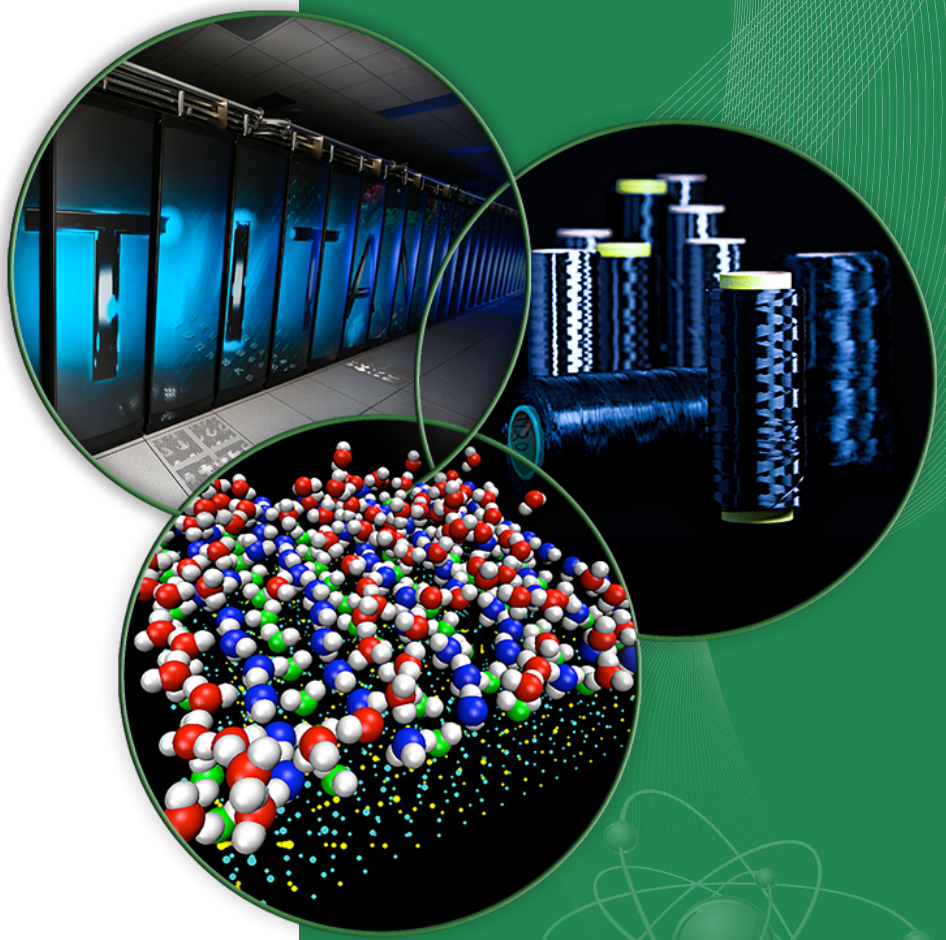
Lustre Job Stats Metric Aggregation at OLCF



From 0 to monitoring in 3 RPMs

Jesse Hanley

Rick Mohr

March 8th, 2016

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory

What we'll cover

- The need for monitoring
- (Not) Reinventing the wheel
- Basic deployment

Why monitor?

- Better monitoring = potential better use of resource
- Understanding file system use cases
- Identifying problems
- Metric collection isn’t new

What to monitor?

- A plethora of metrics available
- Most are listed on the Lustre wiki:
  - http://wiki.lustre.org/Lustre_Monitoring_and_Statistics_Guide
- Concentrating on Job Stats:
  - `obdfilter.*.job_stats` for OST statistics
  - `mdt.*.job_stats` for MDT statistics

Job Stats

- Introduced in LU-694
  - https://jira.hpdd.intel.com/browse/LU-694
- Can be set globally with a conf_param on the MGS, or set per client (lctl set_param jobid_var)
- Modes included:
  - "`disable`" (Default)
  - `procname_uid`
  - Anything else. In this mode, the value is treated as an environment variable

Problems with this method

- The procname_uid setting produces a record for each process, and is more useful for debugging
- The environment variable setting has a few problems:
  - Potential for slowdowns in file-create heavy workloads
  - Not supported for in-kernel client
    - https://www.mail-archive.com/linux-kernel@vger.kernel.org/msg528724.html
  - Not available on all platforms

Performance Differences

File per process (shared directory)

```
mpirun -n 8 -N 8 mdtest -n 131072 -d output/run -F -C -T -r -N 8
```

# of files	Jobstats enabled?	Env var set?	File creates per sec	File stat per sec	File removeal per sec	Tree creates per sec	Tree removal per sec
1048576			1577	2968	2308	1275	36
1048576	x		1474	2882	2198	962	28
1048576	x	x	1445	2877	2171	1067	30
1048576	procname_uid	procname_uid	1563	2953	2277	1192	21

Performance Differences

File per process (unique directory)

```
mpirun -n 8 -N 8 mdtest -n 131072 -d output/run -F -C -T -r -N 8 -u
```

# of files	Jobstats enabled?	Env var set?	File creates per sec	File stat per sec	File removeal per sec	Tree creates per sec	Tree removal per sec
1048576			3399	7434	4002	371	18
1048576	x		3241	6936	3884	343	16
1048576	x	x	3358	6881	3873	317	15
1048576	procname_uid	procname_uid	3377	7452	3940	302	15

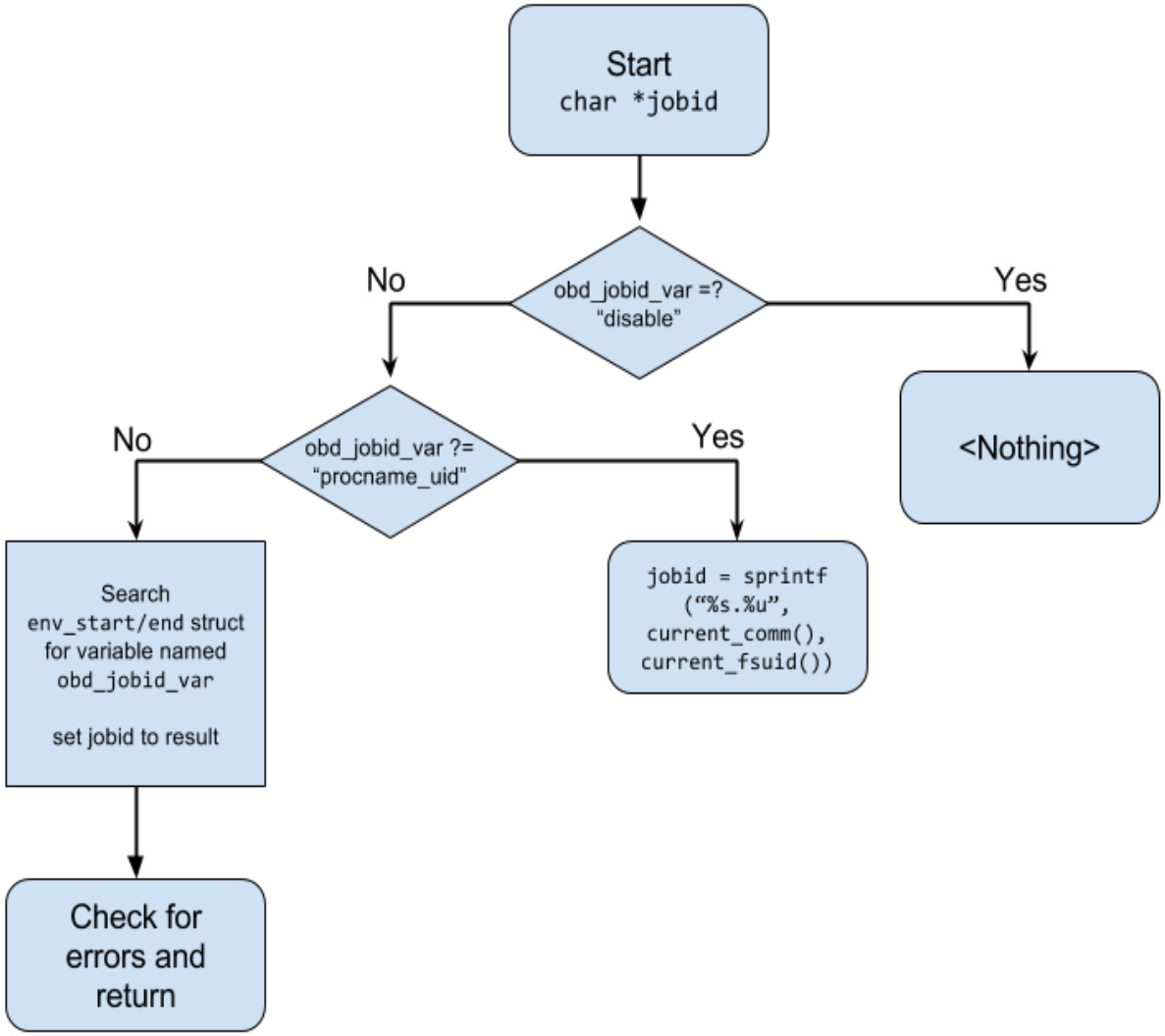
Shared file

File per process (unique directory)

```
mpirun -n 8 -N 8 mdtest -S -C -T -r -n 1 -d output/run -F
```

# of files	Jobstats enabled?	Env var set?	File creates per sec	File stat per sec	File removeal per sec	Tree creates per sec	Tree removal per sec
1048576			1609	5686	11721	1125	3208
1048576	x		1513	5196	16448	1179	3190
1048576	x	x	1573	5163	14704	1143	3091
1048576	procname_uid	procname_uid	1599	4460	11738	1124	2969

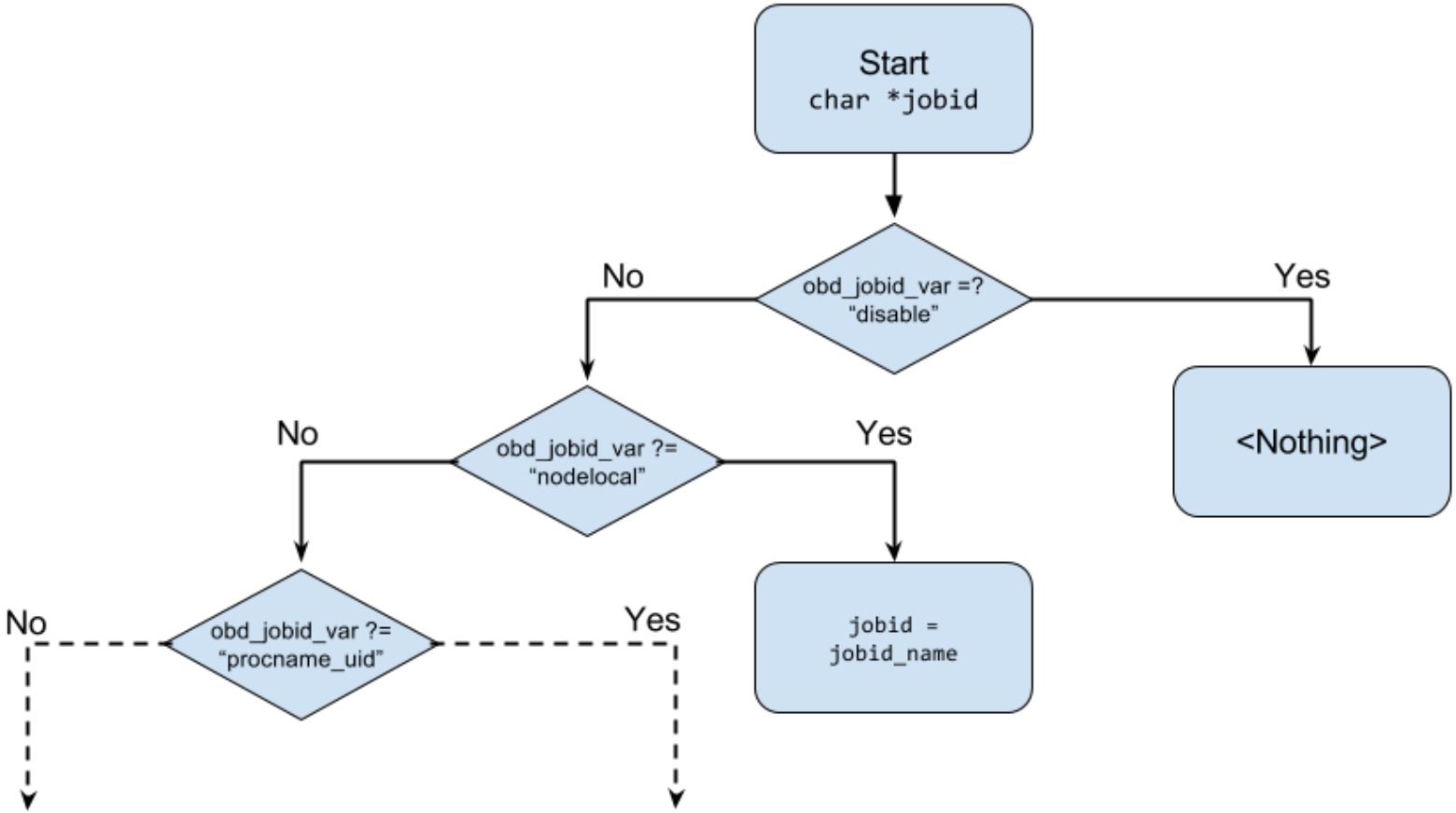
Execution Flow



Solution

- A setting to allow any static string to be configured
- Requested through https://jira.hpdd.intel.com/browse/LU-7195
  - Currently how the upstream in-kernel build works
  - Allows a single string to be configured for the entire node
  - Different functionality and potentially more or less useful depending on use case: shared node vs dedicated

New Execution Flow



What to do with this data

1. ***Gather*** *the data on hosts you are monitoring. Deal with the syntax, extract what you want*
2. ***Collect*** *the data centrally - either pull or push it to your server, or collection of monitoring servers.*
3. ***Process*** *the data - this may be optional or minimal.*
4. ***Alert*** *on the data - optional but often useful.*
5. ***Present*** *the data - allow for visualization, analysis, etc.*

- Lustre Wiki

Gather

- Telegraf
  - Collector written in Go that runs on the MDS and OSS nodes
  - Available via RPM, deb, and source
  - https://github.com/influxdata/telegraf
  - Dozens of input plugins – targets to collect
  - Several of output plugins – targets to send to for collection/storage
  - Includes a Lustre input plugin

Collecting Job Stats

- By default, telegraf only collects stats files:

```
/proc/fs/lustre/obdfilter/*/stats
/proc/fs/lustre/osd-ldiskfs/*/stats
/proc/fs/lustre/mdt/*/md_stats
```

- Patched to support Job Stats:

```
/proc/fs/lustre/obdfilter/*/job_stats
/proc/fs/lustre/mdt/*/job_stats
```

Collect / Process

- InfluxDB
  - Same developers behind Telegraf
  - Time series database
  - Flexible collection and storage rules
  - Job, MDT/OST and host are all tags
  - Query language similar to SQL

Query example

```
> select non_negative_derivative(sum("value"),1s) as "value" from
lustre2_js_open where time > now() - 6h group by time(30s)
fill(previous) limit 10;

name: lustre2_js_open

---------------------
time                    value
1456434030000000000
1456434060000000000     15298.5
1456434090000000000     14172.433333333332
1456434120000000000     13635.366666666667
1456434150000000000     13351.133333333333
1456434180000000000     14706.733333333334
1456434210000000000     12898.733333333334
1456434240000000000     13871.9
1456434270000000000     12684.5
```

Presentation

- Grafana
  - Visualization software for a variety of backend sources, including InfluxDB, Graphite, and Elasticsearch
  - http://grafana.org/

Potential Queries

- Bandwidth
- Usage (Disk and inode)
- System load (memory, CPU)
- Cache access / cache hit / cache miss
- Metadata operations
- All over time, averaged per minute/hour/day, maximums and minimums, rates of change
- With Job Stats, these can be per “job”

Thank You

Jesse Hanley <hanleyja@ornl.gov>

Rick Mohr <rmohr@utk.edu>