

Raphael Brito Reis

**Teste de desempenho de sistemas de arquivos
distribuídos em nuvem computacional privada
no IFSC câmpus São José**

São José - SC

outubro/2015

Raphael Brito Reis

**Teste de desempenho de sistemas de arquivos
distribuídos em nuvem computacional privada no IFSC
câmpus São José**

Monografia apresentada à Coordenação de
Sistemas de Telecomunicações do Instituto
Federal de Santa Catarina para a obtenção
do diploma Bacharel em Sistemas de Telecomunicações.

Instituto Federal de Santa Catarina – IFSC

Campus São José

Sistemas de Telecomunicações

Orientador: Ederson Torresini

São José - SC

outubro/2015

Raphael Brito Reis

Teste de desempenho de sistemas de arquivos distribuídos em nuvem computacional privada no IFSC câmpus São José/ Raphael Brito Reis. – São José - SC, outubro/2015-38 p. : il. (algumas color.) ; 30 cm.

Orientador: Ederson Torresini

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC
Campus São José
Sistemas de Telecomunicações, outubro/2015.

1. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Título

Raphael Brito Reis

**Teste de desempenho de sistemas de arquivos
distribuídos em nuvem computacional privada no IFSC
câmpus São José**

Monografia apresentada à Coordenação de
Sistemas de Telecomunicações do Instituto
Federal de Santa Catarina para a obtenção
do diploma Bacharel em Sistemas de Telecomunicações.

Trabalho aprovado. São José - SC, 15 de outubro de 2015:

Ederson Torresini
Orientador

Professor
Convidado 1

Professor
Convidado 2

São José - SC
outubro/2015

Dedico aos meus pais, pois sem eles este trabalho e muitos dos meus sonhos não se realizariam.

Agradecimentos

Primeiramente, agradeço ao meu orientador Ederson Torresini, pelo pouco tempo que tinha hábil ter me incentivado e ajudado nas correções deste trabalho. E à minha namorada, Aline, companheira que sempre acreditou na realização deste sonho.

*“Que os vossos esforços desafiem as impossibilidades,
lembrai-vos de que as grandes coisas do homem
foram conquistadas do que parecia impossível.
(Charles Chaplin)*

Resumo

Com esse trabalho, pretendo analisar sistemas de arquivos distribuídos para futura implementação no IFSC câmpus São José, para verificar as possibilidades de melhor desempenho e menor custo computacional para os laboratórios. Serão realizados testes com os sistemas comuns em centros de processamento de dados (CPDs), tais como Ceph, GlusterFS e Lustre, bem como os formatos de arquivos EXT4, BTRFS E XFS, para mostrar qual a melhor combinação para o ambiente de nuvem computacional em questão.

Abstract

This work, I analyze distributed file systems for future implementation in the IFSC campus São José, to check the best performance opportunities and lower computational cost for laboratories . Tests with common systems for data processing centers will be performed (CPDs) , such as Ceph , GlusterFS and Black, as well as the EXT4 file formats , BTRFS and XFS , to show what the best combination for the cloud computing environment in question.

Sumário

1	INTRODUÇÃO	19
1.1	Motivação	19
1.1.1	Objetivo	20
2	SISTEMAS DISTRIBUÍDOS	21
2.1	Definição sistemas distribuídos	21
2.2	Sistemas de Arquivos distribuídos	21
2.2.1	Vantagens e desvantagens	21
2.2.1.1	Vantagens	22
2.2.1.2	Desvantagens	22
3	FORMATOS DE ARQUIVOS UTILIZADO NOS TESTES	23
3.0.2	Ext4	23
3.0.3	XFS	24
3.0.4	Btrfs	24
4	SISTEMAS DE ARQUIVOS UTILIZADOS NOS TESTES	27
4.0.5	Ceph	27
4.0.6	GlusterFS	28
4.0.7	Lustre	30
5	CENÁRIO DE TESTES	33
5.0.8	Equipamentos de rede	33
6	METODOLOGIA DE TESTES	35
	Referências	37

1 Introdução

O principio básico de computação em nuvem, consiste em acessar dados ou programas que estão armazenados em um servidor na rede local ou na internet. Quando um cliente solicita esse dado ou programa, a requisição chega ao servidor de arquivos distribuído, que fica encarregado de processar a informação solicitada e depois enviar ao cliente.

O atual cenário do IFSC de São José, consiste em um CPD com muitos recursos disponíveis porém pouco aproveitados. Com várias demandas de processamento na nuvem que aparecem como: Quartus, Matlab, Netkit, ou para publicação de dados na web, como o portal IFSC, wiki IFSC, blog integrado e o moodle de tele. Com o crescimento anual de pessoas acessando esses serviços, resulta-se em congestionamento e até falhas do mesmo.

O que este trabalho propõe, é a realização de alguns testes para verificar o desempenho e estabilidade de um sistemas de arquivos. Utilizando softwares como Ceph, que segundo os desenvolvedores garantem uma grande estabilidade em gerenciar os dados, o Lustre que conforme seus criadores é indicado para sistemas de arquivos muito grande e o GlusterFS que é um produto intermediário.

Com a combinação de formatos de arquivos como o Ext4 que vem de uma evolução do Ext2 e Ext3, possuindo uma grande estabilidade, porem possui pouca performance. O XFS que tem o objetivo principal alta performance e Btrfs que é um novo formato de arquivo. Com esses testes podemos garantir o real propósito de um sistema de arquivos distribuídos, que é a uniformidade, disponibilidade e acesso remoto dos dados.

Posteriormente com os resultados obtidos através dos testes, se pensa em uma nova implementação de um sistema de arquivos distribuído, por parte do administrador de redes do campus IFSC-São José, para se obter uma melhor performance dos dados em nuvem. Onde cada aluno irá ter um login e senha, na qual poderá acessar seus conteúdos de qualquer máquina dentro ou fora do campus. O servidor ficará responsável pelo processamento dos arquivos, entregando ao usuário apenas a tela do programa, ou apenas armazenando e mostrando os dados desejados.

1.1 Motivação

Com o crescimento eminente de dados, é necessário melhorar o armazenamento e processamento do mesmo. No entanto, por melhor que seja o hardware e software, pode-se haver uma falha nos itens citados. Isso levou à descentralização dos serviços e dados do campus, pois, se houver uma perda de hardware ou software, afeta somente aquele servidor

e não compromete o serviço inteiro da rede. Com esse TCC, pretende-se testar diversos sistemas de arquivos, realizando testes de desempenho, disponibilidade, recuperação de dados, processamento, para escolher o mais indicado para o cenário do IFSC de São José.

1.1.1 Objetivo

Realização de testes de desempenho e estabilidade em um sistema de arquivo distribuído, para se ter um melhor aproveitamento da nova infraestrutura do sistema de arquivos distribuídos, tanto para serviços web como página do campus, ambiente wiki , portal do aluno quanto para serviços de ensino, a exemplo das ferramentas de aprendizagem como Quartus, Matlab e Netkit. Podendo ajudar os alunos da instituição a estudar fora do campus, em sua residência, em vista de que nem sempre se têm um laboratório disponível e específico, pois muitas vezes o mesmo está sendo ocupado por uma outra turma.

2 Sistemas distribuídos

Nesse capítulo será abordado alguns pontos importantes deste trabalho, como as definições de sistemas de arquivos e sistemas de arquivos distribuídos, bem com as vantagens e desvantagem de seu uso.

2.1 Definição sistemas distribuídos

Segundo (COULOURIS JEAN DOLLIMORE,), podemos definir sistemas distribuídos como :

Definimos um sistema distribuído como aquele no qual os componentes de hardware ou software, localizados em computadores interligados em rede, comunicam-se e coordenam suas ações apenas enviando mensagens entre si. Exemplos de sistemas distribuídos : jogos online, e-mails, redes sociais, e-commerce.

Apoiado sobre essa ideia, um sistema distribuído são computadores de clientes, estações de trabalho ou servidores, conectados simultaneamente em rede, com um software. A sua função é de reproduzir a percepção de um sistema único ao usuário.

2.2 Sistemas de Arquivos distribuídos

Para (CARLOS; TAVARES,) sistemas de arquivos distribuídos é:

... é um sistema de arquivos no qual os arquivos nele armazenados estão espalhados em *hardware* diferentes, interconectados através de uma rede. Para que esses arquivos sejam acessados, modificados e que sejam criados outros arquivos é necessária uma estrutura que permita tais operações.

Com esse pensamento, um sistema de arquivo distribuído são diferentes servidores que armazenam dados de seus clientes, conectados por uma rede. Ou seja os arquivos ficam em servidores distintos, porém para o usuário, apresenta ser um único servidor. Para fazer alguma alteração nesses arquivos, necessita de um sistema de arquivos.

2.2.1 Vantagens e desvantagens

Nessa sessão, serão listados as principais vantagens e desvantagens em se utilizar um sistema de arquivo distribuídos na infra estrutura da rede.

2.2.1.1 Vantagens

- Econômico: usando um cluster (um conjunto de computadores), se tem um baixo custo em relação aos sistemas centralizados, como o de um mainframe. (ANDRÉ,);
- Crescimento incremental: se pode aumentar o sistema à medida que se precise, apenas adicionando um servidor.(ANDRÉ,);
- Compartilhamento do mesmo dado: pois permite que vários usuários acessem uma base de dados comum. (ANDRÉ,);
- Maior confiabilidade: os dados ficam armazenados em várias máquinas e se pode fazer o backup do mesmo;
- Maior domínio das aplicações, se pode controlar os dados e programas que os usuários acessam;
- Segurança: configura o acesso as informações corretamente.

2.2.1.2 Desvantagens

- Desenvolvimento trabalhoso;
- Deve se ter uma boa infraestrutura de rede para a mesma não ficar lenta;
- Custo: se não houver uma boa infraestrutura, o custo se torna elevado para se ter os equipamentos corretos.

3 Formatos de arquivos utilizado nos testes

Foi escolhido para a realização dos testes, os formatos de arquivos: Ext4, XFS e Btrfs, pois contém escalabilidade, desempenho, confiabilidade e compatibilidade com o sistema de arquivos distribuídos como o Ceph, Lustre e GlusterFS.

3.0.2 Ext4

Conforme, ([LAIRD](#),), em 1992 Ext foi criado por Remy Card e era o primeiro a utilizar sistema de arquivo virtual no Linux, tinha suporte de arquivos até 2 gigabytes, logo em seguida em 1993, foi lançado o Ext2 com suporte para arquivos de 32 TB. O avanço mais importante veio no Ext3, onde se introduziu o conceito de journaling, deixando o sistema extremamente confiável, porém em termos de desempenho ficava atrás dos outros formatos. Posteriormente o Ext4, ficou extremamente estável, pensado para usuários com grandes matrizes de disco, muitos subdiretórios e registro de data e hora precisos, possui compatibilidade com Ext2 e Ext3. Ou seja, se utiliza o Ext4 pensando na estabilidade e não em performance. Principais características são:

- Permite sistemas de arquivos de até 1024 pebibytes.
- Arquivos de 16 TiB.
- Desfragmentação online, ou seja, no momento que o arquivo está sendo alocado ele é desfragmentado.
- Registro de data e hora na escala de nano segundos.
- No formato de arquivo Ext3, se tinha um limite de 32 mil subdiretórios, na versão Ext4 esse limite foi removido.
- Reduzido o tempo de exclusão de grandes arquivos.
- A reserva em disco para o arquivo, atrasa até o último momento, podendo gerar uma performance melhor.
- Existe uma verificação no registros de dados, gerando mais confiabilidade e desempenho.
- Verificação de disco mais rápida, pois com a nova estrutura, o FSK só verifica arquivos que estão sendo utilizados.

3.0.3 XFS

Segundo ([LINUX](#),) XFS é um sistema de arquivo de criado pela Silicon Graphics, Inc., que depois foi adaptado ao Linux. Para ([PEREIRA](#),) o principal objetivo do sistema é alta performance e para trabalhar em grande escala, desenvolvido com arquitetura 64 bits, mas compatível com 32 bits. Muito indicado para aplicações de banco de dados, porém utiliza muitos recursos cache da memória RAM. Segundo ([ABREU](#),) Principais características são:

- Em sistemas de 64bits, tamanho máximo de arquivo é de 8 EB;
- Em sistemas de 32bits, tamanho máximo de arquivo é de 16 TiB;
- Blocos variados de 512 bytes à 64Kb;
- Mantém a gravação dos metadados dos arquivos no journaling, mantendo o sistema de arquivos preservado;
- Possui desfragmentação online;
- Contém redimensionamento online, ou seja, pode aumentar o tamanho do sistemas de arquivos, no momento que quiser. Porém não se pode diminuir o mesmo;
- Executa operação de escrita e leitura melhor em arquivos de tamanho grande;
- Operações que usam o metadados em XFS são mais lentas que em outros sistemas de arquivos;

3.0.4 Btrfs

De acordo com ([RUSSO](#),) começou a ser desenvolvido pela Oracle, porém hoje conta com participação de empresas como Intel, SUSE, Red Hat e Fujitsu. Um dos principais objetivos deste formato é inibir a perda de dados por problemas, permitindo que aumente ou diminua o tamanho do sistema de arquivos montado. Isso faz com que se tenha grande armazenamento de dados e com uma grande performance. Sistema ainda em desenvolvimento, porém considerado estável, se encontra alguns problemas como taxa de ocupação, onde se indica o disco cheio mesmo tendo espaço livre. Principais características para ([BTRFS](#),):

- Tamanho máximo de arquivo: 16 EiB;
- Desfragmentação de espaço online;
- Conversação com sistemas de arquivos Ext3/ Ext4 existente;

- Alocação dinâmica de inodes;
- Snapshots com o sistema de arquivos montados;
- Checksums dos dados e metadados, o que garante a integridade dos dados armazenado;
- Fácil administração.

4 Sistemas de arquivos utilizados nos testes

A escolha dos programas Ceph, GlusterFS e Lustre que gerenciam sistemas de arquivos, se deve à alta confiabilidade, desempenho, escalabilidade, ser open source e permitir uma melhor segurança, disponibilidade e redundância dos dados armazenados nos servidores.

4.0.5 Ceph

Segundo os desenvolvedores do Ceph, ([CEPH, a](#)), tem capacidade para gerenciar grandes quantidades de dados e clientes, compatível com formatos de arquivos como EXT4, XFS e BTRFS . Ele armazena os dados como objetos, que correspondem a um arquivo no sistema de arquivos, como podemos ver na Figura 1. Os objetos tem como parâmetros: dados binários e metadados, que consta em um conjunto de pares com nome e valor.

Este programa não possui gateway centralizado, os clientes interagem diretamente com o Ceph OSD, ele cria réplicas de objetos, garantindo a segurança dos dados e gerando alta disponibilidade. Utilizando de um algoritmo denominado CRUSH, Ceph elimina a centralização, calculando de forma eficiente as informações sobre a localização dos dados.

Antes de um cliente conseguir ler ou escrever dados no sistema de arquivos, ele precisa ter contato com um monitor Ceph, para obter uma cópia mais atual do cluster. Pode-se ter contato com apenas um monitor, porém se o mesmo falhar, o cliente não irá conseguir mais realizar as operações. São recomendados no mínimo três monitores e que use um número ímpar, pois se tem uma resistência maior que número par.

A autenticação do cliente é recomendado, porém não é uma prática obrigatória que se utilize chaves secretas compartilhadas, o cliente e o cluster do monitor possuem uma cópia da chave. Caso as chaves não sejam idênticas, não haverá permissão para o usuário realizar as operações. Na prática quando o usuário realiza o login, o monitor gera

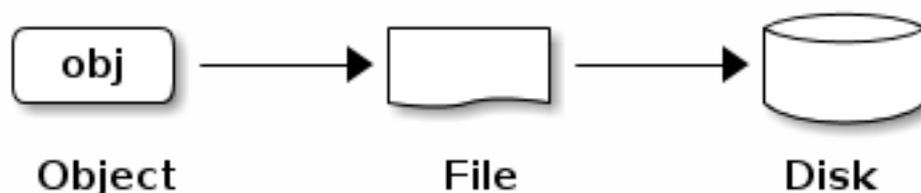


Figura 1 – Armazenamento de Dados ([CEPH, a](#))

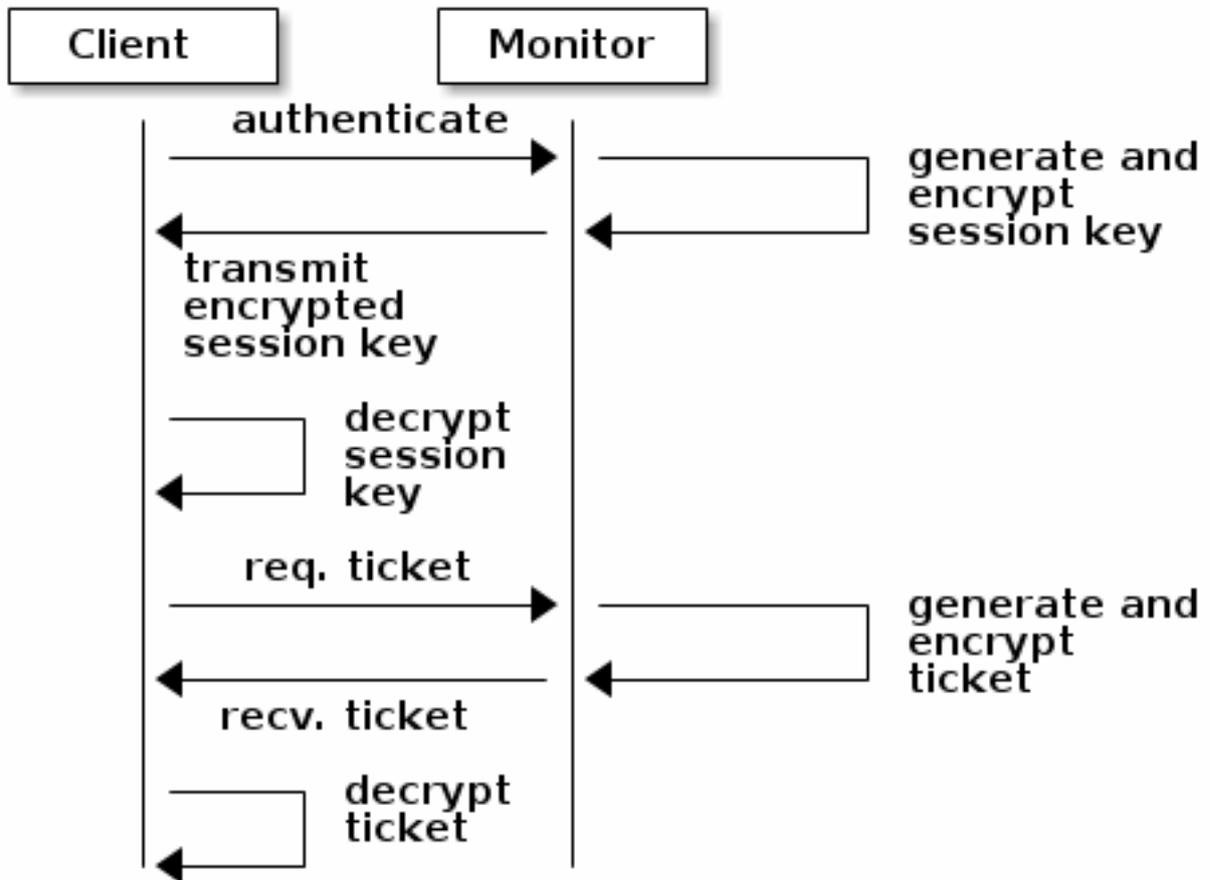


Figura 2 – Autenticação Usuário (CEPH, a)

uma chave de sessão criptografada associada com a senha do cliente, e envia ao usuário. O usuário decifra a chave, com isso, identifica o usuário atual sessão conforme a figura 2.

Temos outros componentes do Ceph, que seria o Ceph OSD responsável por armazenar objetos no sistema de arquivos local e fornecer acesso a eles através da rede (CEPH, c). E o Ceph MDS armazena os metadados dos arquivos, podendo assim operar um objeto como um sistema de arquivos e, por consequência, via MDS gerenciar os metadados (propriedade, permissões, etc.).(CEPH, b).

4.0.6 GlusterFS

Esse sistema tem abordagem em camadas, onde os recursos são adicionados e removidos de acordo com o usuário. Tem compatibilidade com EXT3, EXT4 e XFS, para armazenar os dados e pode-se chegar na escala de petabytes de armazenamento. (GLUSTER,)

Conforme a Figura 3, podemos observar a arquitetura do GlusterFS, onde "distri-

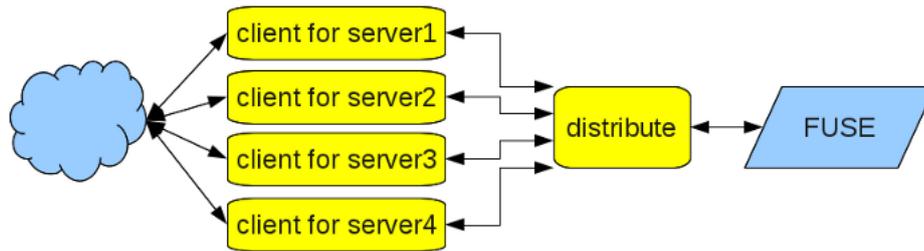


Figura 3 – Arquitetura GlusterFS (GLUSTER,)

bute", recebe uma lista com os sub-volumes e distribui os arquivos entre eles, tornando volume único de armazenamento maior de uma série de outros menores. Cada arquivo poderá ser escrito em servidores diferentes, exemplo: arquivo1.txt no servidor 3 e arquivo2.txt no servidor 1. O hash sabe em qual servidor está o arquivo pelo nome, caso tenha mudança no nome do arquivo, um arquivo ponteiro é escrito para o hash saber a localização. (GLUSTER,)

Alguns componentes principais GlusterFS são:

- Bricks (tijolos): é um diretório no sistema de arquivos que foi atribuído a um volume.
- Réplicas de dados : a réplica é usada visualizando uma redundância de armazenagem e disponibilidade dos dados. Os dados são distribuídos por diferentes nós, o usuário por fim, não sabe em qual servidor está seus dados.
- Stripe: é recomendável que se utilize volumes distribuídos apenas em ambientes que se tem uma alta concorrência para acessar arquivos muito grandes. O usuário pode configurar o número de stripes, diferente do Ceph, porém esse número tem que ser múltiplos dos bricks.
- Filesystem in Userspace (FUSE): fica encarregado de permitir o usuário a criação de um sistema de arquivos.

Segundo (DOCS,), existe a possibilidade no GlusterFS de criar diferentes tipos de volumes, como:

- Distribuídos: distribui arquivos pelo bricks no volume. Utilizado onde a prioridade é escalar o armazenamento, com a redundância sendo fornecida por outras camadas de hardware ou software.
- Replicados: replica o bricks no volume. Cenário indicado: onde se precisa de alta disponibilidade e confiabilidade.
- Stripe distribuídos: dados que estão em dois ou mais nós no cluster. Usado quando se tem prioridade na escala de armazenamento com ambiente de alta concorrência.

- Distribuídos replicados: distribui arquivos entre bricks replicado no volume. Indicado para armazenamento e alta confiabilidade.
- Stripe replicado: volume de dados em stripes em todo brick replicado no cluster. Indicado em ambientes com alta concorrência e acessos paralelos em arquivos grandes.
- Dispersos: fornece proteção de espaço evitando falhas no disco ou servidor. Armazena fragmento codificado em cada brick, com isso, para recuperar o arquivo original, precisa do subconjunto de fragmento.
- Distribuídos dispersos: distribui arquivos nos sub volumes. Se tem as mesmas vantagens em distribuir volumes repetidos.

Vantagem em se utilizar o GlusterFS:

- Podem aumentar o volume de dados adicionando um novo servidor. ([GLUSTER](#),)

Desvantagens em se utilizar o GlusterFS:

- Em caso de perda utilizando um único servidor, você acaba perdendo o acesso aos arquivos hospedados nele.
- Ao tentar escrever um arquivo maior que o sub-volume, irá ter falha. ([GLUSTER](#),)
- Se o arquivo não estiver onde o código hash calcula, a operação deve ser reiniciada, gerando um pouco de atraso. ([GLUSTER](#),)

4.0.7 Lustre

Lustre é um sistema de arquivos paralelo open-source, que não foi baseado em nenhum sistema anterior, com compatibilidade nos formatos de arquivos como EXT4, XFS e BTRFS. Tem o principal objetivo alto desempenho e disponibilidade dos dados. ([LUSTRE](#),). Possui algumas componentes principais, como: o Metadados Target (MDT), Metadata Server (MDS), Objeto de Armazenamento de Destino (OST), Servidores de Armazenamento de Objetos (OSS). Foi utilizado as informações de ([ORACLE](#); [INTEL](#),) para explicar cada item citado a baixo.

- Metadata Server (MDS) : Cada MDS fornece nomes e diretórios para o sistema de arquivos e fornece a solicitação na rede para manipular dados.
- Metadados Target (MDT): Cada sistema de arquivo possui uma MDT, que são os nomes de arquivos, diretórios, permissões e o layout de arquivo, ligado a uma MDS. Um MDT pode estar disponível para vários MDS, porém somente um acessa por

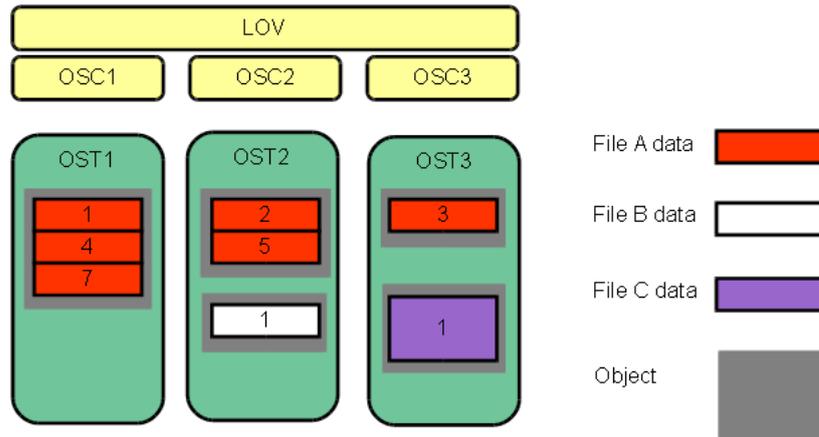


Figura 4 – Striping Lustre (ORACLE; INTEL,)

vez o MDT. Caso uma MDS falhar, a MDS esperar poder servir o MDT e tornar-lo disponível novamente aos clientes.

- Objeto de armazenamento de destino (OST): os dados de cada usuário é armazenado em um ou mais objetos, porém cada objeto possui um OST separado do sistema de arquivos. O número desses objetos por arquivos podem ser configurados, para gerar uma otimização da carga de trabalho.
- Striping: Lustre tem a capacidade de distribuir dados de múltiplas stripes, que seria diferentes pedaços de dados de um arquivo podem ser armazenados em diferentes servidores. Os usuários podem configurar para cada arquivo o número de stripes, tamanho da stripe e OSTs que irão utilizar. Como podemos ver na figura 4:

Cada objeto contém um pedaço de arquivos diferentes, quando um bloco de dados vai ser escrito em um objeto e excedo o tamanho do stripe, ele é armazenado no próximo objeto. O valor padrão da stripe é de 1MB.

Esse procedimento pode melhorar o desempenho, quando a largura de banda agregada a um único arquivo excede a largura de banda de um único OST ou quando um OST não tem espaço livre para armazenar um arquivo inteiro.

- Servidores de armazenamento de objetos (OSS): Oferece o serviço de I/O e tratamento de solicitações na rede de um OST. Podendo servir entre dois e oito OST'S com até 16TB cada. (ORACLE; INTEL,)
- Meta Data Servers: Nele ficam armazenados os dados mais críticos do sistemas de arquivos (árvore de diretórios, gerência de distribuição, alocação e disponibilidade dos OST's) e se houver alguma alteração nos dados, ele realiza o registro das alterações. (OLIVEIRA,)

- Logical Object Volume (LOV) : Para que o cliente consiga visualizar um cluster Lustre localmente, ele precisa receber um código do "LOV". LOV fica responsável por identificar qual MDS ativa. ([OLIVEIRA](#),)

O cliente irá enxergar um único volume de dados, porém sincronizado com o servidor. Vários clientes podem fazer alteração no mesmo arquivo ao mesmo tempo e outros clientes podem ler esse mesmo arquivo. ([ORACLE](#); [INTEL](#),)

5 Cenário de testes

O campus IFSC de São José tem uma rede de médio porte e sua rede externa faz parte da rede nacional de pesquisa (RNP). Com links de 1Gb/s com diversas instituições do sul do país. Na rede interna, o IFSC possui um CPD, onde ficam localizados os equipamentos de redes do campus, como servidores e switch.

O modelo de teste do sistema distribuído irá ser cliente/servidor, ou seja, o cliente irá fazer requisição de serviços ao servidor, que por sua vez irá responder com o dado solicitado, o que acaba gerando mensagens simples de requisição/resposta entre os mesmos. O tempo de espera pela requisição do dado pode ser configurado, portanto, caso o tempo expire, considera-se que a resposta foi perdida e começa um novo processo com a resposta.

5.0.8 Equipamentos de rede

A rede do IFSC de São José, conta com um switch Gigabit, da marca Cisco e modelo Catalyst 2960S. Os três computadores usados para simular os servidores (que é o mínimo recomendável para se ter um sistemas de arquivos distribuído) tem as seguintes configurações:

- Computador 1: Processador: I5-4590 de 3.3 GHz.
Memória RAM: 16GB.
HD: 1TB.
Sistema operacional: Linux.
- Computador 2: Processador: I5-4590 de 3.3 GHz.
Memória RAM: 16GB.
HD: 1TB.
Sistema operacional: Linux.
- Computador 3: Processador: I5-4590 de 3.3 GHz.
Memória RAM: 8GB.
HD: 500GB.
Sistema operacional: Linux.

6 Metodologia de testes

Para verificar qual a melhor combinação de software com formato de arquivos, vou realizar os seguintes testes por trinta vezes, que é a quantidade que o padrão de erro mantém um linear:

- Leitura de arquivos no cliente: se baseia no tempo em que um cliente demora para realizar a leitura de um arquivo, que está localizado no servidor.
- Escrita de arquivos no servidor: verifica-se o tempo em que o cliente consegue inserir ou modificar um arquivo no servidor.
- Uso de processador: certifica o uso do processador no cliente e no servidor.
- Uso de memória: apura o uso da memória no cliente e servidor.
- Uso do tráfego de rede: verifica para qual utiliza-se menos a rede.
- Teste de transição de máquinas: caso algum servidor caia durante uma leitura de arquivo, (qual o tempo e precisão para o sistema de arquivos) busca-se esse arquivo em outro servidor.
- Teste de conexões simultâneas: analisa quantos clientes o servidor pode suportar.
- Variação de tamanho do arquivo: apura se o tamanho dos arquivos mudam, com os formatos de arquivos diferentes.
- Teste de cópia de arquivo (backup/snapshot).

Referências

- ABREU, E. *Anatomia dos Sistemas de Arquivos de journaling do Linux*. [s.n.]. [Online; acessado 18-Dezembro-2015]. Disponível em: <<https://www.vivaolinux.com.br/artigo/Sistemas-de-arquivos-para-GNU-Linux?pagina=5>>. Citado na página 24.
- ANDRÉ, C. U. F. S. *Sistemas Distribuídos*. [s.n.]. [Online; acessado 18-Dezembro-2015]. Disponível em: <<http://www3.fsa.br/localuser/sdtr/conceitos/Sistemas%20Distribuidos%20Vantagens%20e%20Desvantagens.pdf>>. Citado na página 22.
- BTRFS. *BTRFS*. [s.n.]. Disponível em: <https://btrfs.wiki.kernel.org/index.php/Main_Page>. Citado na página 24.
- CARLOS, R. L.; TAVARES, A. *Sistema de arquivos Distribuídos*. [s.n.]. [Online; acessado 10-Outubro-2015]. Disponível em: <<http://www.lrodrigo.com.br/wp/wp-content/uploads/2013/08/arquitetura-de-sistemas.pdf>>. Citado na página 21.
- CEPH. *Architecture*. [s.n.]. [Online; acessado 04-Fevereiro-2016]. Disponível em: <<http://docs.ceph.com/docs/master/architecture/>>. Citado 2 vezes nas páginas 27 e 28.
- CEPH. *CEPH-MDS – CEPH METADATA SERVER DAEMON*. [S.l.: s.n.]. Citado na página 28.
- CEPH. *ceph-osd – ceph object storage daemon*. [S.l.: s.n.]. Citado na página 28.
- COULOURIS JEAN DOLLIMORE, T. K. G. B. G. *SISTEMAS DISTRIBUÍDOS - Conceitos e projetos*. [s.n.]. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=6WU3AgAAQBAJ&oi=fnd&pg=PR3&dq=sistemas+distribuidos&ots=Fg92cFYQnY&sig=yBXAVRVvAPU6D4UxK-bLjp82WfA#v=onepage&q&f=false¬e=>>>. Citado na página 21.
- DOCS, G. *Setting up GlusterFS Server Volumes*. [S.l.: s.n.]. Citado na página 29.
- GLUSTER. *GlusterFS Concepts*. [s.n.]. [Online; acessado 04-Fevereiro-2016]. Disponível em: <http://www.gluster.org/community/documentation/index.php/GlusterFS_Concepts>. Citado 3 vezes nas páginas 28, 29 e 30.
- LAIRD, C. *Migrando para a Ext4*. [s.n.]. Disponível em: <<https://www.ibm.com/developerworks/br/library/wa-ltwebserv/>>. Citado na página 23.
- LINUX, A. *XFS*. [S.l.: s.n.]. Citado na página 24.
- LUSTRE. *Welcome to the official home of the Lustre filesystem*. [S.l.: s.n.]. Citado na página 30.
- OLIVEIRA, E. *Sistemas de Arquivos de Uso Especial*. [S.l.: s.n.]. Citado 2 vezes nas páginas 31 e 32.
- ORACLE; INTEL. *Lustre* Software Release 2.x*. [s.n.]. [Online; acessado 11-Fevereiro-2016]. Disponível em: <https://build.hpdd.intel.com/job/lustre-manual/lastSuccessfulBuild/artifact/lustre_manual.xhtml>. Citado 3 vezes nas páginas 30, 31 e 32.

PEREIRA, D. M. *Filesystem (Sistemas de Arquivos): Várias siglas, um único resumo*. [s.n.]. Disponível em: <<https://x22univale.wordpress.com/tag/xf/>>. Citado na página 24.

RUSSO, R. *Btrfs – O sistema de arquivos da próxima geração GNU/Linux*. [S.l.: s.n.]. Citado na página 24.