User-Centric Workload Analytics: Towards Better Cluster Management

Saurabh Bagchi
Purdue University

Supported by National Science Foundation (NSF)
Jul `15-Jul `18

Joint work with: Subrata Mitra, Suhas Javagal, Stephen Harrell (Purdue), Adam Moody, Todd Gamblin (LLNL)
Presentation available at: *engineering.purdue.edu/dcsl*

Problem Context

- Shared computing clusters at university or government labs is not uncommon
- Users have a varying level of expertise
  - Writing own job scripts
  - Using scripts like a black box
- Varying user needs
  - High computation power
    - Analysis of large structures (Civil, Aerospace engineering)
  - High Lustre bandwidth for file operations
    - Working with multiple large databases/files (Genomics)
  - High Network Bandwidth
    - A parallel processing application

Motivation

- Challenge for the cluster management
  - Need for customer centric analytics to pro-actively help users
  - Improve cluster availability
  - In addition to failures, investigate performance issues in jobs
- Need for open data repository of system usage data
  - Currently, lack of publicly available, annotated quantitative data for analyzing workloads
  - Available public data sets provide only system level information and not up-to-date
  - Dataset must not violate user privacy or IT security concerns

URL: https://github.com/purdue-dcsl/fresco

Cluster Details: Purdue

- Purdue's cluster is called Conte
- Conte is a ``Community" cluster
  - 580 homogeneous nodes
  - Each node contains two 8 core Intel Xeon E5-2670 Sandy Bridge processors running at 2.6 GHz
  - Two Xeon Phi 5110P accelerator card, each with 60 cores
  - Memory: 64GB of DDR3, 1.6 GHz RAM
- 40 Gbps FDR10 Infiniband interconnect along with IP
- Lustre file system, 2GB/s
- RHEL 6.6
- PBS based job scheduling using Torque

Cluster Details: LLNL

- SLURM: Job Scheduler
- TOSS 2.2 OS
- 16-core Intel Xeon processors (Cab)
- 12-core Intel Xeon processors (Sierra)
- 32GB memory (Cab)
- 24GB memory (Sierra)
- 1296 nodes (Cab) and 1944 nodes (Sierra)
- Infiniband network

Cluster Policies

- Scheduling:
  - Each job requests for certain time duration, number of nodes and in some cases, amount of memory needed
  - When job exceeds the specified time limit, it is killed
  - Jobs are also killed by out-of-memory (OOM) killer scripts, if it exhausts available physical memory and swap space
- Node sharing:
  - By default only a single job is scheduled on a an entire node giving dedicated access to all the resources
  - However, user can enable sharing by using a configuration in the job submission scripts

Data Set

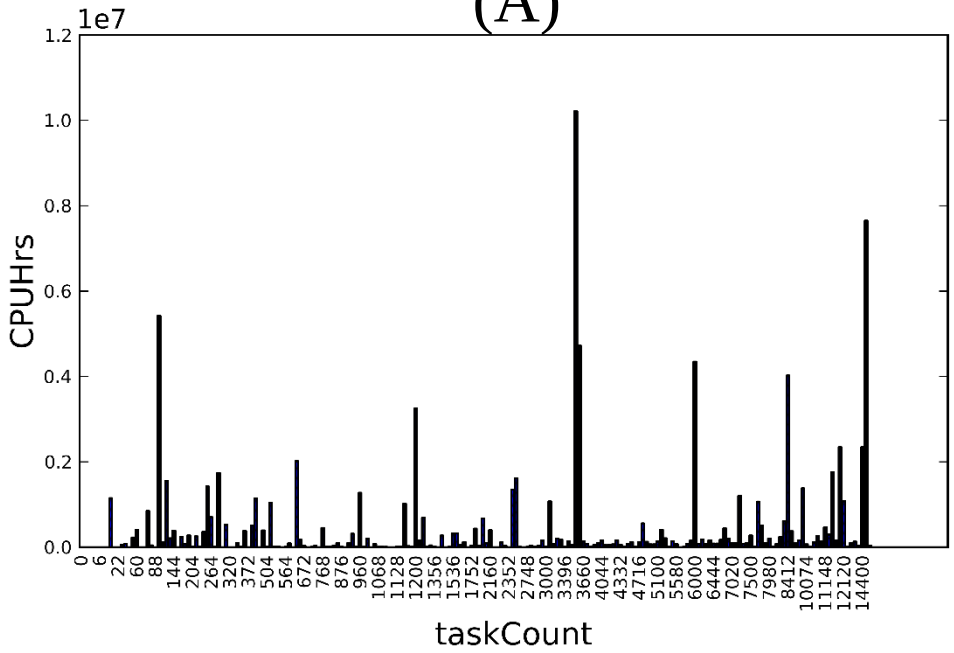
- Accounting logs from the job scheduler, TORQUE
- Node-level performance statistics from TACC stats
  - CPU, Lustre, Infiniband, Virtual memory, Memory and more…
- Library list for each job, called liblist
- Job scripts submitted by users
- Syslog messages

Summary	**Conte**	**Cab** and **Sierra**
Data set duration	Oct'14 – Mar'15	May'15 – Nov'15
Total number of jobs	489,971	247,888 and 227,684
Number of users	306	374 and 207

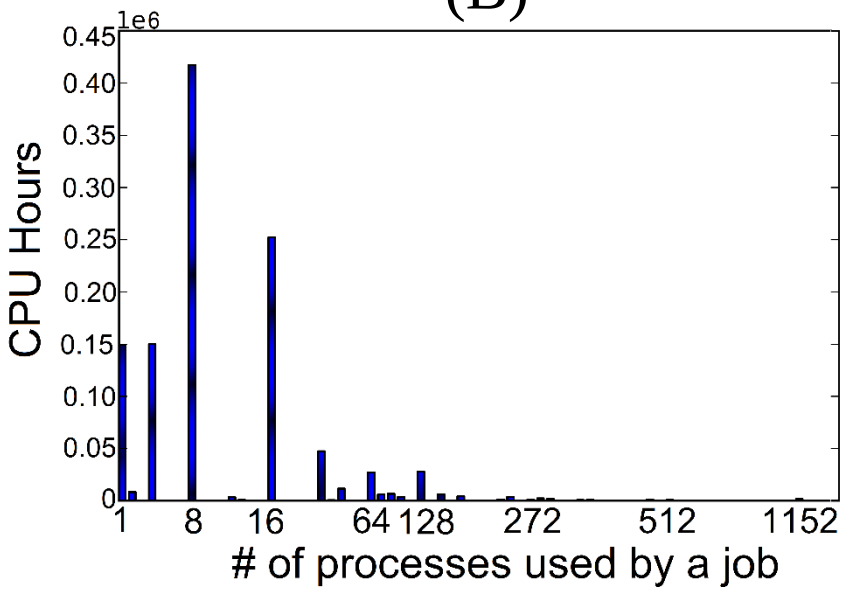
URL: https://github.com/purdue-dcsl/fresco

Analysis: Types of Jobs

(A)



(B)

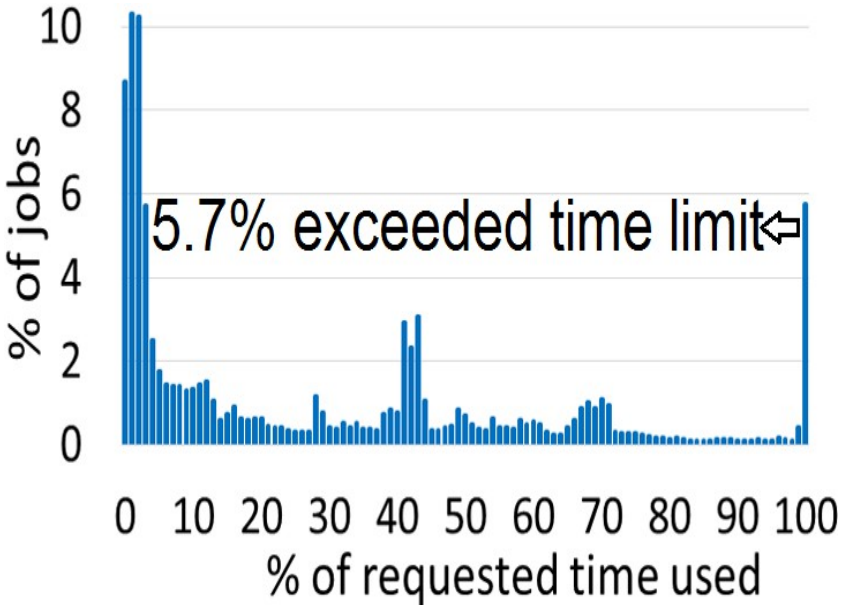


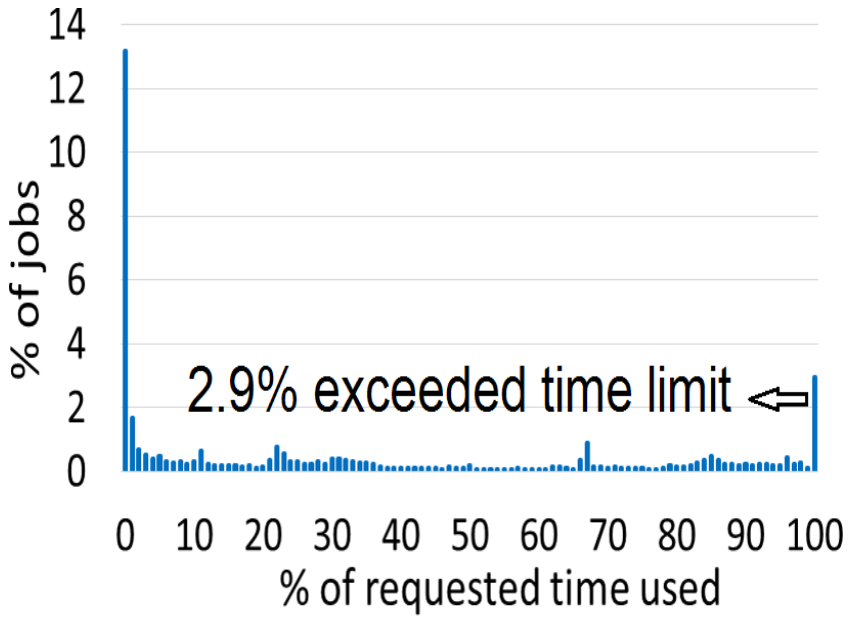
LLNL cluster

Purdue cluster

- Different job sizes
  - Purdue has a large number of “narrow” jobs
  - LLNL jobs span hundreds to thousands of processes

Analysis: Requested versus Actual Runtime



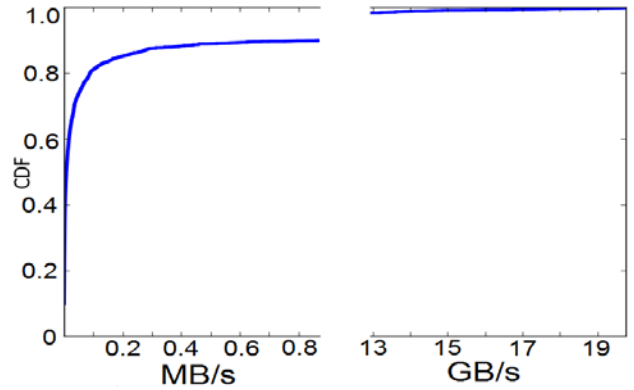


Purdue cluster

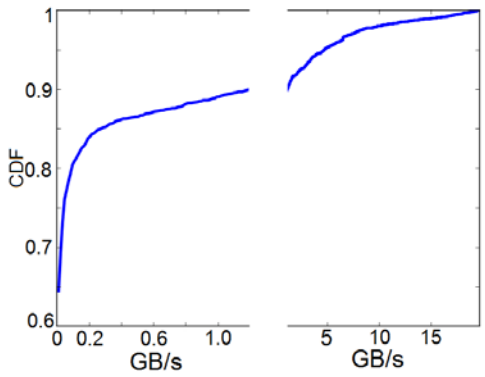
LLNL cluster

- Users have little clue how much runtime to request
  - Purdue: 45% of jobs used less than 10% of requested time
  - LLNL: 15% of jobs used less than 1% of requested time
- Consequence: Insufficient utilization of computing resources

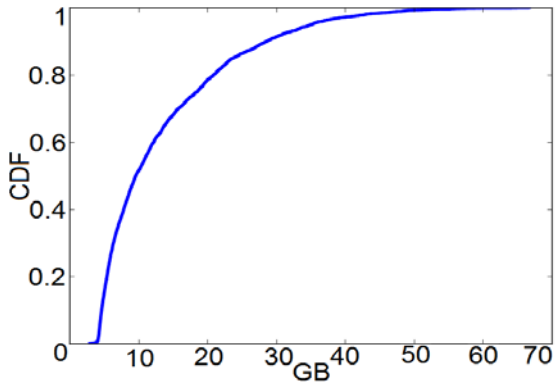
Analysis: Resource Usage by App Groups



Infiniband read rate on Conte



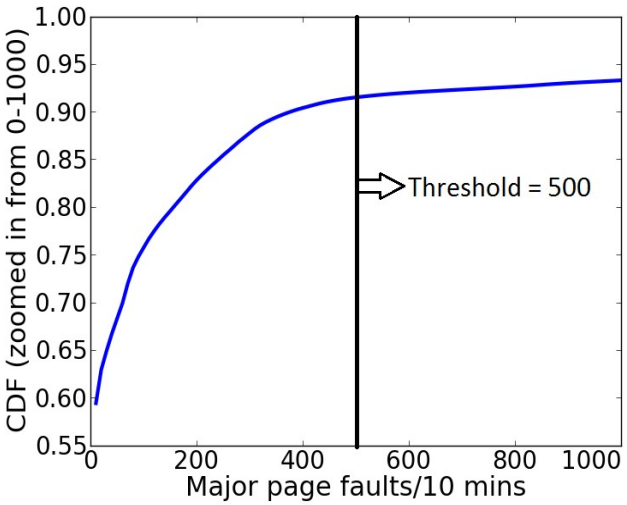
Lustre read rate on Conte



Memory usage on Conte

- Clearly, there are 2 distinct types of jobs
  - Few jobs need high bandwidth backplane for Network and IO
  - In case of memory, such a distinction is not present
- Follow-on: Specialized cluster built in 2015 for high resource demands
  - Has 56 GBps Infiniband network

Analysis: Performance Issues due to Memory



- Find a (quantitative) threshold on major page fault rate
- Find all jobs (and job owners) which exceed the threshold

- In the extreme, memory exhaustion leads to invocation of oom-killer, kernel level memory manager
- Multiple evidence for memory problems: Syslog messages with out-of-memory (OOM) code and application exit code
  - 92% of jobs with memory exhaustion logged OOM messages
  - 77% of jobs with OOM messages had memory exhaustion exit code

Current status of the repository

- Workload traces from Purdue cluster
  - Accounting information (Torque logs)
  - TACC stats performance data
  - User documentation
- Privacy
  - Anonymize machine specific information
  - Anonymize user/group identifiers
- Library list is not shared
  - For privacy reasons

URL: https://github.com/purdue-dcsl/fresco

Conclusion

- It is important to analyze how resources are being utilized by users
  - Scheduler tuning, resource provisioning, and educating users
- It is important to look at workload information together with failure events
  - Workload affects the kinds of hardware-software failures that are triggered
- Open repository started with the goal for different kind of analyses to enhance system dependability

URL: https://github.com/purdue-dcsl/fresco