

## Introduction

### Abstract

This poster presents our new experimentation to utilize Lustre filesystem for CMS analysis with direct POSIX file access while keeping dCache as the frontend for data distribution and management. We describe our implementations to integrate dCache with Lustre filesystem and how to enable users to access data without going through dCache file read protocol. Our initial CMS analysis job measurement shows very good data access performance and the transfer performance is also very good when using Lustre as dCache pool.

### Motivations

- Lustre is a scalable, secure, robust, highly-available cluster file system that has been used successfully among many high performance computing facilities;
- dCache has been successfully used to manage large amount of data, especially HEP/LHC data globally. However, the data access in dCache is relatively slow;
- It will be very good to get the advantages of both systems;

## Implementations

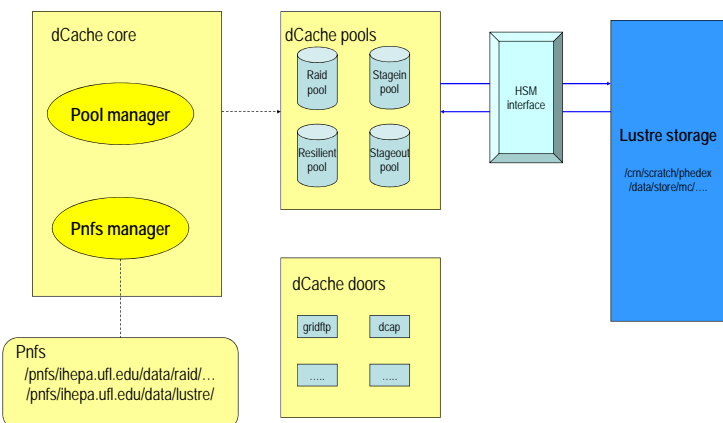
### Use Lustre directly as dCache pool

- Lustre directories are used directly as dCache pool storage;
- Soft links are created separately between pnfids and logical filename;

### Use Lustre as tertiary storage

- Take full advantage of dCache HSM interface with customized callout script;
- Files are first transferred to stageout pools and then migrated to Lustre automatically by dCache;
- Files can be staged in using stagein pools from Lustre filesystem;
- Mapping between pnfids and regular filename during migration for direct data access;

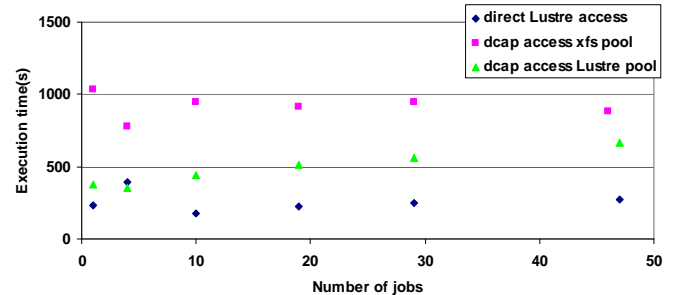
PoolManager and pnfs directory tags are modified for files to use the special Lustre pools. The diagram for using lustre filesystem as tertiary storage is shown below:



## CMS analysis job and data transfer test results

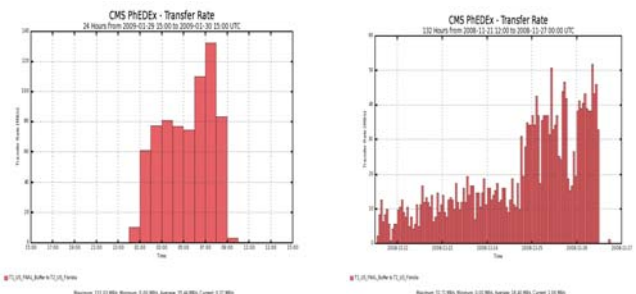
### CMSSW Job execution time

CMS user analysis job executes much more efficient with directly accessing data stored on Lustre filesystem. The following figure shows the CMSSW user analysis job execution time using directly Lustre access, accessing through dcap with xfs filesystem dCache pool, and accessing through dcap with Lustre filesystem dCache pool:



### Transfer performance

Use Lustre directly as dCache pool gives very good transfer performance. The performance using Lustre as tertiary storage is relatively low due to the two-step copying through stagein pool.



## Conclusions

- We described our implementation to utilize Lustre filesystem for CMS analysis with direct POSIX file access while keeping dCache as the frontend for data distribution and management;
- Utilizing Lustre filesystem can significantly improve the data access performance comparing with data access through dCache dcap protocol. Over 60% performance improvement has been observed when running jobs with direct data access through Lustre filesystem. Dcap data access with Lustre filesystem dCache pool is also faster with smaller number of jobs, but slows down with increasing number of jobs;
- dCache with directly Lustre mounted pools shows very good transfer performance;