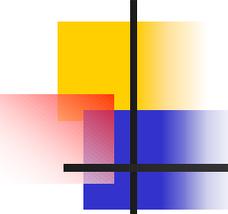


FBSNG Overview

Jim Fromm, Krzysztof Genser, Tanya Levshina, Igor Mandrichenko

**Farms and Clustered Systems Group,
Computing Division,
Fermilab**

July 12, 2000



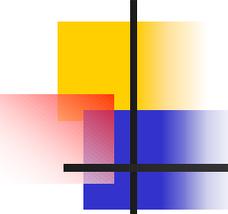
Acknowledgements

FBS developers:

- Mark Breitung
- Marilyn Schweitzer

FBS users and FBSNG testers:

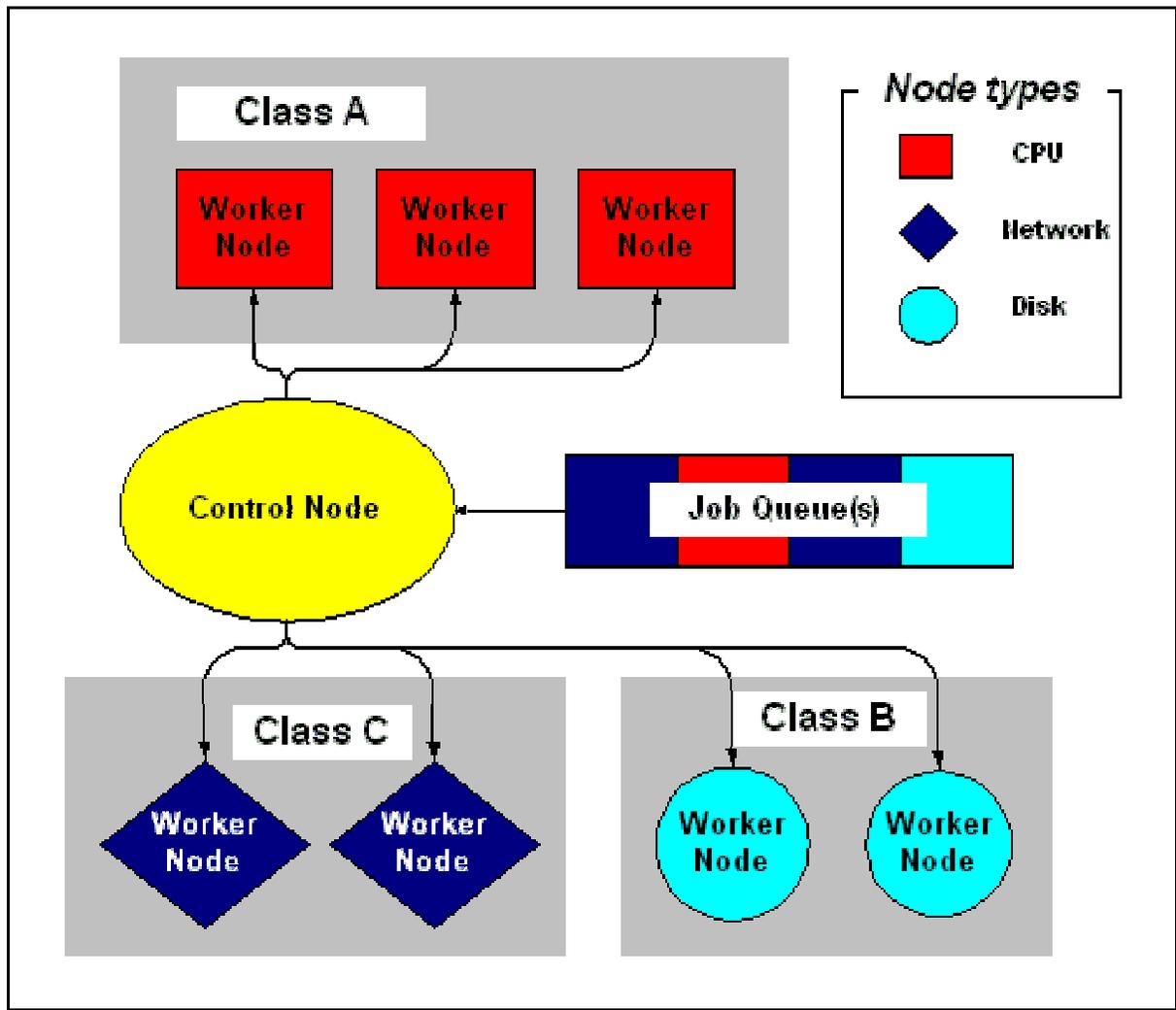
- Antonio Wong Chan (academia Sinica, Taiwan, CDF collaboration)
- Yen-Chu Chen (academia Sinica, Taiwan, CDF collaboration)
- Thomas Las (Manooka, IL Junior High School)
- Miroslav Siket (academia Sinica, Taiwan, CDF collaboration)
- Heidi Schellman (northwestern university, Illinois, D0 collaboration)
- Steve Wolbers (Fermilab, CDF collaboration)
- Ping Yeh (academia Sinica, Taiwan, CDF collaboration)



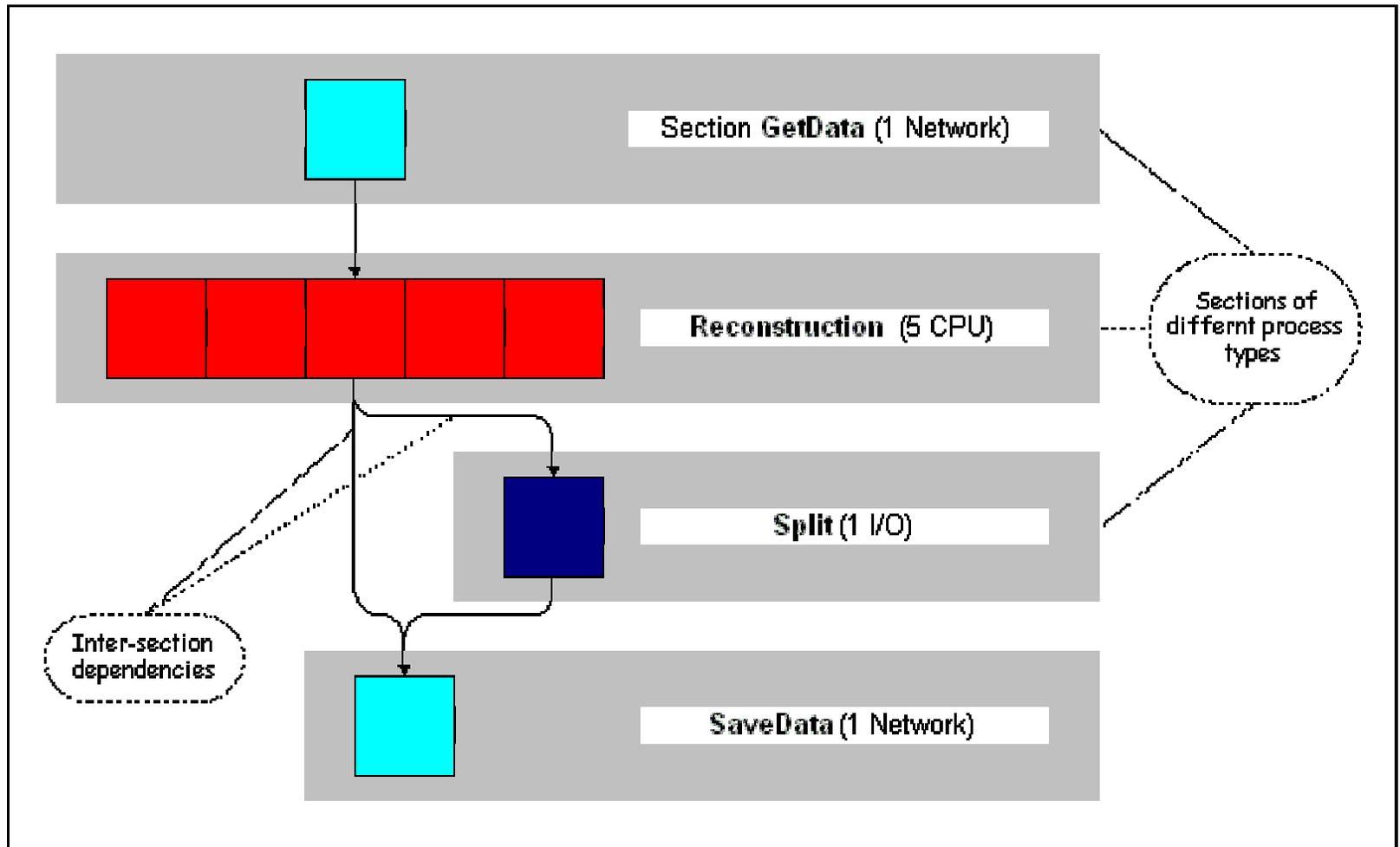
History and Status

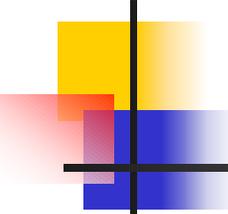
- FBS project goals
 - Replace CPS batch in PC farms environment
 - Develop a farm batch system for file-based parallel data processing style adopted for RunII
 - Do not preclude even-based parallelism
- Milestones
 - Spring 1998 - initial FBS design, first working prototypes
 - Fall 1998 - first production users (E871)
 - Spring 1999 - review by CDF, D0
 - Fall 1999 - FBS v2.2
 - Fall 1999 - beginning of FBSNG project
 - Since may 2000 - alpha testing of FBSNG by CDF
- Currently FBS is running on:
 - Fixed target farms (fnsfh, fnsfo)
 - D0 farm
 - CDF farm

FBS Concepts: Farm Model



FBS Concepts: Job Consists of Sections

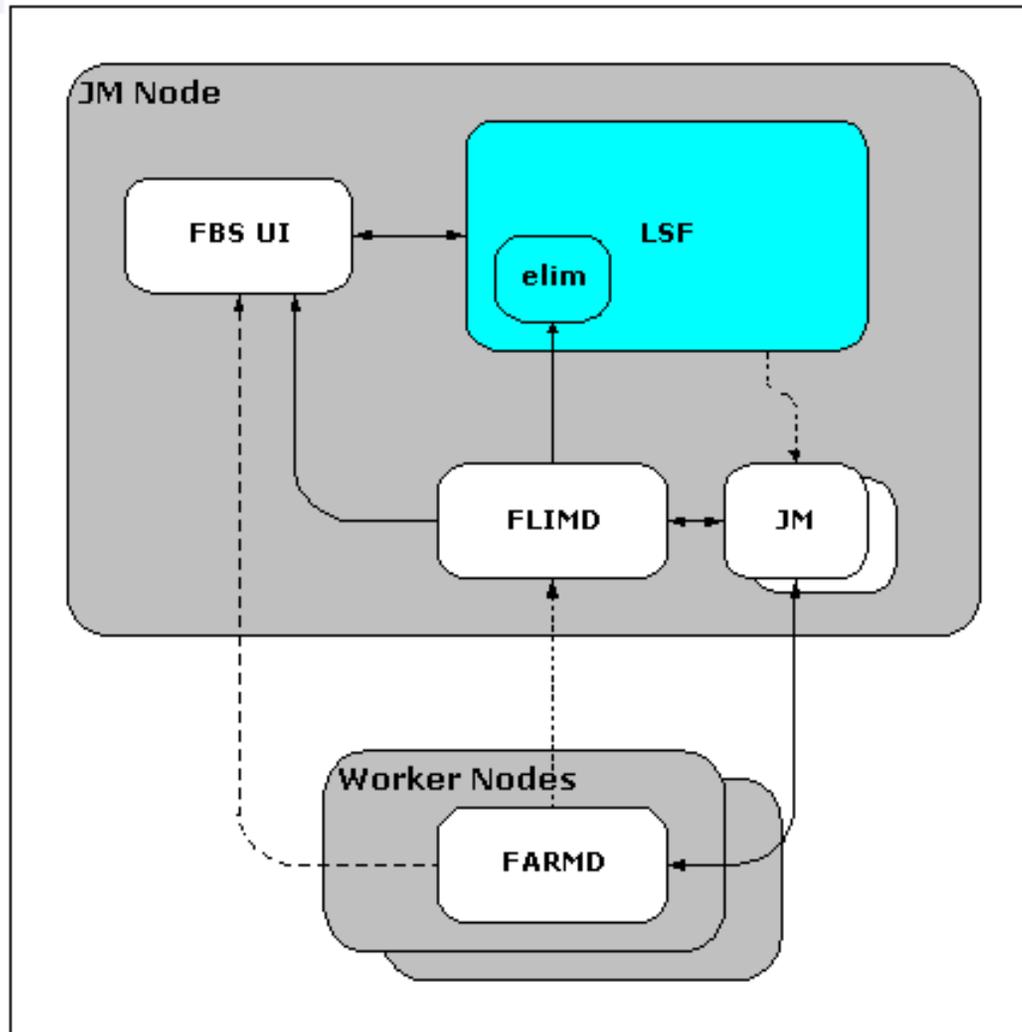




FBSNG Project Goals

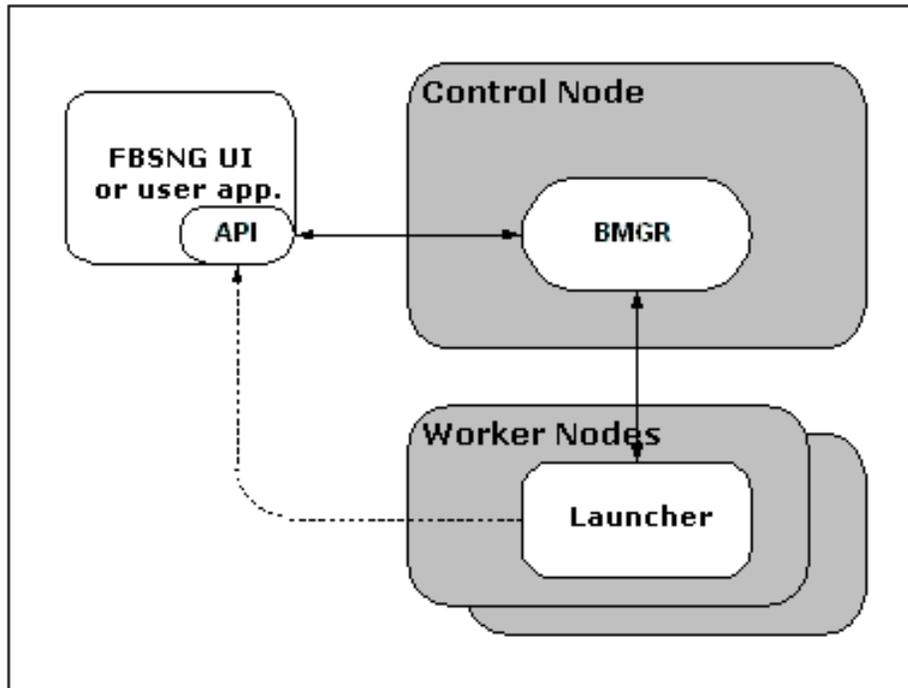
- Immediate
 - Stop using LSF as scheduler and job storage
 - Reduce maintenance and support costs
 - Avoid possible scalability problems
 - Release first version of FBSNG early
 - “Do not brake anything!”
 - Preserve as many features as reasonably possible
 - Add few fundamental features
 - Abstract resources
 - Customizable scheduler
 - Basis for evaluation and future development
- Long-term
 - Having FBSNG as integral product creates possibilities for future development
 - Dynamic re-configuration
 - Further development of resource management
 - Integration with FIPC
 - Open system (API)

FBS Design (Big Picture)



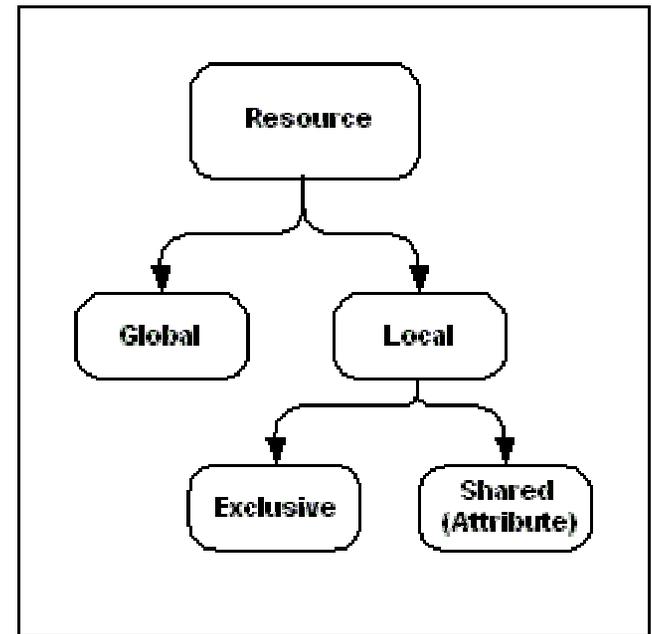
- LSF is used for
 - Scheduling
 - Job storage
 - Some resource management

FBSNG Design (Big Picture)

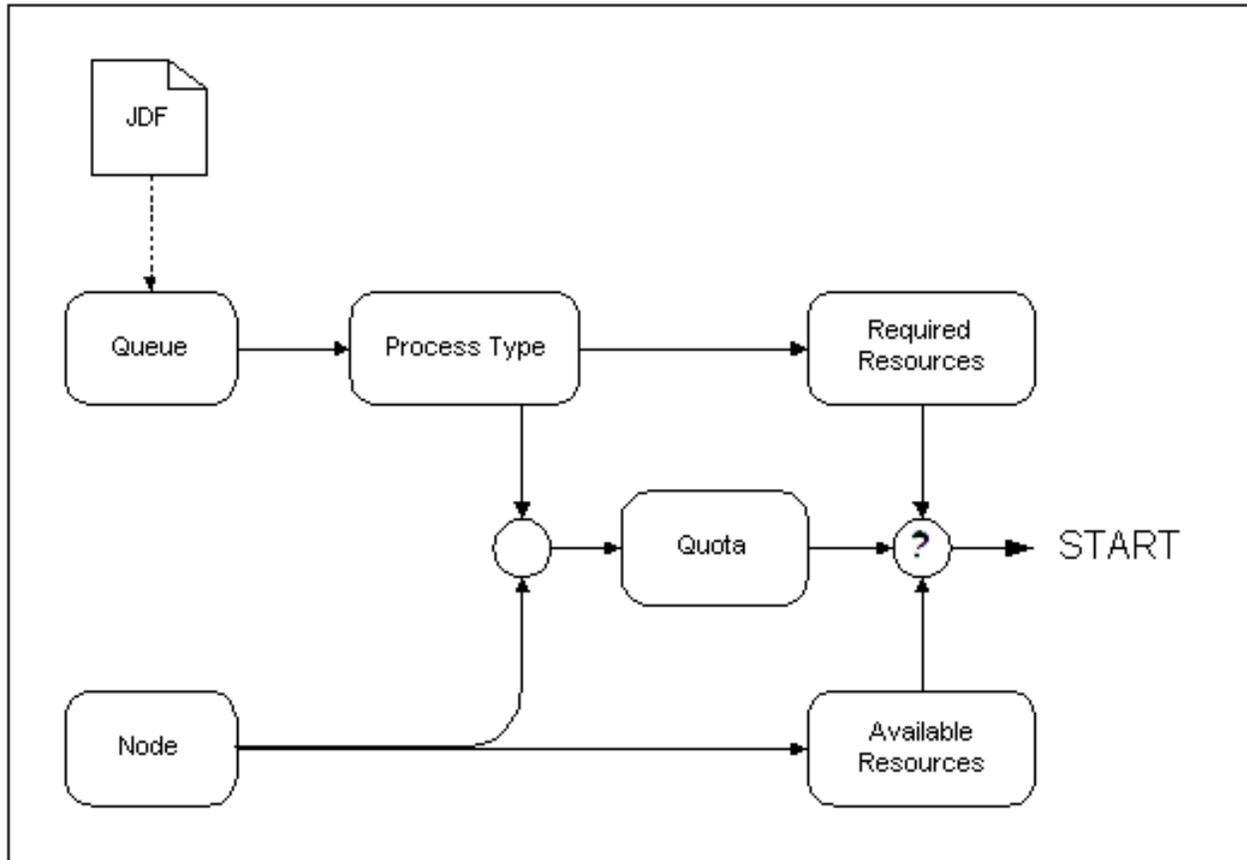


- **BMGR functions:**
 - Scheduling
 - Resource management
 - Job storage
 - Communication with API clients

- **Global resources**
 - Hosted by whole farm
 - Available on any node
 - Examples:
 - Disk space on NFS server
 - Network bandwidth
- **Local resources**
 - Hosted by individual nodes
 - Available only on the node
 - Examples:
 - CPU
 - Local disk
 - Tape drives
- **Node attributes**
 - Unlimited local resources
 - Examples:
 - Special software installed
 - Version of OS

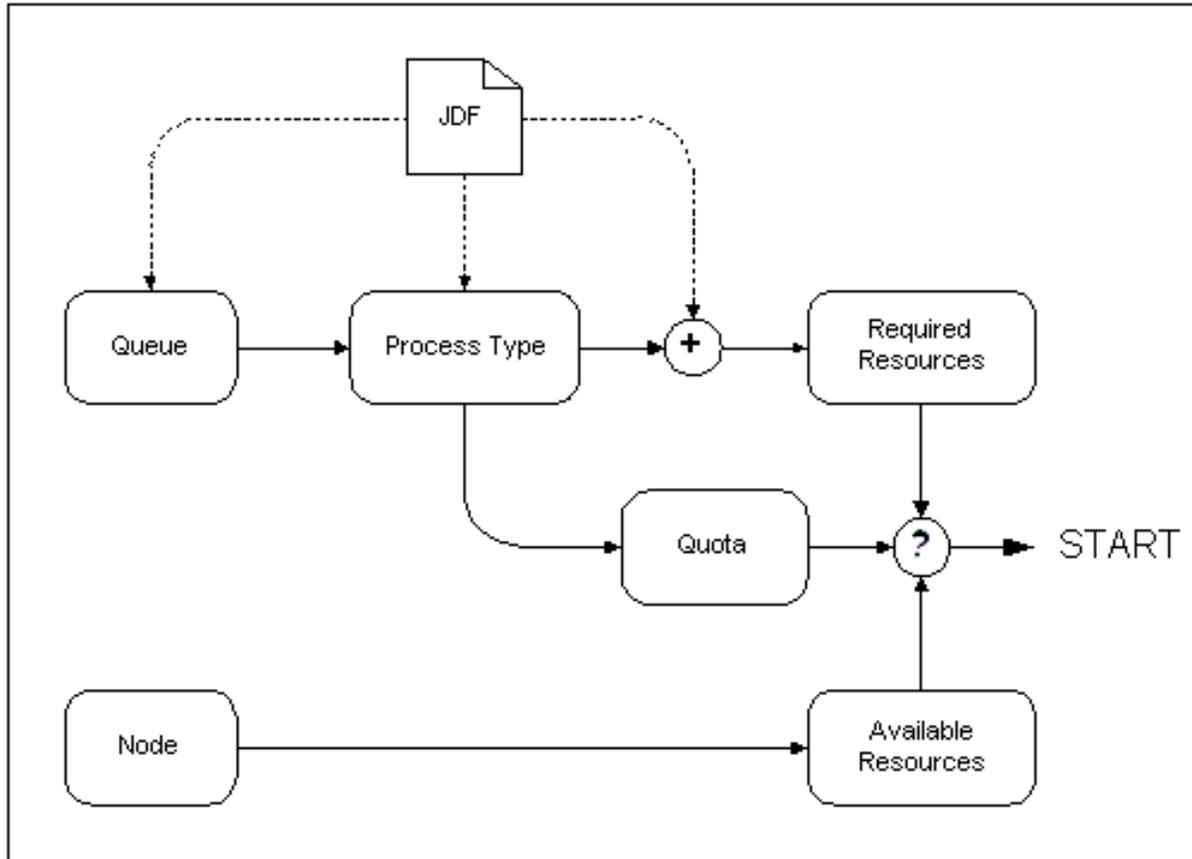


FBS Resources Management

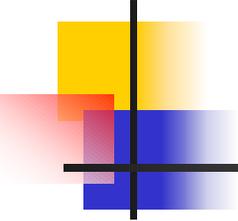


- Queue statically associated with process type and resources
- Per-node per-process type quotas

FBSNG Resource Management

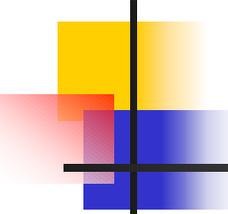


- Only per-process type resource quotas
- Soft association between queue and process type (user can override)
- User can request additional resources



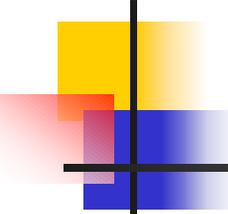
FBSNG Scheduler

- Algorithms are based on the idea of dynamic priorities
- Controllable fair-share scheduling
 - Projects are assigned relative shares of farm resources
- Guaranteed scheduling
 - No infinite delays for *big* jobs
 - Will hold *small* jobs if necessary
- Customizable
 - Wide variety of configuration options



UI: FBS vs. FBSNG: "Do not break anything!"

- Obsolete:
 - JDF END_TIME field
 - "move" command (replaced with "cprio")
- Renamed/changed:
 - "farms" -> "fbs", "fbsng" - synonyms
 - e.g. "farms submit" -> "fbs submit"
 - JDF:
 - BEGIN_TIME -> HOLD_TIME, time specification format
 - Absolute, relative
 - JOB_STDOUT -> SECT_STDOUT
 - LOCAL_DISK -> PROC_RESOURCES
 - e.g. "LOCAL_DISK=5" -> "PROC_RESOURCES = disk:5 tape:1"
- New commands:
 - ls, lj – list jobs, sections
 - cprio – change section priority
 - slog – show section log
 - hold/release job/section/queue
 - ptypes, resources – statistics on process types, resources
- See v1.0 Release Notes



FBSNG: Example of Job Description File (JDF)

```
SECTION Stage
QUEUE=IO_Q
EXEC=stage.sh VSN123 /stage01
NUMPROC=1
```

First section: pre-stage data
Queue to submit to
Command to execute
1 process

```
SECTION Reconstructor
QUEUE=Long_Q
EXEC=reco123.sh /stage01 /stage02
NUMPROC=10
DEPEND=done(Stage)
PROC_RESOURCES=disk:5 Linux Worker
```

Second section: reconstruction

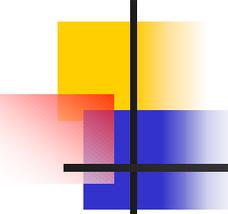
10 processes
Only if pre-staging succeeds
5GB of local disk, Linux, Worker node

```
SECTION Clean-up
QUEUE=Short_Q
PROC_TYPE = Light
EXEC=clean.sh /stage02 VSN123
NUMPROC=1
PRIO_INC = 5
DEPEND=failed(Reconstructor)
```

Emergency clean-up section

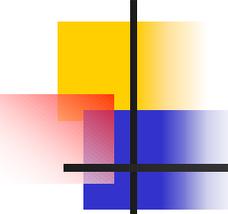
Override default process type

Run at higher priority
Run only if reconstructor fails



FBSNG Users Interface

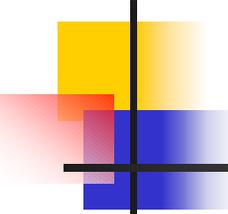
- Job control
 - Submit – *jobs can be submitted from any node of the farm*
 - Monitor status
 - Kill/cancel
 - Hold/release
 - History
 - E-mail notification
- Farm node status
- *Resource utilization statistics*
- *Scheduler status*



FBSNG Batch Process Environment

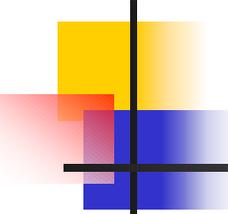
- Environment Variables
 - FBS_JOB_ID
 - FBS_SECTION_NAME
 - FBS_JOB_SIZE - number of processes in the section
 - FBS_PROC_NO - logical process id (1...FBS_JOB_SIZE)
 - FBS_SCRATCH - scratch directory created for the process
 - HOME - home directory
- Current working directory is HOME
- Stderr, stdout as specified in JDF
- Unique UNIX session ID

- Job submission
 - FBSCClient
 - FBSJobDesc
 - FBSSectionDesc
- Job monitoring and control
 - FBSJobInfo
 - FBSSectionInfo
 - FBSProcessInfo
- Resource management and monitoring
 - FBSQueueInfo
 - FBSNodeInfo
 - FBSNodeClassInfo
 - FBSProcessTypeInfo
- Python binding available
- UI, GUI are implemented in terms of API



FBSNG Requirements

- LSF no longer required
- On control node:
 - Bmgr daemon
 - Logd daemon (optional)
- On each worker node:
 - Launcher
 - Rstatd (optional)
- Software/hardware requirements:
 - Python (99% FBSNG sources are python)
 - Tcl/Tk, Tkinter (for GUI)
 - FCSSLIB (common Python module library)
 - Configuration file synchronized on all worker nodes (NFS works well)



FBSNG Project Status and Plans for Future

- FBSNG V1.0 (IRIX, Linux) released July 6th
- Soon to be available from Fermitools
- V1.0 is evaluation version
 - Solicit feedback (suggestions, criticism) from users
 - http://www-isd.fnal.gov/fbsng/fut_feat.html
 - Make plans based on such feedback