D0 File Replication

PPDG SLAC File replication workshop
9/20/00

Vicky White

D0 Data

- Data Files of ~ 1GB
  - A file belongs to a specific Data Tier, specifying its data content - raw, generated, digitized, simulated, reconstructed, summary, thumbnail, etc….
  - Files are clasified many other ways by
    - Trigger stream
    - Run type
    - Production/Decay or PDG process (for MC)
    - Parentage for non-primary files
    - Creating process, e.g. GEANT, Reco V1.5, etc. etc.
  - The file format of different data tiers may be different – the translation from file format to objects is handled by the D0 I/O package d0om
    - EVPACK/DSPACK for most data
    - ROOT for some summary data
    - Other file formats being discussed for thumbnail data, including possibly NO file – all data in Oracle db

D0 File Replication

- RAW, MC, Reconstructed data and summary data files (in fact all permanent files) are stored
  - in AML2 flexible-media robot at Fermilab using the Enstore storage management system for storage and retrieval
    - ➔ Basically 'copy' file to/from disk location to storage system location
  - On disk on central analysis server (~20TB -> 50TB) and other linux cluster disks (even desktops) at FNAL
- MC Data Files and (after March 2001 – summary reconstructed data) will be stored
  - in HPSS controlled storage system at Lyon (started)
  - In storage system at SARA – Nikhef (soon)
  - In small robot or stacker tape system at Lancaster (Jan?)
  - In disk caches and possibly other small storage systems at many other D0 sites including UTA, Maryland, MSU, BU, etc.
    - ➔ As the run starts Mar 2001

SAM system

The D0 SAM system manages and tracks all of the data at all these sites and storage management systems. Currently this uses

- A central Oracle database to record
  - all data files, their classification, event content, history and parentage and also all defined datasets
  - Permanent locations of all files – disk and tape
  - Cached locations of all files
    - ➔ Some files locked in a disk cache
    - ➔ Else cached files eligible for replacement
- A distributed system of CORBA servers to carry out all of the functions
  - with pretty good retry/robustness/error recovery features

Technology and platforms

SAM runs on Irix, Linux, SUN (most servers), OSF1

Uses

- g++ or KAI
- C++, Python and some Java
- Orbacus and Fnorb ORBs
- Enstore encp commands
- bbftp ?
- HPSS copy commands
- Oracle Database interfaced using Python_dcoracle

SAM concepts

A Station is a group of resources managed together – includes a **cache manager** for the disk cache(s) provided to that station

Stations initiate "Projects" each controlled by a **Project master** to provide data files (based on **datasets** requested) to end user applications

Stagers carry out the actual staging of files using multiple processes that each essentially issue some form of 'copy' command for a single file from a source to a destination location known to the database or in a cache

e.g.

encp sim.a....raw /pnfs/sam/mammoth2/....

xcp sim.c chpstc2.in2p3.fr:/hpssbis/in2p3.fr/group/d0/mc_prod/

bbftp xx yy

rcp xx yy

File Replication for read

Files that have been produced and stored at one site are replicated and placed in disk cache at another site, as requested, by means of starting a Project for a particular dataset

Staging of files into cache occurs asynchronously to requests by any one application to 'get next file'.

Staging occurs in batches (configurable) and is held if no consumers consume the files (configurable)

Locations, projects, consumption of files are all stored in the central database at Fermilab

A global optimizer groups, reorders, regulates staging requests (currently only those between storage system and disk)

The model of operation is quite similar to PPDG request architecture coming out of Architecture working group

Replicated Stored Files

A File Storage Server (FSS) on each Station handles storage of files to permanent locations, including routing of files through disk cache on an intermediate station to its final destination, and retries on error

A file name is unique in our SAM catalogue, but the file may be stored in many locations

sam store –descrip=description-file

- store one or more files described in the description-file

sam store can determine destination for nth copy of file based on file characteristics or can use explicit storage destination

Lyon has tested this using HPSS at Lyon and to Enstore at Fermilab. Plans to use for MC production starting now!

Nikhef will also start using this for MC production in October

UTA currently send tapes of MC data and we do the sam store at FNAL

Prague ftps files to Fermilab and we do sam store currently

Replicating a Physics Dataset for Analysis

An example – how datasets will be replicated.

To replicate the 'latest' summary data created on reconstruction farms at Fermilab and stored on tape in Femilab robot -> to Lyon for analysis.

a) Every day (or N hours or ?) start a file-delivery "project" on Lyon station using a dataset 'query' that identifies all the files of type x, physics stream y, and …..

b) Run a 'consuming' file script on Lyon m/c

   a) sam get next file
   b) sam store <file descip> <to local Mass Storage location>
   c) sam release file

THIS REPLICATES the dataset defined and leaves as much of it in Lyon cache as they have room for.

Simultaneous analysis programs will also be able to 'consume' the same dataset as it is being staged and replicated

PPDG File Replication

D0 has an (almost) working system that stores files in multiple locations and replicates files as needed to permanent locations or volatile cache

Can use this testbed to try out 'standard' components or layers in common with PPDG (and other Grid projects)

Well defined interfaces are good! Better than specific implementations. We like IDL.

Have tested the interface approach with LBLs HRM working with Wisconsin - to use both SAM and SRB as source for Condor files

Worried about security infrastructure – Kerberos is being mandated at Fermilab

PPDG File replication components?

D0 distributed file caching and storage could use

- An alternate catalog to 'query' what files are in a dataset
  - Probably not useful – too many D0 specifics
- An alternate/additional replica catalog service to maintain file locations. Globus? SRB, Namespace design?
  - Add location, remove location interfaces
  - Differentiate between permanent locations and cache?
  - When we lock a file in cache it is done on behalf of a specific group so file can be locked N times - lock/unlock primitives
- A parallel gsiftp with authentication - to replace bbftp?
- A bbftp with security? – what will Babar do?
- A network weather service – to help us decide where to fetch files from and how many at a time
- A resource management/matchmaking service to decide where to fetch files from and in what batches (good if mounting tapes)
- Other?