Managing Scientific Computations in Grid Systems

Salman Toor

Division of Scientific Computing
Department of Information Technology
Uppsala University

November 11, 2008

Example-1:

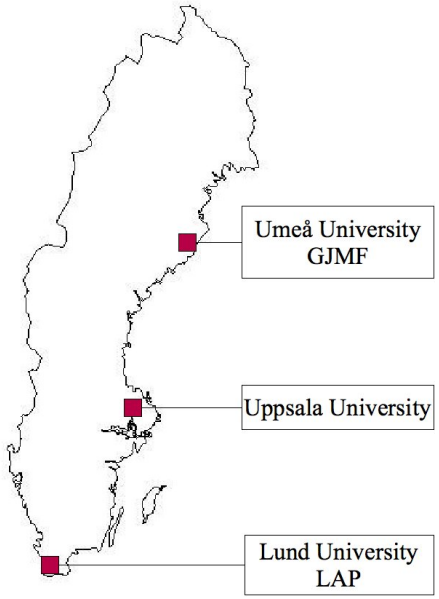
Integration of LUNARC and GJMF

Introduction

A joint project between Lund, Umeå, and Uppsala Universities.

The aim is to build a:

- Robust
- Reliable
- User friendly
- Middleware independent



grid application portal by combining already available building blocks.

Motivation

A grid application portal should be user-friendly and based on a reliable job management system.

- LUNARC Application Portal (LAP) has a user-friendly web interface, that is tightly connected with the underlying middleware (ARC).
- Grid Job Management Framework (GJMF) has a very strong job management framework, which can be used as a robust interface.

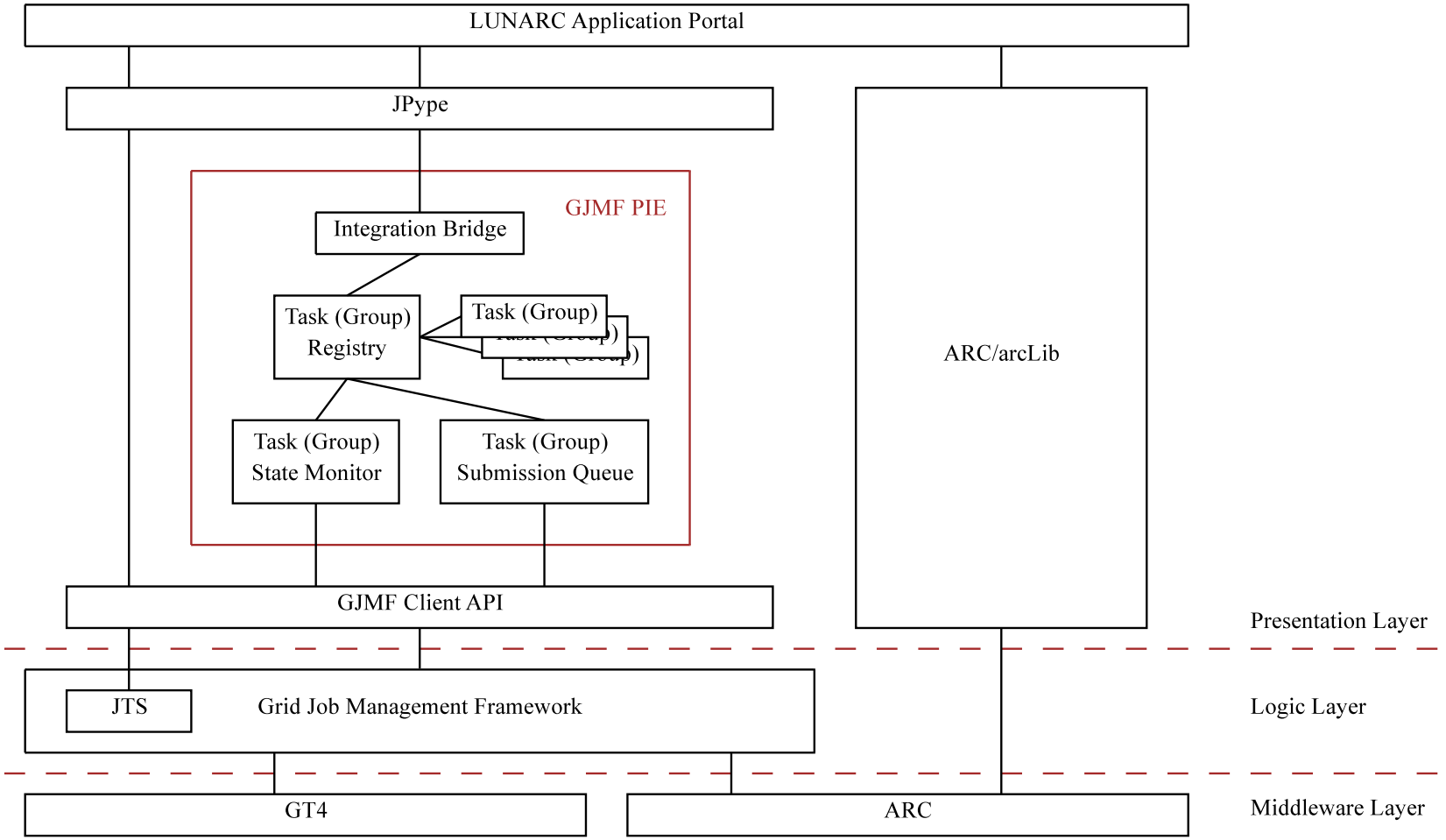
This integration project is the natural extension of these projects to jointly build a more reliable and easy-to-use framework for grid applications.

Integration architecture

The integration architecture is based of the following building blocks:

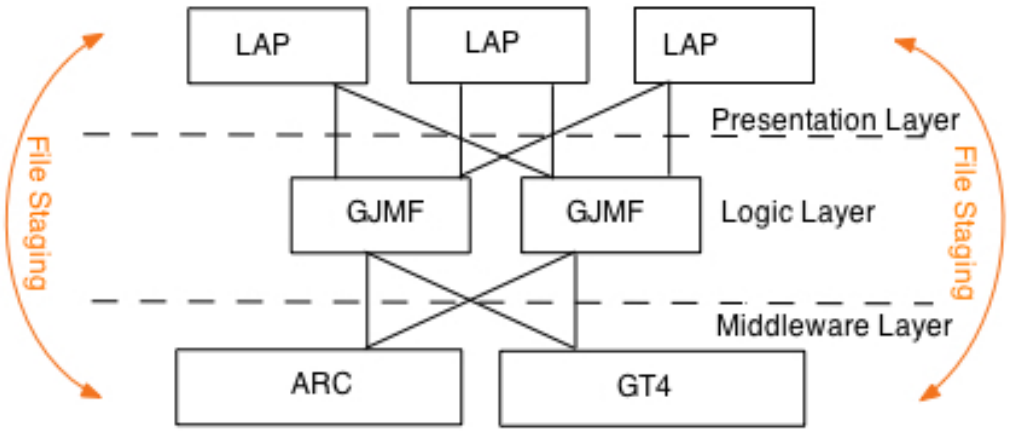
- A 3-tier, loosely coupled model.
- The proposed architecture relies on:
  - Application management by LAP.
  - Grid job management by GJMF.
- The notification-based state monitoring with the fall-back option of polling.

Detailed picture of integration architecture



Advantages of integration architecture

- File staging is completely independent from the logic layer.



GJMF introduces:

- A fault tolerant and robust submission for single and multiple jobs.
- Middleware independent job management.

Based on this architecture, our paper was accepted in the 9th International Workshop on State-of-the-Art in Scientific and Parallel Computing 2008.

Title:
Empowering a Flexible Application Portal with a SOA-based Grid Job Management Framework.

Authors:
E. Elmroth, S. Holmgren, J. Lindemann,
S. Toor, and P-O Östberg.

Example-2:

KnowARC storage system

What is KnowARC?

The KnowARC (Grid-enabled Know-how Sharing Technology Based on ARC Services and Open Standards) is a project within the NorduGrid consortium.

Goal: Improve and extend the existing state-of-the-art technology found in the Advanced Resource Connector (ARC) middleware.

Major components:

- HED (Hosting Environment Daemon) with novel security framework
- A-REX, a standard compliant (BES/JSDL) job execution service
- **Storage system**
- Client tools
- ....

KnowARC storage system

The goal is to implement services which can form a reliable, robust, scalable, high-performance, and consistent data storage system by using the components provided in the KnowARC framework.
Consists of following services:

- Bartender
- Librarian
- Shepherd
- A-Hash
- Gateway

Also provides client tools using FUSE model.

Bartender: Provides a high-level interface for the users by communicating with the Librarian and the Shepherd services.

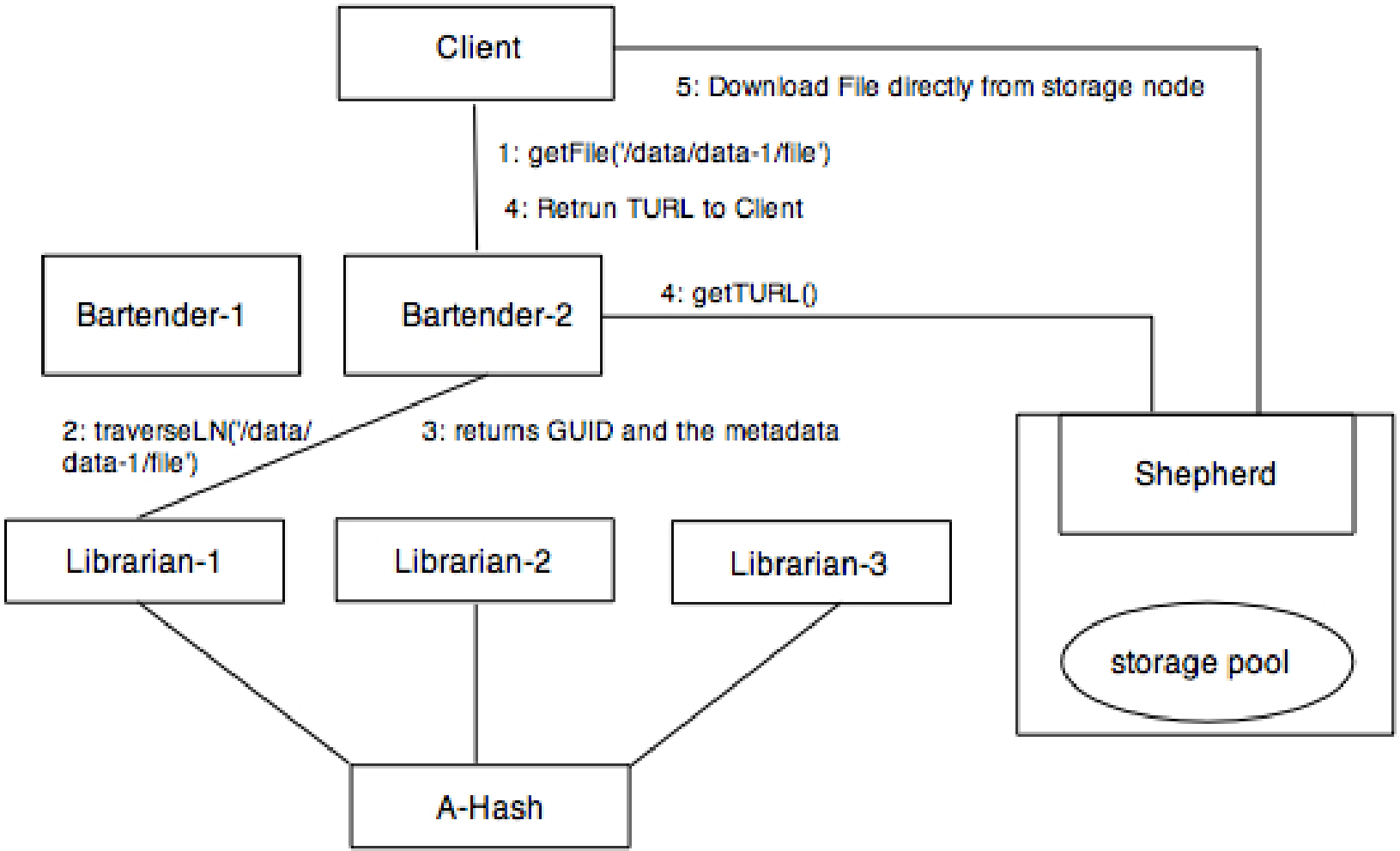
Librarian: Handles the global namespace and registering the Shepherd services.

Shepherd: Use for storing the replicas and the files on different storage solutions.

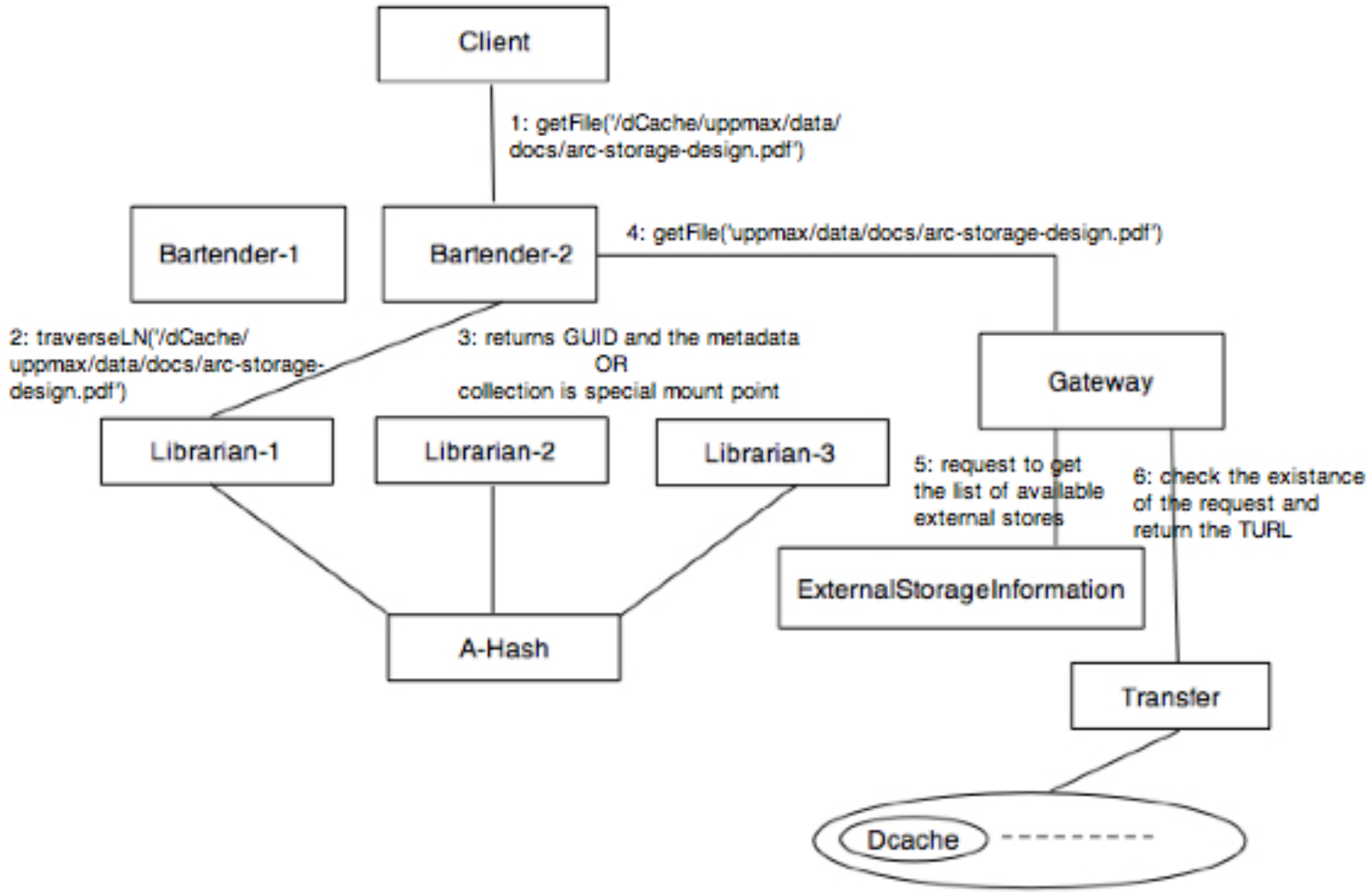
Gateway: Use for communicating with external storage system using different protocols like gridftp and srm. Currently available for the Dcache.

A-Hash: A database service which is used by the Librarian services to store all the data. Currently it is centralized but will be distributed in the final release.

Download file from internal store



Download file from external store



Current status

Recently, we completed the first prototype version of storage system with the basic level of security and centralized A-Hash service. By the start of next year, production version of KnowARC storage system will be available with full security implementation and with the distributed A-Hash service.

Project partners:

Bjarte Mohn (Uppsala University)
Jon Kerr Nilsen (Oslo University)
Salman Toor (Uppsala University)
Zsombor Nagy (NIIF, Hungary)

Example-3:

Brokering in storage systems using Linear Algebra

Problem description

How to design an improved model for load balancing or brokering component for several geographically distributed storage nodes?

The functionality of the resource brokering or load-balancing depends on:

1. The amount of information available about the resources.
2. How efficient is the response of the broker?
3. How reliable the decision is?

Proposed idea

In the heterogeneous environment of the grid, it is more realistic to have an estimation model of finding the resources that may or may not be the best of its kind, but should fulfill the requirement.

Proposed idea

In the heterogeneous environment of the grid, it is more realistic to have an estimation model of finding the resources that may or may not be the best of its kind, but should fulfill the requirement.

For this purpose we want to use the technique of **Mathematical Information Retrieval (Math IR)**.

Math IR example

	Doc-1	Doc-2	Doc-3	Doc-4	Doc-5	Query
run(ning)	1	0	1	0	0	0
help	1	1	0	0	1	1
training	0	0	1	1	1	0
cooking	1	1	0	1	0	1

Dimension of the matrix is:

Terms x Documents

Each document is represented as a four dimensional vector, where each dimension is the required term in the document.
Query: "cooking help" or "help on cooking"

Math IR example: Possible solutions

- Check the similarity of cosine angles(naive approach):

  $$\cos(\theta_j) = \frac{a_j^T q}{||a_j||_2 ||q_j||_2} = \frac{\sum_{i=1}^{t} a_{ij} q_i}{\sqrt{\sum_{i=1}^{t} a_{ij}{}^2} \sqrt{\sum_{i=1}^{t} q_i{}^2}}$$

  Where

  a_{ij} : Elements of the Term x Document matrix.

  q_i : Elements of the Query vector.

- Use the **Reduced Rank Approximation** (advanced approach) by using QR factorization or some other method.

Source:

Matrix Rank Reduction for Data Analysis and Feature Extraction, Haesun Park, Lars Elden

Matrices, Vector Spaces, and Information Retrieval, by Michael W. Berry, Zlatko Drmat and Elizabeth R. Jessup

Proposed model

	Store-1	Store-2	Store-3	Store-4	Store-5	Query
Avaliable Store (GB)	30	10	6	10	4	0.5
Protocol-1	1	1	0	0	1	1
Protocol-2	0	0	1	1	1	0
Protocol-3	1	1	0	1	0	1

(Term x StorageNodes) matrix

1: Store supports the following protocol
0: Dose not support following protocol

query: "upload a file of size 0.5 GB with protocol-1 and protocol-3"

Aim is to use **reduced rank approximation** or some other method to improve the resource brokering in the grid systems.

Proposed model

This approach can be useful for handling the requests of uploading files and creating multiple replicas in the system.

Proposed model

This approach can be useful for handling the requests of uploading files and creating multiple replicas in the system.

Main Challenges:

- Matrix need to be updated recurrently.
- Proposed model cannot help to handle downloading requests. Need a separate matrix.
- Which method to use?