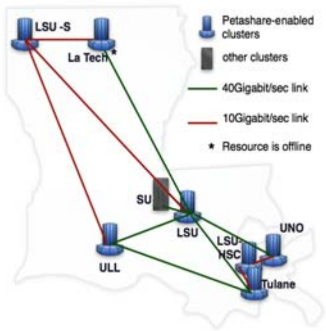
Intermediate Gateway Service to Aggregate and Cache I/O Operations into Data Repositories

Mehmet Balman, Ismail Akturk, Tevfik Kosar
Center for Computation and Technology, Louisiana State University

Today's large-scale scientific applications generate tremendous amount of data. Those data intensive applications usually make use of scratch volumes, which are limited in size, to store output data temporarily in a high performance computational cluster.

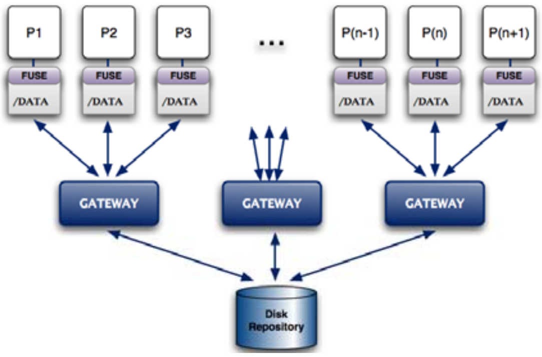
Data Grids provide a distributed environment for indexing and storing large scale scientific data for collaborative science. A general scenario is to use an intermediate storage area and then transfer files to a remote storage for post processing and long term archival. This leads to two major problems. First, there is need to handle data-flow with some external tools; second, client is limited by the storage capacity of the intermediate staging area.

LONI and PetaShare Sites



To provide transparent and efficient direct access for scientific applications (*write once, read many*):
- make client tools accessing remote data storage more intelligent
- upload data transparently while preserving the performance and efficiency

Instead of sending I/O request directly to the remote data resource, we plan to use an intermediate gateway service to aggregate and cache I/O operations. This will provide an asynchronous mechanism and also make I/O accesses efficient since we will not be dealing with high latency for many small data chunks sent for I/O calls. Those clients communicate and forward I/O request to the gateway service.
Gateway act as a staging area, but transparent to users. It does caching and aggregation of I/O requests. We sent data in large chunks and minimize number of calls sent to the remote data repository.



Intermediate Gateway Service

Presented at 7th USENIX Conference on File and Storage Technologies - Feb 24-27, 2009

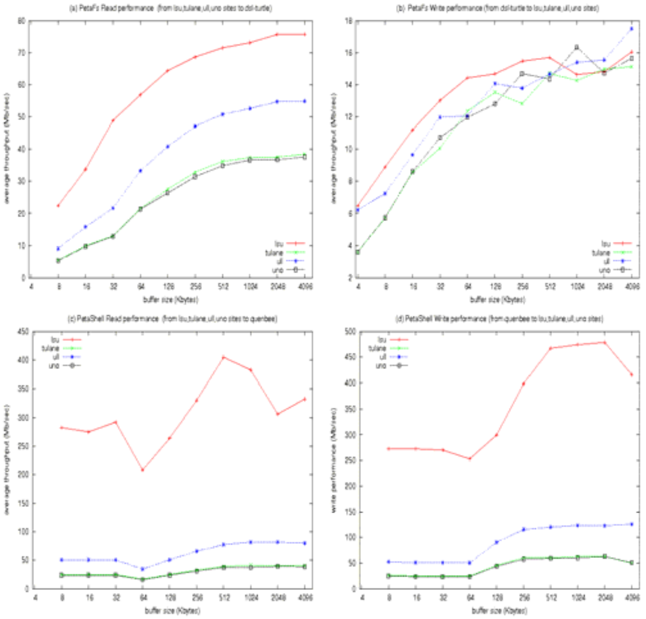
FAST'09

LSU
CENTER FOR COMPUTATION & TECHNOLOGY

PetaShare: A Distributed Data Storage system that span multiple institutions across Louisiana.
Petashare provides a unified namespace using iRODS services and lightweight client tools for efficient and transparent access.

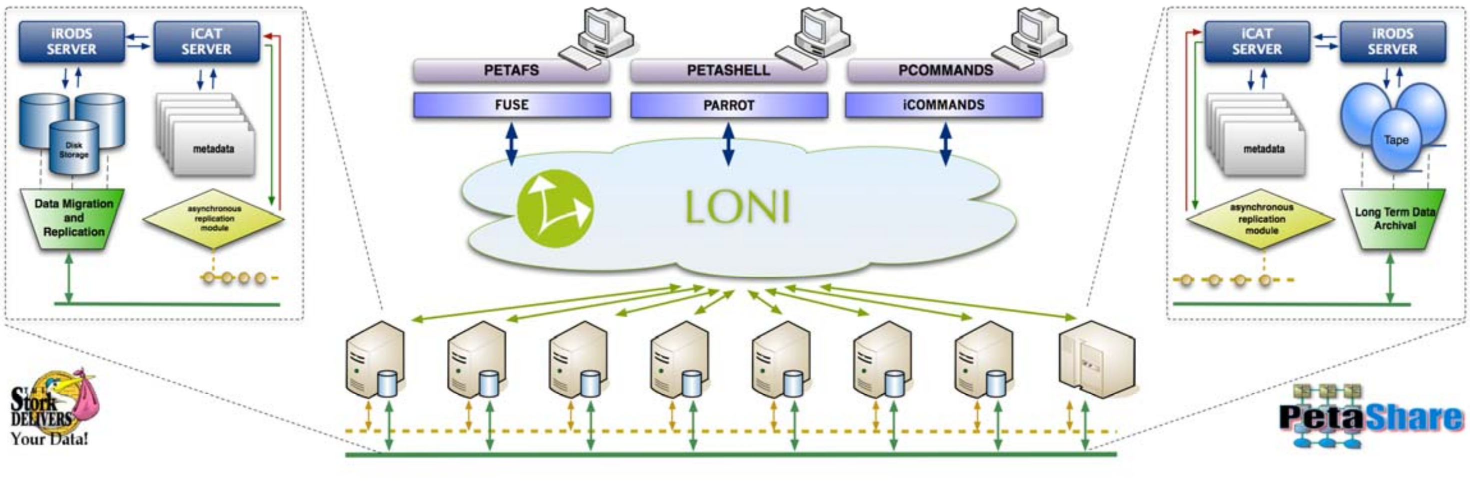
PetaShare client tools:
- ***Petashell***, based on Parrot, provides an interactive shell interface by capturing I/O calls and matching them to corresponding remote I/O operations.
- ***Petafs***, a FUSE client, allows users to access remote data repositories mounted as a local filesystem.
- ***Pcommands,*** *specialized (Posix-like) command set to access* PetaShare resources.

Advance buffer implementation in PetaShare clients:
We have optimized Petafs and Petashell clients by aggregating I/O requests to minimize the number of network messages. We have implemented prefetching for read operations, and caching for write operations such that we delay I/O operations and upload data in large size of chunks to eliminate the effect of high latency between the client and the data resource.



There are 4 major remote sites and the metadata database is on *lsu* site. *Dsl-turtle* is outside of the LONI network and it has slow access to 4 PetaShare sites. *Queenbee* is inside the LONI network and it has much faster access to all of those 4 sites. Results are average values of 3 to 5 separate runs. We have used *cp* command and collected average throughput of 3 to 5 separate runs for copying 1MB, 10MB and 100MB files. The x-axis (buffer size) is in log scale.

PetaShare Architecture



Stork DELIVERS Your Data!

PetaShare

Acknowledgement: This project is in part sponsored by National Science Foundation, Department of Energy, and Louisiana Board of Regents.