

Ambientes de programação paralela que suportam Divisão e Conquista

Guilherme Peretti Pezzi, Nicolas Mailard,
Philippe Olivier Alexandre Navaux

Universidade Federal do Rio Grande do Sul
Instituto de Informática - UFRGS - Porto Alegre, RS
{pezzi, nmaillard, navaux}@inf.ufrgs.br

Resumo

A especificação e implementação de um programa paralelo é um problema complexo e ainda em aberto. Existem várias propostas de arquitetura e modelos de programação paralela: troca de mensagens, memória compartilhada, chamada remota de método, etc. Uma arquitetura que vem despertando interesse de estudo é o *Grid*, que permite agregar uma quantidade de recursos que varia ao longo do tempo. Apesar dessa característica acrescentar flexibilidade ao sistema, é preciso encontrar um ambiente de programação adaptado para essa arquitetura.

Afim de proporcionar uma maior eficiência e melhor aproveitamento dos recursos, o modelo de programação deve poder alterar dinamicamente o grau de paralelismo, ou seja, o número de tarefas em execução em um dado momento. Uma possibilidade para resolver essa questão é utilizar o modelo de programação de divisão e conquista [THO 2002]. Esse modelo permite dividir o trabalho em subtarefas, que podem ser atribuídas dinamicamente para diferentes processadores. Um estudo em andamento sobre esse tipo de escalonamento pode ser encontrado em [CER 2005].

Neste trabalho está sendo feito um levantamento bibliográfico sobre ambientes de programação paralela existentes, que suportam o modelo de divisão e conquista. Este estudo faz parte de um Trabalho Individual e poderá servir como subsídio para uma posterior definição de um ambiente de programação baseado no modelo de divisão e conquista.

Objetivos

O objetivo deste trabalho é fazer uma busca de ambientes paralelos que suportam divisão e conquista e uma análise comparativa dos ambientes estudados. Estão sendo caracterizados, em cada ambiente: os operadores disponíveis para expressar o paralelismo, as arquiteturas alvo (*SMP*, *Cluster*, *Grid*, ...), mecanismo de sincronização dos processos, ao iniciar e finalizar as tarefas e uma análise crítica da facilidade de uso de cada ambiente.

Também está sendo verificado o estado atual da implementação dos ambientes, através da execução prática de programas nos ambientes estudados. Os programas executados também poderão servir para permitir uma comparação entre os ambientes.

Metodologia

Estão sendo buscados e identificados ambientes de programação paralela, por exemplo: Cilk [BLU 95], Satin [NIE 2000], Charm++ [KAL 93], AMPI [HUA 2003], Jade [DES 2003], MPI, MPI-2 [MPI 2005], *MAP₃S* [MEH 2005], Athapascan, Anahy, etc. A partir dessa lista, serão selecionados os ambientes mais apropriados para programação utilizando divisão e conquista. Para cada um dos ambientes selecionados, serão estudadas as características acima mencionadas: funcionalidades oferecidas, arquitetura alvo, meios de sincronização, etc. Também serão estudadas e executadas aplicações teste e *benchmarks* dos ambientes.

Basicamente pode-se dividir esse trabalho em 4 principais etapas: seleção dos ambientes mais apropriados, estudo bibliográfico dos ambientes selecionados, teste dos ambientes e síntese das características.

Referências

- [BLU 95] BLUMOFÉ, R. et al. Cilk: an efficient multithreaded runtime system. In: SYMPOSIUM ON PRINCIPLES AND PRACTICE OF PARALLEL PROGRAMMING, 5., 1995. **Proceedings...** [S.l.: s.n.], 1995.
- [CER 2005] CERA, M. C. et al. Betampi-2: uma biblioteca para o escalonamento dinâmico de processos mpi. In: III WORKSHOP PPD/UFRGS - WSPPD'2005, 2005, Porto Alegre - RS. **Anais...** [S.l.: s.n.], 2005.
- [DES 2003] DESOUSA, J.; KALÉ, L. V. Jade: a parallel message-driven Java. In: WORKSHOP ON JAVA IN COMPUTATIONAL SCIENCE (ICCS 2003), 2003, Melbourne, Australia and Saint Petersburg, Russian Federation. **Anais...** [S.l.: s.n.], 2003.
- [HUA 2003] HUANG, C.; LAWLOR, O.; KALÉ, L. V. Adaptive MPI. In: INTERNATIONAL WORKSHOP ON LANGUAGES AND COMPILERS FOR PARALLEL COMPUTING, 16., 2003, College Station, Texas. **Proceedings...** [S.l.: s.n.], 2003.
- [KAL 93] KALE, L. V.; KRISHNAN, S. CHARM++ : A Portable Concurrent Object-Oriented System Based on C++. In: CONFERENCE ON OBJECT ORIENTED PROGRAMMING SYSTEMS, LANGUAGES AND APPLICATIONS (OOPSLA), 1993. **Proceedings...** ACM Press, 1993. p.91–108.
- [MEH 2005] MEHTA, P.; AMARAL, J. N.; SZAFRON, D. Is mpi suitable for a generative design-pattern system? In: WORKSHOP ON PATTERNS IN HIGH PERFORMANCE COMPUTING, 2005. **Anais...** [S.l.: s.n.], 2005.
- [MPI 2005] MPI message passing interface standard. Disponível em <http://www.mpi-forum.org/>. Acessado em ago/05.
- [NIE 2000] NIEUWPOORT, R. V. van; KIELMANN, T.; BAL, H. E. Satin: efficient parallel divide-and-conquer in java. In: Euro-Par 2000 Parallel Processing, 2000, Munich, Germany. **Anais...** Springer, 2000. n.1900, p.690–699. (Lecture Notes in Computer Science).
- [THO 2002] THOMAS H. CORMEN CHARLES E. LEISERSON, R. L. R. e. C. S. **Algoritmos: teoria e prática**. [S.l.]: Ed. Campus, 2002.