Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Ciência da Computação

Simulação de filas

Avaliação e Desempenho - MAB 515
Prof. Daniel Menasché

Alunos: Bruno Ignácio
Cristiane Monteiro
Filipi Xavier
Pedro Asad
Thalles Rodrigues

Sumário

1. Introdução

Este trabalho tem o objetivo de consolidar os conhecimentos adquiridos na disciplina de Avaliação e Desempenho, em particular filas com prioridades (LCFS - Last-come, First-served; e FCFS - First-come, First-served) e re-entrada. São analisados todos o cenários explicitados na descrição do trabalho, dentre eles a fase transiente com e sem continuidade, o teste do sistema instável, bem como as conclusões e discussões dos resultados. Segue abaixo a imagem que representa o sistema de filas que será analisado durante o trabalho, recebendo as variações de seus parâmetros junto de comparações das mesmas.

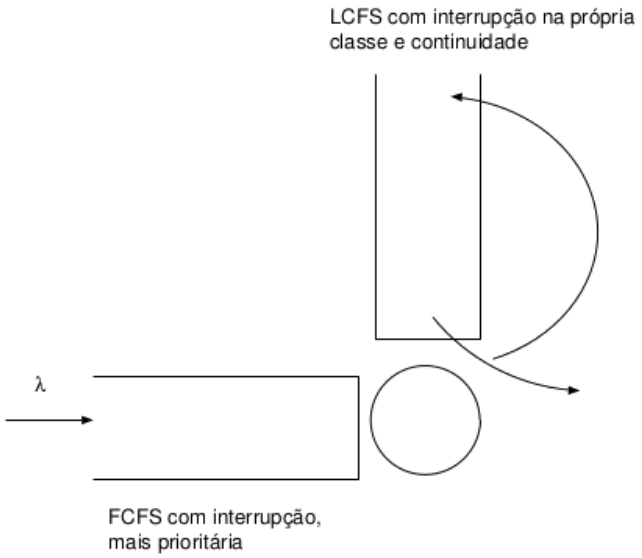


Figura 1: 1 servidor, com re-entrada.

2. Desafios encontrados e soluções

2.1 Determinação da fase estacionária

Algumas estratégias foram testadas para descobrir qual apresentaria melhor resultado. A que apresentou melhor resultado quanto a estabilidade do sistema foi a que verificava se a cada 1000 iterações os 5 estados que tiveram mais ” hits “ (estado i recebe um hit a cada iteração em que o sistema se encontra no estado i) eram os mesmos, na mesma ordem. Somente considerando o sistema após a ocorrencia da primeira igualdade de estados consecutivos. Após a primeira ocorrência desse evento essa ordem permanece praticamente inalterada, e os resultados continuam sem muitas flutuações.

Uma estratégia empregada também foi a verificação do conjunto dos 5 estados que permaneceram mais tempo a cada mil iterações. Uma tentativa foi verificar se o conjunto era mantido de uma iteração para a outra (o tamanho desse conjunto obedeceu a escolhas percentuais e também a escolhas determinísticas), que não produziu resultados expressivos. A outra estratégia, assim como a adotada, verificava se a ordem deles se mantinha inalterada, essa também não produziu bons resultados.

2.2 Resolução analítica dos resultados gerados

A análise do tempo dos clientes na segunda fila para o caso 2 se mostrou mais complexa que as outras análises. A necessidade de tratar os casos em que usuários à frente do usuário padrão (recém interrompido e com menor prioridade que todos) são interrompidos, e consequentemente passam a ter menos prioridade que o usuário padrão, tornou a análise mais complexa. Esse caso e a solução obtida são discutidos mais à frente na seção Derivação dos Resultados Analíticos.

Em geral as dificuldades não se encontraram na interpretação do problema, ou algebrismos, mas sim na própria idealização das fórmulas.

Dificuldades foram encontradas também em gerar os resultados analíticos para a variância do tempo de permanência de usuários no sistema. O que impediu que esses resultados fossem obtidos foi o dificuldade em concluir as soluções analíticas do caso 2 para a fila 2 com respeito ao tempo médio de permanência dos clientes no sistema.

O grupo acabou por ficar sem tempo de gerar os resultados da variância, que obviamente dependiam do preciso cálculo dos primeiro e segundo momentos do tempo de permanência no sistema.

Foi difícil determinar os valores da média e da variância da normal truncada.

Sabemos que o truncamento afeta a distribuição, mas não sabíamos calcular exatamente os parâmetros u e <sigma> referentes à nova distribuição modificada.

Para contornar o problema, utilizamos valores de media e desvio padrão calculados experimentalmente (em cima de amostragens da normal truncada).

Isso fez com que nossos resultados para a fila1 com normal truncada não sejam totalmente analíticos.

3. Decisões de projeto

3.1 Frameworks e ferramentas utilizadas

O desenvolvimento do simulador foi feito em linguagem java utilizando o framework eclipse. O tratamento dos dados gerados foi feito através de scripts awk e bash. Os gráficos foram feitos utilizando shell script e gnuplot.

3.2 Variáveis de estado

Foram considerados como itens da classe de estado do simulador os seguintes itens:

- As filas de clientes e clientes prioritários;
- O número de clientes no sistema;
- As taxas das distribuições;
- Uma variável que diz se o sistema saiu do período transiente;
- O tempo atual do sistema;
- O número máximo de clientes que o sistema quer analisar (retirando a fase tratada como transiente);
- O número total de clientes que o sistema já analisou (retirando a fase tratada como transiente).

3.3 Eventos

A arquitetura dividiu os Eventos nos seguintes tipos:

- Evento de entrada;
- Evento de saída;
- Evento de chegada prioritária;
- Evento de saída prioritária.

No início o simulador dá partida criando o primeiro evento de chegada prioritária. Quando o evento de chegada prioritária ocorre ele agenda um evento de saída prioritária e agenda a próxima chegada prioritária. Quando a saída prioritária ocorre ele agenda a chegada do cliente comum na fila imediatamente. Essa chegada comum agenda a sua saída comum.

4. Depuração

O processo de depuração foi feito através da análise de dados com informações adicionais impressas pelo simulador. Foi feita uma análise de todos os casos que pudemos imaginar de interrupções analisando cuidadosamente os tempos de chegada, tempo de saída do atendimento prioritário, tempo de chegada do atendimento comum, tempo de saída, trabalho restante, trabalho acumulado.

5. Resultados

5.1 Cenários

Caso	λ	Serviço 1	Serviço 2	E[X]	Lim. Inf. Int. Conf.	Lim. Sup. Int. Conf.	Analítico
1	1	determinístico, μ_1 = 2	exponencial, μ_2 = 4	4,007	3,994	4,019	4,000
1	1.5	determinístico, μ_1 = 10	exponencial, μ_2 = 4	0,755	0,754	0,758	0,755
1	2	determinístico, μ_1 = 10	exponencial, μ_2 = 4	1,216	1,211	1,220	1,208
2	1	determinístico, μ_1 = 2	exponencial, μ_2 = 4	3,982	3,969	3,995	4,484
2	1.5	determinístico, μ_1 = 10	exponencial, μ_2 = 4	0,757	0,755	0,759	0,500
2	2	determinístico, μ_1 = 10	exponencial, μ_2 = 4	1,209	1,205	1,214	2,576
1	0.5	normal truncada, μ_1 = 10, σ_1= 1	exponencial, μ_2 = 4	3,378	3,370	3,387	3,098
2	0.5	normal truncada, μ_1 = 10, σ_1= 1	exponencial, μ_2 = 4	3,397	3,395	3,399	1,39

Tabela 1: Esperança do tempo no sistema.

Caso	λ	Serviço 1	Serviço 2	V[X]	Lim. Inf. Int. Conf.	Lim. Sup. Int. Conf.
1	1	determinístico, $\mu_1 = 2$	exponencial, $\mu_2 = 4$	42,32		
1	1.5	determinístico, $\mu_1 = 10$	exponencial, $\mu_2 = 4$	1,210		
1	2	determinístico, $\mu_1 = 10$	exponencial, $\mu_2 = 4$	5,779		
2	1	determinístico, $\mu_1 = 2$	exponencial, $\mu_2 = 4$	42,563		
2	1.5	determinístico, $\mu_1 = 10$	exponencial, $\mu_2 = 4$	1,277		
2	2	determinístico, $\mu_1 = 10$	exponencial, $\mu_2 = 4$	5,588		
1	0.5	normal truncada, $\mu_1 = 10$, $\sigma_1 = 1$	exponencial, $\mu_2 = 4$	18,550		
2	0.5	normal truncada, $\mu_1 = 10$, $\sigma_1 = 1$	exponencial, $\mu_2 = 4$	18,887		

Tabela 2: Variância do tempo no sistema.

5.2 Intervalo de confiança

Escolhemos fixar o número de clientes que passa pelo sistema e, a partir de uma estimativa do desvio padrão dada pelo desvio padrão amostral, obter um intervalo de confiança de 95% de certeza para o tempo médio de permanência. Preferimos esta abordagem em vez de fixar o intervalo e então variar o número de amostras porque estimamos o desvio padrão a partir das amostras e isso teria o potencial de alterar o número de amostras requerido a cada nova medida de tempo de permanência.

Após algumas rodadas de teste, já pudemos determinar heuristicamente alguns valores aproximados para a média e a variância do tempo de permanência. Sabendo desses valores, escolhemos um número de amostras igual a 1000000, que nos possibilita, com uma confiabilidade de 95%, estimar a média com um erro máximo de 0,00196 desvios-padrão. Para

os valores de desvios-padrão obtidos na seção anterior, isto significa que o erro máximo da média se restringia à segunda casa decimal, como também pode ser visto na tabela.

A nossa margem de confiança é tS e sua obtenção é demonstrada abaixo:

$\overline{X}$ - Média amostral
$n = 1.000.000$ - Número de amostras
μ - Média
S - Desvio padrão amostral
α - Tolerância a erro
$\phi()$ - Função de distribuição acumulada da normal padrão

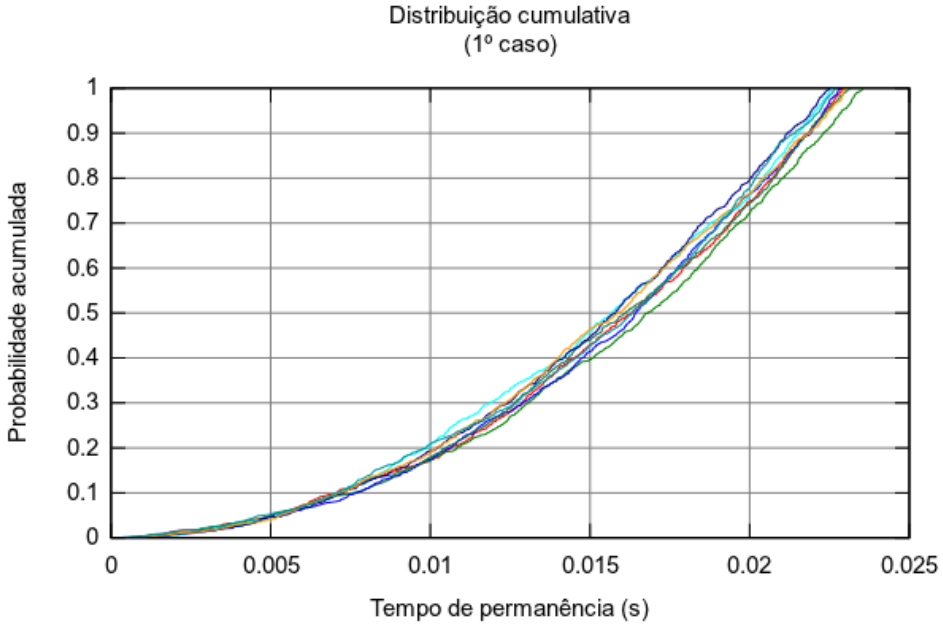
$$P\{-tS < x - \mu < tS\} = 1 - \alpha$$
$$P\left\{-t\sqrt{n} < \frac{x-\mu}{S}\sqrt{n} < t\sqrt{n}\right\} = 1 - \alpha$$
$$\phi(t\sqrt{n}) = 1 - \frac{\alpha}{2}$$
$$t\sqrt{n} = 1{,}96$$
$$t = 0{,}00196$$

5.3 CDF do tempo no sistema

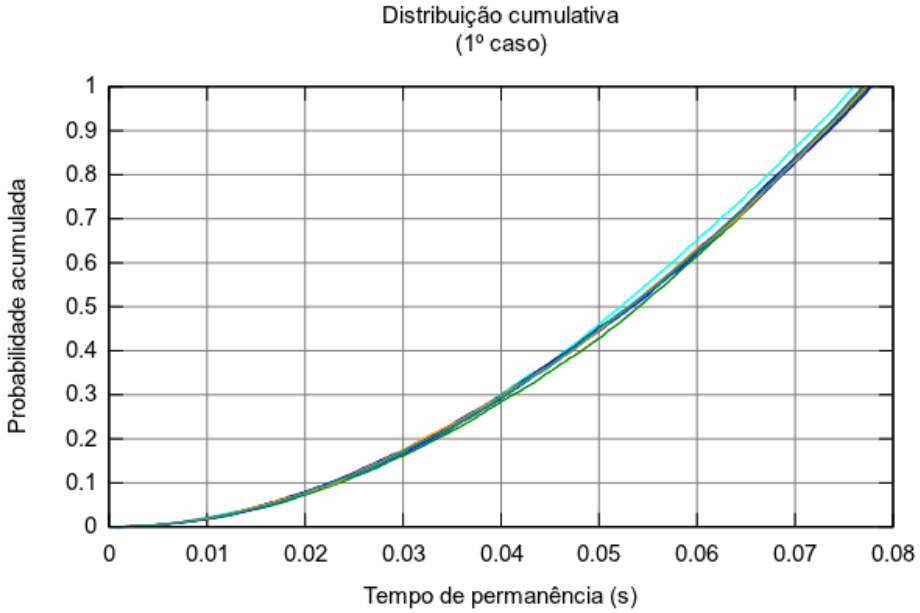
Para obter o gráfico da distribuição acumulada do tempo de permanência no sistema utilizamos scripts externos para tratar a saída crua do simulador. Tomamos o arquivo que continha os tempos de permanência para cada cliente e o ordenamos. A seguir, adicionamos uma segunda coluna correspondendo à soma acumulada dos elementos da primeira coluna dividida pela soma de todos os tempos de permanência, de maneira que o domínio da segunda coluna fosse [0,1].

Os quatro gráficos a seguir representam os dados coletados de oito rodadas da simulação com os parâmetros da primeira linha da tabela. Em cada uma das oito rodadas, simulamos o trânsito de um milhão de clientes a partir do estado estacionário. Porém, para gerar cada um dos gráficos, truncamos os conjuntos de dados, respectivamente, em mil, dez mil, cem mil e um milhão de clientes. Vale ressaltar que somente no primeiro e segundo gráficos é possível visualizar as oito curvas, visto que nos outros, as curvas se sobrepõem bastante.

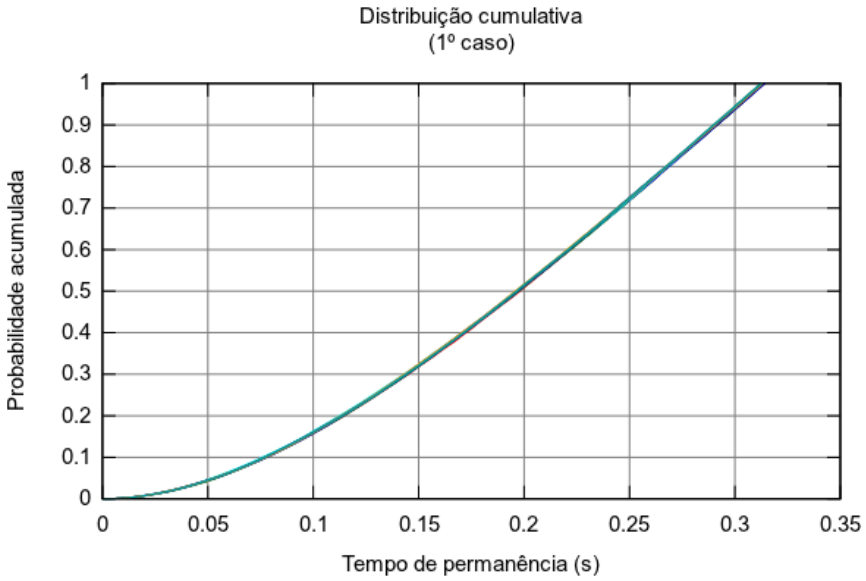
5.3.1 CDF da linha 1 da tabela 1 para 1.000 clientes



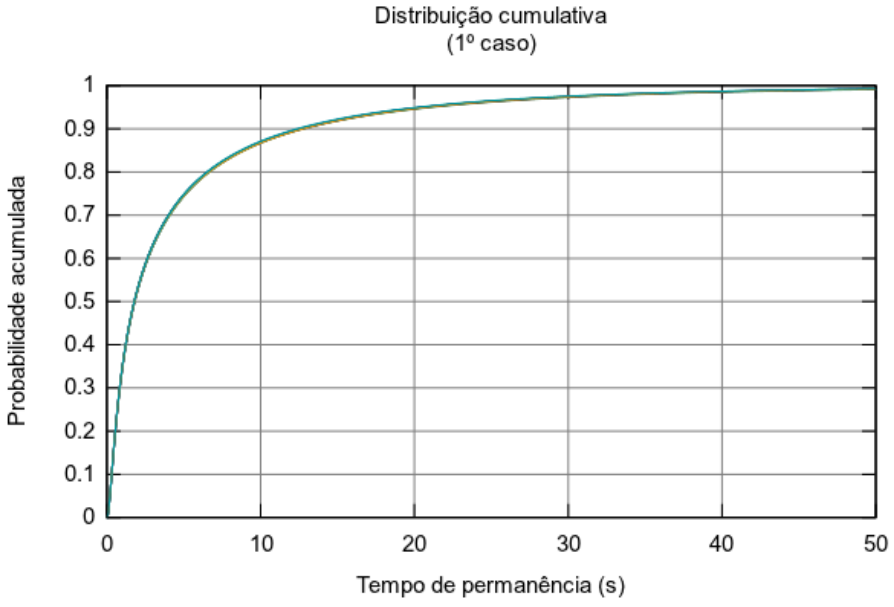
5.3.2 CDF da linha 1 da tabela 1 para 10.000 clientes



5.3.3 CDF da linha 1 da tabela 1 para 100.000 clientes



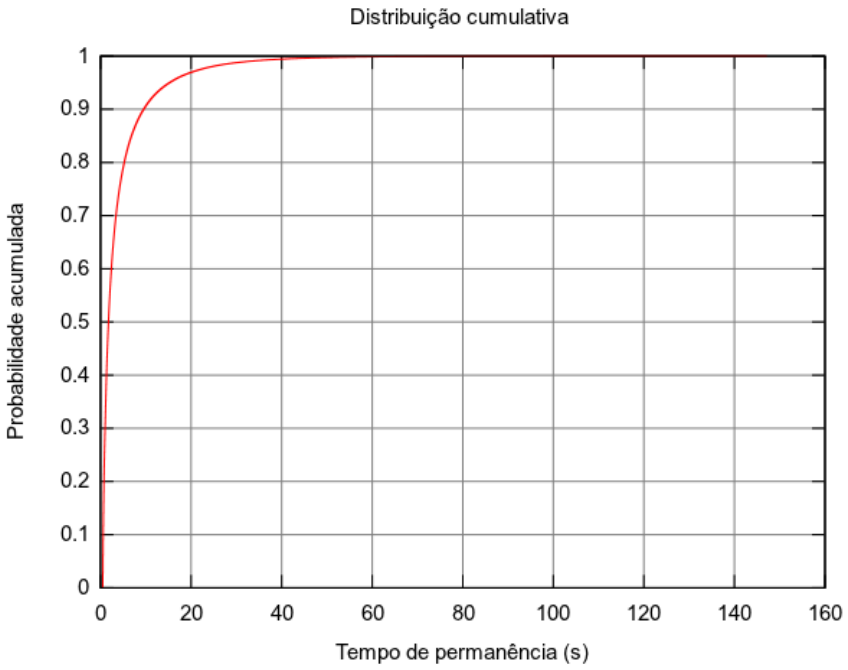
5.3.4 CDF da linha 1 da tabela 1 para 1.000.000 clientes

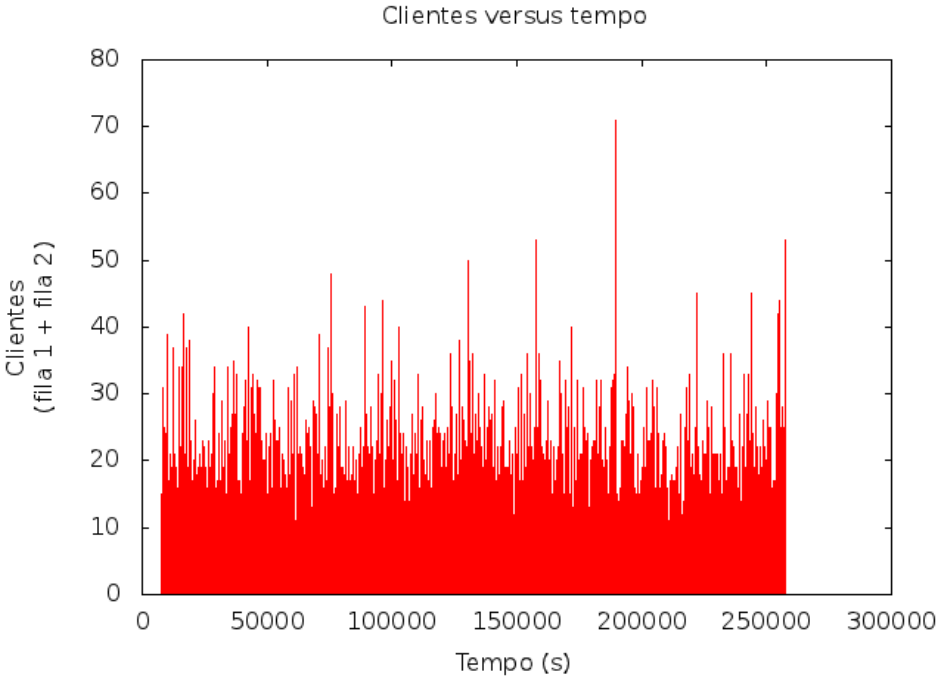


Os dois primeiros gráficos apresentam uma subida mais suave, pois representam um estado menos estável do sistema. O terceiro já mostra tendência a um comportamento assintótico à medida que o tempo de permanência aumenta. O quarto gráfico já representa o que consideramos como um retrato preciso do estado estacionário do sistema.

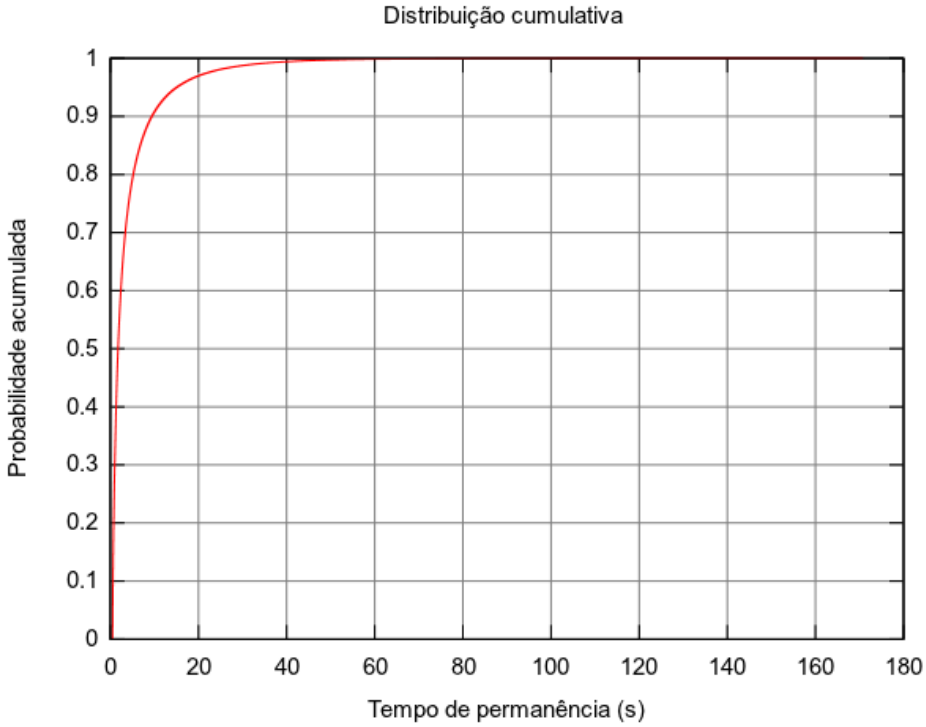
5.4 Fase Transiente

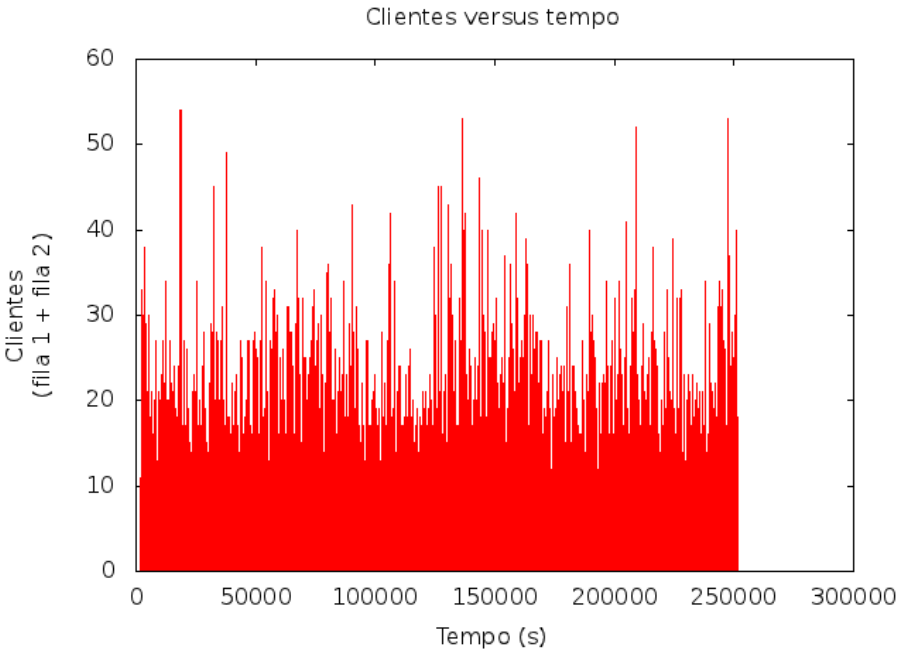
5.4.1 Caso com continuidade





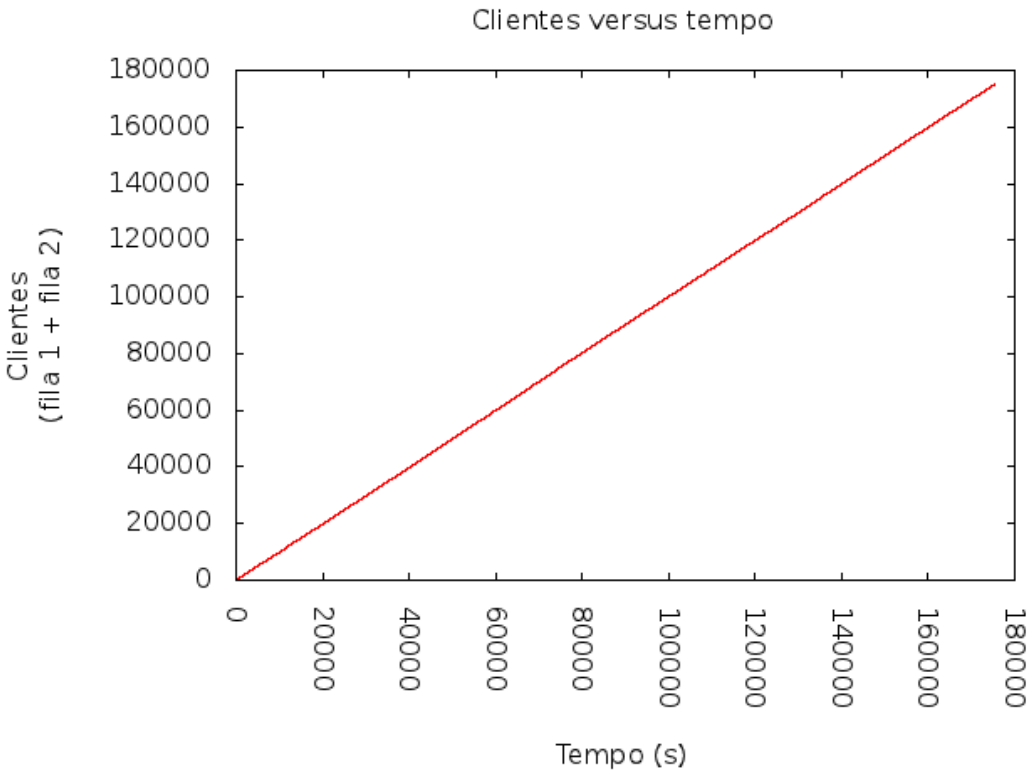
5.4.2 Caso sem continuidade





O caso com continuidade demorou mais a entrar no modo estacionário.

5.5 Sistema Instável



6. Derivação dos resultados analíticos obtidos

Calculamos separadamente o tempo que um cliente passa no sistema como cliente do tipo1 (fila 1 + serviço 1) e o tempo que passa como cliente do tipo 2 (fila 2 + serviço 2).

O tempo total de um cliente no sistema inteiro é então calculado pela soma do tempo como cliente tipo 1 e tipo 2.

Para o cenário da fila 1 as métricas são simples, pois a fila 1 tem prioridade e direito a interrupção. Logo, do ponto de vista da fila 1, o comportamento é de uma M/G/1.

- Cenário da fila 1, serviço determinístico:

  - Média do tempo de serviço: $\frac{1}{\mu_1}$

  - Segundo momento X: E[X^2] = $\frac{1}{\mu_1^2}$

  - Média do tempo em fila: $\frac{\lambda\, E[X^2]}{2(1-\rho)} \rightarrow E[W] = \frac{\rho}{2\mu_1(1-\rho)}$

  - Média do tempo no sistema: $\frac{1}{\mu_1} + \frac{\rho}{2\mu_1(1-\rho)}$

  - Utilização média do servidor: $\lambda E[X] = \frac{\lambda}{\mu_1} = \rho$

  - Número médio de usuários na fila: $\lambda E[W] = \frac{\lambda\rho}{2\mu_1(1-\rho)} = \frac{\rho^2}{2(1-\rho)}$

  - Número médio de usuários no sistema: $\lambda(E[W] + \frac{1}{\mu_1})$

  - Taxa de saída de usuários: Poisson com taxa μ_1.

- Cenário da Fila 1, serviço exponencial:

  - Média do tempo de serviço: $\frac{1}{\mu_1}$

  - Esperança do X_r: $E[X_r] = E[X] = \frac{1}{\mu_1}$

- Média do tempo em fila: $\frac{\rho E[X_r]}{1-\rho} \rightarrow E[W] = \frac{\rho}{\mu_1(1-\rho)}$

- Média do tempo no sistema: $\frac{1}{\mu_1} + \frac{\rho}{\mu_1(1-\rho)}$

- Utilização média do servidor: $\lambda E[X] = \frac{\lambda}{\mu_1} = \rho$

- Número médio de usuários na fila: $\lambda E[W] = \frac{\lambda\rho}{\mu_1(1-\rho)} = \frac{\rho^2}{(1-\rho)}$

- Número médio de usuários no sistema: $\lambda(\frac{1}{\mu_1} + \frac{\rho}{\mu_1(1-\rho)})$

- Taxa de saída de usuários: Poisson com taxa μ_1.

- Cenário da fila 1, serviço normal truncada

  - Tivemos que pegar experimentalmente os valores de média e desvio padrão da normal truncada. Visto que truncar gera efeitos sobre as características da distribuição normal.

  - media do tempo de serviço: 1/μ

  - media do tempo na fila: $\overline{W} = \frac{\lambda E[X^2]}{2(1-\frac{\lambda}{\mu})} = \frac{\lambda E\left[\sigma^2 + \frac{1}{\mu^2}\right]}{2(1-\frac{\lambda}{\mu})}$

  - média do tempo no sistema: $\overline{W} + \frac{1}{\mu}$

  - utilização media do servidor: $\frac{\lambda}{\mu} = \rho$

  - usuários na fila 1: (por little) $\lambda.\overline{W}$

  - taxa media de saida: λ (para estado estacionario o numero de usuarios se conserva, logo entra o mesmo que sai)

- Cenário da fila 2, caso 1, serviço determinístico:

  - Média do tempo de serviço: $\frac{1}{\mu_2}$

  - Média do tempo em fila:

    Será o período ocupado disparado pelos tempos dos casos que possuem prioridade sobre o cliente mais o tempo do próprio cliente. Subtraindo o tempo de serviço do cliente.

    Para: $\lambda\overline{T_1}\,\overline{X_1}$ sendo os prioritários na fila 1 e $\lambda\overline{T_1}\,\overline{X_2}$ sendo os prioritários que foram para a fila 2, temos:

    $$\overline{T_{02}} = \lambda\overline{T_1}\,\overline{X_1} + \lambda\overline{T_1}\,\overline{X_2} + \overline{X_2}$$
    $$\overline{T_2} = \frac{\overline{T_{02}}}{1-(\rho_1+\rho_2)} \rightarrow \overline{W_2} = \overline{T_2} - \overline{X_2}$$

    Obs: $(\rho_1+\rho_2)$, pois é interrompido por clientes que ingressam na fila 1 ou 2.

  - Utilização média do servidor: $\lambda_2 E[X_2] = \lambda_2\frac{1}{\mu_2} = \rho_2$

  - Número médio de usuários em espera: $\lambda_2\overline{W_2} = \lambda\overline{W_2}$

  - Número médio de usuários no sistema: $\lambda_2\overline{T_2} = \lambda\overline{T_2}$

  - Taxa de saída de usuários: $\lambda_2 = \lambda$

- Cenário da fila 2, caso 1, serviço exponencial:

  - Média do tempo de serviço: $\frac{1}{\mu_2}$

  - Média do tempo em fila:

    Será o período ocupado disparado pelos tempos dos casos que possuem prioridade sobre o cliente mais o tempo do próprio cliente. Subtraindo o tempo de serviço do cliente.

    Para: $\lambda\overline{T_1}\,\overline{X_1}$ sendo os prioritários na fila 1 e $\lambda\overline{T_1}\,\overline{X_2}$ sendo os prioritários que foram para a fila 2, temos:

$$\overline{T_{02}} = \lambda \overline{T_1}\,\overline{X_1} + \lambda \overline{T_1}\,\overline{X_2} + \overline{X_2}$$
$$\overline{T_2} = \frac{\overline{T_{02}}}{1-(\rho_1+\rho_2)} \rightarrow \overline{W_2} = \overline{T_2} - \overline{X_2}$$

Obs: $(\rho_1+\rho_2)$, pois é interrompido por clientes que ingressam na fila 1 ou 2.

- Utilização média do servidor: $\lambda_2 E[X_2] = \lambda_2 \frac{1}{\mu_2} = \rho_2$

- Número médio de usuários em espera: $\lambda_2 \overline{W_2} = \lambda \overline{W_2}$

- Número médio de usuários no sistema: $\lambda_2 \overline{T_2} = \lambda \overline{T_2}$

- Taxa de saída de usuários: $\lambda_2 = \lambda$

- Cenário da fila 2, caso 2, serviço determinístico:

- Descrição

Cada vez que um usuário é interrompido ele passa a ter prioridade menor que todos na fila 2. O tempo que este usuário demora para voltar a ter acesso ao servidor depende das chegadas de usuários prioritários na fila 1 e de usuários da fila 1 que são atendidos.

Um usuário (vindo da fila 2) é interrompido toda vez que um usuário prioritário (fila 1) chega ao sistema (taxa λ). Isso pode ocorrer durante o tempo de serviço do usuário ($\overline{X_2}$).

- Média do tempo de serviço: $\frac{1}{\mu_2}$

- Média do tempo na fila:
  - Para cada vez que é interrompido:
    $(\overline{N_{q2}}\,\overline{X_2} + \overline{N_{q1}}\,\overline{X_1} + \overline{N_{q1}}\,\overline{X_2} + \overline{X_1} + \overline{X_2}) = S_1$
  - Quantas vezes é interrompido: $(\lambda_1 \overline{X_2})$
  - Para descontar os usuários que serão interrompidos na minha frente:
    - $(\overline{N_{q1}}\,\overline{X_2} + \overline{N_{q2}}\,\overline{X_2})$. Pessoas que passarão pelo servidor vindo da fila 2;
    - $\overline{-X_{r2}}$. Residual dos interrompidos;
    - $\overline{X_2}$. O que interrompeu e passará pelo servidor vindo da fila 2;
    - λ_1. Taxa com que são interrompidos.
    - = $(\overline{N_{q1}}\,\overline{X_2} + \overline{N_{q2}}\,\overline{X_2} + X_2)\lambda_1(\overline{-X_{r2}}) = S_2$
  - $\overline{T_2} = \frac{S_1+S_2}{1-(\rho_1+\rho_2)}$

- Cenário da fila 2, caso 2, serviço exponencial:

  - Descrição

Cada vez que um usuário é interrompido ele passa a ter prioridade menor que todos na fila 2. O tempo que este usuário demora para voltar a ter acesso ao servidor depende das chegadas de usuários prioritários na fila 1 e de usuários da fila 1 que são atendidos.

Um usuário (vindo da fila 2) é interrompido toda vez que um usuário prioritário (fila 1) chega ao sistema (taxa λ). Isso pode ocorrer durante o tempo de serviço do usuário ($\overline{X_2}$).

- Média do tempo de serviço: $\frac{1}{\mu_2}$

- Média do tempo na fila:
  - Para cada vez que é interrompido:

$$(\overline{N_{q2}}\,\overline{X_2}+\overline{N_{q1}}\,\overline{X_1}+\overline{N_{q1}}\,\overline{X_2}+\overline{X_2})=S_1$$

  - Quantas vezes é interrompido: $(\lambda_1\overline{X_2})$
  - Para descontar os usuários que serão interrompidos na minha frente:
    - $(\overline{N_{q1}}\,\overline{X_2}+\overline{N_{q2}}\,\overline{X_2})$. Pessoas que passarão pelo servidor vindo da fila 2;
    - $\overline{-X_{r2}}$. Residual dos interrompidos;
    - $\overline{X_2}$. O que interrompeu e passará pelo servidor vindo da fila 2;
    - λ_1. Taxa com que são interrompidos.

    - = $(\overline{N_{q1}}\,\overline{X_2}+\overline{N_{q2}}\,\overline{X_2}+X_2)\lambda_1(\overline{-X_{r2}})=S_2$

$$\overline{T_2}=\frac{S_1+S_2}{1-(\rho_1+\rho_2)}\lambda_1$$

  -
  - S_2 é a equação que dispara um período ocupado negativo que está relacionado ao fato de que no caso 2 um cliente típico que aguarda na fila 2 após ser interrompido pode ser "beneficiado" se um cliente que está à frente dele for interrompido (perde prioridade).

7. Comparações entre resultados experimentais e analíticos

Caso 1:

Os resultados obtidos para o caso 1 parecem muito consistentes pois se aproximaram bastante daqueles obtidos por experimentos. Como já dito anteriormente, para este caso, determinar as fórmulas para o tempo de permanência de um usuário tanto na fila 1 quanto na fila 2 foram mais fáceis.

Caso 2:

No caso 2 os resultados obtidos analiticamente não parecem estar em total conformidade com o comportamento no “mundo real”. Acreditamos que o problema esteja nas fórmulas analíticas e não na simulação, pois a geração dos dados experimentais foi extensamente testada, enquanto que para a determinação das fórmulas, encontramos dificuldades e não pudemos contar com o contato com outros grupos para discutir ideias.

Como explicado anteriormente, não finalizamos satisfatoriamente a análise da fila 2 no caso2.

Isso fez com que não encontrássemos tempo e nem razão para tentar expandir as fórmulas em fórmulas de variância.
Entretanto, analisando puramente por "feeling" sabemos que ao truncar a exponencial, seu desvio padrão deve ser reduzido, e espera-se um impacto de redução de tempos em geral no sistema.

8. Referências

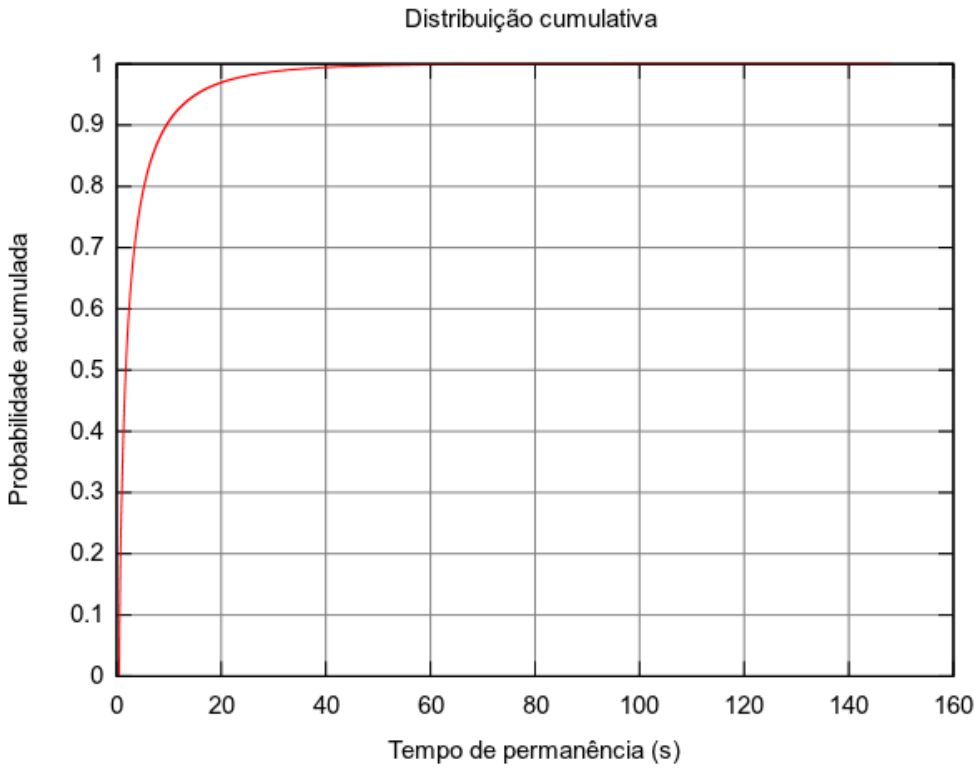
- Descrição do trabalho.
  < http://www.dcc.ufrj.br/~sadoc/ad20121/trabalhoutf8.pdf >
  Acesso em junho de 2012.

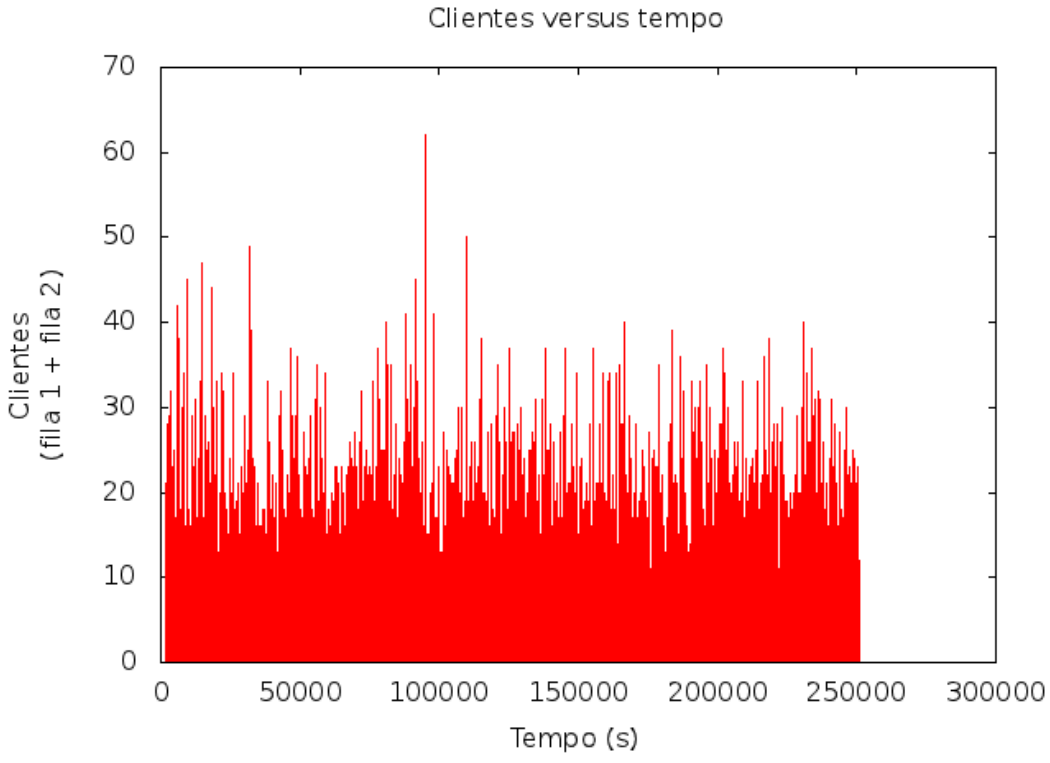
- Aulas de Avaliação de Desempenho, ministradas pelo professor Daniel Sadoc Menasché, em especial a aula 17 sobre filas com prioridades e redes de filas.
  < http://www.dcc.ufrj.br/~sadoc/ad20121/ >
  Acesso em junho de 2012.

- Apostila de Avaliação de Desempenho do professor Paulo Aguiar.
  < http://www.voip.nce.ufrj.br/cursos/images/files/mab515/ap-23-10-2011.pdf >
  Acesso em junho de 2012.

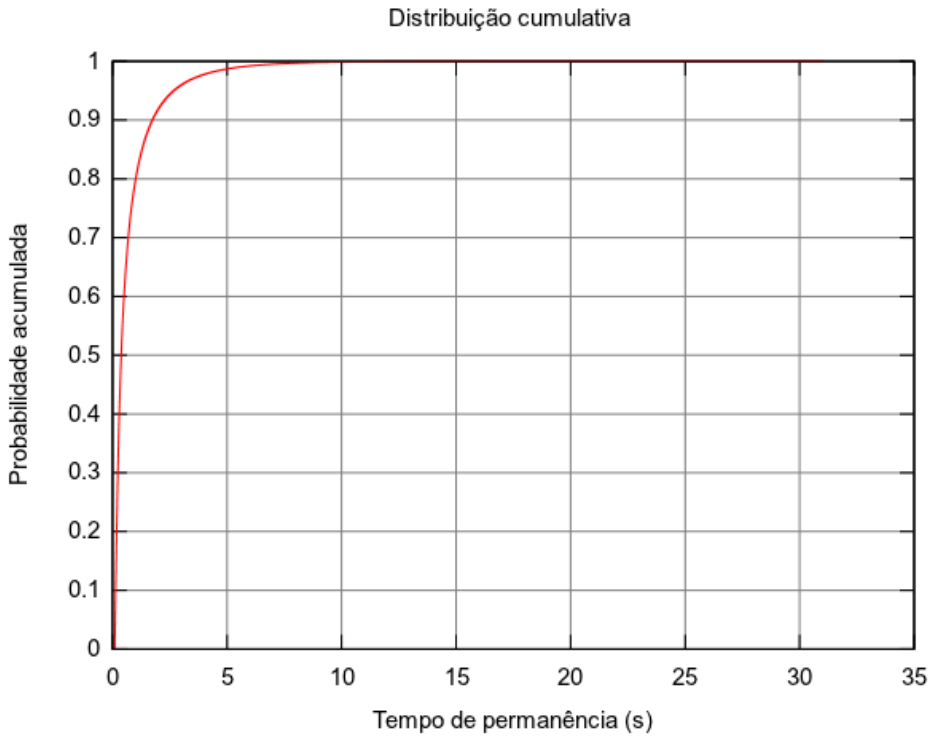
9. Anexos

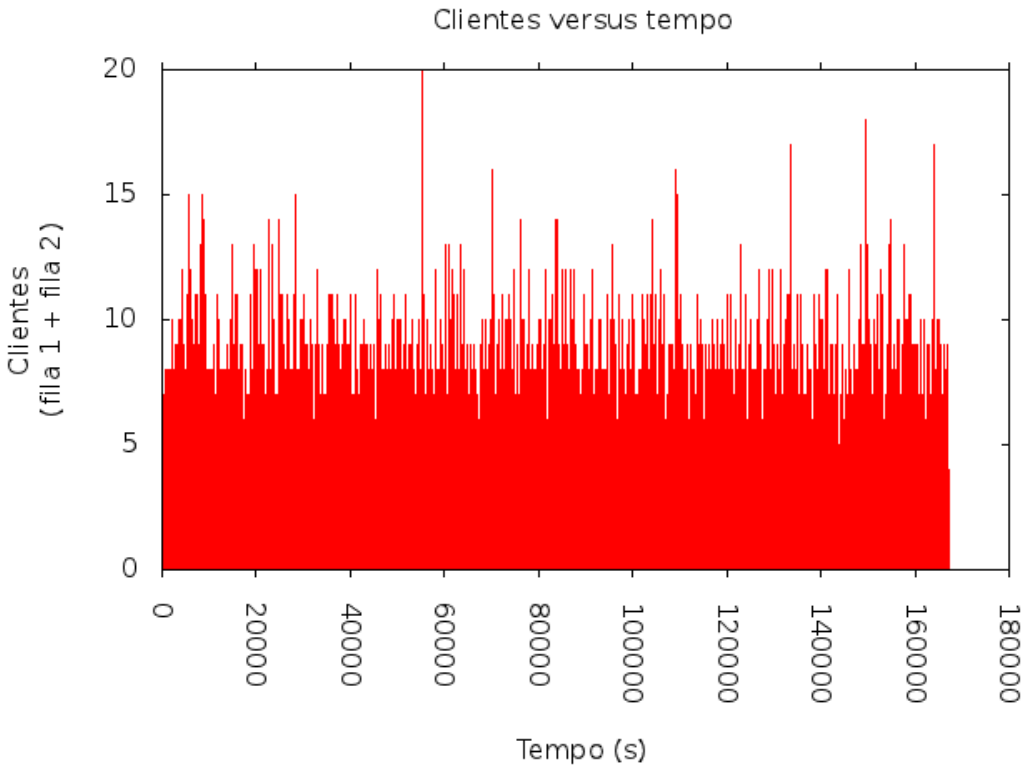
Gráficos da linha 1 da tabela 1



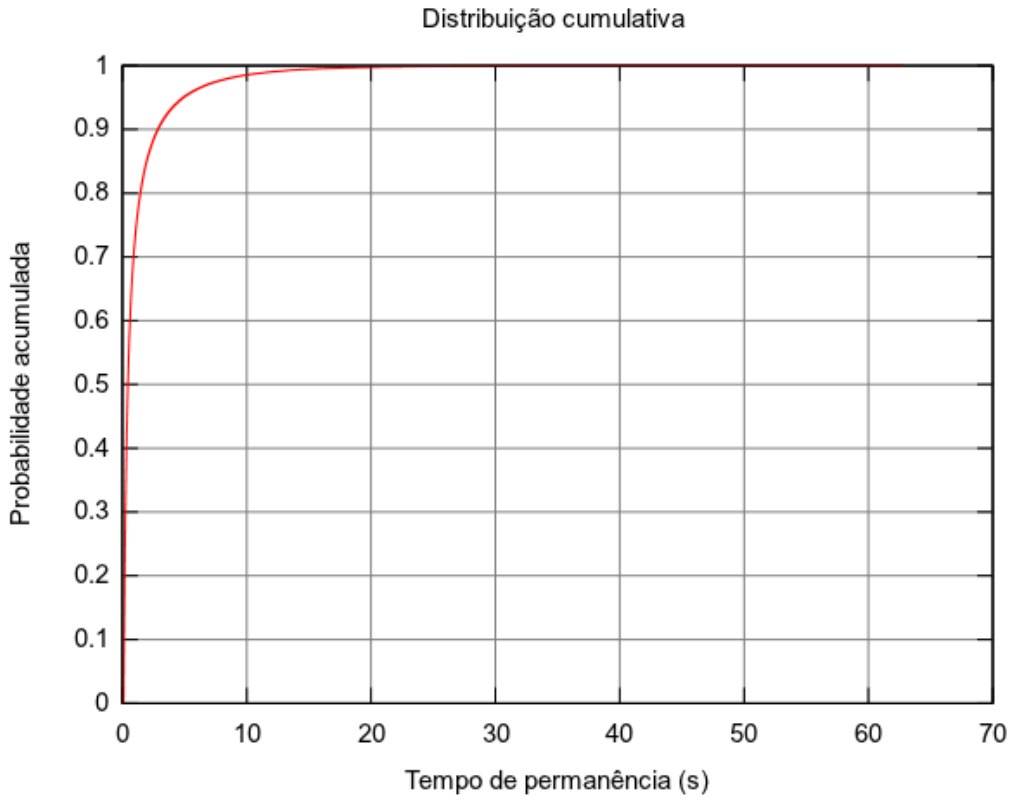


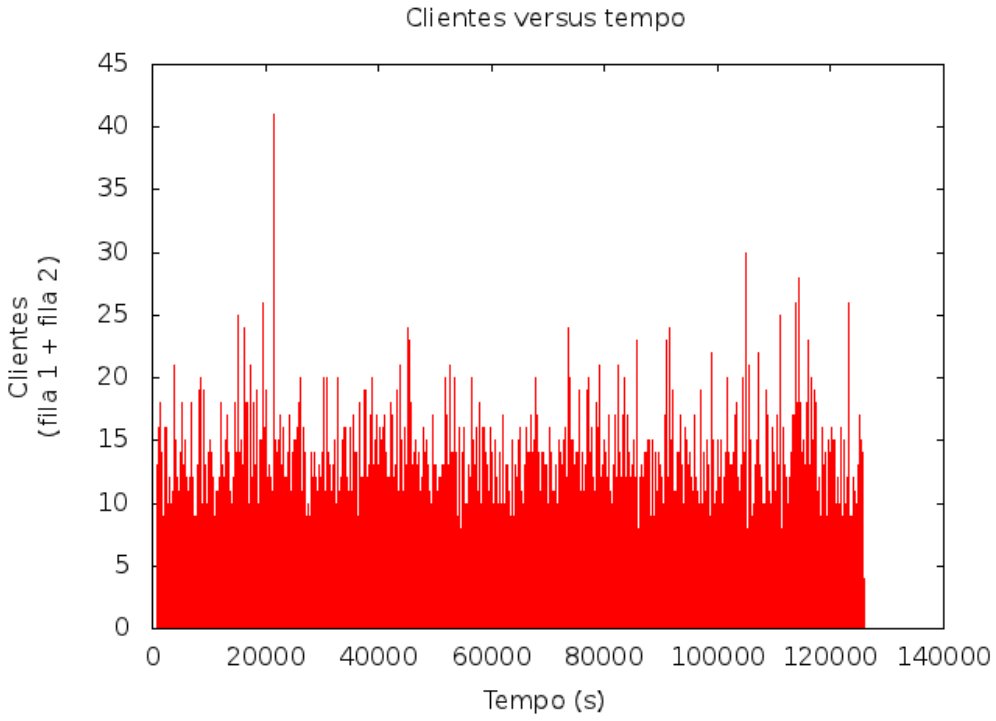
Gráficos da linha 2 da tabela 1



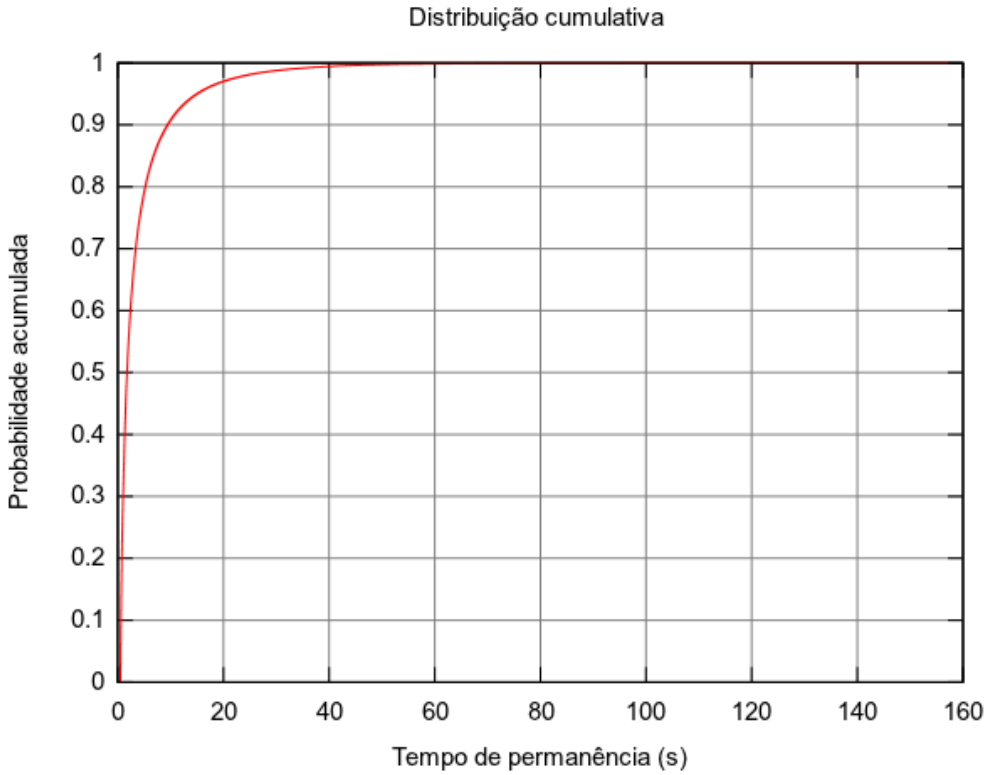


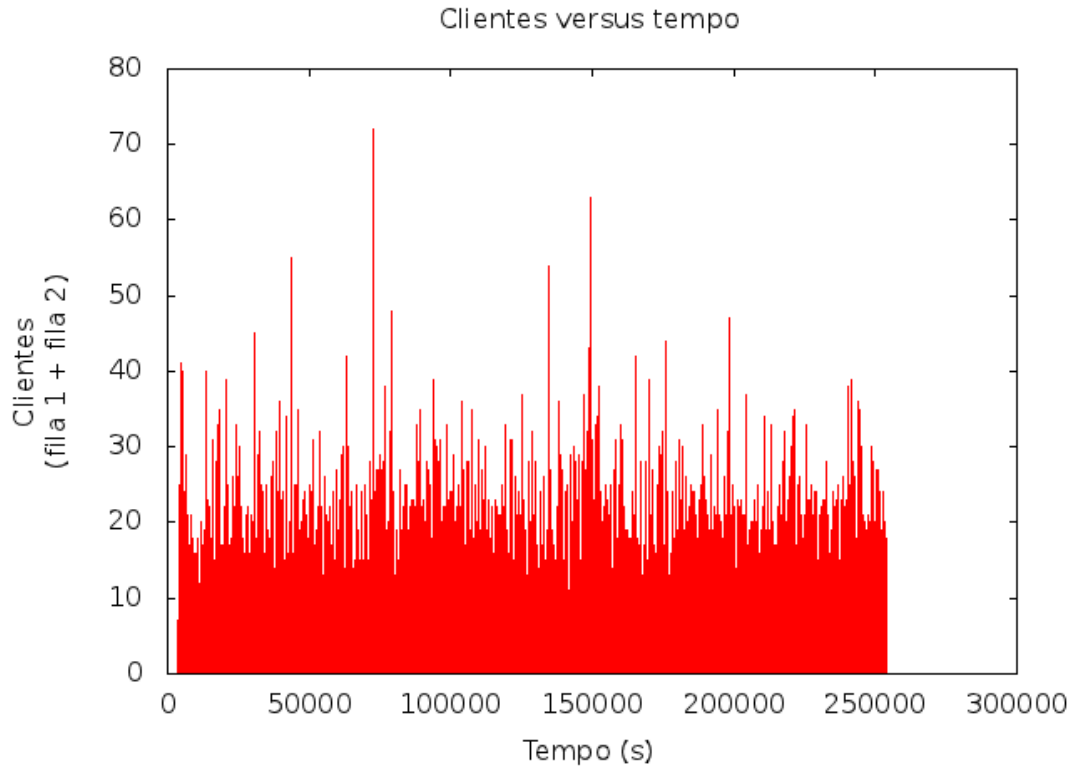
Gráficos da linha 3 da tabela 1



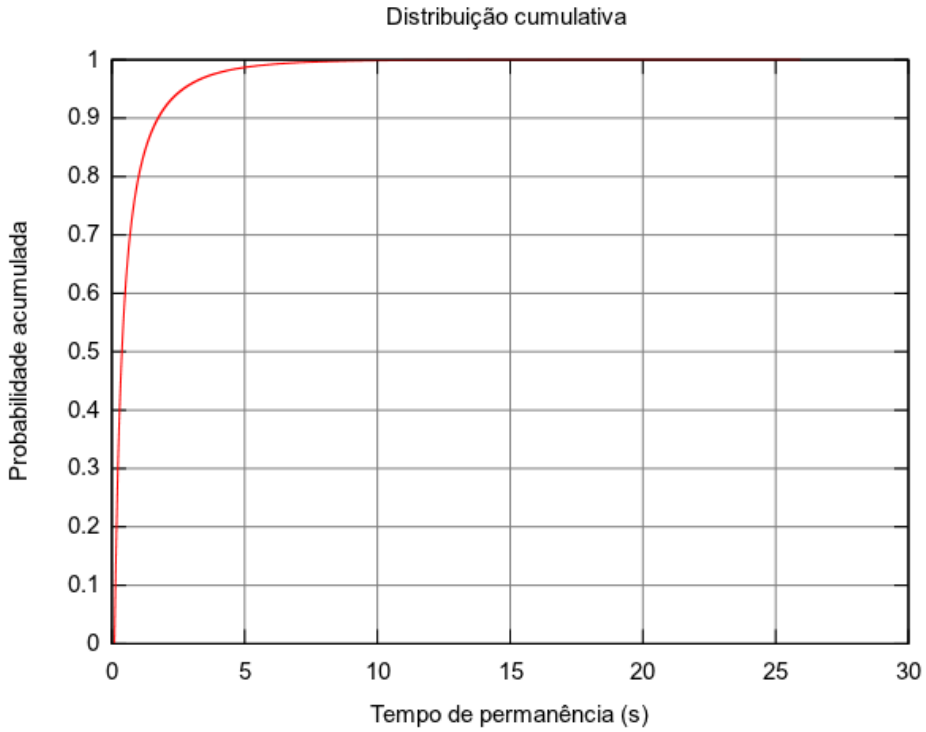


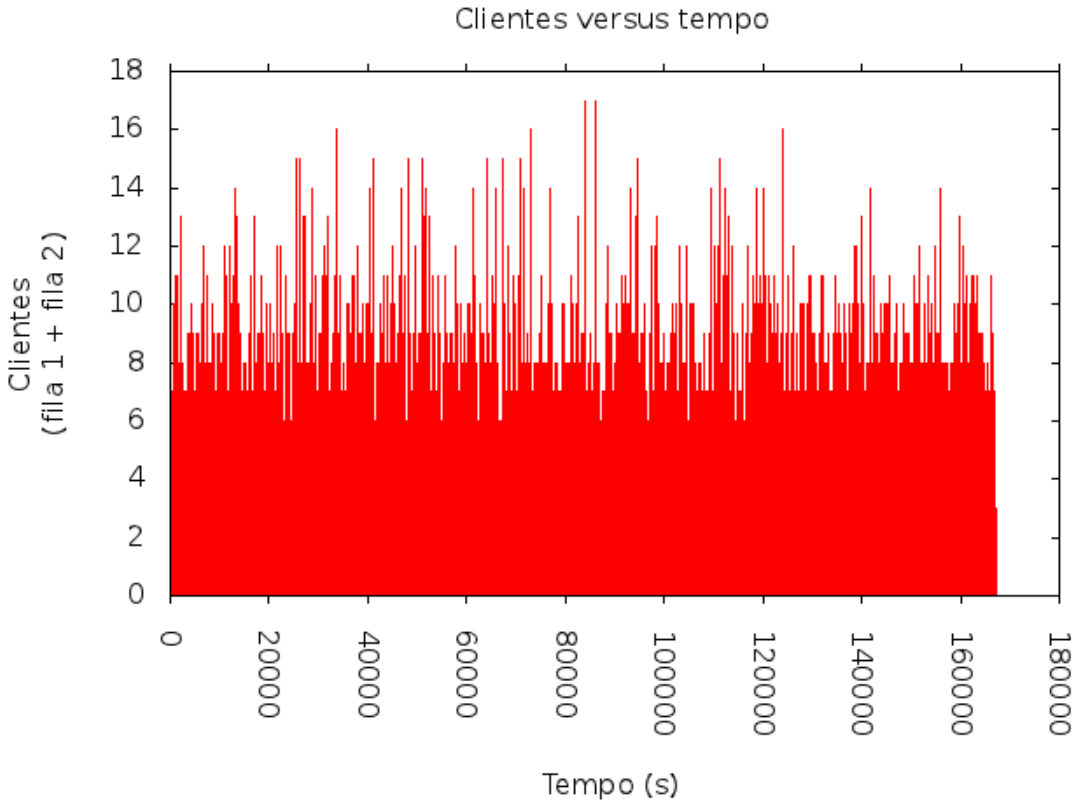
Gráficos da linha 4 da tabela 1



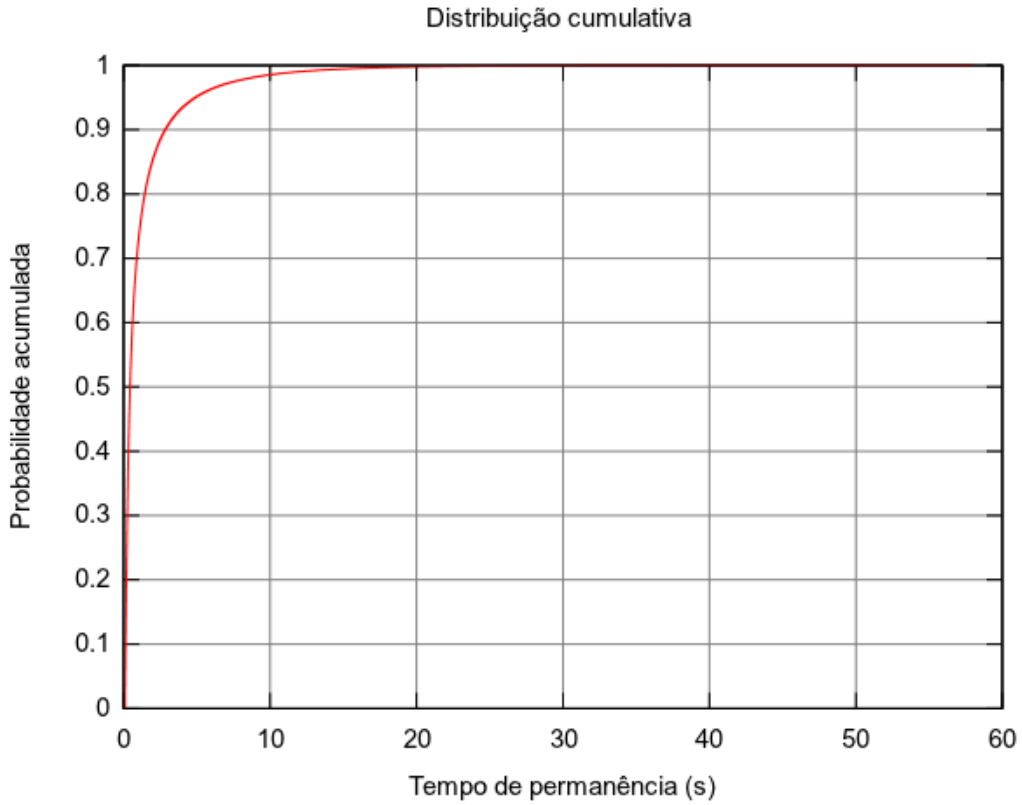


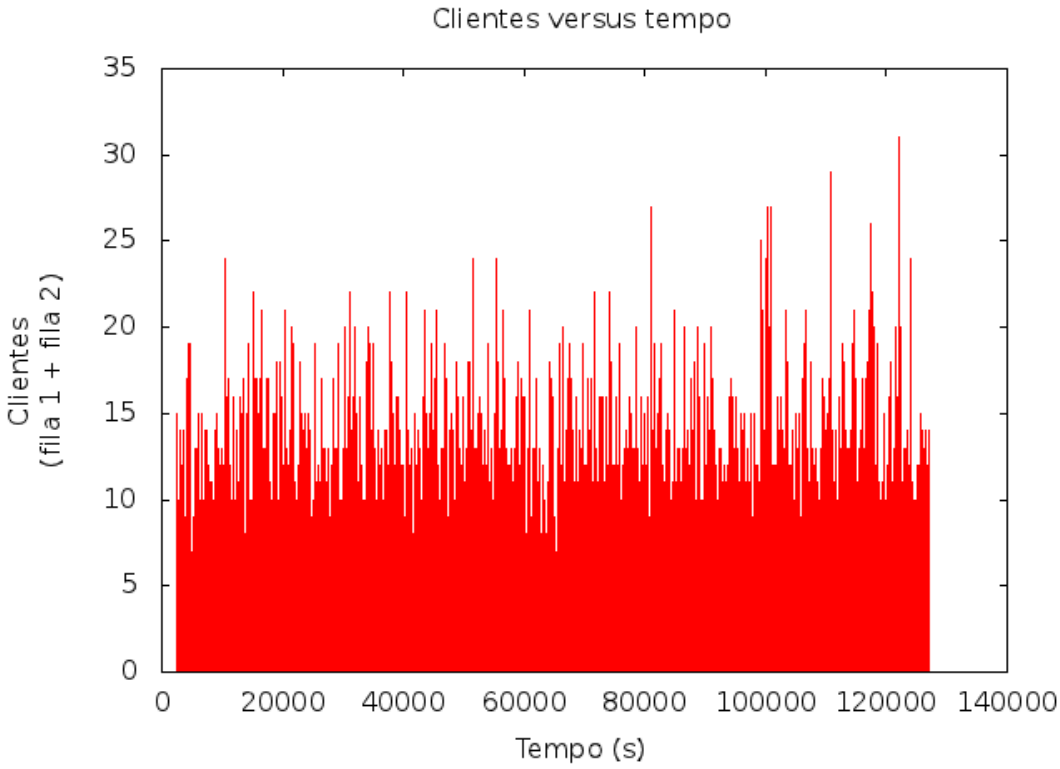
Gráficos da linha 5 da tabela 1



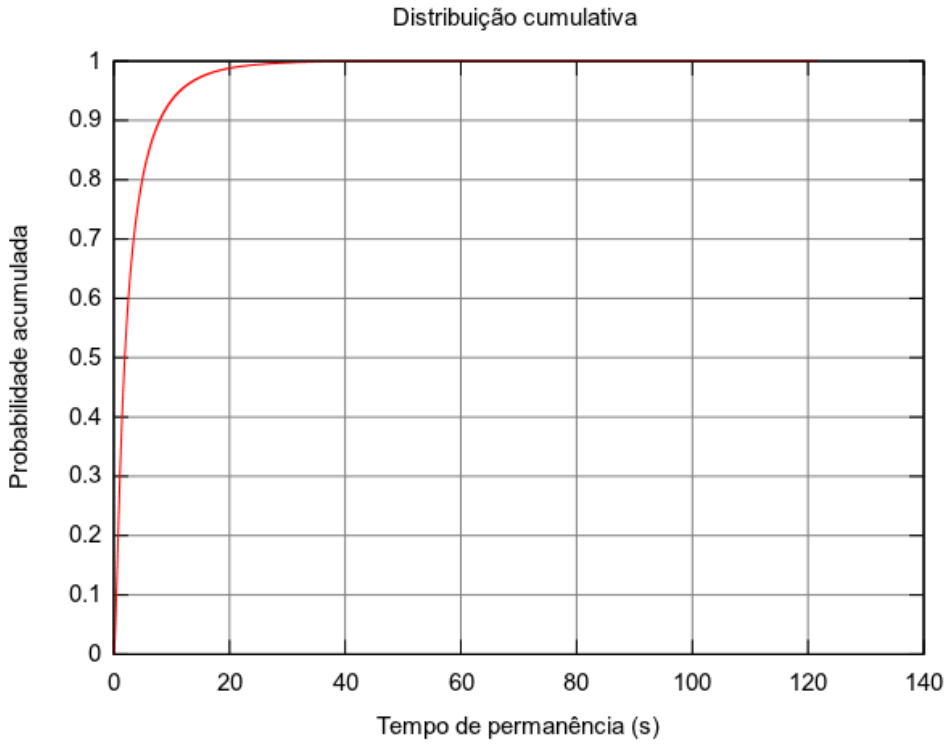


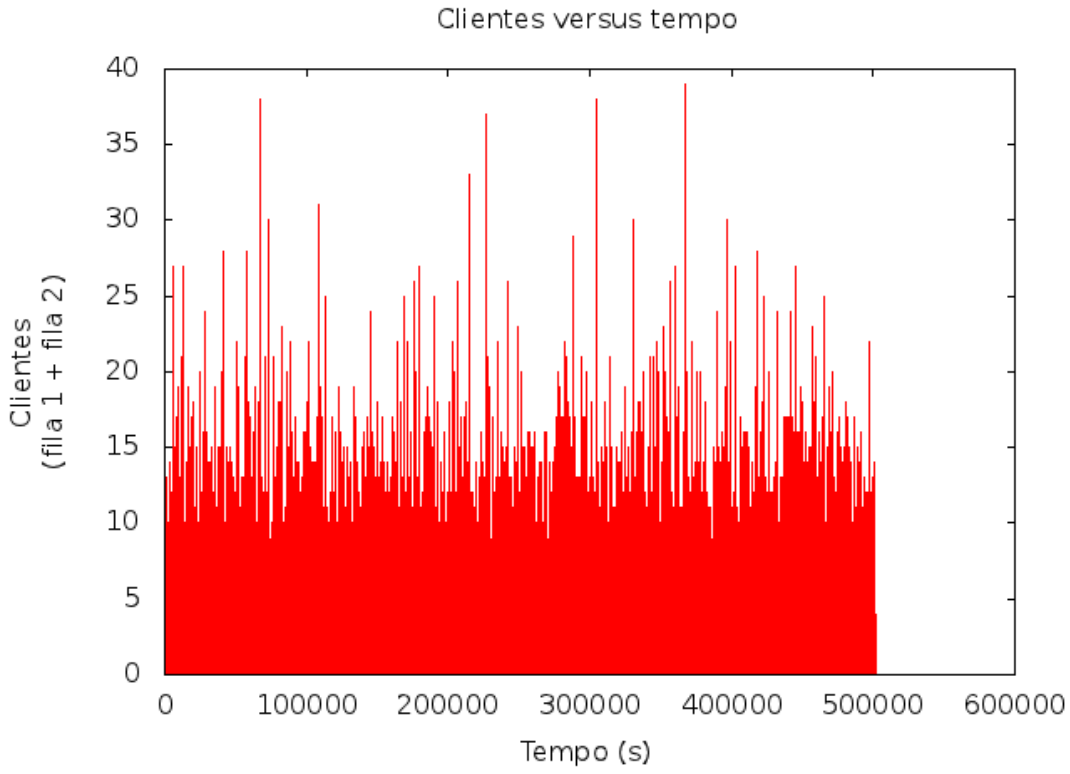
Gráficos da linha 6 da tabela 1



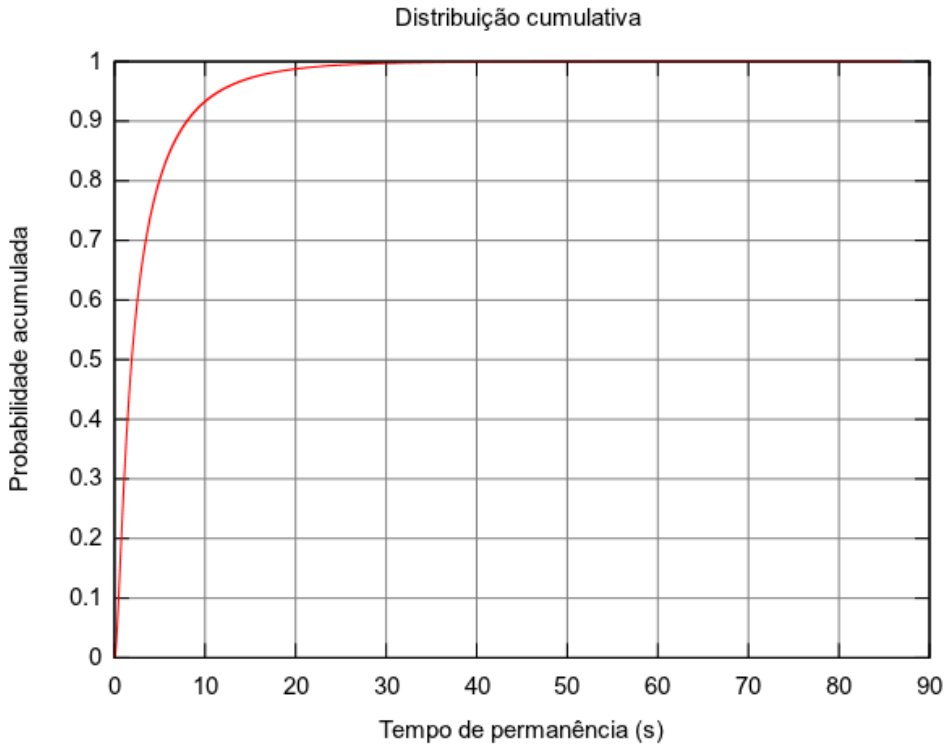


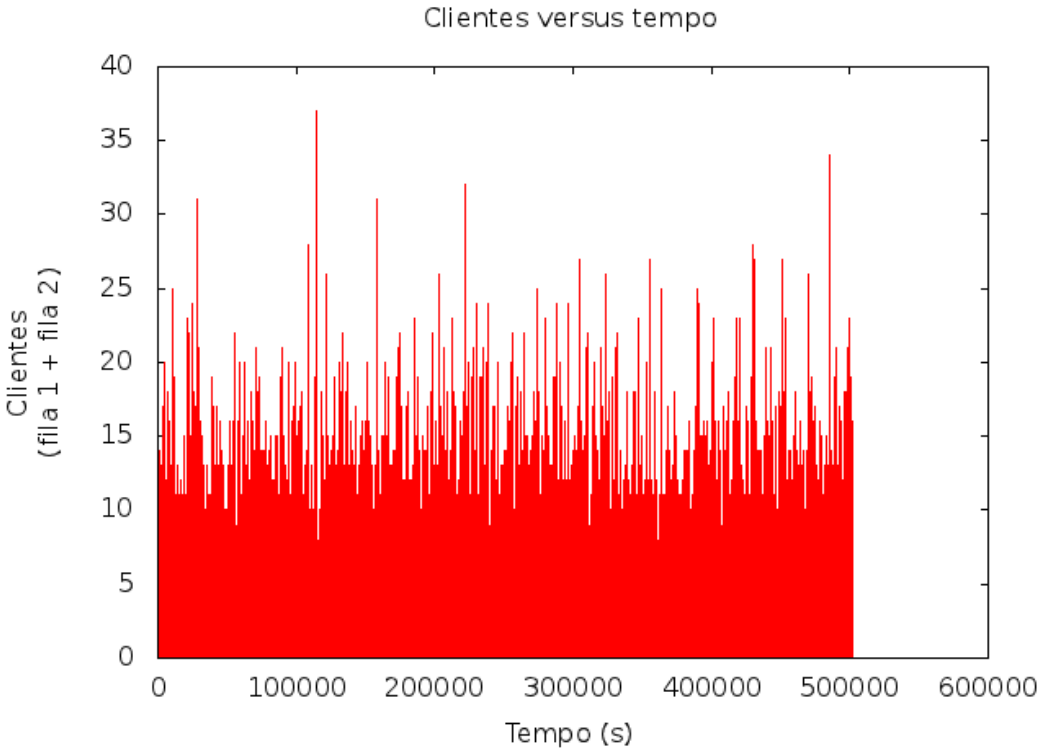
Gráficos da linha 7 da tabela 1





Gráficos da linha 8 da tabela 1





10 - Código Fonte do Simulador

A seguir incluímos o código das principais classes com somente os trechos mais importantes do simulador com comentários sobre o seu funcionamento. Para ver todo o código ver anexo.

Classe Simulador, que contém uma classe que representa o estado do sistema e o looping principal (com vários trechos retirados para facilitar entendimento).

```
public class Simulator {

	public Simulator(){
		eventList = new EventList();
		state= new State();
		state.lambdaArrival=3;
		state.lambdaDeparture=5;
		state.lambdaPriorityArrival=2;
		state.lambdaPriorityDeparture=5;
		freq = new int[1000];
		freqInalterada = new int[1000];
		top10 = new int[10][2];
		top10Values= new int[10][2];
		top10Percentages = new double [10][2];
	}
	public void go() {


		 eventList.enqueue(new PriorityArrivalEvent(state.lambdaPriorityArrival,state.time,EventType.CHEGADA,new
	PriorityArrivalEventAction(state,eventList)));

		double tempoAnterior;
		for(int i = 0; eventList.size() > 0; i++){
			Event event=eventList.dequeue();
			tempoAnterior=state.time;
			state.time=event.getTime();
			event.act();
			if(state.transientIsOver){
				//trecho que escreve no arquivo o número de clientes no intervalo de tempo
			}
			if ((i+1)%1000==0){
					//lógica que calcula se a fazer transiente acabou

			}

		}
	}
	public class State {

		//int n=0;
		double time;
		double lambdaArrival;
		double lambdaDeparture;
		int maxClients = 1000000;
		int clients;
		boolean transientIsOver = false;

		ClientList clientList;
		PriorityClientList priorityClientList;
		public double lambdaPriorityArrival;
		public double lambdaPriorityDeparture;
		public int InterruptedCase=2;
	}
```

Antes de mostrar o código das próximas classes uma breve explicação sobre a estrutura se faz necessária. Notas sobre estrutura:

- Um Evento possui uma variável tempo, que é o tempo que ele vai ocorrer. Ele possui uma classe EventAction, que é a classe responsável por implementar o método act, que será chamado quando o sistema estiver no tempo do evento;

- Todos os Tipos de eventos herdam da classe evento, assim como os tipos de event Action;

- A fila de eventos contém a qualquer momento no máximo quatro eventos, sendo no máximo um de cada tipo.

Sendo assim, quando uma chegada prioritária, por exemplo, é executada, ela enfileira a nova chegada prioritária com o tempo sorteado segundo a distribuição. Com essas observações podemos ver, por exemplo, uma classe que contém as ações a executar quando ocorre uma chegada prioritária (com trechos removidos).

```
public class PriorityArrivalEventAction extends EventAction {
    public void act() {
        if (! state.transientIsOver || state.clients < state.maxClients){
            //cria e enfileira a próxima chegada
            PriorityArrivalEventAction action = newPriorityArrivalEventAction(state,eventList);
                eventList.enqueue(newPriorityArrivalEvent(state.lambdaPriorityArrival,state.time,EventType.CHEGADA,action));
        }
        if (state.transientIsOver)   state.clients++;
        Client cl = new Client(state.time);
        state.priorityClientList.enqueue(cl);
        //se ele é o único prioritário ele deve se colocar em serviço, caso contrário isso será feito pelo saídaDeClienteDePrioriadade
        if(state.priorityClientList.size()==1){
            cl.startJobTime = state.time;
            //cria a sua saída e coloca na fila de enventos
            //caso tenha mais clientes na fila prioritária a saída prioritária será refeita pela ultima saída prioritária a ser realizada antes dessa chegada
            PriorityDepartureEventAction action = new PriorityDepartureEventAction(state,eventList);
            eventList.enqueue(new PriorityDepartureEvent(state.lambdaPriorityDeparture, state.time,EventType.CHEGADA, action));
            }
        //Desfazer a saída do cliente comum que estava sendo atendido, guardando suas informações de atendimento no cliente
        //Essa saída será refeita quando o último cliente da fila prioritária for atendido pela PriorityDepartureArrival
        Event saidaDoComunQueSeraRefeita = eventList.popNextDeparture();
        if(state.clientList.size()>0 && state.priorityClientList.size()==1){
                Client clientComun = state.clientList.dequeue();
                clientComun.commomAcumulatedJobTime= (state.time-clientComun.commomStartJobTime);
                clientComun.commomRemainingJobTime = saidaDoComunQueSeraRefeita.time -state.time;
                    //trata os dois casos , no qual o interrompido vai para o início ou para o final da fila
                    if(state.InterruptedCase==1){
                        state.clientList.enqueue(clientComun);
                    }
                    else if(state.InterruptedCase==2){
                        state.clientList.enqueueLast(clientComun);
                    }
        }
    }
}
```