

Módulo de Escalonadores para Simulação de Sistemas

Exascale

William D’Abruzzo Martins¹, Aleardo Manacero Jr. ¹, Renata S. Lobato¹, Roberta Spolon²

¹Departamento de Ciências da Computação e Estatística
Universidade Estadual Paulista (UNESP) - São José Do Rio Preto - SP - Brasil

²Departamento de Computação
Universidade Estadual Paulista (UNESP) - Bauru - SP - Brazil

{william.dabruzzo, renata.spolon}@unesp.br

Abstract. *Scheduling in computing systems is a fundamental process for performance gains. In order to study schedulers for large-scale computing systems, simulation remains the most viable option in terms of cost and accuracy. However, simulators available in the literature do not provide a user-friendly interface for modeling, requiring knowledge in scripting or a programming language. Thus, this work develops an intuitive and iconic Scheduler Module for iSPD Exa, enabling the creation of simple schedulers in a graphical interface.*

Resumo. *O escalonamento em sistemas computacionais é um processo fundamental para ganho de desempenho. A fim de estudar escalonadores para sistemas computacionais de grande porte, a simulação se mantém a opção mais viável em custo e precisão. Contudo, simuladores disponíveis na literatura não apresentam interface de fácil uso e modelagem, necessitando conhecimento em scripting ou em uma linguagem de programação. Dessa maneira, esse trabalho desenvolve o Módulo de Escalonadores intuitivo e icônico para o iSPD Exa, de modo que seja possível criar escalonadores simples em uma interface gráfica.*

1. Introdução

O escalonamento é um dos principais componentes em sistemas computacionais de grande porte, sendo essencial para desempenho computacional [Menezes et al. 2012], eficiência energética [Albers 2022], e otimizações de múltiplos critérios [Bittencourt et al. 2018].

Simuladores de sistemas de grande porte disponíveis na literatura, como o Grid-Sim [Buyya and Murshed 2002], SimGrid [Casanova 2001] e outros não apresentam uma interface de fácil uso para modelagem do sistema, sendo necessários conhecimentos de *scripts* ou de uma linguagem de programação em específica. Visando melhorar a facilidade de uso e de modelagem, o iSPD (*iconic Simulator of Parallel and Distributed systems*) [Manacero et al. 2012] e o iSPD Exa (*iconic Simulator of Parallel and Distributed system in Exascale*) [Barroso et al. 2023], simuladores utilizados e já validados, utilizam uma interface baseada em ícones para a criação de modelos e visualização de resultados.

Como o princípio do iSPD é a possibilidade de criação de modelos a partir de uma interface simples, a especificação de algoritmos de escalonamento também deve ser realizada com simplicidade. Atualmente, o motor de simulação do iSPD Exa

[Fares et al. 2023] apresenta somente dois escalonadores: *RoundRobin* e *Workqueue*. Para a inclusão de novos escalonadores, será necessário alterar o código-fonte do sistema e escrever as rotinas do escalonador.

Sendo assim, visando manter a simplicidade, esse trabalho propõe a criação de um módulo de escalonadores para o iSPD *Exa* de modo a possibilitar a criação e modelização de escalonadores para sistemas de grande porte.

2. Escalonamento em sistemas de grande porte

O escalonamento é estudado visando distribuir melhor os recursos para execução de determinada tarefa, de modo a obter um resultado desejado. Dessa maneira, o processo de escalonamento consiste em alocar tarefas para recursos, ou alocar recursos para executar determinada tarefa.

Na literatura, os critérios de escalonadores de sistemas distribuídos podem ser separados em cinco principais [Elzeki et al. 2012]:

- **Escalonamento Estático:** Escalona todos os recursos previamente, tendo todas as informações necessárias.
- **Escalonamento Dinâmico:** Escalona em tempo real, podendo escalonar os recursos no exato momento que se tornarem disponíveis e realizar escolhas baseadas no estado atual do sistema.
- **Escalonamento Centralizado:** Um único componente é responsável por realizar decisões globais.
- **Escalonamento Descentralizado:** Componentes distribuídos mantêm controle sobre o processo de escalonamento.
- **Escalonamento Preemptivo:** Permite que uma tarefa seja interrompida no meio de sua execução e/ou transferida para outro recurso
- **Escalonamento não preemptivo:** Não permite a preempção, tarefas não podem ser realocadas até o fim de sua execução.

3. Metodologia

O módulo de escalonadores foi construído considerando cinco passos:

1. **Nome:** O usuário pode adicionar o nome do escalonador;
2. **Propriedade:** O usuário poderá escolher entre um escalonador dinâmico ou estático;
3. **Especificação:** Especificação da fórmula para escolha do recurso. É dividido em ordem, variáveis e operadores:
 - **Ordem:** Crescente, Decrescente, Aleatório e FIFO;
 - **Variáveis:** Número de núcleos CPU (*cpuCores*), Poder de Processamento da CPU (*cpuPP*), Número de núcleos GPU (*gpuCores*), Poder de Processamento da GPU (*gpuPP*), Tamanho de Processamento da Tarefa (*TPS*), Tamanho de Comunicação da Tarefa (*TCS*), *Offloading* Computacional da Tarefa (*TOFF*), MFLOPS Executando no Recurso (*RMFE*) e Instante da última tarefa concluída (*LCT*);
 - **Operadores:** Soma, subtração, divisão, multiplicação e precedência (parênteses);

4. **Restrições:** Restrição do número de tarefas por recurso, ou nenhuma restrição;
5. **Confirmação:** Revisão do escalonador e confirmação.

Ao final, é gerado um arquivo com a extensão *.ims* que será interpretado por um módulo do motor de simulação. No exemplo a seguir, uma tarefa executará no recurso cujo tempo de execução pela CPU é o menor e que o recurso também não esteja executando nenhuma outra tarefa.

Figura 1. Exemplo do arquivo *ims* gerado

```
zSCHEDULER SHORTEST_CPU_TIME
RESTRICT 1 TASKS PER RESOURCE
DYNAMIC
FORMULA INCREASING ( ( 1.00 - TOFF ) * TPS ) /cpuPP
```

4. Conclusão

A interface gráfica para a geração de escalonadores permite a modelagem mais simples e intuitiva. Os critérios disponíveis permitem a modelização levando em conta vários parâmetros, principalmente em sistemas *exascale* como fatores da GPU, CPU, intercomunicação, entre outros [Sandokji and Eassa 2018], [Branco 2004].

Até o momento, resta programar a interpretação do arquivo *.ims* para o motor de simulação, que será realizada com uso da biblioteca TinyExpr¹ para interpretação da fórmula matemática.

No mais, o módulo de escalonadores se limita a modelização de escalonadores centralizados e não preemptivos, uma restrição a ser superada em trabalhos futuros.

Referências

- Albers, S. (2022). *Energy-Efficient Scheduling*, pages 196–212. Springer Nature Switzerland, Cham.
- Barroso, J., Manacero, A., Fares, R., Pereira, D., Jesus, B., Lobato, R., and Spolon, R. (2023). Interface gráfica para simulação de sistemas *exascale*. In *Anais da XIV Escola Regional de Alto Desempenho de São Paulo*, pages 33–36, Porto Alegre, RS, Brasil. SBC.
- Bittencourt, L. F., Goldman, A., Madeira, E. R., da Fonseca, N. L., and Sakellariou, R. (2018). Scheduling in distributed systems: A cloud computing perspective. *Computer Science Review*, 30:31–54.
- Branco, K. R. L. J. C. (2004). *Índices de carga e desempenho em ambientes paralelos/distribuídos - modelagem e métricas*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.
- Buyya, R. and Murshed, M. (2002). Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15):1175–1220.

¹<https://github.com/codeplea/tinyexpr>

- Casanova, H. (2001). Simgrid: A toolkit for the simulation of application scheduling. In *Cluster Computing and the Grid, IEEE International Symposium on*, page 430, Los Alamitos, CA, USA. IEEE Computer Society.
- Elzeki, O., Rashad, M., and Abu Elsoud, M. (2012). Overview of scheduling tasks in distributed computing systems. *International Journal of Soft Computing and Engineering*, 2:470–475.
- Fares, R., Manacero, A., Barroso, J., Genova, L., Juiz, G., Lobato, R., and Spolon, R. (2023). Simulação de sistemas exascale usando time warp. In *Anais da XIV Escola Regional de Alto Desempenho de São Paulo*, pages 29–32, Porto Alegre, RS, Brasil. SBC.
- Manacero, A., Lobato, R. S., Oliveira, P. H. M. A., Garcia, M. A. B. A., Guerra, A. I., Aouqui, V., Menezes, D., and Da Silva, D. T. (2012). ispd: an iconic-based modeling simulator for distributed grids. In *Proceedings of the 45th Annual Simulation Symposium, ANSS '12*, San Diego, CA, USA. Society for Computer Simulation International.
- Menezes, D., Manacero, A., Lobato, R. S., da Silva, D. T., and Spolon, R. (2012). Scheduler simulation using ispd, an iconic-based computer grid simulator. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000637–000642.
- Sandokji, S. and Eassa, F. (2018). Task scheduling frameworks for heterogeneous computing toward exascale. *International Journal of Advanced Computer Science and Applications*, 9(10).