

# Acesso Ubíquo a Arquivos no *Middleware* EXEHDA

Vilnei Neves<sup>1</sup>, Nelsi Warken<sup>3</sup>, Adenauer Corrêa Yamin<sup>1,2</sup>,  
Luthiano Venecian<sup>1</sup>, Sérgio Rodrigues<sup>1,4</sup>

<sup>1</sup>Centro Politécnico – Univ. Católica de Pelotas (CPoli/UCPel)

<sup>2</sup>Centro de Desenvolvimento Tecnológico - Univ. Fed. de Pelotas (CDTec/UFPel)

<sup>3</sup>Núcleo de Tecnologia da Informação - Embrapa Clima Temperado

<sup>4</sup>Instituto Federal Sul-rio-grandense (IFSul)

{vilnei.neves, adenauer, venecian, nelsi.warken, sergiopelotas}@gmail.com

*Resumo. Na Computação Ubíqua, a facilidade para o armazenamento e o acesso a arquivos é um aspecto de grande relevância para sua viabilidade operacional. Este trabalho tem como objetivo central apresentar a proposta de um serviço integrado ao middleware EXEHDA, denominado EXEHDA-SSF (EXEHDA Safe Share Facility), que busca prover um ambiente que permita aos usuários acesso aos seus arquivos em uma perspectiva ubíqua. O trabalho está em andamento, tendo sua fase de modelagem concluída.*

## 1. Introdução

Este trabalho está inserido nos esforços de estudo e pesquisa em andamento no Grupo de Pesquisa em Processamento Paralelo e Distribuído (G3PD/CPoli/UCPel) relacionados a Computação Ubíqua.

A Computação Ubíqua idealiza a existência de ambientes carregados de dispositivos computacionais, que se integram ao cotidiano dos usuários de forma transparente, adaptando-se automaticamente de acordo com as situações observadas no contexto de seu interesse. Deste modo, a computação não seria exclusividade de um computador, mas de diversos dispositivos conectados entre si, formando uma malha de dispositivos inteligentes em torno de cada indivíduo. A Computação Ubíqua, considerando esta perspectiva, constitui um ambiente altamente distribuído, heterogêneo, dinâmico, móvel, mutável e com forte interação entre homem e máquina.

O EXEHDA (*Execution Environment for Highly Distributed Applications*) é um *middleware* direcionado às aplicações distribuídas, móveis e conscientes do contexto da Computação Ubíqua [Yamin 2004]. Este *middleware* é baseado em serviços, e tem por objetivo criar e gerenciar um ambiente ubíquo.

O presente trabalho tem como meta prover o *middleware* EXEHDA de facilidades para acesso, edição, registro e transferência de arquivos em um ambiente ubíquo, e é direcionado a atender as demandas do projeto plenUS (*plentiful Ubiquitous Systems*) [YAMIN 2010], cujo objetivo é disponibilizar Sistemas Ubíquos, que explorem relações pró-ativas entre usuários, softwares e equipamentos, com intuito de promover soluções computacionais que contribuam de forma sinérgica no atendimento das atividades de pesquisa da Embrapa Clima Temperado.

## 2. Sistemas de Arquivos Distribuídos

Nesta seção são discutidos alguns dos principais projetos relacionados aos Sistemas de Arquivos Distribuídos. Uma das referências utilizadas na organização desta seção foi [Madruca 2008]. A revisão destes trabalhos relacionados foi central na concepção do EXEHDA-SSF.

### 2.1. Google File System (GFS)

O *Google File System* propõe um sistema de arquivos que atende os requisitos de uma maior confiabilidade e capacidade de manuseio eficiente de grandes quantidades de dados. O *Google File System* (GFS) foi desenvolvido a partir de observações na carga e necessidades de aplicações específicas da *Google Incorporated*. Dentre suas características, destacam-se:

- arquitetura baseada em *cluster* composta de um servidor mestre (*single master*) e vários escravos (*chunckservers*);
- arquivo dividido em partes (*chunks*) e distribuído entre os *chunckservers*;
- o servidor mestre administra todas as informações do sistema: controle de acesso, localização dos *chunks*, etc;
- tolerância a falhas é proporcionada através de um sistema de sucessivas replicações e de recuperação rápida.

### 2.2. Ivy File System

*Ivy* é um sistema de arquivos baseado em um modelo *peer-to-peer* puro, com elevada escalabilidade quanto ao número de usuários simultâneos.

Principais características:

- baseado no modelo *peer-to-peer*;
- utiliza *logs* armazenados em tabelas de hash distribuídas (*DHT*) como estratégia de coordenação;
- todos participantes consultam todos os *logs*, mas só escrevem no próprio *log*;
- possui mecanismo de controle de versões para resolução de conflitos de arquivos;
- tolerância a falhas através do uso *snapshot*, onde cada participante mantém um retrato do estado atual de todos os *logs* do sistema.

### 2.3. OceanStore

*OceanStore* não é propriamente um sistema de arquivos distribuídos, mas sim uma estrutura global que fornece acesso contínuo à informações persistentes, visando atender as demandas típicas da Computação Ubíqua. Contudo, seus conceitos permitem que seja criado um sistema de arquivos distribuídos usando essa infra-estrutura. Possui duas premissas: (i) executar sobre uma infraestrutura não confiável e (ii) suportar a migração de dados entre os participantes.

Principais características:

- formado por *pools* de servidores próximos;
- dados armazenados na forma de *persistent objects*, identificados através de GUID e replicados entre vários servidores - *floating replicas*;
- atualização através de operações de *update*, que cria uma nova versão do objeto com as mudanças - recuperação rápida;
- tolerância a falhas através de verificação constante dos estados *links* e rotas para réplicas.

## 3. EXEHDA-SSF: Aspectos de Gerais da Concepção

Esta seção apresenta uma proposta de modelagem para o serviço EXEHDA-SSF (*EXEHDA Safe Share Facility*), que considerando as premissas da Computação Ubíqua, busca desenvolver um ambiente que permita ao usuário acesso, manipulação e compartilhamento aos seus arquivos e/ou pastas.

### 3.1. Arquitetura de Software Proposta

A arquitetura do EXEHDA-SSF é constituída de dois componentes: (i) *EXEHDA-SSF Server* e o (ii) *EXEHDA-SSF Client*. Cada componente possui um conjunto de módulos que são responsáveis por uma série de ações que constituem um cenário perseguido no *middleware* EXEHDA, e estão resumidos a seguir:

**Módulo de Acesso:** é responsável por realizar o controle de acesso aos arquivos. Aspectos como a autenticação de usuários e nodos e controle de permissões de acesso e de compartilhamento dos arquivos armazenados, estão contemplados nesse módulo.

**Módulo de Comunicação:** realiza a comunicação entre os nodos através de *Web Services*, e gerencia os protocolos de transferência de dados;

**Módulo de Manipulação de Arquivo:** responsável pela operações sobre os arquivos, bem como da sincronização, gerenciamento de conflitos e controle de versão dos mesmos;

**Módulo de Catálogo:** armazena as informações sobre usuários, arquivos e nodos que utilizam o EXEHDA-SSF e servirá de base de consulta para os demais módulos do serviço;

**Módulo Web:** esse módulo, alojado no *EXEHDA-SSF Server*, é responsável por prover uma interface Web para acesso do usuário aos arquivos armazenados no *EXEHDA-SSF Server*, bem como suas preferências e configurações. Além disso, ele proverá uma interface Web para administração do ambiente de acesso ubíquo a arquivos;

**Módulo de Integração a GUI:** esse módulo, que reside no *EXEHDA-SSF Client*, é responsável por realizar a integração do serviço a interface gráfica do dispositivo do usuário, garantindo ao usuário, maior transparência na manipulação e compartilhamento de seus arquivos.

A figura 1 demonstra a organização da arquitetura do EXEHDA-SSF. Um detalhamento da arquitetura proposta e a discussão de seus módulos está disponível em [Neves 2010].

### 3.2. Principais Funcionalidades do Modelo Proposto

O modelo proposto, EXEHDA-SSF, contempla as seguintes funcionalidades: (i) prover acesso ubíquo a arquivos e pasta de qualquer equipamento gerenciado pelo *middleware* EXEHDA; (ii) permitir sincronização e compartilhamento de arquivos/pastas de forma automática e configurável; (iii) ser adaptativo e sensível ao contexto; (iv) executar em múltiplas plataformas; (v) possuir mecanismos que garantam a segurança e a privacidade dos dados; (vi) gerenciar conflitos de arquivos e possuir controle de versão; (vii) prover suporte a desconexão; (viii) garantir a integridade dos dados que são armazenados e/ou transmitidos; (ix) permitir acesso, configuração e administração via interface Web; (x) possuir mecanismos de backup e replicação.

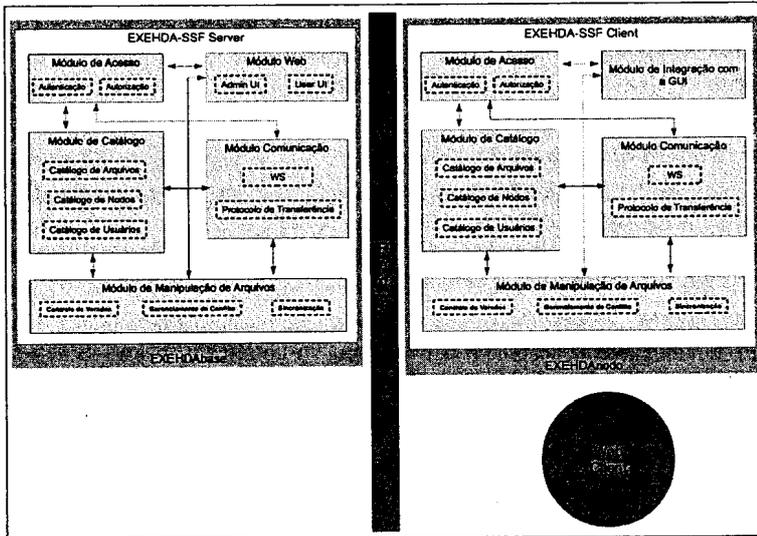


Figura 1. Arquitetura do EXEHDA-SSF

#### 4. Conclusões

Os estudos realizados até o momento contribuíram com subsídios para compreensão dos conceitos pertinentes ao armazenamento de arquivos em uma perspectiva distribuída, bem como a relação desses conceitos com características desejadas na *UbiComp*, e motivaram a concepção de um serviço que atendesse as demandas da *UbiComp* relacionadas ao acesso a arquivos.

A concepção de um serviço de tal natureza, implica em uma série de desafios decorrentes do gerenciamento da mobilidade do usuário e de suas aplicações, bem como outras características inerentes da Computação Ubíqua. O trabalho está em andamento, e um detalhamento maior do que foi desenvolvido pode ser encontrado em [Neves 2010].

#### Referências

- Madruga, M. C. (2008). *Uso de Arquivo Transparentes na Construção de Um Sistema de Arquivos Distribuídos a Faltas*. Tese - Mestrado em Ciência da Computação, Pontifícia Universidade Católica do Paraná, Curitiba, SP.
- Neves, V. M. F. (2010). *Acesso ubíquo a arquivos no Middleware EXEHDA*. Projeto de graduação I (Bacharelado em Ciência da Computação), CPOLI/UCPEL, Pelotas, RS.
- Yamin, A. C. (2004). *Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. PhD thesis, Universidade Federal do Rio Grande do Sul.
- YAMIN, A.C; WARKEN, N. (2010). Projeto plenus: plentiful of ubiquitous systems. Disponível em: < <http://plenus.ucpel.tche.br/> >. Acesso em 10/2010.