Replica Management Component Services

Kurt Stockinger

Scientific Data Management Group

Lawrence Berkeley National Laboratory
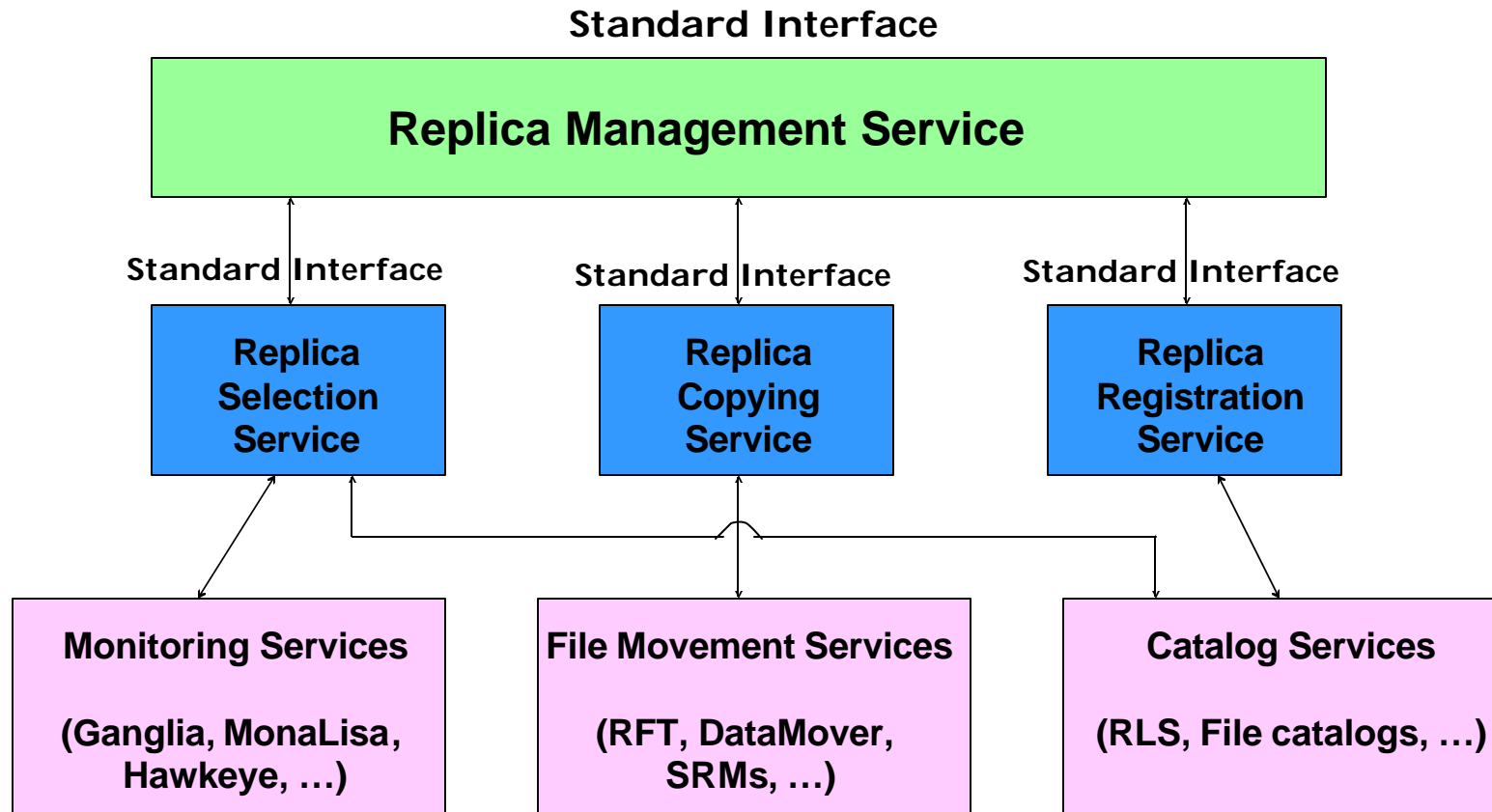
Based on joint work with Arie Shoshani and Alex Sim

Outline

- ◆ Replica Management Components

  - ▪ Replica Management Service (RMS)

  - ▪ Replica Registration Service (RRS)

  - ▪ Replica Copy Service (RCS)

  - ▪ Replica Selection Service (RSS)

- ◆ Main Functionality of RMS

Replica Management Component Services

Standard Interface

Replica Management Service

Standard Interface   **Standard Interface**   **Standard Interface**

Replica Selection Service	Replica Copying Service	Replica Registration Service

Monitoring Services (Ganglia, MonaLisa, Hawkeye, …)	File Movement Services (RFT, DataMover, SRMs, …)	Catalog Services (RLS, File catalogs, …)

Replica Management Service (RMS)

- ◆ Purpose:
  - Copy files and directories
  - Register files in a replica catalog
  - Select best replicas based on access costs (network + storage)

- ◆ Issues:
  - Specifying the source files to be copied and registered
  - Specifying the target directory or location for the files
  - Specifying the catalog to write into
  - Coordinating copying and registration
  - Dealing with failures

- ◆ Current Implementations:
  - Globus + EDG RM: client side tool, do not support replication of directories

Notation

- ◆ { }... tuple, e.g. {LFN, SURL}

- ◆ () ... list (ordered collection)

- ◆ [] ... set (unordered collection)

- ◆ _ ... mandatory (as opposed to optional)

- ◆ | ... choice operator

- ◆ // ... comment

Main Functions of RMS

- register, getRegisterStatus

- copy, getCopyStatus

- copyAndRegister, getCopyAndRegisterStatus

- delete, getDeleteStatus

- unregister, getUnregisterStatus

- unregisterAndDelete, getUnregisterAndDeleteStatus

register

IN:

String userID

{ <u>String LFN | String SURL-source-file</u>,

   <u>String SURL-target-file</u> }[]

|

{ <u>String SURL-source-dir</u>,

   <u>String SURL-target-dir</u> }

String catalogEndpoint ()

Boolean firstTimeRegistration // default: false

Enum registrationErrorDirectives ("stop",

                                "stopAndUndo",

                                "continue") // default: stop

OUT:

<u>String requestID</u>

Int estimatedExecutionTime // in seconds

getRegisterStatus

IN:

String requestID

String catalogEndpoint ()

OUT:

{ String LFN,

String S-SURL,

String T-SURL,

String status }[]   // for each catalog-endpoint,

// the status of all SURLs is returned

// "in progress" | "done" | "suspended"

copyAndRegister

IN:

String userID

{ <u>String source</u>

  <u>String SURL-target-file</u> }[]

|

{ <u>String SURL-source-dir</u>,

  String <u>SURL-target-dir</u> }

String sourceCatalogEndpoint ()

String targetCatalogEndpoint ()

Boolean firstTimeRegistration   // default: false

Boolean retryFromOtherSource // default: true

Int registrationMode // default: 0; register one file at a time vs.

          //   register files after all copies arrived
Enum errorDirectives // see next page on "Mode and Actions"

OUT:

<u>String requestID</u>

Int estimatedExecutionTime // in seconds

copyAndRegister – Modes and Actions

◆ Stop all, delete and unregister

◆ Stop all und unregister

◆ Stop all

◆ Continue copy, stop registration and record failure

◆ Continue copy and registration and record failures

unregisterAndDelete

IN:

String userID

String requestID

{ String LFN,

   String SURL-file}[]

|

{ String SURL-dir }

String catalogEndpoint ()

OUT:

String requestID

Int estimatedExecutionTime // in seconds

Replica Registration Service

- **Register** files into various catalogs:
  - File catalog
  - Replica catalog

- Register set of files (note: replication of directories handled by Replica Management Service)

- Guarantee **fault tolerance**:
  - Keep track of registration status:
    - In progress
    - Done
    - Suspended
  - Retry registration if catalog down

- Main commands:
  - register
  - unregister

Replica Copying Service

◆ Copy files from storage A to storage B

◆ On top of:

  ▪ gridFTP, RFT, SRM, etc

◆ Guarantee fault tolerance

◆ Main command:

  ▪ copy

Replica Selection Service

- Select the best replica based on access costs (network + storage system)

- Need monitoring systems for networks + storage

findReplicas

IN:

{ <u>String LFN</u> |

<u>String SURL-file</u> }

<u>String target-hostname</u>

Boolean includeCost // default: true

OUT:

<u>String requestID</u>

String serviceEndpoint

String estimatedExecutionTime

getFindReplicasStatus

IN:

    <u>String requestID</u>

OUT:

    { <u>String LFN</u>,

      { <u>String SURL-file</u>,

        Int accessTime}()

    } []

Open Issue for SRM

◆ Provide method for access estimation

◆ Prototype implementation of access cost estimator:

Kurt Stockinger, Heinz Stockinger, Lukasz Dutka, Renata Slota, Darin Nikolow, Jacek Kitowski, *Access Cost Estimation for Unified Grid Storage Systems*, *International Workshop on Grid Computing (Grid2003)*, Phoenix, Arizona, November 2003, IEEE Computer Societe Press.

Conclusions

- Layered architectures of Replica Management Component Services:

  - Replica Management Service

  - Replica Selection Service

  - Replica Registration Service

  - Replica Copying Service

- Replication of files and directories

- Focus on fault tolerance