



UNIFACS

UNIVERSIDADE SALVADOR

LAUREATE INTERNATIONAL UNIVERSITIES

**UNIFACS UNIVERSIDADE SALVADOR
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO**

THIAGO SOUZA HOHLENWEGER

**ESTRATÉGIA DE ARMAZENAMENTO DE DADOS DE EXPERIMENTOS
BASEADA NO IRODS PARA ARCABOUÇO DE CONTROLE E MONITORAÇÃO
DA REDE DO PROJETO FIBRE**

Salvador
2014

THIAGO SOUZA HOHLENWEGER

**ESTRATÉGIA DE ARMAZENAMENTO DE DADOS DE EXPERIMENTOS
BASEADA NO IRODS PARA ARCABOUÇO DE CONTROLE E MONITORAÇÃO
DA REDE DO PROJETO FIBRE**

Dissertação apresentada ao Curso de Mestrado Acadêmico em Sistemas e Computação, UNIFACS Universidade Salvador, Universidade Salvador – Laureate International Universities como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Dr. Joberto S. B. Martins.

Salvador
2014

FICHA CATALOGRÁFICA

Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador, Laureate International Universities

Hohlenwege, Thiago Souza

Estratégia de armazenamento de dados de experimentos baseada no IRODS para arcabouço de controle e monitoração da rede do projeto Fibre./ Thiago Souza Hohlenweger. – Salvador, 2014.

102 f.: il.

Dissertação apresentada ao Curso de Mestrado em Sistemas e Computação, UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Dr. Joberto S. B. Martins.

1. Redes de computadores. 2. Projeto FIBRE. I. Martins, Joberto S. B., orient. II. Título.

CDD.004.66

THIAGO SOUZA HOHLENWEGER

ESTRATÉGIA DE ARMAZENAMENTO DE DADOS DE EXPERIMENTOS BASEADA
NO IRODS PARA ARCABOUÇO DE CONTROLE E MONITORAÇÃO DA REDE DO
PROJETO FIBRE

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, UNIFACS Universidade Salvador, Laureate International Universities, pela seguinte banca examinadora:

Joberto Sérgio Barbosa Martins – _____
Doutor em Ciência da Computação, Université Paris VI
Universidade Salvador - UNIFACS

José Augusto Suruagy Monteiro _____
Doutor em Computer Science pela University Of California Los Angeles
Universidade Federal de Pernambuco - UFPE

Romildo Martins da Silva Bezerra
Doutor em Ciência da Computação - Ufba - Unifacs
Instituto Federal da Bahia - IFBA

Salvador, 3 de julho de 2014.

Dedico este trabalho às três mulheres e amores da minha vida, a minha mãe, Maria das Graças Lima Souza, a minha avó, Valdete Lima Souza e a minha esposa, Aline Nunes Silva, pelo amor, incentivo e apoio sempre que preciso.

AGRADECIMENTOS

Agradeço ao meu Deus, por sempre estar ao meu lado.

Ao Instituto Federal da Bahia, em especial à Diretoria de Gestão da Tecnologia da Informação, aos meus colegas, por me proporcionarem condições para a capacitação neste curso tão importante. À Edna, minha diretora e amiga, obrigado pela força e o apoio nesta reta final do curso, serei sempre grato.

Ao meu orientador, professor Dr. Joberto Martins, pelo apoio sempre presente, pela orientação e confiança no meu trabalho.

Ao professor Dr. Augusto Suruagy, que juntamente com o meu orientador, conduziram tão bem os trabalhos semanais do projeto FIBRE I&M.

Aos meus colegas do projeto FIBRE, ilha UNIFACS, Marcelo Pinheiro, Adriano Spínola, Igor Leonardo, Igor Luiz, Paulo Rios e Rafael Augusto (UFPE), que de tantas reuniões, trabalhos em conjunto, viagens e confraternizações já não os considero somente colegas, mas sim amigos.

Aos meus familiares e amigos sempre presentes em minha vida.

“Tudo posso naquele que me fortalece.”

Filipenses 4:13.

RESUMO

Uma Rede para Experimentação (*testbed*) é uma estrutura desenvolvida com o objetivo principal de apoiar a experimentação de novos protocolos, serviços e aplicações de redes, principalmente no contexto da Internet do Futuro. Ela permite que um experimentador possa configurar, instanciar, executar, coletar e salvar as medições de experimentos distribuídos (realizados entre “ilhas” – instâncias físicas do *testbed*) com o gerenciamento da operação sendo realizado por um Arcabouço de Controle e Monitoração (*Control and Monitoring Framework - CMF*). A rede para experimentação do projeto FIBRE¹, cenário de desenvolvimento abordado nesta dissertação, constitui-se um *testbed* com múltiplos CMFs (OFELIA – *OpenFlow in Europe – Linking Infrastructure and Applications*, OMF – *Orbit Management Framework* e ProtoGENI) integrados a fim de propiciar um conjunto mais abrangente de funcionalidades e serviços para o experimentador. Esta dissertação propõe uma estratégia para o armazenamento transparente de dados de medição oriundos de experimentos distribuídos realizados sob a gerência da Rede para Experimentação do Projeto FIBRE. Para tal, foram analisadas três aplicações de sistemas de armazenamento distribuídos, levando em consideração as questões técnicas para o armazenamento transparente e a federação. A partir desta análise, foi adotada a solução iRODS como sistema de gerenciamento de dados e, depois, foram propostas três estratégias de armazenamento no contexto do projeto FIBRE, sendo escolhida a estratégia que permitiu o acesso mais transparente aos dados de monitoração coletados entre os vários repositórios nativos. O estudo de caso validou a estratégia escolhida, que permitiu a federação entre instâncias de armazenamento de dados e a recuperação de dados.

Palavras-chave: Arcabouço de Controle e Monitoração – CMF. Projeto FIBRE. iRODS. Multi-CMF. Federação.

¹ Projeto de pesquisa financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), nº 590022/2011-3, e pela Comissão Europeia (EC) no âmbito do Programa de Cooperação FP7, nº288356.

ABSTRACT

A Network for Experimentation (testbed) is a framework developed aiming at supporting new protocols, network services and applications experimentation, especially in the context of Future Internet. It enables an experimenter to configure, instantiate, execute, gather and save distributed experiments measurements (performed among "islands" – physical instances of the testbed) with the operation management being performed by a Control and Monitoring Framework – CMF. The network for experimentation of the FIBRE² project, testbed used in this dissertation, is comprised of multiple CMFs (OFELIA – OpenFlow in Europe – Linking Infrastructure and Applications, OMF – Orbit Management Framework, and ProtoGENI) integrated in order to provide a more extended set of features and services to the experimenter. This dissertation proposes a strategy for transparent storage of FIBRE's experiments measurement data executed in the distributed management network for experimentation in FIBRE Project. To do this, three applications of distributed storage were analyzed, considering technical issues for the transparent storage and federation. Based on that analysis, the iRODS solution was adopted as data management system and after, three strategies of storage were proposed in the context of project FIBRE, being chosen the strategy that allowed the most transparent access to monitoring data collected from the various native repositories. A case study validated the strategy chosen, which allowed the federation between instances of data storage and data retrieval.

Keywords: Control Monitoring and Framework (CMF). Project FIBRE. iRODS. Multi-CMF. Federation.

² Research project funded by the National Council of Scientific and Technological Development (CNPq), # 590022/2011-3, and the European Commission (EC) under the Cooperation Programme FP7, #288356.

LISTA DE FIGURAS

Figura 1 - Arquitetura geral do OFELIA	21
Figura 2 - Arquitetura de Federação do ProtoGENI	23
Figura 3 - Componentes OMF	25
Figura 4 - FIBRE-BRnet e FIBRE-EUnet interconectados	27
Figura 5 - Infraestrutura básica da Ilha FIBRE-BR	28
Figura 6 - Arquitetura FIBRE-BR I&M	29
Figura 7 - Visão Geral dos componentes iRODS	41
Figura 8 - Visão Geral dos componentes MooseFS	45
Figura 9 - Visão Geral dos componentes HDFS	47
Figura 10 - Arquitetura FIBRE-BR I&M com os componentes iRODS	58
Figura 11- Estratégia de Armazenamento - Tipo 1	59
Figura 12 - Estratégia de Armazenamento - Tipo 2	61
Figura 13 - Estratégia de Armazenamento Tipo 3	62
Figura 14 - Funcionamento iRODS em uma máquina de experimentação	72
Figura 15 - Arquitetura da prova de conceito	75
Figura 16 - Gerenciador VM - VirtualBox	76
Figura 17 - Menu do ArmazenaDados	78
Figura 18 - Envio do dado com o metadado pelo experimentador	79
Figura 19 - Consulta dados e metadados - iRODS Web Browser	79
Figura 20 - Consulta por metadados	80
Figura 21 - Resultado da consulta	80
Figura 22 - Arquitetura da federação - Prova de Conceito	82
Figura 23 - Federação FIBRE com IFBA	84

LISTA DE QUADROS

Quadro 1 - Resumo comparativo entre as alternativas técnicas	56
Quadro 2 - Detalhes dos Componentes iRODS para a Prova de Conceito	75
Quadro 3 - Configuração das máquinas para Prova de Conceito	76
Quadro 4 - Locais dos experimentos realizados pelos experimentadores	77
Quadro 5 - Disposição dos arquivos por máquina de experimentação	78
Quadro 6 - Detalhes dos componentes iRODS da ilha IFBA para a Prova de Conceito	82
Quadro 7 - Descrição de ambiente iRODS do testbed do IFBA	83

LISTA DE ABREVIATURAS E SIGLAS

AM	<i>AGGREGATE MANAGER</i>
API	<i>APPLICATION PROGRAMMING INTERFACE</i>
AVU	<i>ATTRIBUTE – VALUE – UNIT</i>
CMF	<i>CONTROL AND MONITORING FRAMEWORK</i>
DICE	<i>DATA INTENSIVE CYBER ENVIRONMENTS</i>
DNS	<i>DOMAIN NAME SYSTEM</i>
EC	<i>EXPERIMENT CONTROLLER</i>
ED	<i>EXPERIMENT DESCRIPTION</i>
FIBRE	<i>FUTURE INTERNET TESTBED/EXPERIMENTATION BETWEEN BRASIL AND EUROPE</i>
FIRE	<i>FUTURE INTERNET RESEARCH & EXPERIMENTATION</i>
FQDN	<i>FULLY QUALIFIED DOMAIN NAME</i>
GENI	<i>GLOBAL ENVIRONMENT FOR NETWORK INNOVATION</i>
GN	<i>GLOBAL NODE</i>
GSI	<i>GRID SECURITY INFRASTRUCTURE</i>
HDFS	<i>HADOOP FILE SYSTEM</i>
HTTP	<i>HYPertext TRANSFER PROTOCOL</i>
I & M	<i>INSTRUMENTATION AND MEASUREMENT</i>
ICAT	<i>IRODS METADATA CATALOG</i>
IP	<i>INTERNET PROTOCOL</i>
IRODS	<i>INTEGRATED RULE-ORIENTED DATA SYSTEM</i>
LDAP	<i>LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL</i>
MAC	<i>MEDIA ACCESS CONTROL</i>
MDIP	<i>MEASUREMENT DATA INTEGRATION POINT</i>
ML	<i>MEASUREMENT LIBRARY</i>
MOOSEFS	<i>MOOSE FILE SYSTEM</i>
MP	<i>MEASUREMENT POINT</i>
MULTI-CMF	<i>MULTIPLE CONTROL AND MONITORING FRAMEWORKS</i>
NFEXP	<i>NETWORK FOR EXPERIMENTATION</i>
NICTA	<i>NATIONAL ICT AUSTRALIA</i>
NOC	<i>NETWORK OPERATION CENTER</i>

OCF	<i>OFELIA CONTROL FRAMEWORK</i>
ODP	<i>OPEN DISTRIBUTED PROCESSING</i>
OEDL	<i>OMF EXPERIMENT DESCRIPTION LANGUAGE</i>
OFELIA	<i>OPENFLOW IN EUROPE – LINKING INFRASTRUCTURE AND APPLICATIONS</i>
OMF	<i>ORBIT MANAGEMENT FRAMEWORK</i>
OML	<i>OMF MARKUP LANGUAGE</i>
PAM	<i>PLUGGABLE AUTHENTICATION MODULES</i>
PGENI	<i>PROTOGENI</i>
PGPOOL	<i>POSTGRES POOL</i>
PHP	<i>PERSONAL HOME PAGE</i>
RAM	<i>RANDOM ACCESS MEMORY</i>
RC	<i>RESOURCE CONTROLLER</i>
RM-ODP	<i>REFERENCE MODEL FOR OPEN DISTRIBUTED PROCESSING</i>
RNP	<i>REDE NACIONAL DE ENSINO E PESQUISA</i>
SDN	<i>SOFTWARE-DEFINED NETWORK</i>
SFA	<i>SLICE-BASED FACILITY ARCHITECTURE</i>
SGBD	<i>SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS</i>
SRB	<i>STORAGE RESOURCE BROKER</i>
TOR	<i>TOP OF RACK</i>
VLAN	<i>VIRTUAL LOCAL AREA NETWORK</i>
VM	<i>VIRTUAL MACHINE</i>
WAN	<i>WIDE AREA NETWORK</i>

SUMÁRIO

1 INTRODUÇÃO	16
2 ARCABOUÇOS DE CONTROLE E MONITORAÇÃO E PROJETO FIBRE	19
2.1 ARCABOUÇOS DE CONTROLE E MONITORAÇÃO	19
2.1.1 OFELIA (OpenFlow in Europe: Linking Infrastructure and Applications)	20
2.1.1.1 Arquitetura do OFELIA	20
2.1.1.2 Armazenamento dos dados de monitoração	22
2.1.1.3 ProtoGENI	22
2.1.1.4 Arquitetura da federação	23
2.1.1.5 Armazenamento dos dados de monitoração	24
2.1.2 OMF (ORBIT Management Framework)	25
2.1.2.1 Componentes OMF	25
2.1.2.2 Armazenamento dos dados de monitoração	26
2.2 REDE PARA EXPERIMENTAÇÃO DO PROJETO FIBRE	26
2.2.1 Arquitetura de instrumentação e medição (I&M) do FIBRE-BR	28
2.3 CONSIDERAÇÕES FINAIS	30
3 ARMAZENAMENTO DOS DADOS NO CONTEXTO DE REDES PARA EXPERIMENTAÇÃO	31
3.1 QUESTÕES TÉCNICAS	31
3.1.1 Armazenamento transparente	31
3.1.1.1 Semântica/Metadado	31
3.1.1.2 Controle de acesso	32
3.1.1.3 Segurança	33
3.1.1.4 Escalabilidade	33
3.1.1.5 Distribuição de interface cliente	33
3.1.2 Federação	34
3.2 SOLUÇÕES TÉCNICAS	34
3.2.1 Tipos de armazenamento	35
3.2.2 Armazenamento transparente	35
3.2.2.1 Semântica/Metadado	36
3.2.2.2 Acesso	38
3.2.2.3 Segurança	38

3.2.2.4 Escalabilidade	38
3.2.2.5 Distribuição de interface cliente	39
3.2.3 Federação	39
3.3 ALTERNATIVAS TÉCNICAS	39
3.3.1 IRODS	39
3.3.1.1 Visão Geral	40
3.3.1.2 Semântica de dados e metadados	41
3.3.1.3 Controle de acesso	42
3.3.1.4 Segurança	42
3.3.1.5 Escalabilidade	43
3.3.1.6 Distribuição de interface cliente	43
3.3.1.7 Federação	43
3.3.2 MOOSEFS	44
3.3.2.1 Visão Geral	44
3.3.2.2 Semântica de dados e metadados	45
3.3.2.3 Controle de acesso	46
3.3.2.4 Segurança	46
3.3.2.5 Escalabilidade	46
3.3.2.6 Distribuição de interface cliente	46
3.3.2.7 Federação	47
3.3.3 HDFS	47
3.3.3.1 Visão Geral	47
3.3.3.2 Semântica de dados e metadados	48
3.3.3.3 Controle de acesso	48
3.3.3.4 Segurança	49
3.3.3.5 Escalabilidade	49
3.3.3.6 Distribuição de interface cliente	49
3.3.3.7 Federação	49
3.4 AVALIAÇÃO E DEFINIÇÃO DA TECNOLOGIA DE ARMAZENAMENTO DE DADOS	50
3.5 CONSIDERAÇÕES FINAIS	55
4 PROPOSTA PARA O ARMAZENAMENTO DE DADOS DE EXPERIMENTOS EM UM CENÁRIO MULTI-CMF	57

4.1 ARMAZENAMENTO DE DADOS BASEADO NO IRODS PARA O PROJETO FIBRE	57
4.2 ESTRATÉGIAS IRODS PARA O PROJETO FIBRE	58
4.2.1 Estratégia Tipo 1	59
4.2.2 Estratégia Tipo 2	60
4.2.3 Estratégia Tipo 3	62
4.3 DEFINIÇÃO DA ESTRATÉGIA DE IMPLANTAÇÃO DO IRODS	63
4.4 IMPLANTAÇÃO DOS COMPONENTES IRODS PARA O PROJETO FIBRE	65
4.4.1 Aplicação cliente iRODS para máquina de experimentação	65
4.4.2 Aplicação cliente iRODS para acesso externo	66
4.4.3 Configuração da máquina cliente	67
4.4.4 Configuração dos servidores iRODS	68
4.4.4.1 Requisitos	68
4.4.4.2 Criação de recursos	69
4.4.4.3 Autenticação	70
4.5 ESTRATÉGIA DE ACRÉSCIMO E USO DE METADADOS NO CONTEXTO DO PROJETO FIBRE ASSOCIADO ÀS FACILIDADES DO IRODS	70
4.6 ARMAZENADADOS	71
4.7 CONSIDERAÇÕES FINAIS	72
5 PROVA DE CONCEITO	74
5.1 INFRAESTRUTURA DE IMPLANTAÇÃO	74
5.2 TESTE DA INFRAESTRUTURA IRODS (ESTRATÉGIA TIPO 2)	77
5.3 TESTE DA ESTRATÉGIA DE FEDERAÇÃO	81
6 CONCLUSÃO	85
6.1 TRABALHOS FUTUROS	85
REFERÊNCIAS	87
APÊNDICE A – Configuração do ambiente da prova de conceito	90
CONFIGURAÇÃO DO SERVIDOR IRODS SERVER ICAT	90
APÊNDICE B – Código Fonte do Armacenadados	92
APÊNDICE C – Cookbook de Instalação	95

1 INTRODUÇÃO

Redes para Experimentação ou *Network for Experimentation* (NfExp) têm sido amplamente utilizadas em todo o mundo, principalmente por projetos de pesquisa acadêmica. Além disso, grandes redes e empresas de tecnologia estão profundamente envolvidas na experimentação e utilização de redes definidas por software – *Software-Defined Network* (SDN) em seus laboratórios, em seus produtos e, eventualmente, na sua própria infraestrutura. Como exemplo, o Google anunciou que utiliza SDN e o protocolo de comunicação *OpenFlow* para interconectar o seu *datacenter* e enlaces de redes de longa distância (WAN), considerando as questões de engenharia de tráfego (PALMER, 2012). Com efeito, este tipo de desenvolvimento inovador necessita de várias experiências anteriores que podem ser realizadas em *testbeds*, a fim de garantir uma implementação confiável na rede de serviços. Na verdade, as organizações têm fortes preocupações em comprometer seus negócios de milhões de dólares em um desenvolvimento inovador sem a devida validação, testes, implantação, medições e certo nível de confiança. Isso enfatiza a importância de ter redes de experimentação, também mencionadas como *testbeds*, a fim de rodar os experimentos. Como tal, NfExp ajuda no desenvolvimento de novas tecnologias, como abordado no SDN, *OpenFlow* ou qualquer outro novo serviço ou protocolo. Como exemplo, recentemente um estudo foi publicado pela SDNCentral, empresa de consultoria especializada de gestão em computação em nuvem e SDN, estimando-se um mercado SDN de U\$ 35Bilhões até 2018, o que torna uma grande oportunidade para as empresas e as universidades (SDNCENTRAL, 2013).

Estes *testbeds* enfrentam alguns desafios conhecidos como: recursos para orquestração, federação, controle de acesso, disponibilidade, desempenho, instrumentação e suporte de monitoração, integração e operação de múltiplos CMFs (*Control and Monitoring Frameworks*), software de controle e monitoramento, só para citar alguns.

Quanto ao Arcabouço de Controle e Monitoração, conhecido também por CMF existem várias soluções que estão sendo implantadas em todo o mundo, tais como: *Global Environment for Network Innovation* (GENI), OneLab, ProtoGENI, PlanetLab, *OpenFlow* na Europa - Ligando Infraestruturas e Aplicações (OFELIA) e Orbit Management Framework (OMF). Cada CMF visa resolver uma série de problemas com foco em sua comunidade de pesquisa (MONTEIRO, 2010).

Considerando um ambiente com vários CMFs ou multi-CMF, envolvendo múltiplas instituições, todas elas podem se beneficiar com o uso e o reaproveitamento do código de aplicação diferente, com inúmeros usuários, promovendo a inovação entre as diferentes comunidades para arquitetura de Internet do Futuro e a integração e colaboração entre Universidades de diferentes países ao redor do mundo.

Assumimos que este é um cenário comum hoje em dia, em que, o ambiente multi-CMF é uma realidade. Com ele, os múltiplos desafios e questões técnicas estão sendo investigados. Alguns dos desafios multi-CMF são: a federação, instrumentação/orquestração do experimento, segurança, monitoração, armazenamento dos dados de experimentação, controle de acesso, entre outros.

Cada CMF trata estes desafios de forma individualizada onde o trabalho em conjunto destes CMFs na mesma ilha/*testbed* potencializa o problema, por escalar a quantidade de arcabouços diferentes. Um destes desafios em um ambiente multi-CMF é tratado nesta dissertação, ao apresentar uma estratégia para o armazenamento dos dados de monitoração dos experimentos entre múltiplos CMFs de forma que o acesso a estes dados sejam o mais transparente possível ao experimentador.

O experimentador de posse de uma Rede de Experimentação com vários CMFs espalhados em várias instituições poderá fazer uso de um experimento global com múltiplos pontos de medição e coleta dos dados em diferentes pontos ao redor do globo. Certamente, para recuperar e acessar os resultados dos dados dos experimentos seria um grande desafio, o que definitivamente não é o principal objetivo do experimentador.

Para o armazenamento transparente dos dados de medição multi-CMF e instituições envolve várias questões, tais como:

- Qual a arquitetura utilizada?
- Como será tratada a semântica de dados e metadados?
- Como permitir o acesso aos dados armazenados já que envolve várias instituições?
- Como será tratada a disponibilidade do sistema?
- Como atender ao crescimento do *testbed* com a inclusão de novas instituições federando com as instituições já existentes?
- Como implantar as aplicações clientes nos múltiplos CMF de forma uniformizada e transparente para o usuário experimentador?

Neste contexto, será apresentada a estratégia para o armazenamento transparente dos dados de monitoração para a Rede de Experimentação do projeto FIBRE (*Future Internet testbed/experimentation between BRasil and Europe*), que é um *testbed* composto de três CMFs (OFELIA, ProtoGENI e OMF) espalhados por várias instituições do Brasil e da Europa, de forma a mitigar os problemas para o armazenamento dos dados.

Sendo assim, visando atingir os objetivos de um armazenamento transparente dos dados, foi objeto deste trabalho a avaliação de uma estratégia de armazenamento baseado em soluções do tipo sistemas distribuídos com as aplicações: HDFS (*HaDooP File System*), MooseFS (*Moose File System*) e iRODS (*integrated Rule-Oriented Data System*) propondo uma solução para projeto FIBRE e validando a proposta através de uma prova de conceito.

Este trabalho está dividido da seguinte forma: no capítulo 2, é feita a apresentação do arcabouço de controle e monitoração e a descrição da rede para experimentação do projeto FIBRE; o capítulo 3 relata os desafios para o armazenamento dos dados no contexto de redes para experimentação, as soluções técnicas para o armazenamento de dados no contexto de redes para experimentação e as alternativas técnicas para o armazenamento de dados de experimentação; no capítulo 4, são expostas algumas propostas para o armazenamento dos dados multi-CMF; o capítulo 5 traz a prova de conceito, e por último são apresentadas as conclusões do trabalho.

2 ARCABOUÇOS DE CONTROLE E MONITORAÇÃO E PROJETO FIBRE

Neste capítulo, são apresentados arcabouços de controle e monitoração juntamente com a rede para experimentação do projeto FIBRE. Nele são abordados os tipos de arcabouços utilizados pelo projeto FIBRE, sua infraestrutura e arquitetura de instrumentação e medição do FIBRE-BR.

2.1 ARCABOUÇOS DE CONTROLE E MONITORAÇÃO

A inovação tem se consistido no foco das pesquisas em rede de comunicação nos últimos anos. E, para que possa ser aplicada, a inovação necessita de uma rigorosa investigação em larga escala, que se apoia na variedade de ferramentas disponíveis como, por exemplo, simuladores, emuladores e infraestruturas ou *testbeds* para experimentação de redes (MARCONDES *et al.*, 2012). No contexto das infraestruturas têm sido desenvolvidas complexas arquiteturas de controle e monitoramento que habilitam experimentadores sem experiência a testar novas ideias com o mínimo de esforço. Um arcabouço de controle e monitoramento permite registrar os recursos, registrar o experimentador, controlar o acesso aos serviços da infraestrutura, criar e registrar um particionamentos da rede em camadas (fatias ou *slice*) de utilização de recursos, gerenciar o ciclo de experimentos, monitorar os resultados, armazenar os resultados, entre outras é chamado de CMF (*Control and Monitoring Framework*).

Os CMFs normalmente estão inseridos nas suas próprias redes de experimentação ou *testbeds*, possibilitando aos pesquisadores desenvolverem suas pesquisas em larga escala. Como exemplos destes *testbeds* têm-se:

- GENI nos EUA (*Global Environment for Network Innovations*);
- FIRE na Europa (*Future Internet Research & Experimentation*);
- OFELIA (*OpenFlow in Europe: Linking Infrastructure and Applications*) *testbed* baseada em rede *OpenFlow*;
- FIBRE (*Future Internet testbed/experimentation between BRasil and Europe*) ainda em desenvolvimento da qual este trabalho faz parte.

A busca, atualmente, por ambientes para experimentação tem sido grande por parte da comunidade, o que tem gerado um esforço considerável para os desenvolvedores de CMFs para tornar tais sistemas mais genéricos e mais fáceis de usar de forma a suportar a federação de seus recursos. A federação está na junção de vários sistemas independentes distribuídos

que cooperam entre si de acordo com uma política de gestão (PETERSON *et al.*, 2007). No entanto, este objetivo é ambicioso porque permite que recursos independentes e distribuídos na rede formem um grande conjunto de recursos para testes de novas ideias e possam ser estudados em redes e sistemas distribuídos.

Como objeto de estudo deste trabalho serão abordados três CMFs (OFELIA, ProtoGENI e OMF) utilizados no projeto FIBRE, tema da seção 2.2, descrevendo o objetivo e sua arquitetura, detalhando seus componentes no ambiente de experimentação e a forma, caso exista, de armazenamento dos dados de monitoração.

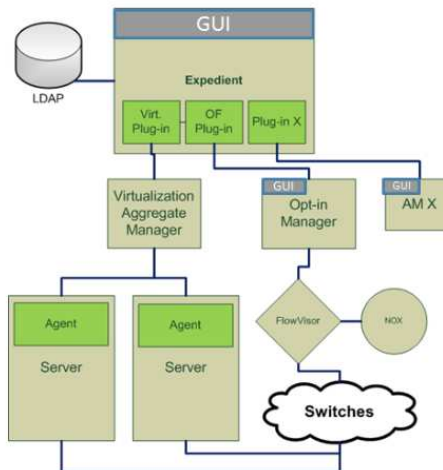
2.1.1 OFELIA (OpenFlow in Europe: Linking Infrastructure and Applications)

O CMF OFELIA possui o seu software de controle público também chamado de OCF (OFELIA *Control Framework*), projeto colaborativo dentro do *7th Framework Programme – FP7* da Comunidade Europeia (OFELIA, 2013), objetivando uma infraestrutura de experimentação (*testbed*) baseada em *OpenFlow*, tecnologia que permite a virtualização, criação de várias redes virtuais em uma rede física, o controle e a gerência da rede através de interfaces seguras. Teve seu início em 2010, com a interligação de oito ilhas *OpenFlow* na Europa permitindo ao experimentador desenvolver seus experimentos em diversas camadas e muitas tecnologias, passando a testar não somente a rede, mas também o controle da rede a partir do substrato *OpenFlow*, o que possibilita ao experimentador montar e configurar a sua própria rede em nuvem (MARCONDES *et al.*, 2012).

2.1.1.1 Arquitetura do OFELIA

A arquitetura do OCF utiliza quatro componentes para o sistema de controle do OFELIA que são: *Expedient*, *VT Aggregate Manager*, *Opt-in Manager* e os agentes, todos configurados normalmente no mesmo servidor. Na Figura 1, segue a arquitetura do OFELIA (OFELIA, 2012).

Figura 1 - Arquitetura geral do OFELIA



Fonte: OFELIA (2012).

Ao se detalhar os componentes da arquitetura OFELIA têm-se:

- *Expedient*: é o elemento central na arquitetura do OCF e possui o controle de todas as funções básicas do OCF como: gerenciar os *logs*, os projetos, os pedidos de acesso, os recursos atuais do sistema. Ele permite que o usuário possa gerenciar seus recursos criando experimentos para testes de sistemas ou serviços de rede. O *expedient* utiliza-se de plug-ins para comunicação com os outros componentes da arquitetura OFELIA, o VT Manager e o Opt-In Manager. O administrador faz a gerência do *FlowVisor*, *proxy* de controladores para o *OpenFlow*, utilizando do *plug-in OpenFlow*. (MARCONDES *et al.*, 2012).
- Os agregados (*Aggregates*) são conjuntos de recurso que podem ser reservados para utilização no experimento.
- O *VirTualization Aggregate Manager* (VT AM) controla os recursos do *testbed*, que são as máquinas virtuais (VMs) dentro dos servidores, dispositivos de rede, entre outros, podendo o experimentador criar as VMs com as características desejadas.
- Os agentes são responsáveis pela provisão dos recursos de virtualização das ilhas *OpenFlow*.
- O *OpenFlow Aggregate Manager* (OF AM) controla os recursos *OpenFlow* do *testbed*, habilitando o OF no switch (*OF-enabled switch*), o experimentador poderá definir um *FlowSpace*, filtragem de fluxos que se baseiam em campos de cabeçalho de protocolos (como IP de origem, *MAC Address*, *DST Port*,

VLAN, entre outros), envolvendo máquinas virtuais, particionando a rede em camadas ou *slices* para a experimentação. Além de, definir o controlador que irá atualizar a tabela de fluxos do *switch* que na Figura 1 está sendo representada pelo NOX.

Assim, com os componentes OCF é possível particionar a rede utilizando conjuntos de campos de cabeçalhos como rótulos de VLAN onde mensagens de controle são enviadas aos respectivos controlador do experimentador. Desta forma, o controlador do experimentador poderá controlar os *switches OpenFlow* da infraestrutura e instanciar máquinas virtuais associadas a esses rótulos de VLAN para o seu experimento.

2.1.1.2 Armazenamento dos dados de monitoração

O OFELIA ainda não dispõe de ferramentas para monitoração do experimento, informações decorrentes do experimento realizado envolvendo métricas de medição significativas para o experimentador como tempo de resposta, atraso e até mesmo, medir a vazão de um determinado fluxo, decorrente da comunicação entre dois computadores. Ele também não possui aplicações para o armazenamento de dados de monitoração. O monitoramento realizado pelo OFELIA até então, está voltado apenas para a infraestrutura e utiliza as ferramenta Zenoss (2014) (KÖRNER, 2013) e OML (VERMEULEN, 2013), para medir processamento, uso da memória, espaço em disco, interface de rede, entre outros dispositivos, armazenando estes dados coletados em uma base de dados MySQL. Entretanto, já existe trabalho em desenvolvimento visando atender a demanda de monitoramento do experimento, como é o caso do MDIP para o OFELIA (PINHEIRO *et al.*, 2012).

2.1.1.3 ProtoGENI

O ProtoGENI é um *Control Framework*, integrante do projeto da *Global Environment for Network Innovations – GENI* que é liderada pela *Flux Reseach Group* da Universidade de Utah, com grande influência do software do *testbed* Emulab (2014).

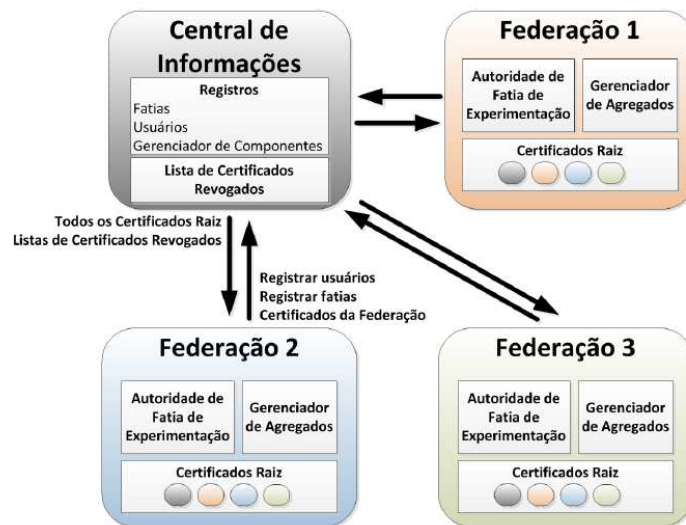
Criada para fazer a integração entre o Emulab e os outros *testbeds* a fim de construir um Cluster C de facilidades para o GENI (PROTOGENI, 2013), o projeto foi baseado no

SFA³ - *Slice-based Facility Architecture*, com a finalidade de provisionar mais recurso para o experimentador com a federação entre os *testbeds*.

2.1.1.4 Arquitetura da federação

A arquitetura da federação proporciona a inclusão de vários agregados federados e seus componentes, possibilitando um *pool* de recurso para o experimentador, que poderá entre outras funções, alocar máquinas físicas e também virtuais, sendo que as virtuais poderão estar todas em uma mesma máquina física, de modo a oferecer todos os recursos necessários para o experimento. A Figura 2 retrata detalhes da arquitetura da federação do ProtoGENI apresentando uma visão das interações dos seus componentes como: *Clearing House*, *Aggregate Manager*, *Slice Authority* e o *RSpec* (MARCONDES *et al.*, 2012).

Figura Erro! Nenhum texto com o estilo especificado foi encontrado no documento.1 - Arquitetura de Federação do ProtoGENI



Fonte: PROTOGENI (2013).

Ao se detalhar os componentes da arquitetura de federação ProtoGENI têm-se:

- Central de Informações (*Clearinghouse*): é o elemento central para se localizar as informações sobre os componentes gerenciados, usuários, responsável por registrar fatias (*slices*), etc. Tem como objetivo ajudar o Gerenciador de Componentes e os experimentadores dos recursos a encontrarem subsídios necessários para elaboração do experimento. Outro ponto importante é a

³ SFA é um conjunto mínimo de interfaces e tipos de dados que permitem a interoperabilidade de uma federação de *testbeds* (MARCONDES *et al.*, 2012)

relação de confiança entre os *testbeds* da federação ao trocarem certificados e Lista de Certificados Revogados;

- Autoridade de Fatia de Experimentação (*Slice Authority*): é responsável por atribuir nomes, registrar e permitir que usuários acessem e controlem seus *slices*. Cada experimentador utiliza certificado para se autenticar com a Autoridade de Fatia de Experimentação local, que por sua vez permite validar o usuário em toda a federação.
- Gerenciador de Agregados (*Aggregate Manager*): define as operações disponíveis para os serviços no nível do usuário de forma a gerenciar as alocações dos recursos para diferentes usuários e experimentos. Ele tem suporte à autenticação de usuários e verificação das credenciais. Anuncia os seus recursos disponíveis, aceita as requisições de criação de *slices* de qualquer membro da federação, aplica política simples para a disponibilização dos recursos para usuários de outros membros da federação, cria múltiplas instâncias (*slivers*⁴) nos computadores e cria as topologias de rede entre os computadores disponíveis.
- RSpec: Apesar de não está representado na figura, ele é a descrição de um formato para a comunicação dos dados entre os *testbeds*. Nele, o experimentador define os recursos desejados para o experimento como também poderá definir a topologia da rede a ser utilizada.

2.1.1.5 Armazenamento dos dados de monitoração

No ProtoGENI (PGENI), as medições são realizadas por Pontos de Medição (MP) dentro da *slice* do usuário. É armazenado em um nó especial chamado Global Node (GN) também dentro da *slice*. Este GN possui, dentre outros serviços, um cliente iRODS (que será abordado na seção 3.3.1) com a função fim de enviar os dados de medição para o servidor de armazenamento.

A *slice* disponível para o experimentador possui um tempo de vida que ao término deste tempo é removida pelo PGENI e conseqüentemente o GN, também, é excluído junto com os dados de experimentação. Caso o experimentador não solicite o armazenamento dos

⁴ *Slivers*: parte ou fragmento de um recurso.

dados coletados pelos experimentos e o tempo de uso da *slice* tenha chegado ao fim, os dados coletados serão perdidos.

2.1.2 OMF (ORBIT Management Framework)

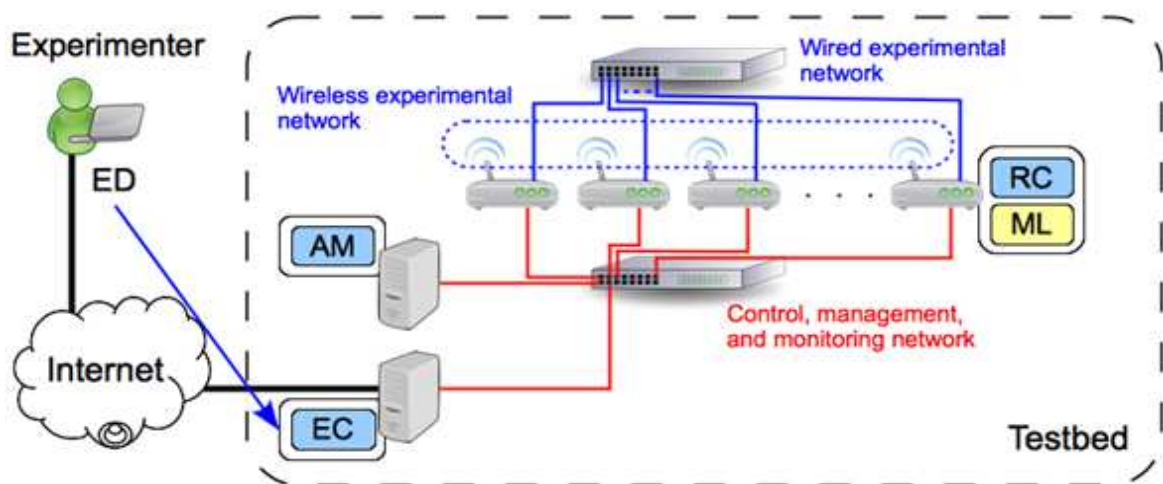
O OMF foi originalmente desenvolvido para o ORBIT *wireless testbed* da Winlab na Universidade Rutgers, com o objetivo de controlar redes sem fio utilizando o projeto de *testbeds* (RAKOTOARIVELO *et al.*, 2010). A partir do ano de 2008, passou a ter a colaboração da *National ICT Australia* (NICTA).

A sua estrutura é composta por uma série de componentes capaz de gerenciar de forma eficiente todos os recursos do *testbed* como: resgatar os nós, reportar o status da rede, instalar novos sistemas, instrumentar, executar o experimento e coletar os resultados. Tais funcionalidades são fundamentais para o pesquisador que irá utilizar-se deste arcabouço na sua pesquisa.

2.1.2.1 Componentes OMF

A sua estrutura é formada por ferramentas e componentes divididos da seguinte forma: Gerenciamento, Controle e Medição. Na Figura 3, são detalhados os componentes que compõe a estrutura OMF.

Figura 2 - Componentes OMF



Fonte: FIBRE (2013).

- Ferramenta de Gerenciamento é formada pelo *Aggregate Manager* (AM) e tem a responsabilidade de gerenciar os recursos de todo o *testbed*.

- Ferramenta de Controle é formada por: *Script* de descrição do experimento (ED - *Experiment Description*), escrito em OEDL (*OMF Experiment Description Language*) que define os recursos e ações para o experimento, tais como, configuração inicial da máquina e tempo de duração do experimento; *Resource Controller* (RC), também chamado de *NodeAgent*, responsável pela execução das ações definidas nos recursos; *Experiment Controller* (EC) faz a comunicação entre o *testbed* e o usuário, processando o ED e sinalizando para o AM sobre quais recursos alocar para o experimento, o que resulta na comunicação com cada RC que irá executar o experimento.
- Ferramenta de Medição é formada pela biblioteca de medição *Measurement Library* (ML), desenvolvida na linguagem C, que precisa estar disponível em todos os recursos que se pretende monitorar.

2.1.2.2 Armazenamento dos dados de monitoração

O OMF tem no ciclo de vida do experimento as seguintes etapas. Primeiro, o usuário experimentador apresenta a descrição do experimento (ED) escrita em texto OEDL para o Controlador de Experimento (EC) que fará a orquestração dos recursos. Este se comunica com o Gerente Agregado ou *Aggregate Manager* (AM) que irá configurar os recursos conforme solicitado no texto e em seguida enviará uma diretiva para o RC executar as ações dos experimentos nos recursos. A medição é feita pela Biblioteca de Medição ML que registra as métricas definidas dos experimentos e envia todos os dados coletados e medidos para o banco de dados MySQL do servidor OML⁵ (FIBRE, 2013c). Cada experimento executado cria um novo banco de dados com o mesmo nome do identificador do experimento.

O dado estará disponível para o experimentador através do EC, que por sua vez fará o acesso ao servidor OML, a fim de recuperar os resultados da experimentação.

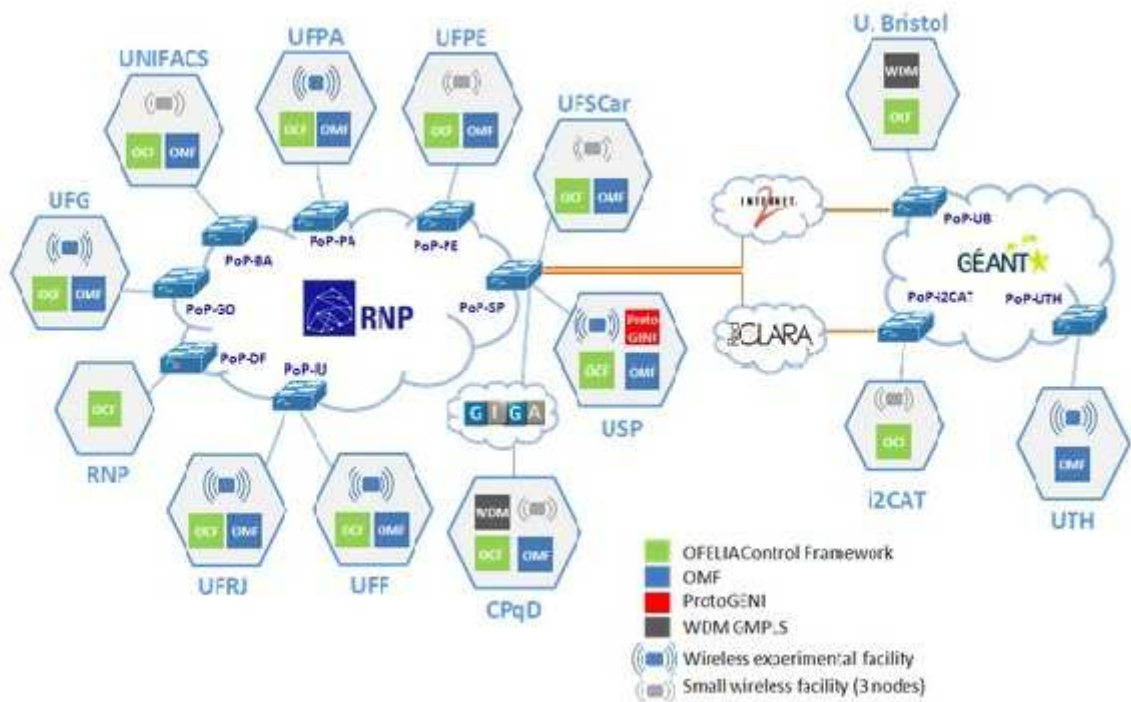
2.2 REDE PARA EXPERIMENTAÇÃO DO PROJETO FIBRE

O FIBRE (*Future Internet testbed/experimentation between Brasil and Europe*) (FIBRE, 2013a) é um projeto de Rede de Experimentação e inovação entre o Brasil e a Europa, tendo como principal objetivo o suporte à pesquisa na Internet do Futuro (IF). É,

⁵ OML é uma biblioteca de medições/monitoramento que permite coletas de diversas aplicações sendo executadas em um experimento (OML, 2013).

basicamente, uma infraestrutura de rede intercontinental composta pelo FIBRE *Backbone* no Brasil (FIBRE-BRnet), outros na Europa (FIBRE-EUnet) e as redes de ilhas. O FIBRE-BRnet está focado na interconexão das dez instituições ou ilhas no Brasil e o FIBRE-EUnet interconecta três instituições na Europa. A Rede Nacional de Ensino e Pesquisa (RNP) e a Rede Gigabit de pesquisa pan-europeia (GÉANT) estão com suas redes de *backbone* interconectadas (Figura 4).

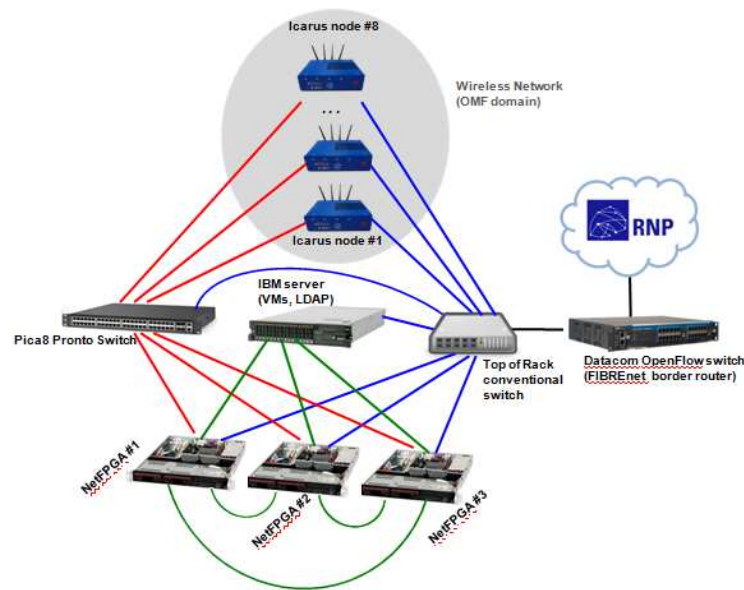
Figura 3 - FIBRE-BRnet e FIBRE-EUnet interconectados



Fonte: FIBRE (2013).

As ilhas situadas no Brasil são compostas por nós Icarus, *switches OpenFlow*, *switch Top Of Rack (ToR)*, servidores netFPGA, servidores utilizados para virtualização e serviços comuns (CMFs, LDAP, DNS, Infraestrutura e Monitoração, etc.) e um roteador de borda conforme detalhe na Figura 4.

Figura 4 - Infraestrutura básica da Ilha FIBRE-BR



Fonte: Marcondes *et al.*(2013).

Todos estes recursos de infraestrutura compõem uma infraestrutura básica de compartilhamento de três diferentes CMFs nas ilhas do FIBRE-BR: *OFELIA Control Framework*, *Orbit and Management Framework* e o ProtoGENI.

Com esta infraestrutura envolvendo várias instituições do projeto FIBRE no Brasil e vários *Control and Monitoring Frameworks* se tem o ambiente de experimentação chamado FIBRE-BR multi-CMF. Diante deste ambiente, será apresentada na próxima seção uma arquitetura de instrumentação e medição para monitoração da infraestrutura e dos experimentos para o projeto FIBRE-BR.

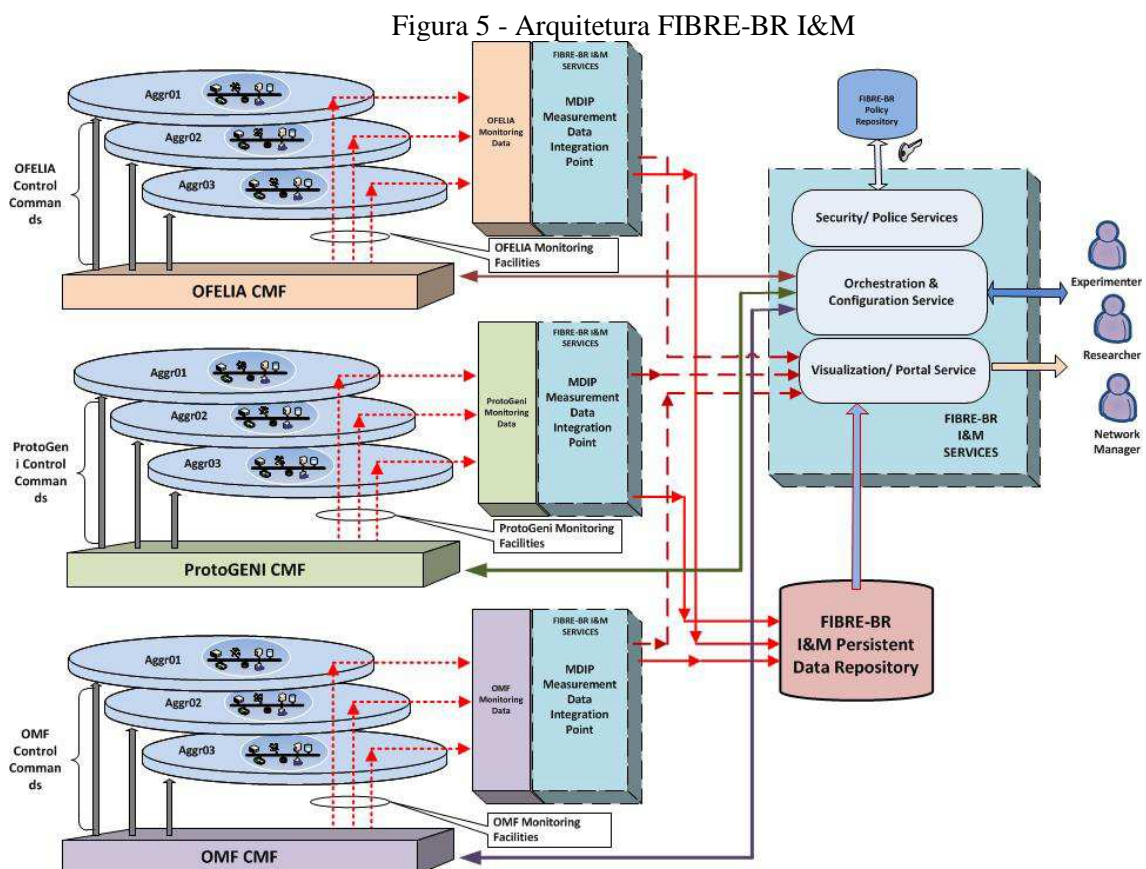
2.2.1 Arquitetura de instrumentação e medição (I&M) do FIBRE-BR

No que se refere ao quesito monitoração, é de responsabilidade do FIBRE-BR multi-CMF atender ao requisito básico de: configurar, monitorar, coletar e exibir os dados de monitoramento da infraestrutura e experimentação.

Para isso, foi desenvolvida a arquitetura de monitoração do projeto FIBRE-BR I&M (PINHEIRO *et al.*, 2012), que poderá ser verificada na Figura 6. Nesta figura, é possível verificar os agregados, conjuntos de recursos reservados para utilização no experimento, nos CMFs. Parte da reserva deste conjunto de recursos a um experimentador é chamado de *slice* ou fatia, representada na figura em cada CMF como: Aggr01, Aggr02 e Aggr03. Os dados

resultantes da monitoração dos experimentos em cada CMF deverão ser acessados *online* ou ser armazenados em um repositório de dados persistente, para isso existe na arquitetura I&M os pontos de integração e medição dos dados o MDIP (*Measurement Data Integration Point*).

A interface MDIP é o ponto de integração dos dados coletados pelos CMFs para o formato padrão NMWG (*Network Measurements Working Group*) do *Open Grid Forum* (OGF), utilizada pelo ambiente perfSONAR (PERFSONAR, 2014). Ela é escrita na linguagem *Extensible Markup Language* (XML) formatada de acordo com seu esquema específico para representar os dados de medição, ou até mesmo a conversão e exportação dos dados coletados para um formato comum a alguma ferramenta de Instrumentação e Medição. O MDIP fornecerá duas interfaces: a primeira interface servirá para transferir os dados monitorados para o repositório de dados persistente, e a segunda interface será utilizada, em tempo real, pelo serviço do portal I&M para acesso aos dados monitorados. O usuário, em conformidade com as políticas de acesso do portal I&M, a qualquer momento, poderá recuperar os dados gerados pelo seu experimento.



Fonte: Pinheiro *et al.* (2012).

2.3 CONSIDERAÇÕES FINAIS

O presente capítulo tratou dos arcabouços de controle e monitoração juntamente com a rede para experimentação do projeto FIBRE. Nele foram abordados os tipos de arcabouços utilizados pelo projeto FIBRE, seus componentes e o armazenamento dos dados de monitoração, onde foi possível verificar que somente o CMF ProtoGENI possuía aplicação para o armazenamento de dados de monitoração. Na seção 2.2.1, foi apresentada a arquitetura de instrumentação e medição do FIBRE-BR, que dentre os componentes existentes em sua arquitetura possui o repositório de dados persistente, base para desenvolvimento desta dissertação.

No capítulo 3, serão abordados, os requisitos e soluções necessários para o armazenamento dos dados no contexto de redes para experimentação.

3 ARMAZENAMENTO DOS DADOS NO CONTEXTO DE REDES PARA EXPERIMENTAÇÃO

Neste capítulo, são apresentados os desafios para o armazenamento dos dados no contexto de redes para experimentação que envolve, dentre outros aspectos, as questões técnicas, a solução e as aplicações utilizadas para a solução. Ao final, é definida uma proposta para o armazenamento de dados no contexto do projeto FIBRE.

3.1 QUESTÕES TÉCNICAS

A decomposição das questões para o armazenamento dos dados de medição multi-CMF envolve vários requisitos técnicos como: armazenamento transparente (incluindo a semântica de dados e metadados), controle de acesso, segurança, distribuição de interface cliente, escalabilidade e federação. A seguir, serão discutidos os problemas existentes em cada um dos aspectos técnicos citado anteriormente.

3.1.1 Armazenamento transparente

De maneira geral, a transparência está em esconder a natureza dos recursos utilizados para a realização da tarefa dos usuários. Para que isso seja possível, uma série de funcionalidades será necessária para tornar o armazenamento e a consulta dos dados de experimentação mais simples para o experimentador, não exigindo do mesmo conhecimento avançado para o armazenamento dos dados de monitoração.

Visando criar um mecanismo de transparência para o ambiente multi-CMF alguns pontos deverão ser tratados, a saber: semântica dos dados, acesso, segurança, escalabilidade e disponibilização dos dados.

3.1.1.1 Semântica/Metadado

A semântica do dado, segundo (VASUDEVAN; PAZANDAK, 2013), é qualquer informação de alto nível relacionada ao arquivo, seja por definição (inclui a extensão do arquivo e seu tipo), associativa (palavras-chaves do arquivo que caracterizam seu conteúdo), estrutural (contém a organização lógica e física dos dados), comportamental (visualização e

modificação, gerenciamento de mudanças) ou de ambiente (contém informações a respeito do proprietário do arquivo, permissões, histórico de revisões, etc.).

Dentre as informações semânticas listadas por Vasudevan e Pazandak, este trabalho visa atender o problema estrutural e a inclusão de informações associativas ao dado de monitoração para o ambiente multi-CMF.

- Estrutural: corresponde à organização física e lógica do dado. O local onde o arquivo será armazenado fisicamente não corresponderá ao caminho lógico apresentado ao pesquisador, pois o arquivo poderá estar distribuído em diversas máquinas da ilha, não sendo trivial para o experimentador o acesso a este dado ilha-a-ilha, caso o mesmo possua arquivos espalhados pelo *testbed*;
- Associativa: descrição do dado. A inserção de metadados ao dado é um item a mais para o seu detalhamento, de forma a possibilitar ao experimentador descrever detalhes sobre o experimento. A ausência de informação na descrição do dado dificulta na identificação e possíveis pesquisas de trabalhos correlatos.

3.1.1.2 Controle de acesso

O acesso do experimentador ao *testbed* FIBRE estará atrelado à gestão de identidade e controle de acesso necessário para federação entre *testbeds* (FERNANDES *et al.*, 2013). Ou seja, a autenticação do usuário de um *testbed* servirá para autenticação e uso dos recursos de outro *testbed* federado, desde que sejam atendidas as políticas locais de cada uma das redes.

A gerência de identidade no projeto FIBRE tem a proposta de adotar o LDAP (*Lightweight Directory Access Protocol*) como o protocolo de autenticação para ilhas de maneira hierárquica, onde o NOC é o nó raiz e as ilhas manteriam suas próprias bases de acesso.

Um dos problemas para o armazenamento persistente consiste na comunicação com a base federada LDAP de forma a não necessitar que seja criado um usuário específico para o armazenamento dos dados.

3.1.1.3 Segurança

A segurança é fundamental para qualquer implantação de sistemas, por envolver os quatro principais atributos básicos, segundo os padrões internacionais (ISO/IEC 17799,2005) que são:

- **Confidencialidade:** fator importante para garantir que o acesso ao dado se aplique ao proprietário ou a quem for dada a permissão para tal, ou seja, um dado de um experimentador não deverá estar disponível para outro experimentador, a não ser que seja dada a este permissão de acesso ao dado ou este tenha se tornando público.
- **Integridade:** garante que o dado não sofra alterações nas suas características originais após o transporte para o repositório de dados.
- **Acessibilidade:** Está atrelado ao controle de acesso (item 3.1.1.2).
- **Disponibilidade:** Os dados, informações e recursos devem estar sempre disponíveis quando requisitados por aqueles que têm o direito de acesso.

No entanto, a disponibilidade e a acessibilidade são os mais relevantes para o armazenamento transparente dos dados, pois a falta de um deles afeta diretamente o acesso do experimentador ao dado armazenado ou até mesmo inviabiliza o seu armazenamento.

3.1.1.4 Escalabilidade

O sistema deverá ser escalável o suficiente para permanecer eficiente quando aumentar de forma significativa o número de recursos e de usuários. A falta de escalabilidade compromete a transparência de acesso ao dado e conseqüentemente a disponibilidade do sistema. Para isso, será necessário implantar um sistema que possibilite um crescimento horizontal de recurso físico (memória, processamento e espaço em disco) de forma a mitigar o risco de indisponibilidade do serviço de armazenamento não comprometendo o desempenho do sistema.

3.1.1.5 Distribuição de interface cliente

A distribuição de aplicações cliente é fundamental para fazer a interface entre as máquinas cliente e o servidor. As aplicações deverão ser capazes de realizar autenticação,

manipular os dados do experimentador de tal maneira que consigam inserir descrição do dado e metadados, podendo renomear, excluir, tornar o dado público, entre outros. Estas aplicações cliente ficarão nas máquinas de experimentação e deverá consumir o mínimo possível de recurso físico, memória, processador e disco.

3.1.2 Federação

A federação consiste, em resumo, numa estratégia de junção de várias instituições independentes distribuídas que cooperam entre si, de acordo com uma política de gestão pré-estabelecidas (PETERSON *et al.*, 2007).

No contexto da rede do projeto FIBRE, rede de experimentação composta por várias instituições e CMFs tanto no Brasil quanto na Europa, são desenvolvidas atividades objetivando a federação de seus recursos. Neste ambiente multi-institucional e com diferentes arcabouços de controle e monitoração, o armazenamento de dados federados deverá ser genericamente, compatível com os diversos CMFs das diversas ilhas, possibilitando atender a novas instituições ou *testbeds* que venham a serem acrescentadas ao projeto FIBRE. Diante deste cenário, convém que a aplicação para o armazenamento de dados de experimentação possua suporte à federação.

3.2 SOLUÇÕES TÉCNICAS

A solução técnica para o armazenamento de dados de experimentação está na adoção de uma ferramenta *OpenSource* que possibilite a gestão dos dados e sua movimentação para o armazenamento persistente, capaz de registrar o usuário/experimentador, suportar o protocolo LDAP para o *testbed* do FIBRE, ser compatível com os diversos sistemas de arquivo e possuir mecanismos de tolerância a falhas que torne o acesso ao dado mais transparente para o experimentador. Devido às ilhas estarem espalhadas em diversas instituições, a solução técnica deverá possuir suporte a sistemas de arquivos distribuídos e a federação, escalando assim a capacidade de expansão do *testbed*.

3.2.1 Tipos de armazenamento

Os tipos de armazenamento de dados podem ser classificados como centralizado ou distribuído.

- **Centralizado** – Também chamado cliente-servidor, este tipo de solução deverá dispor em uma das ilhas do projeto FIBRE, de um servidor para receber os dados dos experimentos. Na máquina do cliente, que terá o dado do experimento, deverá possuir uma aplicação cliente que irá fazer o processo de envio deste arquivo para o servidor.
Neste tipo de solução, o servidor deve ser uma máquina mais robusta visando atender às demandas de todos os integrantes do projeto FIBRE. A desvantagem deste tipo de solução está no gargalo no enlace da rede externa causado pelo crescimento natural do *testbed*, o que poderá tornar indisponível o serviço devido à sobrecarga na rede.
- **Distribuído** – A solução com sistema de arquivo distribuído tende a distribuir a carga em diversos servidores espalhados pelas ilhas do projeto FIBRE, podendo, em cada ilha, ter um ponto coletor dos dados, onde uma camada de abstração (aplicação para gestão dos dados) terá a visão da localização dos dados armazenados. Esta solução reduz a sobrecarga na rede externa, reduzindo o risco de indisponibilidade do serviço, mas, gera uma demanda maior para o administrador pela existência de um servidor a mais por ilha para ser gerenciado.

3.2.2 Armazenamento transparente

A transparência consiste em ocultar, para o experimentador, todo o arcabouço de componentes necessários para que se tenha a visão do sistema como um todo. Este arcabouço para a transparência aparece na literatura de sistemas distribuídos classificado em oito tipos, conforme o *ISO/ITU Reference Model for Open Distributed Processing (RM-ODP)* (RM-ODP, 2008).

- **Transparência de Acesso** – Esconde como os dados são acessados como, por exemplo, tipos de protocolos de comunicação entre os computadores distribuídos.

- Transparência de localização – O experimentador não precisa saber onde os dados estão fisicamente localizados.
- Transparência de falha – É a própria tolerância a falhas. Em caso de falha, o sistema deve possuir uma solução de contingência a fim de manter o sistema disponível.
- Transparência de migração – Os dados podem ser movidos dentro do sistema sem que os usuários percebam.
- Transparência de persistência – Permite que um sistema mova ou recupere dados da memória física ou do disco rígido, sem que o usuário perceba.
- Transparência de relocação – Mudança de local de um recurso é transparente aos experimentadores.
- Transparência de replicação – O experimentador não sabe quantas cópias de um arquivo foi realizado. É utilizado para dividir a carga no servidor e melhora a tolerância a falhas.
- Transparência de concorrência – Vários experimentadores acessam o mesmo arquivo sem que tenham conhecimento disso.

A norma ODP define a estrutura e funções para implantação da transparência, no entanto, a implantação de cada item de transparência impacta diretamente no custo da solução, assim, um sistema ODP deverá implantar a transparência de acordo com a relevância, não sendo obrigado a suportar todos os tipos.

Visando implantar a transparência dos dados no ambiente multi-CMF do projeto FIBRE, de forma a passar a impressão de um sistema único para o experimentador, os problemas abordados na seção 3.1 (semântica dos dados, acesso, segurança, escalabilidade e disponibilização do dado) deverão ser tratados em conformidade com o modelo de referência ODP.

3.2.2.1 Semântica/Metadado

A semântica para a disponibilização do dado em um ambiente multi-CMF deve atentar para alguns detalhes na implementação para que o acesso ao dado pesquisado seja simples para o experimentador. Assim ela deve:

- estar relacionada à transparência de localização. Seria a criação de um endereço lógico para o dado abstraindo o local físico, ou seja, o

experimentador irá verificar no seu *home* os dados armazenados dos diversos CMFs em um único local podendo seguir o padrão:

/testbed/home/usuárioldap/ilha-cmf-arquivo

Exemplo 1: Supondo o seguinte cenário: A experimentadora Aline faz parte do FIBRE, assim o seu repositório persistente é:

/FIBRE/home/aline/...

Onde o FIBRE é o *testbed* e o nome do diretório *aline* deverá corresponder com o nome do usuário ldap.

- contemplar a semântica do nome do arquivo. Isso é importante na identificação do dado do experimento no *home* do usuário, na nomenclatura deverão ser identificados a ilha, o CMF, e o nome do experimento conforme o exemplo 2.

Exemplo 2: Se Aline utiliza diferentes CMFs em diferentes ilhas, cada arquivo que for salvo terá um prefixo identificando o CMF e a ilha. Por exemplo, um resultado de um experimento OMF na ilha UNIFACS seria nomeado assim: UNIFACS-OMF-<nomedoarquivo>. O mesmo é verdadeiro para PGENI (ProtoGENI) e OCF. Assim, o esquema de nomenclatura final seria:

/FIBRE/home/aline/unifacs-omf-nomedoarquivo

- tratar a inserção de metadados (registro eletrônico que contém descrições que permitem saber do que se trata o arquivo sem precisar ler o seu conteúdo). O fato de possuir nomes de arquivos padronizados não é suficiente para descrever um dado, o que torna o uso de metadados importante para a identificação dos dados, facilitando a sua recuperação através da pesquisa. Nesse sentido, ao ser solicitado o armazenamento dos dados de experimentação, será preciso inserir os atributos CMF e ILHA. Para o atributo CMF os valores disponíveis serão: OMF, PGENI e OCF. Para o atributo ILHA o valor será o nome da ilha geradora do dado, ou seja, se o dado foi criado na ilha UNIFACS, o atributo ILHA terá como valor UNIFACS. Estes dois atributos informados serão atribuídos ao dado automaticamente, no entanto, deverá ser facultado ao experimentador inserir manualmente qualquer outro atributo e valor que achar necessário para descrever o dado do seu experimento.

3.2.2.2 Acesso

O acesso transparente para o armazenamento dos dados ao *testbed* do FIBRE está atrelado à gestão de identidade e controle de acesso federada existente. O experimentador, caso queira salvar os dados do experimento, utilizará sua credencial de acesso federada na máquina de experimentação, independentemente de qual ilha do projeto FIBRE esteja fazendo o experimento, enviando uma requisição de autenticação para o servidor de armazenamento, e este por sua vez, fará a consulta na LDAP do projeto FIBRE liberando ou não o acesso para o experimentador. Desta forma, as aplicações estudadas nesta dissertação deverão possuir suporte à comunicação ao protocolo LDAP, tornando possível a sua implantação no projeto FIBRE.

3.2.2.3 Segurança

A segurança é um tópico universal em computação, tanto para o armazenamento centralizado quanto para o armazenamento distribuído, que não deve ser negligenciada, uma vez que, sem segurança não haverá garantia do bom funcionamento do sistema como um todo. O sistema deverá suportar ao menos dois níveis de acesso, um de administrador e outro de usuário, gerando um maior controle de acesso aos dados armazenados por diversos experimentadores. O acesso ao dado se dará por quem o criou ou por quem tem permissão para tal.

A disponibilidade é tópico de segurança necessário para o acesso transparente aos dados de experimentos, a ideia é utilizar um sistema que possua mecanismos para tolerância a falhas onde na ocorrência de indisponibilidade de um servidor o outro assuma.

3.2.2.4 Escalabilidade

Este é outro fator que impacta na transparência. O sistema não deverá perder desempenho com o aumento dos dados ou crescimento da rede ou dos usuários. Para isso, propõe-se utilizar um sistema de arquivo distribuído nas ilhas, de forma a prover possibilidades de crescimento sem maiores impactos, evitando gargalos e melhorando o desempenho global.

3.2.2.5 Distribuição de interface cliente

A distribuição deve atender dois tipos de cliente: o primeiro é o cliente/*host*, ou seja, é a máquina de experimentação onde estará o dado de armazenamento. A ideia é que esta aplicação seja leve, e que consuma o mínimo possível de recursos computacionais (memória, processador e disco), de preferência que o mesmo fique desativado e seja acionado somente quando necessário. O segundo cliente deverá ser uma interface *Web*, em que o experimentador terá acesso aos dados, e que o possibilite inserir metadados, renomear, apagar, compartilhar e baixar o dado para seu computador local.

3.2.3 Federação

Implantar a federação é escalar o *testbed* ofertando novos serviços e interoperando os serviços disponíveis nas instituições de forma a simplificar a autenticação e autorização de usuários a recursos e aplicações entre domínios parceiros. Neste sentido, uma aplicação para o armazenamento persistente deverá prover soluções que contemple a federação.

3.3 ALTERNATIVAS TÉCNICAS

Para cada tipo de solução abordada na seção 3.2 e com base nos problemas listados na seção 3.1, foram avaliadas três aplicações que atendem ao quesito de sistemas de arquivos centralizados e distribuídos. Essas aplicações serviram de propostas para o projeto sendo analisada a existência das funcionalidades que visam reduzir os problemas citados para o armazenamento de dados com a finalidade de se escolher uma aplicação para o uso no projeto FIBRE. As aplicações objeto do estudo foram iRODS, MooseFS e o HDFS.

3.3.1 IRODS

O *Integrated Rule-Oriented Data System* (IRODS, 2013) é um *data grid* desenvolvido pela *Data Intensive Cyber Environments* (DICE, 2013) da Universidade da Carolina do Norte em Chapel Hill e o Instituto de Computação Neural da Universidade da Califórnia que é formado por colaboradores e um grupo de pesquisa do *Storage Resource Broke* (SRB, 2013), software com mais de uma década em gerência dos dados (IRODS, 2013).

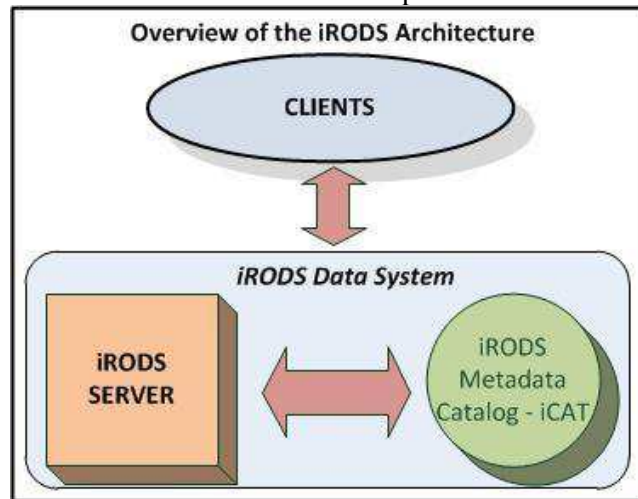
O iRODS é utilizado para gerenciamento de dados distribuídos, possibilitando a pesquisa colaborativa, organizando os dados, compartilhando recursos, protegendo e preservando os dados entre os usuários. Com base em uma estrutura hierárquica, é possível construir uma solução de armazenamento transparente (centralizada ou distribuída) na perspectiva do usuário final. A sua estrutura comporta diversos tipos de sistemas de arquivo como: *Unix File System*, *Windows File System*, *MSS Universal Drive*, *HPSS File System*, e sistemas distribuídos como o drive *Hadoop File System* “*HDFS File Driver*”.

3.3.1.1 Visão Geral

O iRODS é formado por três componentes principais: o servidor de catálogo iCAT (iRODS Metadata Catalog), o servidor de armazenamento (iRODS *Server*) e os clientes de acesso (iRODS *Clients*), conforme poderá ser verificado na Figura 7.

- a) iRODS *Metadata Catalog* (iCAT) é um servidor de catálogo que armazena informações de metadados descritivos em alguns SGBDs (Sistema de Gerenciamento de Banco de Dados), PostgreSQL, MySQL ou Oracle para todos os dados armazenados no servidor de dados iRODS.
- b) iRODS *Server* é um servidor que se responsabiliza pelo armazenamento dos dados, interagindo com o protocolo de acesso de um sistema de arquivos específico, permitindo o armazenamento e compartilhamento de dados distribuídos geograficamente e abrangendo domínios administrativos distintos.
- c) *Clients* são aplicações *Web* ou de linha de comando CLI formada por um conjunto de funcionalidades necessárias para acessar e gerenciar dados no servidor iRODS.

Figura Erro! Nenhum texto com o estilo especificado foi encontrado no documento.6 -
Visão Geral dos componentes iRODS



Fonte: Adaptado de Moore (2010).

O usuário, através de uma aplicação cliente, poderá realizar uma série de ações aos dados armazenados como: localizar, acessar, gerenciar um dado e metadados armazenado no servidor iRODS. Para que isso seja possível, a aplicação cliente solicita deste usuário as credenciais de acesso (login e senha) para então disponibilizar, conforme o nível de acesso, a permissão para manipulação dos dados. Todo o processo de gerência dos dados e metadados é realizado no servidor iCAT, que por sua vez, interage com o iRODS Server aplicando ações originadas do usuário.

Outra característica do iRODS é o conceito de "zona" e "federação". Uma zona é um sistema iRODS independente, formado por um servidor iRODS iCAT e clientes iRODS, podendo existir vários servidores iRODS não-iCAT ou simplesmente servidores iRODS. Ao interoperar duas ou mais zonas iRODS, através da relação de confiança mútua entre as zonas formam-se a federação iRODS, onde um usuário de uma zona poderá ter acesso a outra zona federada com as devidas permissões de acesso.

3.3.1.2 Semântica de dados e metadados

A seguir a forma como o iRODS aborda as questões semânticas e de metadados descritas na seção 3.1.1.1:

- Endereço Lógico: conhecido como *namespace*. Cada arquivo que é armazenado no iRODS possui tanto um endereço físico, local onde o arquivo está armazenado fisicamente no servidor, quanto um endereço lógico, caminho

lógico que o iRODS apresenta para o usuário, abstraindo onde o arquivo está armazenado fisicamente. Ao utilizar o comando “ils”, através do cliente iRODS *icommands*, o usuário irá verificar o endereço lógico do seu dado. Caso queira saber onde este dado está fisicamente, o mesmo poderá executar o comando “ils -L”.

- Inserção de metadados: os metadados no iRODS são gerenciados por banco de dados (PostgreSQL, MySQL, Oracle). São formados por três atributos (AVU), Atributo de Nome, Atributo de Valor e o Atributo opcional de Unidade (SCHROEDER, 2009). É possível em uma aplicação cliente o usuário inserir, remover e localizar um metadados de um arquivo.

3.3.1.3 Controle de acesso

O iRODS tem suporte a vários tipos de autenticação como: GSI, *Kerberos*, *Challenge-response* ou *Shibboleth*. Ele possui um módulo de autenticação PAM (*Pluggable Authentication Module*), disponível a partir da versão 3.2 do iRODS e compatível com vários sistemas operacionais e interage com vários sistemas de autenticação, inclusive o LDAP (SCHROEDER, 2013).

3.3.1.4 Segurança

Existem dois tipos de regras de acesso, a regra de sistema e a regra de usuário. A regra de sistema é o nível de acesso para o administrador iRODS, cujo perfil é “rodsadmin” e serve para gerenciar as políticas do sistema, seja para autenticação, integridade, restrições de acesso, replicação ou atribuição de metadados. A regra de usuário é restrita a atividade deste usuário e possui o perfil “rodsuser” que permite a gestão sobre os dados de sua propriedade, como, por exemplo, inserir um metadados, inserir um comentário, apagar o dado ou torná-lo público, entre outros. As regras normalmente são solicitações feitas pelo usuário que ao chegar ao servidor dispara uma série de operações (micro-serviços⁶) que são realizadas em nome do usuário (TOOBY, 2008).

Além dos níveis de acesso e controle de acesso que são itens de segurança citados no item 3.3.1.3, o iRODS oferece o serviço de alta disponibilidade replicando a sua base dados

⁶ Micro-serviços são pequenos procedimentos ou funções que executam uma determinada tarefa.

entre servidores iRODS utilizando o PgPool (*procedure* que replica a base de dados iCAT entre dois servidores) (REDDAPANI, 2009).

3.3.1.5 Escalabilidade

É possível escalar os serviços do iRODS, em uma mesma zona, dividindo a carga de forma distribuída, onde se tem um servidor de metadados iCAT, também chamado de *Master*, e vários servidores *slaves*, responsáveis pelo armazenamento dos dados. O objetivo dos servidores *slave* é escalar o armazenamento com vários servidores distribuídos na zona para não sobrecarregar o servidor de catálogo.

3.3.1.6 Distribuição de interface cliente

O iRODS possui vários clientes desenvolvidos em diversas plataformas (Windows, Linux e Mac). Os principais são:

- iCOMMANDS, utilizado via linha de comando compatível com Linux, Windows e Mac;
- iRODS *Web Browser*, aplicação *Web* desenvolvida em PHP de fácil navegação que utiliza a interface “explorer-like”. Essa aplicação realiza consulta aos dados e metadados, faz *upload/downloads* dos dados e insere metadados;
- iDROP *Web* desenvolvido em JAVA, tem funções similares ao do iRODS *Web Browser* com uma diferença na transferência de arquivos em massa por utilizar um *applet* iDROP-lite. No entanto, a consulta aos dados está limitada ao nome do arquivo “*tags*” não sendo possível realizar consulta por metadados.

3.3.1.7 Federação

Como visto anteriormente, a união de duas ou mais zonas iRODS, com as devidas permissões de acesso forma a federação iRODS. Isso implica que um usuário pertencente a uma instituição poderá utilizar os recursos de outra instituição com as credenciais de acesso da sua instituição local. Em termos de configuração é necessário registrar na zona local a zona remota, registrar o usuário como um usuário remoto e conceder-lhe as permissões de acesso.

3.3.2 MOOSEFS

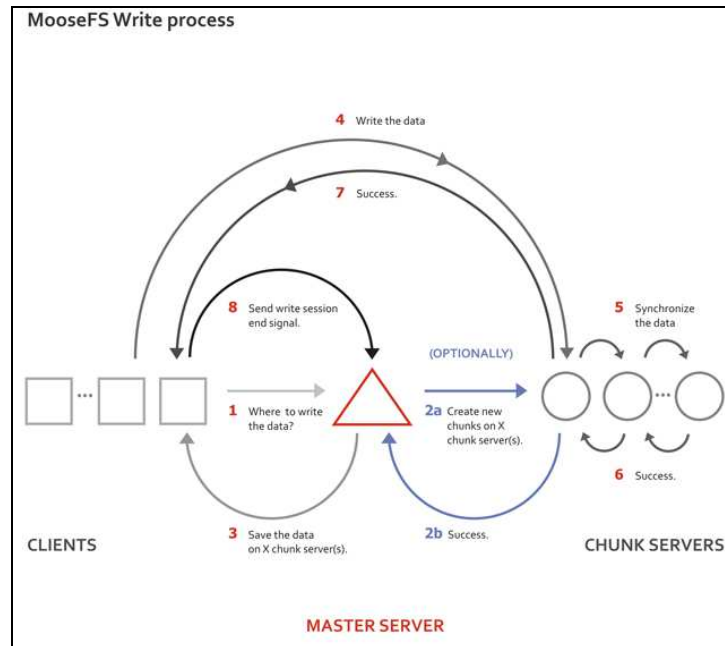
O *System File Moose* (MooseFS, 2013) é um sistema de arquivos distribuído, *Open Source* desenvolvido pelo *Gemius S.A* que se caracteriza pela escalabilidade e pela tolerância a falhas (MOOSEFS, 2013a).

3.3.2.1 Visão Geral

O MooseFS é formado por quatro componentes principais: o servidor de gerência, *Master Server*; os servidores de dados, *Chunk Servers*; o servidor de *backup* dos metadados, *Metalogger Server*; e os clientes, conforme poderá ser verificado na Figura 8.

- Servidor de Gerência – O *Master Server* é um único servidor responsável pela gestão dos dados e metadados do sistema de arquivo. Ele detém a informação de localização do dado, tamanho, atributos, entre outros. Está representado na Figura 8 em forma de triângulo.
- Servidor de Dados – O *Chunk Server* é responsável pelo armazenamento e sincronização dos dados com outros servidores de dados. Está representado na Figura 8 em forma de círculo.
- Servidor de *backup* dos metadados – O *Metalogger Server* é responsável por registrar todas as mudanças no arquivo de metadados do servidor *master* de forma a manter a disponibilidade do serviço. Caso haja uma parada no servidor *master*, o *Metalogger Server* passa a ter o papel do servidor de gerência.
- Cliente – São os usuários clientes que, em uma máquina, montam a sua estrutura de armazenamento no MooseFS, através do *mfsmount*. Está representado na Figura 8 em forma de quadrado.

Figura **Erro! Nenhum texto com o estilo especificado foi encontrado no documento.7** -
Visão Geral dos componentes MooseFS



Fonte: Moosefs (2013).

Os usuários clientes, para terem acesso à estrutura de armazenamento no MooseFS, terão que utilizar o cliente *mfsmount* que fará a comunicação com o servidor *master*, responsável pelo gerenciamento dos metadados. Em seguida, com o servidor de dados ou *chunk servers*, que replicará os dados entre os n servidores *chunk servers* definidos pelo servidor *master*, tornando o sistema de armazenamento de dados mais disponível.

3.3.2.2 Semântica de dados e metadados

A seguir como o MooseFS aborda as questões semânticas e de metadados descritas na seção 3.1.1.1:

- Endereço Lógico: MooseFS gerencia de forma hierárquica o endereço lógico ou *namespace* de diretórios na memória principal do servidor *master*, disponibilizando ao cliente um endereço para o sistema, onde este terá acesso aos dados armazenados nos *chunk servers*.
- Inserção de metadados é realizada somente ao nível de sistema onde o servidor *master* cataloga a descrição dos dados que estão distribuídos em todos os servidores *Chunk*.
-

3.3.2.3 Controle de acesso

O MooseFS não possui mecanismos de criação de usuário ou grupos. Ao utilizar este sistema de arquivo o controle de acesso deverá ser realizado por sistemas externos como é o caso do sistema operacional.

3.3.2.4 Segurança

A disponibilidade do sistema de arquivo MooseFS está na replicação dos Servidores de Dados e na cópia do Servidor de Gerência para o Servidor Metalogger, que tem a função de armazenar os registros de metadados. O servidor Metalogger envia de tempos em tempos arquivos para o servidor de gerência de modo a garantir que caso este falhe, o mesmo assumirá o seu lugar.

3.3.2.5 Escalabilidade

O MooseFS é um sistema escalável por ser distribuído. O crescimento da demanda impacta na criação de novos Servidores de dados ou *Chunk Servers* distribuindo a carga dentro da rede MooseFS. O Servidor *Master* por armazenar os metadados em memória, conseqüentemente, deverá possuir mais memória RAM para que possa armazenar um número maior de registros.

3.3.2.6 Distribuição de interface cliente

O MooseFS disponibiliza um servidor HTTP muito simples escrito apenas para executar os *scripts CGI*⁷ chamado *Mfscgiserv*, no entanto não suporta todos os recursos adicionais como autenticação HTTP (MOOSEFS, 2013b). Na máquina cliente através do cliente *mfsmount* é possível montar uma partição no sistema de arquivo MooseFS *master* desde que o cliente suporte o módulo FUSE (*Filesystem in Userspace*) mecanismo de linha de comando do Unix.

⁷ CGI- Pequenos programas que interpretam parâmetros e geram a página depois de os processar.

3.3.2.7 Federação

O MooseFS não suporta federação.

3.3.3 HDFS

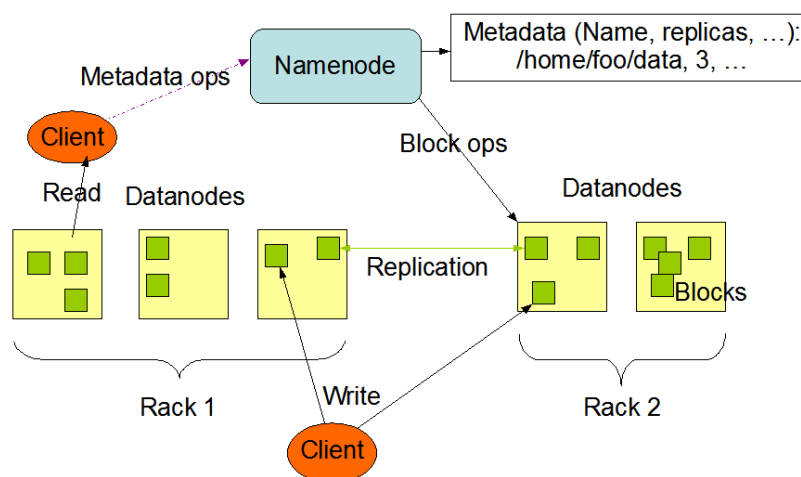
O *Hadoop Distributed File System* (HDFS, 2013) é um sistema de arquivo distribuído e tolerante a falhas, desenvolvido inicialmente, para um projeto de busca do *Apache Nutch* e, hoje, faz parte de um subprojeto do *Apache Hadoop*. O HDFS é um projeto mais voltado para processamento em lote do que para interação com o usuário (HADOOP, 2013a).

3.3.3.1 Visão Geral

O HDFS é formado, basicamente, por dois componentes (*Namenode* e *Datanode*) e os clientes, conforme, poderá ser verificado na Figura 9.

- *Namenodes*: É um servidor responsável por gerenciar os dados e metadados dos clientes, estruturas de nomes, réplicas entre os *Datanodes*.
- *Datanodes*: São os responsáveis para atender a demanda de leitura e gravação de um dado do cliente, além de fazer replicação sob demanda dos *Namenodes*.

Figura 8 - Visão Geral dos componentes HDFS
HDFS Architecture



Fonte: Hadoop (2013).

Um arquivo do cliente é dividido em vários blocos e replicado em diversos *Datanodes*. O conjunto de *Datanodes* é chamado de *Rack*. Assim, um *Namenode* com vários *Datanodes* formam um cluster HDFS (HADOOP, 2013b).

3.3.3.2 Semântica de dados e metadados

A seguir a forma como o HDFS aborda as questões semânticas e de metadados, descritas na seção 3.1.1.1:

- Endereço Lógico: ou *namespace*, local onde será armazenado o dado de um usuário. É organizado de forma hierárquica pelo sistema HDFS, onde o usuário pode criar, apagar e renomear diretório dentro do seu *namespace*. O *namespace* mantém o mesmo esquema de nome do sistema de arquivo pelo *Namenode*, onde se registra qualquer alteração no nome.
- Inserção de metadados: o *Namenode* registra, persistentemente, toda transação de criação de um dado ou alteração de metadados em um arquivo chamado *EditLog*. De forma semelhante todos os arquivos do sistema *namespace* e o mapa dos blocos dos arquivos são armazenados em um arquivo chamado *FsImage* localmente no sistema de arquivos do *Namenodes*. O *Namenodes* mantém uma imagem de todo o *namespace* e o mapa de bloco na memória, assim que o sistema é inicializado.

3.3.3.3 Controle de acesso

A identificação do usuário no *Hadoop* possui dois módulos de identificação do usuário, o *simple* e o *kerberos*. No *simple* o processo de identificação do cliente está limitado apenas ao usuário autenticado no sistema operacional, equivalente ao “*whoami*”. Recentemente, foi disponibilizada na versão 2.2.0, o módulo para o *Kerberos*, onde a identidade de um processo cliente é determinada por suas credenciais. Há pretensão de ser desenvolvida a gestão de identidade também com o LDAP. A gerência de usuário é externa ao HDFS, pois não possui mecanismos de criação de usuário ou grupos implementados (HADOOP, 2013a).

3.3.3.4 Segurança

Disponibilidade: *Namenode* é um ponto único de falha para um cluster HDFS. Se a máquina *Namenode* falhar, é necessária uma intervenção manual. Atualmente, não há suporte para o reinício automático e *failover* do software *Namenode* para outra máquina (HADOOP, 2013b).

3.3.3.5 Escalabilidade

O HDFS é um sistema altamente escalável. O crescimento da demanda impacta na criação de novos *Datanodes*, distribuindo a carga dentro do cluster HDFS administrado pelo *Namenode* que deverá possuir mais memória RAM para que possa armazenar um número maior de registro do *FsImage*.

3.3.3.6 Distribuição de interface cliente

Tipos de clientes: FS Shell, cliente HDFS em linha de comando utilizado para os clientes terem acessos aos seus dados, é voltado para desenvolvimento de *script* para interação com os dados. A sintaxe do comando inicia com “dfs –comando” de acesso limitado e o “dfsadmin –comando” para administração do HDFS cluster. Não foi encontrado na lista de comandos do HDFS nenhum que implemente a inserção de metadados.⁸

Para o acesso externo é disponibilizada uma aplicação *Web* que acompanha a instalação do HDFS, onde o usuário poderá visualizar o conteúdo de seus arquivos utilizando o navegador *Web*.

3.3.3.7 Federação

A federação HDFS é a união de dois ou mais *Namenodes/namespaces* independentes, onde os *Datanodes* são utilizados para armazenar os blocos de todos os *Namenodes*, para isso ele registra todos os *Namenodes* do *cluster* federado (HADOOP, 2013c).

⁸ Disponível em: <<http://developer.yahoo.com/hadoop/tutorial/module2.html#admincommandref>>. Acesso em: 1 Outubro 2013, às 23:41h

3.4 AVALIAÇÃO E DEFINIÇÃO DA TECNOLOGIA DE ARMAZENAMENTO DE DADOS

A proposta escolhida para o armazenamento dos dados multi-CMF visa resolver ou reduzir os problemas abordados na seção 3.1, a partir da análise das três tecnologias apresentadas na seção 3.3, tendo como base os critérios adotados: semântica de dados e metadados, controle de acesso, segurança, escalabilidade, distribuição e federação.

Semântica de Dados e Metadados

De acordo com as questões semânticas, semântico estrutural (contém a organização lógica e física dos dados) e semântica associativa (descrição dos dados ou metadados), listadas por Vasudevan e Pazandak (2013) e descritas na seção 3.1.1.1, foram analisadas três propostas a fim de identificar o suporte a estas questões.

Em relação à semântica estrutural, foi identificada a existência do endereço lógico (*namespace*), nas três propostas, abstraindo o endereço físico de onde os dados se encontram armazenados, conforme pode ser verificado na seção 3.3. A questão semântica associativa é apresentada nas propostas da seguinte forma:

- o MooseFS, possui metadados a nível de sistema onde o servidor *Master* cataloga a descrição dos dados que estão distribuídos em todos os servidores *Chunk*. No entanto, não possui a funcionalidade de inserção de um metadados pelo usuário;
- o HDFS, também utiliza a inserção de metadados a nível de sistema, registrando toda transação de criação de um dado ou alteração de metadados em um arquivo texto chamado EditLog. O conteúdo deste arquivo é carregado na memória quando o servidor Namenodes é inicializado, porém, não possui a funcionalidade de inserção de uma descrição dos dados pelo usuário;
- o iRODS, possui a inserção dos metadados tanto pelo sistema quanto pelo usuário. Estes são gerenciados por banco de dados (PostgreSQL, MySQL, Oracle) e possui três tipos de atributos: Atributo de Nome, Atributo de Valor e o Atributo opcional de Unidade, que o iRODS denomina de AVU (SCHROEDER, 2009). É possível através de uma aplicação cliente iRODS permitir que o usuário insira, remova e localize metadados.

Controle de acesso

O projeto FIBRE utiliza o protocolo LDAP para controle de acesso dos usuários experimentadores à rede de experimentação e aos CMFs, no entanto, as máquinas de experimentação alocadas ao experimentador pelo CMF, ainda não exigem autenticação para o seu manuseio ou teste. Diante deste cenário, ao analisar os sistemas propostos verificou-se que:

- o MooseFS, não possui módulos de controle de acesso, deixando este mecanismo a critério do sistema operacional ou de aplicações externas;
- o HDFS, apesar de possuir dois módulos de identificação do usuário, o *simple* (processo de identificação do cliente, limitado apenas ao usuário autenticado no sistema operacional) e o kerberos (a identidade de um processo cliente é determinada por suas credenciais) ainda não possui módulo para gestão de identidade para o protocolo LDAP, apesar de já existir a pretensão de seu desenvolvimento (HADOOP, 2013a). A gerência de usuário é externa ao HDFS, pois ele não possui mecanismos de criação de usuário ou grupos implementados;
- o iRODS, tem suporte a vários tipos de módulos de autenticação dentre eles estão o GSI, Kerberos, Challenge-response e Shibboleth. A compatibilidade com o protocolo LDAP é possível ao utilizar o módulo de autenticação PAM, disponível para vários sistemas operacionais (SCHROEDER, 2013). Para o seu funcionamento deverá existir uma aplicação cliente instalada na máquina de experimentação alocada pelo CMF, onde fará a autenticação deste experimentador consultando a base de usuários através do protocolo LDAP.

Segurança

As questões de segurança abordadas na seção 3.1.1.3, como acessibilidade e disponibilidade são as mais relevantes para o armazenamento transparente dos dados, pois a falta de uma delas afeta diretamente o acesso do experimentador ao dado armazenado e, até mesmo, inviabiliza o seu armazenamento. Quanto à acessibilidade foi possível observar que:

- o MooseFS e o HDFS não possuem a função de níveis de acesso, este controle é realizado por sistemas externos, como é o caso do sistema operacional que gerencia as contas e perfis de usuário de forma a saber quem é administrador ou o usuário com acesso restrito;

- o iRODS utiliza dois tipos de níveis de acesso, o de administrador e o de usuário. O nível de administrador (rodsadmin) é responsável por gerenciar as políticas do sistema, seja para autenticação, integridade, restrições de acesso, replicação e/ou atribuição de metadados. O segundo nível, o de usuário, possui o perfil “rodsuser” e é restrito à atividade deste usuário, sendo permitida a gestão sobre os dados de sua propriedade, como inserir um metadados, inserir um comentário, apagar o dado, ou torná-lo público, entre outros. Os níveis de acesso são controlados pelo servidor de metadados do iRODS (iCAT) que ao receber a solicitação de um cliente iRODS irá verificar quem foi o usuário solicitante para, em seguida, proceder de acordo com as permissões de acesso.

Em relação à disponibilidade, fundamental para o acesso transparente aos dados de experimentação, as propostas tratam este item de segurança da seguinte forma:

- o MooseFS atua na replicação dos dados entre os servidores de dados, e na cópia do Servidor de Gerência para o Servidor Metalogger, local onde consta as informações dos metadados;
- o HDFS cuida da disponibilidade dos datanodes ao replicar os blocos dos dados em diversos servidores, que em caso de interrupção em um dos servidores de dados outro servidor assume, automaticamente, esta função. No entanto, em seu servidor de gerência, tem o namenode, como seu único ponto de falha por não possuir um reinício automático e/ou *failover* do software para outra máquina. Assim, na ocorrência de uma interrupção no servidor torna-se necessária uma intervenção manual para restabelecer os serviços no outro servidor Metalogger (HADOOP, 2013b);
- no iRODS os dados dos experimentos poderão ser replicados entre servidores, desde que isto seja configurado previamente pelo administrador do iRODS. No entanto, o serviço de alta disponibilidade do servidor de metadados ainda não funciona de forma automática. Já existe o processo de replicação da base de dados entre servidores iRODS utilizando o PgPool (procedure que replica a base de dados iCAT entre dois servidores) (REDDAPANI, 2009), mas é necessária uma intervenção manual no servidor para restabelecer os serviços.

Escalabilidade

O sistema deverá ser escalável o suficiente para atender o crescimento da rede de experimentação mantendo a qualidade do serviço de armazenamento dos dados. Assim, ao analisar as três propostas, todas se mostram altamente escaláveis por se tratarem de sistemas distribuídos, repartindo a carga de processamento, memória e I/O de disco entre as diversas máquinas da rede.

- No MooseFS, o crescimento da demanda impacta na criação de novos Servidores de dados ou Chunk Servers distribuindo a carga dentro da rede MooseFS. Ele armazena no servidor *master* os metadados dos dados na memória, o que se torna um problema por exigir uma capacidade maior de memória RAM ao escalar a rede.
- O HDFS atua de forma semelhante ao MooseFS armazenando um número maior de registro do FsImage na memória, ao expandir a rede com a criação de novos Datanodes.
- O iRODS distribui a carga entre vários servidores *slaves* (iRODS *server*), escalando o armazenamento para não sobrecarregar o servidor de catálogo de dados (iCAT).

Distribuição de interface cliente

Concentra-se nas aplicações cliente apropriadas para realizar a interface entre as máquinas cliente e o servidor. As aplicações deverão ser capazes de realizar autenticação e manipulação dos dados do experimentador de tal maneira que consigam inserir a descrição do dado e metadados, podendo renomear, excluir, tornar o dado público, entre outros. Estas aplicações clientes ficarão nas máquinas de experimentação e deverão consumir o mínimo possível de recurso físico (memória, processador e disco).

- O MooseFS disponibiliza uma aplicação chamada de *Mfscgiserv*, que é um servidor HTTP muito simples desenvolvida apenas para executar os *scripts*, porém não suporta todos os recursos adicionais como autenticação HTTP (MOOSEFS, 2013b). Nas máquinas de experimentação é possível, através do comando *mfsmount*, montar uma partição no servidor *master*, desde que este possua mecanismo de linha de comando do Unix.
- O HDFS possui o cliente em linha de comando FS Shell, que dispõe de comandos com acesso limitado como o “*dfs –comando*” e comandos para

administração, como o “dfsadmin –comando” para gerência do *cluster* DHFS. Entre os comandos existentes nesta aplicação cliente, não foi encontrado nenhum que insira metadados. Para o acesso externo, é disponibilizada uma aplicação *Web* que acompanha a sua instalação, onde o usuário poderá visualizar o conteúdo de seus arquivos utilizando o navegador *Web*.

- O iRODS possui vários clientes desenvolvidos em diversas plataformas (Windows, Linux e Mac). Os principais são: iCOMMANDS, utilizado via linha de comando; iRODS *Web Browser* e o iDROP *Web*, aplicações *Web* desenvolvidas em PHP e JAVA, respectivamente. Estas ferramentas são utilizadas não somente para consulta dos dados, mas também para fazer *upload* e *download* dos dados, inserir metadados, etc..

Federação

O suporte a federação é necessário para o crescimento do *testbed*, possibilitando atender a novas instituições ou *testbeds* que venham a ser acrescidas ao projeto FIBRE compartilhando recursos entre si.

- MooseFS, não tem suporte a federação.
- O HDFS suporta a federação ao unir dois ou mais Namenodes/namespace independentes todos registrados em um cluster federado (HADOOP, 2013c).
- O iRODS suporta a federação com a união de duas ou mais zonas iRODS. Em termos de configuração, é necessário registrar a zona remota na zona local, registrar o usuário com um usuário remoto e conceder as permissões de acesso a este usuário.

Ao utilizar banco de dados para a gerência dos metadados, não necessitando carregar em memória RAM todos os metadados existentes no arquivo e, também, por possibilitar ao usuário a inserção de metadados ao dado, o iRODS respondeu melhor a questão da semântica de dados e metadados.

Em relação ao controle de acesso, a proposta do iRODS se mostrou mais adequada por não depender de sistemas externos para controlar o acesso dos usuários e por possuir um módulo para autenticação com o protocolo LDAP. Ele implementa dois níveis de acesso interno na sua aplicação, não sendo necessária interferência externa para este controle.

Mesmo que, quanto ao critério da disponibilidade, o iRODS não tenha sido a aplicação melhor avaliada por não possuir mecanismos automáticos para o restabelecimento do serviço, após a análise dos critérios avaliados nesta seção, nas três propostas, conclui-se que o iRODS apresentou em seus componentes e funcionalidades uma melhor flexibilidade para atender os requisitos de implantação no processo de armazenamento dos dados no FIBRE I&M.

3.5 CONSIDERAÇÕES FINAIS

Foram apresentados, neste capítulo, os aspectos técnicos para o armazenamento dos dados no contexto de redes para experimentação que envolveu, dentre outras coisas, as questões técnicas, solução e as aplicações utilizadas para a solução. E, por fim, foi definida iRODS como proposta para o armazenamento de dados no contexto do projeto FIBRE seguindo os critérios/requisitos analisados para cada aplicação, conforme poderá ser verificado no Quadro 1, o resumo comparativo entre as alternativas técnicas abordada na seção 3.4 e que serve de base, no capítulo 4, para o desenvolvimento das estratégias de uso do iRODS no contexto da rede experimentação.

Quadro 1 - Resumo comparativo entre as alternativas técnicas

REQUISITOS	iRODS	MOOSEFS	HDFS
Semântica de Dados e Metadados	<ul style="list-style-type: none"> • Existência do endereço lógico. • Possui metadados ao nível de sistema e de usuário. 	<ul style="list-style-type: none"> • Existência do endereço lógico. • Possui metadados ao nível de sistema. 	<ul style="list-style-type: none"> • Existência do endereço lógico. • Possui metadados ao nível de sistema.
Controle de acesso	<ul style="list-style-type: none"> • Suporte a módulos de autenticação dentre eles estão o GSI, <i>Kerberos</i> e <i>Shibboleth</i>. • Suporte ao protocolo LDAP utilizando o PAM. 	<ul style="list-style-type: none"> • Fica a critério do sistema operacional ou de aplicações externas. • Não tem suporte ao protocolo LDAP. 	<ul style="list-style-type: none"> • Possui dois módulos de identificação do usuário, o <i>simple</i> e o <i>kerberos</i>. • Não tem suporte ao protocolo LDAP.
Segurança	<ul style="list-style-type: none"> • Utiliza dois tipos de níveis de acesso, o de administrador (<i>rodsadmin</i>) e o de usuário (<i>rodsuser</i>) • Possui mecanismos de replicação dos dados entre os servidores de dados, porém, para o servidor de metadados, não possui restabelecimento do serviço de forma automática. 	<ul style="list-style-type: none"> • Não trata níveis de acesso, este controle é realizado por sistemas externos como é o caso do sistema operacional. • Possui replicação dos dados entre os servidores de dados, e na cópia do Servidor de Gerência para o Servidor <i>Metalogger</i>. 	<ul style="list-style-type: none"> • Não trata níveis de acesso, este controle é realizado por sistemas externos como é o caso do sistema operacional. • Replica os blocos dos dados em diversos servidores, porém não possui restabelecimento do serviço de forma automática no servidor <i>namenodes</i>.
Escalabilidade	<ul style="list-style-type: none"> • São escaláveis. • Os metadados são armazenados em banco de dados . 	<ul style="list-style-type: none"> • São escaláveis. • Carrega na memória RAM os metadados dos dados. 	<ul style="list-style-type: none"> • São escaláveis. • Carrega na memória RAM os metadados dos dados.
Distribuição de interface cliente	<ul style="list-style-type: none"> • iCOMMANDS • iRODS Web Browser • iDROP Web 	<ul style="list-style-type: none"> • Mfscgiserv • Mfsmount 	<ul style="list-style-type: none"> • FS Shell
Federação	<ul style="list-style-type: none"> • Possui suporte. 	<ul style="list-style-type: none"> • Não possui suporte. 	<ul style="list-style-type: none"> • Possui Suporte.

Fonte: Autoria própria (2013).

4 PROPOSTA PARA O ARMAZENAMENTO DE DADOS DE EXPERIMENTOS EM UM CENÁRIO MULTI-CMF

Este capítulo apresenta o armazenamento de dados baseado no iRODS para o projeto FIBRE, abordando os tipos de estratégias, a implantação dos seus componentes e o uso de acréscimos de metadados associados as facilidades do IRODS. Por fim, é apresentada a aplicação AmazenaDados utilizada para simplificar as configurações de ambiente da máquina de experimento facilitando o envio dos dados de experimentos por parte do experimentador.

4.1 ARMAZENAMENTO DE DADOS BASEADO NO IRODS PARA O PROJETO FIBRE

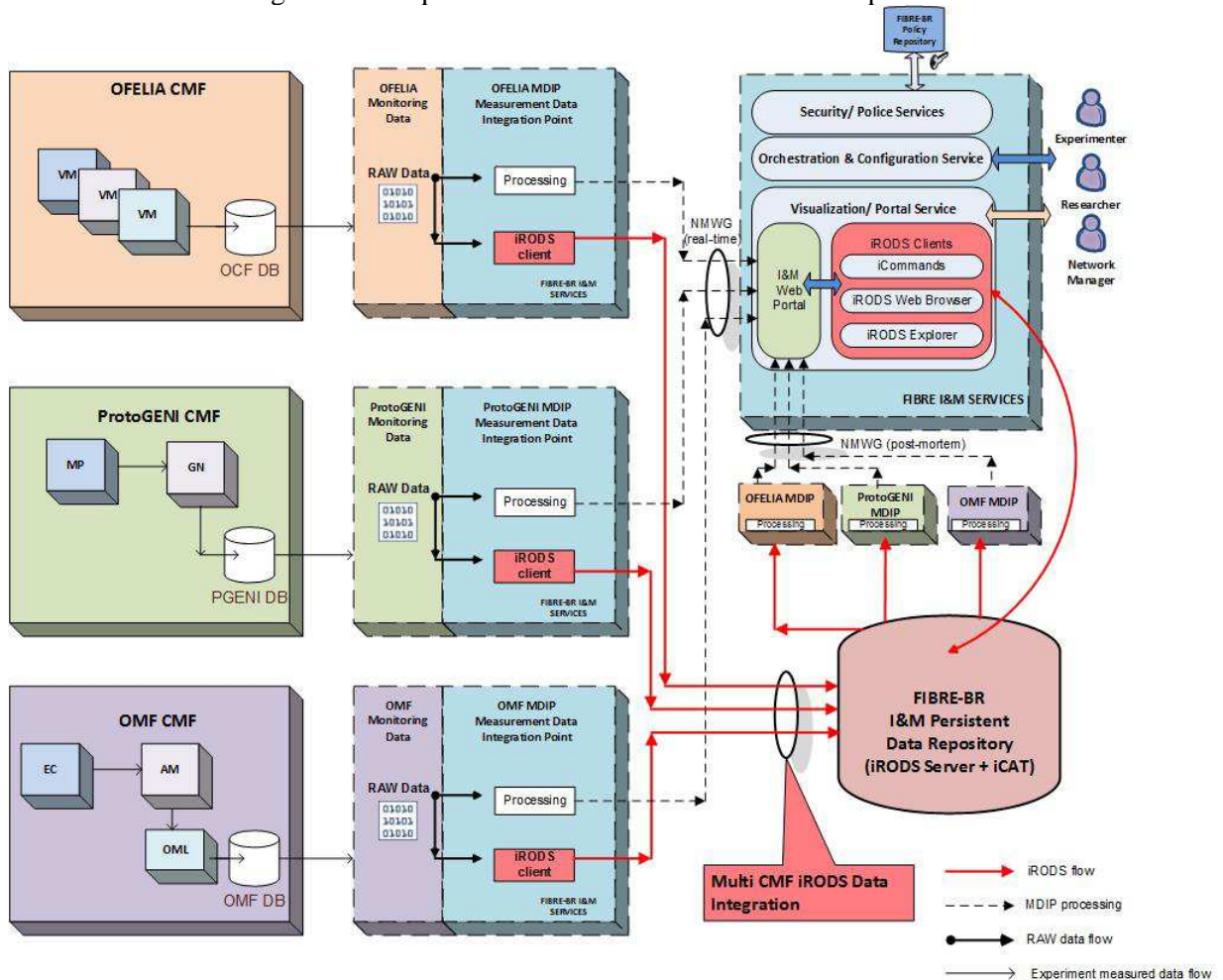
Com a definição do iRODS para a gerência dos dados e metadados de experimentação, originados dos múltiplos CMFs das ilhas participantes do projeto FIBRE, esta seção abordará a estratégia para o armazenamento dos dados utilizando esta aplicação.

A solução de armazenamento de dados baseada no iRODS deve estar em conformidade com a arquitetura do projeto FIBRE-BR I&M (seção 2.2.1), que é composta por diversos CMFs, onde o experimentador poderá ter acesso a parte dos recursos de infraestrutura (agregados) deste CMF chamados de fatia (*slice*). Os dados provenientes da monitoração dos experimentos em cada CMF deverão ser acessados *online* ou devem ser armazenados em um repositório de dados persistente. Uma interface, o MDIP, fará o gerenciamento e a integração do dado para um formato padronizado, o NMWG, possibilitando o acesso *online* do resultado dos experimentos e o armazenamento local dos dados resultantes do experimento.

Diante deste cenário, foram inseridos nesta arquitetura de I&M os componentes iRODS, na visão de uma ilha, para o serviço de armazenamento dos dados no contexto do projeto FIBRE (Figura 10). As máquinas de experimentação e o MDIP possuirão um cliente iRODS com a função de autenticar o usuário experimentador e enviar dados e metadados gerados pela monitoração. O envio dos dados será para o servidor de armazenamento de dados, iRODS *Server*. Para o experimentador ter acesso ao dado armazenado, nesta proposta, o mesmo fará uso de uma aplicação cliente *Web* para o iRODS, como interface cliente de gerenciamento dos dados podendo realizar *download* do dado do experimento, inserir metadados, deletar o dado e até mesmo compartilhar o dado. Detalhes da definição e

implantação dos clientes e servidores iRODS no contexto do projeto FIBRE será apresentada posteriormente ainda neste capítulo.

Figura 10 - Arquitetura FIBRE-BR I&M com os componentes iRODS



Fonte: Adaptado de Pinheiro *et al.* (2012).

A Figura 10 representa, do ponto de vista dos CMFs, as localizações onde ficarão situados os clientes iRODS em uma ilha do FIBRE e o envio dos dados dos experimentos pelos clientes iRODS para o repositório de dados persistente, porém não apresenta a visão da infraestrutura macro de armazenamento com várias ilhas. Neste sentido, foram analisados três tipos de estratégia iRODS para o projeto FIBRE, possíveis de serem implantados como solução de armazenamento de dados.

4.2 ESTRATÉGIAS IRODS PARA O PROJETO FIBRE

As estratégias abaixo, estão relacionadas às formas possíveis de uso dos componentes iRODS no contexto do projeto FIBRE.

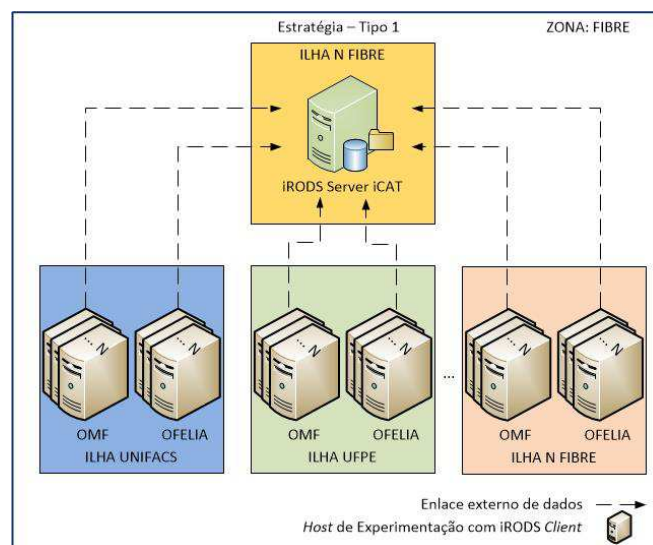
4.2.1 Estratégia Tipo 1

A primeira estratégia utiliza o armazenamento dos dados centralizado. Assim, tem-se o servidor central (iRODS SERVER – iCAT), responsável pelo controle de acesso, armazenamento dos dados e metadados, e os clientes iRODS, instalados nos computadores de experimentação independentemente da ilha. Todos estes, clientes e servidor, fazem parte de uma mesma zona iRODS.

O servidor iRODS poderá ser instalado em qualquer uma das ilhas participantes do *testbed* FIBRE, ficando esta ilha responsável pela gerência do servidor, a fim de manter a disponibilidade do serviço de armazenamento dos dados. Todas as ilhas fariam parte de uma única zona chamada FIBRE.

A Figura 11 representa esta estratégia. Cada ilha do projeto FIBRE possui suas máquinas de experimentação com o cliente iRODS instalado que, por sua vez, se comunica com o servidor iRODS através de um enlace externo, conforme verificado na seção 2.2, rede para experimentação do projeto FIBRE.

Figura 11- Estratégia de Armazenamento - Tipo 1



Fonte: Aatoria própria (2013).

Como proposta para o projeto FIBRE, esta estratégia possui vantagens e desvantagens. A vantagem está na sua simplicidade de implantação, pois se tem um servidor iCAT, responsável por gerenciar os metadados e o armazenamento dos arquivos de experimentação, e as aplicações clientes, que ficam nas máquinas de experimentação, todos configurados para enviar os dados para um único servidor. A desvantagem é possuir um único ponto de falha no servidor iCAT, que nesta estratégia, é o responsável por armazenar os dados e gerenciar os

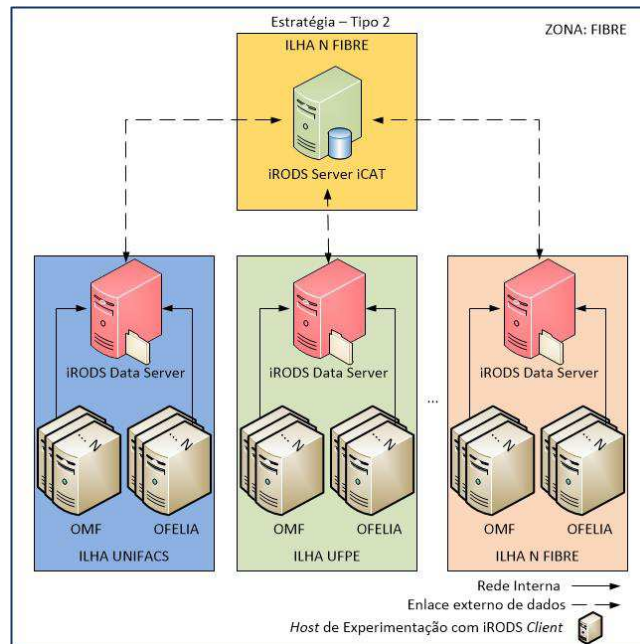
metadados. O segundo fator desfavorável está na sobrecarga que este servidor poderá sofrer (processador, memória e disco), com o crescimento do *testbed* ao incluir novas ilhas ao projeto e por consequência novos experimentadores, gerando um problema na escalabilidade do sistema. E por fim, o uso da rede externa para o armazenamento de dados, sobrecarregando o enlace para envio de dados de experimentação.

Com relação ao custo ou esforço de implantação desta proposta, a princípio é baixo em termos de equipamentos e de pessoas para gerenciar o serviço, pois necessita basicamente de um único servidor e um administrador iRODS para gerenciar este servidor. As máquinas de experimentação cedidas pelo CMF, já virão com os clientes iRODS instalados e configurados de maneira a não necessitar de intervenção técnica local para o seu funcionamento.

4.2.2 Estratégia Tipo 2

A segunda estratégia para o armazenamento dos dados em rede de experimentação consiste em distribuir os locais de armazenamento. Cada ilha passaria a ter um servidor iRODS para o armazenamento dos dados, recebidos dos clientes iRODS dos computadores de experimentação. Este servidor local na ilha se comunicaria com o servidor de catálogo externo (iRODS iCAT), que para esta arquitetura, não teria a função de armazenamento dos dados de experimentação, mas de gerência, controle de acesso, mapeamento dos dados e metadados espalhados pelos servidores iRODS em ilhas diferentes, todos fazendo parte da mesma zona FIBRE, conforme pode ser verificado na Figura 12.

Figura 12 - Estratégia de Armazenamento - Tipo 2



Fonte: Autoria própria (2013).

Assim como na estratégia Tipo 1, o servidor de catálogo poderá ser instalado em qualquer outra ilha do projeto FIBRE.

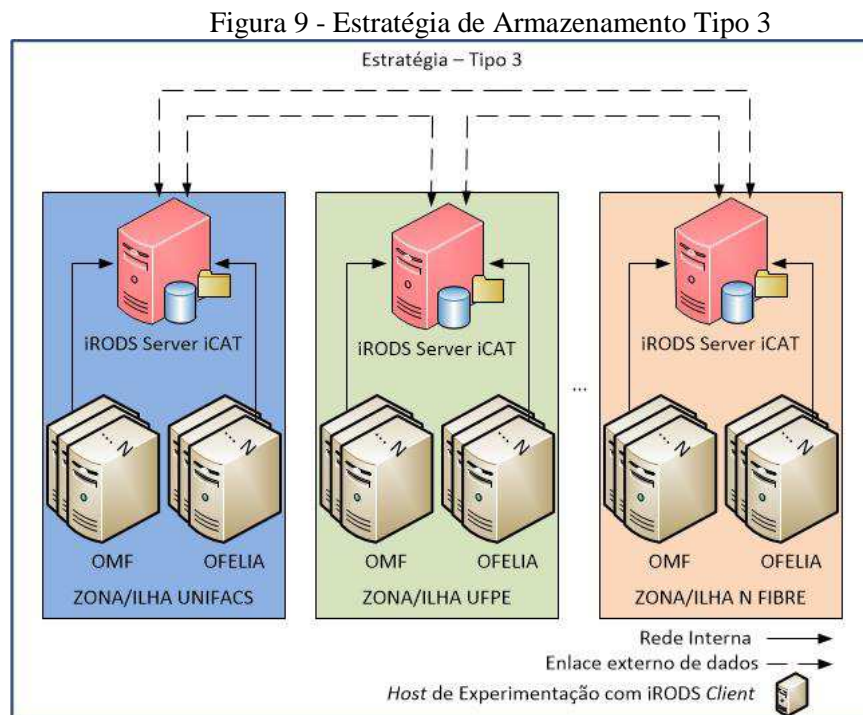
Na estratégia Tipo 2, um computador cliente de experimentação solicitará o armazenamento do dado enviando um pedido de liberação de acesso ao servidor local da ilha, e este, por sua vez, consultará o servidor iCAT que responderá aceitando ou negando o acesso.

Este tipo de estratégia tem como principais vantagens o armazenamento dos dados de forma distribuída, reduzindo a sobrecarga do servidor iCAT por este não ter a função de armazenar os dados do experimento, e no uso do enlace externo, uma vez que será utilizada a rede interna da instituição para o envio dos dados. A desvantagem está na complexidade de implantação e na centralização da gerência que, em caso de falha, acarretará indisponibilidade no serviço e comprometimento no armazenamento transparente. Outra desvantagem está em gerenciar um servidor a mais por ilha.

Trabalhar com sistemas de arquivos distribuídos implica em um esforço maior na gestão da funcionalidade da rede, pela inclusão de um servidor a mais por ilha. Este servidor irá gerar novas demandas técnicas para manter o serviço de armazenamento disponível na ilha, como configuração inicial do servidor e de monitoramento de infraestrutura do mesmo.

4.2.3 Estratégia Tipo 3

A terceira estratégia para o armazenamento dos dados em rede de experimentação utiliza o conceito de federação iRODS, onde cada ilha passaria a ser uma zona iRODS com o nome da instituição. A zona é formada por clientes iRODS instalados nos computadores de experimentação e o servidor iRODS iCAT, responsável pelo controle de acesso, armazenamento dos dados persistente dos experimentadores e gerenciamento dos dados e metadados. O servidor iRODS iCAT de cada ilha, através da relação de confiança, irá se comunicar com os outros servidores iCAT das outras ilhas, assim, após a comunicação entre os servidores, o usuário de uma zona local poderá ser usuário remoto da outra zona. O iRODS chama esta relação de iRODS *Federate* (SCHROEDER, 2008), vide Figura 13.



Fonte: Autoria própria (2013).

Uma das vantagens em utilizar a federação iRODS é a possibilidade, que em caso de falha de um servidor de uma ou mais zonas, o servidor de armazenamento não ficará comprometido para as outras zonas. Outra vantagem é o armazenamento dos dados de forma distribuída, dividindo a carga entre os servidores iCAT das ilhas não sobrecarregando o enlace externo, que só será utilizado, quando for necessário solicitar o dado localizado na outra zona.

A desvantagem em relação à estratégia Tipo 1, está no custo de mais um servidor por ilha, o que implica em uma complexidade maior de implantação, para configurar a federação

iRODS com todas as instituições do projeto FIBRE e na gerência deste servidor, pois cada ilha terá a função de armazenar o dado de experimentação e também controlar o acesso dos usuários. Outro problema referente à federação, é que o iRODS trabalha com estrutura hierárquica, onde cada zona é a raiz da hierarquia, assim, se o experimentador armazenar dados em três ilhas/zonas diferentes ele terá que percorrer a estrutura de diretórios destas ilhas para conseguir acessá-los o que dificulta a transparência do processo.

As três estratégias podem ser utilizadas para o projeto FIBRE. No entanto, existem algumas vantagens e desvantagens que foram consideradas para a definição da estratégia adotada de implantação do iRODS.

4.3 DEFINIÇÃO DA ESTRATÉGIA DE IMPLANTAÇÃO DO IRODS

Para a análise da definição das estratégias foram considerados os seguintes requisitos:

- **Tipos de armazenamento:** O armazenamento distribuído se aplica melhor na infraestrutura com várias ilhas, por não utilizar o enlace da rede externa para o armazenamento do dado. O enlace externo será utilizado para gerência e outras atividades de experimentação. Sendo assim, as estratégias do Tipo 2 e 3 são as que melhores se aplicam neste quesito. Este enlace poderá ser utilizado para medição de dados de experimentação entre CMFs de ilhas diferentes, onde o pico de algum armazenamento poderá interferir no resultado coletado.
- **Transparência:** A transparência é um fator facilitador de acesso ao dado pelo experimentador. Neste quesito os usuários que utilizarem as estratégias Tipo 1 e 2, terão acesso aos seus arquivos de experimentação de forma mais transparente em relação à estratégia Tipo 3 que utiliza a federação. Pois, os usuários situados na mesma zona iRODS terão seus dados disponibilizados no mesmo diretório independentemente da ilha em que este dado de experimentação esteja armazenado. Isso acontece porque o iRODS mapeia o local onde os dados do experimentador foram armazenados e os disponibiliza em um único diretório através de um endereço lógico, abstraindo assim o local de armazenamento físico.

Exemplo: O experimentador Carlos utilizou o armazenamento de dados nas ilhas UNIFACS e UFPE. Ao acessar o cliente *Web* iRODS para fazer o download do dado verificou que a disposição dos seus arquivos estavam distribuídos da seguinte forma:

Exemplo1.1:

/FIBRE/home/Carlos/Exp-UNIFACS-OCF-01

/FIBRE/home/Carlos//Exp-UFPE-OMF-01

Ou seja, Carlos ao fazer uso de um cliente iRODS para acessar os seu dados terá os seus arquivos de experimentação no diretório /FIBRE/home/Carlos, mesmo que este arquivos estejam espalhados por ilhas distintas do projeto FIBRE.

A estratégia Tipo 3 se utiliza do conceito de federação iRODS. Neste tipo de estratégia, o experimentador com as devidas permissões de acesso poderá, a partir de uma zona, acessar os dados de outra zona. Porém, o experimentador para ter acesso aos dados distribuídos entre as zonas deverá percorrer o esquema hierárquico de diretório das zonas, o que dificulta a transparência de acesso ao dado.

Exemplo1.2

/

/unifacs/home/Carlos/Exp-UNIFACS-OCF-01

/ufpe/home/Carlos/Exp-UFPE-OMF-01

- **Autenticação:** Como o projeto FIBRE possui seu esquema de autenticação federada LDAP, nas três estratégias o servidor iRODS iCAT, quando receber uma solicitação de acesso de uma aplicação cliente, fará consulta na base de usuário do FIBRE, liberando ou não o acesso.
- **Disponibilidade:** O iRODS possui mecanismos de replicação da base de dados PostgreSQL do servidor iRODS utilizando o PgPool, como já visto anteriormente. Isso facilita o reestabelecimento do serviço, no entanto, ainda não está funcionando de forma automática. Diante deste problema, a estratégia do Tipo 3 se mostrou um pouco mais eficiente em relação às demais, por permitir o acesso aos dados pelas ilhas que não sofreram falha em seu servidor iRODS.
- **Escalabilidade:** O *testbed* do FIBRE poderá se expandir com a inserção de novas instituições ou até mesmo interligar com outros *testbeds*. As três estratégias poderão ter seu serviço escalado no que se refere à federação entre as zonas. No entanto, a estratégia do Tipo 1 se mostra menos escalável em

relação às do Tipo 2 e 3 por ter um servidor centralizado limitado à capacidade de armazenamento do seu hardware.

Para definição da estratégia a ser utilizada, todas as três possuem suporte a federação, já que, para se federar, deverá existir uma relação de confiança entre os servidores iCAT, caso uma nova instituição ou *testbed* venha a se interligar ao projeto FIBRE. No entanto, as estratégias Tipo 2 e Tipo 3 se mostram mais eficientes no quesito escalabilidade do sistema por utilizarem sistemas de arquivo distribuídos. Assim, a rede externa não ficará sobrecarregada com o envio de dados para um único servidor, já que dispõe de servidor local para o armazenamento dos dados.

Neste sentido e tomando como base o acesso mais transparente, já que os dados não ficarão dispersos em diversos diretórios para o experimentador, será utilizada a estratégia Tipo 2 na prova de conceito deste trabalho para o armazenamento de dados do projeto FIBRE.

4.4 IMPLANTAÇÃO DOS COMPONENTES IRODS PARA O PROJETO FIBRE

Para implantação do sistema de armazenamento de dados de experimentação, foram instaladas as aplicações cliente e servidor conforme existente na estratégia Tipo 2.

Nesta seção, algumas questões são abordadas visando o melhor desempenho e funcionalidades das ferramentas clientes iRODS, tanto para aplicações cliente que ficarão nas máquinas de experimentação, quanto para as aplicações cliente *Web* de acesso externo pelo experimentador.

4.4.1 Aplicação cliente iRODS para máquina de experimentação

Foi realizada uma comparação entre as aplicações clientes iRODS disponíveis, que melhor se encaixam nas máquinas de experimentação, de forma a tornar transparente o armazenamento dos dados dos experimentos nos computadores disponíveis. Assim, as aplicações *Web* como, *iDROP Web* e o *iRODS Web Browser* foram descartadas, pois o uso das mesmas não tornaria o armazenamento de dados transparente ao experimentador, uma vez que teria que utilizar algum navegador para, em seguida, fazer o *upload* do arquivo. Sendo assim, foram testadas duas aplicações *desktop*, *idrop-lite* e o *icommands*, que ficarão nas máquinas de experimentação disponíveis para o envio dos dados gerados pela pesquisa.

A ideia é que estas aplicações consumam a menor quantidade de recurso físico da máquina (memória RAM, espaço em disco e processamento) e que sejam utilizadas quando o experimentador assim o desejar ou de forma automatizada.

O tamanho dos arquivos de instalação utilizados por estas aplicações nas máquinas de experimentação é relativamente pequeno, ocupando poucos MB de espaço em disco. A diferença está no seu funcionamento, pois a aplicação iDROP precisa do JAVA instalado e funcionando em *background* para execução do *idrop-lite*, consumindo memória RAM mesmo quando não existam arquivos para serem armazenados. Já o cliente *icommands* apesar de situado na máquina de experimentação, só consome recurso quando acionado. Devido a esta característica este cliente se mostrou mais interessante para o uso no projeto FIBRE.

A fim de facilitar o armazenamento dos dados de experimentação e torná-lo o mais transparente possível para o experimentador, as máquinas de experimentação disponibilizadas pelos CMFs já virão com o iRODS cliente e suas configurações prévias como, por exemplo, o serviço de autocompletar do Linux para os comandos iRODS, que adicionará no arquivo “*.profile*” do *home* do usuário o endereço dos executáveis do *icommands*.

Exemplo: `PATH=/etc/clients/icommands/bin:$PATH`

4.4.2 Aplicação cliente iRODS para acesso externo

A consulta aos dados de experimentação gravados fora dos CMFs poderá ser feita através do cliente *Web* do iRODS. Foram avaliados dois clientes *Web* existentes no site do iRODS que são: iRODS *Web Browser* desenvolvido em PHP e o iDROP *Web* desenvolvido em JAVA.

Nesta avaliação, foram testadas as seguintes versões para as iRODS *Web Browser* 3.3.0 e iDROP *Web* 2.0.0 jargon:3.3.2 para os seguintes itens: controle de acesso, possibilidade de tornar o dado do experimento público, inserir metadados, consultar metadados, suporte a federação, ou seja, consultar o dado de uma zona remota.

Ao comparar as duas aplicações *Web*, o item consulta por metadados não foi possível ser realizado pela versão do iDROP, o que inviabiliza o seu uso para o contexto do projeto FIBRE. Os demais itens como inserir metadados, consultar metadados, suporte a federação foram atendidos pelas duas aplicações.

4.4.3 Configuração da máquina cliente

Para preparação da máquina de experimentação para envio de dados e metadados, que, por definição, será utilizado o cliente iRODS *icommands*, algumas configurações deverão ser realizadas.

Em primeiro lugar, todas as máquinas deverão estar configuradas no DNS (*Domain Name System*), pois na configuração cliente e servidor iRODS não é permitido o uso de IP (Protocolo de Internet). Segundo, deverão ser configuradas as variáveis de ambiente do iRODS *iCOMMANDS* que ficarão no *home* do usuário no diretório `~/irods/irodsEnv`.

As variáveis de ambiente são:

- `irodsHost`: Nome do servidor iRODS da ILHA
Ex.: `irodsHost 'fibre-unifacs-server'`
- `irodsPort`: Porta de acesso do serviço
Ex.: `irodsPort 1247`
- `irodsDefResource`: Nome do recurso
Ex.: `irodsDefResource 'unifacsResc'`
- `irodsHome`: Nome lógico que será apresentado para o usuário na consulta aos dados.
Ex.: `irodsHome '/FIBRE/home/carlos'`
- `irodsUserName`: Nome do usuário iRODS, este será o mesmo da base LDAP do projeto FIBRE.
Ex.: `irodsUserName 'carlos'`
- `irodsZone`: Nome da zona
Ex.: `irodsZone 'FIBRE'`

As máquinas de experimentação alocadas pelo CMF ao experimentador são disponibilizadas com um usuário e senha padrão do Linux, não exigindo do experimentador o uso das credenciais de autenticação do *testbed* para ter acesso à máquina de experimentação. Isso dificulta a configuração das variáveis de ambiente do cliente iRODS por não se saber quem é o experimentador.

Diante deste problema, não é trivial, para o experimentador, a configuração destas variáveis, nem se obtém a transparência para o armazenamento dos dados de experimentação. A fim de resolver possíveis configurações manuais das variáveis de ambiente do iRODS, foi desenvolvido um script com o nome `ArmazenaDados`, onde o usuário em uma única vez, irá

passar o *login* e a senha do *testbed*, e o sistema irá configurar as variáveis autenticando este usuário de forma que ele não precise informar a senha posteriormente.

A senha digitada pelo experimentador é armazenada de forma criptografada em MD5, para uso futuro em um arquivo (.irodsA).

O *script* poderá ser consultado no apêndice B desta dissertação.

4.4.4 Configuração dos servidores iRODS

Seguindo a implantação da estratégia tipo 2, haverá um servidor de armazenamento iRODS *Server* instalado em cada ilha e um outro servidor iRODS iCAT, que poderá ficar em qualquer ilha do projeto FIBRE. Para a configuração destes servidores iRODS, alguns requisitos de implantação no ambiente foram tratados para criação dos recursos e controle de acesso no iRODS.

4.4.4.1 Requisitos

Alguns requisitos são importantes para configuração do servidor iRODS no projeto FIBRE dentre eles estão:

- **DNS:** Para que a comunicação entre os servidores e clientes ocorra com sucesso no iRODS, é necessário que o serviço de DNS do projeto FIBRE esteja operante, pois o iRODS utiliza somente o nome e não o IP para comunicação entre as máquinas. A falta de DNS configurado na rede, irá gerar uma sobrecarga para o administrador da rede ao inserir possíveis caminhos no arquivo *hosts.conf* do Linux (seja para a máquina cliente seja para a máquina servidor) o que torna inviável a administração do ambiente.
- **LDAP:** Protocolo de autenticação adotado pelo projeto FIBRE para a federação das bases de usuários das ilhas. É fundamental para que os serviços existentes no *testbed* que demandem autenticação o façam juntamente com o LDAP, pois reduz o trabalho de quem está gerenciando o acesso à rede de experimentação, e para o experimentador, por possuir uma única senha para utilização dos serviços disponíveis no *testbed*.

O iRODS se comunica com o sistema de autenticação LDAP através do *Pluggable Authentication Modules – PAM* (SCHROEDER, 2013), podendo através dos atributos dos usuários autorizar ou não o acesso ao dado.

Para isso, no servidor iCAT deverá ser informado que o modo de autenticação será via PAM. Este arquivo fica nas configurações do iRODS em `config/config.mk`, basta desmarcar o comentário as linhas `PAM_AUTH =1` e o `USE_SSL =1` e instalar os pacotes `libpam0g-dev`, `openssl` e `libssl-dev`.

4.4.4.2 Criação de recursos

Os servidores que ficarão nas ilhas serão responsáveis pelo armazenamento persistente dos dados do experimentador. Para que o cliente *icommand* da máquina de experimentação saiba onde armazenar o dado, a variável de ambiente deverá conter o nome do “recurso”. Este recurso é criado pelo administrador iRODS no servidor iCAT (RUSSELL, 2012) contendo informações do servidor onde serão armazenados os dados de experimentação (nome do servidor e endereço físico do dado), o tipo de sistema e a classe do arquivo. A criação dos recursos é fundamental para a definição da arquitetura, porque define o local onde será armazenado o dado e porque exige do administrador o conhecimento do ambiente, configurando o sistema de arquivo correto para o armazenamento do dado. Cada ilha do projeto FIBRE terá seu recurso criado pelo administrador, a descrição das informações necessárias para sua configuração segue abaixo:

- *Name*: nome que será dado ao recurso;
Ex.: `unifacs-Resc`
- *Type*: é necessário saber se é um sistema de arquivo do Linux ou do Windows, uma unidade de FITA, um banco de dados;
Ex.: Para o FIBRE o tipo de recurso é “*unix file system*”
- *Class*: Se é do tipo arquivo, *cache* ou *compound*
Ex.: Para o FIBRE escolhido o recurso *Archive* por possuir uma menor latência no acesso com o “*unix file system*”
- *Host*: Nome FQDN do servidor iRODS da ilha
Ex.: `fibre-unifacs-server`
- *Path*: Local onde ficará armazenado o arquivo no servidor iRODS da ilha.
Ex.: `/home/operador/iRODS/Vault`

Desta forma, cada ilha possuirá o seu recurso criado no servidor iRODS iCAT, de forma que o *host* cliente de uma ilha armazene os dados localmente no servidor iRODS. O *cookbook*, situado no apêndice C desta dissertação, descreve os comandos necessários para a criação do recurso.

4.4.4.3 Autenticação

Como dito anteriormente, o projeto FIBRE possui o seu esquema de autenticação e autorização LDAP e o iRODS consegue, via PAM, a comunicação com este protocolo. Só que, o usuário experimentador ao aceitar a política de acesso criada para entrar no projeto FIBRE, passará a ter as credenciais (*login* e senha) de acesso ao *testbed*, mas não terá ainda acesso ao serviço de armazenamento de dados, pois será necessário a criação deste usuário na base de dados do iRODS.

Então, o administrador da rede fará um procedimento simples que poderá ser automatizado com o desenvolvimento de um *script* para criação de contas no projeto FIBRE incluindo a seguinte linha de comando para criar um usuário no iRODS: “*iadmin mkuser usuarioldap rodsuser*” para perfil de usuário e “*iadmin mkuser usuarioldap rodsadmin*” para perfil de administrador.

Feito isso, o iRODS irá consultar a senha deste usuário via LDAP, ou seja, a senha não estará na base de dados do iRODS. O procedimento é semelhante ao que ocorre em outros serviços como e-mail, onde se cria a conta do usuário, porém a senha é validada via LDAP.

4.5 ESTRATÉGIA DE ACRÉSCIMO E USO DE METADADOS NO CONTEXTO DO PROJETO FIBRE ASSOCIADO ÀS FACILIDADES DO IRODS

Ao escalar um grande número de dados, se faz necessária à inserção de metadados para facilitar consultas futuras, inserindo informações úteis referentes ao dado armazenado. Algumas informações sobre o dado já são inseridas pelo próprio iRODS como: tamanho, data da criação, data da modificação, nome do proprietário do dado, a zona, o caminho físico no servidor onde se encontra o dado, nome do recurso, versão, entre outros. Entretanto, estas informações não são suficientes para consultas específicas que dizem respeito ao ambiente de experimentação onde os dados armazenados são resultados das medições dos experimentos em um *testbed* multi-CMF. Neste sentido, foram definidos alguns metadados básicos importantes para sua descrição, dentre eles estão o CMF e a ILHA:

CMF: OMF ou OCF ou PGENI;

ILHA: UNIFACS ou UFPE ou UFPA,...., e etc.

O iRODS utiliza três tipos de atributos de metadados chamado de AVUs (*Attribute – Value – Units*) que consistem em um atributo de nome, um atributo de valor e um atributo opcional de unidades. Estes atributos poderão ser adicionados através dos clientes iRODS como o *icommands* e o *iRODS Web* (*iDROP* e *Web Browser*), descrevendo o dado de forma a facilitar a consulta dos dados e objetos de correspondência.

Assim, foi desenvolvida na aplicação ArmazenaDados (seção 4.6) uma função para o armazenamento de dados e também de metadados, onde de forma automática o sistema irá informar qual é o CMF utilizado pelo experimentador e a ilha na qual o dado foi criado. O acesso será através de aplicação *Web* onde o experimentador fará acesso aos dados e metadados que foram armazenados persistentemente ou não, distribuídos nas diversas ilhas.

4.6 ARMAZENADADOS

O ArmazenaDados é um *script* que tem como principal função configurar a máquina de experimentação para o iRODS, utilizando o seu cliente *icommands*, de forma simples para o experimentador, abstraindo configurações de variáveis de ambiente, comandos iRODS e os seu parâmetros de uso.

Esta aplicação ficará no Path do Linux, e poderá ser executada pelo usuário experimentador de qualquer diretório do sistema operacional. Ao ser acionada será apresentada uma tela de *menu* com opções necessárias para configurar a máquina de experimentação e enviar os dados do experimento, da seguinte forma:

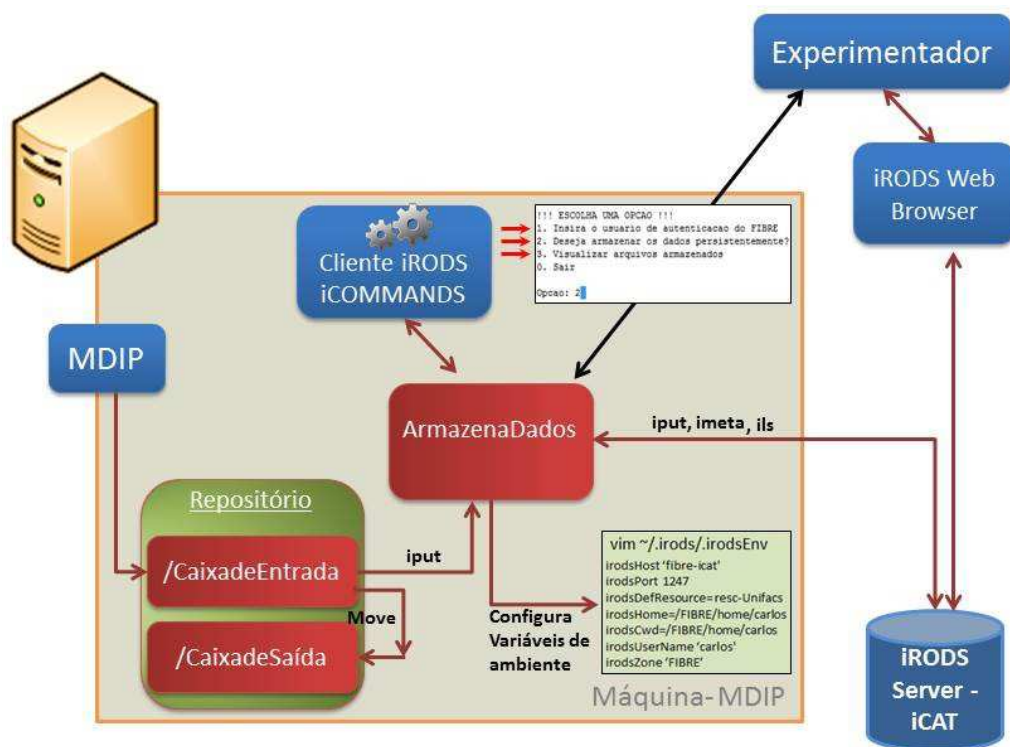
- Opção 1 – Autenticação: Solicita a autenticação deste experimentador, armazenando o *hash* da senha criptografada de forma a não precisar se autenticar novamente. Em seguida a aplicação configura as variáveis de ambiente do iRODS (*.irodsEnv*) com as informações do experimentador, ilha e CMF, verifica a existência dos diretórios */CaixadeEntrada* e */CaixadeSaida*, que por padrão já virão pré-configurados na imagem cedida pelo CMF.
- Opção 2 - Solicita ao experimentador o armazenamento do dado: a aplicação ArmazenaDados executa os comandos *iput* e *imeta*, do cliente iRODS *iCOMMANDS*, enviando o dado e metadados resultado do experimento para o repositório de dados iRODS Server, onde alguns metadados serão inseridos por

padrão, como nome da Ilha e tipo de CMF além de possibilitar ao experimentador colocar alguma descrição sobre o dado, ou seja, um novo metadados. Os arquivos resultados dos experimentos deverão estar no diretório /CaixadeEntrada, diretório definido neste trabalho, como o local de recepção dos dados coletados pelo MDIP do CMF, sem seguida, após escolhido a Opção2, será movido o arquivo para a Caixa de Saída, mantendo assim, o arquivo no servidor iRODS e na máquina de experimentação.

- Opção 3 - Consulta os dados inseridos no servidor iRODS.

A Figura 14, detalha a disposição da aplicação ArmazenaDados na máquina de experimentação.

Figura **Erro! Nenhum texto com o estilo especificado foi encontrado no documento.**10 - Funcionamento iRODS em uma máquina de experimentação



Fonte: Autoria própria (2013).

4.7 CONSIDERAÇÕES FINAIS

Neste capítulo, foi apresentado o armazenamento de dados baseado no iRODS para o projeto FIBRE, representando na arquitetura I&M, Figura 14, no ponto de vista dos CMFs, as localizações onde ficarão situados os clientes iRODS em uma ilha do FIBRE e o envio dos dados dos experimentos pelos clientes iRODS para o repositório de dados persistente. Foi

abordado também, a visão das infraestruturas possíveis para o uso dos componentes iRODS no contexto do projeto FIBRE em três tipos de estratégias, tendo definido a estratégia tipo 2, por possuir o acesso mais transparente aos dados dispersos em diversas ilhas, como a escolhida para o uso na prova de conceito, capítulo 5.

Foi verificado na seção 4.5 o uso de acréscimos de metadados associados às facilidades do IRODS tendo a descrição do dado CMF e ILHA como padrão e o desenvolvimento da aplicação AmazenaDados, seção 4.6, utilizada para simplificar as configurações de ambiente da máquina de experimento facilitando o envio dos dados e metadados de experimentos por parte do experimentador.

5 PROVA DE CONCEITO

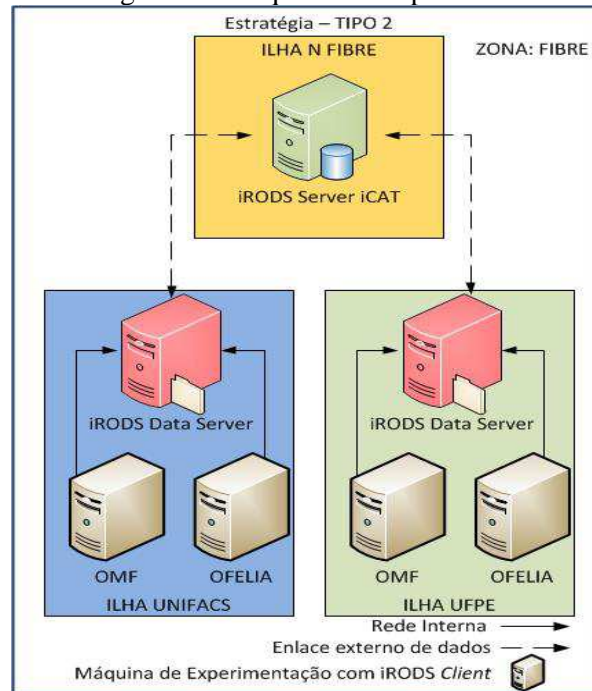
O objetivo desta prova de conceito é aplicar a estratégia de implantação do iRODS para armazenamento de dados no projeto FIBRE buscando validar a proposta (estratégia tipo 2). Neste contexto, será realizada uma implantação com Máquinas Virtuais (VM) simulando a operação do sistema a fim de avaliar a operação do iRODS como proposta de uso para o projeto FIBRE. Nesta implantação será verificada e aferida a viabilidade para o armazenamento dos dados de forma distribuída, a inserção de metadados ao dado pela aplicação cliente, a consulta por metadados, a disposição do acesso ao dado de forma transparente e o suporte a federação.

5.1 INFRAESTRUTURA DE IMPLANTAÇÃO

O ambiente foi configurado simulando duas ilhas do projeto FIBRE, UNIFACS e UFPE, cada uma fazendo uso de dois arca-bouços de controle diferentes, OFELIA e OMF. Assim, conforme o escopo da estratégia tipo 2, foram criadas sete máquinas virtuais, sendo estas representadas da seguinte forma: um servidor iCAT, responsável pela gerência dos dados/metadados e pelo controle de acesso aos serviços iRODS, este servidor poderá ficar em qualquer ilha do projeto FIBRE; três máquinas virtuais para cada ilha (UNIFACS e UFPE), contendo um servidor iRODS, responsável pelo armazenamento dos dados de experimentação gerados pelas máquinas de experimentação existente nesta ilha, e duas máquinas de experimentação, uma representando o OFELIA (OCF) e outra representando o OMF, todas contendo o cliente iRODS *icommands* necessário para identificação do experimentador e o envio dos dados de experimentação para o servidor iRODS, a configuração do ambiente poderá ser verificada no apêndice A.

A Figura 14 ilustra a arquitetura de infraestrutura da prova de conceito com a localização dos componentes iRODS e a representação da sua interligação. O Quadro 2 apresenta os detalhes de configuração das máquinas que compõe o ambiente da prova de conceito contendo: os componentes iRODS, sua quantidade, funcionalidade e os detalhes da configuração. Não foi implantado nesta prova de conceito o serviço de autodisponibilidade do servidor iCAT e a comunicação de autenticação com o servidor LDAP. No entanto, a escolha da tecnologia busca atender tais requisitos.

Figura 15 - Arquitetura da prova de conceito



Fonte: Autoria própria

Quadro 2 - Detalhes dos Componentes iRODS para a Prova de Conceito

Componentes	Quantidade	Função	Configuração
Servidor iRODS iCAT	1	Autenticação, catálogo de metadados, servidor <i>Web</i>	iRODS 3.3, Debian7, Memória de 1GB, Disco Rígido de 8 GB
Servidor iRODS	2	Servidor local da ilha para armazenamento de dados	iRODS 3.3, Debian7, Memória de 512 MB, Disco Rígido de 40 GB
Cliente iRODS	4	Autenticação e envio dos dados de experimentação	icommands, Debian7, Memória de 512 MB, Disco Rígido de 10 GB

Fonte: Autoria própria (2013).

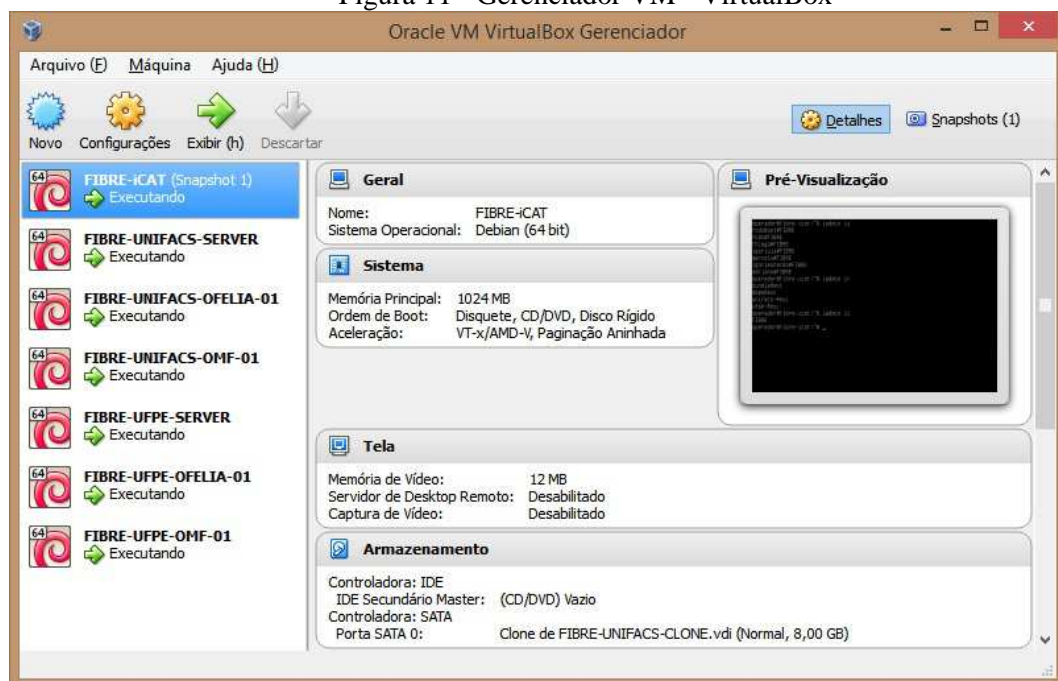
As máquinas virtuais, foram instaladas no Oracle VM VirtualBox 4.3.8 (Figura 16) com as configurações descritas no Quadro 2 e com a nomenclatura (*Hostname*) e endereçamento IP detalhados no Quadro 3, fundamental para comunicação entre os componentes iRODS.

Quadro Erro! Nenhum texto com o estilo especificado foi encontrado no documento.2 - Configuração das máquinas para Prova de Conceito

Localização	Componentes	Hostname: FQDN	Endereço IP
NOC-DO-FIBRE	Servidor iRODS iCAT	fibre-icat	192.168.56.100
UNIFACS	Servidor iRODS	fibre-unifacs-server	192.168.56.101
UNIFACS	Cliente iRODS	fibre-unifacs-ofelia-01	192.168.56.102
UNIFACS	Cliente iRODS	fibre-unifacs-omf-01	192.168.56.103
UFPE	Servidor iRODS	fibre-ufpe-server	192.168.56.111
UFPE	Cliente iRODS	fibre-ufpe-ofelia-01	192.168.56.112
UFPE	Cliente iRODS	fibre-ufpe-omf-01	192.168.56.113

Fonte: Autoria própria (2013).

Figura 11 - Gerenciador VM - VirtualBox



Fonte: Autoria própria (2013).

Para configurar o ambiente de teste, foi necessário adotar o serviço de autenticação do iRODS e o de resolução de nomes do próprio Linux, para suprir o serviço de autenticação e DNS, existente no *testbed* do FIBRE.

Assim, foram configurados em cada VM de teste, no arquivo */etc/hosts*, os possíveis endereços IPs e os FQDNs (*Fully Qualified Domain Names*) / *hostnames* das máquinas que estarão se comunicando entre si nesta arquitetura, conforme identificação de nomes e endereços IP listados no Quadro 3

O processo de instalação dos componentes iRODS foi realizado de acordo com o definido na arquitetura de prova de conceito. Foram instalados o servidor de metadados iCAT,

os servidores de armazenamento dos dados distribuídos, *iRODS Server*, e, por fim, as máquinas de experimentação OMF e OFELIA com o cliente *iRODS*, que representam as máquinas de experimentação alocadas ao pesquisador pelo CMF.

5.2 TESTE DA INFRAESTRUTURA IRODS (ESTRATÉGIA TIPO 2)

Para demonstração de um armazenamento de dados de experimentação, foram criadas cinco contas de usuário (Thiago, Marcelo, Adriano, IgorLuiz, IgorLeonardo) com perfil “*rodsuser*” no servidor *iRODS iCAT*, onde para cada usuário experimentador foi realizado o processo de armazenamento dos “dados coletados do experimento”. Para efeito da simulação da operação do sistema, os dados são arquivos de diversos tamanhos disponibilizados para a realização dos testes. O Quadro 4 retrata os locais utilizados pelos experimentadores para realização dos experimentos.

Quadro 3 - Locais dos experimentos realizados pelos experimentadores

Experimentador	Experimentos Realizados			
	UNIFACS		UFPE	
	OFELIA	OMF	OFELIA	OMF
Thiago	x			x
Marcelo	x	x	x	x
Adriano		x	x	
Igorluiz	x		x	
Igorleonardo		x		x

Fonte: Autoria própria (2013).

Foram colocados alguns arquivos nas máquinas de experimentação no diretório /CaixadeEntrada, diretório definido neste trabalho, como o local de recepção dos dados coletados pelo MDIP do CMF (Figura 14). Ao realizar o processo de armazenamento dos dados utilizando a aplicação ArmazenaDados os metadados com as informações da ilha e do CMF são inseridas automaticamente, ficando a critério de o experimentador inserir o seu metadados com detalhes específicos do seu experimento, conforme será verificado adiante na Figura 17. O Quadro 5 descreve o cenário de teste utilizado para inserção dos dados e metadados pelos experimentadores.

Quadro Erro! Nenhum texto com o estilo especificado foi encontrado no documento.4 - Disposição dos arquivos por máquina de experimentação

ILHA	CMF	Thiago	Marcelo	Adriano	Igorluiz	Igorleonardo
UNIFA CS	OFELIA	host_ ifacebw.rrd	Flowtp.rrd		host_ ifacebw.rrd	
	Metadados		atributo=metrica valor=throughput de fluxo		atributo=metrica valor=consumo de banda	
	OMF		OML001.sql	OML002.sql		OML004.sql
	Metadados		atributo=metrica valor=atraso			atributo=metrica valor=throughput
UFPE	OFELIA		host_ ifacebw.rrd	host_ ifacebw.rrd	Flowtp.rrd	
	Metadados				atributo=metrica valor=throughput	
	OMF	OML001.sql	OML002.sql			OML003.sql
	Metadados					atributo=metrica valor=atraso

Fonte: Autoria própria (2013).

Para demonstrar os passos adotado por um experimentador no envio de um dado de monitoração de um CMF/ILHA segue, na Figura 17, a tela de *menu* da aplicação ArmazenaDados utilizada para o envio dos dados de experimentação, onde o usuário após ter se autenticado no FIBRE (opção 1) procederá com o envio dos dados de monitoração (opção2).

Figura 12 - Menu do ArmazenaDados

```

!!! ESCOLHA UMA OPCAO !!!
1. Insira o usuario de autenticacao do FIBRE
2. Deseja armazenar os dados persistentemente?
3. Visualizar arquivos armazenados
0. Sair

Opcao: 2

```

Fonte: Autoria própria (2013).

Ao escolher a opção 2, a aplicação ArmazenaDados irá proceder com o envio do dado de experimentação que se encontra no diretório /CaixadeEntrada, dando a opção de o experimentador inserir alguns metadados, caso assim o deseje. No Quadro 5 estão alguns metadados inseridos pelos experimentadores. Poderá ser verificado na Figura 18, um exemplo

de inserção de um dado realizado pelo experimentador Igor Leonardo juntamente com os metadados “métrica” e “atraso” ao dado OML003.sql.

Figura 13 - Envio do dado com o metadado pelo experimentador

```

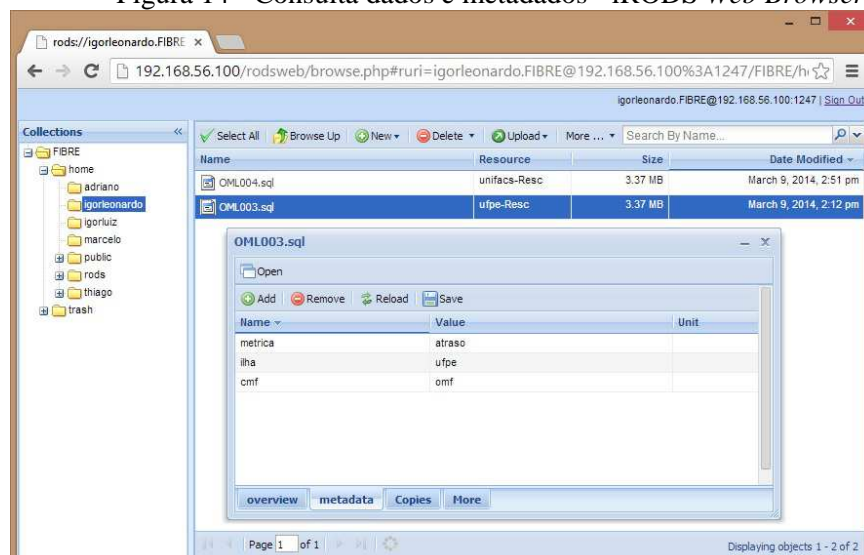
192.168.56.113 - PuTTY
ARMAZENAMENTO DE DADOS PERSISTENTE
=> Verificando a existencia das Caixas de Entrada e Saida
Caixa de Entrada: STATUS OK.
Caixa de Saida: STATUS OK.

=> Verificando arquivos na Caixa de Entrada
Existe(m) 1 arquivo(s) para ser(em) enviado(s) ...
Armazenando o arquivo (OML003.sql) persistentemente...
Inseridos os metadados ilha=ufpe e cmf=omf
Deseja inserir outro metadado ao dado OML003.sql?
  Digite: 1 para Sim e 2 para Nao
1
Insira um Atributo de NOME
metrica
Insira um Atributo de Valor
atraso
  
```

Fonte: Autoria própria (2013).

Para a realização da consulta do dado o experimentador poderá escolher a “opção 3” do *menu* da aplicação ArmazenaDados ou acessar o cliente iRODS *Web Browser* para obter maiores detalhes dos dados e uma maior gerência sobre o mesmo. A Figura 19, mostra o dado armazenado pelo experimentador Igor Leonardo e os metadados inseridos automaticamente pela aplicação ArmazenaDados (ilha=ufpe e cmf=omf) e os metadados inseridos por Igor (métrica=atraso).

Figura 14 - Consulta dados e metadados - iRODS *Web Browser*



Fonte: Aatoria própria (2013).

Conforme poderá ser verificado nas Figuras 20 e 21, através desta aplicação *web*, o experimentador poderá fazer o download do experimento, apagar um dado, replicar o dado para outro servidor iRODS, inserir um metadados e fazer consultas por metadados correlatos, o que é muito importante para um ambiente de experimentação com muitas instituições e CMF diferentes.

Figura 15 - Consulta por metadados

The screenshot shows the 'Advanced Search' window. It is divided into two main sections: 'Attributes' and 'Metadata'.

Attributes Section:

- Name: Name or Partial Name, case sensitive
- Modified Within: Any Time
- Owner: Owner of the file
- Resource: Resource of the file
- Only: Under Current Collection
- Current Collection: /FIBRE/home/igorleonardo

Metadata Section:

- metrica like throughput
- Name Op Value
- Name Op Value
- Name Op Value
- Name Op Value

A 'Search' button is located at the bottom right of the window.

Fonte: Aatoria própria (2013).

Figura 16 - Resultado da consulta

The screenshot shows the 'File Search Results' window with a table of results. The table has five columns: Name, Collection, Resource, Size, and Date Modified.

Name	Collection	Resource	Size	Date Modified
OML004.sql	/FIBRE/home/igorleonardo	unifacs-Resc	3.37 MB	March 9, 2014, 2:51 pm
Flowtp.rrd	/FIBRE/home/igorluiz	ufpe-Resc	299.57 K	March 9, 2014, 1:59 pm
Flowtp.rrd	/FIBRE/home/marcelo	unifacs-Resc	3.37 MB	March 9, 2014, 1:19 pm

At the bottom of the window, there is a pagination bar showing 'Page 1 of 1' and 'Displaying Files 1 - 3 of 3'.

Fonte: Aatoria própria (2013).

Assim, foi possível simular a implantação da estratégia tipo 2 tendo os dados distribuídos pelas duas ilhas representadas pela UNIFACS e UFPE, todas fazendo parte da zona FIBRE.

A decomposição do desafio para o armazenamento transparente dos dados de medição multi-CMF envolveu várias etapas como:

- semântica de dados e metadados – foi possível acessar os dados de experimentação espalhados nas duas ilhas de forma transparente abstraindo o local onde os dados se encontravam fisicamente armazenados, disponibilizando para o experimentador um endereço lógico para o acesso ao dado, conforme poderá ser verificado na Figura 19, onde no endereço lógico “FIBRE/home/igorleonardo” constam dois arquivos de experimentação OML004.sql (existente na ilha UNIFACS) e o OML003.sql (existente na ilha UFPE) situados fisicamente em ilhas distintas. Outro teste realizado foi a inserção de metadados ao dado pela aplicação cliente e consulta por metadados, representadas nas Figuras 20 e 21 ao consultar a métrica *throughput*;
- controle de acesso – realizado pela aplicação cliente que consulta a base de usuário do iRODS Server iCAT para liberar ou não o acesso do usuário para envio dos dados de experimentação, atendendo ao item de segurança correspondente ao nível de acesso estipulado para o experimentador; distribuição de interface cliente, tanto a interface cliente icommands, existentes nas máquinas de experimentação, quanto na interface *Web* foi possível, depois de autenticado o usuário experimentador, realizar a manipulação dos dados podendo apagar, renomear, inserir um metadado, consultar um metadados e fazer download do dado para o seu computador pessoal;
- escalabilidade – é a possibilidade de utilizar o servidor para o armazenamento dos dados distribuídos por ilha, nesta prova de conceito foi utilizado o servidor iRODS UNIFACS e servidor iRODS UFPE como repositório dos dados nos experimentos multi-CMF.

5.3 TESTE DA ESTRATÉGIA DE FEDERAÇÃO

O teste da estratégia de federação iRODS, visa atender às solicitações de adesão de outros *testbeds* ao FIBRE. O *testbed* que for aderir ao projeto FIBRE possuirá seu esquema

particular de autenticação, onde os seus usuários experimentadores poderão fazer uso dos recursos do outro *testbed* utilizando as credenciais de acesso do seu *testbed* local.

Para simular a inserção de um novo *testbed* ao FIBRE, foi criada nesta prova de conceito outra zona iRODS chamada de IFBA, representando outro *testbed*, a fim de testar a comunicação e acesso de um usuário experimentador da zona IFBA na zona FIBRE através da federação iRODS.

A nova zona “IFBA” terá em sua infraestrutura três máquinas distribuídas da seguinte forma: um servidor iRODS iCAT e duas máquinas de experimentação, sendo uma com o CMF OFELIA e outra com o CMF OMF.

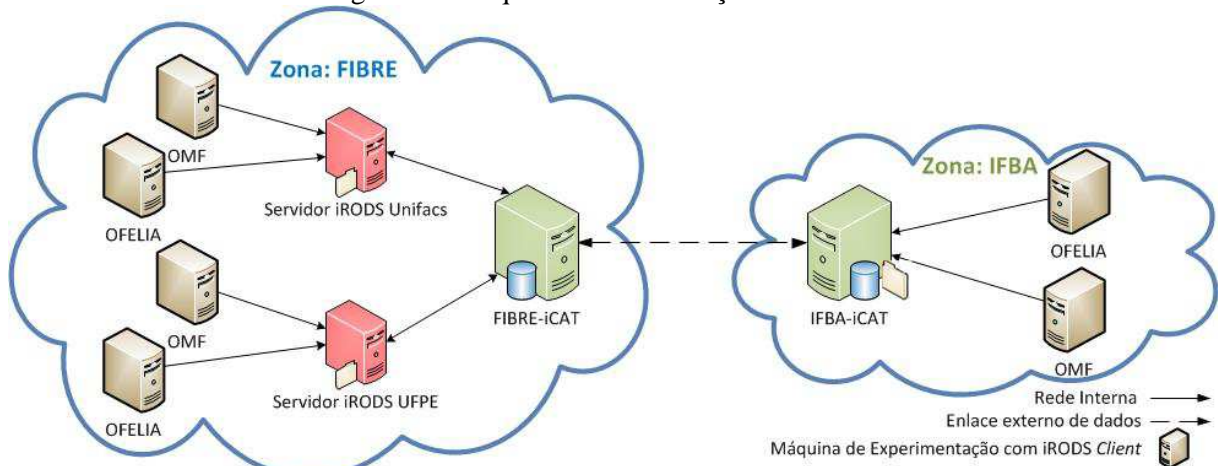
Na Figura 22, poderá ser verificada a arquitetura de infraestrutura da federação da prova de conceito com a localização dos componentes iRODS e a representação da sua interligação e, no Quadro 6, os detalhes de configuração das máquinas que compõem o ambiente da prova de conceito da zona IFBA contendo: os componentes iRODS, sua quantidade, funcionalidade e os detalhes da configuração.

Quadro 5 - Detalhes dos componentes iRODS da ilha IFBA para a Prova de Conceito

Componentes	Quantidade	Função	Configuração
Servidor iRODS iCAT	1	Autenticação, catálogo de metadados, armazenamento dos dados de experimentação e servidor <i>Web</i>	iRODS 3.3, Debian7, Memória de 1GB, Disco Rígido de 8 GB
Cliente iRODS	2	Autenticação e envio dos dados de experimentação	icommands, Debian7, Memória de 512 MB, Disco Rígido de 10 GB

Fonte: Aatoria própria (2013).

Figura 17 - Arquitetura da federação - Prova de Conceito



Fonte: Aatoria própria (2013).

No Quadro 6 e no Quadro 7 poderão ser verificadas as configurações utilizadas nas máquinas virtuais para criação desta nova zona iRODS representando o serviço de armazenamento de outro *testbed*. No Quadro 7 estão os detalhes da nomenclatura (*hostname*) e endereçamento IP, fundamental para comunicação entre os componentes iRODS na zona IFBA e fundamental para a federação entre os *testbeds*.

Quadro 6 - Descrição de ambiente iRODS do *testbed* do IFBA

Localização	Componentes	Hostname: FQDN	Endereço IP
NOC-DO-IFBA	Servidor iRODS iCAT	ifba-icat	192.168.56.200
Campus-Salvador	Cliente iRODS	ifba-ssa-ofelia-01	192.168.56.201
Campus-Salvador	Cliente iRODS	ifba-ssa-omf-01	192.168.56.202

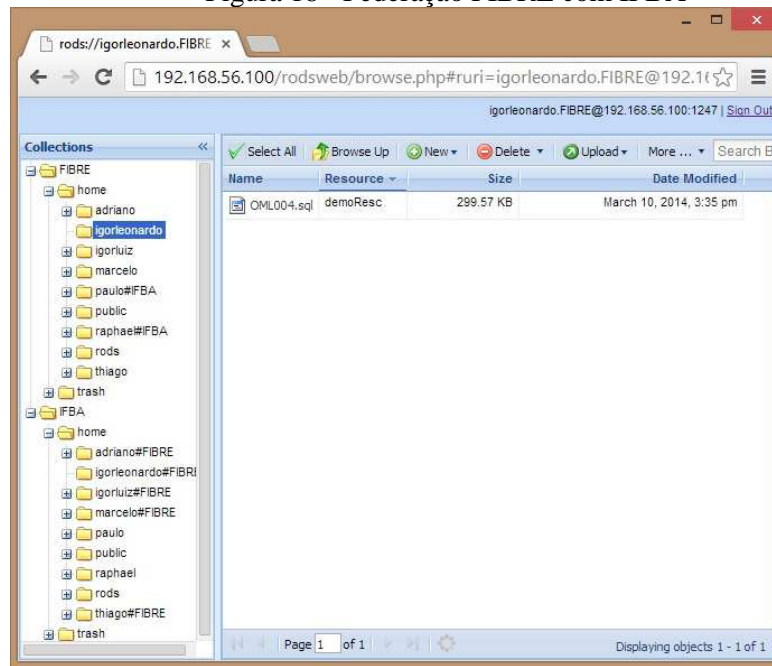
Fonte: Autoria própria (2013).

Para realizar a federação entre as zonas iRODS, foi necessário informar ao servidor iCAT da zona local o nome da zona remota, assim foi informado para zona local “FIBRE” que a zona remota é o “IFBA” com o seguinte comando: *iadmin mkzone IFBA remote ifba-icat:1247*. Da mesma maneira foi realizado no servidor iCAT do IFBA, inserindo o FIBRE como sua zona remota da seguinte maneira: *iadmin mkzone FIBRE remote fibre-icat:1247*.

Uma vez federadas as zonas, os usuários de uma zona poderão ser usuários da zona remota. No entanto, é necessária a criação e a permissão de acesso do usuário remoto na zona local. Como exemplo, foi criado o usuário igorleonardo, pertencente à zona FIBRE, na zona do IFBA da seguinte maneira: *idmin mkuser igorleonardo#FIBRE rodsuser*.

A autenticação do usuário igorleonardo é realizada na sua zona local, neste caso é o FIBRE. Segue, na Figura 23, a tela do iRODS *Web Browser*, as duas zonas federadas FIBRE e IFBA, onde os usuários de uma zona é usuário remoto da outra zona representado da seguinte forma: *login#zona*, que para o exemplo do usuário Igor Leonardo ficou igorleonardo#FIBRE.

Figura 18 - Federação FIBRE com IFBA



Fonte: Autoria própria (2013).

Com esta simulação foi possível verificar a possibilidade de inclusão de uma nova instituição ao testbed do FIBRE através da federação iRODS, onde o usuário com as credenciais de uma zona pôde gravar dados na zona remota.

6 CONCLUSÃO

A estratégia de armazenamento de dados de monitoramento para uma rede de experimentação multi-CMF utilizando o iRODS como ferramenta de gestão dos dados distribuídos se mostrou viável, conforme demonstrado na prova de conceito. A estratégia permitiu a integração de dados de monitoramento levando em conta a transparência de acesso ao dado, a possibilidade de inserção e consulta de metadados, importante para pesquisa colaborativa, dentre outros atributos técnicos.

A escolha da estratégia tipo 2, que consiste em distribuir os locais de armazenamento dos dados, onde cada ilha passaria a ter um servidor iRODS para o armazenamento dos dados recebidos dos clientes iRODS dos computadores de experimentação, sendo estes servidores gerenciados pelo servidor de catálogo (iRODS iCAT), proporciona o crescimento do *testbed* com a inclusão de novas instituições e também um acesso simples ao dado, através de um diretório lógico, onde os dados distribuídos entre as ilhas do projeto FIBRE são apresentados em um único local. Assim sendo, não se faz necessário acessar os serviços dos CMFs das ilhas para ter acesso ao dado do experimento realizado.

Para configuração das máquinas de experimentação, o uso da aplicação ArmazenaDados simplificou o processo de configuração da aplicação cliente (iRODS iCOMMANDS) e também o envio dos dados coletados, não sendo preciso o experimentador conhecer os comandos necessários para a execução do envio dos dados e metadados. Esta aplicação torna mais amigável o processo de configuração da máquina cliente, sendo exigidas somente as credenciais de autenticação do projeto FIBRE.

Outro fator importante para escolha da estratégia tipo 2 é a possibilidade de incluir novos *testbeds* ou instituições à proposta de armazenamento do projeto FIBRE, através da federação, conforme foi demonstrado, na prova de conceito, ao incluir o *testbed* IFBA ao *testbed* FIBRE.

6.1 TRABALHOS FUTUROS

Uma das propostas para futuros trabalhos, é a utilização do MDIP como o intermediador entre o portal I&M do projeto FIBRE e a aplicação cliente do iRODS, de forma a automatizar o processo de identificação do experimentador, trazendo informações do portal como: *login*, senha, ilha e CMF, tornando totalmente transparente o envio do dado do experimentador. Uma vez configurada a aplicação cliente de forma automática, poderá ser

utilizado o micro-serviço de replicação de dados do iRODS através de uma rotina de tempo para envio.

A segunda proposta é desenvolver para o Portal I&M do FIBRE, chamada à API do iRODS de forma a ter acesso aos dados de experimentação em uma interface única para a arquitetura I&M, não sendo necessária a utilização do cliente iRODS *Web Browser*.

A terceira proposta está na demonstração de implantação do mecanismo de tolerância a falhas, de forma automática, utilizando PgPool nos servidores iRODS iCAT do projeto FIBRE mantendo o servidor de catálogo sempre disponível.

REFERÊNCIAS

- FERNANDES, N. C. et al. Gestão de Identidade em Testbeds Brasileiros para a Internet do Futuro. In: WORKSHOP ON EXPERIMENTAL RESEARCH ON THE FUTURE INTERNET (WPEIF), 4., 2013, Brasília. **Anais...** Brasília, 2013 2013.
- FIBRE. **Future Internet testbed experimentation between Brasil and Europe**. 2013a. Disponível em: <<http://www.fibre-ict.eu/>>. Acesso em: 20 nov. 2013.
- FIBRE. **Testbeds**. 2013b. Disponível em: <<http://www.fibre-ict.eu/index.php/testbeds>>. Acesso em: 26 nov. 2013.
- FIBRE. **OMF Control Monitoring Framework**, 2013c. Disponível em: <<http://www.fibre-ict.eu/index.php/cm/omf>>. Acesso em: 23 out. 2013.
- HADOOP. **Permissions Guide**. 2013a. Disponível em: <http://hadoop.apache.org/docs/r1.2.1/hdfs_permissions_guide.html>. Acesso em: 17 set. 2013.
- HADOOP. **HDFS Architecture Guide**, 2013b. Disponível em: <http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html>. Acesso em: 3 nov. 2013.
- HADOOP. **HDFS Federation**. 2013c. Disponível em: <https://hadoop.apache.org/docs/r2.2.0/hadoop-project-dist/hadoop-hdfs/Federation.html#Managing_the_cluster>. Acesso em: 16 dez. 2013.
- IRODS. **IRODS: Data Grids, Digital Libraries, Persistent Archives, and Real-time Data Systems**. 2013. Disponível em: <<https://www.irods.org>>. Acesso em: 12 set. 2013.
- KÖRNER, M. fp7-ofelia. **Second year report on planning development, testing and operation of individual island**. 2013. Disponível em: <<http://www.fp7-ofelia.eu/assets/Public-Deliverables/OFELIAD42.pdf>>. Acesso em: 10 nov. 2013.
- MARCONDES, C. A. C. et al. Estado da arte de sistemas de controle e monitoramento de infraestruturas para experimentação de redes de comunicação. In: ABELÉM, A. J. G.; ALMEIDA, J. M. D.; NETO, D. O. G. **Minicursos livro texto**. Ouro Preto: Sociedade Brasileira de Computação, 2012. v.1, cap. 3, p. 99-159.
- MARCONDES, C. et al. Fibre-ict. **D2.5 Report on the deployment of the first version of the control and monitoring framework for the FIBRE-BR facilities**. [S.l]: [s.n.], 2013.
- MONTEIRO, J. A. S. Measurement Infrastructures for Future Internet Testbeds. In: WORKSHOP OF THE BRAZILIAN INSTITUTE FOR WEB SCIENCE RESEARCH, 5., 2010, Rio de Janeiro. **Annals...** 2010. p.1-10.
- MOORE, R. W. iRODS. **Introduction to iRODS**, 2010. Disponível em: <https://www.irods.org/pubs/iRODS_Workshop_ISGC_Taiwan_03-2010.pdf>. Acesso em: 15 set. 2013.
- MOOSEFS. **About MooseFS**, 2013a. Disponível em: <<http://www.moosefs.org>>. Acesso em: 6 nov. 2013.

MOOSEFS. **FAQ**. 2013b. Disponível em: <http://www.moosefs.org/moosefs-faq.html#mfscgiserv_authentication>. Acesso em: 19 nov. 2013.

OFELIA. **The EU FP7 Project and The European OpenFlow Experimental Facility**. 2012. Disponível em: <<http://www.fp7-ofelia.eu/assets/Publications-and-Presentations/OFELIASeptember2012.pdf>>. Acesso em: 17 out. 2013.

OFELIA. **OpenFlow in Europe: linking infrastructure and applications**. 2013. Disponível em: <<http://www.fp7-ofelia.eu/about-ofelia/>>. Acesso em: 17 jul. 2013.

OML. **OML - Measurement Library**. 2013. Disponível em: <<http://mytestbed.net/projects/oml>>. Acesso em: 5 set. 2013.

PALMER, M. SDN Central. **Google SDN based inter-datacenter WAN case study**, 2012. Disponível em: <<http://www.sdncentral.com/use-cases/just-released-google-sdn-case-study/2012/07/>>. Acesso em: 3 jun. 2013.

PERFSONAR. **Protocols and Messages**. 2014. Disponível em: <<http://www.perfsonar.net/protocols.html>>. Acesso em: 14 jan. 2014.

PETERSON, L. et al. **Slice-Based Facility Architecture, Draft Version 0.8**. [S.l]: [s.n.], 2007.

PINHEIRO, M. M. et al. An Instrumentation and measurement architecture supporting multiple control monitoring frameworks. **TridentCom**, Thessaloniki, p. 363-364, 2012.

PROTOGENI. **Protogeni**. 2013. Disponível em: <<http://www.protogeni.net/>>. Acesso em: 9 dez. 2013.

RAKOTOARIVELO, T. et al. OMF: A Control and Management Framework for Networking Testbeds. **ACM SIGOPS Operating Systems Review**, New York, v.43, n.27, p.54-59, jan. 2010.

REDDAPANI, S. iRODS. **iRODS High Availability**. 2009. Disponível em: <https://www.irods.org/index.php/iRODS_High_Availability>. Acesso em: 14 out. 2013.

RM-ODP. **International Standard ISO/IEC 19793**: 2008. Disponível em: <http://www.lcc.uma.es/~av/download/UML4ODP_IS_V2.pdf>. Acesso em: 6 nov. 2013.

RUSSELL, T. iRODS. **Resource**. 2012. Disponível em: <<https://www.irods.org/index.php/resource>>. Acesso em: 20 set. 2013.

SCHROEDER, W. iRODS. **Federation Administration**. 2008. Disponível em: <https://www.irods.org/index.php/Federation_Administration>. Acesso em: 15 mar. 2013.

SCHROEDER, W. iRODS. **Imeta**. 2009. Disponível em: <<https://www.irods.org/index.php/imeta>>. Acesso em: 23 out. 2013.

SCHROEDER, W. iRODS. **PAM Authentication**, 2013. Disponível em: <https://www.irods.org/index.php/PAM_Authentication>. Acesso em: 7 nov. 2013.

SDNCENTRAL. **SDN Market to reach \$35B by 2018**. 2013. Disponivel em: <<http://www.sdncentral.com/infographic-sdn-market-to-reach-35b-by-2018/>>. Acesso em: 3 jun. 2013.

TOOBY, P. iRODS. **The iRODS Rule System**. 2008. Disponivel em: <https://www.irods.org/index.php/The_iRODS_Rule_System>. Acesso em: 7 jul. 2013.

VASUDEVAN, V.; PAZANDAK, P. Semantic File Systems. **Motivation for Semantic File Systems**, 2013. Disponivel em: <<Http://www.objs.com/survey/OFSExt.htm> >. Acesso em: 5 dez.2013.

VERMEULEN, B. ICT-FIRE. **Fed4FIRE leader architecture WP - iMinds testlab technical manager**. 2013. Disponivel em: <http://www.ict-fire.eu/fileadmin/events/2013-10_FIRE_Forum/20131024_brechtvermeulen_fireforum.pdf>. Acesso em: 19 dez. 2013.

APÊNDICE A – Configuração do ambiente da prova de conceito

CONFIGURAÇÃO DO SERVIDOR IRODS SERVER ICAT

O processo de instalação do servidor iRODS iCAT poderá ser verificado no *cookbook* de instalação apresentado no apêndice C desta dissertação. Este *cookbook* é o resultado validado do processo de configuração e instalação dos componentes iRODS como proposta para o uso no projeto FIBRE. As configurações realizadas no servidor foram:

- Configuração da ZONA iRODS como FIBRE: no processo de instalação do iRODS é solicitado o nome da ZONA, que por definição foi colocado FIBRE.
- Autenticação do usuário: foram criadas pelo usuário “rods”, administrador do iRODS, cinco contas de usuários com perfil limitado para realização da prova de conceito.

```
iadmin mkuser nomedousuario rodsuser
iadmin moduser nomedousuario password openflow
```

- Criação dos recursos: cada ilha terá o seu recurso criado no servidor iRODS iCAT. É através da configuração do recurso que será definido o local onde serão armazenados os dados dos experimentos, e qual tipo do sistema de arquivo que será utilizado para armazenar os dados. Neste sentido, foi criado neste servidor, um recurso para cada ilha da seguinte forma:

```
iadmin mkresc unifacs-Resc "unix file system" archive
fibre-unifacs-server "/home/operador/iRODS/Vault"
iadmin mkresc ufpe-Resc "unix file system" archive
fibre-ufpe-server "/home/operador/iRODS/Vault"
```

- iadmin mkresc: Comando iRODS para criação do recurso;
- unifacs-Resc e ufpe-Resc: Nome dado ao recurso no padrão nomedailha-Resc;
- unix file system: Tipo do sistema de arquivo utilizado para o armazenamento do dado;
- archive: Classe do tipo arquivo;
- fibre-unifacs-server e fibre-ufpe-server: Nome FQDN do servidor iRODS da ilha;
- /home/operador/iRODS/Vault: Local físico onde os dados serão armazenados no servidor iRODS da ilha.

Este comando será realizado pelo administrador do iRODS, que criará os recursos para todas as ilhas existentes e para as que aderirem ao FIBRE.

- Instalação do Cliente *Web*: para a instalação do iRODS *Web Browser*, foi necessária a instalação do PHP5 e do APACHE2, habilitando o módulo de leitura e escrita do Apache e2enmod. O processo de instalação do iRODS *Web Browser* poderá ser verificado no apêndice C desta dissertação.

CONFIGURAÇÃO DO SERVIDOR IRODS

A configuração do servidor iRODS por ilha, responsável pelo armazenamento dos dados de experimentação, se dá basicamente no processo de instalação (apêndice C), onde será solicitado o nome do servidor iRODS iCAT, que para esta prova de conceito será o fibre-

icat, o nome da zona (FIBRE), o local para o armazenamento dos dados (/home/operador/iRODS/Vault) e o nome do recurso que para o servidor da UNIFACS será unifacs-Resc e para UFPE será ufpe-Resc.

CONFIGURAÇÃO DOS CLIENTES IRODS

Para a preparação das máquinas clientes, necessárias para o envio dos dados de experimentação e disponibilizadas pelos CMFs ao pesquisador, as mesmas já virão com o cliente iRODS icommands e suas configurações prévias, como o serviço de autocompletar do Linux para os comandos iRODS, que adicionará no arquivo “.profile” do *home* do usuário o endereço dos executáveis do icommands.

Exemplo: `PATH=/home/root/iRODS/clients/icommands/bin:$PATH`

No entanto, as máquinas de experimentação cedidas pelos CMFs ao experimentador, não possuem informações de autenticação do mesmo, sendo o acesso feito por um usuário e senha padrão. Estas máquinas também não vêm com as variáveis de ambiente do iRODS configuradas, até por não se saber quem é o experimentador que está fazendo uso delas. Para que o iRODS seja configurado na máquina de experimentação demandaria do experimentador configurar as variáveis de ambiente do iRODS e ter um conhecimento mais técnico do ambiente de armazenamento dos dados, além de uma série de comandos iRODS para autenticação e envio dos dados de experimentação, o que definitivamente não é o foco de pesquisa deste experimentador.

Não é objeto desta implantação que o experimentador saiba utilizar os comandos iRODS e saiba configurar as variáveis de ambiente do cliente iRODS para fazer o envio dos seus dados de experimento para o servidor iRODS. Neste sentido, visando disponibilizar uma interface amigável e mais transparente ao experimentador, foi desenvolvida uma aplicação em *Shell Script* com o nome de ArmazenaDados (apêndice B). Através dela será possível fazer todas as configurações na variável de ambiente do iRODS, de forma simples e sem fazer uso dos comandos iRODS. Esta aplicação ficará no *Path* do Linux, e poderá ser executada pelo usuário experimentador de qualquer diretório do sistema operacional. Será apresentado um menu com opções necessárias para configurar a máquina de experimentação. Esta aplicação realizará os seguintes procedimentos:

- Autenticação: Solicita a autenticação deste experimentador, armazenando o *hash* da senha criptografada de forma a não precisar se autenticar novamente.
- Instrumentação da máquina: Configura as variáveis de ambiente do iRODS com as informações do experimentador, ilha e CMF, verifica a existência dos diretórios /CaixadeEntrada e /CaixadeSaida, que por padrão já virão pré-configurados na imagem cedida pelo CMF.
- Solicita ao experimentador o armazenamento do dado juntamente com o metadados, onde alguns metadados já serão inseridos por padrão, como nome da Ilha e tipo de CMF e possibilita ao experimentador colocar alguma descrição sobre o dado.
- Consulta os dados inseridos no servidor iRODS.

Para refletir o ambiente das máquinas de experimentação dos CMFs na prova de conceito, foram utilizadas VMs com usuário padrão de root e senha “openflow” de forma semelhante às utilizadas pelos CMFs em produção.

APÊNDICE B – Código Fonte do Armazenados

```
#!/bin/bash
#script backup
Main_Menu()
{
zona=FIBRE
ilha=unifacs
cmf=ofelia
opcao=1
clear
while [ $opcao -ne 0 ]
do
echo
echo "!!! ESCOLHA UMA OPCAO !!!"
echo "1. Insira o usuario de autenticacao do FIBRE"
echo "2. Deseja armazenar os dados persistentemente?"
echo "3. Visualizar arquivos armazenados"
echo "0. Sair"
echo
echo -n "Opcao: ";

read opcao
case $opcao in
"1")
clear
# Verificando o diretorio da Variavel de Ambiente do iRODS"
iRODSdir=~/.irods
if [ ! -d $iRODSdir ]; then
# "Criando Diretorio $iRODSdir"
`mkdir -p $iRODSdir`
touch $iRODSdir/.irodsEnv
# if [ $? -eq 0 ]; then
# echo "Diretorio de ambiente criado com sucesso."
# fi
# else
# echo "Diretorio ambiente iRODS: STATUS OK"
fi
# Deslogando usuario caso exista
iexit
mv -f ~/.irods/.irodsEnv ~/.irods/.irodsEnvOld
echo
echo "Digite seu login: ";
read nome
# Configurando as variaveis de ambiente do cliente iRODS
touch ~/.irods/.irodsEnv
echo "irodsHost 'fibre-$ilha-server'" >> ~/.irods/.irodsEnv
echo "irodsPort 1247" >> ~/.irods/.irodsEnv
```

```

echo "irodsDefResource '$ilha-Resc'" >> ~/.irods/.irodsEnv
echo "irodsHome '$zona/home/$nome'" >> ~/.irods/.irodsEnv
echo "irodsCwd '$zona/home/$nome'" >> ~/.irods/.irodsEnv
echo "irodsUserName '$nome'" >> ~/.irods/.irodsEnv
echo "irodsZone '$zona'" >> ~/.irods/.irodsEnv
# Autenticando o usuario no iRODS
iinit

iuserinfo $nome > /dev/null
if [ $? -eq 0 ]; then
    echo "!!!USUARIO $nome AUTENTICADO!!!";
else
    echo
        echo "!!!USUARIO $nome NAO FOI ENCONTRADO!!!";
fi
;;

"2")
clear
echo "ARMAZENAMENTO DE DADOS PERSISTENTE";
echo "=> Verificando a existencia das Caixas de Entrada e Saida"
cxentrada="/CaixadeEntrada"
cxsaida="/CaixadeSaida"
# Verificando a existencia da Caixa de Entrada
if [ ! -d $cxentrada ]; then
# echo "Criando Diretorio $cxentrada"
`mkdir -p $cxentrada`
if [ $? -eq 0 ]; then
    # echo "Diretorio $cxentrada criado com sucesso!!!"
    echo "Caixa de Entrada: STATUS OK"
fi
else
    echo "Caixa de Entrada: STATUS OK."
fi
# Verificando a existencia da Caixa de Saida
if [ ! -d $cxsaida ]; then
# echo "Criando Diretorio $cxsaida"
`mkdir -p $cxsaida`
if [ $? -eq 0 ]; then
    # echo "Diretorio $cxsaida criado com sucesso!!!"
    echo "Caixa de Saida: STATUS OK"
fi
else
    echo "Caixa de Saida: STATUS OK."
fi
# Verificando arquivos na Caixa de Entrada para o armazenamento persistente
echo
echo "=> Verificando arquivos na Caixa de Entrada"
numarq=`ls $cxentrada | wc -l`
while [ $numarq -ne 0 ]

```

```

do
    if [ $numarq != 0 ]; then
        echo "Existe(m) $numarq arquivo(s) para ser(em) enviado(s) ..."
        ls $cxentrada > /var/tmp/tmpstorage;
        arquivo=`sed -n 1p /var/tmp/tmpstorage`;
        # Armazenando o arquivo persistentemente
        iput -f $cxentrada/$arquivo;
        echo "Armazenando o arquivo ($arquivo) persistentemente..."
        # Inserindo metadado ao dado informado a Ilha e o CMF
        imeta add -d $arquivo ilha $ilha
        imeta add -d $arquivo cmf $cmf
        echo "Inseridos os metadados ilha=$ilha e cmf=$cmf"
        echo "Deseja inserir outro metadado ao dado "$arquivo"?"
        echo " Digite: "1" para Sim e "2" para Nao"
        read op
        if [ $op -eq 1 ]; then
            echo "Insira um Atributo de NOME"
            read attnome
            echo "Insira um Atributo de Valor"
            read attvalor
            imeta add -d $arquivo $attnome $attvalor
        fi
        # Movendo o arquivo da Caixa de Entrada para Caixa de Saida
        mv -f $cxentrada/$arquivo /$cxsaida;
    else
        echo " Nao existem arquivos para serem enviados "
    fi
    numarq=${$numarq-1}
done
echo "Nao existem arquivos para serem enviados"
;;
"3")
clear
ils -l
;;
"0")exit
;;
*) echo "Opcao desconhecida. Digite a opcao correta";
esac
done
}
Main_Menu

```

APÊNDICE C – Cookbook de Instalação

Este *cookbook* é voltado para implantação do serviço de armazenamento dos dados de monitoração que estarão distribuídos nas diversas ilhas do FIBRE. Para a sua realização foi dividido o processo de instalação em quatro etapas.

1. **INSTALAÇÃO E CONFIGURAÇÃO DO IRODS SERVER ICAT:** Servidor responsável pela autenticação, permissão de acesso, criação de recursos e catalogar os dados e metadados, entre outros. A ideia é que este servidor fique em uma das ilhas do FIBRE.
2. **INSTALAÇÃO DO IRODS SERVER:** Servidor que estará armazenando os dados persistentemente ou não do experimentador. Será instalado um servidor deste por ilha.
3. **CONFIGURAÇÃO DO CLIENTE IRODS:** Estarão instalados basicamente nas máquinas de experimentação dos CMFs e nos MDIPs. Este cliente irá se autenticar com o iRODS Server iCAT e enviar os dados de monitoração para o iRODS SERVER. Este cliente também terá a função de inserir metadados para os dados armazenados.
4. **FEDERAÇÃO IRODS FIBRE:** Outras instituições poderão se juntar ao FIBRE, assim, serão apresentadas as configurações necessárias para a comunicação entre as zonas FIBRE e IFBA.

Considerações:

- Todas as máquinas instaladas no laboratório para o desenvolvimento deste *cookbook* utilizaram o Linux Debian 7, 64 bits, com o iRODS 3.3.
- Abstraímos neste documento o processo de instalação do sistema operacional.

Para instalar os servidores iRODS configure o Debian da seguinte forma:
Primeiro, informe os nomes dos repositórios do Debian no arquivo “source list”.

```
vim /etc/apt/source.list
```

```
deb http://debian.pop-sc.rnp.br/debian/ wheezy main
deb-src http://debian.pop-sc.rnp.br/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main
deb http://debian.pop-sc.rnp.br/debian/ wheezy-updates main
deb-src http://debian.pop-sc.rnp.br/debian/ wheezy-updates
main
```

Segundo, instale os pacotes necessários para o funcionamento do iRODS:

```
apt-get install perl gcc make g++
```

Terceiro, crie uma conta de usuário local no Debian com perfil de *user*, que neste documento foi criado o usuário “operador”.

Quarto, faça o download da aplicação iRODS, através do seu site, e coloque o instalador na pasta /home/operador/irods3.3.tgz

Site do iRODS: <https://www.irods.org/index.php/Downloads>

Depois de realizadas as quatro etapas, inicie o processo de instalação dos servidores iRODS SERVER e o iRODS SERVER iCAT, utilizando o usuário operador do linux.

```
su operador
tar -xvzf irods3.3.tgz
cd iRODS
./irodssetup
```

A partir desta etapa os processos de instalação dos servidores iRODS iCAT e o Servidor iRODS se divergem como segue.

1. INSTALAÇÃO E CONFIGURAÇÃO DO IRODS SERVER ICAT:

1.1. PASSO-A- PASSO DA INSTALAÇÃO

Ao executar o script “./irodssetup” uma série de perguntas serão exibidas e que deverão ser respondidas como segue.

```
Include additional prompts for advanced settings [no]? enter
Build an iRODS server [yes]? enter
Make this Server ICAT-Enabled [yes]? enter
iRODS zone name [tempZone]? FIBRE enter
iRODS login name [rods]? enter
Password [rods]? openflow enter
Download and build a new Postgres DBMS [yes]? enter
New Postgres directory? /home/operador/postgres enter
New database login name [operador]? enter
Password? Openflow enter
PostgreSQL version [postgresql-9.2.4.tar.gz]? enter
ODBC version [unixODBC-2.2.12.tar.gz]? enter
Include GSI [no]? enter
Include Kerberos [no]? enter
Include the NCCS Auditing extensions [no]? enter
Save configuration (irods.config) [yes]? Enter
Start iRODS build [yes]?
```

Um resumo da configuração será exibido ao termino da instalação.

```
Confirmation
-----
Please confirm your choices.

-----
Build iRODS data server + iCAT metadata catalog
  directory  '/home/operador/iRODS'
  account    'rods'
  password   'openflow'

Build Postgres
```

```

host      'localhost'
directory '/home/operador/postgres'
account   'operador'
password  'openflow'
pg version 'postgresql-9.2.4.tar.gz'
odbc version 'unixODBC-2.2.12.tar.gz'
control   start & stop along with iRODS servers

```

```

GSI not selected
Kerberos not selected
NCCS Audit Extensions not selected
Build iRODS command-line tools

```

Depois de realizada a instalação, adicione no arquivo “.profile” do *home* do usuário, o PATH para facilitar a execução dos comandos.

```

vim ~/.profile
Insira a seguinte linha no final do arquivo.
PATH=/home/operador/iRODS/clients/icommands/bin:$PATH

```

Em seguida reinicie o servidor.

Para inicializar, parar e reinicializar o Servidor iRODS, utilize o *script* ‘irodsctl’ que fica dentro da pasta iRODS da seguinte forma:

```

./irodsctl start
./irodsctl stop
./irodsctl restart

```

Obs.: Para inicializar o servidor iRODS automaticamente, utilize o *script*, vide tópico 5 “SCRIPT DE INICIALIZAÇÃO DO iRODS SERVER”.

1.2. CRIAÇÃO DOS RECURSOS

O recurso é formado pelo nome, tipo, classe, host, e o path. Para o projeto FIBRE será criado um recurso para cada ilha.

- Nome: Nome do recurso
- Tipo: Sistema de arquivo
- Classe: pode ser tipo arquivo, cache, ..., .
- Host: Nome do servidor que irá armazenar os dados
- Path: local de armazenamento do dado

Abaixo um exemplo de criação de um recurso para a ilha UNIFACS que deverá ser replicada para o restante das ilhas, substituindo o nome UNIFACS pelo nome da outra ilha. Crie o recurso conforme a ilha que estiver configurando. Exemplo: UFPE, USP, UFRJ, UFG, etc.

```

iadmin mkresc unifacs-Resc "unix file system" archive fibre-
unifacs-server "/home/operador/iRODS/Vault"

```

1.3. CRIAÇÃO DE USUÁRIO NO iRODS

Para criação de usuários no iRODS, deverá atender aos dois perfis de usuário, o administrador “rodsadmin” e o usuário comum “rodsuser”. Crie o usuário para uso no iRODS com perfil comum. Segue o exemplo do comando.

Exemplo:

```
iadmin mkuser igorluiz rodsadmin ou rodsuser
iadmin moduser igorluiz password openflow
```

Obs.: Se o iRODS estiver se autenticando via LDAP, que não é objeto deste *cookbook*, será necessário criar somente a conta do usuário na base do iRODS com o mesmo userid do LDAP e a senha será consultada via PAM do iRODS.

1.4. CONFIGURANDO O CLIENTE iRODS *Web Browser*

Para configurar o cliente iRODS *Web Browser*, será necessário instalar alguns pacotes para o seu funcionamento como: o apache2, php5, php5-xmllrpc e unzip. Instale os pacotes da seguinte forma:

```
apt-get install apache2 php5 php5-xmllrpc unzip
```

Habilite o apache para o modo de leitura e gravação.

```
cd /usr/apache2/modules
a2enmod rewrite
```

Configure o apache, em sites-available, alterando o arquivo *default* como segue.

```
vim /etc/apache2/sites-available/default
<VirtualHost *:80>
  RewriteEngine On
  RewriteOptions Inherit
  DocumentRoot /var/www/rodsweb
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /var/www/rodsweb>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
  </Directory>
```

Faça o *download* do arquivo *irodsphp.zip* e coloque dentro do diretório do iRODS

```
wget
https://code.renci.org/gf/download/frsrelease/158/1232/irodsphp.zip
```

Descompacte o arquivo *irodsphp.zip*

```
unzip irodsphp.zip
```

Acesse o diretório *irodsphp* e o seu subdiretório *web*

```
cd /home/operador/iRODS/irodsphp/web
```

Descompacte o arquivo extjs.zip

```
unzip extjs.zip
```

Ao descompactar o arquivo extjs.zip será criada uma pasta como o nome “ext-1.1.1” que deverá ser renomeada para “extjs”.

```
mv ext-1.1.1/ extjs/
```

Crie um *link* simbólico em /var/www apontando para o local onde se encontra o site.

```
ln -s ~operador/iRODS/irodsphp/web rodsweb
```

Reiniciar o apache

```
/etc/init.d/apache2 restart
```

2. INSTALAÇÃO DO IRODS SERVER:

Ao executar o script “./irodssetup” uma série de perguntas serão exibidas e que deverão ser respondidas como segue.

```
Include additional prompts for advanced settings [no]? enter
Build an iRODS server [yes]? enter
Make this Server ICAT-Enabled [yes]? no enter
Host running iCAT-enabled iRODS server? fibre-icat enter
Resource name [demoResc]? unifacs-Resc enter
Resource storage area directory [/home/operador/iRODS/Vault]? enter
Existing iRODS admin login name [rods]? enter
Password [rods]? openflow enter
iRODS zone name [tempZone]? FIBRE enter
Include GSI [no]? enter
Include Kerberos [no]? enter
Include the NCCS Auditing extensions [no]? enter
Save configuration (irods.config) [yes]? enter
Start iRODS build [yes]? enter
```

Um resumo da configuração será exibido ao termino da instalação.

```
Confirmation
-----
Please confirm your choices.

-----
Build iRODS data server without iCAT
iCAT host   'fibre-icat'
directory  '/home/operador/iRODS'
account    'rods'
password   'openflow'

GSI not selected
```

```
Kerberos not selected

NCCS Audit Extensions not selected

Build iRODS command-line tools
```

Depois de realizada a instalação, adicione no arquivo “.profile” do *home* do usuário, o PATH para facilitar a execução dos comandos.

```
vim ~/.profile
Insira a seguinte linha no final do arquivo.
PATH=/home/operador/iRODS/clients/icommands/bin:$PATH
```

Em seguida, reinicie o servidor.

Para inicializar, parar e reinicializar o Servidor iRODS, utilize o *script* ‘irodsctl’ que fica dentro da pasta iRODS da seguinte forma:

```
./irodsctl start
./irodsctl stop
./irodsctl restart
```

Obs.: Para inicializar o servidor iRODS automaticamente, utilize o *script*, vide tópico 5 “SCRIPT DE INICIALIZAÇÃO DO iRODS SERVER”.

3. CONFIGURAÇÃO DO CLIENTE iRODS

Faça o download do iRODS.

Site do iRODS: <https://www.irods.org/index.php/Downloads>

Em seguida proceda com a descompactação do iRODS. Será necessário instalar um descompactador e realizar os seguintes passos:

```
apt-get install unzip
tar -xvzf irods3.3.tgz
cd iRODS/clients
mv icommands /etc/
```

Depois de inserir a aplicação cliente no diretório /etc/icommands será necessário adicionar no arquivo “.profile” do home do usuário, o PATH para facilitar a execução dos comandos.

```
vim ~/.profile
Insira a seguinte linha no final do arquivo.
PATH=/etc/icommands/bin:$PATH
```

E por fim, coloque a aplicação ArmazenaDados (apêndice B) no diretório /etc/icommands/bin/

Altere a permissão do arquivo ArmazenaDados.

```
chmod 755 armazenadados
```

Com esta aplicação será possível configurar as variáveis de ambiente do iRODS e realizar a autenticação do usuário experimentador.

4. FEDERAÇÃO iRODS

Outras instituições/zonas poderão se juntar ao FIBRE, cada uma com o seu sistema de autenticação. Neste caso, será necessário criar uma federação entre as zonas. Segue o exemplo da federação entre as zonas FIBRE e IFBA.

No servidor de metadados de cada zona, o administrador do iRODS deverá através do cliente icommands, informar o servidor remoto da zona a ser federada, executando os seguintes comandos:

```
Federando as zonas FIBRE com o IFBA
Na zona FIBRE
iadmin mkzone IFBA remote ifba-icat:1247
Na zona IFBA
iadmin mkzone FIBRE remote fibre-icat:1247
```

Onde ifba-icat:1247 e fibre-icat:1247 representa o nome do servidor e a porta de acesso. Para que usuários de uma zona tenha acesso à zona remota, será necessária a criação deste usuário na zona remota da seguinte forma:

```
Na zona FIBRE será criado a conta de usuário (Aline) que pertence à zona remota IFBA.
iadmin mkuser aline#IFBA rodsuser
```

Assim, a usuária Aline, com a sua senha de acesso, poderá utilizar o sistema de armazenamento de dados da outra zona.

De forma semelhante, poderá ser criado na zona IFBA um usuário pertente à zona FIBRE.

5. SCRIPT DE INICIALIZAÇÃO DO IRODS SERVER

Crie um *script* para inicializar o iRODS assim que o servidor for ligado. Primeiro deve instalar o pacote sudo.

```
# apt-get install sudo
```

Crie um script com permissão 755 colocando na pasta de inicialização do Linux.

```
vim /etc/init.d/iniciandoirods
```

```
#!/bin/bash
iRODS_HOME=/home/operador/iRODS

start(){
  echo "Iniciando o irods..."
  sudo -u operador $iRODS_HOME/irodsctl start
}

stop(){
```

```
echo "Parando o irods..."
sudo -u operador $IRODS_HOME/irodsctl stop
}

restart(){
  stop
  start
}

case "$1" in
  start)
  start
  ;;
  stop)
  stop
  ;;
  restart)
  restart
  ;;
  *)
  echo "Utilize: iniciandoirods {start|stop|restart}"
  exit 1
esac
exit 0
```

```
chmod 755 iniciandoirods
update-rc.d iniciandoirods defaults
```

Depois de inserido o *script* “inicializandoirods” na inicialização do Linux, reinicie esta máquina.