Relatório de Simulação com Política de Escalonamento EDF em Sistemas Multiprocessados

Luciano Pinheiro dos Santos.
Tiago Mesquita de Araujo Cunha.

16 de outubro de 2014

Data: 16 de outubro de 2014
Instrutor: Prof.ª Flávia Maristela

1 Introdução

Sistemas multiprocessadores passaram a fazer parte do nosso cotidiano, já sendo difícil encontrar computadores monoprocessados à venda. O desenvolvimento de sistemas para computadores com multiprocessadores requer novos paradigmas no desenvolvimento de soluções. Para obter um melhor aproveitamento desses computadores e uso efetivo de múltiplos núcleos, novas técnicas de escalonamento e divisibilidade de tarefas passaram a ser adotadas, conforme aponta Luciano Bertini (2004).
Os sistemas de tempo real também foram afetados pela adoção de multiprocessadores. Em virtude da grande variedade de processadores *multithread* possibilitando a execução de tarefas com técnicas como o *pipeline*, algumas abordagens foram propostas para escalonar os processos. Uma destas abordagens define um escalonamento global: Neste modelo, um único escalonador gerencia todos os processos que, por sua vez, estão dispostos em uma fila (também única) e executados de acordo as prioridades definidas, sendo que é permitido a um processo migrar de um processador para outro. Deste modo, partes de um único processo (*threads*) podem ser executados em diferentes processadores de acordo a disponibilidade e a gerência do escalonador.
Em um cenário em que o sistema é monoprocessado, compete ao escalonador a preocupação com os atributos das tarefas (custo, *deadline*, período) e as preempções (quando houver), de acordo a política de escalonamento adotada. Ao enveredar para um ambiente com multiprocessamento, outras atribuições são adicionadas às atividades do escalonador, o que requer preocupação com alguns aspectos não considerados quando existe um único processador, como o fato de haver memória compartilhada (memória *cache*), barramentos que interligam os processadores, o custo computacional com interrupções e troca de tarefas entre

os processadores, dentre outros. Logo, uma das principais metas desses algoritmos para definição de políticas de escalonamento em sistemas multiprocessados consiste em diminuir as preempções e otimizar o tempo de execução global das tarefas do sistema. Starke (2012)
Neste trabalho iremos abordar alguns aspectos dos sistemas de tempo real e realizar experimentos com foco na política de escalonamento G-EDF (*Global-Earliest Deadline First*), que traduz-se na política EDF, utilizada em sistemas monoprocessados, aplicada em sistemas multiprocessados.
Esse trabalho é formado por 6 seções. Na seção II, apresentada abaixo, são abordados cinco artigos relacionados a área. Na seção III é feita a contextualização do problema, abrangendo o referencial teórico apresentado na seção II. Na seção IV são realizados experimentos com a política de escalonamento G-EDF em um simulador denominado SimSO, escolhido para uso neste trabalho. Na seção V são apresentados os resultados e é feita a análise do comportamento e valores encontrados. Na seção VI é apresentada a conclusão.

1.1 Definições

Earliest Deadline First Política de escalonamento adotada em alguns sistemas de tempo real onde as tarefas com menor *deadline* absoluto são executadas primeiro.

2 Estado da Arte

Foram selecionados cinco trabalhos correlatos, usados como referencial teórico para esse documento. Os trabalhos selecionados possuem características específicas abordadas na área de sistemas de tempo real com foco em sistemas multiprocessados e estão listados na subseção abaixo:

2.1 Trabalhos relacionados

Um Breve Survey: Escalonamento em Sistemas de Tempo Real com Otimização do Consumo de Energia

Apresentado em Luciano Bertini (2004) apresenta um *survey* sobre métodos e políticas para economia de energia em sistemas de tempo real. Apresenta os sistemas *power-aware*, como aqueles que conseguem reduzir o seu consumo em tempo de execução e os *low-power* como aqueles de projeto específico, objetivando o melhor consumo sem perda de desempenho. Outras questões relacionadas a redução de energia ainda são abordadas ao longo desse trabalho.

Uma Abordagem De Escalonamento Heterogêneo Preemptivo E Não Preemptivo Para Sistemas De Tempo Real Com Garantia Em Multiprocessadores

Na dissertação apresentada em Starke (2012) o autor aborda criteriosamente diversos aspectos dos sistemas de tempo real em ambientes multiprocessados. Dentre os diversos aspectos destacados no trabalho de dissertação foi utilizado nesse trabalho a abordagem do autor aos diversos mecanismos de escalonamento para esses sistemas, como: escalonamento particionado; global; híbrido.

A Survey of Hard Real-Time Scheduling for Multiprocessor Systems.

Neste *survey*, Robert I. Davis (2011) apresenta alguns algoritmos de escalonamento em tempo real críticos (*hard*), bem como técnicas de análise de escalonabilidade para sistemas com múltiplos processadores homogêneos. Os autores focam nos principais resultados neste campo desde as suas origens na década de 1960 para uma pesquisa publicada no final de 2009. São apresentadas as diferentes características e métodos de alogoritmos de escalonamento, sendo consideras as diversas métricas de desempenho que podem ser usadas para fins de comparação. Uma análise detalhada é fornecida destacando os tipos de escalonadores (particionado, global, híbridos) e os mais recentes resultados de pesquisas empíricas de compartilhamento de recursos. Ao final, são propostas questões a serem solucionadas e os principais desafios de pesquisa na área.

Techniques for Multiprocessor Global Schedulability Analysis.

Em Baruah (2011) o autor faz uma abordagem a diferentes agendadores, focando na utilização do *Global EDF*, também apresenta uma modificação no funcionamento do *Global EDF* a fim de solucionar um problema específico, tratado no artigo. O Autor também realiza testes com o agendador *Global EDF*.

Efficiency Assessment of Job-level dynamic Scheduling Algorithms on Identical Multiprocessors. Em VAHID SALMANI and NEJAD (2010) Os autores fazem uma abordagem aos diferentes agendadores possíveis para o escalonamento de tarefas. Os autores também abordam os aspectos do agendamento em sistemas multiprocessador e realizam uma discussão entre política de escalonamento EDF e LLF. Os autores ainda fazem um experimento onde analisam tempo de resposta e outros aspectos de um sistema de tempo real.

3 Contextualização do Problema

Um sistema passa a ser considerado como sistema de tempo real (STR) quando, além de executar suas tarefas de maneira logicamente correta, atende às restrições de temporalidade. Ao atender estas restrições, o sistema torna-se previsível, sendo a previsibilidade outra característica que distingue os sistemas convencionais de STRs. O que faz, por exemplo, um sistema de controle da malha ferroviária ser considerado um STR não é o fato deste ser rápido ou não: A execução das tarefas no instante planejado garante categorização deste sistema como de tempo real.
Garantir que sistemas críticos como o de controle de uma malha ferroviária, das aeronaves, de equipamentos médico-hospitalar para monitoramento de pacientes tenham previsibilidade, exige o entendimento dos atributos de uma tarefa, bem como das políticas de escalonamento para gerenciá-las. Estes atributos e seus conceitos são apresentados na tabela a seguir.

ATRIBUTOS	**CONCEITO**
Período (T_i)	Intervalo regular de ativação da tarefa.
Deadline (D_i)	Tempo de vida (validade) da tarefa.
Custo (C_i)	Duração mínima de uma tarefa.

Tabela 1: Atributos de uma tarefa.

Há outras propriedades inerentes às tarefas que devem ser destacadas, segundo Rômulo S. de Oliveira (2000), como as que dizem respeito a criticidade e a regularidade de ativação, elencadas a seguir:

- Críticas (tarefas "hard"): Uma tarefa é dita crítica quando, ao ser completada depois de seu *deadline*, pode causar falhas catastróficas no sistema de tempo real e em seu ambiente;
- Brandas ou Não Críticas (tarefas "soft"): Essas tarefas, quando se completam depois de seus *deadlines*, no máximo implicam numa diminuição de desempenho do sistema;
- Periódicas: Quando as ativações do processamento de uma tarefa ocorrem, numa sequência infinita, uma vez por período;
- Aperiódicas ou Assíncronas: Quando a ativação ocorre em tempos irregulares (períodos não previstos);
- Esporádica: São tarefas aperiódicas com característica de execução em um intervalo mínimo entre duas ativações consecutivas e conhecidas.

Entender as propriedades inerentes às tarefas possibilita melhor compreensão e definição das políticas de escalonamento adotadas no gerenciamento das execuções

destas no sistema. Para estas políticas, também foram associados atributos, resumidos na quadro comparativo abaixo, onde os campos da coluna nomeada TIPO 1 são características opostas à coluna TIPO 2:

TIPO 1	**TIPO 2**
Otimizada: quando o algoritmo provê uma solução ótima.	Heurística: por que não existe uma solução ótima.
Preemptiva: a tarefa pode ser interrompida por um processo de maior prioridade.	Não-Preemptiva: a tarefa não sofre interrupção. Executa até ser concluída.
Estáticos: as decisões das tarefas são baseadas em parâmetros fixos.	Dinâmicos: as decisões das tarefas são baseadas em parâmetros que mudam durante a evolução do sistema.
Offline: as decisões são definidas antes da execução da tarefa.	*Online*: as decisões são tomadas em tempo de execução, quando uma nova tarefa chega ou outra ativa é concluída.

Tabela 2: Quadro comparativo dos atributos de um escalonador.

As características até então apresentadas são de grande relevância quando trata-se de sistemas monoprocessados. Com o multiprocessamento, além da preocupação em garantir a execução das tarefas respeitando seu *deadline*, sejam estas periódicas ou aperiódicas, as políticas de escalonamento visam contemplar soluções para outros problemas.
Sejam tais políticas preemptivas ou não, estáticas ou dinâmicas, no cenário em que há a utilização de mais de um processador, o fato de existir uma memória compartilhada pode ter um impacto significativo sobre a execução de outra tarefa (surge uma região crítica). Além disto, o custo computacional com a troca de *threads* entre processadores (considerando também os barramentos) e as possíveis preempções geradas por estas ações, tornam-se mais um fator de diferenciação de algoritmos escalonadores para sistemas de tempo real multiprocessados.
Robert I. Davis (2011) destacam características a serem consideradas acerca dos sistemas multiprocessados:

- Heterogêneos: Os processadores são diferentes, fazendo com que a taxa de execução de uma tarefa dependa tanto do processador quanto da tarefa. O impacto neste caso, é que nem todas as tarefas podem ser capazes de executar em todos os processadores;
- Homogêneos: Os processadores são idênticos, garantindo a mesma taxa de execução das tarefas em todos os processadores;
- Uniformes: A taxa de execução de uma tarefa depende apenas da velocidade do processador.

Diversos algoritmos de escalonamento globais foram propostos visando reduzir custos computacionais gerais, limitando assim, a quantidade de preempções. Neste experimento foi utilizada política G-EDF com o intuito de analisar e comparar o número de preempções e migrações em função do número de tarefas e a quantidade de processadores para o cenário exposto na seção subsequente.

4 Experimento e Simulação

Para a realização do experimento foi utilizado o simulador SimSO (2014), no qual é possível programar um agendador e analisar seu comportamento em diferentes cenários, variando em tarefas, processador e parâmetros possíveis de serem estabelecidos. Este trabalho não aborda os aspectos de utilização do simulador, mas é feita uma breve análise do algoritmo utilizado para o agendador estudado nesse trabalho, o Global EDF.

4.1 Análise de algoritmo do Global EDF

No código demonstrado na figura 1 foi utilizado a política de escalonamento Global EDF. Nessa política o uso de múltiplos processadores é contemplado.
Na execução do código escrito em *Python* na figura supracitada existem quatro funções para o funcionamento do agendador. Na função "*init*" o agendador é inicializado com a sua lista de tarefas zerada. Nas funções "*on_activate*" e "*on_terminated*" o agendador notifica o processador um novo evento, no primeiro caso de início e no segundo de término de sua execução. Na função "*schedule*" o agendador faz o processo interno para escalonamento das tarefas.
No código de escalonamento *Global EDF* utilizado, conforme demonstrado em código, o algoritmo segue a lógica de buscar por processadores livres e então atribuir uma tarefa baseada naquela que tiver o menor *deadline* absoluto. Caso não existam processadores disponíveis o agendador então localiza, dentre os processadores em utilização, aquele que contém uma prioridade inferior a tarefa que precisa ser executada para realizar a preempção dela em detrimento da atividade de maior prioridade.

```
from simso.core import Scheduler

class G_EDF(Scheduler):
    def init(self):
        self.ready_list = []

    def on_activate(self, job):
        self.ready_list.append(job)
        # Send a "schedule" event to the processor.
        job.cpu.resched()

    def on_terminated(self, job):
        # Send a "schedule" event to the processor.
        job.cpu.resched()

    def schedule(self, cpu):
        decision = None # No change.

        if self.ready_list:
            # Look for a free processor or the processor
            # running the job with the least priority.
            key = lambda x: (1 if not x.running else 0,
                             x.running.absolute_deadline if x.running else 0)
            cpu_min = max(self.processors, key=key)

            # Obtain the job with the highest priority within the ready list.
            job = min(self.ready_list, key=lambda x: x.absolute_deadline)

            # If the selected job has a higher priority
            # than the one running on the selected cpu:
            if (cpu_min.running is None or
                    cpu_min.running.absolute_deadline > job.absolute_deadline):
                self.ready_list.remove(job)
                if cpu_min.running:
                    self.ready_list.append(cpu_min.running)
                # Schedule job on cpu_min.
                decision = (job, cpu_min)

    return decision
```

Figura 1: *Global Earliest Deadline First scheduling* utilizado na simulação.Maxime Cheramy and Deplanche (2014)

4.2 Experimento Realizado

Nessa experimentação será suposto um sistema de fechamento de válvulas para uma hidrelétrica. Partindo do suposto que o sistema da hidrelétrica analisado possui seis válvulas de fechamento e abertura para passagem de água que são distintas, cada válvula possui como possibilidade de tarefa a sua abertura e o seu fechamento. Neste trabalho será abordado apenas a atividade de seu fechamento. Serão analisadas seis tarefas de fechamento onde T1 = Tarefa de fechamento de válvula 1 até T6, onde T6 = Tarefa de fechamento de válvula 6.

TAREFA	C_i	T_i	D_i
T1	5	10	10
T2	15	30	30
T3	15	25	25
T4	10	20	40
T5	5	40	45
T6	20	40	50

Tabela 3: Tabela de tarefas do experimento e seus valores.

Na tabela 3 é possível visualizar as tarefas, assim como seu custo associado em WCET e seu período de *deadline*.
Na primeira experimentação foi realizada a execução do experimento com 2 processadores, o gráfico *Gantt* gerado pode ser visualizado na figura 2.

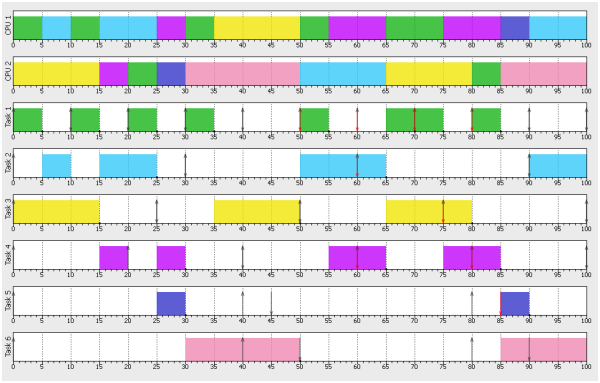

Figura 2: Gráfico *Gantt* de execução com dois processadores.

No segundo estudo de simulação foi adicionado mais um processador, totalizando assim três processadores. O resultado em gráfico *Gantt* dessa simulação pode ser visualizado no figura 3.

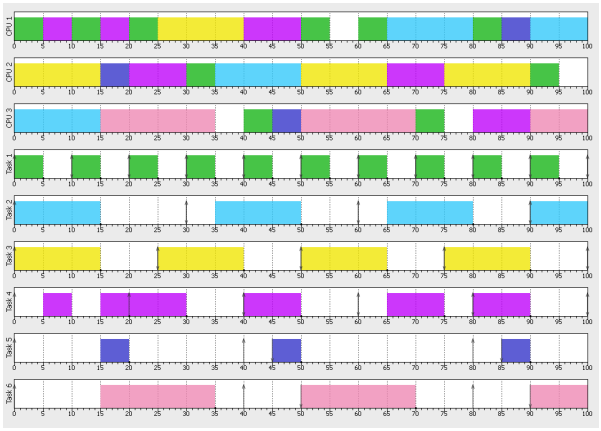

Figura 3: Gráfico *Gantt* de execução com três processadores.

5 Análise de Resultados

Conforme já apresentado nos gráficos *Gantt* anexo, o sistema não obteve um escalonamento ideal sendo executado com dois processadores. Nesse cenário ocorreria a perda de *deadlines* nas tarefas T1, T2, T3 e T4, onde seus tempos máximos respectivamente apresentados foram de: 20, 35, 30 e 45, ultrapassando seus deadlines previstos. Apesar do uso de 100% de ambos os processadores o simulador nesse cenário não conseguiu estipular a possibilidade de escalonamento com a política Global EDF, conforme visualizado na figura 2.

Em uma situação real esse sistema poderia se tornar perigoso no momento em que 4 válvulas teriam relativo atraso no seu fechamento, sendo assim poderia ocorrer o desperdício de água ou até mesmo a inviabilidade de operação da hidrelétrica.

Em um segundo experimento foi adicionado um novo processador na execução, deixando o sistema com três processadores com o objetivo de resolver o problema de escalonamento com cumprimento dos deadlines previstos. Nesse novo cenário foi observado através do gráfico *Gantt* exibido na figura 3 e dos resultados gerados da simulação, que o sistema passa a ser escalonado com a utilização média de 90% dos processadores, possibilitando também a adoção futura de técnicas de tolerância a falha.

Nesse cenário foi possível observar que todos os *deadlines* foram cumpridos com a ocorrência de preempção nas tarefas T2, T3, T4 e T6.

6 Conclusão

A apresentação desse estudo demonstra como um algoritmo de escalonamento pode apresentar diferentes resultados e como a utilização da simulação pode-se antever possíveis problemas na implementação de sistemas de tempo real.
No estudo apresentado podemos verificar que a utilização de dois processadores no cenário observado não seria ideal, gerando inconsistência e falha no sistema com a perda do *deadline* específico de algumas das tarefas. Apesar do objeto de estudo desse trabalho ter sido o algoritmo de escalonamento *Global EDF* é válido também citar que outros algoritmos de escalonamento para sistemas multiprocessados poderiam ser utilizados para atingir o objetivo do sistema ainda que com apenas dois processadores.

Referências

Baruah, S. (2011). Techniques for multiprocessor global schedulability analysis.

Luciano Bertini, J. L. (2004). Um breve survey: Escalonamento em sistemas de tempo real com otimização do consumo de energia. WTR2004.pdf.

Maxime Cheramy, P.-E. H. and Deplanche, A.-M. (2014). Simso - a simulation tool to evaluate real-time multiprocessor scheduling algorithms.

Robert I. Davis, A. B. (2011). A survey of hard real-time scheduling for multiprocessor systems. Journal ACM Computing Surveys.

Rômulo S. de Oliveira, Joni S. Fraga, J.-M. F. (2000). Sistemas de tempo real. livro-tr.pdf.

SimSO (2014). Simso - simulation of multiprocessor scheduling with overheads.

Starke, R. A. (2012). Uma abordagem de escalonamento heterogêneo preemptivo e não preemptivo para sistemas de tempo real com garantia em multiprocessadores.

VAHID SALMANI, MAHMOUD NAGHIBZADEH, A. T. M. B. and NEJAD, S. K. (2010). Efficiency assessment of job-level dynamic scheduling algorithms on identical multiprocessors.