

Franklin Job Completion Analysis

Yun (Helen) He, Hwa-Chun Wendy Lin, and
Woo-Sun Yang

National Energy Research Scientific Computing Center

CUG 2010, May 24-27, Edinburgh



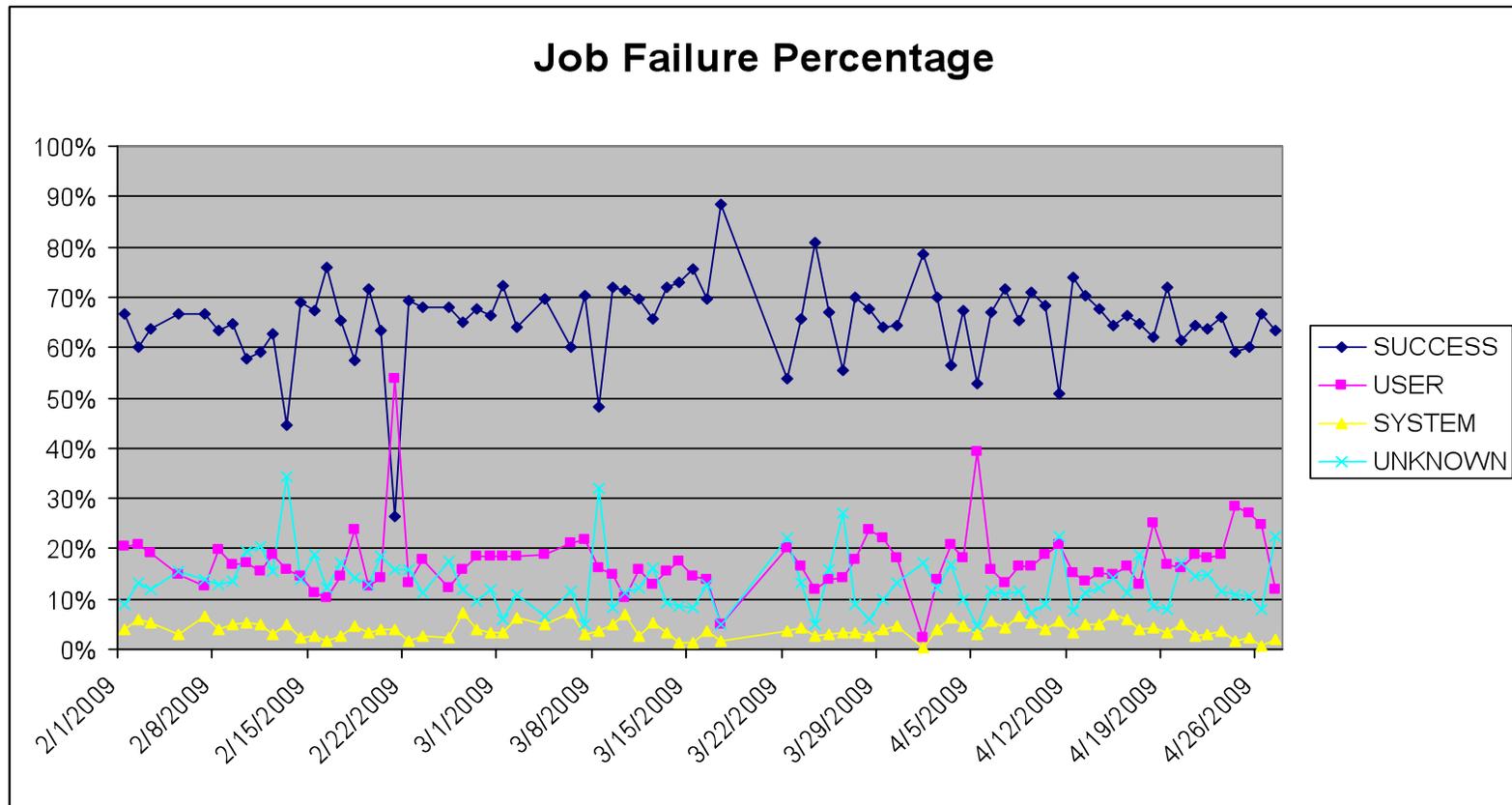


Our Goal

- **Identify and track system issues that cause user jobs to fail. Work with Cray to get them fixed.**
- **Job completion report, i.e. how many jobs ran successfully and how many jobs failed for what reasons.**

Our Data

Job Completion rate = Success + User related failures



User Related Job Failures

- **Application Errors: APEXIT, APNOENT, APRESOURCE, APWRAP**
- **Runtime Errors: CCERUNTIME, PATHRUNTIME**
- **MPI Errors: MPIABORT, MPIENV, MPIFATAL, MPIIO**
- **IO Errors: PGFIO**
- **PTL Errors: PTLUSER**
- **Signal: SIGSEGV, SIGTERM,**
- **Misc: XBIGOUT, DISKQUOTA, OOM, NIDTERM**

System Related Job Failures

- **LUSTREIO: input/output error**
- **NODEFAIL**
- **PTLSYS: PTL_NAL_FAILED, PTL_PT_NO_ENTRY**
- **SHMEMATOMIC**
- **IDENTERM: identifier removed**
- **JOBSTART: MOM could not start job**
- **JOBPROLOG: prolog script error**
- **JOBQUEUE: usually after SWO**
- **User or System related Job Failures:**
 - **JOBCOPY, JOBWALLTIME, NOBARRIER**

System Issues

- **System wide outages**
 - Lustre node crashes
 - Link failures, HSN failures
 - Power issues ...
- **MOM node crashes**
 - Warmbooting a MOM node prevents a system crash, and saves jobs running on other MOM nodes.
 - Login/MOM node separation helps a lot too. A login node crash is not causing batch job failures any more.
- **LDAP lookup failures**
- **Hardware failures**



System Issues (cont'd)

- “sick” nodes left by previous job
- Hang applications
- aprun awaiting barriers
- /tmp or /var filled
- Programming environment related issues
- Portals bug related issues
- Portals related system issues
- Lustre IO related issues
- DVS Server failures

LDAP Lookup Failures

- **LDAP: Lightweight Directory Access Protocol**
- ***nscd*: Name Service Cache Daemon**
- **Failure mode 1:**
 - NSCD dies, users could not login
- **Failure mode 2:**
 - JOBSTART
- **Failure mode 3:**
 - JOBCOPY
- **Failure mode 4:**
 - JOBPROLOG

LDAP Lookup Failures

- **Failure mode 5:**
 - “identifier removed” error while accessing files
 - Happens interactively or in batch job
 - Traced to LDAP time out with `I_getgroups` failures.
 - Bug filed for `I_getgroups` to use `nscd` group caching
 - Initial upgrade of `nscd` daemon did not help
 - `nscd` configuration change in the setting of the shared attributes for user and group lookups improved the situation substantially
 - Plan to test with new `nss_ldap` and `nscd`.

Hung Applications

- **Most hung jobs hung before aprun starts.**
- **Waste valuable allocation time. Impact user productivity.**
- **NOBARRIER error:**
 - **job killed: walltime xxxx exceeded limit xxxx**
aprun: Caught signal Terminated awaiting barrier, sending to apid xxxxxx
 - **MPI or SHMEM applications send barrier message to aprun. Working with Cray to set timeout for aprun (via aprun wrapper with an env variable) waiting for the barrier.**

Hung Applications (cont'd)

- **Possible cause: Portals issues?**
 - Console log message: “[c5-4c1s0n2]Lustre Error 31373:0: mdc_locks.c:586:mdc_enqueue())ldlm_cli_enqueue: -4”.
 - Traced to a portals issue related to “transmit credit accounting”. Applied patch.
- **Possible cause: Lustre issues?**
 - Console log message: “The mds_connect operation failed with -16”
 - Changed the Lustre “group_acquire_expire” setting for MDS from 15 to 60, then to 240 seconds.



Hung Applications (cont'd)

- **Possible cause: “bad” nodes left by previous jobs?**
 - OOM
 - /tmp memory usage
 - slab memory usage
 - orphan process
 - node segfault
- **Node Health Checking**
 - Improvement in OS 2.1 and 2.2 helped to better identify “sick” nodes and set them to “admindown”.
 - Detecting “bad” nodes with insufficient useable memory is on our wish list for NHC.

Case Study

- User “aaaa” ran a total of 109 jobs on 1/11-2/11/2009.
- **15 succeeded, 94 failed.**
- 59 jobs failed due to the a user environment issue caused by inconsistency between xtpe-quadcore and xt-asyncpe module installation. The system problem has been fixed. System error.
- 6 job failures were due to system crashes. System error.
- 2 job failures were due to transient ALPS error. System error.
- 1 job failure was due to TCP socket connection time out. System error.
- 3 job failure was due to “identifer removed” error. System error.
- 2 job failures were due to “PGFIO” issue. System error.
- 2 job failure was due to node failures. System error.

Case Study (cont'd)

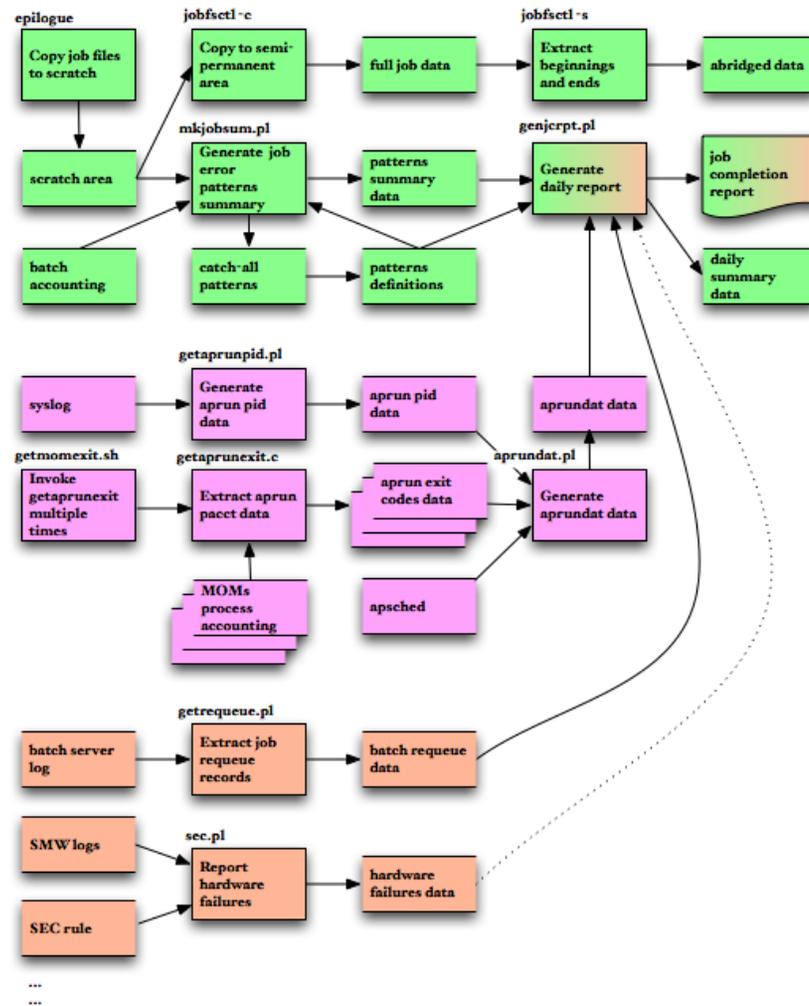
- **5 job failures were due to user executable files not exist. User error.**
- **4 job failures were due to user running from a wrong repo number. User error.**
- **9 job failures were due to various errors in codes: seg fault, floating point exception. User error.**
- **1 job failure was run out of wall clock time. Possible user error.**
- **Total of 75 jobs failed due to system error, 19 jobs failed due to user error.**



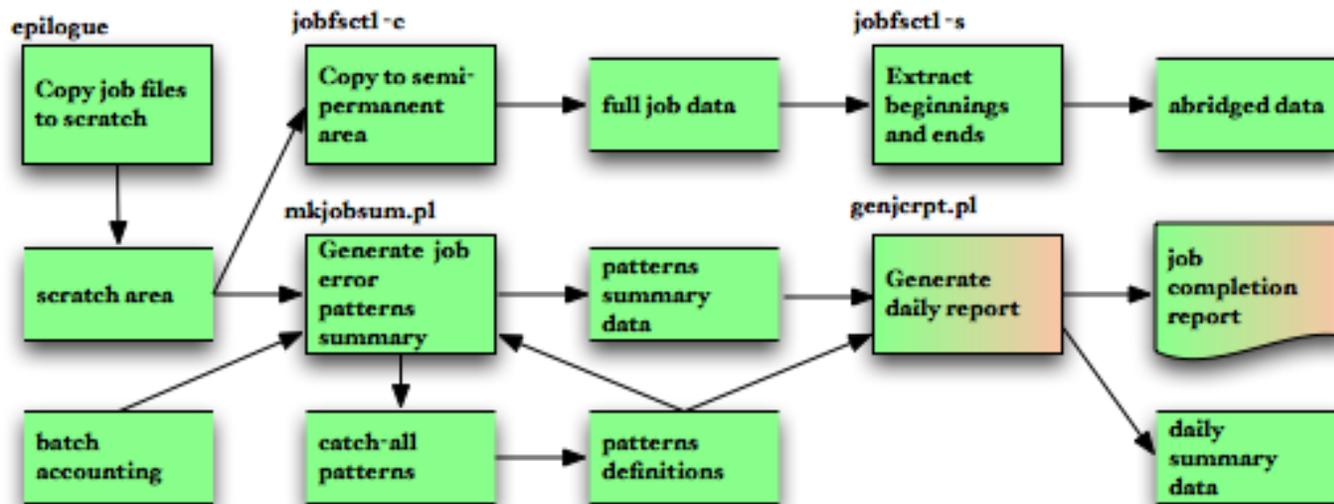
Job Completion Report Generation

- **Previous report generation**
 - Analyze job stderr/stdout in batch epilogue at the end of a job
 - Generate daily summary from job data collected
- **New report generation**
 - Approach: Save job files in epilogue; post-process all at once
 - Design goal: maximize accuracy in deciding whether a job completed successfully, and for jobs that failed, whether the cause was user or system originated.
- **Three phases in implementation**
 - Based on error message patterns and batch job exit status
 - Supplement with aprun exit codes
 - Supplement with system log messages

Report Generation Implementation Phases Overview



Implementation Phase I: Components





Implementation Phase I: Players

- **Epilogue saves user job files: script, stderr, stdout**
- **Batch accounting log provides job IDs and exit statuses**
- **Jobcompinc.pl defines attributes for known patterns: text strings, labels, causes (user, system, or user_or_system)**
- **Mkjobsum.pl finds all known patterns shown in stderr/stdout, write out their labels**
- **Genjcrpt.pl generates daily report, summary**



Error Message Patterns: Sources

- **USG tickets and archived job files**
 - Combine and generalize messages
- **Documentation on message prefixes**
 - CCE, PathScale runtime errors
- **Visual inspection of messages caught by “catch-all” patterns such as “aprun: Apid”, “[NID \d+]”**
 - aprun: Apid 2277067 RCA ec_node_failed event received for nid 2943
 - aprun: Apid 2219443 close of the compute node connection before app startup barrier (local fd 3, port 25763)
 - [NID 05738] Apid 2292125: cannot execute: exit(111) fork error:



Error Message Patterns and Labels

- **Appendix A**
- **Label = a group of similar patterns**
- **Hierarchy of labels**
 - Labels weigh differently
 - Highest ranked: **APDVS, APCONNECT, APWRAP, APRESOURCE, FILEIO, etc**
 - Low ranked: **NIDTERM, MPIABORT, etc**



Derived Labels: Exit_status from Batch Accounting

- **-2: JOBSTART**
 - Authentication error in MOM
- **-1: JOBPROLOG**
 - Prologue error (repo check)
- **143, 271: SIGTERM**
- **139, 267: SIGSEGV**
- **(More to be identified)**
- **Other non-zero: JOBEXIT**
- **(See flow chart in paper)**



Sample First Phase Report

Exit Status	Count	Percent	Cause
APDVS	1	0.0	S
APEXEC	2	0.1	U
APNOENT	36	1.7	U
APRESOURCE	12	0.6	U
CCERUNTIME	1	0.0	U
JOBEXIT	122	5.9	U
JOBPROLOG	4	0.2	US
JOBSTART	3	0.1	S
JOBWALLTIME	240	11.5	US
MPIABORT	3	0.1	U
MPIENV	4	0.2	U
MPIFATAL	7	0.3	U
NIDTERM	128	6.1	U
NOCMD	20	1.0	U
NODEFAIL	1	0.0	S
NOENT	48	2.3	U
NOKNOWNERR	1287	61.8	N/A
OOM	11	0.5	U
PATHRUNTIME	1	0.0	U
PERMISSION	1	0.0	U
PGFIO	41	2.0	U
SHAREDLIB	1	0.0	U
SIGSEGV	25	1.2	U
SIGTERM	57	2.7	U
STALENFS	27	1.3	S
XBIGOUT	1	0.0	U
Total	2084		





Sample First Phase Report (cont.)

Job Failure Statistics

Type	Count	Percent
No known err	1287	61.8
System	32	1.5
User/system	244	11.7
User	521	25.0

High Counts for Category+User

Category	User	Count
APNOENT	userabc	10
APRESOURCE	userb	9
JOBEXIT	usercd	55
JOBWALLTIME	userdef	31
JOBWALLTIME	useref	19
NIDTERM	userfg	18
NIDTERM	userg	61
NOCMD	userhi	9
NOCMD	userjkl	8
NOENT	userklm	11
OOM	usermno	6
PGFIO	usernop	14
SIGSEGV	usero	8
SIGTERM	userp	5

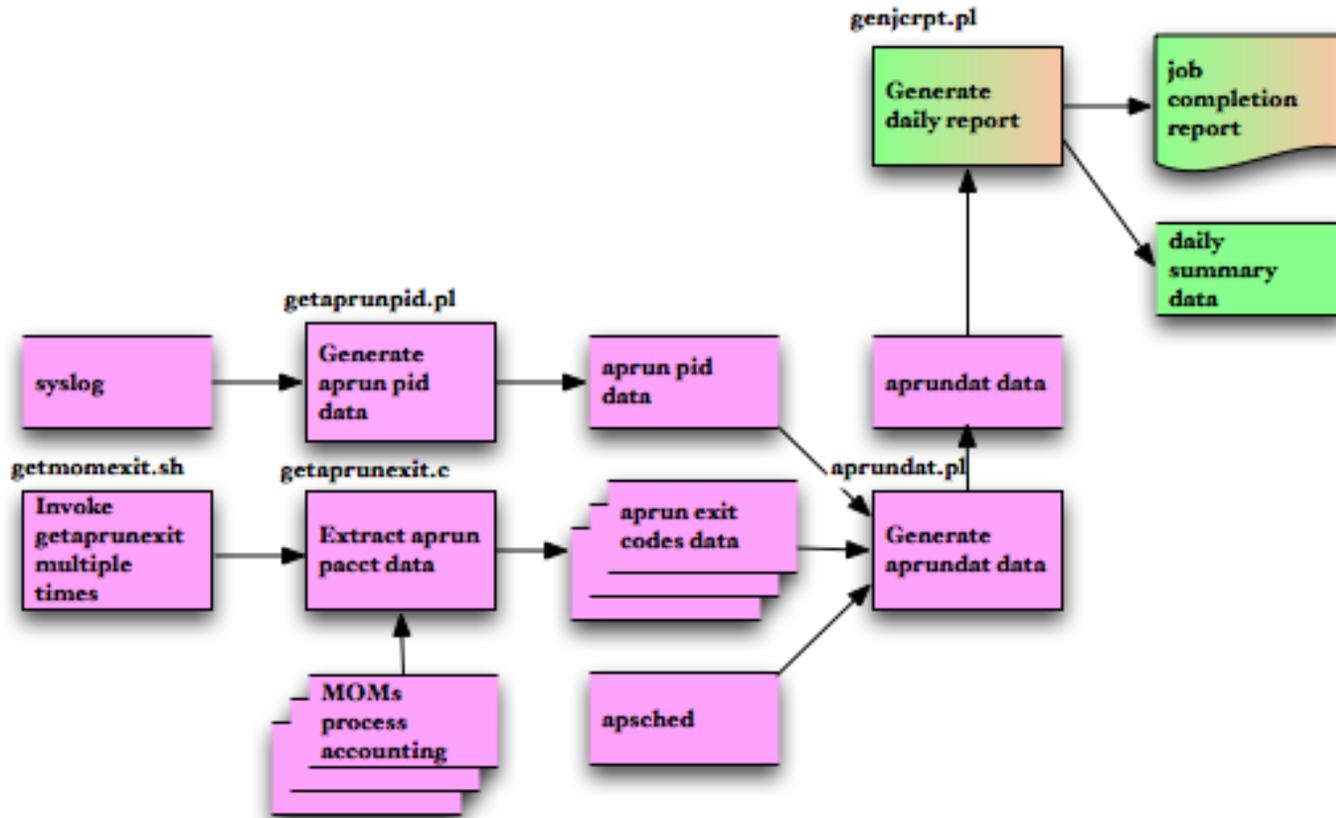
...



Error Message Patterns: Issues

- **Missing patterns**
 - Catch-all patterns
- **Label hierarchy**
 - Labels weigh differently
- **Multiple apruns**
 - Labels of same weight
- **Aprun stderr redirection**
 - Bug to have aprun save stderr
 - Could affect ordering of stdout/stderr messages if merged
 - Would not handle multiple apruns with mixed success and failure

Implementation Phase II: Components





Implementation Phase II: Players

- **Getaprunpid.pl** extracts aprun pid data from syslog
- **Getaprunexit.pl** extracts “aprun*” exit codes from process accounting log
- **Enhanced aprundat.pl** generates aprun info, including exit codes, using apsched log, and the above two output

Application Exit Codes

- **Output from the original aprundat.pl**

Job ID	...	User	Command/node list
6467851	...	abcdef	hostname/9031-9046

- **Output from the enhance aprundat.pl**

Job ID	...	User	exitcode	Command/node list
6467851	...	abcdef	0x0000	"aprun -n 64 hostname "/9031-9046

- **Aprun exit code == application exit code?**

- Not really
- Issue: aprun borrows exit codes 1, 0x80?? - 0x8f??
- Bug 75252, scheduled: June/July time frame
- Application exit codes/signals (up to 4) to be on aphys records in syslog
- Process accounting no longer needed

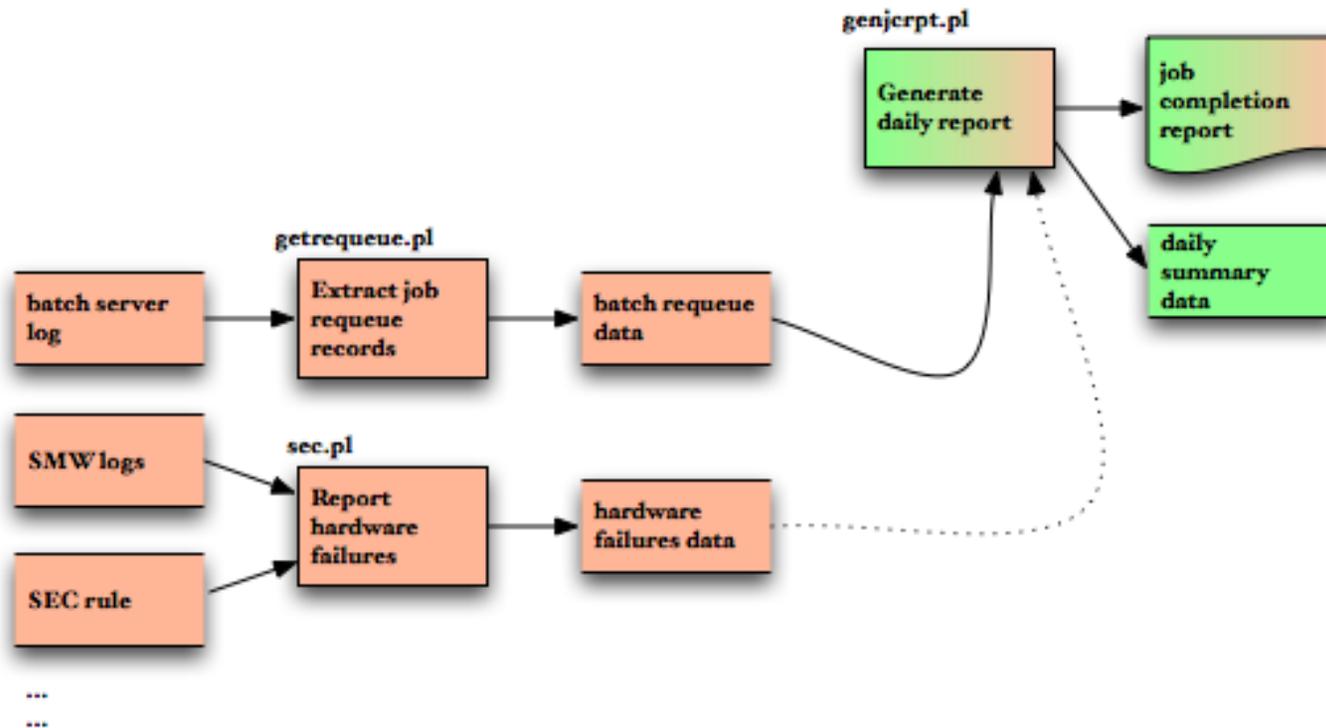
- **Good enough?**

- Some labels still ambiguous

Cause: User or System?

- **Ambiguous Errors**
 - **JOBWALLTIME**
 - Batch server restart (SWO)
 - Application hang
 - User checkpoint
 - User code loop
 - User job flow design
 - **JOBCOPY**
 - Filesystem issue
 - Directory non-existent
 - **NOBARRIER**
 - Should apply to MPT implementation only
 - Bug 755008
- **Tie-breaker**
 - **System logs**

Implementation Phase III





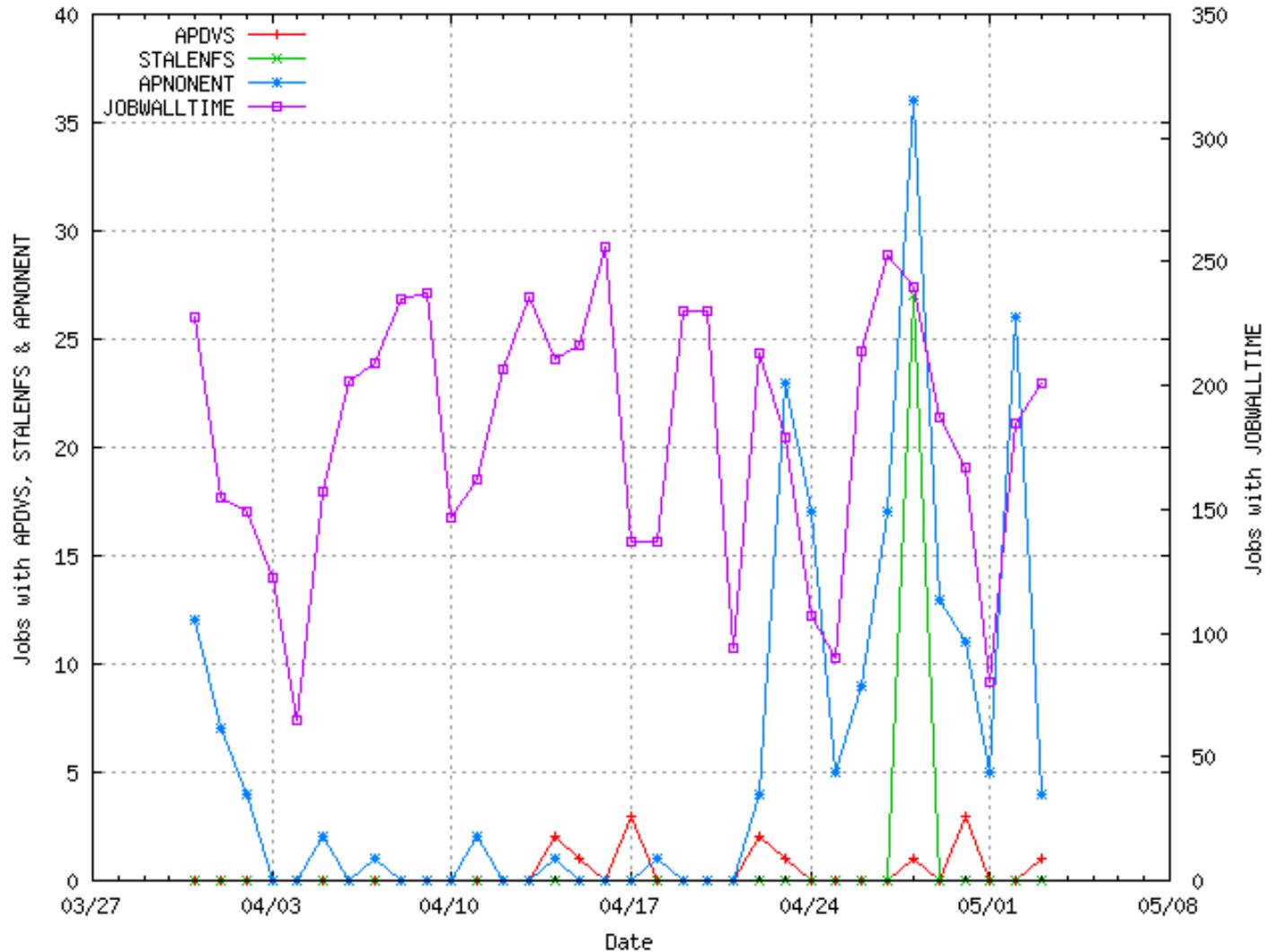
Implementation Phase III: Players

- **Batch server log**
 - job requeued (SWO)
- **SMW logs: console, netwatch, consumer**
 - Backend data available via e-mail
- **Planned:**
 - More from batch server log
 - Batch MOM log
 - syslog

Beyond Reporting

- **Motivation for job completion report: DOE Operation Assessment requirement**
- **Information useful other ways**
 - **Raw data**
 - Spot system wide issues
 - “Cannot connect to default server host <host>”
 - **Report data**
 - “High Counts for Category+User”
 - Promote proactive user services
 - **Daily summary data**
 - Error trend chart

Beyond Reporting: Error Trend



Conclusion

- **Printing error messages on user output helps debugging and facilitates job analysis**
- **Moving error analysis out of epilogue enables more in-depth analysis**
- **Catch-all patterns allows expanding error patterns**
- **Requests to Cray**
 - Tag all Cray messages (portals, ALPS)
 - Publish message catalogues

Future Work

- **Finish up phase III implementation**
- **Modify existing data collection modules**
 - When application exit codes/signal become available in syslog
 - When CMS becomes available
- **Complete the abridged data study**
 - Huge files not feasible to save for too long
 - Beginning 100 and ending 400 lines good enough for analysis?
- **Add data to daily summary file**
 - Job size
 - Compute resource usage by node-hours
 - Per request



Thank You

We'd like to thank:

- **DOE for supporting NERSC computing.**
- **Cray for providing resources to look into our requests.**
- **Steve Luzmoor of Cray for assisting with problem analysis. Steve also authored and tracked many Bugs.**
- **Steve and Rita Wu, also of Cray, for responding our queries quickly.**
- **Tina Declerck of NERSC for helping with the hung jobs investigation.**



NERSC Contacts

Wendy Lin: hclin@lbl.gov

Helen He: yhe@lbl.gov

Woo-Sun Yang: wyang@lbl.gov