Introdução de Mensagens Ativas em um Ambiente de Execução para Processamento de Alto Desempenho *

Evandro Clivatti Dall'Agnol†, Lucas Correia Villa Real‡,
Daniela Saccol Peranconi§, Gerson G. H. Cavalheiro

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada
Universidade do Vale do Rio dos Sinos
Av. Unisinos, 950 - São Leopoldo - RS - Brasil
Fone/Fax: 590-8161
{ecd, lucasvr, danielap, gersonc}@exatas.unisinos.br

Introdução

A popularização dos aglomerados de computadores (também chamados de *clusters*) para o Processamento de Alto Desempenho (PAD) deve-se, principalmente, aos baixos custos para aquisição e manutenção de hardware para este tipo de arquitetura. No entanto, a exploração eficiente do PAD em aglomerados demanda, entre outras, o desenvolvimento de bons mecanismos de comunicação. Tais mecanismos necessitam tirar o máximo proveito da rede utilizada na comunicação entre os nodos do aglomerado, de forma a contribuir para um melhor aproveitamento dos recursos disponibilizados pela arquitetura.

Um paradigma bastante utilizado para a implementação desses mecanismos é o de RPC (*Remote Procedure Call*), que provê a execução remota de procedimentos. Contudo, cabe salientar que as implementações de RPC estão mais focadas em aumentar o nível de abstração na programação de aplicações em ambientes com memória distribuída do que em explorar eficientemente os recursos disponibilizados por estes ambientes. Uma maneira de implementar eficientemente um modelo de comunicação próximo ao RPC faz uso de Mensagens Ativas, cujo foco está na exploração dos recursos para PAD disponíveis em arquiteturas com memória distribuída.

Neste contexto, este trabalho apresenta a implementação e avaliação de desempenho de um mecanismo de comunicação baseado no modelo de Mensagens Ativas, cujo objetivo é explorar eficientemente o processamento de alto desempenho em arquiteturas do tipo aglomerados de computadores. Destaca-se, ainda, a interação entre o mecanismo proposto e o núcleo executivo de Anahy.

Modelo de Execução

O ambiente Anahy [CAV 2003] oferece ao programador a visão de uma arquitetura do tipo aglomerado de computadores como sendo uma arquitetura virtual multiprocessada com memória compartilhada. Neste caso, a arquitetura real é composta por

*Projeto Anahy – CNPq (55.2196/02-9). Este trabalho foi parcialmente desenvolvido em colaboração com a HP Brasil P&D.

†ITI - CNPq

‡ITI - CNPq

§Bolsista PROSUP/CAPES

um conjunto de nodos de processamento, dotados de memória local e unidades de processamento (CPU's). Sobre esta arquitetura existe uma arquitetura virtual composta por um conjunto de processadores virtuais (PVs) alocados sobre os nodos e por uma memória compartilhada pelos PV's. Cada PV é responsável por executar tarefas (atividades concorrentes nas quais uma aplicação é dividida) criadas pelo usuário.

As tarefas a serem executadas pelos processadores virtuais são manipuladas pelo escalonador Anahy, baseado em um algoritmo de listas [GRA 69], de forma a explorar eficientemente um *grafo de dependências*. Tal grafo é formado pelo conjunto de tarefas a serem executadas e pelas conexões entre elas, de maneira a descrever as dependências existentes entre as tarefas. Para atingir a eficiência esperada, o grafo é percorrido em profundidade e em ordem lexicográfica.

O núcleo executivo de Anahy realiza, basicamente, duas operações: atribuição de tarefas aos processadores e controle da dependência de dados, tendo como base o grafo de dependências. A partir do grafo pode-se obter uma ordem de execução que maximize a eficiência, evitando que tarefas sejam lançadas para execução e acabem bloqueando esperando o término de outra. A decisão sobre qual tarefa irá executar em um certo instante é tomada no momento em que uma nova tarefa é buscada na lista de tarefas prontas.

Considerando-se um ambiente com memória compartilhada em que todos os processadores compartilham o acesso ao grafo de dependências, pode-se facilmente modelar e gerenciar as operações descritas acima. Busca-se obter igual facilidade para ambientes com memória distribuída. Para tanto, é necessário que os serviços disponibilizados para ambientes com memória compartilhada sejam mapeados para ambientes com memória distribuída. Dessa forma, são necessários serviços de comunicação entre os nodos que possibilitem a troca de dados e tarefas entre eles, obedecendo as decisões de escalonamento. Para a implementação destes serviços, será utilizado o modelo de Mensagens Ativas.

Mensagens Ativas

Mensagens Ativas [EIC 92, CAR 99, ROL 2004] são soluções clássicas ao problema de comunicação em ambientes para o processamento de alto desempenho, permitindo realizar comunicações sem introduzir grande quantidade de sobrecustos de execução. O mecanismo de comunicação com Mensagens Ativas é assíncrono, não sendo necessário que o processo transmissor da mensagem fique bloqueado até que o receptor processe a mensagem recebida. Com isso, há a possibilidade de exploração mais eficiente dos recursos de processamento e comunicação existentes. Cada Mensagem Ativa contém em seu cabeçalho as informações necessárias para que o processo receptor execute determinada tarefa. Ou seja, a mensagem contém basicamente a identificação de uma função tratadora e os dados necessários à execução da tarefa. Opcionalmente, pode-se acrescentar dados de controle, como tamanho dos dados ou mesmo o destino para o retorno dos dados gerados pela execução da tarefa indicada.

Uma Mensagem Ativa tem caráter de urgência, ou seja, quando criada é enviada assim que possível e, quando recebida, é tratada o mais breve possível. Na recepção, a função tratadora é imediatamente lançada com a responsabilidade de somente retirar a mensagem e seus dados da rede [EIC 92, CAR 99] e inserí-la na computação existente, concluindo sua execução o mais brevemente possível. Desse modo, é utilizada a menor

quantidade possível de recursos, como processamento e *buffers*, a não ser os necessários pelo sistema operacional. Fica a cargo do escalonador do receptor a responsabilidade de concluir a tarefa indicada na mensagem.

A utilização conjunta de Mensagens Ativas com multiprogramação leve (uso de *threads*) oferece as condições necessárias para a sobreposição dos tempos de comunicação com computação efetiva [VAL 90]. Um mecanismo de Mensagens Ativas foi implementado para a adoção no ambiente Anahy, visando o suporte à execução distribuída de tarefas. Os resultados obtidos com esta implementação são apresentados na próxima seção.

A Figura 1 ilustra a integração do mecanismo de Mensagens Ativas com Anahy. A implementação realizada provê uma interface para que Anahy se utilize dos recursos de comunicação do sistema operacional (SO) de acordo com o conceito de Mensagens Ativas. Essa interface se dá através da chamada de funções como act_msg_create() e act_msg_send(). A interação de Anahy com o SO em relação ao recursos de execução permanece inalterado.

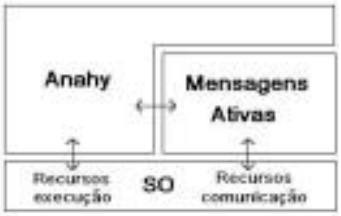

Figura 1: Integração de Mensagens Ativas em Anahy

Resultados

Sendo um ambiente de execução para processamento de alto desempenho, é de interesse que Anahy suporte a distribuição de carga entre diferentes nodos de um aglomerado. Visando este objetivo, foram avaliadas execuções sobre o mecanismo de Mensagens Ativas implementado para identificar sua escalabilidade e sobrecarga associados à invocação dos serviços remotos. Nota-se que os testes realizados fazem uma simulação do comportamento de Anahy, permitindo avaliar apenas os pontos relevantes à comunicação entre os nodos.

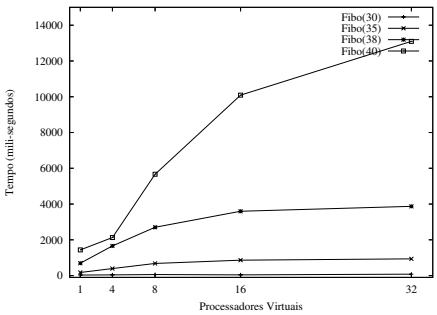

Figura 2: Comportamento do serviço de Mensagens Ativas com diferentes PVs

A Figura 2 apresenta os resultados obtidos em 4 invocações de serviços remotos com diferentes cargas de trabalho, sendo atendidos com diferentes quantidades de processadores virtuais. A arquitetura utilizada são dois nodos bi-processados Xeon 2.8GHz com

2GB de memória RAM cada. Foram considerados como carga a execução seqüencial do algoritmo recursivo de Fibonacci para os valores 30, 35, 38 e 40. A quantidade de processadores virtuais como suporte à execução foi variado entre 1, 4, 8, 16 e 32 unidades para cada invocação de serviço. Em cada uma destas execuções, foram enviadas 16 requisições de trabalho com a mesma carga associada.

Na Figura 2, podemos observar que, conforme aumenta a quantidade de PVs, o tempo aumenta com uma inclinação maior até a quantidade de 16 PVs e, após isso, a inclinação diminui. Como a quantidade de serviços requisitada também é de 16, os testes apontam para o fato de que, em menor número em relação às tarefas requisitadas, os PVs ficam ocupados por mais tempo processando as requisições. Isso faz com que o *overhead* associado à criação dos PVs seja menor e a sobreposição do tempo de comunicação com cálculo efetivo seja melhor aproveitada do que se fosse utilizada uma quantidade maior de PVs. Como Anahy se propõe a explorar eficientemente máquinas SMP normalmente encontradas em aglomerados de computadores, ou seja, com poucas unidades de processamento reais, este comportamento é o esperado e o mecanismo de comunicação implementado mostrou-se não alterar este comportamento.

Conclusão

O uso do mecanismo de Mensagens Ativas mostrou-se adequado para a quantidade de processadores reais a que Anahy se propõe explorar. Este mecanismo está atualmente sendo incorporado ao ambiente de execução Anahy, possibilitando a sua utilização em aglomerados de computadores. Como trabalhos futuros encontra-se o término da integração do mecanismo de comunicação implentado ao Anahy e sua validação.

Referências

[CAR 99] CARISSIMI, A. S. **Le noyau exécutif athapascan-0 et l'exploitation de la multiprogrammation légère sur les grappes de stations multiprocesseurs**. 1999. Thèse de doctorat — Institut National Polytechnique de Grenoble, Grenoble, France.

[CAV 2003] CAVALHEIRO, G. G. H.; REAL, L. C. V.; DALL'AGNOL, E. C. Uma biblioteca de processos leves para a implementação de aplicações altamente paralelas. In: IV WORKSHOP EM SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO, 2003, São Paulo, SP. **Anais...** [S.l.: s.n.], 2003.

[EIC 92] EICKEN, T. von et al. **Active messages**: a mechanism for integrated communication and computation. University of California, Berkeley, CA 94720.

[GRA 69] GRAHAM, R. L. Bounds on multiprocessing timing anomalies. **SIAM Journal on Applied Mathematics**, v.17, n.2, p.416–429, Mar. 1969.

[ROL 2004] ROLOFF, E.; CARISSIMI, A. S.; CAVALHEIRO, G. G. H. Variações de mensagens ativas para aglomerados de computadores. In: ESCOLA REGIONAL DE ALTO DESEMPENHO, 4., 2004, Pelotas - Rio Grande do Sul - Brasil. **Anais...** SBC, 2004. p.289–292.

[VAL 90] VALIANT, L. G. A bridging model for parallel computation. **Commun. ACM**, v.33, n.8, p.103–111, 1990.