BLUE WATERS

SUSTAINED PETASCALE COMPUTING

May 12, 2016

Drain Time Analysis at Scale

Joshi Fullop

National Center for Supercomputing Applications

GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION

Background

- As the size of machines continues to grow, the greater the importance for efficiency.
- Capability computing is often characterized by running jobs with greater node counts.
- Jobs have a drain cost in addition to their execution time.
- Drain time goes against node occupancy metrics (utilization).

Defining 'Drain Time'

- What it is NOT.
  - Scheduler definition of Drain/Draining state.

the state where a node is not currently able to accept additional workload due to administrative action

- What it IS – for this paper's purposes
  - Common definition

amount of time that node is held in reserve and prohibited from running workload in order to allow enough nodes to become available to start another job

Available Data

- Moab server logs
  - Node state dump every iteration.

```
2015-09-01T00:02:11.000-0500     Node '1' status: state='Busy'
rsvlist='2190780,jmo8k.147195,2183821,2183815,2183819' joblist='2190780'.

2015-09-01T00:02:11.001-0500     Node '4' status: state='Busy'
rsvlist='2201168,jmo8k.147195,2183821,2183815,2183819' joblist='2177494'.

2015-09-01T00:02:11.001-0500     Node '5' status: state='Idle'
rsvlist='jmo8k.147195,2183821,2183815,2183819' joblist='none'.
```

1.7TB over about 2 years

Available Data

- Moab node 'state' has 6 possibilities
  - *Down, Idle, Busy, Running, Drained*, and Draining**
- Rsvlist – 2 possibilities
  - Populated or 'None'
- Joblist – 2 possibilities
  - Populated or 'None'

- Totals 24 separate accumulators per node.

System-wide Accounting

```
struct node_acc {
    bool bxk;               // flags XK vs XE nodes
    long last_utime;        // last state timestamp
    int last_state;         // last Moab node state
    int last_idx;           // last rsvlist/joblist state
    long node_basis;        // total accounted for seconds
    long accum[7][4];       // 24 + 4 accumulators
};
```

Extra 4 accumulators for an Error state where iteration time is out of bounds or other problems.

Importance of Basis

- Basis is the total number of accumulated node seconds that are considered.
- Should be compared to the total number of expected node seconds as a quality of data check.
- Great indicator on the quality of the data.

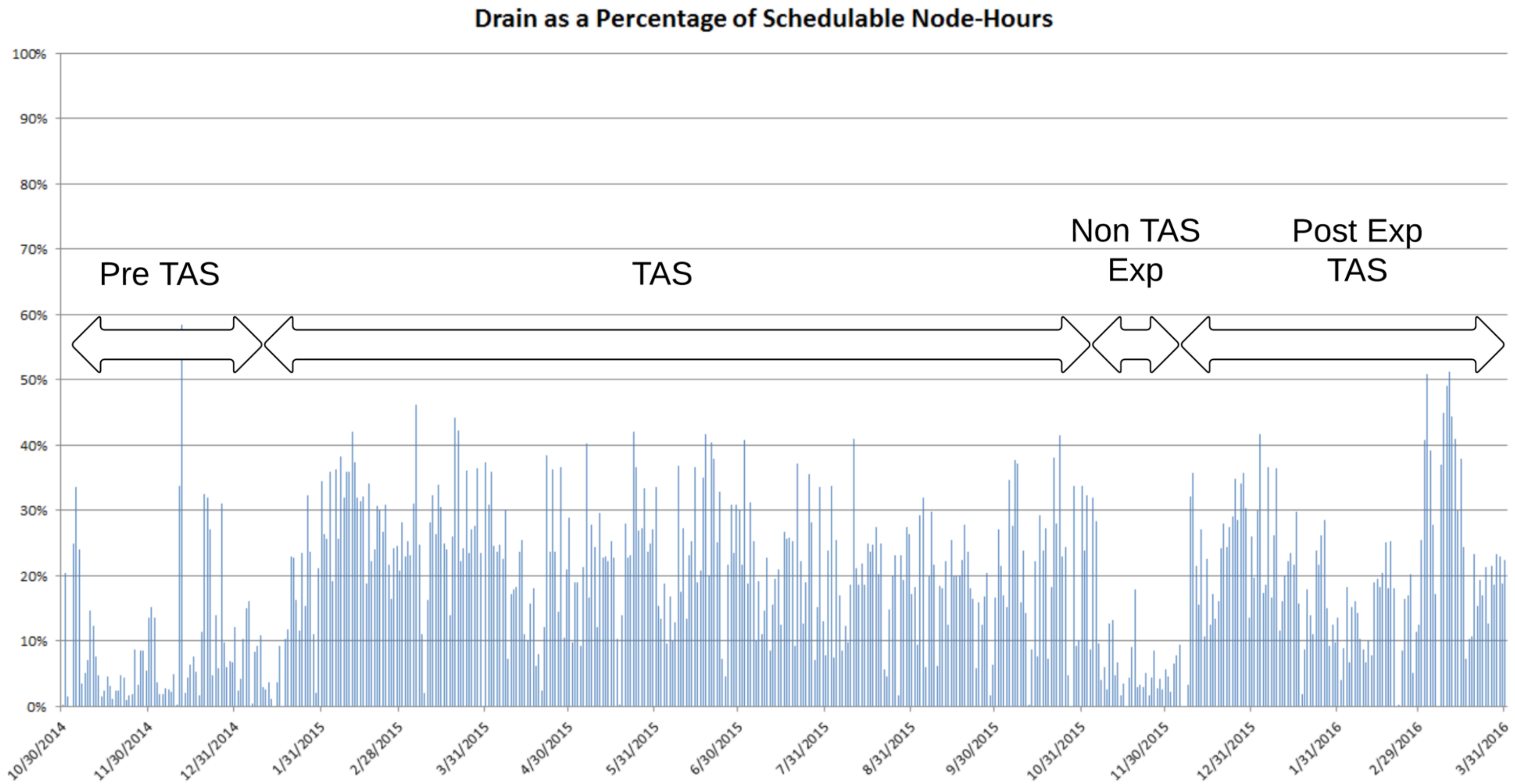
System-wide Analytics

- Identifying Policy Oversight
  - Jobs with outlying drain times should be scrutinized.
- Perpetually Sliding Jobs
  - Jobs that continue to accrue drain time may be constantly sliding and may never get launched.
- Evaluate Changes in Scheduler Policy
  - Gives one factor for comparison.

Example: Topology Aware Scheduling (TAS)

- Jobs are scheduled in contiguous bricks of nodes on the torus network.
- Reduce or eliminate cross-node network traffic and minimize average hops between nodes.
- Results in improved performance and consistency.
- Causes decreased node-occupancy.

Topology Aware Scheduling Example



Topology Aware Scheduling Example

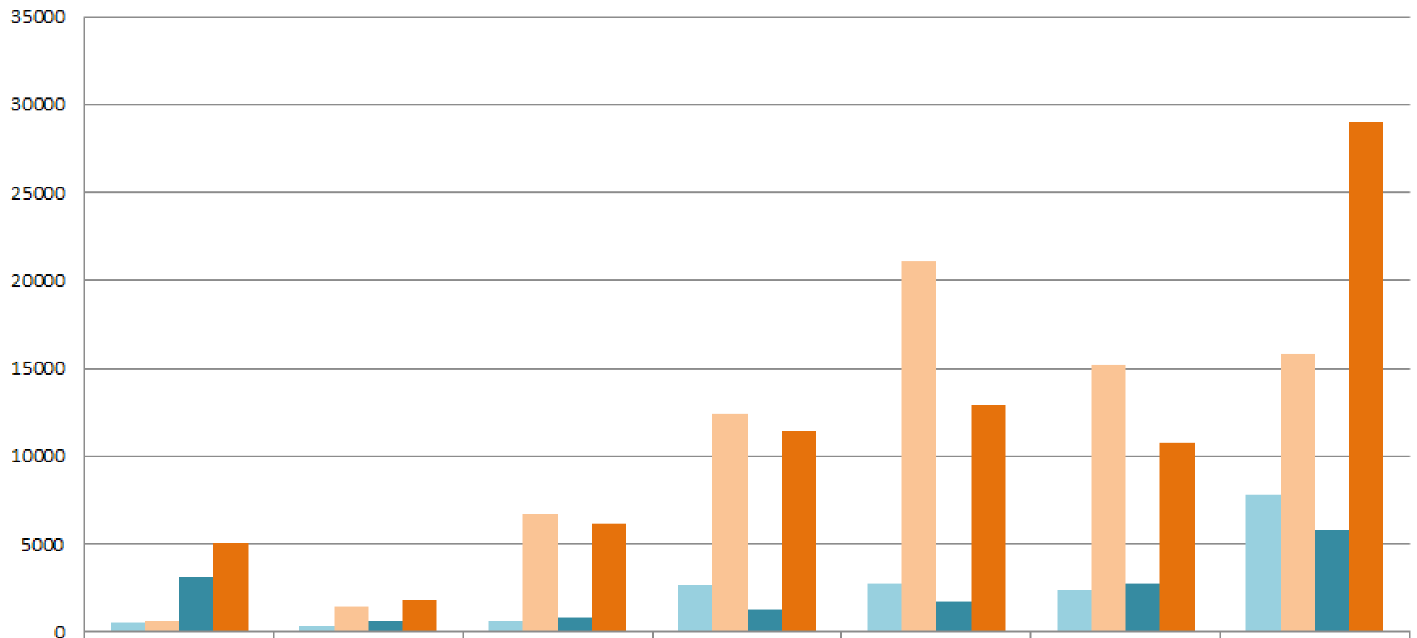
Period	Drain Time %
Pre-TAS	**9.1%**
TAS	**22.3%**
Non-TAS Experiment	**5.8%**
Post Experiment TAS	**21.3%**

TAS increased drain cost 14%.

Per-Job Accounting

- Similar to node-accounting,
  1. Use an associative array indexed by job/reservation.
  2. Only accumulate drain time.

Average Drain Time (node-seconds) to Job Size Ratio



	Tiny	Sub1k	1k+	2k+	4k+	8k+	16k+
Pre TAS	493	329	606	2,645	2,759	2,376	7,787
TAS	580	1,455	6,751	12,399	21,092	15,223	15,830
Non-TAS Exp	3,146	661	819	1,304	1,709	2,739	5,742
Post Exp TAS	5,055	1,825	6,192	11,438	12,924	10,776	28,986

Valuation of Preparedness

- Examination of jobs that cause drain and then fail to run.
  - Job walltime < 30 seconds
  - Over a period of 1.4 years, 1.77 Million Node hours were wasted draining for these jobs.
  - However, that equates to only 0.5% of the system.
  - Equivalent to 128 nodes for the entire period.

Conclusions

1. Drain time is part of running large jobs.
2. Before you can manage something, you must measure it.
3. Drain time can accumulate to significant node-hours.
4. Drain time grows incrementally more for larger job.
5. There are many other factors to consider when selecting a scheduling policy. Drain time should be one of them.

ACKNOWLEDGMENT & DISCLAIMER

This work was supported by the National Science Foundation (NSF) and the DOE Office of Science (Office of Basic Energy Sciences).

Special to thank Mark Dalton of Cray and Rick McKay of Adaptive Computing for their verification of the interpretation of the log file information.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Fin

Joseph 'Joshi' Fullop IV

fullop@illinois.edu

http://www.ncsa.illinois.edu/People/jfullop