

Construção de um Modelo de Simulação para Escalonamento de Processos não-preemptivos

Jhonatan T. Cabral Néry¹, Franciny M. Barreto¹, Joslaine C. Jeske de Freitas¹

¹Bacharelado em Ciência da Computação –
Universidade Federal de Goiás - Regional Jataí (UFG - Jataí)
BR 364, km 195, nº 3800 CEP 75801-615 – Jataí – GO – Brazil

{jhonatanthallisson, francinymedeiros, joslaine}@gmail.com

Abstract. *Modern society has increasingly demanded technology requirements, some of them agility, availability and the ability to handle different tasks simultaneously. Thus, one of the researches to meet the current scenario has its focus on the treatment of running applications (so-called processes) that may be in a functioning Operating System. On a computer, the element responsible for handling these processes is the scheduler. This research aims to construct a simulator that specifically shows the behavior of the First-in First-time scheduler in scenarios with one, two and three processors with random duration processes.*

Resumo. *A sociedade moderna tem exigido cada vez mais requisitos da tecnologia, sendo alguns deles agilidade, disponibilidade e a possibilidade de tratar diferentes tarefas simultaneamente. Assim, uma das pesquisas para atender o atual cenário tem seu foco no tratamento das aplicações em execução (os chamados processos) que podem estar em um Sistema Operacional em funcionamento. Em um computador, o elemento responsável por tratar estes processos é o escalonador. Esta pesquisa tem por objetivo construir um simulador que mostre especificamente o comportamento do escalonador First-in First-out em cenários com um, dois e três processadores com processos de duração aleatória.*

1. Introdução

Todo computador possui recursos finitos, sendo isto verdade, todos estão sujeitos a atrasos e escassez de modo geral. Por este motivo, é importante que se utilizem políticas de alocação bem estruturadas e que não se deixe evidente a carência de recursos. O gerenciamento destes computadores atuais é feito pelo Sistema Operacional (SO) e este realiza tarefas complexas e uma delas é o escalonamento de processos, que de acordo com [Maziero 2014] é o mecanismo que aloca recursos de processamento aos programas da máquina, ditando quem usará, em que momento e por quanto tempo.

Atualmente existem várias técnicas de escalonamento diferentes, cada uma delas propondo uma abordagem distinta. Tendo isto em vista, manifesta-se uma pergunta: Quais os possíveis comportamentos de um escalonador e qual sua conduta quando o número de processadores é alterado? Então, este artigo tem como objetivo responder a esta indagação, mostrando através de simulação, como um escalonador First-Come First-Served (FCFS), também chamado de First-In First-Out (FIFO) pode se comportar quando aplicado a cenários distintos e como isto pode contribuir para a otimização do uso do processador.

2. Conceitos

2.1. Sistemas Operacionais

[Deitel et al. 2005, Silberschatz 2009] definem SO como sendo um *software* que possibilita a interação entre as aplicações e o *hardware* de um computador, além disto fornece serviços que permitem que cada aplicação seja executada com segurança e efetividade. [Machado and Maia 2013] ainda reiteram dizendo que aplicações projetadas sobre um SO são mais racionais, pois detêm o foco em sua tarefa central (alto nível de abstração), não precisam se comunicar diretamente com o hardware, o que as torna quase sempre portáteis, além de exigirem menor custo nas etapas de criação e manutenção.

2.2. Redes de Petri

As redes de Petri (ou RdP) surgiram em 1962, por iniciativa de Carl Adam Petri. Redes de Petri são consideradas como uma ferramenta gráfica e matemática de representação formal que permite modelagem, análise e controle de sistemas a eventos discretos que comportam atividades paralelas, concorrentes e assíncronas [Murata 1989]. Os elementos básicos que permitem a definição das redes de Petri são:

- **Lugar:** Representado por um círculo. Um lugar pode ser interpretado como uma condição, um estado parcial, um procedimento, a existência de um recurso, etc..
- **Transição:** Representada graficamente por uma barra ou retângulo. As transições são associadas a eventos que ocorrem em um sistema.
- **Ficha:** É representada por um ponto em um lugar. É um indicador significando que a condição associada ao lugar é verificada, como, por exemplo, um recurso disponível em um certo processo.

As redes de Petri Coloridas foram idealizadas por Kurt Jensen, pesquisador da Universidade de Aarhus, na Dinamarca. Seu artigo, datado do ano de 1981, intitulado *Coloured Petri Nets and the Invariant Method* foi um trabalho promissor, que induziu uma série de estudos sobre coloração das fichas. As redes de Petri coloridas foram projetadas para tornar mais compactos os modelos de sistemas complexos, permitindo a individualização de fichas (usando as cores que lhe são atribuídas). Assim, diferentes processos ou recursos podem ser representados em uma mesma estrutura gráfica.

CPN Tools é uma ferramenta desenvolvida pelo grupo CPN da Universidade de Aarhus (2000-2010), capaz de trabalhar com vários tipos de redes de Petri, como as redes Temporizadas, redes Clássicas e as redes de Petri Coloridas, possibilitando a criação, simulação de modelos e extração de relatórios e probabilidades que determinam se estes modelos são viáveis, ou se são possíveis [Jensen and Kristensen 2009].

3. Implementação

3.1. Definição do Modelo

[Maziero 2014] define o escalonador de processos como sendo o nível mais baixo, complexo e o processo com maior prioridade em um SO. [Tanenbaum 2003] cita diferentes tipos de SO multiprogramados possíveis. Dentre estes, a presente pesquisa buscará mostrar o escalonador FIFO, que se encaixa no grupo dos escalonadores não preemptivos.

Para criar um modelo de simulação será necessário encontrar um cenário de escalonamento base, com o objetivo de retirar alguns dados fundamentais, como tempo de chegada de processos e tempo de execução de cada um deles.

[Stallings 2012] descreve um cenário de escalonamento de cinco processos que, como ilustrado na tabela 1, é composto pelos dados de hora de chegada (tempo de chegada de cada processo) e tempo de serviço (tempo de execução), que são exatamente os dados necessários para dar andamento as simulações. [Stallings 2012] especifica que neste cenário os processos exigem uso alternativo de processador e E/S de forma repetitiva.

Processo	Hora de Chegada	Tempo de Serviço
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Table 1. Exemplo de escalonamento de processo,[Stallings 2012].

Ainda usando este cenário, é possível identificar o tempo médio de chegada e o tempo médio de execução dos processos. Usando a tabela 1 como base, tem-se o seguinte:

- Média chegada = $(0+2+4+6+8)/5 = 4$
- Média execução = $(3+6+4+5+2)/5 = 4$

[Maziero 2014] descreve o algoritmo de escalonamento FIFO apresentado a seguir.

```

01 ALGORITMO FIFO
02   t = 0
03   ENQUANTO (t < tmax), FAÇA
04     PARA cada processo i
05       SE (o processo i chegou agora), ENTÃO
06         coloque o processo i no fim da fila de prontos
07       fim SE
08     fim PARA
09     SE ((o processador estiver livre) E
10       (houver processo na fila de prontos)),
11       ENTÃO
12         encontre o primeiro processo da fila de prontos
13         migre o processo para o estado "executando"
14       fim SE
15     SE (o processo executando chegou ao fim da execução),
16       ENTÃO
17       libere o processador
18     SENÃO
19       SE (o processo solicitou E/S),
20         ENTÃO
21         migre o processo para o estado "bloqueado"
22         libere o processador
23       fim SE
24     fim SE
25     t <-- t+1      /*incrementando o tempo*/
26   fim ENQUANTO
27 fim ALGORITMO

```

A Figura 1 mostra a rede de Petri gerada baseada no funcionamento do escalonador FIFO.

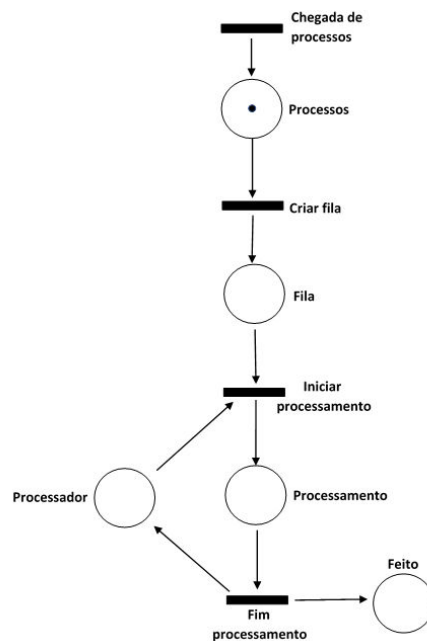


Figure 1. Rede de Petri do escalonador FIFO.

Baseado no cenário proposto por [Stallings 2012] e no algoritmo apresentado por [Maziero 2014] este trabalho propõe construir o algoritmo de escalonamento FIFO utilizando CPN Tools e simulá-lo em diferentes cenários. Esta proposta difere-se do apresentado por [Stallings 2012] pois deseja-se mostrar uma situação mais próxima da realidade. Para tanto, funções exponenciais serão utilizadas para gerar valores aleatórios de tempo de chegada e tempo de serviço de cada processo.

3.2. Desenvolvimento

A figura 2 mostra o modelo de simulação completo do escalonador FIFO, criado na ferramenta *CPN Tools*. As funções utilizadas e definidas como **iat** e **Mtime** são funções prontas da ferramenta e segundo [CPNTools 2018] são definidas pelo seguinte:

- **fun iat(ET) = round (exponential(1.0/ET))** : Cria valores aleatórios levando em conta uma média (ET), neste caso 4.
- **fun Mtime() = IntInf.toInt(time())** : Retorna o tempo atual no modelo como um número inteiro.

Para a simulação do funcionamento do escalonador FIFO com 2 e 3 processadores o lugar *Processador* recebeu o valor 2() e 3() respectivamente, demonstrando que o mesmo trabalhará com mais de um processo por vez.

O *CPN Tools* possui uma ferramenta eficiente para a observação do comportamento de modelos simulados, os chamados monitores. Foram criados neste trabalho os monitores **Tamanho da Fila**, **Tempo de Espera** e **Tempo de Processamento** para observar o comportamento do escalonador FIFO.

4. Análise dos resultados

À partir das simulações geradas pela ferramenta *CPN Tools* é possível identificar vários aspectos da execução e suas nuances quando alterado o número de processadores.

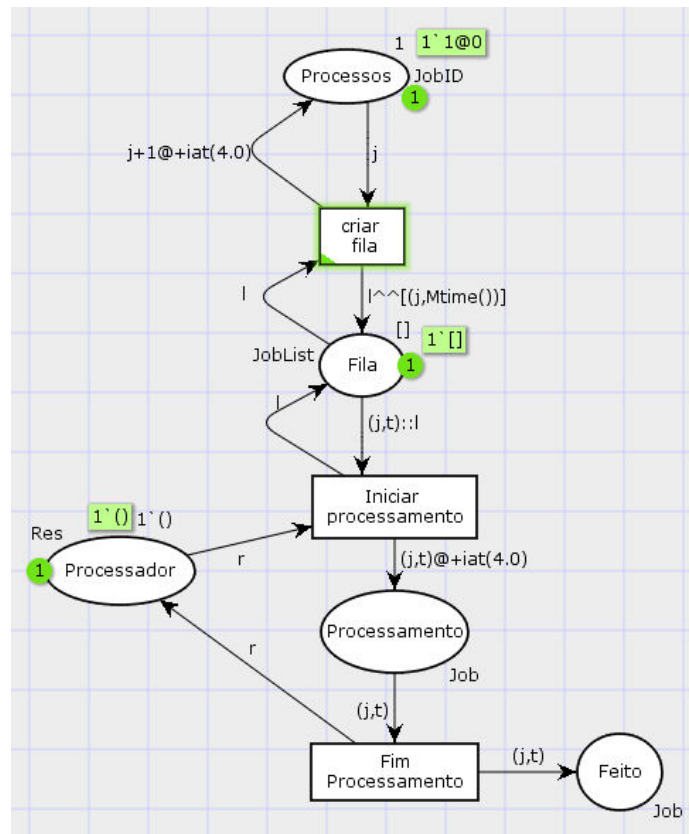


Figure 2. Modelo de simulação completo do escalonador FIFO.

Com o monitor **Tamanho da Fila** foi possível observar que a média de elementos na fila diminui substancialmente com o aumento do número de processadores.

O monitor **Tempo de Espera** mostra que no modelo com 1 processador esta espera penaliza as aplicações, sendo em média 178,585786 unidades de tempo o que cada processo aguarda na fila. O modelo com 2 processadores já mostra uma redução deste tempo em 99,24% e com 3 processadores a redução é de 99,90% em relação ao modelo com 1 processador.

Por fim o monitor **Tempo de Processamento** que visa expressar o tempo total que cada processo gastou no escalonador (tempo de espera + tempo de execução), consegue abstrair conhecimentos bastante pertinentes, já que o tempo total de processamento é fator determinante para avaliar uma arquitetura de escalonamento. Nestas estatísticas é possível ver que quanto menos processadores, mais sobrecarregado o sistema fica, gerando grandes filas e com isto uma espera proporcional, o que faz com que o tempo total de processamento seja maior.

O monitor em questão mostra os seguintes valores: O tempo médio total de processamento com 1 processador é de 182,566413 unidades de tempo, com 2 esta média fica em 5,33624 unidades e se colocando 3 processadores para trabalhar em paralelo este número decresce para 4,150723 unidades de tempo. Sabendo estas informações é possível alegar que o cenário menos atrativo é o de 1 processador, o de 2 e 3 processadores quase se equiparam em eficiência, à vista que o modelo de 2 processadores é 97,08% mais

rápido que o modelo com 1 e o modelo de 3 processadores é pouco diferente, ficando com 97,73% de rapidez a mais que o de 1 processador apenas.

5. Considerações Finais

Sabendo da diferença próxima de desempenho do escalonador FIFO com 2 e 3 processadores é possível concluir que com o aumento da quantidade de processadores, esta diferença de performance tenderá a diminuir e a ser cada vez menos significativa dentro de um escalonador FIFO. Com isto indivíduos e instituições podem usar estes dados para comparação com outros escalonadores e também para decidir a respeito do melhor custo/benefício ante a questão quantidade de processadores *versus* ganho de desempenho.

Sobre a ferramenta *CPN Tools*, pode-se afirmar que esta se mostrou maleável e prática, conseguindo implementar elementos teoricamente complexos de forma simples e isto foi decisivo para a realização deste trabalho. Quanto a visão do poder computacional das simulações para explicar e auxiliar na análise de situações e arquiteturas, é correto dizer que este método é eficaz quando se consegue gerar um modelo assertivo. Assim sendo, as simulações conseguem fornecer informações precisas e relevantes sobre situações hipotéticas ou reais, dando plenas condições para a tomada de decisão.

6. Trabalhos futuros

Algumas sugestões de trabalhos futuros vem de uma das vertentes que não foi abordada aqui, que é a comparação entre o escalonador FIFO e outros escalonadores. Onde a ferramenta *CPN Tools* poderia facilmente gerar relatórios e provocar discussões construtivas a respeito do melhor escalonador para determinados cenários. Trabalhos posteriores podem também explorar simulações com um número maior de processadores, podendo talvez levar a novas descobertas sobre escalonamentos. Pode-se também usar a metodologia exposta aqui para criar novas abordagens que tratem da eficiência do uso de processadores e da relação custo x benefício que estas novas abordagens podem oferecer.

References

- CPNTools (2018). Documentation - <http://cpntools.org/2018/01/16/documentation-2/>.
- Deitel, H., Deitel, P., and Choffnes, D. (2005). *Sistemas Operacionais*. PRENTICE HALL BRASIL.
- Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri Nets*. Springer, Germany.
- Machado, F. B. and Maia, L. P. (2013.). *Arquitetura de sistemas operacionais*:. LTC,, Rio de Janeiro-RJ:, 5.ed. edition.
- Maziero, C. A. (2014). Sistemas operacionais: Conceitos e mecanismos. *Livro aberto*.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.
- Silberschatz, A. (2009). *Operating System Concepts*. John Wiley & Sons Software, 8th edition.
- Stallings, W. (2012). *Operating Systems: Internals and Design Principles*. Prentice Hall.
- Tanenbaum, A. (2003). *Sistemas Operacionais Modernos*. Prentice-Hall do Brasil.