Escalonamento Distribuído de Tarefas Dinâmicas em Dois Níveis com MPI-2

Stéfano Drimon Kurz Mór[1], **Nicolas Maillard**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{sdkmor,nicolas}@inf.ufrgs.br

1. Introdução & Motivação

Aglomerados de computadores (*clusters*) tornaram-se alternativa prática e cada vez mais comum em PAD, visto que podem ser construídos através da composição em redes de *hardware* menos especializado [Patterson e Hennessy 2003, Patterson e Hennessy 2005]. Mesmo que avanços na construção de processadores de propósito geral possam trazer melhoria no desempenho de execuções seqüenciais, é necessário suporte da arquitetura de cluster para o uso ótimo dessas melhorias. Graças às tecnologias *Simultaneous Multiprocessors* (SMP) e *Simultaneous Multithread* (SMT) [Navaux e Rose 2003] tem-se (em conformidade com [Flynn 1972]) multicomputadores MIMD (clusters) de multiprocessadores MIMD (*chip multicore*).

A teoria clássica de escalonamento trata da alocação de *tarefas* em *recursos* (1 nível) [Casavant e Kuhl 1994]. O cenário atual, todavia, evidencia a necessidade de escalonamento multinível. *v.g.*, a Figura 1 ilustra um possível cenário de escalonamento multinível.

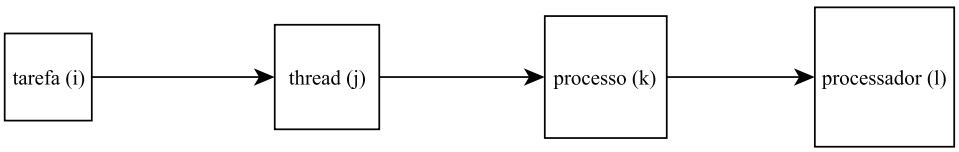

Figura 1. Escalonamento multinível. As setas significam "é escalonado em". O tamanho dos blocos é proporcional à granularidade de um componente em relação ao próximo. Todos os relacionamentos são do tipo 1:1.

Não há na literatura algum padrão para comunicação e escalonamento em clusters que faça escalonamento dinâmico multinível e seja largamente utilizado; hoje, MPI é o padrão *de facto* para comunicação em clusters.

MPI-1 centrava-se na comunicação entre processos, que se dá por troca de mensagens [MPI Forum 1994]. MPI-2 [MPI Forum 2008] é uma extensão da primeira norma que oferece uma série de facilidades para o programador, em especial a *criação dinâmica de tarefas*. No entanto, não há especificação de como o escalonamento destes processos deve ser feito em relação aos recursos disponíveis, tampouco uma correspondência explícita entre a noção de tarefa e a noção de processo. Não há restrição de dependência de dados. Embora MPI suporte a utilização de *threads* pelo usuário, não provê suporte próprio ao seu uso.

No quesito de granularidade da tarefa (*threads vs.* processos), a LibSpawn [Lima e Maillard 2008] oferece um mapeamento transparente de processos MPI-2 em *threads* POSIX, ao invés do tradicional mapeamento em processos do Sistema Operacional; sempre que o número de *threads* em um dado recurso superar uma constante M (definida em tempo de compilação), devido à criação dinâmica de uma dada tarefa, um novo processo de Sistema Operacional é alocado em um novo recurso. Apoiando-se em um padrão amplamente difundido, ainda falta à LibSpawn a capacidade de escalonar eficientemente as tarefas de acordo com a carga de trabalho dos processadores participantes da computação.

2. Objetivos

Não temos conhecimento de alguma implementação do padrão MPI-2 capaz de fornecer um escalonamento multinível, distribuído, adaptável e eficiente de tarefas dinamicamente criadas e de maneira transparente ao programador. Este trabalho visa preencher essa lacuna, fornencendo um escalonador com balanceamento de carga em dois níveis (*threads* e processos) para a LibSpawn.

Esse trabalho está em seu estágio inicial e a descrição apresentada é fruto de uma Proposta de Estudos e Pesquisa para Mestrado em Ciência da Computação.

Referências

Casavant, T. L. e Kuhl, J. G. (1994). A taxonomy of scheduling in general-purpose distributed computing systems. Em *Readings in Distributed Computing Systems*. IEEE Computer Society Press, Los Alamitos, CA.

Flynn, M. J. (1972). Some computer organization and their effectiveness. Em *IEEE Transactions on Computers*, volume C-21.

Lima, J. V. F. e Maillard, N. (2008). Controle de granularidade com threads em programas mpi dinâmicos. In SBC, editor, *IX Simpósio em Sistemas Computacionais - WSCAD-SSC 2008*, páginas 117–124, Campo Grande, Brasil.

MPI Forum (1994). MPI: A Message-Passing Interface Standard. Relatório Técnico CS-94-230, University of Tennessee, Knoxville, Tennessee.

MPI Forum (2008). Mpi: A message-passing interface standard ver. 2.1. Relatório Técnico, Message Passing Interface Forum.

Navaux, P. O. A. and Rose, C. A. F. D. (2003). *Arquiteturas Paralelas*. Número 15 in Série Livros Didáticos. Editora Sagra Luzzato, Edição 1.

Patterson, D. A. e Hennessy, J. L. (2003). *Arquitetura de Computadores: Uma Abordagem Quantitativa*. Editora Campus, Edição 3.

Patterson, D. A. e Hennessy, J. L. (2005). *Organização e Projeto de Computadores: A Interface Hardware Software*. Editora Campus, Edição 3.