

## Validação de um modelo de custos de escalonamento em um ambiente de processamento de alto desempenho\*

Marcelo Augusto Cardozo Jr, Gerson G. H. Cavalheiro

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada  
Universidade do Vale do Rio dos Sinos  
São Leopoldo – RS – Brasil  
{mcardozo, gersonc}@exatas.unisinos.br

### Resumo

A exploração de aglomerado de computadores para processamento de alto desempenho requer que seus recursos sejam utilizados de maneira eficiente. Uma maneira para obter tal eficiência se dá utilizando-se de técnicas de escalonamento para exploração da concorrência definida pela aplicação. Assim, o principal objetivo do escalonamento é distribuir a carga computacional do programa em execução sobre os recursos de processamento da arquitetura de forma a otimizar algum índice de desempenho (ex. tempo de execução). Em [Feitelson, 1994] encontramos uma classificação dos mecanismos de escalonamento em dois níveis: aplicativo e sistema.

O escalonamento realizado em nível aplicativo tem um interesse particular em ambientes para o processamento de alto desempenho: por ser realizado de forma associada ao programa em execução, informações da aplicação, e da própria evolução do programa, podem ser exploradas de forma a otimizar as estratégias de distribuição de carga.

Com o escalonamento em nível aplicativo podemos utilizar informações sobre dependência entre tarefas montando um grafo orientado. As informações contidas no grafo, representando a execução em curso, permitem uma análise mais aprofundada sobre as características de execução do programa, podendo ser empregadas heurísticas para aprimorar a estratégia de escalonamento utilizada. Além das informações sobre precedência de execução entre tarefas, é possível obter outros dados sobre o programa, tal como custo total de execução ( $T_1$ ) e tempo mínimo de execução paralela ( $T_\infty$ ), este último dado pelo caminho crítico definido no grafo pelas relações de dependência. Estes custos,  $T_1$  e  $T_\infty$ , consistem nas principais informações com as quais pode-se estimar o desempenho de execução sobre um programa (cf. Graham [Graham, 1969]). Uma vez conhecidos estes custos, pode-se ter informações sobre os limites teóricos sobre o seu desempenho ([Graham, 1969] [David B. Shmoys and Williamson, 1995]).

A prática no entanto mostra que diversos outros custos computacionais são inseridos na execução de um programa. Em muitos casos estes são introduzidos pelo o próprio ambiente de suporte à execução. Alguns destes custos são facilmente identificáveis, como

---

\*Projeto Anahy - CNPq(55.2196/02-9) Projeto parcialmente desenvolvido em parceria com HP Brasil P&D

por exemplo o *overhead* gerado pela criação de uma tarefa. Uma vez identificadas as fontes de tais custos, um trabalho de modelagem do ambiente de execução pode ser realizado de forma a reduzir o impacto destes no desempenho final.

Neste trabalho esta sendo realizado um estudo dos custos envolvidos na execução de programas desenvolvidos para Anahy [Cavalheiro et al., 2003][Dall'Agnol et al., 2003]. Anahy possui uma interface de programação de alto nível que permite a construção de um grafo de dependências entre tarefas. Seu núcleo executivo é baseado em um algoritmo de listas[Graham, 1969] que explora este grafo para otimizar o tempo total de execução do programa. Para tanto, um modelo de simulação para Anahy foi criado e implementado de forma a permitir a análise das possíveis fontes de custos, assim como também esta sendo utilizado programas experimentais que têm por objetivo validar o modelo simulado.

O estudo realizado até o presente momento definiu um modelo de simulação do ambiente Anahy, permitindo representar a execução de programas sintéticos. Busca-se, com este trabalho, identificar diferentes fontes de *overhead* introduzidos por Anahy, obtendo uma maior precisão na identificação dos limites de desempenho. Também busca-se uma equação expandida de Graham que defina estes limites.

## Referências

- [Cavalheiro et al., 2003] Cavalheiro, G. G. H., Real, L. C. V., and Dall'Agnol, E. C. (2003). Uma biblioteca de processos leves para a implementação de aplicações altamente paralelas. In *IV Workshop em Sistemas Computacionais de Alto Desempenho*, São Paulo, SP.
- [Dall'Agnol et al., 2003] Dall'Agnol, E. C., Real, L. C. V., Benitez, E. D., and Cavalheiro, G. G. H. (2003). Portabilidade na programação para o processamento de alto desempenho. In *IV Workshop em Sistemas Computacionais de Alto Desempenho*, São Paulo, Brasil.
- [David B. Shmoys and Williamson, 1995] David B. Shmoys, J. W. and Williamson, D. P. (1995). Scheduling parallel machines on-line. *SIAM Journal on Computing*, 24(6):1313–1331.
- [Feitelson, 1994] Feitelson, D. (1994). A survey of scheduling in multiprogrammed parallel systems. Technical Report 19790(87657), IBM T. J. Watson Research Center, Chemnitz.
- [Graham, 1969] Graham, R. L. (1969). Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429.