GSPAM: Um Escalonador para Sistemas Operacionais de Tempo Real em Ambientes Multicore

Vitor P. David[1], **Douglas C. de Araujo**[1], **Danyel M. do Nascimento**[1], **Juliana M. N. S. Zamith**[1]

[1]Universidade Federal Rural do Rio de Janeiro

{vitor.pito, custodiodouglas6, matiasnascimento4496}@gmail.com

julianananascente@gmail.com

Resumo. *Sistemas Operacionais de Tempo Real devem garantir a execução das tarefas (ou processos) em um deadline previamente definido para que se obtenha o comportamento temporal desejado. Estes sistemas têm aplicação em diversas áreas como, por exemplo, em sistema de controle de voos, temperatura, entre outros. Neste trabalho é proposto um escalonador denominado GSPAM, capaz de escalonar tarefas críticas e não críticas que considera os múltiplos núcleos da máquina.*

1. Introdução

Sistemas Operacionais de Tempo Real (SO-TR) possuem grande importância no mercado e tendem a receber ainda mais importância, devido ao crescimento de diversas áreas como, por exemplo, a IoT (Internet of Things) [Jayavardhana Gubbi 2012]. Este crescimento traz ainda grandes desafios para área SO-TR, uma vez que estes sistemas computam tarefas complexas com *deadlines* predefinidos e executam em sistemas com limitação de recursos (bateria, memória, etc). Algoritmos de escalonamentos eficientes para SO-TR poderiam contribuir para garantir a execução da tarefa no tempo desejado, evitando o uso excessivo dos recursos limitados dos dispositivos.

Nesse artigo é proposta uma versão preliminar de um escalonador para SO-TR que executa em ambientes multicores. O escalonador, denominado GSPAM, produz um escalonamento considerando, além dos *deadlines* das tarefas, os múltiplos núcleos presentes nas máquinas atuais. Foi utilizado o simulador SimSo [Maxime Chéramy 2014] para realizar os testes e, neste estudo preliminar, o GSPAM produziu resultados melhores que o algoritmo EDF somente quando as tarefas são periódicas.

2. Descrição da abordagem

Este trabalho apresenta uma solução preliminar para o escalonamento de tarefas periódicas, esporádicas e aperiódicas em SO-TR. Tarefas esporádicas e periódicas sempre são tratadas como tarefas críticas, ou seja, sempre precisam ter suas execuções concluídas em um tempo menor do que o *deadline* predefinido. Tarefas aperiódicas são tratadas como não críticas, ou seja, o seu tempo de execução pode ser maior do que o tempo de resposta predefinido.

Cada tarefa periódica τ_i^p é definida como uma tupla $\tau_i^p = (T_i, D_i, C_i, P_i)$, sendo T_i o período, D_i o *deadline* relativo, C_i o custo de pior caso de execução (WCET - *Worst-case execution time*) e P_i a prioridade da tarefa. Já a tarefa esporádica τ_i^e é definida como

uma tupla $\tau_i^e = (min_i, D_i, C_i, P_i)$, sendo min_i o menor tempo entre duas ocorrências da tarefa τ_i^e. Tarefas aperiódicas são definidas como $\tau_i^a = (S_i, D_i, C_i, P_i)$, onde S_i é o tempo de surgimento da tarefa.

O GSPAM utiliza diferentes algoritmos de escalonamento, dependendo do tipo de tarefa que será escalonada. Os algoritmos utilizados são descritos a seguir e foram retirados de [Jean-Marie Farines 2000]:

I **DM (*Deadline Monotonic*)**: este algoritmo escalona um conjunto de tarefas periódicas. As tarefas são selecionadas para execução em ordem de prioridade, sendo que a maior prioridade é atribuída para a tarefa com o menor *deadline* relativo. A prioridade é estática.

II **EDFS (*Earliest Deadline First for Sporadic*)**: este escalonador é baseado no EDF (*Earliest Deadline First*), porém considera um conjunto de tarefas composto por tarefas aperiódicas e esporádicas. O EDF executa a tarefa com o *deadline* mais próximo, por isso é definido como um escalonador dinâmico. A prioridade da tarefa pode mudar durante a execução;

III **DS (*Deferrable Server*)**: este algoritmo escalona tarefas periódicas e aperiódicas. O DS cria uma tarefa servidora, S, que tem a maior prioridade dentre as tarefas periódicas e possui uma capacidade c que é definida inicialmente com valor igual ao maior WCET das tarefas aperiódicas ou esporádicas. As demais prioridades são definidas da seguinte forma: i) tarefas periódicas tem prioridade definida de acordo com a mais frequente, ou seja, quanto menor o período maior a prioridade; ii) tarefas aperiódicas possuem prioridade nula. A execução das tarefas periódicas é realizada de acordo com a prioridade, mas as tarefas aperiódicas apenas executam quando a tarefa S libera a utilização do processador. Vale ressaltar que S executa somente quando existe tarefas aperiódicas requisitando poder de processamento e S possui capacidade para executá-la. Ainda, a capacidade de S é preenchida no inicio do período de S.

2.1. GSPAM

Inicialmente o GSPAM divide as tarefas em dois conjuntos, um de tarefas periódicas outro de aperiódicas. Logo, são criadas n listas, sendo cada uma delas associada a um núcleo da máquina e em para cada lista será executado um tipo de escalonamento. As tarefas são associadas às listas de acordo com os seguintes casos. Através do teste de escalabilidade, é verificado a quantidade mínima de núcleos necessários para executar todas as tarefas periódicas. Caso todos os núcleos sejam necessários então as tarefas periódicas são divididas entre todas as listas criadas e sobre $n - 1$ listas é executado o escalonador DM. À última lista são adicionadas as demais tarefas (aperiódicas e esporádicas) e sobre esta lista é executado o algoritmo DS. Caso não seja necessário todos os núcleos, as tarefas aperiódicas e esporádicas são atribuídas às listas dos núcleos não utilizados para executar as tarefas periódicas. Estas são escalonadas com o EDFS e as tarefas periódicas, como anteriormente, são escalonadas com DM.

Após a distribuição é realizado novamente o teste de escalonabilidade para cada lista de tarefas para verificar a viabilidade da solução, evitando assim a utilização de uma distribuição que pode ter falhas em tarefas críticas. A análise de escalonabilidade é realizada como descrito em [Jean-Marie Farines 2000], contudo o teste referente ao

EDFS foi alterado com o objetivo de fazer a verificação tanto para tarefas esporádicas e aperiódicas.

3. Resultados

Como o trabalho ainda está em desenvolvimento, foi realizado um teste preliminar com um conjunto de 9 tarefas, mescladas entre periódicas e aperiódicas com o intuito de avaliar a proposta deste trabalho. O EDF foi utilizado como comparativo. A Tabela 1 mostra que na ocupação geral (em porcentagem) o EDF se mostrou melhor do que o GSPAM, pois aproveita os intervalos de ociosidade entre as tarefas periódicas para executar tarefas aperiódicas ou esporádicas. Contudo, na mesma tabela, foi observado que os processadores específicos para tarefas periódicas (núcleo 1) são melhor preenchidos no GSPAM do que no EDF.

Tabela 1. Porcentagem de ocupação dos processadores

	GSPAM		EDF	
	Carga total	Ocupação total	Carga total	Ocupação total
CPU1	0.8000	0.8000	0.6700	0.6700
CPU2	0.4000	0.4000	0.7700	0.7700
CPU3	0.6400	0.6400	0.6550	0.6550
CPU4	0.9900	0.9900	0.8500	0.8500
Média	0.7075	0.7075	0.7362	0.7362
Perda de soft-deadlines	5		1	

4. Conclusão e Trabalhos Futuros

Foi apresentado uma nova abordagem para o escalonamento de tarefas periódicas, esporádicas e aperiódicas em ambientes multicores. Os testes iniciais foram importantes para mostrar que o GSPAM produz um escalonamento compatível com o EDF, mas com um melhor aproveitamento dos processadores destinados a tarefas periódicas. Contudo, apresenta uma taxa de ocupação menor e um maior número de perda de *soft-deadline* do que o EDF.

Os próximos passos serão destinados a realizar melhorias na distribuição das tarefas entre os núcleos, de forma a diminuir o tempo ocioso de cada um e implementar políticas de escalonamentos mais eficientes que aceitem compartilhamento de recursos e dependências entre tarefas.

Referências

Jayavardhana Gubbi, Rajkumar Buyya, S. M. M. P. (2012). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*.

Jean-Marie Farines, Joni da Silva Fraga, R. S. d. O. (2000). *Sistemas de Tempo Real*. UFSC, 1 edition.

Maxime Chéramy, Pierre-Emmanuel Hladik, A.-M. D. (2014). Simso: A simulation tool to evaluate real-time multiprocessor scheduling algorithms. *5th International Workshop on AnalysisTools and Methodologies for Embedded and Real-time Systems (WATERS)*.