NORDUGRID-TECH-10

15/12/2005

THE NORDUGRID "SMART" STORAGE ELEMENT

Description and Administrator's Manual

A.Konstantinov*

*aleks@fys.uio.no

Contents

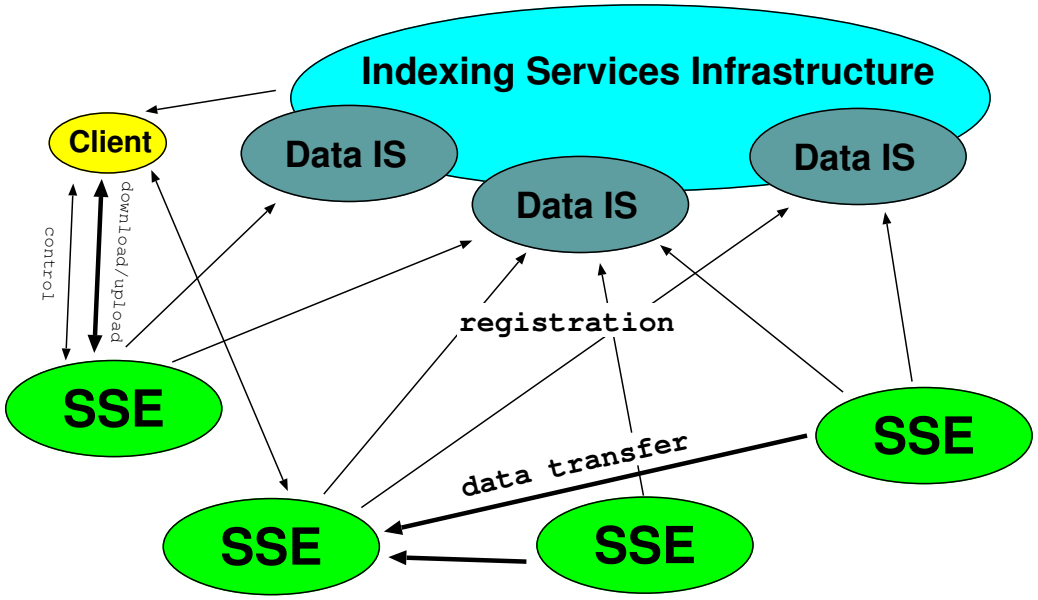

Figure 1: Architecture of Data Storage Infrastructure

1 Introduction

A ”Smart “ Storage Element (SSE) is supposed to be autonomous service implementing at least most important data management functions without user’s intervention. It should be also capable of resolving failure situation in safest way. It is part of HTTPSD server. So please read this manual first [1].

The SSE is developed by NorduGrid project[2] as part of ARC software.

2 Architecture

Main intention of SSE is to serve as part of full data storage system. Following are main features of SSE and Data Storage Infrastructure which can be implemented using SSE (Fig.1):

- SSE is only part of full infrastructure and must be complemented with Data Indexing Services (IS).
- IS can be either common purpose or application specific. Modular internal architecture may be used to provide interface to unsupported IS type.
- SSE has no internal structure for storing data units (files). Files are identified by their identity used in IS (Logical File Name, GUID, Logical Path, etc.)
- All operations on data units (creation, replication, deletion of replica, etc.) are done on request of Client through SSE. Hence all operations are atomic from from Client’s point of view.
- Entry points to IS infrastructure should be used by Client only for operations not involving data storage units.
- Access to data units on SSE is based on Grid Identity of Client and uses GACL [3] for flexibility.
- SSE implements HTTPS and HTTPg protocols for data transfer.
- Data transfer between SSEs is done on request of Client by SSEs (third-party transfers).

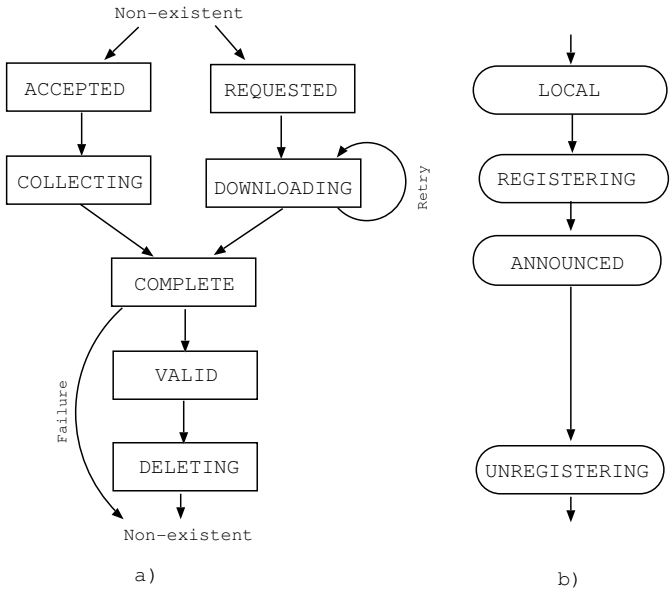

Figure 2: Execution flow.

3 How it works internally

Each data unit (a file) passes few stages inside SSE. Those can be divided into two threads - related to a state of content (Fig.2a) and registration (Fig.2b).

Below is description of all actions taken by the SSE at every state:

- **Accepted** - the SSE just received request from client to create a slot for a new file. This file will be filled by client which can present same credentials.
- **Requested** - the SSE received request to create and download new file. Only difference from **accepted** is that file has one or more *sources* (URLs) associated with it. If *source* is empty that means replication is requested and location of sources is established from the IS.
- **Collecting** - the SSE is waiting for the client to fill content if the file. As soon as last chunk is uploaded state is changed to **complete**.
- **Downloading** - the SSE is taking necessary actions to fill content of the file from associated *sources*.
- **Complete** - content of the file is fully available. The SSE is trying to validate file by comparing it's *checksum* with one available from other sources.
- **Valid** - the file has proper content and can be downloaded from the SSE. This is a normal state of files stored at the SSE.
- **Deleting** - the SSE has received request to remove the file. It is waiting for other operations on that file to be completed.

Each SSE can be associated with some IS (multiple ISes in a future). Information about each stored file is passed to that IS along with some meta-data. That meta-data includes Logical File Name (LFN), size of file, creation time and some sort of checksum (MD5 by default).

- **Local** - file is know to SSE only.
- **Registering** - information about the file is being passed to the IS.
- **Announced** - information has been passed to IS.
- **Unregistering** - information is being removed from IS. This normally happens while file is in a **deleting** state.

4 Configuration

The SSE is part of HTTPSD server and is configured in corresponding section of server's configuration file. It supports following service-specific configuration commands:

- *acl_create="group_name[group_name[...]]"* - list of groups of users allowed to create new files.
- *acl_read="group_name[group_name[...]]"* - list of groups of users which are always allowed to retrieve content and information of all stored files.
- *acl_replicate="group_name[group_name[...]]"* - list of groups of users allowed to initiate replication.
- *storage=directory_path* - place to store data. *directory_path* specifies directory used to store content of data and associated meta-data.
- *url =URL* - defines URL used to create URL for file access. File with logical name LFN will be reported as accessible under URL/LFN.
- *registration="[option[,option[,...]]*" - defines how SSE behaves during collection and registration of new data unit. Options are
  - *immediate* or *delayed* - determines if SSE should try to register incoming data before it allows to store it (*immediate*) or accept new data without that and try registration later periodically (*delayed*).
  - *retry* or *noretry* - defines if failed *immediate* registration should cause error passed to client (*noretry*) or it should fall-back to *delayed* behavior (*retry*).
  - *showincomplete* or *hideincomplete* - defines if data units which have not completed collection and registration will be reported to client (*showincomplete*) or will be hidden from it (*hideincomplete*).
- *ns=configuration_string* - defines an Indexing Service (IS) to use for data registration. *configuration_string* determines type of service. For RC and RLS it contains URL of indexing service. *none* can be used to configure SSE to not use any Indexing Service.

5 Registration of files

The SSE can register stored files at Globus Replica Catalog (RC) and Replica Location System (RLS)[4] indexing services. The type of IS is determined from URL which follows configuration command *ns*. These URLs are supported:

- *rc://hostname[:port]/logical_collection_distinguished_name* - for RC.
- *rls://hostname[:port]* - for experimental registration to RLI part of RLS. This way is not fully compatible with RLS infrastructure and should be used only for testing and future development.

- *lrc://hostname[:port]* - for registering to LRC part of RLS. Because RLS does not support storing information about Storage Elements, the SSE registers itself under special name "__storage_service__". Client utilities can use it to find place to store data files.

6 SRM interface

The SSE is also accessible through Storage Resource Manager (SRM [5]) interface. Following methods of SRM v1.1 are implemented:

- *get*
- *put*
- *copy* - only from remote to local destination.
- *ping*
- *pin*
- *unPin*
- *mkPermannent* - all files at SSE are treated as "permanent".
- *getEstGetTime* - there is no secondary storage support in SSE, hence all files are available immediately.
- *getEstPutTime* - no file staging between storages is done by SSE, hence files can be stored immediately.
- *setFileStatus* - files stored at SRM do not have SRM states, hence request to change status does not really affect files.
- *getRequestStatus* - status in SSE can is only in "pending" state as long as it does exist. Request is removed as soon as it completed.
- *getProtocols* - SSE supports HTTPS and HTTPg for data transfers.
- *advisoryDelete* - this deletes file immediately if it not pinned.
- *getFileMetadata*

Some methods of SRM v2.1.1 are implemented too:

- *srmLs*
- *srmRm*
- *srmSetPermission*
- *srmRmdir*

All interfaces - SSE, SRMv1.1, SRMv2.1.1 are accessible through same endpoint.

7 Clients

Client part of the SSE is integrated into utilities provided as part of ARC middleware. For more information about usage and supported options please read “The NorduGrid Grid Manager and GridFTP Server” [6] or The “NorduGrid/ARC User Guide” [7]. Currently the SSE is supported by *ngls*, *ngcopy*, *ngrequest* and *ngacl* utilities. To access SSE directly following URL must be used
se://hostname[:port]/service_path[?filename]
This URL corresponds to service accessible at *httpg://hostname[:port]/service_path*. And *filename* corresponds to LFN to be used while registering file to indexing service.

To store new file at SSE one could use command
ngcopy file:///somefile_path_to_file se://host:port/service?lfn .
This will store file at SSE and make it register in indexing service.

To access SSE through SRM interface one may use same tools with URL
srm://hostname[:port]/[service_path?SFN=]filename
with default *service_path* equal to *srm/managerv1*.

But it is much better to use Indexing Service to choose SSE. Hence advisable way to store file is
ngcopy file:///some_path_to_file rls://index_host:port/lfn .
Here RLS URL rls://index_host:port is one of Indexing Services used by SSE to register files.

8 Setup

After installation following files must be present in installation root:

- *sbin/httpsd*
- *lib/se.so*
- *lib/srm.so*
- *sbin/httpsd.sh* (optional)

Edit */etc/arc.conf* (preferred way) or *etc/httpsd.conf* configuration file as described in [8], [1] and Section 4. Use either plugin *se.so* for SSE only interface or *srm.so* for all 3 interfaces.

Make sure You have valid host certificate and key at /etc/grid-security/host{cert|key}.pem .

Start HTTPSD by using SysV startup script (/etc/init.d/httpsd start) or by simplified startupt script *sbin/httpsd.sh* or directly ($NORDUGRID_LOCATION/sbin/httpsd).

Look at */var/log/httpsd.log* (default log file) for possible errors.

Appendix A. Configuration Example

General configuration part, applies to every service. For more information see [8].

```
[common]
hostname="home.takas.lt"
globus_tcp_port_range="9000,9300"
```

```
x509_user_key="/etc/grid-security/hostkey.pem"
x509_user_cert="/etc/grid-security/hostcert.pem"
x509_cert_dir="/etc/grid-security/certificates"
```

Definition of authorization group named "*admin*" which includes only one person. For more information see [8].

[group]

name=admin

subject="/O=Grid/O=NorduGrid/OU=uio.no/CN=Aleksandr\ Konstantinov"

Configuration of httpsd server. For more information see [1]. This block defines server which listens on TCP/IP ports 8000 (HTTPg) and 8001 (HTTPS).

```
[httpsd]
logfile= /var/log/httpsd.log
pidfile=/var/log/httpsd.pid
debug=4
gsiport=8000
sslport=8001
```

This shared library contains plugin named "*se*". It provides SSE service. For SRM interfaces use *srm.so* instead.

```
plugin=/opt/nordugrid/lib/se.so
```

Configuration block which activates and defines behavior of "se" service. For *srm.so* plugin use *srm* as name of service

```
httpsd/se
```

Path where service can be contacted.

```
path=/se
```

User from group "admin" is allowed to perform any action.

```
acl_create="admin"
acl_read="admin"
acl_replicate="admin"
```

Files and associated information are stored under */tmp/se*.

```
storage=/tmp/se
```

Stored content is registered in LRC (RLS) server listening on host *home.takas.lt*.

```
ns=lrc://home.takas.lt
```

Files are not being registered immediately (*delayed*) and registration is retried in case of failure (*retry*). Information about files is available as soon as it is allocated at SSE service.

```
registration=delayed,retry,showincomplete
```

References

[1] A. Konstantinov. The HTTPS(S,G) and SOAP Server/Framework. [Online]. Available: http://www.nordugrid.org/documents/HTTP_SOAP.pdf

[2] The NorduGrid Collaboration. [Online]. Available: http://www.nordugrid.org

[3] The Gridsite. [Online]. Available: http://www.gridsite.org

[4] Replica Location Service. [Online]. Available: http://grid-data-management.web.cern.ch/grid-data-management/replica-location-service/index.html

[5] Storage Resource Management Working Group. [Online]. Available: http://sdm.lbl.gov/srm-wg/

[6] A. Konstantinov. The NorduGrid Grid Manager. [Online]. Available: http://www.nordugrid.org/documents

[7] The NorduGrid/ARC User Guide. [Online]. Available: http://www.nordugrid.org/documents/userguide.pdf

[8] A. Konstantinov. Configuration and Authorisation of ARC (NorduGrid) Services. [Online]. Available: http://www.nordugrid.org/documents/Config_Auth.pdf