

# Porte de soquetes Java para operar sobre DECK e Infiniband\*

Rodrigo da Rosa Righi<sup>†</sup>, Philippe Olivier Alexandre  
Navaux<sup>‡</sup>, Marcelo Pasin<sup>‡</sup>

<sup>†</sup> PPGC-II-UFRGS, Av. Bento Gonçalves 9500, Bloco IV, Porto Alegre/RS

<sup>‡</sup> UFSM, Informática CT, Campus UFSM, Santa Maria/RS  
{rrrighi, navaux}@inf.ufrgs.br, pasin@inf.ufsm.br

## Introdução

Java tem emergido com uma linguagem bastante utilizada para a escrita de aplicações paralelas e distribuídas. Ela torna mais fácil a escrita desses tipos de aplicações, pois além das vantagens da orientação a objetos, ela possui mecanismos nativos para tratar com fluxos concorrentes de execução, compartilhamento de memória e troca de mensagens. Mais especificamente, essa última característica pode ser dividida em sistemas de RMI e de soquetes Java. O RMI impõe um tipo específico de interação: a chamada e retorno de procedimento. Por outro lado, a comunicação com soquetes permite a construção de formas de interação totalmente genéricas.

A interface de soquetes Java possui classes para a conexão e comunicação entre objetos e a implementação atual usa o protocolo TCP/IP. Este protocolo foi construído para proporcionar confiabilidade na troca de dados para ambientes heterogêneos e largamente distribuídos. Para isso, ele impõe sobrecargas de *software* que vão de encontro com a obtenção de alto desempenho em sistemas controlados, como os aglomerados (*clusters*). Com o objetivo de adaptar os soquetes Java para operar sobre redes de alto desempenho, capazes de proporcionar alta vazão e baixa latência de comunicação, foi desenvolvido o sistema de **soquetes Aldeia**[da Rosa Righi et al., 2004]. O presente artigo apresenta uma breve descrição sobre esse sistema e a sua utilidade para a escrita de aplicações Java.

## Sistema de Soquetes Aldeia

Os soquetes Aldeia oferecem troca de mensagens em Java sobre diferentes equipamentos de interconexão. Ele possibilita comunicação em redes Myrinet, SCI, VIA e Ethernet, através da biblioteca DECK[Barreto, 2000], e Infiniband, ao fazer uso da biblioteca VAPI[Mellanox, 2000]. Infiniband é uma arquitetura de interconexão de alta velocidade, que oferece primitivas de comunicação assíncrona em nível de usuário. Nenhuma dessas duas bibliotecas possui primitivas de programação em alto nível, muito menos orientadas a objetos. Nesse sentido, os soquetes Aldeia escondem os detalhes das bibliotecas de baixo nível, oferecendo uma interface popular e versátil como a dos soquetes Java.

O sistema em questão é composto de três conjuntos de classes: (i) para a conexão entre computadores (AldeiaSocket e AldeiaServerSocket); (ii) para o envio e recepção de dados (AldeiaOutputStream e AldeiaInputStream); (iii) para a configuração do ambiente (AldeiaConfiguration). Por razões de desempenho, todas estas classes fazem uso

---

\*Financiamento: CNPq (processo 552256/02-1), FIPE e projeto Labtec-DELL

de métodos nativos. Cada um deles possui a sua implementação escrita em linguagem C e a reunião dessas funções formam o Adaptador VAPI, que é responsável por interagir diretamente com a biblioteca de baixo nível. Com o intuito de utilizar o DECK, foi construída a biblioteca microVAPI, que implementa as funções da VAPI necessárias para os métodos nativos das classes citadas, servindo-se de chamadas à biblioteca DECK.

O sistema de soquetes Aldeia foi elaborado para minimizar as cópias de memória e para fornecer alto desempenho na comunicação. As classes do segundo conjunto são extensões de classes Java, que originalmente definem vários métodos para realizar a troca de mensagens. Os soquetes Aldeia sobrescrevem esses métodos, de modo a reduzir chamadas encadeadas de métodos e operações de E/S dentro de estruturas iterativas. Além disso, as classes Aldeia usam as funcionalidades do pacote `nio` do Java, que permitem que objetos Java acessem diretamente dados alocados em modo nativo, diminuindo o número de cópias necessárias para a comunicação.

Para utilizar os soquetes Aldeia basta adicionar, nas classes `Socket` e `ServerSocket`, o prefixo `Aldeia`. A partir daí, toda a comunicação passa a utilizar uma das bibliotecas suportadas por esse sistema. Os canais de dados dos soquetes Aldeia substituem perfeitamente as classes Java que manipulam a serialização de objetos e as que escrevem e lêem dados de tipos primitivos. Assim, pode-se reunir as facilidades de programação distribuída em Java com comunicação eficiente de alto desempenho.

## Considerações finais

O sistema de soquetes Aldeia proporciona uma interface que o programador já está acostumado a usar e pode ser facilmente integrado a outros sistemas Java que utilizem soquetes, devido a característica de polimorfismo dessa linguagem. Esse sistema integra o projeto Aldeia, que objetiva a escrita de aplicações RMI. A próxima etapa da pesquisa envolve a integração dos soquetes Aldeia com um sistema RMI. Nesse sentido, está sendo analisado o sistema ProActive [Caromel et al., 1998]. Também como trabalho futuro, tem-se a avaliação de desempenho dos soquetes Aldeia e a sua execução com a versão do DECK instrumentado. Essa versão gera rastros das operações de E/S que podem ser analisados em uma ferramenta de visualização.

## Referências

- [Barreto, 2000] Barreto, M. E. (2000). DECK: um ambiente para programação paralela em agregados de multiprocessadores. Master's thesis, PPGC - UFRGS.
- [Caromel et al., 1998] Caromel, D., Klauser, W., and Vayssiere, J. (1998). Towards seamless computing and metacomputing in Java. In *Concurrency Practice and Experience*, volume 10, pages 1043–1061. Wiley & Sons, Ltd.
- [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. aceito para publicação.
- [Mellanox, 2000] Mellanox (2000). Introduction to Infiniband. Technical report, Mellanox Technologies Inc., Santa Clara, California (EUA).