Porte de um Escalonador de Processos MPI-2 na Implementação LAM-MPI

Márcia Cristina Cera, Nicolas Maillard e
Philippe O. A. Navaux

Instituto de Informática - UFRGS - Porto Alegre, RS
{mccera, nicolas, navaux}@inf.ufrgs.br

Resumo

No contexto de programação paralela, a biblioteca de comunicação MPI (*Message Passing Interface*) [GRO 94] é tida como um padrão para a troca de mensagens e é a mais utilizada em sua categoria. A especificação da norma MPI-2 acrescenta à norma original a criação dinâmica de processos (em tempo de execução), acesso remoto a memória (RMA - *Remote Memory Access*) e entrada/saída paralela. Tais características ampliam as possibilidades de emprego de MPI, em especial a criação dinâmica de processos favorece desde o uso de arquiteturas dinâmicas (por exemplo as grades de computadores - *grids* [FOS 2003]) até às aplicações que criam processos sob demanda. Porém, para obter um bom aproveitamento dessa característica, é necessário o emprego de técnicas de escalonamento e distribuição de cargas eficiente.

As normas MPI não prevêem nenhuma especificação quanto ao escalonamento de processos, em geral as implementações de MPI oferecem apenas uma abordagem Round-Robin. Entre as implementações de MPI que implementam a norma MPI-2, nosso estudo concentrou-se na LAM-MPI [SQU 2003] pois esta apresenta o maior conjunto de características implementadas e é a mais estável. A LAM-MPI permite que se habilite uma chave (chamada `lam_spawn_sched_round_robin`) para distribuir os processos criados com `MPI_Comm_spawn` numa abordagem Round-Robin. Ao testar-se este mecanismo, identificou-se que o mesmo apresentava problemas dependendo da forma como é empregado. O mecanismo funciona adequadamente quando vários processos são criados a partir de uma única chamada `MPI_Comm_spawn`. Porém, quando deseja-se criar vários processos fazendo-se várias chamadas `MPI_Comm_spawn`, os processos acabam sendo criados sem que ocorra um balanceamento entre os nós de computação disponíveis. Isto porque não existe uma informação que determine quais nós já receberam processos novos. A Figura 1 exemplifica esta situação.

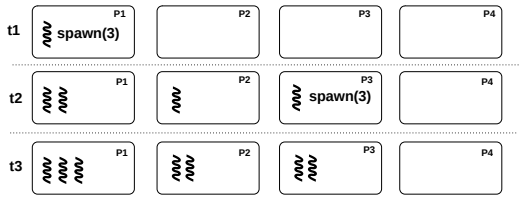

Figura 1: Distribuição Round-Robin da LAM-MPI

Uma solução para este problema foi apresentada em [CER 2006], onde foi proposta uma biblioteca capaz de efetivamente obter uma distribuição Round-Robim em todos os casos. Complementando o trabalho realizado para oferecer uma abordagem Round-Robin funcional, também foi desenvolvida uma abordagem capaz de distribuir os processos conforme a carga de processamento dos nós. Dessa forma, é possível obter um melhor desempenho graças ao melhor aproveitamento dos recursos, como pode ser visto na Figura 2 que apresenta os resultados obtidos para uma aplicação que faz a computação de números primos (para maiores informações consulte [CER 2006a]). Os resultados obtidos com as duas abordagens de escalonamento impulsionaram à que se buscasse passar de uma biblioteca auxiliar para escalonamento, que deve ser compilada com os programas MPI, para uma versão integrada diretamente na LAM-MPI. Tal integração do escalonador direto na implementação da LAM-MPI trará vantagens como: um menor sobrecusto para a aplicação e uma maior transparência para os usuários.

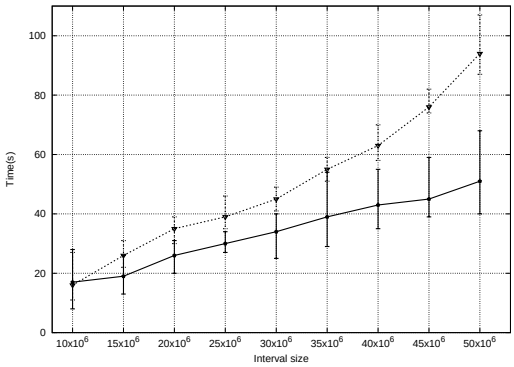

Figura 2: Computação de números primos com Round-Robin (linha tracejada) e com base na carga (linha sólida)

Referências

[CER 2006] CERA, M. C. et al. **Scheduling Dynamically Spawned Processes in MPI-2**. accepted in 12th Workshop on Job Scheduling Strategies for Parallel Processing, jun, 2006, Saint-Malo, France.

[CER 2006a] CERA, M. C. et al. Improving the Dynamic Creation of Processes in MPI-2. In: LECTURE NOTES IN COMPUTER SCIENCE - 13TH EUROPEAN PVMMPI USERS GROUP MEETING, 2006, Bonn, Germany. **Anais...** Springer Berlin / Heidelberg, 2006. v.4192/2006, p.247–255.

[FOS 2003] FOSTER, I.; KESSELMAN, C. **The grid**: blueprint for a new computing infrastructure. 2.ed. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2003.

[GRO 94] GROPP, W.; LUSK, E.; SKJELLUM, A. **Using MPI**: Portable Parallel Programming with the Message Passing Interface. Cambridge, USA: MIT Press, 1994.

[SQU 2003] SQUYRES, J. M.; LUMSDAINE, A. A Component Architecture for LAM/MPI. In: EUROPEAN PVM/MPI USERS' GROUP MEETING, 10., 2003, Venice, Italy. **Proceedings...** Springer-Verlag, 2003. n.2840, p.379–387. (Lecture Notes in Computer Science).