

# Comunicação entre processos através de memória compartilhada para o DECK

Caciano S. Machado<sup>1</sup>  
Rafael B. Ávila<sup>2</sup>  
Philippe O. A. Navaux<sup>3</sup>

388705

## Resumo

As arquiteturas de clusters de multiprocessadores permitem a exploração da memória compartilhada na comunicação entre processos executando no mesmo nodo. Este documento aborda os aspectos mais importantes do mecanismo de troca de mensagens através de memória implementado para o DECK.

## 1 Introdução

DECK (BARRETO; NAVAUUX; RIVIÈRE, 1998) (BARRETO, 2000) é uma biblioteca de programação paralela para clusters de alto desempenho (BUYA, 1999) que possui suporte para várias tecnologias de interconexão. Visando buscar o máximo desempenho que os clusters podem oferecer foi implementado um mecanismo alternativo que explora a memória das máquinas em eventuais trocas de mensagens entre processos de um mesmo nodo.

## 2 DECK shmem

Cada processo cria um segmento de memória que é anexado a cada um dos outros processos da aplicação no mesmo nodo para que possa receber as mensagens dos processos locais. A criação e anexação dos segmentos de memória é feita com as rotinas de IPC do Linux (CURRY, 1996).

Um dos fatores responsáveis pelo bom desempenho do dispositivo é a utilização de um mecanismo de exclusão mútua menos oneroso para o controle das estruturas de dados, as primitivas de *spin-lock* do Kernel do Linux (SCHIMMEL, 1994). No entanto o uso de *spin-locks* acaba limitando o número de processos de um nodo ao número de processadores disponíveis

<sup>1</sup>caciano@inf.ufrgs.br Bolsista CNPq

<sup>2</sup>avila@inf.ufrgs.br Bolsista CAPES

<sup>3</sup>navaux@inf.ufrgs.br

no nodo, pois no caso em que se tem um número maior de processos que processadores o mecanismo de *spin-lock* poderá prejudicar a execução dos outros processos em certas circunstâncias. A utilização de exclusão mútua com o desescalonamento dos processos é uma alternativa possível para o caso de mais de  $n$  processos numa máquina com  $n$  processadores.

Este dispositivo foi adaptado para ser utilizado em conjunto com o dispositivo TCP de *mail boxes*. Temos então como resultado dois tipos de rotinas de envio e recebimento de mensagens. O gerenciamento de qual tipo de comunicação deve ser utilizado é feito com o armazenamento de dados redundantes sobre a localização das *mail boxes* nas suas estruturas. Estas informações são atualizadas durante as operações de clonagem das *mail boxes* e consultadas nas operações de envio e recebimento de mensagens.

Nos testes de desempenho as versões do dispositivo *shmem* do DECK apresentaram ótimos resultados. Em avaliações com aplicações *ping-pong* os tempos de latência e largura de banda melhoraram consideravelmente (SANTOS MACHADO, 2003).

O estudo de técnicas para mecanismos de exclusão mútua e cópia de dados na memória são as áreas mais críticas dos dispositivos de troca de mensagens através de memória pois são principais fatores responsáveis pelo *overhead*. A utilização de dispositivos como o *shmem* do DECK permitem uma exploração mais eficiente de clusters de nodos multiprocessados, logo seu desenvolvimento, estudo e aprimoramento se fazem necessários para tais arquiteturas.

## Referências

- BARRETO, M. E. *DECK: Um Ambiente para Programação Paralela em Agregados de Multiprocessadores*. Dissertação (Mestrado) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- BARRETO, M. E.; NAVAU, P. O. A.; RIVIÈRE, M. P. DECK: a new model for a distributed executive kernel integrating communication and multithreading for support of distributed object oriented application with fault tolerance support. In: *Trabajos Seleccionados del 4o.* Neuquén, Argentina: Neuquén, Universidad Nacional del Comahue, 1998. v. 2, p. 623–637.
- BUYA, R. (Ed.). *High Performance Cluster Computing: Architectures and Systems*. Upper Saddle River: Prentice Hall PTR, 1999. 849 p.
- CURRY, D. A. *UNIX Systems Programming for SVR4*. Sebastopol: O'Reilly, 1996.
- SANTOS MACHADO, C. dos. Troca de mensagens através de memória compartilhada para o ambiente DECK. In: *Anais: ERAD 2003 - Escola Regional de Alto Desempenho*. Santa Maria, Brasil: Universidade Federal de Santa Maria, 2003. p. 285–288.
- SCHIMMEL, C. *UNIX Systems for Modern Architectures: symmetric multiprocessing and caching for kernel programmers*. Massachusetts: Addison-Wesley, 1994.