Um mecanismo dinâmico para diminuição da latência e do tempo de resposta em sistemas de arquivos paralelos

Edilson A. Spessoto
Universidade Federal de São Carlos, Departamento de Computação
São Carlos, Brasil, 13560-660
edilson@virgos.com.br

and

Hélio C. Guardia
Universidade Federal de São Carlos, Departamento de Computação
São Carlos, Brasil, 13560-660
helio@dc.ufscar.br

Abstract

This work presents a mechanism developed to minimize the latency and response time on the communication between clients and servers in parallel file systems. This mechanism concerns the communication delays in the disk and network accesses in an attempt to estimate appropriate values for the timeout in read and write operations. Executed at each client application that issues requests to the parallel file servers, the mechanism tries to reduce the overhead on the communication subsystem caused by unnecessary retransmissions. The excessive delay prior to retransmissions is also avoided.

Keywords: Parallel File Systems, Latency Reduction, Response Time

Resumo

Este trabalho apresenta um mecanismo desenvolvido para minimizar a latência e o tempo de resposta na comunicação entre clientes e servidores de sistemas de arquivos paralelos. Este mecanismo aborda as causas dos atrasos na comunicação, notadamente o acesso ao meio de comunicação e aos discos dos servidores. Este mecanismo procura estimar os tempos de *timeout* das requisições de leitura e escrita de clientes para servidores de um sistema de arquivos paralelos distribuídos, evitando sobrecarga no meio de comunicação causada por retransmissões desnecessárias de requisições ou ociosidade causada por demora na detecção de perda de requisições.

Palavras-chave: Sistemas de Arquivos Paralelos, Redução da Latência, Tempo de Resposta

1 Introdução

Os avanços experimentados na microeletrônica têm proporcionado CPUs mais rápidas a cada dia. Este cenário possibilitou o surgimento de aplicações que manipulam volumes de dados da grandeza de muitos gigabytes até alguns petabytes. Por outro lado, os avanços na tecnologia de armazenamento de dados não se deram na mesma velocidade que na microeletrônica, mesmo porque envolvem componentes mecânicos, e, neste caso, é necessário vencer a inércia. Tudo isto fez com que as aplicações que antes eram limitadas pela CPU se tornassem limitadas pela entrada e saída (E/S) de dados, o que é conhecido como o Problema da Entrada/Saída.

Em resposta à necessidade de armazenamento de grandes volumes de dados acessíveis a altas taxas de transferência surgiu a técnica de arquivos paralelos [1]. Esta técnica consiste em particionar os arquivos de dados em blocos (*striping units*) que são armazenados em discos distintos. Estes discos podem estar em um único servidor (RAID) ou em discos servidores diferentes acessíveis via rede em um sistema de arquivos paralelos. Cada conjunto de blocos armazenados em um mesmo disco ou servidor constitui um segmento do arquivo [1]. No caso da distribuição em mais de um servidor, o acesso aos arquivos se dá de forma descentralizada, proporcionando balanceamento de carga através de múltiplos servidores e eliminando a contenção existente no acesso centralizado nos sistemas de arquivos comuns.

Na distribuição de arquivos em servidores diferentes, apesar dos ganhos obtidos no acesso descentralizado aos arquivos, há atrasos proporcionados pela comunicação em rede e no acesso ao disco de cada um dos servidores. Tais atrasos devem-se ao fato de que a comunicação em rede, na maioria dos casos, ocorre através de recursos compartilhados. Em um sistema computacional há diversas aplicações competindo pelo meio de transmissão, incluindo os próprios servidores de arquivos paralelos, aumentando a latência no envio de requisições e no tempo de resposta.

Além dos atrasos no meio de comunicação, há atrasos no acesso aos discos dos servidores de arquivos paralelos. Estes atrasos são causados pela operação normal do disco como posicionamento do braço e das cabeças de leitura [6] e ainda pelo recebimento de eventuais requisições simultâneas de clientes distintos. Para minimizar tais atrasos foi desenvolvido um mecanismo de ajuste dinâmico para diminuição da latência e do tempo de resposta nestes sistemas de arquivos.

Em sistemas de arquivos paralelos existem clientes e servidores. Os clientes têm a função de enviar requisições de escrita e leitura de dados aos servidores, que são responsáveis pela escrita em seus discos dos dados recebidos do cliente e pela leitura dos dados requisitados por eles. Após a realização da operação solicitada, os servidores devem informar aos clientes apropriados os dados solicitados ou então a confirmação de escrita dos dados recebidos.

O mecanismo proposto neste trabalho é localizado nos clientes e calcula e ajusta dinamicamente os tempos de *timeout* das requisições para cada um dos servidores acessados pelos clientes.

Na seção 2 são apresentados os assuntos relevantes ao trabalho. Na seção 3 é descrito o mecanismo de ajuste dinâmico para a diminuição da latência e do tempo de resposta desenvolvido. Na seção 4 são apresentados os trabalhos atuais e os resultados esperados. Na seção 5 são descritos os trabalhos futuros.

2 Arquivos Paralelos em Rede

Na implementação de um sistema de arquivos paralelos em rede, o modelo de comunicação cliente x servidor é utilizado para permitir que aplicações armazenem e recuperem dados sobre os discos presentes nas estações da rede. Vistos como estruturas lógicas únicas, arquivos paralelos nessa arquitetura têm seus dados divididos de acordo com algum padrão de distribuição determinado pelo sistema de arquivos. Partições dos dados de um arquivo paralelo atribuído a cada servidor são armazenados de maneira contígua num disco local, na forma de um *segmento*. O mapeamento dos dados de um arquivo lógico sobre os segmentos nos servidores pode ser determinado no cliente ou nos próprios servidores [1].

Assim como em outros sistemas de arquivos paralelos em rede, no sistema de arquivos NPFS - *Network Parallel File System* [2], a mesma arquitetura cliente x servidor pode ser encontrada. Este sistema de arquivos é composto por um processo mestre, N processos servidores e M processos clientes. Os processos clientes, que são bibliotecas incluídas nas aplicações, solicitam a abertura e o fechamento de arquivos ao mestre. Este, por sua vez, é responsável pela inicialização dos servidores, pela determinação de quais servidores armazenam os arquivos requisitados pelos clientes e pela requisição de abertura e fechamento de arquivos aos servidores. Uma vez atendida esta requisição, o mestre retorna uma resposta aos clientes, que a partir de então passam a enviar requisições de leitura e escrita diretamente aos servidores apropriados.

Para evitar o *overhead* causado pelo controle existente em protocolos orientados à conexão e, tendo em vista a demanda por altas taxas de transferência, as comunicações entre processos do NPFS utilizam um protocolo não-orientado à conexão, o UDP. Por não garantir a entrega de pacotes de dados, uma requisição pode falhar na ocorrência de perda de pacotes devido a falhas na rede, no envio de uma requisição ou de sua resposta. Também pode ocorrer o descarte de pacotes pelos servidores caso o buffer de recepção esteja cheio. Estes são os motivos mais comuns de falhas. Neste caso, faz-se necessária a detecção da perda de pacotes e a retransmissão dos pacotes perdidos.

De maneira semelhante à utilizada no protocolo TCP [3], para detectar a perda de pacotes de dados, no momento do envio o cliente inicia um temporizador que contém um valor inicial (valor de *timeout*) representando o tempo que se deve aguardar até o recebimento da confirmação do servidor. Caso o temporizador atinja o tempo máximo estabelecido e a resposta aguardada não tenha chegado, considera-se que a requisição foi perdida e procede-se o seu reenvio, iniciando-se o temporizador novamente. Esta operação é repetida até que se obtenha sucesso ou até que um número máximo de tentativas seja atingido, causando o cancelamento das transmissões e um erro de operação.

A determinação de valores apropriados para o tempo de espera de confirmações das requisições é um fator crítico no desenvolvimento de um sistema de arquivos paralelos em rede. Com a utilização de valor muito pequeno podem ocorrer retransmissões desnecessárias, gerando sobrecargas no meio de transmissão e no acesso aos discos dos servidores. Por outro lado, quando o tempo de espera é muito grande, o atraso para identificar uma falha de comunicação por qualquer motivo pode deixar ociosos os discos nos servidores e a própria rede de comunicação.

Num sistema de arquivos paralelos em rede NPFS, discos e rede são utilizados no processamento das requisições. Nas operações de leitura, uma lista de blocos necessários é propagada pela rede para os servidores, que devem recuperar os dados de seus discos locais e transmiti-los ao cliente. Na escrita, fragmentos de dados são enviados pela rede para os servidores apropriados, que devem gravá-los em seus discos locais, notificando o sucesso da operação aos clientes posteriormente. Determinar os atrasos envolvidos em cada uma dessas operações constitui um problema a ser equacionado para o bom desempenho de um sistema de arquivos paralelos em rede.

O tempo de operação da rede nas comunicações entre clientes e servidores pode ser determinado em função do tempo gasto entre a emissão de uma requisição e o retorno de uma resposta. Uma abordagem para a determinação do atraso na rede consiste em iniciar o valor do tempo de espera (*timeout*) com um valor pré-definido e então ajustá-lo no decorrer das operações de comunicação, de acordo com os tempos de resposta apurados nas transmissões bem-sucedidas. Para se atualizar o *timeout* deve-se utilizar o tempo transcorrido enre ida de uma mensagem ao servidor e o retorno da confirmação de recebimento. Este tempo é conhecido como *Round-Trip Time* (RTT) [5]. O cálculo do RTT é realizado subtraindo-se do tempo de recebimento de uma confirmação o tempo de envio da requisição correspondente. Como a carga de mensagens no meio de comunicação varia com o tempo, o tempo necessário para o tráfego de requisições também varia, por isso o valor de RTT é útil, uma vez que reflete as variações do meio de comunicação.

Para a obtenção do *timeout*, cada vez que se obtém um novo RTT é possível ajustar-se o RTT estimado para a transferência de dados, segundo o algoritmo de Karn [6]:

$$RTT = (\alpha * RTT\_anterior) + ((1 - \alpha) * RTT\_novo)$$

Onde $0 <= \alpha < 1$ (Recomendado $\alpha = 0,9$)

timeout = β * RTT ($\beta > 1$, se recomenda $\beta = 2$)

Os valores recomendados neste caso referem-se à implementação original do TCP.

Neste algoritmo, o parâmetro α determina o papel desempenhado pelas novas medidas apuradas de RTT no cálculo do *timeout*, isto é, determina o quanto os novos valores de RTT irão influenciar no novo RTT. O valor recomendado (0,9) implica que os novos valores de RTT serão baseados nos valores antigos em 90% e em 10% no último valor medido. A razão para a recomendação deste valor deve-se ao fato de que variações bruscas na carga da rede devem causar pouca influência nos valores de RTT, visto que podem ser passageiras e, caso se mantenham, serão incorporadas paulatinamente no valor de RTT. Já o parâmetro β ser maior que 1 significa que o *timeout* deve ser maior que o tempo estimado para a ida e volta de uma mensagem. Valores próximos de 1 tornam o algoritmo mais sensível a possíveis perdas, com o inconveniente de poder causar instabilidade.

3 Abordagem para o desenvolvimento do mecanismo

Visando prover confirmação de recebimento de pacotes de dados aos clientes de sistemas de arquivos paralelos sem causar *overhead* na comunicação encontrado em alguns protocolos, deve-se detectar a perda de pacotes e retransmití-los eficientemente. Isto é conseguido enviando-se as requisições com os dados de clientes para servidores e aguardando-se pela confirmação. Caso esta não ocorra, é necessário retransmitir as requisições não confirmadas até um certo período de tempo após o envio (*timeout*) [3].

Para se ajustar os tempos de espera até que uma requisição seja considerada perdida por falta de resposta, é necessário que se conheça o tempo necessário para que uma mensagem chegue até o servidor, seja processada através do acesso ao disco e então seja transmitida ao cliente a confirmação de recebimento. Contudo, o tempo necessário para o tráfego das requisições pela rede e para acesso ao disco pode variar em função do tempo, do tipo de requisição (leitura ou escrita) e do tamanho da requisição na comunicação com cada um dos servidores.

3.1 Tempo de acesso aos discos dos servidores

Os tempos de acesso aos discos nos servidores são apurados tomando como base as requisições bem-sucedidas dos clientes. No recebimento de uma requisição, o servidor guarda seu o tempo de chegada e procede ao atendimento. Ao término do atendimento de cada requisição, o servidor calcula o tempo gasto através da diferença entre o tempo do término e o tempo de chegada e insere o valor na resposta ao cliente.

No recebimento de confirmações de atendimento das requisições aos servidores, os clientes mantêm em uma tabela as informações a respeito do tempo gasto no atendimento de cada servidor, isto é, do tempo gasto no acesso ao disco. Além disso, é calculado o RTT segundo o algoritmo descrito na seção 2. Estas informações são mantidas e atualizadas em uma tabela como a Tabela 1 a cada comunicação com os servidores, para posterior consulta e utilização no uso de ajuste do *timeout* para a comunicação com os servidores.

3.2. Tempo da comunicação com os servidores

Para o ajuste do *timeout* das requisições enviadas, o cliente deve possuir estimativas de tempo para o atendimento de requisições. Este tempo é composto pelo tempo gasto no tráfego da requisição no meio de comunicação (ida e volta) e pelo tempo gasto no atendimento da requisição, isto é, no acesso ao disco. Chamando a soma dos tempos de *Round-Trip Time* (RTT), tem-se

$$RTT = T_{recebimento} - T_{envio} - T_{disco}$$

Cabe notar que os tempos de envio e recebimento são relativos ao cliente.

Considerando que o RTT varia em função do tempo devido à variação da carga de trabalho no meio de comunicação, o valor do *timeout* deve ser reajustado dinamicamente a cada interação com os servidores. O cálculo do novo valor de *timeout* deve levar em consideração o valor atual e o último valor de RTT apurado.

Serv ID	Op. (E/S)	Tam. Req. (Bytes)	Tempo Disco(ms)	RTT(ms)
1	Entrada	450000	5	3
1	Saída	...	...	...
...	...	...	...	...
N	...	...	...	...

Tabela 1 – Exemplo da tabela contendo Valores de Comunicação com Servidores

O valor de RTT apurado na comunicação com cada um dos servidores é armazenado em uma tabela como a Tabela 1. Nela, cada entrada apresenta a identificação do servidor (Serv. ID), o tipo de operação realizado (Entrada/Saída), o tamanho da requisição em *Bytes* ao qual referem-se os tempos (Tam. Req.), o tempo gasto (em milisegundos) no acesso ao disco (Tempo Disco) e o tempo gasto (em milissegundos) desde o envio da requisição até o recebimento de confirmação correspondente (RTT).

3.3. Cálculo de *timeout* das requisições

O *timeout* para requisições enviadas aos servidores é calculado consultando-se a entrada da tabela referente ao servidor apropriado considerando o tipo de operação e o tamanho da requisição para se obter o tempo estimado de acesso ao disco e à rede, desta forma:

$$timeout = (T\_Disco + T\_Rede) * P$$

onde

T_Disco = Tabela[Serv. ID][Op.][T. Disco]

T_Rede = Tabela[servidor id][Op.][T. Rede]

P = Tabela[Serv.ID][operação][Tam. Req] / Tamanho_Requisição_Atual

O cálculo do *timeout* leva em consideração os tempos de RTT (tempo gasto na comunicação) e de acesso ao disco dos servidores. Além disso, o parâmetro P serve para ajuste do tempo em função do tamanho da requisição a ser realizada (Tamanho_Requisição_Atual) em relação ao tamanho da requisição da qual foram tomados os valores (valor P).

4 Resultados esperados

Com este mecanismo espera-se diminuir a latência na comunicação entre clientes e servidores do sistema de arquivos paralelos NPFS através da detecção de perda de requisições e retransmissão no intervalo de tempo apropriado.

Para averiguar se há ganhos de desempenho em relação à arquitetura NPFS anterior está-se procedendo avaliações de desempenho comparando-se as taxas de transferência utilizando-se o NPFS com o mecanismo e sem ele para verificação e interpretação dos resultados.

Em testes realizados para apurar o comportamento do meio de comunicação, transmitiu-se dados no tamanho de 1GB utilizando-se o protocolo UDP, variando-se o tamanho das requisições, conforme demonstrado na Fig. 1. Os

resultados sugerem que, no envio de requisições pelo cliente, determinando-se corretamente o tempo de *timeout* considerando a rede e o disco, será possível conseguir uniformidade nas taxas de transferência através da utilização eficiente dos recursos de armazenamento e transmissão de dados.

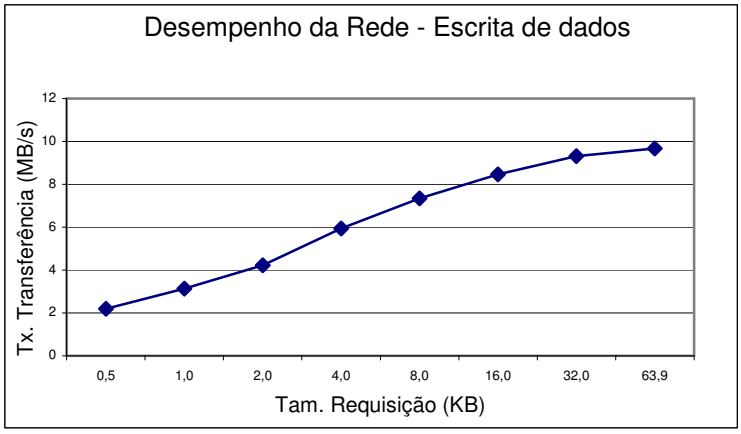

Fig. 1 – Desempenho da rede

Considerando que, em um sistema de arquivos paralelos, cada um dos segmentos dos arquivos armazenados é acessado localmente pelos servidores, como arquivos seqüenciais, foram realizados testes para se avaliar o desempenho do disco, realizando-se requisições de leitura e escrita seqüenciais, de diferentes tamanhos de dados, em sistemas de arquivos locais. Para verificar a interferência de processos concorrentes realizando acessos ao disco, foram realizados testes com um único processo acessando o disco e com dois processos acessando o disco simultaneamente.

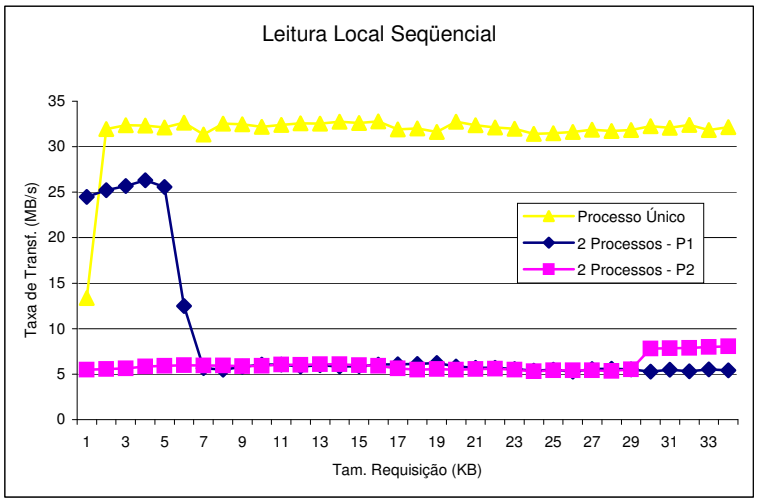

Fig. 2 – Desempenho no acesso ao disco

Conforme demonstrado na Fig. 2, os resultados obtidos na utilização de um único processo realizando requisições de leitura seqüencial demonstram uma taxa de transferência baixa para requisições de 0,5KB devido às leituras de blocos de dados completos do disco rígido (4 KB), dos quais é aproveitado (contabilizados no cálculo da taxa) apenas 0,5KB. Para tamanhos de requisição maiores de 0,5KB, houve uma certa constância na taxa de transferência.

Na Fig. 2, nota-se a perda de desempenho na ocorrência de mais de um processo realizando acessos ao disco simultaneamente, justificando a necessidade de uma política de tomada de valores durante a utilização do sistema, uma vez que a carga de trabalho varia com o tempo. A variação na taxa de transferência obtida na utilização de dois processos sugere o tratamento diferenciado de requisições segundo a três faixas de tamanhos: requisições menores que 7KB, maiores ou iguais a 7KB e menores que 30KB e requisições maiores ou iguais a 30KB. Desta forma, pode

haver necessidade de as entradas da tabela contendo Valores de Comunicação com Servidores deverem contemplar também as faixas de tamanho onde as requisições se enquadram.

5 Resultados esperados e trabalhos futuros

Com a minimização dos acessos desnecessários à rede em função do ajuste apropriado dos valores de *timeout*, o desempenho do sistema de arquivos paralelos deve melhorar significativamente. A diminuição das oscilações de desempenho para diferentes tamanhos de requisição também deve ser observada. Tais ajustes devem tornar o sistema mais estável e menos vulnerável a interferências no meio de transmissão e na concorrência pelo acesso aos discos nos servidores.

De acordo com os resultados dos testes realizados, caso mostrem ganhos de desempenho medidos pelo aumento da taxa de transferência de dados entre clientes e servidores NPFS, será realizado um estudo para a melhora do mecanismo de *timeout*, baseando-se em algoritmos mais recentes empregados em implementações de protocolos de comunicação que utilizam ajustes de *timeout* .

Esse estudo deverá avaliar o impacto causado pela inserção de complexidade no cálculo dos valores, uma vez que a complexidade existente em protocolos orientados a conexão os torna inapropriados para a transferência dos grandes volumes de dados requeridos pelas aplicações atualmente.

Em relação à tabela mantida pelos clientes, serão realizados estudos para uma melhor utilização dos valores referentes ao tempo de acesso ao disco, contemplando diferentes tamanhos de requisições de leitura e de escrita, para que se possa estimar com maior precisão o valor de *timeout*, baseando-se no tipo e tamanho das requisições enviadas.

6 Referências

[1] Crockett, T. W. File Concepts for Parallel I/O. *Proc. Supercomputing 89*, IEEE CS Press, Los Alamitos, Calif., Order No. 2021, 1989, pp. 574-579.

[2] Guardia, H.C. Considerações Sobre as Estratégias de um Sistema de Arquivos Paralelos Integrado ao Processamento Distribuído, *Tese de Doutorado*, Escola Politécnica da Universidade de São Paulo, São Paulo, 1999.

[3] Jacobson, V. and Karels, M.J. Congestion Avoidance and Control, Proc. of *The Sigcomm'88* Symposium in Stanford, CA, August, 1988

[4] Jacobson, V., Braden R. and D. Borman. TCP extensions for high performance. RFC 1323, May 1993.

[5] Karn, P. and Partridge, C. Improving round-trip time estimates in reliable transport protocols. *ACM Transactions on Computer Systems (TOCS)*, vol. 9, pp. 365–373, 1991.

[6] Ruemmler, C. and Wilkes, J. An Introduction do Disk Drive Modeling. *IEEE Computer*, March 1994.