Stephan Wiesand
November 13, 2002

Sharing Compute Farms

- Experience from
  - Zeuthen
  - CERN
    - info taken from Thorsten Kleinwort's recent talk on linux@CERN given in Zeuthen
  - HEPIX
    - info taken from slides from HEPIX fall meeting

Zeuthen

- ice farm
  - 55 x dual PIII 800
  - + former husky farm
    - included recently
    - 10 x dual PIII 600
  - + new nodes RSN
    - 10 x P4 Xeon 2.4
- fully maintained by DV

- SGEEE batch system
  - "Sun Grid Engine, Enterprise Ediditon"
  - the batch system formerly known as "Codine"
  - Open source
  - AFS support

Zeuthen (continued)

- Job submission from all desktops & public hosts
- Interactive batch being introduced
- Groups in Zeuthen are too small not to share the work horse
- => Single farm
- dynamically shared between "projects" owning different "shares"
  - good utilization, little waste of resources
  - peak demands usually served easily

Zeuthen: Problems

- sharing fairly is easier with finer granularity
  - need to reduce time limit of "long" queues
    - avoids hogging
    - avoids latencies
  - BUT: "we've had 2 weeks for years, all our scripts depend on this"
  - some analyses <u>are</u> much easier to do in a single long job
- memory hogs may render farm unusable (SGE copes well with CPU hogs)
  - challenge to set the right limits

CERN

- LXBATCH
  - ~600 nodes
  - LSF batch system
- 1997
  - Version 3.2
  - Multicluster setup
  - Fixed partitions per group/experiment
- 2002
  - Version 4.3
  - Single cluster
    - using "fairshare"
    - much better utilisation
- RedHat 6.1 -> 7.3 RSN
  - support ~~outsourced~~

HEPIX Large Cluster Workshop

- From report by Alan Silverman:
  - "HEP is starting to get practical experience in running large clusters, practically all on Linux running on commodity hardware."
  - "More and more of these share the resources among several or many client groups"
- But from the same report:
  - "The larger the cluster, the more professional you must become at all levels from the ground up (literally). "

Sharing Farms: PROs & CONs

- PROs
  - Projects can buy for average load instead of peak load
  - Fewer idle nodes (which consume the same resources as a busy one)
  - The GRID will work like this anyhow
- CONs
  - Configuration needs to be identical across projects
    - -> Virtual servers ?
  - Batch system configuration tricky
  - Social problems
    - humans are lazy, greedy, careless...