21/12/2005

THE NORDUGRID "SMART" INDEXING SERVICE
FLOATING OBJECTS

Description

A.Konstantinov*

*aleks@fys.uio.no

Contents

1 Common

- Lessons learned from installing (sorry, deploying) and using RC and RLS.

RC Never use anything You can't control. LDAP+DB was not a best solution from flexibility point of view.

RLS Even good idea can be killed by poor implemnetation.

Fancy features are usually never used. Have anybody ever used attribute types other than ordinary string?

- Requirements
  - More flexibility in access control. And more flexibility everythere.
  - Set of services is supposed to be quite static, so no P2P tricks are needed. Services are put into hierarchical mesh. But same service can be at different levels of hierarchy for diferent objects.
  - No immediate propagation of updates is needed.

2 Levels

System is defined in 3 levels.

2.1 Level 0

Set of services with integrated capabilities to talk to each other. Preferably same interface everythere.

Services are added to system by registering themselves to other services. Registration probably includes information about what kind of object (described below) service wants to host.

2.2 Level 1

Objects are active in sense that type of object and not type of service defines which operations are performed on them and linked objects.

There are ordinary objects (*objects*) and reference objects (*refobjects*).

Objects contain:

Type

ID - unique in whole Grid. One ID is reserved for top level object.

links - to refer to other objects. Each link contains ID, name and URLs of services with per-URL state.

backlinks - links to objects which want to receive information about object's modification. Typically those are parent objects, refobjects and replicas of object.

attributes - properties of object of name=value kind. Typical property is a 'modification time'.

Objects take care of creating/linking to refobjects and provide interface for propagating update notifications and defining actions for those notifications.

Reference objects are used to link objects through higher level services (equivalent of RLI-LRC link). They are also responsible for synchronising updates in objects, maintaining replicas of objects and directing clients to real instances.

Refobjects contain:

Type

ID - same as of ordinary object

Links - to ordinary objects

Maybe also backlinks. I'm not sure about that.

2.3 Level 2

This level defines types of objects, minimal set of attributes and behavior.

Object types:

Collection - it links to objects of different types and represents a way to group objects. Backlinks of linked objects point to it. Predefined attributes are 'name' (optional), 'modification time' and 'access control'. Other attributes may be project specific and/or created dynamically.

Knot - represents data file. Predefined attributes are 'name' (optional), 'modification time', 'access control', 'size' and 'hash' (type of hash is encoded in value). It links to physical instances/replicas of file.

File - file itself if SE has functionality of Indexing Service.

Operations performed by objects:

Collection - accept modification of links and send modification notifications through refobjects, modify backlinks on request.

Knot - initiate new replicas on request and automatically (for backward compatibility also accept newreplicas). Send modification notifications and pass them to SEs (mostly for access control).

By adding new types of objects different kinds of data entities can be supported. Like database fields for example.

File is referenced through any valid chain #ID/link name/.../link name, with ID either top level reserved ID or any ID known in advance (like ID reserved for VO).

References