MicroVAPI: Utilização da biblioteca DECK em programas Java*

Juliano Foletto Reckziegel†, Rodrigo da Rosa Righi‡, Marcelo Pasin†

† UFSM, Informática CT, Campus UFSM, Santa Maria/RS
‡ PPGC - UFRGS, Av. Bento Gonçalves 9500, Bloco IV, Porto Alegre/RS
reck@inf.ufsm.br, rrrighi@inf.ufrgs.br, pasin@inf.ufsm.br

Introdução

Java tem se tornado uma das linguagens mais utilizadas para a escrita de aplicações, inclusive para aquelas que utilizam programação paralela e distribuída. Isso se deve ao fato dela apresentar as facilidades da programação orientada a objetos, a portabilidade entre diferentes arquiteturas de computadores e uma boa documentação. Aliado a isso, ela oferece mecanismo para trabalhar com múltiplos fluxos concorrentes de execução e com memória distribuída, através dos sistemas de RMI e de soquetes.

Com o intuito de prover mais poder de processamento são utilizadas máquinas paralelas, como os aglomerados (*clusters*). Essa arquitetura é composta de nós que trocam informações através de uma rede dedicada [Baker and Buyya, 1999]. Visando construir um ambiente de programação capaz de proporcionar trocas de mensagens em redes de alta velocidade e integrar aglomerados, foi desenvolvida a biblioteca de comunicação DECK (*Distributed Execution and Communication Kernel*) [Barreto et al., 1998]. Ela possibilita comunicação em redes Myrinet, SCI e VIA, além da tecnologia padrão Ethernet.

Para integrar as vantagens da tecnologia Java com redes de alto desempenho, está sendo implementado o sistema **Aldeia** [da Rosa Righi et al., 2004]. Ele visa a escrita de aplicações Java RMI que tenham como ambiente de execução redes de alta velocidade. Para alcançar esse objetivo, ela utiliza a biblioteca VAPI [Mellanox, 2000], que possibilita comunicação em redes Infiniband [IBTA, 2002]. Procurando agregar mais plataformas de comunicação ao Aldeia, foi desenvolvida uma biblioteca de adaptação chamada **microVAPI**, que tornou possível usar a biblioteca DECK como meio de comunicação. Esse trabalho descreve a biblioteca microVAPI, as vantagens de sua utilização e alguns testes do sistema Aldeia ao utilizá-la.

Sistema Aldeia

O sistema Aldeia busca proporcionar RMI através de redes que forneçam alta vazão e baixa latência de comunicação. A sua organização pode ser visualizada na figura 1. Para prover desempenho, ele implementa classes com a interface de soquetes, servindo-se da interface VAPI. As classes do sistema de soquetes Aldeia possuem métodos nativos,

*Financiamento: CNPq e FIPE

cuja as implementações estão escritas em linguagem C e estão reunidas no Adaptador VAPI.

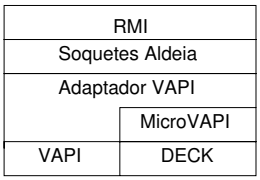

Figura 1: Utilização de DECK e VAPI como suporte a comunicação

Com a finalidade de englobar outras tecnologias de redes de alto desempenho, foi integrado ao Aldeia a biblioteca microVAPI. Ela é encarregada de portar chamadas VAPI no adaptador de baixo nível para utilizarem o DECK, aproveitando assim todas as arquiteturas de redes suportadas por este.

Biblioteca MicroVAPI

A MicroVAPI é uma biblioteca de adaptação que, através do DECK, torna o Aldeia capaz de operar sobre redes SCI, VIA, Myrinet e Ethernet, além da original Infiniband. Isso acontece através da implementação de algumas funções da VAPI. A biblioteca VAPI é muito complexa, pois a tecnologia Infiniband provê diversos recursos, os quais, necessitam chamadas muito específicas para serem habilitados ou desabilitados. A microVAPI implementa 13 das funções da VAPI, somente aquelas necessárias ao adaptador VAPI.

Como as interfaces de chamada DECK e VAPI são ligeiramente diferentes, foi necessário escrever funções de adaptação de uma para outra. Algumas dessas funções nem chegam a chamar a biblioteca DECK, como por exemplo as funções de registro de memória. Por outro lado, a inicialização do DECK se baseia em um *script* de lançamento (deckrun), que foi substituído por funções da microVAPI.

O principal ponto a ser resolvido foi a comunicação. As bibliotecas Infiniband e DECK utilizam paradigmas diferentes para descrever uma troca de mensagem. O DECK utiliza a abstração de caixa de mensagens (*Mail Box* - MB). Cada caixa, em um dado instante, pode somente receber ou enviar mensagens. Para uma caixa de mensagens poder receber dados, ela deve ser criada, já para enviar ela deve clonar a caixa de mensagens que o nó receptor criou. Todavia, Infiniband utiliza um único ponto final de conexão, chamado par de fila (*Queue Pair* - QP). Através de uma única QP, pode-se enviar e receber mensagens. A microVAPI redefine a estrutura de uma QP como sendo composta de duas caixas de mensagens. A figura 2 apresenta essa organização, com o encapsulamento das caixas de mensagens dentro de uma QP.

Para enviar mensagens através do soquetes Aldeia no nível do adaptador VAPI, é necessário a criação de um descritor de requisição. Esse descritor é uma estrutura que possui dois campos principais: um ponteiro para a região onde estão ou devem ser colocados os dados; a quantidade de dados a serem enviados ou recebidos. Com a utilização de descritores VAPI, no momento da chegada dos dados, o receptor sabe a quantidade

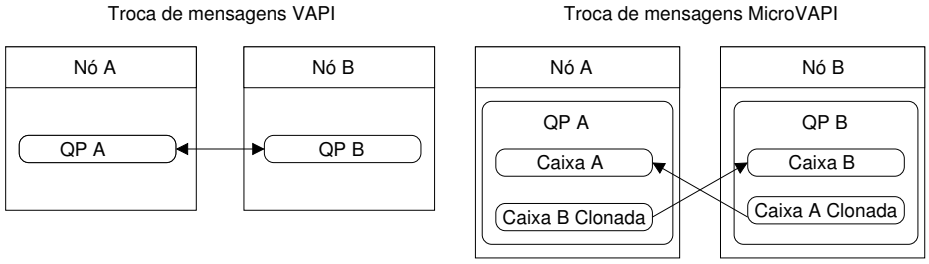

Figura 2: Caixa de Mensagens na MicroVAPI

enviada pelo outro ponto final. Entretanto, o ambiente DECK não possibilita que o receptor saiba a quantidade de dados enviados. Para resolver isso, é empacotado na própria mensagem DECK o tamanho dos dados e logo após, são empacotadas as informações. Da mesma forma, no lado receptor, é desempacotado o tamanho, e logo após os dados efetivamente.

Avaliação

Para demonstrar a potencialidade dos soquetes Aldeia sobre a biblioteca DECK foram desenvolvidas duas aplicações. A primeira envia, de uma máquina para outra, um vetor de inteiros, de modo a testar a eficiência do sistema. A segunda se refere a uma implementação do filtro de mediana para imagens, com o intuito de testar a correta transmissão dos dados.

O ambiente utilizado é composto por dois computadores que possuem um processador Intel Pentium 4 de 2,4GHz com 512 Kb de cache, 768 Mb de memória principal. Foram utilizados o sistema operacional Linux (Gentoo), com Kernel 2.4.26, e a JVM Blackdown versão 1.4.1. As duas máquinas estavam interligadas através de uma rede Ethernet de 100 megabits.

A primeira aplicação realiza a serialização e o envio de um vetor de inteiros. Essa aplicação foi executada sobre dois diferentes cenários. O primeiro se refere a implementação com os soquetes Aldeia, servindo-se da biblioteca DECK versão TCP. O segundo cenário faz uso das classes padrão de soquetes Java, que também utiliza o protocolo TCP. Foi enviado um vetor composto por 10 milhões de elementos inteiros. O sistema operacional utilizado trabalha com o tipo inteiro no qual possui 4 bytes. Assim, totalizando 40 megabytes de dados transferidos. Para uma melhor validação desse teste foi realizada a média aritmética de 250 repetições em cada cenário. Na versão que utiliza soquetes Java foi alcançada uma média de 3699 milisegundos e desvio padrão de 145, enquanto os soquetes Aldeia obtiveram uma média de 3692 milisegundos e desvio padrão 59. Esse teste demonstrou que a microVAPI conseguiu bons resultados ao utilizar DECK para prover comunicação para programas Java.

A segunda aplicação realiza o filtro de mediana, que foi executada novamente nos cenários anteriormente citados. A figura utilizada possui 341 linhas por 306 colunas. A interação entre as máquinas aconteceu da seguinte maneira. A máquina 1 mapeia a figura para uma matriz e envia para o outro ponto várias mensagens. Cada uma dessas contendo

os dados referentes a uma das linhas da matriz original. A máquina 2, após receber todas as linhas, calcula o filtro da mediana para a matriz inteira e retorna, novamente, cada linha da matriz em uma nova mensagem.

Para a segunda aplicação, o tempo final leva em conta o tempo de cálculo do filtro, juntamente com o de comunicação. Este tempo resultante é produzido através da média de também 250 execuções. Nessa aplicação, os soquetes Aldeia apresentaram um tempo final de 716 milisegundos e um desvio padrão de 28. O segundo cenário atingiu o valor de 719 milisegundos e de 50 para o desvio padrão no término da aplicação. De posse dos resultados da execução seqüencial do filtro, foi possível compará-los com os dados obtidos como resultado da aplicação distribuída. Constatou-se que os resultados da execução seqüencial, juntamente com aqueles obtidos ao fazer uso dos soquetes Aldeia e Java padrão, são idênticos.

Conclusão

A integração do DECK ao sistema Aldeia, através da biblioteca microVAPI, possibilitou a programação JAVA sobre diversas tecnologias de redes de uma forma fácil e clara. Sem falar que essa integração proporciona uma programação de alto nível e orientada a objetos sobre um ambiente de comunicação de baixo nível. A biblioteca microVAPI também pode ser útil para portar programas Infiniband, escritos com a VAPI, para as tecnologias suportadas pelo DECK.

Os testes realizados são preliminares e demonstraram, até aqui, um bom resultado dos soquetes Aldeia rodando sobre o ambiente DECK TCP. As otimizações de desempenho realizadas dentro dos soquetes Aldeia e da biblioteca microVAPI, mostraram a viabilidade das suas utilizações para a programação distribuída em Java. A cargo de trabalhos futuros utilizando a microVAPI, pretende-se a avaliação dos soquetes Aldeia sobre o DECK configurado para trabalhar em redes Myrinet.

Referências

[Baker and Buyya, 1999] Baker, M. and Buyya, R. (1999). Cluster computing at a glance. In Buyya, R., editor, *High Performance Cluster Computing*, volume 1, Architectures and Systems, pages 3–47. Prentice Hall PTR, Upper Saddle River, NJ.

[Barreto et al., 1998] Barreto, M. E., Navaux, P. O. A., and Rivière, M. P. (1998). DECK: a new model for a distributed executive kernel integrating communication and multiheading for support of distributed object oriented application with fault tolerance support. In *Congreso Argentino de Ciencias de la Computacion*, volume 2, pages 623–637, Argentina.

[da Rosa Righi et al., 2004] da Rosa Righi, R., Pasin, M., and Navaux, P. O. A. (2004). Aldeia: Invocação remota e assíncrona de métodos sobre Infiniband e DECK. In *Quinto Workshop em Sistemas Computacionais de Alto Desempenho*, Foz do Iguaçu - PR.

[IBTA, 2002] IBTA (2002). Inifiniband architecture specification, vol. 1, release 1.1, 2002. Disponível em http://www.infinibandta.org/specs.

[Mellanox, 2000] Mellanox (2000). Introduction to Infiniband. Technical report, Mellanox Technologies Inc., Santa Clara, California (EUA).