Resumo: Using Virtual Filesystem to avoid metadata bottlenecks

Gabriel Dieterich Cavalcante — RA 079738 — mo401 - 2010

E. Artiaga and T. Cortes . *Using Filesystem Virtualization to Avoid Metadata Bottlenecks. Virtualization technologies.* DATE, Março-2010, p. 562. Dresden, Alemanha.

Sistemas de *cluster* dependem do compartilhamento de recursos entre vários nós, onde, aplicações paralelas podem dividir recursos espalhados ao longo do do *cluster*. Os sistemas de arquivos distribuídos apresentaram um modelo de armazenamento compatível com esta realidade, provendo mecanismos de distribuição de dados em vários servidores de armazenamento, que fazem com que as informações fiquem disponíveis de modo que nós do *cluster* enxerguem-nas como dados locais. Entretanto, o trabalho de Artiaga & Cortes identifica alguns casos onde existe degradação de *performance* no sistema, principalmente pela herança de comportamento dos sistemas de arquivos clássicos.

Nos sistemas de arquivos clássicos os diretórios são criados para indicar algum tipo de afinidade entre os arquivos, tendendo a compactar metadados desses arquivos em um só local. Os nós de *clusters* que rodam pequenas aplicações rotineiramente armazenam arquivos de controle/*checkpoint* em um mesmo diretório compartilhado, o que acarreta a criação muitos arquivos em paralelo dentro do mesmo. Estudos quantitativos observaram queda de rendimento principalmente pelo sistema de arquivos tentar manter um pequeno e atulizado controle sobre os metadados deste diretório compartilhado. O agravante principal está na punição empregada a todos os nós do *cluster*, e não somente a aqueles que estão "infringindo" as boas práticas de acesso.

Este estudo propôs COFS—*The Composite File System*, uma camada virtual entre os nós do *cluster* e o servidor de armazenamento distribuído GPFS—*General Parallel File System*—da IBM. Esta camada visa o desacoplamento do gerenciamento de arquivos e de metadados, e, além disso, cria diferentes visões de organização dos arquivos e de usuário. Para isto, divide grande diretórios compartilhados em sub-diretórios no nível de organização, porém abstrai esta visão no nível de usuário, mostrando apenas um diretório. Desta forma as aplicações conseguem seguir seu padrão de criação de arquivos—muitos em um mesmo diretório. Transparentemente o sistema armazena metadados em uma organização—um nó adicional—que evita o excesso sincronização criado, pois não há *metadata request* em vários nós do *cluster*.

Um protótipo desenvolvido adicionou um nó, responsável por armazenar informações sobre os metadados, além de uma camada extra em cada nó, que monta a "visão do usuário" do sistema de arquivos. No momento de criação do arquivos é gerado um *hash* das seguintes informações: o nó que requisitou a criação, o diretório pai na "visão do usuário" e o processo que está criando o arquivo. Após este cálculo será decido onde o arquivo ficará armazenado no sistema de arquivos distribuído. Isso faz com que os arquivos sejam organizados em diretórios diferentes, ou seja, evita conflitos entre nós. Porém os arquivos continuam completamente próximos na visão do usuário.

Testes realizados demonstraram que o sistema de Artiaga & Cortes otimizou de 5 à 10 vezes o processo de criação de arquivos em diretórios compartilhados por nós do *cluster*, para outras operações—*open/close, stat*—o fator de *speedup* foi menor, porém ainda considerável. Alterar as estrutura geral dos arquivos poderia causar impactos negativos nas operações de leitura/escrita, pois eventualmente a localização dos dados poderia ser alterada pela camada de virtualização, o que causaria sobrecarga da banda de rede. Testes foram realizados e comprovaram que COFS teve consumo de rede similar ao GPFS nativo.

O estudo de Artiaga & Cortes mostrou que técnicas de virtualização são uma ferramenta válida para desacoplar o controle de nomes e metadados do gerenciamento de dados em baixo nível. Para isto uma prova de conceito implementada otimizou o funcionamento de um consagrado sistema de arquivos, o GPFS da IBM. O destaque principal vai para a melhoria de *performance* quando o sistema lida com ambientes compartilhados contendo grande número de arquivos, o que é muito comum em *clusters* que possuim vários nós rodandos pequenas aplicações independentes ou paralelas.