

Servidor Esporádico Dinâmico

Francisca Beatriz Precebes da Silva¹, Johnny Marcos Silva Soares¹, José Robertty de Freitas Costa¹

¹Campus Quixadá - Universidade Federal do Ceará (UFC)

{be.precebes, johnnymarcos1998, roberttyec}@gmail.com

1. Introdução

Este trabalho visa apresentar a técnica de escalonamento *Dynamic Sporadic Server*, também conhecido como DSS e que pode ser traduzido como Servidor Esporádico Dinâmico. Essa técnica se assemelha bastante com a técnica *Sporadic Server* e surgiu da necessidade de se trabalhar com políticas de prioridades dinâmicas, mais especificamente com o EDF [Farines et al. 2000].

Quando falamos de servidor, nos referimos a uma fila de tarefas que ficará responsável por encaixar as tarefas aperiódicas que chegam na fila de execução. Dessa forma o servidor precisa ser visto pelo sistema como uma tarefa adicional.

O *Sporadic Server*, ou simplesmente SS, técnica de escalonamento precursora ao DSS, permite o melhoramento do tempo médio de resposta das tarefas aperiódicas sem degradar o limite de utilização do conjunto de tarefas periódicas. Ele cria uma tarefa de alta prioridade para atender solicitações aperiódicas e mantém a capacidade do servidor em um nível de alta prioridade até que ocorra uma solicitação aperiódica. O SS, diferente de outras técnicas de escalonamento, reabastece sua capacidade apenas após ter sido consumido pela execução de tarefas aperiódicas [Spuri and Buttazzo 1996].

O trabalho está dividido em 5 seções, onde a presente seção visa apresentar a técnica DSS. A segunda seção trata da definição da técnica e, em seguida, na terceira seção, apresentamos o teste de escalonabilidade para esta técnica. Por fim, na quarta seção trazemos um exemplo da técnica apresentada.

2. Definição

Semelhante a outros servidores existentes, tratamos C_s como capacidade e P_s como o período de execução. Uma das características mais relevante do Servidor Esporádico Dinâmico é que a sua capacidade só é consumida depois da ativação de uma tarefa aperiódica e a capacidade não é restabelecida por completo após o período do servidor, mas somente quando ele foi consumido, desta forma tenta-se usar o consumo máximo do processador [Spuri and Buttazzo 1996].

Como já foi citado, a técnica de escalonamento que estamos apresentando é uma extensão da técnica *Sporadic Server* e podemos citar como a principal diferença entre as duas técnicas o fato de o servidor no DSS apresentar a prioridade dinâmica, ou seja, a prioridade das tarefas no sistema (incluindo a tarefa servidor) serem atribuídas ao longo da sua execução através de um prazo adequado.

A atribuição do deadline e a capacidade de reposição são definidas pelas seguintes regras:

- Quando o servidor é criado, C_s é inicializado no seu valor máximo.
- O próximo tempo de reposição (RT, do inglês *replenishment time*) e o deadline atual do servidor d_s são definidos assim que $C_s > 0$ e haja um pedido aperiódico pendente. Se t_A for esse tempo, então $RT = d_s = t_A + P_s$.
- A quantidade de reposição (RA, do inglês *replenishment amount*) a ser feita no tempo RT é calculada quando o último pedido aperiódico é concluído ou C_s foi esgotado. Se t_I for esse tempo, então RA é ajustado igual à capacidade consumida dentro do intervalo $[t_A, t_I]$.

Abaixo está a descrição dos estados e das transições possíveis de um escalonamento que utilize DSS apresentada em [Spuri and Buttazzo 1996].

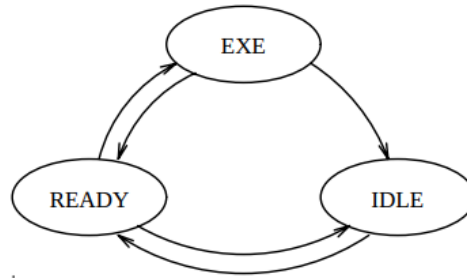


Figura 1. Modelo de transições

A transição de Idle (ocioso) para Ready (pronto) acontece quando chega uma ativação de tarefa aperiódica e $C_s > 0$ ou quando temos o restabelecimento de C_s e uma tarefa já tinha sido ativada. A transição contrária acontece quando o servidor possui a maior prioridade do sistema mas não pode executar por não possuir tarefas aperiódicas ativas.

Outra transição é do estado Ready para Exe (executando), que acontece quando o servidor possui a maior prioridade do sistema, tarefas aperiódicas ativas e capacidade de execução disponível. A transição contrária ocorre quando o servidor não é mais a tarefa mais prioritária do sistema, ou seja, o servidor irá sofrer uma preempção.

Por fim, a transição entre Exe e Idle acontece quando terminamos de executar a tarefa aperiódica e não possui outra tarefa aperiódica ativa ou a capacidade do servidor é esgotada, ou seja, $C_s = 0$.

3. Teste de Escalonabilidade

Para sabermos se o conjunto de tarefas é executável ou não, temos que atender a seguinte condição:

$$U_p + U_s \leq 1 \quad (1)$$

onde U_p é a taxa de utilização de tarefas periódicas dada por:

$$U_p = \sum_{i=1}^n \frac{C_i}{P_i} \quad (2)$$

E U_s é dado por:

$$U_s = \frac{C_s}{P_s} \quad (3)$$

Esse teste de escalonabilidade lembra muito ao teste de escalonabilidade do EDF. A única diferença que podemos citar é que adicionamos a taxa de utilização do nosso servidor.

4. Exemplo

Abaixo está descrito um exemplo de um escalonamento utilizando o *Dynamic Sporadic Server* apresentado em [Farines et al. 2000]. A intenção é exemplificar melhor a técnica estudada.

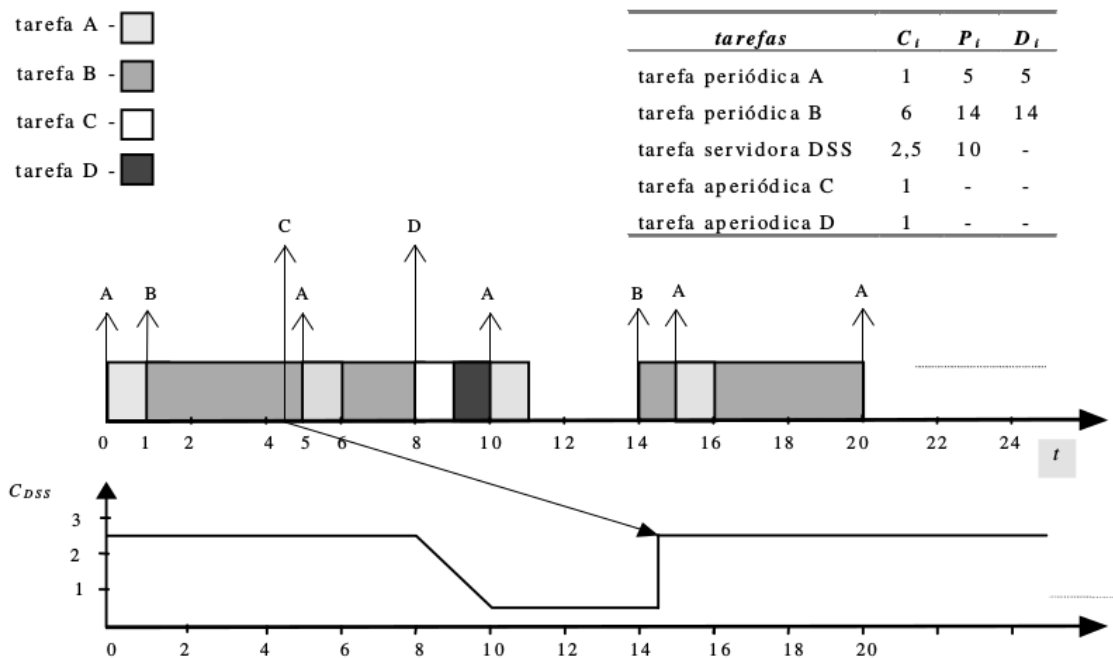


Figura 2. Exemplo de escalonamento DSS

No canto superior direito da figura 2 vem apresentando a tabela que contém o tempo de computação e o período das tarefas. Observe que o servidor de tarefas aperiódicas também é encarada como mais uma tarefa do sistema. Em $t = 0$ temos uma ativação da tarefa A e seu d_s será 5, como só tem ela na fila de execução, ela será executada e em $t = 1$ terá cumprida a sua execução. Neste momento tem uma ativação de B e seu d_s será 14 (o exemplo apresenta um erro com deadline da tarefa B). Como só existe a tarefa B ativa ela fica em execução até $t = 4,5$.

Em $t = 4,5$ existe uma ativação do servidor DSS (devido a ativação da tarefa C), porém o $d_s = 14,5$ e B não foi finalizado e possui maior prioridade nesse momento, logo B continua sua execução até $t = 5$. Neste instante ocorre uma ativação da tarefa A e seu $d_s = 10$, logo ele passará a ser a tarefa mais prioritária do sistema e irá executar até $t = 6$ que

é quando o seu tempo de computação finaliza. De $t = 6$ a $t = 8$ a tarefa mais prioritária é a B então ela irá executar e terminar o seu tempo de computação neste período.

No instante $t = 8$, temos a ativação da tarefa aperiódica D. Como a tarefa C chegou primeiro ela irá executar até $t = 9$ e de $t = 9$ a $t = 10$ teremos a execução da tarefa D. Observe nos instantes de $t = 8$ até $t = 10$ C_{DSS} diminui o seu valor até que seja restaurado em $t = 14,5$. No instante $t = 10$ temos uma ativação da tarefa A e como só tem esta tarefa ativa ela será executada até $t = 11$. De $t = 11$ até $t = 14$ não temos nenhuma tarefa ativa. A partir de $t = 14$ não temos mais ativações de tarefas aperiódicas então a execução irá ser realizada como no escalonamento utilizando EDF.

Referências

- Farines, J.-M., Fraga, J. d. S., and Oliveira, R. d. (2000). Sistemas de tempo real. *Escola de Computação*, 2000:201.
- Spuri, M. and Buttazzo, G. (1996). Scheduling aperiodic tasks in dynamic priority systems. *Real-Time Systems*, 10(2):179–210.