Batch Queueing System (BQS)

Overview

Fabio Hernandez
fabio@in2p3.fr

Lyon, June 30th 2006

Contents

- Context
- Brief history of BQS
- Current status
- Issues

Context

- Computer Centre's mission is to acquire and operate the computing resources to support scientific experiments
  - 30+ international experiments in nuclear, particle and astro-physics
  - Biomedical applications since a few years
- 20+ years record providing computing services round-the-clock
  - Mainframe based in until early 90s, Unix-based since then
- Demand is higher than offer, so we need a resource management system
  - We must make sure that the resources we are able to buy are used to the maximum possible capabilites

History of BQS

- Home-grown software being developed since 1992
  - Currently 2.5 FTEs work on it
- Maturity
  - First generation initially based on NASA's NQS
    - *12 worker nodes in 1992*
  - Second generation: reengineered and got rid of NQS layer
  - Third generation started in early 2000s
    - *Database-centric architecture*
    - *By the end of 2006: ~1000 worker nodes, 2000 CPUs*
- Extremely well suited to our needs
  - Several experiments competing for resources
    - *Several parameters for « programming » the scheduler by the operations people for meeting production targets*
  - Fair-share implemented since the beginning
  - Tools for extracting accounting data available since the early days

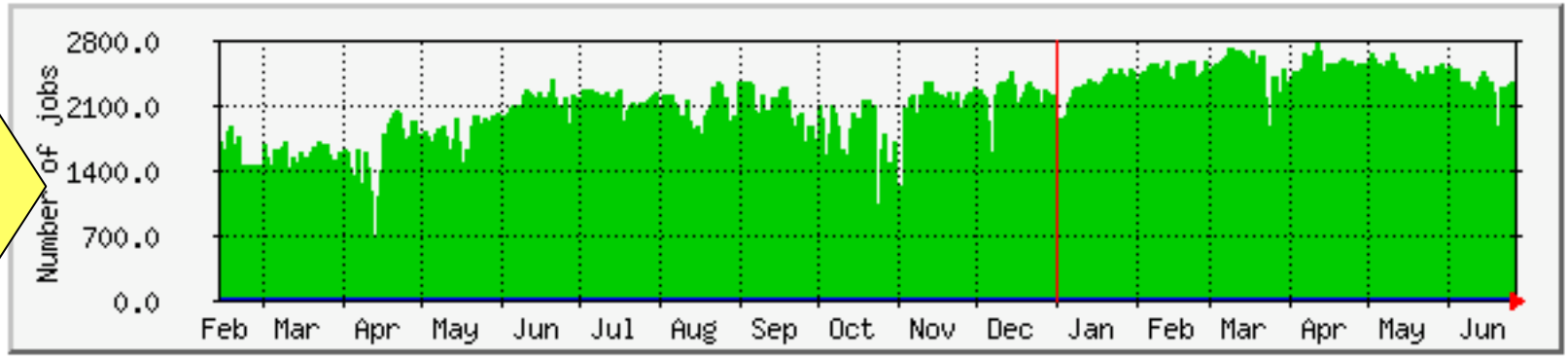
Current Status

- Database-centric architecture
  - 2 master hosts: 1 for database and another for job handling
    - *Spawning, getting status reports from jobs, collecting data on job resource consumption, ...*
- Some figures
  - Jobs in queue: 17.000
  - Jobs in simultaneus execution: 2.700
  - Throughput: 15.000 jobs/day

Current Status (cont.)

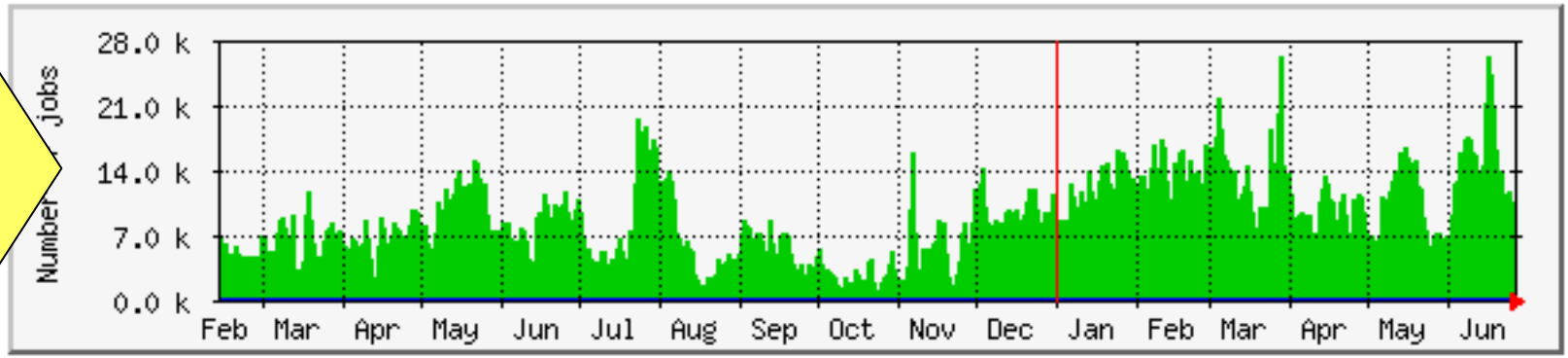
Running jobs

`Yearly' Graph (1 Day Average)



Max running jobs: 2790 Average running jobs: 2098 Current running jobs: 2360

Queued jobs



Max queued jobs: 27 k Average queued jobs: 8921 Current queued jobs: 9507

Issues

- Based on the planned capacity of the facility, by 2010 we would need to handle
  - 6000-8000 jobs in simulatenous execution
  - 30.000-40.000 jobs in the queue
  - Throughput: 40.000 jobs/day
- So, it would be very useful fo us to
  - Have a tool to **predict** the behavior of the system well in advance
    - *Currently, we observe how the system reacts to the load and then implement ways of improving it*
  - Have a tool to allow us to **optimise** the configuration of the system based on actual load
- Tools for modeling or simulating the whole system may be needed
  - This is complex task and we don't have the neither the manpower nor the know-how to to this