Metrics Project Briefing

December 3, 2003

Steve Wolbers

Jeff Mack

Acknowledgements

- Many helped in the following described work:
  - Terry Jones
  - Igor Mandrichenko
  - Steve Timm
  - Dave Fagan, Jason Allen and Joe Boyd (CAB), Lance Weems and Marc Neubauer (CAF), Joe Kaiser (CMS), Dan Yocum (SDSS), Don Holmgren (Lattice QCD), Mike Stolz
  - Ruth Pordes and Lothar Bauerdick (MonALISA – CMS)

URL For Metrics Charts

- All of the reports/charts in the presentation are available via links at
  - http://www-csd.fnal.gov/metrics
- Charts are setup with the java viewer specified in the url:
    - ?init=java
- IE users can also use the activex viewer by specifying
    - ?init-actx

Presentation Overview

- Goals
- Usage Tracking at FNAL – Current Status and Prototyping Activities
- Key Metrics and Collection Methodologies
- How to benefit from the Grid
- Proposed Next Steps – Establishing a Project

Goals

- Define a project that will
  - implement a framework for collecting, maintaining and reporting on computational usage data. i.e. cpu, I/O, memory, disk. … for all production systems supported by the Computing Division.
  - enable analysis of historical data to determine needs for increasing processor, I/O or network capacity or to understand better usage patterns.
  - enable appropriate allocation of resources across groups/experiments and to determine who is using the resources. Provide feedback for changing priorities and allocations.
  - assist in discerning broad performance problems due to configuration issues such as poor I/O to cpu mix or job queue imbalances.

Goals – continued

- Develop list of tasks for the project:
  - Define metrics
  - Estimate effort, M&S costs and schedule for the project
  - Explore solutions
    - Home grown – developed from the ground up.
    - Borrowed and modified
    - Commercial
    - Other
- Compare solutions
- Recommend strategy

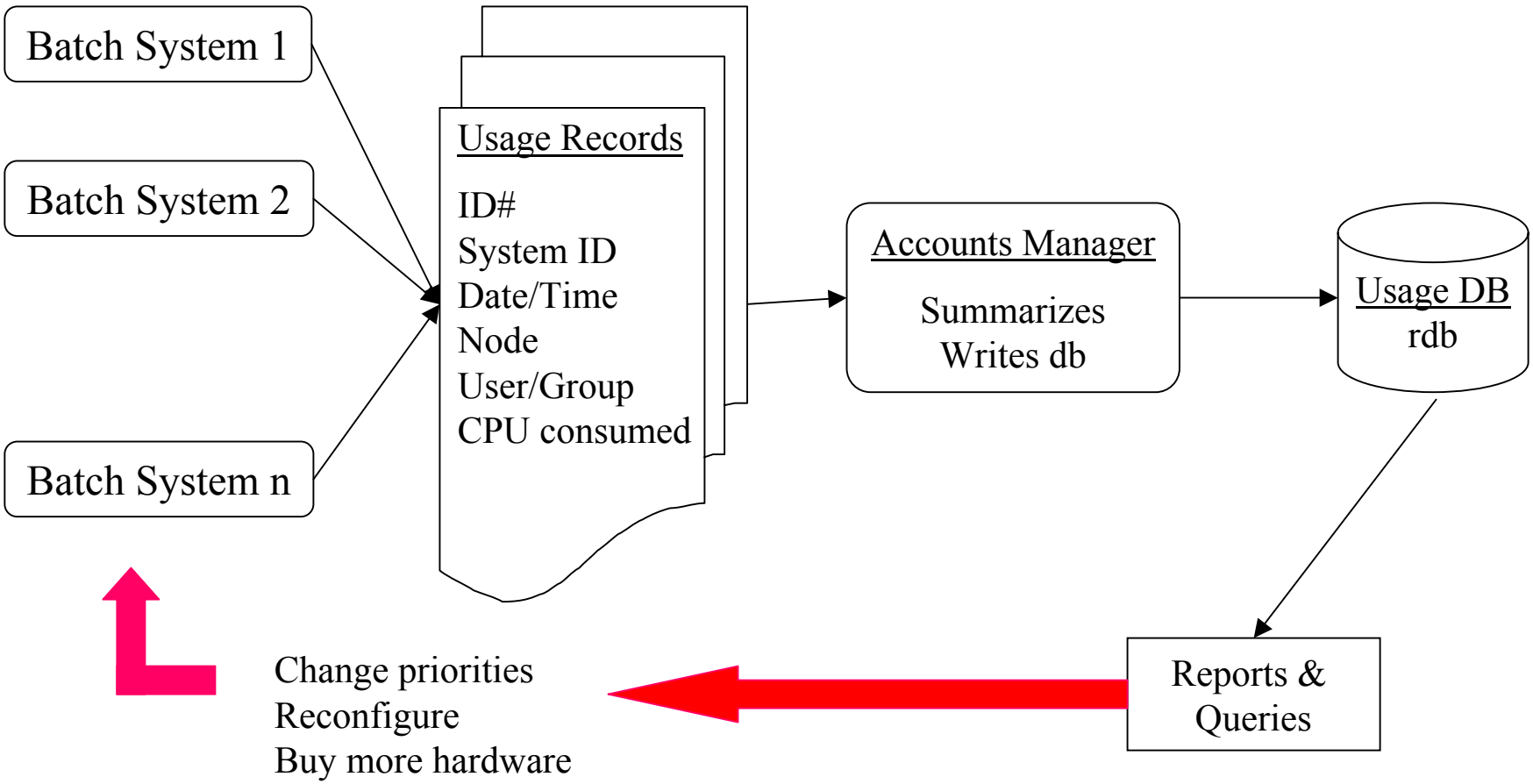
Current Process Accounting

- Process accounting model used by FARMS (and most other Unix systems) for last 12 years reports cpu usage on a daily, weekly and monthly basis.
- System uses standard Unix accounting methods.
- Reports can include summary usage by node, user and group.
- No relational db involved. Data is maintained in flat-files. (The original plan was for this to feed a db, but this was never built)
- This system is difficult to maintain and does not facilitate adhoc reporting and querying inherent in db-based systems.

Status – Process Accounting Efforts

- Recent effort (since beginning of year) has been put into installing software and updating configuration files to implement process accounting on all of the major production systems.
  - Farms. D0 CAB and CDF CAF - complete
  - CMS - work in progress
  - SDSS - ?
- FNALU, KTeV,CDF SGI, D0 SGI & MISCOMP already had process accounting enabled.
- This work involved mainly Terry Jones working together with the respective sys admins.

Basic Architecture



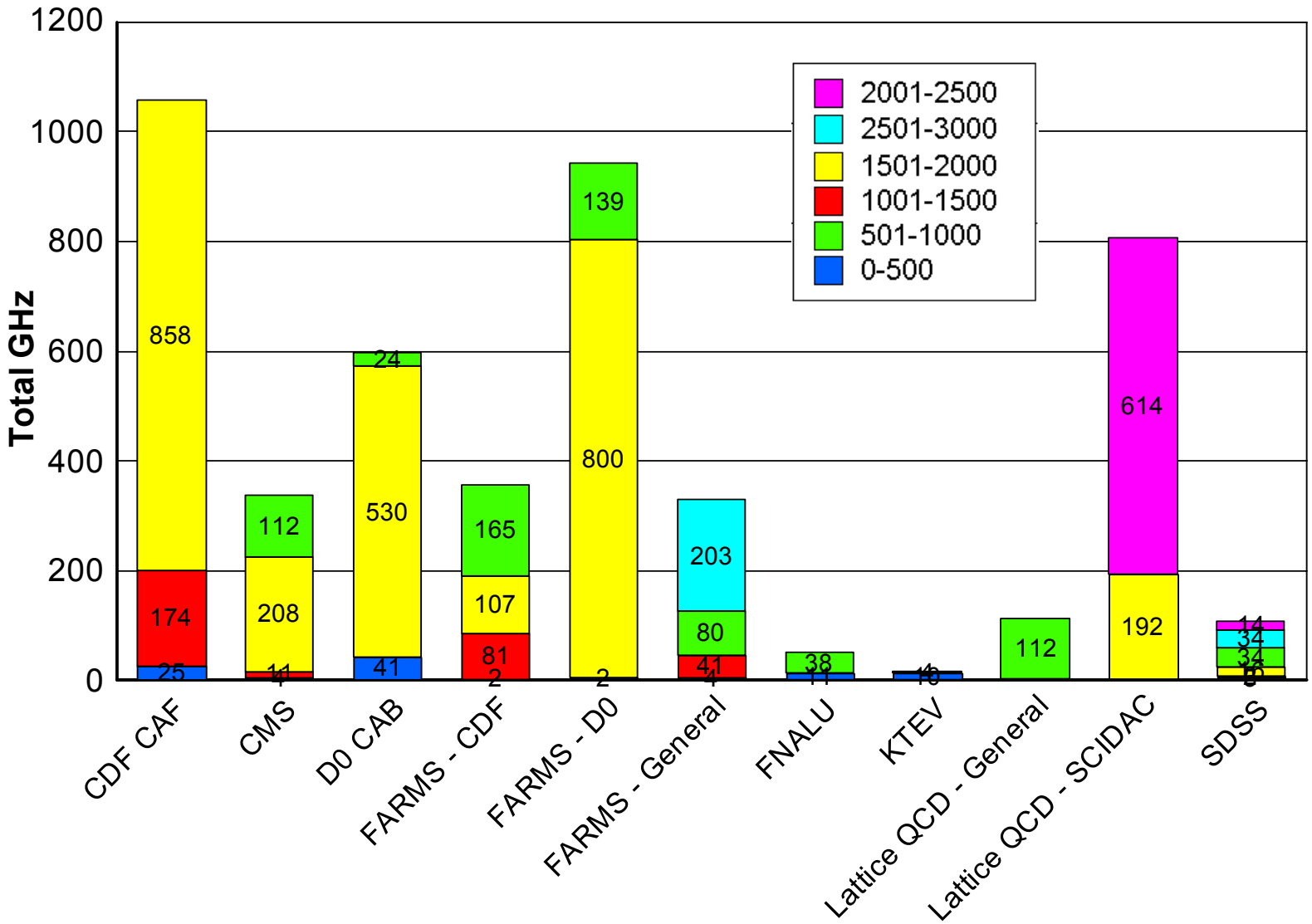
Status – Prototyping Activities

- Most of the recent effort has been spent on collecting monthly process accounting data and storing it in a prototype db.
  - Created db for maintaining historical data.
  - Import report data into db (MS Access).
- Processor performance is factored in to usage reporting.
- Reports have been designed to provide concise historical view of cpu utilization using this data. No other metrics besides cpu usage being stored and analyzed.

Status – continued

- System configuration tables have been created that contain info on all nodes for most of our major production systems. CPU List db
- Equipdb was not used because it did not have complete mapping of node to system, cpu speeds and number of processors.

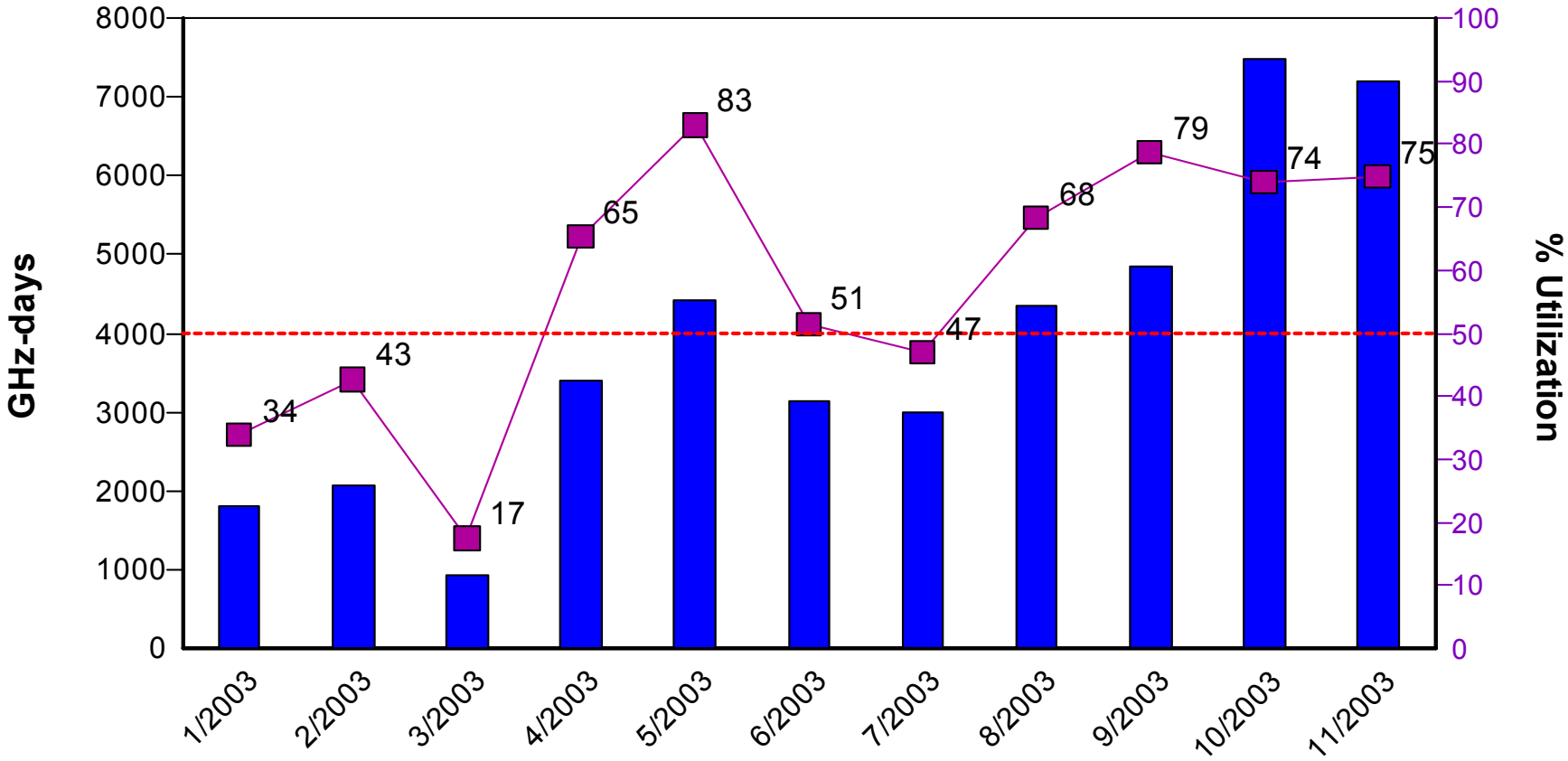
System Capacity and Processor Speeds



Status – Prototyping Activities

- Crystal Reports used as reporting tool.
  - Windows-based with license needed to design reports
  - Very powerful and programmable reporting capability.
  - Provides charting.
  - Creates web viewable report and chart objects.
- Crystal Enterprises (and IIS)
  - Manages and serves Crystal Reports on web. Licenses (which can be shared) are needed to view reports.

General Farms



Process Accounting Reports

- Farms (menu for a variety of reports)
- D0 CAB
- CDF CAF

Job Accounting - PBS

- Used by D0 CAB and Lattice QCD Cluster
- D0 CAB data is loaded into mySQL db
- Standard reporting tools such as used for process accounting, e.g. Crystal Reports, can be used via mySQL ODBC driver.
- QCD Cluster creates text based accounting records
- Accounting is job oriented but multi-node jobs on lqcd not reporting cpu usage for all nodes in job
- Processor speed differences may not be accounted for in usage records.

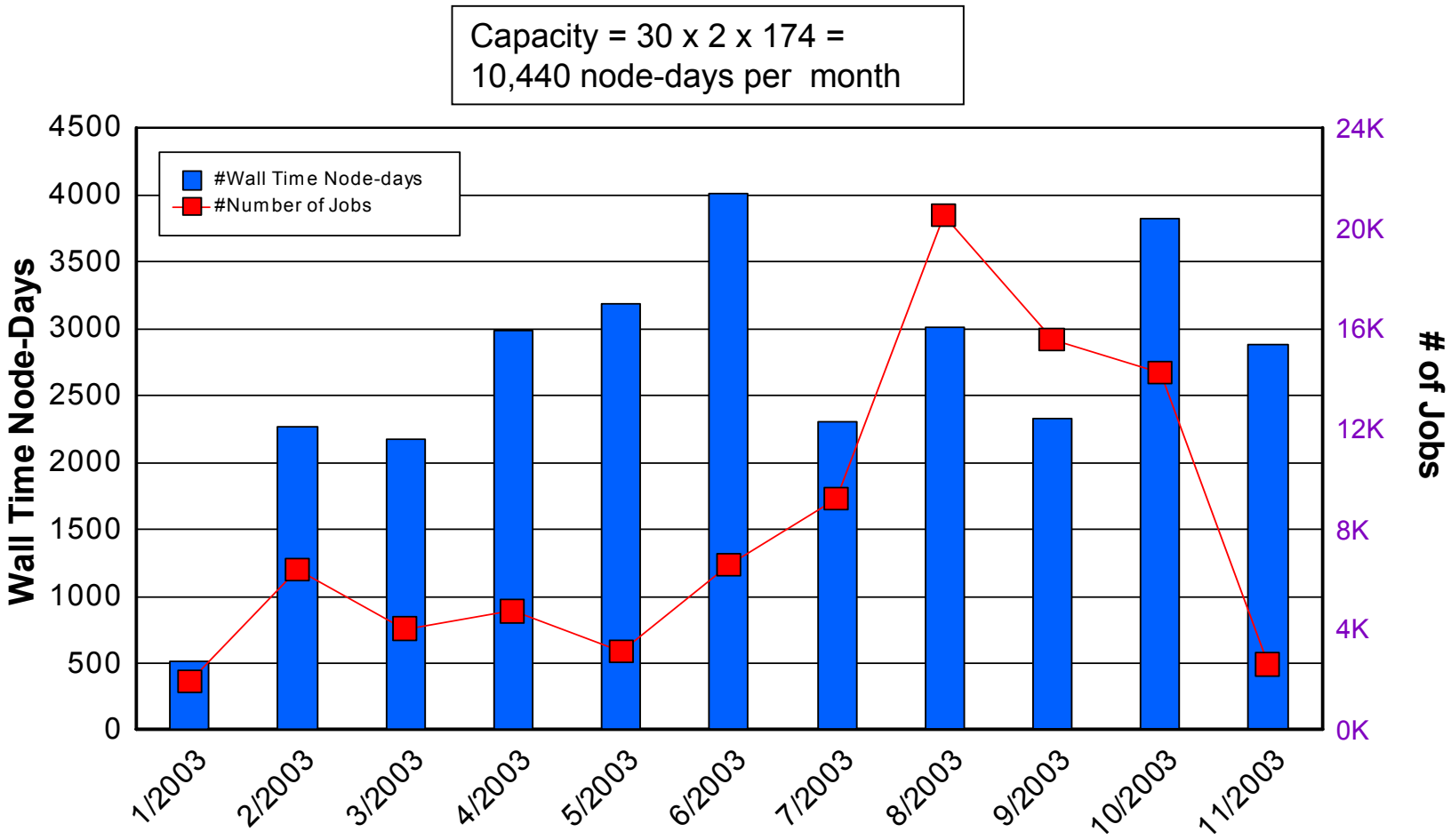
Job Accounting – FBSNG

- Used on Farms, CAF and CMS cluster.
- Accounting data maintained in flat files, not relational db.
- Provides job and process accounting but data currently rolls off after 1 week.
- Varying cpu speeds can be factored into normalized cpu usage metric because each process belonging to job generates usage log record containing node name.

Job Accounting Reports (pbs)

- SCIDAC Lattice QCD Cluster
- D0 CAB

Lattice QCD – SCIDAC Cluster



Which Metrics are Important?

- Cpu, I/O, storage, network, queue times …
- How or should we factor in or account for effects due to:
  - Power outages
  - Hardware repairs
  - Software maintenance

  FARMS Uptime
- Should we account for processor speed differences?
- Should we consider compiler effects, firmware performance options (hyperthreading, e.g.), etc. ?

Sampling vs. Actual Use

- Sampling methodology may simplify data collection processes and provide more control over amount of data collected.
- However, can we defend its use when negotiating user limits and priority changes?
  - Probably yes, if sampling interval small enough relative to workloads, e.g. 5 minutes.

Database Issues

- Some things to keep in mind:
  - Nodename key presumes no reuse of old nodenames for new processors.
  - Start and end of production service information must be included.
  - While design should be relatively straightforward, could impact ease of query/report design.
- Should we maintain db for operational management of our systems?

Systems Operational DB

- Would maintain table of all computers supported in Fermilab's datacenter
- Could include information such as nodename, system name, cputype, speed, support classification (24x7…), power, etc.
- Could be used by a variety of applications
  - Remedy/TelAlert
  - Usage Accounting
  - Computer room planning/management
  - Security
  - NGOP

Grid Work – Does It Help Us?

- A lot of work underway to define schemas and services to account for grid resource use:
  - Usage Record format is in its final draft
  - Resource Usage Service
  - Accounting Service
  - Economic Service
- Usage record data will be provided from underlying batch system such as LSF, Condor, pbs or fbsng.

Usage Record WG Charter Abstract

"In order for resources to be shared, sites must be able to exchange basic accounting and usage data in a common format. This working group proposes to define a common usage record based on those in current practice. The record format will be specific enough to facilitate information sharing among grid sites, yet general enough that the usage data can be used for a variety of purposes - traditional usage accounting, service usage monitoring, perfomance tuning, etc. This group will therefore be concentrating on collecting and disseminating resource consumption data. We will not be addressing how that data is to be collected by the resource sites, nor how it will be used by its recipients."

Resource Usage Service WG Charter

"To define a Resource Usage Service (RUS) for deployment within an OGSA hosting environment that will track resource usage (accounting in the traditional UNIX sense) and will not concern itself with payment for the use of the resource."

Make, Leverage or Buy?

- Build our own – estimates to be given
- Leverage NGOP, SETI …
- SNUPI (from SDSC)
- Open source – Ganglia, Already being used on many systems
- MonALISA, used by CMS
- PLATFORM – Workload Analytics

Next Steps

- Establish project and project leader
- Define requirements…some are already done.
- Investigate solutions
- Choose solutions
- Build project
- Staff it
- Run it
- Finish it.

Needed Staff Resources

- Following are the types of staff resources we will need to complete the project:
  - Systems integrator
  - Developer (if build our own)
  - DB analyst
  - Systems administrator

Locally Developed System – Project Outline

Collection and Storage of Accounting Information
Work Breakdown Structure

Draft by Igor Mandrichenko

1. Research
1.1. Requirements collection (1 week)
1.1.1. Data structures
1.1.2. Timing issues
1.1.3. Data representation and access
1.2. Decide on the technology and hardware requirements for storage and access to the data (1 week)
1.2.1. Database schema
1.2.2. Hardware
1.2.3. Network connectivity
1.2.4. Data access means
1.2.5. Long-term data storage
2. Development
2.1. Review existing sources of information (1 week)
2.1.1. Structure and accessibility of the data provided
2.1.2. State of data collection tools
2.2. Research other data sources (1 week)
2.2.1. Conceptual overview of the information they can provide
2.2.2. Available interfaces
2.3. Summarize requirement for necessary modifications of existing and new information sources (data missing, missing functionality in the interface, etc.) (1 week)
2.4. Parallel development
2.4.1. Develop set of tools needed to collect information from different sources and represent it in common format (1 month)
2.4.1.1. UNIX accounting
2.4.1.2. FBSNG
2.4.1.3. LSF
2.4.1.4. PBS
2.4.1.5. ?
2.4.2. Develop data storage solution (1 week)
2.4.3. Develop basic report generators (1 week)
2.4.4. Develop set of tools for historical data reduction and long-term storage (1 week)

Example of Costs for Locally Developed System

- Define metrics to be collected. – 1-2 weeks
- Decide build vs. buy – 2-4 FTE-weeks
- If build, see previous outline for development effort needed. ~ 3-6 FTE-months.
- Deploy system – sys admins for each system plus accounting systems manager effort.
- Collect data – system does this!
- Design reports – initially 2-4 weeks with small effort when new views/reports are desired.
- Maintain tables and manage data collection system. Ongoing 15-20% FTE (based on current effort by Terry Jones, et al.)

Ongoing Operational and Support Costs

- Whichever solution is chosen, there will be ongoing operational and support costs that may include some or all of the following:
  - Software management/administration including bug fixes, configuration changes, new release installation, etc. (FTE's)
  - Report and query design (FTE's)
  - Software license costs. (M&S)

Batch Scheduling – Related Work To Do

- Not in scope of this project but important and strongly related.
- Examine current schedulers in use and determine whether or not they meet our needs for resource management and accounting
  - LSF, PBS, Condor, fbsng
- Identify and recommend extensions, configuration changes, etc. needed to meet resource accounting/mgt requirements.

Monitoring Links

- Farms:
  - http://fnpcg.fnal.gov/ganglia-webfrontend
  - http://www-oss.fnal.gov/scs/farms/integrated_uptime.html
- CAF:
  - http://fcdfmon2.fnal.gov
  - http://cdfcaf.fnal.gov/cgi-bin/caf/admin/cafmon
- D0 CAB:
  - http://d0om/d0admin/ganglia/