Implementação da biblioteca de comunicação DECK sobre o padrão de protocolo de comunicação em nível de usuário VIA

Leonardo A. de Paula e Silva [*], Philippe O. A. Navaux

UFRGS - Av. Bento Gonçalves, 9500, Bloco IV, Porto Alegre/RS
lapys@inf.ufrgs.br,navaux@inf.ufrgs.br

Resumo

Redes do tipo SAN são melhores exploradas por protocolos de comunicação desvinculados do *kernel* do sistema operacional e que evitem cópias intermediárias entre o espaço de endereçamento do usuário e do *kernel* , chamados protocolos em nível de usuário. Neste artigo, apresenta-se o DECK e como as primitivas da API da arquitetura VIA foram utilizadas na implementação da μDECK e os resultados esperados.
Palavras-chave: programação paralela, *cluster computing*, DECK e VIA.

Introdução

Com o surgimento de redes que oferecem baixa latência de comunicação e alta de largura de banda, conhecidas como SANs (*System Area Networks*), os protocolos de comunicação integrados ao *kernel* do sistema operacional, como TCP/IP, tiveram de ser substituídos por outros protocolos que pudessem melhor explorar as potencialidades destas redes. Na concepção destes novos protocolos, nomeados pela literatura por protocolos em nível de usuário, foram utilizadas técnicas de **cópia zero**, para evitar cópias intermediárias entre o espaço de endereçamento do usuário e do *kernel*, e de **sobrepasso do sistema operacional**, para permitir que a aplicação do usuário acesse o dispositivo de rede sem a necessidade de chamadas ao sistema operacional. A utilização de protocolos de comunicação em nível de usuário é bastante interessante na implementação de abstrações de comunicação de mais alto nível, como o DECK.

DECK – *Distributed Execution and Communication Kernel* – é um ambiente de programação paralela, desenvolvido no GPPD do Instituto de Informática da UFRGS, que proporciona o emprego de suas primitivas em aplicações paralelas, pela filosofia SPMD, através da sobreposição de multiprogramação com comunicação [Barreto et al., 1998].

VIA, *Virtual Interface Architecture* [COMPAQ et al., 1997], é um padrão de protocolo de comunicação em nível de usuário que oferece a abstração de **Interface Virtual** (VI) que dispensa a necessidade de chamadas de sistema para realizar operações de comunicação. Assim, o dispositivo de rede pode ser acessado de forma protegida e diretamente pelo processo usuário através de uma VI. Cada VI representa um ponto de comunicação. Um processo pode ter múltiplas VIs relacionados a um dispositivos de rede.

Estudou-se a arquitetura VIA, suas primitivas e de sua semântica, com o objetivo de utilizá-la como protocolo de comunicação de baixo nível para implementação da

[*]Bolsa fomentada pelo CNPq. Apoio do convênio LabTeC/DELL

camada μDECK da biblioteca DECK. A exploração das características de protocolo de nível de usuário de VIA permitirá que a biblioteca DECK alcance menores latências de comunicação e um melhor aproveitamento da largura de banda oferecida pelas redes as quais o padrão VIA é implementado.

A biblioteca de comunicação DECK/VIA

O ambiente DECK está dividido em camada de serviços e μDECK. A camada μDECK é dependente da tecnologia de rede, sendo composta pelos módulos de iniciação, de mensagem e de caixa postal, os quais são implementados utilizando a API VIA. O módulo de iniciação prepara o dispositivo de rede para posterior comunicação, através do ajuste dos atributos do endereço de rede, da iniciação de serviço de nomes VIA e da iniciação variáveis de condição e *mutexes*. Ao módulo de mensagens, estão relacionadas as primitivas de preparação e alocação dos dados a serem comunicados. Para cada *buffer* de mensagens criado com a chamada da primitiva `deck_msg_create`, uma região de memória do tamanho especificado é criada, cadastrada e relacionada a um descritor. A primitiva `deck_msg_destroy` providencia o descadastramento da região de memória e liberação do descritor relacionado à mensagem destruída.

O módulo de caixa postal lida com primitivas relacionadas ao canal de comunicação entre os processos e *threads* DECK, através da abstração de caixas postais. Ao criar uma caixa postal, pela invocação de `deck_mbox_create`, um conjuto de VIs é criado e cada uma é colocada em estado de espera de conexão em uma *thread* diferente, ou seja, no aguardo de uma clonagem, segundo a semântica de DECK. O processo que deseja clonar uam caixa postal, faz uma chamada a primitiva `deck_mbox_clone`, a qual cria uma nova VI no processo invocador e requisita uma conexão desta a uma das VIs anteriormente criadas pelo dono da caixa postal. A primitiva `deck_mbox_post` coloca o descritor de envio na fila de envio da VI do processo remetente. De forma análoga, a primitiva `deck_mbox_retrv`, coloca o descritor de recebimento na fila de recebimento da VI do processo destinatário. Note que esta comunicação é assíncrona, assim como sugere a abstração de caixas postais. A primitiva `deck_mbox_destroy` providencia a desconexão e a destruição das VIs utilizadas na comunicação entre os processos.

O DECK/VIA está sendo implementado utilizando a implementação de VIA para redes Myrinet, denominada VI-GM e atualmente é a única implementação de VIA disponível e mantida. Os resultados permitirão confirmar que o emprego do protocolo de comunicação em nível de usuário do padrão VIA, traz uma diminuição significativa da latência de comunicação e melhor aproveitamento da largura de banda da rede para qual o padrão foi implementado.

Referências

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, Neuquém, Argentina.

COMPAQ, INTEL, and MICROSOFT (1997). Virtual interface architecture specification version 1.0. Dezembro 1997. Disponível em `http://developer.intel.com/design/servers/vi/developer/ia_imp_guide.htm`.