

Computação Sustentável e Mecanismos de Sincronização

Timóteo M. Rico¹, Felipe L. Teixeira^{1*}, Maurício L. Pilla¹, André R. Dubois¹

¹Universidade Federal de Pelotas (UFPel)

Computação - CDTec

LUPS - Laboratory of Ubiquitous and Parallel Systems

{tmrico, flteixeira, pilla, dubois}@inf.ufpel.edu.br

1. Introdução

O poder computacional está a cada dia mais presente no cotidiano. Percebe-se isso, com o aumento no número de dispositivos móveis, computadores pessoais, e a computação na nuvem, que em muitos casos utiliza-se de grandes *data centers*. Estima-se que somente os *data centers* sejam responsáveis por cerca de 1,5% de eletricidade total consumida no planeta, e este montante deve crescer a menos que utilize-se mecanismos que reduzam esse consumo e ainda assim atendam aos requisitos de processamento [Sharavanan et al. 2010].

Em relação aos dispositivos móveis (celulares, *notebooks*, *tablets*, entre outros) a autonomia está relacionada diretamente à duração de suas baterias, portanto otimizar recursos computacionais de modo que esses reduzam o consumo de energia é de suma importância. Nesse sentido, pesquisadores de computação tem tomado consciência sobre o consumo energético em sistemas computacionais e os impactos causados por esses no meio ambiente, gerando então uma área de pesquisa ainda recente, a Computação Sustentável (*Green Computing*), que tem por objetivo prover um conjunto de práticas voltadas para a computação de modo que minimize-se o impacto ambiental [Kurp 2008].

Em sistemas computacionais, a eficiência energética pode ser obtida através de otimizações tanto de *hardware* como *software*, desde o projeto do circuito, passando pelo sistema operacional, compiladores, chegando até o nível de aplicação. Contudo, tem-se como limitante da redução de consumo que os requisitos temporais dos aplicativos continuem sendo atendidos, mesmo que possam ocorrer perdas no desempenho.

2. Proposta

Com o advento de arquiteturas multicore, a programação paralela tem papel fundamental na melhoria da eficiência, por fazer melhor proveito dos recursos de *hardwares* disponíveis e aumentar o desempenho das aplicações [Mór et al. 2010]. Um modelo de programação concorrente muito utilizado é o *multithreaded* [Andrews 2000]. Neste modelo, o paralelismo é explicitado por *threads*, que compartilham a memória por padrão. A comunicação entre *threads* concorrentes é feita através da escrita e/ou leitura em variáveis compartilhadas e há necessidade de utilizar-se de mecanismos de sincronização para garantir a integridade e consistência de execução entre *threads* concorrentes [Andrews 2000].

*Bolsista PIBITI/CNPq

Nesse contexto, este trabalho tem por objetivo **realizar um estudo investigativo de comparações entre mecanismos de sincronização em ambientes de memória compartilhada**, tomando como métricas o desempenho e o consumo de energia sobre cada mecanismo.

Foram estudados conceitos e aplicações de Computação Sustentável relacionada a mecanismos de sincronização em ambiente de memória compartilhada. Tem-se como relacionados a este projeto os seguintes trabalhos: [Baldassin et al. 2009], [Moreshet et al. 2006] e [Moreshet et al. 2005]. Nesta perspectiva, para cada trabalho, diferentes técnicas e implementações de sincronização foram exploradas visando a redução no consumo de energia.

O foco do projeto é analisar o consumo de programas usando memórias transacionais em *software* e comparar com outras técnicas de sincronização, como *mutexes*. As métricas usadas serão desempenho e consumo de energia, sendo que a eficiência das diferentes técnicas e implementações de memórias transacionais serão comparadas para obter o mais eficiente método de sincronização para cada tipo de ambiente simulado dentre os analisados.

Referências

- Andrews, G. R. (2000). *Foundations of multithreaded, parallel, and distributed programming*. Addison Wesley, Boston.
- Baldassin, A., Klein, F., Araujo, G., Azevedo, R., and Centoducatte, P. (2009). Characterizing the energy consumption of software transactional memory. *IEEE Comput. Archit. Lett.*, 8:56–59.
- Kurp, P. (2008). Green computing. *Commun. ACM*, 51:11–13.
- Moreshet, T., Bahar, R. I., and Herlihy, M. (2005). Energy reduction in multiprocessor systems using transactional memory. In *Proceedings of the 2005 international symposium on Low power electronics and design, ISLPED '05*, pages 331–334, New York, NY, USA. ACM.
- Moreshet, T., Bahar, R. I., and Herlihy, M. (2006). Energy-aware microprocessor synchronization: Transactional memory vs. locks. In *4th Workshop on Memory Performance Issues, held in conjunction with the International Symposium on High-Performance Computer Architecture*.
- Mór, S. D. K., Alves, M. A., Lima, J. V. F., Mailard, N. B., and Navaux, P. O. A. (2010). Eficiência energética em computação de alto desempenho: Uma abordagem em arquitetura e programação para green computing. *XXX Congresso da Sociedade Brasileira de Computação, CSBC*, pages 346–360.
- Sharavanan, S., Poongodi, D., and Kumar, A. (2010). Towards green computing: Energy efficient data centers. *Journal of Mathematics and Technology*, pages 131–135.