

Distribuição de um Sistema de Arquivos para um Ambiente de Cluster

Rejane Elsing¹, Luis Fernando Friedrich¹

¹CPGCC – Universidade Federal de Santa Catarina (UFSC)
Florianópolis - SC – Brasil

rejane@inf.ufsc.br, fernando@inf.ufsc.br

Resumo. *Este artigo descreve a distribuição de um sistema de arquivos num ambiente de cluster. A partir de um servidor de arquivos centralizado foi implementado um servidor de arquivos multithreaded, que foi disponibilizado em dois nós de trabalho do cluster, visando distribuir e aumentar a disponibilidade do serviço. Cada nó do cluster disponibilizava parte de seus arquivos para o espaço de nomes compartilhado. Conseqüentemente, houve a necessidade de projetar e implementar um servidor de diretório multithreaded, com o objetivo de identificar qual servidor de arquivos, no ambiente do cluster, estava disponibilizando o arquivo que o processo cliente desejava manipular, sem que este soubesse sua localização física.*

1 Introdução

A necessidade de interconexão entre computadores localizados em diversos locais diferentes e que executam um trabalho em comum, com necessidade de maior troca de informações, maior velocidade de processamento e maior disponibilidade de informações, provoca o surgimento de sistemas cada vez mais transparentes, escaláveis, confiáveis e com maior desempenho. Essa necessidade motivou o surgimento de pesquisas e projetos que levaram ao aparecimento de novos ambientes capazes de reduzir custos e otimizar a obtenção de resultados.

No Laboratório de Computação Paralela e Distribuída (LaCPaD) da Universidade Federal de Santa Catarina (UFSC) foi desenvolvido o projeto de montagem de um *cluster* de computadores. Inicialmente, o *cluster* contava com um servidor de arquivos centralizado que tinha a função de atender requisições de processos clientes que desejavam acessar e manipular arquivos (PLENTZ, 2001). Baseado nesse cenário iniciou-se a pesquisa, projeto e implementação de uma abordagem distribuída para um sistema de arquivos num ambiente de *cluster*. Para atingir este objetivo, optou-se por transformar o servidor de arquivos centralizado num servidor de arquivos *multithreaded* e disponibilizar este em mais de um nó de trabalho do *cluster* para aumentar sua disponibilidade. O servidor de arquivos foi disponibilizado em dois nós de trabalho, os quais disponibilizariam parte dos arquivos locais para compor o espaço de nomes, o qual seria visto de maneira uniforme por todos os processos clientes do *cluster*, sendo chamado de espaço de nomes compartilhado.

Para que os processos clientes pudessem acessar um arquivo do espaço de nomes compartilhado seria necessário implementar algum serviço que realizasse a localização deste no *cluster*. Desse modo, o outro objetivo desse trabalho foi implementar um

pdfMachine - is a pdf writer that produces quality PDF files with ease!

Get yours now!

"Thank you very much! I can use Acrobat Distiller or the Acrobat PDFWriter but I consider your product a lot easier to use and much preferable to Adobe's" A.Sarras - USA

servidor de diretório *multithreaded*, para identificar o servidor de arquivos que estaria disponibilizando o arquivo que o processo cliente desejava acessar. O servidor de diretório seria disponibilizado num único nó de trabalho do *cluster*, de maneira dedicada, e deveria permitir que um cliente acessasse um arquivo no ambiente distribuído, sem que o cliente tomasse conhecimento de sua localização física, ou seja, a localização do arquivo era transparente para o cliente.

2 Sistemas Distribuídos

No início dos anos oitenta, começaram a surgir novas tecnologias, como o desenvolvimento de microprocessadores com um poder de processamento cada vez maior e a invenção das redes locais de alta velocidade. A aplicação destas duas tecnologias resultou na construção de sistemas de computação poderosos, compostos por um grande número de processadores, ligados através de redes de alta velocidade, chamados de sistemas distribuídos (TANENBAUM, 2001).

Em geral, um sistema distribuído é uma coleção de computadores independentes que aparece para seus usuários como um único sistema. As diferenças entre os vários computadores e o modo como eles se comunicam, são escondidos dos usuários (TANENBAUM, 2001).

Segundo TANENBAUM (2003) entre os fatores que favoreceram o surgimento de sistemas distribuídos tem-se: o uso de microprocessadores que ofereciam uma melhor relação de preço e desempenho do que os *mainframes*; maior interoperabilidade das aplicações que envolvem máquinas separadas fisicamente; maior confiabilidade e tolerância à falhas - se uma máquina falha não prejudica todo o sistema; a possibilidade de crescimento do poder de computação; a possibilidade de compartilhamento de dados, de dispositivos e de carga de trabalho, de forma controlada; e a capacidade de conectar usuários e recursos facilmente, escondendo o fato que recursos são distribuídos através da rede.

Segundo HOCHSTETLER (2003) quando um sistema distribuído é composto por um conjunto de computadores - também chamados de nós - com somente um processador ou vários processadores, interconectados por uma rede de comunicação de alta velocidade, que dão a impressão de ser um único computador, temos o que nós chamamos de um *cluster* de computadores.

Conforme VAZ (2003) existem vários tipos de *clusters*, os quais se diferenciam pelo tipo e número de processadores, o tamanho e velocidade da memória *cache*, o tipo e a velocidade da memória principal, a velocidade do barramento, a qualidade das interfaces de rede, cabeamentos, *switches*, entre outros.

As principais vantagens encontradas no uso de *clusters* são a possibilidade de tornar o sistema mais escalável, tolerante à falhas, de menor custo e independente de fornecedores. Eles podem ser vistos como um conjunto de servidores agrupados, com a intenção de aumentar o desempenho, a disponibilidade ou a facilidade do gerenciamento (VAZ, 2003).

3 Sistemas de Arquivos Distribuídos

Segundo COULOURIS (2001, p. 309-350) um sistema de arquivos distribuído tem o objetivo de fornecer os mesmos serviços e recursos de um sistema de arquivos convencional, pelo menos na visão dos clientes que o acessam, com a diferença que o sistema de arquivos distribuído pode ser acessado de qualquer máquina da rede. Ele suporta o compartilhamento de informações na forma de arquivos e o compartilhamento de recursos de hardware, como armazenamento em disco, através da rede.

Um serviço de arquivos bem projetado provê acesso a arquivos armazenados em servidores remotos, com desempenho e confiabilidade similar a arquivos armazenados em discos locais. Atualmente, os serviços de arquivo tentam explorar a maior largura de banda entre redes locais e novos modos de organização em disco, para alcançar maior desempenho, tolerância à falhas, e alta escalabilidade dos sistemas de arquivos (COULOURIS, 2001, p. 309-350).

Segundo TANENBAUM (2001) os principais requisitos procurados para um sistema de arquivos distribuído são praticamente os mesmos procurados pelos sistemas distribuídos de modo geral, como: transparência, controle de concorrência, replicação de arquivos, heterogeneidade de sistemas operacionais e hardware, tolerância à falhas e segurança.

Segundo KON (1996) os principais componentes de um sistema de arquivos distribuído são o serviço de arquivos, o serviço de diretórios e o serviço de nomes. O serviço de arquivos é responsável por operações relacionadas a arquivos individuais, como leitura, gravação ou remoção de arquivos, mantendo a integridade das operações realizadas nos arquivos. Por exemplo, quando uma aplicação altera algum arquivo, todas as demais aplicações que estiverem acessando o arquivo devem perceber essa alteração o mais rápido possível (KON, 1996). O serviço de diretório, por sua vez, está relacionado com operações em diretórios, como a criação e manutenção de diretórios, a adição e remoção de entradas no diretório, entre outras (KON, 1996). Já o serviço de nomes realiza operações para mapear nomes de arquivos, identificar sua localização física e seus atributos, entre outras (KON, 1996).

4 O Modelo Distribuído do Sistema de Arquivos

O principal objetivo deste trabalho foi distribuir o sistema de arquivos num ambiente de *cluster*. O primeiro passo para distribuir o sistema de arquivos foi disponibilizar o serviço de arquivos em vários nós de trabalho dedicados, onde cada nó, disponibilizava parte dos arquivos locais para serem compartilhados por todos os processos clientes do *cluster*.

Estando o serviço de arquivos disponível em vários nós de trabalho, era necessário implementar algum serviço que permitisse identificar, de forma transparente para o cliente, no ambiente do *cluster*, qual dos servidores disponibilizava determinado arquivo. Para isso, propôs-se a criação de um servidor de diretório.

O modelo arquitetural proposto para distribuir o sistema de arquivos no ambiente do *cluster* é apresentado na Figura 1. O ambiente montado era composto por 4 microcomputadores completos, chamados de nós de trabalho, os quais eram interligados por uma rede conectada a um *switch*, a qual representa a rede de trabalho. Os principais

componentes dessa arquitetura são os processos clientes, a interface do sistema, a interface da rede de trabalho, os servidores de arquivos e o servidor de diretório. Cada um deles é descrito a seguir.

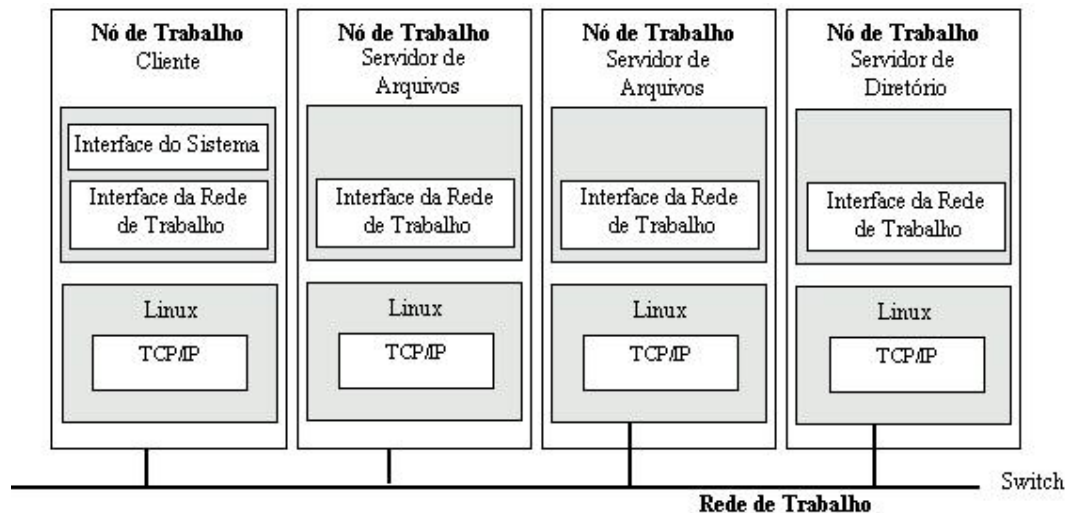


Figura 1 Modelo distribuído do sistema de arquivos.

4.1 Processos clientes

A Figura 1 mostra somente um processo cliente, porém, num ambiente distribuído podem existir vários processos clientes fazendo requisições por serviço ao mesmo tempo. Quando um processo cliente realiza uma chamada ao sistema, esta é interceptada e tratada pela interface do sistema.

Cada processo cliente possui uma *cache* local, que tem como objetivo diminuir a quantidade de conexões estabelecidas entre os processos clientes e o servidor de diretório, reduzindo a carga do servidor, aumentando o desempenho e a escalabilidade do sistema. A estrutura definida para a *cache* local possui dois campos: o prefixo, que indica o nome do caminho desde a raiz até o primeiro nível da árvore hierárquica, e o número IP do servidor de arquivos que disponibiliza a entrada.

4.2 Interface do Sistema

Na implementação do ambiente distribuído, a interface do sistema foi implementada pela biblioteca *libcsa*, que implementa grande parte das chamadas de sistema relativas ao sistema de arquivos do Linux. A interface é composta por um conjunto de operadores que representam algumas das chamadas de sistema executadas pelos processos clientes. As chamadas que foram implementadas na interface e que constituem a biblioteca *libcsa* são: *open*, *r_close*, *read*, *write*, *link*, *unlink*, *symlink*, *truncate*, *lseek*, *ftruncate*, *chdir*, *rename*, *mkdir*, *rmdir*, *mknod*, *chown*, *dup*, *creat*, *access*, *stat*, *sync*, *chmod* e *utime*. As outras chamadas de sistema que não foram implementadas nesta interface são executadas localmente no cliente, a partir da biblioteca *libc* original do Linux.

Para a execução das chamadas de sistema realizadas pelos clientes, os operadores e as funções da biblioteca *libcsa* interagem com os servidores de arquivos e com o servidor de diretório, através da troca de mensagens. Para cada chamada de sistema realizada por um processo cliente a interface intercepta essas chamadas, identifica o servidor que contém o arquivo que será manipulado pela chamada e monta uma mensagem, contendo a identificação da chamada e seus argumentos, a qual é enviada ao servidor apropriado. Todo nó de trabalho que roda um processo cliente necessita ter a interface do sistema.

4.3 Interface da Rede de Trabalho

Na arquitetura proposta, a comunicação entre processos clientes e servidores é realizada pela interface da rede de trabalho, que foi implementada com *sockets* TCP/IP (Protocolo de Controle de Transmissão / Protocolo da Internet).

A interface implementa três operadores que são descritos abaixo:

- ***s_Send***: Usado para enviar mensagens de requisição de serviço dos processos clientes para os processos servidores e para enviar mensagens de resposta dos processos servidores para os processos clientes.
- ***s_Receive***: Usado pelos processos clientes para receber mensagens de resposta dos processos servidores. Essas mensagens, geralmente, contêm as respostas das execuções dos servidores.
- ***ss_ReceiveAny***: Usado por processos servidores para receber mensagens de requisição de serviço, tanto de processos clientes e servidores.

4.4 Servidor de Arquivos

Considerando um *cluster* de computadores, com somente um nó disponibilizando o serviço de arquivos e atendendo todas as requisições clientes do *cluster*, quando a quantidade de requisições clientes fica elevada, o servidor de arquivos perde desempenho, pois o tempo gasto com leitura ou escrita em disco é bastante significativo.

Como o principal objetivo deste trabalho era distribuir o sistema de arquivos no ambiente do *cluster*, o serviço de arquivos foi disponibilizado, inicialmente, em dois nós de trabalho. Isso permitiu a distribuição da carga de requisições clientes, melhorando a disponibilidade dos servidores, pois cada servidor ficou responsável por gerenciar uma parte do espaço de nomes compartilhado.

Cada um dos servidores de arquivos constitui um processo, que faz a inicialização das estruturas de dados e fica em laço, aguardando que os processos clientes estabeleçam conexões. Para cada conexão estabelecida é criada uma *thread* que fica aguardando mensagens de requisição de serviço.

4.5 Servidor de Diretório

O principal objetivo em implementar um servidor de diretório foi permitir que um processo cliente, de qualquer nó de trabalho do *cluster*, realizasse operações sobre arquivos ou diretórios que compõem o espaço de nomes compartilhado, sem que ele

conhecesse a localização física dos mesmos. O processo cliente não precisa saber em qual dos nós de trabalho está localizado o arquivo que ele deseja manipular. Ele vê um único espaço de nomes, onde não há indícios da localização física dos arquivos. Assim, pode-se dizer que o sistema implementa transparência de localização para o cliente.

O servidor de diretório é um processo que se encontra localizado num único nó de trabalho. Ele conhece o espaço de nomes compartilhado somente até o primeiro nível da árvore hierárquica, mantendo na Estrutura de Diretório da memória principal os prefixos das entradas de diretório disponibilizadas por cada servidor de arquivos, para serem compartilhadas pelos processos clientes do *cluster*, e seu IP. Isso é suficiente para determinar o servidor de arquivos que disponibiliza determinada informação. Por exemplo, o servidor de arquivos com número IP 150.164.112.76 está disponibilizando para o espaço de nomes compartilhado os prefixos */etc* e */home*.

Assim, o servidor de diretório constitui um processo, que faz a inicialização das estruturas de dados e fica em laço, aguardando que os processos clientes estabeleçam conexões. A comunicação é realizada usando *sockets* TCP/IP.

5 Testes Realizados

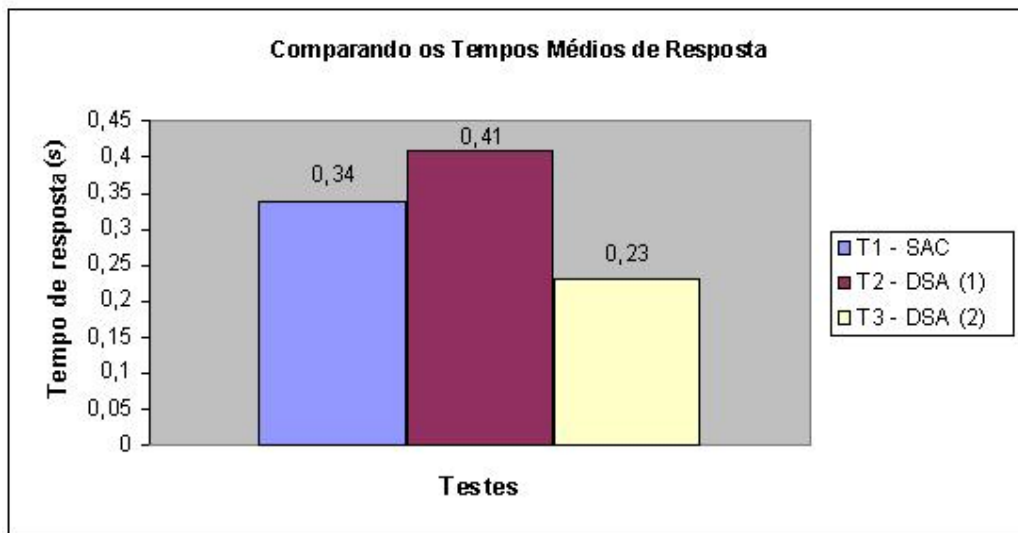
Para verificar se os objetivos do trabalho foram atingidos, resolveu-se fazer vários testes e analisar os resultados. Utilizou-se como referência o *Benchmark* Bonnie (BONNIE, 1996), que implementava funções que permitiam analisar a quantidade de *bytes* transferidos por segundo, a utilização da CPU, o tempo do sistema e o tempo de CPU.

Entre os operadores implementados pela *libcsa*, os mais requisitados pelos processos clientes são: *open*, *read*, *lseek*, *write* e *r_close*. Assim, acrescentou-se ao *Benchmark* Bonnie linhas de instruções que permitiram analisar o desempenho do sistema de arquivos distribuído quando essas cinco chamadas eram executadas.

Na realização dos testes, analisou-se o tempo de resposta (tempo decorrido) desde o instante que um processo cliente executava um determinado número de chamadas até o momento que ele recebia as respostas. Para compreender melhor os resultados dos testes realizados, o Gráfico 1 faz uma comparação entre os tempos médios de resposta destes. Na legenda do gráfico, T1 representa o teste 1, T2 o teste 2 e T3 o teste 3.

No Gráfico 1, para os testes T1 e T2 conclui-se que, num *cluster* com somente um nó de trabalho disponibilizando o servidor de arquivos, implementado para a DSA (Distribuição do Sistema de Arquivos), o tempo médio de resposta será maior, do que num *cluster* com um SAC (Servidor de Arquivos Centralizado). Isso ocorre, porque na implementação da DSA são usadas várias estruturas para manter e gerenciar determinadas informações, como a *Cache* Local do Cliente e a Estrutura de Diretório, as quais não existem no SAC.

Gráfico 1 Comparando os tempos médios de resposta dos testes



Comparando T1 com T3, observa-se que quando o serviço de arquivos é disponibilizado em dois nós de trabalho, o tempo médio de resposta para o cliente diminui. Para confirmar os resultados do terceiro teste, disponibilizou-se o servidor de arquivos, implementado para a abordagem distribuída, em três nós de trabalho, o servidor de diretório em um único nó de trabalho e os seis processos clientes, executando ao mesmo tempo, em outro nó de trabalho. Observou-se, que o tempo médio de resposta para cada cliente diminuiu ainda mais.

Portanto, pode-se dizer que a distribuição do serviço de arquivos num ambiente de *cluster* ajuda a diminuir o tempo de resposta para os processos clientes, aumentando a disponibilidade do serviço e conseqüentemente o desempenho do sistema.

6 Conclusões

Este artigo apresentou a distribuição do sistema de arquivos para um ambiente de *cluster*. A partir dos estudos realizados começou-se a definir como o sistema de arquivos seria distribuído no ambiente do *cluster*. Em seguida, iniciou-se um trabalho de implementação, onde os produtos finais foram um servidor de arquivos e um servidor de diretório *multithreaded*, além da interface do sistema adaptada para um ambiente distribuído.

Para a distribuição do sistema de arquivos no *cluster*, disponibilizou-se o servidor de arquivo em dois nós de trabalho e o servidor de diretório num único nó de trabalho. Em seguida, realizou-se testes em diferentes ambientes, observando-se que o tempo médio de resposta do sistema diminuía quando o serviço de arquivos era disponibilizado em mais de um nó de trabalho.

Portanto, a distribuição do serviço de arquivos no *cluster* foi implementada, permitindo aos processos clientes acessar os arquivos do espaço de nomes compartilhado de forma transparente, sem conhecer sua localização física; diminuindo a quantidade de requisições enviadas ao processo servidor de diretório, com o uso da

estrutura de *cache* local no cliente; aumentando a disponibilidade do serviço de arquivos e conseqüentemente o desempenho do sistema.

Referências

- BONNIE, Benchmark. (2004) “Benchmark Bonnie”, Revisado em 1996. Disponível em: <<http://www.textuality.com/bonnie/>>. Acesso em: 16 set. 2004.
- COULOURIS, G; DOLLIMORE, J e KINDBERG, T. (2001) Distributed Systems. Concepts and Design, 3 ed., Pearson Education.
- HOCHSTETLER, S e BERINGER, B. (2004) “Linux Clustering with CSM and GPFS”, IBM Brasil. Disponível em: <<http://www.redbooks.ibm.com/redbooks/SG246601.html>>. Acesso em: 15 out. 2004.
- KON, F. (1996) “Distributed File Systems Past, Present and Future”, Technical report. Department of Computer Science - University of Illinois at Urbana-Champaign. Disponível em: <<http://www.ime.usp.br/~kon/>>. Acesso em: 10 out. 2004.
- PLENTZ, P. D. M. (2001) “Um servidor de arquivos para um cluster de computadores”, Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis.
- TANENBAUM, A. S. e Steen, M. van. (2001) Distributed Systems. Principles and Paradigms, Prentice Hall.
- TANENBAUM, A. S. (2003) Sistemas Operacionais Modernos, Prentice Hall.
- VAZ, T. B. (2004) “Clusters Beowulf. Arquitetura de computadores”, Instituto de Matemática, DCC, UFBA. Disponível em: <http://coisa.im.ufba.br/~tiago/arq/apresentacao_beowulf_1.0.pdf>. Acesso em: 23 nov. 2004.