UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA E GESTÃO DO CONHECIMENTO

Douglas Dyllon J. de Macedo

Um Estudo de Estratégias de Sistemas Distribuídos Aplicadas a Sistemas de Telemedicina

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Engenharia e Gestão do Conhecimento.

Prof. Dr. Luis Fernando Jacinto Maia

Prof. Dr. Mario Antonio Ribeiro Dantas

Florianópolis, Fevereiro de 2008

Um Estudo de Estratégias de Sistemas Distribuídos Aplicadas a Sistemas de Telemedicina

Douglas Dyllon J. de Macedo

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia do Conhecimento, área de concentração Sistemas Distribuídos e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento.

Prof. Dr. Roberto Pacheco

Banca Examinadora

Prof. Dr. Luis Fernando Jacinto Maia

Prof. Dr. Mario Antonio Ribeiro Dantas

Prof. Dr. Renato Fileto

Prof. Dr. Eros Comunello

Prof. Dr. Martius Vicente Rodriguez y Rodriguez

“Perfection is not achieved when there is nothing left to add, but when there is nothing left to take away”
Antoine de St. Exupery

“Para às mulheres da minha vida”

Agradecimentos

Agradeço primeiramente a Deus, por iluminar-me e abençoar-me em todos os momentos de minha vida.

Aos meus pais, pelo amor, pelo carinho incondicional, pela dedicação, pelo apoio e pela presença em todos os momentos de minha vida. Às minhas amadas irmãs, por sempre acreditarem em mim e pela credibilidade sempre concedida. À Josiane Cordeiro, pelo amor, pela confiança, paciência e pelo apoio sempre dado quando precisei.

Ao meu orientador Prof. Dr. Luis Maia, por seu humor sempre contagiante e inteligente e pelas lições de vida ensinadas a mim em todos estes anos de convívio.

Ao meu co-orientador Prof. Dr. Mario Dantas, por seus ensinamentos, pela dedicação, disponibilidade e pelas conversas sempre animadas, incentivadoras e altamente técnicas.

Ao Projeto Cyclops, em nome do Prof. Dr. rer.nat. Aldo von Wangenheim e Prof. Dr. rer.nat. Eros Comunello, por proporcionarem o ambiente propício e o apoio financeiro necessário para a elaboração deste trabalho.

Aos amigos do Laboratório de Telemedicina da UFSC: Rafael Andrade, Jader Wallauer, Harley Wagner, Rodrigo Pavezi, Cloves Barcelos, Luiz Tomazella, Rodrigo Cabral, Marcone Magnus e a todos os outros, pelos momentos de descontração e companheirismo nestes anos de projeto.

Ao meu obstinado e competente “PS” Hilton W. G. Perantunes (o *Paris*) pelo comprometimento total com o trabalho. E a Thiago Coelho Prado (o *Coeio*), pela ajuda fundamental na reta final do trabalho.

E a todos aqueles que ajudaram direta ou indiretamente na concepção e elaboração deste trabalho, meus mais sinceros agradecimentos.

MUITO OBRIGADO!!

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Desde o surgimento da Telemedicina na década de 60, ela vem sendo sugerida e aplicada como uma forma de prover acesso a saúde das pessoas, que estão isoladas dos grandes centros médicos. A sua disseminação e popularização trouxe alguns desafios no que tange à disponibilidade e escalabilidade dos sistemas que as suportam e, conseqüentemente, do conhecimento embutido em seus bancos de dados. Desta forma, este trabalho se propõe a realizar um estudo de estratégias de sistemas distribuídos aplicadas aos sistemas de telemedicina. Foram desenvolvidos três estudos, baseados em cenários de aplicação. São eles: agregados computacionais, bancos de dados distribuídos e sistemas de arquivos distribuídos. No primeiro cenário proposto, foi realizado um estudo para o provimento de alta disponibilidade e alto desempenho, usando agregados computacionais, para o sistema de telemedicina do Laboratório de Telemedicina da Universidade Federal de Santa Catarina. Neste cenário foram avaliados e comparados o sistema atual *versus* o sistema utilizando agregados computacionais. No segundo cenário foi avaliado o uso de replicação de dados assíncrona entre bancos de dados distribuídos, no sentido de integrar o conhecimento embutido nos bancos de dados médicos. Por fim, no terceiro e último cenário, foi avaliado o uso de sistemas de arquivos distribuídos como dispositivo de armazenamento de imagens médicas, no sentido de promover altos níveis de escalabilidade e desempenho para as informações.

Palavras-chave: Sistemas Distribuídos, Telemedicina, Agregados Computacionais, Bancos de Dados Distribuídos, Sistemas de Arquivos Distribuídos.

Abstract

Since the coming of Telemedicine, in the decade of 1960, it has been suggested and applied as a mean to provide access to the health of people that by other means, are isolated from large medical centers. Its spread and increasing popularity brought some challenges on the availability and scalability of the supporting systems and, in consequence, of the embedded knowledge on its databases. Thus, this work proposes to perform a study on strategies of the appliance of distributed systems on the referred telemedicine systems. Three studies will be developed, based on application scenarios: cluster computing, distributed databases and distributed file systems. On the first one, a study will be performed on providing high availability and performance, using cluster computing for the referred telemedicine system. On this scenario, will be assessed and compared the current system versus a system using cluster computing on telemedicine. On the second scenario will be assessed the use of asynchronous data replication on medical databases. Finally, on the third scenario, will be assessed the use of distributed file systems as a storage device for medical images, on with the objective of promoting higher levels of scalability and performance to the access of information.

Keywords: Distributed Systems, Telemedicine, Cluster Computing, Distributed Databases, Distributed File Systems.

Capítulo 1

Introdução

Desde o surgimento da Telemedicina, na década de 60 [BAS 02], ela vem sendo sugerida e aplicada como uma forma de prover acesso à saúde das pessoas ou de comunidades que, de uma forma ou de outra, estejam isoladas ou desprovidas de pessoal médico qualificado. Sua utilização nestes casos pode reduzir o custo de transporte e maximizar a utilização do parque tecnológico instalado em hospitais e clinicas médicas [MCN 02].

No decorrer destes anos, a Telemedicina e conseqüentemente os Sistemas de Telemedicina evoluíram e vem se disseminando por muitos países. Os principais fatores que culminaram neste desenvolvimento foram a sofisticação das redes de comunicação, em conjunto com os protocolos de rede, sistemas operacionais, equipamentos de nova geração para imagens médicas, entre outros fatores.

Porém ainda existem muitos desafios que cerceiam este crescimento, que vão desde melhores regulações e normas de boas práticas na medicina até melhores infra–estruturas para suportar sua distribuição em escala. Atualmente, a maioria dos sistemas de telemedicina em operação se encontra dentro de hospitais e clínicas. Visando uma amplicação destes serviços para fora das barreiras dos hospitais, existem alguns desafios tecnológicos ainda a serem resolvidos.

Neste âmbito, a Engenharia do Conhecimento tem papel fundamental, pois sua principal função dentro das organizações é a codificação do conhecimento, neste caso, o conhecimento médico embutido nestes sistemas. Entretanto, para fornecer modelos, métodos e técnicas para a codificação do conhecimento médico, a Engenharia do Conhecimento necessita que haja tecnologia envolvida.

Neste trabalho, serão utilizadas estratégias de Sistemas Distribuídos agregadas a Sistemas de Telemedicina, visando prover a eles: maior escalabilidade, prevendo um crescimento

futuro; maior interoperabilidade, para que as informações e processos entre os sistemas de telemedicina sejam naturais; maior desempenho, pois devido ao crescimento, a agilidade e rapidez são fundamentais; maior integração entre os sistemas, para se obter históricos de pacientes consolidados, evitando assim gastos desnecessários.

Para isto, serão estudados o desenvolvimento de cenários baseados em: Agregados Computacionais (*Cluster Computing*) para prover alta disponibilidade e alto desempenho aos sistemas de telemedicina, Replicação de Dados entre Bancos de Dados Distribuídos (*Distributed Databases*) para prover a interoperabilidade e integração necessárias a estes sistemas e, por fim, Sistemas de Arquivos Distribuídos (*Distributed File Systems*) para distribuição de imagens médicas no formato HDF5.

1.1 Motivação

Devido à tendência dos sistemas de telemedicina no Brasil e no mundo se disseminarem, um estudo profundo sobre as possibilidades e os desafios da sua ampliação corresponde um desafio para a área da Engenharia do Conhecimento, no que diz respeito ao provimento de modelos, métodos e técnicas para a codificação do conhecimento embutido neste tipo de sistema. Diante disto, detectou-se uma oportunidade de elaborar estudos baseados em cenários para aplicação de estratégias de sistemas distribuídos em sistemas de telemedicina.

1.2 Objetivos

1.2.1 Objetivo Geral

Realizar um estudo para aplicação de estratégias de sistemas distribuídos, sendo elas: agregados computacionais, bancos de dados distribuídos e sistemas de arquivos distribuídos em sistemas de telemedicina, avaliando assim sua viabilidade.

1.2.2 Objetivos Específicos

- Implementar uma estrutura de agregados computacionais em sistemas de telemedicina, procurando prover uma maior disponibilidade, escalabilidade e desempenho;
- Utilizar técnicas de replicação assíncrona em bancos de dados médicos distribuídos, visando

integrar o conhecimento médico embutido nos bancos de dados;

- Implementar uma hierarquização de dados para imagens DICOM no formato HDF5 e avaliar sua utilização em sistemas de arquivos distribuídos, em contrapartida com o uso de bancos de dados;

1.3 Multidisciplinaridade da Proposta

Este trabalho está caracterizado pela união de várias metodologias de diversas áreas de conhecimento. A multidisciplinaridade do tema apresentado engloba aspectos da área de ciências da computação, e de suas sub-áreas, tais como: sistemas distribuídos, redes de computadores, sistemas de informação, bancos de dados distribuídos, agregados computacionais, sistemas de arquivos distribuídos; e em áreas correlatas, tais como: gestão do conhecimento médico, engenharia do conhecimento, integração de informações e interoperabilidade entre bancos de dados médicos.

A gestão do conhecimento médico será tema fundamental nesta pesquisa, pelo fato que todo o esforço a ser aplicado será em prol do melhor gerenciamento, armazenamento e disseminação das informações médicas. A interoperabilidade e integração do conhecimento médico que esta armazenado nos sistema de telemedicina e nos bancos de dados distribuídos caracteriza um forte cenário multidisciplinar, onde as entidades e instituições envolvidas trabalharão em conjunto para solucionar os problemas.

Todo este envolvimento multidisciplinar, tanto das entidades, quanto das áreas de concentração que serão abordadas, além de ser imprescindível à execução do trabalho, dará oportunidade à equipe de pesquisadores de ampliar a visão da realidade e do potencial dos sistemas distribuídos integrados aos sistemas de telemedicina.

1.4 O Projeto Cyclops

O Projeto Cyclops [CYC 08] é um grupo de pesquisas e desenvolvimento direcionado para a informática em saúde, criado em 1997 através de uma proposta de projeto aprovada pelo programa de cooperação internacional CNPq/GMD/DLR German-Brazilian Cooperation Programme on Information Technology, entitulada: The Cyclops Project. O projeto Cyclops é coordenado pelo professor Dr. rer. nat. Aldo von Wangenheim, do Departamento de Informática e

Estatística (INE) da Universidade Federal de Santa Catarina (UFSC).

Desde a sua criação o projeto tem encaminhado suas pesquisas para a área de informática médica, tendo participado de diversos projetos em nível estadual, nacional e internacional. O Cyclops desenvolve tecnologias de software para os mais variados domínios de aplicação de Sistemas de Informação Hospitalar, principalmente nas áreas de Telemedicina, Prontuário Eletrônico de Pacientes (PEP), Sistemas de Arquivamento e Comunicação de Sinais e Imagens Médicas, Planejamento Cirúrgico e Suporte Automatizado para Diagnóstico por Imagem.

Atualmente, o Cyclops é o principal centro de pesquisa, desenvolvimento e aplicação de tecnologias de informação para a área da saúde dentro do Hospital Universitário, sendo tido como referência na área de Telemedicina no Brasil e colocando o Estado de Santa Catarina num patamar de destaque. Para isso o projeto conta com uma equipe multidisciplinar, composta por: médicos, dentistas, enfermeiros, engenheiros e gestores. O projeto atualmente envolve mais de 80 pessoas envolvidas.

Este trabalho foi desenvolvido no âmbito do Projeto Cyclops, utilizando dois serviços desenvolvidos pelo projeto. São eles: Portal de Telemedicina da Secretária de Saúde de Santa Catarina e o CyclopsDCMServer, que é o servidor de imagens médicas DICOM (CyclopsDCMServer) utilizado na infra–estrutura do Portal de Telemedicina.

1.5 Estrutura do Trabalho

Este capítulo apresentou uma introdução ao tema, descrevendo os objetivos propostos, a multidisciplinaridade da proposta e o âmbito de aplicação do trabalho.

Capítulo 2 – Sistemas Distribuídos

O capítulo 2 são apresentados todos os conceitos relativos aos sistemas distribuídos. Descreve aspectos técnicos de Agregados Computacionais (*Cluster Computing*), Bancos de Dados Distribuídos (*Distributed Databases*), Sistemas de Arquivos Distribuídos (*Distributed File Systems*) e o Hierarchical Data Format (HDF5), visando embasar os estudos apresentados no Capítulo 4.

Capítulo 3 – Sistemas de Telemedicina

O capítulo 3 apresenta os conceitos de Sistemas de Telemedicina. Apresenta ainda uma breve descrição do tema de Telemedicina e dos projetos e aplicações no Brasil e no mundo, descrevendo o padrão DICOM e o servidor de imagens médicas CyclopsDCMServer. Também apresenta o ambiente de aplicação, o Portal de Telemedicina.

Capítulo 4 – Estudos Desenvolvidos

O capítulo 4 apresenta os estudos desenvolvidos, baseados em cenários de aplicação. São implantandos três cenários, integrando sistemas de telemedicina com estratégias de sistemas distribuídos, sendo estes: agregados computacionais, bancos de dados distribuídos e sistemas de arquivos distribuídos.

Capítulo 5 – Conclusão e Trabalhos Futuros

O capítulo 5 apresenta a conclusão dos estudos realizados nos cenários propostos. A título de trabalhos futuros, também são relacionadas as possíveis direções que o projeto pode tomar, bem como os trabalhos que ainda serão desenvolvidos.

Capítulo 2

Sistemas Distribuídos

2.1 Introducão

Neste capítulo serão abordados o estado da arte, os conceitos e as aplicações dos Sistemas Distribuídos (*Distributed Systems*), visando embasar os estudos propostos no capítulo 4. São abordados os temas de Agregados Computacionais (*Cluster Computing*), Bancos de Dados Distribuídos (*Distributed Databases*), Sistemas de Arquivos Distribuídos (*Distributed File Systems*) e por fim, o Formato de Dados Hierárquico, HDF5 (*Hierarchical Data Format 5*).

2.2 Sistemas Distribuídos

O desenvolvimento tecnológico dos microprocessadores na década de 80, em conjunto com o alto grau de sofisticação que os protocolos e as redes de comunicação em geral atingiram, foram os dois adventos principais que culminaram no surgimento de uma nova tecnologia, a dos Sistemas Distribuídos (*Distributed Systems*).

De acordo com Coulouris *et al.* [COU 05b] um sistema distribuído pode ser definido como sendo aquele no qual os componentes de hardware ou software, localizados em computadores interligados em uma rede, se comunicam e coordenam suas ações apenas enviando mensagens entre si. Essa simples definição cobre todo o conjunto de sistemas computacionais, nos quais computadores ou outros dispositivos de comunicação, ligados em rede, podem ser distribuídos de maneira útil, compartilhando recursos e informações.

Um sistema distribuído pode ser definido também como um conjunto de computadores independentes que se apresentam a seus usuários como um sistema único e coerente

[TAN 02]. Essa definição tem vários pontos importantes a serem considerados. O primeiro deles é que um sistema distribuído consiste em componentes (computadores, celulares, sensores, etc.) autônomos. Um segundo aspecto a ser considerado é que os usuários, sejam pessoas ou programas, devem ter a impressão que estão tratando com um único sistema. Isso significa que, de um modo ou de outro, os componentes autônomos precisam cooperar.

Como estabelecer essa colaboração entre os componentes é o cerne do desenvolvimento de sistemas distribuídos. É importante observar que nenhuma premissa é adotada em relação ao tipo de computadores. Em princípio, até mesmo dentro de um único sistema, eles poderiam variar desde computadores centrais (*mainframes*) de alto desempenho até pequenos nós em redes de sensores. Dá mesma maneira, nenhuma premissa é adotada quanto ao modo como os computadores são interconectados [TAN 02].

Na figura 2.1, pode-se visualizar um cenário clássico da aplicação de sistemas distribuídos: a Internet. Nela pode-se visualizar as relações entre os computadores pessoais, os servidores de rede, os provedores ou *ISP* (*Internet Service Provider*) e os links de comunicação entre todas as entidades envolvidas.

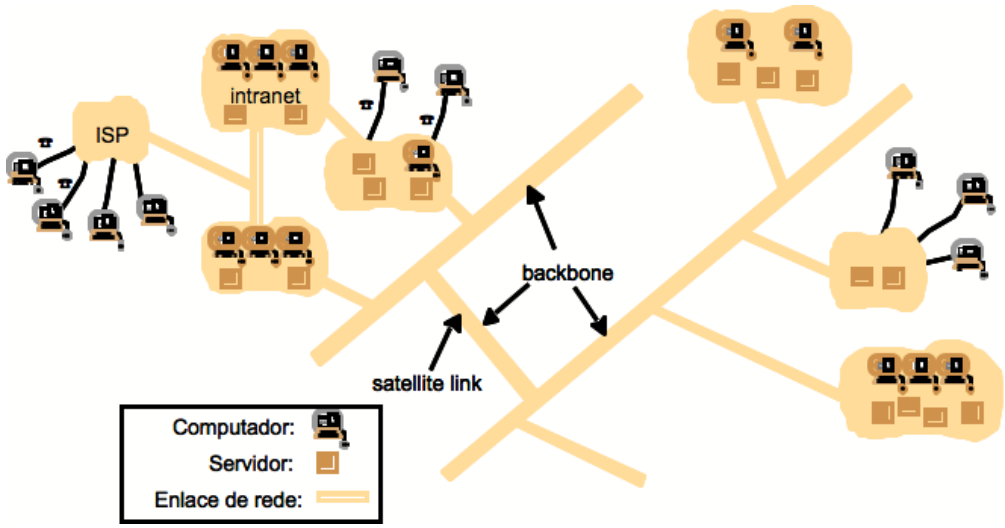


Figura 2.1: Um exemplo de Sistemas Distribuídos [COU 05b]

2.2.1 Metas dos Sistemas Distribuídos

O fato de ser possível, tecnologicamente falando, montar sistemas distribuídos não quer dizer necessariamente que seja viável sua implantação. Afinal, dada a tecnologia corrente, também é possível colocar processadores modernos de até quatro núcleos em um computador. Nessa seção serão discutidas quatro metas importantes que devem ser cumpridas no desen-

volvimento de um sistema distribuído. Um sistema distribuído deve oferecer fácil acesso aos seus recursos; deve ocultar razoavelmente bem o fato de que os recursos estão distribuídos por uma rede; deve ser aberto e deve poder ser expandido [TAN 02].

2.2.1.1 Acesso a Recursos

Os sistemas distribuídos têm como principal meta facilitar aos usuários, programadores e aplicações o acesso aos recursos (de hardware e software) remotos e o compartilhamentos destes, de maneira segura, controlada e eficiente. Existem algumas razões plausíveis para que as organizações queiram compartilhar recursos. Entre elas pode-se citar a principal: diminuição dos custos.

Como exemplo da aplicação do compartilhamento de recursos para a redução dos custos, pode-se citar que é mais barato para uma organização ter 40 impressoras funcionando de modo compartilhado para seus 400 funcionários, do que 400 impressoras, uma para cada. Neste segundo caso, gerando um custo de manutenção e operação desnecessário.

Para alcançar esta meta, os projetistas de sistemas distribuídos usam técnicas de aplicação de graus de transparência de distribuicação dos recursos, para que o usuário não perceba que está trabalhando em um sistema no qual os recursos estão geograficamente distribuídos. Utilizam também sistemas abertos, com padrões e métodos normalizados, para que os programadores possam integrar suas aplicações ou expandir funcionalidades, mesmo que não conheçam a infra-estrutura distribuída em que esta trabalhando. Alguns conceitos sobre sistemas abertos são discutidos na sub-seção 2.2.1.2.

2.2.1.2 Sistemas Abertos

Uma meta que deve ser atingida na construção de sistemas distribuídos é a abertura. Um sistema distribuído aberto é um sistema que oferece serviços de acordo com regras padronizadas que descrevem sua sintaxe e semântica [TAN 02], [EMM 94], [CLA 03]. Diz-se ainda, que um sistema é aberto quando ele pode ser estendido e reimplementado de várias maneiras [COU 05b].

Como um exemplo de sucesso do uso de sistemas distribuídos abertos pode-se citar a Internet, que utiliza protocolos de comunicação conhecidos e que normalizam a troca de informações na rede. Geralmente os serviços distribuídos são especificados e normalizados por meio de interfaces, que costumam ser descritas em uma Linguagem de Definição de Interface ou

IDL (*Interface Definition Language*).

2.2.1.3 Escalabilidade

No sentido contrário à centralização dos serviços e algoritmos, uma das principais metas dos sistemas distribuídos é a escalabilidade. Um sistema é descrito como *escalável* se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários a ele conectados [COU 05b].

O provimento de escalabilidade é um dos assuntos mais pesquisados na computação distribuída pelo fato que o crescimento das redes de computadores, sistemas e bancos de dados é uma tendência, e à medida que são ampliados, este crescimento muitas vezes vem acompanhado de perda de desempenho.

2.2.1.4 Transparência

De acordo com Coulouris *et al.* [COU 05b], a transparência é definida como sendo a ocultação, para um usuário final ou para um programador de aplicativos, da separação dos componentes em um sistema distribuído de modo que o sistema seja percebido como um todo, em vez de uma coleção de componentes independentes. As implicações da transparência têm grande influência sobre o projeto do software a ser executado sobre um sistema distribuído.

Uma meta importante de um sistema distribuído é ocultar o fato de que seus processos e recursos estão fisicamente distribuídos por vários computadores. Um sistema distribuído que é capaz de se apresentar a usuários e aplicações como se fosse apenas um único sistema de computador é denominado transparente [TAN 02].

Existem alguns tipos de trasparência, e este conceito pode ser aplicado a diversos aspectos de um sistema. O modelo de referência RM-ODP (*Reference Model for Open Distributed Processing*) da ISO (*International Standardization Organization*) normaliza os principais tipos de transparência em sistemas distribuídos [MEE 95], [ISO 95]. A tabela 2.1 descreve os sete principais tipos de transparência que são aplicados em sistemas distribuídos.

Dos sete tipos principais de transparência citados, os dois mais importantes são: a transparência de acesso e a de localização. A sua presença ou ausência nas aplicações distribuídas afetam fortemente a utilização de recursos distribuídos. Muitas vezes na literatura, as duas são referenciadas em conjunto como transparência de rede. São estas características que provêm aos usuários a abstração necessária, sobre onde as informações estão e como elas podem ser acessadas

Tabela 2.1: *Formas de Transparência em um Sistema Distribuído [MEE 95], [ISO 95]*

Transparência	Descrição
Acesso	Oculta diferenças na representação de dados e no modo de acesso a um recurso
Localização	Oculta o lugar em que um recurso está localizado
Migração	Oculta que um recurso pode ser movido para outra localização
Relocação	Oculta que um recurso pode ser movido para outra localização enquanto em uso
Replicação	Oculta que um recurso é replicado
Concorrência	Oculta que um recurso pode ser compartilhado por diversos usuários concorrentes
Falha	Oculta a falha e a recuperação de um recurso

acessadas.

Podemos usar como exemplo da transparência de acesso, um sistema distribuído que fornece uma interface com usuário na Web e cujos dados estão geograficamente distribuídos. Neste caso, para o usuário final é transparente o acesso aos dados, pois a aplicação gerencia a massa de dados e entrega para o usuário somente o resultado de sua solicitação. Em contrapartida, podemos citar como exemplo de falta de transparência de acesso, o mesmo sistema, porém neste caso, ele não permite que o usuário acesse um determinado arquivo em um computador remoto, a não ser que ele utilize um programa FTP (*File Transfer Protocol*).

2.2.2 Tipos de Sistemas Distribuídos

De acordo com Tanenbaum e Steen [TAN 02] os sistemas distribuídos podem ser classificados em 3 grandes tipos: Sistemas Distribuídos Pervasivos, Sistemas de Informação Distribuídos e em Sistemas de Computação Distribuídos.

Nos dias de hoje, com o avanço da tecnologia, pode-se encontrar sistemas distribuídos nos quais a instabilidade é o comportamento esperado. Nesses sistemas, que são denominados Sistemas Distribuídos Pervasivos, os equipamentos costumam ser caracterizados por seu pequeno tamanho, pela alimentação por bateria, por sua mobilidade e por terem somente uma conexão sem fio [TAN 02]. Porém tais características não precisam ser necessariamente interpretadas como restritivas, como ilustrado pelas possibilidades dos modernos *smart phones* [ROU 05].

Um outro importante tipo de sistema distribuído, pode ser encontrado em empresas e organizações, nas quais devido ao grande número de aplicações de rede o baixo nível de interoperabilidade entre elas, se tornou um problema. Os Sistemas de Informação Distribuídos são

sistemas constituídos por *middlewares* de aplicação que têm o papel de fornecer a interoperabilidade necessária para as aplicações de rede.

Os Sistemas de Computação Distribuídos, objeto de estudo deste trabalho, podem ser entendidos como um segmento da ciência da computação que tem como objetivo a melhoria do desempenho de aplicações distribuídas e paralelas, utilizando-se de complexas infraestruturas computacionais [DAN 05]. Nesta classe, em termos estritos, pode-se fazer uma distinção entre dois grupos: os agregados computacionais ou aglomerados computacionais (*cluster computing*) e a computação em grade ou computação em malha (*Grid Computing*). As aplicações que são executas nesses ambientes são geralmente separadas em duas grandes classes: as distribuídas e as paralelas. Neste trabalho, serão abordadas as aplicações distribuídas, não levando em consideração o paralelismo envolvido nos processos dos cenários propostos.

Com a disseminação da área, outras classificações e denominações surgiram, tais como: Sistemas Multimída Distribuídos, que tratam diretamente com sistemas de tempo real (*Real Time Systems*) para aúdio e video; Sistemas de Arquivos Distribuídos, que buscam a distribuição de informações em sistemas de arquivos e os Sistemas Peer-to-Peer, que trabalham em um paradigma para a construção de sistemas onde dados e recursos computacionais são provenientes da colaboração de muitas máquinas.

Nas próximas seções serão ilustrados alguns conceitos de Sistemas de Computação Distribuídos, sendo estes: Agregados Computacionais, Bancos de Dados Distribuídos e os Sistemas de Arquivo Distribuídos, visando detalhar as características técnicas que serão utilizadas no cenários propostos.

2.3 Agregados Computacionais

Um agregado computacional ou aglomerado de computadores (*Cluster Computing*) é um sistema de computação distribuída composto por um conjunto de computadores independentes (elementos) que trabalham de forma integrada como se fosse um recurso computacional único [BUY 99]. Os agregados são caracterizados pelo agrupamento físico considerando-se um pequeno limite geográfico, ou virtual, de inúmeros computadores para a execução de aplicações [DAN 05].

Os elementos que compõem um agregado podem residir em um único gabinete, ligados por um dispositivo de interconexão ou estar fisicamente separados e conectados por uma rede local de comunicação. O que determina a posição física dos elementos é o tipo de rede

de interconexão usado. Geralmente, as redes usadas tendem a se encontrar no limiar máximo do seu desempenho e são construídas com componentes padronizados, além do fato da distância raramente ultrapassar alguns metros [PAS 03].

A grande motivação para o surgimento e a criação dos agregados foi prover a possibilidade de se construir um equipamento de alto poder computacional, simplesmente empilhando diversos computadores comuns. E, se algum dia houver a necessidade de se ampliar este poder computacional, basta inserir mais computadores no agregado, que automaticamente ele será expandido.

Devido à inexistência de uma taxonomia amplamente aceita para a classificação dos ambientes de agregados computacionais, não há uma coerência nas definições e nas características básicas deste tipo de ambiente. Porém, existem alguns pontos relevantes que podem auxiliar neste processo, são eles: limite geográfico, modo de utilização dos nodos, tipo de hardware e conexão entre os nodos, requerimentos de recursos solicitados pelas aplicações e os tipos dos nodos que compõem a configuração do agregado [DAN 05].

De outra forma, existem algumas características, que na maioria das configurações dos agregados, estão presentes. São elas:

- **Independência**: cada elemento do sistema possui um ou mais processadores, memória, dispositivos de entrada e saída e executa seu próprio sistema operacional. Esta característica determina ainda que o elemento na maioria dos casos pode ser removido do agregado sem que isto comprometa o funcionamento do sistema.

- **Imagem Única do Sistema**: por teoria um agregado computacional é um recurso computacional único e indivisível, desta forma muitas ferramentas para construção de agregados utilizam-se deste conceito (*Single System Image ou SSI*).

- **Conexão Especializada**: na maioria dos casos os elementos do agregado são conectados por algum tipo de rede de comunicação veloz e de tecnologia aberta.

Um exemplo clássico de agregado computacional pode ser visualizado na figura 2.2, na qual podem-se ver os elementos de hardware em cada nodo do sistema, sendo eles: o processador, a memória e a NIC (*Network Interface Card*), todos interligados por um barramento de comunicação.

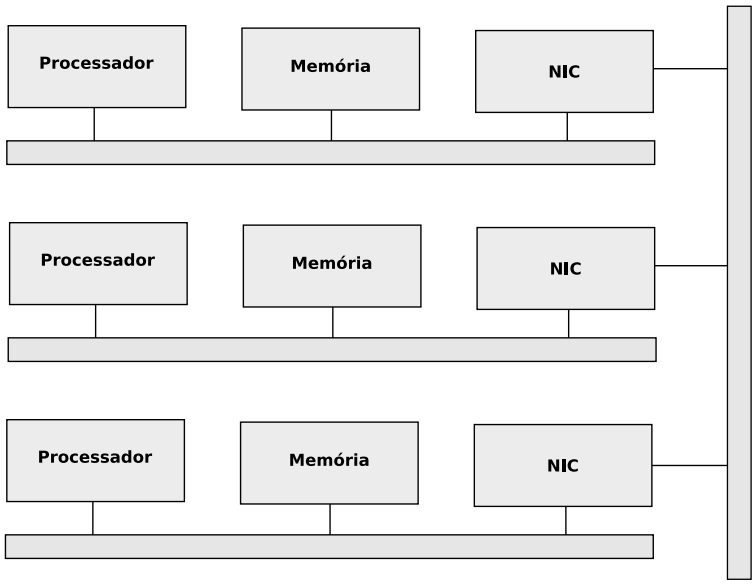


Figura 2.2: Exemplo de Clássico de um Agregado

2.3.1 Tipos de Agregados

Uma característica importante na construção dos agregados é a forma de interligação física e a forma de utilização dos diversos computadores através de um dispositivo de interconexão de rede. De maneira geral, os agregados podem ser interligados de duas maneiras, na forma não–dedicada e de forma dedicada [DAN 05].

Nos agregados com configurações não-dedicadas, todos os computadores que os compõem, têm softwares locais que são responsáveis pelas tarefas executadas nos mesmos. Neste tipo de agregado, não existe uma interação natural entre os computadores, pois é necessário que exista um software instalado, que será responsável por prover tal interoperabilidade. Este software será responsável também por abstrair todas as diferenças de hardware e dos softwares locais, para as aplicações que irão ser suportadas pelo ambiente.

As configurações dedicadas de agregados, por outro lado, são a opção mais indicada para a execução de aplicações críticas. Em outras palavras, o ambiente dedicado corresponde à melhor resposta para a execução de aplicações vitais para as organizações [DAN 05].

Os pacotes de software utilizados para construção de agregados têm algumas funções vitais para o correto funcionamento das aplicações que vão utilizar o ambiente. Eles são responsáveis, além de abstrair o hardware e software do ambiente, por duas facilidades básicas fundamentais: alta disponibilidade e balanceamento de carga.

A funcionalidade de alta disponibilidade determina que o software tem que ser capaz de detectar automaticamente um problema em um dos nodos do agregado e redirecionar suas

funções para os outros nodos do agregado. Esta facilidade é fundamental em ambientes onde os serviços não podem ficar fora do ar ou indisponíveis.

A funcionalidade de balanceamento de carga determina que o software instalado no nodo mestre do agregado, tem que ser capaz de distribuir homogeneamente a carga das solicitações para os nodos escravos do agregado processares, provendo assim alta desempenho para as solicitações. Na figura 2.3 pode-se visualizar uma arquitetura com as facilidades de balanceameto de carga e alta disponibilidade.

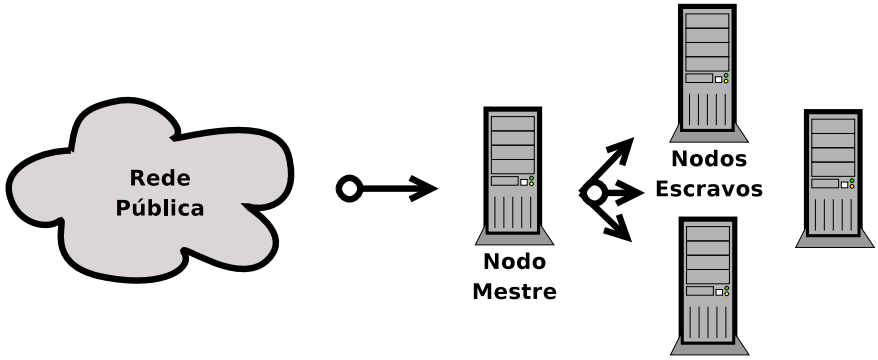


Figura 2.3: Exemplo de Agregado Computacional

2.3.2 Trabalhos Relacionados

Existem algumas alternativas de ferramentas para a aplicação de agregados computacionais em configurações dedicadas e não-dedicadas. Entre as opções desenvolvidas com o conceito de software livre podemos citar o Linux Virtual Server [LVS 08], Beowulf [BEO 08], OpenSSI [OPE 08b] e o openMOSIX [OPE 08a]. Por fatores de projeto, as ferramentas analisadas serão baseadas em software livre.

Os agregados Beowulf são soluções de alto desempenho e altamente escaláveis usando computadores pessoais, não especializados baseadas no sistema operacional Linux. O projeto do Beowulf foi elaborado para melhorar o desempenho das aplicações, proporcionalmente à medida que máquinas são adicionadas em sua infra-estrutura, desde um aglomerado de dois nodos até um de 1024 nodos. O projeto foi criado por Donald Becker da NASA, e hoje é utilizado em todo o mundo, principalmente para processamento de dados com finalidade científica [BEO 08], [STE 95].

O OpenSSI é definido como uma solução de agregados computacionais abrangente que oferece um Sistema de Imagem Única ou SSI (*Single System Image*) altamente disponí-

vel para Linux. As principais metas do OpenSSI incluem prover às arquiteturas suportadas, altas taxas de disponibilidade, escalabilidade e gerenciabilidade para as aplicações [OPE 08b].

O openMOSIX é outra opção para a construção de agregados computacionais do tipo Sistema de Imagem Única, possibilitando assim a conversão de uma rede comum de computadores em um super-computador para aplicações baseadas no sistema operacional Linux. Ele é caracterizado como uma extensão ao núcleo (*kernel*) do sistema operacional. O openMOSIX utiliza seu próprio sistema de arquivos (openMOSIX Filesystem ou oMFS), permitindo assim, uma melhor troca de dados e mensagens entre os vários processos do agregado [OPE 08a].

O LVS ou *Linux Virtual Server* é um projeto aberto iniciado por Wensong Zhang em 1998. A missão do projeto é construir um servidor de alto desempenho, altamente disponível para o sistema operacional Linux, utilizando-se da tecnologia de agregados, fornecendo assim, a escalabilidade, confiabilidade, usabilidade e gerenciabilidade necessárias para aplicações de missão critica [LVS 08].

O objetivo do LVS é fornecer um *framework* para construção de ambientes altamente disponíveis e escaláveis para serviços de rede utilizando um grande grupo de servidores *commodity* (computadores comuns). O TCP/IP do núcleo do sistema operacional Linux foi extendido para suportar três técnicas de balanceamento de carga baseados em IP (*Internet Protocol*). Esta implementação permite a paralelização de diferentes tipos de agregados. A escalabilidade do LVS é proporcionada pela adição ou remoção de um nodo do agregado, e a alta disponibilidade é assegurada pelos mecanismos internos de detecção de falhas, que em tempo de execução podem reconfigurar o sistema e garantir sua disponibilidade [ZHA 00].

O LVS tem sido muito utilizado em ambientes cujos serviços não podem falhar e onde sua disponibilidade é vital para o negócio. Por este motivo e por tratar-se de um software onde as facilidades de balanceamento de carga, alta disponibilidade e alto desempenho são bem desenvolvidas, este software foi designado para fazer parte da configuração dos cenários propostos no capítulo 4.

2.4 Bancos de Dados Distribuídos

A primeira geração de bancos de dados, foi desenvolvida com seu foco voltado para centralização dos dados, pois até então, mecanismos que pudessem distribuir e disseminar os dados não estavam maduros o suficiente. No final da década de 70 e início da década 80 esta tendência se inverteu, como pode-se constatar nos trabalhos de [BER 78], [ROT 80], [BER 81]

,[BER 84], [SMI 86], e a descentralização e autonomia de processamento em bancos de dados cresceu.

A tecnologia de Bancos de Dados Distribuídos (*Distributed Databases*) ou BDD é originada da junção de outras duas tecnologias: bancos de dados e comunicação de dados em redes. Portanto, esta mudança de paradigma ocorreu devido ao grau de sofisticação que estas duas tecnologias atingiram, somada à necessidade das organizações na descentralização das suas informações. Novos protocolos de rede, aumento na velocidade das conexões, redes de alto desempenho e a Internet, foram alguns dos fatores que fomentaram a disseminação desta nova tecnologia.

As organizações têm se mostrado muito interessadas na descentralização do processamento (a nível de sistema) ao mesmo tempo que obtêm uma integração dos recursos de informação (no nível lógico) dentro de seus sistemas geograficamente distribuídos de bancos de dados, aplicações e usuários. Juntamente com os avanços nas comunicações, existe hoje um endosso da abordagem cliente-servidor para o desenvolvimento de aplicações, que assume muitas das questões sobre BDD [ELM 99].

Os BDDs podem ser definidos como uma coleção de vários bancos de dados logicamente inter-relacionados, distribuidos ao longo de um sistema de rede de computadores e um sistema de gerência de banco de dados distribuídos (SGBDD) como um sistema de software que gerencia um banco de dados distribuído, ao mesmo tempo em que torna a distribuição transparente para o usuário [ÖZS 99], [ELM 99].

A figura 2.4 apresenta o cenário de aplicação clássico dos BDDs. Pode-se visualizar os Nodos 1, 2 e 3, juntamente com seus bancos de dados geograficamente distribuídos, interligados a uma rede de comunicação que provê a intercomunicação entre os nodos do sistema. O Nodo 1, neste cenário age como uma entidade cliente, que apenas usa a rede de comunicação para acesso aos nodos do sistema.

A distribuição de informação neste tipo de ambiente acarreta uma maior complexidade no projeto e na implementação dos sistemas. De acordo com Elmasri e Navathe [ELM 99], um BDD é responsável por fornecer algumas funções adicionais às aplicações, além das funções de um SGBD (Sistema Gerenciador de Banco de Dados), sendo elas:

- Controlar os dados: capacidade de controlar a distribuição, a fragmentação e a replicação dos dados, expandindo o catálogo do SGBD.

- Processamento de consultas distribuídas: capacidade de acessar sites remotos e transmitir consultas e dados entre os varios sites através de uma rede de comunicação.

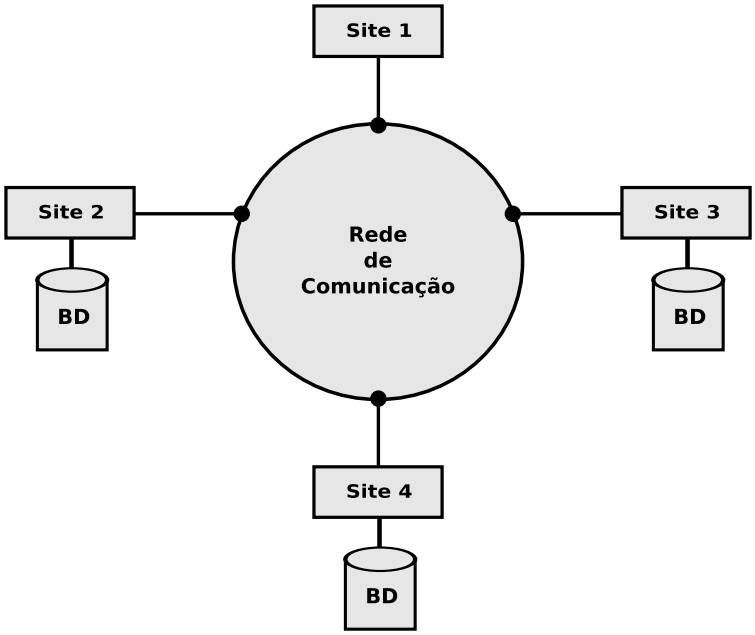


Figura 2.4: Ambiente de Banco de Dados Distribuídos

- Gerenciamento de transações distribuídas: capacidade de planejar estratégias de execução para consultas e transações que acessam dados de mais de um site e de sincronizar o acesso a dados distribuídos e mantar a integridade do banco de dados geral.

- Gerenciamento de dados replicados: capacidade de decidir qual cópia de um item de dados replicado acessar e manter a consistência de cópias de um item de dado replicado.

- Recuperação de bancos de dados distribuídos: capacidade de se recuperar de colapsos em sites individuais e de novos tipos de falhas tais como a falha de um link de comunicação.

- Segurança: transações distribuídas devem ser executadas com o gerenciamento apropriado da segurança dos dados e os privilégios de acesso de usuários.

- Gerenciamento do diretório distribuído: um diretório contém informações (*metadados*) sobre dados do banco de dados. O diretório pode ser geral para todo o BDD ou local para cada site.

Nas próximas sub-seções serão tratados os aspectos sobre fragmentação e replicação em BDD. Aspectos estes, que serão usados nos cenários propostos no capítulo 4. Também serão relatados alguns trabalhos relacionados sobre o tema.

2.4.1 Fragmentação de Dados

A metodologia de desenvolvimento de BDD necessita de fragmentação de dados, alocação de dados e de otimização [BEL 96] para se obter uma maior flexibilidade na implementação das soluções. O problema da fragmentação de dados em Bancos de Dados Distribuídos Relacionais (BDDR) foi reconhecido por seu impacto sobre o desempenho dos sistemas como um todo [COR 87], [NAV 84], [ÖZS 99].

Pode-se definir fragmentação de dados como um conjunto de técnicas, que são utilizadas para dividir o banco de dados em unidade lógicas, chamadas *fragmentos*, que podem ser armazenados em vários sites [ELM 99]. Como já foi citado, a fragmentação de dados impacta diretamente no desempenho das aplicações, porém muitas vezes este custo envolvido não é levado em consideração pelos desenvolvedores de software, o que acarreta em soluções com baixo desempenho.

Segundo Ozsu e Valduriez [ÖZS 99] existem dois tipos conhecidos de fragmentação de dados: Fragmentação Horizontal (*Horizontal Fragmentation*) e Fragmentação Vertical (*Vertical Fragmentation*). Existe ainda a Fragmentação Híbrida ou Mista (*Hibrid Fragmentation*), na qual os fragmentos são obtidos pelo uso dessas duas estratégias em conjunto.

A fragmentação horizontal é caracterizada por promover a divisão das *tuplas* de uma relação em subconjuntos de dados, ou seja, ela divide uma relação horizontalmente, agrupando linhas para criar subconjuntos de tuplas. Por outro lado, a fragmentação vertical, provê um conjunto de relações diferentes, a partir de uma mesma relação, ou seja, ela divide uma relação verticalmente através de colunas.

2.4.2 Replicação de Dados

Replicação de dados tem sido muito estudada no contexto dos bancos de dados distribuídos [ÖZS 99]. No âmbito de agregados de bancos de dados, o principal assunto estudado é como prover escalabilidade, ou seja, como conseguir desempenho com um grande número de nodos, e ainda prover autonomia, para vários tipos de replicação, tais como mestre-escravo (*master-slave*), multi-mestre (*multimaster*), replicação total (*full replication*), replicação parcial (*partial replication*) [COU 05a].

A replicação de dados consiste em manter múltiplas cópias de dados, chamadas réplicas, em dispositivos diferentes em uma ou mais redes, sendo uma importante tecnologia para serviços distribuídos e alvo de muito estudo nos últimos anos. A replicação aprimora a disponibi-

lidade dos dados, permitindo o seu acesso mesmo que algumas das réplicas estejam indisponíveis [SAI 05].

As técnicas tradicionais de replicação buscam manter a consistência dos dados distribuídos, fazendo parecer ao usuário que existe apenas uma única cópia, altamente disponível, dos dados [BER 87]. As técnicas que implementam essa solução são chamadas “pessimistas” e podem ser observadas, por exemplo, em sistemas que elegem uma réplica primária como responsável por atender a todas as solicitações a um objeto em particular [BER 87].

Essas técnicas se comportam bem e são altamente recomendadas para redes locais (LAN), em que as latências são pequenas e as falhas pouco comuns. Porém, em redes de longa distância ou geograficamente distribuídas (WAN), como a Internet, técnicas pessimistas de replicação têm baixo desempenho [SAI 05] por razões que envolvem desde a maior latência e menor confiabilidade das redes de longa distância, até problemas de escalabilidade com os algoritmos pessimistas mais comuns. Contanto, esta não é a única abordagem de replicação difundida, existindo ainda a replicação otimista.

A replicação otimista é o nome dado a um conjunto de técnicas para o compartilhamento de dados de forma eficiente em redes de longa distância ou ambientes com dispositivos móveis. A característica que diferencia os algoritmos de replicação otimista de suas contrapartes pessimistas é a forma como é abordado o controle de concorrência [GOE 98]. Os algoritmos pessimistas coordenam as réplicas de forma síncrona durante os acessos e bloqueiam os outros usuários durante uma atualização. Algoritmos otimistas permitem que os dados sejam acessados sem a necessidade de uma sincronização prévia, assumindo de forma “otimista” que eventuais problemas podem ocorrer. As atualizações são propagadas em *background* e os conflitos ocasionais são tratados posteriormente. Exemplos de uso das técnicas otimistas podem ser observadas no sistema de gerenciamento de nomes de domínio da Internet (*DNS – Domain Name System*) e em sistemas de colaboração assíncrona entre usuários, como no software CVS [SAI 05].

Os algoritmos de replicação otimista oferecem algumas vantagens em relação aos pessimistas, como a possibilidade de os nodos poderem continuar operando normalmente com suas réplicas, mesmo no caso da comunicação entre os mesmos ser interrompida ou se tornar instável, além de possibilitarem maior escalabilidade devido à comunicação entre os sites ser menos frequente [SAI 05]. Os custos dessas vantagens surgem na área da consistência dos dados, pois na replicação otimista, em um determinado momento as réplicas podem possuir dados divergentes, possibilitando o surgimento de conflitos entre as mesmas [GOE 98]. Dessa forma, o grande desafio dos algoritmos de replicação otimista é manter a consistência dos dados, ou seja, manter as

réplicas suficientemente similares entre si.

Um aspecto a ser considerado na replicação de dados é o número de réplicas que possuem permissões para gravar informações nas outras. Sistemas com um único mestre (*single-master ou master-slave*) têm todas as suas atualizações originadas de uma única réplica e passadas para as outras, denominadas escravos. Já os sistemas multi-mestre (*multi-master*) permitem que as atualizações sejam submetidas por múltiplas réplicas independentemente, permitindo uma maior disponibilidade dos dados, mas também tornando o seu gerenciamento mais complexo.

2.4.3 Trabalhos Relacionados

Existem alguns projetos de software livre que tratam de replicação em bancos de dados, entre estes, pode-se citar os mais populares: PGPool [PGP 08], PGCluster [PGC 08], Postgres-R [POS 08] e o Slony-I [SLO 08]. Entre os software proprietários pode-se citar: IBM DB2 Replication [IBM 08], Oracle (RAC) Real Application Clusters [ORA 08] e o SyBase Replication Server [SYB 08]. Entretanto, pelo fato da arquitetura do Portal de Telemedicina, se basear no sistema gerenciador de banco de dados relacional PostgreSQL, as soluções abordadas e estudadas são soluções baseadas neste sistema.

PGCluster é um sistema de replicação síncrona de composição multi-mestre para PostgreSQL. Devido a sua característica síncrona, não há uma latência considerável com os dados duplicados entre os nodos do cluster e considerando sua composição multi-mestre, o sistema fica disponível para concorrência de acesso entre vários usuários [PGC 08]. O PGCluster é um sistema com uma arquitetura robusta, porém o custo de largura de banda envolvido e o baixo grau de flexibilidade que ele provê, inviabiliza a sua utilização neste projeto.

O Slony-l é um sistema de replicação com único-mestre (*single-master*) para múltiplos escravos. Ele é um sistema concebido para trabalhar em centro de dados e em *backup sites,* em que todos os nodos do sistema estão disponíveis e por estas razões trabalha com algoritmos pessimistas de replicação assíncrona [SLO 08]. Se o ambiente de operação for suscetível a indisponibilidade, o Slony-l não é a solução ideal para o projeto.

O PGPool é um *middleware* que funciona entre o PostgreSQL e o cliente de dados, que oferece os seguintes recursos: limitador de conexões, *pooling* de conexões, replicação, balanceador de carga e *queries* paralelas. A sua característica de replicação permite criar um *backup site* em tempo real para múltiplos nodos [PGP 08]. A principal aplicação do PGPool é ser um servidor de controle de conexões, mas apesar de ter esta funcionalidade, não trabalha muito

bem como ferramenta de replicação de dados.

O Postgres-R é uma extensão do sistema de banco de dados relacional PostgreSQL, e proporciona uma eficiente, rápida e consistente replicação para agregados de bancos de dados [POS 08]. A principal utilização do Postgres-R é construir um sistema de balanceamento de carga e alta disponibilidade de dados. Porém, seu modo de replicação sendo síncrono, determina um alto consumo de comunicação de dados.

Devido aos softwares correlatos não atenderem totalmente as questões relativas ao projeto e aos objetivos do cenário proposto que trata de replicação de dados assíncrona, multimestre, com fragmentação híbrida e do tipo pessimista, foi desenvolvido um protótipo chamado PGR (*Postgres Replication*), relatado no capítulo 4.

2.5 Sistemas de Arquivos Distribuídos

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 [TAN 02].

Alguns tipos de sistemas de arquivos distribuídos são construídos de acordo com a arquitetura cliente-servidor, sendo que o NFS (*Network File System*) pode ser citado como um dos exemplos de maior utilização. Ele é muito utilizado em ambientes UNIX para troca e compartilhamento de arquivos, servindo também como camada básica para aplicações que rodam neste tipo de ambiente.

De outra forma, algumas vezes este modelo de operação tende a ser estendido para os agregados computacionais. Levando em consideração que agregados computacionais costumam ser utilizados por aplicações paralelas, muitas vezes os sistemas de arquivos são ajustados para este fim [TAN 02]. Desta forma, além dos nodos do sistema compartilharem processador e memória, eles compartilham espaço em disco para armazenamento de dados.

Existem algumas vantagens na utilização desta tecnologia, e entre elas, pode-se citar: os arquivos disponibilizados nestes ambientes tendem a estar mais disponíveis, pois na maioria dos software já estão embutidas características de alta disponibilidade; as rotinas de backup e segurança são mais fáceis de administrar, devido ao fato de se tratar de um único grande volume de armazenamento distribuído; podem fornecer grande espaço de armazenamento, devido à possibili-

dade de agregar o espaço disponível de vários servidores e por este motivo também são altamente escaláveis.

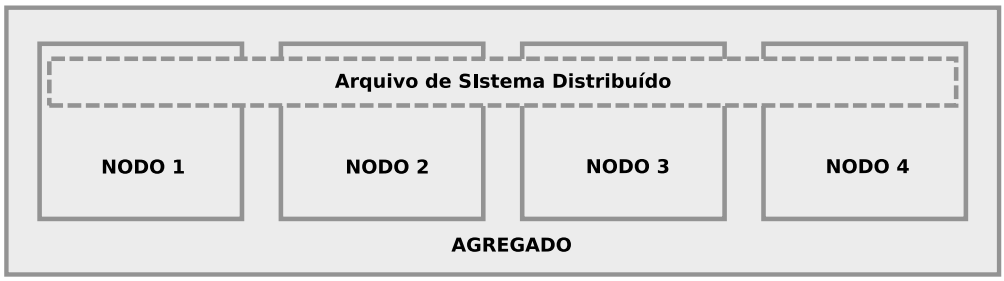


Figura 2.5: Ambiente Clássico de Sistemas de Arquivos Distribuídos

Na figura 2.5 pode-se visualizar um ambiente clássico de sistemas de arquivos distribuídos em agregados computacionais. Nela, os nodos do agregado computacional estão compartilhando espaço de armazenamento em disco, formando assim, um único grande volume de armazenamento. Na próxima seção serão expostos alguns trabalhos relacionados ao tema.

2.5.1 Trabalhos Relacionados

Sistemas de arquivos distribuídos estão em crescimento exponencial, e muito softwares são e estão sendo desenvolvidos para este fim. Nesta seção serão relacionados alguns exemplos destes softwares para construção de sistemas de arquivos distribuídos. Serão relatados o Andrew File System ou AFS [AFS 08], o CODA [COD 08], o Google File System ou GFS [GHE 03] e o Parallel Virtual File System ou PVFS [PVF 08].

O CODA é um sistema de arquivos distribuídos avanądo, desenvolvido no estado da arte de técnicas experimentais em sistemas de arquivos. Ele é desenvolvido pelo grupo de M. Satyanarayanan na Universidade de Carnegie Mellon. O CODA tem algumas características que fazem com que ele se destaque: ele suporta ser desligado em estado de operação do sistema, congelando seu estado e o restaurando quando o sistema for reestabelecido. A sua estrutura interna de prevenção a falhas e replicação é muito similar a do Andrew File System [COD 08], [SAT 90], [KIS 92].

O Andrew File System ou AFS é um software para construção de sistemas de arquivos distribuídos que foi criado na Universidade de Carnegie Mellon e é apoiado e desenvolvido pela Transarc Corporation, que agora chama-se IBM Pittsburgh Labs. Ele fornece uma arquitetura cliente-servidor para compartilhar arquivos e replicação de arquivos para somente leitura (*read-only*), proporcionando assim transparência de localização, alta escalabilidade, segurança e

migração de dados transparente. A IBM disponibiliza o código-fonte do AFS livrevemente para o desenvolvimento da comunidade de software livre, chamada OpenAFS [AFS 08], [TOB 94].

O trabalho de Ghemawat *et al.* descreve o desenvolvimento do Google File System ou GFS [GHE 03], sistema de arquivos distribuídos do Google. O GFS é um sistema de arquivos distribuídos altamente escalável para aplicações que necessitam de alta vazão de dados. Tem características de tolerância a falha e roda sobre hardwares comuns (*commodity*).

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.

O PVFS é desenvolvido conjuntamente entre o Laboratório de Pesquisas em Arquiteturas Paralelas (*Parallel Architecture Research Laboratory*) da Clemson University e a Divisão de Matemática e Ciência da Computação (*The Mathematics and Computer Science Division*) do Argonne National Laboratory.

Carns *et al.* [CAR 00] descrevem em seu trabalho a concepção e o desenvolvimento do PVFS. No trabalho é desenvolvida uma série de testes de desempenho, baseados em leituras e escritas simultâneas nos nodos do agregado. Foram avaliados ainda a integração e o desempenho do MPI-IO sobre PVFS, para grande quantidade de acessos concorrentes. Os testes foram realizados em redes Myrinet *versus* rede Fast Ethernet.

Neste trabalho o *Parallel Virtual File System* foi o software para construção de sistemas de arquivos distribuídos/paralelos, por sua robutez, alto desempenho, pela vasta documentação que a comunidade de software livre proporciona à ferramenta e pela excelente integração com o ambiente de operação proposto, baseado no sistema operacional Linux.

2.6 HDF – Hierarchical Data Format 5

HDF é a sigla em inglês para Formato de Dados Hierárquico, um formato portável de dados desenvolvido no parque de pesquisas da Universidade de Illinois nos Estados Unidos. Trata-se de uma biblioteca para manipulação de alta performance de dados científicos, possibilitando o armazenamento de objetos com grandes quantidades de dados, como *arrays* multidimensionais e tabelas com grandes conjuntos de dados, que podem ser utilizados em conjunto, de forma que atenda às aplicações.

A sigla HDF também se refere ao conjunto de softwares, interfaces de aplicativos e utilitários que compõem a biblioteca e que possibilitam aos usuários a manipulação de arquivos no formato. O software está em desenvolvimento desde 1988 e atualmente encontra-se em sua quinta versão, a HDF5. Esta, por questões de projeto, não possui compatibilidade com as versões anteriores, mas possui diversos aprimoramentos em relação a elas, em especial no que se refere ao acesso paralelo a dados [SHA 04b].

2.6.1 Modelo de Dados e de Programação

O HDF5 implementa um modelo para o gerenciamento e armazenamento de dados que é um modelo abstrato de representação dos dados e um modelo abstrato de armazenamento. Também estão presentes bibliotecas para implementar o modelo abstrato e mapeá-lo para diferentes mecanismos de armazenamento.

A biblioteca também implementa um modelo de transferência de dados que provê uma forma eficiente de transferência de dados de uma representação de armazenamento para outra representação. Entre suas características, destaca-se a capacidade de auto-descrição através de metadados sobre o conteúdo dos arquivos, flexibilidade no uso de diferentes tipos de dados em um único ou múltiplos arquivos, portabilidade entre plataformas, padronização de tipos e formatos de dados e o fato de a biblioteca ser um software de código aberto.

O seu modelo de programação suporta desde pequenos sistemas até grandes multi-processadores e clusters, manipulando por meio de instanciação, registro e recuperação de dados, objetos do modelo abstrato. A biblioteca é a implementação concreta do modelo de programação, e exporta as APIs HDF5 e suas interfaces.

As estruturas conceituais mais importantes do HDF5 são o *dataset* e o *group*. Um *dataset* é um *array* multi-dimensional de elementos de um tipo de dados (*datatype*) especificado. Os tipos de dados podem ser atômicos (números inteiros, de ponto flutuante, cadeias de caracteres, etc) ou tipos compostos. Os grupos são similares a estruturas de diretório, provendo uma forma de organizar explicitamente os conjuntos de dados em um arquivo HDF5.

A biblioteca HDF5 possui uma camada virtual de arquivos (*VFL ou Virtual File Layer*), que consiste de uma API para o tratamento de operações de I/O em baixo nível, de forma que uma aplicação possa gravar dados utilizando diferentes recursos de armazenamento [ROS 01], como pode ser observado na figura 2.6.

O modelo abstrato de dados (*ADM ou Abstract Data Model*) define conceitos

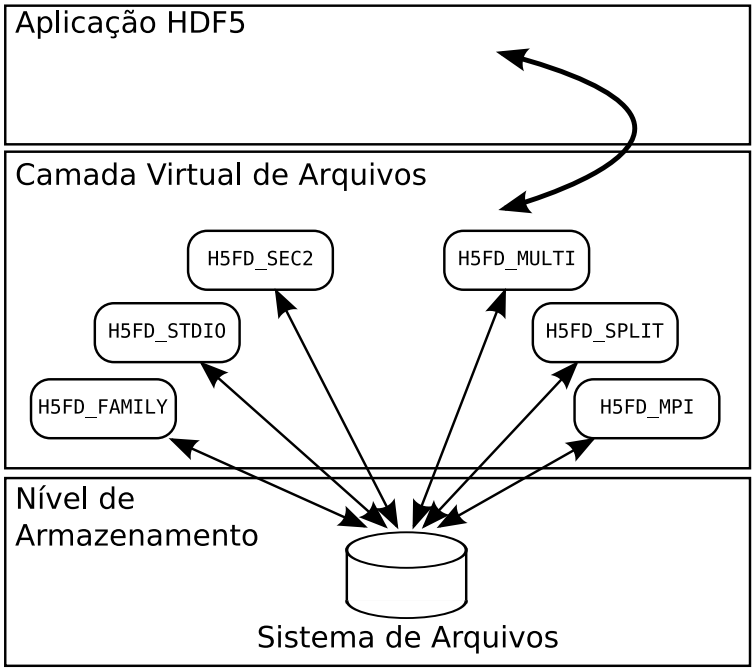


Figura 2.6: Relacionamento Entre os Modelos

para a e descrição de dados complexos armazenados nos arquivos. É um modelo genérico projetado para conceitualmente abranger diversos modelos específicos de dados. Diferentes tipos de dados podem ser mapeados para objetos do modelo de dados HDF5, e conseqüentemente armazená-los e recuperá-los utilizando a biblioteca HDF5 [HDF 08].

Os conceitos-chave do modelo incluem:

- **Arquivo:** um arquivo binário armazenado em disco ou memória de um computador;
- **Grupo:** uma coleção de objetos, que pode incluir outros grupos;
- **Dataset:** um array multidimensional de elementos de dados, com atributos e outros metadados;
- **Datatype:** uma descrição de um tipo específico de dados para um elemento de dados, incluindo seu padrão de armazenamento;
- **Dataspace:** uma descrição das dimensões de um array multidimensional;
- **Atributo:** um valor que possui um nome, associado a um grupo, dataset ou um datatype que também possua nome;
- **Lista de Propriedades:** uma coleção de parâmetros que controlam as opções da biblioteca. Algumas propriedades são permanentemente armazenadas como parte do objeto, enquanto

outras são transitórias e aplicadas em acessos específicos. Cada classe da lista de propriedades tem suas propriedades específicas.

Para uma melhor compreensão do modelo de dados e de programação e de todos os conceitos envolvidos em sua teoria, cada um deles será relatado em detalhes no ANEXO 4 deste documento.

2.6.2 APIs de Alto Nível

O HDF5 Lite possui dois conjuntos de funções, relacionados à manipulação de *datasets* e *atributos*. Os métodos da API consistem de funções de maior nível de abstração que realizam mais operações por chamada do que as funções básicas da biblioteca HDF5. O propósito é servir como interface por meio de funções intuitivas que utilizam conjuntos específicos de características da API existente. Mesmo com a API de alto nível, porém, algumas funções de criação e acesso aos arquivos do HDF5 básico podem precisar ser utilizadas por uma aplicação.

Por outro lado, a API HDF5 Image define um sistema padrão de armazenamento para *datasets* HDF5 que espera-se que sejam interpretados como imagens. Esta versão da API possui métodos para armazenar imagens indexadas de 8 bits, nas quais cada *pixel* é armazenado como um índice em uma paleta de cores, e imagens *true color* de 24 bits, em que o armazenamento de cada *pixel* contém os planos de cores vermelho, verde e azul.

Ambas as APIs serão utilizadas em uma ferramenta construída para servir como ponte de acesso entre o servidor de imagens DICOM e as funções da biblioteca HDF5 para a criação de arquivos neste formato a partir de um conjunto de imagens médicas. Na próxima seção serão mostrados alguns trabalhos relacionados a implementação de sistemas de armazenamento baseados em HDF5.

2.6.3 Trabalhos Relacionados

O HDF5 é utilizado em uma vasta área de aplicação, com diferentes abordagem, mas principalmente em aplicações científicas e tecnológicas, como podemos constatar nos trabalhos de Cohen *et al.* [COH 06], Gosink *et al.* [GOS 06], Lee e Hung [LEE 00] e Yu *et al.* [YU 06].

No trabalho de Cohen *et al.* [COH 06] é apresentado um estudo baseado em extensões de sistemas gerenciadores de bancos de dados relacionais, que permitem a representação

de dados científicos e operações estatísticas comuns. Foram utilizados o NetCDF e o HDF, que são os dois dos formatos científicos mais populares. Neste trabalho, ainda foram realizadas operações estatísticas, usando as extensões dos SGBDs em comparação com as operações nativas do NetCFD e do HDF.

Gosink *et al.* [GOS 06] apresentam uma nova abordagem para acelerar os acessos a grandes arquivos HDF5, utilizando indexação semântica multi–dimensional, denominada HDF5–FastQuery. Os resultados deste trabalho demostraram que esta nova abordagem é duas vezes mais rápida que o os métodos padrões do HDF5.

Lee e Hung [LEE 00] descrevem o desenvolvimento de uma ferramenta, elaborado pelo time CERES (*Clouds and Earth's Radiant Energy System*) de Gerenciamento de Dados, que é parte do Sistema de Observação da Terra da NASA. Esta ferramenta tem a função de visualização gráfica via interface com o usuário para análises de *datasets* HDF.

Yu *et al.* [YU 06] realizam em seu trabalho uma série de avaliações de desempenho em gravações de dados paralelas. São realizados vários testes de leitura e escrita em interfaces paralelas de alto nível tais como HDF e o NetCDF, em ambientes com um grande número de processadores, no caso é utilizado o supercomputador Blue Gene/L.

2.7 Considerações

Neste capítulo, foram descritos o estado da arte e os aspectos envolvidos na construção de Sistemas Distribuídos. Foram descritas as estratégias: Agregados Computacionais, Bancos de Dados Distribuídos, Sistema de Arquivos Distribuídos e por fim, o Formato de Dados Hierárquico, HDF5. No próximo capítulo será abordado o estado da arte da Telemedicina no Brasil e no Mundo, bem como suas aplicações em ambientes reais.

Capítulo 3

Sistemas de Telemedicina

3.1 Introdução

Neste capítulo serão abordados os aspectos dos sistemas de telemedicina, bem como as tecnologias e processos envolvidos. Será feita uma breve discussão sobre a origem da telemedicina e de alguns casos correlatos existentes pelo mundo. Após isto, será apresentado o Portal de Telemedicina da Rede Catarinense de Telemedicina. Na próxima seção serão discutidos aspectos do padrão DICOM, sendo seguida pela apresentação do CyclopsDCMServer, objeto de estudo neste trabalho.

3.2 Telemedicina

Desde seu surgimento, na década de 60 [BAS 02], a Telemedicina vem sendo sugerida e aplicada como uma forma de prover acesso à saúde das pessoas ou de comunidades que, de uma forma ou de outra, estejam isoladas ou desprovidas de pessoal médico qualificado. Sua utilização nestes casos pode reduzir o custo de transporte e maximizar a utilização do parque tecnológico instalado em hospitais e clinicas médicas [MCN 02].

De acordo com a ATA (*American Telemedicine Association*), a Telemedicina pode ser definida como "o uso de informação médica veiculada de um local para outro, por meio de comunicação eletrônica, visando a saúde e educação dos pacientes e do profissional médico, para assim melhorar a assistência de saúde [ATA 08].

Essa é uma definição extremamente abrangente que inicia relacionando a arte médica que envolve contato com o paciente, diagnóstico, tratamento ou até intervenção cirúrgica,

com qualquer meio de comunicação que possa unir dois ou mais pontos distantes fisicamente, desde cartas escritas até prontuários eletrônicos de pacientes, interligados por redes de comunicação de alta velocidade.

Um exemplo clássico da aplicação da Telemedicina, pode ser visualizado também em um sistema onde um profissional de saúde envia imagens e dados médicos de uma localidade para outra. Pode se configurar ainda por um simples telefonema onde um médico solicita uma segunda opinião a um outro médico especialista, ou até mesmo o envio de imagens médicas, como tomografias computadorizadas, para que o laudo seja provido por outro profissional especialista.

Existem atualmente diversos projetos de Telemedicina em muitos Estados e países desenvolvidos ou em desenvolvimento. Exemplos destes podem ser vistos nos Estados de Arizona [MCN 02] e Utah [PET 96] nos EUA, que ainda conta com projeto nacional que busca a normalização e padronização dos projetos para uma melhoria na interoperabilidade entre os sistemas [CHO 06]. Outros exemplos como estes se dão, por exemplo, no Estado da Bavária [JAH 05] na Alemanha, no Equador [MIJ 04], no Japão [TOM 04], em Taiwan [LUH 05], no Kazaquistão [SHA 04a], na China [XUE 07], na Índia [PAL 05] e no Brasil [MAI 06].

Embora todos estes projetos tenham como objetivo comum: diminuir o custo e facilitar o acesso de pacientes em ambientes distantes, ou mesmo de especialistas médicos de grandes centros, eles diferem completamente em termos de implementação e funcionalidade. Na grande maioria dos trabalhos, inclusive no Brasil, a Telemedicina tem o foco voltado para a teleconsulta entre um médico e um paciente, para geração de laudos virtuais ou a realização de teleconferências entre dois ou mais médicos com o objetivo de analisar um caso médico complicado.

A maioria desses projetos de telemedicina já esta em vigor nesses países há algum tempo, e como no caso dos Estados Unidos, fazem parte do plano de governo para a melhoria da saúde. Embora já consolidada nos países desenvolvidos, a telemedicina ainda caminha a pequenos passos na maioria dos países em desenvolvimento. A telemedicina pode diminuir os custos de transporte de pacientes aos centros especializados, e permitir ainda, um melhor controle da quantidade e origem dos exames, além de promover a melhora do atendimento ao paciente, tornando mais rápido o diagnóstico e gerando um histórico de informações sobre o paciente. Ajudando assim a diminuir a replicação desnecessária de exames, prática atualmente muito comum [WAN 97].

3.3 Aplicações de Telemedicina

No Brasil, existem alguns projetos de telemedicina nos Estados, porém a falta de normalização no desenvolvimento destes sistemas impossibilita a interação e a troca de informações entre eles. Desta forma, não existe a possibilidade do médico fazer pesquisas a fim de encontrar registros do histórico do paciente e realizar consultas sobre este paciente realizadas em outros Estados, o que poderia reduzir custos desnecessários com exames repetidos, e ainda, auxiliar a futuras tomadas de decisão. Para que estas informações possam ser recuperadas e utilizadas de forma plena pelos profissionais da área da saúde, elas necessitam estar disponíveis, organizadas e padronizadas, facilitando seu entendimento, o acesso e cruzamento.

A recuperação destes dados, no entanto, não se apresenta como uma tarefa simples. Os dados precisam ser re-formatados e padronizados (normalizados), o que implica em considerável esforço computacional para a conformação dos dados. A re-formatação dos dados implica possivelmente, entre outras técnicas no uso de ontologias, que forneçam vocabulários necessários para auxiliar, por exemplo, na estruturação de um laudo médico. A padronização desses textos de laudos é um grande desafio, pois muitos hospitais possuem sua própria forma de modelagem e estruturação dos dados. Esse processo de registro demanda um grande trabalho adicional para analisar e recuperar informações que poderiam auxiliar, por exemplo, no diagnóstico de doenças [WAN 97].

3.3.1 Portal de Telemedicina

O desenvolvimento de um projeto de pesquisa elaborado pelo Hospital Universitário da Universidade Federal de Santa Catarina (UFSC), em parceria com a Secretaria de Estado da Saúde (SES) de Santa Catarina, possibilitou a disponibilização em rede, de imagens, sinais e laudos médicos gerados a partir de estabelecimentos de saúde espalhados pelo Estado, a chamada Rede Catarinense de Telemedicina (RCTM).

A concepção desta rede foi motivada pela necessidade de diminuir os custos de transporte de paciente para os centros especializados mais próximos podendo desta forma, melhorar o controle sobre a quantidade de exames realizados e suas origens, tornando o atendimento e o diagnóstico dos pacientes mais rápido.

Esta rede de conhecimento é composta por algumas tecnologias que foram desenvolvidas localmente, entre as quais podem ser citadas: Dicomizer, Nutrição Parenteral, Teledermatologia, CyclopsDCMServer e o Portal de Telemedicina, sendo que os dois últimos serão

objetos de estudo neste trabalho. Na figura 3.1 pode-se visualizar a tela principal de operação do Portal.

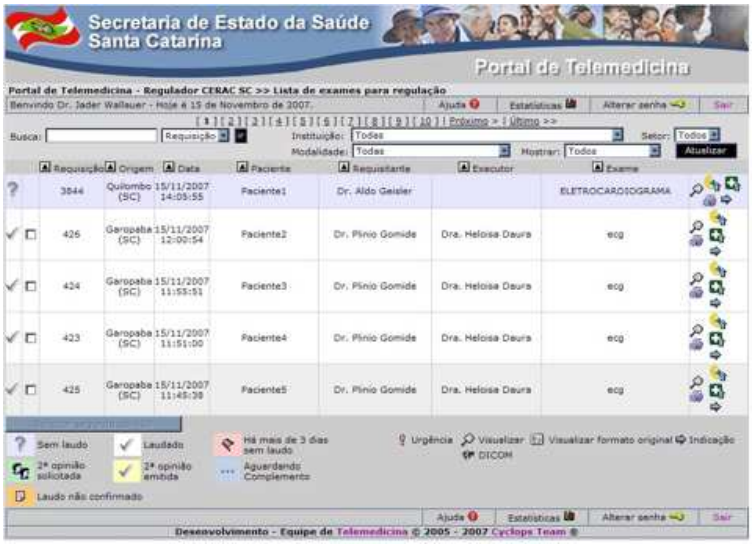


Figura 3.1: O Portal de Telemedicina do Estado de SC

O Portal de Telemedicina [TEL 08] é um sistema baseado na *web*, construído na linguagem de programação PHP (*PHP Hypertext Preprocessor*), sendo que seus dados são armazenados em um sistema gerenciador de bancos de dados relacionais PostgreSQL. Trata-se de um sistema com arquitetura de três camadas, simples, porém com funcionalidades de grande valia.

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. A partir deste processo, um médico especialista pode acessar as informações de qualquer ponto da Internet, podendo assim laudar exames, discutir o caso com outros médicos e diagnosticar um laudo de segunda opinião, entre outras operações.

3.4 DICOM

Na década de 70, com o avanço tecnológico, surgiu uma nova modalidade de equipamentos para diagnóstico médico, a Tomografia Computadorizada (*Computed Tomography*). Dois exemplos deste tipo de imagem podem ser visualizados na figura 3.2. Ela introduziu o uso de técnicas de arquivamento e transmissão de imagens digitalizadas nos ambientes hospitalares.

Após o surgimento deste tipo de exame, outras modalidades de diagnóstico elaborados por imagens digitais emergiram. Porém, a inexistência de um padrão único de troca de imagens tornou-se um problema, frente a necessidade de interconectar sistemas de diferentes fa-

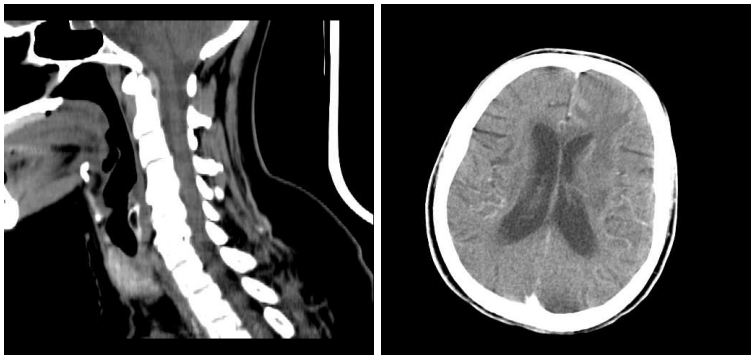


Figura 3.2: Exemplo de uma Tomografia Computadorizada

bricantes. Por este fato, cada fornecedor criava sua própria solução para arquivamento, impressão, transmissão e visualização de imagens digitais e informações de cada modalidade de exame [DEL 05].

A partir deste problema o Colégio Americano de Radiologia ou ACR (*American College of Radiology*), em conjunto com a Associação de Fabricantes de Equipamentos Elétricos dos Estados Unidos ou NEMA (*National Electrical Manufactures Association*), reconheceram a necessidade da existência de um padrão único para a intercomunicação entre equipamentos fabricados por diferentes fornecedores.

A primeira versão deste padrão foi publicada em 1985 e foi denominada ACR-NEMA Standards Publication No. 300-1985. Logo após a primeira versão, ocorreram duas revisões, em 1986 e 1988, sendo a última denominada ACR-NEMA Standards Publication No. 300-1988. Estas regulações proviam especificações de uma interface de hardware, um conjunto de formatos de dados e um conjunto mínimo de comandos de software.

A maior deficiência deste padrão era em relação ao suporte aos ambientes de redes de computadores. Por este motivo, o padrão resistiu até 1992, quando foi lançada a ACR-NEMA Standards Publication PS3, também chamada de DICOM 3 (*Digital Imaging and Communications in Medicine*), um padrão muito mais robusto que os primeiros.

Atualmente o DICOM 3 é o padrão de fato para o PACS (*Picture Archiving and Communications System*), sendo suportado pela grande maioria dos aparelhos que trabalham com informações médicas digitais. O fato de um aparelho o suportar é a garantia de que este poderá ser facilmente integrado em um PACS já existente devido à utilização de tecnologias de rede acessíveis e baratas (TCP/IP ou OSI) para a sua implementação, e ao fato deste poder utilizar-se de serviços disponibilizados por outros aparelhos que também suportem o padrão DICOM [DEL 05].

3.4.1 Servidor DICOM

A partir da versão 3 do DICOM, o Projeto Cyclops desenvolveu o seu próprio servidor para armazenamento e recuperação de imagens médicas, chamado CyclopsDCMServer. Ele foi desenvolvido para funcionar em ambientes de Sistemas de Arquivamento e Comunicação de Imagens ou PACS, dentro de hospitais ou clínicas radiológicas. Entretanto, devido às necessidades de expansão, ele foi remodelado para trabalhar em redes de longa distância.

Um dos principais objetivos deste servidor é armazenar e fornecer arquivos DICOM indexados em um repositório de dados gerenciados por um sistema de gerenciador de bancos de dados relacionais, tais como: PostgreSQL, MySQL, Oracle, etc. Toda a comunicação entre os equipamentos médicos e o servidor é realizadas através de redes de comunicação baseadas em TCP/IP [CYC 08]. Atualmente o CyclopsDCMServer suporta os seguintes tipos de modalidades de exames médicos:

- Radiografia Computadorizada (CR)
- Tomografia Computadorizada (CT)
- Ressonância Magnética (MR)
- Medicina Nuclear (NM)
- Ultra-Som (US)
- Cintilografia (XA)
- Eletrocardiogramas (ECG)

No seu modo de operação, visualizado na figura 3.3, os equipamentos de imagens médicas instalados nos hospitais, após realizar os exames no pacientes, enviam em tempo real as imagens para o CyclopsDCMServer. Ele então, reconstrói a imagem e realiza um pré-processamento para selecionar os objetos que serão para a base de dados DICOM. Após esta etapa, ele constrói a imagem, a converte para o formato JPEG Baseline e a envia para a base de dados do Portal de Telemedicina.

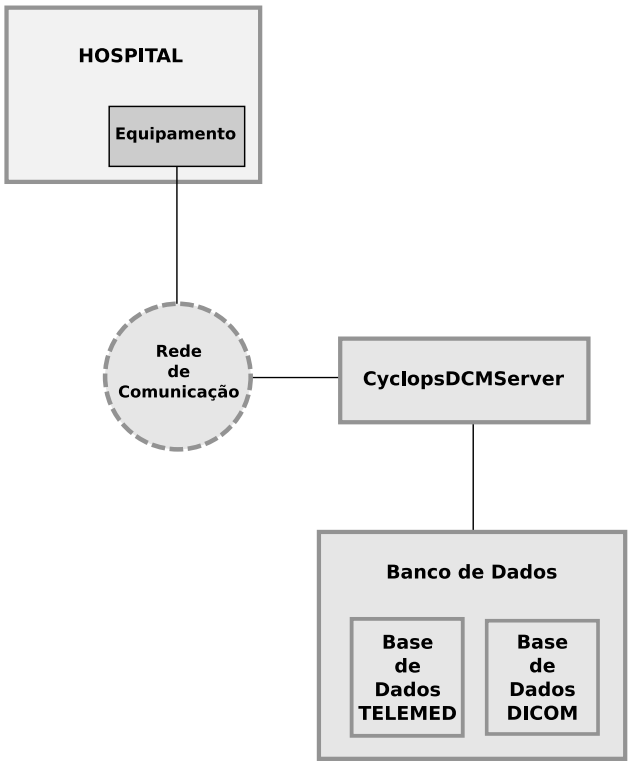


Figura 3.3: Arquitetura do CyclopsDCMServer

3.5 Considerações Finais

Neste capítulo foram abordados os aspectos dos sistemas de telemedicina e suas aplicações. Ainda foram mostradas as características do Portal de Telemedicina da Rede Catarinense de Telemedicina (RCTM) e o CyclopsDCMServer, que serão objetos de estudo nos cenários de aplicação de técnicas de sistemas distribuídos.

Capítulo 4

Estudos Desenvolvidos

4.1 Introdução

Neste capítulo serão abordados os cenários de aplicação propostos, para fins de avaliar o uso de algumas ténicas de Sistemas Distribuídos em aplicações de Telemedicina. No primeiro cenário proposto será avaliado o uso de Agregados Computacionais, visando prover desempenho e alta disponibilidade para a aplicação em questão. No segundo cenário será abordado um cenário utilizando Bancos de Dados Distribuídos, com o intuito de integrar as informações dos sistemas, utilizando replicação de dados. Por fim, no terceiro cenário, será avaliado o uso de Sistemas de Arquivos Distribuídos para armazenamento distribuído de imagens médicas no formato HDF5.

4.2 Ambiente de Aplicação

Todos os cenários avaliados foram desenvolvidos em um ambiente de teste. Para isto, foram utilizadas 5 máquinas, descritas na Tabela 4.1. Estas máquinas foram interconectadas por meio de uma LAN (*Local Area Network*) de 100Mbits. Todas as máquinas utilizam o sistema operacional Debian GNU/Linux, com *kernel* versão 2.6.22.

Para efeitos ilustrativos o Anexo 5 mostra a disposição física dos equipamentos no ambiente. Importante salientar que foram elaboradas configurações diferenciadas para cada tipo de cenário proposto. Em cada um deles será exposta a arquitetura utilizada de software, visando assim prover um melhor entendimento do problema e das soluções propostas.

Tabela 4.1: *Especificações das Máquinas do Ambiente*

Nome	Descrição
Master	Intel Celeron 2.53GHz, 512Mb RAM, 40Gb HD
Nodo1	Intel Celeron 2.53GHz, 256Mb RAM, 40Gb HD
Nodo2	Intel Celeron 2.53GHz, 256Mb RAM, 40Gb HD
Nodo3	Intel Celeron 2.53GHz, 256Mb RAM, 40Gb HD
Banco de Dados	Intel Xeon 2.4GHz, 512Mb RAM, 120Gb HD

4.3 Agregados Computacionais

Neste cenário, será exposto o uso de agregados computacionais em sistemas de telemedicina. Serão demostrados o problema do cenário atual, o desenvolvimento de uma proposta para a solução deste problema, bem como os resultados experimentais obtidos.

4.3.1 Descrição do Problema

À medida que os sistemas de telemedicina se popularizam e crescem, a quantidade de acessos simuntâneos a eles acompanha esta tangente. O caso do Portal de Telemedicina da SES (Secretária de Estado da Saúde) de Santa Catarina não é diferente, recebendo diariamente centenas de acessos de médicos, enfermeiros, pessoal administrativo e de outros sistemas que estão interconectados a ele.

Pelos fatores expostos, o seu desempenho fica comprometido, pois todos os usuários compartilham os mesmos recursos do servidor de aplicação, causando lentidão que em alguns casos podem afetar a disponibilidade do sistema. Levando em consideração que este tipo de sistema lida diretamente com apoio à vida, a sua indisponibilidade pode causar sérios problemas.

A figura 4.1 demostra o cenário atual de operação, onde temos apenas um servidor de aplicação e um servidor de banco de dados. Este tipo de arquitetura está propensa a falhas pelo fato da falta de redundância no nível de aplicação. Neste caso, se ocorrer uma falha de sistema ou de hardware, o sistema ficará indisponível a acessos.

4.3.2 Desenvolvimento da Proposta

A partir do problema apresentado no cenário atual, foi proposto um cenário de aplicação buscando resolver os problemas apresentados, fornecendo um estudo do uso de agrega-

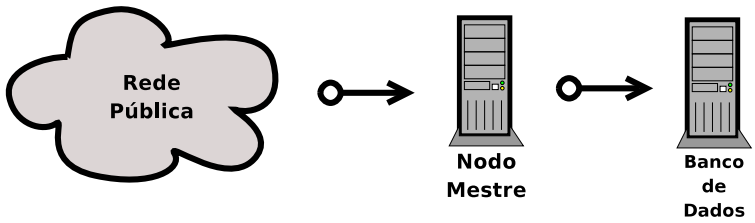


Figura 4.1: Arquitetura Atual Utilizada no Portal de Telemedicina

dos computacionais de alta disponibilidade e alto desempenho, aplicados ao sistema de telemedicina em questão, o Portal de Telemedicina da SES.

Na figura 4.2 é apresentado o cenário proposto, no qual foi utilizada uma arquitetura de agregados a nível de aplicação buscando prover ao sistema alta disponibilidade e alto desempenho, se valendo de técnicas de balanceamento de carga. Importante salientar que todas as requisições advindas da Internet são direcionadas para o Nodo Mestre (balanceador de carga) e ele tem o papel de direcionar as requisições aos Nodos Escravos do agregado.

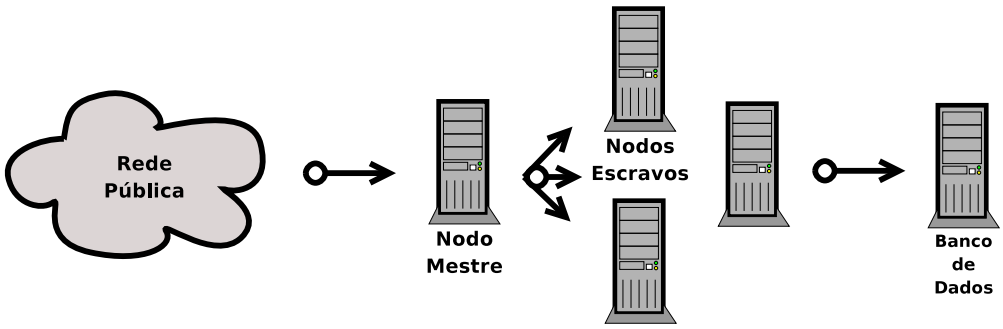


Figura 4.2: Arquitetura Proposta Utilizada no Portal de Telemedicina

Para a construção do agregado, foi utilizada a aplicação *Linux Virtual Server* ou LVS nos elementos. Na figura 4.3 pode-se visualizar o agregado em funcionamento, com todos os 3 nodos integrados (nodo1, nodo2 e nodo3) e aguardando requisições enviadas pelo balanceador de carga (master).

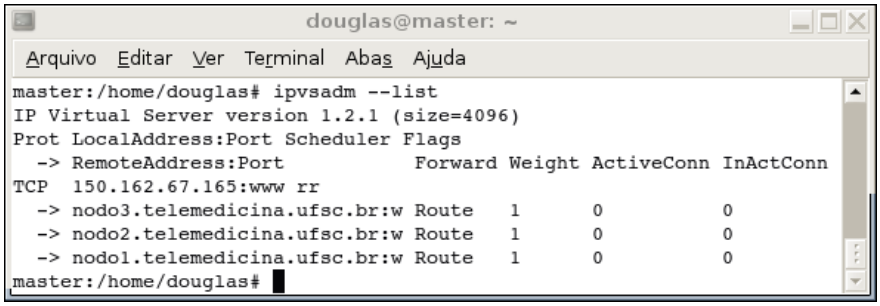


Figura 4.3: Lista das máquinas do cluster em operação

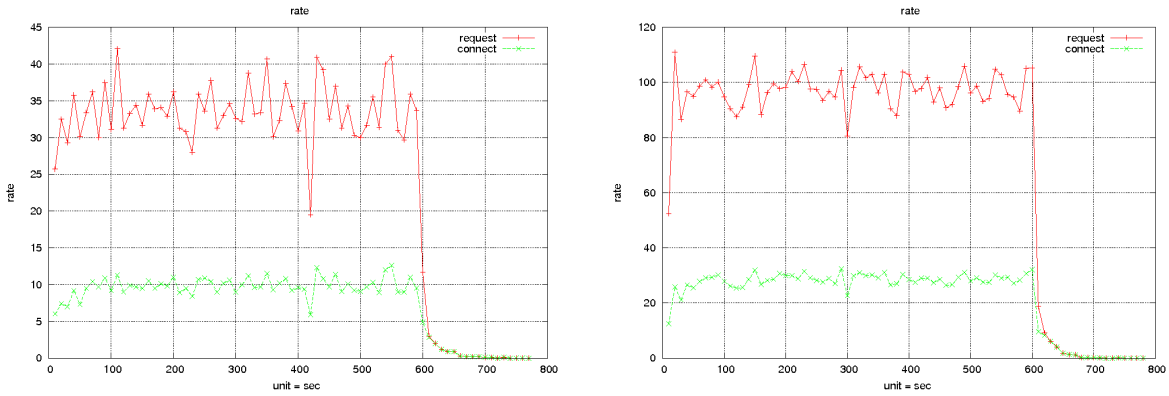
4.3.3 Resultados Experimentais

Diante da arquitetura implantada, foram elaborados testes de simulação de carga e stress nos servidores do agregado, para se obter as métricas de desempenho e escalabilidade do cenário atual de aplicação em comparação com o cenário usando agregados computacionais.

Para os testes de “stress” e “carga” no agregado, foi utilizada a aplicação Tsung [TSU 08]. O Tsung, anteriormente conhecido como IDX-Tsunami, pode ser usado para simular usuários, a fim de testar a escalabilidade e o desempenho dos servidores de aplicação. Atualmente ele pode ser usado para simular testes em servidores: HTTP, PostgreSQL, SOAP e Jabber.

Serão avaliados os tempos de pedidos de conexão, os tempos para o estabelecimento de conexão, o número de transações de páginas por segundo, número máximo de usuários conectados por segundo e o número máximo de usuários simultaneamente conectados por segundo, de ambos os cenários para que se tenha métricas mais precisas para as comparações.

Na figura 4.4(a) e na figura 4.4(b) são avaliados os tempos de pedidos de conexão (*requests*) e os tempos para o estabelecimento de conexão (*connect*). Sendo que na figura 4.4(a) esta representado o cenário atual e na figura 4.4(b) o cenário usando agregados computacionais. Pode-se observar nos resultados que o cenário atual obteve uma média de pedidos de conexão em torno de 33, em contrapartida com o cenário com o uso de agregados computacionais que teve um média de pedidos de conexão em torno de 95 solicitações a cada 10 segundos em média.

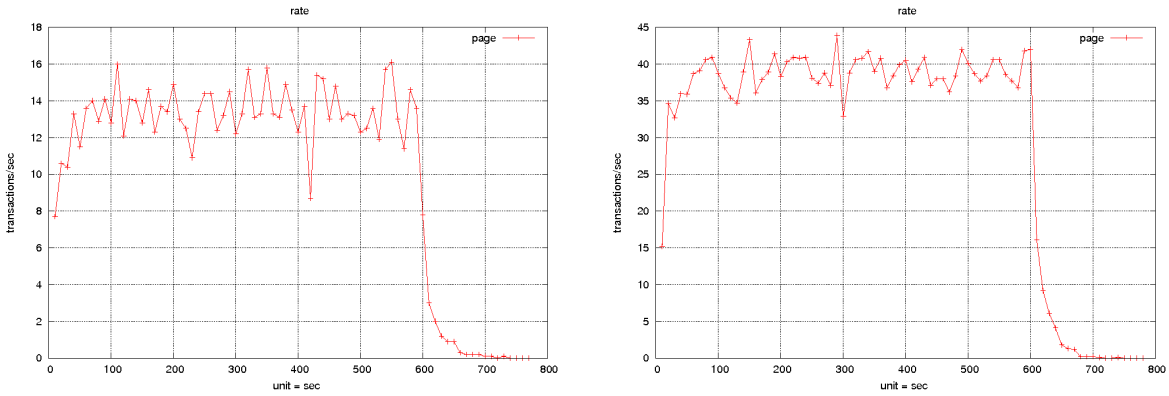


(a) Gráfico Sem Agregados (b) Gráfico com Agregados

Figura 4.4: *Tempos de Conexão*

Na figura 4.5(a) e na figura 4.5(b) são avaliadas as transações de páginas por segundos (*page*). Páginas são representadas pelo tempo de resposta para cada conjunto de solicitações. Na figura 4.5(a) são exibidos os tempos obtidos com o cenário atual, enquanto na figura 4.5(b) são mostrados os tempos obtidos usando agregados computacionais. Pode-se visualizar nos

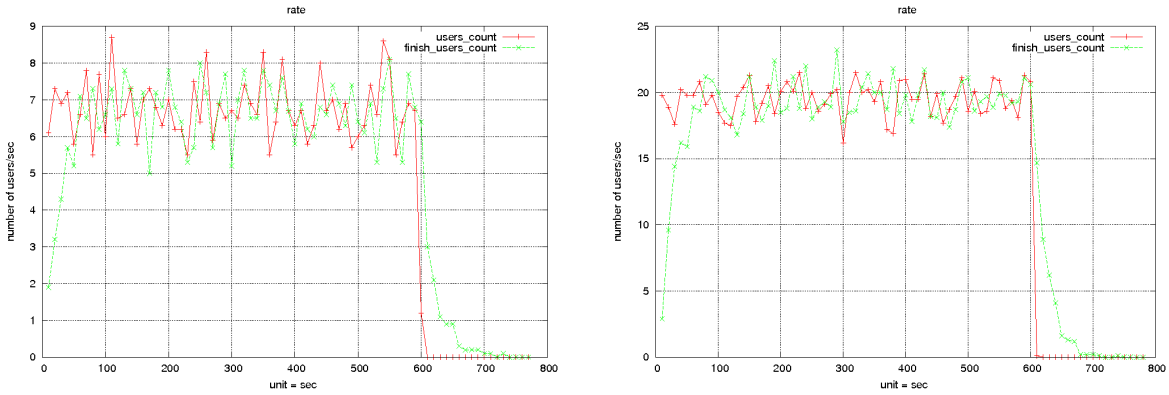
resultados que o cenário atual obteve um número médio de transações de 13, com contrapartida com o cenário com uso de agregados computacionais, que teve um número médio de 38 transações a cada 10 segundos em média.



(a) Gráfico Sem Agregados (b) Gráfico com Agregados

Figura 4.5: *Transações de Páginas por Segundo*

Na figura 4.6(a) e na figura 4.6(b) são avaliados o número máximo de usuários conectados por segundo. Na figura 4.6(a) são mostrado tempos obtidos no cenário atual e na figura 4.6(b) são mostrados os tempos obtidos usando agregados computacionais. Pode-se visualizar nos resultados que o cenário atual obteve um número máximo de usuários conectados em torno de 7, em contrapartida com o cenário com uso de agregados computacionais, que teve um número médio de usuários conectados em torno de 19 a cada 10 segundos em média.

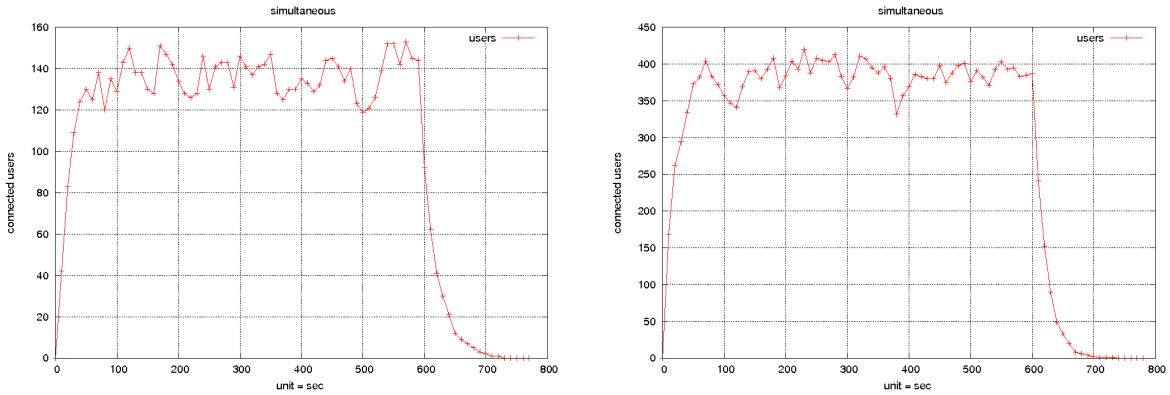


(a) Gráfico Sem Agregados (b) Gráfico com Agregados

Figura 4.6: *Número de Usuários Conectados por Segundo*

Na figura 4.7(a) e na figura 4.7(b) são avaliados o número máximo de usuários simultaneamente conectados. Na figura 4.7(a) são mostrados os resultados obtidos no cenário atual e na figura 4.7(b) são mostrados os tempos obtidos usando agregados computacionais. Pode-se

visualizar nos resultados obtidos que o cenário atual obteve um número médio de usuários simultaneamente conectados em torno de 132, em comparação com o cenário com uso de agregados computacionais, que obteve um número médio em torno de 380, a cada 10 segundos em média.



(a) Gráfico Sem Agregados (b) Gráfico com Agregados

Figura 4.7: *Usuários Simultâneos*

Como conclusão dos resultados experimentais deste estudo, viu-se que em todas as métricas retiradas, os resultados obtidos pelo cenário de agregados computacionais foram superiores, em comparação com o cenário atual. Outro resultado positivo da implantação de agregados computacionais é a alta disponibilidade obtida pela configuração, pois se um dos nodos escravos (Nodo1, Nodo2 ou Nodo3) falharem, isto não afetará a disponibilidade dos serviços.

Um ponto fraco desta configuração, como pode ser visualizado na arquitetura, é o Nodo Mestre, pois se ele falhar, o sistema ficará indisponível. Para solucionar este problema, a inserção de mais uma máquina em conjunto com a implantação de um protocolo de alta disponibilidade, como o HeartBeat [HEA 08], poderia solucionar este problema.

4.4 Bancos de Dados Distribuídos

Neste cenário serão abordados os aspectos de Bancos de Dados Distribuídos (BDD) em Sistemas de Telemedicina. A utilização de BDD em aplicações desta natureza são fundamentais no que tange à integração do conhecimento médico embutido nestes sistemas. Do contrário, as informações e os exames realizados em outras localidades ficariam isolados. Neste ponto, o uso de BDD e técnicas de replicação se tornam fundamentais no auxílio à integração do conhecimento deste tipo de sistema.

4.4.1 Descrição do Problema

No modelo adotado pela Rede Catarinense de Telemedicina (RCTM), os hospitais dos municípios, realizam exames nos pacientes, tais como: eletrocardiogramas, hemodinâmicas, cintilografias, tomografias computadorizadas e ressonâncias magnéticas. Após realizados, estes exames são enviados em tempo real, através do Servidor DICOM, para um banco de dados centralizado, localizado no Hospital Universitário da Universidade Federal de Santa Catarina (UFSC). A partir deste, os médicos então podem acessar as informações dos exames através do Portal de Telemedicina e efetuar os procedimentos necessários com os dados.

Este modelo é ideal quando não há a necessidade de disponibilizar as informações de forma integrada. Entretanto, prevendo sua expansão para outros Estados, esta arquitetura de bancos de dados centralizados não suportaria uma disseminação a nível nacional, pois a integração dos dados ficaria comprometida. A aplicação deste cenário se destina a estudar e identificar maneiras de prover escalabilidade para o sistema de telemedicina em questão, no que diz respeito à integração dos bancos de dados médicos distribuídos.

Sistemas de telemedicina são muito especializados, pois contém todas as regras, regulações e diretrizes médicas em sua programação. Seus bancos de dados armazenam grandes quantidades de dados, desde informações pessoais de pacientes, médicos e instituições, até todas as imagens dos exames realizados. Por este motivo, bancos de dados médicos costumam ter tamanho significativo, chegando a vários *terabytes*.

Nestes bancos de dados, são armazenados todos os estudos de cada paciente. Um estudo definido como um conjunto de imagens e informações estudadas que resultam em um único laudo. Para efeito de ilustrar o tamanho deste tipo de banco de dados, é mostrada na Tabela 4.2 a relação de alguns exames e seus respectivos tamanhos. É importante salientar, que diariamente são realizados vários exames, de vários pacientes diferentes e o volume dos bancos de dados tende a aumentar.

Tabela 4.2: *Tipos de exames e seus tamanhos*

Tipo de Exame	Tamanho Médio	Número de Itens	Média Total
Tomografia Computadorizada	500Kb	100	50Mb
Hemodinâmica	70Mb	8	560Mb
Cintilografia	1Mb	5	5Mb
Ressônancia Magnética	250Kb	200	50Mb

O maior problema encontrado quando existem múltiplos bancos de dados distribuídos é no que tange a integração dos dados. No caso de sistemas de telemedicina, os exames realizados em outras localidades ficam isolados, não podendo ser acessados de outras localidades. Desta forma, causando aumento nos custos operacionais para o Estado, falta de informação para auxilio na tomada de decisão médica e não permitindo que se tenha um histórico dos pacientes consolidado, com todas as informações de seus registros, consultas, exames, estudos e prescriçõs já realizadas.

Projetando a expansão do Portal de Telemedicina a nível nacional, integrando assim todos os bancos de dados, ficaria economicamente inviável em termos de largura de banda e em termos financeiros, a replicação de todo o banco de dados para cada um dos Estados, pois a quantidade de exames realizados em alguns Estados, é superior a média nacional. Como exemplo, podemos citar um Estado do porte de São Paulo replicando todos seu banco de dados para um Estado como Roraima.

Por este motivo neste trabalho será utilizado replicação parcial dos dados. A figura 4.8, ilustra o cenário real de aplicação, onde existem os bancos de dados estaduais isolados, sem nenhum nível de interoperabilidade.

Figura 4.8: Cenário com os Bancos de Dados Médicos Isolados

4.4.2 Desenvolvimento da Proposta

Avaliando as necessidades apresentadas por um sistema de telemedicina no que se refere a bancos de dados distribuídos, foi verificada a oportunidade de aplicar os conceitos de replicação parcial e integração dos dados do ambiente, utilizando as técnicas dos algoritmos de

replicação otimista de bases de dados. Para isso, foi desenvolvido um protótipo como prova de conceito utilizando o SGBD PostgreSQL.

A ferramenta, chamada internamente de PGR (*Postgres Replication*), permite que sejam especificados subconjuntos de tabelas das bases de dados a serem monitoradas. Dessas tabelas, é possível efetuar operações de fragmentação horizontal e vertical [COU 04] nos dados, ou seja, filtrar os registros por meio de condições nas consultas e obter apenas os atributos que interessam à *engine* de replicação.

O protótipo é composto de uma estrutura de tabelas relacionais criadas em cada um dos bancos de dados (réplicas) do ambiente, utilizada para armazenar informações de controle para o sistema. Também existe, no núcleo do sistema de replicação, uma *engine*, que será executada em cada um dos servidores que hospedam as bases de dados realizando as conexões nos outros bancos de dados e coordenando as replicações.

A ferramenta determina que a estrutura relacional das tabelas que são monitoradas pelo sistema de replicação seja a mesma em todas as bases de dados do sistema distribuído. Essa característica é necessária, pois a estratégia utilizada para a propagação dos dados é a multi-mestre, e cada uma das réplicas assume que as tabelas das outras réplicas possuem campos com os mesmos nomes e os mesmos tipos de dados de suas próprias tabelas. No ambiente, cada réplica armazena informações sobre as outras em tabelas específicas de sua própria base, o que lhe permite realizar conexões e consultas nas mesmas.

Durante uma atualização na base de dados local, se a mesma fosse um mestre em ambiente síncrono, uma conexão seria imediatamente realizada nas bases secundárias e as informações sobre a atualização seriam a elas transmitidas. Ao invés de utilizar essa abordagem, o PGR apenas grava as informações sobre a alteração em seu sistema, em uma tabela de eventos. Tais informações fazem referência a qual operação foi realizada, em que tabela e quando a mesma ocorreu.

Em dado momento, uma segunda réplica irá conectar na base de dados local e recuperar essas informações, obtendo os identificadores dos registros que foram atualizados e copiando os dados das respectivas tabelas para sua própria base. O momento da conexão escolhido por cada réplica do sistema, pode ocorrer posteriormente, caracterizando a propagação assíncrona dos dados. O momento da transação é armazenado na base do PGR, permitindo que em uma próxima conexão apenas as atualizações posteriores ao último acesso sejam identificadas e capturadas.

A figura 4.9 ilustra o processo de replicação dos dados no ambiente. No quadro 1 as réplicas A e B sofrem, cada uma, alterações distintas em tabelas que são monitoradas pela

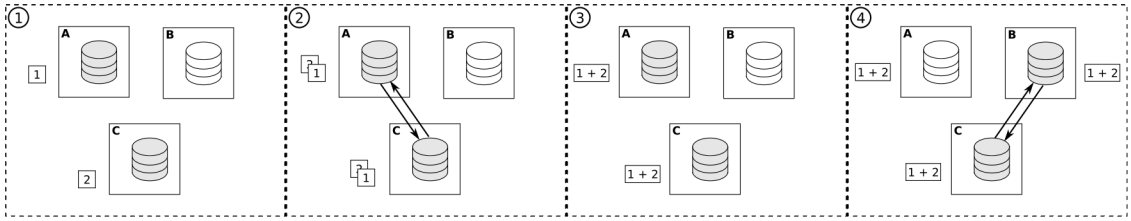


Figura 4.9: Processo de Replicação de Dados

engine. Quando uma comunicação for estabelecida entre as réplicas (quadro 2), as informações sobre as alterações nas bases de dados serão transmitidas entre as bases de dados. A partir deste ponto, cada base passa a observar as mudanças realizadas nas outras em conjunto com seus próprios dados locais, como mostra o quadro 3. Esse conjunto de dados misto entre os seus próprios e os obtidos de outras bases são transmitidos a réplicas as quais ainda não foram propagadas as novas alterações (quadro 4).

Os dados sobre as atualizações obtidos das réplicas são armazenados em tabelas chamadas internamente de *shadow*, cada uma com a mesma estrutura das tabelas monitoradas pela ferramenta, além de uma coluna extra que relaciona cada registro sua base de dados de origem. Após essa etapa, as consultas às tabelas monitoradas na base de dados passam a ser feitas através de uniões entre as tabelas originais e suas tabelas *shadow*, possibilitando inclusive a filtragem de dados de acordo com sua origem.

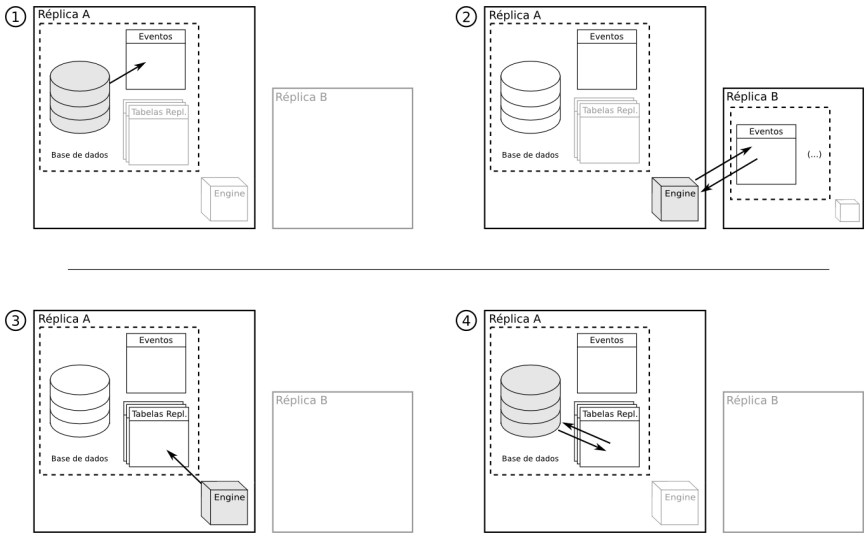


Figura 4.10: Processo Detalhado de Replicação de Dados

De uma forma mais detalhada, na figura 4.10 pode ser observada a comunicação entre bases de dados distintas quando alterações são realizadas em uma réplica. No quadro 1, as alterações realizadas na Réplica A são armazenadas em uma tabela de eventos do PGR. Tal tabela está presente em todas as réplicas, e é acessada pela *engine* para obter as atualizações mais recentes

de suas tabelas monitoradas, como ilustra o quadro 2.

A ferramenta é responsável por gravar as informações obtidas das outras bases de dados em tabelas especiais de replicação (as tabelas *shadow*, quadro 3). Uma vez concluído esse procedimento, a base de dados passa a utilizar as uniões relacionais com as tabelas *shadow* nas consultas às suas tabelas monitoradas, tendo acesso as alterações realizadas nas demais réplicas do grupo (quadro 4), concluindo assim o processo de replicação de dados.

4.4.3 Resultados Experimentais

Para testar a funcionalidade e o desempenho dos métodos de replicação utilizados na ferramenta PGR, foram elaborados alguns testes simulando replicação de dados entre os bancos de dados distribuídos. Na figura 4.11, é mostrado o cenário pretendido de operação da aplicação. Nela, pode-se visualizar todas as entidades já conectadas, trocando dados de exames, imagens, vídeos e outras informações relevantes para os sistemas.

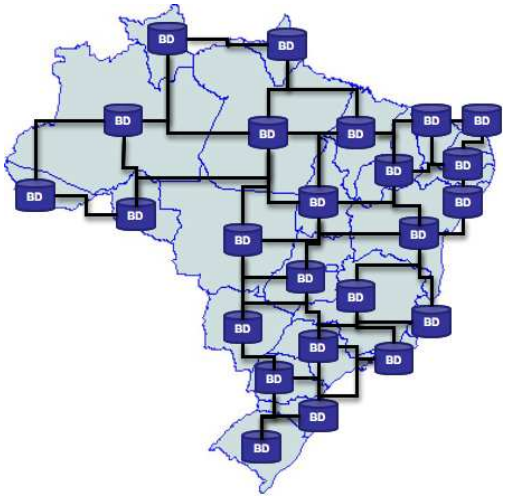


Figura 4.11: Cenário com os Bancos de Dados Médicos Distribuídos Integrados

Na elaboração dos testes de desempenho da aplicação, foram usados conjuntos de dados (*chunks*) de variados tamanhos, para a obtenção de métricas mais consistentes. Foram realizados vinte testes experimentais de replicação de dados entre as entidades remotas. No primeiro teste foram usados conjuntos de dados, compostos de objetos binários (*binary large objects*) e no segundo teste, dados em texto plano. Na tabela 4.3, pode-se observar os tamanhos dos conjuntos de dados utilizados nos testes.

Foram realizadas vinte (Anexo II e Anexo III) replicações de um conjunto de imagens médicas e cadastros simples, divididos em conjuntos de dados de 10, 100, 1.000 e 10.000 registros. Na experiência, foram medidos os tempos de transferência entre dois nodos do ambiente

Tabela 4.3: *Tamanho dos conjuntos de dados*

Tipo de Dados	10	100	1000	10.000
Texto Plano	8Kb	16Kb	104Kb	1,128Kb
Binário	0.34Mb	26.20Mb	275.07Mb	1,712Mb

computacional utilizado para testes, em frações de segundo para cada conjunto de dados. Foram utilizados recursos do sistema operacional do nodo responsável pela requisição da réplica para registrar o tempo no momento em que a *engine* controladora do processo inicia a transferência dos dados e do momento em que o sucesso da transação é confirmado.

Para o agrupamento dos registros em conjuntos, foram utilizadas tabelas temporárias no banco de dados do nodo que contém os dados originais para cada conjunto a ser transmitido. O controlador do nodo primário precisa obter a estrutura de tais tabelas para reproduzí-las em sua própria base e então replicar os dados de cada uma através dos procedimentos de cópia de tabelas do PostgreSQL. O tempo médio das transações, para os registros em formato texto e binário podem ser observados na tabela 4.4.

Tabela 4.4: *Tempo médio de replicação, em segundos*

Tipo de Dados	10	100	1000	10.000
Texto Plano	0.0048	0.0091	0.0170	0.1836
Binário	0.0665	9.6952	104.4264	627.1346

Entretanto, para auxiliar a visualização dos dados obtidos a partir da replicação, foi utilizado um algoritmo de normalização de amplitude nos dados obtidos, de acordo com a fórmula mostrada na figura 4.12. Este procedimento foi necessário para que se possa comparar os valores de forma igual, já que se tratam de valores em escalas diferentes.

$$y = \frac{|\bar{x} - x_i|}{x_{max}}$$

Figura 4.12: Fórmula de Normalização

A normalização foi obtida através da divisão do valor absoluto da diferença entre cada valor e a média das observações pelo valor máximo de cada categoria. A fórmula resulta em valores entre 0 e 1 que permitem a comparação dos dados obtidos a partir de variáveis com dispersões distintas. Esses valores foram multiplicados por 100 para criar uma unidade relativa de custo de transferência, e estão listados na tabela 4.5, para cada categoria de dados.

Tabela 4.5: *Custo de Replicação por Registro*

Tipo de Dados	10	100	1000	10.000
Texto Plano	18.25	3.40	7.46	7.39
Binário	0.97	0.47	0.59	0.08

A partir dos dados normalizados, que resultaram no *custo* de replicação por registro, pode-se ver na figura 4.13 um gráfico explanativo sobre os resultados das transações.

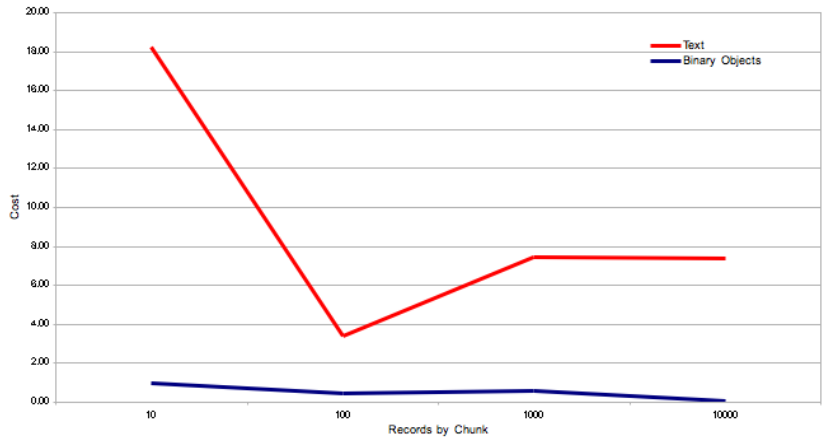


Figura 4.13: Gráfico de Desempenho do Processo de Replicação de Dados

Avaliando os resultados experimentais na figura 4.13 pode-se observar, quando usada a mesma escala, que há uma menor variação em relação a média de tempo de transferência em conjuntos de determinados tamanhos, como em conjunto de dados de texto de 100 registros em comparação com os de 10. Isso sugere que há a possibilidade de encontrar valores para a quantidade de registros, que possibilitem um menor custo em tempo de transferência de acordo com cada tipo de dados através da escolha do tamanho dos conjuntos utilizados pela ferramenta de replicação.

4.5 Sistemas de Arquivos Distribuídos

Neste cenário serão abordados os aspectos de Sistemas de Arquivos Distribuídos, aplicados para armazenamento de imagens médicas no formato DICOM. Será utilizado o HDF5 para hierarquização das imagens, e posterior armazenamento em um ambiente distribuído construído com o *Parallel Virtual File System*.

4.5.1 Descrição do Problema

Sistemas de Telemedicina em geral são compostos por imagens e videos de exames médicos. Estas informações são utilizadas pelos médicos para elaborar laudos. Estas imagens, na maioria dos casos estão no formato DICOM, que será o formato utilizado neste cenário, e tem tamanho significativo como pode ser constatado na tabela 4.2.

Como já foi citado na capítulo 3, um dos principais objetivos do CyclopsDCMServer, é armazenar e fornecer arquivos DICOM indexados em um repositório de dados gerenciados por um sistema de gerenciador de bancos de dados relacionais, tais como: PostgreSQL, MySQL, Oracle, SyBase, etc. No caso do CyclopsDCMServer, as imagens DICOM são tratadas em tempo de execução e segmentadas em dados do paciente e a imagem propriamente dita, e armazenadas no base de dados do Portal de Telemedicina. Após este tratamento, as imagens em formato DICOM, ficam armazenadas em outra base de dados, sendo raramente utilizadas.

Entretanto, por questões de legislação e regulações do CNM (Conselho Nacional de Medicina), elas não podem ser descartadas em um curto período de tempo e o seu armazenamento pode tornar-se um problema, pois muitas instituições não possuem infra-estrutura com capacidade de armazenamento de *gigabytes* ou até *terabytes* de dados ociosos.

Pensando nesta problemática, este cenário abordará o uso de sistemas de arquivos distribuídos, baseados em agregados computacionais para o armazenamento destas imagens. Entretanto, armazenar imagens em formato DICOM em formato “puro” impossibilitaria ou dificultaria uma futura recuperação das informações. Para isto, as informações serão hierarquizadas e armazenadas no formato de hierarquização de dados, HDF5.

4.5.2 Desenvolvimento da Proposta

Com o uso da tecnologia descrita nos capítulos 2 e 3, foi iniciada a preparação de um ambiente em que imagens médicas no formato DICOM pudessem ser armazenadas em um meio distribuído no formato HDF5 em adição aos métodos já utilizados, como gravação em banco de dados relacional e armazenamento de longo prazo em disco. Tal ambiente possibilita a realização de comparações, buscando analizar os custos de tempo de gravação e recuperação entre os diferentes métodos.

O primeiro passo para a preparação do ambiente, foi a criação de uma biblioteca composta de um objeto *wrapper* contendo métodos de criação e armazenamento de informações específicas de imagens DICOM, obtidas através do servidor de imagens. O objeto é utilizado como

um módulo a parte do CyclopsDCMServer e instanciado de forma similar aos outros métodos de armazenamento disponíveis no servidor (figura 4.14).

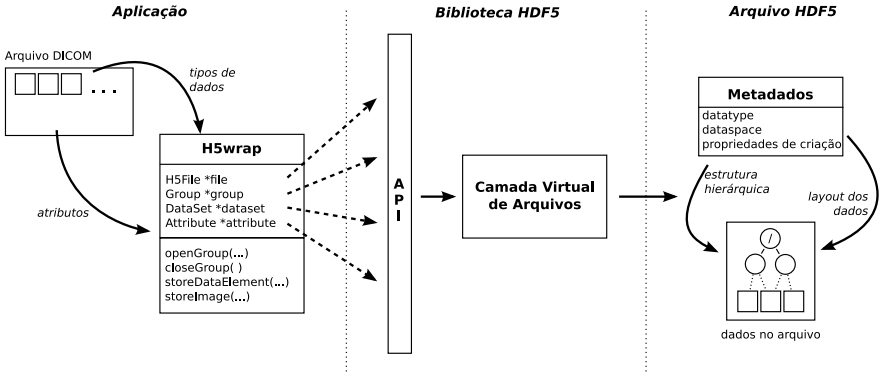


Figura 4.14: Modelo de Operação

Em seguida, foi construído um sistema compartilhado de arquivos entre os diversos nodos de um agregado computacional. Esse sistema foi obtido por meio da utilização do *Parallel Virtual File System* (PVFS) em cada um dos nodos do agregado e também no servidor de imagens médicas. No momento em que o CyclopsDCMServer solicita a criação de um novo arquivo HDF5, este é criado diretamente no ambiente compartilhado, como pode ser observado na figura 4.15.

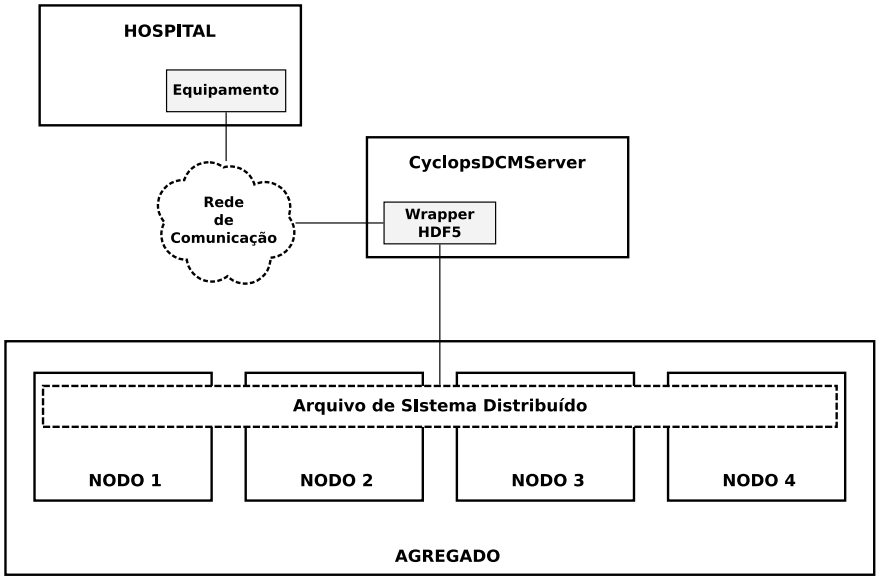


Figura 4.15: Arquitetura Proposta

Para a leitura inicial dos dados, foi necessário separá-los em duas categorias: elementos de dados comuns e imagem. Os elementos de dados comuns são informações sobre a imagem, como dados sobre o paciente (Nome, CPF, RG, etc.) para o qual a imagem foi gerada, dimensões e outras características da imagem, entre outras. No caso da imagem, é a representa-

ção binária da figura gerada pelo equipamento médico, como uma tomografia ou cardiograma, por exemplo. Para o armazenamento no formato HDF5 optou-se por referenciar todos os elementos de dados comuns como *strings*, enquanto as imagens deveriam ser armazenadas em uma representação numérica gerada pela API HDF5 Image, própria para este fim.

Outro aspecto considerado durante o projeto foi a estrutura hierárquica adotada para representar cada imagem dentro de um arquivo HDF5. Considerando as informações disponíveis dentro das imagens em formato DICOM, optou-se por utilizar a estrutura exibida na figura 4.16. As cinco camadas da hierarquia (abaixo do grupo raiz) possibilitam uma organização satisfatória para o experimento realizado, permitindo uma boa visualização e praticidade de acesso aos dados.

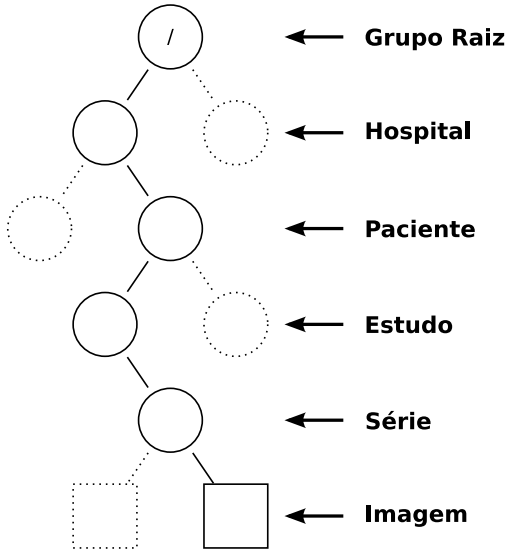


Figura 4.16: Hierarquização dos Dados no Formato HDF5

No CyclopsDCMServer, no momento da obtenção dos dados da imagem, a hierarquia é criada por meio de chamadas ao método do objeto *wrapper* que cria grupos, sendo necessário informar o “caminho” do grupo que está sendo criado no arquivo HDF5 e um nome, que será utilizado como identificador dentro do arquivo. Uma vez que o objeto esteja aberto no grupo desejado, dois métodos de acesso estão disponíveis para o armazenamento dos dois diferentes tipos de informações obtidos do arquivo original. Tais métodos também permitem que informações sobre esses dados sejam recebidos pelo objeto *wrapper* e associados às informações originais.

Uma vez que todas as informações de um determinado arquivo DICOM sejam obtidas e gravadas no arquivo HDF5 criado de início, um ou mais grupos são fechados, dependendo da imagem, e uma nova iteração é realizada para o armazenamento de uma nova imagem no mesmo arquivo. Quando o ciclo acaba, observa-se a existência de possíveis erros em sua criação, e em

caso negativo o processo de gravação iniciado pelo CyclopsDCMServer é encerrado.

4.5.3 Resultados Experimentais

Para avaliar o modelo proposto, decidiu-se por realizar um comparativo entre o armazenamento das imagens médicas no formato HDF5 e pelo método atualmente utilizado pelo CyclopsDCMServer, que é a gravação dos dados das imagens em banco de dados relacional. Neste banco de dados as informações sobre as imagens DICOM obedecem à seguinte estrutura: quatro tabelas para armazenar os elementos de dados comuns e meta-informações sobre as imagens DICOM e uma tabela com o objetivo de gravar os arquivos em formato binário, como *large objects*. A estrutura das imagens no arquivo HDF foi descrita anteriormente e pode ser observada na figura 4.16.

Para a realização dos experimentos sobre a arquitetura proposta anteriormante, um ambiente de testes foi criado de forma a atender aos requisitos necessários para a execução da ferramenta. Foi montado um agregado computacional composto por quatro nodos, cada um possuindo uma área compartilhada de disco rígido sob o PVFS.

O primeiro 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. Os arquivos foram originalmente organizados pelo nome do paciente, sendo que cada paciente pode possuir um ou mais estudos com diversas séries de imagens. As imagens são geradas por equipamentos de tomografia computadorizada, gerando figuras monocromáticas de 512 *pixels* de altura e largura que podem ser convertidas para para o formato JPEG.

Na tabela 4.6, é possível observar 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. Na figura 4.17 é possível visualizar o gráfico baseado nos valores das transferências.

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. Também é possível observar a menor variação nos tempos de armazenamento no sistema de arquivos distribuído. As médias de tempo foram de **858,73** segundos para a gravação no PVFS em formato HDF5 e de **1018,77** segundos para a gravação no banco da dados relacional.

Uma segunda experiência foi realizada para a obtenção dos tempos de acesso

Tabela 4.6: *Tempos de Armazenamento*

	HDF5	Bancos 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

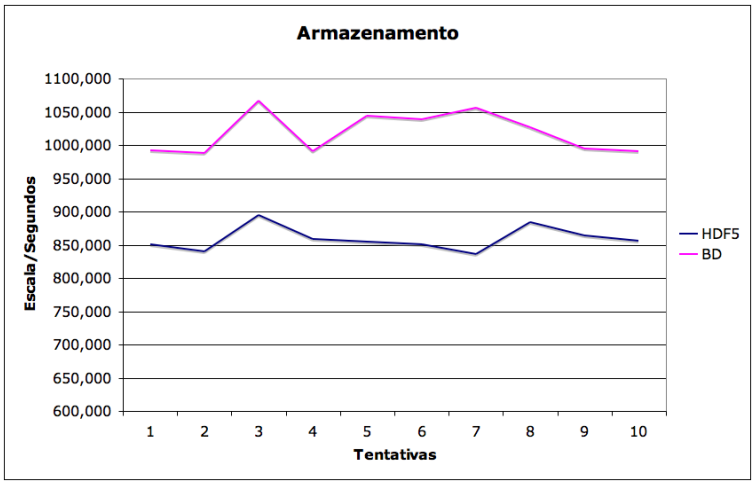


Figura 4.17: Gráfico de Desempenho de Armazenamento

a uma imagem específica, novamente em um banco de dados relacional e em um arquivo HDF5 gravado em um sistema de arquivos distribuídos. As tabelas utilizadas na consulta e o arquivo armazenado são as mesmas utilizadas na obtenção dos tempos de gravação dos dados.

Vinte iterações foram realizadas no acesso aos dados de uma imagem escolhida de forma aleatória para ser recuperada das duas estruturas de armazenamento. No banco de dados relacional, a consulta aos dados da imagem foi realizada de forma que ao final da consulta a imagem estivesse carregada, na memória principal do servidor, em formato binário, pronta para ser armazenada em disco.

Já no acesso ao arquivo no formato HDF5, devido ao fato de os dados não estarem armazenados em formato binário, foi criada uma extensão no objeto *wrapper* de forma que os

dados de uma imagem DICOM pudessem ser armazenados em uma estrutura de dados projetada apenas para recuperar para a memória os elementos de dados de uma imagem específica.

Os resultados dos acessos às imagens podem ser observados no Anexo 6, e um gráfico explanativo é apresentado na figura 4.18. É possível perceber que existe uma diferença constante entre os tempos obtidos a partir do banco de dados em comparação aos acessos ao arquivo HDF5, sendo que a vantagem agora passa a ser do banco de dados relacional.

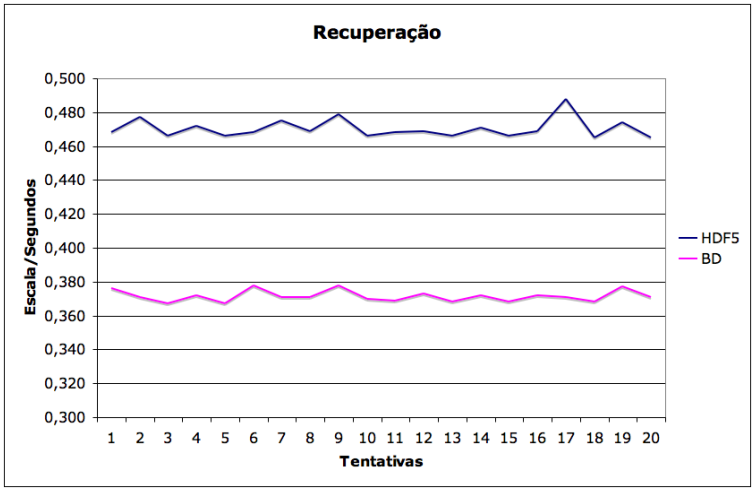


Figura 4.18: Gráfico de Desempenho da Recuperação

É importante ressaltar que a obtenção dos tempos de armazenamento e recuperação dos dados entre as duas diferentes tecnologias é influenciada pelos equipamentos utilizados em cada método empregado. Dessa forma, as diferenças na quantidade de memória e capacidade de processamento nos ambientes computacionais utilizados em cada caso devem ser levadas em consideração quando forem realizadas comparações utilizando as métricas obtidas nos experimentos descritos neste trabalho.

4.6 Considerações Finais

Neste capítulo foram avaliados três cenários de sistemas distribuídos aplicados em sistemas de telemedicina, sendo sendo o primeiro um cenário com agregados computacionais no Portal de Telemedicina visando prover desempenho, disponibilidade e escalabilidade ao sistema. No segundo cenário foram aplicadas técnicas de bancos de dados distribuídos e replicação para prover a integração do conhecimento médico embutido nos bancos de dados distribuídos. Por fim, no terceiro cenário foi avaliado o armazenamento de imagens médicas do formato DICOM, em formato HDF5, em sistemas de arquivos distribuídos.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho foram estudadas estratégias de sistemas distribuídos aplicadas a sistemas de telemedicina. Foram elaborados três cenários para a aplicação destas estratégias, sendo eles: agregados computacionais, replicação entre bancos de dados distribuídos e armazenamento de imagens médicas no formato HDF5 em sistemas de arquivos distribuídos.

No primeiro cenário foram comparados o desempenho e a escalabilidade do cenário atual do Portal de Telemedicina em relação com um cenário usando agregados computacionais no mesmo sistema. Foram avaliados os tempos de pedidos de conexão, os tempos de estabelecimento de conexão, o número de transações de páginas por segundo, número máximo de usuários conectados por segundo e o número máximo de usuários simultaneamente conectados por segundo, de ambos os cenários. Em todos os testes propostos, o uso de agregados computacionais resultou em um desempenho superior em comparação a cenários sem agregados, o que sugere a viabilidade do uso de agregados computacionais em sistemas de telemedicina.

A título de trabalhos futuros para este cenário, poderiam ser estudados os aspectos relacionados a, de que forma prover uma maior escalabilidade e disponibilidade ao sistema. Poderia ser inserida uma quantidade maior de máquinas ao agregado e analisar os resultados, comparando com os resultados atuais obtidos. Desta forma seria possível retirar as métricas para traçar a relação entre o número de máquinas e um maior desempenho. Um outro estudo possível, seria a inserção de uma camada de alta disponibilidade no Nodo Mestre do sistema, para no caso de falhas, o sistema não se tornar indisponível.

No segundo cenário foi apresentado um modelo de interoperabilidade baseado em replicação assíncrona entre bancos de dados médicos distribuídos. Como prova de conceito, no intuito de validar o modelo, foi desenvolvida uma *engine* que tem a missão de gerenciar todas

as operações de integração e replicação das informações entre os bancos de dados. Foram realizados testes experimentais com a ferramenta, onde ela se mostrou estável e funcional, atingindo os objetivos esperados. Como resultado teórico do cenário, foi validado o uso de técnicas de replicação otimista e parcial, com fragmentação híbrida em um ambiente de replicação assíncrona com bancos de dados distribuídos.

Como resultado prático deste trabalho, esta ferramenta provê a possibilidade das aplicações e sistemas de telemedicina a serem futuramente implantadas no Brasil, terem todos os dados dos pacientes integrados, facilitando a tomada de decisão de profissionais da área médica, consolidando o histórico dos pacientes e evitando redundâncias de exames, o que resultará diretamente em redução de custos e indiretamente a melhores tratamentos médicos.

Nos testes experimentais realizados, as métricas retiradas baseadas nos tempos de replicação entre os sites foram satisfatórias. É importante notar que a replicação dos conjuntos de dados de 100 registros de texto plano tiveram um melhor desempenho, em relação aos seus pares. Para conjuntos de dados binários, os compostos por 10.000 registros tiveram um melhor desempenho em relação aos seus pares, o que sugere que quanto maior a quantidade de registros binários a serem replicados, melhor o desempenho da replicação.

No cenário de bancos de dados distribuídos ainda existem muitos desafios a serem resolvidos como trabalhos futuros. Como exemplo, pode-se citar os aspectos que envolvem a consistência dos dados e a detecção e resolução de conflitos. Outro exemplo é o desenvolvimento de mecanismos que assegurem a integridade dos dados, permitindo que o site possa saber em determinado momento se sua cópia dos dados está consistente com os dados dos demais nodos do sistema. Com tais tópicos resolvidos, a aplicação pode se tornar uma boa candidata à grande parte das situações que envolvam a necessidade de replicação parcial e assíncrona de dados entre bancos de dados distribuídos.

No terceiro e último estudo proposto, onde é avaliado o armazenamento de imagens médicas em sistemas de arquivos distribuídos, constatou-se que mesmo não utilizando acesso e escrita paralelos de imagens DICOM em arquivos HDF5, e mesmo não tendo realizado comparações entre as diversas formas de utilização do formato, a fim de escolher os tipos de dados mais adequados para cada situação. O desempenho do formato de dados hierárquico em casos de armazenamento de imagens médicas, é um pouco melhor que a solução atual baseada em bancos de dados relacionais, entretanto, quanto a recuperação das imagens o sistema atual se mostrou melhor.

Estudos mais profundos em relação a paralelização das consultas realizadas no

arquivo HDF5 sobre o sistema de arquivo distribuído, sugerem uma melhora significativa no desempenho do sistema, visto que, da mesma forma, a aplicação de métodos de escrita paralela no ambiente distribuído também pode melhorar significativamente o desempenho do armazenamento. Adicionalmente, um outro estudo possível seria a utilização de armazenamento das imagens em formato HDF5 em malhas computacionais, utilizando de software para armazenamento distribuídos, tal como o GridFTP. Para a recuperação das imagens, seria possível utilizar *webservices* para intermediar as operações entre as entidades do sistema.

Por fim, a contribuição principal deste trabalho foi constatar que o uso de estratégias de sistemas distribuídos aplicadas a sistemas de telemedicina são, além de viáveis, necessárias para uma maior disseminação do conhecimento embutido neste tipo de sistema.

Referências Bibliográficas

[AFS 08] **OpenAFS**. Disponível em: http://www.openafs.org/. Acessado em: 16/01/2008.

[ATA 08] **ATA**. Disponível em: http://www.atmeda.org/. Acessado em: 16/01/2008.

[BAS 02] BASHSHUR, R. L. Telemedicine and health care. **Telemedicine Journal and E-Health**, [S.l.], v.8, 2002.

[BEL 96] BELLATRECHE, L.; SIMONET, A.; SIMONET, M. Vertical fragmentation in distributed object database systems with complex attributes and methods. In: DEXA WORKSHOP, 1996. [s.n.], 1996. p.15–21.

[BEO 08] **Beowulf Cluster - Project Overview**. Disponível em: http://www.beowulf.org. Acessado em: 16/01/2008.

[BER 78] BERNSTEIN, P. A. et al. The concurrency control mechanism of sdd-1: A system for distributed databases (the fully redundant case). **IEEE Transaction Software Engineering**, Piscataway, NJ, USA, v.4, n.3, p.154–168, 1978.

[BER 81] BERNSTEIN, P. A.; GOODMAN, N. Concurrency control in distributed database systems. **ACM Computer Surveys**, New York, NY, USA, v.13, n.2, p.185–221, 1981.

[BER 84] BERNSTEIN, P. A.; GOODMAN, N. An algorithm for concurrency control and recovery in replicated distributed databases. **ACM Transaction on Database Systems**, New York, NY, USA, v.9, n.4, p.596–615, 1984.

[BER 87] BERNSTEIN, P. A.; HADZILACOS, V.; GOODMAN, N. **Concurrency Control and Recovery in Database Systems**. Addison-Wesley, 1987.

[BUY 99] Buyya, R., editor. **High Performance Cluster Computing**, v.1, Architectures and Systems. Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[CAR 00] CARNS, P. H. et al. Pvfs: a parallel file system for linux clusters. In: ALS'00: PROCEEDINGS OF THE 4TH CONFERENCE ON 4TH ANNUAL LINUX SHOWCASE & CONFERENCE, ATLANTA, 2000. **Proceedings...** Berkeley, CA, USA: USENIX Association, 2000. p.28–28.

[CHO 06] CHOI, Y. et al. Telemedicine in the usa: standardization through information management and technical applications. **Communications Magazine, IEEE**, [S.l.], v.44, n.4, p.41–48, April 2006.

[CLA 03] CLARK, D. D. Economics and the design of open systems. **IEEE Internet Computing**, [S.l.], v.7, n.2, p.94–96, 2003.

[COD 08] **CODA**. Disponível em: http://www.coda.cs.cmu.edu. Acessado em: 16/01/2008.

[COH 06] COHEN, S. et al. Scientific formats for object-relational database systems: a study of suitability and performance. **SIGMOD Record**, [S.l.], v.35, n.2, p.10–15, 2006.

[COR 87] CORNELL, D. W.; YU, P. S. A vertical partitioning algorithm for relational databases. In: PROC. IEEE INT'L. CONF. ON DATA ENG., 1987. **Proceedings...** Los Angeles, CA: [s.n.], 1987. p.30.

[COU 04] COULON, C.; PACITTI, E.; VALDURIEZ, P. Scaling up the preventive replication of autonomous databases in cluster systems. In: Daydé, M. J. et al., editors, VECPAR, 2004. Springer, 2004. v.3402 of **Lecture Notes in Computer Science**, p.170–183.

[COU 05a] COULON, C.; PACITTI, E.; VALDURIEZ, P. Consistency management for partial replication in a high performance database cluster. In: ICPADS, 2005. IEEE Computer Society, 2005. p.809–815.

[COU 05b] COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Distributed Systems**. Boston: Addison-Wesley, 2005.

[CYC 08] **The Cyclops Project**. Disponível em: http://www.cyclops.ufsc.br. Acessado em: 16/01/2008.

[DAN 05] DANTAS, M. **Computao Distribuda de Alto Desempenho**. 1. ed. Rio de Janeiro: Axcel Books, 2005.

[DEL 05] DELLANI, P. R. **Desenvolvimento de um Servidor de Imagens Médicas Digitais no Padrão DICOM**. Universidade Federal de Santa Catarina, 2005. Dissertação de Mestrado.

[ELM 99] ELMASRI, R. A.; NAVATHE, S. B. **Fundamentals of Database Systems**. 3. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[EMM 94] EMMERICH, P. et al. Benefits, problems and issues in open systems architectures. **IEEE Transactions on Power Systems**, [S.l.], v.9, n.1, 1994.

[GHE 03] GHEMAWAT, S.; GOBIOFF, H.; LEUNG, S.-T. The google file system. In: PROCEEDINGS OF THE NINETEENTH ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 2003. **Proceedings...** New York: ACM Press, 2003. v.37, 5 of **Operating Systems Review**, p.29–43.

[GOE 98] GOEL, A.; PU, C.; POPEK, G. J. View consistency for optimistic replication. In: SYMPOSIUM ON RELIABLE DISTRIBUTED SYSTEMS, 1998. [s.n.], 1998. p.36–42.

[GOS 06] GOSINK, L. J. et al. HDF5-fastquery: Accelerating complex queries on HDF datasets using fast bitmap indices. In: SSDBM, 2006. IEEE Computer Society, 2006. p.149–158.

[HDF 08] **HDF5 – Hierarchical Data Format 5. User's Guide.** Disponível em: http://hdf.ncsa.uiuc.edu/products/hdf5/. Acessado em: 16/01/2008.

[HEA 08] **Heart Beat**. Disponível em: http://www.beowulf.org.

[IBM 08] **A Pratical Guide to DB2 UDB Data Replication V8**. Disponível em: http://www.ibm.com. Acessado em: 16/01/2008.

[ISO 95] ISO/IEC, editor. **Basic reference model of open distributed processing: Overview**. Junho, 1995.

[JAH 05] JAHNT, K.; REIHER, M.; STUHL, T. Telemedical projects in bavaria: what is the current position and what needs to be done. In: INTERNATIONAL CONGRESS SERIES 1281, 2005. [s.n.], 2005.

[KIS 92] KISTLER, J. J.; SATYANARAYANAN, M. Disconnected operation in the coda file system. **ACM Trans. Comput. Syst.**, New York, NY, USA, v.10, n.1, p.3–25, 1992.

[LEE 00] LEE, K.; HUNT, A. view–hdf: Visualization and analysis tool for hierarchical data format files. 2000. [s.n.], 2000. p.03.

[LUH 05] LUH, J. et al. Evaluation of the low cost telemedicine system in taiwan. In: ENTERPRISE NETWORKING AND COMPUTING IN HEALTHCARE INDUSTRY, 2005. HEALTHCOM 2005. PROCEEDINGS OF 7TH INTERNATIONAL WORKSHOP ON, 2005. [s.n.], 2005. p.316–319.

[LVS 08] **LVS - Linux Virtual Server**. Disponível em: http://www.linuxvirtualserver.org. Acessado em: 16/01/2008.

[MAI 06] MAIA, R.; VON WANGENHEIM, A.; NOBRE, L. A statewide telemedicine network for public health in brazil. In: COMPUTER-BASED MEDICAL SYSTEMS, 2006. CBMS 2006. 19TH IEEE INTERNATIONAL SYMPOSIUM ON, 2006. [s.n.], 2006. p.495–500.

[MCN 02] MCNEILL, K. M.; WEINSTEIN, R. S.; HOLCOMB, M. J. Arizona telemedicine program: Implementing a statewide health care network. [S.l.], Janeiro 18, 2002.

[MEE 95] MEER, J. The ISO reference model for open distributed processing. **Computer Networks and ISDN Systems**, [S.l.], v.27, n.8, p.1211–1214, 1995.

[MIJ 04] MIJARES, M. Telemedicine in ecuador: failure or a learning experience? In: ENTERPRISE NETWORKING AND COMPUTING IN HEALTHCARE INDUSTRY, 2004. HEALTHCOM 2004. PROCEEDINGS. 6TH INTERNATIONAL WORKSHOP ON, 2004. [s.n.], 2004. p.41–43.

[NAV 84] NAVATHE, S. et al. Vertical partitioning algorithms for database design. **ACM Transactions on Database Systems**, [S.l.], v.9, p.680–710, 1984.

[OPE 08a] **openMOSIX - An Open Source Linux Cluster Projetct**. Disponível em: http://openmosix.sourceforge.net. Acessado em: 16/01/2008.

[OPE 08b] **OpenSSI - Open Single System Image Clusters for Linux**. Disponível em: http://www.openssi.org. Acessado em: 16/01/2008.

[ORA 08] **Oracle - Oracle Real Application Clusters (RAC) 11g**. Disponível em: http://www.oracle.com. Acessado em: 16/01/2008.

[ÖZS 99] ÖZSU, M. T.; VALDURIEZ, P. **Principles of Distributed Database Systems, Second Edition**. Prentice-Hall, 1999.

[PAL 05] PAL, A. et al. Telemedicine diffusion in a developing country:the case of india (march 2004). **Information Technology in Biomedicine, IEEE Transactions on**, [S.l.], v.9, n.1, p.59–65, March 2005.

[PAS 03] PASIN, M.; KREUTZ, D. L. **Arquitetura e Administração de Aglomerados**. Escola Regional de Alto Desempenho, 2003. 4–34 p.

[PET 96] PETERSEN, M. et al. Telemedicine in utah: the rural utah telemedicine pilot project. In: FRONTIERS IN EDUCATION CONFERENCE, 1996. FIE '96. 26TH ANNUAL CONFERENCE., PROCEEDINGS OF, 6-9 Nov 1996. [s.n.], 6-9 Nov 1996. v.2, p.986–989 vol.2.

[PGC 08] **PGCluster - The multi-master and synchronous replication system for PostgreSQL**. Disponível em: http://pgcluster.projects.postgresql.org.

[PGP 08] **PGPool**. Disponível em: http://pgpool.projects.postgresql.org. Acessado em: 16/01/2008.

[POS 08] **Postgres-R - Eager multi-master replication for Postgres**. Disponível em: http://www.postgres-r.org. Acessado em: 16/01/2008.

[PVF 08] **PVFS – Parallel Virtual File System**. Disponível em: http://www.pvfs.org. Acessado em: 16/01/2008.

[ROS 01] ROSS, R. et al. A case study in application I/O on linux clusters. In: SC'2001 CONFERENCE CD, 2001. **Proceedings...** Denver: ACM SIGARCH/IEEE, 2001.

[ROT 80] ROTHNIE, J. B., J. et al. Introduction to a system for distributed databases (sdd-1). **ACM Trans. Database Syst.**, New York, NY, USA, v.5, n.1, p.1–17, 1980.

[ROU 05] ROUSSOS, G.; MARSH, A.; MAGLAVERA, S. Enabling pervasive computing with smart phones. **IEEE Pervasive Computing**, [S.l.], p.39–47, 2005.

[SAI 05] SAITO, Y.; SHAPIRO, M. Optimistic replication. **ACM Computing Surveys**, [S.l.], v.37, 2005.

[SAT 90] SATYANARAYANAN, M. et al. Coda: A highly available file systems for a distributed workstation environment. **IEEE Transactions on Computers (IEEE TOC)**, [S.l.], v.C-39, n.4, p.447–459, Abril, 1990.

[SHA 04a] SHARMAN, D. et al. The first telemedicine project in kazakhstan. In: COMPUTER-BASED MEDICAL SYSTEMS, 2004. CBMS 2004. PROCEEDINGS. 17TH IEEE SYMPOSIUM ON, 2004. [s.n.], 2004. p.440–445.

[SHA 04b] SHASHARINA, S. G.; WANG, N.; CARY, J. R. Grid service for visualization and analysis of remote fusion data. In: CLADE, 2004. IEEE Computer Society, 2004. p.34.

[SLO 08] **Slony-I - Enterprise-level Replication System**. Disponível em: http://www.slony.info. Acessado em: 16/01/2008.

[SMI 86] SMITH, J. M. et al. Multibase Ñ- integrating heterogeneous distributed database systems. In: ON AFIPS CONFERENCE PROCEEDINGS; VOL. 55 1986 NATIONAL COMPUTER CONFERENCE, 1986. **Proceedings...** Arlington, VA, USA: AFIPS Press, 1986. p.335–347.

[STE 95] STERLING, T. et al. BEOWULF: A parallel workstation for scientific computation. In: PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, 1995. **Proceedings...** Boca Raton, FL: CRC Press, 1995. v.I, Architecture, p.I:11–14.

[SYB 08] **SyBase - Replication Server: Move and synchronize data across the enterprise with Replication Server**. Disponível em: http://www.sybase.com. Acessado em: 16/01/2008.

[TAN 02] TANENBAUM, A. S.; VAN STEEN, M. **Distributed systems: principles and paradigms**. Prentice Hall, 2002.

[TEL 08] **Telemedicina**. Disponível em: http://www.telemedicina.ufsc.br. Acessado em: 16/01/2008.

[TOB 94] TOBBICKE, R. Distributed file systems: focus on andrew file system/distributed file service (afs/dfs). **Proceedings of Thirteenth IEEE Symposium on Mass Storage Systems. 'Towards Distributed Storage and Data Management Systems.'**, [S.l.], p.23–26, 1994.

[TOM 04] TOMIOKA, Y. et al. Survey on patent applications for medical communications and telemedicine in japan. In: ENTERPRISE NETWORKING AND COMPUTING IN HEALTHCARE INDUSTRY, 2004. HEALTHCOM 2004. PROCEEDINGS. 6TH INTERNATIONAL WORKSHOP ON, 2004. [s.n.], 2004. p.121–124.

[TSU 08] **Tsung**. Disponível em: http://tsung.erlang-projects.org/. Acessado em: 16/01/2008.

[WAN 97] WANGENHEIM, A. et al. Cyclops - expert system shell for the development of applications in the area of medical image analysis. In: PROCEEDINGS OF THE 4TH GERMAN-BRAZILIAN WORKSHOP ON INFORMATION TECHNOLOGY, 1997. **Proceedings...** Porto Alegre: [s.n.], 1997.

[XUE 07] XUE, Y.; LIANG, H. Analysis of telemedicine diffusion: The case of china. **Information Technology in Biomedicine, IEEE Transactions on**, [S.l.], v.11, n.2, p.231–233, 2007.

[YU 06] YU, H. et al. High performance file i/o for the blue gene/l supercomputer. In: THE TWELFTH INTERNATIONAL SYMPOSIUM ON HIGH-PERFORMANCE COMPUTER ARCHITECTURE, 2006. [s.n.], 2006.

[ZHA 00] ZHANG, W. Linux virtual server for scalable network services. In: PROCEEDINGS OF THE OTTAWA LINUX SYMPOSIUM 2000, 2000. [s.n.], 2000.

ANEXO 1 - Publicações

– ARTIGOS APROVADOS

Título: Dinâmica da Digitação Aplicada a Ambientes Web.
Autores: PAVEZI, R.; MACEDO, D. D. J.; ANDRADE, R.; WANGENHEIM, A. v.
Evento: VII Simpósio Brasileiro em Seguranạ da Informação e de Sistemas Computacionais / I Workshop de Trabalhos de Iniciação Científica e de Graduação
Sigla: SBSeg / WTICG
Local: Rio de Janeiro – RJ – Brasil.
Ano: 2007
Qualis CC: B Nacional
Prêmio: 1o Lugar no Concurso de Melhor Artigo. Indicação para publicação na Revista Eletrônica de Iniciação Científica – REIC (submetido e aguardando revisão por pares).

Título: Definição de Métricas de Acordos Sec–SLA em Conformidade com as Normas Internacionais de Seguranạ.
Autores: BARBOSA, P.; RIGHI, R.; KREUTZ, D. L.; MACEDO, D. D. J.
Evento: XXXIII Conferência Latinoamericana de Informática
Sigla: CLEI
Local: San Jose – Costa Rica.
Ano: 2007
Qualis CC: B Nacional

Título: Um Estudo Comparativa de Ferramentas Baseadas em Código Livre para o Controle e Monitoramento de Redes.
Autores: MACEDO, D. D. J.; MATOS, A.; RIGHI, R. R.; KREUTZ, D. L.; DANTAS, M. A. R.

Evento: V Escola Regional de Redes de Computadores

Sigla: ERRC

Local: Santa Maria – RS – Brasil.

Ano: 2007

Qualis CC: Não Avaliado.

Título: Creating a Statewide Public Health System starting from Telemedicine Network.

Autores: WALLAUER, J.; MACEDO, D. D. J.; ANDRADE, R.; WANGENHEIM, A. v.

Journal: IEEE IT Professional Journal

Sigla: IEEE ITPro

Número: 2

Volume: 10

Mês: Março

Ano: 2008

Qualis CC: Não Avaliado.

Título: Replicação Assíncrona entre Bancos de Dados Médicos Distribuídos

Autores: MACEDO, D. D. J.; PERANTUNES, H. W. G.; MAIA, L. F. J.; WANGENHEIM, A. v.; DANTAS, M. A. R.

Evento: IV Escola Regional de Bancos de Dados

Sigla: ERBD2008

Local: Florianópolis – SC – Brasil.

Ano: 2008

Qualis CC: Não Avaliado.

Título: Asynchronous Data Replication: A National Integration Strategy for Databases on Telemedicine Network

Autores: MACEDO, D. D. J.; PERANTUNES, H. W. G.; ANDRADE, R.; MAIA, L. F. J.; WANGENHEIM, A. v.; DANTAS, M. A. R.

Evento: The 21Th IEEE International Symposium on Computer-Based Medical Systems

Sigla: CBMS2008

Local: X – Finlândia.

Ano: 2008

Qualis CC: Qualis A Internacional

Título: An Interoperability Approach Based on Asynchronous Replication among Distributed Internet Database

Autores: MACEDO, D. D. J. ; PERANTUNES, H. W. G. ; MAIA, L. F. J. ; WANGENHEIM, A. V. ; DANTAS, M.A.R.

Evento: The IEEE Symposium on Computers and Communications

Sigla: ISCC2008

Local: Marrakeche - Marrocos.

Ano: 2008

Qualis CC: Qualis A Internacional

ANEXO 2 - Tempo de Replicação de Texto Plano

Tabela 5.1: *Tempo das Operações de Replicação de Texto Plano*

Texto Plano			
10	**100**	**1000**	**10.000**
0,008097	0,002498	0,016295	0,194715
0,001351	0,002522	0,016698	0,176752
0,069604	0,139961	0,036498	0,174085
0,000954	0,002131	0,015190	0,162571
0,000965	0,002291	0,016071	0,175140
0,000975	0,002434	0,016126	0,175786
0,001006	0,002153	0,016159	0,159776
0,000997	0,002251	0,016079	0,176684
0,000993	0,002271	0,016258	0,174766
0,000949	0,002204	0,015111	0,160495
0,001999	0,002128	0,016113	0,176641
0,000994	0,002299	0,016076	0,175574
0,001043	0,002135	0,015421	0,164103
0,001001	0,002269	0,015998	0,181371
0,001116	0,002167	0,016095	0,175645
0,000983	0,002156	0,016316	0,171742
0,001102	0,002250	0,016087	0,174928
0,000970	0,002295	0,016133	0,173645
0,001746	0,002246	0,016090	0,187447
0,000967	0,002269	0,016074	0,360577

ANEXO 3 - Tempo de Replicação de Binários

Tabela 5.2: *Tempo das Operações de Replicação de Texto Plano*

Binários			
10	**100**	**1000**	**10.000**
0,071243	9,167423	107,661195	629,568098
0,072289	8,741315	108,583590	635,222137
0,172075	10,388399	108,333777	633,308550
0,056817	9,172938	106,195820	605,662033
0,057632	8,606416	99,918583	614,252065
0,062365	10,198111	107,244779	623,227692
0,057268	8,467666	100,893021	610,137883
0,057400	10,007583	101,865754	606,454484
0,062434	9,856839	101,882570	635,674873
0,057178	8,706079	108,174354	605,447847
0,057271	9,664754	103,355253	629,678288
0,064917	8,882488	101,418856	624,229319
0,057359	8,443993	101,347067	601,105453
0,057724	9,827734	103,176704	683,428129
0,062318	12,372667	101,014167	630,607216
0,056921	10,676873	102,780165	615,804214
0,057614	10,030553	104,436173	693,909820
0,074960	10,037250	107,942872	613,471973
0,056858	9,308499	106,674676	618,190911
0,058026	11,346591	105,630550	633,311968

ANEXO 4 - HDF5 - Conceitos

Neste anexo serão mostrados detalhes dos conceitos envolvidos no modelo de dados e de programação do HDF5. Conceitos estes que foram retirados da documentação oficial do site [HDF 08].

Arquivo

De forma abstrata, um arquivo HDF5 é um recipiente para um conjunto organizado de objetos. Tais objetos são grupos, datasets ou outros objetos. Cada arquivo HDF5 possui pelo menos um objeto, o grupo raiz, como pode ser visualizado na figura 5. Todos os objetos são membros do grupo raiz ou descendentes do mesmo. Objetos HDF5 possuem ao menos uma identidade nica dentro de um arquivo HDF5, e podem ser acessados apenas por meio dos nomes na hierarquia do arquivo. Objetos HDF5 em diferentes arquivos não possuem necessariamente identidades únicas e não é possível acessar um objeto HDF5 permanente exceto por meio de um arquivo.

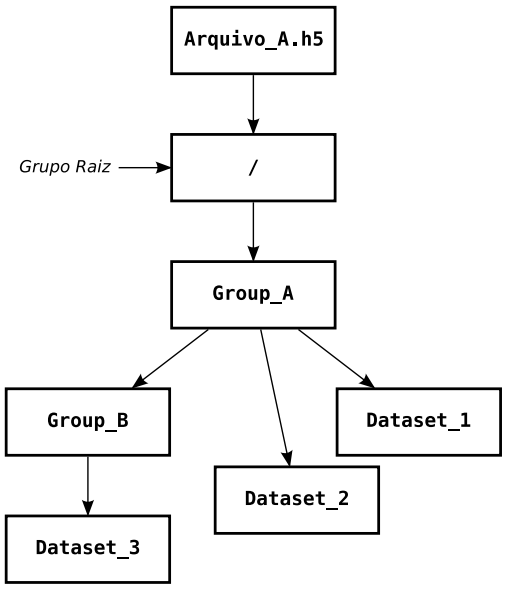


Figura 5.1: Exemplo de Arquivo no Formato HDF5

Quando o arquivo é criado, as propriedades de criação especificam suas características, que incluem informações sobre a versão e parâmetros globais de dados. Quando o arquivo é aberto, as propriedades de acesso do arquivo especificam as características para aquela instância específica de acesso, além de incluírem parâmetros para os métodos de armazenamento e para coleta de lixo. As propriedades de criação de arquivo são permanentes, enquanto as de acesso podem ser alteradas fechando e abrindo novamente o arquivo.

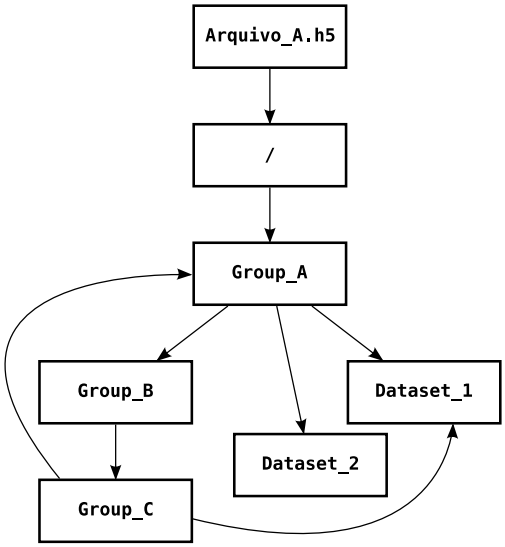


Figura 5.2: Relacionamento Entre os Modelos

Um arquivo HDF5 pode ser montado como parte de outro arquivo HDF5 de forma análoga à forma de montagem utilizada pelo sistema de arquivos do Unix, como pode ser constatado na figura 5. A raiz do arquivo montado é referenciada no grupo do arquivo em que a montagem é solicitada e todo o seu conteúdo pode ser acessado como se o arquivo montado fosse parte do primeiro arquivo.

Grupo

Um grupo HDF5 é análogo a um sistema de arquivos baseado em diretórios. De forma abstrata, um grupo contém zero ou mais objetos, e todo objeto precisa ser membro de pelo menos um grupo (exceto o grupo raiz, que não precisa pertencer a outro grupo). Essa relação implementada através de objetos de ligação, que pertencem a um grupo e aponta para um “nome de objeto”. Cada objeto de ligação possui um nome e aponta para apenas um objeto, enquanto cada “nome de objeto” tem um ou mais objetos de ligação o referenciando.

Dataset

O dataset é um array de dados multi-dimensional (retangular) de elementos de dados. O formato específico do array (número de dimensões e seu tamanho) é descrito por um objeto dataspace. Um elemento de dados é uma unidade atômica de dados que pode ser um número, um caractere ou um registro de elementos de dados heterogêneos. Um elemento de dados é um conjunto de bits, o layout dos bits descrito pelo datatype.

O dataspace e o datatype são definidos quando o dataset é criado, e não podem ser modificados posteriormente. As propriedades de criação do dataset são definidas quando o mesmo é criado. O objeto dataset gerencia o armazenamento e acesso aos dados. Apesar de os dados estarem conceitualmente armazenados em um array contíguo retangular, eles são fisicamente armazenados e transferidos de diferentes formas dependendo das propriedades e do mtodo de armazenamento utilizado. O dataset realiza um mapeamento entre o array conceitual de elementos e os dados armazenados fisicamente.

Dataspace

Dataspaces descrevem o layout dos elementos de um array multi-dimensional. Conceitualmente, o array é um hiper-retângulo possuindo de uma a 32 dimensões, podendo ser extendido. Cada dimensão possui um tamanho atual e um tamanho máximo, que pode ser ilimitado.

Eles também podem ser utilizados para descrever seleções hyperslab de um dataset. Qualquer subconjunto dos elementos de um dataset pode ser selecionado para leitura ou escrita especificando um conjunto de hyperslabs. Uma região não-retangular pode ser selecionada através da união de vários dataspaces (que são retangulares).

Datatype

Um datatype descreve o layout de um único elemento de dados. Um elemento de dados é um único elemento do array; pode ser um único número, um caractere, um array de números ou outros dados. Datatypes são categorizados em 11 classes. Cada classe é interpretada de acordo com um conjunto de regras e possui um conjunto específico de propriedades que descrevem sua forma de armazenamento. Por exemplo, números de ponto flutuante possuem características que são interpretadas de acordo com os padrões apropriados para a representação desse tipo de número. Dessa

forma, a classe de datatype pode informar o significado do elemento, enquanto o Datatype descreve como ele é armazenado.

Estre as classes possíveis para utilização por um datatype estão números inteiros, números de ponto flutuante, caracteres, representações de data e hora, sequências de bits (bitfield) e sequências de bytes não interpretados armazenados em bloco (opaque). Datatypes podem ser atômicos ou compostos de múltiplos elementos de datatypes atômicos. Além dos datatypes padrão, usuários podem também definir outros tipos, como inteiros de 24 bits ou números de ponto flutuante de 16 bits.

Um dataset ou atributo têm um único objeto datatype associado. O datatype pode ser usado na definição de vários objetos, mas por padrão, uma cópia do datatype será privativa para o dataset. Opcionalmente, um datatype pode ser armazenado em um arquivo HDF5. O datatype é referenciado em um grupo, e consequentemente a ele é dado um nome. Um datatype com nome pode ser aberto e utilizado de qualquer modo que um datatype possa ser usado.

Atributo

Qualquer objeto HDF5 que possua um nome (grupos, datasets ou named datatyes) pode ter zero ou mais atributos definidos, que são utilizados para documentar o objeto. Os atributos são armazenados dentro do objeto, possuindo um nome e armazenando dados que são descritos de forma análoga a um dataset: um dataspace define o layout do array de elementos de dados e um datatype define a forma de armazenamento e interpretação de tais elementos.

Dessa forma, um atributo é muito similar a um dataset, mas com as seguintes limitações:

- Um atributo pode apenas ser acessado através do objeto, sendo que nomes de atributos são significantes apenas dentro de tal objeto. Atributos não podem ser compartilhados;
- Por razões práticas, um atributo deve ser um objeto pequeno, com não mais de 1000 bytes;
- Os dados de um atributo precisam ser lidos ou gravados em um único acesso, sendo que não é permitido selecionar subconjuntos dos mesmos;
- Atributos não podem possuir atributos.

O valor de um atributo pode ser um objeto referência, de forma que um atributo

compartilhado ou um atributo com uma grande quantidade de informações pode ser implementado por meio de uma referência a um Dataset.

O nome, dataspace e datatype de um atributo são especificados em sua criação e não podem ser alterados posteriormente. O atributo pode ser aberto utilizando seu nome, índice ou realizando uma interação através de todos os atributos do objeto.

Lista de Propriedades

O HDF5 possui um objeto que representa uma lista genérica de propriedades, sendo uma coleção de pares (nome, valor). Cada classe da lista de propriedades possui um conjunto específico de propriedades. Cada propriedade possui um nome implícito, um datatype HDF5 e um valor. Uma lista de propriedades é criada e utilizada de forma similar aos outros objetos da biblioteca. Algumas propriedades são permanentes, como a estratégia de fragmentação para um dataset, que possibilita o armazenamento de dados em diversas formas (figura 5), enquanto outras são transitórias.

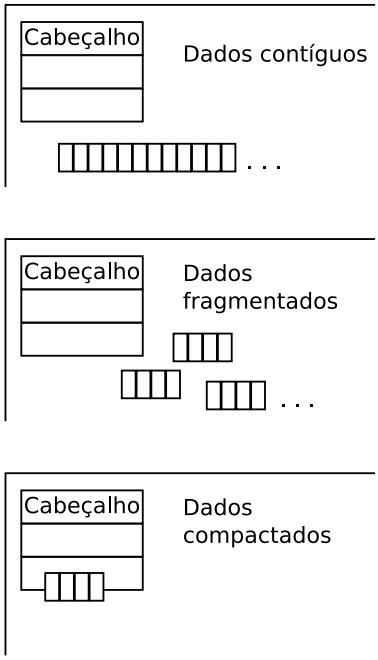


Figura 5.3: Propriedades do HDF5

Um uso comum para a lista de propriedades é a passagem de parâmetros para a camada virtual de arquivos. As listas de propriedades são conceitualmente similares a atributos. As listas de propriedades contém informações relevantes ao comportamento da biblioteca, enquanto atributos são relevantes aos dados e a aplicação do usuário.

ANEXO 5 - Foto do Ambiente

Figura 5.4: Ambiente dos Testes

ANEXO 6 - Tempos de Recuperação das Imagens

Tabela 5.3: *Tempos de Recuperação das Imagens*

	HDF5	Bancos de Dados
1	0,468	0,376
2	0,477	0,371
3	0,466	0,367
4	0,472	0,372
5	0,466	0,367
6	0,468	0,378
7	0,475	0,371
8	0,469	0,371
9	0,479	0,378
10	0,466	0,370
11	0,468	0,369
12	0,469	0,373
13	0,466	0,368
14	0,471	0,372
15	0,466	0,368
16	0,469	0,372
17	0,488	0,371
18	0,465	0,368
19	0,474	0,377
20	0,465	0,371