Sincronização Parcial de Servidores de Dados no Sistema de Arquivos dNFSp

Danilo Fukuda Conrad, Rodrigo Virote Kassick, Philippe O. A. Navaux
Instituto de Informática - Universidade Federal do Rio Grande do Sul
{dfconrad, rvkassick, navaux}@inf.ufrgs.br

Abstract

This article proposes a new synchronization method for data servers in the dNFSp file system. This new mechanism will allow for some servers to store just a subset of the data that is usually destined for their storage group. The new method will also allow the data servers to keep their data coherent even though one of them gets full. As a result of that it will be possible to support data servers with different storage capacities in the file system.

1. Introdução

O uso de *clusters* (agregados) de computadores tem sido uma alternativa eficaz e de baixo custo para computação de alto desempenho. Contudo, as aplicações executadas nessas arquiteturas podem lidar com uma quantidade muito grande de dados, sendo a velocidade ao acesso deles um fator importante. O uso de um sistema de arquivos que tire proveito da arquitetura paralela pode aumentar consideravelmente o desempenho no acesso aos dados.

O dNFSp[1] é um exemplo de sistema de arquivos distribuído e paralelo para *clusters* compatível com o NFS padrão[4]. Ele é composto por servidores de metadados (metasservidores) e servidores de dados (I/O daemons, IODs). Nos servidores de metadados são armazenadas as informações sobre os arquivos tais como atributos e a sua localização nos IODs. É possível haver mais de um metasservidor a fim de distribuir os clientes entre eles. Ao tratarem as requisições em paralelo aumenta-se o desempenho nas operações de escrita. Os IODs armazenam os blocos de dados, enviando-os diretamente aos clientes em paralelo, o que aumenta consideravelmente o desempenho de leitura[3]. Os IODs podem ser reunidos em grupos de armazenamentos (Virtual IODs, VIODs). IODs pertencentes ao mesmo grupo servem como réplicas para aumentar a tolerância a falhas do sistema e os diversos grupos servem para determinar o fracionamento dos arquivos.

Assim como a maioria dos sistemas de arquivos distribuídos, o dNFSp foi projetado para trabalhar em *clusters* de máquinas homogêneas, ou seja, com a mesma configuração. Entretanto, isso acaba se tornando uma restrição, pois diversos *clusters* são compostos por máquinas com configurações diferentes. Neste artigo consideraremos apenas a heterogeneidade em relação ao tamanho dos discos, ou seja, serão considerados ambientes heterogêneos aqueles com IODs com diferentes capacidade de armazenamento. Ao executar o sistema num ambiente heterogêneo pode ocorrer um problema nas operações de escrita, pois os dados são distribuidos entre os VIODs através de uma política *round-robin*. Quando um dos IODs de um VIOD estiver cheio, ocorrerá um erro, e o sistema deixará de funcionar. Isso limita a capacidade de armazenamento do sistema aos IODs com menor espaço. A fim de solucionar esse problema, este artigo propõe novos estados de sincronização dos dados, e os fatores que devem ser considerados para implementá-los.

A seção seguinte apresenta os novos estados de sincronização propostos, e o que é preciso considerar a fim de implementá-los. Em seguida, serão apresentados os possíveis casos de escrita utilizando os estados de sincronização parcial, e como eles serão tratados. Por fim, serão apresentadas as conclusões e trabalhos futuros.

2. Estados de Sincronização Parcial

Para implementar os novos estados, dois atributos serão adicionados aos IODs: *full* (cheio) e *append* (anexo). Quando um IOD estiver sem espaço, ele será marcado como *full*, e poderá ser associado a outro IOD, pertencente a um grupo de armazenamento (VIOD) diferente, que será marcado como *append*.

A Figura 1 ilustra essa situação. Durante a escrita do bloco de arquivo foo.bar1 no IOD 2, ele fica sem espaço, e é marcado como *full*. Então, ele é associado ao IOD 4 que é marcado como *append* e recebe os blocos restantes do arquivo foo.bar1. O IOD 4 continua recebendo as requisições de escrita enviadas ao VIOD 2 normalmente, mas também pode receber as requisições do VIOD 1.

O IOD anexo funcionará como uma extensão lógica do IOD cheio, recebendo os dados gravados subsequente-

mente. Essa associação pode se estender além, caso o IOD *append* fique cheio. Neste caso ele receberia também o atributo de *full*, e seria associado a outro IOD pertencente a um VIOD diferente.

Ao associar um IOD a mais de um VIOD ele poderá receber requisições de escrita de dois grupos, o que poderá afetar o seu desempenho. Entretanto, essa associação tem como objetivo principal manter o sistema operante, e os possíveis efeitos no desempenho serão estudados posteriormente.

Quando dados forem deletados do servidor *full*, em princípio ele continuará com status de cheio, e as novas escritas continuarão sendo passadas para o IOD *append*. Entretanto, ele poderá voltar ao seu estado normal. Haverá um limiar para um servidor *full* voltar ao estado normal, baseado em quanto de espaço há livre no IOD *append* e quanto há livre nos outros IODs do VIOD. Entretanto, como o sistema de arquivos suporta rebalanceamento de IODs [2], esse limiar é difícil de ser determinado.

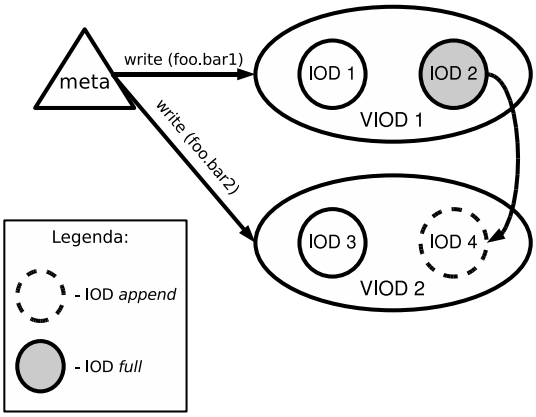

Figura 1. Associação entre IODs *full* e *append*

3. Divisão dos Dados Entre IODs *full* e IODs *append*

Ao tratarmos os IODs associados como um único disco lógico será preciso identificar onde serão escritos os blocos de dados. Existem duas situações principais a serem diferenciadas: operações de escrita e de sobrescrita.

As operações de escrita (contendo dados novos) serão sempre enviadas ao IOD *append* após a associação. Nas operações de sobrescrita, será necessário verificar onde estão os blocos de dados originais. Se estiverem no IOD *full*, os dados antigos serão sobrescritos. Contudo, caso a quantidade de dados seja maior que a quantidade de blocos anterior, será preciso enviar o restante para o IOD anexo, e diferenciar esse tipo de operação. Caso os dados sobrescritos estejam no IOD *append*, eles serão sobrescritos normalmente.

Exemplificando com a Figura 1, se um dado fosse sobrescrito no IOD 2, seria preciso verificar se a quantidade de blocos é maior que a original. Caso haja um número maior de blocos que o anterior, será preciso enviar os restantes ao IOD 4. Caso contrário, os blocos poderão ser escritos no próprio IOD 2.

4. Conclusões e Trabalhos Futuros

Com a criação dos novos atributos para os IODs, foi possível definir diferentes estados de sincronização entre os servidores de dados. Utilizando os novos estados, espera-se obter uma maior disponibilidade no sistema de arquivos, permitindo que o sistema continue operante mesmo após um servidor de dados ficar sem espaço livre.

Foram apresentados casos específicos gerados pelos novos estados, nos quais os dados podem ser gravados tanto no servidor *append*, quanto no servidor *full*. A distribuição dos dados nesses servidores necessitará de um tratamento especial.

Foi ressaltado também que o desempenho poderá ser afetado devido a associação de um IOD a mais de um VIOD. Entretanto, o foco da solução será inicialmente na disponibilidade e não no desempenho. Os impactos da implementação serão estudados após ela ter sido realizada.

Dentre os trabalhos futuros estão implementar os novos estados, realizando o tratamento dos casos específicos anteriormente mencionados. Em seguida, pretende-se testar o sistema em um ambiente heterogêneo, verificando o impacto no desempenho do sistema. Através dos testes espera-se encontrar valores aproximados para determinar o limiar no qual um IOD *full* pode retornar ao estado normal.

Referências

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

[2] E. Hermann. Dinamismo de servidores de dados no sistema de arquivos dnfsp. Master's thesis, PPGC-UFRGS, Porto Alegre, 2006.

[3] R. Kassick, C. Machado, E. Hermann, R. Ávila, P. Navaux, and Y. Denneulin. Evaluating the performance of the dNFSP file system. In *Proc. of the 5th IEEE International Symposium on Cluster Computing and the Grid, CCGrid*, Cardiff, UK, 2005. Los Alamitos, IEEE Computer Society Press. CD-ROM Proceedings, ISBN 0-7803-9075-X.

[4] S. Microsystems. Nfs: Network file system protocol specification, 1989.