JOSIANE M. DINIZ DUSZCZAK

UMA ABORDAGEM ARQUITETURAL DE GRADE DE DADOS PARA IMAGENS FITS

CURITIBA

2012

JOSIANE M. DINIZ DUSZCZAK

UMA ABORDAGEM ARQUITETURAL DE GRADE DE DADOS PARA IMAGENS FITS

Dissertação apresentada como requisito parcial à obtenção do grau de mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profª Drª Maria Salete Marcon Gomes Vaz

CURITIBA

2012

JOSIANE M. DINIZ DUSZCZAK

UMA ABORDAGEM ARQUITETURAL DE GRADE DE DADOS PARA IMAGENS FITS

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Informática da Universidade Federal do Paraná, pela Comissão formada pelos professores:

Orientadora: Profª Drª Maria Salete Marcon Gomes Vaz
Departamento de Informática, UFPR

Prof. Dr. Márcio Augusto de Souza
Departamento de Informática, UEPG

Prof. Dr. Marcos Sfair Sunye
Departamento de Informática, UFPR

Curitiba, 29 de fevereiro de 2012

AGRADECIMENTOS

Primeiramente, quero agradecer a Deus pela saúde, fé e perseverança que tem me dado. Aos meus pais Edivaldo e Nadir que sempre me apoiaram e me deram condições de estudar. Ao meu esposo Thiago que me apoiou e entendeu as horas que eu estava ausente devido aos estudos, apoiando-me em momentos difíceis e comemorando nos momentos de vitórias. Meus irmãos Valéria e Fernando e meu cunhado Diogo que sempre me deram forças de sempre seguir em frente. A minha sogra que não está mais entre nós, mas tenho certeza que está torcendo por mim.

Não poderia deixar de agradecer as minhas amigas e orientadoras, Salete e Lucélia, que sempre me ajudaram com sabedoria e me deram forças para não desistir. Muito obrigada.

Gostaria também de agradecer ao professor Márcio. Obrigada pelas horas dedicadas comigo no laboratório, me ajudando na instalação do *middleware* Globus.

O Senhor é meu pastor e nada me faltará – Bíblia Sagrada.

Tenha firmeza em suas atitudes e persistência em seu ideal, mas seja paciente, não pretendendo que tudo lhe chegue de imediato. Há tempo para tudo. E tudo o que é seu virá as suas mãos, no momento oportuno. Saiba esperar o momento exato e receberá os benefícios que pleiteia. Aguarde com paciência que os frutos amadureçam para que possa apreciar devidamente sua doçura – C. Torres Pastorino.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SIGLAS

API= Application programming interface

ASCII= American Standard Code for Information Interchange

CAS=Catalog and Archive Server and Client

CSDGM= Content Standard for Digital Geospatial Metadata

DFDL= Data Format Description Language

FITS=Flexible Image Transport System

FGDC= Federal Geographic Data Committee

GRAM= Globus Resource Allocation Manager

GSI= Globus Security Infrastructure

GAT = Grid Application Toolkit

GASS= Global Access to Secondary Storage

GRIDFTP= Grid File Transfer Protocol

GNU= General Public License

MPPs= Processadores Maciçamente Paralelos

MDS= Monitoring and Discovery System

MCS = Metadata Catalog Service

NASA = National Aeronautics and Space Administration

OGSA-DAI= Innovative Solution for Distributed Data Access and Management

PBS= Deploying Torque (Open PBS)

RDF= Resource Description Framework

RSH= Remote shell, a UNIX command-line utility for remotely executing commands

SPARQL= RDF Query Language

SVN= Subversion (também conhecido por svn) é um sistema de controle de versão

URL = Uniform Resource Locator

URI = Uniform Resource Identifier

XML = Extensible Markup Language

Web= World Wide Web

RESUMO

A necessidade de acesso, armazenamento e processamento de grande quantidade de dados, resulta em uma alta demanda de processamento, principalmente em astronomia, medicina, física, biologia e engenharia. Grades de dados são meios para prover e gerenciar recursos computacionais distribuídos para aplicações científicas, as quais demandam poder computacional ou utilizam equipamentos de uso específico na ciência. Alguns dos benefícios da utilização da arquitetura de grade são: tempo para executar uma tarefa complexa, possibilidade de um sistema unificado, interface padronizada, maneira comum de executar cálculos e gerenciar dados resultantes, transferência eficiente de dados, velocidade e armazenamento de grande quantidade de dados. As grades de dados permitem a manipulação de grandes quantidades de dados e o compartilhamento coordenado e heterogêneo dos dados distribuídos. Já os metadados descrevem os dados, como meio de identificar e fornecer acesso e recuperação dos mesmos. Para manipulação dos metadados, é necessária uma padronização para favorecer a interoperabilidade dos dados. Quando não existe padronização dos metadados criados, o acesso e a recuperação dos dados e recursos da grade tornam-se tarefas difíceis devido a heterogeneidade dos metadados criados. O formato FITS é utilizado para manipular, armazenar e transmitir imagens científicas. Tal formato é muito utilizado na Astronomia. Esta dissertação apresenta uma arquitetura de grade dados para imagens FITS e a especificação de um padrão de metadados para gerenciamento desse tipo de imagens, estendido para o contexto das grades, como forma de promover a interoperabilidade, facilitando o acesso e a recuperação dessas imagens.

ABSTRACT

The need for accessing, storaging and processing large data amounts, implies in a high demand of processes, especially in astronomy, medicine, physics, biology and engineering. Data Grids are main to provide and manage distributed computing resources for scientific applications, which require computational power or the use of specific equipment in science. Some of the benefits that come from grid architecture are: time to perform a complex task, the possibility of a unified system, standardized interface, common way to perform calculations and manage the resulting data, efficient data transfer, speed and storage of large amounts of data. Data grids allow large amounts of data handling and sharing coordinated heterogeneous and distributed data. Metadata describes the data in order to identify and provide its access and retrieval. If handling this metadata, standardization is required to ease the data interoperability. When the created metadata is not standardized, data and resources access/retrieval from the grid become a difficult task, due to heterogeneity of this metadata. The FITS format is used to manipulate, store and transmit scientific images. This format is widely used in astronomy. This dissertation presents a data grid architecture to FITS images and specify the metadata standard for management of such images, extending it to the grids context, in order to promote interoperability, turning the access and retrieval of these images not complicated.

CAPÍTULO 1

1. INTRODUÇÃO

Neste capítulo é apresentada a motivação para o desenvolvimento desta dissertação, onde são enfatizadas a importância e relevância do trabalho. São descritos os objetivos da dissertação, bem como um resumo dos objetivos de cada capítulo.

1.1. Motivação

Cada vez mais existe a necessidade de acesso, armazenamento e processamento de grande quantidade de dados por entidades empresariais, acadêmicas e/ou governamentais. Em algumas áreas da ciência, a necessidade pela alta demanda de processamento se tornou obrigatória, como na astronomia, medicina, física, biologia e engenharia.

Neste cenário, o conceito de grades (*grids*) corresponde a um meio para prover e gerenciar recursos computacionais distribuídos para aplicações científicas, as quais demandam grande poder computacional ou utilizam equipamentos de uso específico na ciência [10].

As grades podem ser definidas como uma plataforma para execução de aplicações paralelas, em um ambiente heterogêneo e amplamente distribuído, não possuindo um controle central [35]. Existe uma série de aplicações de ambientes de grades a serem desenvolvidas, assim como existem tecnologias que oferecem funcionalidades necessárias para esses ambientes.

O objetivo das grades computacionais é proporcionar recursos como um serviço público, semelhante aos sistemas elétricos ou telefônicos, onde usuários usufruem dos recursos, pagando pela utilização dos mesmos. No caso das grades, dependendo da política adotada, pode-se ou não pagar pelo recurso em uso.

Alguns dos benefícios da utilização da arquitetura de grade são: executar tarefas complexas com maior velocidade, possibilitar sistema unificado, permitir interface padronizada, executar/gerenciar cálculos e dados resultantes, transferir/trocar dados, e armazenar grande quantidade de dados.

De modo geral, as grades podem ser divididas nas seguintes categorias [22]: (i) Grade computacional, utilizada para solucionar problemas computacionais complexos, tais como processamento de imagens médicas; (ii) Grade computacional oportunista,

focando na utilização de ciclos computacionais ociosos; (iii) Grade de dados, a qual gerencia e distribui os dados, sendo o foco desta dissertação; e (iv) Grade de serviços, que oferece serviços viabilizados pela integração de recursos na grade.

Uma grade de dados pode ser definida como um ambiente capaz de gerenciar os dados distribuídos, fornecendo suporte de acesso, sincronização e coordenação dos dados distribuídos, em locais remotos [1].

Nesses ambientes, usuários têm a possibilidade de acessar repositórios de dados e executar aplicações intensivas, aplicações que demandam grande poder computacional, proporcionando a análise de dados e o compartilhamento dos resultados gerados. Esses usuários podem visualizar os dados de locais diferentes.

Em grades de dados são manipuladas grandes quantidades de dados, permitindo o compartilhamento coordenado e heterogêneo dos dados distribuídos. Podem ser definidos metadados para descrever os dados, como meio de identificá-los para fornecer acesso e recuperação dos mesmos.

Para manipulação dos metadados, é necessária padronização para favorecer a interoperabilidade dos dados. Não havendo padronização, a heterogeneidade dos metadados criados torna as tarefas, como acesso e a recuperação dos dados e recursos da grade, não triviais.

No contexto desta dissertação os dados gerenciados em grades de dados serão dados em Formato *FITS* – Sistema de Transporte de Imagens Flexíveis, usado para manipular, armazenar e transmitir imagens científicas. Esse formato é muito utilizado na Astronomia. Ao contrário de muitos formatos de imagem, é projetado especificamente para dados científicos e, portanto, inclui opções para descrever informações de dados espaciais, juntamente com os metadados de origem da imagem.

Na área de astronomia, muitos projetos de grades de dados estão em desenvolvimento, e não existe uma padronização específica para imagens do tipo FITS. Para tanto uma arquitetura de grades de dados definida e o uso de padrões de metadados é necessário. Um padrão de metadados corresponde a um conjunto de informações (esquema de metadados) definido para atender um determinado contexto. Através da identificação de problemas no armazenamento e recuperação de informação por falta de padronização, vários esquemas são criados para atender diferentes propósitos [28].

Existem algumas questões críticas referentes aos metadados em ambientes de grades de dados [1]:

- ***Distribuição e Uniformidade de Acesso***. Dados no ambiente de grades são distribuídos e gerenciados por organizações diferentes, onde os metadados associados aos dados seguem o mesmo princípio. Existe a necessidade de acesso uniforme aos repositórios de metadados e ontologias, para oferecer visões integradas em grandes coleções de metadados distribuídos.
- ***Metadados devem ser acessíveis através de protocolos orientados a serviços***. Embora não possua um modelo de dados comum, é necessário um modelo comum para gestão de metadados. No mínimo, é necessário manter a associação entre os recursos e metadados e apoiar as linguagens de consulta com os metadados específicos do modelo.
- ***Gerenciamento do ciclo de vida de metadados.*** Os metadados são dinâmicos e seu ciclo de vida pode variar em ordens de magnitude. É necessária a automação de todos os aspectos de gerenciamento de metadados, ou seja, formas de verificar mudanças de estado de metadados e propagá-los usando uma infraestrutura de notificação.
- ***Controle de acesso uniforme e granular aos metadados***. Duas questões são levantadas no controle de acesso. Deve ser permitida a permissão para acessar e manipular elementos de metadados, em diferentes níveis de granularidade, conforme o formato dos metadados. E os mecanismos de controle de acesso devem ser uniformes em todo repositório de metadados.

Atualmente, não existe a definição de um padrão para armazenamento de metadados de imagens FITS, em ambientes de grades de dados. Cada organização define um formato, conforme suas necessidades, afetando o desempenho na consulta e recuperação das imagens.

Na astronomia, para tratamento de imagens terrestres, existe um padrão mundialmente reconhecido denominado CSDGM – *Content Standard for Digital Geospatial Metadata* [4]. Esse padrão não é adequado para o gerenciamento de imagens geoespaciais do tipo FITS, pois possui diversos metadados desnecessários, excessivos e/ou que não se aplicam para esse tipo de imagens, assim como não trata ambientes específicos de grade de dados.

Muitas organizações e pesquisadores estão usando tecnologia da grade de dados, porém falta uma interface transparente e amigável para manipulação dos dados, devido

à complexidade desses ambientes. A utilização de portais é uma opção para fornecer, ao usuário final, um ambiente intuitivo, amigável e de fácil utilização.

1.2. Objetivos

Esta Dissertação tem como objetivo principal apresentar uma abordagem arquitetural de grades de dados para imagens FITS e a especificação de um padrão de metadados para essas imagens científicas. A arquitetura tem como objetivo ajudar os cientistas a construir uma grade de dados e a especificação de um padrão de metadados busca facilitar os processos de consulta e recuperação das imagens.

Os objetivos específicos são os que seguem: (i) levantar o estado da arte de grades de dados e metadados; (ii) apresentar uma arquitetura de grades de dados para imagens FITS; (iii) especificar um padrão de metadados para imagens FITS em ambientes de grades de dados; e (iv) fazer uma análise comparativa da contribuição com trabalhos correlatos.

1.3. Estrutura da Dissertação

A Dissertação está estruturada como segue. Além deste capítulo introdutório, existem mais 4 (quatro) capítulos e um anexo onde são especificados os detalhes de instalação dos componentes da arquitetura. No Capítulo 2 são descritos conceitos inerentes a grades de dados, metadados, portais para grades envolvendo arquiteturas existentes. Também são apresentados conceitos de imagens FITS e padrões de Metadados. No Capítulo 3 é apresentado o padrão de metadados especificado para imagens FITS em grade de dados e a arquitetura de grade de dados para essas imagens. No Capítulo 4 são abordados trabalhos relacionados e um comparativo entre eles é apresentado. O Capítulo 5 apresenta as conclusões e perspectivas de pesquisas futuras.

CAPÍTULO 2

2. GRADES DE DADOS E METADADOS

Neste capítulo são abordados conceitos, características, benefícios e aplicações de grades de dados e do *middleware* Globus [12]. Além disso, são descritos conceitos inerentes a grades de dados e padrões de metadados. Também são apresentados conceitos de Imagens FITS, portais para grades e são descritos os Padrões de Metadados CSDGM - *Content Standard for Digital Geospatial Metadata* [9] e Metafits [29].

2.1. Grades

Uma grade pode ser definida como uma rede, onde o usuário se conecta para prestar e receber serviços computacionais. Corresponde a uma composição de infraestrutura de hardware e software, permitindo acesso a grandes capacidades computacionais, geograficamente distribuídas, de forma confiável, consistente, econômica e persistente [33].

Um serviço pode ser uma rede elétrica, disponibilizando eletricidade de acordo com a demanda, escondendo do solicitante os detalhes, tais como a origem da energia elétrica, complexidade de transmissão e distribuição. De forma semelhante, a infraestrutura da grade permite o compartilhamento de recursos, tais como dados, capacidade de processamento e armazenamento [34].

As grades utilizam recursos independentes e amplamente dispersos como plataforma de execução de aplicações paralelas [11]. Uma grade provê uma coordenação de recursos com o mínimo controle centralizado, baseado em padrões abertos, provendo uma qualidade não trivial de serviços.

As grades foram possíveis devido à melhoria do desempenho e à redução de custos, tanto de redes de computadores quanto de microprocessadores. As grades podem ser utilizadas para propósitos computacionais (grades computacionais) ou para armazenamento de dados em larga escala (grades de dados), ainda podendo haver combinação de ambas.

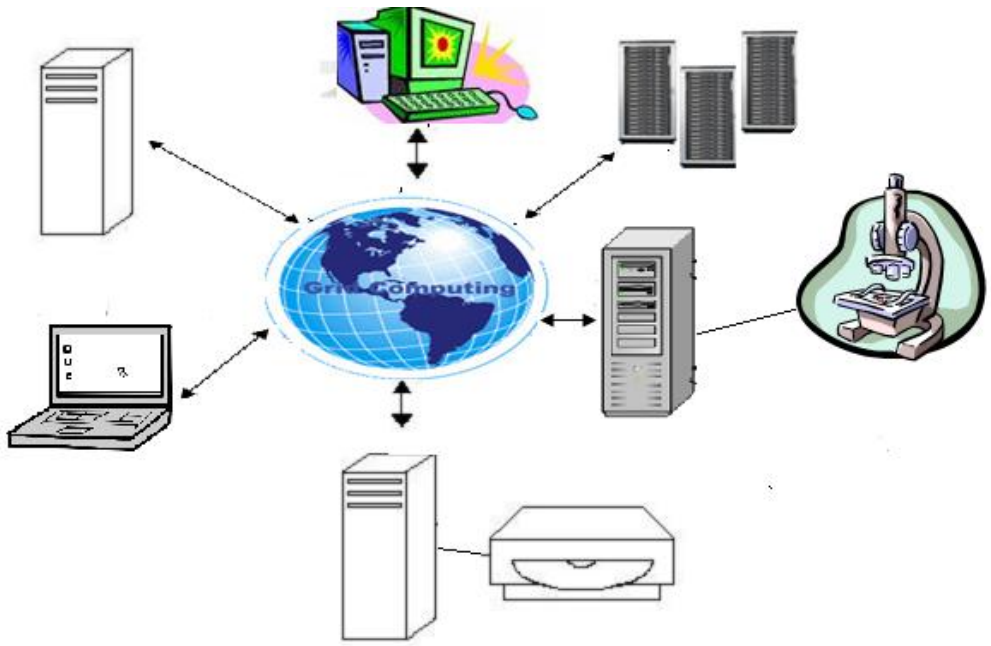

Figura 1. Ambiente de Grade [34].

A arquitetura de um ambiente de grade (Figura 1) mostra computadores pessoais, servidores, *notebooks*, discos e recursos interligados através da grade. Essa interligação proporciona compartilhamento de recursos, capacidade de processamento e armazenamento.

Forster e Kesselman (1999) traduzem grades de duas formas clássicas: (i) compartilhamento de recursos coordenados, com resolução de problemas em organizações virtuais multi-institucionais dinâmicas; e (ii) suporte à execução de aplicações paralelas que acoplam recursos heterogêneos distribuídos, oferecendo acesso consistente e barato aos recursos, independente de sua localização.

2.1.1. Características das Grades

Algumas características das grades são [11]: heterogeneidade, compartilhamento, alta dispersão geográfica, diversos domínios administrativos e controle distribuído.

A heterogeneidade das grades trata com diversos recursos diferentes, softwares de várias versões, equipamentos e serviços dos mais variados tipos. Uma grade pode ser dedicada a várias aplicações, permitindo compartilhamento de recursos.

As grades podem atingir uma escala global, agregando serviços localizados em várias partes do planeta. Assim, têm como característica a alta dispersão geográfica.

No que se refere aos diversos domínios administrativos, podem existir várias políticas de acesso e uso dos serviços, de acordo com as diretrizes de cada domínio participante da grade.

Em geral, várias entidades possuem poder sobre a grade, pois cada organização pode estabelecer sua política local, com necessário controle distribuído. Nas próximas seções serão apresentados os benefícios oferecidos pelas grades e sua aplicação.

2.1.2. Benefícios de utilização

Um dos maiores benefícios das grades é o fornecimento de diversos serviços computacionais, utilizando recursos localizados em instituições diferentes e geograficamente dispersos. Os recursos podem ser alocados utilizando vários computadores, conectados via *Internet* com menor custo, pois não é necessária a utilização de computadores de grande porte.

Alguns benefícios proporcionados pelas grades são: (i) agregação de recursos; (ii) melhoria de qualidade e velocidade de serviços, acesso distribuído aos recursos, colaboração entre pesquisadores e; (iii) aproveitamento de recursos ociosos.

As organizações podem agregar recursos com toda a infraestrutura, não importando a localização geográfica. Na melhoria de qualidade e velocidade de serviços, as organizações podem melhorar os serviços disponibilizados devido à colaboração transparente dos recursos compartilhados.

O acesso distribuído aos recursos permite acesso e compartilhamento das bases de dados, de forma remota. A colaboração entre pesquisadores possibilita auxílio em projetos, por conta da habilidade no compartilhamento de experimentos, desde conceitos, estudos de casos e validações.

Existe aproveitamento dos ciclos de processamento ociosos disponíveis dos computadores, encontrados em várias localidades. Por exemplo, os computadores ociosos durante a noite em uma empresa em Tóquio, podem ser utilizados durante o dia para operações na América do Sul. Portanto, a tecnologia de grades possibilita agregar recursos computacionais, variados e dispersos, em um único supercomputador virtual, acelerando a execução de várias aplicações paralelas.

2.1.3. Aplicações de Grades

As grades estão sendo utilizadas nas mais diversas áreas, tais como ciência, educação, engenharias e comercial/industrial. Um exemplo é o estudo do genoma, desenvolvido pela Rede Biomedicinal de Pesquisa da Informação - BIRN.

Um dos benefícios das grades para a engenharia está relacionado na melhora do desempenho computacional. O *Network for Earthquake Engineering Simulation Grid* - NEESgrid [27], interliga vários centros de pesquisa, tendo como foco a pesquisa e análise de tremores de terra.

As grades proporcionam à física o desempenho computacional necessário para as experiências científicas. O *Grid Physics Network* - GriPhyN [16] corresponde a uma tecnologia em grade desenvolvida para pesquisas físicas, com característica de coletar e analisar dados de forma distribuída.

As grades trazem benefícios financeiros às instituições, sendo que as soluções podem ser voltadas ao armazenamento de dados, jogos on-line, sistemas de busca, entre outros.

O projeto da *NS Solutions* [30] testa o uso da tecnologia de grade para melhorar o planejamento de sistemas de produção em usinas de ferro. São realizados cálculos, em tempo real, como base nas especificações e prazos de entrega.

As grades também produzem benefícios para a Astronomia onde, muitas vezes, são necessários vários dias para concluir um experimento, levando em consideração apenas uma máquina. Com a utilização de uma grade, o mesmo experimento pode ser concluído em algumas horas, ou ainda, em se tratando de compartilhamento de dados, uma instituição no Brasil pode consultar dados de uma instituição do Japão ou vice versa.

2.2. Grades de Dados

Uma Grade de Dados tem como objetivo fornecer serviços para descobrir, transferir e manipular grandes quantidades de dados armazenados, assim como criar e gerenciar cópias desses conjuntos de dados. Alguns dos desafios e características das grades de dados são:

- Grandes conjuntos de dados: os dados armazenados nas grades de dados são do tamanho de TB (*terabytes*) e PB (*petabytes*). Existe uma grande preocupação em

minimizar latências de transferências de dados, criando assim estratégias de réplicas;

- Coleções de dados compartilhados: participantes dentro de grades de colaboração científica, muitas vezes, necessitam utilizar os mesmos repositórios como fonte de dados e armazenamento dos resultados de sua análise;
- Unificação do *namespace*: o nome do arquivo lógico é mapeado para um ou mais nomes de arquivos físicos em vários recursos; e
- Restrições de acesso: usuários desejam garantir a confidencialidade dos seus dados, restringindo o acesso para determinados colaboradores.

Para garantir autenticação e autorização em grades de dados, existe a necessidade de desenvolver grandes controles de acessos.

Foster [12] propôs uma arquitetura de grade para compartilhamento de recursos entre diferentes entidades, baseado no conceito de VOs (*Virtual Organization*).

Uma VO é formada quando diferentes organizações colaboram para atingir um objetivo comum. Uma VO define recursos disponíveis e regras de acesso para utilização dos recursos, como por exemplo, em quais condições um recurso pode ser utilizado. Recursos podem ser definidos como: recursos de armazenamento, rede, software, instrumentos científicos e dados.

Uma VO, também, fornece protocolos e mecanismos para determinar a adequação e acessibilidade dos recursos disponíveis. A arquitetura de grade de dados (Figura 2) está dividida em camadas [1].

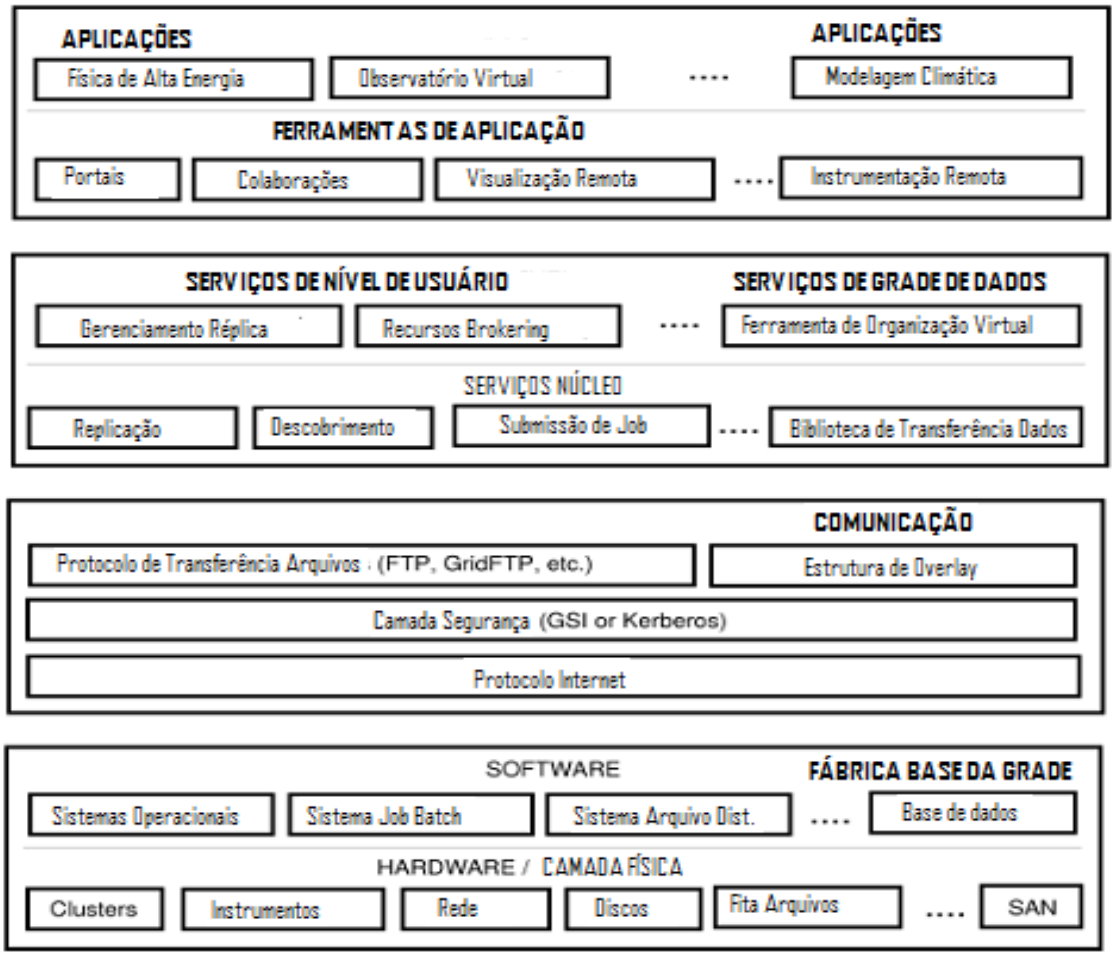

Figura 2. Arquitetura de uma Grade de Dados [1].

Cada camada tem por base os serviços oferecidos pela camada inferior, além de interagir e cooperar com os componentes no mesmo nível (por exemplo, *Resource Broker* invoca ferramentas VO):

- *Fábrica Base da Grade*: consiste em recursos computacionais distribuídos (*clusters*, supercomputadores), recursos de armazenamento (RAID *arrays*, arquivos de fita) e instrumentos (telescópio, aceleradores) conectados por rede.
- *Comunicação:* possui protocolos utilizados para recursos de consulta na camada *Fábrica Base da Grade*, realizando transferências de dados.
- *Serviços da Grade de Dados*: provê serviços para gerenciar e processar dados em uma grade de dados.
- *Aplicações:* consiste dos serviços específicos que atendem aos usuários, personalizando serviços conforme os domínios.

Algumas caracteríticas das grades de dados são:

- **Propósito**: grades de dados têm como propósito possibilitar a colaboração através de compartilhamento de recursos, com o objetivo de obter os benefícios da agregação.
- **Agregação:** pode ser definida através de um processo *ad hoc*, pois os nós se conectam na rede, sem acordos prévios ou um processo específico, reunidos para uma finalidade específica. Grades de dados são criadas com o objetivo de unir recursos para atingir um objetivo comum, sendo que configurações dinâmicas são possíveis devido à introdução ou retirada dos recursos na grade.
- **Organização:** em grades de dados existem quatro diferentes tipos de organizações: monádica, hierárquica, federada e híbrida:
    - *Monádica*: dados são agrupados em um repositório central, respondendo consultas de usuários para prover dados, os quais são de diversas fontes. A diferença é que existe um ponto central e os dados não precisam ser distribuídos.
    - *Hierárquica*: existe uma única fonte de dados e esses precisam ser distribuídos (através de colaborações), podendo ser através de réplicas.
    - *Federada*: criada por instituições que desejam compartilhar dados em bases já existentes.
    - *Híbrida*: esse combina os três tipos anteriores.
- **Tipo de Acesso aos Dados:** geralmente, os acessos são operações de leitura. Porém, grades de dados podem também apoiar a atualização de réplicas de dados, se a fonte for modificada.
- **Descoberta dos Dados:** Grades de dados são organizadas em catálogos que mapeiam a descrição lógica dos dados para o dado real (armazenamento físico). Dados são localizados por meio de consultas aos catálogos que resolvem a localização física dos conjuntos lógicos. Para descobrir dados, algumas grades utilizam metadados.
- **Desempenho e Gerenciamento de Latência:** Um elemento chave de desempenho em redes de dados com grande intensidade é a maneira pela qual eles reduzem a latência de transferência de dados. Algumas técnicas são: replicação, cache, entre outras. A diferença entre replicação é que é feita a partir da fonte de dados (lado do provedor) e o cache é ao lado do consumidor. A

replicação envolve a criação e a manutenção de cópias de dados em diferentes locais na rede, dependendo das taxas de acesso, por exemplo. A replicação tem como objetivo aumentar o desempenho, reduzir a latência e aumentar a confiabilidade, criando diversas cópias de *backup*. Grades de dados transferem grande conjunto de dados, motivando o desenvolvimento de mecanismos de alta velocidade para transferência de dados. Mensagens de controle são enviadas separadas das transferências de dados reais. Métodos de otimização de transferências de dados, como acesso a dados próximos ao seu ponto de consumo, são empregados na grade.

- **Consistência:** grades de dados, atualmente, não possuem apoio à recuperação e reversão dos dados, sendo assim não é possível garantir que os dados são consistentes.

2.3. Metadados e Padrões

Os metadados são definidos como dados que descrevem dados, ou, como a descrição detalhada das instâncias de dados, dos formatos, das características e dos valores dependentes dos mesmos [38]. Podem ser utilizados para descrever objetos ou, tornar pública sua existência.

Os metadados disponibilizam, descrevem, localizam e auxiliam na compreensão dos dados, transformando-os em conhecimento [40]. Ao ter conhecimento de quais dados estão disponíveis, entender o seu contexto e onde estão localizados, informações precisas são obtidas e melhores decisões podem ser tomadas [38].

Ao serem identificados e registrados, os metadados são gerenciados como elementos dentro do repositório, nomeados e devem possuir um contexto, podendo conter informações sobre o domínio de negócio, áreas, sistemas, banco de dados, modelagem ou ambiente.

Os metadados podem ser classificados como [36]: administrativos, descritivos, de preservação, técnicos, estruturais e pelo seu uso. Os metadados administrativos são utilizados na gestão de recursos de informação. Já os descritivos descrevem características de um documento, facilitando sua identificação, pesquisa e o gerenciamento das informações.

Os metadados de preservação são aqueles que salvaguardam as informações. Os técnicos estão relacionados ao funcionamento do sistema e do comportamento dos

metadados. Os estruturais descrevem a forma como os objetos se interligam. Finalmente, os metadados de uso relacionam-se ao nível e ao tipo de uso dos recursos tecnológicos.

O conjunto de informações corresponde a um esquema de metadados definido para atender um determinado contexto. Através da identificação de problemas no armazenamento e recuperação de informação por falta de padronização, vários esquemas são criados para atender diferentes propósitos, chamados de padrões de metadados [28].

A utilização de padrões já foi vista como forma de limitação entre a comunidade de desenvolvedores. Atualmente, com o crescimento dos dados armazenados, padrões são vistos como aliados. O uso de padrões resulta em benefícios para a comunidade, tais como facilitar a atividade de análise, pois geralmente são amplamente documentados, e facilitam a comunicação entre os usuários, proporcionando uniformidade e integração entre soluções.

Dentre os padrões existentes para descrição de objetos, destacam-se:

- *Dublin Core Metadata Initiative* – DCMI [5]: define um grupo de quinze atributos que pode ser utilizado para descrever seus próprios recursos na *Web*, destacando-se pela simplicidade, interoperabilidade semântica, consenso internacional, extensibilidade e modularidade. Esse padrão pode ser utilizado nas mais diversas áreas de conhecimento, devido a utilizar atributos genéricos.
- *Metadata Encoding and Transmission Standard* – METS [25]: é um padrão implementado em XML para a codificação de metadados descritivos, administrativos e estruturais, utilizados para a gestão e a troca de objetos de repositórios de bibliotecas de objetos digitais. Estas bibliotecas requerem a manutenção de vários tipos de metadados estruturados. Esse padrão possui sete seções, em cada seção possui um grupo de atributos, sendo: Cabeçalho METS, Metadados Descritivos, Metadados Administrativos, Seção de Arquivos, Mapa Estrutural, Ligações Estruturais e Comportamento. Esse padrão tem como vantagem, ser coerente e flexível para descrição de objetos de bibliotecas digitais.
- *Movie Picture Experts Group - MPEG 7* [26]: utilizado para descrever dados de áudio e vídeo. A interface de descrição de conteúdo multimídia foi desenvolvida para prover um *template* utilizado em repositórios de dados

para recuperação automatizada em aplicações. Esse padrão é adequado para a descrição das informações estruturais e semânticas de conteúdos multimídia.

- *Learning Object Metadata – LOM* [6]: utilizado para descrição de objetos de aprendizagem, podendo ser qualquer entidade, digital ou não, usada, reusada ou referenciada no aprendizado, em meio tecnológico. O Padrão LOM usa uma abordagem estruturada para criação de metadados, utilizando nove categorias: Geral, Ciclo de vida, Meta-metadados, Técnica, Educacional, Direitos, Elação, Anotação e Classificação. Apesar de criar relacionamentos hierárquicos complexos, facilitando a descoberta de recursos, é um padrão de difícil utilização por usuários iniciantes, podendo resultar em registros de metadados incompletos ou insuficientes para pesquisa/recuperação de recursos com qualidade.
- *Content Standard for Digital Geospatial Metadata – CSDGM* [9]: possui um total de 245 elementos de metadados, divididos em sete grupos (descritos na próxima seção), requerendo uma excessiva quantidade de informações. É de aplicação questionável devido à exigência do preenchimento de um conjunto de formulários exaustivos.
- *MetaFits* [29]*:* padrão de metadados para documentação de imagens geoespaciais do formato FITS, sendo formado por 30 atributos, distribuídos em cinco categorias. Esse padrão é descrito em mais detalhes na próxima seção.

O uso de padrões de metadados permite facilitar a decisão de quais metadados devem ser coletados e mantidos, visto que os metadados podem ter uma variedade de formas. Além do fato de que diferentes comunidades podem usar o mesmo padrão e não propor diferentes tipos de metadados e/ou adotar vocabulários diferentes, facilitando comunicação e a interoperabilidade.

2.3.1. Padrão de Metadados CSDGM e Imagens FITS

O *Content Standard for Digital Geospatial Metadata* – CSDGM [9] foi um dos primeiros Padrões de Metadados, desenvolvido pelo *Federal Geographic Data Committee* - FGDC, sendo aprovado em 1994 e passando a vigorar a partir de 1995. Esse órgão do governo norte americano coordena o desenvolvimento, uso, compartilhamento e disseminação de dados geográficos.

O FGDC definiu um conjunto único de metadados para a documentação de dados geoespaciais digitais, estabelecendo os nomes dos metadados e seus agrupamentos, além de informações a respeito dos valores que devem ser fornecidos para cada elemento de metadados. Na aplicação prática, as normas têm servido de referência para, praticamente, todos os demais padrões geoespaciais propostos.

Os metadados que formam o padrão CSDGM são organizados em sete grupos: Identificação, Qualidade de Dados, Organização Espacial de Dados, Referência Espacial, Entidade e Atributo, Distribuição e Referência de Metadados.

O grupo de Identificação contém informações básicas a respeito do conjunto de dados como a descrição, palavras-chave, restrições de acesso e frequência de atualização e manutenção.

O grupo de Qualidade de dados compreende os metadados referentes à qualidade do conjunto de dados, como precisão dos dados, completude e fontes de informação, bem como métodos utilizados para a produção dos dados.

A organização espacial dos dados engloba metadados referentes aos mecanismos utilizados para representar a informação espacial do conjunto de dados. Já o grupo de Referência Espacial contém informações referentes ao sistema de projeção utilizado no conjunto de dados.

Entidade e atributo contêm detalhes sobre a informação contida no conjunto de dados, incluindo os tipos de entidade, seus atributos e o domínio dos valores que podem ser atribuídos aos metadados. Já a distribuição (ou fonte) contém informações a respeito do distribuidor e opções para a obtenção do conjunto de dados.

Finalmente, o grupo Referência de metadados contém informações sobre a parte responsável pelos metadados e a frequência com que os metadados são atualizados.

O Padrão de Metadados CSDGM é utilizado para a documentação de imagens geoespaciais digitais, ou seja, imagens obtidas da superfície terrestre, a partir de satélites.

As imagens FITS são obtidas a partir de telescópios localizados na superfície terrestre que fotografam, por exemplo, galáxias, constelações e superfícies planetárias (Figura 3).

Nebulosa Trífida

Galáxia

Figura 3. Imagens no formato *FITS* fotografadas pelo Telescópio Espacial Hubble

O formato de arquivo FITS [20] é comumente utilizado em astronomia, sendo um formato utilizado, principalmente, por instituições de pesquisa espacial, para armazenamento e troca de imagens astronômicas. Ele oferece parâmetros que podem estar associados às imagens [18].

Um arquivo FITS é composto pela imagem e por um cabeçalho associado à mesma. O arquivo utiliza o formato binário para o armazenamento das imagens e o formato ASCII para o armazenamento do cabeçalho (Figura 4), permitindo aos pesquisadores e máquinas o reconhecimento das informações contidas nesse cabeçalho.

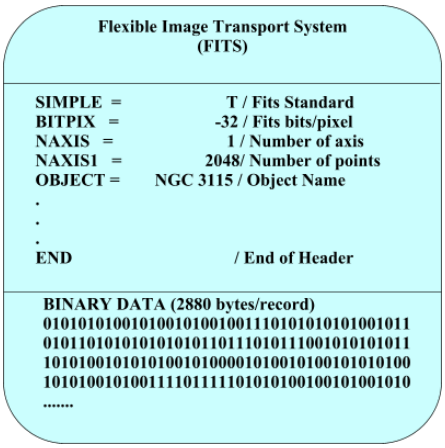

Figura 4. Esquematização de um arquivo no formato *FITS*.

Cada cabeçalho é constituído por vários registros, os quais são compostos por um valor precedido por uma palavra-chave correspondente. Esses pares palavra-

chave/valor fornecem metadados como nome, data, tamanho, origem, instrumento, coordenadas, comentários e história da imagem ao qual estão associados.

Em um cabeçalho, são seis as palavras-chaves obrigatórias [43]: 1) SIMPLE (tipo lógico) especifica se o arquivo está de acordo com as normas FITS; 2) BITPIX (tipo inteiro) especifica o número de bits utilizado para representar cada valor de *pixel*; 3) NAXIS (tipo inteiro) especifica o número de dimensões (eixos de coordenadas) da imagem; 4) NAXISn (tipo inteiro) informa o número de *pixels* utilizados na dimensão n; 5) OBJECT (tipo *string*) identifica o nome do objeto; 6) END indica o fim do arquivo.

O Metafits foi criado para a documentação das imagens FITS baseado nos grupos de metadados do Padrão CSDGM, uma vez que este tem sido a base para o desenvolvimento dos principais padrões de metadados existentes, porém o Metafits tem como maior benefício uma quantidade menor de atributos obrigatórios, focando em imagens obtidas através de telescópios localizados na superfície terrestre.

2.3.2. Padrão de Metadados METAFITS

Um padrão de metadados precisa ser claro, compreensível, consistente, completo, flexível e, principalmente, fácil de usar. Um número mínimo de metadados pode garantir uma estrutura de metadados homogênea, confiável e robusta [44].

A maioria dos padrões utiliza uma quantidade excessiva de metadados de caráter administrativo, quando aspectos de qualidade teriam maior relevância no uso e implementação de metadados. Como no Padrão CSDGM, as informações incluídas no MetaFits [29], para a documentação de imagens FITS, foram selecionadas com base em quatro características necessárias [44]: (i) Disponibilidade, metadados para descrever o conjunto de dados para localização geográfica; (ii) Adequação para uso, determinando se um conjunto de dados preenche uma necessidade específica; (iii) Acesso, estabelecendo os dados necessários para introdução de um conjunto de dados; e (iv) Transferência, definindo os dados necessários para processar e utilizar um conjunto de dados.

Baseado nisso, foi desenvolvido um padrão com as características de metadados inerentes às imagens FITS e adaptado a partir dos grupos de metadados do CSDGM. As palavras-chave do cabeçalho foram empregadas como metadados do MetaFits e novos metadados foram adicionados, objetivando reduzir a utilização dos metadados administrativos e focando nos metadados que referenciam os aspectos de qualidade.

Para documentação de imagens FITS, cinco grupos de metadados foram definidos baseados nos grupos utilizados pelo padrão CSDGM:

- *Identification*: compreende os metadados responsáveis pela caracterização básica das imagens, bem como pela identificação da mesma;
- *Quality*: agrupa os metadados inerentes à qualidade da imagem, como sua definição e clareza.
- *Coordinate System*: abrange os metadados responsáveis pela localização espacial do objeto presente na imagem.
- *Source*: contêm informações a respeito das pessoas, entidades e/ou órgãos de pesquisa responsáveis pela geração da imagem FITS.
- *Reference*: contém os metadados responsáveis por informar quando os metadados foram gerados e quem foi responsável pela geração dos mesmos.

A categoria Referência Espacial, do CSDGM, não tem correspondente no padrão criado para a documentação de imagens FITS, uma vez que informações sobre sistemas de projeção não são necessárias para imagens espaciais, pois essas não são de superfície terrestre.

As categorias Entidade e Atributo, por serem substancialmente administrativas, não têm correspondentes no padrão. As demais categorias de metadados do padrão para documentação de imagens geoespaciais têm seus correspondentes no Padrão CSDGM.

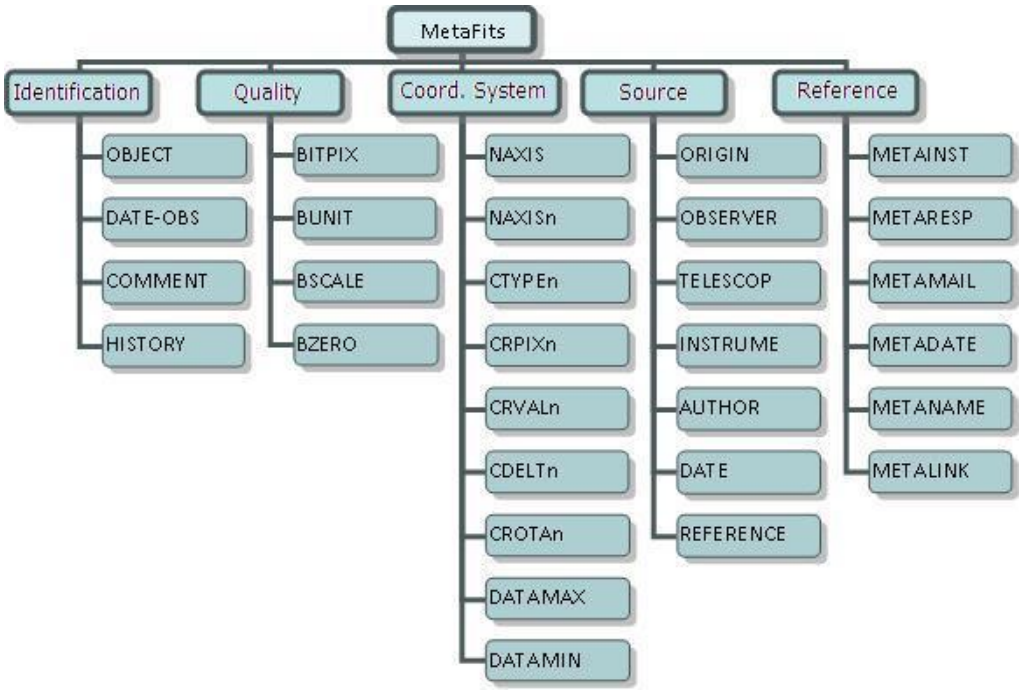

Figura 5. Organograma do Padrão MetaFits.

Os metadados definidos para o MetaFits foram estabelecidos com base nas características: disponibilidade, adequação para uso, acesso e transferência. Assim, o Padrão MetaFits possui um total de 30 (trinta) metadados distribuídos em cinco categorias (Figura 5): Identificação, Qualidade, Sistema de Coordenadas, Fonte e Referência.

Na Tabela 1 são apresentados os metadados da Categoria *Identification* compreendendo a caracterização básica das imagens.

Tabela 1. Metadados do Padrão MetaFits para a Categoria *Identification*

Metadados	Tipo de Dados	Descrição
OBJECT	String	Nome do objeto.
DATE-OBS	Date	Data de aquisição.
COMMENT	String	Comentários a respeito do objeto.
HISTORY	String	Descrição do processamento do objeto.

Na Tabela 2 são apresentados os metadados da Categoria *Quality* referentes à qualidade da imagem, como sua definição e clareza. Esses metadados estão diretamente relacionados com o conteúdo da imagem.

Tabela 2. Metadados do Padrão MetaFits para a Categoria *Quality*

Metadados	Tipo de Dados	Descrição
BITPIX	Int	Número de *bits*, por *pixel*, utilizado.
BUNIT	String	Unidade física de quantidade do objeto.
BSCALE	Float	Fator de escala usado na conversão dos elementos de imagem armazenados no conjunto de dados *FITS*, para valores físicos.
BZERO	Float	Valor físico correspondente ao valor zero armazenado na imagem.

Na Tabela 3 são apresentados os metadados da Categoria *Coordinate System*, responsáveis pela localização espacial do objeto presente na imagem.

Tabela 3. Metadados do Padrão MetaFits para a Categoria *Coordinate System*

Metadados	**Tipo de Dados**	**Descrição**
NAXIS	Int	Número de eixos.
NAXIS*n*	Int	Número de elementos no eixo *n.*
CTYPE*n*	String	Nome da coordenada física do eixo *n.*
CRPIX*n*	Float	Posição ao longo do eixo *n,* chamada de *pixel* ou ponto de referência, usada para definir a escala dos valores da coordenada física do eixo *n*.
CRVAL*n*	Float	Valor da coordenada física identificada por CTYPE*n,* no ponto de referencia do eixo *n.*
CDELT*n*	Float	Taxa de mudança da coordenada física, ao longo do eixo *n,* alterando a unidade no índice de contagem, obtida a partir do ponto de referência.
CROTA*n*	Float	Ângulo de rotação, em graus, do eixo *n* real.
DATAMAX	Float	Valor máximo de dados referentes ao sistema de coordenadas, após as transformações escalares terem sido aplicadas.
DATAMIN	Float	Valor mínimo de dados referentes ao sistema de coordenadas, após as transformações escalares terem sido aplicadas.

Na Tabela 4 são apresentados os metadados da Categoria *Source,* que contêm informações a respeito das pessoas e/ou entidades e/ou órgãos de pesquisa responsáveis pela geração da imagem *FITS.*

Tabela 4. Metadados do Padrão MetaFits para a Categoria *Source*

Metadados	Tipo de Dados	Descrição
ORIGIN	String	Instituição que gerou a imagem.
OBSERVER	String	Responsável pela geração da imagem.
TELESCOP	String	Telescópio utilizado na aquisição.
INSTRUME	String	Instrumento utilizado na aquisição.
AUTHOR	String	Autor que gerou os dados associados ao cabeçalho.
DATE	Date	Data em que o cabeçalho foi gerado.
REFERENCE	String	Referência bibliográfica da imagem.

A Tabela 5 apresenta a Categoria Reference, que contém os metadados responsáveis por informar quando os metadados foram gerados e quem foi responsável pela geração dos mesmos.

Tabela 5. Metadados do Padrão MetaFits para a Categoria *Reference*

Metadados	Tipo de Dados	Descrição
METAINST	String	Instituição responsável pelos metadados.
METARESP	String	Pessoa responsável pelos metadados.
METAMAIL	String	E-mail do responsável pelos metadados.
METADATE	Date	Data de geração dos metadados.
METANAME	String	Nome do arquivo de metadados.
METALINK	String	Endereço W*eb* contendo os metadados.

O MetaFits tem como vantagem a garantia de uma estrutura de dados sem o emprego excessivo de metadados administrativos, comparado com a maioria dos padrões de metadados para a documentação de imagens geoespaciais.

A desvantagem da sua utilização é a pequena quantidade de dados cadastrados e disponíveis para manipulação. Essa desvantagem tende a ser superada, de médio e longo prazo, através da utilização desse padrão e do aumento da quantidade de dados cadastrados segundo os critérios estabelecidos. Padrões utilizados há algum tempo, como o CSDGM possuem grandes bases de dados catalogadas de acordo com seus critérios.

2.4. Portais para Grade

A requisição de quaisquer serviços na grade não é uma tarefa trivial, devido à complexidade do ambiente da grade. Uma solução é a utilização dos portais [39], definidos como aplicações web que proveem personalização e conteúdo agregado de diferentes fontes e *hosts.*

Alguns dos benefícios dos portais são: facilidade de utilização, segurança, personalização, capacidade de agregação dos recursos e conteúdo, opções estas indispensáveis na utilização da grade. Portais permitem que usuários comuns e cientistas utilizem a grade, por proporcionar interfaces *web* padrão, sendo que o desenvolvedor pode adaptar o leiaute de acordo com a sua necessidade. Portais podem ser acessados via *browsers.* De maneira simplificada, os portais trabalham da seguinte maneira acessando os serviços das grades:

1 - Usuários, através de seus computadores requisitam informações e serviços através do portal;

2 - O portal analisa se o usuário possui acesso (autenticação na grade);

3 - Caso o usuário possua a permissão para a solicitação, o portal envia a solicitação do usuário para um componente da grade que esteja ocioso.

O Portal Gridsphere [15] permite 4 (quatro) tipos de papéis de usuários sendo eles: convidado, usuário, administrador e super usuário. Trabalha com conceitos de *portlets*, correspondendo a componentes visuais que podem ser assimilados dentro de páginas de um portal web. Eles provêm pequenos aplicativos que podem mostrar conteúdo informacional ou prover ingresso a outros serviços. Usuários podem customizar a aparência e funcionalidades que desejam acessar. Isso devido ao fato de que todo *portlet* possui características que permitem ao usuário configurá-las.

Os *portlet*s possuem suporte nos seguintes idiomas: francês, inglês, alemão, polonês, italiano, arábico e chinês. O Gridsphere possui biblioteca e coleções de *portlets* para gerenciamento de credencial, execução de *job* e transferência de dados. Para que o Gridsphere funcione de maneira correta um dos pré-requisitos é que *Apache Ant* esteja instalado e executando.

O *Gridsphere* é um f*ramework* que provê *portlets open-source* para portal *web*, permitindo o desenvolvimento rápido e fácil. Já o *Vine* [41] é uma biblioteca Java extensível e modular que oferece aos desenvolvedores uma ferramenta de fácil utilização, interface de programação de alto nível para grade. Vine dá suporte a uma

quantidade de *middleware* e serviços de terceiros, tais como Globus Toolkit [17], GLITE [13] e GRIA [14].

2.5. Grades de Dados e Metadados

Metadados em ambientes de grades auxiliam na recuperação, localização, acesso e gerência de dados. Algumas informações que os metadados descrevem são: proveniência dos dados, informações físicas e autoridades de acesso sobre os dados.

A proveniência dos dados é inerente a como os itens de dados são criados ou transformados, e por quais instrumentos científicos. Os metadados de informações físicas fornecem informações de tamanho e localização. Os metadados inerentes a autoridades de acesso sobre os dados descrevem as informações sobre os proprietários e leitores de dados.

Os metadados para grades de dados incluem três aspectos: informações de sistemas, de réplicas e de aplicativos [24]. As informações de sistemas abrangem informações estruturais sobre as grades de dados. Por exemplo, condições de serviços sobre a *Internet*, capacidade de armazenamento dos dispositivos de armazenamento, *status* de ociosidade do computador e políticas de uso.

As informações de réplicas descrevem o mapeamento entre arquivos lógicos e cópias físicas. Quanto às informações de aplicativos, estas descrevem atributos definidos pela comunidade. Por exemplo, conteúdo dos dados e as informações semânticas sobre os dados.

A publicação e funções de descoberta/recuperação de dados são baseadas em serviços de metadados. A publicação de dados é um processo de pesquisa de dados através de informações publicadas e fazendo delas atributos associativos e acessíveis aos usuários.

O processo de publicação necessita que os serviços de metadados possam combinar determinadas informações dos metadados com o conjunto de dados armazenados, para que após isso os dados possam ser utilizados pelo serviço de descoberta dos dados.

A descoberta de dados é um processo de identificação dinâmica, que identifica o conjunto de dados pertinentes e sua localização através de informações de atributos de consultas publicadas pelo serviço de metadados, ou informações como especificação da estrutura interna, membros, proveniência ou propriedades físicas, como tamanho e caminho de acesso.

Após a descoberta dos requisitos dos itens de dados, é possível ao usuário acessar os dados originais ou réplicas. Os serviços de descoberta são capazes de mostrar o conteúdo dos recursos de dados de uma forma estática.

2.5.1. Serviços de Metadados em Grades de Dados

Os metadados descrevem informações sobre item de dados e incluem significado para os dados, que podem ser encontrados facilmente ou ser combinados com outros dados, de acordo com metadados similares.

A principal razão de utilizar metadados em ambientes de grades é a enorme quantidade de dados armazenados. Além disso, as técnicas de pesquisa de dados tradicionais não tratam com uma quantidade tão grande de dados.

Os dados em ambiente de grades, geralmente, são heterogêneos. As diferenças estão no formato de armazenamento, representação dos dados e como eles são controlados. Além de serem de vários domínios científicos e possuírem anotações escritas por cientistas. Estas anotações devem ser organizadas de forma apropriada, a fim de serem associadas com dados originais tornando-se úteis para consulta de dados relevantes.

2.5.2. Tipos de Metadados para Grades de Dados

Em um arquivo de dados, com o uso de metadados, é possível descrever características físicas dos arquivos, tais como tamanho dos arquivos, localização no sistema de armazenamento, descrever informações sobre suas réplicas ou uma parte de seu conteúdo. Destaca-se que metadados podem ser de quatro tipos [24]: metadados de dados, definidos pelos usuários, de aplicações e de recursos.

Metadados de Dados

Os metadados de dados correspondem à entidade base em uma grade. Possui metadados físicos, de réplicas e de domínio. Os metadados físicos incluem características de armazenamento físico e propriedades, tais como tamanho de dados do arquivo ou data do objeto, localização, hora de criação, proprietário de criação, data de criação, formato ou tipo de arquivo. Outros metadados desta categoria são tipos de dados naturais. Cada sistema gerenciador de dados tem sua própria definição para tipo de dados e estas definições não são únicas.

Os metadados de réplicas estão relacionados aos registros de ligação entre o arquivo lógico de dados e suas réplicas de um ou mais arquivos físicos, e podem ser armazenadas em sistemas de arquivos diferentes ou base de dados. Esse tipo de metadados é frequentemente usado pelo sistema gerenciador de metadados.

Os metadados de domínio descrevem tipos de atributo de dados usados especificamente em domínios de dados. Metadados são provenientes de um conjunto de convenções definidas por cientistas e engenheiros que trabalham neste domínio.

Metadados Definidos pelo Usuário

O usuário é quem cria, usa ou modifica os dados e metadados. Os metadados definidos pelo usuário incluem informações como nome do usuário, endereço, e-mail, número de telefone e senha.

O domínio de atributos especifica para qual organização ou projeto o usuário trabalha, e o domínio pode ser justamente o nome do projeto ou organização.

O domínio de usuários pode ser agrupado por outros critérios. Um usuário pode ser registrado em um grupo, e todos os usuários têm os mesmos direitos para acessos aos recursos e controle dos mesmos. O uso de metadados facilita o gerenciamento de direitos de acesso aos metadados.

Metadados de Aplicações

Os dados são gerados por aplicativos e são utilizados na entrada e saída de aplicativos. As aplicações de metadados podem representar conteúdo de dados.

Algumas vezes, os aplicativos são divididos em vários componentes, e a entrada de dados é dividida. Informações de relacionamento entre os dados utilizados por aplicações de componentes devem ser registradas.

Aplicações de metadados incluem a informação de relacionamento de dados usados por toda a aplicação e seus componentes.

Metadados de Recursos

Os recursos são importantes elementos utilizados em gerenciamento de dados. São utilizados para criar, armazenar e transferir dados. Metadados de recursos descrevem características dos mesmos. Isto inclui endereçamento de acessos, localização física, tipo de recurso e lista de controle de acessos para recursos.

Existem duas formas para armazenamento de metadados, através de tabelas em banco de dados relacionais ou em arquivos XML. Um dos aspectos a serem considerados é a escalabilidade. Serviços de metadados executam um importante papel na descoberta e publicação de dados, onde de modo geral, os dados são gerados de diversos instrumentos científicos e armazenados em formatos padrões sendo disponibilizados para a comunidade.

Os serviços de dados possibilitam que cientistas adicionem metainformações que facilitem a descoberta dos dados.

2.6. Grades de Dados e Metadados no Globus

O Globus [12] é um *middleware* composto por um conjunto de serviços para a construção de uma infraestrutura de grade. Os serviços do Globus podem ser utilizados para garantir segurança, descoberta de recursos, controle e submissão de aplicações e movimentação de dados. Esses são essenciais em um ambiente de grade.

O Globus pode ser considerado o *middleware* de maior aceitação, devido aos serviços independentes oferecidos, existindo a possibilidade que seja utilizado parte dos seus serviços. Porém, o Globus não pode ser considerado uma solução completa para grades.

Para obter benefícios de um sistema em grade são necessários diversos protocolos, padrões e ferramentas de software. Várias organizações criam e validam *middlewares*. Globus é um *middleware* que contém um conjunto de serviços para segurança, gestão de dados, comunicação, detecção de falhas e gerenciamento de recursos.

Os serviços formados pelo *middleware* Globus podem ser usados separados ou em conjunto, utilizando computação em grade, por meio de *API - Application Programming Interface*. Este projeto cresceu através de uma estratégia *open-source*, o que motivou rápida e ampla adoção devido às melhorias contínuas na ferramenta.

É constituído por um conjunto de serviços que facilitam a construção de infraestruturas para Computação em grades. *Os* serviços *Globus Security Infrastructure - GSI, Globus Resource Allocation Manager - GRAM, Grid File Transfer Protocol - GRIDFTP, Reliable File Transfer – RFT, Replica Location Service - RLS e OGSA-DAI,* são descritos na sequência.

Globus Security Infrastructure – GSI

Na Computação em grade, serviços precisam efetuar ações de forma automática como solicitar o armazenamento ou acesso a um determinado arquivo. Essas ações necessitam de autenticação pela GSI [17], a qual viabiliza um *login* único na grade, utilizando criptografia de chave pública, certificados X.509 e comunicação *SSL - Secure Socket Layer* para identificação do usuário.

Após a identificação do usuário junto à GSI, todos os demais serviços Globus sabem que o usuário é de fato quem diz ser. Depois é necessário saber quais operações o usuário pode realizar, sendo feito através do mapeamento da identidade Globus para um usuário local.

Globus Resource Allocation Manager - GRAM

Quando um usuário submete uma aplicação para ser executada na grade, não há escalonador para controlar todo o sistema. O usuário utiliza um escalonador de aplicação e esse escolhe quais recursos serão utilizados, dividindo o trabalho com tais recursos e enviando as tarefas para os escalonadores dos recursos (Figura 6).

O *GRAM* [17] é o serviço da Arquitetura Globus que fornece uma interface para submissão e controle de tarefas, escondendo os múltiplos escalonadores dos demais serviços de grade. Fornece informações sobre o *status* do recurso ao MDS - *Monitoring and Discovery System,* que *é um* serviço Globus que fornece informações sobre a grade.

GRAM é um serviço Globus que permite aos usuários localizar, submeter, monitorar e cancelar trabalhos remotos em recursos computacionais baseados em grade. GRAM não é um escalonador de tarefas, mas provê um protocolo de comunicação com diferentes escalonadores de tarefas.

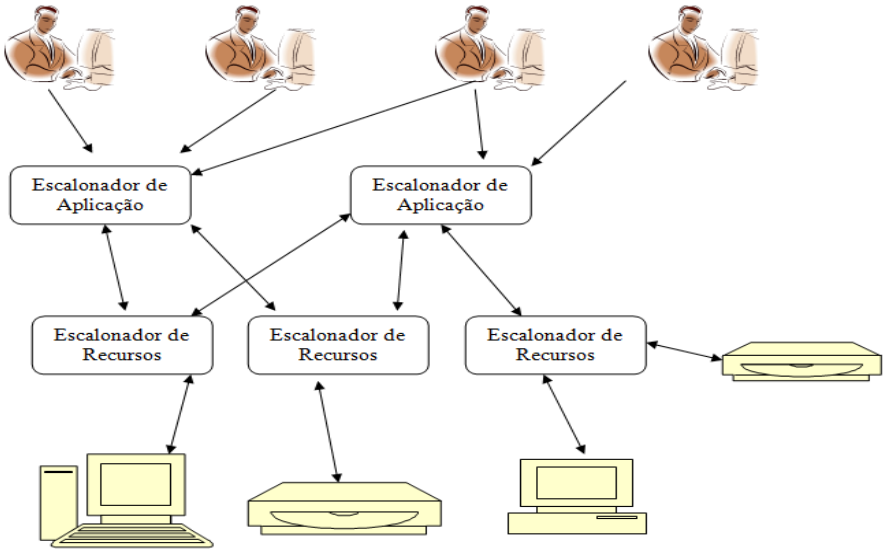

Figura 6. Cenário de Escalonamento do Globus [12].

O GRAM possibilita o monitoramento e o gerenciamento dos processos em execução, como também gera chamadas para os escalonadores locais de cada domínio integrante da grade.

A vantagem de utilizar o GRAM pelo cliente é a manipulação uniforme de tarefas e a submissão e o controle das mesmas, não importando qual é o escalonador de recursos usado para controlar a máquina. Isso é possível, visto que as requisições enviadas ao GRAM são escritas em *RSL - Resource Specification Language*, independente do escalonador de recursos que está sendo utilizado.

O gerenciador de tarefas é responsável em converter a requisição RSL em um formato que o escalonador de recursos entenda. O escalonador de aplicação pode utilizar os serviços de outros escalonadores de aplicação, devido à utilização da Linguagem RSL.

Um componente importante para a execução de aplicações fortemente acopladas é o *co-alocador*, escalonador de aplicação responsável em garantir que tarefas localizadas em máquinas distintas executem simultaneamente. Em aplicações fortemente acopladas, as tarefas precisam se comunicar para que a aplicação tenha progresso. Portanto, todas as tarefas da aplicação têm que ser executadas simultaneamente.

Um exemplo de submissão de uma aplicação em uma grade Globus pode ser descrita conforme o que segue:

1. O usuário envia uma solicitação de simulação interativa, envolvendo 10.000 entidades, para um escalonador de aplicação especializado em simulação interativa distribuída;

2. O escalonador converte a solicitação em outra mais específica, descrevendo a necessidade do usuário em termos de ciclos, memória e latência de comunicação; A solicitação é então enviada a um escalonador de aplicação especializado em MPPs - Processadores Maciçamente Paralelos. Esse escalonador consulta o MDS para descobrir quais MPPs, dentre aqueles que o usuário tem acesso, são os melhores para utilizar no momento;
3. O escalonador especializado em MPPs faz a partição do trabalho entre os MPPs escolhidos e envia a solicitação mais refinada para o *co-alocador*; e
4. O *co-alocador* garante que as tarefas submetidas aos distintos MPPs comecem a executar simultaneamente.

O GRAM não é escalonador de *job*s. Ele fornece protocolo único para comunicação com diferentes escalonadores. É um dos principais serviços do Globus Toolkit, permitindo que o Globus submeta tarefas através de outros gerenciadores de *jobs*, como [37]: Condor, PBS, Fork e Torque PBS. Os gerenciadores de *jobs* Condor, PBS, Fork e Torque PBS estão sendo explicados em seguida.

Condor

Condor é um sistema especializado de gerenciamento de cargas, fornecendo mecanismos de enfileiramento e priorização de aplicações, políticas de escalonamento e monitoração de recursos. Usuários do Condor têm suas aplicações escalonadas, monitoradas e com valor de retorno informado automaticamente.

Condor transfere arquivos de forma transparente ao usuário. Se uma tarefa estiver em execução em uma determinada máquina e esta for requisitada pelo usuário local, Condor é capaz de realizar um *checkpoint* no processo e migrá-lo para outra máquina sob seu domínio. O usuário não perde a autonomia sobre o seu recurso.

O escalonamento de processos no Condor utiliza um mecanismo denominado *matchmaking* que decide como, onde e quando será executada uma determinada tarefa.

Fork

Fork é utilizado pelo Sistema Unix e cria uma réplica do processo solicitante e todo o espaço de memória do processo é replicado. É considerado um gerenciador de *jobs* limitado.

PBS

O PBS consiste em um sistema de gerenciamento de recursos baseado em filas e capaz de atuar em redes Unix multiplataforma [32]. O PBS recebe requisições de aplicações, as coloca em execução, realiza o monitoramento das mesmas e entrega os resultados ao submissor. Algumas das principais características do PBS são:

- Balanceamento de Carga - o escalonador provê diversas maneiras de distribuir a carga de trabalho entre as máquinas disponíveis.
- Transferência Automática de Arquivos - usuários PBS podem especificar os arquivos necessários para uma execução remota. O PBS transfere esses arquivos para a máquina executora e, somente após o término da transferência, é que a aplicação tem início;
- Interdependência entre Comandos - o PBS permite a definição de dependência entre comandos, incluindo ordenação de execuções, sincronização e execução condicionada a sucesso ou falha de um comando anterior;
- Autorização de Acesso - é possível permitir ou negar acesso a determinadas máquinas ou usuários;
- Contabilidade - o PBS mantém arquivos detalhados de descrição das atividades realizadas por usuários;
- Ambiente Gráfico - interface gráfica para a submissão, consulta e enfileiramento de comandos; e
- Interface de Programação - disponibilização de uma API para escrever novos comandos, integrar as funcionalidades do PBS nos códigos das aplicações ou implementar novas políticas de escalonamento.

Torque PBS

O Torque *(Terascale Open-Source Resource and Queue Manager)* é um gerenciador de recursos de código aberto baseado previamente na Versão 2.3.12, do OpenPBS. Com mais de 1200 pontos de modificação no código do OpenPBS, o Torque incorporou significativas melhorias ao PBS com contribuições de importantes organizações da área de Computação de Alto Desempenho, tais como NCSA (*National Center for Supercomputing Applications*), OSC (*Ohio Supercomputer Center*), USC (*University of Southern California*) e TeraGrid.

Algumas das características inseridas ao OpenPBS, pelo Torque, estão nas áreas de:

- Tolerância a Falhas - foram adicionadas novas condições, detecções e tratamento de falhas;
- Interface de Escalonamento - adição de novas e mais acuradas informações para consulta a respeito dos recursos. Também, foram adicionadas interfaces para um maior controle do comportamento das aplicações. Além disso, houve a inclusão de permissão para a coleta de estatísticas a respeito das aplicações já executadas;
- Escalabilidade - melhorias significativas no servidor para a implementação do modelo MOM (*Message-Oriented Middleware*) de comunicação. O gerenciador tornou-se escalável para *cluster*s de até 2500 processadores, podendo servir aplicações maiores (que utilizem até 2000 processadores) e dando suporte a mensagens maiores do servidor.

GridFTP – Grid File Transfer Protocol

O Protocolo *GridFTP* [17], estende o Protocolo FTP para torná-lo adequado para as necessidades da computação em grade. A extensão se deve à ampla utilização e o suporte dado aos servidores de dados. Têm características como autenticação GSI; transferência em paralelo com várias conexões *TCP - Transmisson Control Protocol* entre fonte e destino; transferência com conexões TCP entre várias fontes e um destino, ou vice-versa; controle manual dos *buffers* TCP, usado para afinamento de desempenho; e instrumentação embutida.

RFT - Reliable File Transfer

O RFT [17] é uma interface de serviço baseada em protocolos de *web services.* O RFT é um mecanismo que faz a persistência do estado das transferências em armazenamento confiável, utilizando o banco de dados PostgreSQL, funcionando como um escalonador de *jobs* para transferência de arquivos.

RLS - Replica Location Service

O RLS [17] é uma ferramenta que fornece o acompanhamento de réplicas de arquivos nas grades. RLS é um registro distribuído, o que significa que pode consistir de vários servidores em diferentes locais. Ao distribuir os registros RLS, aumenta-se a escala global do sistema e armazena mapeamentos mais do que seria possível em um catálogo

único e centralizado. Também evita a criação de um ponto único de falha no sistema de gestão da rede de dados. O RLS também pode ser implementado como um único servidor centralizado.

No RLS, o nome de um arquivo lógico é um identificador único para o conteúdo do arquivo. O nome do arquivo físico é a localização de uma cópia do arquivo em um sistema de armazenamento.

O trabalho do RLS é manter associações, ou mapeamentos, entre nomes de arquivos lógicos e um ou mais nomes de arquivos físicos de réplicas, conforme Figura 7. Um usuário pode fornecer um nome de arquivo lógico para um servidor RLS e pedir para todos os inscritos, os nomes de arquivos físicos de réplicas. O usuário também pode consultar um servidor RLS para encontrar o nome de arquivo lógico associado a uma localização de um determinado arquivo físico.

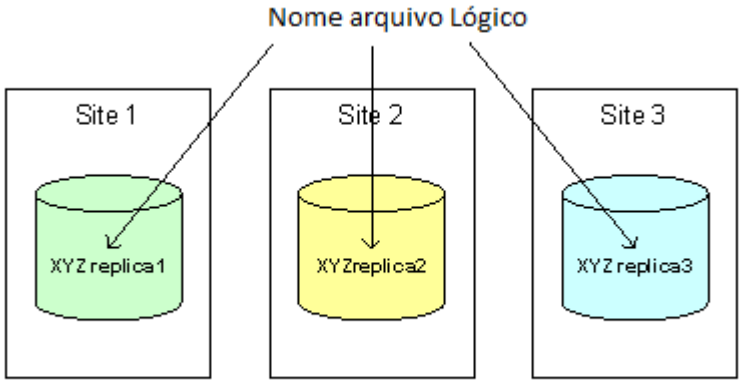

Figura 7. Funcionamento RLS [17]

RLS permite aos usuários associar atributos ou informações descritivas (como tamanho ou *checksum*) com nomes de arquivos lógicos ou físicos que estão registrados no catálogo. Os usuários, também, podem consultar RLS com base nesses atributos.0

Grid-mapfile

O Arquivo Grid-Mapfile [17] possue o nome dos usuários que têm acesso a um serviço. O arquivo também mapeia cada nome distinto para uma conta de usuário. O formato do arquivo é muito simples. Uma linha para cada usuário que tem permissão de acesso. Cada linha tem dois campos separados por espaços: o nome distinto e a conta de usuário.

OGSA- DAI

O OGSA-DAI [31] é um *framewo*rk que permite executar atividades que envolvem acesso e alterações em dados que estão distribuídos em bases de dados, em locais diferentes e em vários formatos.

Ele consiste de um executor de fluxo de trabalho e um processador de consulta distribuída que permite em uma única consulta referenciar tabelas distribuídas em vários bancos de dados. Automaticamente o processador analisa a consulta e especifica um plano de consulta para obter dados necessários de cada consulta.

O OGSA-DAI fornece os resultados de acordo com a recuperação de cada banco, ou seja, enquanto outros dados da mesma pesquisa ainda estão sendo recuperados, os resultados já recuperados estão sendo apresentados, proporcionando tempos de execução mais eficientes e despesas gerais de memória reduzidas.

2.7. Considerações Finais

A Computação em grade propicia a redução de custos e tempo, aumento de produtividade, compartilhamento de recursos e de informações. Correspondem a soluções para aplicações que precisam de uma grande capacidade de cálculos e/ou enormes quantidades de dados transmitidos de um lado para o outro e/ou ambos. Porém, alguns desafios são: segurança, tolerância a falhas, escalonamento e gerenciamento.

A garantia da segurança em grades não é uma tarefa fácil, devido à quantidade de nós se comunicando de diversas localidades. É necessário garantir a privacidade dos usuários, integridade dos dados e confidencialidade das informações. A utilização de mecanismos eficientes de verificação e identificação de usuários e recursos é necessária.

Os nós são suscetíveis a falhas ou interrupções a qualquer momento. A garantia da não interrupção de uma tarefa é quase impossível. Portanto, existe necessidade de garantir o início de uma tarefa do ponto onde foi interrompida e que seja executada com consistência.

As grades podem enfrentar diversos problemas com escalonamento. A garantia da transparência de acesso aos recursos computacionais e de dados, além de possibilitar a execução de uma tarefa com melhor desempenho, sem muitas informações, não é uma tarefa fácil. Isso porque, se deve decidir qual serviço de execução é o melhor para a aplicação/recurso que usa a grade e, em paralelo, proporcionar o melhor desempenho da aplicação e utilização da melhor forma o recurso.

O gerenciamento da demanda de recursos e a troca de dados não é uma tarefa trivial. O gerenciador tem que ser capaz de administrar os componentes operacionais essenciais na grade, tais como políticas, processos, equipamentos e dados. O gerenciador é responsável pela priorização e reserva de recursos.

Consultas e acessos aos dados em uma grade exige uma padronização, pois dependendo da complexidade da consulta e quantidade de dados armazenados, a consulta demorará algumas horas. A utilização de um padrão de metadados em uma grade de dados tem como objetivo maior rapidez em resultados de consultas e interoperabilidade dos dados.

O escopo da Globus é genérico, comparado com outras ferramentas de grades, sendo a solução mais difundida. Um dos benefícios é que seus serviços podem ser utilizados em conjunto ou totalmente independentes, conforme a necessidade/preferência do usuário.

CAPÍTULO 3

3. ARQUITETURA DE GRADE DE DADOS PARA IMAGENS FITS

Neste capítulo é apresentada a arquitetura de grade de dados para imagens FITS, com o Padrão de Metadados MetafitsGrid.

3.1. Arquitetura Proposta e sua Instanciação

A arquitetura é dividida em 4 camadas (Figura 8): Camadas de Metadados, de Aplicação, Operacional e de Hardware.

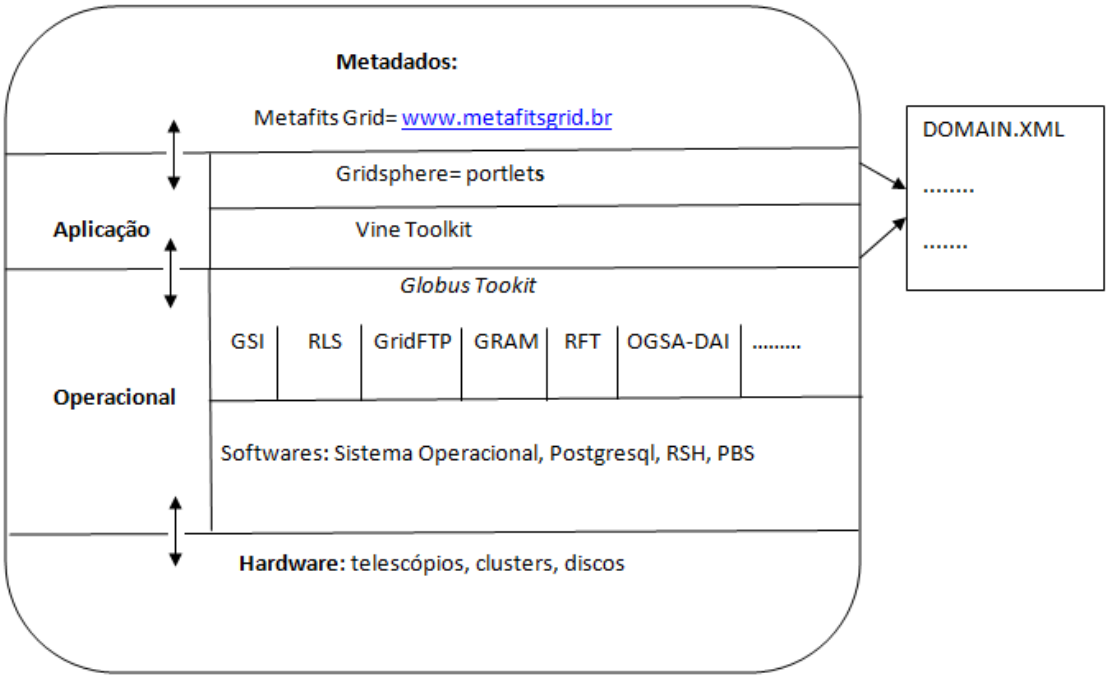

Figura 8. Visão geral – Arquitetura de Grade de Dados para Imagens FITS

A camada de Metadados permite aos usuários entenderem o significado dos dados e o contexto da aplicação. Permite ao administrador mapear os dados de acordo com um esquema de classificação visando criar uma visão das informações da aplicação.

A Camada de Aplicação tem as funcionalidades necessárias à adaptação dos processos de aplicação ao ambiente de comunicação. A camada de aplicação é estruturada para permitir a flexibilidade das funções e de forma, para se determinar os requisitos de comunicação de cada aplicação distribuída.

A camada operacional é responsável pelos softwares instalados para o funcionamento da grade. Essa camada corresponde ao núcleo da grade de dados, com configuração de aspectos de segurança, escalonadores e gerenciadores de réplicas. Esta camada faz a comunição com os recursos de hardware e fornece informações para a camada de aplicação (permissões de acesso, por exemplo).

A camada de hardware é composta pelos computadores pessoais, *cluster*s e quaisquer outros recursos de hardware conectados à grade de dados.

Nas seções seguintes é instanciada a arquitetura proposta, com a apresentação do Padrão MetaFitsGrid e softwares específicos.

3.1.1. Camada de Metadados

Na Camada de Metadados foi o definido o Padrão MetafitsGrid para descrever imagens FITS, em ambientes de Grades, o qual estende o Padrão MetaFits. Os metadados foram divididos em 4 grupos, sendo definidos 34 metadados (Figura 9).

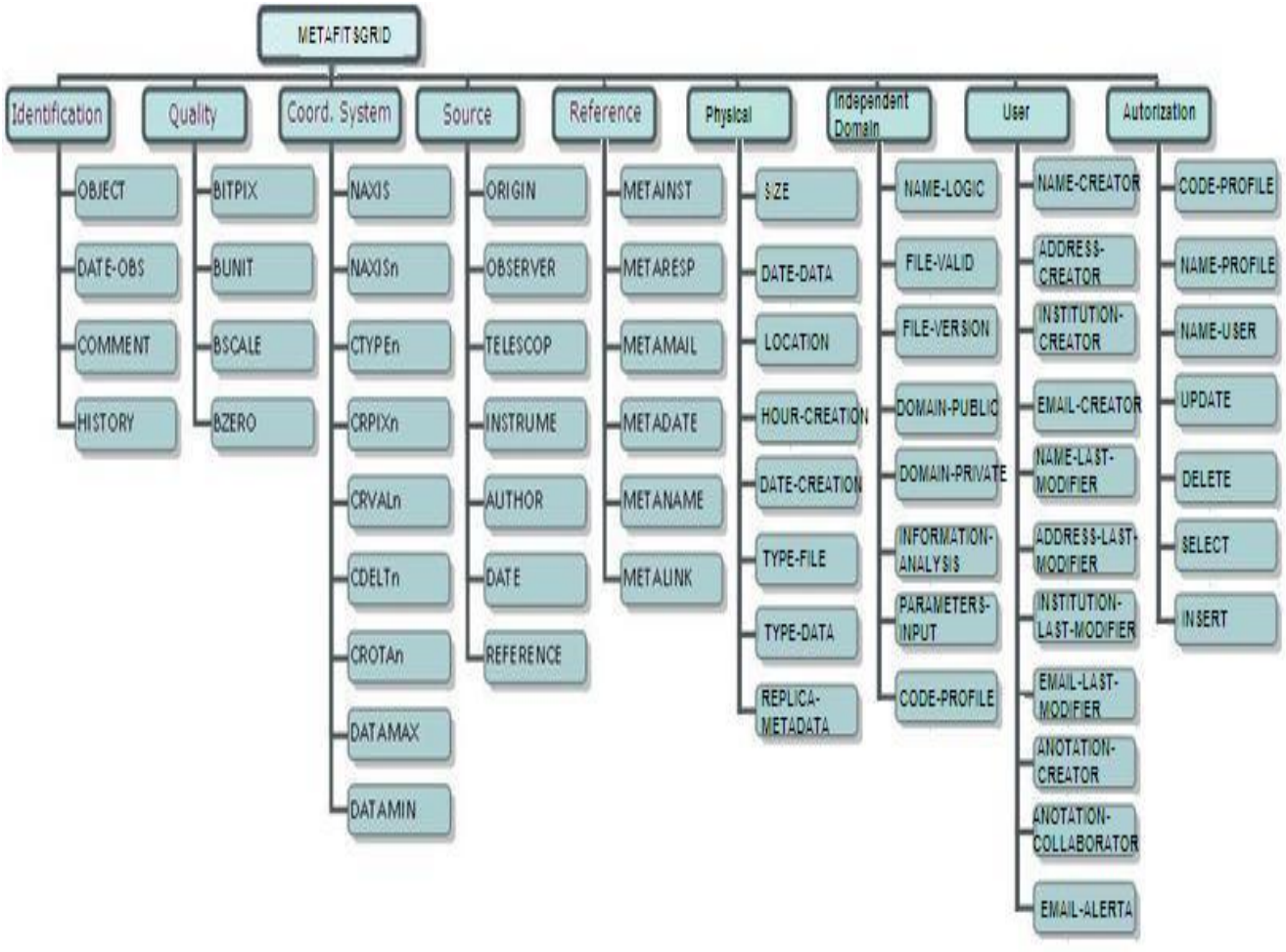

Figura 9. Organograma do Padrão MetafitsGrid

Metadados físicos: incluem informações sobre características físicas dos dados. No caso do *Type-Data*, cada sistema gerenciador de dados tem sua própria definição para tipo de dados e, estas definições não são únicas. Como exemplo, em *MySQL* e Oracle, existem dois diferentes nomes para o mesmo tipo de dados, *char* e *varchar*.

O tipo de metadados *Replica-Metadata* registra a ligação entre o arquivo lógico de dados e a réplica de um ou mais arquivos físicos, podendo estar armazenados em sistemas de arquivo ou base de dados diferentes. Este tipo de metadados é frequentemente usado pelo sistema gerenciador de metadados.

Tabela 6: Metadados do Padrão MetafitsGrid para a Categoria Metadados Físico

Metadados	**Tipo de Dados**	**Descrição**
SIZE	String	Tamanho do objeto
DATE-DATA	Date	Data de inclusão do objeto
LOCATION	String	Localização do objeto, endereço URI (*Uniform Resource Identifier*), fornecido pelo serviço RLS
HOUR-CREATION	Hour	Hora criação objeto
DATE-CREATION	Date	Data criação objeto
TYPE-FILE	String	Tipo do arquivo (.txt, . bin)
TYPE-DATA	String	Tipo de dados natural
REPLICA-METADATA	String	Descreve ligações entre ligações entre o arquivo lógico e suas réplicas de um ou mais físicos

Metadados de domínio independente: são metadados que se aplicam a qualquer item de dados. Nesta categoria, os metadados de proveniência estão especificados, os quais descrevem transformações ocorridas na grade.

O *File-Valid* é um tipo de metadados válido e indica se um item de dados é atualmente válido, permitindo rapidamente invalidar arquivos lógicos. Se existirem várias versões especificadas, tanto o nome de arquivo lógico e o número de versão devem ser fornecidos para permitir que o gerenciador de metadados identifique o item de dados desejado. Isso é determinado pelo tipo de metadados *File-Version*.

Tabela 7: Metadados do Padrão MetafitsGrid para a Categoria Metadados de Domínio

Metadados	Tipo de Dados	Descrição
NAME –LOGIC	String	Nome Lógico, correspondendo a um nome único que referencia um tipo de metadados
FILE-VALID	String	Identifica se o arquivo é válido ou não (T=True or F=False)
FILE-VERSION	String	Identifica a versão do arquivo lógico
DOMAIN-PUBLIC	String	Especifica se o tipo de metadados pode ser visualizado por todos os membros da comunidade
DOMAIN – PRIVATE	String	Especifica se o tipo de metadados não pode ser visualizado por todos os membros da comunidade
INFORMATION – ANALYSIS	String	Informações sobre análises executadas
PARAMETERS INPUT	String	Parâmetros de entrada utilizados
CODE-PROFILE	String	Código do perfil associado ao dado, caso o perfil seja do tipo privado

Independente

Metadados inerentes aos Usuários: incluem informações referentes aos usuários.

Tabela 8: Metadados do Padrão MetafitsGrid para a Categoria Metadados Inerentes aos Usuários

Metadados	Tipo de Dados	Descrição
NAME-CREATOR	String	Nome criador
ADDRESS-CREATOR	String	Endereço criador
INSTITUTION CREATOR	String	Instituição criador
EMAIL-CREATOR	String	E-mail criador
NAME-LAST-MODIFIER	String	Nome último modificador
ADDRESS- LAST-MODIFIER	String	Endereço último modificador
INSTITUTION LAST – MODIFIER	String	Instituição último modificador
EMAIL-LAST–MODIFIER	String	E-mail último modificador
ANOTATION-CREATOR	String	Anotação Criador
ANOTATION-COLLABORATOR	String	Anotação membro da comunidade
EMAIL-ALERTA	String	e-mail do responsável quando um registro de domínio privado for acessado

Metadados de autorização: são metadados que especificam os privilégios de acesso em arquivos lógicos.

Tabela 9: Metadados do Padrão MetafitsGrid para a Categoria Metadados de Autorização

Metadados	Tipo de Dados	Descrição
CODE-PROFILE	Int	Código Perfil
NAME-PROFILE	String	Nome Perfil
NAME-USER	String	Nome usuários que estão vinculados ao perfil
UPDATE	String	Especifica se o perfil pode alterar um metadado
DELETE	String	Especifica se o perfil pode excluir um metadado
SELECT	String	Especifica se o perfil pode consultar um metadado
INSERT	String	Especifica se o perfil pode incluir um metadado

O principal objetivo da camada de metadados é possuir padrão para descrição de imagens FITS, na grade, facilitando o acesso aos dados, através de consultas.

3.1.2. Camada de Aplicação

Esta camada apresenta uma interface amigável para o usuário final da grade de dados. A camada de Aplicação pode ser instanciada através de dois componentes, o *Gridsphere* e *Vine Toolkit.* O Gridsphere se comunica diretamente com a camada de Metadados para consultar as descrições realizadas através do padrão de metadados proposto e o Vine Toolkit se comunica diretamente com o *Middleware Globus.* A comunicação entre o middleware Globus e o *framework* Gridsphere é feita através do Vine *Toolkit.*

Vine Toolkit

Vine é um sistema modular e uma biblioteca Java extensível, oferecendo aos desenvolvedores APIs para aplicações de grade. O Vine dá suporte a uma ampla gama de *middleware* e serviços de terceiros, sendo um *framewo*rk *open source*.

Na arquitetura proposta, o Vine é responsável pela comunicação entre o *middleware* Globus Toolkit e a ferramenta para criação de Portais Gridsphere. O arquivo que faz tal comunicação é o Domain.xml. Um exemplo básico do arquivo Domain.xml é o que segue.

```
<domain name="test" label="Test domain" description="Vine Test domain">
   <!-- Portlet authentication -->
  <authenticationModule key="PortletAuthModule"/>
  <!-- Role resource -->
 <vineRoleResource/>
  <!-- Portal -->
 <hostResource name="portal"
     hostname="localhost"
     label="Portal"
     description="Portal">
     <!-- Portal file system -->
    <portalFileSystem label="Portal File System" description="Portal File System"/>
    <accountResource name="AccountManager"
        label="Account"
       description="Account Manager"/>
 </hostResource>
</domain>
```

Onde,

- domain: corresponde ao conceito semelhante a *namespace*, para os casos com múltiplos domínios.
- Portlet authentication: é responsável pela autenticação do usuário.
- vineRoleResource**:** é o recurso obrigatório exigido pelo sistema de autorização Vine.
- hostResource: é o recurso mais comum utilizado. É a máquina host física com alguns serviços instalados nele.
- accountResource: é responsável pela gestão de usuários.

No Anexo A está a configuração do Domain.xml utilizado para fazer a comunicação entre o Globus e o *framework* Gridsphere.

Gridsphere

O Gridsphere é um projeto *open source* e permite a criação de um *framewo*rk para o desenvolvimento de portais de grades baseado no conceito de *portlet*s. Desenvolvedores tem a possibilidade de criar aplicações que possam ser executadas e administradas dentro do Gridsphere. No Anexo A são apresentados os *portlet*s criados para implementar o padrão de metadados criado para imagens FITS, em ambiente de grade.

3.1.3. Camada Operacional

A camada operacional é formada pelos softwares que devem ser instalados, ou seja, são pré-requisitos para que a grade funcione de maneira correta e o *middleware* Globus Toolkit e seus componentes correspondentes fazem parte da instanciação desta camada.

Softwares

Um dos pré-requisitos para que o *middleware* Globus Toolkit funcione corretamente é a instalação de um escalonador para gerenciar os *jobs*. Gerenciadores de *jobs* tem como objetivo apresentar mecanismos de descoberta, seleção e monitoramento de recursos. Alguns escalonadores disponíveis são: Condor, Fork, PBS e Torque PBS. O escalonador proposto e instalado na arquitetura é o Torque PBS, devido as suas condições de tolerância a falhas, envolvendo detecções e tratamento. Além, dos novos recursos de interface de escalonamento para controle do comportamento das aplicações.

Alguns dos softwares que são pré-requisitos para a correta instalação e funcionamento do Globus são: zlib, gcc, g++, tar, sed, make, Perl, sudo, postgresql, java e Apache Ant.

Globus Toolkit

O *Globus Toolkit* é uma ferramenta (*middleware*) que facilita a utilização da tecnologia de grade. Além do Globus existem outros *middleware*s, tais como: Glite [13], Legion [21] e EasyGrid [7], porém o foco foi o Globus pois o mesmo representa um grande avanço na implementação de *web services* de forma integrada e padronizada [2]. O Globus trabalha com o conceito de componentes, o GSI, GRIDFTP, GRAM, RFT, RLS, Grid-Mapfile e OGSA-DAI.

GSI (Globus Security Infrastructure)

O componente GSI trabalha a segurança dentro da grade de dados. O GSI utiliza criptografia de chave pública. A motivação para utilização do GSI é a comunicação segura (autenticada e confidencial), suporte de segurança através das fronteiras organizacionais e por trabalhar com conceitos de certificados. O certificado inclui 4 (quatro) componentes:

- Um assunto que identifica a pessoa ou objeto que o certificado representa;

- A chave pública pertencente ao assunto;
- A identidade da unidade certificadora (CA) que assina o certificado para atestar que a chave pública e a identidade pertencem ao assunto; e
- Os certificados GSI são codificados no formato de certificados X.509, formato de dados padrão para os certificados estabelecidos pelo IETF (*Internet Engineering Task Force*).

O GSI oferece uma extensão do Protocolo Padrão SSL que reduz o número de vezes que o usuário deve digitar sua senha, necessidade esta exigida pelo Gridsphere que é resolvida através da criação de um proxy.

Um proxy é constituído de um novo certificado (com uma nova chave pública nele) e uma nova chave privada. O novo certificado contém a identidade do proprietário, ligeiramente modificado para indicar que ele é um proxy. Proxys tem vida útil limitada.

GRIDFTP (Grid File Transfer Protocol)

O serviço do GridFTP é utilizado para realizar transferências de arquivos em ambientes de grade, proporcionando maior segurança e confiabilidade dos dados no momento das transferências. O Globus Toolkit fornece a implementação mais comumente usada do protocolo. O Globus provê uma implementação de servidor chamado *globus-gridftp-server*, um script de linha de comando programável, e um cliente chamado *globus-url-copy*, conjunto de bibliotecas personalizadas de desenvolvimento para clientes.

RFT (Reliable File Transfer)

O RFT permite consultar o estado das transferências. Seu servidor está disponível através de um serviço no *container* do Globus Toolkit e o mesmo deve estar a um banco de dados de terceiro, como PostgreSQL, por exemplo.

RLS (Replica Location Service)

O RLS fornece o acompanhamento de réplicas de arquivos nas grades. RLS é um registro simples que mantém o controle do armazenamento físico. Usuários ou serviços registram os arquivos em RLS quando os mesmos são criados e após os usuários consultam servidores RLS para encontrar estas réplicas.

Grid-mapfile

O Globus Toolkit 4 requer um mapeamento entre todos os usuários do ambiente grade e de seus domínios, afim de prover um mapa de todos os usuários que tenham acesso ao ambiente, este mapeamento é fornecido pelo arquivo mapfile.

GRAM* (*Grid Resource and Management)

O GRAM é um projeto Globus, que produz tecnologias que permitem aos usuários localizar, submeter, monitorar e cancelar *jobs* em grade de dados.

OGSA-DAI

O OGSA-DAI é uma solução para acesso e gerenciamento de dados distribuídos, que permite que recursos de dados, como por exemplo: banco de dados relacionais ou XML, serem acessados via *web services* na *web* ou em grades de dados, sendo que os dados podem ser consultados, transformados e combinados de várias maneiras.

3.1.4. Camada de Hardware

A camada de hardware é composta pelos computadores pessoais, *cluster*s e qualquer outro recurso de hardware conectado à grade de dados.

3.2. Considerações Finais

A arquitetura foi criada devido aos benefícios oferecidos pela computação em grades. Esses benefícios correspondem a pré-requisitos para cientistas da área científica, que necessitam de grande poder computacional e de armazenamento e a possibilidade de compartilhar experimentos.

A arquitetura criada possui uma camada de metadados facilitando o acesso aos dados, com um padrão de metadados definido. Tal padrão, apesar de ser específico para imagens FITS, possui metadados que podem ser utilizados para ambientes genéricos de grades.

A arquitetura proposta tem como maior benefício à simplicidade para criação de uma grade de dados, sendo que a mesma possui a camada de metadados. Tal camada tem como objetivo facilitar o posterior acesso aos dados, pois a mesma propõe um padrão de metadados para imagens FITS, sendo assim os dados devem ser cadastrados de acordo com o padrão especificado, sendo acessados e recuperados posteriormente com maior rapidez e qualidade.

A Camada de Aplicação tem as funcionalidades necessárias para ambiente de comunicação, permitindo flexibilidade das funções e de forma, para se determinar os requisitos de comunicação de cada aplicação distribuída na grade.

A camada operacional é responsável pelos softwares instalados para o funcionamento da grade, correspondendo ao seu núcleo, com configuração de aspectos de segurança, escalonadores e gerenciadores de réplicas. Esta camada faz a comunição com a camada de hardware e fornece informações para a camada de aplicação.

CAPÍTULO 4

4. TRABALHOS RELACIONADOS

Neste capítulo são apresentados trabalhos relacionados, enfatizando grades de dados e padrões de metadados. Por fim uma análise comparativa dos trabalhos relacionados com a arquitetura proposta é apresentada.

4.1. AstroGrid-D

O AstroGrid-D [8] foi um esforço de astrofísicos e cientistas para explorar tecnologias de grade para aplicações cientificas, ciência astronômica para tecnologias de grades. É baseado no GT4 - *Globus Toolkit middleware*. O projeto envolve institutos de astronomia tais como *AIP – American Institute of Physics*, *AEI – American Enterprise Institute*, *MPA - Max-Planck Institute for Astrophysics*, MPE - *Max-Planck-Institut for Extraterrestrische Physik* e *ZAH - Zentrum for Astronomie.*

O principal objetivo deste projeto é estabelecer um trabalho colaborativo para ambientes de astronomia, fornecendo ferramentas de software aos usuários, permitindo fácil acesso a computação e facilidades de armazenamento.

O Arquitetura AstroGrid (Figura 10) provê um conjunto de integrações: computação, dados e recursos de hardware especiais, como telescópios robóticos, de modo que eles podem ser acessados como qualquer outro nó de computação.

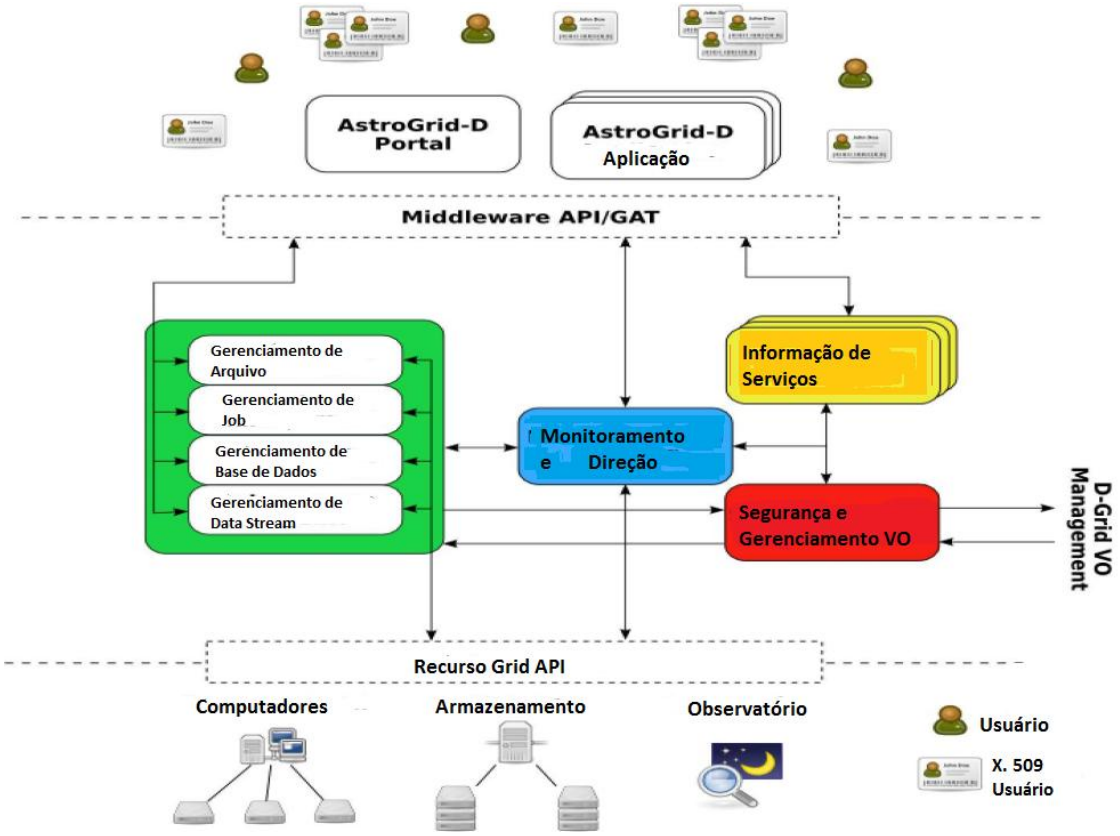

Figura 10. Arquitetura do AstroGrid [8]

O AstroGrid é composto por cerca de 20 componentes: *cluster*s, estações de trabalho, servidores de armazenamento de dados, bem como um servidor telescópio. Ele pode ser acessado via interface como segue.

- GAT (*Grid Application Toolkit*): uma API que fornece uma alternativa de interface, fornecendo uma maneira transparente de acessar a grade.
- GridSphere: permite que desenvolvedores rapidamente desenvolvam *portlets* para a grade.

O projeto AstroGrid define 4 (quatro) tipos de metadados:

- Metadados de recursos: descreve propriedades dos recursos compartilhados, por exemplo, para um telescópio: abertura, filtros, capacidade;
- Metadados de estado de atividade: registra o estado atual e registros de atividades na grade tais como: a localização de determinado recurso, informações de transferências de arquivo. Por exemplo, nome de usuário e nome telescópio;
- Metadados de aplicação: descreve o programa e seus parâmetros de entrada;

- Metadados científicos: inclui informações sobre a proveniência dos conjuntos de dados que são usados. Por exemplo: projeto de ciências, tipo de dados (imagem, tabela) e referências.

4.2. BD-GRID

O BD-Grid [42] é uma grade de dados baseada na Arquitetura GD-Grid, a qual provê uma plataforma de dados, integrando recursos de dados biológicos distribuídos e heterogêneos, permitindo aos usuários autorizados acesso fácil por meio de uma série de protocolos e interfaces.

Metadados são classificados no domínio BD-Grid, onde os usuários podem encontrar dados de seu interesse com facilidade, utilizando um Portal *Web*. A Arquitetura BD-Grid é dividida em 3 (três) camadas (Figura 11): fornecimento, integração e acesso aos dados.

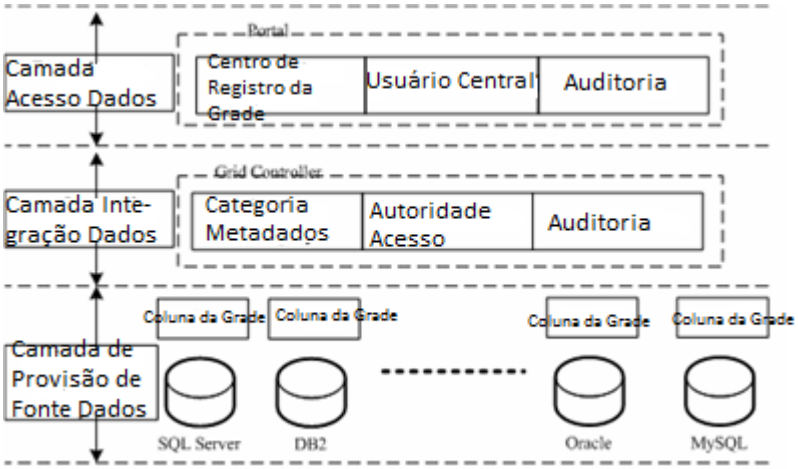

Figura 11. Arquitetura BD-Grid [42]

1. *Camada de Fornecimento dos dados:* O objetivo principal é disponibilizar os dados de forma ordenada e segura aos usuários. Essa camada está ligada à camada de acesso aos dados.
2. *Camada de Integração de Dados:* Quando um recurso é integrado, os mesmos podem ser de diferentes tipos de bancos de dados, podendo existir a necessidade de recuperação de dados de múltiplas fontes diferentes ou integrar dados de diferentes sistemas. A camada de integração de dados é projetada para resolver esse problema, integrando recursos de dados fornecidos pela camada de

fornecimento de dados de recursos. Através dessa camada, usuários recuperam dados de múltiplas fontes, de diferentes tipos com facilidade e segurança. Existem três módulos principais: categoria de metadados, auditoria e autoridade de acesso.

- *Categoria de metadados:* Cada categoria de metadados é mapeada para um Catálogo no BD-Grid. Os catálogos de entrada são organizados em uma hierarquia, como uma estrutura de diretório, mas com a diferença de que os dados do catálogo são independentes de localização. Ou seja, usuários não precisam saber onde os dados estão fisicamente armazenados para encontrá-los e usá-los. Proprietários dos dados não criam réplicas dos seus dados, em vez disso, criam um *link* a partir da entrada de catálogo de dados existente, enviando o *link* de parâmetros para o módulo de autoridade de acesso, tais como: tipo de banco de dados, nome do banco de dados, nome de usuário, senha, certificado simbólico, entre outros.
- *Auditoria:* usada para ajudar a garantir a segurança dos recursos. Administradores podem usar o *log4j* para configurar o *log* de todos os eventos do acesso que executam ou modificam itens nos metadados do catálogo. Entretanto, nesta plataforma, é possível emitir alertas (como e-mails) para notificar os administradores quando itens particularmente sensíveis são acessados.
- *Autoridade de acesso:* esse módulo trabalha com autoridade de acesso, com a autenticação de *login* de acesso ao BD-Grid.

3. *Camada de Acesso aos Dados:* inclui uma interface *web* simples e amigável, uma central de registro da grade, um usuário central e um módulo de auditoria. O Portal permite aos usuários executarem tarefas comuns. A concepção de domínio é empregada nesse módulo e os usuários podem encontrar facilmente seus dados ou publicar seus metadados para o catálogo no BD-Grid. Metadados são adicionados livremente, ou seja, não existe um critério de consistência de inserção dos mesmos. A ferramenta não garante a qualidade dos metadados inseridos, porém, pretende-se criar um mecanismo de acesso que pode dimensionar a confiabilidade de metadados.

4.3. Portal GridMACS

O GridMACS [3] é um portal web com o objetivo de disponibilizar uma interface para o GROMACS - *GRONingem MAchine for Chemical Simulations*, executando uma simulação dinâmica molecular num ambiente de grade. O portal foi desenvolvido de acordo com a arquitetura básica de uma grade e foi utilizado o *framewo*rk Gridsphere, o Vine Toolkit, e o Globus Toolkit.

O Portal GridMACS prove um *portlet* de interface de usuário, em um ambiente grade, para o GROMACS. GROMACS é baseado em LINUX. Devido à alta complexidade do GROMACS para não usuários LINUX, o uso do portal Gridsphere age como uma interface web onde a aplicação do GROMACS reside. A complexidade de baixo nível, de acesso aos recursos computacionais do ambiente de grade, é oculta aos usuários.

Foram utilizadas as seguintes ferramentas: Gridsphere 3.1 como *framewo*rk de portal, Vine Toolkit 1.1 como serviço de grade e o Globus 4.0.5 como *middleware* de Grade e o apache 5.5.26 como ambiente de hospedagem para o portal.

A Figura 12 apresenta uma visão simplificada da infraestrutura das camadas no portal. O *Framewo*rk Gridsphere foi utilizado para criar os *portlets*, o Vine Toolkit integrado no Gridsphere disponibiliza *portlets* que dão suporte ao Globus Toolkit.

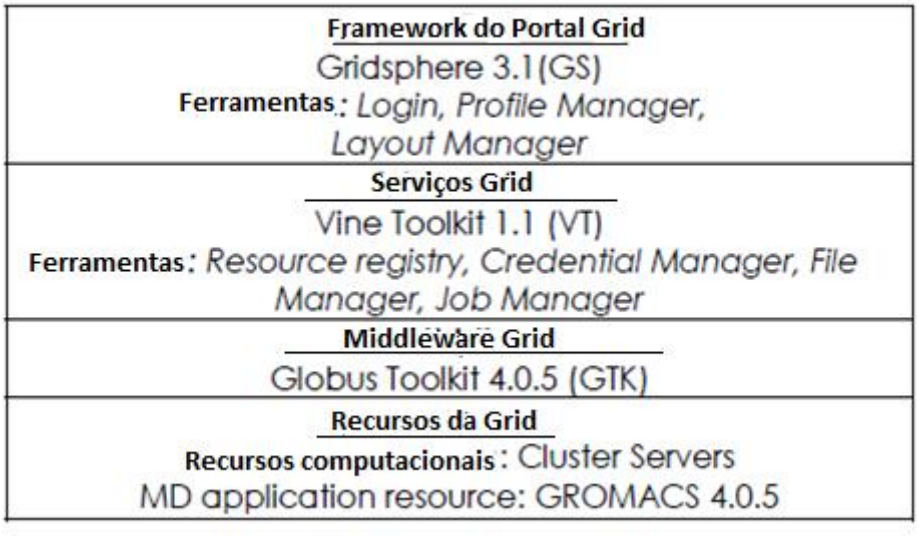

Figura 12. Arquitetura GridMACS

Desenvolvedores do GridMACS realizaram alguns testes na arquitetura criada utilizando o *benchmarck* d.*dppc* disponível no site do Gromacs. Foram enfrentados vários obstáculos, por exemplo: a versão mais nova do Gridsphere 3.1 é compatível com o Vine Toolkit 1.1 e para submeter um *job* do portal, o Globus Toolkit deve estar

instalado. Durante o desenvolvimento do projeto, o Gridsphere e o Vine Toolkit foram compatíveis apenas com o Globus na Versão 4.0.5 e o Apache Tomcat é compatível apenas na Versão 5.5.x. Outras versões mais novas, não funcionaram.

Conclui-se que o Portal GRIDMACS emprega tecnologias de grade, provendo uma plataforma para facilitar simulações GROMACS, aumentando a disponibilidade e a usabilidade.

4.4. Grades de Dados para Arquivo de Imagens Médicas

A Grade de Dados para arquivo de imagens médicas [19] tem como objetivo recuperar dados de imagens médicas e realizar análises, possuindo um sistema de armazenamento de backup e tolerância à falhas.

A grade de dados é um aglomerado de 3 (três) sites. O primeiro é o IPI - *Image Processing and Informatics Laboratory*. O segundo e o terceiro são os hospitais *Saint John's Health Center* (SJHC) e o *Healthcare Consultation Center II* (HCC II), respectivamente. Os 2 (dois) centros de saúde possuem computadores e recursos integrados de imagens radiológicas (PACS), correspondendo a sistemas de arquivos SAN (redes de serviços de área).

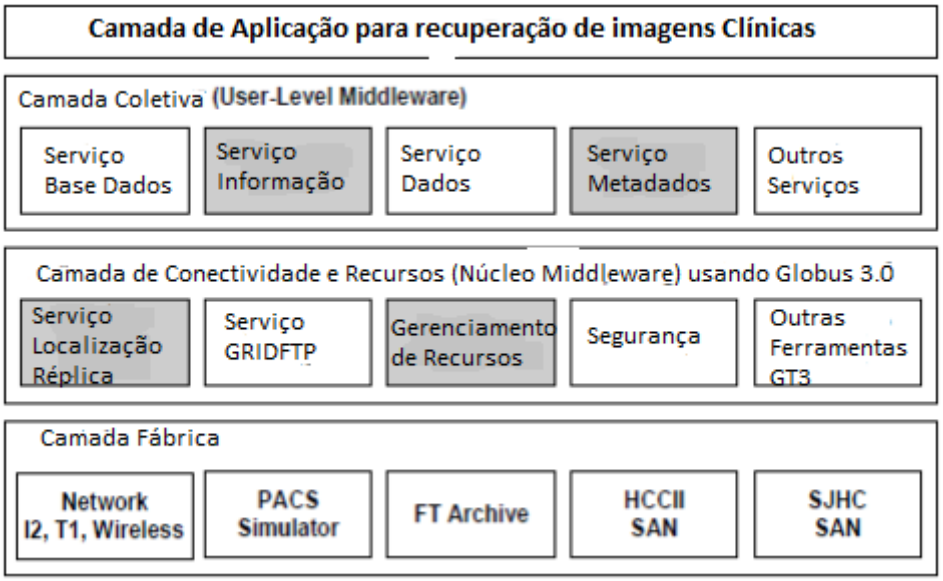

Figura 13 - Grade de Dados para arquivos de imagens médicas e análise

A grade de dados foi implementada utilizando Globus 3.0. A arquitetura da grade de dados é formada por 4 (quatro) camadas, conforme Figura 13, descritas como segue.

Camada 1 – Fábrica

Esta camada consiste do DICOM, correspondendo ao backup de arquivos de tolerância à falhas, servidores de backups de arquivos, simuladores PACS, 2 SANS (Redes de área de armazenamento) em dois locais clínicos e sistemas de comunicações de rede, incluindo LAN, Internet e banda larga WAN.

Camada 2 – Conectividade e Recursos

*E*sta camada consiste de serviços do Globus 3.0. Possui o núcleo middleware usando o Globus Toolkit 3.0.

Camada 3 – Coletiva

Esta camada consiste de serviços para interagir entre aplicações de usuários e serviços do núcleo do *middleware* ao nível do usuário, como serviços de base de dados, para encontrar os melhores bancos de dados disponíveis na grade de dados; serviços de informação, para supervisionar os serviços ativos na grade de dados; e serviços de dados, para encontrar o endereço físico dos dados lógicos, como outros serviços.

Nesta camada, o Globus 3 tem apenas serviços de informação. Todos os outros, tais como base de dados, metadados, dados e outros serviços, não estão disponíveis. Estão sendo desenvolvidos outros serviços em conjunto com outras aplicações de recuperação de dados de imagens.

Camada 4 – Camada de Aplicação da Grade de dados

Esta camada consiste de diversas aplicações, como recuperação de dados de imagens clínicas PACS.

4.5. Pesquisa do Sono e Análise de Polissonografia

A pesquisa do sono e Análise de Polissonografia [23] é uma grade de dados para experimentos relacionados à saúde, tendo como objetivo principal fazer análises de imagens relacionadas ao sono utilizando a base de dados SIESTA. A estrutura de funcionamento da grade de dados é apresentada na Figura 14.

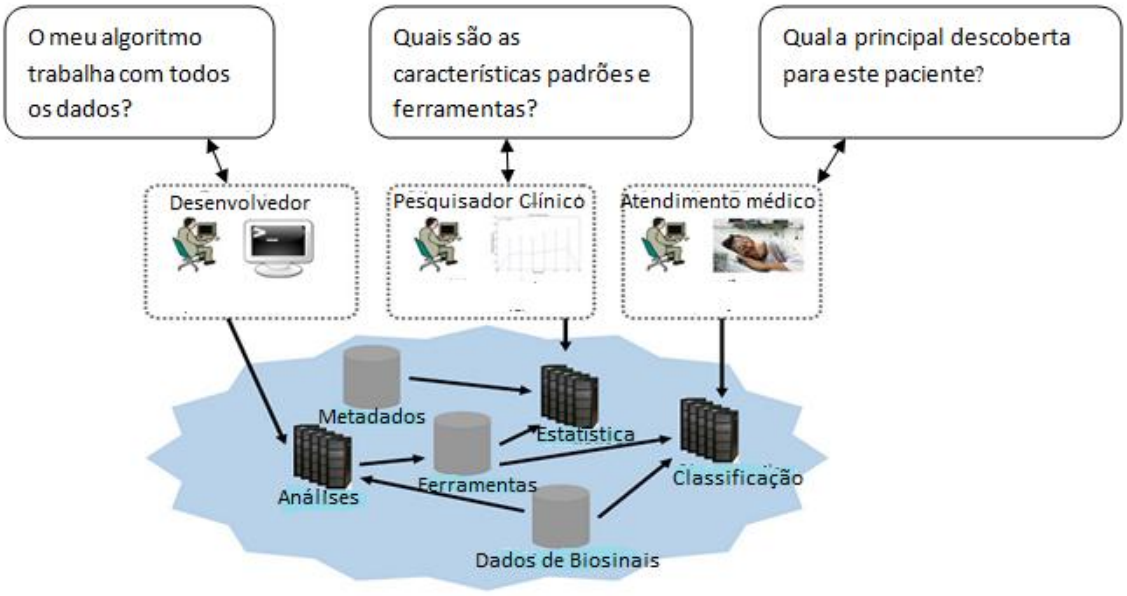

Figura 14 – Estrutura funcionamento do projeto Pesquisa do Sono e Análise de Polissonografia

A grade de dados possui 3 (três) grupos de usuários principais:

- *Desenvolvedores de software:* utilizam a base de dados para efetuar testes de algoritmos, considerando dados heterogêneos. Esses testes são feitos para verificar a robustez do algoritmo;
- *Pesquisadores médicos:* exploram a base de dados para realizar estudos clínicos, utilizando ferramentas de análises disponíveis; e
- *Médico assistente:* o objetivo dos pesquisadores é utilizar sistemas, onde computadores possam ajudar no diagnóstico de uma determinada doença. É utilizado algoritmos para encontrar relacionamentos entre biosinais e status da saúde do paciente.

A arquitetura da grade apresentada tem sua implementação na grade biomédica nacional MediGrid. O MediGrid é uma iniciativa do D-Grid, oferecendo componentes de *middleware* genéricos, tais como Globus e Glite.

A arquitetura do MediGrid esta retratada na Figura 15. Os componentes que são utilizados do D-Grid nesta arquitetura são:

- SRB (*Storage Resource Broker*);
- GWES (*Grid Workfkow Execution Service;* e
- Portal GridSphere.

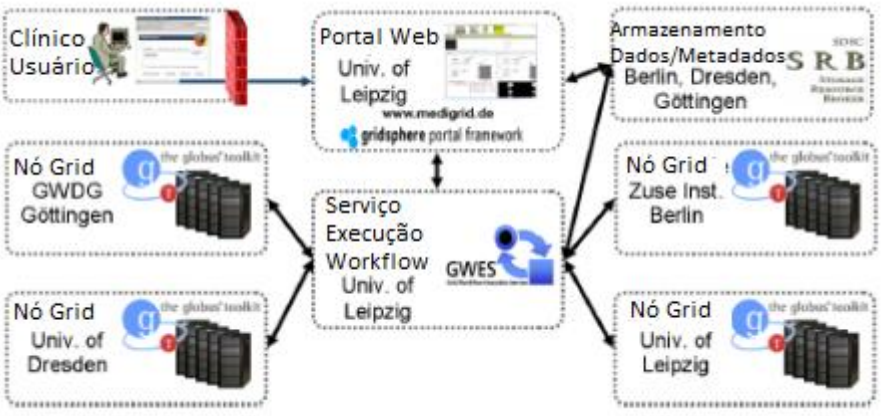

Figura 15 - Arquitetura do MediGrid

SRB - *Storage Resource Broker*

SRB é um sistema gerenciador de dados na grade, baseado na arquitetura cliente-servidor. Cada usuário tem uma pasta *home* própria no SRB, que é semelhante a um diretório *home* em um sistema de arquivo local. Direitos de acessos de usuários e grupos, como ler e escrever podem ser configurados para arquivos e diretórios, individualmente. Como SRB é amplamente utilizado em ambientes de grades, há muitas ferramentas para acessar SRB. MediGRID executa uma instalação SRB com recursos distribuídos em Berlim, Dresden e Gottingen, gerenciando cerca de 80 TB de espaço de armazenamento.

GWES - *Grid Workflow Execution Service*

GWES é um gerenciador de fluxo de trabalho designado por uma aplicação da grade. O núcleo do GWES é o *Grid Workflow Description Language* (GWorkflowDL), um padrão de rede de Petri para descrever fluxos de trabalho usando XML. GWES implementa em alto nível Redes de Petri (HLPN) para a descrição do fluxo de trabalho, usados diretamente para a transferência e modelo de armazenamento de entrada e saída de dados, bem como dados de controle.

Portal Grid

O *framewo*rk GridSphere provêm uma fonte aberta de *portlet* baseada em portal web. Os Metadados são integrados no SRB como tags de metadados definidos pelo usuário (Tabela 10).

Tabela 10: Tags de metadados definidos pelos usuários

Nr.	Metadata-key	Description
1	Record	Filename without Fileending in upper letters
2	ECG	Flag, if an ECG is recorded
3	ODI	Oxygen Desaturation Index
4	TST	Total Sleep Time [min]
5	AI	Apnea Index [per minute]
6	AHI	Apnea Hypopnea Index [per minute]
7	Age	Age [years]
8	Sex	Sex-Flag [male = 1, female = 2]
9	Status	Health Status according SIESTA convention
10	Height	Height [cm]
11	Weight*	Weight [kg]
12	Pulse*	Pulse rate [per minute]
13	BPsys*	blood pressure, systolic reading [mmHg]
14	BPdia*	blood pressure, diastolic reading [mmHg]

Os desenvolvedores apontam alguns problemas na utilização do SRB, tais como: não existe a possibilidade de alterar metadados. É necessário exclui-los e criar novamente.

Na próxima seção é apresentado um comparativo das arquiteturas descritas nesta seção com a arquitetura proposta nesta dissertação.

4.6. Análise Comparativa

Na sequencia é apresentada uma análise comparativa de trabalhos correlatos, conforme Tabela 11, com a arquitetura proposta.

Tabela 11: Análise Comparativa dos Projetos de Grade

Projeto	**Aplicação**	**Criação de Metadados**	**Padronização de Metadados**	**Catálogo de Metadados**	**Middleware**	**Interface**
Grade Proposta na dissertação	Dados científicos – Imagens FITS	Sim	Sim	Não	Globus Toolkit 4	Gridsphere e Vine Toolkit
AstroGrid	Dados Astronômicos	Metadados definidos pela comunidade AstroGrid	Sim	Sim	Globus Toolkit 4	GAT ou Gridsphere
BD-GRID	Dados Biológicos	Livre	Não	Sim	GD-Grid	Portal BD-Grid amigável

GridMACS	Dados Biológicos (Moleculares)	Não	Não	Não	Globus Toolkit 4	Gridsphere e Vine Toolkit
Medical Image	Dados Médicos	Não	Não	Não	Globus 3	Não especifica tecnologia, porém comenta que existe um portal
Sleep Research Analysis of Polysomnographies	Dados Médicos (Sono)	Sim	Sim	Sim	Baseado no MediGrid (compo-nentes do Globus e Glite)	Gridsphere

A grade de dados proposta para imagens FITS, foca na área astronômica. Sua arquitetura é dividida em camadas: Hardware, Operacional, Aplicação e Metadados. Define um padrão de metadados para imagens FITS em ambientes de grades de dados, com o objetivo de facilitar o acesso aos dados cadastrados.

Os metadados foram divididos nas categorias Identificação, Qualidade, Sistema de Coordenadas, Fonte, Referências, Físicos, Independente de Domínio e Usuários. Utiliza o *middleware* Globus Toolkit 4, Vine Toolkit e Gridsphere.

O AstroGrid constitui de grade de dados astronômica, projeto desenvolvido pela Alemanha, que tem como objetivo compartilhamento de experimentos e maior rapidez. Utiliza o *middleware* Globus. Desenvolvedores do AstroGrid criaram 4 (quatro) categorias de metadados: metadados de recursos, metadados de estado de atividade, metadados de aplicação e metadados científicos.

O BD-Grid trata de grade de dados para analisar conjunto de dados biológicos, baseado no M*iddleware* GD-Grid, e possui um módulo de categorias de metadados, onde são mapeadas para um catálogo do BD-Grid. Através do Portal do BD-Grid, o usuário pode publicar seus metadados ou encontrar seus dados. Porém, no BD-Grid não existe um mecanismo para validar metadados incluídos, ou seja, não se pode garantir que os metadados inseridos sejam confiáveis.

O GRIDMACS trata grade de dados biológicos, focada em simulações de moléculas de proteínas que tem como objetivo oferecer uma interface amigável para usuários finais. Utiliza ferramentas para construção da grade de dados: Globus Toolkit 4, Gridsphere e Vine Toolkit. O GridMACS não abordou conceitos de metadados e o projeto teve problemas para implementação da grade, devido a incompatibilidade de versões.

A Grade de Dados para arquivos de imagens médicas e análise é formada por hospitais e centros de pesquisa com o objetivo de recuperar dados de imagens médicas, tendo grande preocupação com o sistema de backup. Utiliza o *middleware* Globus Toolkit 3. Não apresenta detalhes de como foi implementado o portal e nem de padrões de metadados.

O projeto Pesquisa do Sono e Análise de Polissonografia possui uma estrutura de grade de dados de saúde focada em experimentos relacionados ao sono. Possui bem definido o conceito de grupo de usuários, a implementação da grade segue o padrão da grade *MediGrid*. O *MediGrid* utiliza componentes de *middleware*s genéricos como *Globus Toolkit* e *Glite*. No portal utilizam o *framewo*rk *Gridsphere*. Na grade de dados são abordados conceitos de metadados, sendo que usuários podem definir metadados abertamente, porém alguns problemas são encontrados, pois metadados não podem ser alterados e algumas consultas de metadados não estão funcionando.

4.7. Considerações Finais

Grades de dados tornaram-se uma tendência nas mais variadas áreas da ciência devido a possibilidade das mesmas oferecerem grande poder computacional e armazenamento e de, principalmente, oferecer a possibilidade de compartilhar os experimentos.

A maioria das grades utiliza o *middleware* Globus, devido o mesmo ser o mais completo *middleware* para grades e o que está em constantes mudanças para sua melhoria. O ambiente de grades possui uma interface de baixo nível, baseada em linha de comando, porém existe a necessidade de uma interface amigável. A maioria das interfaces é criada utilizando o *framewo*rk Gridsphere que fornece uma maneira fácil de criar portais *web* em conjunto com o Vine Toolkit.

As grades estão em constante processo de melhoria. Muitos pesquisadores enfrentam dificuldades para sua implementação, devido à incompatibilidade de versões

e uma documentação dispersa. Algumas grades de dados implementam o conceito de metadados, porém ainda são muito poucas.

A arquitetura proposta tem como benefício auxiliar profissionais que não possuem muito conhecimento de informática, construir uma grade de dados, seguindo as especificações sugeridas e instalando os componentes necessários, independente da área de conhecimento.

Foi escolhido o *middleware* Globus, o Gridsphere e o Vine Toolkit, devido os mesmos estarem em constantes melhorias e serem *open source.* Comparando com os trabalhos relacionados, os mesmos mostram arquitetura de grade de acordo com necessidade do seu ambiente.

A proposta de um padrão de metadados para imagens FITS objetivou melhorar recuperação e acesso aos dados na grade. As arquiteturas dos trabalhos correlatos não possuem um padrão de metadados definido. O Padrão MetafitsGrid pode ser utilizado em outros ambientes de grade, alterando apenas metadados que são específicos de cada área de conhecimento

CAPÍTULO 5

5. CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões desta dissertação, com as principais contribuições e a relevância das mesmas, e as perspectivas de trabalhos futuros são evidenciadas.

Grades de dados permitem o compartilhamento de dados e experimentos, necessidade dos cientistas que estão localizados em lugares geograficamente distantes. Muitos experimentos necessitam de grande poder computacional e armazenamento. Construir e utilizar uma grade de dados não é uma tarefa trivial.

Esta dissertação atingiu o objetivo fornecendo uma abordagem arquitetural de Grade de dados para imagens FITS, descrevendo as camadas de um ambiente de grade e componentes, sendo mostrado a instalação do *middleware* Globus Toolkit e a configuração de componentes.

Também, foi apresentado e implementado o Portal para apresentar o padrão de metadados para imagens FITS, o MetafitsGrid, utilizando o Gridsphere e o Vine Toolkit. Devido a utilização do Globus Toolkit, Gridsphere e Vine, a versão do Globus que estava em conformidade com o Gridsphere era a versão do Globus 4.0.5. O Gridsphere e o Vine estão passando por constantes modificações.

Esta dissertação, também, alcançou o objetivo de facilitar e ajudar na criação de um ambiente de grade de dados para imagens FITS, apresentando componentes que são essenciais e alguns softwares que podem tornar o ambiente de grade mais amigável através de portais utilizados, o Gridsphere e o Vine.

A arquitetura proposta pode ser utilizada para construir arquiteturas de grades de dados de diversas áreas de conhecimento. O padrão de metadados proposto facilita a recuperação e a consulta aos dados de imagens FITS, onde o padrão proposto pode ser reutilizado para ambientes de grades de dados de outras áreas de conhecimento, alterando apenas os metadados da área de conhecimento.

Como Trabalhos Futuros, pretende-se finalizar a comunicação entre o Globus Toolkit, Vine e Gridsphere; realizar testes com o padrão de metadados proposto armazenando e consultando imagens FITS e também automatizar o processo de inclusão dos metadados no Portal MetafitsGrid.

REFERÊNCIAS

[1] BUYYA, R; VENUGOPAL, S; RAMAMOHANARAO, K. *A Taxonomy of Data Grids for Distributed Data Sharing, Management, and Processing*. University of Melbourne, Australia. ACM Computing Surveys, Vol. 38. Article 3, 2006.

[2] CERAMI, E. *Web services essentials*. O Reilly, Vol. IV, 2005.

[3] CHIA, Elizabeth; SHAMSIR, Shahir Mohd; HUSSEIN, Azura Zeti; HASHIM, Siti Zaitom Mohd. *GridMACS Portal: A Grid Web Portal for Molecular Dynamics Simulation using GROMACS*. 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, 2010.

[4] CSDGM – *Content Standard for Digital Geospatial Metadata*. Disponível em: http://www.fgdc.gov/metadata/csdgm/. Acesso em: fev/2011.

[5] DCMI - *Dublin Core Metadata Initiative*. Disponível em: http://dublincore.org/ Acesso em: jan/2010.

[6] DUVAL, E., HODGINS, W., SUTTON, S., WEIBEL S. *Metadata Principles and Practicalities*. D-Lib Magazine 8(4). Disponível em: http://www.dlib.org/dlib/april02/weibel/04weibel.html. Acesso em: jan/2010.

[7] EasyGrid. Disponível em: http://easygrid.ic.uff.br/grid/EasyGrid.html. Acesso em: mai/2011.

[8] ENKE, H.; STEINMETZ, M.; ADORF, H. ; BECK-RATZKA A.; BREITLING, F.; BRUSEMEISTER, T.; CARLSON, A.; ENSSLIN, T.; HOGQVIST, M.; NICKELT, I.; RADKE, T.; REINEFELD, A.; REISER, A.; SCHOLL, T.; SPURZEM, R.; STEINACKER, J.; VOGES, W.; WAMBSGANß, J.; WHITE, Steve. *AstroGrid-D: Grid Technology for Astronomical Science*, 2010.

[9] FGDC, Federal Geographic Data Committee. *Content Standard for Digital Geospatial Metadata*. User Guide, 1998.

[10] FOSTER, I. *What is the Grid? A Threee Point*. Disponível em: http://dlib.cs.odu.edu/WhatIsTheGrid.pdf. Acesso em: dez/2010.

[11] FOSTER, I; KESSELMAN, C. *The Grid: Blueprint for a Future Computing Infrastructure*, 1999.

[12] FOSTER I; KESSELMAN ,C. *Globus: A metacomputing infrastructure toolkit.* International Journal of Supercomputer Applications. vol. 11, no. 2, pp. 115–128, 1997.

[13] Glite. Disponível em: http://glite.cern.ch/. Acesso em: nov/2011.

[14] GRIA. Disponível em: http://www.gria.org/documentation/5.2/manual/ developer-guide. Acesso em: nov/2011

[15] GRIDSPHERE 3.2. Disponível em: www.gridsphere.org/gridsphere/ download/ download. Acesso em: jan/2011.

[16] GRIDPP. *Grid for UK Particle Physics*. Disponível em: http://www.gridpp.ac.uk. Acesso em: mar/ 2010.

[17] GLOBUS Toolkit 4.0.5. Disponível em: http://www.globus.org/toolkit/downloads /4.0.5/. Acesso em: jan/2011.

[18] HANISCH R. J.; FARRIS A; GREISEN E. W.; PENCE W. D.; SCHLESINGER B. M.; Teuben, P. J.; THOMPSON R. W.; WARNOCK III A. *Definition of the Flexible Image Transport System (FITS).* NASA/ Science Office of Standards and Technology, 2001.

[19] HUANG, H. K.; ZHANG, Aifeng; LIU, Brent; ZHOU, Zheng; DOCUMET, Jorge; KING, Nelson; CHAN, L.W.C. *Data Grid for Large-Scale Medical Image Archive and Analysis.* Image Processing & Informatics Laboratory, Departments of Radiology and Biomedical Engineering, University of Southern California, ACM, 2005.

[20] IAU. *International Astronomical Union*. Disponível em: http://www.iau.org/ Acesso em: nov/2010.

[21] Legion. Disponível em: http://legion.virginia.edu/index.html. Acesso em: mar/ 2011

[22] KRAUTER, K; BUYYA, R; MAHESWARAN, M. *A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing*. Software: Practice&Experience,32(2):135–164, 2002.

[23] KREFTING, Dagmar; CANISIUS, Sebastian; HOHEISEL, Andreas; TOLXDORFF, Thomas; PENZEL, Thomas. *Grid Based Sleep Research Analysis of Polysomnographies using a Grid Infrastructure.* 9th IEEE/ACM International Symposium on Cluster Computing and the Gridv, 2009.

[24] MAGOULÈS, F; PAN, J; TAN, An K; KUMAR, A. *Introduction to Grid Computing*. Taylor & Francis Group, LLC, 2009.

[25] METS-Metadata Encoding and Transmission Standard (METS). Disponível em http://www.loc.gov/standards/mets/. Acesso em: jan/2010.

[26] MPEG-Moving Picture Experts Group (MPEG) ISO/IEC JTC1 SC29 WG11. Disponível em http://www.chiariglione.org/mpeg/. Acesso em jan/2010.

[27] NEESGRID. *Building The National Virtual Collaboratory for Earthquake Engineering.* Disponível em: http://www.neesgrid.org. Acesso: mar/2010.

[28]NISO. *Understanding Metadata.*Disponível em: http: //www.niso.org/publications/ press/UnderstandingMetadata.pdf. Acesso em: jan/2010.

[29] NOGUEIRA, Eduardo Dimas Andrino ; VAZ, M. S. M. G. ; SOUZA, Lucélia de . *Um Padrão de Metadados para Descrição de Imagens Astronômicas do Tipo FITS.* Revista Ciências Exatas e Naturais, v. 12, p. 55-72, 2010.

[30] NSSOLUTION. *Strengths of NS Solutions.* Disponível em: http://www.nssol.co.jp/en/corporate/index.html. Acesso em: mar/2010.

[31] OGSA-DAI. Disponível em: http://www.ogsadai.org.uk/about/index.php. Acesso em set/2011.

[32] Pbs gridworks: Openpbs. (2011). Disponível em http://www.openpbs.org. Acesso em: jan/2011.

[33] RAICU, I.; FOSTER, I.T.; WILDE, M.; ZHANG, Z.; ISKRA, K., BECKMAN, P.H.; ZHAO, Y.; SZALAY, A.S., CHOUDHARY, A.N.; LITTLE, P. *Middleware support for many-task computing*. Cluster Computing(2010) 291-314, 2010.

[34] REIS, V. Q. *Escalonamento em grids computacionais: estudo de caso*. Dissertação de Mestrado, Programa de Pós-Graduação em Ciências da Computação e Matemática Computacional, USP, 2005.

[35] ROURE, D.; JENNINGS, N. R.; SHADBOLT, N. R. *The Semantic Grid: Present, past and future*. Proceedings of IEEE 2005. Pages 669-681, 2005.

[36] SENSO, J. A.; PIÑERO, A. R. *El concepto de metadato. Algo más que descripción de recursos electrónicos*. Ciência da Informação, Brasília, v. 32, n. 2, maio/2011, 2003.

[37] TANNENBAUM, T.; WRIGHT, D.; MILLER, K. ; LIVNY, M. *Beowulf cluster computing with Linux, chapter Condor: a distributed job scheduler* p. 307_350. MIT Press, Cambridge, MA, USA, 2202, 2002.

[38] TANNENBAUM, A. *Metadata Solutions: Using Metamodels, Repositories, XML and Enterprise Portals to Generate Information on Demand.* Addison Wesley, 2002.

[39] THOMAS, M. P, BOISSEAU, J.R. *Building Grid Computing Portals: The NPACI Grid Portal Toolkit*. Grid Computing: Making the Global Infrastructure a Reality, 2003.

[40] VAZ, M. S. M. G. *MetaMídia – Um Modelo de Metadados na Indexação e Recuperação de Objeto Multimídia*. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco, 2002.

[41] VINE TOOLKIT 1.3.2. Disponível em: http://vinetoolkit.org/software_releases, Acesso em: fev/2011.

[42] XIE, Jiang; Zhang, Wu; Mei, Jian. *A Data Grid System oriented Biologic Data.* IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Workshops, 2007.

[43] WELLS D. C.; GREISEN E. W.; HAETEM R. H. *FITS: A Flexible Image Transport System*. Astronomy & Astrophysics Supplement Series. Páginas 363-370, 1981.

[44] WEBER E.; ANZOLCH R.; LISBOA F. J.; COSTA A. C.; IOCHPE C. *Qualidade de Dados Geoespaciais*. Relatório de Pesquisa. Universidade Federal do Rio Grande do Sul, 1990.

ANEXO A

A.1. Implementação do Ambiente MetafitsGrid

Nesta seção é descrita a instalação das ferramentas necessárias para construção e implementação do portal para grades de dados MetafitsGrid. O protótipo MetafitsGrid foi criado utilizando o *middleware* Globus, uma ferramenta livre e disponível via internet, o servidor web GNU Apache, o *framework* para desenvolvimento de portais de Grade, Gridsphere e o Vine *Toolkit*, responsável por efetuar a comunicação entre o portal Gridsphere e o *middleware* Globus. Na Seção A.1.1 é apresentado a instalação do Globus. Na seção A.1.2 e A.1.3 são descritas as instalações do Gridsphere e Vine *Toolkit*, respectivamente.

A.1.1. Instalação Globus Toolkit

A versão do globus utilizada foi a 4.0.5, sendo que o Globus pode ser instalado em diferentes sistemas operacionais: Debian, Arch Linux e Ubuntu. Podem ocorrer algumas diferenças, considerando a instalação da Versão 4 para a Versão 5. Os pré-requisitos para a instalação do Globus são: Java JDK, Apache ant, GNU tar, GNU sed, GNU Make, zlib, sudo, PostgreSQL.

As seguintes variáveis de ambiente NOME_HOST e GLOBUS_LOCATION são utilizadas para denotar o nome da máquina e a localização da instalação do Globus. Por exemplo:

NOME_HOST=josiane-laptop

GLOBUS_LOCATION=/usr/local/globus-4.0.5

1 - Download do Globus Toolkit 4

O download do instalador do Globus Toolkit está disponível através do site: http://globus.org/toolkit/downloads/4.0.5/

2- Descompactação do arquivo:

A descompactação do arquivo é feito através do seguinte comando:

tar –jxf gt4.0.5-all-source-installer.tar.bz2

Para fazer a instalação do Globus, é necessário instalar o Globus com outro usuário que não seja o *root*, a instalação é feita com o usuário chamado globus.

3 - Exportar variáveis de ambiente:

Antes de começar a instalação do Globus, é necessário exportar as seguintes variáveis de ambiente, para setar o local de instalação do Java e Ant e incluí-los no *PATH*.

export JAVA_HOME=/usr/lib/jvm/java-6-openjdk

export ANT_HOME=/usr/bin/ant

export PATH=ANT_HOME/bin:$JAVA_HOME/bin:$PATH

4- Instalação do Globus

Para a instalação do Globus, executar os seguintes comandos com o usuário *root*:

cd gt4.0.5-all-source-installer

./configure --prefix=/usr/local/globus-4.0.5;

make

make install

O comando make install poderá demorar alguns minutos. Após o comando ser executado com sucesso o Globus está instalado. Porém, são necessárias algumas configurações e a instalação de alguns componentes dependendo da finalidade que será utilizado o Globus. Por exemplo, caso o objetivo seja gerenciamento de dados, será necessário a instalação do RLS.

5- Criando e configurando a segurança do Globus

Para configurar a segurança do Globus, após definir a variável de ambiente GLOBUS_LOCATION, especificando a localização de instalação do Globus é necessário primeiramente criar a Unidade Certificadora (CA) para a grade. Para criar a Unidade Certificadora executa-se o comando setup-simple-ca, no seguinte caminho: GLOBUS_LOCATION/setup/globus, considerando o caminho da instalação definido no comando <*configure*>. Durante a execução do comando setup-simple-ca, é solicitado uma frase, usada para autenticar os certificados, posteriormente. Também, é gerado o diretório globus_simple_ca_(sequencia de numeros)_setup dentro do diretório *setup* localizada dentro de GLOBUS_LOCATION.

Após o comando anterior ter sido executado, é necessário executar o comando <setup-gsi>, responsável por tornar o certificado criado como padrão de certificado. O comando <setup-gsi> também cria em /etc um diretório chamado grid-security. O comando setup-gsi deve ser executado dentro do diretório globus_simple_ca_(sequencia de numeros)_setup da seguinte maneira: *setup-gsi –defaut*

6 – Obtendo certificados para computadores da grade

Em um ambiente de grade, é necessário ter uma Unidade Certificadora. O comando responsável por criar os pedidos de certificação é o comando <grid-cert-request> passando como parâmetro o nome do domínio da máquina. Por exemplo: *grid-cert-request -host 'NOME_HOST'.* O comando grid-cert-request gera no diretório grid-security os arquivos hostcert_request.pem e o hostkey.pem

7 – Assinando o certificado

Após a solicitação do Certificado, é necessário a solicitação ser assinada pela Unidade Certificadora. Para tanto, é necessário utilizar o comando <grid-ca-sign>. Para execução deste comando, é necessário digitar a frase definida no ínicio da instalação, sua sintaxe é: *grid-ca-sign –in hostcert_request.pem -out hostsigned.pem.*

Após a execução do comando, o Certificado assinado é armazenado em: /home/usuario/.globus/simpleCA;newcerts/01.pem. Este certificado deve ser renomeado para *hostsigned.pem* e copiado para a pasta */etc/grid-security*.

8 – Obtendo certificados para usuários da grade

Usuários que irão utilizar a grade, também, precisam ser autenticados. O usuário que irá utilizar a grade deverá executar o seguinte comando: *grid-cert-request*. Este comando irá solicitar uma senha que deverá ser utilizada para o usuário acessar a grade. Também será criado um arquivo chamado usercert_request.pem em: */home/usuário_solicitante_certificado/.globus.*

O usuário que instalou o Globus deverá autenticar o certificado através do seguinte comando: *grid-ca-sign -in usercert_request.pem -out signed.pem.*

Conforme passo 7, o Certificado assinado é armazenado em: */home/usuario/.globus/simpleCA;newcerts/02.pem*. Este certificado deve ser renomeado para signed.pem e copiado para o diretório do usuário solicitante .globus.

9 – Criando o arquivo grid-mapfile

O grid-mapfile tem como objetivo fazer o mapa da grid, fornecendo a lista dos usuários da grade com seu respectivo domínio. O arquivo grid-mapfile deve ser criado dentro do diretório /etc em GLOBUS_LOCATION.

O conteúdo do arquivo grid-mapfile é o seguinte:

/O=Grid/OU=GlobusTest/OU=simpleCA-josiane-laptop/CN=grade_josi

***10 – Criando e estartando o* container**

Alguns componentes do Globus, como o OGSA-DAI, utilizado para acesso e gerenciamento dos dados na grade, necessitam que o *container* esteja iniciado. Para que o *container* seja iniciado com sucesso, é necessário executar os seguintes comandos:

cp hostcert.pem containercert.pem

cp hostkeyt.pem containerkey.pem

Para iniciar o *container* é necessário executar o comando: *globus-start-container* Este comando está localizado dentro da pasta bin no diretório GLOBUS_LOCATION.

Para que o comando anterior seja executado com sucesso, é necessário que o arquivo global_security_descriptor.xml esteja configurado apontando para a localização correta dos arquivos: containerkey.pem, containercert.pem e grid-mapfile, conforme exemplo:

```
<?xml version=”1.0” enconding=”UTF-8” ?>
<securityConfig xmlns=http://www.globus.org>
<credential>
        <key-file value=”path de containerkey.pem”/>
        <cert-file value=”path de containercert.pem”/>
</credential>
        <gridmap value=”path de  grid-mapfile”/>
</securityConfig>
```

11 – Instalando e configurando o gerenciador de jobs Torque

O GRAM-WS, serviço de submissão de *jobs* padrão do Globus Toolkit, possui um gerenciador padrão chamado *fork*, porém este serviço possui algumas limitações quando se trata de submissão de tarefas mais complexas.

O instalador GT4 possui o PBS em seu fonte. Para instalar o PBS, é necessário entrar no instalador e executar os seguintes comandos:

make gt4-gram-pbs

make install

Após o comando ter sido executado com sucesso, é necessário configurar o *jobmanager* para que ele saiba que está sendo usado o *rsh*. Dentro do diretório $GLOBUS_LOCATION/setup/globus, é necessário executar o seguinte comando: *./setup-globus-job-manager-pbs –remote-shell=rsh*

A.1.2. Instalação do Gridsphere

O Gridsphere proporciona um Portal Web, utilizando o conceito de *Portlets*. Os *Portlets* são definidos como componente que provêm pequenos aplicativos que mostram

conteúdo informacional, podendo ser alterada a aparência e funcionalidades pelo usuário. O Gridsphere funciona da seguinte maneira:

- O cliente utiliza o navegador *web* (*browser*) e envia uma requisição ao portal;
- O Gridsphere *servlet* é invocado; e
- O Gridsphere *servlet* requisita ao mecanismo de *layout* do portal que converte os dados formando uma saída para o usuário.

Para controlar o acesso, Gridsphere utiliza o conceito de controle de acesso baseado em papéis (roles). Grupos podem ser criados e um conjunto de *portlets* é associado a ele. Um usuário pode pertencer a mais de um grupo, sendo que ele pode ser considerado um visitante, usuário, administrador e/ou super usuário.

Os pré-requisitos para a instalação do Gridsphere são: Java JDK, Apache ant e Apache Tomcat-5.5.26. O apache a ser instalado é a versão 5.5.26 para que o Gridsphere e o Vine, aplicativo que faz a comunicação entre o Gridsphere e o Globus, funcionem de maneira esperada.

Passos para instalação do Gridsphere:

1 - Download do Gridsphere

O Gridsphere deve ser baixado do site em sua Versão 3.2. A Versão correspondente disponível é Versão 3.1.

2 - Descompactação do arquivo baixado:

Para descompactar o arquivo baixado, executar o seguinte comando:

tar –vzxf GridSphere-3.1-src.tar.gz

3 - Definição das variáveis de ambiente

As seguintes variáveis de ambiente devem ser definidas antes da instalação do Gridshpere: JAVA_HOME, ANT_HOME e CATALINA_HOME. Após a descompactação do arquivo e definição das variáveis de ambiente e *PATH* deverá efetivar a instalação, conforme próximo passo.

4 - Efetivação da instalação

Para fazer instalação do Gridsphere é necessário acessar o diretório do Gridsphere descompactado e executar o comando:

ant install

Após a execução do comando, será mostrada a mensagem:

Do you ant to install Gridsphere JavaDoc API ([y], n]

É escolhida a opção [n], pois quando executava a opção para instalar a documentação [y] a instalação não conclui com sucesso. Após alguns minutos, a instalação do Gridsphere é executada com sucesso. É criado o diretório Gridsphere dentro do diretório Apache e um diretório gridsphere no diretório <home> do usuário responsável pela instalação do Gridsphere.

A.1.3. Instalação do Vine toolkit

O Vine Toolkit é um *framework open source*, modular, que oferece aos desenvolvedores uma biblioteca java extensível de fácil utilização e uma API de alto nível.

1 – Pré-requisitos

Antes de instalar o Vine, é necessário definir as seguintes variáveis de ambiente e defini-las no CLASSPATH, conforme exemplo abaixo:

export SERVLET_API=/root/srv/apache-tomcat-5.5.26/common/lib/servlet-api.jar
export JSP_API=/root/srv/apache-tomcat-5.5.26/common/lib/jsp-api.jar
export CLASSPATH=$SERVLET_API:$JSP_API:$CLASSPATH

2 - Download do Vine Toolkit

O Vine Toolkit pode ser baixado do seguinte endereço http://vinetoolkit.org/software_releases. Na instalação foi utilizada a versão vine-1-3-2.

3 - Descompactação do arquivo baixado:

Para descompactar o arquivo baixado, executa-se o seguinte comando:

tar –vzxf vine-1.3.2.tar.gz

4 -Instalação do vine

Dentro do diretório Vine extraído, executar o comando:

ant install

Após a execução do comando será mostrado os instaladores disponíveis para o Vine.
Na instalação foi escolhida a 14ª. Opção: Vine for Tomcat 5.X, GridSphere 3.1 and Globus Toolkit 4. Depois de alguns minutos o Vine é instalado com sucesso.

O computador que está executando a instalação deve ter acesso à Internet, pois o instalador consulta o *SVN* para fazer algumas atualizações no decorrer da instalação Após a finalização das instalações, é necessário iniciar o Apache, no seguinte caminho:

apache-tomcat-5.5.26/bin

./catalina.sh start

Utilizando o browser digita-se: http://localhost:8080/gridsphere/gridsphere. Será mostrada a configuração do Gridsphere. É escolhida a opção *Embedded Database,* onde será usado um banco de dados interno do Gridsphere. É necessário preencher onde os dados para criação do administrador do portal. O usuário e a senha criados serão utilizados posteriormente, para executar o login no Gridsphere.

Após a criação do login, já é possivel logar no portal. Para que o Gridsphere visualize os recursos da grade, é necessário alterar o Arquivo Domain.xml, localizado no seguinte caminho:

/srv/apache-tomcat-5.5.20/webapps/vine/WEB-INF/vine/resources.

Abaixo está sendo apresentado a configuração do arquivo Domain.xml utilizado pelo metafitsGrid, em itálico estão as alterações executadas no arquivo.

<domain name="gt4" label="Globus Toolkit 4" description="Globus Toolkit 4">
<!-- Portlet authentication -->
<authenticationModule key="PortletAuthModule" order="1"/>
<!-- Credential repository authentication (For use with "myproxyResource") -->
<authenticationModule key="CredentialRepositoryAuthModule" order="2"/>
<!-- Role resource -->
<vineRoleResource/>
<!-- Portal -->
<hostResource name="portal" *hostname="localhost"* label="Portal" description="Portal">
<!-- Portal file system (Do not remove!) -->
<portalFileSystem label="Portal File System" description="Portal File System"/>
<accountResource name="AccountManager" label="Account" description="Account Manager">
<gridsphereRegistrationResource name="GridsphereRegistration" label="GridsphereRegistration"/>
<!-- GSS demo certificate registration -->
<gssCertificateRegistrationResource name="GssDemoCertRegistration" label="GssDemoCertRegistration" *caCertFilePath="/root/.globus/simpleCA/cacert.pem" caKeyFilePath="/root/.globus/simpleCA/private/cakey.pem"caKeyPassword="m4r14n01"*/>
<!-- GT4 registration -->

<gt4RegistrationResource name="Gt4Registration" label="Gt4Registration" *targetHost="josiane-laptop.pl" superUserName="root" mkdirCmd="/bin/mkdir" chownCmd="/bin/chown" gridMapfileAddEntryCmd="sudo /usr/local/globus-4.0.5/sbin/grid-mapfile-add-entry" gridMapfileDelEntryCmd="sudo /usr/local/globu-4.0.5/sbin/grid-mapfile-delete-entry"/>*

</accountResource>

</hostResource>

<!-- MyProxy resource PCSS fury -->

<hostResource name="josiane-laptop"hostname="josiane-laptop" label="josiane-laptop" description="Linux Host">

<!-- MyProxy -->

<myproxyResource label="josiane-laptop (MyProxy)" useCredential="false" checkConnection="true" timeoutMiliseconds="5000"/>

</hostResource>

<!-- Globus GT 4.0.X josiane-demo Cluster -->

<hostResource name="Josiane-laptop" hostname="josiane-laptop" label="josiane Demo Machine" description="Linux Host">

<!-- GridFtp -->

<gridftpResource label="josiane (Grid-FTP)" description="josiane (Grid-FTP)"/>

<!-- WS-GRAM -->

<wsGramResource label="josiane-Demo (WS-GRAM)" port="8443">

<resourceAttribute name="description" value="GT4"/>

<!-- possible values: FORK, LSF, PBS, MULTI, CONDOR -->

<resourceAttribute name="factoryType" value="PBS"/>

<resourceAttribute name="WsrfResource.AuthorizationType" value="host"/>

<resourceAttribute name="WsrfResource.DelegationEnabled" value="true"/>

<resourceAttribute name="WsrfResource.MessageProtectionType" value="2"/>

</wsGramResource>

<!--WS MDS-->

<wsMdsResource label="PSNC (WS-MDS)"/>

<!--Proxy cert used by mds 4.0.X client-->

<gssCertConfiguration proxyFile="/tmp/x509up_u_dejw"/>

</hostResource>

</domain>

A.1.4. *Portlet*s criados no Gridsphere

Nesta seção são apresentados os portlets criados para apresentar o Padrão de Metadados para imagens FITS em ambientes de grades. A Figura 16 mostra a tela inicial do Portal para cadastrar imagens FITS de acordo com o padrão proposto.

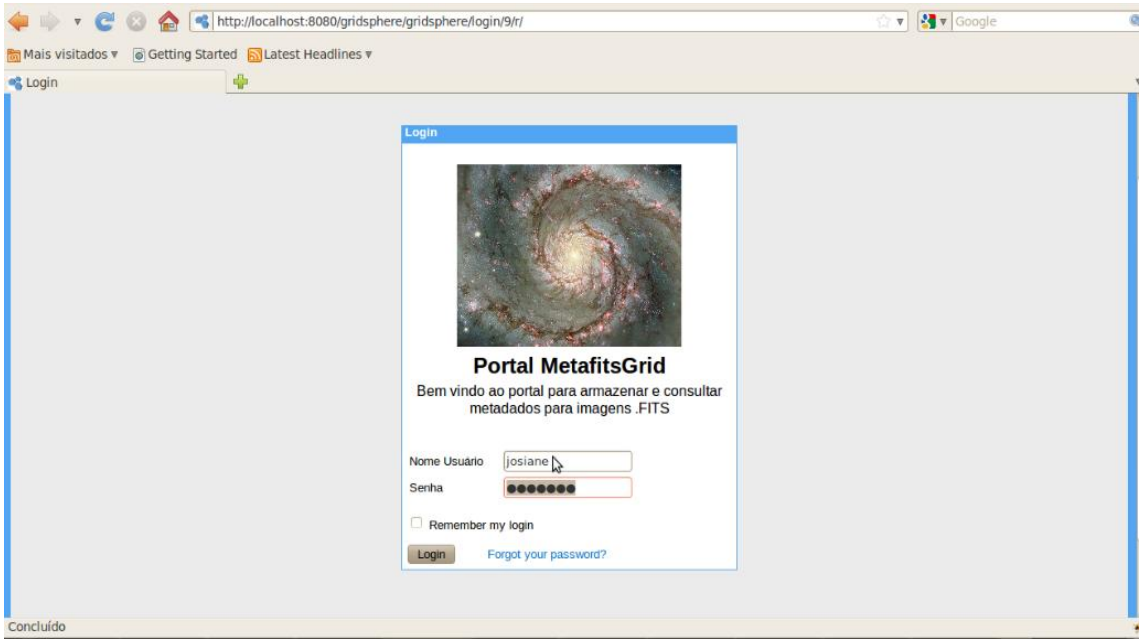

Figura 16. Portal MetafitsGrid

Foi criada uma aba para cada classificação dos metadados sugerida no padrão. As abas criadas foram: Identificação, Qualidade, Sistemas de Coordenadas, Fonte, Referências, Físicos, Independente de Domínio e Usuários.

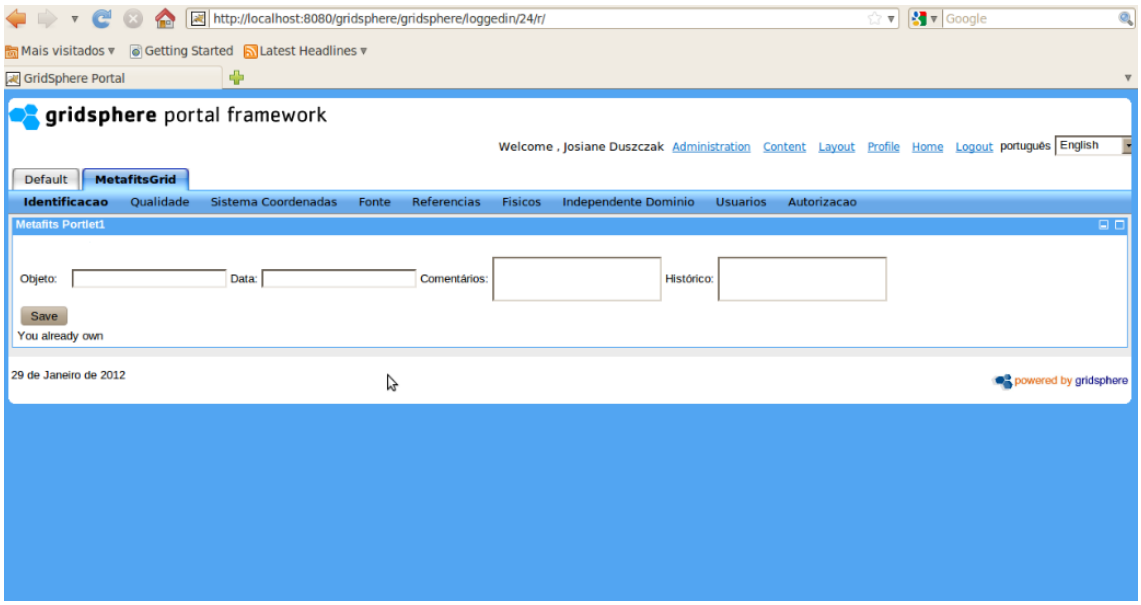

Figura 17. Aba Identificação

Na Figura 17 é mostrada a aba Identificação, onde permite a definição de entrada dos dados de identificação de objeto, data, comentários e histórico.

JOSIANE M. DINIZ DUSZCZAK

UMA ABORDAGEM ARQUITETURAL DE GRADE DE DADOS PARA IMAGENS FITS

Dissertação apresentada como requisito parcial à obtenção do grau de mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profª Dra. Maria Salete Marcon Gomes Vaz

CURITIBA

2012