MigBSP: Nova Abordagem para Reescalonamento de Processos em Aplicações BSP

Rodrigo Righi[1], Alexandre Carissimi[1], Philippe Navaux[1], Hans-Ulrich Heiss[2]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

[2]Kommunikations- und Betriebssysteme - Technische Universität Berlin (TU Berlin)

{rodrigo.righi,asc,navaux}@inf.ufrgs.br, heiss@cs.tu-berlin.de

1. Introdução

Migração de processos é um pertinente mecanismo para oferecer balanceamento dinâmico de carga, principalmente em ambientes dinâmicos e heterogêneos[Bonorden 2007]. Nesse contexto, foi desenvolvido um modelo chamado MigBSP[Righi et al. 2008] que controla o reescalonamento de processos em aplicações BSP (*Bulk Synchronous Parallel*). Uma aplicação BSP é dividida em um ou mais superpassos, cada qual contendo ambas fases de computação e comunicação seguidas por uma barreira de sincronização. Uma vez que a barreira espera pelo processo mais lento, a idéia final de MigBSP é ajustar a localização dos processos com o intuito de reduzir o tempos dos superpassos. Considerando o escopo de aplicações BSP, as novas idéias de MigBSP são: (i) combinação de três métricas em outra escalar chamada Potencial de Migração; (ii) emprego de padrões de computação e comunicação para controlar a regularidade de cada processo; (iii) adaptação eficiente quanto a periodicidade de lançamento do reescalonamento de processos BSP.

2. MigBSP: Modelo de Reescalonamento de Processos

O Potencial de Migração (PM) indica quais são os processos candidatos para a migração e pode ser encontrado através da Equação 1. Cada processo i computa n funções $PM(i,j)$, onde n é o número de Conjuntos e j significa um Conjunto em especial. Um Conjunto é a abstração de uma rede e pode ser entendido como sendo um cluster, uma rede local ou um supercomputador. Cada um possui um gerente responsável pelo escalonamento hierárquico. A estratégia do cálculo do PM é não fazer todos os testes processos-recursos possíveis no momento do reescalonamento. $PM(i,j)$ é encontrado através da combinação das métricas Computação, Comunicação e Memória. A relação entre elas é baseada na noção de força da física. As métricas Computação e Comunicação atuam como forças a favor da migração, enquando a Memória trabalha em sentido contrário e adiciona ao modelo uma idéia de custo. Quanto maior o valor de $PM(i,j)$, maior é a tendência de migração do processo. Um alto $PM(i,j)$ significa que o processo i possui um alto tempo de computação, bastante comunicação com processos que pertencem ao Conjunto j e um baixo custo de migração para o Conjunto destino.

$$PM(i,j) = Comp(i,j) + Comm(i,j) - \frac{Mem(i,j)}{\frac{\alpha}{2}} \quad (1)$$

A métrica Computação - $Comp(i,j)$ - considera três informações: (i) predição do tempo de computação do processo i; (ii) Padrão de Computação do processo i; (iii) grau de desempenho do Conjunto j. A predição é baseada no conceito de Aging e usa dados entre duas chamadas para reescalonamento de processos. O Padrão de Computação

mede a regularidade do processo quanto as instruções que realiza em cada superpasso. O seu valor é próximo de 1 se o processo é regular (considerando um intervalo de erro) e perto de 0, caso contrário. O tratamento do escalonamento usando padrões foi empregado pois espera-se que o processo desempenhe um comportamento similar no lugar de destino. Assim como a métrica Computação, a Comunicação - $Comm(i, j)$ - computa o Padrão de Comunicação entre processos e Conjuntos. A cada superpasso, esse padrão considera a quantidade de dados recebidos pelo processo i provenientes de envios de processos do Conjunto j. Além disso, essa métrica usa predição do tempo de comunicação considerando dados entre duas ativações de balanceamento de carga. A métrica Memória - $Mem(i, j)$ - trabalha com os seguintes dados: (i) memória do processo; (ii) taxa de transferência entre o processo i e o gerente do Conjunto alvo j; (iii) custos de migração.

A decisão de remapeamento de processos é tomada no final de uma superpasso (depois da barreira de sincronização e antes do próximo superpasso). Esse ponto de migração foi escolhido porque neste momento é possível analisar dados de todos os processos BSP nas suas fases de computação e comunicação. Contrário a biblioteca PUBWCL[Bonorden 2007], MigBSP não realiza o reescalonamento a cada superpasso mas sim de acordo com o estado do ambiente. Para isso, foram aplicadas duas adaptações que gerenciam o valor de α. α é atualizado a cada chamada para reescalonamento e irá indicar o próximo intervalo para tal. Na Equação 1, a métrica Memória é dividida por $\frac{\alpha}{2}$ pois assume-se que os processos vão executar em média mais $\frac{\alpha}{2}$ superpassos após a chamada para reescalonamento. Para gravar as variações no estado do sistema entre duas chamadas para migração, uma variável temporária chamada α' foi usada. Ela é atualizada a cada superpasso através do incremento ou decremento de uma unidade. α é completada com o valor de α' no momento da chamada de reescalonamento. As idéias gerais das adaptações são: (i) atrasar a chamada para rebalanceamento de carga se o sistema estiver estável (processos estão balanceados entre os recursos) ou torná-la mais frequente, caso contrário; (ii) atrasar a chamada de reescalonamento se o sistema apresentar um padrão sem migrações nas últimas ω chamadas de reescalonamento. A segunda adaptação é importante para cenários onde os processos estão desbalanceados e migrações são inviáveis.

3. Resultados e Conclusão

Os resultados do desempenho de MigBSP podem ser vistos em [Righi et al. 2008]. Eles mostraram a viabilidade de uso do modelo, o qual impõe uma baixa sobrecarga na execução normal da aplicação quando migrações não acontecem (custo de 5%). Além disso, ganhos acima de 20% foram obtidos com migrações no ambiente multi-cluster testado. MigBSP oferece uma definição e uma relação precisa de informações para rescalonamento de processos. O trabalho de unificação de métricas de escalonamento para os cenários de sistemas distribuídos complexos é visto com um grande desafio para a área.

Referências

Bonorden, O. (2007). Load balancing in the bulk-synchronous-parallel setting using process migrations. In *21th International Parallel and Distributed Processing Symposium (IPDPS 2007)*, pages 1–9. IEEE.

Righi, R., Pilla, L., Carissimi, A., and Navaux, P. O. A. (2008). Controlling processes reassignment in bsp applications. In *20th International Symposium on Computer Architecture and high Performance Computing (SBAC-PAD 2008)*, pages 37–44. IEEE.