# How to Make LSF Work for You

Iwona Sakrejda
PDSF/LBNL

MSU, 08/13/2003

# Outline

- Introducing PDSF (the other useful farm).
- Account configuration for STAR users.
- Batch system and why do we need it.
- Job submission.
- Job monitoring (When is my job going to run?)
- Job manipulation (Killing, changing properties and requirements, suspending, moving in priority)
- Interactive processing.
- Help resources.

Batch system will make your life easier, but you have to understand it…..

# Introducing PDSF

RH7.2

7 interactive nodes

pdsfint.nersc.gov

~10GB local /scratch

RH7.2

3 Globus Gatekeepers

pdsfgrid1

pdsfgrid2

pdsfgrid3

RH8.0

pdsflx008

BATCH

RH 7.2

~380 batch CPU's

PIII 650MHz –

Athlons 1.8GHz

~10GB local /scratch

Queues normalized to

1GHz

pdsflx009-pdsflx250

NFS mounted STORAGE

~60TB

pdsfdv08 - pdsfdv60

2 mirrors of the STAR database

stardb.nersc.gov

HPSS

# Account configuration for STAR users.

## DO NOT

- Do not create a .tcshrc file, tcsh reads .cshrc file if the other one is missing.
- PDSF STAR user accounts come configured with the new version of STAR libraries from BNL via AFS.
- If you need to make another version your default add starver SL0xy at the end of your .cshrc file.
- Software packages available as modules.
    - module avail
    - module (un)load
    - module list
- Local software environment - .pdsf_setup file

#!/bin/tcsh –f

setenv STAR_PDSF_LINUX_SETUP use_local

(other options – use_debug, use_afs, use_none)

# Why do we need a batch system?

- Manage individual CPUs and do load balancing and job monitoring and control across the cluster (faster CPUs are loaded first).
- Fair Share scheduling
  - on PDSF STAR users have to share with other groups (~70%)
  - this works to everybody's advantage (prior to last QM STAR got between 90% and 100% of the cluster).

- Shares and priorities

dynamic priority = (number of shares)/
((1+number of jobs running)*RUN_JOB_FACTOR +
(total run time of jobs in execution)*RUN_TIME_FACTOR))

- What is my share and priority?

bhpart
bhpart –l

# LSF & job submission

Use STAR job scheduler if possible

(they will end up in LSF anyway)

- What is my environment in batch?
  - -default:a copy of your environment at the time of submission (without LD_LIBRARY_PATH) + bash
  - -L option of bsub (man bsub) – login environment
  - printenv
  - AFS gateway misteries (PDSF only)
- Resource requirements
  - dvio – disk vault access moderator
  - scratch requirement
  - memory requirement
  - combining requirements

# More about the dvio resource

## Use it, even if you think you do not need it.

•Each disk valt has only 2CPU's so it can talk to a finite number of jobs.

•If too many jobs try to read data at the same time, the disk vault goes into a thrashing mode (spends all the time trying to figure out whom to serve) and then gets "wedged" and only reboot can bring it back ( 510 486 6821, but suspend some of your jobs first!!!!! )

Each disk vault has 1000 units assigned to it

Each disk vault can handle about 30 I/O intensive jobs ( /aztera is an exception, it can handle couple hundred of jobs simultaneously)

Syntax is important – check PDSF FAQ!

bsub –q medium –R "select[defined(dv27io)&&defined(dv34io)&&scratch>2000] rusage[dv27io=33:dv34io=33] " <your job>

Read from your disk vault write to /scratch

# The Wonders of Spool

- Spool is a work area for the batch system
  - Info about your submitted jobs is kept there
  - If you spool your input script, it is kept there too (copied at the time of submission, only then LSF directives embedded in a script are taken into account)
  - Your standard output and standard error go there even if you send them to a local file - that's where bpeek looks for info.
- Spool can be overwhelmed if abused the same way as any other disk vault (it is a disk vault) and if there is no spool, the whole system stops.
- E-mail from your jobs
  - If standard output and error are sent to a file, there is no e-mail.
  - On PDSF files larger than 100MB are dropped.
  - You can ask to be e-mailed when your job finishes.

# Job monitoring

- bjobs - shows info about your pending, running and recently completed jobs.
  bjobs –l
  - helps you understand why are your jobs pending
  - lets you check how your jobs are doing (CPU/wall clock)
- bpeek <job ID> - gives you access to the standard output of your job while your job is running (does not work if /dev/null specified as std I/O)
- bhist – lets you examin a history of any job anybody EVER ran on the cluster
  It also shows Error code even if you got no output.
  - n <number> (0, all the past logs)
  - d – completed jobs
  - l – long version

# Job manipulation

- xlsbatch (an X window, job monitoring too, great for beginners)
- bmod
  – any bsub parameter can be modified for a pending job
  – only –R part can be modified for running jobs
- bkill
  – you can kill one job if you specify job ID or all (bkill 0)
  – selective killing requires scripting or xlsbatch
  – Sys admin cannot kill any job you cannot kill
- btop n
- bbot n

  seting priority in bsub does not work with the fair share scheduling

# Interactive processing.

You can run interactive on batch nodes (and at PDSF it is actually encouraged).

bsub -I -q short <your application> (but no LD_LIBRARY_PATH)

bsub -Is -q short /bin/tcsh

or

bsub -I -q short xterm

Then you can configure your environment and run until time limit.

lsrun strongly discouraged – jobs with no companion batch process are killed automatically

# *PDSF help resources*

- *Web pages*
- *FAQ*
- *Man pages*
- *Mailing lists (pdsf-announce@lbl.gov)*
- *Hypernews forums (pdsf-hn)*
- *FAQ (PDSF and STAR)*
- *Web form*
- *E-mail (consult@nersc.gov)*
- *Call*

Batch system (LSF) is there to make your life easier –
so USE IT!