Documentos

ISSN 1677-9274
Agosto, 2017
151

Sistemas de arquivos distribuídos em apoio ao processamento científico

Embrapa

ISSN 1677-9274
Agosto, 2017

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento

Documentos 151

Sistemas de arquivos distribuídos em apoio ao processamento científico

Leandro Carrijo Cintra

Embrapa Informática Agropecuária
Campinas, SP
2017

Embrapa Informática Agropecuária
Av. Dr. André Tosello, 209 - Cidade Universitária, Campinas - SP
Fone: (19) 3211-5700
https://www.embrapa.br/informatica-agropecuaria

Supervisão editorial: Kleber X. Sampaio de Souza
Revisão de texto: Adriana Farah Gonzales
Normalização bibliográfica: Maria Goretti G. Praxedes
Editoração eletrônica: Tuíra Santana Favarin, sob supervisão de Flávia Bussaglia Fiorini.
Imagem da capa: Tuíra Santana Favarin

1ª edição publicação digital - 2017

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Informática Agropecuária

Carrijo, Leandro Cintra.
Sistemas de arquivos distribuídos em apoio ao processamento científico. - Campinas : Embrapa Informática Agropecuária, 2017.
47 p. il.: color. ; 16 cm x 21 cm. - (Documentos / Embrapa Informática Agropecuária, ISSN 1677-9274; 151).

1.Sistemas distribuídos. 2. GlusterFS. 3. Software. I. Título II. Embrapa Informática Agropecuária. III. Série.

CDD 363.728

Autor

Leandro Carrijo Cintra
Cientista da Computação, Doutor em Bioinformática e Analista da Embrapa Informática Agropecuária, Campinas, SP.

Apresentação

Os avanços recentes no desenvolvimento de tecnologias para a obtenção de dados na área de biologia molecular tornaram possível a execução de projetos de pesquisa que se baseiam na geração, no armazenamento e na análise de grandes volumes de informações. Do ponto de vista computacional, isso se torna um grande desafio a ser superado, dado que a demanda por armazenamento e processamento vai além das necessidades tradicionais; exigindo-se assim, novas abordagens, tanto relacionadas com o armazenamento quanto com o processamento dos dados.

No caso do armazenamento, os dados científicos têm características interessantes para o uso de Sistemas de Arquivos Distribuídos. Dentre tais características, pode-se mencionar o fato de se gerar grande quantidade de dados temporários no decorrer da execução de análises e, também, o acesso paralelo às informações que ocorre, frequentemente, no ambiente computacional.

Os Sistemas de Armazenamento Distribuídos têm a vantagem de operar eficazmente quando a demanda de acesso concorrente aumenta; podem fornecer um considerável espaço de armazenamento e apresentam um custo acessível quando comparado com os Sistemas de Armazenamento Dedicados (*storages*).

O presente trabalho apresenta o Sistema de Arquivo Distribuído GlusterFS discutindo suas principais características, formas de instalação e possíveis configurações em ambiente de produção. Os resultados descritos neste documento foram obtidos no âmbito do projeto "Tecnologias para computação distribuída, armazenamento de grandes volumes de dados e *workflow* científico, em suporte à pesquisa agropecuária" e são aplicáveis a instituições científicas que demandem o armazenamento e processamento de grandes volumes de dados.

Sílvia Maria Fonseca Silveira Massruhá
Chefe da Embrapa Informática Agropecuária

Sumário

Sistemas de arquivos distribuídos em apoio ao processamento científico

Leandro Carrijo Cintra

Introdução

A ciência caminha rapidamente para uma condição na qual a computação deixa de ser apenas uma ferramenta auxiliadora e passa a fazer efetivamente parte do método científico. Jim Gray, cientista da computação e pesquisador da Microsoft, devotou sua carreira profissional à integração da Tecnologia da Informação (TI) às ciências. A ele se atribui a afirmação de que a ciência está vivendo a fase do quarto paradigma, com o que ele indicava que atualmente é possível fazer ciência explorando grandes volumes de dados e/ou executando simulações complexas. Essa metodologia tem se difundido também com o termo e-Science. Historicamente, há milhares de anos houve a fase empírica, na qual fazer ciência resumia-se basicamente em coletar e descrever fenômenos naturais; há alguns séculos iniciou-se a fase teórica na qual se criaram modelos e generalizações; há algumas décadas introduziram-se as simulações computacionais e, atualmente, pratica-se a exploração de dados (HEY, 2011).

Tradicionalmente, as aplicações computacionais científicas são altamente demandantes de processamento; com a e-Science, essas aplicações têm se tornado também altamente dependentes de capacidade para armazenamento de grandes volumes de dados. Apenas

como um exemplo, o Centro de Supercomputação de San Diego (San Diego Supercomputer Center - SDSC) mantém armazenados em seu data center 27 PB de dados em aproximadamente 100 bases de dados em temas diversos, tais como bioinformática e recursos hídricos.

Quando o obstáculo a ser superado é o espaço para o armazenamento de dados, então invariavelmente as soluções disponíveis utilizam o armazenamento distribuído. E apesar dos fabricantes aumentarem constantemente a capacidade de armazenamento dos *storages* modernos, isso ocorre a um ritmo muito inferior à demanda. Sendo assim, surge como opção a utilização de vários dispositivos de armazenamento, simultaneamente, para se obter um ambiente com alta capacidade. Essa foi a abordagem adotada por várias organizações comerciais e científicas, tais como Google, Facebook, Yahoo, SDSC, Instituto de Física Corpuscular (IFC), National Center for Biotechnology Information (NCBI), European Bioinformatics Institute (EBI) dentre outros, com grande sucesso.

Evidentemente, existem disponíveis várias tecnologias que implementam e possibilitam o armazenamento distribuído; mas como não poderia deixar de ser, o componente central dessas tecnologias é sempre um Sistema de Arquivos Distribuído (Distributed File System - DFS). Dentre os principais DFS disponíveis atualmente, pode-se citar o LustreFS, Hadoop File System (HDFS) o CEPH e o GlusterFS. Neste trabalho, apresenta-se uma comparação das principais características desses diversos sistemas e as razões que motivaram a escolha do GlusterFS para uso no ambiente de produção do Laboratório Multiusuário de Bioinformática (LMB) da Empresa Brasileira de Pesquisa Agropecuária (Embrapa). Além disso, tem-se uma discussão detalhada dos procedimentos de sua instalação e informações sobre testes de desempenho realizados.

Certamente, existem diferenças significativas entre sistemas de armazenamento distribuído, que utilizam um conjunto de servidores commodities (não específicos para armazenamento); e os *storages*, que são desenvolvidos unicamente com a finalidade de serem

armazenadores de dados. Evidentemente estes últimos são muito mais seguros e apresentam uma performance otimizada. Por outro lado, têm um custo muito mais elevado e dificuldades para escalar em ambientes que exigem grandes volumes de armazenamento. Dessa forma, os ambientes de armazenamento distribuídos não são adequados para o armazenamento de todos os tipos de dados, antes, são projetos para aquelas categorias de dados que demandam menos segurança e têm um volume que cresce consideravelmente. Esse é o caso de várias iniciativas em áreas científicas nos últimos tempos. Mais especificamente, os projetos científicos tendem a gerar vários conjuntos de dados intermediários nas análises que levam dos dados primários aos conjuntos de resultados. Esses dados intermediários são excelentes candidatos a residirem em um DFS.

Neste sentido, defende-se aqui que em um ambiente de processamento científico, o qual demande alta capacidade de armazenamento; uma estrutura mista, que envolva o uso de *storages* para os dados primários e dados de resultados, e armazenamento distribuído para dados intermediários, teria um bom custo-benefício.

Além desta introdução, o presente documento constitui-se da seção 2, que discutirá os principais sistemas de arquivos distribuídos e apresentará uma argumentação para a seleção do GlusterFS para uso no ambiente de produção do LMB; da seção 3, que apresentará detalhes sobre a instalação do GlusterFS; da seção 4, que discorrerá sobre a arquitetura adotada no presente trabalho; da seção 5, que apresenta uma análise de desempenho do sistema; da seção 6, que apresenta uma análise de custos da solução; e da seção 7 que traz uma conclusão dos trabalhos.

Sistemas de Arquivos Distribuídos

Os sistemas de arquivos são um componente importante em sistemas computacionais. Eles se referem à camada de software do sistema operacional responsável pelo gerenciamento e controle dos dados armazenados em memória secundária. Tradicionalmente,

a implementação ocorria de forma local, ou seja, cada máquina mantinha apenas os sistemas de arquivos para as mídias sob sua responsabilidade. Sistemas de arquivos desses tipos são amplamente discutidos em (TANENBAUM, 2010) .

No entanto, com a ampliação do uso de redes de computadores, tornou-se relevante o uso de sistemas de arquivos compartilhados entre diversas máquinas. Iniciava-se um intenso desenvolvimento de sistemas de arquivos distribuídos, conforme relata (COULOURIS, 2008) e (TANENBAUM, 2007).

Com relação à arquitetura desses sistemas, pode-se dizer que as propostas são bastante variadas. Existem tanto propostas baseadas no padrão cliente-servidor, como o Network File System (NFS); quanto em *peer-to-peer*, como o GlusterFS. Ainda, existem aquelas soluções cliente-servidor altamente elaboradas com diversos servidores especializados, como o LustreFS e CephFS. Nas subseções seguintes discutem-se os principais pontos de cada um desses sistemas.

LustreFS

O LustreFS é um sistema de arquivos distribuído e paralelo utilizado principalmente em supercomputadores. Foi desenvolvido para contornar as limitações de escalabilidade e performance do NFS, que tem sua arquitetura centralizada, ou seja, um único servidor de dados para todos os clientes. A premissa que envolve todo o desenvolvimento do LustreFS é justamente o inverso disso: o sistema de armazenamento será constituído por diversos *storages* (máquinas de alto rendimento especializadas em armazenar dados) e o LustreFS irá integrá-los, possibilitando que os clientes acessem esses *storages* como um único ambiente de armazenamento. Como há diversos equipamentos de armazenamento no sistema, pode-se efetivar a recuperação paralela de arquivos.

Para realizar o trabalho, o LustreFS faz uma distinção entre dados e os metadados de um arquivo. No ambiente existirão máquinas servidoras especializadas em servir metadados, *Metadata Server* (MDS) e outras

especializadas em servir os próprios dados dos arquivos, *Object Storage Server* (OSS). A essência da habilidade de escalonamento do LustreFS consiste no fato de se poder adicionar quantos OSSs forem necessários para se atingir a capacidade de armazenamento objetivada para o ambiente, conforme ilustrado na Figura 1. Ainda, caso o número de clientes seja muito alto e haja uma sobrecarga no servidor de metadados, esse pode ser replicado também repetidas vezes. Existe um servidor (MGS-Management Server) com informações globais da arquitetura do sistema, que permite o gerenciamento do ambiente.

No âmbito deste trabalho, é importante ter-se em consideração que o LustreFS é apropriado para o uso em ambiente com equipamentos especializados para armazenamento de dados (*storages*) e dessa forma, é uma solução que requer um investimento considerável.

A Figura 1 ilustra a arquitetura do sistema LustreFS, mostrando três máquinas cliente, um servidor de metadados e dois servidores de dados.

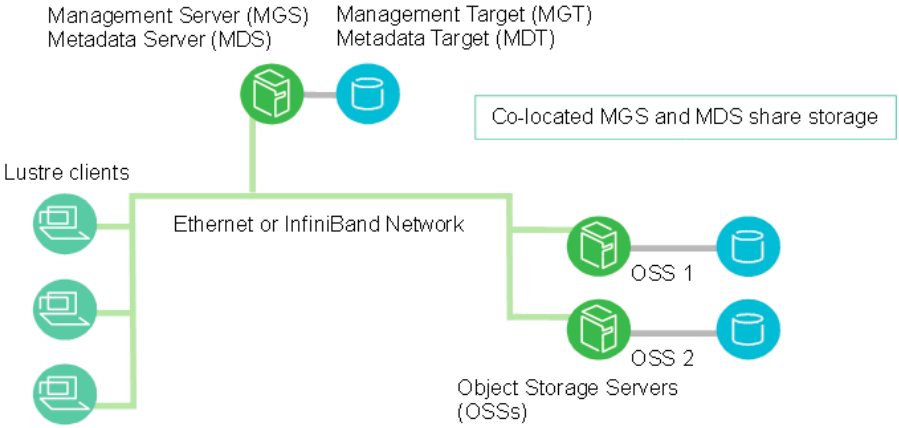

Figura 1. Arquitetura do sistema LustreFS.

Fonte: Intel Corporation (2017).

CephFS

O CephFS é um sistema de arquivos distribuídos fortemente inspirado no LustreFS, mas desenvolvido sob a premissa de operar com máquinas de armazenamento commodities. Dessa forma, está implícito que as unidades de armazenamento podem falhar eventualmente e o sistema só será eficiente se operar com réplicas dos objetos armazenados. Por padrão, o CephFS trabalha com três réplicas, mas essa quantidade pode ser modificada durante a configuração de novos volumes de dados. Esse princípio de se utilizar réplicas é a principal diferença entre os sistemas de arquivos desenvolvidos para serem utilizados em ambientes com equipamentos especializados em armazenamento de dados e sistemas de arquivos desenvolvidos para serem utilizados em ambientes com equipamentos commodities (servidores de boa qualidade, mas não especializados em armazenamento).

Outra característica marcante do CephFS é sua capacidade de trabalhar com as três formas de armazenamento: em blocos, em objetos e em arquivos. Dessa maneira, o sistema é capaz de suportar aplicações que exigem acesso ao mais baixo nível como os sistemas gerenciadores de banco de dados (SGBDs), aplicações de computação em nuvem que exigem o armazenamento de objetos e aplicações que obtêm arquivos como entradas e produzem arquivos como saída.

A arquitetura básica do CephFS baseia-se no uso de servidores responsáveis pelos dados, Ceph OSDs *Daemon*; servidores de monitoramento, que mantêm a estrutura do *cluster*, Ceph Monitor; e para o caso de armazenamento de arquivos, servidores de metadados, que mantêm os metadados dos arquivos, *Ceph Metadata Server* (MDS).

Por ocasião do início dos trabalhos com sistemas de arquivos distribuídos no LMB da Embrapa o Ceph ainda não era recomendado para uso em sistemas em produção.

GlusterFS

A GlusterFS é um sistema de arquivos distribuído baseado em duas premissas principais: o sistema de armazenamento será constituído de máquinas commodities e o efetivo gerenciamento dos arquivos no disco ficará a cargo de um sistema de arquivos local. Dessa forma, o GlusterFS concentra-se em resolver as questões relacionadas à integração dos diversos dispositivos de armazenamento e replicação dos dados.

O fato de a manipulação dos dados no disco ser implementada via um sistema de arquivos comum, como o XFS ou Ext4, faz com que o uso do GlusterFS seja extremamente simples. Do ponto de vista gerencial, é muito similar ao tradicional sistema NFS, porém, sem os gargalos operacionais desse, uma vez que a arquitetura do GlusterFS é peer-to-peer, com cada nó do *cluster* exercendo as mesmas funções de gerenciamento dos dados, como ilustrado na figura 2. Isso possibilita que o sistema escale consideravelmente sua capacidade de armazenamento, bastando a adição de novos nós. Ainda, o *throughput* do ambiente é ampliado no caso de escalonamento, pois os novos nós também irão atuar recebendo e enviando dados para os clientes.

A Figura 2 ilustra o sistema GlusterFS evidenciando os servidores físicos que são integrados para fornecer um volume de armazenamento homogêneo para os clientes.

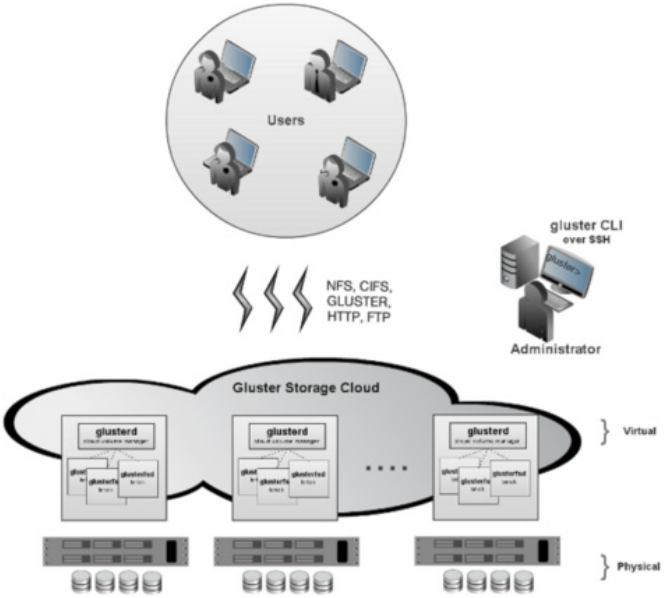

Figura 2. Sistema GlusterFS.
Fonte: Introducing Gluster... (2017).

Sobre a Seleção do GlusterFS

O projeto Tecnologias para computação distribuída, armazenamento de grandes volumes de dados e *workflow* científico, em suporte à pesquisa agropecuária (CompDist) avaliou os sistemas de armazenamento distribuídos anteriormente descritos e, ao final, optou pelo uso do GlusterFS. Para se compreender os motivos de tal escolha é necessário ter-se em consideração que os requisitos do ambiente distribuído a ser implementado incluíam o uso de máquinas commodities para armazenamento de dados intermediários gerados nas análises de dados científicos.

Dessa forma, o sistema LustreFS foi descartado, apesar de ser um sistema robusto e amplamente utilizando em centros de supercomputação. O problema é que ele exige equipamentos de armazenamento dedicados, ou seja, storages; e não há garantias de

que o LustreFS operaria adequadamente tendo máquinas commodities como suas fontes primárias de armazenamento.

Com relação ao CephFS, sua proposta é bastante interessante e promissora, no entanto, o sistema encontrava-se, ainda, em um estágio de maturação e a própria equipe de desenvolvimento não indicava o seu uso em ambiente de produção. Como o projeto tinha claro o objetivo de utilizar o sistema inicialmente no ambiente computacional do LMB e, posteriormente, em outras aplicações nas quais a abordagem de armazenamento distribuído se mostrasse adequada, então, descartou-se o sistema CephFS.

Por fim, restou escolhido para uso o sistema GlusterFS. Menos porque os outros se mostraram inviáveis e mais pelas suas próprias características. O GlusterFS é um sistema em estágio de desenvolvimento bastante estável, tendo como casos de usos diversas organizações que demandam o armazenamento de enormes quantidades de dados. O sistema foi projetado para atuar sobre equipamentos de armazenamento commodities, ou seja, servidores não especializados em armazenamento, mas que tenham uma quantidade significativa de discos; exatamente como ambicionado pelo projeto. Os testes efetuados com o sistema a priori mostraram um bom desempenho em relação ao *throughput* e, ainda, deixaram claro que o desempenho melhora com a adição de novos nós ao *cluster* de armazenamento. Essas são características essenciais buscadas em um ambiente de processamento intensivo.

Instalação e Configuração do GlusterFS

Esta seção objetiva apresentar as ações necessárias para a implantação bem-sucedida de um ambiente de armazenamento distribuído. Como foi discutido quando se apresentou o sistema GlusterFS, esse baseia-se na existência de servidores que trabalharão em conjunto seguindo uma

estratégia para o armazenamento de dados distribuídos e na existência de clientes, que acessarão esses dados utilizando um protocolo próprio do GlusterFS ou o difundido protocolo NFS. Não há nenhuma restrição em uma mesma máquina atuar como servidor e cliente nesse ambiente, se necessário.

Dessa forma, o procedimento de implantação consiste na instalação dos componentes de software que permitirão a operação dos servidores e, caso se deseje, na instalação dos componentes de software que permitirão a operação dos clientes via o protocolo específico do GlusterFS. Note que essa instalação nos clientes não é obrigatória, uma vez que eles poderão utilizar o protocolo NFS para se comunicar com os servidores GlusterFS. No entanto, ela é altamente recomendada, dado que o uso dos clientes GlusterFS irá possibilitar o acesso paralelo aos servidores com um considerável incremento de desempenho em diversas situações.

O GlusterFS roda sobre a plataforma Linux. Não há a necessidade de se escolher uma distribuição Linux em especial, a não ser o fato de atualmente as versões mais recentes do sistema serem distribuídas e pré-compiladas em pacotes .rpm. Dessa forma, o uso de uma distribuição que suporte a instalação desse tipo de pacote facilita consideravelmente o procedimento de implantação.

Evidentemente, a instalação do GlusterFS inicia-se uma vez que se tenha instalado e configurado adequadamente os servidores com o sistema operacional Linux. Em especial, é necessário que todo o ambiente de rede esteja operando adequadamente, antes de se iniciar a instalação dos componentes do GlusterFS. Particularmente, os nomes das máquinas nas quais ocorrerão as instalações são de vital importância para o processo, visto que serão frequentemente utilizados nos comandos de configuração. Também, enfatiza-se que é viável a instalação em um número qualquer de servidores; no entanto, para possibilitar exemplo eficaz, considera-se nesta seção a instalação hipotética em quatro servidores nomeados “server1”, “server2”, “server3” e s”erver4”. Logicamente, em uma instalação específica

deve-se trabalhar com os nomes locais das máquinas em questão. Pode-se também usar os endereços IPs dos equipamentos sem prejuízo para o correto funcionamento do ambiente distribuído, no entanto, essa abordagem não é cômoda para o administrador do sistema.

Basicamente, a implantação da plataforma de armazenamento distribuído passa pela instalação dos servidores GlusterFS, configuração dos volumes de dados e posterior instalação dos clientes GlusterFS. A seguir, uma discussão utilizando como exemplo os quatro servidores hipotéticos server1, server2, server3 e server4.

Instalando os servidores GlusterFS

A forma mais prática de se proceder com a instalação dos servidores GlusterFS é utilizando pacotes específicos para cada distribuição. No entanto, a indicação é de que não se utilizem os pacotes que estão disponíveis nos repositórios oficiais, por meio de sistemas tais como apt-get e yum; isso porque, o desenvolvimento do GlusterFS tem sido muito intenso e, invariavelmente, essas fontes tornam-se desatualizadas muito rapidamente. Assim, a melhor estratégia é obter os pacotes da versão mais estável diretamente no site https://www.gluster.org/download. A partir desse link, deve-se navegar segundo a distribuição de interesse. Para as necessidades deste trabalho, utilizou-se os pacotes .rpm disponibilizados para a distribuição CentOS, versão 6.7, na arquitetura x86_64. Para a obtenção dos arquivos necessários:

```
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-server-3.7.8-4.el6.x86_64.rpm
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-3.7.8-4.el6.x86_64.rpm
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-libs-3.7.8-4.el6.x86_64.rpm
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-api-3.7.8-4.el6.x86_64.rpm
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-client-xlators-3.7.8-4.el6.x86_64.rpm
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-cli-3.7.8-4.el6.x86_64.rpm
wget http://download.gluster.org/pub/gluster/GlusterFS/LATEST/CentOS/epel-6.7/x86_64/GlusterFS-fuse-3.7.8-4.el6.x86_64.rpm
```

Uma vez tendo-se os pacotes necessários, foram instalados os mesmos “em cada um dos servidores” server1, server2, server3 e server4. Para tanto, copiaram-se os referidos arquivos .rpm em cada servidor e executou-se, também em todos os servidores, os comandos:

```
rpm -i  GlusterFS-libs-3.7.8-4.el6.x86_64.rpm
rpm -i  GlusterFS-3.7.8-4.el6.x86_64.rpm
rpm -i GlusterFS-client-xlators-3.7.8-4.el6.x86_64.rpm
rpm -i GlusterFS-api-3.7.8-4.el6.x86_64.rpm
rpm -i GlusterFS-cli-3.7.8-4.el6.x86_64.rpm
rpm -i GlusterFS-fuse-3.7.8-4.el6.x86_64.rpm
rpm -i  GlusterFS-server-3.7.8-4.el6.x86_64.rpm
```

Por fim, a depender dos demais pacotes que o sistema operacional tenha instalado, é possível que alguma dependência externa tenha que ser satisfeita. Nesse caso, pode-se simplesmente utilizar o gerenciador de pacotes (ex.: apt-get ou yum) para resolver tais pendências. Apenas a título de exemplo, durante o procedimento relatado, foi necessário providenciar a instalação das bibliotecas *pyxattr* e *liburcu*. Para tanto, utilizou-se o gerenciador de pacotes yum:

```
yum install pyxattr
yum install userspace-rcu
```

Tendo-se procedido com a instalação sem erros, pode-se verificar a lista de pacotes instalados com o comando a seguir. Obtendo-se a listagem completa, tem-se um indicativo de que a instalação ocorreu com sucesso.

```
rpm -qa|grep gluster  #Este comando deve produzir a saída abaixo, caso a instalação tenha ocorrido adequadamente
GlusterFS-3.7.8-4.el6.x86_64
GlusterFS-libs-3.7.8-4.el6.x86_64
GlusterFS-cli-3.7.8-4.el6.x86_64
GlusterFS-server-3.7.8-4.el6.x86_64
GlusterFS-api-3.7.8-4.el6.x86_64
GlusterFS-client-xlators-3.7.8-4.el6.x86_64
GlusterFS-fuse-3.7.8-4.el6.x86_64
```

O próximo passo é realizar a inicialização do servidor GlusterFS em cada uma das máquinas server1, server2, server3 e server4. Utilize os comandos a seguir, em cada uma delas:

```
/etc/init.d/glusterd start
#E para garantir que o serviço seja reinicializado automaticamente quando o servidor for ligado, execute o comando a seguir em cada servidor.
chkconfig glusterd on
```

Configurando os volumes de dados

Esta seção discute a configuração dos volumes de dados nos ambientes de armazenamento distribuídos baseados no GlusterFS.

Ela se divide em duas subseções, sendo que na primeira tem-se uma discussão geral sobre as possibilidades de configurações disponíveis no GlusterFS e a segunda apresenta a configuração utilizada na arquitetura modelo adotada neste documento.

Primeiramente, deve-se enfatizar que o GlusterFS apoia a criação dos volumes de dados no conceito de *bricks*. Sendo sucinto, um *brick* nada mais é que um diretório reservado para uso dos servidores do GlusterFS. Como discutido anteriormente, o GlusterFS não realiza diretamente operações nos blocos dos discos; mas, ao contrário, o sistema terceiriza essas atividades para algum sistema de arquivo como o EXT4 e o XFS, por exemplo. Dessa forma, para que o GlusterFS possa operar e implementar as suas estratégias de armazenamento é necessário que os espaços de armazenamento (os *bricks*) estejam disponíveis e montados em algum ponto da árvore de diretórios da máquina. É altamente recomendável, apesar de não obrigatório, que os *bricks* sejam partições que tenham sido formatadas utilizando algum sistema de arquivo de confiança e estejam montadas. O fato de se utilizar partições completas impede que outros sistemas possam interferir no espaço de armazenamento disponibilizado para o GlusterFS; do contrário, o fato de dados serem escritos em outros diretórios poderia influenciar a disponibilidade de espaço para o GlusterFS se os *bricks* e tais diretórios estivessem compartilhando a mesma partição.

Possíveis estratégias de armazenamento com o GlusterFS

Para se compreender as estratégias de armazenamento do GlusterFS, deve-se levar em consideração que o ambiente de armazenamento foi projetado para ser implantado utilizando-se equipamentos commodities, ou seja, servidores que não são especializados em armazenamento. Nesta situação, o esperado é que alguns equipamentos irão falhar eventualmente, e para se garantir que o ambiente de armazenamento continue operando é necessário que se tenham réplicas dos arquivos armazenados em diferentes máquinas. Além do número de réplicas, pode-se optar por manter os arquivos armazenados integralmente ou divididos em chunks (*strippers*) nos vários nós do *cluster*. Com essas

opções, o GlusterFS oferece 5 (cinco) configurações alternativas para seus volumes de dados. A seguir uma breve apresentação de cada uma delas.

Volumes Distribuídos *(Distributed GlusterFS Volume)*

Nessa configuração, o GlusterFS permite que se agreguem quantos *bricks* desejar para a formação de um volume de armazenamento. Isto permite que se gere espaços de armazenamento extremamente grandes, apesar de não haver réplicas e, portanto, a falha em uma das máquinas do cluster levará o sistema de armazenamento a um estado de instabilidade, no qual parte dos arquivos deixam de estar disponíveis para acesso. A Figura 3 ilustra uma configuração de volumes distribuídos. Nota-se que o espaço de cada *brick* é acoplado para formar um volume de armazenamento com a capacidade de todos eles. Essa configuração é interessante quando se deseja ampliar a capacidade de armazenamento dos servidores individuais, no entanto, deve-se levar em consideração que neste caso não houve nenhuma réplica dos arquivos armazenados e isso implica que, na eventualidade de um dos servidores do *cluster* sofrer uma falha, o sistema como um todo estará comprometido, dado que alguns arquivos deixam de estar disponíveis no ambiente.

A forma genérica do comando para se criar esse tipo de configuração no GlusterFS é:

```
gluster volume create NEW-VOLUME_NAME NEW-BRICKS
```

Como exemplo, considere a criação de um volume chamado vol-agregado constituído de quatro *bricks*

```
gluster volume create vol-agregado server1:/var/gluster/brick server2:/var/gluster/brick server3:/var/gluster/brick server4:/var/gluster/brick
```

Na Figura 3 pode-se perceber que no caso dos volumes distribuídos do gluster, um arquivo estará inteiramente armazenado em um único brick. Desta forma, o espaço de cada *brick* se compõe para gerar um volume maior.

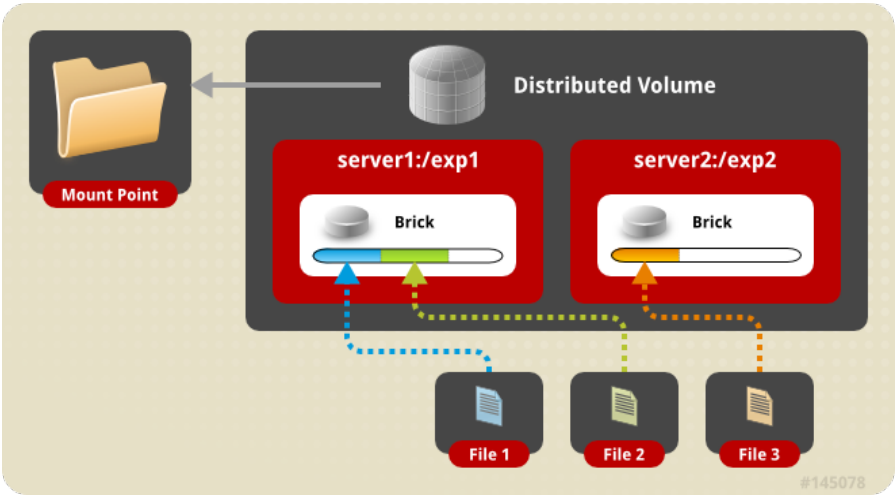

Figura 3. Volumes Replicados (Replicated GlusterFS Volume).
Fonte: Architecture (2017).

Nessa configuração, o objetivo é aumentar a resiliência do sistema, não a sua capacidade de armazenamento. Dessa forma, inserem-se *bricks* replicados, o que garante que haja mais de uma cópia, em servidores diferentes, para cada arquivo armazenado no sistema. A quantidade de réplicas pode ser estabelecida conforme a demanda do ambiente de armazenamento. A figura 4 apresenta uma visão da operação do sistema nessa configuração. Pode-se perceber que são mantidas cópias dos arquivos em servidores distintos no ambiente. Evidentemente, todas as questões relativas à manutenção da integridade dessas cópias são tratadas diretamente pelo GlusterFS sem a necessidade de nenhuma intervenção do usuário.

A forma genérica do comando para se criar esse tipo de configuração no GlusterFS é:

```
gluster volume create NEW-VOLUME_NAME replica COUNT NEW-BRICKS
```

Como exemplo, considere a criação de um volume chamado vol-replica constituído de quatro réplicas

```
gluster volume create vol-replica replica 4 server1:/var/gluster/brick server2:/var/gluster/brick server3:/var/gluster/brick server4:/var/gluster/brick
```

Apesar de os comandos que criaram os volumes vol-agregado e vol-replica serem bastante similares, o fato de ter se inserido a *flag* replica, modifica por completo a operacionalização dos volumes. No primeiro caso, tem-se um volume cujo espaço é constituído da soma dos espaços dos 4 *bricks* utilizados; no segundo caso, o espaço disponibilizado corresponde ao espaço do menor *brick*, mas por outro lado, cada arquivo inserido nesse volume terá 4 cópias.

A Figura 4 mostra que cada arquivo está sendo mantido de forma redundante em servidores separados.

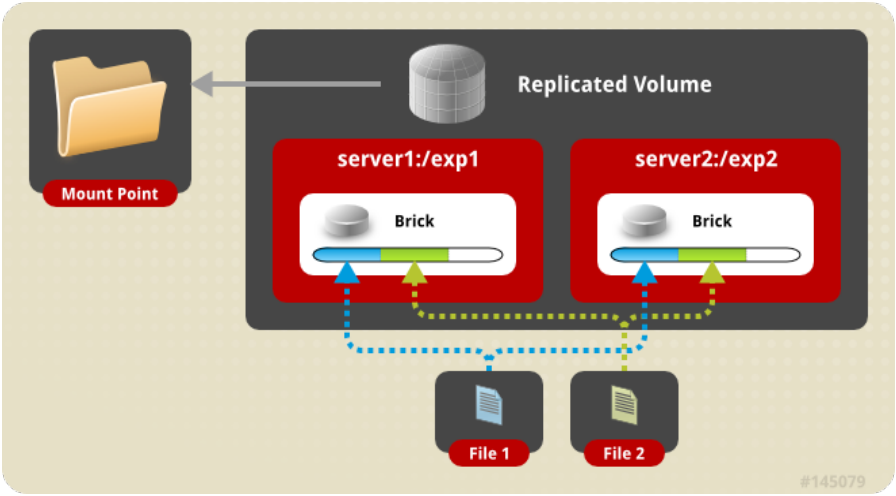

Figura 4. Volumes Replicados Distribuídos (Distributed Replicated GlusterFS Volume)
Fonte: Replicated... (2017).

Nessa modalidade tem-se uma junção dos dois fatores anteriores. Pode-se conseguir a resiliência do sistema com a inserção de réplicas e se pode conseguir grandes volumes de dados com o uso de *bricks* acoplados. A vantagem nesse caso é poder se trabalhar com conjuntos de dados realmente grandes, podendo ser superiores a 1 petabytes, de uma forma bastante segura. A figura 5 permite a visualização do mecanismo de operação do GlusterFS nessa configuração. Pode-se perceber que os arquivos estão sendo mantidos replicados em servidores distintos e, além disto, tem-se ainda a agregação de espaço de armazenamento entre as réplicas.

A forma genérica do comando para se criar esse tipo de configuração no GlusterFS é idêntica à anterior, mas o número de *bricks* deve ser múltiplo do valor COUNT:

```
gluster volume create NEW-VOLUME_NAME replica COUNT NEW-BRICKS
```

Como exemplo, considere a criação de um volume chamado vol-replicasDistribuidas constituído de duas réplicas com agregação de dois grupos de réplicas. No caso do exemplo, os *bricks* do server1 e server2 serão replicados. O mesmo se dará com os *bricks* do server3 e server4. Por fim, esses grupos replicados serão agregados em um volume maior.

```
gluster volume create vol-replicasDistribuidas replica 2 server1:/var/gluster/brick server2:/var/gluster/brick server3:/var/gluster/brick
server4:/var/gluster/brick
```

Como apresentado na Figura 5, cada arquivo é mantido redundantemente em servidores distintos. Além disto, fica claro que os espaços do volume 0 e do volume 1 estão acoplados em um volume maior.

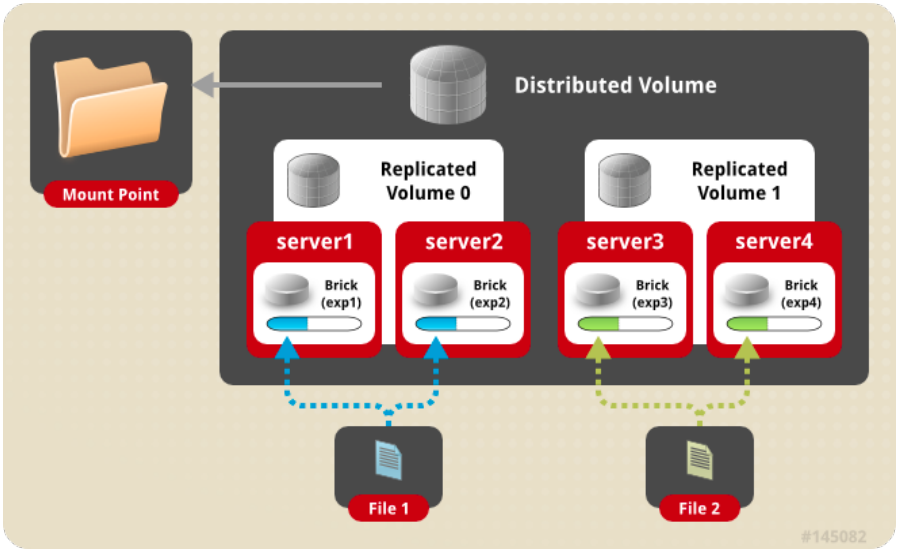

Figura 5. Volumes Segmentados (Striped GlusterFS Volumes).
Fonte: Distributed... (2017).

Até o presente, discutiram-se as possibilidades de configuração em que os arquivos são armazenados integralmente em um mesmo *brick* com ou sem redundância. Existe uma situação relacionada a desempenho que pode exigir uma configuração diferente. Considere o caso em que se tenham grandes arquivos que recebem uma considerável quantidade de acessos (acessos de clientes diferentes). Nessa situação pode-se sobrecarregar o servidor que mantém o *brick* no qual tal arquivo foi armazenado. Uma forma de tratar essa questão é dividir o arquivo em segmentos e armazenar tais segmentos em *bricks* diferentes. Dessa forma, a carga no acesso ao conteúdo do arquivo será distribuída no pool de máquinas que fazem parte do *cluster* de armazenamento; desde que existam acessos a diferentes partes do arquivo ocorrendo concomitantemente, ou seja, desde que vários clientes acessem o mesmo arquivo.

A configuração que acaba de ser descrita pode ser alcançada utilizando-se os volumes segmentados disponíveis no GlusterFS. A figura 6 permite uma compressão do mecanismo de funcionamento dos volumes segmentados. Pode-se concluir que o arquivo foi dividido em alguns segmentos (6 segmentos na ilustração) e esses são mantidos em bricks diferentes de acordo com a disponibilidade.

A forma genérica do comando para se criar esse tipo de configuração no GlusterFS é:

```
gluster volume create NEW-VOLUME_NAME stripe COUNT NEW-BRICKS
```

Como exemplo, considere a criação de um volume chamado "vol-segmentado" constituído de quatro *bricks* para segmentação.

```
gluster volume create vol-segmentado stripe 4 server1:/var/gluster/brick server2:/var/gluster/brick
server3:/var/gluster/brick server4:/var/gluster/brick
```

Na Figura 6 pode-se perceber que um mesmo arquivo foi dividido em vários segmentos e estes armazenados em diferentes *bricks* de acordo

com a disponibilidade de servidores. Desta forma, pode-se conseguir um ganho de desempenho para ocorrências que exijam intenso paralelismo.

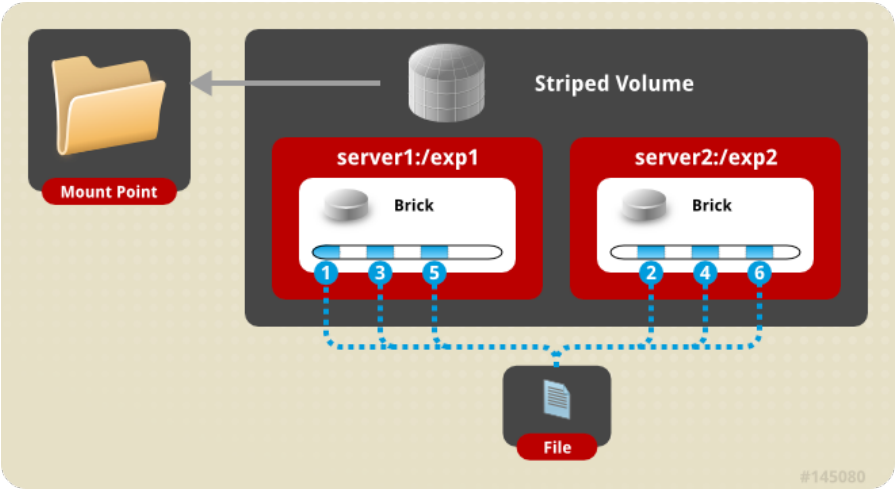

Figura 6. Volumes Segmentados Distribuídos (Distributed Striped GlusterFS Volume). Fonte: Striped ... (2017).

Para os casos em que além de se ter em grandes arquivos com uma alta carga de acessos, se tenha grande quantidade desses arquivos e que, portanto, seja necessário um amplo espaço de armazenamento, pode-se agregar diversos volumes segmentados, perfazendo um volume com enorme capacidade. Nesse caso, têm-se os volumes segmentados e distribuídos.

A forma genérica para se criar este tipo de volume no GlusterFS é:

```
gluster volume create NEW-VOLUME_NAME stripe COUNT NEW-BRICKS
```

Sendo assim, a quantidade de New-Bricks deve ser um múltiplo no número de *strip*. A figura 7 ilustra um volume segmentado agregado constituído da agregação de dois servidores sendo que cada um gerencia dois *bricks* segmentados. Como exemplo, considere a criação de um volume chamado vol-seg-agregado que seja segmentado entre

dois servidores e agregado também entre outros dois:

```
gluster volume create vol-seg-agregado stripe 2 server1:/var/gluster/brick server2:/var/gluster/brick
server3:/var/gluster/brick server4:/var/gluster/brick
```

A Figura 7 mostra *pools* de *bricks* operando na forma segmentada agregados para permitir a expansão na capacidade de armazenamento.

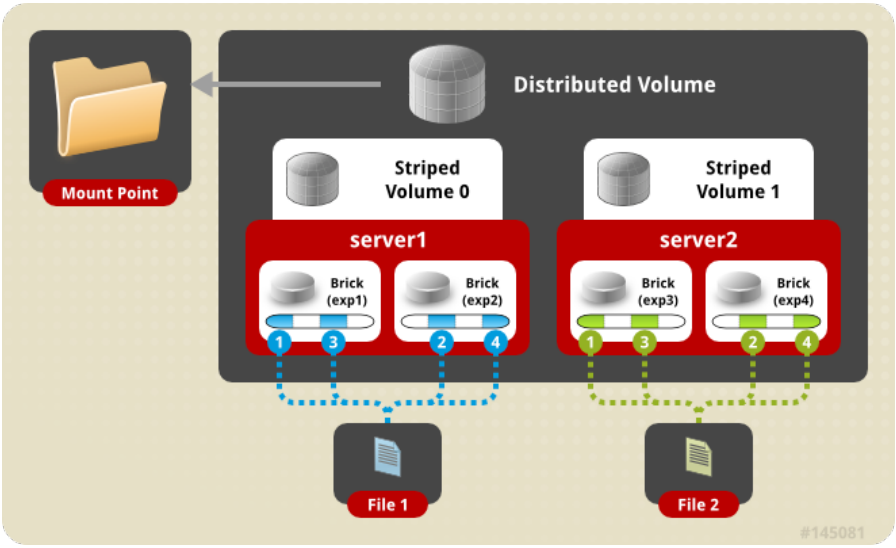

Figura 7. Pools de bricks.
Fonte: Distributed ... (2017).

Configuração da arquitetura utilizada no LMB

A arquitetura selecionada para uso no projeto trabalha com o armazenamento replicado distribuído utilizando duas cópias. Isso implica que, para uma garantia de disponibilidade, cada arquivo presente no sistema é mantido replicado em dois servidores; e para ganho de escalabilidade, pode-se adicionar quantos pares de servidores se queira. No momento da confecção deste documento, a infraestrutura de armazenamento distribuído presente no LMB constituía-se de seis servidores, cada qual trabalhando com oito discos rígidos de 4Tb. Dessa forma, havia uma capacidade de armazenamento bruta de 192Tb. No entanto, em função das réplicas, tinha-se 96Tb de espaço útil.

Para efeito de gerenciamento, atribuíram-se os nomes de gs1, gs2, gs3, gs4, gs5 e gs6 a cada um dos servidores que foram integrados ao cluster de dados baseado no sistema GlusterFS. Esses servidores tiveram cada um dos seus oito discos rígidos formatados com o sistema de arquivos XFS e as partições geradas foram montadas nos diretórios /gfs/d1, /gfs/d2, /gfs/d3, /gfs/d4, /gfs/d5, /gfs/d6, /gfs/d7 e /gfs/d8. Todos estes *bricks* foram agregados em um único volume de dados com replicação 2, conforme descrito na seção e discutido no subitem "volumes replicados distribuídos" da seção 12.

O GlusterFS permite duas formas de os clientes acessarem os dados do sistema: pode-se utilizar um cliente NFS comum ou um cliente próprio do GlusterFS. A segunda opção é a que garante o maior grau de paralelismo no acesso aos dados, pois uma máquina cliente não fica limitada à conexão com apenas uma máquina servidora, mas ao contrário, o cliente GlusterFS permite a comunicação com todos os servidores no cluster do GlusterFS. Dessa forma, a máquina cliente irá se comunicar com aquele servidor que efetivamente mantém uma cópia do arquivo a que se deseja ter acesso.

No ambiente implantado no âmbito do LMB, utilizaram-se os clientes próprios do GlusterFS como forma de acesso aos dados por parte das máquinas pertencentes ao *cluster* de processamento. Para tanto, foi necessário realizar as instalações e configurações descritas na seção a seguir.

Instalando e configurando os clientes GlusterFS

Como mencionado anteriormente, o acesso aos dados armazenados pelo GlusterFS não necessariamente depende da utilização de um cliente especial, isso pode ser feito utilizando-se um cliente NFS comum. Para tanto, deve-se selecionar uma das máquinas do *cluster* de dados (servidores GlusterFS) e efetuar a montagem do volume de dados na máquina cliente utilizando o comando mount. Como exemplo, considere uma possível montagem do volume de dados descrito na seção 17 em uma máquina de processamento tendo como referência o

servidor gs1. O comando necessário seria

```
mount -t nfs -o vers=3 mountproto=tcp gs1:/workspace /mnt/workspace
```

Nesse caso, o workspace em gs1:/workspace se refere ao nome dado ao volume de dados do GlusterFS no momento da sua criação. Esse nome poderia ser outro qualquer e foi escolhido simplesmente porque é mnemônico para as atividades de análise de dados do laboratório.

Por outro lado, apesar de simples, essa estratégia tem a desvantagem de limitar o paralelismo ao sistema de arquivos, visto que toda a comunicação com o GlusterFS deverá ocorrer via máquina servidora selecionada, impactando consideravelmente no seu desempenho. Sendo assim, a estratégia mais aconselhável é proceder com a instalação dos clientes nativos GlusterFS nas máquinas que irão acessar o *cluster* de dados.

Para efetuar a instalação, devem ser obtidos os pacotes do GlusterFS, GlusterFS-fuse, GlusterFS-libs e GlusterFS-client-xlators tal como discutido na seção 10 e executados na máquina cliente os comandos:

```
rpm -i  GlusterFS-libs-3.7.8-4.el6.x86_64.rpm
rpm -i  GlusterFS-3.7.8-4.el6.x86_64.rpm
rpm -i GlusterFS-client-xlators-3.7.8-4.el6.x86_64.rpm
rpm -i GlusterFS-fuse-3.7.8-4.el6.x86_64.rpm
```

Uma vez instalados tais pacotes com sucesso, basta executar a carga do módulo fuse:

```
#para efetuar a carga, execute o comando a seguir:
modprobe fuse
#Para verificar se o módulo foi carregado, faça:
dmesg | grep -i fuse
#Será exibida a mensagem "fuse init (API version 7.14)"
```

Posteriormente, basta montar o volume de dados desejado utilizando o comando:

```
mount -t GlusterFS  gs1:/workspace  /mnt/workspace
```

Nesse caso, apesar de o comando utilizado ainda se referir ao servidor de dados gs1, a operacionalização das consultas é completamente diferente. A máquina cliente irá utilizar os serviços do servidor gs1 apenas para determinar qual dos servidores do *cluster* detém uma cópia do arquivo desejado. Após essa etapa inicial, toda a comunicação para a efetiva transferência dos dados ao cliente se dará diretamente com o servidor detentor dos dados. Pode-se perceber que essa estratégia irá ampliar consideravelmente o paralelismo de acesso aos dados e contribuirá efetivamente para um aumento no desempenho do sistema de armazenamento.

Operações de manutenção da infraestrutura

O GlusterFS disponibiliza uma ferramenta em linha de comando para o gerenciamento da plataforma de armazenamento. Existe uma ampla gama de operações que podem ser realizadas e a lista continua a ser ampliada, dado que o sistema se encontra em efetivo desenvolvimento. No entanto, não faz parte do objetivo deste documento discutir todas as operações exaustivamente e, dessa forma, apresentar-se-ão as operações disponíveis apenas para ciência e se fará uma discussão mais elaborada apenas de duas operações diretamente relacionadas com as questões deste trabalho. Antes, no entanto, deve-se esclarecer que a ferramenta mencionada se chama GlusterFS. Dessa forma, para se poder ter acesso às ações que permitirão a operacionalização e manutenção do sistema de arquivos, deve-se utilizar o comando:

```
GlusterFS volume <operação> <opções>
```

Sendo que a operação a ser executada é uma dentre as seguintes:

- GlusterFS volume delete <nomeVolume> - exclui o volume especificado

- GlusterFS volume start <nomeVolume> - ativa o volume especificado

- GlusterFS volume stop <nomeVolume> - inativa o volume especificado

- GlusterFS volume add-brick <nomeVolume> <novoBrick> - adiciona um novo brick ao volume especificado

- GlusterFS volume remove-brick <nomeVolume> <brick> - remove o brick do volume especificado

- GlusterFS volume status – apresenta informações sobre volumes ou bricks

- gluster volume list – lista todos os volumes do cluster

Maiores detalhes serão fornecidos sobre as operações de inicialização dos volumes de dados e da adição de novos *bricks* ao sistema para expansão da capacidade de armazenamento. Com relação à primeira questão, deve-se considerar que o GlusterFS implementa um conceito de volumes ativos e inativos. Volumes recém-criados estão no estado inativos, impossibilitando que qualquer operação possa ser executada sobre os dados desses volumes. Além disso, sempre é possível inativar um volume que esteja operacional, em situações em que a plataforma de armazenamento irá passar por alguma manutenção. Desta forma, impede-se que novas escritas/leituras de dados sejam executadas sobre o volume em questão. Para se inativar/ativar um volume no cluster deve-se utilizar o comando GlusterFS com a sintaxe a seguir:

```
#Para ativar o volume nomeado workspace deve-se fazer
GlusterFS volume start workspace
#Para inativar o volume nomeado workspace deve-se fazer
GlusterFS volume stop workspace
```

Outra operação comum diz respeito à adição e remoção de *bricks* no sistema. Como exemplo, considere que um novo par de servidores (máquina gs7 e gs8) fosse adicionado ao *cluster* atual. Ainda, considere-se que essas novas máquinas tivessem oito discos rígidos cada. Assim, para adicionarem-se esses discos ao volume já operacional aumentando a sua capacidade de armazenamento útil para 128Tb, deve-se fazer:

```
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d1 sg8:/gfs/d1
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d2 sg8:/gfs/d2
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d3 sg8:/gfs/d3
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d4 sg8:/gfs/d4
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d5 sg8:/gfs/d5
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d6 sg8:/gfs/d6
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d7 sg8:/gfs/d7
GlusterFS volume add-brick workspace replica 2 sg7:/gfs/d8 sg8:/gfs/d8
```

Análise de Desempenho

Ao se analisar o desempenho de um sistema de arquivos pode-se focar em suas diversas características. As mais relevantes e mais comumente consideradas são o *throughput* e o número de Input/Output Operations Per Second (IOPS) ou operações de entradas e saídas por segundo. O *throughput* indica a capacidade de transferência disponibilizada pelo sistema por unidade de tempo. Normalmente é medido em megabytes por segundo (MB/s) ou mesmo gigabytes por segundo (GB/s). Essa característica dos sistemas de arquivos é particularmente importante para aplicações científicas que exigem a massiva manipulação de dados. Já o número de IOPS é indicativo da capacidade de operações que o sistema de arquivos pode executar, não importando a quantidade de dados que são manipulados em cada operação. Essa característica é mais importante para ambientes transacionais, nos quais o número de operações de entrada e saída podem ser elevados. Por fim, a latência indica o tempo necessário para se completar uma operação.

Existem várias suítes de aplicativos específicas para o teste de

desempenho de sistemas de arquivos, que possibilitam a realização de testes diversificados para determinar os mais diversos parâmetros de desempenho. No entanto, neste trabalho optou-se por uma abordagem diferente que estivesse relacionada mais diretamente com as exigências mais comuns no ambiente de processamento do LMB da Embrapa.

Dessa forma, fez-se uma análise de desempenho considerando-se os procedimentos de "leitura" e "escrita" de arquivos, visando identificar o *throughput* fornecidos pelos sistemas de arquivos em avaliação. Utilizou-se a ferramenta "dd", disponível no ambiente Linux, para emular as operações de leitura e escrita.

Com relação aos sistemas de arquivos, comparou-se o GlusterFS com o XFS e o NFS. O XFS é um sistema de arquivos local de alto desempenho e forneceu informações do desempenho esperado quando não estão envolvidos todos os gargalos necessários para disponibilizar as informações em rede. O NFS dos sistemas de arquivos mais simples é o mais difundido em ambientes de processamento distribuído para garantir que todos os nós do *cluster* possam acessar o mesmo conjunto de arquivos. No caso específico do presente trabalho, tem-se um particular interesse no NFS pois ele é utilizado com o sistema de arquivos padrão no ambiente de processamento de alto desempenho do LMB.

Cada um dos três sistemas de arquivos mencionados teve suas capacidades avaliadas quanto ao acesso linear e paralelo. O acesso linear significa os casos em que um único processo trabalha sobre um único arquivo. Já o acesso paralelo sinaliza as situações em que vários processos rodando em uma única máquina (caso do XFS) ou em várias máquinas do *clusters* (caso do GlusterFS e XFS) trabalham simultaneamente sobre vários arquivos diferentes.

Em resumo, foram analisados dois procedimentos sobre três sistemas de arquivos diferentes em duas situações distintas. Dessa forma, foram realizados doze testes cujos resultados são apresentados e discutidos a seguir. Veja as tabelas 1 e 2 para obter uma compilação dos tempos de

execução de cada um dos testes realizados.

Analisando-se os resultados dos testes efetuados pode-se perceber que o sistema de armazenamento local é o que apresenta melhor desempenho, tanto nos requisitos de capacidade de transferência de dados quanto no número de operações de IO por unidade de tempo. Esse resultado é exatamente o esperado, dado que um sistema de arquivos local não está sujeito a uma série de gargalos envolvendo a sincronização de operações e transferência via rede. No entanto, o melhor desempenho do sistema local não ajuda na prática, dado que em um *cluster* o usuário ficaria muito sobrecarregado se fosse obrigado a ficar transferindo os dados de máquina em máquina para o processamento. Dessa forma, deve-se estar claro que o sistema de armazenamento local é utilizado apenas como um *benchmark* e ainda são preferíveis sistemas que permitam acessos distribuídos, mesmo que com menor desempenho. As informações constantes na Tabela 1 são referentes a testes de desempenho linear dos sistemas de arquivos. Dessa forma, apenas um único arquivo está sendo lido ou escrito a cada instante. Essas operações se deram em blocos de 1K ou 1M, para se avaliar o desempenho mediante à variação na taxa de requisições.

Tabela 1. Testes de desempenho linear dos sistemas de arquivos.

Leitura	Blocos de 1K					
		10G		20G		40G
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
	XFS	29,03±3,14	338,89±23,24	71,90±9,22	275,55±31,88	136,70±19,55
	NFS	164,28±19,53	60,11±5,73	359,83±64,37	55,60±7,70	682,34±117,69
	GFS	262±1,15	37,27±0,16	541,75±26,04	36,11±1,70	993,25±20,07
	Blocos de 1M					
		10G		20G		40G
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
	XFS	14,25±0,5	685,92±23,25	27,5±0,58	710,46±14,92	57,5±0,58
	NFS	11,75±0,5	832,33±6,99	23±0,82	849,99±30,22	45,25±0,5
	GFS	22,75±0,96	429,82±17,78	46,25±1,26	422,53±11,33	92,75±3,59
Escrita	Blocos de 1K					
		10G		20G		40G
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
	XFS	43,84±3,46	224,19±18,96	70±2,81	279,02±10,89	143±10,93
	NFS	83,68±2,10	116,78±2,90	148,6±2,09	131,46±1,84	252,15±3,85
	GFS	313,07±8,42	51,46±0,45	765,80±56,81	25,61±1,84	1557,8±80
	Blocos de 1M					
		10G		20G		40G
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
	XFS	10±0,48	976,56±50,9	21±0,51	930,06±23,49	53±3,42
	NFS	9,8±0,61	1000,24±62,37	20,77±0,63	941,33±28,27	52,23±4,21
	GFS	50,85±6,52	195,09±25,23	102,50±15,37	195,16±32,59	213,85±26,29

As informações constantes na Tabela 2 são referentes a testes de desempenho paralelos dos sistemas de arquivos. Dessa forma, vários arquivos estavam sendo lidos ou escritos a cada instante. Essas operações se deram em blocos de 1K ou 1M, para se avaliar o desempenho mediante a variação na taxa de requisições.

Tabela 2. Testes de desempenho paralelos dos sistemas de arquivos.

		Blocos de 1K				
		10 arquivos		20 arquivos		40 arquivos
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
Leitura	XFS	1316±9,64	74,21±0,54	2669,33±4,51	73,17±0,12	5588,33±241,95
	NFS	2865±46,51	37,09±0,55	3480±99,74	56,15±1,60	4799,67±390,87
	GFS	11865±1009,36	8,27±0,7	13600,3±1784,5	14,53±1,93	18055±2027,51
		Blocos de 1M				
		10 arquivos		20 arquivos		40 arquivos
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
	XFS	99,33±17,56	1004,8±184,82	240±14,73	815,89±51,07	417±46,7
	NFS	339±31,75	289,69±25,98	740±161,79	273,83±67,92	2420,33±114,67
	GFS	225,57±111,11	510,03±191,41	394,67±43,46	500,21±58,21	660,33±203,5
		Blocos de 1K				
		10 arquivos		20 arquivos		40 arquivos
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
Escrita	XFS	425,9±3,48	229,31±1,88	889,1±11,37	219,71±2,79	2781,4±25,97
	NFS	575,25±11,91	169,83±3,44	1122,9±29,94	174,05±4,71	2132±83,53
	GFS	5843,6±1615,7	17,56±4,64	7440±964,21	26,57±3,72	12085±5714,99
		Blocos de 1M				
		10 arquivos		20 arquivos		40 arquivos
		Tempo(s)	Throughput(MB/s)	Tempo(s)	Throughput(MB/s)	Tempo(s)
	XFS	261,67±58,32	384,49±75,94	535,33±131,2	379,32±89,41	1007±104,65
	NFS	550,5±12,45	177,47±4,10	1179,67±173,49	167,8±22,75	3525,33±2352,42
	GFS	448,4±218,85	307,02±255,72	629,1±136,93	326,33±83,93	902±284,28

Fica claro, também, que o GlusterFS apresentará seu melhor desempenho nos casos de acessos paralelos (acessos concorrentes a diversos arquivos por diversos usuários) e em operações em que a transferência de dados se dê em grandes blocos. O GlusterFS sofre de uma série de gargalos na sincronização de operações e há um custo considerável para se iniciar uma operação. Dessa forma, para um melhor desempenho é desejável que uma vez iniciada, a operação transfira uma quantidade razoável de dados. Nos testes efetuados, fica evidente uma diferença de desempenho entre extremos que envolveram a transferência de 1KB e 1MB por operação.

Análise de Custos

A análise de custos relacionados à implantação de um sistema de armazenamento distribuído baseado em equipamentos commodities utilizará as aquisições realizadas no âmbito do LMB como referência.

Irá se comparar os custos envolvidos na aquisição de dois *storages* utilizados em um sistema de armazenamento dedicado e seis servidores *commodities* utilizando em um sistema de armazenamento distribuído. Para se acompanhar o cálculo dos custos envolvidos na aquisição dos dois sistemas mencionados, deve-se ter em consideração que o sistema de armazenamento dedicado foi adquirido em três fases: aquisição de cada um dos storages e aquisição de um upgrade de um deles. O sistema de armazenamento distribuído também foi adquirido em três fases: aquisição de dois servidores, aquisição dos discos rígidos a serem usados e, então, aquisição de mais quatro servidores. Todos os equipamentos foram adquiridos via pregão eletrônico, na modalidade de melhor preço, razão pela qual se têm equipamentos na mesma categoria fornecidos por fabricantes distintos. Também, as aquisições não foram realizadas nas mesmas datas e, assim, para buscar uma melhor equalização dos preços, esses foram corrigidos pela inflação até a data da última compra. Esse não seria o índice mais adequado para equipamentos de tecnologia, mas por falta de uma alternativa melhor, trabalhou-se dessa maneira. As Tabelas 3 e 4 resumem os preços envolvidos. Nas tabelas, tornava-se mais intuitivo considerar os custos envolvidos na aquisição dos discos, diretamente nos equipamentos que receberam esses discos. Dessa forma, as informações de custos dos discos não aparecem explicitamente, mas foram totalmente consideradas nos cálculos.

Tabela 3. Custos envolvidos na infraestrutura de armazenamento dedicada.

Data Aquisição	Equipamento	Capacidade Útil(TB)	Custo	Custo por Terabyte	Custo por Terabyte Geral
23/02/2011	Storage IBM DS	102TB	R$325800,50	R$3194,12	R$2739,96
13/08/2014	SGI InfiniteStorage	154TB	R$375628,22	R$2439,15	

A aquisição do sistema de armazenamento distribuído envolveu 3 fases pelos motivos a seguir. Inicialmente, o projeto estava investigando a viabilidade do sistema para uso no ambiente de produção do LMB. Nesse momento, foi realizada a aquisição de apenas dois equipamentos para serem utilizados nos procedimentos de avaliação. Posteriormente, quando já se tinham resultados constando a efetividade do sistema

para uso em algumas categorias de análises de dados, então se realizou uma nova aquisição de mais quatro equipamentos. Por outro lado, por uma questão de disponibilidade de recursos de custeio e restrições em recursos de investimento, fez-se a aquisição dos discos rígidos separadamente. Essa estratégia, quando considerado todo o esforço empregado, não tem um custo-benefício vantajoso e deve ser evitada.

Tabela 4. Custos envolvidos na estrutura de armazenamento distribuído.

Data Aquisição	Equipamento	Capacidade Útil(TB)	Custo	Custo por Terabyte
09/12/2013	Dois servidores IBM	30TB	R$42193,69	R$1406,45
28/08/15	Quatro servidores HP	60TB	R$46627,38	R$777,12

Cumpre esclarecer que o custo médio por terabyte no sistema de armazenamento distribuído apresentado na tabela 4 já considera a necessidade de se trabalhar com o sistema replicado, ou seja, o ambiente tem a capacidade de armazenar até 180TB de dados, mas como se está trabalhando com duas réplicas, então o ambiente armazena 90TB. No entanto, essa configuração é muito flexível e pode-se facilmente conviver com situações em que a replicação dos dados não fosse necessária. Nesse caso, ter-se-ia um espaço útil superior ao considerado na tabela 4 e o custo seria ainda menor do que esse apurado.

Por outro lado, mesmo trabalhando-se com a hipótese de que o sistema de armazenamento distribuído apenas seria utilizado na sua configuração replicada, ainda assim, o custo médio por terabyte armazenado é bastante inferior. O sistema de armazenamento dedicado custa R$2739,96 por terabyte armazenado, ao passo que o sistema de armazenamento distribuído custa R$986,89 por terabyte armazenado. Dessa forma, o custo do armazenamento distribuído equivale a 36% do valor gasto no armazenamento dedicado.

Os custos do armazenamento distribuído são muito mais vantajosos, no entanto deve-se lembrar mais uma vez de que esse tipo de armazenamento não é eficaz para todos as categorias de dados,

daqueles sensíveis para a organização, que demandaram grandes esforços para serem gerados, não podem ser armazenados senão em uma infraestrutura altamente especializada que possua as mais extensas garantias de segurança à informação que a tecnologia atual possa fornecer. Por outro lado, dados temporários gerados em análises a partir dos dados primários podem tranquilamente ser mantidas, enquanto necessário, em uma infraestrutura com menores garantias.

Conclusão

O GlusterFS mostrou-se uma efetiva ferramenta para fazer frente aos desafios de se trabalhar com grandes volumes de dados para análises científicas. É altamente flexível nas configurações da arquitetura de armazenamento, garante uma boa disponibilidade quando se utiliza a redundância, tem uma performance aceitável trabalhando-se com um número adequado de máquinas no *cluster* e seu custo é acessível.

O sistema é bastante flexível em relação às possibilidades de esquemas de redundância disponíveis para uso, e disponibiliza configurações que podem maximizar os tamanhos dos volumes de dados a serem armazenados ou o desempenho do sistema ou a segurança da informação armazenada. O interessante é que, se necessário, pode-se utilizar essas configurações diversas concomitantemente de uma forma bastante simples.

Com relação ao desempenho, o GlusterFS não é ideal quando se está trabalhando em acessos lineares ao sistema. No entanto, seu desempenho melhora consideravelmente quando se consideram os acessos em paralelo. Como em um *cluster* uma grande parte das rotinas de análises executadas rodam em paralelo, as limitações do sistema para os casos de acesso lineares não devem impactar negativamente na maior parte do tempo. Ainda há que se considerar que mesmo aquelas análises que dependam de acesso linear aos arquivos podem não sofrer impacto significativo se rodarem sobre o GlusterFS. O motivo é que a grande maioria dos programas utilizados nessas análises têm um comportamento cpu-bound (muito dependente

de cpu e pouco acesso de IO). Nesse caso, os programas geralmente carregam os dados para a memória principal e processam-nos por horas ou mesmo dias. Dessa forma, se por exemplo, o tempo de carga passar de 15 minutos para 150 minutos; mesmo assim, o tempo total da análise ainda pode sofrer um impacto mínimo, se o processamento exigir centenas de horas.

O custo de armazenamento de cada megabyte é extremamente atrativo para o sistema de armazenamento distribuído e, além do mais, a flexibilidade de se ampliar a capacidade de armazenamento ao longo do tempo, de uma maneira bastante simples, é também muito conveniente.

Fica claro, portanto, que o uso do GlusterFS deve ser mantido e intensificado no âmbito do ambiente computacional para análise de grandes volumes de dados do LMB.

Referências

ARCHITETURE. [2017]. Disponível em: <http://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture/>. Acesso em: 30 maio 2017.

ARCHITETURE types of volumes. [2017]. Disponível em: <https://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture/>. Acesso em: 30 maio 2017.

COULOURIS, G. S.; DOLLIMORE, J.; KINDBERG, T. **Sistemas distribuídos**: conceitos e projeto. Tradução João Tortello. 4. ed. Porto Alegre, 2008. 784 p. il.

DISTRIBUTED replicated glusterfs volumes. [2017]. Disponível em: <http://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture>. Acesso em: 30 maio 2017.

DISTRIBUTED striped glusterfs volume. [2017]. Disponível em: <http://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture>. Acesso em: 30 maio 2017.

HEY, T.; TANSLEY, S.; TOLL K. **O quarto paradigma**: descobertas científicas na era da eScience. Tradução Leda Beck. São Paulo, 2011. 263 p. il.

INTEL CORPORATION. **Lustre* software release 2.x**. Disponível em: <xhttp://doc.lustre.org/lustre_manual.xhtml>. Acesso em: 30 maio 2017.

INTRODUCING gluster file system. Disponível em: <http://gluster.readthedocs.io/en/latest/Administrator%20Guide/GlusterFS%20Introduction>. Acesso em: 30 maio 2017.

REPLICATED glusterfs volume. Disponível em: <https://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture/>. Acesso em: 30 maio 2017.

STRIPED Glusterfs Volume (2017). Disponível em: <http://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture/>. Acesso em: 30 maio 2017.

TANENBAUM, A. S. E STEEN, M. V. **Sistemas distribuídos**: princípios e paradigmas. 2. ed. São Paulo, 2007. 402 p. il.

TANENBAUM, A. S. **Sistemas operacionais modernos**. 3. ed. São Paulo: Pearson, 2010. 653 p. il.

Embrapa

Informática Agropecuária

MINISTÉRIO DA
AGRICULTURA, PECUÁRIA
E ABASTECIMENTO

BRASIL
GOVERNO FEDERAL

CGPE 14075