



*Nordic Testbed for Wide Area Com-
puting And Data Handling*

18/7/2004

THE NORDUGRID “SMART” STORAGE ELEMENT

*Description and Administrator’s Manual**

A.Konstantinov

*Comments to: aleks@fys.uio.no

Contents

1	Introduction	4
2	Configuration	4
3	How it works	4
4	Registration of files	6
5	Clients	6
6	Setup	6

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 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.

3 How it works

Each data unit (a file) passes few stages inside SSE. Those can be divided into two threads - related to a state of content (Fig.1a) and registration (Fig.1b).

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.

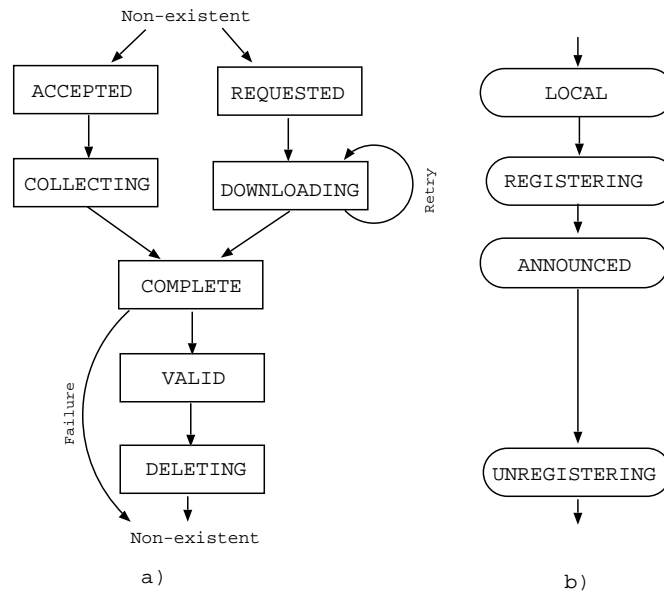


Figure 1: Execution flow.

- **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 of 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 its *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 known 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 Registration of files

The SSE can register stored files at Globus Replica Catalog (RC) and Replica Location System (RLS)[3] 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.

5 Clients

Client part of the SSE is integrated into utilities provided as part of NorduGrid Grid Manager. For more information about usage and supported options please read “The NorduGrid Grid Manager and GridFTP Server”[4]. 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 *http://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.

But it is much better to use index 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 ISes used by our SSE to register files.

6 Setup

Both HTTPSD and SSE are in development state. As result it's setup can be a bit cumbersome. After installation following files must be present in installation root:

- *sbin/httpsd*
- *sbin/httpsd.sh*
- *etc/httpsd.conf* or *share/doc/grid-manager/httpsd.conf*

Edit *etc/httpsd.conf* as described in [1] and Section 2. Or use central configuration file */etc/nordugrid.conf* as described in [5].

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.

References

- [1] The HTTP(S,G) and SOAP Server/Framework. <http://www.nordugrid.org/documents/>
- [2] NorduGrid project. <http://www.nordugrid.org>
- [3] Replica Location Service. <http://www.globus.org/rls/>
- [4] The NorduGrid Grid Manager And GridFTP Server: Description and Administrator's Manual. <http://www.nordugrid.org/documents/>
- [5] Configuration and Authorisation of ARC (NorduGrid) Services. <http://www.nordugrid.org/documents/>