UNIVERSIDADE FEDERAL DE ALFENAS

TIAGO COSTA SOARES

Trabalho Final

ALFENAS

2023

TIAGO COSTA SOARES

Trabalho Final

Trabalho apresentado à disciplina de Análise de Desempenho, ministrada pelo professor Dr. Flávio Barbieri Gonzaga, no curso de Ciência da Computação na Universidade Federal de Alfenas, Campus Santa Clara.

ALFENAS

2023

Resumo

Esse relatório tem como objetivo modificar o código apresentado e desenvolvido durante as aulas da disciplina Análise de Desempenho a fim de fixar o aprendizado do conteúdo ministrado. Os objetivos específicos do trabalho são: Etapa 1: Simular um roteador que recebe a chegada de pacotes de navegação web; Coletar os dados equivalentes às Medidas de Desempenho (Ocupação, E[N], E[W], Erro de Little); Executar a simulação para 4 cenários com taxas de tempo médio de serviço diferentes; Gerar os gráficos para os valores obtidos.

Palavras-chave: Simulação. Ocupação. E[N]. E[W]. Erro de Little.

SUMÁRIO

1. INTRODUÇÃO

Para o desenvolvimento deste trabalho foi disponibilizado um código fonte chamado simulacao_little.c. O objetivo principal é o de modificar o código fonte, fixando parâmetros para a coleta de dados e posterior análise.

Para isso foram fixados os seguintes parâmetros: Taxa média de chegada; Tempo de simulação; Tempo médio de serviço.

O objetivo de fixar os parâmetros, assim como a semente aleatória, é a de permitir que o processo seja reproduzido com a obtenção dos mesmos resultados.

2. FIXANDO OS PARÂMETROS

Nessa primeira etapa foram fixados os parâmetros necessários para a simulação, assim como calculados os valores de ocupação a serem usados posteriormente em cada um dos cenários exigidos. Para o tempo de simulação foi fixado o valor de 36000, correspondente ao número de segundos simulados. Já a taxa média de chegada foi fixada em 0.01, ou seja, ocorrem, em média, 100 chegadas por segundos.

```
int main()
{
    double tempo_simulacao = 36000;
    double intervalo_media_chegada = 0.01;
```

Para fixar as taxas de ocupação primeiro foi necessário calcular a taxa de transmissão média, dada pela média dos tamanhos dos pacotes multiplicado pela média da taxa de chegada. (0.5*550 + 0.4*40 + 0.1*1500)*100. Com o resultado obtido foi possível calcular o tamanho do link, através da fórmula: Capacidade de atendimento do Link = taxa de transmissão / ocupação. Por fim pode-se calcular o tempo de atendimento médio para cada pacote, através da fórmula: L/R sendo L o tamanho do pacote e R a capacidade de atendimento do link. Para facilitar o desempenho do código, os valores foram colocados em uma matriz que irá ser percorrida durante as iterações de cada cenário.

```
// taxa de transmissão média = (0.5*550 + 0.4*40 + 0.1*1500)*100 = 44100 bytes/segundo
// tamanho do link(ocupação 60%) = taxa de transmissão / ocupação = 44100 / 0.6 = 73500 bytes/segundo
// tamanho do link(ocupação 80%) = taxa de transmissão / ocupação = 44100 / 0.8 = 55125 bytes/segundo
// tamanho do link(ocupação 95%) = taxa de transmissão / ocupação = 44100 / 0.95 = 46421 bytes/segundo
// tamanho do link(ocupação 99%) = taxa de transmissão / ocupação = 44100 / 0.99 = 44545 bytes/segundo
```

```
// Ocupação 60% = 73500 bytes/segunds
// tempo de atendimento = L/R = 550 / 73500 =  0.007482993197278911 segundos
// tempo de atendimento = L/R = 40 / 73500 =  0.0005442176870748299 segundos
// tempo de atendimento = L/R = 1500 / 73500 = 0.02040816326530612  segundos
// Ocupação 80% = 55125 bytes/segunds
// tempo de atendimento = L/R = 550 / 55125 = 0.009977324263038548  segundos
// tempo de atendimento = L/R = 40 / 55125 = 0.0007256235827664399 segundos
// tempo de atendimento = L/R = 1500 / 55125 = 0.027210884353741496 segundos
// Ocupação 95% = 46421 bytes/segunds
// tempo de atendimento = L/R = 550 / 46421 = 0.011848085995562353 segundos
// tempo de atendimento = L/R = 40 / 46421 = 0.0008616789814954439 segundos
// tempo de atendimento = L/R = 1500 / 46421 = 0.03231296180607914 segundos
// Ocupação 99% = 44545 bytes/segunds
// tempo de atendimento = L/R = 550 / 44545 = 0.012347064765967 segundos
// tempo de atendimento = L/R = 40 / 44545 = 0.012347064765967 segundos
// tempo de atendimento = L/R = 1500 / 44545 = 0.033673812998091815 segundos
```

```
double tempos_medios_servicos[4][3] = {{550.0/73500.0, 40.0/73500.0, 1500.0/73500.0}, {550
```

A semente aleatória também foi fixada, para permitir que os mesmos resultados possam ser obtidos posteriormente em outras análises.

```
double aleatorio()
{
    double u = rand() / ((double)RAND_MAX + 1);
    // limitando entre (0,1]
    u = 1.0 - u;
```

Para garantir as taxas de chegada exigidas para cada tipo de pacote, sempre que uma nova chegada (em casos em que não há fila) ou saída ocorrem é feita uma verificação que irá atribuir o tempo médio de serviço de acordo com o tamanho do pacote. Como os números aleatórios estão no intervalo entre 0 e 1, sempre que o número é menor que 0.5 o tempo médio de serviço atribuído será referente ao pacote de 550. Caso o número esteja entre 0.5 e 0.9 o tempo médio de serviço atribuído será referente ao pacote de 40. E caso o número seja maior que 0.9 o tempo médio atribuído será referente ao pacote de 1500 Bytes.

```
if (fila)
{
    num_aleatorio = aleatorio();
    if (num_aleatorio < 0.5)
    {
        tempo_medio_servico = tempos_medios_servicos[i][0];
        pacote550++;
    }
    else if (num_aleatorio >= 0.5 && num_aleatorio < 0.9)
    {
        tempo_medio_servico = tempos_medios_servicos[i][1];
        pacote40++;
    }
    else if (num_aleatorio >= 0.9)
    {
        tempo_medio_servico = tempos_medios_servicos[i][2];
        pacote1500++;
    }
    servico = tempo_decorrido + tempo_medio_servico;
    soma_tempo_servico += servico - tempo_decorrido;
```

Junto com a verificação para atribuir o tempo médio de serviço também há uma checagem para averiguar que a chegada de cada tipo de pacote atendeu aos pré-requisitos estabelecidos. Os resultados obtidos são extremamente próximos do esperado.

```
float total_pacotes = pacote550 + pacote40 + pacote1500;
printf("A porcentam de pacotes de tamanho 550 Bytes foi igual a: = %lF\n", pacote550 / total_pacotes);
printf("A porcentam de pacotes de tamanho 40 Bytes foi igual a: = %lF\n", pacote40 / total_pacotes);
printf("A porcentam de pacotes de tamanho 1500 Bytes foi igual a: = %lF\n", pacote1500 / total_pacotes);
```

```
A porcentam de pacotes de tamanho 550 Bytes foi igual a: = 0.499694
A porcentam de pacotes de tamanho 40 Bytes foi igual a: = 0.400213
A porcentam de pacotes de tamanho 1500 Bytes foi igual a: = 0.100093
```

3. COLETA DOS DADOS

Com os parâmetros fixados e o tempo médio de serviço para cada cenário já definido, as demais modificações já podem ser implementadas. Primeiramente foi implementado o processo que permite a coleta de dados a cada 100 segundos exatos. Nota-se a importância de definir a coleta em segundos exatos, assim a coleta não será feita necessariamente quando ocorrer uma chegada ou saída, mas em um tempo determinado.

Para permitir essa coleta o tempo decorrido recebia o valor mínimo (em segundos) entre a coleta de dados, o tempo da última chegada e o tempo da última saída. A primeira coleta de dados ocorre no segundo 100 e então a variável correspondente é incrementada a cada coleta.

```
while (tempo_decorrido <= tempo_simulacao)
{
    tempo_decorrido = !fila ? minimo(chegada, coleta_dados) : minimo(minimo(coleta_dados, chegada), servico);
```

```
coleta_dados = 100;
```

```
coleta_dados += 100;
```

A coleta de dados foi realizada para a Ocupação, para as medidas de Little: E[N] e E[W], e para o Erro de Little. Esses dados foram calculados conforme já disponibilizados no código fonte.

Essa coleta foi realizada para cada um dos cenários individualmente. Ao fim de cada cenário as variáveis são “resetadas” e os valores possíveis para o tempo médio de serviço são dados pela próxima linha da matriz.

```
for (int i = 0; i < 4; i++)
{
```

```
if (num_aleatorio < 0.5)
{
    tempo_medio_servico = tempos_medios_servicos[i][0];
    pacote550++;
```

4. GERAÇÃO E ANÁLISE DOS GRÁFICOS

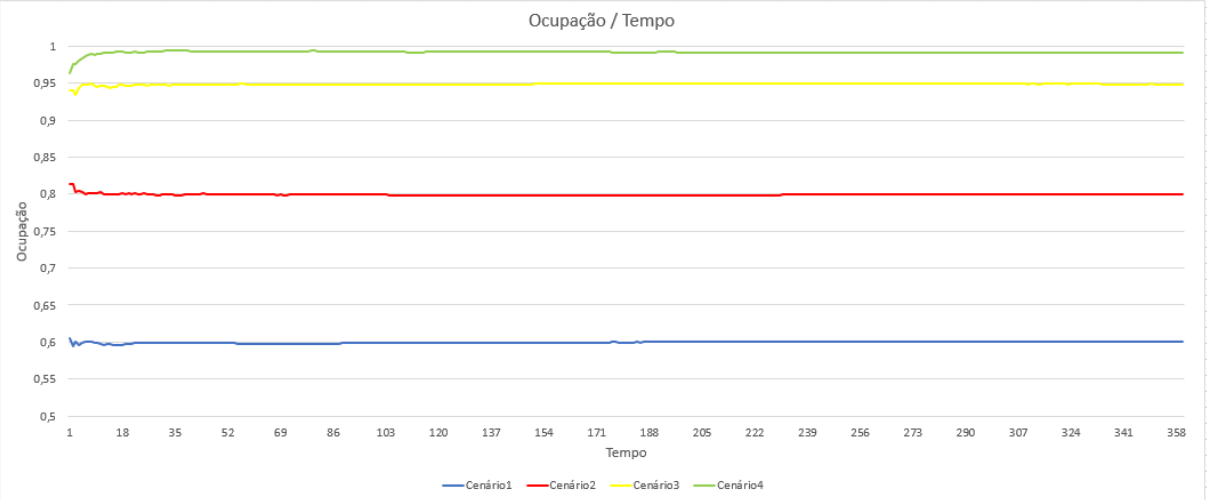
Com os dados já coletados foi necessária uma maneira de poder analisá-los, para isso os dados foram colocados em um arquivo .txt e posteriormente exportados para o Excel, onde os gráficos foram gerados.

```
printf("%d\t%lF\t%lF\t%lF\t%.20lF\n", coleta_dados, soma_tempo_servico / maximo(tempo_decorrido, servico), e_n_final, e_w_final, erro_little );
```

```
$ gcc simulador.c -lm -o simulador
$ ./simulador > simulador.txt
```

4.1. GRÁFICO DA OCUPAÇÃO

Gráfico 1 - Variação da Ocupação com o passar do tempo para os diferentes tempos médios de serviço.

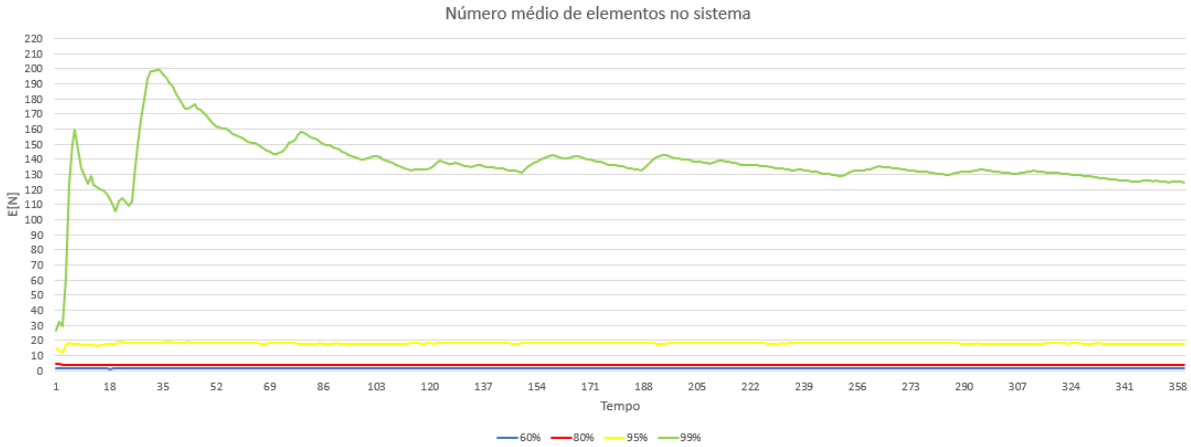


O primeiro gráfico gerado foi o de ocupação pelo tempo. O eixo y representa a taxa de ocupação e o eixo x a passagem de tempo, sendo que cada ocorrência representa uma coleta, ou a passagem de 100 segundos. As linhas representam a variação do tempo médio de serviço para cada cenário. Ao analisar o gráfico é possível perceber que no início da simulação as taxas de ocupação apresentaram uma variação maior, ainda que muito pequena, e de acordo com a passagem de tempo a ocupação alcançou uma estabilidade consideravelmente boa, apresentado resultados semelhantes e próximos dos esperados.

Esse resultado corresponde com o esperado, uma vez que as chegadas e saídas foram geradas de forma aleatória, assim os resultados iniciais são imprevisíveis e não representam um valor concreto e confiável, no entanto, com o decorrer da simulação e o aumento no número de eventos esse número tende a atingir uma média.

4.2 GRÁFICO DO NÚMERO MÉDIO DE ELEMENTOS NO SISTEMA

Gráfico 2 - Número médio de elementos no sistema

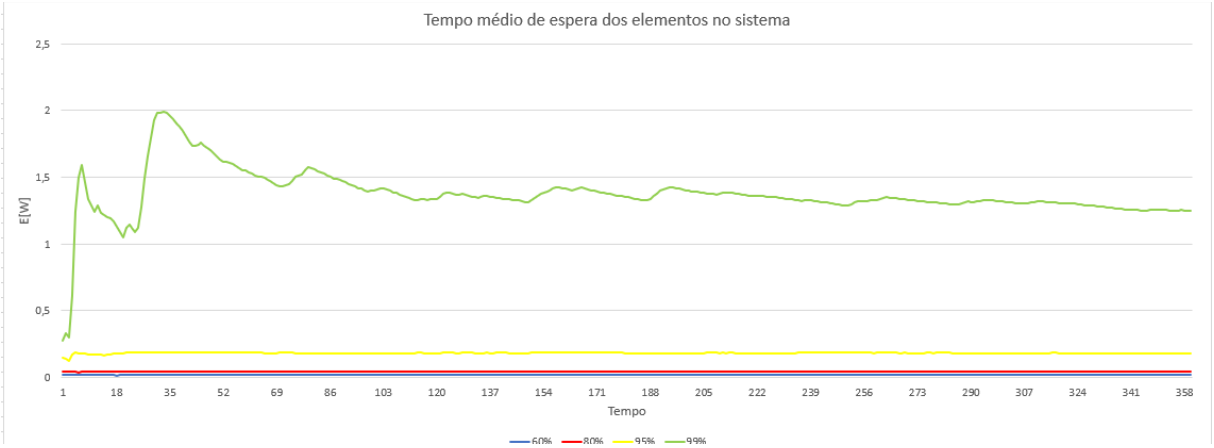


O segundo gráfico gerado foi o de número médio de elementos no sistema. O eixo y representa a quantidade de elementos e o eixo x a passagem de tempo. As linhas representam a variação da ocupação para cada cenário. No início de todas os cenários temos um aumento no número de elementos e com a passagem do tempo uma estabilização ou queda. Esse aumento repentino novamente pode ser explicado pela aleatoriedade do sistema.

Quanto maior a taxa de ocupação maior o número de elementos, uma vez que quanto mais ocupado um sistema está mais propenso ele é a formar filas. Esse comportamento fica explícito no gráfico, as maiores taxas de ocupação apresentaram uma taxa média de elementos mais elevada.

4.3. GRÁFICO DO TEMPO MÉDIO DE ESPERA DOS ELEMENTOS NO SISTEMA

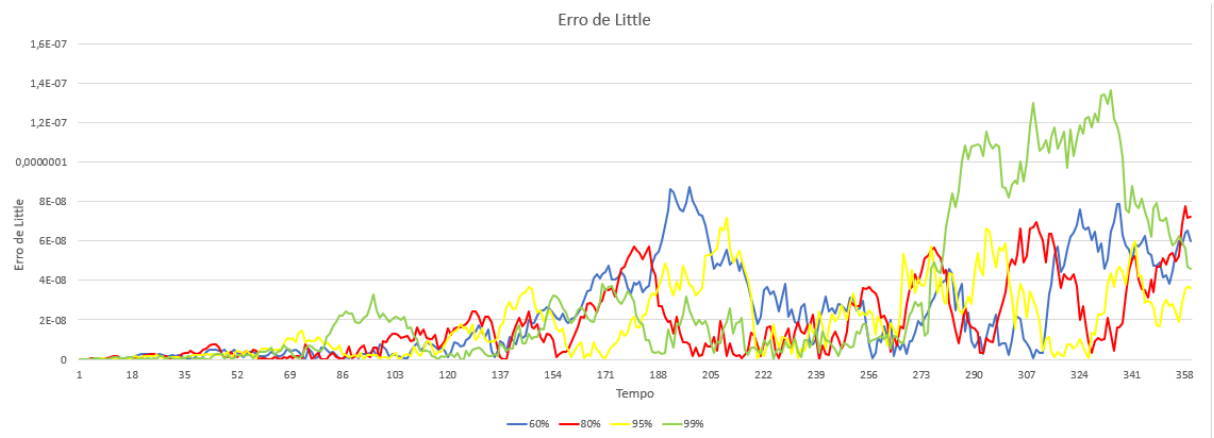
Gráfico 3 - Tempo médio de espera dos elementos no sistema.



O terceiro gráfico gerado foi o do tempo médio de espera dos elementos no sistema. O eixo y representa o tempo médio de espera e o eixo x a passagem de tempo. As linhas representam a variação da ocupação para cada cenário. O gráfico de tempo médio de espera dos elementos está extremamente relacionado com o gráfico de número de elementos no sistema, uma vez que a quantidade de elementos no sistema determina o tempo de espera. Quanto mais elementos em um sistema maior a fila e maior o tempo de espera, e quanto menos elementos menor a fila e o tempo de espera.

4.4. GRÁFICO DO ERRO DE LITTLE

Gráfico 4 - Erro de Little



O quarto gráfico gerado foi o de Erro de Little. O eixo y representa o Erro de Little e o eixo x a passagem de tempo. As linhas representam a variação da ocupação para cada cenário. É possível notar pelo gráfico que no início da simulação o valor do Erro de Little é extremamente próximo de zero e com o passar do tempo esse número tende a aumentar, ainda que não se afaste de zero.

Esse aumento é compreensível em um prazo curto de tempo como o da simulação, no entanto espera-se que quanto maior o tempo de simulação mais próximo de zero esse número se aproxime.

4.5. RESULTADOS FINAIS

Tabela 1 – Resultados finais

Ocupação / Medidas	Ocupação	E[N]	E[W]	Erro de Little
60%	0.600238	1.477128	0.014772	0.00000005995599017261
80%	0.799268	3.910741	0.039132	0.00000007257985012998
95%	0.948814	17.865336	0.178692	0.00000003634060163904
99%	0.991304	124.662499	1.245012	0.00000004562490119042

5. CONCLUSÃO

A simulação permitiu visualizar e analisar os conceitos teóricos apresentados nas aulas da disciplina, fornecendo uma melhor compreensão da teoria assim como uma fixação do aprendizado.

Os resultados, em sua maioria, atenderam ao esperado, apenas os dados obtidos com o Erro de Little não foram suficientes para gerar uma conclusão certeira, uma vez que o comportamento do Erro de Little é melhor visualizado em simulações com longos prazos de tempo, uma vez que quando o tempo tende ao infinito mais próximo de zero o Erro de Little se aproxima. Logo, para gerar dados conclusivos a respeito do Erro de Little seria necessário aumentar o prazo de tempo do simulador.