

Marcone Schardosim Magnus

*Alternativa de Armazenamento de Imagens
Médicas Com Alto Desempenho*

Florianópolis

Maio 2010

Marcone Schardosim Magnus

*Alternativa de Armazenamento de Imagens
Médicas Com Alto Desempenho*

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

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

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

Florianópolis

Maio 2010

Monografia de graduação sob o título “*Alternativa de Armazenamento de Imagens Médicas Com Alto Desempenho*”, defendida por Marccone Schardosim Magnus e aprovada em 8 de julho de 2010, em Florianópolis, Santa Catarina, pela banca examinadora constituída pelos professores:

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

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

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

Sumário

Resumo

Abstract

1	Introdução	p. 7
1.1	Justificativa	p. 7
1.2	Cenário de Aplicação	p. 8
1.3	Objetivos	p. 10
1.3.1	Objetivo Geral	p. 10
1.3.2	Objetivos Específicos	p. 10
1.4	Estrutura do Trabalho	p. 11
2	Revisão Bibliográfica	p. 12
2.1	Rede Catarinense de Telemedicina - RCTM	p. 12
2.2	DICOM	p. 13
2.3	DCMServer	p. 15
2.4	HDF5	p. 16
2.4.1	HDF5 e DCMServer	p. 17
2.5	NetCDF	p. 18
2.5.1	Trabalhos Relacionados	p. 21
2.5.1.1	NASA- HALOE	p. 21
2.5.1.2	Lamont-Doherty Earth Observatory	p. 21
2.5.1.3	CHAMMP - <i>Climate Change Prediction Program</i>	p. 22

2.5.1.4	CSIRO <i>Commonwealth Scientific and Industrial Research Organisation</i>	p. 22
3	Metodologia	p. 23
3.1	Análise do cenário	p. 23
3.2	A Proposta	p. 26
3.3	O Modelo Proposto	p. 27
3.3.1	Modelo Hierárquico	p. 28
4	Ambiente e Resultados Experimentais	p. 31
4.1	Ambiente	p. 31
4.2	Resultados preliminares	p. 31
4.3	Comportamento	p. 32
5	Conclusões e Trabalhos Futuros	p. 34
5.1	Trabalhos Futuros	p. 34
	Referências	p. 36
	Anexo A – Metadados parcial de um arquivo DICOM no modelo NetCDF	p. 38
	Anexo B – E-mail enviado por desenvolvedor da UCAR Unidata Program	p. 43

Resumo

Os sistemas de Telemedicina vem aumentando sua necessidade de performance e segurança no armazenamento das informações médicas. Para suprir essas necessidade vários trabalhos vem sendo desenvolvidos na área de computação de alto desempenho. Este trabalho apresenta uma proposta de modelo de armazenamento de dados científicos aplicados para dados médicos, usando como sistema base a Rede Catarinense de Telemedicina, desenvolvida pelo grupo Cyclops, na Universidade Federal de Santa Catarina em parceria com o Governo do Estado de Santa Catarina. A ferramenta NetCDF é apresentada como uma proposta para melhora do sistema de armazenamento das imagens médicas no formato DICOM. Após a apresentação do modelo apresentados os testes de desempenho comparados com a ferramenta HDF5 já implementada anteriormente no mesmo sistema pelo Grupo Cyclops. Por fim é avaliado a viabilidade da implantação da ferramenta no sistema atual.

Palavras-chave: DICOM, NetCDF, PACS, HDF5, Computação de alto desempenho.

Abstract

Telemedicine systems has increased their need for performance and safe storage of medical information as is become more popular. To meet these needs several researches have been done in the high performance computing area. This paper presents a proposal model applied scientific data storage for medical data using as a base system the Santa Catarina Telemedicine Network developed by Cyclops Group at the Federal University of Santa Catarina in partnership with State Government of Santa Catarina. The NetCDF tool will be presented here as a proposal to improve the system for storing medical images in DICOM format. After the model's presentation will be presented performance tests compared with the HDF5 tool already implemented in the same system by The Cyclops Group. Finally will be assessed the feasibility of implementing the tool in the current system.

Keywords: DICOM, NetCDF, PACS, HDF5, High Performance Computing.

1 *Introdução*

A tecnologia da informação vem se afirmando como necessidade básica da para evolução da sociedade contemporânea, e conforme sua importância vai se consolidando novos desafios vem surgindo para fazer com que a tecnologia atual atenda de forma satisfatória a demanda de serviço. Justamente para suprir os problemas de recursos computacionais exigidos por essa demanda que surge a computação de alto desempenho. A computação distribuída é um ramo da ciência da computação que tem como objetivo a melhoria do desempenho de aplicações distribuídas e paralelas, utilizando de complexas infraestruturas computacionais(DANTAS, 2005, p. 5).

Entre as evoluções sociais que o desenvolvimento da tecnologia da informação está ajudando a sustentar uma delas será abordada nesse trabalho, a saúde. Motivados pelo cenário crítico no sistema público de saúde, foi desenvolvido no estado brasileiro de Santa Catarina pelo grupo de pesquisa *Cyclops Group* com o apoio da Secretaria Estadual da Saúde, um sistema computacional para melhorar e agilizar o atendimento aos pacientes Rede Catarinense de Telemedicina(RCTM). Com finalidade de levar amparo médico às comunidades mais distantes e descongestionar o sistema concentrado de atendimento existente na capital do estado, agilizar o processo de atendimento aos paciente, isso tudo com poucos recursos financeiros e humanos.

1.1 **Justificativa**

A RCTM funciona centralizando os exames do estado inteiro no Hospital Universitário, onde uma equipe de médicos pode fazer o laudo dos exames a distância. Os municípios catarinenses tem seus equipamentos ligados aos servidores localizados na capital, assim quando um paciente precisa fazer um exame em uma comunidade distante que não possui médicos especializados, ele não precisa se locomover até a capital em busca desse atendimento, pode realizar os exames na própria comunidade e enviá-lo através da rede de telemedicina para os servidores localizados no Hospital Universitário onde ele terá seu

laudo feito por um especialista, agilizando o resultado do laudo e riscos de locomoção entre estados.

Imagine agora todos os municípios catarinenses ligados nessa rede. Hoje são 167 municípios que enviam vários tipos de exames entre eles : Eletrocardiogramas, Tomografia Computadorizada, Ressonância Magnética, Raio X entre outros. Um total de mais de 2000 exames enviados todo mês, o que pode ultrapassar 100GB de dados enviados para os servidores da rede catarinense todo mês. Esse grande número de dados gera um constante desafio, armazená-los em segurança e confiabilidade, garantindo o sigilo de cada exame e garantindo que ele estará disponível para o paciente em qualquer lugar e quando ele precisar.

Esse fluxo de dados vem aumentando conforme a rede de telemedicina vem se consolidando. A previsão é que 1,5 TB de dados sejam enviados aos servidores da RCTM no mês de dezembro de 2010. É essencial para a expansão do projeto medidas preventivas que garantam o armazenamento desse volume de dados. Esse trabalho é uma das vertentes de estudo para otimização desse serviço que vem sendo realizados no *Cyclops Group* em conjunto com o LAPESD(Laboratório de Pesquisa em Sistemas Distribuídos) ambos da Universidade Federal de Santa Catarina.

1.2 Cenário de Aplicação

Uma das formas estudadas para suportar essa crescente taxa de armazenamento de dados é através de recursos de sistemas de arquivos distribuídos, Eles tem sido usados como camada básica para sistemas e aplicações distribuídas pois permitem que vários processos compartilhem dados por longos períodos, de modo seguro e confiável(TANENBAUM; STEEN, 2002, p. 296).

No início da década de 80 o Colégio Americano de Radiologia (ACR - *American College of Radiology*) se reuniu com a Associação de Fabricantes de Equipamentos Elétricos dos Estados Unidos (NEMA – *National Electrical Manufactures Association*) para criar um padrão único para intercomunicação entre equipamentos de diferentes fornecedores. Em 1985 surge a primeira versão desse padrão único, após diversas melhorias e várias versões desse padrão surge, em 1992, o DICOM 3(*Digital Imaging and Communications in Medicine*). Atualmente DICOM é o padrão adotado para os Sistemas de Arquivamento e Comunicação de Imagens (PACS – *Picture Archiving and Communications System*)(MILDENBERGER; EICHELBERG; MARTIN, 2002), falaremos mais sobre esse

padrão no capítulo 2.2.

O *Cyclops Group* desenvolveu seu próprio sistema PACS o DCMServer. Essa ferramenta recebe as imagens no padrão DICOM e armazena todas suas informações em um banco de dados relacional, atualmente é usado o Postgres. Dessa forma os exames podem ser acessados pelos médicos através do Portal de Telemedicina, outra ferramenta desenvolvida pelo grupo, onde o médico poderá diagnosticar o paciente pelos seus exames disponíveis no portal. Por sua vez o paciente poderá receber o laudo médico de seus exames sem sair de sua cidade, basta acessar o portal e receber o resultado de seus exames. Mais detalhes sobre o DCMServer serão vistos no capítulo 2.3.

Um sistema de telemedicina hoje, precisa atender certos requisitos fundamentais para fornecer um serviço aceitável para sociedade, como grande espaço para armazenamento e largura de banda para aceitar grandes volumes de trocas de dados. A área de telemedicina não é a única a ter esses requisitos, eles são clássicos nos estudos de simulação física, sistemas meteorológicos, astronômicos, entre outras áreas que geram grande quantidade de dados que devem ser armazenados de maneira a facilitar seu estudo. Essas áreas geralmente fazem uso de um sistema sofisticado de armazenamento (SHASHARINA et al., 2007), por que não aplicar métodos semelhantes nos sistemas de armazenamento de dados médicos atuais? Baseado nessas semelhanças e na necessidade de uma melhoria do sistema de armazenamento na área da telemedicina, surgiram diversos estudos fazendo uso de modelos de dados sofisticados, que foram originados para armazenar e indexar dados científicos, para armazenar e indexar dados médicos.

Um exemplo de formato de dados muito comum nesses armazenamentos de dados científicos é o *Hierarchical Data Format*(HDF). Esse formato é muito comum entre pesquisas que realizam simulações físicas, estudam ciências geográficas, astronomia e inclusive na área médica. Uma das vantagens de se utilizar o formato HDF é que ele previne incompatibilidades binárias já que seus dados são escritos em binário universal e evitam conflitos entre diferentes arquiteturas, além de permitir acesso de leituras e escrita paralelos. Esse modelo de formato de dados foi implementado e integrado ao DCMServer(MACEDO et al., 2008).

Foi realizado no grupo Cyclops um experimento usando o formato HDF para armazenar as imagens médicas. O DCMServer foi estendido para suportar o formato de dados HDF, agora a imagem no padrão DICOM não seria mais armazenada em banco de dados mas sim em um formato hierárquico auto descritivo o modelo proposto será melhor explicado no capítulo 3.3. O objetivo principal era provar que um modelo mais sofisticado de

armazenamento teria desempenho superior ao modelo convencional de base de dados relacional, e associado a uma arquitetura de computação distribuída de alto desempenho essa performance poderia alcançar resultados ainda mais promissores(MACEDO et al., 2008).

Outra ferramenta clássica muito utilizada para armazenamento de dados científicos é o NetCDF. De fato esse será a principal abordagem deste trabalho. NetCDF (*network Common Data Form*) é um conjunto de bibliotecas de software e formato de dados independente de máquina que suporta a criação, acesso e compartilhamento de dados científicos array-orientados(UNIDATA, 2010) e, dependendo da versão, também pode adotar o modelo hierárquico de dados como HDF. O projeto NetCDF é mantido pela *University Corporation for Atmospheric Research*(UCAR), e foi baseado em um modelo conceitual da NASA o CDF (NASA, 2010a). Mais detalhes sobre o modelo NetCDF será discutido na sessão 2.5 deste trabalho.

1.3 Objetivos

1.3.1 Objetivo Geral

Achar um meio alternativo de alto desempenho ao modelo de banco de dados relacional em funcionamento hoje no DCMServer;

1.3.2 Objetivos Específicos

- Elaborar um modelo de dados sofisticado para armazenar dados de imagens médicas no padrão DICOM utilizando NetCDF;
- Integrar o DCMServer com o modelo de dados proposto;
- Armazenar dados médicos que estão em formato dicom usando modelos utilizados para armazenar dados científicos;
- Comparar de desempenho entre NetCDF e HDF5 para armazenamento de imagens médicas no formato DICOM;
- Avaliar viabilidade de uso do modelo proposto .

1.4 Estrutura do Trabalho

Este trabalho será dividido em 5 principais etapas. Na segunda apresentaremos os conceitos envolvidos bem como trabalhos anteriores relacionados com este. Na terceira parte faremos uma descrição mais detalhada sobre o modelo implementado, suas características e objetivos. No quarto capítulo são apresentados os ambientes e resultados experimentais, comparando com os modelos já existentes, modelo relacional e hierárquico com HDF, finalizando, é efetuada a avaliação dos resultados junto com propostas e expectativas para trabalhos futuros.

2 Revisão Bibliográfica

2.1 Rede Catarinense de Telemedicina - RCTM

Entende-se por Telemedicina qualquer meio que auxilie o diagnóstico ou o tratamento médicos realizados a distância. E devido ao desenvolvimento e popularização dos sistemas computacionais modernos os sistemas de telemedicina vem se consolidando como sistema seguro prático e barato para melhoria dos sistemas de saúde.

A rede catarinense de Telemedicina consiste num conjunto de aplicações computacionais que servem para auxílio do atendimento médico. Ela tem como principal objetivo dar atendimento médico especializado para pessoas de todos os municípios do estado de Santa Catarina, da capital aos municípios rurais mais isolados, sem a necessidade de locomoção dos pacientes para os grandes centros com hospitais e clínicas especializadas, evitando assim gastos e riscos do transporte e agilizando o processo de atendimento ao paciente. O processo consiste em realizar o exame no município onde o paciente reside e enviar o exame por uma ferramenta conectada a internet para um centralizador onde será feita uma avaliação especializada e o paciente terá o diagnóstico feito a distância.

Entre as ferramentas que fazem parte da RCTM estão o portal de telemedicina, Dicomizer e DCMServer. O portal de telemedicina é a ferramenta centralizadora, onde médicos e pacientes terão acesso a exames e laudos, DCMServer é uma ferramenta PACS desenvolvida pelo grupo Cyclops para enviar exames médicos para a base de dados DICOM e do portal de Telemedicina, e o Dicomizer é um visualizador Dicom também desenvolvido no grupo Cyclops q auxilia na visualização e manipulação dos exames pelo médico para um diagnóstico mais preciso.

Atualmente 167 municípios de Santa Catarina são cobertos pelo serviço de Telemedicina. Muitos exames entre eles Tomografias Computadorizadas, Ressonâncias Magnéticas e Hemodinâmicas são enviados diariamente para os servidores, um total de mais de 100 GB de dados todo mês.

A expectativa é que até o final do ano de 2010 todos os municípios catarinenses estarão conectados a RCTM. Isso fará um movimento mensal de mais de 20000 exames, mais de 1,5 TB e pacientes de todo estado atendidos pelo sistema. Esse crescente fluxo e quantia de dados para armazenamento foi uma das motivações para o estudo desse trabalho.

2.2 DICOM

Em meados da década de 70 cada fabricante de equipamentos para exames médico criava seu próprio padrão de comunicação e armazenamento de dados. Conforme esse tipo de exame foi se popularizando e mais empresas fabricavam esses equipamentos, a necessidade de um padrão único foi aumentando. Foi justamente por isso que no início dos anos 80 a universidade Americana de Radiologia (*American College of Radiology-ACR*) em conjunto com a associação nacional de fabricantes eletrônicos *National Electrical Manufacturers Association-NEMA* se juntaram para criar um padrão único (MILDENBERGER; EICHELBERG; MARTIN, 2002).

O primeiro padrão foi publicado em 1985 e se chamava ACR-NEMA, foi o primeiro padrão não proprietário para armazenamento e comunicação de dados médicos. ainda na década de 80 surgiram algumas atualizações e em 1993 o que seria a versão 3.0 do ACR-NEMA foi denominado DICOM (*Digital Communications in Medicine*) esse padrão já era bem mais elaborado e já apresentava um protocolo de comunicação mais moderno. A estrutura de dados era baseada em um modelo que possuía um único identificador para objetos e serviços. Esses objetos incluem imagens, dados do paciente, ou laudos médicos. O DICOM desde então serve de base para a maioria dos sistemas de armazenamento e comunicação de imagens médicas (*Picture Archiving and Communication Systems-PACS*)(MILDENBERGER; EICHELBERG; MARTIN, 2002).

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

2.3 DCMServer

DCMServer faz parte do sistema PACS (*Picture archiving and communications system*) implementado pelo grupo Cyclops. PACS é um sistema integrado de gerenciamento de distribuição e armazenamento de dados de imagens médicas (CAO; HUANG; ZHOU, 2003). Ele é responsável por receber as imagens médicas no padrão DICOM enviados dos aparelhos de exames através da internet e armazená-los.

O PACS surgiu, assim como o DICOM, da tentativa de substituir o convencional papel por alternativas da tecnologia da informação. Apesar de um meio convencional tão difundido não se pode dar as costas para as grandes vantagens da nova era digital como agilidade, segurança, praticidade, durabilidade, redução de custos entre tantos outros benefícios.

O DCMServer foi implementado para atender à RCTM. Além de armazenar os exames que são enviados todos os dias, ele os deixa disponível para visualização e manipulação na base de dados do Portal de Telemedicina, outra ferramenta que integra a Rede Catarinense de Telemedicina.

Hoje a maioria dos aparelhos médicos transmitem as imagens já no padrão DICOM diretamente para o DCMServer, porém por se tratar de uma rede pública que existe a alguns anos, ainda existem aparelhos que não estão no formato DICOM, esses equipamentos são ligados a um intermediador chamado *bridge* onde os exames serão passados para o padrão DICOM antes de se comunicar com o DCMServer. Quando um paciente vai a um hospital ou um centro especializado ligados à RCTM ele fará os exames necessários no local, o equipamento médico se comunicará com o servidor localizado hoje no Hospital Universitário, onde ele será armazenado e feito uma cópia para uso através do Portal de Telemedicina, um médico pode então acessar os exames do paciente através do portal diagnosticar e dar laudo através da internet diretamente no portal sem precisar receber os exames diretamente das mãos do paciente ou da entidade que o fez.

Atualmente os exames são armazenados em banco de dados relacional, apesar de existir estudos para armazená-los em sistema de arquivos Unix comum ou formatos de dados mais complexos como desse estudo, esses meios alternativos embora promissores ainda estão em fase de teste.

2.4 HDF5

HDF5 é a versão atual da *Hierarchical Data Format*(HDF), é um formato de dados auto descritivo muito usado para o armazenamento de dados científicos, originados de pesquisas de simulações físicas, na astronomia ou mesmo na medicina. Existem interfaces para o uso do modelo de dados HDF5 em diversas linguagens como C, C++, java, python, entre outras.

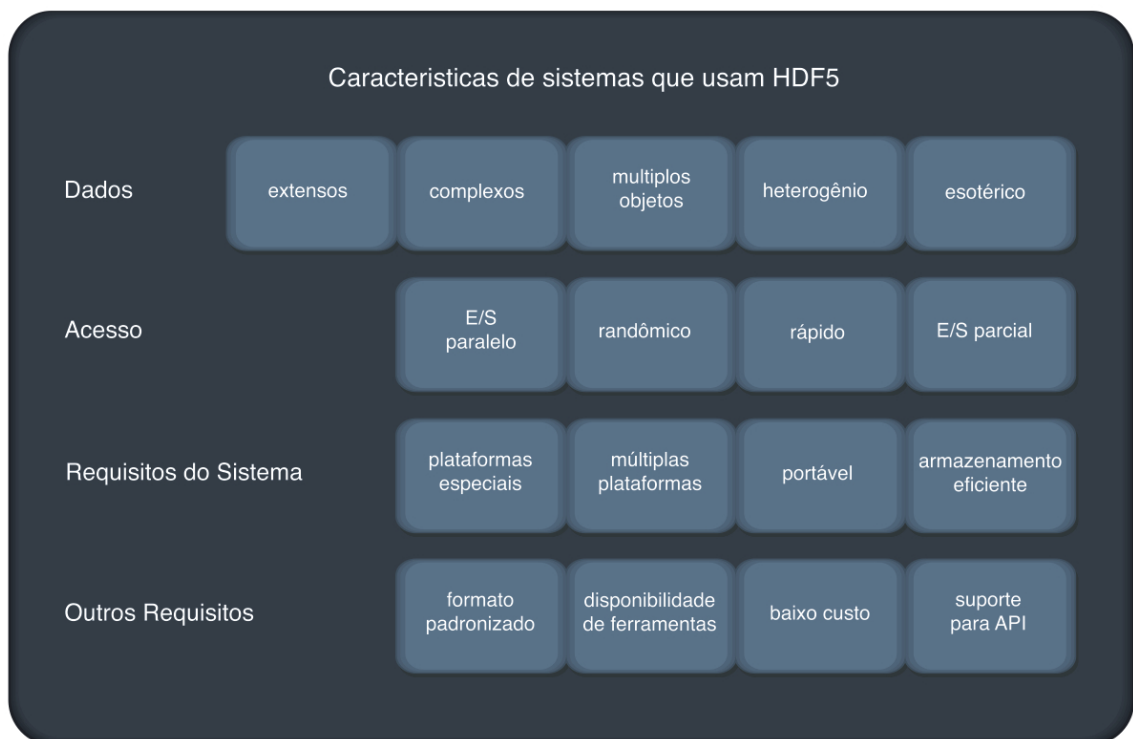


Figura 3: Características de um sistema que usa formato HDF.

Similar ao XML, HDF é auto descritivo e permite o usuário especificar relações entre os dados e dependências. Porém diferente do XML, HDF armazena os dados em formato binário e permite acessar partes específicas dos dados em um arquivo sem precisar carregar todo o conteúdo.

Em contraste com banco de dados relacionais, HDF permite que objetos de dados hierárquicos sejam expressados de maneira natural. Enquanto banco de dados convencional suporta tabelas o formato HDF suporta conjunto de dados multi dimensionais e cada elemento do conjunto de dados pode ser um objeto complexo. Base de dados relacional oferece um excelente suporte para pesquisas baseados na comparação de campo, portem não tem bom desempenho para processamento sequencial (GROUP, 2010).

2.4.1 HDF5 e DCMServer

O grupo cyclops, motivados pelas características e pelo crescimento da RCTM, iniciou uma pesquisa para encontrar meios alternativos para armazenamento dos dados médicos fugindo do sistema de banco de dados relacional usado atualmente. A primeira ferramenta escolhida foi o HDF5.

Então o DCMServer foi alterado e a API do HDF5 foi integrada ao sistema. Agora os dados no padrão DICOM são hierarquizados e salvo em um arquivo no formato HDF5.

Outro grande motivo para essa mudança foi a tentativa de achar soluções de alto desempenho para salvar as imagens médicas. Já que HDF5 foi projetado para suporta operações de E/S paralelas, se mostrou uma ferramenta promissora pra ser usada em um sistema de arquivos distribuídos.

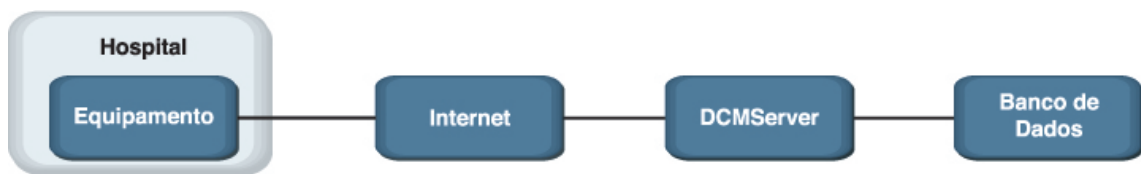


Figura 4: Sistema convencional com banco de dados relacional.

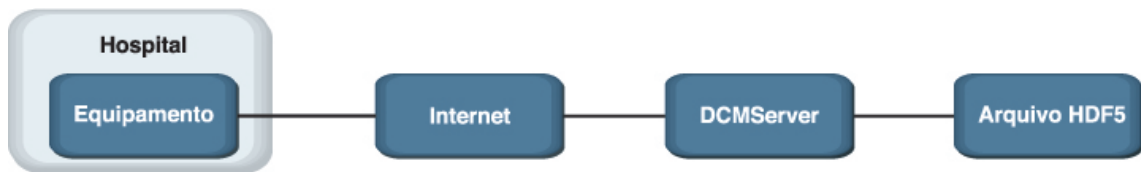


Figura 5: Sistema remodelado com suporte a HDF5.

Os testes preliminares se mostraram muito promissores, embora ainda existia a necessidade de diversas melhorias, o armazenamento de dados foi superior em tempo de gravação comparada com o sistema convencional de banco de dados relacional. Porém a recuperação dos dados foi inferior ao antigo sistema. Mas mesmo assim esses resultados são animadores pois o banco de dados convencional já possui um sistema de indexação consolidado e seria uma surpresa os testes se revelarem superiores logo nos experimentos iniciais, sem muito aprofundamento em algum sistema de indexação mais complexo no uso da ferramenta HDF5 (MACEDO et al., 2008).

2.5 NetCDF

Network Common Data Form(NetCDF) é um conjunto de interfaces para acesso a dados array-orientados e uma coleção livre de bibliotecas de acesso a dados para diversas linguagens como C, Fortran, C++, Java, entre outras. As bibliotecas NetCDF suportam um formato independente de maquina de representação de dados científicos. Juntos, as interfaces e bibliotecas suportam a criação, acesso e compartilhamento de dados científicos (UNIDATA, 2010).

NetCDF foi desenvolvida e é mantida pelo programa Unidata da UCAR(*University Corporation for Atmospheric Research*). Assim como HDF5 NetCDF é muito popular entre aplicações de simulações físicas e aplicações astronômicas e climáticas. E NetCDF é faz parte do estudo deste trabalho, um outro formato para armazenamento de dados científicos para comparar o desempenho com o, já em andamento, projeto implementado em HDF5.

NetCDF é uma ferramenta mais antiga que HDF, de fato quando HDF surgiu um dos objetivos era substituir o NetCDF em muitas aplicações. Por causa disso muitos trabalhos comparativos entre essas duas ferramentas são realizadas. muitas inovações que não existiam ou não funcionavam bem no NetCDF foram implementadas no HDF. Tanto que a última versão do NetCDF (NetCDF4) a principal mudança foi a integração com HDF5. Assim NetCDF4 usa muitos recursos novos do HDF sem perder compatibilidade com as versões mais antigas.

NetCDF a princípio não usava o formato hierárquico como HDF ou XML. Na nova versão NetCDF4, ela importa essa funcionalidade do HDF, podendo agora organizar os dados de forma mais natural. Esse não foi a única integração feita pela ferramenta, mas essa característica foi fundamental para escolhermos o formato de dados que implementamos com NetCDF (veremos o modelo implementado com mais detalhes no capítulo 3.3.1).

Embora o modelo do netCDF-3 tinha a virtude da simplicidade, ele também tinha limitações significantes. Existe pouco suporte para outras estruturas de dados além dos arrays multidimensionais. Apenas uma dimensão por arquivo poderia ser ilimitada o que significava que muitos conjuntos de dados precisam ser representados em múltiplos arquivos. Array de caracteres podem representar strings, porém precisam que o usuário trabalhe com seu tamanho explicitamente. Faltam tipos não sinalizados e tipos inteiros de 64 bits, que são essenciais para algumas aplicações.

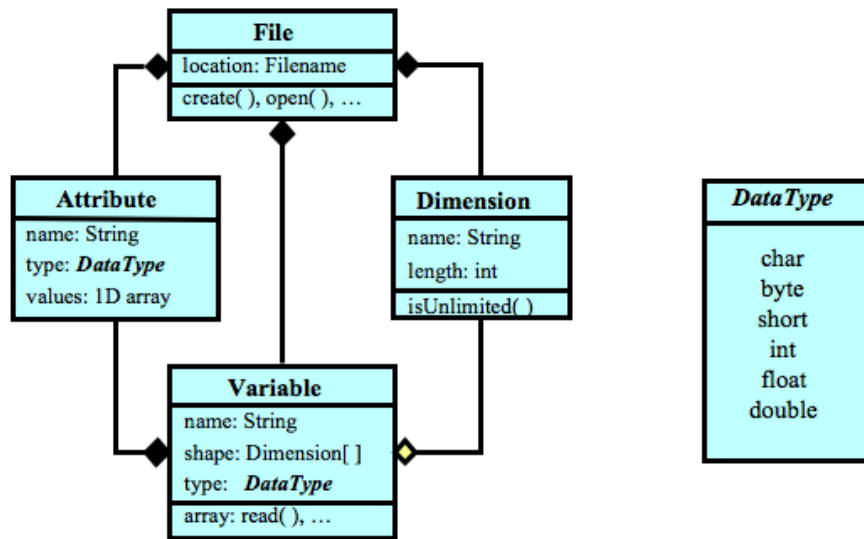


Figura 6: Diagrama UML para o modelo clássico do NetCDF.

O modelo de dados NetCDF-4, implementado usando uma camada de armazenamento baseado no HDF5, trabalha em todas essas limitações. nesse modelo de dados reforçado, um arquivo tem um grupo inicial sem nome. Cada grupo pode conter um ou mais variáveis nomeadas, dimensões, atributos, grupos e tipos. Uma variável ainda é um array multidimensional cujos elementos são todos do mesmo tipo, cada variável pode possuir atributos, e cada formato de variável é especificada por sua dimensão, que pode ser compartilhada. Entretanto, nesse modelo reforçado, uma ou mais dimensões devem ser de tamanho ilimitado, então os dados podem ser adicionados eficientemente às variáveis junto com qualquer uma dessas dimensões(UNIDATA, 2010).

Mesmo com essa adaptação do NetCDF com o HDF os conceitos do modelo não alteraram. os principais são:

- *Variável*: Variáveis são usadas para armazenar a estrutura dos dados em um arquivo NetCDF. Uma variável representa um array de valores do mesmo tipo. Um valor escalar é tratado como um array 0-dimensional. Uma variável tem um nome, um tipo de dados e um formato descrito pela lista de dimensões especificadas quando a variável é criada. A variável pode também ser associada a atributos, que podem ser adicionados, excluídos e modificados depois que a variável é criada (UNIDATA, 2010).
- *Dimensão*: A Dimensão pode ser usada para representar um dimensão física real, por exemplo, tempo, latitude, longitude ou peso. Uma dimensão NetCDF tem

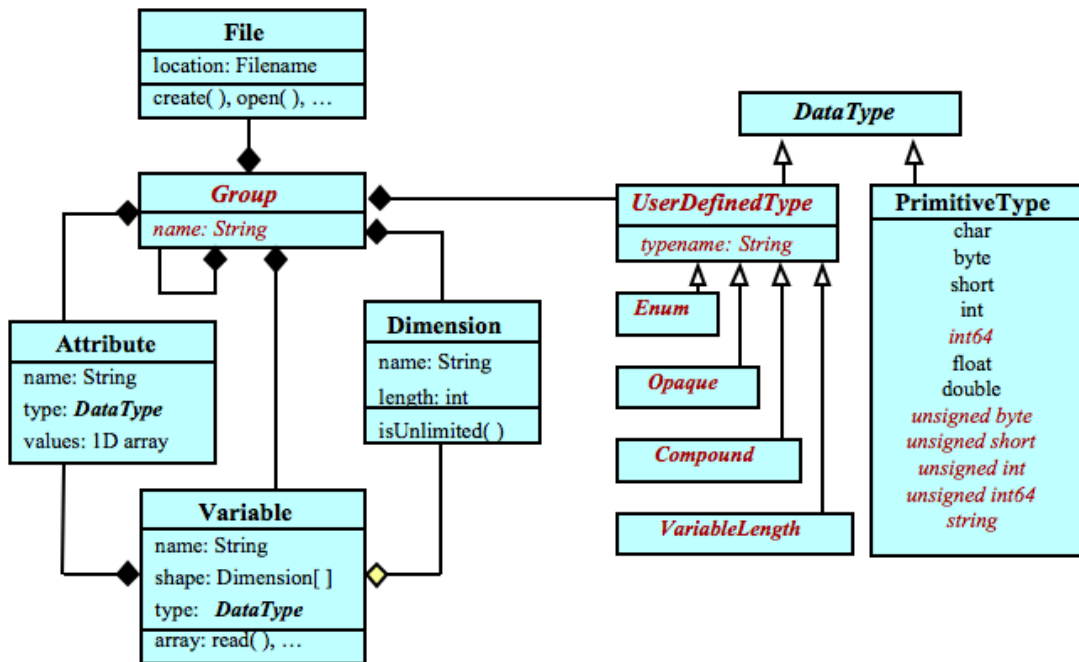


Figura 7: Diagrama UML para o modelo NetCDF4 integrado com HDF5.

nome e tamanho. Um tamanho de dimensão é um inteiro positivo arbitrário, exceto quando uma dimensão em um arquivo NetCDF pode ter um tamanho ilimitado (UNLIMITED). Tal dimensão é chamada de dimensão ilimitada. Uma variável com uma dimensão ilimitada pode atingir qualquer tamanho naquela dimensão (UNIDATA, 2010).

- *Atributos*: São usados para gravar dados sobre os dados (auxiliar dados ou metadados), similar em várias formas as informações nos dicionários de dados e esquemas nos sistemas de base de dados convencional. A maioria dos atributos fornecem informação sobre uma variável específica. Eles são identificados pelo nome (ou ID) dessa variável, junto com o nome do atributo (UNIDATA, 2010).
- *Grupo*: Esse conceito é exclusivo da versão NetCDF-4. Grupos, como diretórios no sistema de arquivos Unix, são organizados hierarquicamente, com profundidade arbitrária. Eles podem ser usados para organizar grande número de variáveis. Cada grupo atua como um conjunto de dados inteiro do modelo clássico do netCDF. Ou seja, em cada grupo pode conter atributos, dimensões e variáveis, como em qualquer outro grupo (UNIDATA, 2010).

2.5.1 Trabalhos Relacionados

NetCDF é um modelo muito popular e bem consolidado a muitos anos. Existem diversos projetos em diversas áreas científicas que possuem seus sistemas baseados ou com suporte a NetCDF. Nesse capítulo conheceremos alguns projetos empresas e entidades que usam ou desenvolvem sistemas baseados nessa ferramenta.

2.5.1.1 NASA- HALOE

O HALOE (*Halogen Occultation Experiment*) foi lançado no satélite de pesquisas atmosféricas (SRVAS) em 12 de setembro de 1991, e após um período de ajustes, começou as observações científicas em 11 de outubro de 1991. O experimento utiliza ocultação solar para medir os perfis verticais de O₃, HCl, HF, CH₄, H₂O, NO, NO₂, e a temperatura versus pressão com um campo de vista vertical instantânea de 1,6 quilômetros na Terra. A cobertura latitudinal é de 80 graus S e 80 graus N, ao longo de um ano e inclui extensas observações da região da Antártica durante a primavera (NASA, 2010b).

Os principais objetivos desse projeto são:

- Estudo da dinâmica de diversas regiões atmosféricas principalmente a polar;
- Estudo sobre a importância da relação entre fontes naturais e antrópicas de cloro;
- Medição de gases atmosféricos e aerossol;
- Análise detalhada do desenvolvimento e recuperação do buraco na camada de ozônio na Antártida.

Os dados dos experimentos do HALOE são organizados no formato NetCDF. O *HALOE Data Viewer* disponibiliza cada tipo de dado com uma interface guiada por menu para auxiliar na localização dos arquivos baseados na data, tempo, espécie e versão dos dados (UNIDATA, 2010).

2.5.1.2 Lamont-Doherty Earth Observatory

O LDEO é um laboratório de pesquisas que observa os comportamentos climáticos e geológicos da Terra. O observatório estuda terremotos, vulcões, mudanças climáticas afim de entender e prever o comportamento da Terra e como isso afeta e afetará a sociedade.

Os cientistas da LDEO (*Lamont-Doherty Earth Observatory*) da Universidade de Columbia observam a Terra numa escala global, do seu mais profundo interior até as camadas exteriores da Atmosfera, em cada continente e em cada oceano. Eles decifram a longa história do passado, monitoram o presente e procuram prever o futuro da Terra.

O instituto de pesquisa da Universidade de Columbia converteu todos seus dados geofísicos coletados nos últimos 40 anos por cientistas da LDEO para o formato de dados NetCDF (LDEO, 2010).

2.5.1.3 CHAMMP - *Climate Change Prediction Program*

CHAMMP é um programa do Departamento de Energia que faz pesquisas de previsão climática. Um componente importante do programa CHAMMP é que ele une as tecnologias emergentes em Computação de Alto Desempenho para o desenvolvimento de modelos computacionalmente eficientes de previsão numérica precisa do clima. Patrocinado pelo Instituto de Investigação Biológica e Ambiental (OBER), o programa envolve um esforço para o desenvolvimento computacional de métodos com capacidade de simulação da atmosfera futura e modelo geral de circulação dos oceanos. Estes programas de computador constituem o núcleo do avançado modelos de previsão que podem ser usadas para estudar as mudanças climáticas. E o programa selecionou o NetCDF como formato de armazenamento para seus dados computacionais(CHAMMP, 2010).

2.5.1.4 CSIRO *Commonwealth Scientific and Industrial Research Organisation*

CSIRO é uma agência científica nacional da Austrália e uma das maiores e mais diversificadas do mundo. Entre suas áreas de pesquisa destacam-se a de adaptação climática, saúde preventiva, energia transformada, agricultura sustentável, riqueza dos oceanos entre outros. A divisão de pesquisas atmosféricas e os dados originados de modelos oceânicos usam NetCDF para armazenamento(CSIRO, 2010).

3 Metodologia

Devido ao crescimento acelerado dos sistemas convencionais de telemedicina nos últimos anos e uma expectativa de um aumento ainda maior para os próximos, o grupo cyclops está pesquisando novos meios para armazenar os dados médico com mais confiabilidade e eficiência. Já que os requisitos do sistema atual tem grande semelhança com trabalhos científicos que usam meios mais sofisticados de armazenamento essa é uma das vertentes atuais no qual se encaixa esse trabalho. Adequar as ferramentas usadas atualmente na RCTM à alguns desses modelos sofisticados de dados e testar recursos.

Se hoje já é enfrentado problemas de armazenamento dos dados na RCTM, e existe uma grande demanda por largura de banda para que as ferramentas de auxílio médico funcionem com um desempenho aceitável, podemos imaginar os problemas que nos reservam os próximos anos, quando a RCTM se tornar mais popular e atingirá todos os municípios catarinenses ou ainda se pensarmos em uma rede no cenário nacional.

E é justamente essa a expectativas para os sistemas de Telemedicina atuais, que eles se tornem cada vez mais populares e aumente ainda mais o fluxo de dados e consequentemente a quantidade de dados armazenados(BASHSHUR, 2002).

3.1 Análise do cenário

No Cenário atual temos um sistema funcionando com uma arquitetura cliente servidor. Tem como principal características o número ilimitado de clientes e a centralização das informações em um único servidor.

Seu funcionamento é simples, cada cliente localizado em algum hospital ou clínicas especializadas se comunica com um servidor localizados hoje no Hospital Universitário, esse servidor é responsável pelo armazenamento tanto para o portal de Telemedicina quanto para a base DICOM que é o que nos interessa já que nosso objetivo deste trabalho ainda não é substituir a base de dados da aplicação do portal de Telemedicina para um

formato mais complexo, os estudos estão se concentrando na base DICOM.

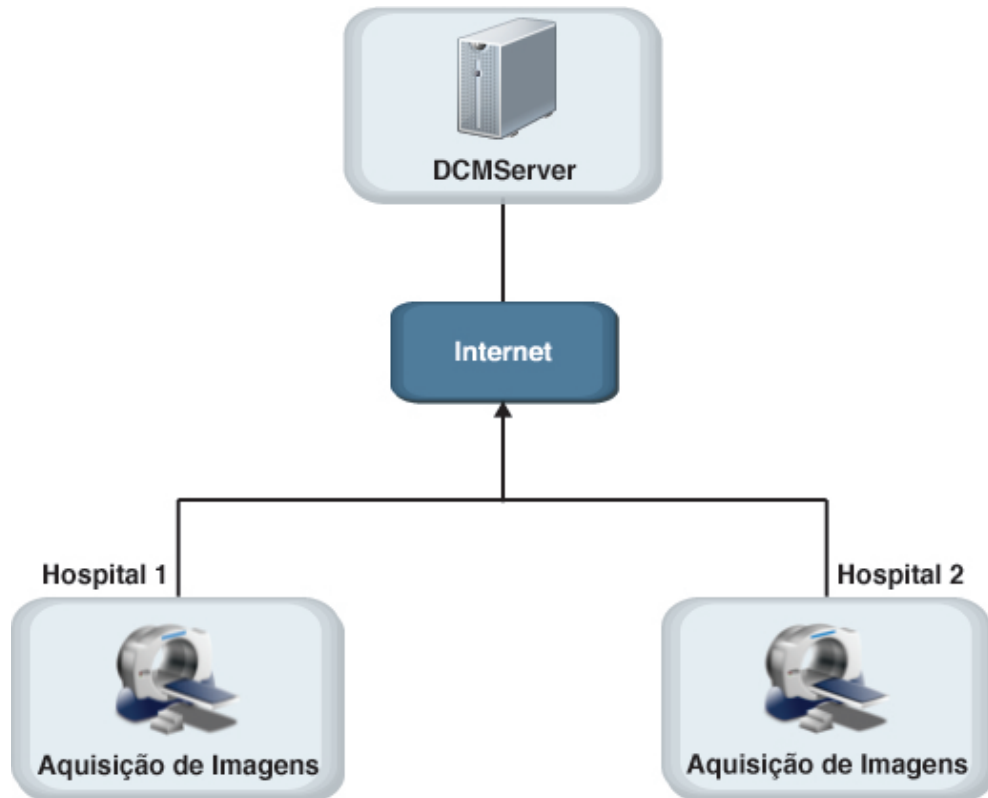


Figura 8: Arquitetura cliente servidor.

O servidor (DCMServer) desconstrói os arquivos no formato DICOM que ele recebe dos clientes. Ele passa todos os IODs, as informações relacionadas à imagem e os *headers* para um banco de dados relacional.

O formato DICOM possui uma complexa estrutura de dados entre os principais elementos dessa estrutura estão:

- **Data Set:** O conjunto de dados (Data Set) é composto de diversos elementos de dados (*Data Element*), cada elemento de dado é composto de três ou quatro campos, etiqueta (tag), representação de valor (VR), tamanho do valor, e campo de dados. Os elementos de dados são definidos unicamente por uma etiqueta (Tag). Os elementos de dados devem ser ordenados de acordo com a etiqueta em ordem crescente;
 - **Etiqueta:** A etiqueta é composta de 32 bits (4 bytes), 2 bytes para o grupo e 2 bytes para o elemento, os grupos pares são os que são definidos pelo padrão DICOM e podem ser encontrados na parte 6 do padrão, os grupos ímpares são para uso privado;

Etiqueta		VR		tamanho	Valor
Número do Grupo	Número do Elemento	Caractere de 2 bytes estabelecendo a representação do valor (tipo do dado)	Reservado	tamanho em bytes do campo valor	
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	variável

Tabela 1: Dataset

Grupo	Descrição
0002	Grupo de meta-elementos
0008	Grupo de identificação
0018	Grupo de Aquisição
0028	Grupo de apresentação das imagens

Tabela 2: Exemplo de Grupos

Grupo	Elemento	Tipo	Descrição
0002	0010	Transfer Index UID	Define a sintaxe de transferência
0008	0016	AE Fonte	Título da entidade de aplicação
0018	0016	SOP Class UID	Define a classe de serviço-objeto (SOP)
0028	0060	Modalidade	Modalidade da imagem (MR, CT, XR)
0018	0070	Fabricante	GE, Siemens, Toshiba
0018	0050	Espessura do corte	Perímetro de aquisição da imagem
0018	0010	Pixel Data	Imagem

Tabela 3: Exemplo de Etiquetas

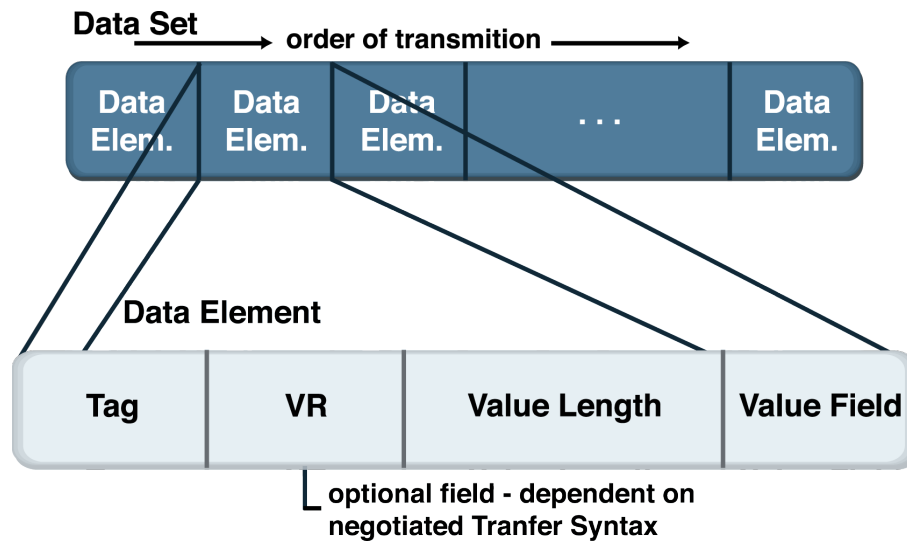


Figura 9: Arquitetura cliente servidor.

- **REPRESENTAÇÃO DE VALOR (VR):** O campo VR (Value Representation) do elemento de dados deve conter 2 bytes que especificam o tipo de dados e a sintaxe de transferência que estará no campo de dados, o VR pode ser explícito ou implícito, no implícito os dados são utilizados na sintaxe de transferência padrão DICOM (implicit VR Little Indian).
- **IOD:** Um IOD é um modelo de dado orientado a objeto utilizado para especificar objetos reais. Com base nos IODs, as entidades de aplicação partilham uma visão comum das informações que serão trocadas. Os IODs estão especificados na parte 3 do padrão DICOM.

3.2 A Proposta

A proposta desse trabalho é armazenar essas estruturas complexas de um arquivo médico no formato DICOM em um modelo mais elaborado do que a banco de dados relacional. Existe hoje uma estudo já realizado com a ferramenta HDF5(MACEDO et al., 2008). Então neste trabalho propomos a integração do DCMServer com a ferramenta NetCDF e uma comparação de desempenho entre esses dois modelos de armazenamento de dados científicos.

Esse modelo de armazenamento de dados científicos é recomendado para sistemas que tenham características específicas como:

- **Grande quantidade de dados:** Um único arquivo no formato DICOM pode

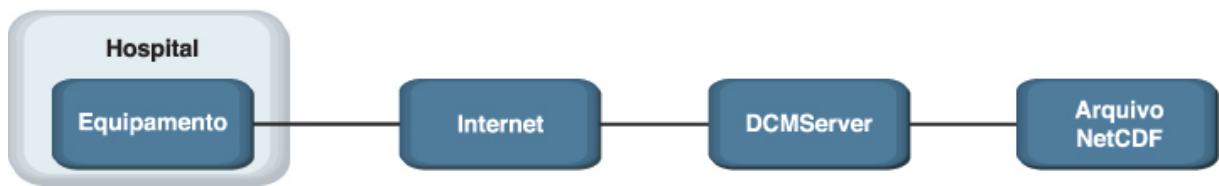


Figura 10: DCMServer com integração com o NetCDF.

conter várias imagens com tamanho significativamente grande, um único exame pode gerar mais de 500 Mb de dados, A previsão para dezembro de 2010 é que a RCTM receba 1,5 TB de dados médicos.

- **Dados complexos:** Como vimos o padrão DICOM possui descrições sobre o exame e sobre o paciente muito detalhadas e a estrutura que é organizado os dados é bem complexa.
- **Dados Heterogêneos:** Atualmente o DCMServer suporta diversos tipos de modalidades de exame, e o padrão DICOM descreve mais de 40 modalidades de exames, cada tipo de exame tem tipos de imagens diferentes podendo conter vídeos, som ou imagens estáticas.
- **Acesso com E/S paralelos:** Entre os planos de desenvolvimento do sistema de armazenamento dos servidores da RCTM estão estratégias de computação de alto desempenho utilizando cluster ou grids para maximizar o desempenho da aplicação (MACEDO, 2008).

3.3 O Modelo Proposto

O principal teste a ser realizado com o modelo é o teste de desempenho comparado com o modelo utilizando HDF5. Então as características entre os modelos foram aproximadas. A versão do NetCDF escolhida foi a versão NetCDF-4, que suporta o modelo padrão do HDF e trabalha com diversos defeitos que as antigas versões do NetCDF tinham.

O modelo será armazenado em um único arquivo e assim como HDF adotará um modelo hierárquico para representação dos dados. Outro motivo para a versão escolhida da ferramenta, uma vez que as versões anteriores do NetCDF tinham limitações correspondentes ao tamanho do arquivo e não possuíam suporte ao modelo hierárquico. Mas a base fundamental de arrais multidimensionais ainda continuava.

A nova versão da ferramenta foi projetada para suportar operações de E/S em paralelo.

Essa é uma propriedade que não foi explorada aqui nesse trabalho, mas é um projeto para ser explorado no futuro conforme é descrito em (MACEDO, 2008).

3.3.1 Modelo Hierárquico

O modelo implementado para armazenar as imagens DICOM pode ser entendido de forma análoga ao sistema de arquivos UNIX, com seu padrão de pastas e subpastas. Definimos uma estrutura de pastas fixas baseadas na estrutura de tabelas da base DICOM atual da RCTM. Cada imagem DICOM inserida vai ser organizada de acordo com essa estrutura pré definida.

Essa estrutura está dividida em 4 níveis. Eles foram escolhidos para funcionar de maneira similar à maneira implementada em (MACEDO et al., 2008), onde esse modelo foi escolhido considerando as informações contidas nas imagens do formato DICOM, o modelo pode ser visualizado na Figura 11. O modelo possibilitou uma organização satisfatória para o experimento realizado, permitindo uma boa visualização, um excelente desempenho e praticidade de acesso aos dados (MACEDO et al., 2008).

O modelo original pode ser visto na Figura 12, o nível que representa Hospital foi representado porque aquele modelo considerava um cenário nacional da aplicação, o que ainda não corresponde a realidade. No modelo deste trabalho o nível do Hospital foi removido considerando o fato de um paciente poder realizar exames em mais de uma instituição de saúde, o que, no modelo da Figura 12, poderia significar um aumento desnecessário de quantidade de grupos criados na hierarquia.

Assim como no sistema de arquivo UNIX o topo da hierarquia começa sempre com o nível “/” ele é o único grupo desse nível.

No segundo nível temos o nome do paciente. Neste modelo ainda abstraímos situações como dois pacientes com o mesmo nome, sabemos que isso é possível e frequente mas não é o foco de estudo deste trabalho resolver esse tipo de problema. O nome do paciente está representado no arquivo DICOM pela *Tag* 0010x0010. O terceiro nível é o estudo, *Tag* 0020x000D. O quarto e último nível é a série, *Tag* 0020x000E do padrão DICOM.

Quando inserimos uma imagem no padrão DICOM, a aplicação verifica se já existem os níveis referente aquela imagem, se já existir os diretórios correspondes são abertos e tem os dados armazenados de forma organizada. Se não existirem os diretórios, caso seja a primeira vez que um paciente tem o seu exame inserido no sistema ou seja um novo estudo ou série, os diretórios respectivos aos novos dataelements são criados.

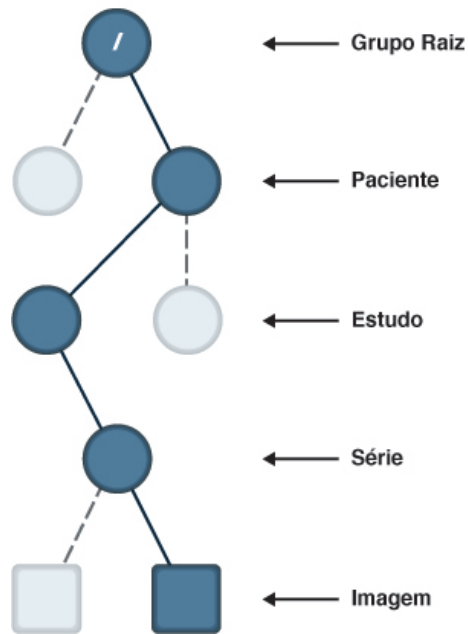


Figura 11: Hierarquização dos dados usando NetCDF.

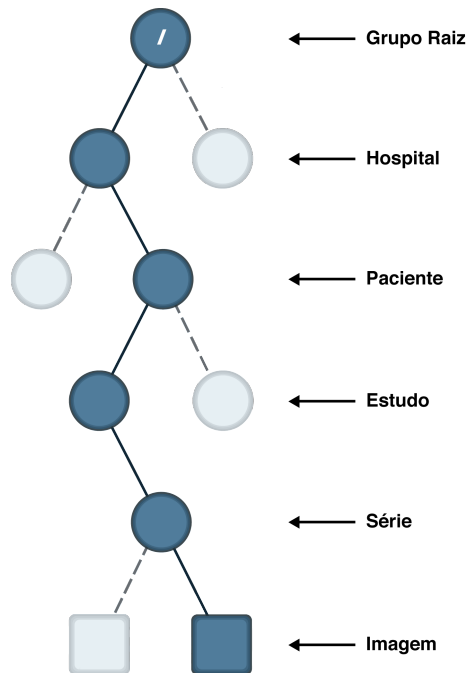


Figura 12: Hierarquização dos dados usando HDF5.

Para cada um dos *DataElements* do arquivo DICOM criamos uma nova variável do NetCDF e uma nova dimensão. Cada variável é relacionada a uma dimensão exclusiva, essa dimensão é do tamanho do *Dataelement* que a variável armazenará. Isso faz com que cada variável do NetCDF seja um array do tamanho exato para cada *Dataelement* que

ele representa. Tanto a variável quanto a dimensão são nomeados com o nome da Tag que elas representam, deixando assim o modelo em uma forma bastante natural para a leitura.

Uma biblioteca nova foi criada para servir de interface de comunicação entre o NetCDF e o DCMserver. A biblioteca é escrita utilizando a interface em C do NetCDF. Essa biblioteca possui as chamadas de abertura, leitura e escrita da API do NetCDF.

O DCMServer possui uma classe lógica referente a um modelo de armazenamento genérico. Então uma nova classe foi implementada especializando essa classe genérica do servidor. A nova classe de armazenamento não faz mais chamadas para o banco de dados ela controla os grupos da hierarquia do arquivo NetCDF e faz chamadas para biblioteca que faz a interface entre o DCMServer e a API do NetCDF.

4 *Ambiente e Resultados Experimentais*

Neste capítulo apresentaremos os testes realizados com o DCMServer integrado com a biblioteca de interfaceamento com o NetCDF. Vamos apresentar os resultados obtidos comparar com os resultados do DCMServer integrado com HDF5.

4.1 Ambiente

Para realizar os testes 100 imagens médicas foram escolhidas, não será realizado, nesta etapa, testes de imagens médicas enviadas diretamente de um equipamento médico, já que o objetivo é avaliar o comportamento do servidor usando a biblioteca NetCDF e não a comunicação do DCMServer com aparelhos médicos. Elas foram enviadas através da ferramenta *storescu* (DCMTK, 2010).

Assim nenhuma falha de comunicação interferiu nos resultados dos teste. O computador é equipado com processador AMD Athlon X2 2.0 com 2 Gb de memória RAM. Este com certeza não será um ambiente onde o DCMServer estaria rodando para prestar serviço para toda a RCTM, mas esse será apenas um teste comparativo, ambos, NetCDF e HDF5 serão usados nessa mesma máquina, e o que nos interessa é o desempenho relativo entre esses dois cenários.

A imagens no formato DICOM foram escolhidas de maneira aleatória, sem intenção de favorecer ou prejudicar o desempenho de nenhuma das aplicações. O dados usados são dados reais de exames médicos, mas os dados dos pacientes permanecerão em sigilo.

4.2 Resultados preliminares

De acordo com os testes iniciais o DCMServer com NetCDF recebeu e armazenou as 100 imagens totalizando 51Mb de dados em 89.22282490 segundos. Esse tempo passou

muito abaixo das expectativas, já que o DCMServer com HDF5 armazenou a mesma quantia de dados em 32.393293 segundos. Analgizando o tempo de gravação de cada imagem no modelo que usa NetCDF nota-se que o tempo levado para armazenar cada imagem vai aumentando conforme mais imagens são armazenadas.

4.3 Comportamento

De acordo com o modelo escolhido, as imagens médicas são armazenadas em um único arquivo NetCDF, dentro desse arquivo elas são hierarquizadas e os dados são organizados e armazenados. Notou-se que conforme esse arquivo aumentava seu tamanho algumas operações levavam mais tempo para terminar a execução.

Podemos observar que a primeira imagem que tem 515 Kb é armazenada em 0.0941359 segundos, já a última imagem armazenada que tem o mesmo tamanho leva 1.84557 segundos, podemos observar que o tempo de armazenamento vai aumentando gradativamente. Isso não ocorre no modelo implementado com HDF5, apesar do primeiro arquivo ser armazenado mais rápido que os demais eles não vão aumentando conforme o arquivo HDF5 aumenta eles tem um comportamento linear.

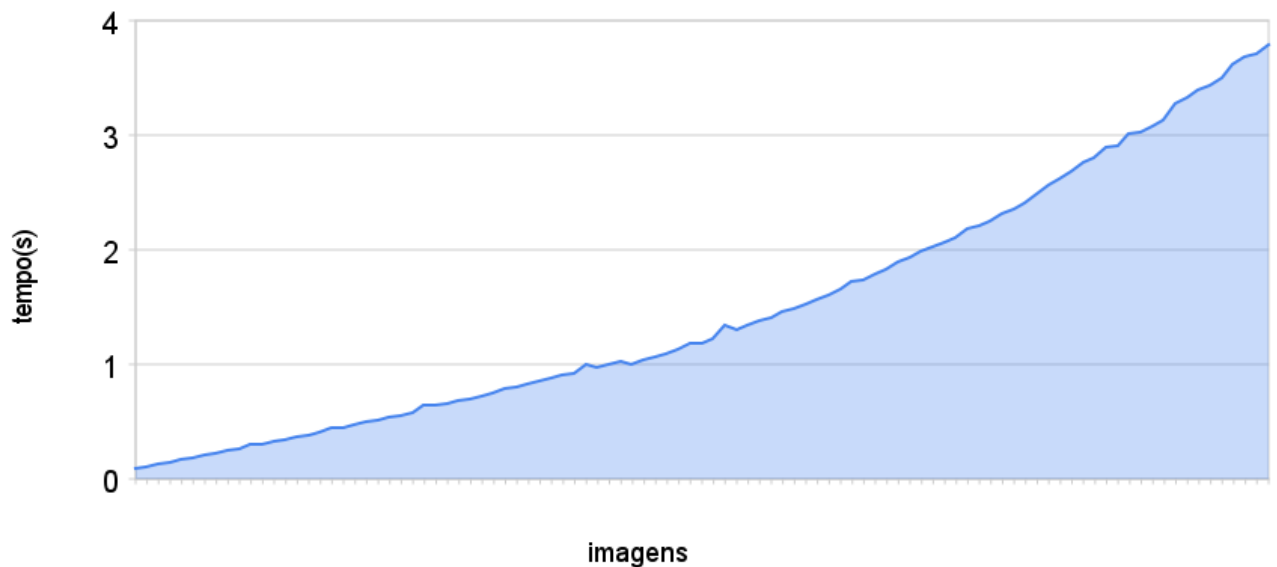


Figura 13: Comportamento do armazenamento de imagem com NetCDF.

Para cada imagem enviada o servidor faz as seguintes operações para armazenar os dados: abre o arquivo netcdf, isso vai fazer o arquivo abrir no diretório “/” da hierarquia, então abre o segundo grupo, insere os dados referentes aquele nível, então abre o terceiro

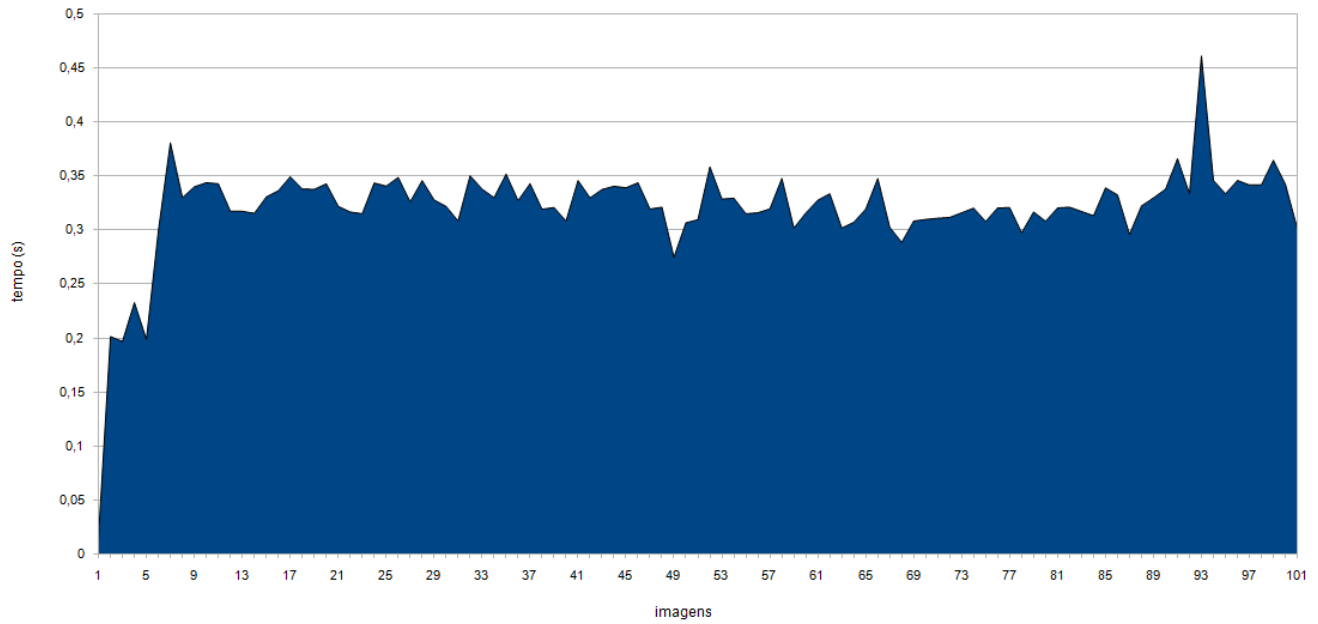


Figura 14: Comportamento do armazenamento de imagem com HDF5.

grupo, faz grava os dados referentes aquele nível, abre o ultimo nível, insere o restante dos dados nesse último grupo e por fim fecha o arquivo.

Analisando cada operação individualmente observou-se que as operações que abrem escrevem e fecham o arquivo aumentam o tempo de execução conforme o arquivo aumenta de tamanho.

5 *Conclusões e Trabalhos Futuros*

Os resultados dos testes apresentaram um comportamento que não era esperado nesse trabalho. O modelo implementado foi um modelo similar ao implementado em (MACEDO et al., 2008) e como o NetCDF usa uma camada da gravação originada do HDF5, era esperado um comportamento similar. Não foi o que ocorreu, uma vez que armazenando uma pequena quantidade de dados o tempo aumentava significativamente para cada arquivo.

O formato padrão do NetCDF tinha um tamanho limitado de arquivo. Usamos esse formato para o padrão ficar parecido com o o trabalho implementado em (MACEDO et al., 2008). Porém verificamos que apesar do NetCDF usar uma camada de armazenamento implementada pelo HDF5 o comportamento não é igual.

O modelo que se mostrou eficiente para HDF5 não teve o mesmo desempenho com NetCDF. Usar o mesmo arquivo para armazenar todas as informações não é uma alternativa viável para o problema enfrentado na RCTM. Isso não significa que a ferramenta não é adequada para trabalhar nesse ambiente. Significa que a ferramenta não é adequada para esse modelo.

Em contato com a equipe de desenvolvimento do NetCDF, como podemos ver no anexo B, fomos informados que, com o objetivo de acelerar o acesso aos dados gravados no arquivo NetCDF, o arquivo é inteiro carregado em memória. Isso não acontece no HDF5 que carrega apenas os dados que solicitamos. Assim confirmamos a inviabilidade dessa ferramenta para esse modelo.

5.1 **Trabalhos Futuros**

Como a ferramenta não teve um comportamento esperado para o modelo, um novo modelo poderá ser proposto para substituir o modelo hierárquico para armazenamento das imagens médicas com NetCDF. Esse novo modelo deverá considerar as limitações da ferramenta detectadas nesse trabalho.

Um modelo usando o formato clássico do NetCDF seria de grande contribuição também, já que, apesar de ser bem distinto do formato hierárquico usado hoje pelo HDF5 na RCTM, existem muitas ferramentas e projetos que utilizam esse modelo.

A continuação desse projeto não se restringe ao uso dessa ferramenta específica. Alguns outros mecanismos em ascensão no cenário mundial para sistemas com grande volume de dados e grande tráfego, como o modelo descrito por Chang em (CHANG et al., 2006), fariam uma grande contribuição os sistemas de Telemedicina.

Referências

- BASHSHUR, R. Telemedicine and health care. *Telemedicine Journal and e-health*, v. 8, n. 1, p. 5–12, 2002.
- CAO, F.; HUANG, H.; ZHOU, X. Medical image security in a HIPAA mandated PACS environment. *Computerized Medical Imaging and Graphics*, Elsevier, v. 27, n. 2-3, p. 185–196, 2003.
- CHAMMP. *Climate Change Prediction Program*. 2010. Acessado em: 28 abr. 2010. Disponível em: <http://www.csm.ornl.gov/chammp/chammp.html>.
- CHANG, F. et al. Bigtable: A distributed storage system for structured data. In: *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*. [S.l.: s.n.], 2006.
- CSIRO. *Commonwealth Scientific and Industrial Research Organisation*. 2010. Acessado em: 28 abr. 2010. Disponível em: http://www.its.csiro.au/csiro/channel/_ca_dch2t-.html.
- DANTAS, M. *Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais*. [S.l.: s.n.], 2005.
- DCMTK, O. *OFFIS DCMTK Documentation*. 2010. Acessado em: 28 abr. 2010. Disponível em: <http://support.dcmthk.org/docs/storescu.html>.
- GROUP, H. *Hierarchical Data Format*. 2010. Acessado em: 28 abr. 2010. Disponível em: <http://www.hdfgroup.org/>.
- LDEO. *Lamont-Doherty Earth Observatory*. 2010. Acessado em: 28 abr. 2010. Disponível em: <http://www.ldeo.columbia.edu/>.
- MACEDO, D. D. J. de. *Um Estudo de Estratégias de Sistemas Distribuídos Aplicadas a Sistemas de Telemedicina*. Dissertao (Mestrado) — Universidade Federal de Santa Catarina, 2008.
- MACEDO, D. de et al. Armazenamento distribuído de imagens médicas DICOM no formato de dados HDF5. In: ACM. *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web*. [S.l.], 2008. p. 20–27.
- MILDENBERGER, P.; EICHELBERG, M.; MARTIN, E. Introduction to the DICOM standard. *European radiology*, v. 12, n. 4, p. 920, 2002.
- NASA. *Common Data Format (CDF)*. 2010. Acessado em: 28 abr. 2010. Disponível em: <http://cdf.gsfc.nasa.gov/>.

NASA. *HALOE*. 2010. Acessado em: 28 abr. 2010. Disponível em: <<http://haloe.gats-inc.com/about/index.php>>.

SHASHARINA, S. et al. Distributed Technologies for Remote Access of HDF Data. *proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2007)*, 2007.

TANENBAUM, A.; STEEN, M. V. *Distributed systems*. [S.l.]: Citeseer, 2002.

UNIDATA. *Network Common Data Format*. 2010. Acessado em: 28 abr. 2010. Disponível em: <<http://www.unidata.ucar.edu/software/netcdf/>>.

ANEXO A – Metadados parcial de um arquivo DICOM no modelo NetCDF

Essa é parte de uma imagem médica após realizado um *dump* usando a ferramenta *ncdump*, que faz parte do conjunto de ferramentas do NetCDF. Aqui não estão incluídos os dados da imagem médica, apenas os metadados, uma vez que os dados são muito extensos.

```
netcdf dicom {
```

```
group: MAGNUS\,MARCONE {
```

```
  dimensions:
```

```
  \0010x0010 = 20 ;
```

```
  variables:
```

```
  ubyte \0010x0010(\0010x0010) ;
```

```
group: \1.2.840.113619.2.22.287.1.220.20060303.282359 {
```

```
  dimensions:
```

```
  \0020x000d = 46 ;
```

```
  variables:
```

```
  ubyte \0020x000d(\0020x000d) ;
```

```
group: \1.2.840.113619.2.22.287.1.220.5.20060303.290116 {
```

```
  dimensions:
```

```
  \0020x000e = 48 ;
```

```
  variables:
```

```
  ubyte \0020x000e(\0020x000e) ;
```

```
group: \1.2.840.113619.2.22.287.1.220.5.3.20060303.290234 {
```

```
  dimensions:
```

\0008x0018 = 50 ;
\0008x0005 = 10 ;
\0008x0008 = 22 ;
\0008x0016 = 26 ;
\0008x0020 = 8 ;
\0008x0021 = 8 ;
\0008x0022 = 8 ;
\0008x0023 = 8 ;
\0008x0030 = 6 ;
\0008x0031 = 6 ;
\0008x0032 = 6 ;
\0008x0033 = 6 ;
\0008x0050 = 1 ;
\0008x0060 = 2 ;
\0008x0070 = 18 ;
\0008x0080 = 28 ;
\0008x0090 = 18 ;
\0008x1010 = 2 ;
\0008x1030 = 4 ;
\0008x103e = 20 ;
\0008x1050 = 12 ;
\0008x1060 = 12 ;
\0008x1090 = 8 ;
\0010x0020 = 10 ;
\0010x0030 = 1 ;
\0010x0040 = 2 ;
\0010x1010 = 4 ;
\0010x1030 = 6 ;
\0018x0010 = 6 ;
\0018x0015 = 8 ;
\0018x0022 = 12 ;
\0018x0050 = 4 ;
\0018x0060 = 4 ;
\0018x0090 = 6 ;
\0018x1000 = 16 ;


```
\0018x1020 = 4 ;
\0018x1030 = 14 ;
\0018x1040 = 2 ;
\0018x1050 = 10 ;
\0018x1100 = 12 ;
\0018x1110 = 12 ;
\0018x1111 = 12 ;
\0018x1120 = 4 ;
\0018x1130 = 4 ;
\0018x1140 = 2 ;
\0018x1150 = 4 ;
\0018x1151 = 4 ;
\0018x1170 = 2 ;
\0018x1190 = 10 ;
\0018x1210 = 4 ;
\0018x5100 = 4 ;
variables :
ubyte \0008x0018(\0008x0018) ;
ubyte \0008x0005(\0008x0005) ;
ubyte \0008x0008(\0008x0008) ;
ubyte \0008x0016(\0008x0016) ;
ubyte \0008x0020(\0008x0020) ;
ubyte \0008x0021(\0008x0021) ;
ubyte \0008x0022(\0008x0022) ;
ubyte \0008x0023(\0008x0023) ;
ubyte \0008x0030(\0008x0030) ;
ubyte \0008x0031(\0008x0031) ;
ubyte \0008x0032(\0008x0032) ;
ubyte \0008x0033(\0008x0033) ;
ubyte \0008x0050(\0008x0050) ;
ubyte \0008x0060(\0008x0060) ;
ubyte \0008x0070(\0008x0070) ;
ubyte \0008x0080(\0008x0080) ;
ubyte \0008x0090(\0008x0090) ;
ubyte \0008x1010(\0008x1010) ;
```

```
ubyte \0008x1030(\0008x1030) ;
ubyte \0008x103e(\0008x103e) ;
ubyte \0008x1050(\0008x1050) ;
ubyte \0008x1060(\0008x1060) ;
ubyte \0008x1090(\0008x1090) ;
ubyte \0010x0020(\0010x0020) ;
ubyte \0010x0030(\0010x0030) ;
ubyte \0010x0040(\0010x0040) ;
ubyte \0010x1010(\0010x1010) ;
ubyte \0010x1030(\0010x1030) ;
ubyte \0018x0010(\0018x0010) ;
ubyte \0018x0015(\0018x0015) ;
ubyte \0018x0022(\0018x0022) ;
ubyte \0018x0050(\0018x0050) ;
ubyte \0018x0060(\0018x0060) ;
ubyte \0018x0090(\0018x0090) ;
ubyte \0018x1000(\0018x1000) ;
ubyte \0018x1020(\0018x1020) ;
ubyte \0018x1030(\0018x1030) ;
ubyte \0018x1040(\0018x1040) ;
ubyte \0018x1050(\0018x1050) ;
ubyte \0018x1100(\0018x1100) ;
ubyte \0018x1110(\0018x1110) ;
ubyte \0018x1111(\0018x1111) ;
ubyte \0018x1120(\0018x1120) ;
ubyte \0018x1130(\0018x1130) ;
ubyte \0018x1140(\0018x1140) ;
ubyte \0018x1150(\0018x1150) ;
ubyte \0018x1151(\0018x1151) ;
ubyte \0018x1170(\0018x1170) ;
ubyte \0018x1190(\0018x1190) ;
ubyte \0018x1210(\0018x1210) ;
ubyte \0018x5100(\0018x5100) ;
} // group \1.2.840.113619.2.22.287.1.220.5.3.20060303.290234
} // group \1.2.840.113619.2.22.287.1.220.5.20060303.290116
```

```
    } // group \1.2.840.113619.2.22.287.1.220.20060303.282359
  } // group MAGNUS\,MARCONE
}
```

ANEXO B – E-mail enviado por desenvolvedor da UCAR Unidata Program

O resultado peculiar dos testes realizados com o NetCDF gera preocupação com a correteza do modelo implementado. E um questionamento inevitável surge, se a ferramenta foi usada de maneira adequada e se não existe outra forma de se aproveitar suas funcionalidades neste problema. Com esses questionamentos foi enviado um e-mail para o grupo de desenvolvimento do conjunto de ferramentas NetCDF *UCAR Unidata Program*. A resposta confirma nosso método e não nos mostra outra maneira melhor de implementar o modelo proposto.

Hi Marcone,

Sorry to take so long to respond to your question ...

When a netCDF-4 file is opened, it reads all the metadata into memory once, so that later references to the metadata can be accessed quickly. By "metadata", I mean

- names and sizes of dimensions - names, shapes, and types of variables - names and types of attributes - names of groups and their subgroups - definitions of all user-defined types

This could be slow if you add a large amount of metadata to the file before closing and re-opening it. However, if you are just adding more data to the file (values for variables already defined), it should not slow down the nc-open calls significantly. Most of the use cases for netCDF-4 that we have seen benefit from reading in all of the metadata when the file is first opened, to speed up access to the data and metadata on subsequent calls while the file is still open.

The underlying HDF5 library works differently, only reading in metadata as needed, so it is faster for cases such as a large number of nested groups where the common case is to only read data from a small subset of those groups before closing the file. That makes the open much faster, but each read that has to access metadata slower.

We have considered implementing an optional "fast open" by following the HDF5 model, but so far there has not been enough demand for that feature to make it a high priority for development.

The only suggestions I have are to

- consider making more use of data and less use of metadata for representing your data structures. For example, instead of using thousands of separate small variables, use a smaller number of variables with indexing, or use large multidimensional variables instead of many small variables.

- similarly, if you have thousands of deeply nested groups, consider a design that uses indexing in a few groups instead of relying on recursion in deeply nested groups.

- consider using HDF5 directly instead of netCDF-4, to see if it's model of lazy evaluation of metadata is better suited for your data representations

- try to keep the file open while it is used, to amortize the cost of opening and reading in all the metadata

-Russ