Practical Support Solutions for a Workflow-Oriented Cray Environment

Adam G. Carlyle, Ross G. Miller, Dustin B. Leverman, William A. Renaud, Don E. Maxwell

Oak Ridge National Laboratory

National Climate-Computing Research Center (NCRC)

- Joint computing initiative between Oak Ridge National Laboratory (ORNL) and the National Oceanic and Atmospheric Administration (NOAA).
- "Gaea" - Cray XT6 and XE6 partitions
- Automated workflow = unique challenges

Outline:
Unique Support Challenges

- Lustre soft quotas need to be enforced on subsets of end-users; directory size reporting tools needed.
- Geographically remote support and development teams at (3) physical locations.
- Single job failures can cause workflow disruptions and must be investigated; unscheduled file system downtimes are particularly impactful.

Practical Support Solutions

1. Lustre reporting: LustreDU
2. Remote Support Teams: NCRC System Dashboard
3. File System Outages: Lustre-Aware Moab

Issue 1: Lustre Directory Reporting

- Need to generate an overview of file system structure and sizes regularly.
- `du` over a 1-10 PB Lustre file system with (1) MDS returns… in days? In weeks? Never?
- Need a utility that returns on “human” time-scales.

Solution: LustreDU for Non-intrusive Directory Reporting

1. Simple user-facing command line binary

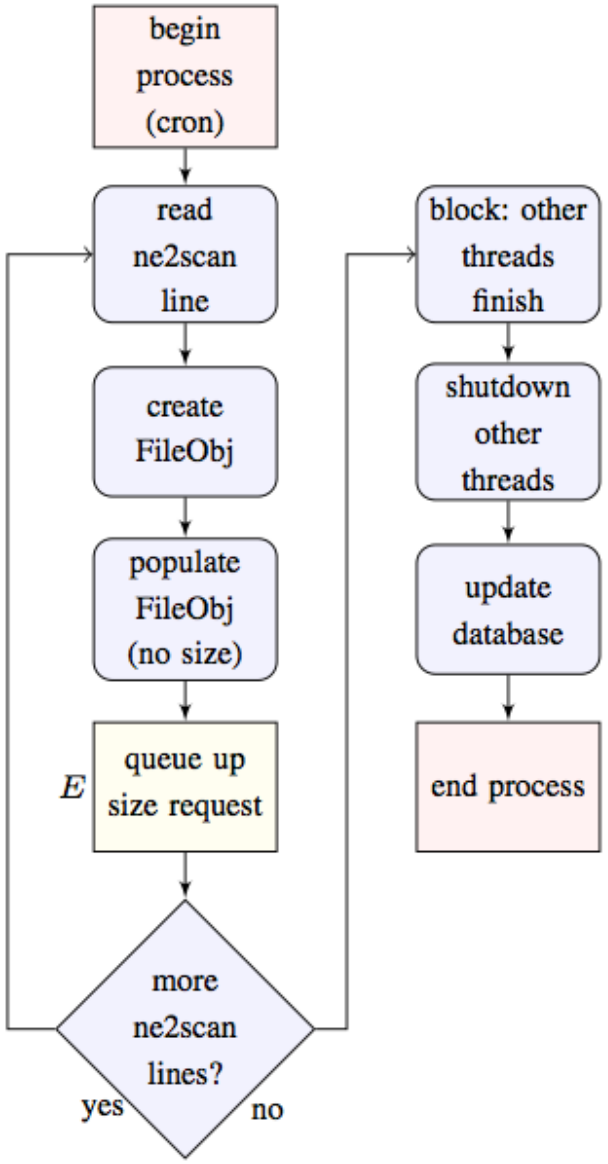
```
[user@host] (/scratch) $ lustredu ./somedir
Last Collected Date        Size    File Count  Directory
2012-04-10 07:29:33     103.91 KB           1  ./somedir
```

2. Database backend

3. Main processes to populate database
   - Runs twice a day, returns in a few hours

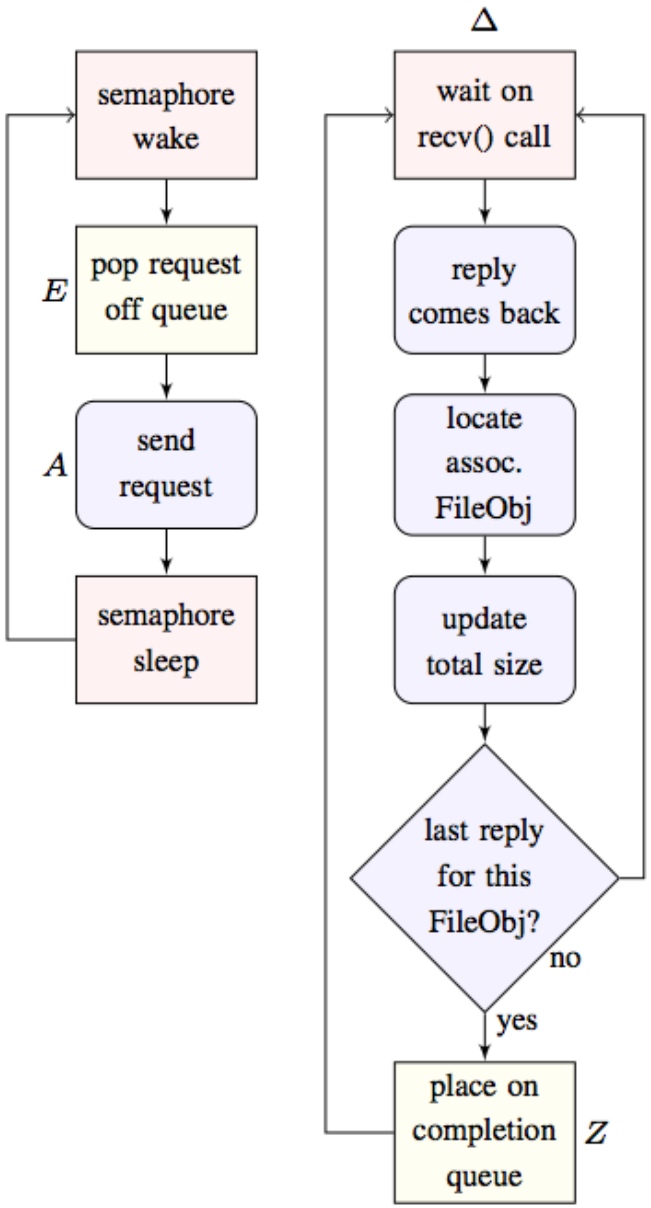
LustreDU Internals: Main Process

- Starts with ne2scan output
- Runs on server with access to ne2scan output
- Main process sets in motion numerous asynchronous parallel threads



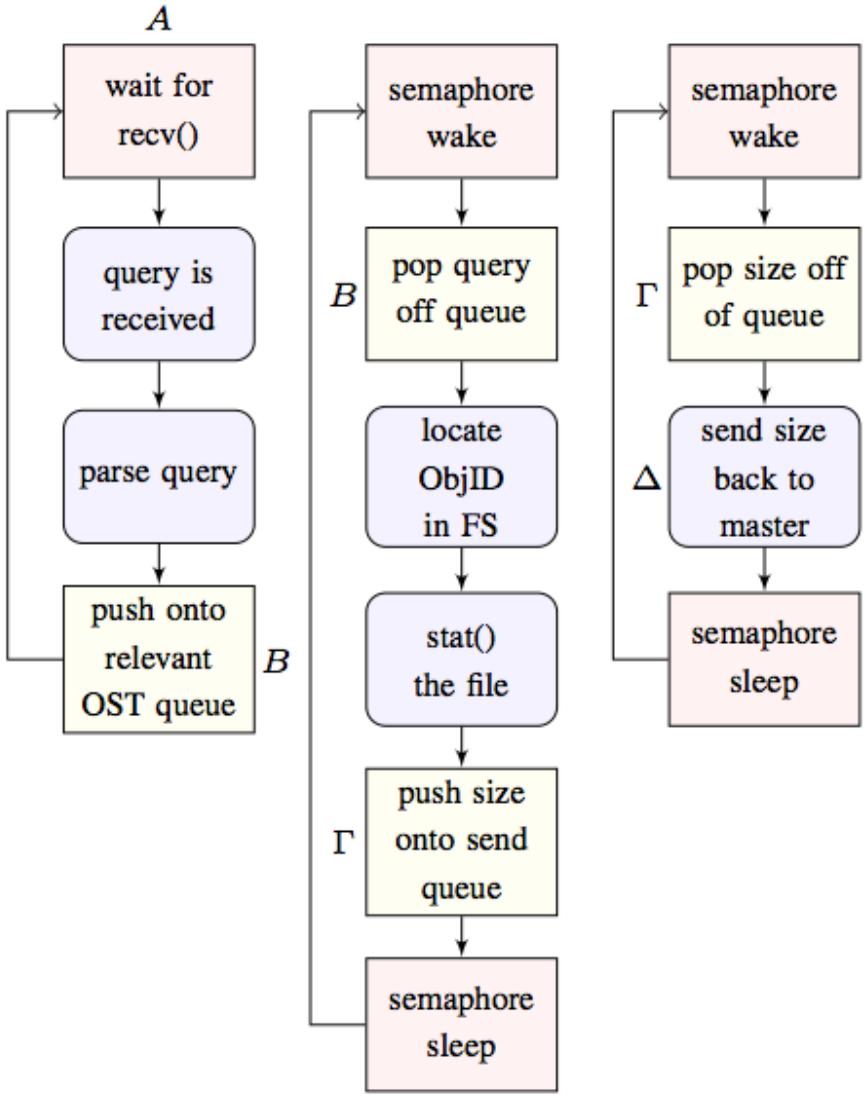
LustreDU Internals: Network Threads

- Runs alongside master process.
- Network send (left) and receive (right) threads process.



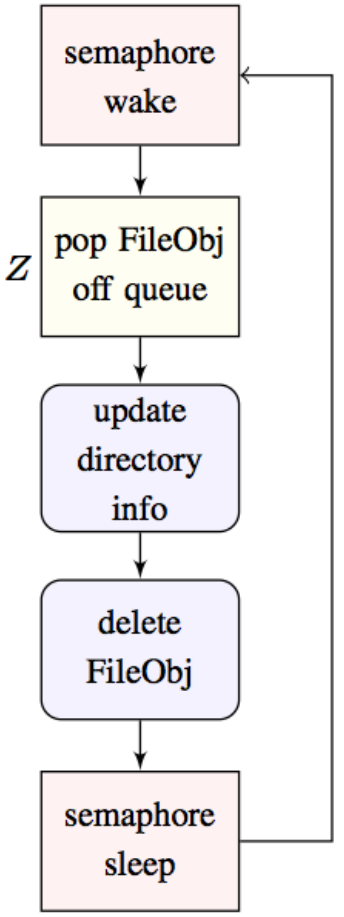
LustreDU Internals: OSS Daemons

- OSS daemon threads: receive (left), send (right),
- `stat()` thread (center) handles fetching of size requests.



LustreDU Internals: Completion Thread

- Completion thread completes FileObject, updates directory records and deletes the FileObject.
- Runs alongside master process.

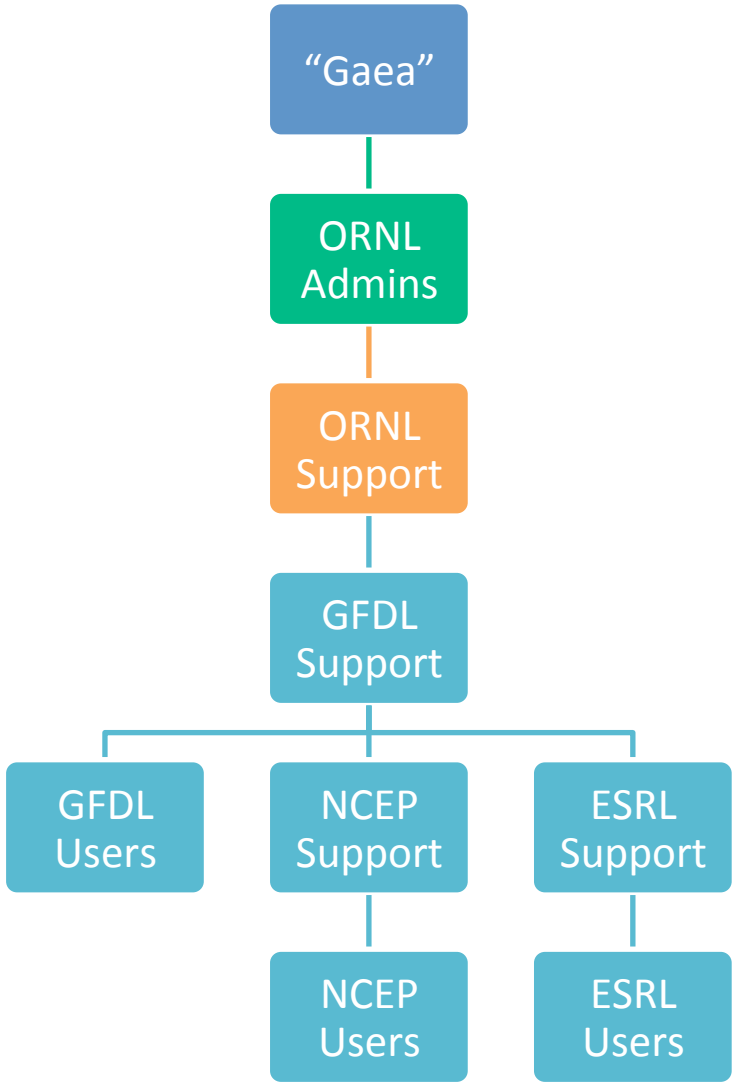


Outcome

- Interactive directory reporting becomes feasible on human time scales (i.e. seconds)
- End-users have a functional tool for self-monitoring.
- Support teams have a tool for reporting to management.
- Drawback: Data are the state of the file system as of last scan.

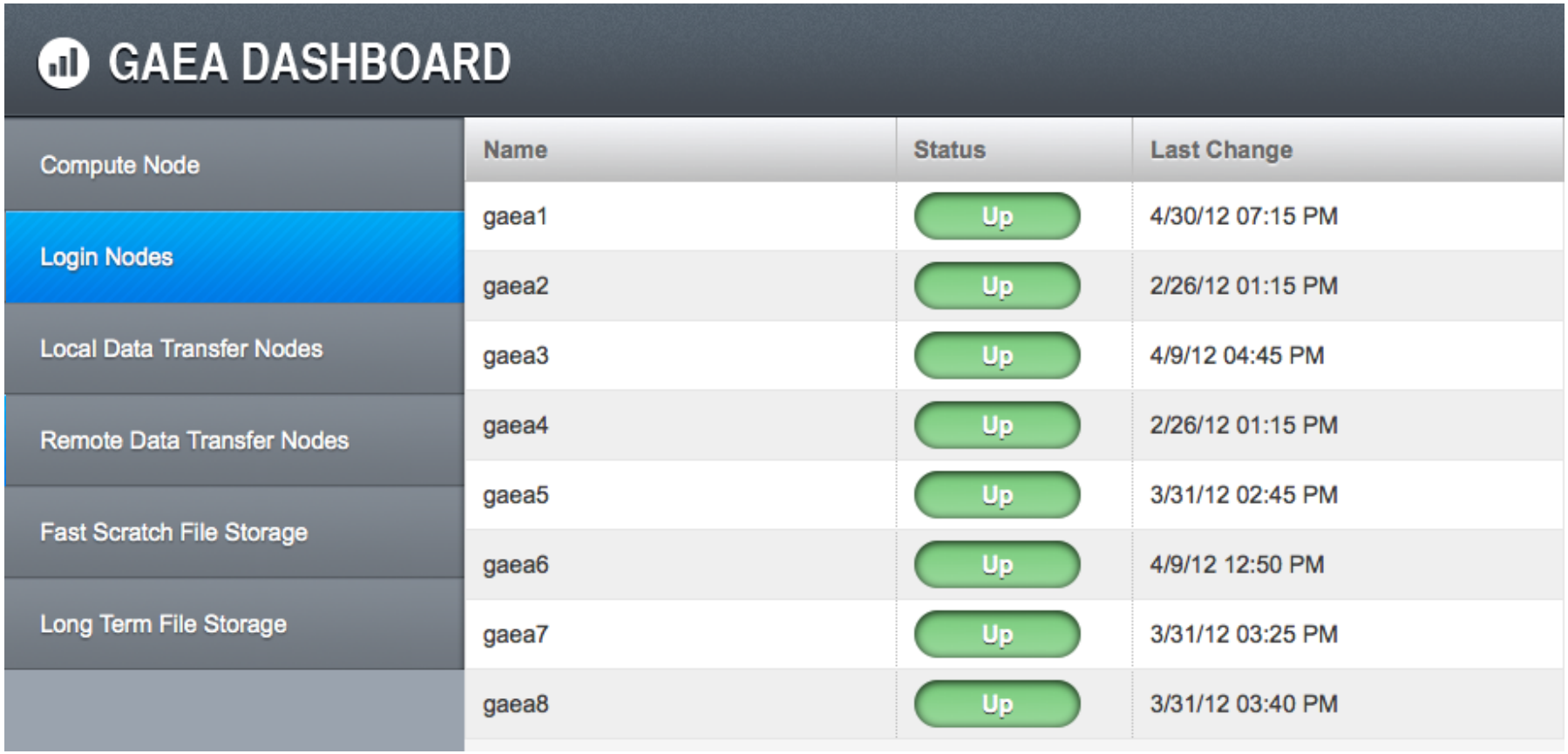
Issue 2: Remote Support Teams

- Primary support team is remote from primary admins.
- Other support teams are remote from primary support teams
- Needed system monitoring for non-admins and users alike.



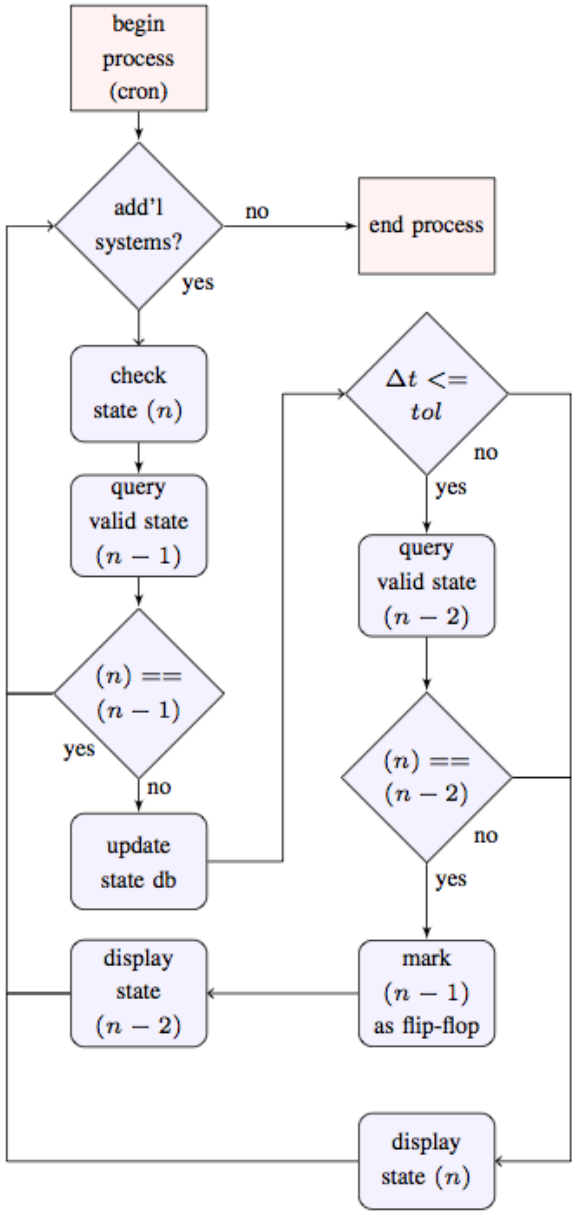
Solution: System Dashboard

- Provides concise system status to geographically remote support teams.



System Status Dashboard

- Straightforward loop over system-level Nagios checks
- Catching false positives requires a bit more effort.



Outcome

- System events disseminated to teams and end users in near-real time.
- Support teams and users can try to correlate job failures with system events.

Issue 3: Workflow Disruption From File System Outages

- In the event of a file system outage:

```
10   running jobs crash
20   resources are freed
30   scheduler starts new jobs
40   goto 10
```

- Leaves the workflow in an unknown state… analogy:

```
CONFLICT (content) Automatic merge failed; fix
conflicts and then commit the result.
```

Solution: Lustre-Aware Moab

- Combination of FS checks and Lustre File System Checks via Nagios plugins
  - File system mounting checks
  - `health_check`
- Nagios Event Handler to communicate with Moab

Lustre-Aware Moab

- How to Integrate into Moab?
- Moab *generic consumable resource(s)* to associate computational resource(s) and file systems
- Moab *submit filters* to associate jobs to dependent computational resources

Outcome

- Job workflow can respond to unscheduled file system outages before too much damage is done.
- Running jobs are lost, but queued jobs are spared. Failed jobs can be resubmitted at given a priority boost to run before those queued.


Questions?
carlyleag@ornl.gov