UFRJ
UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

Trabalho 1 - Relatório Técnico

Simulação de uma Rede Aberta de Filas: Análise de Tempo de Serviço e Métricas de Desempenho

Autores
Pedro Saito : 122149392
Milton Salgado : 122169279

Professor Orientador
Vinícius Gusmão Pereira de Sá

Modelagem e Avaliação de Desempenho
Universidade Federal do Rio de Janeiro
2024

Sumário

1. Introdução

A modelagem de redes de filas constitui uma abordagem analítica essencial para compreender o desempenho de sistemas computacionais complexos. Mediante simulação de fluxos de serviços através de múltiplos servidores, essa técnica permite avaliar métricas críticas como tempo de espera, utilização de recursos e throughput.

Este estudo propõe implementar um simulador de rede de filas aberta, explorando experimentalmente como diferentes distribuições probabilísticas de tempos de serviço impactam o comportamento sistêmico.

1.1 Objetivos

O presente estudo tem como objetivo simular uma rede aberta de filas em diferentes condições de operação, avaliando métricas de desempenho relacionadas ao tempo de permanência dos jobs no sistema.

Especificamente, estamos interessados em:

- Implementar um simulador de rede aberta de filas com três servidores, considerando filas ilimitadas, probabilidades de roteamento e diferentes distribuições de tempos de serviço.
- Coletar métricas dos *jobs* no sistema de rede de filas, tal como o tempo médio no sistema e o desvio médio do tempo calculado.
- Analisar o desenvolvimento do sistema de rede de filas em que os tempos de serviços dos *jobs* são distribuídos de forma: **Determinística**, **uniforme** e, por fim, **exponencial**.
- Explicar as estruturas e funções do código, abordando o tratamento de processos estocásticos e a coleta de métricas.

Ao final, espera-se demonstrar a eficácia do simulador em capturar as dinâmicas do sistema e facilitar a visualização de diferentes distribuições de tempos de serviço no desempenho global da rede.

1.2 Premissas

Para o desenvolvimento do simulador, foram estabelecidas algumas premissas, listadas a seguir:

- **Três Servidores**: Conforme descrito no enunciado, há três servidores, identificados como "S1", "S2" e "S3". Os trabalhos chegam inicialmente ao servidor "S1" e, em seguida, têm probabilidade igual de serem direcionados para os demais servidores.
- **Processo de Poisson e Exponencial**: As chegadas dos *jobs* aos servidores constituem um **Processo de Poisson** com taxa $\lambda = 2$ jobs/s. Portanto, o tempo entre a chegada de dois *jobs* consecutivos pode ser descrito por uma **variável exponencial** com média de $0,5s$.
- **Distribuições do Tempo de Serviço**: Iremos analisar a distribuição do tempo de serviço sob diferentes circunstâncias, tal como:
  - Tempos de serviço fixos, isto é, determinísticos.
  - Tempos de serviço dados por uma variável uniforme.
  - Tempos de serviço distribuídos de forma exponencial.
- **Espera até início do Estado Estacionário**: Antes de coletar as métricas, deve-se aguardar a chegada dos primeiros 1.000.000 *jobs*, isto é, dando tempo o suficiente para que a cadeia entre no estado estacionário.

A representação do sistema que desejamos modelar está demonstrado abaixo:

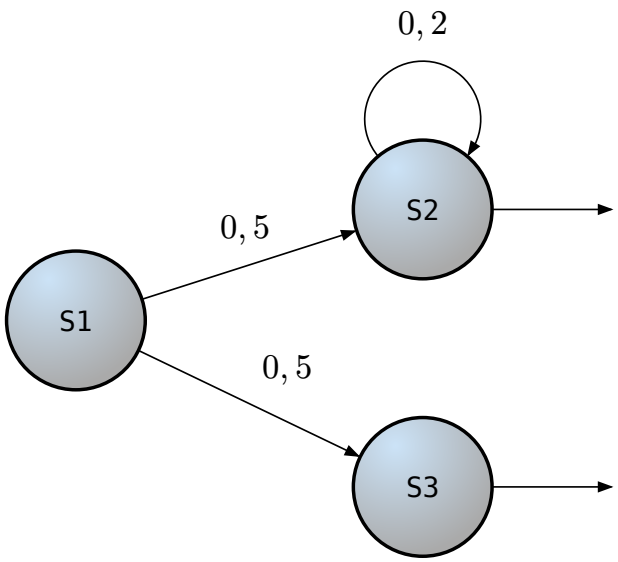

Figura 1: Representação da Rede de Filas com os Servidores "S1", "S2" e "S3".

2. Simulação

A simulação da cadeia de Markov aberta consiste em analisar um sistema de filas em rede, no qual serviços (ou tarefas) chegam ao sistema, passam por diferentes servidores, podendo seguir fluxos específicos ou deixar o sistema. Esse tipo de simulação é amplamente empregado em estudos de teoria de filas, análise de desempenho de sistemas e modelagem probabilística.

2.1 Funcionamento

Para conduzir a simulação da cadeia de Markov aberta, estabelecemos as seguintes premissas:

1. O sistema é composto por múltiplos servidores (por exemplo, S1, S2 e S3).
2. Os serviços chegam ao sistema seguindo um processo estocástico (geralmente Poisson), o que garante uma chegada contínua e independente.
3. Cada servidor possui seu próprio tempo de serviço, que pode ser determinístico ou seguir distribuições de probabilidade (exponencial, uniforme, etc.).
4. O objetivo da simulação é analisar métricas como tempo médio de permanência no sistema, throughput e taxas de ocupação, sob diferentes cenários.

Com base nessas premissas, descrevemos a dinâmica da simulação:

- Serviços chegam inicialmente ao primeiro servidor (S1).
- Após o processamento em S1, os serviços podem ser roteados para outros servidores (por exemplo, S2 ou S3) de acordo com probabilidades pré-definidas.
- Alguns servidores podem apresentar realimentação interna: um serviço, após ser atendido, pode retornar ao mesmo servidor com certa probabilidade, antes de prosseguir para outro estágio ou deixar o sistema.
- Quando um serviço conclui o último estágio de atendimento, ele sai do sistema, permitindo analisar a distribuição dos tempos de permanência.

Na prática, para a simulação:

- Utiliza-se uma abordagem por eventos, na qual cada chegada e saída de serviço é um evento agendado.
- Ao processar um evento, o estado dos servidores, as filas e o tempo global da simulação são atualizados.
- A simulação é executada até que um número significativo de serviços seja processado, descartando-se um período inicial de aquecimento, a fim de se obter métricas estatisticamente estáveis.

2.2 Exemplo

Considere um exemplo simples de um sistema com três servidores, S1, S2 e S3, no qual:

- Os serviços chegam ao S1 a uma taxa λ.
- Após o atendimento em S1, um serviço pode ser encaminhado para S2 ou S3 com probabilidades p(S2) e p(S3).
- Em S2, existe a possibilidade de o serviço retornar ao mesmo servidor (feedback) com probabilidade p_f, prolongando seu tempo no sistema.
- Ao concluir o atendimento em S3 ou sair de S2 sem feedback, o serviço deixa o sistema.

As taxas de chegada, tempos médios de serviço e probabilidades de roteamento podem ser ajustados para estudar diferentes cenários. Por exemplo, se definirmos:

Servidor	Distribuição de Serviço	Próximo Servidor	Probabilidade
S1	Exponencial (média 0.4s)	S2	0.5
S1	Exponencial (média 0.4s)	S3	0.5
S2	Exponencial (média 0.6s)	S2	0.2
S2	Exponencial (média 0.6s)	Saída	0.8
S3	Exponencial (média 0.95s)	Saída	1.0

Ao executar a simulação, podemos observar estatísticas como tempo médio que os serviços passam no sistema, a carga média de cada servidor e a eficiência do sistema para diferentes distribuições e parâmetros.

3. Código

Nesta seção, descrevemos a implementação do simulador de sistema de servidores, incluindo as classes de dados, módulos e funções principais. O código em Python foi desenvolvido seguindo práticas de modularidade e simplicidade, facilitando a manutenção e a expansão do simulador.

3.1 Organização do Projeto

O projeto tem quatro arquivos de cabeçalho, cada um com um arquivo `.c` correspondente, exceto o `main.c`. As funções estão declaradas nos arquivos de cabeçalho. Abaixo está a estrutura do projeto:

```
├── README.md
├── app
│   ├── exemplo.py
│   ├── modelos.py
│   ├── simulacao.ipynb
│   └── simulacao.py
└── utils
│   └── requirements.txt
└── docs
    └── relatorio.pdf
```

3.2 Classes e Funções

Durante a implementação da simulação da cadeia de Markov aberta, foram definidas diversas classes e funções para representar servidores, serviços e eventos no sistema. Cada classe encapsula um aspecto fundamental da simulação, permitindo organizar e manipular o estado do sistema de forma estruturada.

Abaixo, detalhamos as classes e estruturas utilizadas, assim como suas funções mais relevantes.

3.2.1 Serviços

A classe `Servico` representa um trabalho que chega ao sistema, podendo passar por um ou mais servidores antes de sair. Ela contém as seguintes variáveis como:

Variável	Descrição
`id`	Identificador único do serviço.
`tempo_chegada`	Tempo de chegada do serviço ao sistema.
`tempos_servico`	Dicionário contendo tempos gastos em cada servidor.
`tempo_saida`	Tempo em que o serviço saiu do sistema.

3.2.2 Servidores

A classe `Servidor` representa um servidor dentro da rede de filas. Cada servidor possui:

Variável	Descrição
`nome`	Nome do servidor (por ex.: "S1", "S2", "S3").
`fila`	Lista de serviços aguardando processamento.
`ocupado_ate`	Tempo até o qual o servidor estará ocupado.
`funcao_tempo_servico`	Função que determina o tempo de serviço para novos atendimentos.
`servico_atual`	Serviço atualmente em processamento, se houver.

3.2.3 Eventos

A classe `Evento` modela qualquer mudança de estado que ocorre na simulação. Há dois tipos de eventos principais: "chegada" e "saida". Cada `Evento` possui:

Variável	Descrição
`tempo`	Tempo no qual o evento ocorre.
`tipo`	Tipo do evento ("chegada" ou "saida").
`servico`	Serviço associado ao evento (se aplicável).
`servidor`	Servidor associado ao evento (se aplicável).

3.2.4 Simulação

As funções que implementam a lógica da simulação são agregadas na classe `Simulacao`. Alguns dos principais métodos são:

1. `__init__(self, funcoes_tempo_servico: dict)`

   - Inicializa a simulação, criando os servidores e definindo funções de tempo de serviço para cada um.
   - Não possui valor de retorno.

2. `executar(self)`

   - Executa a simulação até que um determinado número de serviços seja coletado.
   - Processa eventos de forma cronológica, atualizando o estado do sistema (filas, servidores e métricas).
   - Ao final, calcula e exibe estatísticas (tempo médio no sistema, desvio padrão).
   - Não possui valor de retorno.

3. `agendar_evento(self, evento: Evento)`

   - Insere um evento na fila de eventos, ordenada pelo tempo.
   - Não possui valor de retorno.

4. `processar_chegada(self, evento: Evento)`

   - Trata a chegada de um novo serviço ou o repasse de um serviço já existente a um servidor específico.
   - Caso o servidor esteja livre, inicia o processamento imediatamente. Caso contrário, o serviço é inserido na fila de espera do servidor.
   - Agenda o próximo evento de chegada (se for um novo serviço externo) e o evento de saída deste serviço quando concluído.
   - Não possui valor de retorno.

5. `processar_saida(self, evento: Evento)`

   - Trata a conclusão do atendimento de um serviço em um servidor.
   - Caso haja serviços na fila de espera, inicia o próximo imediatamente.
   - Dependendo do servidor, o serviço pode ser encaminhado a outro servidor, voltar ao mesmo ou sair do sistema.
   - Registra o tempo total de permanência no sistema, se aplicável.
   - Não possui valor de retorno.

3.2.5 Distribuições de Tempo de Serviço

As funções responsáveis por gerar os tempos de serviço para cada servidor são fornecidas via dicionário no construtor da simulação. Podem ser determinísticas, uniformes ou exponenciais:

1. **Tempos Determinísticos:** Na situação 1, os tempos de serviço são constantes e específicos para cada servidor. Por exemplo, o servidor S1 sempre leva 0.4 segundos para processar um serviço.

2. **Distribuições Uniformes:** Na situação 2, os tempos de serviço seguem uma distribuição uniforme dentro de um intervalo específico. Por exemplo, o servidor S1 tem tempos de serviço entre 0.1 e 0.7 segundos, escolhidos aleatoriamente usando random.uniform.

3. **Distribuições Exponenciais:** Na situação 3, os tempos de serviço seguem uma distribuição exponencial. Para gerar tempos de serviço exponenciais, utilizamos random.expovariate. A função expovariate(1/λ) gera tempos de serviço exponenciais com média λ.

3.2.6 Filas de Prioridade com Heap

A fila de eventos é implementada como uma fila de prioridade usando um heap. Em Python, a biblioteca heapq é utilizada para manter a fila ordenada por tempo, permitindo que os eventos sejam processados em ordem cronológica. O heap é eficiente para inserir elementos e extrair o menor elemento, ambos em tempo O(log n).

3.2.7 Métricas e Ajustes

Para análise de desempenho, a simulação ignora um conjunto inicial de serviços (período de aquecimento) e, posteriormente, coleta estatísticas dos serviços seguintes. Essas métricas incluem:

- **Tempo médio no sistema:** Média do tempo que cada serviço dura no sistema.
- **Desvio padrão:** Variação do tempo de permanência, permitindo medir a dispersão.

Ao término da execução, o simulador imprime os resultados obtidos, permitindo a análise do comportamento do sistema sob diferentes parâmetros de chegada, roteamento e tempo de serviço.

4. Saída

```text
Simulação - Situação 1
Tempo médio no sistema: 7.091735829777251
Desvio padrão do tempo no sistema: 8.543577938831138

# Exibição do gráfico com tempo de serviço determinístico

Simulação - Situação 2
Tempo médio no sistema: 8.768989782193337
Desvio padrão do tempo no sistema: 10.966880115767493

# Exibição do gráfico com tempo de serviço uniforme

Simulação - Situação 3
Tempo médio no sistema: 12.594940058080958
Desvio padrão do tempo no sistema: 15.260461154376697

# Exibição do gráfico com tempo de serviço exponencial
```

5. Resultados

O primeiro gráfico ilustra a dinâmica do tempo de serviço sob um modelo determinístico, revelando o comportamento temporal dos processos em um contexto de variação controlada.

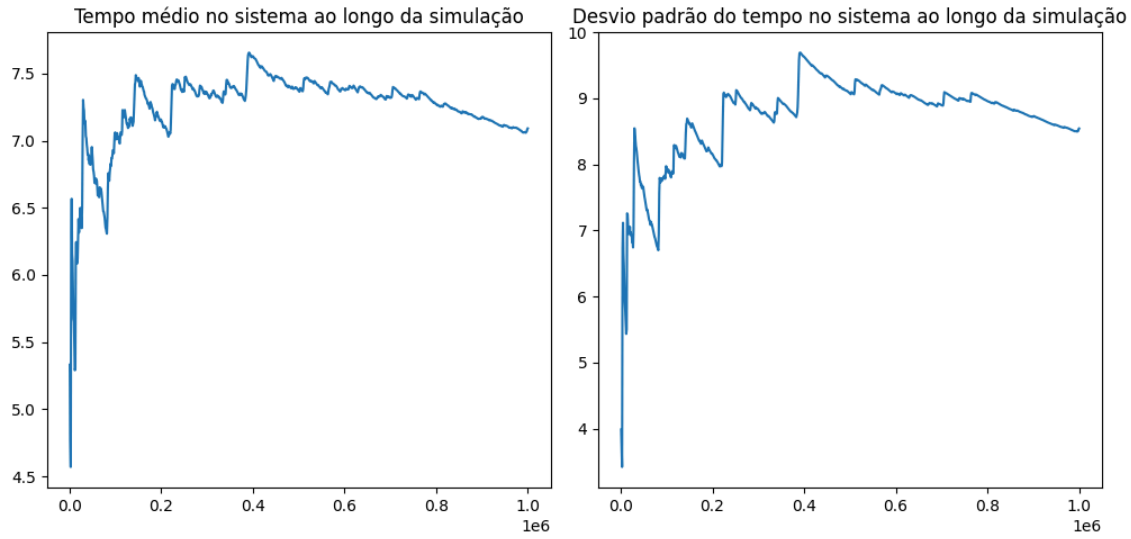

Figura 2: Resultado para tempos de serviços determinísticos

O gráfico do tempo médio mostra uma progressão inicial de 4.5 para 7.0, com flutuações que sugerem variações no processamento de serviços. O desvio padrão segue trajetória similar, aumentando de 4 para 8.5, indicando maior variabilidade nos tempos de serviço à medida que o sistema evolui. Essas oscilações refletem a complexidade operacional e as nuances do processamento do sistema.

O gráfico subsequente apresenta a dinâmica do tempo de serviço segundo um modelo uniforme, evidenciando a distribuição equitativa dos intervalos de processamento em um espectro de variação padronizada.

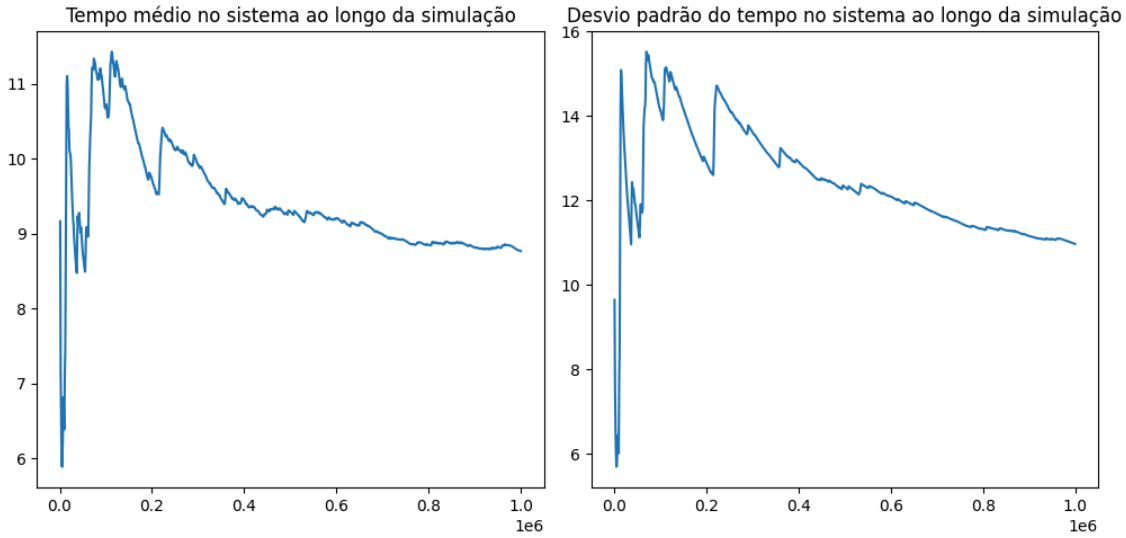

Figura 3: Resultado para tempos de serviços uniformes

A análise dos gráficos revela uma tendência decrescente tanto no tempo médio quanto no desvio padrão dos serviços. Essa redução sugere uma melhoria progressiva na eficiência do sistema, com processamentos mais rápidos e consistentes. Os resultados indicam um possível equilíbrio operacional, onde os tempos de serviço se tornam mais previsíveis e ágeis ao longo da simulação.

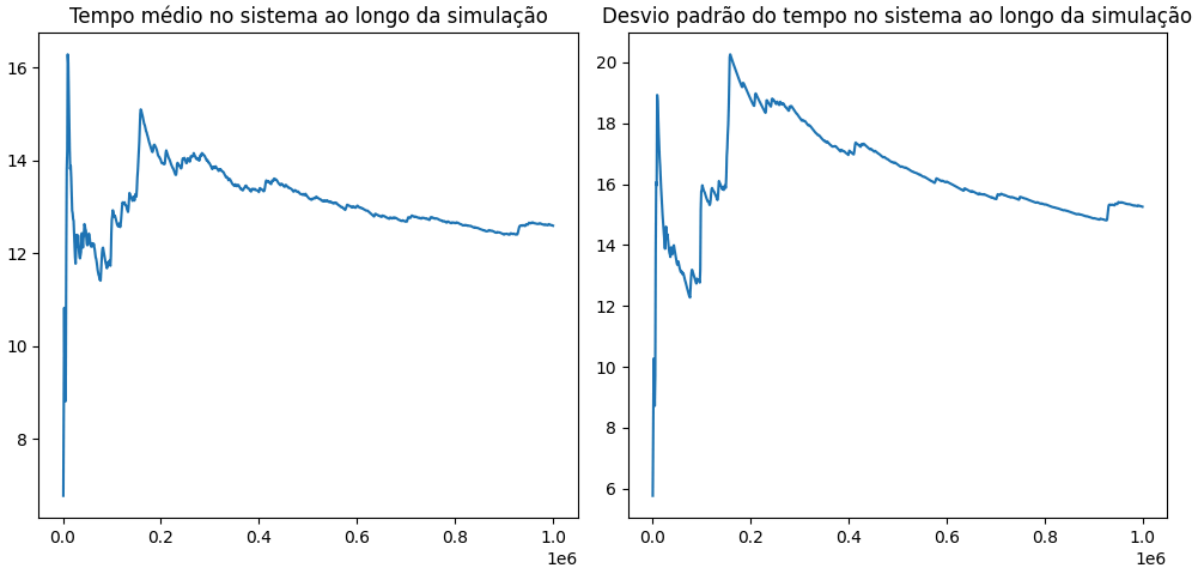

Figura 4: Resultado para tempos de serviços exponenciais

No primeiro gráfico observa-se que a linha começa em torno de 8, atinge um pico em torno de 16 e depois diminui gradualmente para cerca de 12. Isso indica que, no início, os tempos médios no sistema aumentaram significativamente, possivelmente devido ao aquecimento do sistema e à chegada inicial de muitos serviços. Com o tempo, o sistema começou a processar os serviços de forma mais eficiente, resultando na diminuição do tempo médio. Já no segundo gráfico, a linha começa em torno de 6, atinge um pico em torno de 20 e depois diminui gradualmente para cerca de 14. Esse aumento inicial no desvio padrão sugere uma maior variabilidade nos tempos de serviço no início da simulação. À medida que a simulação avança, essa variabilidade diminui, indicando que o sistema se estabiliza com tempos de serviço mais consistentes.

5.0.1 Caso Determinístico

No caso determinístico, todos os tempos de serviço têm valores fixos e iguais. Isso resulta em um comportamento altamente previsível, onde os tempos de resposta e a ocupação dos servidores refletem diretamente as diferenças nas taxas de chegada e distribuição do trabalho.

1. Gráfico de Duração dos Jobs no Sistema:

   - O histograma apresenta um pico único em torno de um valor fixo. Isso reflete a natureza determinística, onde todos os serviços levam o mesmo tempo para serem processados.
   - A fila se mantém pequena, já que o tempo é previsível e os servidores processam os jobs de maneira uniforme.

2. Utilização Média dos Servidores:

   - Os valores de utilização variam de acordo com a carga de trabalho distribuída para cada servidor. Um servidor pode ter alta utilização (ex.: 95% em S3) se for mais frequentemente escolhido, enquanto outros, como S2, podem apresentar menor carga.

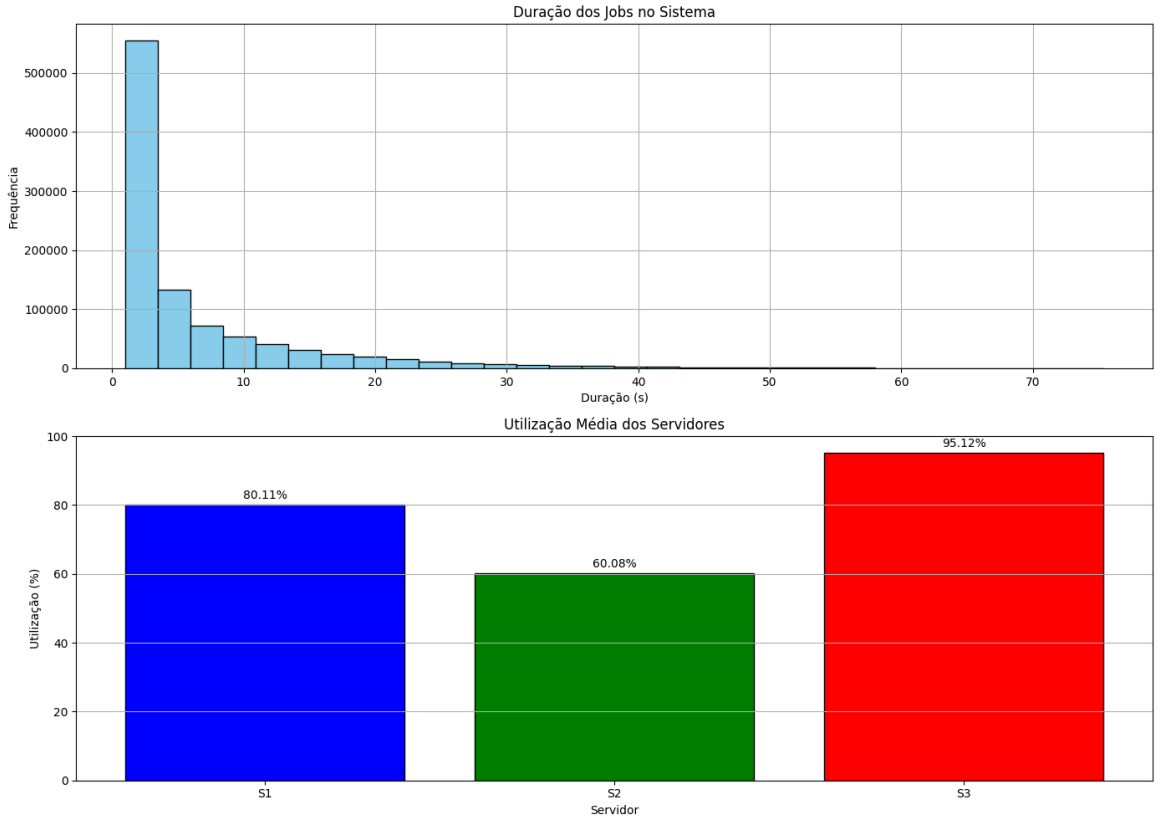

Figura 5: Resultado para tempos de serviços determinísticos

5.0.2 Caso Uniforme

No caso uniforme, os tempos de serviço são distribuídos aleatoriamente dentro de um intervalo fixo (ex.: 2 a 5 segundos). Isso introduz uma variabilidade moderada no sistema, influenciando os tempos de espera e a ocupação dos servidores.

1. Gráfico de Duração dos Jobs no Sistema:

   - O histograma apresenta uma distribuição uniforme, com frequências aproximadamente iguais para os tempos de serviço dentro do intervalo definido.
   - A maior variabilidade nos tempos de serviço pode levar a flutuações no tamanho das filas.

2. Utilização Média dos Servidores:

   - A utilização dos servidores será mais balanceada em comparação ao caso determinístico, devido à maior distribuição aleatória dos tempos de serviço.
   - Pequenas diferenças podem surgir devido a eventos de alta carga momentânea.

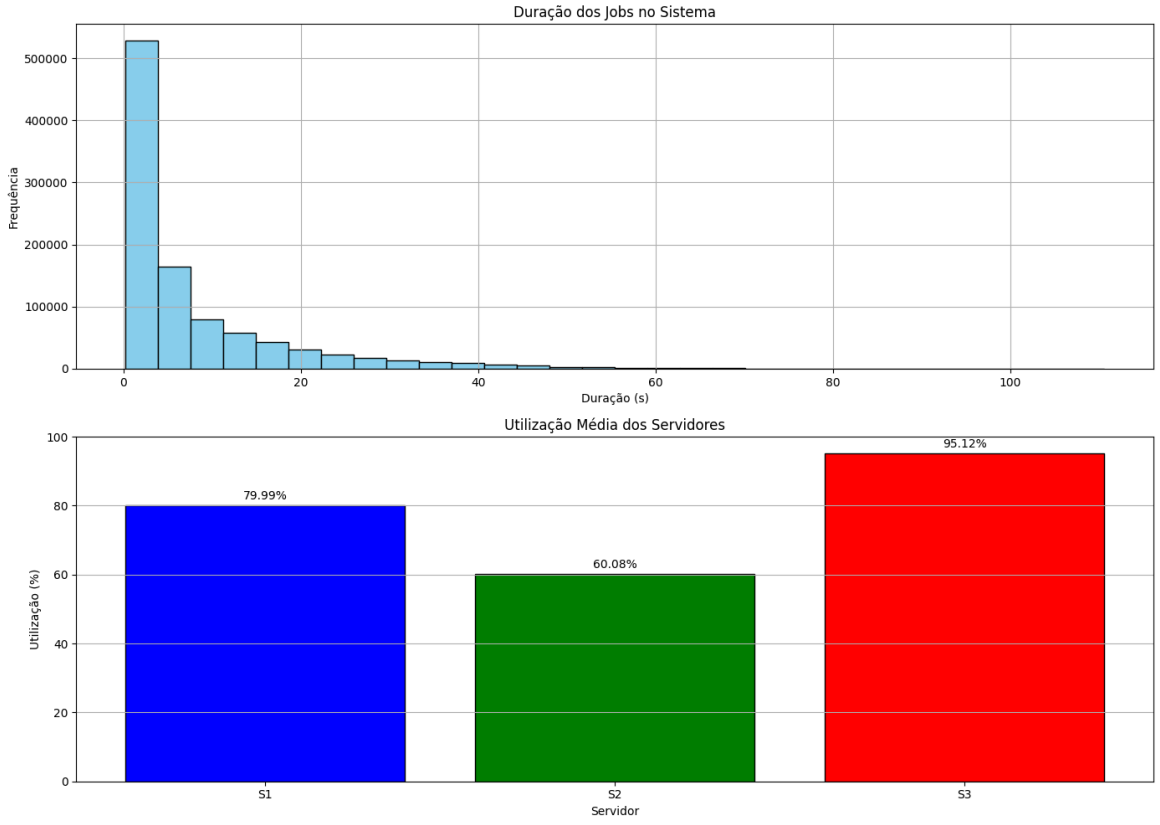

Figura 6: Resultado para tempos de serviços determinísticos

5.0.3 Caso Exponencial

No caso exponencial, os tempos de serviço são distribuídos de forma assimétrica, com muitos serviços rápidos e alguns significativamente mais longos. Essa é uma distribuição comum em sistemas reais, como redes e filas.

1. Gráfico de Duração dos Jobs no Sistema:

   - O histograma apresenta uma concentração de serviços com duração curta, mas uma "cauda longa" de serviços com tempos elevados.
   - Esse comportamento pode levar ao acúmulo de filas em momentos de maior carga, principalmente para serviços de longa duração.

2. Utilização Média dos Servidores:

   - A utilização será mais imprevisível, com picos elevados em servidores que recebem serviços longos. Isso pode levar a desequilíbrios temporários na carga dos servidores.
   - Em média, a utilização pode ser similar ao caso uniforme, mas com maior variância entre os servidores.

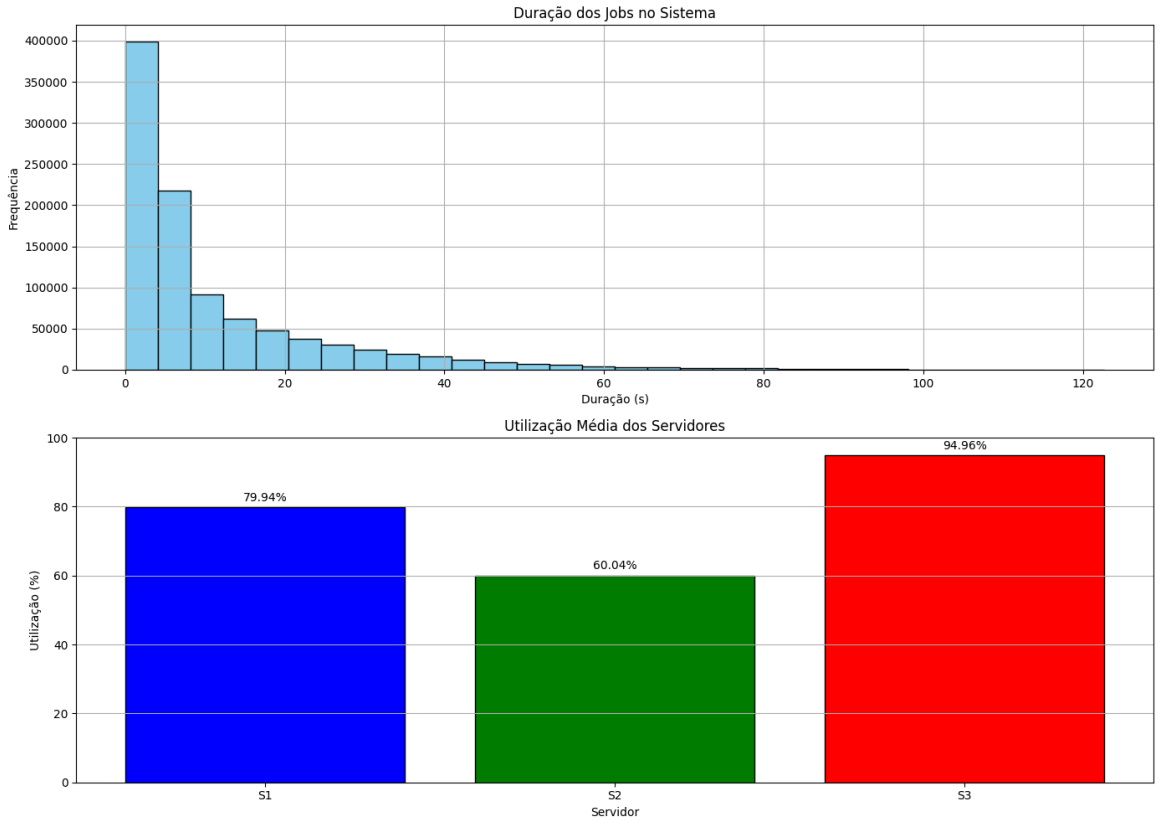

Figura 7: Resultado para tempos de serviços determinísticos

6. Conclusão

Analiticamente, esperávamos que o servidor 2 tivesse uma utilização maior por conta do “self-loop”, garantindo que mais serviços chegassem na fila e fossem processados, mas aparentemente, a tendência do tamanho médio de fila maior se concentrou no servidor 3, aumentando drasticamente sua utilização, deixando o servidor 2 ocioso por mais tempo.

Além disso, ao calcularmos analiticamente o tempo médio (essencialmente para a primeira situação), percebemos que o tempo na fila influencia em praticamente 90% o tamanho do tempo médio no sistema, dado que o tempo médio no servidor (não contando a fila) foi de aproximadamente 1.25.

Também notamos que a amostragem pela inversa das distribuições probabilísticas utilizadas revela as variâncias no comportamento do sistema ao se adaptar às naturezas dos eventos e seus serviços, em cada uma das situações.

Bibliografia

[1] Harchol-Balter M. Performance Modeling and Design of Computer Systems: Queueing Theory in Action. Cambridge, UK: Cambridge University Press; 2013