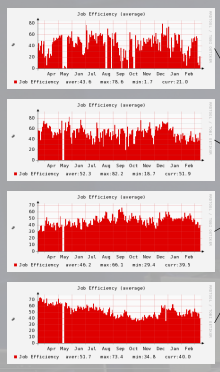
Batch Efficiency at CERN

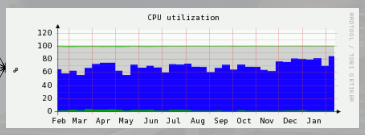
CERN IT Department

Problem description



User job efficiency
LHC Experiments

Public batch resources last year



Box utilization averaged 67%

Low user job efficiency ⟷ **Poor resource utilization**

Reasons

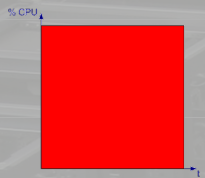
Site problems:
- Slow network ?
- Site provided job wrappers
- Software bugs
- Service incidents
- Tracking of nodes with problems
- Protections against misbehaving users

User job problems:
- Badly coded programs
- Bad or wrong usage of the system
- Bugs in the code
- Large job overhead
- Processing of large amounts of data
- Waiting for tape recalls
- "Nature of the job": I/O intensive activity
- Misbehaving users

CPU bound jobs

- Little I/O
- full use of CPU
- highly efficient in terms of CPU/Wall time

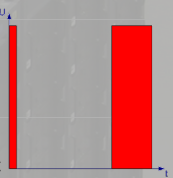
Batch system queues:
- New queues created
- CPU/Wall time fixed to 80%
- enhanced priority



I/O bound jobs

- Waiting for network
- Waiting for slow storage
- Reading large amount of data
- Wait for data recalls from tape

Batch system queues:
- Standard queues
- Wall time limits up to 3x larger than CPU time limit

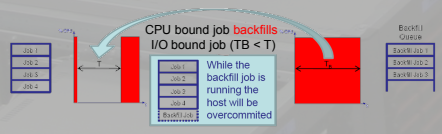


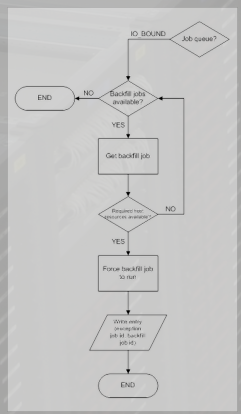
Improving the box usage

Basic idea: Use short CPU bound jobs and run them on underused machines (backfilling)

Results: Increased box utilization

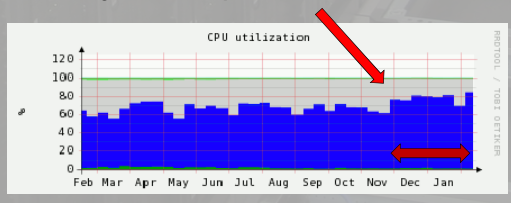
I/O job queues:
- Identify jobs which are waiting for tape recalls
- Get an estimated waiting time from the tape system
  - Based on historical data
  - Used to estimate the window size
- Check CPU queues for fitting jobs
- Force the CPU bound job to run on the worker node





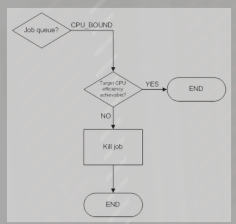
- Able to identify 28% of the idle jobs as jobs to be backfill during initial tests, without any tuning and no input from the users
- Implementation in C using the LSF API
- Triggered by the LSF idle job detection feature

The changes went into **production** in December 2008:



Noticeable increase on the average box utilization

CPU job queues:
- Careful idle job detection
- Kill jobs as soon as they cannot fulfill the 80% efficiency threshold



Challenges:
- Do not overload the worker node
- Respect user requirements for backfilled jobs
- Respect overall fair share
- Do not overload CASTOR server with queries

Conclusions:

- Improved the individual box utilization even under a chaotic mixture of different types of jobs
- Improved overall throughput
- Better turn around time for jobs submitted to the new queues for CPU efficient jobs
- Users are invited to use CPU queues whenever possible
  - By submitting to these queues users are categorizing their jobs which allows for optimized scheduling.

CERN - IT Department
CH-1211 Genève 23
Switzerland

www.cern.ch/it-fio