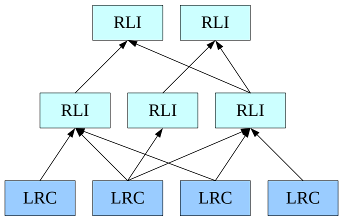
The Globus Replica Location Service



The Replica Location Service (RLS) is a tool that provides the ability to keep track of one or more copies, or *replicas,* of files in a Grid environment. This tool, which is included in the Globus Toolkit, is especially helpful for users or applications that need to find where existing files are located in the Grid.

RLS Overview

The Replica Location Service is one component of data management services for Grid environments. It is a simple registry that keeps track of where replicas exist on physical storage systems. Users or services register files in the RLS when the files are created. Later, users query RLS servers to find these replicas.

The RLS is a *distributed* registry, meaning that it may consist of multiple servers at different sites. By distributing the RLS registry, we are able to increase the overall scale of the system and store more mappings than would be possible in a single, centralized catalog. We also avoid creating a single point of failure in the Grid data management system. If desired, the RLS can also be deployed as a single, centralized server.

Using the RLS: An Example

One example of a system that uses the RLS as part of its data management infrastructure is the Laser Interferometer Gravitational Wave Observatory (LIGO) project. LIGO scientists have instruments at two sites that are designed to detect the existence of gravitational waves. During a run of scientific experiments, each LIGO instrument site produces millions of data files. Scientists at eight other sites want to copy these large data sets to their local storage systems so that they can run scientific analysis on the data. Therefore, each LIGO data file may be replicated at up to ten physical locations in the Grid. LIGO deploys RLS servers at each site to register local mappings and to collect information about mappings at other LIGO sites. To find a copy of a data file, a scientist requests the file from LIGO’s data management system, called the Lightweight Data Replicator (LDR). LDR queries the Replica Location Service to find out whether there is a local copy of the file; if not, the RLS tells the data management system where the file exists in the Grid. Then the LDR system generates a request to copy the file to the local storage system and registers the new copy in the local RLS server.

LIGO currently uses the Replica Location Service in its production data management environment. The system registers mappings between more than 11 million logical file names and 120 million physical file locations.

Components of the Replica Location Service

The RLS design consists of two types of servers: the Local Replica Catalog and the Replica Location Index.

The *Local Replica Catalog (LRC)* stores mappings between logical names for data items and the physical locations of replicas of those items. Clients query the LRC to discover replicas associated with a logical name. The simplest RLS deployment consists of a single LRC that acts as a registry of mappings for one or more storage systems. Typically, when an RLS is deployed on a site, an administrator populates it to reflect the contents of a local file or storage system. If new data files are produced by a workflow manager or a data publishing service, these services typically register newly created files with the RLS as part of their publication process. Our performance studies for an LRC deployed on a moderately powerful workstation with a MySQL relational database back end show that the catalog can sustain rates of approximately 600 updates and 2,000 queries per second.

For a distributed RLS deployment, we also provide a higher-level *Replica Location Index (RLI)* server. Each RLI server collects information about the logical name mappings stored in one or more LRCs. An RLI also answers queries about those mappings. When a client wants to discover replicas that may exist at multiple locations, the client will pose that query to an RLI server rather than to an individual Local Replica Catalog. In response to a query, the RLI will return a list of all the LRCs it is aware of that contain mappings for the logical name contained in the query. The client then queries these LRCs to find the physical locations of replicas.

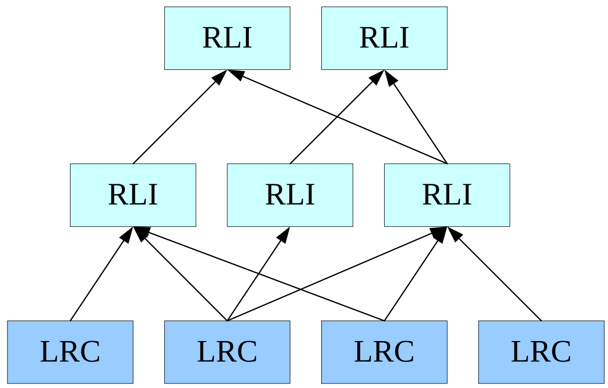


Figure 1. Example deployment of a Replica Location Service

Information is sent from the LRCs to the RLIs using *soft-state update protocols*. Each LRC periodically sends information about its logical name mappings to a set of RLIs. The RLIs collect this mapping information and respond to queries regarding the mappings. Information in RLIs times out and gets periodically refreshed by subsequent updates. An advantage of using such soft-state update protocols is that if an RLI fails and later resumes operation, its contents can be reconstructed using these updates.

Because each LRC may hold millions of logical file name mappings, updates from LRCs to RLIs can become large. Sending them around the network may be slow, especially in the wide area; when updates arrive at an RLI, they may consume considerable storage space there. One option for making these updates more efficient is to compress their contents. Various compression strategies are available. The one that we chose to implement for the RLS is based on Bloom filter compression. Each Local Replica Catalog periodically creates a bit map that summarizes its contents by applying a series of hash functions to each logical name registered in the LRC and setting the corresponding bits in the bit map.

For more information:

Replica Location Service Documentation: http://www.globus.org/toolkit/docs/4.0/data/rls/

"Performance and Scalability of a Replica Location Service," Ann L. Chervenak, Naveen Palavalli, Shishir Bharathi, Carl Kesselman, and Robert Schwartzkopf, High Performance Distributed Computing (HPDC-13) Conference, June 2004.

Contacts: Ann Chervenak (annc@isi.edu) or Robert Schuler (schuler@isi.edu)