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
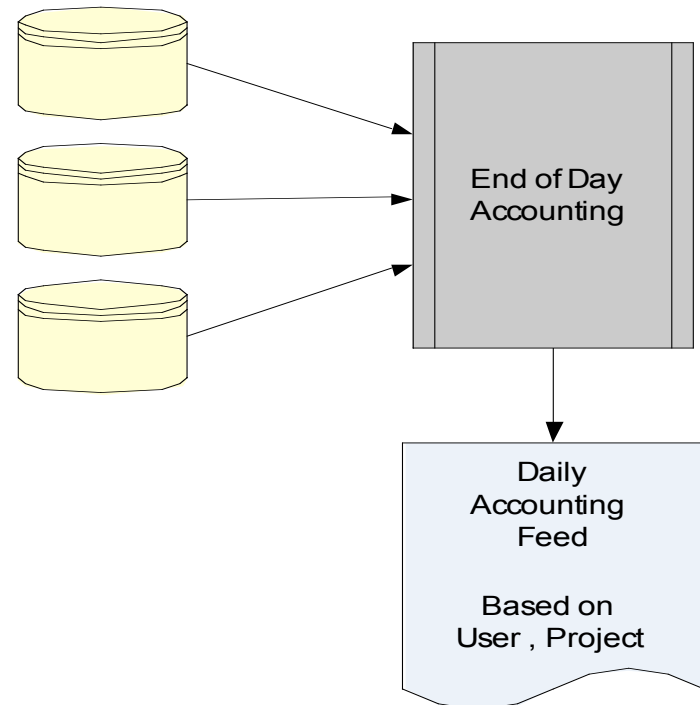
Accounting Goals

- Accuracy

- Repeatability

- User visibility

- Support of multiple projects per user

- Meaningful reporting granularity

- Reliability of daily processes

Initial Accounting Implementation

- Reporting based on user, project

- Rolled up based on 24 hour period

- No "end-of-job" reporting

- Minimal solution

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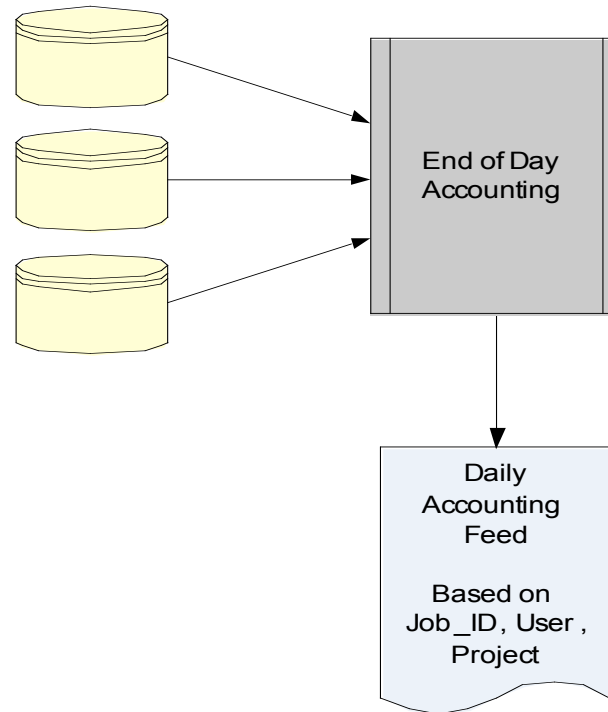Accounting Files
/var/adm /Spacct *. MMDD

End of Day
Accounting

Daily
Accounting
Feed

Based on
User , Project

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

When is a Job Complete?

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

End of Day
Accounting

UNICOS /mp
Job Table

(list of
currently
running jobs)

Daily
Accounting
Feed

Based on
Job _ID, User ,
Project

Dealing with Jobs that Cross Accounting Boundaries

Boeing Technology | **Information Technology**

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

End of Day
Accounting

Job Recycle File

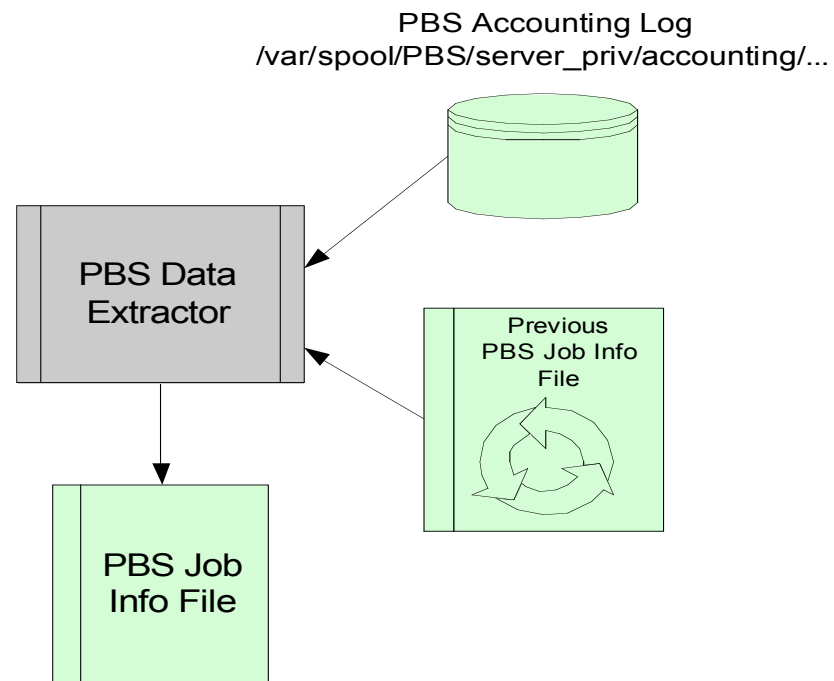Daily
Accounting
Feed

Based on
Job_ID, User,
Project
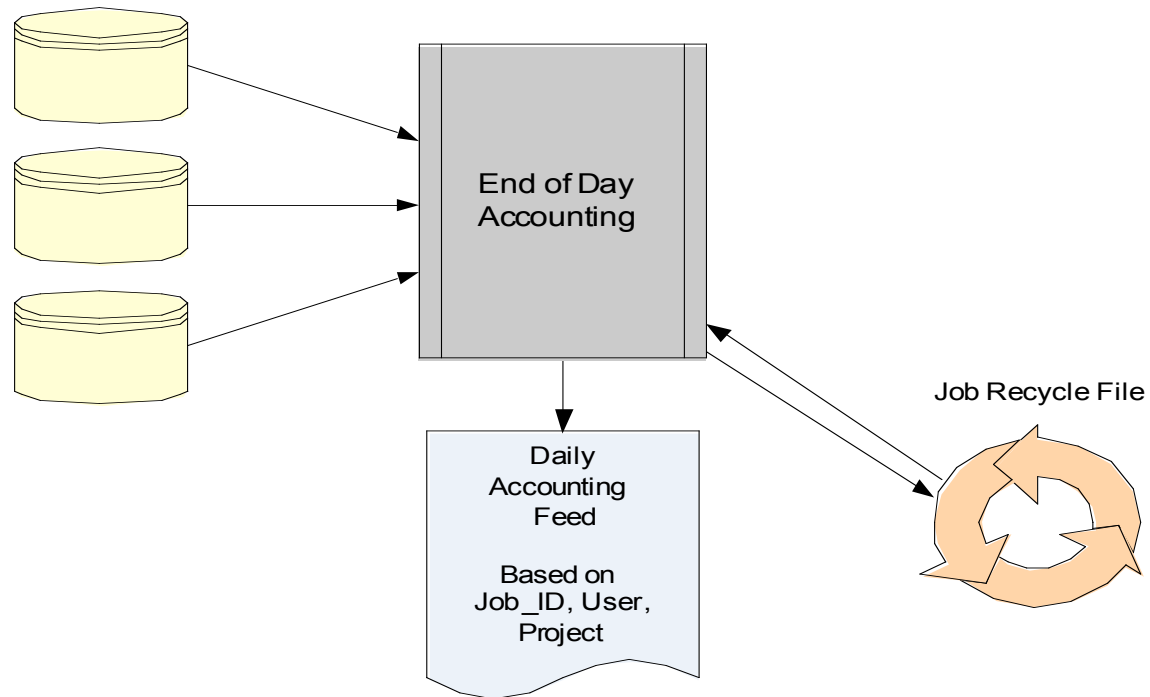
Additional Job Data from PBS

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…

PBS Accounting Log

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

PBS Job
Info File

PBS Data
Extractor

End of Day
Accounting

UNICOS/mp
Job Table

(list of currently
running jobs   )

Job Recycle File

Daily
Accounting
Feed

Based on
Job _ID, User,
Project
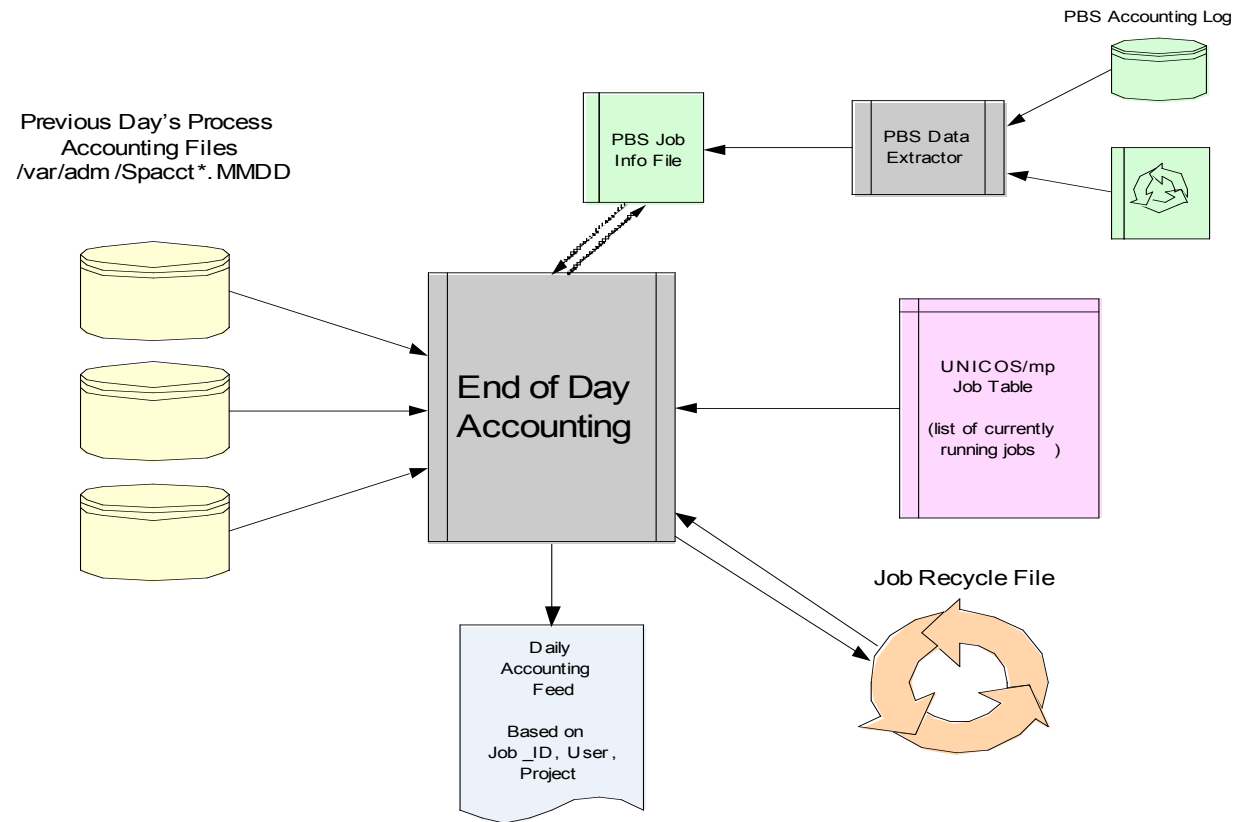
User Job Cost Reporting – "jobcost"

Current Process
Accounting Files
/var /adm /pacct *

PBS Server

PBS Job Table

"jobcost"
command

Job Cost Report

Job Recycle File
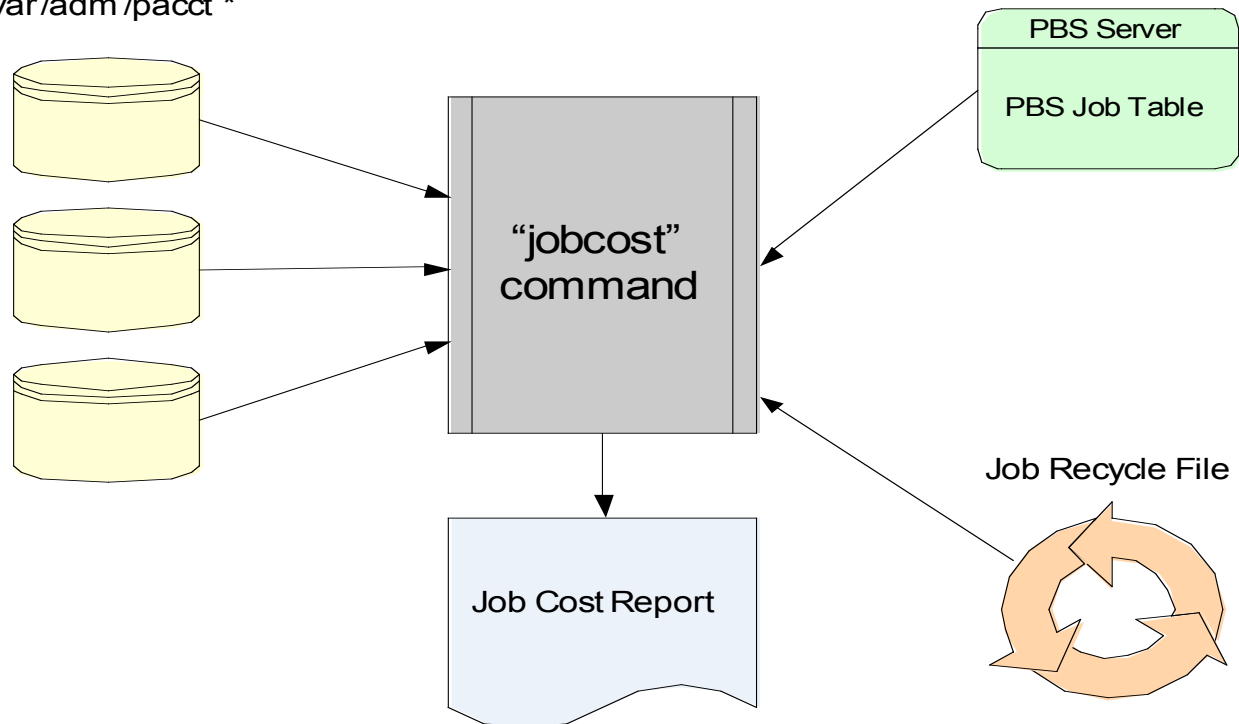
Output from "jobcost" Command

```
% 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


                Units         CRUs      Percent
          CPU   1.283        73.466      100.0%
       Memory   0.011         0.021        0.0%
Total Raw CRUs                73.487

Factored CRUs                 15.653
```

"jobcost" Command Options

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"

  - /etc/init.d/acct -> /usr/lib/acct/startup -> /usr/lib/acct/turnacct -> /usr/lib/acct/accton

  - 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
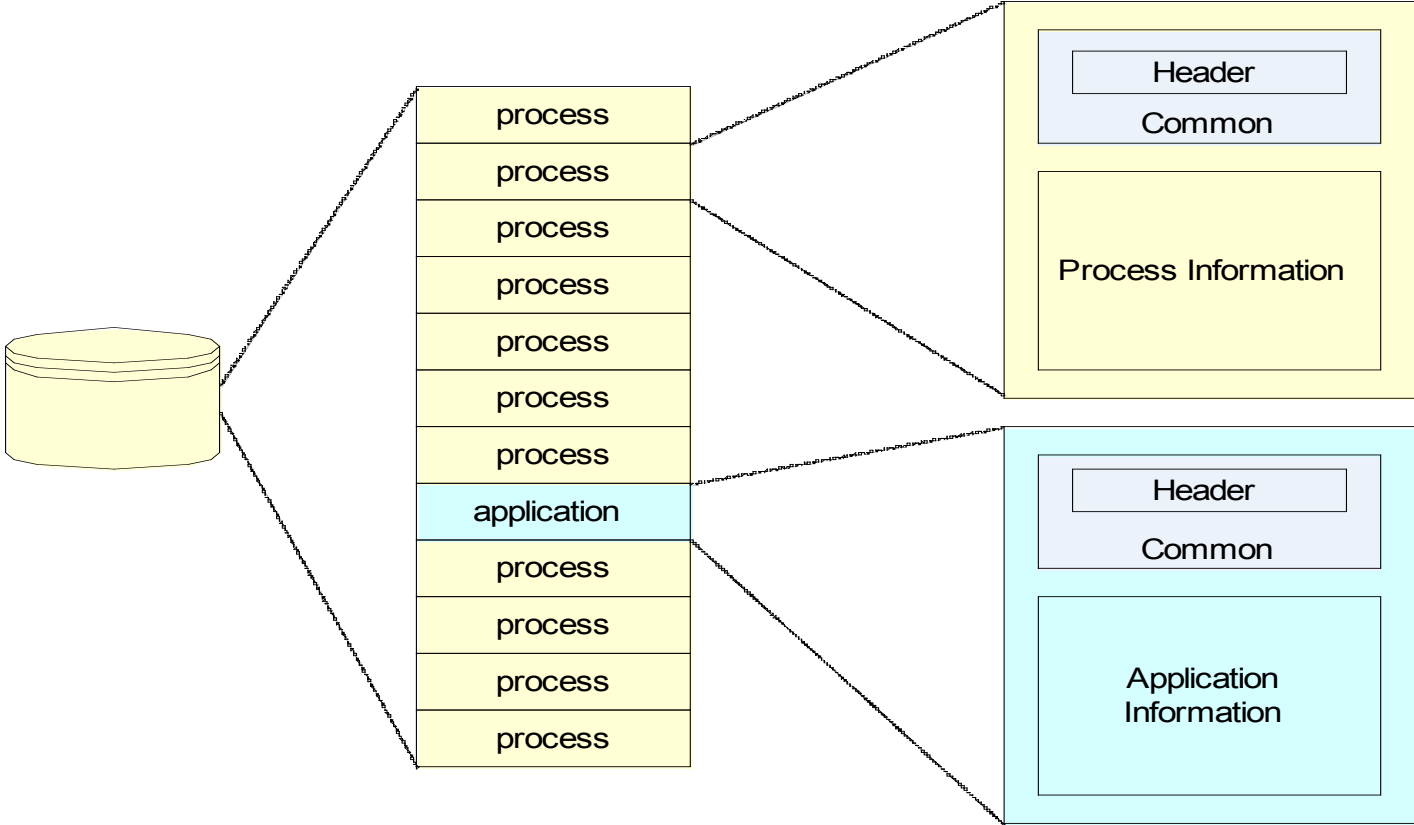
Process Accounting Fields Available from the "acctcom" Command

Boeing Technology | **Information Technology**

Accounting field	"acctcom" option	Definition
ah_flag	-f	Process flags
acc_uid	*	User
acc_status	-f	Exit status
acc_btime	*	Start time
acc_etime	*	Elapsed time
acc_comm	*	Command name
ac_gid		Group
ac_acid	-p	Account ID (project)
ac_tty	*	Terminal
ac_pid	-p	Process ID
ac_apid	-p	Application ID
ac_sid	-p	Session ID
ac_utime	-t	User CPU time
ac_stime	-t	System CPU time
ac_mem	-k	Memory integral
ac_himem	*	Memory hi-water
ac_io	-i	Characters transferred
ac_rw	-i	Blocks read/written

Application Accounting Fields Available from the "acctcom -A" Command

Accounting field	"acctcom –A" option	Definition
acc_uid	*	User
acc_status		Exit status
acc_btime	*	Start time
acc_etime	*	Elapsed time
acc_comm	*	Command name
acap_apid	*	Application ID
acap_sid		Session ID
acap_ltime	*	Launch time
acap_flags	*	Application flags
acap_width	*	Application width
acap_depth	*	Application depth
acap_ctime	*	Connect time
acap_acid		Account ID
acap_txt_pgsiz	-L	Text page size
acap_oth_pgsiz	-L	Non-text page size
acap_place_cnt	-L	Placement count

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