



**Universidade Federal do Rio de Janeiro**  
Departamento de Ciência da Computação

## **Trabalho de Simulação**

Autores:

Marco Vinícius Lima Reina de Barros

Pedro Henrique Pereira de Jesus

Ronald Andreu Kaiser

Todos os membros do grupo participaram da implementação do código e da documentação do relatório.

28 de Novembro de 2010

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Funcionamento geral . . . . .	2
1.2	Estruturas internas utilizadas . . . . .	3
1.3	Linguagem de Programação . . . . .	3
1.4	Geração de variáveis aleatórias . . . . .	3
1.5	Métodos utilizados . . . . .	4
1.6	Implementação do conceito de cores . . . . .	4
1.7	Escolha dos parâmetros . . . . .	4
1.8	Máquina utilizada . . . . .	5
<b>2</b>	<b>Teste de Correção</b>	<b>6</b>
<b>3</b>	<b>Estimativa da fase transiente</b>	<b>7</b>
3.1	No simulador: . . . . .	7
3.2	Gráficos: . . . . .	7
<b>4</b>	<b>Resultados</b>	<b>13</b>
4.1	Tabelas . . . . .	13
4.2	Comentários . . . . .	13
<b>5</b>	<b>Otimização</b>	<b>16</b>
<b>6</b>	<b>Conclusões</b>	<b>17</b>
<b>7</b>	<b>Implementação</b>	<b>18</b>

# Capítulo 1

## Introdução

### 1.1 Funcionamento geral

O simulador possui uma lista de eventos que é processada continuamente, até alcançar um número máximo de clientes que desejamos atender por rodada.

São executadas tantas rodadas quanto forem necessárias até todos os intervalos de confiança dos valores que estão sendo estimados forem válidos, ou seja,  $\leq 10\%$  da média do estimador.

Inicialmente, calculamos o tempo de chegada do primeiro cliente que representa um evento de chegada no sistema. A passagem de um cliente pelo sistema possibilita a criação dos seguintes eventos:

- <tempo, tipo: chegada no sistema>
- <tempo, tipo: entrada no servidor pela primeira vez>
- <tempo, tipo: saída do servidor>
- <tempo, tipo: entrada no servidor pela segunda vez>

Quando um evento de chegada ocorre, outro evento de chegada é criado com o tempo definido com o tempo de chegada baseado em uma distribuição exponencial, que representa o tempo de chegada do próximo cliente. Deste modo, os clientes vão chegando no sistema e a lista de eventos é processada.

Quando um evento é processado, ele é removido da lista de eventos e os novos eventos gerados a partir deste são criados e adicionados na lista, ordenada pelos tempos em que cada evento ocorre.

Todos os parâmetros, descritos na seção 1.7 são passados para o simulador em sua inicialização.

## 1.2 Estruturas internas utilizadas

Para viabilizar a implementação da ideia geral apresentada acima, dividimos o simulador em alguns módulos, abaixo estão explicitados os mais importantes:

### Módulos utilitários:

- Estimator: Módulo que possui métodos para retornar os estimadores de média, variância e calcula intervalos de confiança.
- Dist: Módulo com o método que retorna os tempos aleatórios de chegada de uma distribuição exponencial.

### Classes:

- Client: classe que representa um cliente que entra no sistema. Possui seus tempos de entrada e saída da fila, tempo no servidor e cor.
- EventHeap: classe que representa a lista de eventos que é processada durante uma rodada de simulação.
- Simulator: classe que implementa a lógica principal do simulador, processa as rodadas tratando os eventos e as chegadas dos clientes. E calcula as estimativas das variáveis aleatórias.
- Analytic: classe que serve para calcular os resultados de forma analítica.

## 1.3 Linguagem de Programação

Para a codificação do simulador foi utilizada a linguagem de programação Python, versão 2.5.5.

## 1.4 Geração de variáveis aleatórias

A linguagem Python utiliza o gerador de números aleatórios "Mersenne Twister", um dos métodos mais extensivamente testados existentes.

O método garante que a sequência de números gerados pela chamada `random()` só se repetirá em um período de  $2^{19937} - 1$ . Como o período é bem extenso, não precisamos nos preocupar com redefinir seeds que gerassem sequências sobrepostas.

A semente inicial utilizada pelo gerador, por default, é o timestamp corrente no momento do import do módulo `random`.

## 1.5 Métodos utilizados

Foi utilizado o método replicativo para a simulação.

## 1.6 Implementação do conceito de cores

O conceito de cores foi implementando adicionando o atributo “color” no objeto Client, que possui 2 valores: TRANSIENT ou EQUILIBRIUM. O número de clientes que representam a fase transiente são associados à cor TRANSIENT e os outros clientes são associados à cor EQUILIBRIUM.

Ao final da rodada de simulação os clientes que possuem a cor TRANSIENT são descartados do cálculo dos estimadores.

## 1.7 Escolha dos parâmetros

Ao iniciar o simulador, são executados em sequência todas as simulações necessárias para obtermos todos os dados requeridos para ambas as políticas de atendimento com os parâmetros:

- $\rho = 0.2$  - # de clientes na frase transiente = 30000
- $\rho = 0.4$  - # de clientes na frase transiente = 40000
- $\rho = 0.6$  - # de clientes na frase transiente = 80000
- $\rho = 0.8$  - # de clientes na frase transiente = 400000
- $\rho = 0.9$  - # de clientes na frase transiente = 500000

A escolha do número de clientes da fase transiente para cada utilização foi estimada de acordo com o que é exposto no capítulo 3.

O número de clientes processados a cada rodada é um parâmetro de entrada para o simulador. Para o cálculo dos resultados foram utilizados apenas os dados de 100000 clientes, sem contar os presentes na fase transiente.

## 1.8 Máquina utilizada

As configurações da máquina utilizada para executar a simulação e os tempos de cada experimento são mostrados abaixo:

### **Configurações:**

- Processador: Intel Core Duo 2 GHz
- Memória: 2GB DDR 2 667Mhz
- Sistema Operacional: MAC OS X 10.5.8 (Leopard)

### **Duração dos experimentos:**

- $\rho = 0.2$  - F.C.F.S : 24.88s.
- $\rho = 0.2$  - L.C.F.S : 19.15s.
- $\rho = 0.4$  - F.C.F.S : 68.93s.
- $\rho = 0.4$  - L.C.F.S : 27.58s.
- $\rho = 0.6$  - F.C.F.S : 120.42s.
- $\rho = 0.6$  - L.C.F.S : 27.97s.
- $\rho = 0.8$  - F.C.F.S : 627.45s.
- $\rho = 0.8$  - L.C.F.S : 1037.33s.
- $\rho = 0.9$  - F.C.F.S : 3403.95s.
- $\rho = 0.9$  - L.C.F.S : 4732.28s.

## Capítulo 2

### Teste de Correção

Nesta seção você descreverá os testes de correção que foram efetuados para garantir o pleno funcionamento do simulador. Você deve demonstrar que o seu programa está simulando exatamente e com correção o esquema proposto. As fórmulas analíticas não podem ser utilizadas para garantir a correção. Servem apenas de orientação, pois na maioria das vezes partimos para a simulação exatamente por não termos os resultados analíticos. Procure rodar o simulador com cenários determinísticos com estatística conhecida, demonstrando que o programa está correto. Você deverá anexar comentários sobre a boa qualidade dos intervalos de confiança obtidos e como os valores exatos se encaixam nestes intervalos, para os diversos valores de  $r$ .

## Capítulo 3

# Estimativa da fase transiente

Os valores usados para a fase transiente de cada experimento foram estimados desenhando gráficos mostrando a geração dos valores da variância do tempo de espera na fila 2 ( $V(W_2)$ ) em 5 rodadas.

Esse estimador foi usado pois ele é o que converge mais lentamente para o intervalo de confiança válido. Este fato foi comprovado executando o simulador um número considerável de vezes para cada tipo de utilização do servidor.

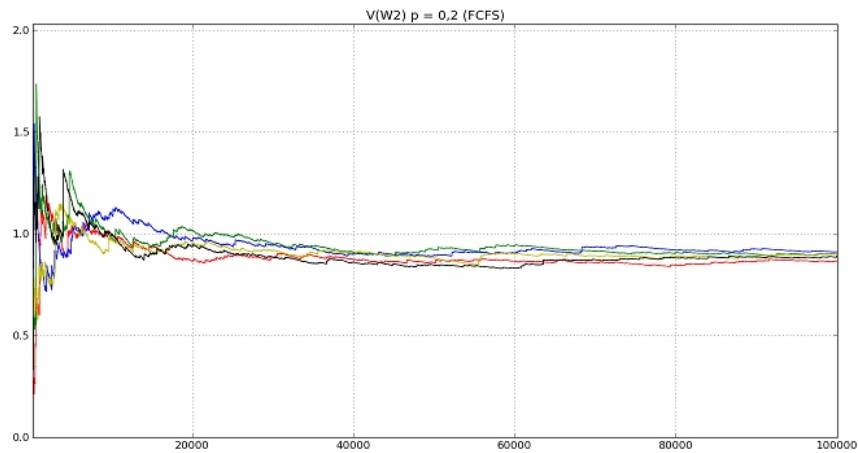
Com os gráficos gerados foi possível ter uma boa noção de que em que ponto o simulador começa a entrar na fase de equilíbrio. Eles são mostrados na seção 3.2.

### 3.1 No simulador:

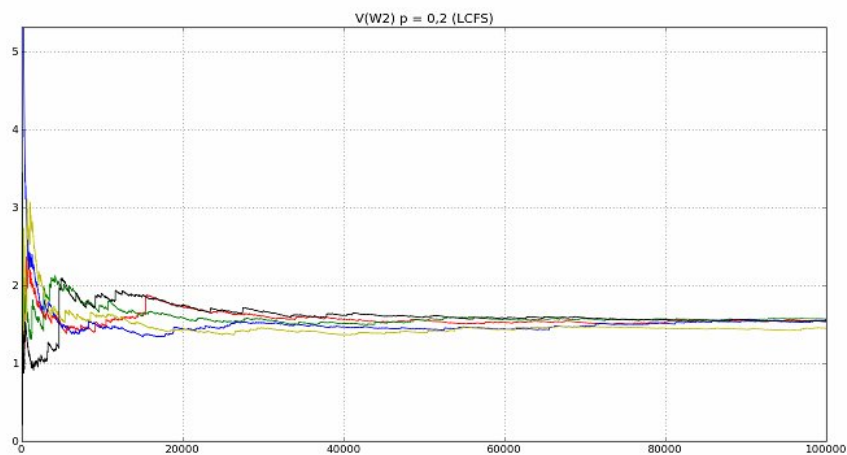
Dentro do simulador esses valores de fase transiente para cada experimento são dados em uma lista junto com os valores das taxas de entrada.

### 3.2 Gráficos:



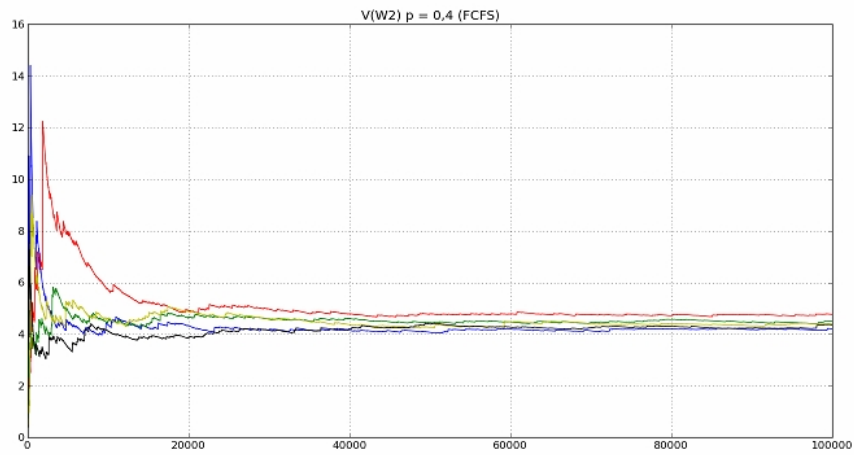


(a) First Come First Served

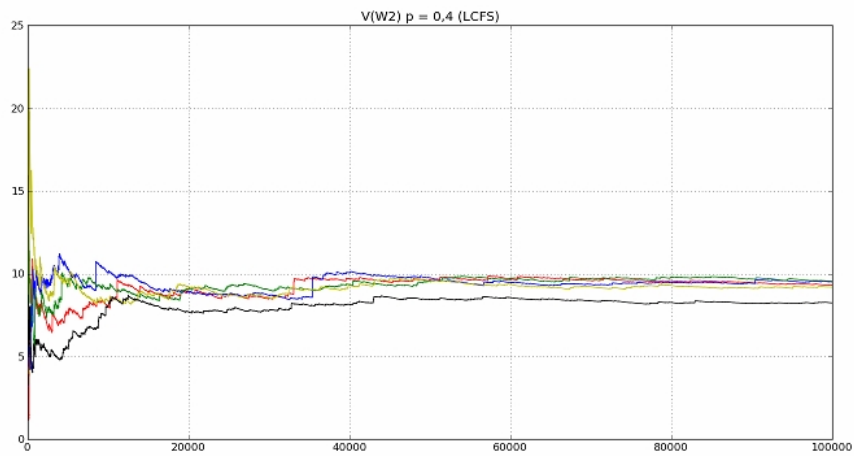


(b) Last Come First Served

Figura 3.1:  $V(W2)$  para  $\rho = 0.2$ . Nesse caso identificamos que a fase de equilíbrio começa quando aproximadamente 30.000 clientes já passaram pelo sistema.

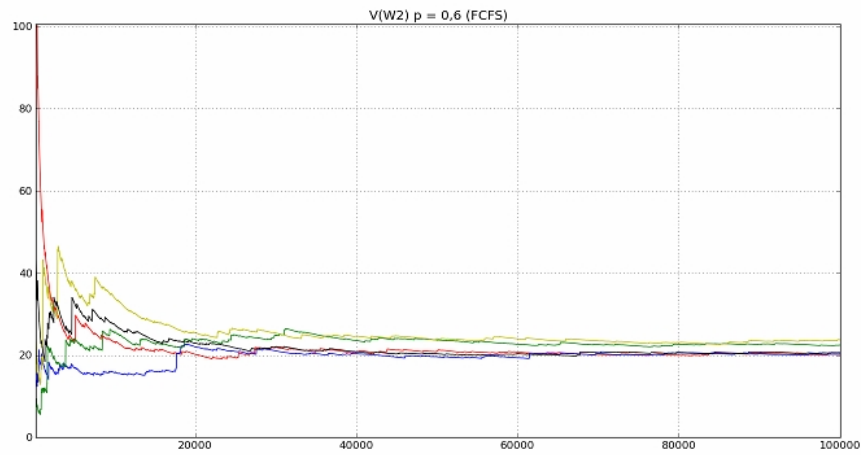


(a) First Come First Served

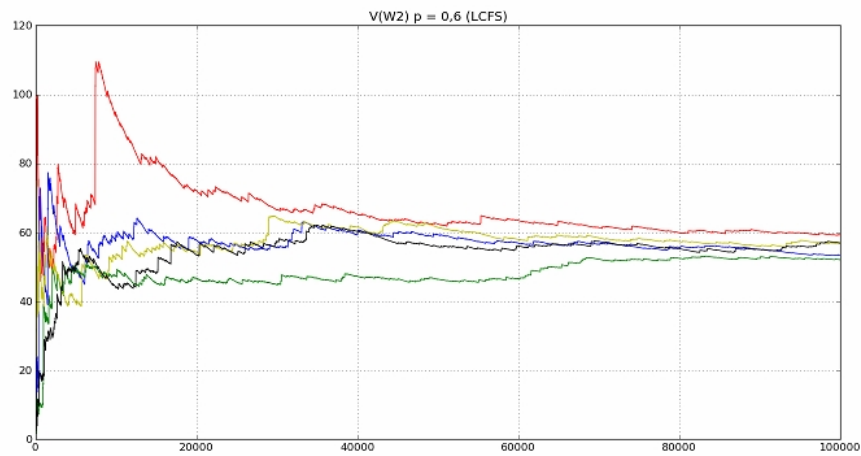


(b) Last Come First Served

Figura 3.2:  $V(W2)$  para  $\rho = 0.4$ . Nesse caso identificamos que a fase de equilíbrio começa quando aproximadamente 40.000 clientes já passaram pelo sistema.

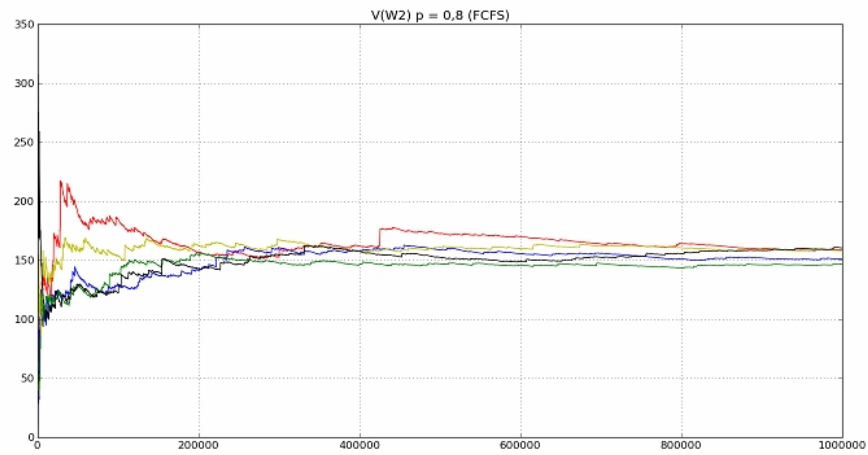


(a) First Come First Served

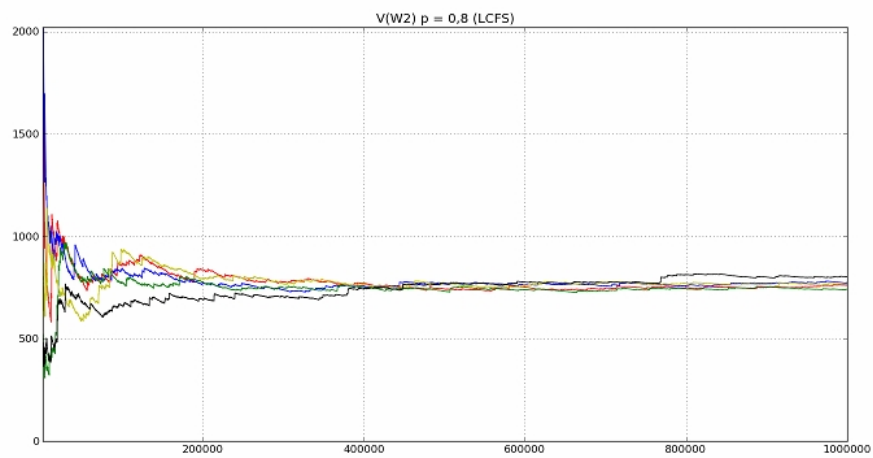


(b) Last Come First Served

Figura 3.3:  $V(W2)$  para  $\rho = 0.6$ . Nesse caso identificamos que a fase de equilíbrio começa quando aproximadamente 80.000 clientes já passaram pelo sistema.

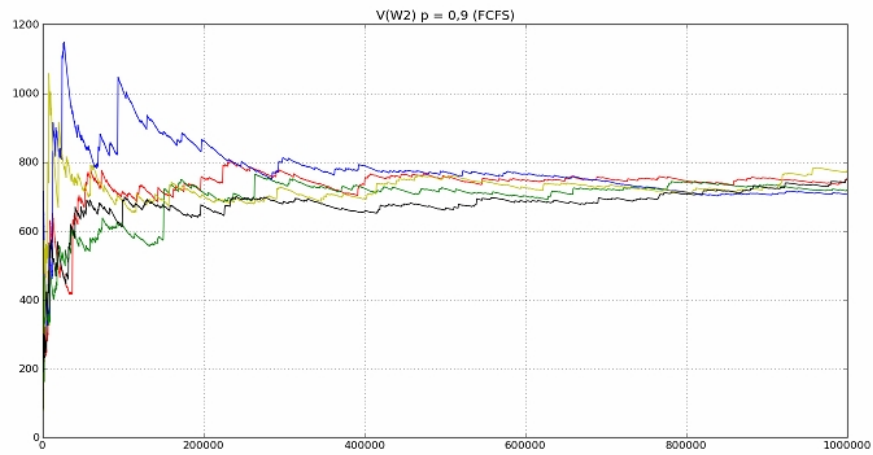


(a) First Come First Served

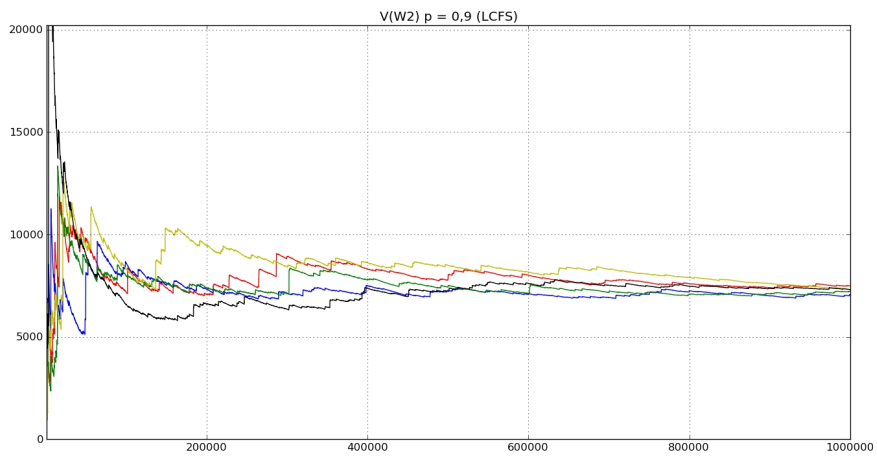


(b) Last Come First Served

Figura 3.4:  $V(W2)$  para  $\rho = 0.8$ . Nesse caso identificamos que a fase de equilíbrio começa quando aproximadamente 400.000 clientes já passaram pelo sistema.



(a) First Come First Served



(b) Last Come First Served

Figura 3.5:  $V(W2)$  para  $\rho = 0.9$ . Nesse caso identificamos que a fase de equilíbrio começa quando aproximadamente 500.000 clientes já passaram pelo sistema.

# Capítulo 4

## Resultados

Neste capítulo apresentamos os resultados obtidos na seção 4.1 e os comentários a respeito deles na seção 4.2.

### 4.1 Tabelas

Os resultados gerados pelo simulador são mostrados nas duas tabelas abaixo. O formato delas segue o seguinte padrão:

- Cada coluna representa um tipo de resultado (Tempo de espera na fila 1, etc.).
- Cada linha representa uma forma de utilização do servidor diferente.
- Cada célula contém respectivamente o valor analítico do resultado, o valor estimado pelo simulador e o tamanho do intervalo de confiança em % do valor estimado.

### 4.2 Comentários

Tabela com os resultados para a política de atendimento F.C.F.S										
uti.	E[N1]	E[N2]	E[T1]	E[T2]	E[Nq1]	E[Nq2]	E[W1]	E[W2]	V(W1)	V(W2)
<b>0.2</b>	0.12222	0.13056	1.22222	1.30556	0.02222	0.03056	0.22222	0.30556	0.44444	X
	0.12235	0.13034	1.22245	1.30357	0.02232	0.03048	0.22284	0.30408	0.45328	0.89774
	1.94%	2.64%	1.55%	2.31%	3.5%	5.98%	4.11%	6.5%	6.73%	9.38%
<b>0.4</b>	0.3	0.4	1.5	2.0	0.1	0.2	0.5	1.0	1.0	X
	0.29969	0.39933	1.49965	2.00103	0.09962	0.19957	0.49885	1.00104	0.98963	4.52897
	0.73%	1.63%	0.58%	1.87%	1.55%	2.9%	1.52%	3.46%	2.85%	8.97%
<b>0.6</b>	0.55714	1.13571	1.85714	3.78571	0.25714	0.83571	0.85714	2.78571	1.71429	X
	0.55845	1.13964	1.86446	3.81291	0.25813	0.83965	0.86328	2.81424	1.73843	22.21279
	0.71%	1.6%	0.7%	2.51%	1.08%	2.12%	1.21%	3.39%	2.4%	9.74%
<b>0.8</b>	0.93333	4.13333	2.33333	10.33333	0.53333	3.73333	1.33333	9.33333	2.66667	X
	0.93344	4.12212	2.33201	10.29418	0.53334	3.72216	1.33238	9.29394	2.66799	150.76599
	0.38%	1.34%	0.53%	3.0%	0.54%	1.48%	0.9%	3.32%	2.13%	9.85%
<b>0.9</b>	1.18636	11.12727	2.63636	24.72727	0.73636	10.67727	1.63636	23.72727	3.27273	X
	1.18532	11.06688	2.63626	24.62149	0.73556	10.61707	1.63607	23.62118	3.27872	766.03746
	0.17%	1.24%	0.39%	3.0%	0.24%	1.29%	0.59%	3.12%	1.52%	9.99%

Figura 4.1: Tabela com os valores para a política de atendimento First Come First Served.

Tabela com os resultados para a política de atendimento L.C.F.S										
uti.	E[N1]	E[N2]	E[T1]	E[T2]	E[Nq1]	E[Nq2]	E[W1]	E[W2]	V(W1)	V(W2)
<b>0.2</b>	0.12222	0.13056	1.22222	1.30556	0.02222	0.03056	0.22222	0.30556	0.49931	X
	0.12224	0.13054	1.22401	1.30979	0.02214	0.03063	0.22194	0.30905	0.4945	1.54658
	1.43%	2.15%	0.82%	0.93%	2.76%	4.97%	0.79%	3.69%	6.33%	8.55%
<b>0.4</b>	0.3	0.4	1.5	2.0	0.1	0.2	0.5	1.0	1.3125	X
	0.2998	0.40095	1.49855	1.99848	0.09978	0.20087	0.49807	0.99834	1.32128	9.16262
	1.93%	1.89%	0.61%	0.85%	2.92%	3.67%	0.81%	2.35%	3.0%	7.34%
<b>0.6</b>	0.55714	1.13571	1.85714	3.78571	0.25714	0.83571	0.85714	2.78571	2.76385	X
	0.55606	1.12351	1.85354	3.76388	0.25627	0.82439	0.85329	2.76932	2.77179	59.26071
	0.44%	1.91%	1.02%	1.92%	1.1%	2.21%	1.15%	2.64%	8.99%	5.89%
<b>0.8</b>	0.93333	4.13333	2.33333	10.33333	0.53333	3.73333	1.33333	9.33333	5.62963	X
	0.93431	4.14681	2.33523	10.44639	0.5341	3.74652	1.33541	9.44607	5.61005	790.73029
	0.28%	1.18%	0.53%	2.4%	0.42%	1.29%	0.83%	2.64%	2.67%	9.96%
<b>0.9</b>	1.18636	11.12727	2.63636	24.72727	0.73636	10.67727	1.63636	23.72727	8.14125	X
	1.18593	11.11241	2.63293	24.5574	0.73595	10.66243	1.63331	23.55766	8.12877	7429.88892
	0.15%	0.96%	0.29%	2.18%	0.21%	0.99%	0.43%	2.27%	1.55%	9.95%

Figura 4.2: Tabela com os valores para a política de atendimento First Come First Served.



## Capítulo 5

# Otimização

Para cada valor de utilização e para cada cenário, obtenha os resultados otimizados, isto é, considerando sempre o número de eventos de partida computados, procure determinar  $\text{FATOR MÍNIMO (disciplina)} = \text{número de rodadas} \times \text{tamanho de rodada (número de partidas)} + \text{fases transientes (número de partidas desprezadas)}$ , que satisfaz no seu simulador aos requisitos do tamanho do intervalo de simulação, independente do valor de utilização. Este fator mínimo deve ser independente da semente e isso tem que ser demonstrado. Obviamente, este FATOR será obtido para o caos mais crítico, entre parâmetro e valor de utilização. Com seu simulador operando corretamente, a busca da combinação ótima dará um bônus extra ao grupo que conseguir os menores valores, dentro de uma margem de  $20 \times \text{MÍNIMO}$  e outro tenha conseguido 1.200.000, eles estarão tecnicamente empatados e ambos ganharão bônus (20

# Capítulo 6

## Conclusões

Coloque aqui seus comentários finais. Descreva dificuldades encontradas, as otimizações feitas, e outras conclusões que você tirou do trabalho. Comente o que poderia ser melhorado, como, por exemplo, o tempo de execução do seu programa. Adicione quaisquer comentários que você julgar relevante. Cada uma das seções terá sua avaliação. Portanto, não deixe de colocar nenhuma seção no seu relatório. Se você não incluir uma seção, deixe-a em branco, mas não altere a numeração. Recomendamos fortemente que isso não ocorra. Não deixe de ler o capítulo de simulação na apostila.

# Capítulo 7

## Implementação

Anexo - Listagem documentada do programa A documentação do programa fonte deverá ser feita com rigor, explicando cada sub-rotina ou passo da programação. Código fonte sem comentários não é aceitável. Mande a listagem do código fonte como um anexo ao relatório. Uma versão eletrônica do documento impresso deve ser disponibilizada. O Grupo deve apresentar um executável funcionando em PC Windows ou Linux Ubuntu. O relatório completo deve ser entregue impresso (e não em mídia eletrônica). O programa executável deverá ser enviado para [aguiar@nce.ufrj.br](mailto:aguiar@nce.ufrj.br). Se o Grupo usar alguma linguagem específica, ele deve compilar o ambiente e apresentar um executável que rode sem necessidade de instalação especial, em Windows 7 ou Ubuntu. O programa será testado com alguns valores particulares para averiguar sua correção e integridade.