



LCG Disk Pool Manager

July 9, 2004

James Casey, Jean-Philippe Baud, Ian Bird

Overview

SRM is emerging as the standard for grid accessible storage. Developers for MSS systems such as CASTOR and ENSTORE have already implemented SRM interfaces to their storage solutions. Also, there have been some disk cache solutions provided by the dCache, CASTOR, and LBNL DRM projects. Furthermore HPSS is accessible via the LBNL HRM.

It is expected that the Tier 1 sites (and other sites with tape MSS) will be responsible for ensuring the integration SRM interfaces into those systems. The concept of a Storage Element in LCG (and other grids) requires a managed disk pool, capable of providing an SRM interface and local POSIX I/O, providing managed space (space reservation, garbage collection, etc.).

Significant work has been expended by LCG and the dCache development teams in the last 6 months into packaging and testing dCache as a candidate as a general disk management component of a Storage Element, with the hope of being able to deploy it to any LCG sites as an SE.

It has become clear that although dCache certainly presents the functionality and interfaces required, it is nevertheless quite heavyweight in terms of installation and management, and might be difficult in situations (like smaller Tier 2 centres) where limited system management effort is available. dCache represents an excellent solution for large sites with many Terabytes of disk storage.

In the gLite architecture of EGEE, two types of storage element are foreseen – strategic and tactical storage. In those terms what we propose here, fits closely with the disk management scale and needs of the gLite tactical SE.

A small Tier-2 tends to be characterized by the following features:

- 1-10TB of storage, usually system-attached to nodes
- No SAN architecture
- No full-time support for storage solutions. Only a fraction of an FTE available to manage the system

At the moment, the EDG 'classic SE' is used as a solution in LCG. This has the main drawback to sites that they must run such a 'classic SE' (in essence a gridftp server with some configuration parameters stored in the Information System) on each storage node, and manage the disk space separately on each node. Also, it does not provide an SRM interface, requiring replica management tools to deal with it on a special basis.

Systems like dCache SRM are quite complicated to setup and manage, and this effort is prohibitive for deployment at small Tier-2 sites.

Thus we see a need to provide a solution for smaller sites in this situation (and indeed larger sites that may require secondary small SEs for specific purposes), and outline our proposed solution in this note. This proposal is not intended to replace, but rather to complement the other solutions such as custom integration of existing large systems, and dCache for very large disk pools.



Proposed Solution

We propose a lightweight disk pool manager to be developed with the following characteristics:

- Support SRM v1.1 and/or v2.1 interface
- Manage 1-10Tb of storage, which can be spread across a set of disk server nodes with system-attached storage
- Provides a single point of entry for management and monitoring of all disk exposed to the grid
- Be simple to deploy and manage, requiring a fraction of an FTE for these operations
- The system would 'own' all files on disk
- All access would be GSI authenticated, and audit trails of access would be provided

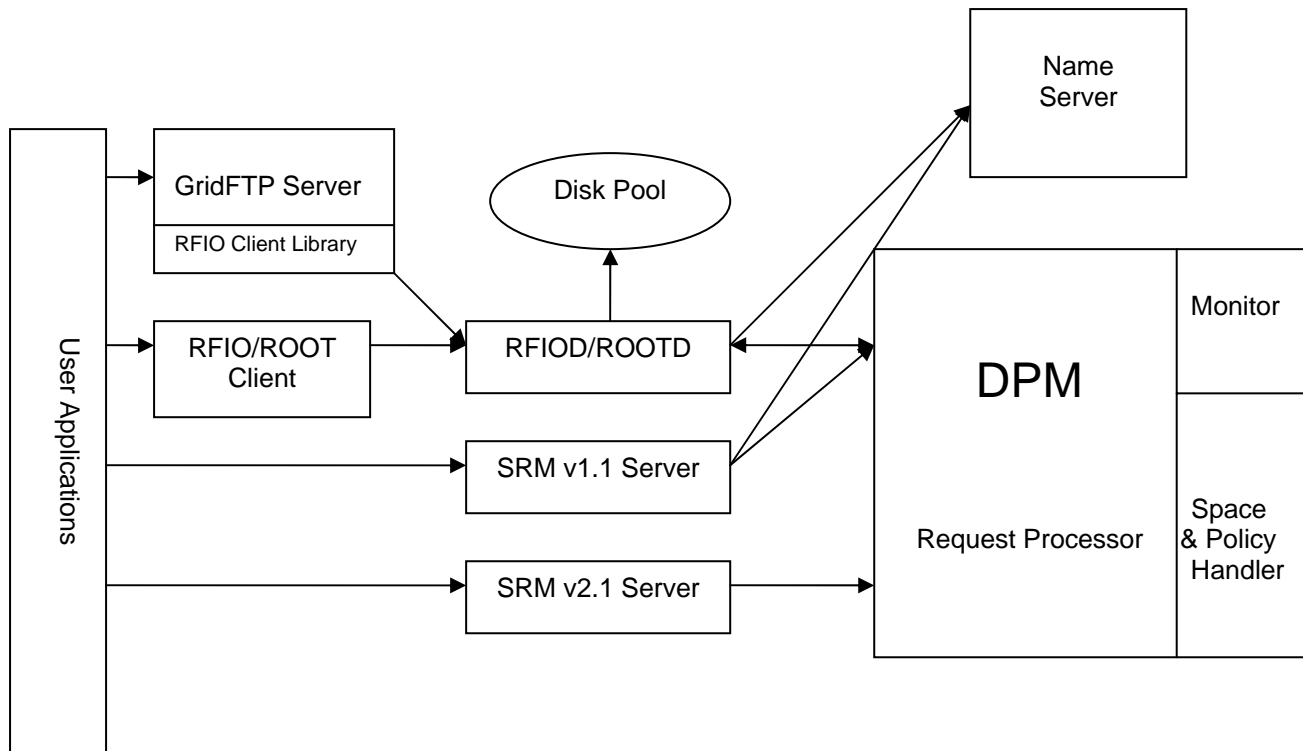
The aim of the solution is to provide such a solution in a short time frame (first version by Autumn 2004), so we would reuse existing components where available, and development work would mainly consist of gluing these components together.

Components of a Disk Pool System

A grid-exposed disk pool manager consists of the following components:

- SRM server. Exposing either SRM v1.1 or v2.1 APIs
- Name Server. This provides a global logical namespace over the set of disks, and the location within the disk pool of the actual physical files. This would have POSIX semantics for both file system operations as well as access control.
- Disk Pool Manager. This maintains a queue of ongoing and historical requests, decides onto which disks to place new files, and load balances the access to existing files
- Transfer Servers. These expose the actual files on disk via standard transfer protocols, such as gridftp, rfiio and rootd.

All interaction with the system goes through the name server and the disk pool manager and all the logic is centralized here. This makes all the other 'services' very light.



Advantages of approach

We believe that such a solution would provide several features that experiments require and have requested, but are not available in current systems.

- SRM v2.1 provides many needed features that we would expose to users. These include
 - Global Space reservation
 - Access Control Lists
- Using this unified architecture, we believe we can provide better security, with all server interactions being authenticated and logged for audit purposes.
- Finally, we believe that our approach not only works with classical 'disk servers' which provide high quality and high volume disk space, but also with disk space available on worker nodes. We could use this as volatile space, but that would be accessed securely and managed in the same way as space on the disk server.