

Restoring the CPA to CNL


NATIONAL CENTER
FOR COMPUTATIONAL SCIENCES



presented by

Mark Fahey (for Don Maxwell)

Oak Ridge National Laboratory
U.S. Department of Energy

- 
- Why?
 - How?
 - Database Layout
 - Populating the Database
 - Job Failures
 - Uses of Database
 - Issues
 - Future

ORNL ALPS Accounting Database Implementation

- Why?
 - Need for same functionality that existed in CPA (Catamount)
 - Accounting
 - Statistics
 - Number of failed jobs, etc.
 - Troubleshooting
 - Site scripts used to determine which application is causing problems on a given node at a given time
 - Detecting orphaned reservations
- How?
 - Use SEC (Simple Event Correlator) to watch the MOAB event logs
 - SEC (realtime) approach needed to support troubleshooting tools
 - Start and End records call perl script which populates database tables
 - Perl script gathers information from 5 different sources
 - MOAB event logs
 - TORQUE accounting logs
 - MOAB partition logs
 - ALPS apsched logs
 - Syslogs

Database Organization

- Mostly modeled after the CPA database
 - **Jobs**
 - Job table
 - Job processor table
 - Job failure table
 - **ALPS**
 - ALPS table
 - ALPS processor table
 - Potentially multiple apruns in a job
 - Tied to Job table using keys

Job Tables

```
CREATE TABLE job_accounting (  
  hostname VARCHAR(80),  
  reservation_id BIGINT UNSIGNED NOT NULL,  
  session_id BIGINT UNSIGNED NOT NULL,  
  queue VARCHAR(80),  
  job_id VARCHAR(80),  
  job_name VARCHAR(80),  
  job_duration INTEGER UNSIGNED,  
  walltime INTEGER UNSIGNED,  
  account VARCHAR(80),  
  uid VARCHAR(64) NOT NULL,  
  exec_host VARCHAR(80),  
  create_time DATETIME NOT NULL,  
  destroy_time DATETIME,  
  job_err INTEGER UNSIGNED,  
  num_of_compute_processors INTEGER UNSIGNED NOT NULL,  
  num_of_service_processors INTEGER UNSIGNED NOT NULL,  
  cleaned_by ENUM ('client', 'ras'),  
  INDEX (hostname, reservation_id, session_id)  
) TYPE=InnoDB;
```

Job Tables (cont'd)

```
CREATE TABLE job_accounting_processor_list (  
  hostname VARCHAR(80),  
  reservation_id BIGINT UNSIGNED NOT NULL,  
  session_id BIGINT UNSIGNED NOT NULL,  
  processor_id INTEGER UNSIGNED NOT NULL,  
  INDEX (hostname, reservation_id, session_id),  
  PRIMARY KEY (hostname, reservation_id, session_id,  
  processor_id),  
  FOREIGN KEY (hostname, reservation_id, session_id)  
  REFERENCES job_accounting(hostname, reservation_id,  
  session_id) ON UPDATE CASCADE  
)  
TYPE=InnoDB;
```

ALPS Tables

```
CREATE TABLE alps_accounting (  
  hostname VARCHAR(80),  
  apid BIGINT UNSIGNED NOT NULL,  
  reservation_id BIGINT UNSIGNED NOT NULL,  
  session_id BIGINT UNSIGNED NOT NULL,  
  login_processor INTEGER UNSIGNED NOT NULL,  
  process_id INTEGER UNSIGNED NOT NULL,  
  command VARCHAR(255),  
  create_time DATETIME NOT NULL,  
  destroy_time DATETIME,  
  num_of_compute_processors INTEGER UNSIGNED NOT NULL,  
  num_of_service_processors INTEGER UNSIGNED NOT NULL,  
  exit_info VARCHAR(255),  
  INDEX (hostname, reservation_id, session_id),  
  PRIMARY KEY (hostname, apid),  
  FOREIGN KEY (hostname, reservation_id, session_id) REFERENCES  
    job_accounting(hostname, reservation_id, session_id) ON UPDATE CASCADE  
) TYPE=InnoDB;
```

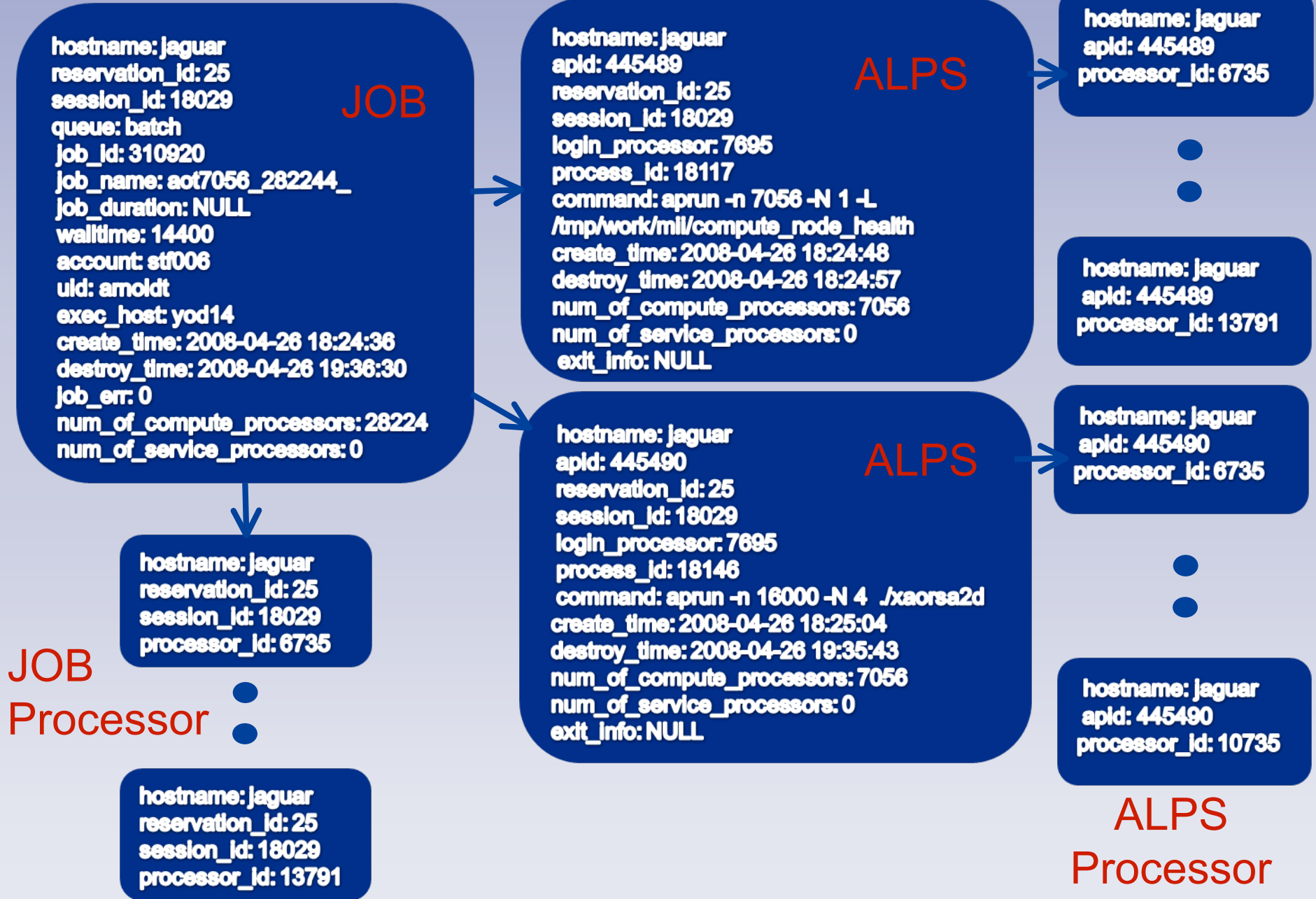
ALPS Tables (cont'd)

```
CREATE TABLE alps_accounting_processor_list (  
  hostname VARCHAR(80),  
  apid BIGINT UNSIGNED NOT NULL,  
  processor_id INTEGER UNSIGNED NOT NULL,  
  PRIMARY KEY (hostname, apid, processor_id),  
  INDEX (hostname, apid),  
  FOREIGN KEY (hostname, apid) REFERENCES  
  alps_accounting(hostname, apid)  
) TYPE=InnoDB;
```


Job Failure Table

```
CREATE TABLE job_failure (  
  hostname VARCHAR(80),  
  reservation_id BIGINT UNSIGNED NOT NULL,  
  session_id BIGINT UNSIGNED NOT NULL,  
  job_id VARCHAR(80),  
  fail_time DATETIME NOT NULL,  
  category ENUM ('hardware', 'software'),  
  reason ENUM ('user', 'system'),  
  description VARCHAR(80),  
  text VARCHAR(512),  
  INDEX (hostname, reservation_id, session_id),  
  FOREIGN KEY (hostname, reservation_id, session_id)  
  REFERENCES job_accounting(hostname, reservation_id,  
  session_id) ON UPDATE CASCADE  
) TYPE=InnoDB;
```

ORNL ALPS Database



ORNL ALPS Database (cont'd)

JOB

hostname: jaguar
reservation_id: 25
session_id: 18029
queue: batch
job_id: 310920
job_name: aot7056_282244_
job_duration: NULL
walltime: 14400
account: stf006
uid: arnoldt
exec_host: yod14
create_time: 2008-04-26 18:24:36
destroy_time: 2008-04-26 19:36:30
job_err: 0
num_of_compute_processors: 28224
num_of_service_processors: 0

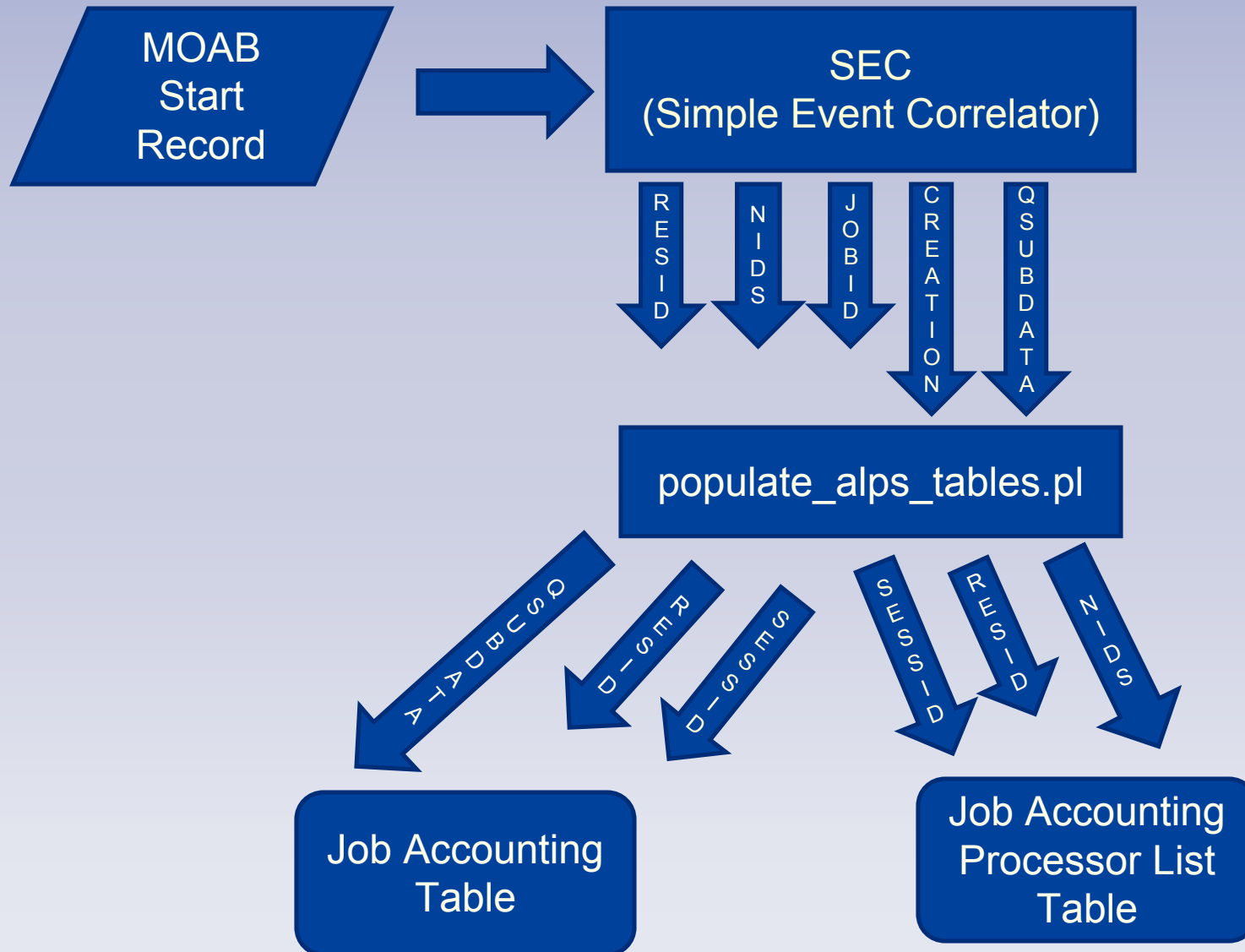


**JOB
Failure**

hostname: jaguar
reservation_id: 25
session_id: 18029
job_id: 310920
fail_time: 2008-04-26 19:34:34
category: hardware
reason: system
description: Machine Check Exception
text: Node c28-2c0s0n3 Machine Check
Exception Bank 4 Status
fe1aa00064080813 Addr f8152ad0

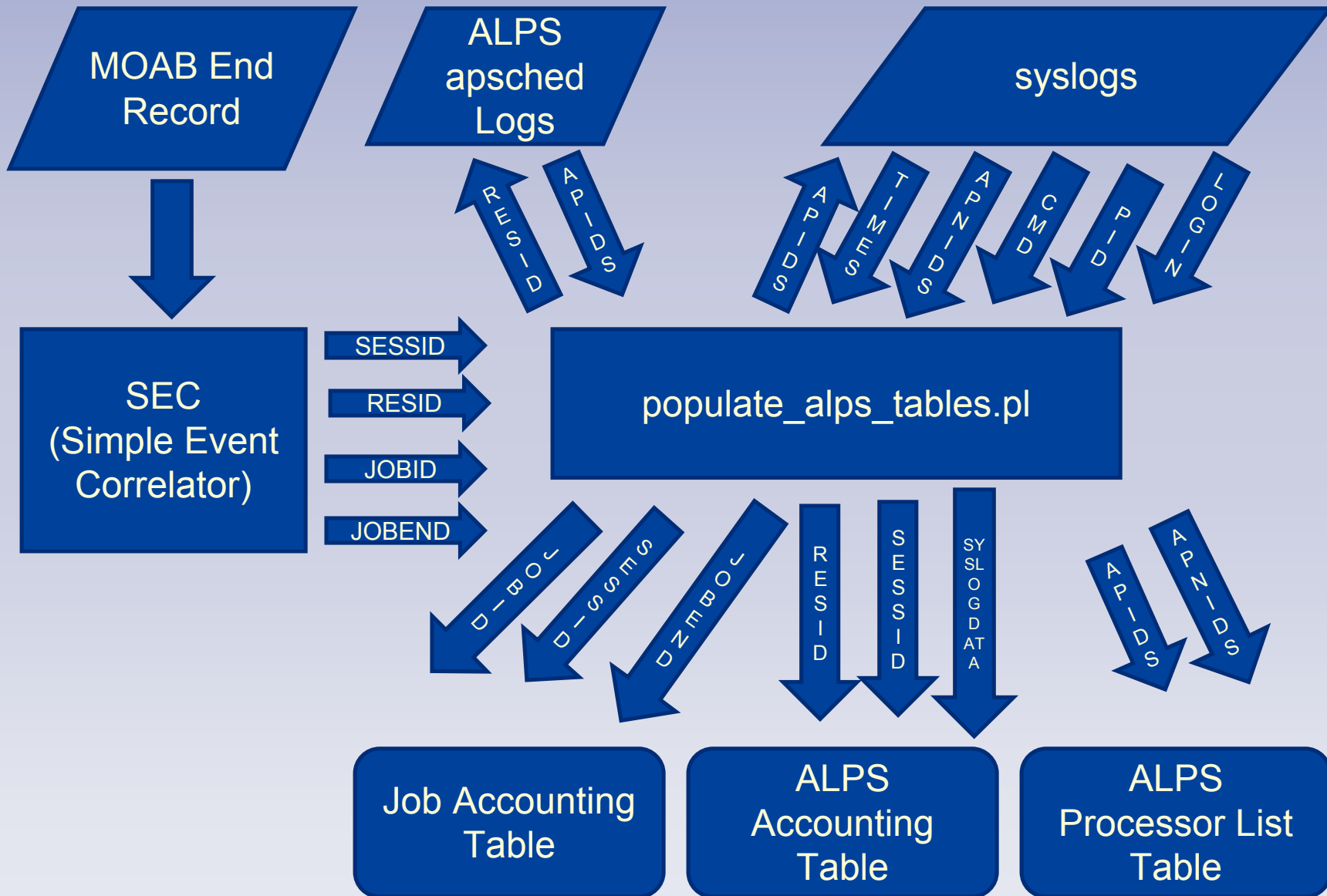
ORNL ALPS Database Sources

Job Accounting Table



ORNL ALPS Database Sources

ALPS Accounting Table



Job Failures

- Primary focus to this point has been hardware failures
 - **SEC watching console/netwatch/consumer logs on SMW**
 - **Failure records generated**
 - Date/Time
 - Node
 - Category (hardware/software)
 - Reason (user/system)
 - Description (e.g.)
 - Machine Check Exception
 - Seastar Heartbeat Fault
 - Kernel Panic
 - Seastar Lockup
 - Link Inactive
 - Out of Memory
 - **Using Job tables, exact job killed by hardware event is found and job failure record created**

Job Failures

- Catastrophic errors (link inactive/SCSI errors) are handled by determining from the database what was running at the time the event happened. Failure records are then generated for each job.
- Many SEC rule dependencies developed to attempt to capture the real issue when multiple events are seen for one problem.
- Further work
 - **Capturing errors from aprun**
 - **aprun wrapper has been developed**
 - Save the exit status of each aprun command
 - Update the ALPS table exit_info field
 - **Could this instead be tied into xtok (node health) via a userexit?**

Job Failures

- A nice outcome to all this work was the development of a concise machine status

2008-04-18 20:49:58 Machine Boot

2008-04-19 16:05:07 Node c25-0c0s4n0 Machine Check Exception Bank 4 Status fe0020003f080813 Addr 1f0092ac0

2008-04-19 16:05:59 Node c25-0c0s4n0 SeaStar Heartbeat Fault Explicit Portals firmware panic - Check the opteron

2008-04-20 00:43:57 Node c17-2c2s6n1 Machine Check Exception Bank 4 Status fe46200085080813 Addr 178062c40

2008-04-20 00:44:11 Node c17-2c2s6n1 SeaStar Heartbeat Fault Explicit Portals firmware panic - Check the opteron

2008-04-20 02:39:12 Node c11-3c0s2n3 Machine Check Exception Bank 4 Status fe5fa00094080813

2008-04-20 02:39:22 Node c11-3c0s2n3 Heartbeat Fault with No Seastar Heartbeat Fault

2008-04-20 05:47:57 Node c30-3c1s1n0 Heartbeat Fault with No Seastar Heartbeat Fault

2008-04-20 09:30:10 Node c30-3c1s1n0 Kernel Panic pop

2008-04-20 12:05:29 Node c23-2c0s5n2 SeaStar Heartbeat Fault Explicit Portals firmware panic - Check the opteron

2008-04-20 19:41:10 Node c10-2c0s5n0 Machine Check Exception Bank 4 Status fc03a000aa080a13 Addr 15910e600

2008-04-20 19:41:51 Node c10-2c0s5n0 SeaStar Heartbeat Fault Explicit Portals firmware panic - Check the opteron

2008-04-20 19:44:27 Node c29-0c2s1n0 SeaStar Heartbeat Fault Explicit Portals firmware panic - Check the opteron

2008-04-20 22:16:42 Recv Sequence Error c10-2c0s4s0I2 c10-2c0s5s0I3

2008-04-20 22:16:42 Link Inactive c10-2c0s4s0I2 c10-2c0s5s0I3

2008-04-20 22:18:24 Machine Shutdown

What can be done with all this data?

- Daily troubleshooting
 - **Tools can be written to query the database**

[2008-05-01 00:39:48][c25-1c1s0n0]Kernel panic - not syncing: Machine check ← console message

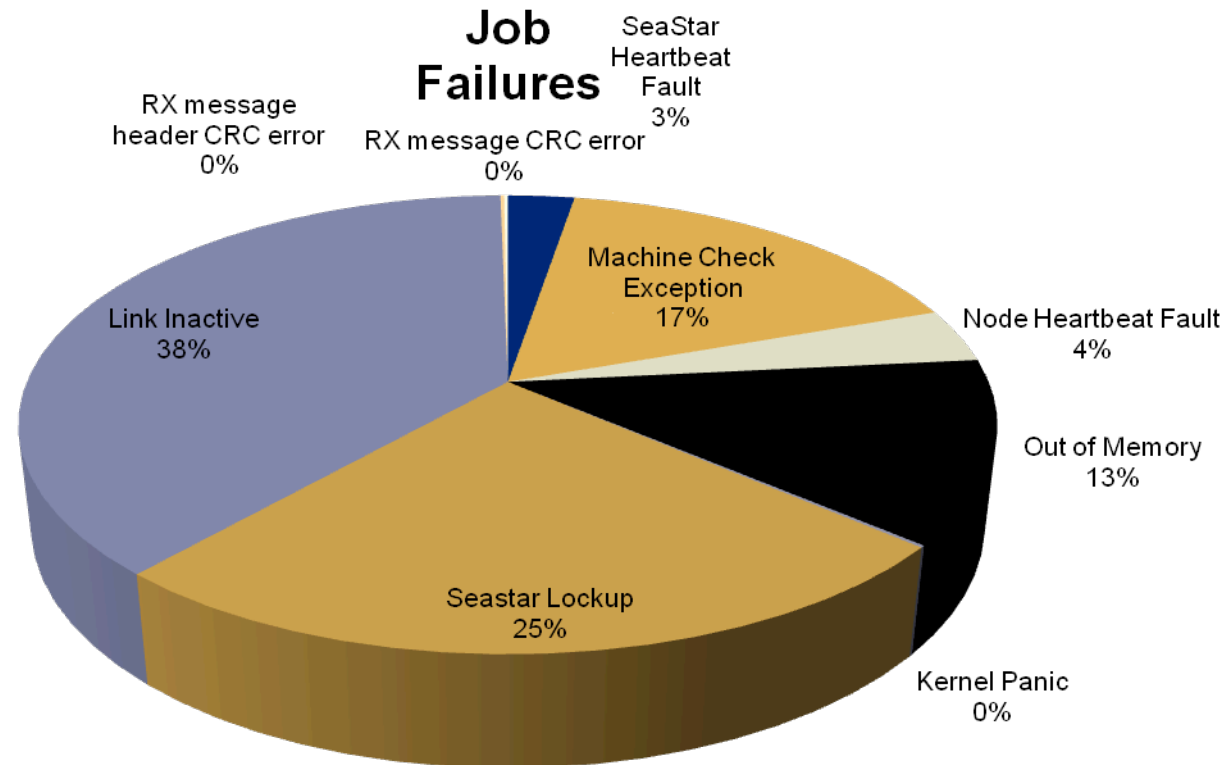
> find_job [2008-05-01 00:39:48][c25-1c1s0n0]
Searching for job on 9888 at time 2008-05-01 00:39:48...

← utility to find the job that was impacted

```
***** 1. row *****
hostname: jaguar
reservation_id: 174
session_id: 15397
queue: batch
job_id: 333801
job_name: ibtc12000_s3000_N4
job_duration: NULL
walltime: 3600
account: stf006bf
uid: rsankar
exec_host: yod9
create_time: 2008-05-01 00:32:06
destroy_time: 2008-05-01 00:41:27
job_err: 0
num_of_compute_processors: 12000
num_of_service_processors: 0
cleaned_by: NULL
hostname: jaguar
reservation_id: 174
session_id: 15397
processor_id: 9888
```

What can be done with all this data?

- Statistical analysis of failures by category
 - Which failures are killing more jobs?
 - Size distribution of jobs being killed
 - Possibilities are endless



Issues

- Database keys require multiple fields
 - **Reservation ids cannot be primary since ids repeat at each reboot**
 - **Session ids are just pids of TORQUE mom processes, so they repeat**
 - **Job ids repeat after a crash (a currently running job gets rerun)**
 - **All three certainly provide a level of uniqueness but some records have not loaded**
- Numerous data sources error prone
 - **Requires tweaking to coordinate timestamps among various log files**
 - **Log files can miss data under heavy load or due to bugs in various systems**

Requirements/Desires/Promises

- Hooks in ALPS to retrieve this information in a reasonable way that doesn't involve 5 sources, log files, etc.
- Desirable that Cray create and populate a database, but if not, at least provide the information so that the customer can do as they wish
- Cray has committed to providing a unique PAGG in UNICOS/lc 2.1
 - **Should solve the unique key problem**
- Other discussions at CUG regarding long-term system management issues

- Contact:
 - **Don Maxwell**
 - **maxwellde@ornl.gov**