Modelos de Redes de Filas com Uma Classe (Análise de Valores Médios)

Profa. Jussara M. Almeida

1º Semestre de 2014

Modelos com Uma Classe

- Provê estimativas de medidas de desempenho (não simplemente limites)
- Uma única classe pode ser usada quando:
  - Próximo passo para modelos mais detalhados
  - Uma única carga de trabalho de interesse
  - Cargas homogêneas
- Uma única classe pode não ser apropriada quando:
  - Cargas de trabalhos múltiplas e distintas (batch x timesharing, CPU-bound x I/O bound)
  - Entradas/Saídas do modelo dependentes da classe

  Neste caso, usar modelos com múltiplas classes (a seguir)

Representação da Carga de Trabalho

- Conjunto de demandas + intensidade da carga
- Intensidade da carga = λ , N ou N e Z
- Carga aberta ou fechada:
  - Número de clientes ilimitado?
- Nota:
  - Tempo de resposta de modelos abertos tendem a ser maiores que os correspondentes em modelos fechados para o mesmo throughput: longas filas em potencial

Modelo de Filas X Sistema Real: Carga Fixa X Carga variável

- Modelos são úteis em:
  - Cenários com cargas muito pesadas (há sempre processos esperando por memória): modelo batch
  - Intensidade da carga não é inteiro: interpolação
  - Estudos de capacidade: Qual a máxima taxa de processamento de transação suportada com um tempo de resposta médio inferior a 3 segs?
  - Robustez: estimativa de crescimento de carga imprecisa; avaliar modelo para valores em torno do valor esperado
  - Intensidade da carga dada como distribuição

Simplicidade dos modelos

Modelo de Filas X Sistema Real: Carga Fixa X Carga variável

- Intensidade da carga dada como distribuição

n	$P[N=n]$	$U_{CPU}(n)$	$X(n)$	$R(n)$
0	.1	0	0	0
1	.2	.032	.0525	.787
2	.3	.062	.1031	1.546
3	.3	.092	.1515	2.273
4	.1	.119	.1974	2.961

$$U_{CPU} = \sum_{n=1}^{4} P[N=n]\, U_{CPU}(n) = .0645$$

$$R = \sum_{n=1}^{4} \left[\frac{X(n)P[N=n]}{\sum_{j=1}^{4} X(j)P[N=j]} R(n) \right] = 2.492$$

Estudo de Casos Práticos

1. Um modelo para um sistema interativo
   (problema clássico/histórico – talvez o primeiro a aplicar modelos de filas em CS)

2. Um modelo com análise de modificações

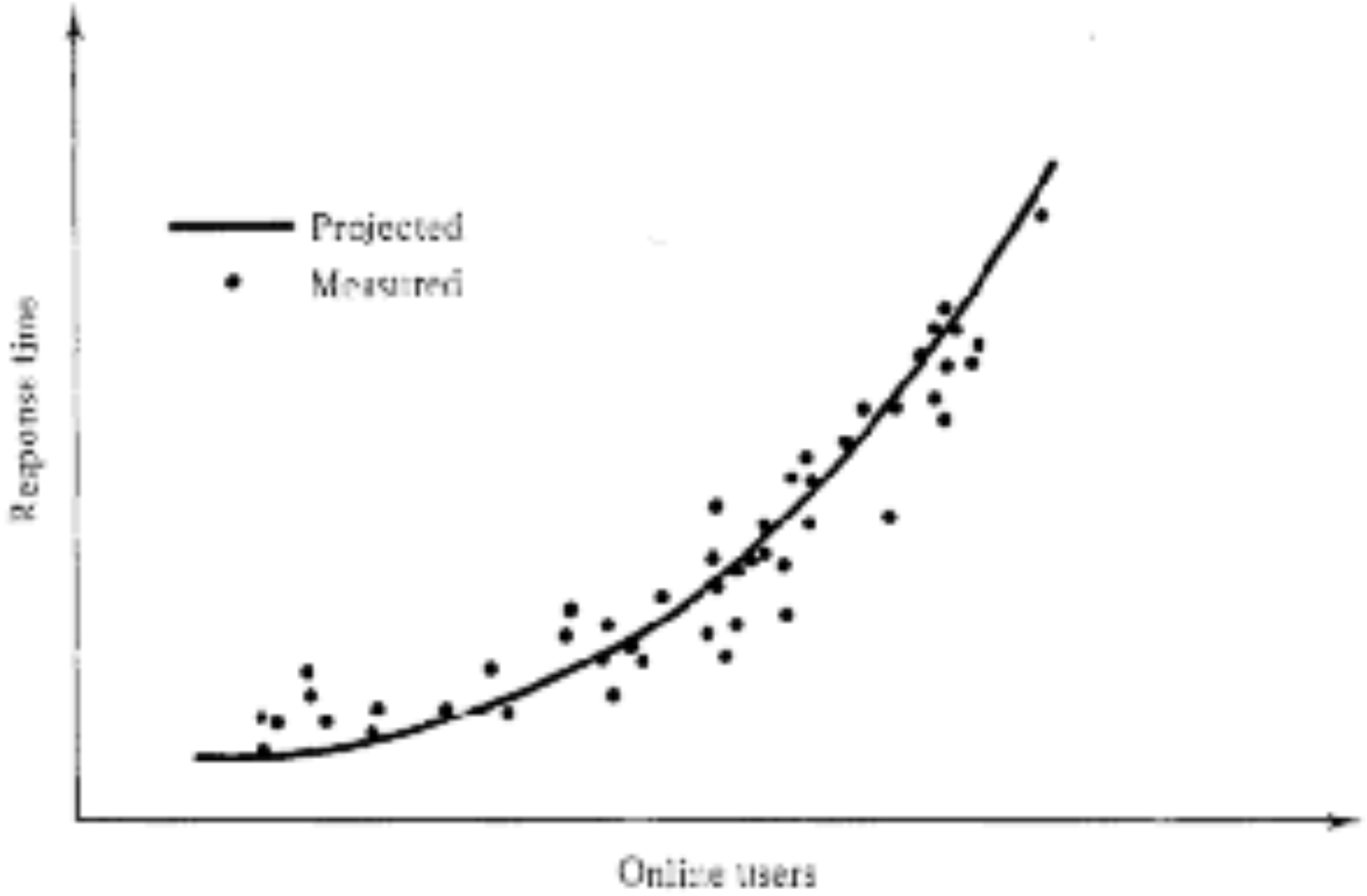
3. Planejamento de capacidade
   (Lazowska, ch. 6)

Modelo para um Sistema Interativo
[1965]

- Avaliar tempo de resposta do sistema CTSS em função do # de usuários
  - CPU + discos + memória é time-sliced: um unico usuário ativo no sistema a cada instante
- Carga interativa com um único centro de serviço
  - Caso contrário: múltiplos usuários usando CPU e discos simultaneamente leva a modelos mais imprecisos
  - Agregação de múltiplos centros em um único (solucao para problemas de restricao de memoria)
- Medições para parametrização do modelo
  - Think time médio,
  - Tempos médio de processamento de CPU e disco
  - Demanda no centro de serviço = tempo de processamento de CPU + tempo de serviço no disco (swapping job)

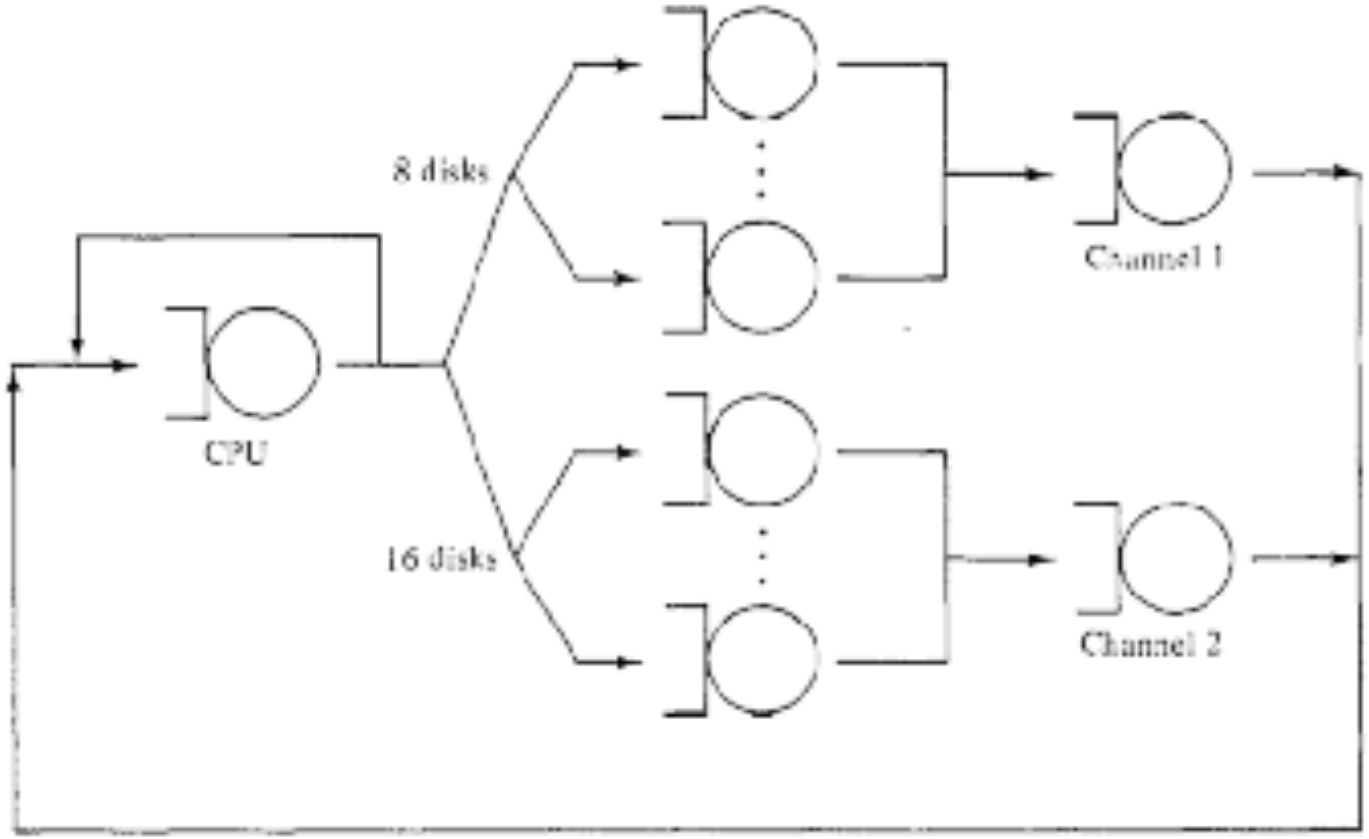
Modelo para um Sistema Interativo

[1965]



Modelo para Análise de Modificações

- Avaliar os benefícios de propostas de mudanças alternativas no hardware e software de um sistema
  - Sistema + 24 discos + 2 canais (controladoras)



Modelo para Análise de Modificações: Parametrização

- Demanda por CPU: tempo total CPU ocupada dividida pelo número de jobs completaram serviço
  - Alocação homogênea de overhead de CPU (escalonamento, interrupção de I/O)

Modelo para Análise de Modificações: Parametrização

- Subsistema de I/O: complicado
  no sistema real, tanto canal como disco eram ocupados durante latência rotacional e transferência
  - Demanda pelo canal = tempo médio de latência + tempo médio de transferência
  - Demanda por disco= tempo médio de seek.
  - Imprecisão:
    - No modelo, disco e canal não são ocupados ao mesmo tempo pelo mesmo cliente: paralelismo artificial
      - Desafio para modelagem de subsistemas de I/O
    - No caso em questão: # de discos muito maior que nível de multiprogramação médio: pequena prob. paralelismo

Modelo para Análise de Modificações: Avaliação

- Medicoes indicaram grande variacao no nivel de multiprogramacao
- Avaliação para cada nível de multiprogramação observado
  - Projeções de desempenho foram obtidos tomando a média ponderada das soluções distintas (% de tempo observado em cada nível)
- Avaliação de (combinações) dos upgrades:
  - Trocar 8 discos em 1 canal por discos mais rápidos
  - Substituir CPU por uma mais rápida
  - Otimização do SO: reduz porção da demanda de CPU correspondente ao processamento no SO.
- Modelo simples forneceu resultados precisos (Lazoska, ch.6)

Soluções para Modelos de Filas

- Conjunto de medidas de desempenho que descrevem o comportamento médio (no tempo), ou a longo prazo, do modelo.
- Soluções para redes de filas genéricas são difíceis, mas existem soluções simples para redes que exibem características específicas
- Premissa Básica para os procedimentos descritos a seguir:
  - Modelos de redes de filas separáveis
- Soluções diferentes para modelos abertos e modelos fechados

Redes de Filas Separáveis: Premissas

- Equilíbrio de fluxo em cada centro de serviço: número de chegadas no centro = número de saídas
- Comportamento *one step*: dois clientes não mudam de estado no sistema ao mesmo tempo
- Homogeneidade de roteamento: prob. de cliente saindo do centro k ir para centro j não depende do tamanho da fila em qualquer centro
- Homogeneidade de dispositivo: taxa de processamento de clientes em um centro não depende do número e localização de clientes na rede como um todo (mas pode depender do # clientes no centro em particular)
- Homogeneidade de chegadas externas: Os momentos de chegadas de clientes (externos ao sistema em avaliação) não dependem do número e localização de clientes dentro da rede.

Premissa Extra para Algoritmos Dados (MVA)

- Homogeneidade de tempo de serviço: taxa de serviço de um centro não depende do número de clientes no centro (*load independent)* e do número e localização de clientes na rede

- Redes com load-dependent centers exigem outras técnicas para resolução (Cap 8 e 20 do Lasowska)

Redes Separáveis com Uma Única Classe: Definição Alternativa

- Cada centro de serviço deve ter uma das seguintes políticas de escalonamento:
  - FIFO, Processor Sharing,
    Last-Come First-Served Preemptive Resume,
    além de outras menos comuns
  - Priority scheduling: **não pode**
- Se a política é FIFO, os tempos de serviço têm que seguir uma distribuição exponencial
- Chegadas externas devem seguir processo Poisson
- Probabilidades de roteamento de clientes não podem depender do estado da rede

Aplicação das Premissas de Redes Separáveis

- Premissas necessárias para garantir, matematicamente, que soluções dos modelos são exatas
- Mundo real: premissas nem sempre são razoáveis (ex: homogeneidade)
- Na prática: a violação das premissas pode não ser uma fonte de imprecisão significativa
  - Experiência mostra que a precisão dos modelos é muito robusta a violações nas premissas
  - Tipicamente, a falha na validação vem de imprecisões na representação do sistema ou nos valores dos parâmetros
- Exceção: Casos em que as restrições da estrutura do modelo não capturam aspectos primários
  - Ex: restrições de memória, escalonamento com prioridade
  - Neste caso, precisa técnicas mais avançadas: coleção de modelos separáveis avaliados conjuntamente
- O investimento em modelos mais avançadas deve ser feito com critério!
  - Compromisso PRECISÃO vs. SIMPLICIDADE

Modelos de Filas Abertos: Soluções

- Capacidade de Processamento: taxa de chegada de clientes em que o modelo satura

$$\lambda_{sat} = 1 / D_{max}$$

- Throughput

$$X = \lambda \qquad X_k(\lambda) = \lambda V_k$$

- Utilização

$$U_k(\lambda) = X_k(\lambda) S_k = \lambda D_k$$

(a utilização de um delay center deve ser interpretada como o # médio de clientes presente)

Modelos de Filas Abertos: Soluções

- Tempo de Residência: tempo total gasto por um cliente em um centro, incluindo tempo de fila e tempo recebendo serviço

Delay center: $R_k = V_k S_k = D_k \quad (Z)$

Centros de serviço: $R_k = V_k S_k$ + tempo de fila

Seja: $A_k(\lambda)$ = número médio de clientes na fila na visão do cliente que acabou de chegar

Tempo de fila = $V_k (A_k(\lambda) S_k)$

Premissa: tempo esperado para o cliente *em serviço* terminar de ser servido é igual ao tempo de serviço

Modelos de Filas Abertos: Soluções

$$R_k(\lambda) = V_k S_k + V_k A_k(\lambda) S_k = D_k [\ 1 + A_k(\lambda)\ ]$$

Teorema da Chegada (*Arrival Theorem*): cliente chegando em um centro de serviço vê um tamanho de fila igual ao *steady state*

$A_k(\lambda) = Q_k(\lambda) =$ tamanho da fila médio (no tempo)

(redes separáveis)

$$R_k(\lambda) = D_k [\ 1 + Q_k(\lambda)\ ]$$

$$R_k(\lambda) = D_k [\ 1 + \lambda R_k(\lambda)\ ]$$ Lei de Little

$$R_k(\lambda) = D_k / (\ 1 - U_k(\lambda)\ )$$

$U_k(\lambda) \to 0,\ R_k(\lambda) \to D_k$ $U_k(\lambda) \to 1,\ R_k(\lambda) \to \infty$

Modelos de Filas Abertos: Soluções

- Tamanho da fila : inclui quem está em serviço

$Q_k(\lambda) = \lambda R_k(\lambda) = U_k(\lambda)$ (delay center)

$U_k(\lambda) / (1 - U_k(\lambda))$ (centros c/ fila)

- Tempo de Resposta do Sistema

$$R(\lambda) = \sum_{k=1}^{K} R_k(\lambda)$$

- Número Médio de clientes no sistema

$$Q(\lambda) = \lambda R(\lambda) = \sum_{k=1}^{K} Q_k(\lambda)$$

Modelos de Filas Abertos: Algoritmo

processing capacity : $\lambda_{sat} = 1 / D_{max}$

throughput : $X(\lambda) = \lambda$

utilization : $U_k(\lambda) = \lambda D_k$

residence time :

$$R_k(\lambda) = \begin{cases} D_k & \text{(delay centers)} \\ \dfrac{D_k}{1 - U_k(\lambda)} & \text{(queueing centers)} \end{cases}$$

queue length : $Q_k(\lambda) = \lambda R_k(\lambda)$

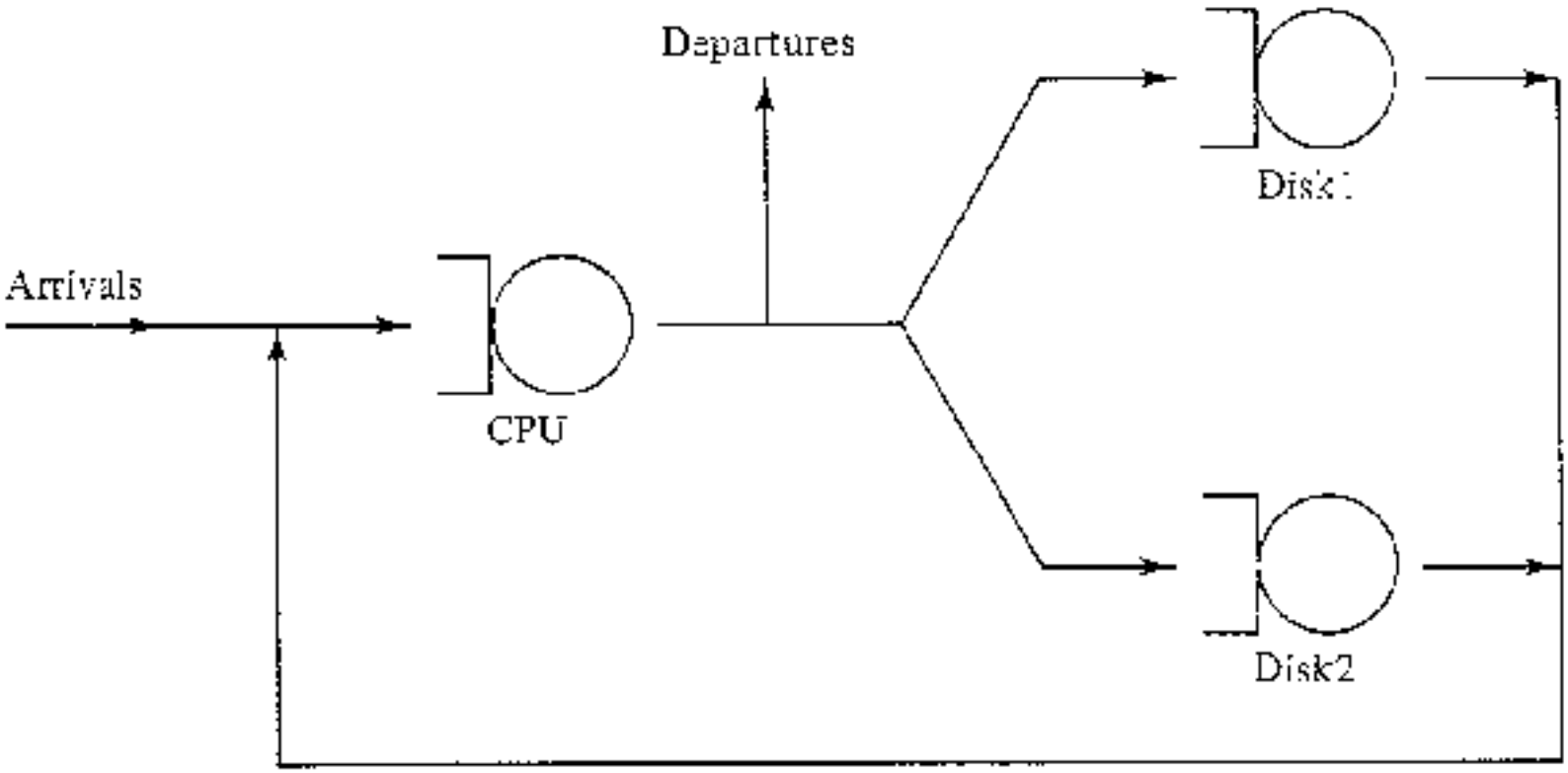
$$= \begin{cases} U_k(\lambda) & \text{(delay centers)} \\ \dfrac{U_k(\lambda)}{1 - U_k(\lambda)} & \text{(queueing centers)} \end{cases}$$

system response time : $R(\lambda) = \sum_{k=1}^{K} R_k(\lambda)$

average number in system : $Q(\lambda) = \lambda R(\lambda) = \sum_{k=1}^{K} Q_k(\lambda)$

Complexidade: O(K)

Modelos de Filas Abertos: Exemplo



Modelos de Filas Abertos: Exemplo

- $V_{CPU} = 121$ $V_{disk1} = 70$ $V_{disk2} = 50$
- $S_{CPU} = 0.005$ $S_{disk1} = 0.030$ $S_{disk2} = 0.027$
- $D_{CPU} = 0.605$ $D_{disk1} = 2.1$ $D_{disk2} = 1.35$

$\lambda = 0.3$ processos/seg

Modelos de Filas Abertos: Exemplo

$$\lambda_{sat} = \frac{1}{D_{max}} = \frac{1}{2.1} = .476 \text{ jobs/sec.}$$

$$X_{CPU}(.3) = \lambda V_{CPU} = (.3)(121) = 36.3 \text{ visits/sec.}$$

$$U_{CPU}(.3) = \lambda D_{CPU} = (.3)(.605) = .182$$

$$R_{CPU}(.3) = \frac{D_{CPU}}{1 - U_{CPU}(.3)} = \frac{.605}{.818} = .740 \text{ secs.}$$

$$Q_{CPU}(.3) = \frac{U_{CPU}(.3)}{1 - U_{CPU}(.3)} = \frac{.182}{.818} = .222 \text{ jobs}$$

$$R(.3) = R_{CPU}(.3) + R_{Disk1}(.3) + R_{Disk2}(.3)$$

$$= .740 + 5.676 + 2.269 = 8.685 \text{ secs.}$$

$$Q(.3) = \lambda R(\lambda) = (.3)(8.685) = 2.606 \text{ jobs}$$

Modelos de Filas Fechados

- Solução por MVA = mean value analysis (análise de valores médios)
- Throughput $$X(N) = \frac{N}{Z + \sum_{k=1}^{K} R_k(N)}$$
- Tamanho médio de cada fila $Q_k(N) = X(N)R_k(N)$
- Tempo de Residência

Delay center: $R_k = V_k S_k = D_k \quad (Z)$

Centros de serviço: $R_k = D_k (1 + A_k(N))$

Modelos de Filas Fechados

- Diferentemente do modelo aberto, nos modelos fechados $A_k(N) \neq Q_k(N)$
  - $K = 2$ e $N = 1$: $Q_k(1) = \frac{1}{2}$ mas $A_k(1) = 0$
- Soluções exatas e aproximadas (com relacao ao modelo, nao ao sistema real)

Análise de Valores Médios (MVA): Solução Exata

- Tamanho da fila no instante de chegada $A_k(N)$ calculado de forma exata

  Teorema da Chegada (*Arrival Theorem*)

  $$A_k(N) = Q_k(N - 1)$$

  "Um cliente chegando ao sistema o ve em um estado que nao inclui o novo cliente".

- Solução iterativa

Análise de Valores Médios (MVA) Exata

for $k \leftarrow 1$ to K do $Q_k \leftarrow 0$

for $n \leftarrow 1$ to N do

begin

for $k \leftarrow 1$ to K do $R_k \leftarrow \begin{cases} D_k & \text{(delay centers)} \\ D_k(1 + Q_k) & \text{(queueing centers)} \end{cases}$

$$X \leftarrow \frac{n}{Z + \sum_{k=1}^{K} R_k}$$

for $k \leftarrow 1$ to K do $Q_k \leftarrow XR_k$

end

Complexidade: O(NK)

Análise de Valores Médios (MVA) Exata

- Throughput do sistema: X
- Tempo de Resposta do Sistema: N/X – Z
- Número Médio no Sistema: N – XZ
- Throughput do dispositivo *k* : XV_k
- Utilização do dispositivo *k* : XD_k
- Tamanho da fila do dispositivo *k*: Q_k
- Tempo de residência do dispositivo *k* : R_k

Análise de Valores Médios (MVA) Exata: Exemplo

- D_{CPU}= 0.605 D_{disk1} = 2.1 D_{disk2} = 1.35 N = 3 Z = 15

	k	$n=0$	$n=1$	$n=2$	$n=3$
R_k	*CPU*	-	.605	.624	.644
	*Disk*1	-	2.1	2.331	2.605
	*Disk*2	-	1.35	1.446	1.551
X		-	.0525	.1031	.1515
Q_k	*CPU*	0	.0318	.0643	.0976
	*Disk*1	0	.1102	.2403	.3947
	*Disk*2	0	.0708	.1490	.2350

Análise de Valores Médios (MVA): Solução Aproximada

- Tamanho da fila no instante de chegada $A_k(N)$ calculado de forma aproximada

$$A_k(N) = h\,(Q_k(N)\,)$$

- O que é a função h()?
  - Várias aproximações

Análise de Valores Médios Aproximada (AMVA): Aproximação de Bard-Schweitzer

- Aproximação de Bard Schweitzer:

$$h() = (N - 1) / N \times Q_k(N)$$

- Premissa: $Q_k(N) / N = Q_k(N-1) / (N-1)$
- Aproximação boa nos dois extremos: assintoticamente exata para valores grandes de N e trivial para modelos com N = 1
- Boa na prática para outros cenários também.

Análise de Valores Médios Aproximada (AMVA): Algoritmo

1. Inicialização: $Q_k(N) = N / K$ para todos os centros k
3. Aproximação: $A_k(N) = (N - 1) / N \times Q_k(N)$
4. Calcule:

$R_k = D_k \ (= Z)$ centro de atraso

$= D_k (1 + (N - 1)/ N \times Q_k(N))$ centro de serviço

$$X(N) = \frac{N}{Z + \sum_{k=1}^{K} R_k(N)}$$

$Q^{new}_k(N) = X(N)R_k(N)$

4. Se $| Q^{new}_k(N) - Q_k(N) | > 0.1\% \ Q_k(N)$:

$Q_k(N) = Q^{new}_k(N)$, retorne ao passo 2

Análise de Valores Médios (AMVA) Aproximada: Exemplo

- D_{CPU}= 0.605 D_{disk1} = 2.1 D_{disk2} = 1.35 N = 3 Z = 15

iteration	Q_{CPU}	Q_{Disk1}	Q_{Disk2}	X	R
0	1.00	1.00	1.00		
1	.1390	.4826	.3102	.1379	6.7583
2	.0988	.4150	.2436	.1495	5.0659
3	.0972	.4043	.2366	.1508	4.8950
4	.0972	.4024	.2359	.1510	4.8732
5	.0973	.4021	.2359	.1510	4.8700
exact solution	.0976	.3947	.2350	.1515	4.8020

Exercício 1

Um servidor de arquivos com dois discos idênticos é observado durante 1 hora. Processando os logs de acesso do servidor coletados durante o período em questão, você observa que as 360 requisições processadas foram para arquivos com tamanho médio de 0.5 MB. Além disto, a saída do comando de iostat executado durante o período de medição mostra uma utilização de CPU de em torno de 60%, e um tempo médio de serviço em cada disco de 12 ms.

Sabendo que o tamanho de um bloco de disco é 512 bytes (parâmetro do SO) e que os arquivos estão armazenados nos discos seguindo o modelo *striping* (os blocos dos arquivos são divididos igualmente entre os dois discos), calcule:

a) Throughput e tempo de resposta médios do servidor
b) Tempo médio gasto por cada requisição esperando por servico
c) Número médio de requisições servidas simultaneamente pelo servidor
d) Número médio de requisições simultaneamente disputando a CPU
e) Numero médio de requisicoes na fila de cada disco

Exercício 2

Imagine que o mesmo servidor de arquivos do exemplo anterior seja agora configurado para atender a um máximo N de requisições simultâneas. Assumindo que não haja nenhuma razão para crer que o tamanho médio dos arquivos requisitados mude, responda as perguntas a), b) e c) para cenários fictícios em que o servidor está configurado com N= 2, 4, e 6.

Dentre estes valores, qual você sugeriria para configurar o servidor se o tempo de resposta médio deve ser inferior a 40 seg?

Qual o erro de aproximação no tempo de resposta e throughput médios do sistema para N = 6 quando AMVA é usado?

Exercício 3

Você é contratado pela Companhia Xulambis para realizar um estudo do desempenho do seu sistema de armazenamento. Você começa o estudo reduzindo o escopo para o cache e os cinco discos *idênticos,* que, segundo o gerente da companhia, são os principais gargalos de desempenho identificados previamente.

Cada operação de I/O (requisição para um bloco de dado) submetida ao sistema passa primeiro pelo cache, uma memória mais rápida que armazena os blocos mais recentemente acessados. Caso o bloco requisitado não seja encontrado no cache, o pedido é enviado para um dos 5 discos do sistema.

Sabe-se que cada acesso ao cache é servido em 2 milissegundos, independente do bloco ser ou não encontrado. Além disto, sabe-se que os blocos de dados são distribuídos aleatoriamente entre os 5 discos, isto é, a probabilidade de um bloco requisitado estar em um determinado disco é igual para todos os discos.

A especificação do hardware estima um tempo médio de serviço no disco de 10 ms.

Sabendo que a probabilidade de um bloco ser encontrado no cache é de 0.6 e que a taxa de chegada de requisições para blocos é de 50 requisições por segundo, faça:

- Mostre o modelo de filas correspondente ao sistema analisado
- Calcule o tempo médio de resposta do sistema.
- Calcule a utilização de cada disco.
- Calcule o número médio de requisições na fila de cada disco (não inclui em serviço).
- Calcule o tempo médio de espera de cada requisição nas filas do sistema.

Exercício 4

Um amigo seu, iniciante em modelagem de sistemas, lhe apresenta os dados da tabela abaixo, obtidos a partir de medições em um sistema real e também de um modelo MVA de filas separáveis, fechado, que ele construiu. O sistema (real e modelado) tem 3 centros de serviço, uma carga única e 4 clientes. O seu amigo lhe pede ajuda pois, apesar da concordância entre modelo e dados medidos ser excelente para várias métricas de desempenho, ela é muito ruim para algumas outras. Ele lhe pergunta: as discrepâncias são provavelmente causadas por erro nas medições realizadas no sistema ou no código de solução do modelo MVA? Justifique sua resposta apresentando resultados quantitativos que a suportem.

Dados Experimentais	Estimativas a partir do Modelo MVA
$S_1 = 70$	
$S_2 = 12$	
$S_3 = 25$	
$V_1 = 1$	
$V_2 = 2$	
$V_3 = 2$	
$X = 0.01$	$X = 0.01$
$R_1 = 198$	$R_1 = 198$
$U_1 = 70\%$	$U_1 = 70\%$
$R_2 = 18$	$R_2 = 14$
$U_2 = 30\%$	$U_2 = 12\%$
$R_3 = 100$	$R_3 = 87$
$U_3 = 50\%$	$U_3 = 40\%$