

Marcus Vinícius Florêncio e Souza

*Abordagem para avaliação de desempenho para o armazenamento
de informações médicas em sistemas de arquivos distribuídos*

Florianópolis

2010

Marcus Vinícius Florêncio e Souza

Abordagem para avaliação de desempenho para o armazenamento de informações médicas em sistemas de arquivos distribuídos

Monografia submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciências da Computação.

Orientador: Prof. Dr. rer. nat. Aldo von Wangenheim

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Florianópolis

2010

Monografia de graduação sob o título “Abordagem para avaliação de desempenho para o armazenamento de informações médicas em sistemas de arquivos distribuídos”, defendida pelo acadêmico Marcus Vinícius Florêncio e Souza e aprovada em ____/____/2010, em Florianópolis, Santa Catarina, pela banca examinadora constituída pelos professores:

Prof. Dr. rer. nat. Aldo von Wangenheim
Universidade Federal de Santa Catarina
Orientador

MSc. Douglas Dyllon J. de Macedo
Universidade Federal de Santa Catarina
Co-Orientador

Prof. Dr. Mario Antonio Ribeiro Dantas
Universidade Federal de Santa Catarina
Membro da Banca

Agradecimentos

Agradeço a Deus, por estar sempre presente nos momentos em que necessito fortalecendo-me para a realização de meus objetivos.

A minha família, por estar sempre ao meu lado oferecendo de maneira incondicional amor, respeito, carinho e por ser um grande exemplo de superação. Por acreditar e me guiar por caminhos que sozinho não seria capaz de andar.

A Claudia Alice Goecks, pelo amor, paciência, carinho, cumplicidade e tudo o que se pode desejar de uma companheira durante a estrada da vida. Que a estrada seja longa e cheguemos ao final dela juntos.

Ao meu orientador Prof. Dr. rer.nat. Aldo von Wangenheim, pela oportunidade de realização deste projeto, pela paciência, disponibilidade e orientação para o desenvolvimento de um bom trabalho.

Ao meu co-orientador MSc. Douglas Dyllon J. de Macedo, por todo o tempo disponibilizado, pelo conhecimento compartilhado, pela paciência nos momentos de nervosismo, pela motivação nas horas difíceis e acima de tudo pela amizade.

Ao Tiago Coelho Prado pelo companheirismo durante algumas noites sem dormir e pela obstinação em ajudar no projeto de forma direta e eficiente. Sem dúvida um grande amigo.

A todos aqueles que contribuíram e torceram pelo sucesso desta pesquisa, meus sinceros agradecimentos.

Muito Obrigado!

Sumário

Sumário.....	6
Lista de Figuras	8
Lista de Tabelas	9
Resumo.....	10
Abstract	11
1. Introdução.....	12
1.1 Pergunta de Pesquisa.....	12
1.2 Cenário de Aplicação	13
1.3 Objetivos.....	13
1.3.1 Objetivo Geral	13
1.3.2 Objetivos Específicos.....	14
1.4 Metodologia.....	14
1.5 Estrutura do Trabalho	14
2 Conceitos Fundamentais	15
2.1 Telemedicina.....	15
2.1.1 Benefícios do uso da Telemedicina	16
2.1.2 Telemedicina no Brasil e em Santa Catarina	17
2.2 DICOM.....	18
2.3 Desempenho.....	20
2.4 Sistemas de Arquivos Distribuídos.....	21
2.4.1 Requisitos de um sistema de arquivos distribuído	22
2.5 Sistemas de Arquivos Paralelos	23
2.5.1 Sistema de arquivos Lustre.....	24
2.5.2 dNFSp (Distributed Network File System Parallel).....	25
2.5.3 FhGFS (Fraunhofer File System).....	27
2.5.4 Sistema de arquivos distribuído Hadoop (HDFS)	28
2.5.5 PVFS (Parallel Virtual File System)	30
3 Revisão do Estado da Arte e Prática.....	31
3.1 Avaliação de desempenho.....	31
3.2 Técnica / Modelos para prover melhor desempenho de armazenamento utilizando sistemas de arquivos distribuídos.....	30
3.3 Estudos Comparativos	33
3.4 Análise crítica dos trabalhos	36
4 Ambientes	41
4.1 Parallel Virtual File System (PVFS)	42
4.2 Fraunhofer File System (FhGFS).....	42
4.3 Hadoop Distributed File System (HDFS)	42
4.4 Sistema de Arquivos Lustre.....	43
4.5 Distributed Network File System Parallel (dNFSp)	44
5 Resultados	45
5.1 Resultados para o Experimento com Uma Imagem	47
5.2 Resultados para o Experimento com Dez Imagens	48

5.3	Resultados para o Experimento com Cem Imagens	49
5.4	Resultados para o Experimento com Mil Imagens	50
5.5	Resultados para o Experimento com Dez Mil Imagens	51
6	Conclusões e Trabalhos Futuros.....	54
	Referências	55

Lista de Figuras

Figura 1 - Visualização da Rede Catarinense de Telemedicina (RCTM).....	18
Figura 2- Arquitetura do Sistema de Arquivos Lustre.....	25
Figura 3 - Arquitetura do Sistema de Arquivos NFSp.....	26
Figura 4 - Arquitetura do Sistema de Arquivos dNFSp.....	27
Figura 5 - Arquitetura do Sistema de Arquivo Fraunhofer.....	28
Figura 6 - Arquitetura do Hadoop Distributed File System.....	29
Figura 7 - Teste SFWA que varia o tamanho de bloco da requisições.....	33
Figura 8 - Teste SFWA variando o número de clientes envolvidos.....	33
Figura 9 - Teste MFWA variando o número de clientes envolvidos.....	33
Figura 10 - Teste de operações de leitura MFWA e SFWA variando número de clientes envolvidos.....	33
Figura 11 - Throughput de leitura agregado obtido com o protótipo, em comparação com o modelo original.....	32
Figura 12 - (a) Velocidade do PVFS2 e Ext3 e (b) ganho do PVFS2 sobre o Ext3 ao aumentar a concorrência no acesso, usando discos na velocidade 2,5MB/s.....	34
Figura 13 - (a) Velocidade do PVFS2 e do Ext3 e (b) ganho do PVFS2 sobre o Ext3 ao aumentarmos a velocidade dos discos, usando 32 threads.....	35
Figura 14 - Velocidade do PVFS2 e do Ext3 em (a) um cliente com processador Pentium 4 2,8GHz e em (b) um cliente com processador Pentium 4 1,8GHz, ambos em rede Gigabit.....	35
Figura 15 - Gráfico de Desempenho de Armazenamento.....	46
Figura 16 - Gráfico Comparativo de Desempenho de Armazenamento (Uma Imagem).	47
Figura 17 - Gráfico Comparativo das Métricas Obtidas (Uma Imagem).	47
Figura 18 - Gráfico Comparativo de Desempenho de Armazenamento (Dez Imagens).	48
Figura 19 - Gráfico Comparativo das Métricas Obtidas (Dez Imagens).	48
Figura 20 - Gráfico Comparativo de Desempenho de Armazenamento (Cem Imagens).	49
Figura 21 - Gráfico Comparativo das Métricas Obtidas (Cem Imagens).	49
Figura 22 - Distribuição dos Tempos de Armazenamento.	50
Figura 23 - Gráfico Comparativo de Desempenho de Armazenamento (Mil Imagens).	50
Figura 24 - Gráfico Comparativo das Métricas Obtidas (Mil Imagens).....	51
Figura 25 - Comparativo de Desempenho de Armazenamento (Dez mil Imagens).....	51
Figura 26 - Gráfico Comparativo das Métricas Obtidas (Dez mil Imagens).....	52
Figura 27 - Desempenho Individual do PVFS para cada classe de experimento.	52
Figura 28 - Desempenho Individual do FhGFS para cada classe de experimento.	53

Lista de Tabelas

Tabela 1 - Apresentação de Benefícios do uso da Telemedicina em pacientes e em instituições.....	17
Tabela 2 - Resumo dos trabalhos citados.....	38
Tabela 3 - Especificações das Máquinas do Ambiente.....	41
Tabela 4 - Sistemas Operacionais das Máquinas do Ambiente.....	41
Tabela 5 - Tempos de Armazenamento.....	46
Tabela 6 - Ganho de Desempenho do FhGFS em relação ao PVFS.....	52

Resumo

Com as constantes evoluções na velocidade de processamento dos computadores e nas taxas de transmissões de dados em rede, as aplicações geram um volume cada vez maior de dados. Um exemplo real para esse tipo de situação é a Rede Catarinense de Telemedicina que gera conjuntos de dados na ordem de terabytes que necessitam ser armazenados e recuperados em tempo hábil.

É com o intuito de prover melhoria no desempenho de armazenamento desses dados que é apresentada uma abordagem para avaliação de desempenho no armazenamento de informações médicas utilizando cinco sistemas de arquivos distribuídos distintos (Lustre, dNFSp, FhGFS, HDFS e PVFS) .

Para realização da avaliação foi configurado um aglomerado computacional com o número de clientes fixo e, para cada uma das ferramentas abordadas, foram realizados experimentos caracterizados pelo armazenamento de conjuntos de imagens distintos e para cada experimento foram colhidas métricas referentes ao: tempo máximo, tempo mínimo, tempo total, média e desvio padrão. Por fim, foram realizados estudos comparativos entre essas métricas para avaliar qual ferramenta obteve melhor desempenho para determinado conjunto de dados, assim como qual obteve melhor desempenho médio.

Fazendo uma avaliação e sintetização desses resultados é possível analisar os ganhos e vantagens na implantação de um ambiente desse tipo.

Palavras-chave: Desempenho, Armazenamento, Sistemas de Arquivos Distribuídos, Telemedicina.

Abstract

With the constant changes in processing speed of computers and the rates of data transmission network, the applications generate an increasing volume of data. A real example for this type of situation is the Santa Catarina Telemedicine Network that generates data sets in the order of terabytes that need to be stored and retrieved in a timely manner.

It is aiming to provide improved performance data storage that is presented an approach for evaluating performance of storing medical information using five different distributed file systems (Lustre, dNFSp, FhGFS, HDFS and PVFS).

To perform the evaluation was set up a computational cluster with a fixed number of clients and, for each of the tools discussed, experiments were performed characterized by storing different sets of images and for each experiment were collected for the metrics: maximum time, minimum time , total time, average and standard deviation. Finally, studies comparing these metrics to evaluate which tool showed the best performance for a given data set, and which had the best average performance.

Making an assessment and synthesis of these results is possible to analyze the gains and advantages in implementing such an environment.

Keywords: Performance, Storage, Distributed File Systems, Telemedicine.

1. Introdução

Com as constantes evoluções na velocidade de processamento dos computadores e nas taxas de transmissões de dados em rede, as aplicações geram um volume cada vez maior de dados. Como se isso não fosse suficiente a evolução nos componentes físicos que permitem o armazenamento dessas informações não segue essa mesma proporção gerando gargalos e queda de desempenho para o armazenamento das mesmas. Um exemplo real para esse tipo de situação é a Rede Catarinense de Telemedicina (RCTM) que gera conjuntos de dados na ordem de terabytes que são armazenados em servidores localizados no Hospital Universitário em Florianópolis.

Para resolver esse tipo de situação, os sistemas de arquivos distribuídos aparecem como uma solução natural para prover melhoria no desempenho de armazenamento e recuperação dessas informações. Devido à considerável quantidade de sistemas de arquivos distribuídos existentes, torna-se interessante avaliar o desempenho de algumas dessas ferramentas para que possa ser analisado qual é mais vantajosa para esse cenário e também para que seja obtido um conjunto de métricas que forneçam informações que possam auxiliar na escolha de qual ferramenta é mais consistente e coerente para determinada abordagem dentro de cenários distintos do aqui proposto.

1.1 Pergunta de Pesquisa

Levando em consideração a grande quantidade de informações médicas providas da Rede Catarinense de Telemedicina e a necessidade de armazenar e recuperar essas informações em tempo hábil, é necessário utilizar ferramentas que forneçam melhor desempenho para esse tipo de aplicação. Com a utilização de sistemas de arquivos distribuídos é possível gerar um conjunto de métricas para avaliar o desempenho no armazenamento desses dados. Fazendo uma avaliação e sintetização desses resultados poderá ser analisado os ganhos e vantagens na implantação de um ambiente desse tipo.

Neste trabalho de conclusão de curso a principal questão a ser investigada é: Dado conjuntos com quantidades diferenciadas de informações médicas, qual sistema de arquivos distribuído oferece melhor desempenho no armazenamento desses dados?

1.2 Cenário de Aplicação

A Rede Catarinense de Telemedicina (RCTM) foi desenvolvida pelo grupo de pesquisas *Cyclops Group* com o apoio da Secretaria Estadual da Saúde com a finalidade de auxiliar o atendimento médico a localidades mais distantes descongestionando o atual sistema concentrado de atendimento apresentado pelo sistema público de saúde.

Atualmente a RCTM conta com 259 municípios enviando diversos tipos de exames, entre eles: Eletrocardiogramas, Tomografia Computadorizada, Ressonância Magnética, Raios-X, entre outros. Esses exames atualmente geram dados na ordem de terabytes que precisam ser armazenados e recuperados de forma segura, confiável e com alto desempenho para prover os dados e exames dos usuários no momento que forem solicitados.

Uma das formas estudadas para suportar essa crescente taxa de armazenamento de dados é dada através de recursos de sistemas de arquivos distribuídos. Eles têm sido usados como camada básica para sistemas e aplicações distribuídas, pois permitem que vários processos compartilhem dados por longos períodos, de modo seguro e confiável (TANENBAUM; STEEN, 2002).

É com o intuito de prover melhoria no desempenho de armazenamento dessas informações médicas que é apresentado neste trabalho uma abordagem para avaliação de desempenho no armazenamento de informações médicas utilizando cinco sistemas de arquivos distribuídos (Lustre, dNFSp, FhGFS, HDFS e PVFS) para que possa ser analisada a oportunidade de implantação e vantagens apresentadas pelos mesmos neste cenário.

1.3 Objetivos

1.3.1 Objetivo Geral

Avaliar o desempenho de armazenamento de um grande volume de informações médicas, providas da Rede Catarinense de Telemedicina em diferentes sistemas de arquivos distribuídos, bem como comparar os resultados obtidos para que possa ser feita uma análise do desempenho das mesmas.

1.3.2 Objetivos Específicos

- Levantar o estado da arte relacionado ao armazenamento em sistemas de arquivos distribuídos;
- Desenvolver uma metodologia para avaliação de desempenho de sistemas de arquivos distribuídos utilizando informações médicas;
- Estruturar ambientes computacionais baseados em sistemas de arquivos distribuídos para aplicar a metodologia proposta;
- Implantar um conjunto de procedimentos para sintetização dos resultados visando à avaliação da metodologia proposta.

1.4 Metodologia

Este estudo será realizado sobre um aglomerado computacional, com o número de clientes fixo, e consiste em, para cada uma das ferramentas abordadas, realizar 25 repetições para o experimento caracterizado pelo armazenamento de conjuntos de imagens distintos: 1, 10, 100, 1000 e 10000. Para cada repetição do experimento serão colhidas métricas referentes ao: tempo máximo, tempo mínimo, tempo total, média e desvio padrão. Por fim serão realizados estudos comparativos entre essas métricas para avaliar qual ferramenta obteve melhor desempenho para determinado conjunto de dados, assim como qual obteve melhor desempenho médio.

Com esses dados em mãos será possível avaliar e justificar as vantagens e desvantagens na implantação desta proposta para a RCTM.

1.5 Estrutura do Trabalho

Este trabalho é dividido em seis etapas principais. No segundo capítulo são apresentados os conceitos fundamentais para o entendimento da proposta abordada, no terceiro capítulo é feito o levantamento do estado da arte referenciando trabalhos relacionados à avaliação de desempenho e armazenamento de dados em sistemas de arquivos distribuídos. Na quarta parte é feita uma descrição sobre o ambiente utilizado. No quinto capítulo são apresentados os resultados experimentais comparando as cinco ferramentas utilizadas e, finalizando, é efetuada a avaliação dos resultados junto com propostas e expectativas para trabalhos futuros no sexto capítulo.

2 Conceitos Fundamentais

2.1 Telemedicina

Anteriormente ao conceito de telemedicina é interessante avaliar a origem desse termo. Telemedicina é um termo híbrido, ou seja, ele é constituído por elementos de diferentes línguas. Em um dos principais dicionários de sinonímia da língua portuguesa – Michaelis e em um dicionário de etimologia – Dicionário Etimológico Nova Fronteira da Língua Portuguesa – *Tele* é parte do termo telemedicina, de origem grega, a qual é atribuída o significado de longe, ao longe, longe de. Já a parte *medicina*, é de origem latina e significa arte e ciência de curar e prevenir as doenças. Apesar de genéricos, os significados apresentados nesses dicionários de sinonímia e de etimologia possibilitam identificar duas características comuns presentes nos conceitos de telemedicina, que são: curar, prevenir doenças (medicina); à distância (tele).

É possível perceber a existência de diferentes conceitos para telemedicina. Conforme apresenta De Faria (2003), o termo telemedicina foi inicialmente utilizado por Thomas Bird, em 1975, que definia a telemedicina como uma intervenção médica por meios de sistemas de comunicação áudio-visual interativos. Na década de 80, Conrath apresentou uma definição mais ampla para o termo telemedicina. Para Conrath, a telemedicina poderia ser definida como o uso de tecnologias de comunicação com a finalidade de auxiliar práticas médicas. Já na década atual (2010), a telemedicina é definida pela ATA - American Telemedicine Association, como o acesso a informações médicas em diferentes locais geográficos, por meio de sistemas de comunicação eletrônica, com o objetivo de aperfeiçoar a assistência médica à pacientes e aperfeiçoar o ensino de profissionais da área da saúde. É perceptível que o conceito de telemedicina foi sendo modificado no decorrer das décadas e essas mudanças podem ser decorrentes tanto de alterações nos sistemas de comunicação utilizados, os quais são constantemente desenvolvidos e modificados, como também de mudanças ocorridas nas finalidades e possibilidades de uso da telemedicina.

2.1.1 Benefícios do uso da Telemedicina

A utilização da telemedicina repercute em benefícios diversos, tanto para pacientes como também para as instituições que a utilizam. Em diferentes obras, (De Faria, 2003; Ferreira, 2005; De Macedo, 2008; Bonho, 2006; Savaris et al, 0000;) os autores apresentam benefícios provindos da utilização da telemedicina, que foram subdivididos em benefícios aos pacientes e benefícios às instituições que a utilizam e que estão apresentados na tabela 1.

Benefícios aos pacientes:	Benefícios às instituições:
<ul style="list-style-type: none">- reduzir custos aos pacientes com transportes para atendimentos em instituições melhor especializadas;- maior conforto a pacientes, sem necessitar de deslocamentos para outras instituições melhor especializadas, assim como, possibilita o monitoramento doméstico de pessoas idosas ou em pós-operatório;- acesso à saúde para pessoas e comunidades isoladas ou desprovidas de profissionais qualificados;- redução do risco de infecção hospitalar aos pacientes;- melhora do atendimento aos pacientes: mais rapidez no diagnóstico de pacientes; possibilidade de gerar um histórico do paciente com informações sobre ele;- reduzir o tempo de espera por resultados de exames, diagnósticos, tratamentos e procedimentos médicos;- diminuir a replicação desnecessária de exames;- diminuir o tempo necessário entre o processo de diagnóstico e o início da	<ul style="list-style-type: none">- reduzir custos com atendimentos: diminuição dos custos com a locomoção de profissionais da saúde devido à possibilidade de discussão, à distância, de exames e do acompanhamento de exames por videoconferência; diminuição dos custos com o transporte de pacientes para instituições melhor especializadas a certos tipos de doenças;- agilizar trocas de informações ao facilitar as discussões de casos entre médicos e especialistas por meio de encontros virtuais entre entidades separadas geograficamente;- gerar laudos virtuais e realizar teleconferências entre dois ou mais médicos com o objetivo de analisar casos médicos complicados;- permitir melhor controle da quantidade e origem dos exames;- possibilitar aumento no número de vagas em instituições (hospitais, clínicas,...), já que a internação para investigação diagnóstica ou tratamento de doenças crônicas torna-se desnecessária;- reciclar e aperfeiçoar profissionais à distância: médicos e demais profissionais da área da saúde podem discutir casos complicados, solucionar dúvidas e conduzir

<p>terapia, o que aumenta a possibilidade de sucesso em tratamentos.</p>	<p>treinamentos à distância, o que possibilita que profissionais localizados em locais de difícil acesso ou distantes de grandes centros podem atualizar-se frequentemente.</p>
--	---

Tabela 1 - Apresentação de Benefícios do uso da Telemedicina em pacientes e em instituições

2.1.2 Telemedicina no Brasil e em Santa Catarina

Segundo De Faria (2003), o início dos processos de implantação da telemedicina no Brasil está relacionado às redes metropolitanas de alta velocidade baseadas em tecnologias que provêm conexões de alta velocidade e oferecem suporte à qualidade de serviço. Na área metropolitana de São Paulo, instituições de pesquisa, apoiadas pelo CNPq, tem desenvolvido a Rede Metropolitana de Alta Velocidade de São Paulo (REMAV-SP). A respeito dessa infraestrutura, um conjunto de aplicações em telemedicina foi desenvolvido, possibilitando a distribuição de imagens e informações médicas entre instituições da área da saúde.

A infra-estrutura implantada e o conjunto de padrões utilizados (ATM, TCP/IP e DICOM) possibilitam a distribuição de informações médicas entre os dois maiores hospitais universitários públicos na área metropolitana de São Paulo. Os recursos oferecidos pelo protocolo ATM, como a baixa latência e a possibilidade de estabelecer prioridade no envio de pacotes, possibilitaram a implantação de serviços de saúde à distância com alta eficiência. Além disso, um conjunto de aplicações foi desenvolvido para transmissão, armazenamento e visualização de imagens médicas, o que permite oferecer serviços de saúde ainda não disponíveis nos meios de comunicação convencionais (DE FARIA, 2003).

Atualmente, 193 municípios catarinenses pertencem à rede catarinense de telemedicina (RCTM) e 259 municípios catarinenses participam do programa Telessaúde Brasil. A rede catarinense de telemedicina (RCTM) é resultado de uma parceria entre a Universidade Federal de Santa Catarina (UFSC), por meio do grupo Cyclops, e a Secretaria de Estado da Saúde de Santa Catarina. Por meio dessa parceria, serviços são oferecidos com o objetivo de coletar, armazenar e disponibilizar dados referentes a exames provenientes de instituições de saúde distribuídas por todo o estado (SAVARIS ET AL, 0000)

Segundo De Macedo (2008), o portal catarinense de telemedicina é um sistema baseado na web, construído na linguagem de programação PHP, sendo que seus dados são armazenados em um sistema gerenciador de bancos de dados relacionais PostgreSQL. No

modo de operação do portal, os exames realizados nos municípios são automaticamente enviados ao banco de dados central, localizado nas dependências do hospital universitário (HU). A partir desse processo, um médico especialista pode acessar as informações de qualquer local por meio da internet e pode, com isso, laudar exames, discutir casos com outros médicos dentre outras operações.

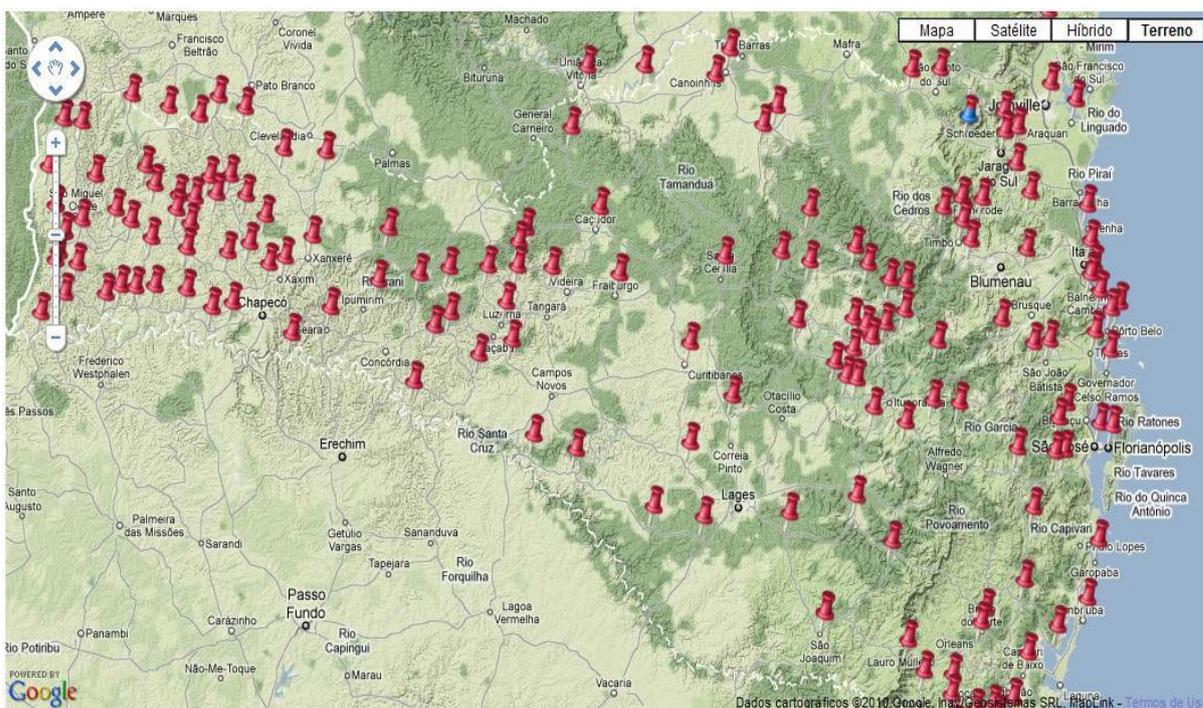


Figura 1 - Visualização da Rede Catarinense de Telemedicina (RCTM).

2.2 DICOM

Na década de 70, surge uma modalidade de equipamentos para diagnósticos médicos, denominada de Tomografia Computadorizada (Computed Tomography). Com o surgimento da Tomografia Computadorizada tornou-se possível arquivar e transmitir imagens médicas digitalizadas em hospitais e demais instituições afins (DE MACEDO, 2008).

Após o surgimento da Tomografia Computadorizada, surgiram outras modalidades de diagnóstico por imagens digitais, as quais eram elaboradas de maneiras distintas por diferentes fabricantes. Pelo fato de cada fabricante desenvolver distintamente a solução para o arquivamento, impressão, transmissão, visualização de imagens digitais e informações de cada modalidade de exame, muitas vezes, era impossível integrar esses equipamentos diversos (DELLNI, 2005).

Como forma de solucionar esse problema, foi publicado em 1985 um primeiro sistema padrão para armazenamento e transmissão de dados e informações médicas, desenvolvido a partir de uma ação integrada entre a Faculdade Americana de Radiologia (American College of Radiology - ACR) com a Associação Nacional de Fabricantes de Equipamentos Elétricos - NEMA (National Electrical Manufacturers Association). Essa primeira versão desse sistema padrão foi denominada de ACR- NEMA Standards Publication No. 300-1985. Após essa primeira versão ocorreram, em 1986 e 1988, duas revisões desse sistema e em 1993 foi lançada a ACR-NEMA Standards Publication PS3, também chamada de DICOM (Digital Imaging and Communications in Medicine) (MILDENBERGER; EICHELBERG; MARTIN, 2002).

O DICOM é um padrão mais estável e mais completo do que os primeiros padrões. Além disso, ele dispõe de um protocolo para transmissão de informações mais moderno do que os padrões anteriores. A estrutura de dados do DICOM é baseada em um modelo que possui um único identificador para objetos (imagens; dados de pacientes; laudos médicos;...) e serviços. Atualmente o DICOM é o padrão de fato para o PACS (Picture Archiving and Communications System), sendo suportado pela maioria dos aparelhos que trabalham com informações médicas digitais (MILDENBERGER; EICHELBERG; MARTIN, 2002).

Os principais componentes da estrutura de dados do padrão DICOM são os IODs (Information Object Definitions). Eles são definidos pelos dados de uma imagem e as informações relacionadas a ela. Existem ainda um cabeçalho (header) contendo uma lista de atributos descrevendo os tipos dos objetos, dados do paciente e outras informações como laudos e procedimentos já realizados. Cada atributo de um IOD tem um significado bem definido. Os dados são divididos em vários grupos (tags) e, dependendo do grupo, os detalhes técnicos são muito importantes. Para exemplificar, em um raio X, há detalhes como voltagem, disposição das imagens entre outros que são importantes nesse tipo de exame. Portanto, dependendo da modalidade, diferentes estruturas de dados são definidas (MAGNUS, 2010).

2.3 Desempenho

Desempenho pode ser definido como o conjunto de características ou capacidades de comportamento e rendimento de um indivíduo, produto, sistema, empreendimento ou processo, em especial quando comparados com metas, requisitos ou expectativas.

Quando se está interessado em avaliar o desempenho de SAP, é necessário encontrar um meio de medir as alterações em seu desempenho quando o mesmo é submetido a diferentes configurações. Isso significa que quando falamos em medir o desempenho não estamos necessariamente falando em comparar diferentes sistemas de arquivos paralelos, e sim na habilidade de avaliar as mudanças no desempenho conforme fazemos mudanças no cenário de aplicação e na configuração dos mesmos. Com essas informações é possível adaptar as aplicações para obter um máximo desempenho ou, dado um sistema de arquivos, configurá-lo para obter melhor desempenho para um conjunto de código ou dados a serem armazenados (LAYTON, 2006).

Os SAPs têm como objetivo principal aumentar o desempenho total de I/O de arquivos para aplicações. Por outro lado, simplesmente testando o sistema de arquivo para avaliar a quantidade de dados possível de ser transferida para os clientes numa determinada quantidade de tempo apenas oferece uma idéia geral do desempenho do SAP, mas não nos garante que as aplicações terão o mesmo nível de desempenho apresentado pelos benchmarks pelo fato de fatores como cache de dados e a distribuição das informações entre os nós representarem um papel crítico no desempenho de armazenamento dos grandes conjuntos de dados pelos feito pelos sistemas de arquivos paralelos (Kassick, 2008).

A cache de dados torna-se importante não apenas por evitar que repetidas requisições sejam tratadas múltiplas vezes como também por evitar que quantidades significativas de dados sejam transportadas dos clientes para os servidores. No caso da distribuição dos dados, uma vez que os que o armazenamento nos SAP é distribuído entre diversos servidores, a maneira como é feita a distribuição dos dados de uma aplicação entre eles certamente influenciará no desempenho geral do sistema I/O (KASSICK, BOITO & NAVAUX, 2008).

Técnicas de acesso à memória, técnicas de mapeamento dos arquivos, mecanismos utilizados na escrita de dados, tamanho de buffer e operações de timing são mais alguns exemplos de fatores influentes no desempenho de um sistema de arquivos paralelo (KASSICK, BOITO & NAVAUX, 2008).

2.4 Sistemas de Arquivos Distribuídos

Com a dinâmica evolução de novas tecnologias, assim como da nanotecnologia, que nos permite colocar cada vez mais transistores numa pastilha de silício, torna-se cada vez mais possível aumentar a capacidade e velocidade de processamento dos computadores, conforme percebemos na lei de Moore. Porém essa evolução vem sendo limitada devido a algumas leis da termodinâmica e o elevado custo para se fabricar novos processadores. A utilização de algoritmos otimizados aparece como alternativa para obter um aumento de desempenho, porém, na prática não é isso que percebemos nas organizações. Supercomputadores também aparecem como solução para obter ganho de desempenho, porém seus elevados custos financeiros tornam limitado o número de organizações que têm acesso a eles.

Mesmo com todas essas limitações a velocidade de acesso aos arquivos através de seus meios físicos de armazenamento não evoluíram na mesma proporção que a velocidade de processamento e taxa de transmissão de dados em redes presentes nos computadores mais modernos, transformando o desempenho de recuperação e armazenamento da robusta quantidade de informações geradas pelas aplicações num cenário de Computação de Alto Desempenho (empresas como FaceBook, Google, Twitter e Amazon armazenam hoje dados na ordem de petabytes), em um dos grandes desafios da computação científica (DE CARVALHO, 2010) É nesse âmbito que a tecnologia de sistemas distribuídos aparece como uma opção adequada para melhoria de desempenho nas aplicações.

Em sistemas distribuídos compartilhar informações e dados é crucial para sua operação, e por este fato, os sistemas de arquivos distribuídos (*Distributed File Systems*) são a base para aplicações desta natureza. Sistemas de arquivos distribuídos permitem que vários processos compartilhem dados por longos períodos de tempo, de modo seguro e confiável, e por este motivo eles têm sido usados como a camada básica para sistemas e aplicações distribuídas (TANENBAUM & VAN STEEN, 2002).

Devido à considerável quantidade de Sistemas de Arquivos Distribuídos existentes com comportamentos e características diferenciadas, dependendo das decisões de projeto tomadas durante sua construção, é necessário avaliar o desempenho dos mesmos na prática para que a partir desse conjunto de métricas seja possível avaliar qual SAD é mais apropriado para uma dada necessidade. Utilizando um conjunto de ferramentas para obter uma comparação confiável entre os resultados obtidos, é possível auxiliar os desenvolvedores a

adaptarem as operações de acesso aos dispositivos E/S para obterem melhores vantagens e como consequência melhor desempenho nas aplicações desenvolvidas.

2.4.1 Requisitos de um sistema de arquivos distribuído

Para resolver qual sistema de arquivos utilizar em um sistema distribuído, costuma-se tentar saber qual o tipo de aplicação que se utilizará desse sistema, e a partir disso tentar descobrir o que é mais importante para o sistema, como tolerância a falhas, acesso concorrente, etc. Abaixo serão descritos alguns requisitos importantes dos SAD cuja análise possa auxiliar nessa escolha.

Transparência: É um conceito relativo a tornar menos perceptível alguns detalhes que são tratados pela ferramenta. O SAD deve prover transparência nos seguintes contextos:

- **Acesso:** aplicações que acessam os arquivos do SAD não devem estar cientes da localização física deles.
- **Localização:** Todas as aplicações devem ter sempre a mesma visão do espaço de arquivos.
- **Mobilidade:** Com a movimentação dos arquivos, nem programas do cliente e nem tabelas de administração precisam ser modificadas de modo a refletir essa movimentação.
- **Desempenho:** programas clientes devem executar satisfatoriamente independente de variação de carga do serviço de arquivos.

Modificações concorrentes de arquivos: Alterações feitas por um cliente não devem interferir nas operações de outros clientes, mesmo que esses clientes estejam manipulando o mesmo arquivo.

Replicação de arquivos: Os SADs devem manter cópias atualizadas dos arquivos em diferentes locais. Essa cópia pode ser também apenas dos arquivos mais acessados ou simplesmente os pedaços dos arquivos que costumam ter um nível de acesso maior. O uso de cache em um sistema de arquivos pode ser encarado como uma replicação de arquivos, embora seu objetivo seja puramente desempenho (DE CARVALHO, 2003).

A replicação de arquivos fornece vantagens como balanceamento de carga, pois uma quantidade maior de servidores pode atender uma determinada requisição, melhora a escalabilidade e tolerância a faltas, pois se um arquivo falhar, ele pode ser requisitado a outro

servidor que o tenha, retornando a um estado consistente, sem que os clientes tenham conhecimento.

Heterogeneidade: Os SADs devem levar em consideração a heterogeneidade de plataformas (sistemas operacionais, arquiteturas de processador, dentre outros fatores), que podem ser utilizadas tanto pelos servidores, como pelos clientes.

Tolerância a falhas: Durante a transmissão de dados entre servidores e clientes, podem ocorrer falhas, seja por excesso de tráfego de pacotes pela rede, seja por um dos servidores estar sobrecarregado. Além disso, podem ocorrer falhas de hardware, especialmente dos mecanismos de armazenamento, de transmissão, etc (DE CARVALHO, 2003).

Seja qual for o problema, o sistema deve evitar que o cliente fique aguardando uma resposta por muito tempo, ou que seus dados sejam danificados ou até mesmo perdidos.

Consistência: Só deve haver uma versão disponível de cada arquivo no SAD, ou seja, se um cliente ler um arquivo em um dado momento e outro cliente ler o mesmo arquivo em outro momento ou lugar, caso o arquivo não tenha sido alterado, ambos os clientes devem visualizar o mesmo conteúdo.

Segurança: Compartilhar arquivos entre vários ambientes e usuários é uma das vantagens que os sistemas de arquivos distribuídos proporcionam, dessa forma, torna-se necessário adotar mecanismos de segurança, para evitar que pessoas desautorizadas tenham acesso aos arquivos do sistema (DE CARVALHO, 2003). Por isso os SADs devem se preocupar com:

- Permissão (a pessoa que está acessando o arquivo pode fazê-lo).
- Autenticação (essa pessoa é realmente ela?).
- Privacidade (dados secretos podem ser vistos por outras pessoas?).
- Integridade (Esse arquivo foi danificado no transporte?).

Eficiência: O SAD deve fazer todas as atividades listadas nos requisitos acima com um desempenho comparável a, ou melhor, que um sistema de arquivos centralizado, de modo que, mesmo se a carga varie constantemente, o funcionamento de aplicações clientes do SAD não seja muito penalizadas.

2.5 Sistemas de Arquivos Paralelos

Um Sistema de Arquivos Paralelo, segundo De Carvalho (2003), nada mais é que um SAD especializado em acesso concorrente e alto desempenho. Algumas implementações se

preocupam simplesmente com desempenho, esquecendo da segurança dos dados, ou até mesmo de fornecer alta disponibilidade dos dados. Por outro lado, outras implementações estão mais interessadas em disponibilizar um ambiente mais confiável para as aplicações, mesmo que isso acarrete uma perda de desempenho.

Outra definição mais abstrata sobre sistemas de arquivos paralelos é apresentada mostrando que esses sistemas é um subconjunto dos sistemas de arquivos distribuídos, que são geralmente sistemas de arquivos que fornecem o acesso compartilhado aos dados distribuídos, mas não apresentam necessariamente foco em desempenho ou acesso paralelo (PVFS Project, 2003).

O fornecimento de acesso compartilhado aos dados distribuídos permite que os sistemas de arquivos paralelos utilizem vários discos para armazenar os dados reduzindo o gargalo provocado pelo acesso constante a apenas um disco (DE CARVALHO, 2003).

2.5.1 Sistema de arquivos Lustre

Desenvolvido pela Sun Microsystems, o sistema de arquivos Lustre almeja providenciar alto desempenho e escalabilidade. Sua arquitetura é composta por um servidor de meta dados centralizado (MDS) e muitos servidores de dados (Object Storage Targets, OSTs). O MSD é responsável por todas as operações no espaço de nomes do sistema de arquivo, enquanto os OSTs cuidam dos acessos aos dados e sua interação com os dispositivos de armazenamento, os Object-BasedDisks (OBDs).

Embora os dispositivos de armazenamento sejam chamados OBDs, eles não precisam estar em discos. Como a sua interface com os OSTs usa um dispositivo de driver que esconde sua identidade, uma tecnologia de armazenamento sny pode ser aplicada, por exemplo, sistema de arquivos Linux (ext3, RaiserFS, etc). Para a comunicação, Lustre usa o Lustre Networking (LNET) que suporta diversas tecnologias de rede com drivers ajustáveis, chamados Lustre Network Drivers (LNDs). Os clientes acessam o serviço de arquivos através de uma camada virtual fornecida pelo sistema de arquivo. Os servidores de dados (data servers) não aplicam nenhum tipo de cache. Não obstante, cache pode ser providenciada pelo OBD.

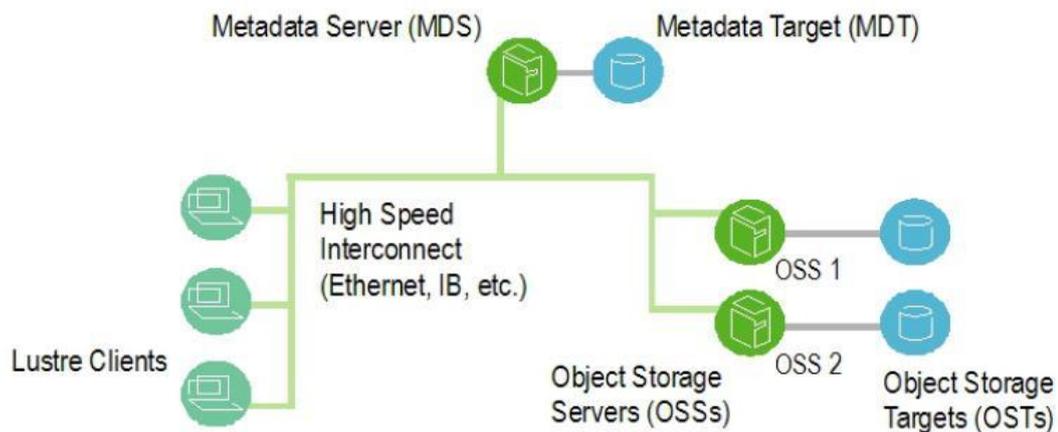


Figura 2- Arquitetura do Sistema de Arquivos Lustre.

2.5.2 dNFSp (Distributed Network File System Parallel)

Antes de descrevermos algo sobre o dNFSp vamos primeiramente falar sobre o pNFS. O pNFS é um projeto iniciado no Laboratório de Informática Distribuída, em Grenoble, na França com o objetivo de providenciar um incremento na performance do Servidor NFS. Dessa forma, esse projeto deseja oferecer um aumento no desempenho de aplicações paralelas mantendo o nível de intrusão numa instalação de cluster tradicional o mínimo possível.

O pNFS consiste de vários nodos E/S, que podem ser chamados de iods, que são responsáveis pelo armazenamento e recuperação dos blocos de dados resultantes do file striping e um meta Servidor responsável por encaminhar as requisições recebidas para os iods mais apropriados para responder ao cliente que solicitou a tarefa.

Uma vez recebidas essas tarefas, o meta servidor encaminha as requisições para os iods, que por sua vez podem trabalhar em paralelo aumentando assim o desempenho. Os iods podem rodar nas máquinas clientes sem nenhuma restrição.

Uma das características mais comum das variações implementadas do pNFS é que o meta servidor é somente um processo rodando em um dos nodos. Esse design permite um aumento de desempenho no caso de operações de leitura, porque a maioria das informações envolvidas no processo são enviadas diretamente dos iods para os clientes, porém nos casos de escrita os dados tem que ser enviados para o meta servidor e o tradicional gargalo persiste.

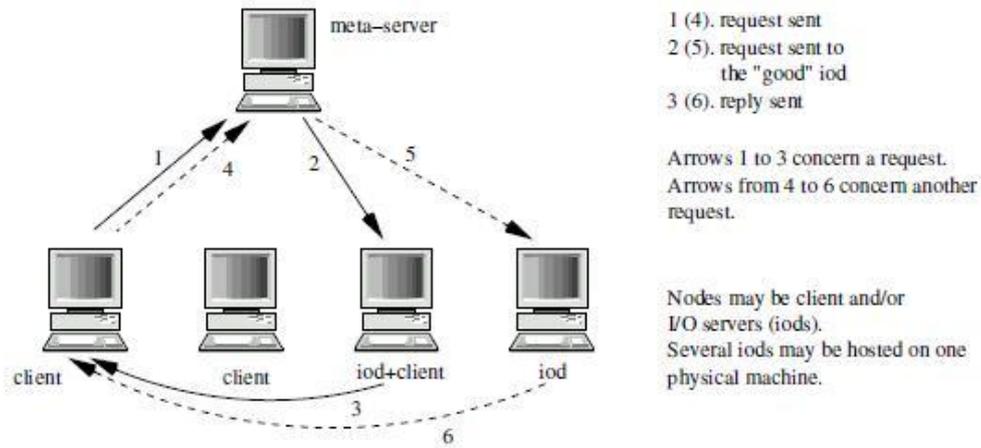


Figura 3 - Arquitetura do Sistema de Arquivos NFSp.

Para resolver esse problema, o dNFSp (Distributed NFS Parallel) foi proposto como uma variante do pNFS. O dNFSp consiste em criar réplicas do meta servidor em mais nodos do sistemas aumentando assim a largura de banda. Todavia, cada subconjunto de clientes visualiza apenas o meta servidor ao qual está associado como se fosse o único meta servidor de todo o sistema. Cada meta servidor continua capaz de visualizar todos os iod do sistema.

Com a criação das réplicas do meta servidor um novo problema se apresenta: a consistência dos dados. Se um cliente cria um novo arquivo, ele existirá apenas no meta servidor cujo qual o cliente está associado e será visualizado apenas pelos clientes associados ao mesmo meta servidor. Isso significa que os clientes associados a outros meta servidores não conseguirão visualizar tal arquivo.

Como solução para esse problema é utilizado um mecanismo baseado em LRC. Dessa forma um meta servidor só irá copiar um arquivo existente em outro meta servidor quando realmente necessário. Esse fato normalmente ocorre quando um meta servidor tenta acessar um arquivo inexistente e então passa a procurá-lo em outros meta servidores copiando o arquivo quando esse for encontrado. Para isso, é partido do princípio de que nunca ocorrerá uma requisição de acesso a um arquivo que definitivamente não exista.

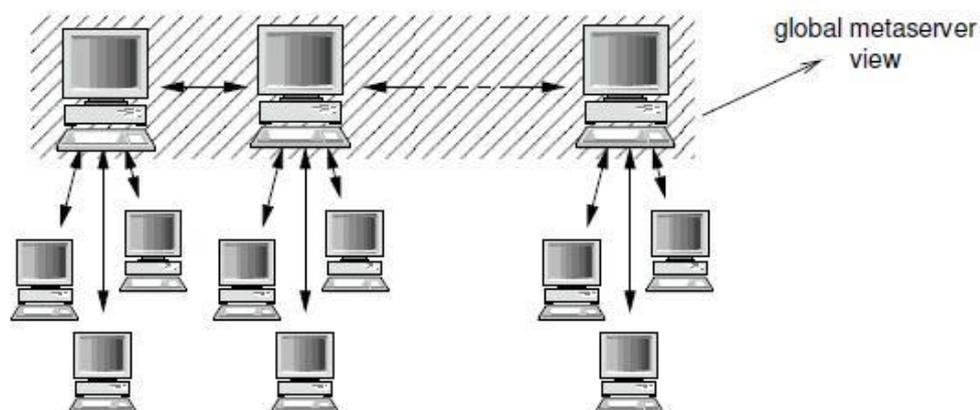


Figura 4 - Arquitetura do Sistema de Arquivos dNFSp

2.5.3 FhGFS (Fraunhofer File System)

Fraunhofer é um sistema de arquivos paralelo de alto desempenho para Linux. Ele foi desenvolvido combinando os últimos resultados de pesquisas e otimizações para Computação de Alto Desempenho. O sistema de arquivo foi desenhado para permitir escalabilidade, flexibilidade e uma arquitetura paralela robusta.

Vantagens: Conteúdo de arquivos distribuídos e metadados: Um dos conceitos mais fundamentais do FhGFS é sua arquitetura feita para evitar gargalos. Dividir o conteúdo dos arquivos em múltiplos servidores de armazenamento é apenas uma parte desse conceito.

Outro aspecto importante é a distribuição dos metadados do sistema de arquivo através de múltiplos metaservidores.

FhGFS combina múltiplos servidores de armazenamento para permitir o recurso de armazenamento em rede compartilhada com divisão de arquivos. Desse modo, o sistema supera as limitações de desempenho de servidores únicos, conexões de internet únicas, número limitado de drivers de disco, etc. Nesse tipo de sistema, as demandas elevadas de produção de um grande número de clientes podem ser facilmente satisfeitas e mesmo um único cliente pode lucrar com o desempenho agregado de todos os nós de armazenamento no sistema.

Tudo isso é possível graças à separação de metadados e conteúdo de arquivos. Os Clientes FhGFS tem acesso direto as servidores de armazenamento e comunicam-se com múltiplos servidores simultaneamente dando à aplicação acesso paralelo real as dados do

arquivo. Para manter mínima a latência de acesso aos metadados, FhGFS permite distribuir os metadados através de múltiplos servidores.

Além dos três papéis básicos no FhGFS (clientes, servidores de metadados, servidores de armazenamento) existe mais dois serviços de sistema que fazem parte do FhGFS. O primeiro é o gerenciamento padrão (management daemon) que serve como um ponto central de informação para clientes e nodos do grupo. O segundo é o serviço de administração e monitoramento que fornece um web browser gráfico para a instalação e a monitoração permanente do status de sistema.

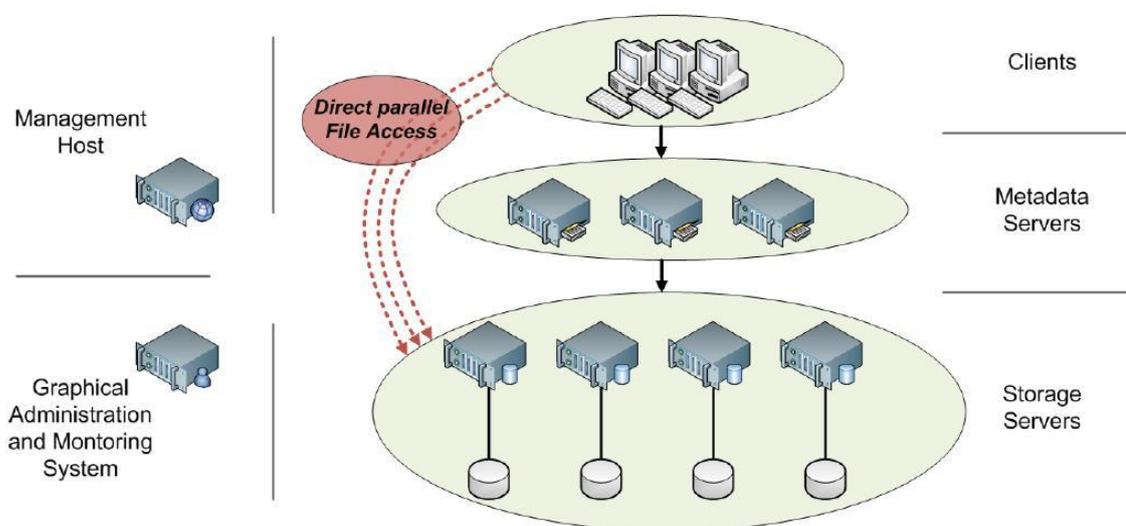


Figura 5 - Arquitetura do Sistema de Arquivo Fraunhofer.

2.5.4 Sistema de arquivos distribuído Hadoop (HDFS)

Hadoop é uma plataforma de software em Java de computação distribuída voltada para clusters. Esse projeto foi criado por Doug Cutting e surgiu como um pequeno subprojeto dentro do projeto apache Lucene e tornou-se um grande framework para computação distribuída sendo usado cada vez mais por grandes empresas da web. Como paradigma de programação ele utiliza Map Reduce para a realização de computação distribuída em clusters com grande quantidade de máquinas e vem consolidando-se em um padrão de fato quando se trata de em processamento distribuído de alta escala.

Os componentes mais conhecidos do Hadoop são seu sistemas de arquivos distribuídos (HDFS) e o Hadoop MapReduce.

O Hadoop MapReduce é um modelo de programação paralela e framework para criação de aplicações que processam rapidamente vastas quantidades de dados em paralelo através de clusters montados com uma quantidade grande de computadores comuns. O Algoritmo MapReduce especifica uma função Map que processa um par chave/valor gerando um par chave/valor intermediário e uma função Reduce que mescla os pares chave/valor intermediários até a saída de uma única chave final (DELMERICO, 2009).

O framework controla o escalonamento, monitoramento e execução dos Jobs MapReduce juntamente com a gestão dos nós de armazenamento do HDFS (HDFS Architecture Guide, 2010).

O Hadoop Distributed File System (HDFS) é o sistema básico de armazenamento utilizado por aplicações Hadoop. O HDFS cria réplicas de blocos de dados e que são distribuídos no cluster para permitir computações extremamente rápidas.

O Sistema de arquivos Distribuído Hadoop foi desenvolvido para ser executado em *commodity hardware* sob sistema operacional GNU/Linux. Ele possui muita similaridade com os outros sistemas de arquivos distribuídos já existentes, mas possui diferenças significativas, tais como ser altamente tolerante a falhas e ser projetado para ser aplicado em hardware de baixo custo. HDFS fornece alto *throughput access* aos dados da aplicação e é apropriado para aplicações que produzem um grande conjunto de dados.

O HDFS possui uma arquitetura mestre/escravo e um HDFS cluster consiste de um nameNode, um servidor mestre que gerencia o espaço de nomes do sistema de arquivos e regula o acesso aos arquivos pelos clientes. Além desses, possui ainda nós de dados, geralmente um para cada nó do cluster e que gerencia o armazenamento anexado aos nós sobre cujo quais estão rodando.

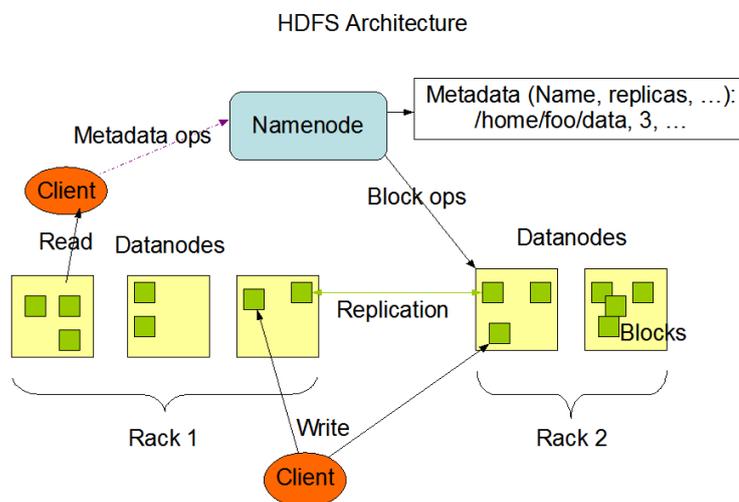


Figura 6 - Arquitetura do Hadoop Distributed File System.

O HDFS possui um espaço de nomes e permite que os dados do usuário possam ser armazenados em arquivos. Internamente esses arquivos são separados em um ou mais blocos e armazenados em um conjunto de nós de dados. O namenode executa operações no sistema de arquivos, como abrir, fechar e renomear arquivos e diretórios. Ele também determina o mapeamento dos blocos para os datanodes. Os datanodes são responsáveis pelo fornecimento de pedidos de leitura e escrita do sistema de arquivos dos clientes. Os datanodes também executam a criação, exclusão e replicação de blocos sob instruções do namenode.

2.5.5 PVFS (Parallel Virtual File System)

O PVFS é desenvolvido conjuntamente entre o Laboratório de Pesquisas em Arquiteturas Paralelas (*Parallel Architecture Research Laboratory*) da Clemson University, a Divisão de Matemática e Ciência da Computação (*The Mathematics and Computer Science Division*) do Argonne National Laboratory e o Centro de Supercomputação de Ohio (*Ohio Supercomputer Center*). Seu desenvolvimento foi financiado pela NASA Goddard Space Flight Center, Laboratório Nacional de Argonne - DOE, pelo programas NSF PACI e HECURA, e outras agências do governo e privadas.

O projeto do *Parallel Virtual File System* ou PVFS [PVF 08] tem o objetivo de explorar os vários métodos para concepção, implementação e utilização de I/O paralelo. Ele pode ser utilizado tanto como plataforma para pesquisas em I/O paralelo, quanto para construção de sistemas de arquivos distribuídos para aplicações que necessitam distribuir suas informações em agregados computacionais.

Ele foi desenvolvido para aplicações paralelas compartilhando um robusto conjunto dados através de vários clientes. Para fazer isso com alto desempenho, muitos servidores são usados para providenciar múltiplos paths aos dados (PVFS Project, 2003).

PVFS usa uma arquitetura inteligente de servidor. Isso significa que ele faz mais do que simplesmente fornecer blocos de dados dos discos aos clientes, e possui alto nível de abstração se tratando de arquivos e diretórios. Ele utiliza uma alternativa de armazenamento compartilhado, onde os dispositivos de armazenamento (geralmente acessados pela granularidade de um bloco) são acessados diretamente pelos clientes (PVFS Project, 2003).

3 Revisão do Estado da Arte e Prática

Foi realizado um levantamento de publicações e ferramentas utilizando os seguintes critérios para a inclusão dos mesmos:

- Temas com enfoque em técnicas e modelos de avaliação de desempenho no armazenamento em sistemas de arquivos distribuídos;
- Avaliação de fatores estruturais e configuração de ambientes que exerçam influência no desempenho das ferramentas;
- Modelos práticos e comparativos que pudessem justificar a utilização de sistemas de arquivos distribuídos na proposta deste trabalho;
- Além dos critérios supracitados, foi utilizado como premissa que os trabalhos estivessem relacionados com as ferramentas aqui sugeridas.

Foram excluídos em sua maioria trabalhos com:

- Propostas de implementação de algoritmos em nível de software para as ferramentas utilizadas com o intuito de prover melhor desempenho;
- Artigos de autores distintos que avaliassem os mesmos fatores influentes no desempenho para ferramentas distintas, procurando assim abranger o máximo possível de métricas diferentes para desempenho.

A busca foi feita preferencialmente em base de dados de pesquisa acadêmica, tais como Science Direct, Portal ACM, IEEE Explore e em sites de busca como Google Scholar.

Foram lidos 100 abstracts e escolhidos 30 artigos inicialmente. Dentre esses, 10 são apresentados a seguir por conterem técnicas de avaliação de desempenho, modelos comparativos e análise prática dos resultados, à luz dos objetivos deste trabalho.

3.1 Avaliação de desempenho

No trabalho de Kassick et al (2008) foram apresentadas seis classes de testes para simular as cargas de trabalho observadas em aplicações científicas: SFWR, SFWR-SMP, SFSSR, SFSS, MFNS e MFSS para avaliar dois diferentes SAPs: Lustre e dNFSp.

- SFWR (Single File, Whole Read): O teste consiste numa grande quantidade de clientes realizando leitura de todo o arquivo compartilhado.

- SFWR-SMP (Single File, Whole Read, SMP-nodes): Basicamente a mesma coisa que o teste SFWR, mas muitas instâncias da aplicação são executadas no mesmo nó computacional.
- SFSR (Single File, Segmented Read): Esse teste utiliza leituras sequenciais como o SFWR, mas não há sobreposição em dados lidos por duas instâncias do aplicativo.
- SFSS (Single File, Share Segments): Esse teste é dividido em duas fases distintas: primeiro cada instância da aplicação paralela lê e escreve operações no seu próprio segmento de um arquivo compartilhado. Na próxima fase, os offsets do segmento são alterados de modo que eles incluam dados escritos por uma instância do segmento vizinho.
- MFNS (Multiple File, No Sharing): Nesse teste, cada instância da aplicação lê e escreve em seu próprio arquivo em vez de seu segmento exclusivo (Como no SFSR).
- MFSS (Multiple File, Shared Segments): Esse avalia o comportamento observado em SFSS, mas em sua segunda fase ao invés das instâncias da aplicação acessar dados de segmentos vizinhos ele realizará leitura de arquivos diferentes.

Pelo fato de diferentes aspectos, tais como cache e escalonamento do conjunto de dados entre os nós, influenciarem no desempenho de sistemas de arquivos distribuídos, os testes propostos são capazes de avaliar o comportamento dessas ferramentas em situações similares e gerar dados consistentes considerados importantes para o estudo das mesmas.

Em seu trabalho Boito (2009), sugere que além de aplicar as classes de testes acima citadas para avaliar o sistema de arquivos Lustre, que ainda sejam aplicados dois grandes testes para todas as classes:

- Variando o número de clientes: a quantidade e o tamanho dos segmentos são fixos (32 objetos com 64 MB cada, para a maioria dos testes) enquanto o número de clientes envolvidos é incrementado, partindo de 1 até 40. O tamanho dos dados cresce de acordo com o número de clientes.
- Variando o número de segmentos: o número de clientes é fixado em 40, enquanto o número de segmentos aumenta. O tamanho de cada um dos segmentos é adaptado para manter a quantidade de dados acessados.

A figura 7 mostra os resultados obtidos para a classe SFWR testando a variação no tamanho de bloco das requisições. Já a figura 8 mostra os resultados para teste com variação no número de clientes e com tamanho de bloco fixo de 2KB. É possível perceber que o desempenho aumenta até tornar-se estável e depois não sofre degradação com o aumento no número de clientes (BOITO, 2009).

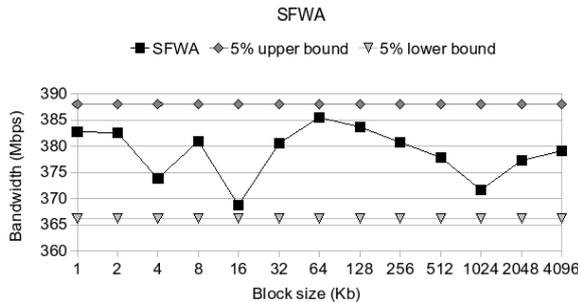


Figura 7 - Teste SFWA que varia o tamanho de bloco da requisições.

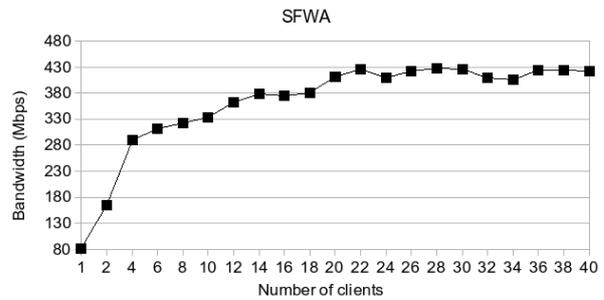


Figura 8 - Teste SFWA variando o número de clientes envolvidos.

Com esses resultados ratifica-se claramente que não apenas o tamanho de bloco das requisições interfere no desempenho do SAP Lustre, como não adianta configurar um ambiente com inúmeros clientes para prover melhor desempenho, pois uma hora esse desempenho torna-se estável de acordo com a configuração das máquinas, rede e da ferramenta em si.

Os resultados para o teste da classe MFWA variando o número de clientes é apresentado na figura 9. Assim como para a classe de teste anterior o desempenho aumenta até tornar-se estável.

A figura 10 mostra um comparativo para o teste de operações de leitura entre as classes SFWR e MFWA. Para a classe SFWR temos um desempenho melhor devido ao sistema de cache dos servidores, uma vez que os dados acessados são os mesmos para todos os clientes (BOITO, 2009).

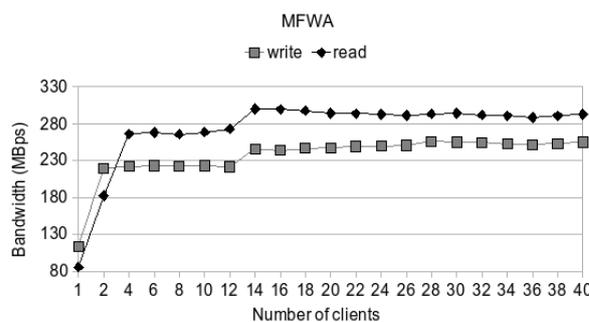


Figura 9 - Teste MFWA variando o número de clientes envolvidos.

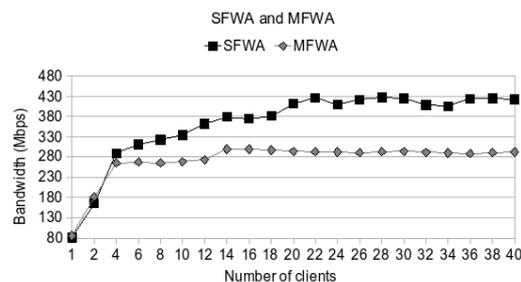


Figura 10 - Teste de operações de leitura MFWA e SFWA variando número de clientes envolvidos.

Já o autor Shvachko avalia uma ferramenta diferente do Lustre, o HDFS (Hadoop Distributed File System), e sugere que para o principal objetivo desta ferramenta, que é fornecer uma largura de banda E/S elevada para grandes conjuntos de dados, que há três métricas capazes de testar esse objetivo:

- Qual é a largura de banda observada de um benchmark artificial?
- Que largura de banda é observada em um cluster de produção com múltiplos Jobs de usuários misturados?
- Que largura de banda pode ser obtida por uma aplicação de usuário de grande-escala cuidadosamente construída?

O autor ainda relata que dependendo da escala da quantidade de nós, a largura de banda é linear e que a estatística interessante de ser feita passa a ser a largura de banda em cada nó.

3.2 Técnica / Modelos para prover melhor desempenho de armazenamento utilizando sistemas de arquivos distribuídos

Piernas et al (2007), propõe em seu trabalho uma implementação de armazenamento ativo (inteligente), no sistema de arquivos Lustre, para prover melhor desempenho e obter vantagens do tempo de CPU subutilizados nos nós de armazenamento. A idéia subjacente de armazenamento ativo baseia-se na utilização do poder de processamento dos nós de armazenamento para executar tarefas de computação sobre dados que são lidos e gravados nos arquivos locais. Isto pode ser completamente feito no espaço de usuário se os nós de armazenamento também forem clientes do sistema de arquivos paralelo. Assim sendo, os nós de armazenamento podem acessar todos os arquivos no sistema de arquivos, especificamente, para os arquivos armazenados localmente.

Os resultados experimentais obtidos ratifica que a implementação é capaz de tirar proveito dos nós de armazenamento, aumentando o desempenho das aplicações que processam uma massiva quantidade de dados, reduzindo significativamente o tráfego total da rede ao mesmo tempo, o que pode prevenir um gargalo na mesma (PIERNAS; NIEPLOCHA; FELIX, 2007).

Embora implementações de códigos não seja um dos focos deste trabalho, o trabalho de Piernas et al (2007), enfatiza como é importante a extração do máximo de desempenho possível das ferramentas de sistemas de arquivos distribuídos, além de uma configuração correta da mesma quando se trata de quantidades intensivas de dados. Além das vantagens já oferecidas por estas essas ferramentas, é proposto implementações com conceito de

armazenamento ativo em nível de espaço de usuário e de kernel como opção para se ter o máximo desempenho possível.

Em seu trabalho, Zhao et al (2010) apresentada uma proposta de um modelo de desempenho relativo capaz de prever diferenças de desempenho entre um par de sistemas Lustre equipados com diferentes fatores de desempenho. O experimento foi projetado para avaliar quatro categorias de fatores de desempenho importantes: o número de OSSes, abordagem sobre conexões de armazenamento, o tipo de discos utilizados e o número de threads/OST.

O modelo de desempenho relativo pôde obter melhor a precisão da previsão de desempenho de um dispositivo, e os resultados confirmam que para uma determinada carga de trabalho, o desempenho de um dispositivo geralmente é o melhor predictor para o desempenho de outro (ZHAO; MARCH; DONG, 2010).

Avaliando esse modelo, é possível perceber que pelo fato de serem comparadas ferramentas indênticas, com uma mesma carga de trabalho, e equipadas com diferentes fatores de desempenho, os resultados de uma configuração, embora sirvam como base para previsão do desempenho de outra configuração poupando tempo e esforço, serão diferentes, ratificando a importância desses fatores para a avaliação de desempenho dessas ferramentas.

Na tentativa de alcançar um equilíbrio entre desempenho e simplicidade de gestão, o projeto NFSP (LOMBARD; DENNEULIN, 2002) propõe uma versão distribuída do tradicional servidor NFS que trate de uma maneira melhor a carga gerada por múltiplos acessos concorrentes feitos pelos nós computacionais, mas que ainda permaneça compatível com o cliente NFS presente em todo sistema Linux.

No trabalho apresentado por Ávila (2004) é proposto um modelo de replicação de servidor para o projeto NFS original. Este, por sua vez, é projetado para ter um único servidor (nfsd) sendo acessado por todos os clientes conectados. O modelo proposto consiste em replicar o servidor NFS para múltiplas máquinas numa tentativa de aumentar a escalabilidade para um cenário de computação distribuída.

Esses meta-servidores são executados independentemente, mas em cooperação entre si, realizando comunicação em rede e formando uma noção de servidor único e global. O principal objetivo desse modelo é distribuir a carga das operações de meta-dados, evitando o

saturamento frente múltiplas requisições concorrentes do servidor único apresentado no projeto original (ÁVILA, 2004). Para o experimento de desempenho de leitura, o teste consiste na leitura concorrente de um arquivo com 1 GB de tamanho fixo por todos os clientes.

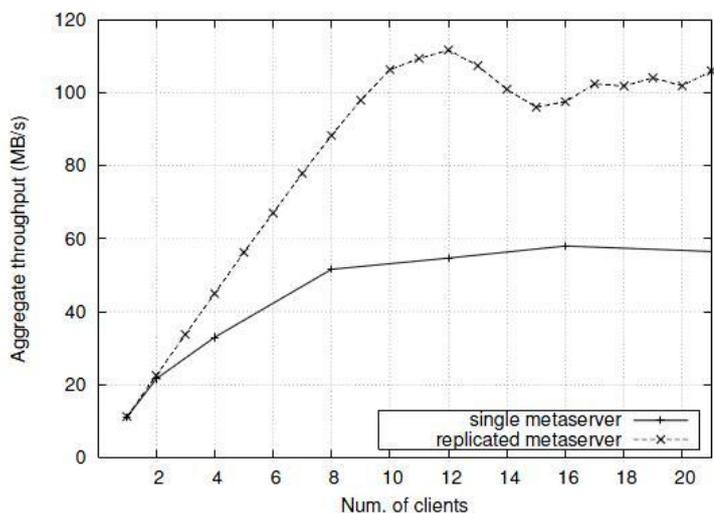


Figura 11 - Throughput de leitura agregado obtido com o protótipo, em comparação com o modelo original.

Como observado na figura acima, ter múltiplas instâncias do meta-servidor distribui a carga imposta pelos acessos simultâneos e, portanto, permite um ganho de desempenho maior em relação ao servidor NFS centralizado.

A pesquisa de Hildebrand, Ward e Honeyman (2006) demonstra que utilizando pNFS, que segundo Fuhrmann (1997) é uma extensão do NFS, que foi introduzido para prover uma visão dos dados de uma forma mais amigável e para resolver algumas limitações do NFS, como: operações sem estado (stateless), o desempenho das operações de E/S é considerado lento, o número de operações do NFS é fixo e limitado, entre outras.

Com esta pesquisa pôde ser verificado um aumento do throughput no armazenamento de dados paralelos. A avaliação desta metodologia foi realizada em cima de uma aplicação com dados científicos reais e demonstra os benefícios da utilização desta tecnologia. Que ratifica que a verificação do protocolo de metadados heterogêneo pNFS permite realizar escrita de forma paralela, o que possibilita desta forma um aumento de performance.

3.3 Estudos Comparativos

Em seu trabalho, Sebepon et al (2008) faz uma avaliação comparativa experimental de dois sistemas de arquivos paralelos, PVFS2 e Lustre, em uma série de benchmarks modelando aplicações do mundo real. Os resultados indicam que ambos os sistemas de arquivos são bem sucedidos na oferta de E/S em paralelismo para o caminho de armazenamento, atingindo bom desempenho sobre uma ampla gama de aplicações. Em áreas onde as arquiteturas dos sistemas de arquivos diferenciam, como na cache do cliente, garantia de consistência dos dados e gerenciamento de metadados os resultados mostram que: (a) o custo de gestão da consistência dos dados pode ser uma sobrecarga desnecessária para aplicações que a exijam; (b) O gerenciamento eficiente de metadados pode ser uma aptidão diferenciadora, fundamental para aplicações necessitem de altas taxas de transação sobre grandes conjuntos de pequenos objetos de dados.

Muitas aplicações, tanto científicas como comerciais, necessitam transferir grande quantidade de dados de e para um armazenamento secundário. O desempenho de E/S pode assumir um papel crítico no tempo de conclusão total dessas aplicações (GARG; MACHE, 2001).

A idéia de E/S paralelo é aumentar tanto a capacidade de armazenamento como o desempenho de E/S, armazenando dados através de múltiplos discos.

Garg e Mache (2001) comparam em seu trabalho o desempenho de E/S de dois sistemas de arquivos populares: Intel PFS para ASCII Red e PVFS para clusters Linux.

O método utilizado explorou a eficácia do PVFS no aumento do tempo de conclusão total de uma aplicação ray tracing (MACHE, 2000). Ray tracing é uma técnica para renderizar uma cena composta de objetos matematicamente definidos seguindo o conjunto de raios que emanam do observador e passa através de vários pixels da tela de projeção. Ao invés de exibir os quadros calculados, eles foram salvos em disco.

O programa de ray tracing computou 10 quadros de uma cena simples em 840x840 pixels cada, resultando em 27 MBytes de dados que foram gravados em um número especificado de nós E/S. Foram rodados dez experimentos, cada qual com uma média dos resultados de 40 execuções do programa ray tracing.

Os resultados mostram que é possível reduzir o tempo de conclusão total em um fator de até 1.563 e com a utilização do PVFS é possível obter um throughput de E/S de mais de 1Gigabyte/seg. Isso tudo depende das escolhas de configuração e programação do ambiente(GARG; MACHE, 2001).

Goldman e De Carvalho (2005) propõe uma comparação entre os sistemas de arquivos paralelos PVFS2 e o sistema de arquivos local Ext3, usando uma única máquina como cliente. Parece estranho essa comparação já que essas ferramentas possuem propósitos distintos, mas ela pode ser justificada caso haja largura de banda suficiente na rede para o tráfego dos dados, o que pode tornar o sistema de arquivos paralelo mais rápido que o sistema de arquivos local (GOLDMAN; DE CARVALHO, 2005).

A figura 12 apresenta os resultados obtidos quando a velocidade dos discos locais das máquinas envolvidas foi reduzida para simular uma rede de dados mais rápida que o sistema de armazenamento local. Como pode ser observado, há um ganho considerável ao usar o PVFS2 no lugar do Ext3 para esse tipo de situação (GOLDMAN; DE CARVALHO, 2005).

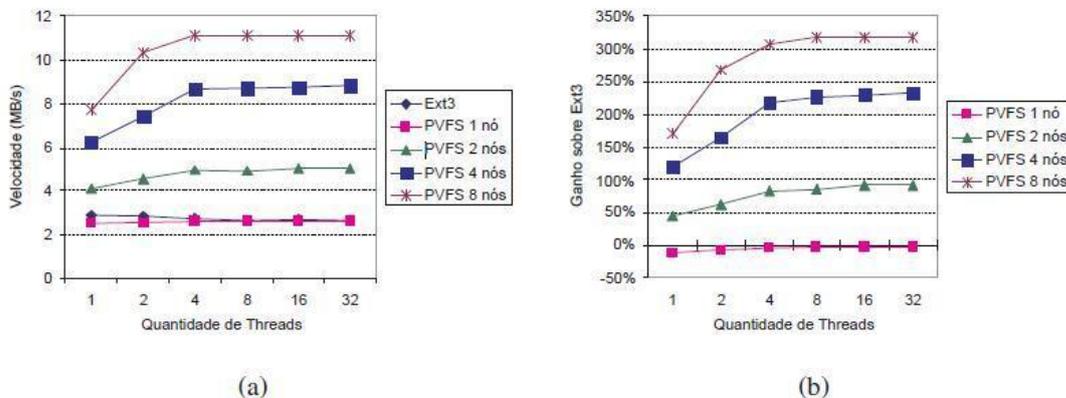


Figura 12 - (a) Velocidade do PVFS2 e Ext3 e (b) ganho do PVFS2 sobre o Ext3 ao aumentar a concorrência no acesso, usando discos na velocidade 2,5MB/s.

Ao aumentar a velocidade dos discos foi analisado o impacto no desempenho do PVFS2. A partir dos resultados obtidos foi estimado que, através de uma simples regra de três, que o desempenho do PVFS2 em uma rede conectada a 1Gbit/s seria de aproximadamente 109MB/s, considerando que a velocidade máxima da rede seria de 125MB/s, e que os discos não usam mais do que 92% da banda e que não haveria nenhum outro gargalo a não ser o próprio disco (GOLDMAN; DE CARVALHO, 2005).

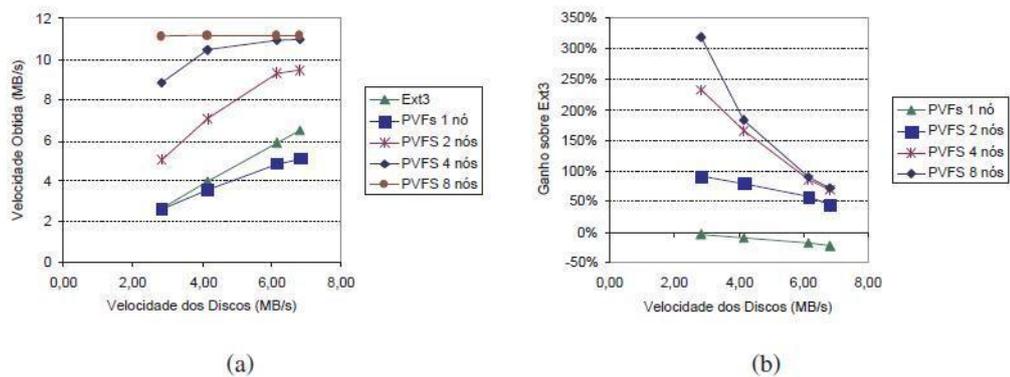


Figura 13 - (a) Velocidade do PVFS2 e do Ext3 e (b) ganho do PVFS2 sobre o Ext3 ao aumentarmos a velocidade dos discos, usando 32 threads.

Na etapa de verificação, teve-se acesso a um aglomerado de 12 máquinas, todas conectadas entre si através de placas de rede Gigabit, ligadas a um switch de mesma velocidade. Foram realizados testes comparativos de desempenho entre o PVFS2 e o Ext3 com os discos na velocidade máxima (GOLDMAN; DE CARVALHO, 2005).

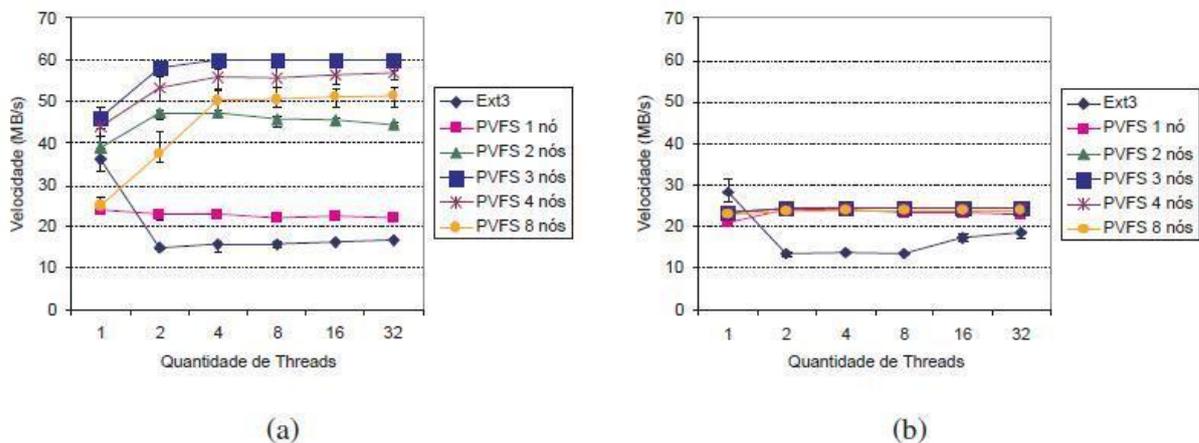


Figura 14 - Velocidade do PVFS2 e do Ext3 em (a) um cliente com processador Pentium 4 2,8GHz e em (b) um cliente com processador Pentium 4 1,8GHz, ambos em rede Gigabit.

Os resultados obtidos, conforme figura 14, mostra que o Ext3 perde muito desempenho quando se tem algum acesso concorrente, enquanto que o PVFS2 mantém-se constante. É possível notar também que o desempenho dos PVFS2 com 8 servidores é pior do que com 4. Isso se deve ao fato dos testes terem sido realizados numa rede não homogênea, sendo priorizadas as melhores máquinas para os testes com menos servidores (GOLDMAN; DE CARVALHO, 2005).

Com esses resultados é possível perceber que o poder de processamento e transmissão de dados do cliente passa a ser o gargalo principal. Com um cliente mais potente é possível obter

melhor desempenho do PVFS2, acessando o mesmo sistema de arquivos, mostrando que tanto a rede como a CPU são mais importantes que a velocidade do disco quando o objetivo é atingir alto desempenho no acesso ao sistema de arquivos.

3.4 Análise crítica dos trabalhos

A manipulação de imagens médicas ainda é um tema pouco abordado pelo fato que na maioria dos casos os servidores destas imagens vêm preparados para realizar o armazenamento de suas informações em banco de dados relacionais.

O objetivo dessa revisão foi esclarecer o contexto da avaliação de desempenho no armazenamento de dados em sistemas de arquivos distribuídos além de metodologias de avaliação entre as ferramentas.

Nota-se que, em sua grande maioria, os trabalhos analisados utilizam sistemas de arquivos distribuídos para prover melhor desempenho no armazenamento e recuperação de uma grande quantidade de dados provindos de aplicações de computação de alto desempenho, porém, esse desempenho pode ser alterado por uma série de fatores que vão desde a configuração das ferramentas utilizadas, passando por técnicas de cache e gerenciamento de consistência de dados e finalizando com a configuração da rede e largura de banda, entre outros.

É em cima de alguns desses fatores, e outros, que estão focados a maioria dos temas dos trabalhos acima citados e outros estudados que aqui não foram listados. Em grande parte, descrevem e propõem alterações e até mesmo novas implementações de suas características, tais como consistência, replicação, etc., para que se possa extrair o máximo desempenho possível dessas ferramentas.

Já outros trabalhos fazem uma abordagem mais técnica sobre técnicas de avaliação do desempenho dessas ferramentas ou utilizam benchmarks já criados para definirem um conjunto melhor de critérios que estabeleçam a avaliação do sistema de arquivos de uma maneira mais ampla, e não focada apenas na quantidade de dados transferidos por unidade de tempo.

Até o presente momento, foi apurada a importância dos sistemas de arquivos distribuídos e sua relevância para o armazenamento de uma massiva quantidade de dados, além de técnicas para extrair o melhor desempenho plausível, fundamentando assim a escolha da utilização dessas ferramentas neste trabalho.

Ainda foram apresentados trabalhos com análises comparativas entre sistemas de arquivos distribuídos diferentes e com sistemas de arquivos centralizados. A revisão do estado da arte forneceu análises que contribuíram para o embasamento teórico, apresentando resultados práticos, para a utilização de um método que fosse capaz de fundamentar as necessidades desta pesquisa.

É baseado nesses métodos que a proposta deste trabalho é, para um conjunto fixo de clientes, variando o tamanho da quantidade de dados a serem armazenadas, gerar e avaliar um conjunto de métricas que sejam capazes de definir qual ferramenta fornece melhor desempenho para conjuntos de dados distintos e depois uma avaliação geral entre as ferramentas utilizadas com o propósito de analisar as vantagens e desvantagens da implantação desta proposta para o armazenamento das informações médicas provindas da RCTM.

A tabela 2 mostra um resumo dos trabalhos aqui citados, apresentando os títulos dos trabalhos, os nomes dos autores, a proposta do estudo assim como uma breve descrição do modelo ou método aplicado. São apresentados os resultados e o ambiente de aplicação para alguns dos trabalhos, assim como algumas observações consideradas pertinentes.

Ao lado do título de cada um dos trabalhos relacionados é apresentado entre parênteses a qual categoria pertence o tema descrito no artigo de acordo com a seguinte divisão:

- 1 - Avaliação de desempenho;
- 2 - Técnicas / Modelos para prover melhor desempenho de armazenamento utilizando sistemas de arquivos distribuídos;
- 3 - Estudos comparativos.

Tabela 2 - Resumo dos trabalhos citados.

Titulo	Autores	Proposta do estudo	Descrição	Ambiente	Resultados	Observações
Testing the Performance of Parallel File System - (1)	Rodrigo Virote Kassick, Franciele Zanon Boito, Philippe O.A. Navaux	Este trabalho propõe um conjunto de testes para avaliar os sistemas de arquivos distribuídos em situações semelhantes às experimentadas durante a execução de aplicações científicas.	São propostas seis classes de testes (SFWR, SFWR-SMP, SFSR, SFSS, MFNS, MFSS) que visam avaliar os sistemas de arquivos dNFS e Lustre de acordo com alguns padrões de acesso observada na bibliografia.	-	Por ser um trabalho teórico não apresenta resultados práticos.	Este trabalho trata-se de uma proposta teórica de classes de testes para avaliação de desempenho de sistemas de arquivos distribuídos com o intuito de prover informações mais completas e válidas na escolha de uma ferramenta desse tipo para determinada aplicação.
Evaluating the performance of Lustre File System – (1)	Franciele Zanon Boito, Rodrigo Virote Kassick, Philippe O.A. Navaux	Avaliar as classes de testes acima sugeridas e simular as cargas de trabalho de aplicações científicas para obter resultados que possam ser utilizados na criação de um perfil sobre o comportamento do sistema de arquivos Lustre.	Para a execução dos testes, foi utilizado o teste MPI-IO, uma ferramenta de código aberto que funciona em cima das chamadas MPI-IO. O teste é descrito por argumentos passados, como operação (leitura ou escrita), arquivos envolvidos, barreiras e segmentação do arquivo (Strided or non-strided).	O Lustre foi usado com 4 OSTs e no máximo 40 clientes. Nos OSTs o sistema de armazenamento é o sistema de arquivos local (ext3). A quantidade de dados acessados em cada cliente é de 2 GB. O cluster utilizado é o Helios, de Grid 5000. Tem nós AMD Opteron com 4 GB de RAM e uma rede Ethernet Gigabit.	Os resultados para alguns dos testes sugeridos permitiram obter algumas conclusões sobre o comportamento do Lustre: ele escala bem não sofrendo degradação com o crescimento do número de clientes, operações de leitura têm melhor desempenho do que as de escrita, o desempenho é melhor quando todos os clientes iniciam suas requisições no mesmo servidor, etc.	Quando a principal proposta de um trabalho for avaliar exclusivamente uma ferramenta de sistema de arquivos distribuído, a aplicação destas classes de teste são muito válidas pois ajudam a avaliar algumas características da ferramenta e a traçar um perfil de comportamento para a mesma.
The Hadoop Distributed File System – (1)	Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler	Descreve a arquitetura do HDFS (Hadoop Distributed File System) e relata uma experiência utilizando o HDFS para gerir 25 petabytes de dados corporativos do Yahoo!.	Descreve toda a arquitetura do HDFS e descreve o funcionamento de algumas de suas características, tais como: leitura e escrita de arquivos, consistência, gerenciamento de replicação, Balanceamento, etc. Depois descreve o comportamento de uma experiência utilizando o HDFS para gerir 25 petabytes de dados do Yahoo!.	-	A principal contribuição deste trabalho é sugerir que o principal objetivo do HDFS é fornecer uma largura de banda E/S elevada para grandes conjuntos de dados, além de perguntas cujas respostas podem auxiliar em um melhor desempenho neste quesito.	Este trabalho além de prover uma visão geral da arquitetura do Hadoop e do funcionamento de algumas de suas características, ele é referenciado nesta pesquisa por levantar questões que podem prover métricas a mais no desempenho da largura de banda.

Tabela 2 – Resumo dos trabalhos citados (cont).

Título	Autores	Proposta do estudo	Descrição	Ambiente	Resultados	Observações
Evaluation of active storage strategies for the Lustre parallel file system – (2)	Juan Piernas, Jarek Nieplocha, Evan J. Felix	É apresentado uma implementação em espaço de usuário de armazenamento ativo para o sistema de arquivos Lustre, e comparado com uma implementação tradicional kernel-based.	É apresentado a avaliação de duas implementações com conceito de armazenamento ativo para o sistema de arquivos Lustre 1.6. Foram realizadas métricas de duas aplicações em três configurações diferentes: sem armazenamento ativo, com implementação de armazenamento ativo em user-space e outra em kernel-space.	-	Os resultados experimentais obtidos mostram que a implementação de armazenamento ativo é capaz de tirar proveito dos nós de armazenamento, aumentando o desempenho das aplicações de uma massiva quantidade de dados e reduzir significativamente o tráfego total da rede.	Este trabalho exalta como o desempenho é importante quando se trata do armazenamento de quantidades expressivas de dados. Além das vantagens já apresentadas pelas ferramentas SADs, é apresentado uma estratégia de armazenamento ativo para extrair mais desempenho ainda.
Evaluation of a performance model of Lustre File System – (2)	Tiezhu Zhao, Verdi March, Shoubin Dong, Simon See	Apresenta uma avaliação sobre o sistema de arquivos Lustre e propõe um modelo de desempenho relativo para prever a sobrecarga sob diferentes fatores que exercem influência no desempenho dessas ferramentas.	A avaliação abrange principalmente o número de OSSes, abordagens sobre conexões de armazenamento, o tipo de disco e o número de threads/OST. Para os testes foram utilizados pares de sistemas Lustre equipados com diferentes fatores de desempenho.	-	Os resultados mostram que com o modelo é possível obter melhor precisão na previsão e mostram que para uma determinada carga de trabalho, o desempenho de um dispositivo é o melhor predictor para o desempenho de outro.	Este trabalho reforça que o desempenho é a consequência da configuração de uma série de fatores diferentes além de definir que a avaliação do desempenho de um dispositivo é o melhor predictor para o desempenho de outro.
NFSp: a distributed NFS server for cluster of workstations – (2)	P. Lombard, Yves Denneulin	Este trabalho apresenta um servidor NFS que visa a utilização do espaço de disco não utilizado distribuído nos nós do cluster e proporcionar melhorias no desempenho e escalabilidade.	Neste trabalho é introduzido uma modificação de um clássico servidor NFS para adaptá-lo ao contexto de clusters. A proposta é dividir o servidor NFS em dois servidores menores: a servidor de metadados e servidores de dados.	O SO utilizado é o Linux Mandrake 7.1. O servidor NFS dedicado do i-cluster é equipado com P3-1GHz, 512 MiB RAM e 1 GiB swap.	-	Propõe uma versão distribuída do tradicional servidor NFS que trate de uma maneira melhor a carga gerada por múltiplos acessos concorrentes feitos pelos nós computacionais.
Performance evaluation of a prototype distributed NFS Server – (2)	Rafael Bohrer Ávila, Philippe O.A. Navaux, Yves Denneulin	São investigadas as possibilidades de um modelo de replicação para o servidor NFS que se baseia em Lasy Release Consistency (LRC).	O modelo proposto consiste em replicar o servidor NFS para múltiplas máquinas numa tentativa de aumentar a escalabilidade para um cenário de computação distribuída.	O sistema foi configurado com 12 iods, 7 meta servidores e 21 clientes (3 clientes por meta-servidor) num total de 40 nós.	Os resultados mostraram que é possível um desempenho 5 vezes maior para escrita e dobrar o desempenho de leitura(atingindo quase 100% da eficiência da rede.	Ter múltiplas instâncias do meta-servidor distribui a carga imposta pelos acessos simultâneos permitindo um ganho maior de desempenho.

Tabela 2 – Resumo dos trabalhos citados (cont).

Titulo	Autores	Proposta do estudo	Descrição	Ambiente	Resultados	Observações
Large files, small writes, and pNFS – (2)	Dean Hildebrand, Lee Ward, Peter Honeyman	O trabalho investiga o desempenho dos sistemas de arquivos paralelos com pequenas escritas. É demonstrado que os sistemas de arquivos paralelos podem aumentar a capacidade de gravação paralela dos dados independentemente do tamanho do arquivo.	Consiste na utilização do sistema de arquivos NFSp para aplicações científicas e avaliação de desempenho realizando escrita de forma paralela.	Foram utilizados vários benchmarks distintos para a realização dos testes.	Os resultados ratificam que a utilização do NFSp promove melhoria no desempenho de SAPs utilizando E/S de forma paralela para grandes requisições de escrita e para os SADs pequenas requisições de escrita.	-
Performance evaluation of parallel file systems for pc clusters and ASCI Red – (3)	Sharad Garg, Jens Mache	O trabalho avalia o desempenho de E/S do Intel PFS, um sistema de arquivos comercial para ASCI Red e o PVFS, um sistema de arquivos open-source para clusters Linux. O método utilizado explora a eficácia do PVFS no aumento do tempo de conclusão total de uma aplicação Ray Trancing	O programa de Ray Tracing computou 10 quadros de uma cena simples em 840x840 pixels cada, resultando 27 MBytes de dados que foram gravados em um número especificado de nós de E/S.	Foram utilizados 2 ambientes: - Chiba city cluster, em Argonne National Laboratory (ANL) com até 256 nós Pentium III Dual 500 MHz - PC cluster em Lewis & Clark College (LC) com até 12 nós Pentium III 450 MHz.	Os resultados mostraram que é possível reduzir o tempo de conclusão total em um fator de até 1.563 e com a utilização do PVFS é possível obter um throughput de E/S de mais de um 1Gigabyte/seg.	-
Sistemas de arquivos paralelos: Alternativas para a redução do gargalo no acesso ao sistema de arquivos – (3)	Alfredo Goldman, Roberto Pires de Carvalho	O trabalho pretende mostrar que um sistema de arquivos paralelo, no caso o PVFS2, pode ser mais rápido do que um sistema de arquivos local, mesmo no caso em que se tem apenas uma máquina como cliente.	O objetivo é propor uma comparação entre o sistema de arquivos paralelo PVFS2 e o sistema de arquivos local Ext3, usando apenas uma máquina como cliente, em uma situação que haja largura de banda suficiente na rede para o tráfego dos dados.	-	Os resultados mostram que o gargalo no acesso aos dados do disco foi resolvido com o PVFS2. Como os servidores são acessados aleatoriamente, acabam recebendo as requisições de forma balanceada, o que melhora as respostas do PVFS2. Já o Ext3 não possui um desempenho muito bom quando se tem acesso concorrente. Mesmo quando o acesso é mono-tarefa, o PVFS2 pode ser melhor, desde que a máquina cliente tenha capacidade suficiente de processamento e no acesso à rede de alta velocidade.	Com os resultados apresentados é possível perceber que o poder de processamento e transmissão de dados do cliente passa a ser o gargalo principal. Com um cliente mais potente é possível obter melhor desempenho do PVFS2, acessando o mesmo sistema de arquivos, mostrando que tanto a rede como a CPU são mais importantes que a velocidade do disco quando o objetivo é atingir alto desempenho no acesso ao sistema de arquivos.

4 *Ambientes*

O ambiente utilizado para o desenvolvimento deste estudo, bem como para a realização dos testes e colheita de métricas foram disponibilizados pela infra-estrutura do Laboratório de Telemedicina situado no Hospital Universitário de Santa Catarina.

Para a configuração do ambiente foram utilizadas seis máquinas heterogêneas que foram interconectadas por meio de uma rede Ethernet Gibabit. Para melhor visualização do ambiente, a configuração individual de cada uma das máquinas é apresentada na tabela 3 enquanto o sistema operacional e versão do kernel são apresentados na tabela 4.

Máquina	Processador	RAM	HD
1	Amd X2 Dual Core 2.1 GHz	2 Gb	180 Gb
2	Amd X2 2.8 GHz	3 Gb	250 Gb
3	Amd X2 2.8 GHz	3 Gb	250 Gb
4	Intel i5 3.2 GHz	8 Gb	320 Gb
5	Intel i5 3.2 GHz	8 Gb	320 Gb
6	Intel Dual Core 1.8 GHz	1 Gb	70 Gb

Tabela 3 - Especificações das Máquinas do Ambiente.

Máquina	Sistema Operacional
1	Ubuntu GNU/Linux, com kernel versão 2.6.32-23-generic
2	Ubuntu GNU/Linux, com kernel versão 2.6.32-24-generic
3	Ubuntu GNU/Linux, com kernel versão 2.6.32-25-generic
4	Ubuntu GNU/Linux, com kernel versão 2.6.32-24-generic
5	Ubuntu GNU/Linux, com kernel versão 2.6.32-25-generic
6	Debian GNU/Linux, com kernel versão 2.6.26-2

Tabela 4 - Sistemas Operacionais das Máquinas do Ambiente.

4.1 Parallel Virtual File System (PVFS)

Para a construção de um sistema compartilhado de arquivos utilizando o PVFS, o ambiente foi configurado com:

- Máquina 1: Servidor de metadados, que é o nodo que mantém os metadados (como permissões e carimbos do tempo).
- Máquina 1: Cliente, que é o nodo que pode ler e gravar arquivos.
- Máquinas 1-6: Seis servidores E/S (máquina 1-6) que são os nós que realmente armazenam porções de dados dos arquivos.

4.2 Fraunhofer File System (FhGFS)

O ambiente utilizando o sistema de arquivos Fraunhofer foi configurado similar ao ambiente configurado com o PVFS, com as máquinas oferecendo os seguintes serviços:

- Máquina 1: Servidor de metadados.
- Máquina 1: Cliente.
- Máquinas 1-6: Servidores de armazenamento.

4.3 Hadoop Distributed File System (HDFS)

Uma configuração típica de cluster utilizando o Hadoop consiste em um nodo mestre e múltiplos nodos escravos. O nodo mestre consiste de um namenode e de um datanode, enquanto os nodos escravos consistem apenas de um datanode.

O namenode executa operações no sistema de arquivos, como abrir, fechar e renomear arquivos e diretórios. Ele também determina o mapeamento dos blocos para os datanodes. Os datanodes são responsáveis pelo fornecimento de pedidos de leitura e escrita do sistema de arquivos dos clientes. Os datanodes também executam a criação, exclusão e replicação de blocos sob instruções do namenode.

Com isso em mente, o ambiente utilizando o sistema de arquivos Hadoop foi configurado da seguinte maneira:

- Máquina 1: Namenode (Mestre)
- Máquinas 1-6: Datanodes (Escravos)

Uma limitação do HDFS é que ele não pode ser montado diretamente por um sistema operacional existente. Obter dados dentro e fora do sistema de arquivos HDFS pode ser

inconveniente. Um sistema de arquivos em espaço de usuário (FUSE) foi desenvolvido para resolver esse problema, pelo menos para o Linux e outros sistemas Unix.

O Fuse-DFS permite que o HDFS possa ser montado como um sistema de arquivos padrão usando o comando mount. Uma vez montado, o Fuse-DFS suporta leituras, escritas e operações de diretório (por exemplo: cp, ls, more, cat, find, less, rm, mkdir, mv, rmdir). Comandos como touch, chmod, chown e permissões ainda estão sendo trabalhadas.

Algumas limitações do HDFS, tal como a impossibilidade de sobrescrever bytes de um arquivo que já tenha sido escrito, ocasionou erros internos específicos, cujos quais não foi possível ter controle, ou mesmo acesso, e isso impossibilitou que fosse feita a coleta de resultados para a avaliação desse ambiente, embora o mesmo tenha sido devidamente montado.

4.4 Sistema de Arquivos Lustre

O sistema de arquivos Lustre consiste dos seguintes componentes básicos:

- Metadata Server (MDS): Cada MDS gerencia os nomes e diretórios no sistema de arquivos Lustre e prevê o tratamento de solicitações de rede para um ou mais MDTs.
- Metadata Target (MDT): O MDT armazena os metadados (tais como nome de arquivos, diretórios, permissões e layout de arquivo) em um MDS.
- Object Storage Servers (OSS): O OSS fornece o service de E/S de arquivos, e manipulação de requisição de rede para um ou mais OSTs.
- Object Storage Target (OST): O OST armazena os dados de arquivos (pedaços de arquivos do usuário) como objetos de dados em um ou mais OSSs.
- Clientes: Os clientes são nodos computacionais, de visualização ou desktops que executam o software Lustre que lhes permite montar o sistema de arquivos Lustre.

O MDT, OSTs e o cliente podem ser executados concorrentemente em um único nodo, entretanto, uma configuração típica consiste de um MDT em um nodo dedicado, dois ou mais OSTs em cada nodo OSS e um cliente para cada nodo computacional.

Pelo fato do ambiente disponibilizado ser heterogêneo, houve uma grande dificuldade para configurar o ambiente, inviabilizando assim, a sua montagem.

Credita-se essa dificuldade ao fato de termos distribuições Linux diferentes, assim como as versões do kernel, para algumas máquinas. Embora tenhamos seguido manuais

oferecidos pela documentação e outros scripts encontrados em páginas de discussão da ferramenta na web, a tentativa de instalação do Lustre ocasionou erros que não foram possíveis de serem corrigidos pelo fato das descrições serem muito específicas para determinada distribuição e para determinada versão do kernel que não correspondiam as características apresentadas pelo ambiente disponibilizado.

Vale observar que a tentativa de instalação foi feita utilizando o source disponível na página de downloads da Oracle, e que além do source, estão disponíveis versões de download para Oracle Enterprise Linux 5, Red Hat Enterprise Linux 5 e SUSE Linux Enterprise Server. Dessa forma, uma alternativa seria configurar uma ambiente com um dos sistemas operacionais oferecidos acima e realizar a instalação, alternativa essa que não foi implementada neste trabalho.

4.5 Distributed Network File System Parallel (dNFSp)

O pNFS consiste de vários nodos E/S, que podem ser chamados de iods, que são responsáveis pelo armazenamento e recuperação dos blocos de dados resultantes do file striping e um meta servidor responsável por encaminhar as requisições recebidas para os iods mais apropriados para responder ao cliente que solicitou a tarefa (KASSICK, MACHADO, 2005).

Uma vez recebidas essas tarefas, o meta servidor encaminha as requisições para os iods, que por sua vez podem trabalhar em paralelo aumentando assim o desempenho. Os iods podem rodar nas máquinas clientes sem nenhuma restrição (KASSICK, MACHADO, 2005).

O dNFSp (Distributed NFS Parallel) foi proposto como uma variante do pNFS. O dNFSp consiste em criar réplicas do meta servidor em mais nodos do sistemas aumentando assim a largura de banda. Todavia, cada subconjunto de clientes visualiza apenas o meta servidor ao qual está associado como se fosse o único meta servidor de todo o sistema. Cada meta servidor continua capaz de visualizar todos os iods do sistema (KASSICK, MACHADO, 2005).

As mesmas dificuldades encontradas e descritas na seção anterior, adicionada a falta de documentação mais objetiva e consistente, impossibilitaram a instalação e avaliação desta ferramenta.

5 *Resultados*

Para os ambientes descritos no capítulo anterior, foi possível avaliar três ferramentas, levando em consideração as cinco que foram propostas. Das três ferramentas avaliadas, foi possível coletar métricas para duas, PVFS e FhGFS, e para a terceira ferramenta, HDFS, embora o ambiente tenha sido montado, não foi possível fazer a coleta de dados devido a limitações oferecidas pela própria ferramenta, conforme analisado anteriormente na descrição do ambiente.

Para as duas ferramentas restantes, também devido a algumas limitações das ferramentas, somado as limitações de um ambiente heterogêneo e não dedicado, não foi possível configurá-los e realizar os experimentos para que fosse possível obter dados significativos para a avaliação proposta.

Desta maneira, neste capítulo serão apresentados os resultados individuais destas duas ferramentas, assim como gráficos comparativos para tempo médio de armazenamento para cada um dos experimentos, 1, 10, 100, 1000 e 10000 imagens.

Antes de apresentar os resultados, para que haja uma validação da proposta desta pesquisa, faz-se necessário apresentar dados que realmente justifique o ganho de desempenho no uso de sistemas de arquivos distribuídos em relação ao método atualmente utilizado, que é a gravação dos dados das imagens em banco de dados relacional.

De Macedo (2008), propõe em um dos objetivos de seu trabalho, realizar um comparativo entre o armazenamento das imagens médicas utilizando sistemas de arquivos distribuídos e a gravação destas imagens em banco de dados relacional. Para a realização do experimento foi montado um agregado computacional composto por quatro nodos, cada um possuindo uma área compartilhada de disco rígido sob o PVFS.

O experimento consistiu na gravação de um conjunto de dados, composto por 2.570 imagens DICOM, ocupando um espaço total em disco de 1.019 Mb.

A tabela 5 e a figura 15 mostram o registro dos tempos, em segundos, utilizado na transferência do conjunto de imagens para o servidor do banco de dados relacional e para o agregado computacional com sistema de arquivos distribuído, PVFS (DE MACEDO, 2008).

	HDF5	Banco de Dados
1	850,663	992,463
2	840,380	988,079
3	894,973	1066,320
4	858,084	990,905
5	854,340	1043,406
6	850,273	1038,364
7	835,660	1056,084
8	883,583	1027,002
9	863,837	994,848
10	855,500	990,177

Tabela 5 - Tempos de Armazenamento.

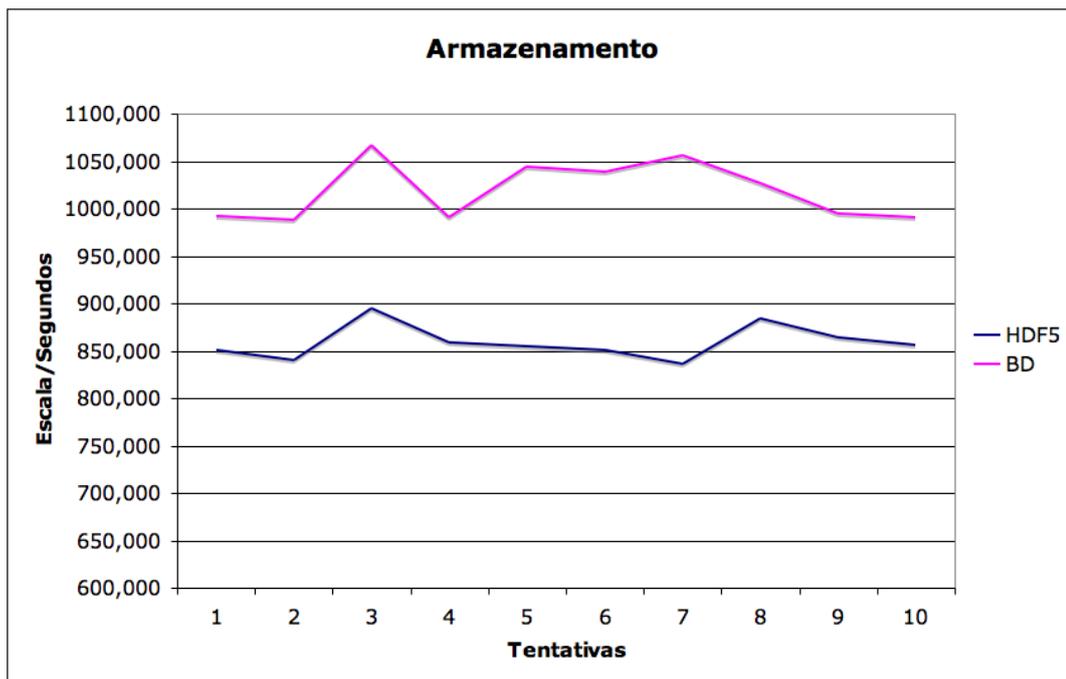


Figura 15 - Gráfico de Desempenho de Armazenamento.

Pela análise dos resultados obtidos, pode-se perceber que existe um ganho de desempenho no armazenamento dos arquivos DICOM no formato HDF5 em sistemas de arquivos distribuídos, em comparação ao armazenamento em um banco de dados relacional (DE MACEDO, 2008).

Validado o ganho de desempenho oferecido pela utilização de sistemas de arquivos distribuídos em relação ao atual método utilizado, é possível finalmente apresentar os dados obtidos pelos experimentos aqui propostos.

5.1 Resultados para o Experimento com Uma Imagem

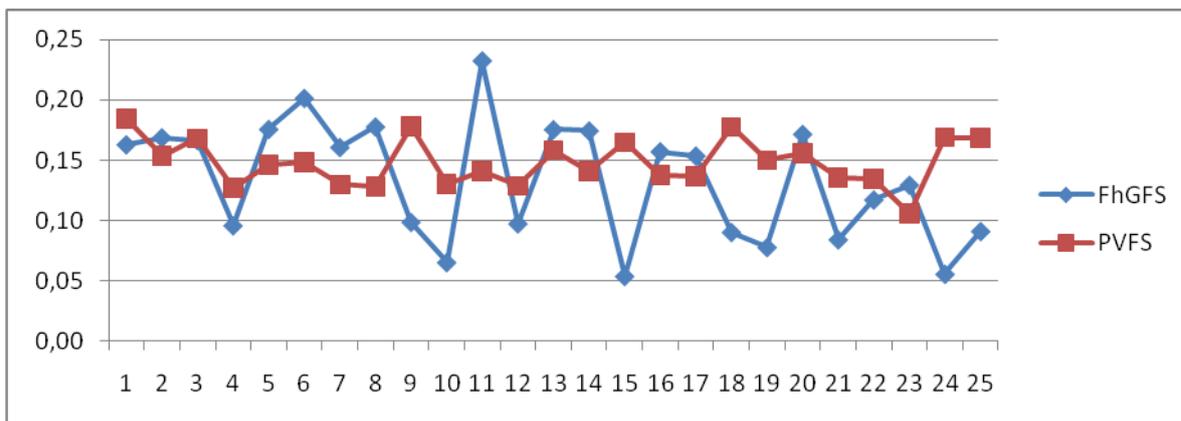


Figura 16 - Gráfico Comparativo de Desempenho de Armazenamento (Uma Imagem).

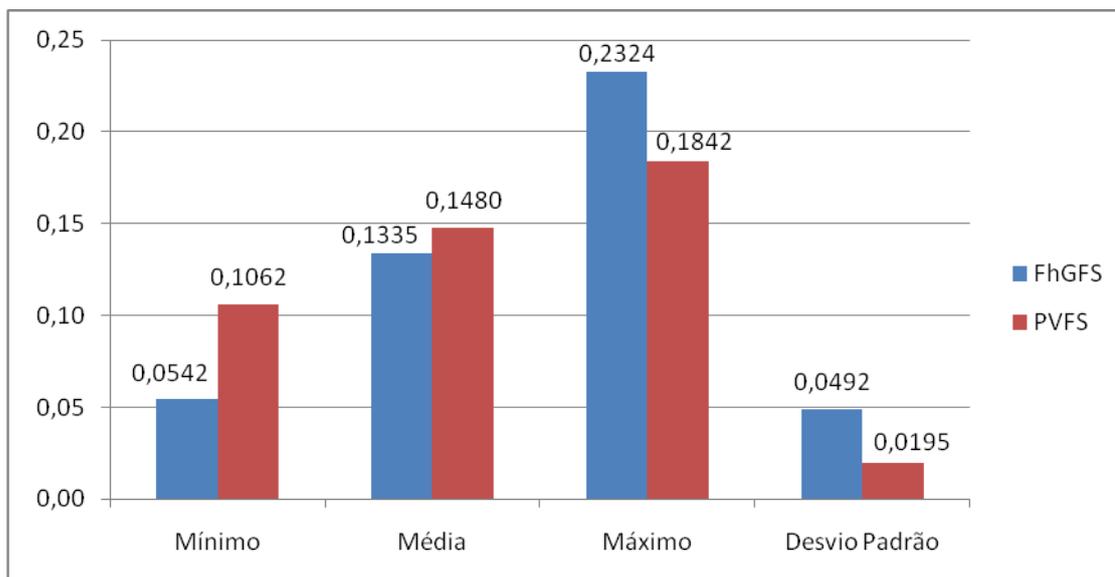


Figura 17 - Gráfico Comparativo das Métricas Obtidas (Uma Imagem).

Para o experimento realizado com uma imagem, embora o FhGFS tenha apresentado um valor médio de desempenho de armazenamento menor que o PVFS, também é possível observar que o PVFS teve menor variação nos tempos de armazenamento, o que representa uma regularidade maior na avaliação de desempenho para este experimento.

5.2 Resultados para o Experimento com Dez Imagens

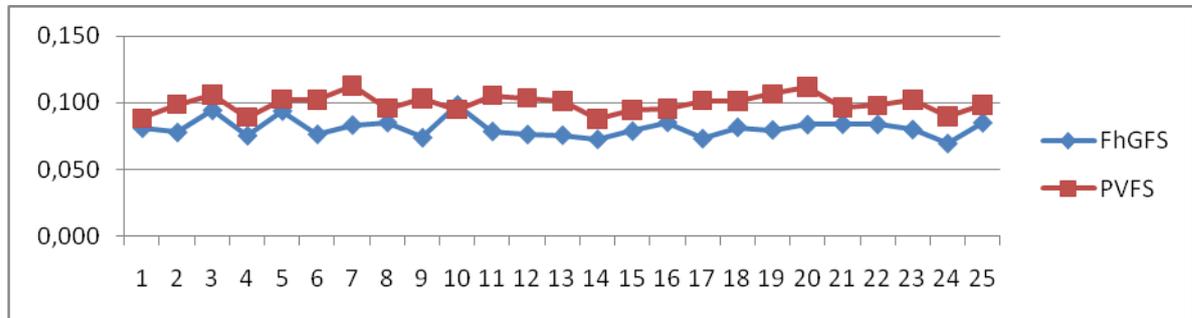


Figura 18 - Gráfico Comparativo de Desempenho de Armazenamento (Dez Imagens).

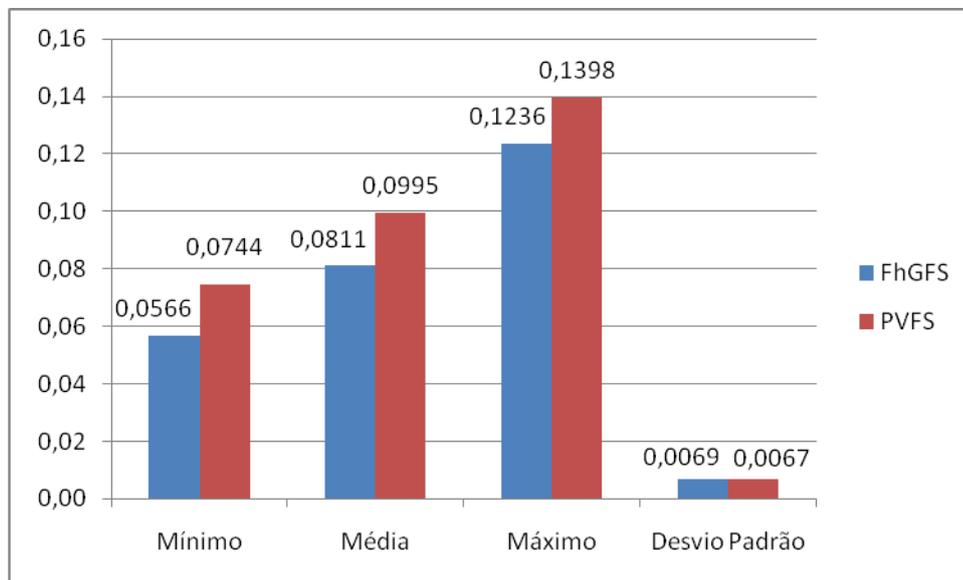


Figura 19 - Gráfico Comparativo das Métricas Obtidas (Dez Imagens).

Para o experimento realizado com dez imagens é possível perceber que novamente o FhGFS apresentou melhor desempenho em relação ao PVFS e, diferentemente do experimento realizado com uma imagem, o FhGFS apresentou menor variância nos tempos de armazenamento. Mesmo assim, embora a diferença seja mínima, o PVFS apresentou menor variância nos resultados.

Outra análise a ser feita é, dado o tempo total de armazenamento para o FhGFS e PVFS, 0,811 e 0,995 respectivamente, é possível constatar que o FhGFS obteve um ganho de desempenho de 19,5% em relação ao PVFS.

5.3 Resultados para o Experimento com Cem Imagens

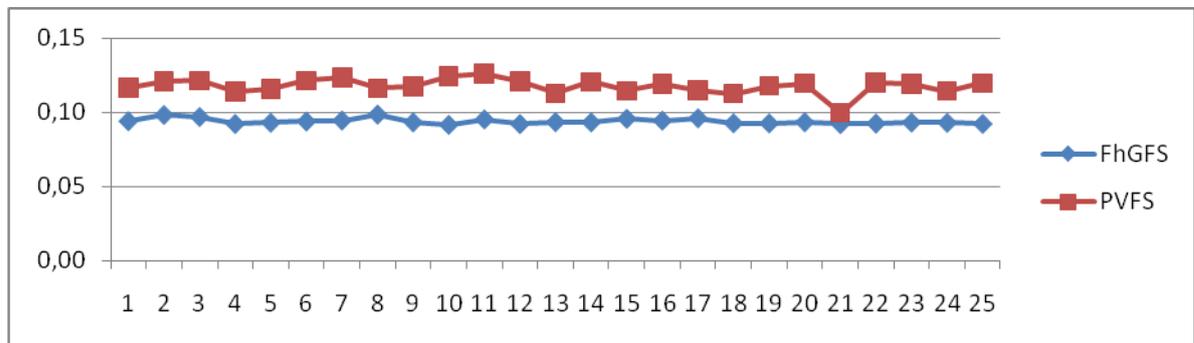


Figura 20 - Gráfico Comparativo de Desempenho de Armazenamento (Cem Imagens).

Para o experimento realizado com cem imagens, novamente o FhGFS obteve melhor desempenho médio e, diferentemente das duas classes de experimentos apresentadas anteriormente, também obteve menor variância para os tempos médios de armazenamento em relação ao PVFS, conforme ilustrado a seguir.

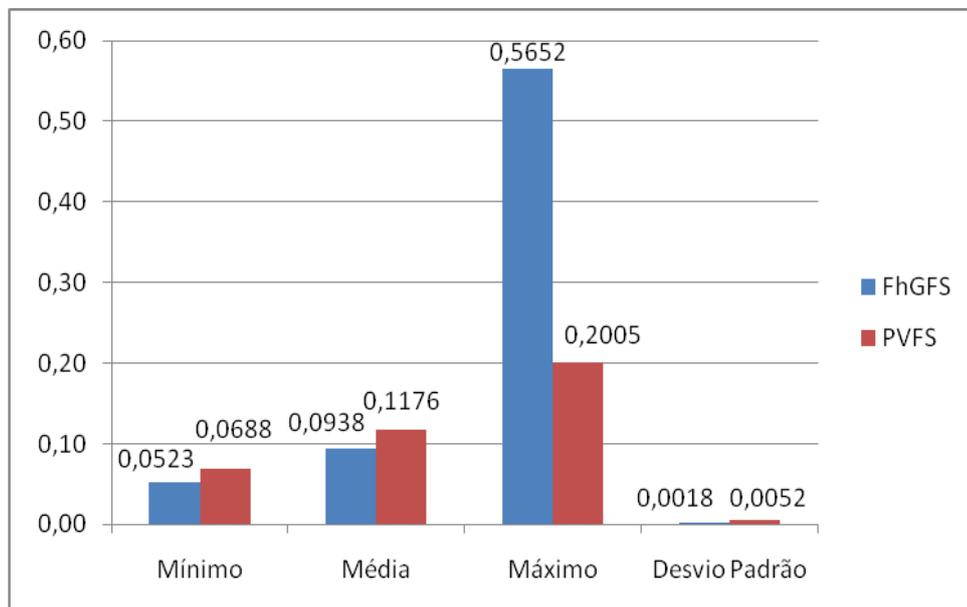


Figura 21 - Gráfico Comparativo das Métricas Obtidas (Cem Imagens).

Observando o gráfico acima é possível perceber que houve um valor máximo muito alto comparado aos experimentos anteriores, porém, é possível notar que este é um valor discrepante único dos demais, conforme ilustrado na figura 22 e, assim sendo, não altera de forma significativa o valor da média calculada.

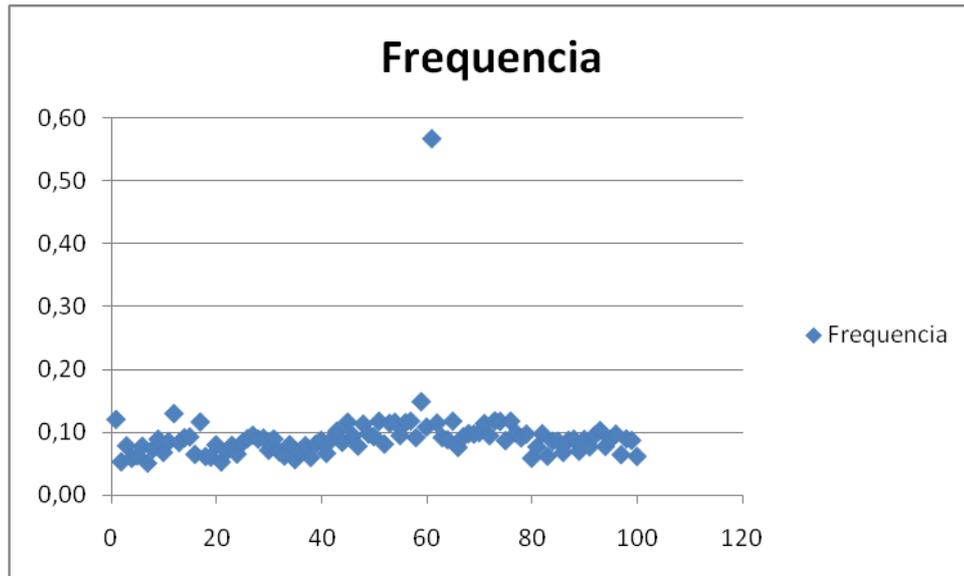


Figura 22 - Distribuição dos Tempos de Armazenamento.

Avaliando o tempo total de armazenamento das ferramentas para este experimento, 9,38 segundos para o FhGFS e 11,76 segundos para o PVFS é constatado que o FhGFS obteve ganho de aproximadamente 20,25% no desempenho em relação ao PVFS.

5.4 Resultados para o Experimento com Mil Imagens

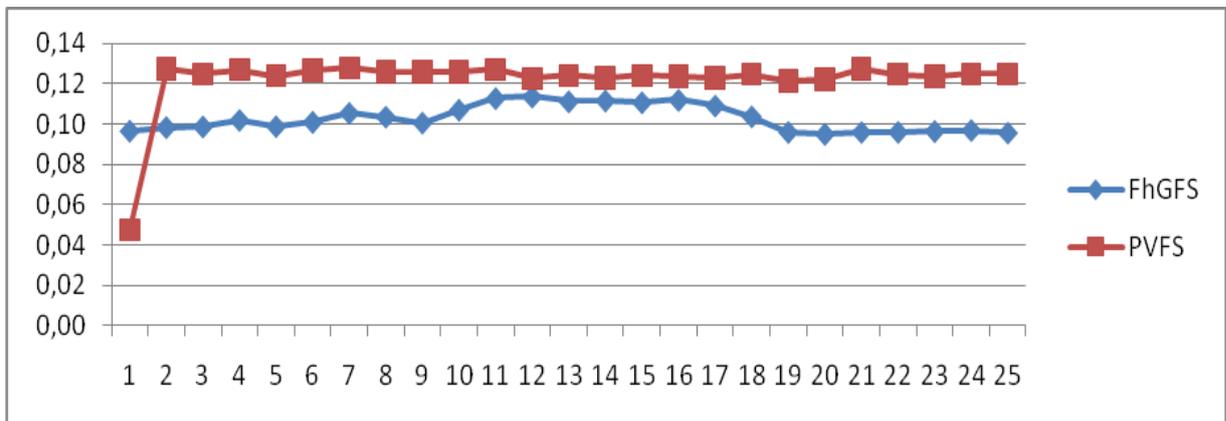


Figura 23 - Gráfico Comparativo de Desempenho de Armazenamento (Mil Imagens).

Para o experimento realizado com 1000 imagens, novamente é possível perceber que o FhGFS obteve um desempenho médio melhor que o PVFS, conforme ilustrado na figura 24, além de também obter uma variância menor para os tempos registrados.

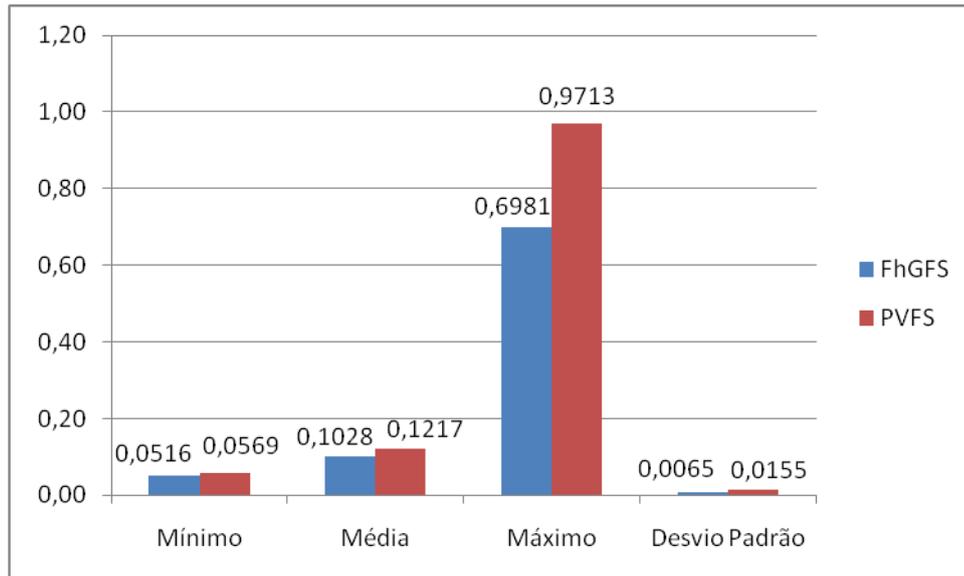


Figura 24 - Gráfico Comparativo das Métricas Obtidas (Mil Imagens).

Analisando o tempo total de armazenamento das ferramentas, 102,8 segundos para o FhGFS e 121,7 segundos para o PVFS, é possível calcular um ganho de aproximadamente 15,5% no tempo de armazenamento para o FhGFS em relação ao PVFS.

5.5 Resultados para o Experimento com Dez Mil Imagens

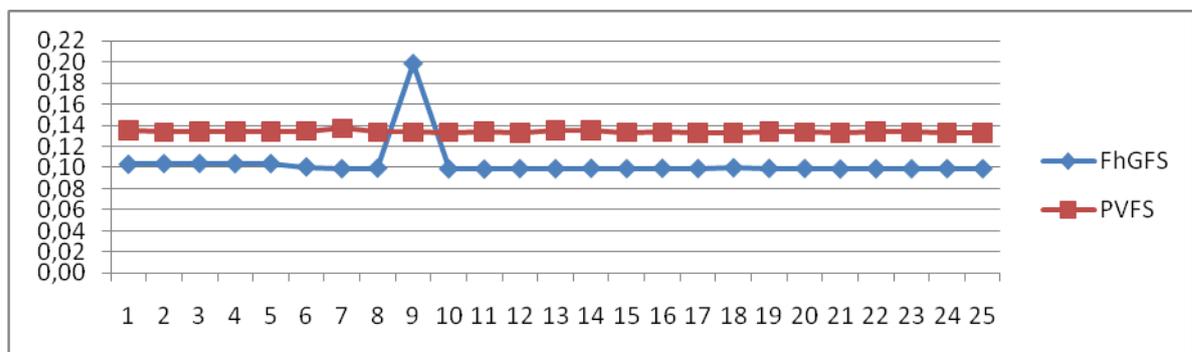


Figura 25 - Comparativo de Desempenho de Armazenamento (Dez mil Imagens).

Para o experimento realizado com dez mil imagens, o FhGFS obteve um ganho de aproximadamente 22,35% em relação ao PVFS, levando em consideração o tempo total de armazenamento, 1038,78 segundos para o FhGFS e 1337,79 segundos para o PVFS.

Para este experimento também é possível notar que o PVFS obteve menor variância para os tempos registrados.

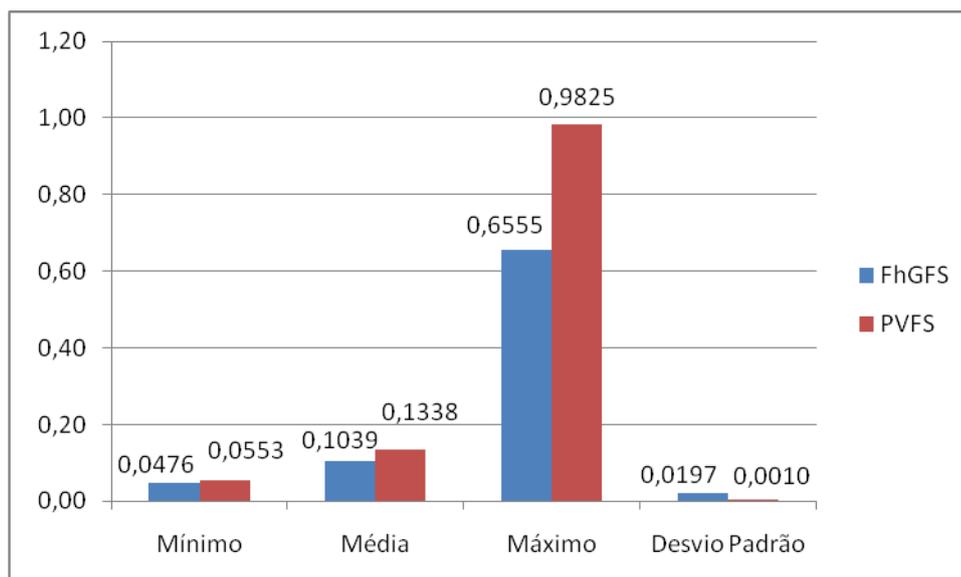


Figura 26 - Gráfico Comparativo das Métricas Obtidas (Dez mil Imagens).

Feito o comparativo para as duas ferramentas avaliadas com resultados, é ilustrado nas figuras 27 e 28 as métricas individuais de cada ferramenta, categorizadas por cada uma das classes de experimentos. A partir destas figuras é possível analisar o comportamento de cada sistema de arquivos distribuídos conforme a quantidade de imagens a serem armazenadas aumenta.

Como o FhGFS obteve melhor desempenho para todas as classes de experimentos, para melhor visualização dos resultados, a tabela 6 ilustra o tempo total de armazenamento, em segundos, para cada conjunto de imagens, assim como o ganho obtido pelo FhGFS em relação ao PVFS.

Conjunto de Imagens	FhGFS	PVFS	GANHO
1	0,1335465	0,14802569	9,8%
10	0,8107787	0,99482777	19,5%
100	9,3837995	11,7633604	20,25%
1000	102,76122	121,741186	15,5%
10000	1038,7822	1337,79269	22,35%

Tabela 6 - Ganho de Desempenho do FhGFS em relação ao PVFS.

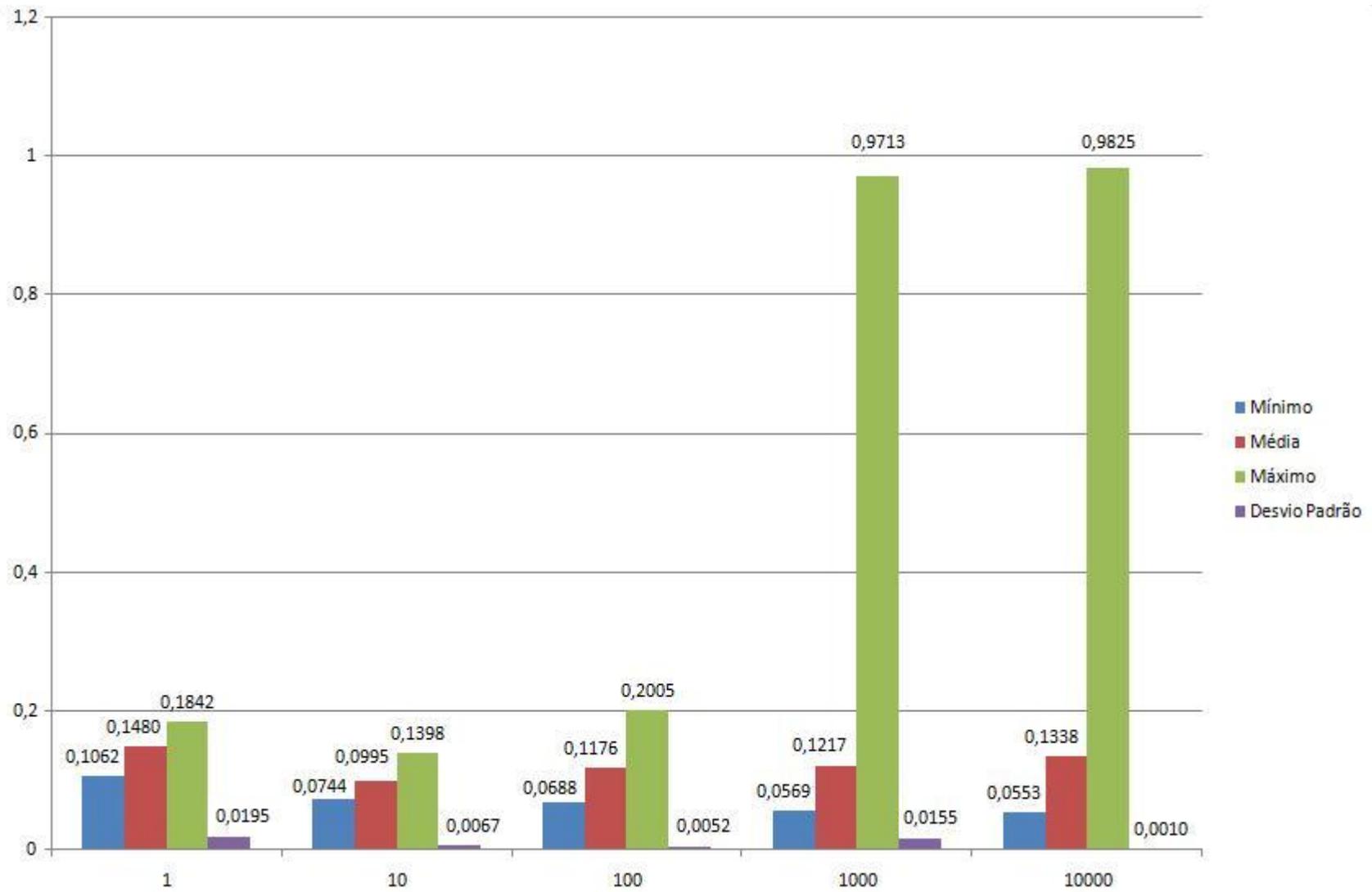


Figura 27 - Desempenho Individual do PVFS para cada classe de experimento.

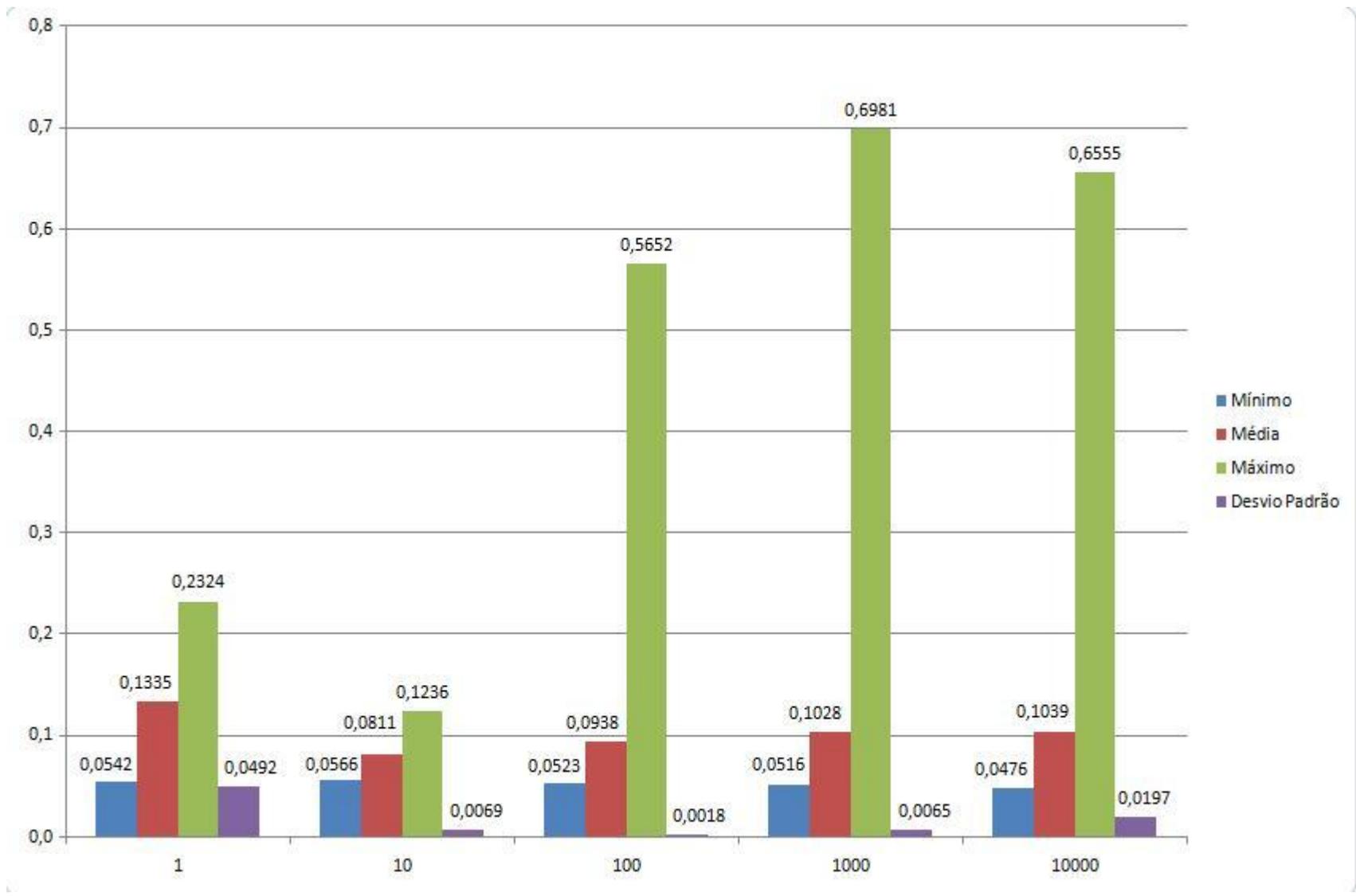


Figura 28 - Desempenho Individual do FhGFS para cada classe de experimento.

6 *Conclusões e Trabalhos Futuros*

Baseado nos resultados obtidos com os experimentos realizados, podem ser realizadas algumas observações em cima das métricas colhidas.

Para ambas ferramentas, embora o valor máximo de algumas classes de experimentos tenham sido discrepantes, a métrica de desvio padrão mostra que não houve grande variância nos valores médios de armazenamento registrados. Pelo fato de todos os experimentos terem sido realizados no mesmo ambiente, e este por sua vez tratar-se de um ambiente heterogêneo e não dedicado, não é possível afirmar com precisão os motivos que levaram a esses valores divergentes, uma vez que não se tem total controle sobre a utilização das máquinas, assim como da utilização da rede de interconexão.

Analisando a métrica relacionada a média dos tempos de armazenamento individualmente para cada ferramenta, é possível perceber que esse valor cresce conforme a quantidade de imagens a serem armazenadas também aumenta. O fato desse incremento ser mínimo para o ambiente aqui proposto, demonstra a capacidade que os sistemas de arquivos distribuídos possuem em prover desempenho no armazenamento de grandes volumes de dados. Enquanto a quantidade de dados a serem armazenados aumentam na ordem de dez vezes, os tempos médios aumentam numa taxa de 10% comparando os três primeiros experimentos e menos de 2% comparando os dois últimos.

Finalizando, conforme figuras e tabelas apresentadas no capítulo anterior, é possível afirmar que o sistema de arquivos distribuído Fraunhofer obteve melhor desempenho em relação ao PVFS, para todas as classes de experimentos aqui realizadas, consolidando-se assim, como a melhor ferramenta, entre as duas avaliadas com resultados, para o cenário descrito.

A título de trabalhos futuros, caberia a configuração de um ambiente dedicado, configurado de maneira propícia para cada ferramenta, para obtenção de resultados de ferramentas distintas que pudessem ser agregados aos já obtidos. Também seria interessante a realização de experimentos onde o número de clientes variasse, obtendo métricas para os tempos de armazenamento de múltiplas requisições sendo feitas concorrentemente.

Por fim, poderiam ser realizados experimentos com maior número de nodos, para verificar possíveis relações entre o tempo de armazenamento e quantidade de máquinas configuradas para o ambiente.

Referências

ÁVILA, R. B., NAVAU, P. O. A., LOMBARD, P., LEBRE, A., and DENNEULIN, Y. **Performance evaluation of a prototype distributed NFS server.** In J.-L. Gaudiot, M. L. Pilla, P. O. A. Navau, and S. W. Song, editors, Proc. of the 16th Symposium on Computer Architecture and High Performance Computing, pages 100-105, Foz do Iguaçu, Oct. 2004. Washington, IEEE.

BASHSHUR, R. **Telemedicine and health care.** Telemedicine Journal and e-health, v. 8, n. 1, p. 5–12, 2002.

BOITO, Francieli Z.; KASSICK, Rodrigo; NAVAU, P. O. A. **Evaluating the Performance of Lustre File System.** In: VII Workshop de Processamento Paralelo e Distribuído, 2009, Porto Alegre. Anais do VII Workshop de Processamento Paralelo e Distribuído, 2009. Disponível em: < <http://gppd.inf.ufrgs.br/wsppd/2009/apresentacoes.php> > Acesso em: 02 set 2010.

BONHO, Samira. **Sistema micro controlado para Transmissão de sinais de ECG pela Internet.** 2006. 97f. Dissertação (Mestrado em Engenharia Elétrica) – Curso de Pós-graduação em Engenharia Elétrica. Universidade Federal de Santa Catarina, Florianópolis, 2006.

CARNS, Philip H.; LIGON III, Walter B. et al. **PVFS: A Parallel File System for Linux Clusters.** In: Proceedings of the Extreme Linux Track: 4th Annual Linux Showcase and Conference, October 2000. Disponível em:<<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.7313>>. Acesso em: 10 set 2010.

DA CUNHA, Antônio Geraldo. **Dicionário Etimológico Nova Fronteira da Língua Portuguesa.** 2ª ed. Rio de Janeiro: Nova Fronteira, 1986.

DE CARVALHO, Roberto Pires. **Sistemas de Arquivos Paralelos e Distribuídos.** 2003. 75f. Exame de Qualificação de Mestrado (Mestrado em Ciências da Computação) – Curso de Pós-graduação em Ciências da Computação. Universidade de São Paulo, São Paulo, 2003. Disponível em: < <http://www.ime.usp.br/~carvalho/quali/quali.pdf> > Acesso em: 02 set 2010.

DE FARIA, Bernardo Moreira. **E-Cath: um Sistema de Telemedicina para Hemodinâmicas Utilizando Redes de Baixas Velocidades.** 2003. 83f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-graduação em Ciência da Computação. Universidade Federal de Minas Gerais, Belo Horizonte, 2003.

DE MACEDO, Douglas Dyllon J. **Um Estudo de Estratégias de Sistemas Distribuídos Aplicadas a Sistemas de Telemedicina.** 2008. 86f. Dissertação (Mestrado em Ciências da

Computação) - Curso de Pós-graduação em Ciências da Computação. Universidade Federal de Santa Catarina, Florianópolis, 2008.

DE MARTINI, Richard Augusto Schafer. **Um Modelo de Gerenciamento de Rede PACS para o Sistema Catarinense de Telemedicina**. 2008. Trabalho de conclusão de curso em Ciências da Computação. Universidade Federal de Santa Catarina, Florianópolis, 2008.

DELLANI, P. R. **Desenvolvimento de um Servidor de Imagens Médicas Digitais no PadrãoDICOM**. Universidade Federal de Santa Catarina, 2005. Dissertação de Mestrado.

DELMERICO, Jeffrey A.; BYRNES, Nathaniel A.; BRUNO, Andrew E.; et al. **Comparing the Performance of Clusters, Hadoop, and Active Disks on Microarray Correlation Computations**. Department of Computer Science and Engineering, University at Buffalo. Universidade de Buffalo. 2009. Disponível em: www.cse.buffalo.edu/~vipin/papers/2009/2009_9.pdf Acesso em: 23 set 2010.

DOS SANTOS, Elias Amaral. **Um Estudo Aplicado de Armazenamento de Imagens Médicas em Ambientes Distribuídos**. 2009. 110f. Dissertação (Curso de Mestrado Acadêmico em Computação Aplicada) - Curso de Pós-graduação em Ciências da Computação. Universidade do Vale do Itajaí, Florianópolis, 2009.

FERREIRA, Geraldo Antônio. **Aplicações MIDP em aparelhos móveis celulares e monitoramento remoto de bio-sinais: considerações e desenvolvimento de uma solução**. 2005. 89f. Dissertação (Mestrado em Ciências da Computação) – Curso de Pós-graduação em Ciências da Computação. Universidade Federal de Minas Gerais, Belo Horizonte, 2005.

FUHRMANN, P. **A Perfectly Normal Namespace for the DESY Open Storage Manager**. Conf. on Computing in High Energy Physics, Berlin, 1997.

GARG, Sharad; MACHE, Jens. **Performance Evaluation of Parallel File Systems for PC Clusters and ASCII Red**. In: IEEE International Symposium on Cluster Computing, 2001, Newport Beach, CA. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/CLUSTR.2001.959973>>. Acesso em: 06 out 2010.

GOLDMAN, Alfredo; DE CARVALHO, Roberto Pires. **Sistemas de Arquivos Paralelos - Alternativas para a redução do gargalo no acesso ao sistema de arquivos**. 2005. Dissertação (Mestrado em Ciências da Computação) - Universidade de São Paulo, . *Orientador:* Alfredo Goldman vel Lejbman. Disponível em: <http://www.ime.usp.br/~carvalho/defesa/defesa.pdf>> Acesso em: 02 set 2010.

HILDEBRAND, Dean; WARD, Lee; HONEYMAN, Peter. **Large files, small writes, and pNFS**. In: INTERNACIONAL CONFERENCE ON SUPERCOMPUTING, 20., 2006, Cairns, Queensland, Australia. Proceedings... . New York, Ny, Usa: Acm, 2006. p. 116 – 124.

HU, Lei; GORTON, Ian. **Performance Evaluation for Parallel Systems: A Survey**. Pandora electronic collection, Sydney, 1997. Disponível em: <<ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/9707.pdf>>. Acesso em: 02 set 2010.

Institut Techno-und Wirtschaftsmathematik. **Fraunhofer Parallel File System**. HPC wire Editors' Choice Awards 2009. Disponível em: <http://www.itwm.fraunhofer.de/hpc/flyer/hpc_flyer_FhGFSFraunhofer_FileSystem_ISC2009.pdf>. Acesso em: 02 set 2010.

Institut Techno-und Wirtschaftsmathematik. **User Guide**. FraunhoferFS. 2009. Disponível em: <http://www.fhgfs.com/docs/FraunhoferFS_User_Guide.pdf> Acesso em: 02 set 2010.

KASSICK, Rodrigo ; BOITO, Francieli Z. ; NAVAUUX, P. O. A. **Testing the Performance of Parallel File Systems**. In: VI Workshop de Processamento Paralelo e Distribuído, 2008. Porto Alegre. Anais do VI Workshop de Processamento Paralelo e Distribuído, 2008. Disponível em: <<http://gppd.inf.ufrgs.br/wsppd/2008/papers/Kassick.pdf>>. Acesso em: 10 set 2010.

KASSICK, Rodrigo Virote; MACHADO, Caciano dos Santos; HERMANN, Everton; ÁVILA et al. **Evaluating the Performance of the dNFSP File System**. In: IEEE International Symposium on Cluster Computing and the Grid, 5., 2005, Cardiff, UK. Proceedings... Los Alamitos: IEEE Computer Society, 2005. Disponível em: <<http://gppd.inf.ufrgs.br/new/papers/Kassick05CCGrid.pdf>>. Acesso em: 14 set 2010.

LAYTON, Jeff. **Benchmarking Parallel File Systems**. Cluster Monkey. 2006. Disponível em: <<http://www.clustermonkey.net//content/view/62/28/>>. Acesso em: 15 set 2010.

LOMBARD, P.; DENNEULIN, Y. **NFSP: a distributed NFS server for clusters of workstations**. In: *Proc. of the 16th International Parallel & Distributed Processing Symposium, IPDPS*. Ft. Lauderdale, Florida, USA: Los Alamitos, IEEE Computer Society, 2002. p. 35. Abstract only, full paper available in CD-ROM.

MACHE, J.; BROADHURST R.; ELY J.; **Ray tracing on cluster computers**. In *Proceedings of the International Workshop on Cluster Computing - Technologies, Environments, and Applications (CC-TEA)*, 2000.

MADHYASTHA, Tara M., REED, Daniel A., **Input/output access pattern classification using hidden markov models**, in *In Proceedings of the Fifth Workshop on Input/Output in Parallel and Distributed Systems*. 1997, pp. 57–67, ACM Press.

MADHYASTHA, Tara M., REED, Daniel A., **Learning to classify parallel input/output access patterns**. IEEE Trans. Parallel Distrib. Syst., vol. 13, no. 8, pp. 802–813, 2002.

MAGNUS, Marcone Schardosim. **Alternativa de Armazenamento de Imagens Médicas com Alto Desempenho**. 2010. Trabalho de conclusão de curso em Ciências da Computação. Universidade Federal de Santa Catarina, Florianópolis, 2010.

MAIA, Rafael Simon. **Um sistema de telemedicina de baixo custo em larga escala**. 2006. 90f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-graduação em Ciência da Computação. Universidade Federal de Santa Catarina, Florianópolis, 2006.

Michaelis: Moderno Dicionário da Língua Portuguesa. São Paulo: Companhia Melhoramentos, 1998.

MILDENBERGER, P.; EICHELBERG, M.; MARTIN, E. **Introduction to the DICOM standard.** European radiology, v. 12, n. 4, p. 920, 2002.

O'MALLEY O.; MURTHY, A. C.. **Hadoop Sorts a Petabyte in 16.25 Hours and a Terabyte in 62 Seconds.** May 2009. Disponível em: <http://developer.yahoo.net/blogs/hadoop/2009/05/hadoop_sorts_a_petabyte_in_162.html>. Acesso em: 06 out 2010.

Oracle. **Lustre™ 2.0 Operations Manual.** Sun Oracle, 2010. Disponível em: <<http://wiki.lustre.org/images/3/35/821-2076-10.pdf>> Acesso em: 02 set 2010.

PIERNAS, Juan; NIEPLOCHA, Jarek; FELIX, Evan J. **Evaluation of Active Storage Strategies for the Lustre Parallel File System.** In: *CiteSeerX Scientific Literature Digital Library and Search Engine*, 2007, Nevada, USA. Disponível em: <<http://sc07.supercomputing.org/schedule/pdf/pap287.pdf>> Acesso em: 06 out 2010.

PVFS Project. **Parallel Virtual File System, Version 2.** PVFS2 Development Team, 2003. Disponível em: <<http://www.pvfs.org/documentation/>> . Acesso em: 14 set 2010.

SAVARIS, Alexandre; ANDRADE, Rafael; DE MACEDO, Douglas D. J. et al. **O Uso da Telemedicina Assistencial Assíncrona em Larga Escala no Setor Público de Saúde.** Anais e Programação do CBIS'2008. Disponível: <www.sbis.org.br/cbis11/arquivos/734.pdf>. Acesso em: 10 set. 2010.

SEBEPOU, Zoe; MAGOUTIS, Kostas; MARAZAKIS, Manolis et al. **A Comparative Experimental Study of Parallel File Systems for Large-Scale Data Processing.** In: Institute of Computer Science (ICS). Foundation for Research and Technology - Hellas (FORTH), 2008, Boston, MA. Disponível em: <http://www.usenix.org/event/lasco08/tech/full_papers/sebepou/sebepou.pdf> . Acesso em: 04 out 2010.

SHAFER, Jeffrey; RIXNER, Scott; COX, Alan L. **The Hadoop Distributed Filesystem: Balancing Portability and Performance.** IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2010), White Plains, Nova Iorque. 2010. Disponível em: <http://www.jeffshafer.com/publications/papers/shafer_ispass10.pdf/view> Acesso em: 02 set 2010.

SHEN, Xiaohui; CHOUDHARY, Alok. **DPFS: A Distributed Parallel File System.** International Conference on Parallel Processing, Washington, 2001. Disponível em: <<http://www.computer.org/portal/web/csdl/doi/10.1109/ICPP.2001.952101>> Acesso em: 02 set 2010.

SHVACHKO, Konstantin; KUANG, Hairong; RADIA, Sanjay; et al. **The Hadoop Distributed File System.** IEEE 26 th Symposium. 2010. Disponível em:

<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5488875%2F5496967%2F05496972.pdf%3Farnumber%3D5496972&authDecision=-203> Acesso em: 23 set 2010.

SMIRNI, Evgenia; REED, Daniel A., **Workload characterization of input/output intensive parallel applications**, in Proceedings of the 9th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation. 1997, pp. 169–180, Springer-Verlag.

TANENBAUM, A. S.; VAN STEEN, M. **Distributed systems: principles and paradigms**. Prentice Hall, 2002.

The Apache Software Foundation. HDFS Architecture Guide. HDFS Project. Disponível em: < http://hadoop.apache.org/hdfs/docs/current/hdfs_design.html>. Acesso em: 14 set 2010.

ZHAO, Tiezhu; MARCH, Verdi; DONG, Shoubin et al. **Evaluation of A Performance Model of Lustre File System**. In: The Fifth Annual ChinaGrid Conference. 2010. Disponível em: <<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5562735%2F5562862%2F05562884.pdf%3Farnumber%3D5562884&authDecision=-203>>. Acesso em: 07 out 2010.