Job-Based Accounting for UNICOS/mp

Jim Glidewell
Network, HPC, and Access Support
james.glidewell@boeing.com
Topics

- Background and Goals
- Why accounting?
- Available accounting resources
- Design of a Job-based Accounting System for UNICOS/mp
- Other accounting tools & data sources
Site Overview

- Primary HPC Datacenter supporting Boeing Commercial Aircraft division
- Mix of Cray, SGI, and Linux cluster systems
- Long time users of CSA (Cray/Comprehensive System Accounting)
Why Accounting?

- Cost recovery
- Usage profiling & trend analysis
  - Application usage
  - Overall demand trends
  - Hardware acquisition planning
- Security monitoring
- Performance tracking & tuning to maximize overall system performance, as well as looking for troublesome applications
- User Benefits
  - Individual application tuning
  - Selection of appropriate computing platform
Resources Available

- UNICOS/mp “job container”
- Project accounting and “newacct” command
- Enhanced process accounting records, including
  - Job, User, and Project ID’s
  - Detailed resource information
- Application accounting records
  - Included in process accounting (“pacct”) files
  - Further detail on application resources and placement
- PBS Pro accounting logs
- UNICOS/mp job table
Accounting Goals

- Accuracy
- Repeatability
- User visibility
- Support of multiple projects per user
- Meaningful reporting granularity
- Reliability of daily processes
- Reporting based on user, project
- Rolled up based on 24 hour period
- No “end-of-job” reporting
- Minimal solution
Basic Job-based Accounting

Previous Day's Process
Accounting Files
/var/adm/Spacct*.MMDD

End of Day Accounting

Daily Accounting Feed
Based on Job_ID, User, Project

Copyright © 2006 Boeing. All rights reserved.
When is a Job Complete?

Previous Day’s Process
Accounting Files
/var/adm/Spacct*.MMDD

End of Day Accounting

Daily Accounting Feed
Based on Job_ID, User, Project

UNICOS /mp Job Table
(list of currently running jobs)

Copyright © 2006 Boeing. All rights reserved.
Dealing with Jobs that Cross Accounting Boundaries

Previous Day's Process Accounting Files
/var/adm/Spacct*.MMDD

Daily Accounting Feed Based on
Job_ID, User, Project

End of Day Accounting

Job Recycle File

Copyright © 2006 Boeing. All rights reserved.
Additional Job Data from PBS

PBS Accounting Log
/var/spool/PBS/server_priv/accounting/...

PBS Data Extractor

Previous PBS Job Info File

PBS Job Info File
Putting It All Together…

Previous Day’s Process
Accounting Files
/var/adm/Spacct*.MMDD

PBS Job Info File

PBS Data Extractor

PBS Accounting Log

UNICOS/mp Job Table
(list of currently running jobs)

Daily Accounting Feed
Based on Job _ID, User, Project

End of Day Accounting

Job Recycle File
User Job Cost Reporting – “jobcost”

Current Process
Accounting Files
/var/adm/pacct *

“jobcost” command

Job Cost Report

PBS Server
PBS Job Table

Job Recycle File
Output from “jobcost” Command

% jobcost

  X1 Job Cost Report

  Job ID :       238433
  User :       xxxxxxx
  CWA :       xxxxxx
  Project :
  Start date :   10/05/2005
  Start time :     11:53:00
  End date :   10/05/2005
  End time :     12:03:04
  User CPU :        0.000 MSP-seconds
  System CPU :        0.000 MSP-seconds
  User CPU :        2.200 SSP-seconds
  System CPU :        2.933 SSP-seconds
  Memory :        0.011 gigabyte-seconds
  I/O :       29.442 megabytes
  Processes :           17

<table>
<thead>
<tr>
<th>Units</th>
<th>CRUs</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.283</td>
<td>73.466</td>
</tr>
<tr>
<td>Memory</td>
<td>0.011</td>
<td>0.021</td>
</tr>
<tr>
<td>Total Raw CRUs</td>
<td>73.487</td>
<td></td>
</tr>
<tr>
<td>Factored CRUs</td>
<td>15.653</td>
<td></td>
</tr>
</tbody>
</table>
Usage: jobcost [-t | -i]

jobcost
- reports all resource usage by job

jobcost -i
- reports resource usage since last jobcost command

jobcost -t
- no report – marks start of segment for jobcost -i
Job-Based Accounting – A Summary

- Two major components
  - “End-of-job cost” report for users
  - “End-of-day” accounting for
    - Finance & billing
    - User management reports
    - Resource and application tracking & trend analysis
- Relies heavily on Cray’s process accounting, and particularly the “job” container
- Significant improvement over usage reporting based on 24 hour period
- Robust and reliable
- Well received by customers
Other Accounting Options

Sites that wish to examine process, application, or job resource usage have a number of options:

- Process records in /var/adm/pacct*
- Application records in /var/adm/pacct*
- Output of “acctcom” command (process data)
- Output of “acctcom –A” command (application data)
- PBS Pro accounting logs (job-based data, limited resource detail)
Enabling Process Accounting

- To enable process accounting: “chkconfig acct on”
  - Enables cron entries in root and adm, including `/usr/lib/acct/runacct`
- `runacct` script leaves process accounting files in `/var/adm` with names of the form `Spacct*.MMDD`
- `/usr/lib/acct/runacct.local` is provided as a “site hook” to allow sites to post-process and cleanup or archive the `Spacct*` files
- Defining a policy on process accounting retention is essential
Structure of the Process Accounting Files

- Process Information
  - Header
  - Common

- Application Information
  - Header
  - Common

- Application
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process
  - Process

Copyright © 2006 Boeing. All rights reserved.
# Process Accounting Fields Available from the “acctcom” Command

<table>
<thead>
<tr>
<th>Accounting field</th>
<th>“acctcom” option</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ah_flag</td>
<td>-f</td>
<td>Process flags</td>
</tr>
<tr>
<td>acc_uid</td>
<td>*</td>
<td>User</td>
</tr>
<tr>
<td>acc_status</td>
<td>-f</td>
<td>Exit status</td>
</tr>
<tr>
<td>acc_btime</td>
<td>*</td>
<td>Start time</td>
</tr>
<tr>
<td>acc_etime</td>
<td>*</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>acc_comm</td>
<td>*</td>
<td>Command name</td>
</tr>
<tr>
<td>ac_gid</td>
<td></td>
<td>Group</td>
</tr>
<tr>
<td>ac_acid</td>
<td>-p</td>
<td>Account ID (project)</td>
</tr>
<tr>
<td>ac_tty</td>
<td>*</td>
<td>Terminal</td>
</tr>
<tr>
<td>ac_pid</td>
<td>-p</td>
<td>Process ID</td>
</tr>
<tr>
<td>ac_apid</td>
<td>-p</td>
<td>Application ID</td>
</tr>
<tr>
<td>ac_sid</td>
<td>-p</td>
<td>Session ID</td>
</tr>
<tr>
<td>ac_utime</td>
<td>-t</td>
<td>User CPU time</td>
</tr>
<tr>
<td>ac_stime</td>
<td>-t</td>
<td>System CPU time</td>
</tr>
<tr>
<td>ac_mem</td>
<td>-k</td>
<td>Memory integral</td>
</tr>
<tr>
<td>ac_himem</td>
<td>*</td>
<td>Memory hi-water</td>
</tr>
<tr>
<td>ac_io</td>
<td>-i</td>
<td>Characters transferred</td>
</tr>
<tr>
<td>ac_RW</td>
<td>-i</td>
<td>Blocks read/written</td>
</tr>
</tbody>
</table>
# Application Accounting Fields Available from the “acctcom -A” Command

<table>
<thead>
<tr>
<th>Accounting field</th>
<th>“acctcom –A” option</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>acc_uid</td>
<td>*</td>
<td>User</td>
</tr>
<tr>
<td>acc_status</td>
<td></td>
<td>Exit status</td>
</tr>
<tr>
<td>acc_btime</td>
<td>*</td>
<td>Start time</td>
</tr>
<tr>
<td>acc_etime</td>
<td>*</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>acc_comm</td>
<td>*</td>
<td>Command name</td>
</tr>
<tr>
<td>acap_apid</td>
<td>*</td>
<td>Application ID</td>
</tr>
<tr>
<td>acap_sid</td>
<td></td>
<td>Session ID</td>
</tr>
<tr>
<td>acap_ltime</td>
<td>*</td>
<td>Launch time</td>
</tr>
<tr>
<td>acap_flags</td>
<td>*</td>
<td>Application flags</td>
</tr>
<tr>
<td>acap_width</td>
<td>*</td>
<td>Application width</td>
</tr>
<tr>
<td>acap_depth</td>
<td>*</td>
<td>Application depth</td>
</tr>
<tr>
<td>acap_ctime</td>
<td>*</td>
<td>Connect time</td>
</tr>
<tr>
<td>acap_acid</td>
<td></td>
<td>Account ID</td>
</tr>
<tr>
<td>acap_txt_pgsiz</td>
<td>-L</td>
<td>Text page size</td>
</tr>
<tr>
<td>acap_oth_pgsiz</td>
<td>-L</td>
<td>Non-text page size</td>
</tr>
<tr>
<td>acap_place_cnt</td>
<td>-L</td>
<td>Placement count</td>
</tr>
</tbody>
</table>
Making use of process accounting

- The “acctcom” command greatly simplifies use of process and/or application accounting

- Sites wishing to use project accounting in conjunction with “acctcom –A” will need to use the application_ID to tie the data together

- Reading the binary pacct files is slightly more efficient, but any site creating new accounting processes should probably look carefully at “acctcom” first

- When reading the binary pacct files, the application record usually, but not always, directly follows the associated process record
Summary

- We have created a local job-based reporting system for use by end users and downstream processes.
- Cray’s inclusion of a job id in process and application accounting records allows a site to tie together resource and job information from multiple sources.
- The enhanced process and application records are relatively easy to process, and provide sites with valuable information.
- Job-based accounting:
  - Is a good tradeoff of detail versus data volume.
  - Is easier for users to understand and utilize.
  - Allows more options for data center utilization tracking and analysis.
  - Works well for our site.
CUG 2007
NEW FRONTIERS

SEATTLE • April 30-MAY 3, 2007