Chamadas de Procedimentos Remotos no dNFSp

Francieli Zanon Boito, Rodrigo Virote Kassick,
Everton Hermann, Philippe O. A. Navaux

Instituto de Informática - UFRGS
fzboito, rvkassick, ehermann, navaux@inf.ufrgs.br

1 Introdução

Com a possibilidade de executar aplicações cada vez mais complexas, a tendência é o aumento do volume de dados envolvidos. Obter proveito da arquitetura distribuída pode melhorar o desempenho do armazenamento dessas informações. Isso pode ser conseguido colocando os dados em máquinas diferentes e acessando-os em paralelo, através de um sistema de arquivos distribuído. Um exemplo de sistema de arquivos para cluster é o dNFSp[1], que é baseado no sistema NFSp [3]. Em suas rotinas, o dNFSp precisa executar comandos remotamente. O mecanismo utilizado até então, o *rsh*, é um programa que permite executar comandos *shell* em outro computador através de uma rede. No entanto, esse mecanismo mostrou-se pouco eficiente, tornando necessária a sua substituição por algum outro método.

2 A mudança no dNFSP

O uso de *rsh* precisou ser revisto, pois tornava o sistema de arquivos menos eficiente. Isso acontecia principalmente pelo fato de que chamar um programa , que por sua vez executará outro, geralmente será mais lento do que simplesmente chamar uma subrotina. Além de perder eficiência, o uso desse mecanismo diminuía a portabilidade, pois colocava o servidor de *rsh* (*rshd*) como um pré-requisito a mais para a instalação do dNFSP.

O RPC (*Remote Procedure Call*) é um mecanismo utilizado em aplicações que utilizam a arquitetura cliente-servidor. Permite que um programa, rodando em uma máquina, solicite que outro programa, rodando em outra máquina, execute uma determinada função. Pode ser implementado sobre diferentes protocolos de transporte. Se sobre TCP (*Transmission Control Protocol*), não é necessário se preocupar com retransmissões, duplicatas, etc. No entanto, se implementado sobre UDP (*User Datagram Protocol*), esses aspectos passam a merecer atenção. Além de melhorar o desempenho do sistema de arquivos, outro motivo para usar esse mecanismo é manter um bom nível de abstração, facilitando a manutenção do código.

No dNFSp, o RPC foi implementado sobre UDP, pois o uso de TCP pode tornar a implementação mais complexa, uma vez que o sistema de arquivos precisa manter um estado da comunicação. O UDP ainda tem a vantagem de ser mais rápido, pois, ao contrário do TCP, não faz verificações nem estabelece sessões. Essa escolha implica preocupações

com a confiabilidade da transmissão dos dados. Esse tratamento tem que ser feito no código da aplicação.

O DAB (*Distributed Andrew Benchmark*)[2] tem por objetivo medir o desempenho nos vários nós acessando o sistema. É composto por cinco etapas, e só passa de uma para a outra quando todos os clientes tiverem terminado a atual. O desempenho das duas versões do dNFSp foram medidos com o DAB. Seus resultados para 4 metaservidores e 4 servidores de dados, com 16 clientes acessando o sistema estão representados na figura abaixo.

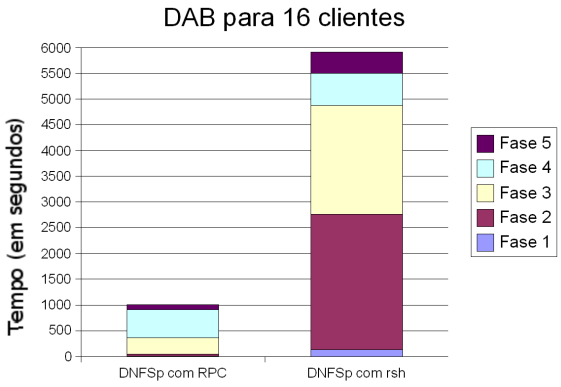

Figura 1: Comparativo entre as versões do dNFSp

3 Conclusão e Trabalhos Futuros

Feita essa alteração, e observados seus bons resultados, o próximo passo é tentar melhorar o protocolo. A cada operação realizada, todos os metasservidores devem ser consultados para determinar a localização do arquivo. A melhoria pode ser feita buscando diminuir o número de mensagens trocadas, acrescentando desempenho ao sistema de arquivos.

Referências

[1] R. B. Ávila. *Uma Proposta de Distribuição Servidor de Arquivos em Clusters*. PhD thesis, PPGC-UFRGS/ID-IMAG, Porto Alegre/Grenoble, 2005.

[2] R. Kassick, C. Machado, E. Hermann, R. Ávila, P. Navaux, and Y. Denneulin. Evaluating the performance of the dnfsp file system. In *5th IEEE International Symposium on Cluster Computing and the Grid*. IEEE Computer Society Press, 2005.

[3] P. Lombard and Y. Denneulin. nfsp: A distributed nfs server for clusters of workstations. In *16th International Parallel and Distributed Processing Symposium (IPDPS 2002)*. IEEE Computer Society, 2002.