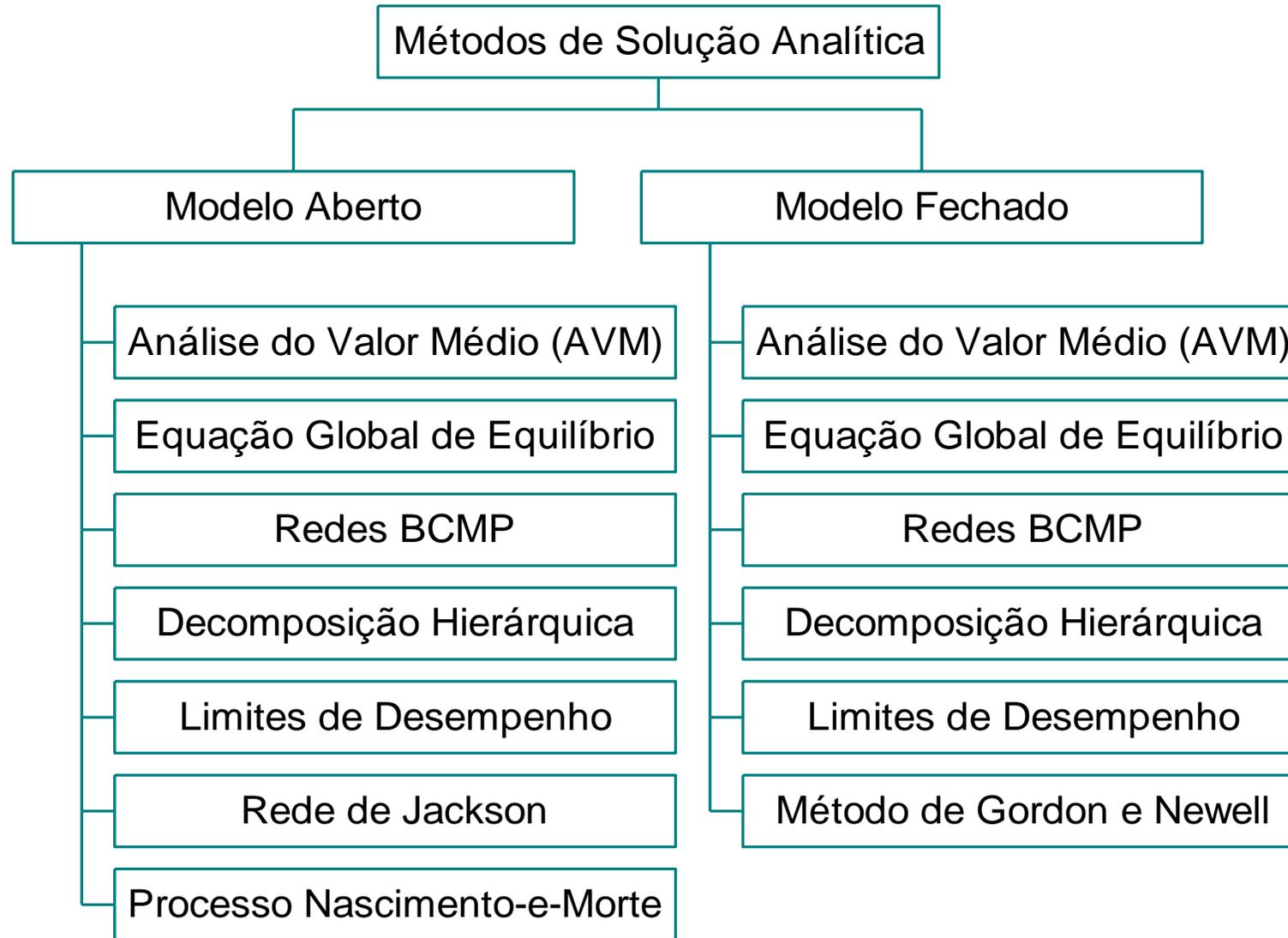
Soluções para o Modelo

- Solução Analítica
 - Modelos analíticos descrevem a operação do sistema e sua carga em termos completamente abstratos.

Soluções para o Modelo

- Solução Analítica
 - Estimativas de desempenho são obtidas através da resolução analítica ou numérica do modelo matemático
 - Relação funcional entre parâmetros do sistema e critérios de desempenho escolhidos
 - Rápido
 - Resultados exatos
 - Restrição - Simplificações
 - Modelo baseado em redes de filas

Solução Analítica



Soluções para o Modelo

- Solução por Simulação
 - Modelos de simulação são programas de computador nos quais a operação de um sistema e sua carga são descritas utilizando-se algoritmos apropriados.

Soluções para o Modelo

- Solução por Simulação
 - Estimativas de desempenho são obtidas através da execução de um programa que simula o modelo
 - Resultados estocásticos
 - Necessidade de tratamento estatístico
 - Tempo para obtenção dos resultados
 - Flexível

Técnicas de Avaliação de Desempenho

