

# **Integrated Systems Development Department Projects**

## **CDF Central Analysis Computing Workshop**

October 18-19, 2001

Igor Mandrichenko,  
Farms and Clustered Systems Group

---

# ISD Projects

---

- Data Management and Storage
  - Enstore – mass storage system
  - Disk Cache – WAN, GRID interface to Enstore
- Farms and Clusters: management and infrastructure
  - PCQCD – PC farm for lattice QCD calculations
  - NGOP – Next Generation of Operations
    - **FBSNG – Farm Batch System**
    - **Disk Farm – distributed disk storage**

# Mass Storage

- Enstore
  - Robotic tape storage system developed for Run II era
  - Staging system with UNIX file name space
  - Main and only storage system supported by CD at Fermilab
  - Hardware:
    - STK robot, 9940 tape drives
    - ADIC robot, LTO tape drives

URL: <http://www-isd.fnal.gov/enstore/>

- Disk Cache
  - Developed in collaboration with DESY
  - WAN-, FTP-accessible disk buffer for Enstore
  - Work in progress on Data Grid standards
    - Grid-FTP interface
    - HRM interface
  - Throughput optimization
  - Remote user authentication

## PCQCD

- Replacement for ACMPAPS supercomputer used for lattice QCD calculations with specialized PC farm
- Being developed by ISD, CP departments, MILC collaboration
- Research areas include:
  - Low latency intrer-node communication
  - Centralized OS upgrades, version control, OS boot configuration (Intel's Preboot Execution Environment, PXE)
  - Disk-less boot/operation
  - PC benchmarks (<http://qcdhome.fnal.gov/benchmarks/>)
  - Trace macros for low level kernel and application performance measurements and tuning (<http://linux-rep.fnal.gov/software2.html>)
  - Computer health monitoring/logging (<ftp://linux-rep.fnal.gov/pub/ipmi/>)
- Currently 80 dual 750MHz Pentium III nodes
- Plans: add 48+128 dual 1.7GHz+ Pentium IV nodes

URL: <http://qcdhome.fnal.gov/>

# NGOP – Next Generation of Operations

- Migration to farms adds new dimensions to old problems:
  - How to monitor ~1000-10000 computers and software components
- NGOP is an *open framework* for building distributed software/OS/hardware monitoring systems for large and small computing centers and systems
  - Comes with ready to use components
  - Is to be extended/customized by end users
  - Can perform simple corrective actions
- Being developed by ISD in collaboration with OSS
- NGOP is being moved into production
  - Goal: integration into centralized CD operations and support infrastructure
  - Already used to generate Helpdesk tickets for 24x7 systems including production farms

URL: <http://www-isd.fnal.gov/ngop>

## FBSNG – Farm Batch System (Next Generation)

- Initially developed for Run II production farms
- From event parallelism (CPS) to file parallelism (FBS)
- FBS and FBSNG have been in production since 1998
- Currently managed farms:
  - CDF, D0 on-line production farms
  - 2 “Fixed Target” (common use) farms recently merged into one farm
  - CMS USA Tier 1 center
  - NIKHEF – D0 collaborators
  - NWU – D0 (and CDF?) collaborators
  - Other HEP sites, one known corporate site
  - Number of micro-farms (1-2 computers)

URL: <http://www-isd.fnal.gov/fbsng>

# FBSNG Concepts

- No load measurement
- Instead, **resource counting**:
  - Know resource capacity of farm nodes
  - Know process resources requirements
  - Know which process runs where
  - Start new process when and where resources are available
  - Makes the system *simple, robust, flexible, portable*
- Resource counting can be used for SMPs and farms
- Unit of operation is an array of batch processes (*job section*)
- FBSNG job consists of (dependent) sections

# FBSNG Concepts

- Abstract Resources
  - All resources in FBSNG are *abstract* semaphores created by administrator
  - Local resources – associated and available locally on farm nodes
    - CPU
    - Disk
    - Tape drives
  - Node attributes – features of farm nodes
    - OS flavor
    - Installed software
    - Logical attributes (“red”, “green”, used to partition the farm)
  - Global resources – resources shared by all the processes on the farm
    - network throughput
    - NFS-exported disks
    - global semaphores
- No predefined resources
- Allows high flexibility in farm/cluster configuration and management

# FBSNG Features

- Scheduler
  - Task/project/group prioritization
  - Fair-share scheduling
  - Guaranteed scheduling
  - Resource utilization quotas
- Interactive batch jobs
  - Run my process along with and in the same way as other batch processes
  - But... give me interactive connection to it
- Kerberos support
  - Client authentication – ability to access over WAN
  - Creates credentials for batch processes
- Easily portable
  - 99% Python. Some C, C++ code, mostly non-essential
  - Supported/runs on Linux, IRIX, SunOS, OSF1

# FBSNG User Interface

The screenshot displays the FBSNG User Interface, which includes a command-line interface (CLI) and a web interface (FBSWWW).

**CLI Interface:** The top window shows a list of jobs with columns for JOB\_ID, SECTION\_NAME, USER, QUEUE, STATUS, START, and FINISHED. The jobs are listed in a table format.

**Web Interface:** The bottom window shows the FBSWWW interface, which includes a navigation menu (Queues, Jobs, Nodes, Process Types, Graphs) and a table of queues. The table has columns for Name, Status, Default Process Type, Share, Prio, Waiting, Ready, Running, and Total.

Name	Status	Default Process Type	Share	Prio	Waiting	Ready	Running	Total
BlueQ	OK	Worker_6	1.00	0	0	0	1	1
CadmiumQ	OK	Worker_13	1.00	0	0	0	0	0
ChartreuseQ	OK	Worker_9	1.00	0	0	0	0	0
CobaltQ	OK	Worker_14	1.00	0	0	0	0	0
DoveQ	OK	Worker_test_1	1.00	0	0	0	0	0
EndQueue	OK	EndSAM	1.00	50	15	0	0	18
GreenQ	OK	Worker_2	1.00	0	0	0	0	0

**Graphs:** The bottom window shows a graph titled "CPU utilization by project (week)". The Y-axis is "Number of CPUs" (0 to 120) and the X-axis is "Time" (Fri, Sat, Sun, Mon, Tue, Wed). The graph shows the number of CPUs utilized over time, with a legend for various projects and workers.

Legend for CPU utilization graph:

- EndSAM
- IO
- StartSAM
- Test
- Worker\_1
- Worker\_11
- Worker\_12
- Worker\_13
- Worker\_14
- Worker\_15
- Worker\_2
- Worker\_3
- Worker\_4
- Worker\_5
- Worker\_6
- Worker\_7
- Worker\_8
- Worker\_9
- Worker\_test
- Worker\_test\_1

Updated: Thu Aug 30 11:33:13 2001

- Command line interface
  - Job submission, control
  - Farm management
- GUI
  - Job monitoring, control
  - Farm management
- Python API
  - Provides full functionality
  - UI, GUI, FBSWWW use API
- Minimal requirements for client-only installation. Provides an ability to access the farm over WAN
- Web interface (FBSWWW)
  - Resource/job monitoring

## FBSNG – big picture

---

- FBSNG is full scale batch system for farms and clusters
  - Has worked well on farms with various kinds of users and resource utilization patterns
  - Robust, easy to support and manage
- As the result of CPS, FBS, FBSNG projects, ISD has the expertise in the area of development and maintaining of batch control systems
- ISD has the code of FBSNG – potential for further development:
  - Analysis vs. production: what is common and what is different ?
  - Grid interface ?

## Disks as data storage media

- It is recognized that disks approach tapes in \$/MB
  - Disks are random access devices, more efficient, easier to deal with
  - Hard to get PC with small disk = disk comes for free these days
- ISD is exploring possibilities of wider use of disks as data storage media in HEP applications
- Some investigative work has been done in using disks instead of tapes in long-term mass storage applications
- Typical farm as an array of disks controlled by array of CPUs, or **disk farm**:
  - Capacity:  $100 \text{ nodes} * 2 * 30 \text{ GB} = 6\text{TB}$
  - Throughput:  $10 \text{ MB/s} / \text{node} * 100 \text{ nodes} = 1 \text{ GB/s}$
- Disk Farm is a product which helps utilize large unused disk capacity of farm nodes

## Disk Farm – distributed farm disk storage

- Organizes distributed disk space spread over nodes of the farm into global *virtual* name space
  - Physical file path: node1:/local/stage1/dfarm/xyz123
  - Virtual file path: /e123/runII/data/mc123.dat
- User interface:
  - get, put, mkdir, rmdir, rm, ls commands – similar to UNIX FS access commands. E.g.:

```
$ dfarm mkdir /e123/runII/data
$ dfarm put /scratch/mc123.dat /e123/runII/data
$ dfarm ls /e123/runII/data
```
  - On the node where the data happens to be:
    - “get” is local – faster, cheaper
    - Ability to read data without copying out of disk farm
    - “put” is almost always local
  - Users are limited by (optional) global quotas, not physical sizes of individual volumes

## Disk Farm features and status

- Data replication:
  - User: here is my file, make 3 copies of it on 3 different nodes
  - In case a node goes down, 2 copies are still available
- Relatively new project – mostly developed in July-August 2001
- Current installations:
  - CFD production farm:
    - Total disk capacity of 150 worker nodes: 8TB
    - Currently, 1.4TB on 90 nodes under disk farm management
  - Fixed Target farm:
    - 1.8 TB on 90 nodes
  - NWU farm
- Being considered for CMS Mosix-based analysis farm

URL: <http://www-isd.fnal.gov/dfarm>