Understanding HPC Application I/O Behavior Using System Level Statistics

Arnab K. Paul*, Olaf Faaland‡, Adam Moody‡, Elsa Gonsiorowski‡, Kathryn Mohror‡, Ali R. Butt*

*Virginia Tech, ‡ Lawrence Livermore National Laboratory

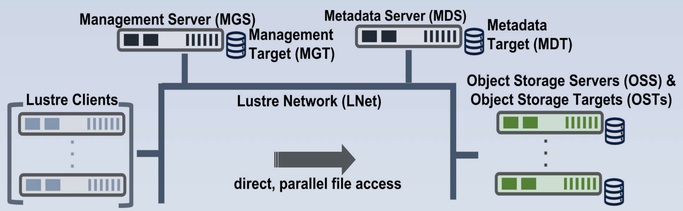
Introduction

- The compute performance of HPC systems is increasing at a much higher rate than storage performance.
- HPC system designers and administrators need to understand file system and application behavior to improve storage systems.
- We need to analyze statistics captured from the file system itself independently of user applications to draw meaningful conclusions.
- The Lustre file system is one of the most widely-used parallel file systems supporting seven of the top ten supercomputers in the latest Top-500 list (November, 2018).

Objective: *Collect* and *analyze* file system statistics from two LLNL systems (Cab and Quartz) each having a 15PiB Lustre file system.

Background

Lustre Distributed File System



Cluster Configuration

	Cab	Quartz
Processor Architecture	Xeon 8-core E5-2670	Xeon 8-core E5-2695
Operating System	TOSS 2	TOSS 3
Processor Clock Rate	2.6 GHz	2.1 GHz
Number of Nodes	1,296	2,634
# Cores per Node	16	36
Total # Cores	20,736	96,768
Memory per Node	32 GB	128 GB
Total Memory	41.5 TB	344.06 TB
Interconnect	QDR Infiniband	Intel Omni-Path 100 Gb/s
Tflops	426.0	3,251.4

Data Collection

Aggregate Job Statistics

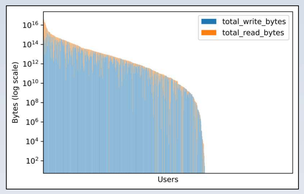
- The SLURM job scheduler is used to run a prolog and an epilog script for each node of an application which reads the proc file - */proc/fs/lustre/llite/lustre-file-system/stats;* which are then added to RDBMS database.
- Statistics are:
  - *starttime, endtime, duration, uid, nodes, mkdir, mknod, open, rename, rmdir, unlink, read/write_bytes, read/write_bytes_count, recv/send_bytes, recv/send_count*

Time-Series Job Statistics

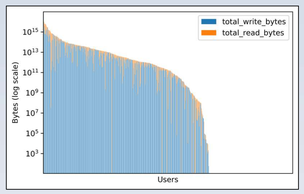
- Telegraf is used to collect Lustre JobStats data from servers - */proc/fs/lustre/mdt/*/job_stats* on MDS and */proc/fs/lustre/obdlter/*/job_stats* on OSSes, which are stored in influxdb.
- Statistics are:
  - **MDS** –*create, mkdir, mknod, open, rename, rmdir, unlink*
  - **OSS** – *jobstats_read/write_bytes, jobstats_read/write_calls*
  - *jobid, time*

Analysis

	Cab	Quartz
Duration of data collection	04/2015 – 03/2018	04/2017 – 03/2018
Total # of jobs	2,854,478	1,401,897
Total # of users	994	584
% Jobs with no I/O	47.9%	22.63%
% Users doing no I/O	35.2% (350)	33.5% (196)
Write Intensive Jobs	50.1%	76.23%

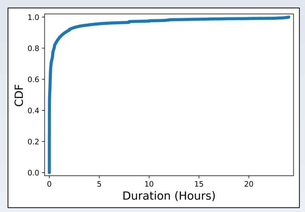


Cab

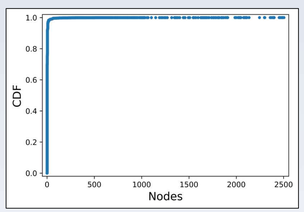


Quartz

Most of the users (> 65%) perform significant I/O in mostly write-intensive jobs.



Quartz: CDF Duration



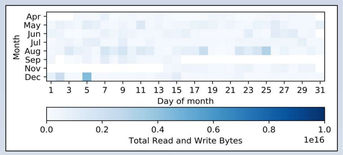
Quartz: CDF #Nodes

90% of jobs run for less than 2 hours and allocate less than 100 nodes. Mean duration < 52 minutes.

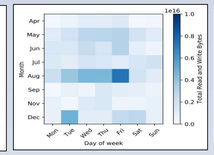
Analysis (contd.)

Users with inefficient writes (users who do more writes with less writes per call) – (bytes written > mean bytes written) & (bytes written per call < mean bytes written per call)

- **Cab:** 138 users (46.9%) – out of 294 users who write a lot of data.
- **Quartz:** 111 users (66%) – out of 168 users.

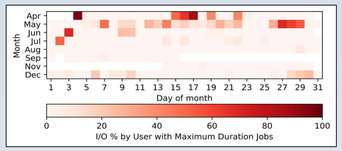


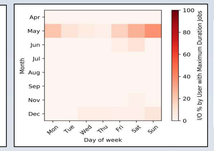
Quartz: I/O per Day



Quartz: I/O per Day of Week

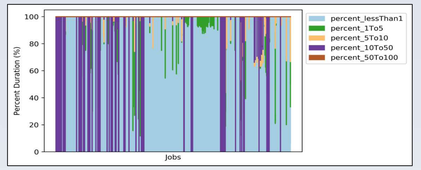
I/O performed by a job is not correlated with month, day of month/week or holiday.





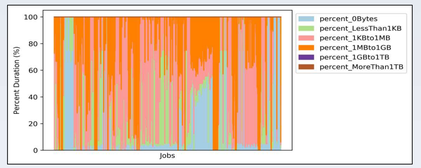
Quartz: Contribution of Job with Maximum Duration on I/O for a day or a day of week in 2017.

Long running jobs do not have a huge impact on the overall I/O traffic.



Quartz: Percent I/O duration vs write burst size as % of memory

90% of jobs never write burst data larger than 1% of memory size



Quartz: Percent I/O duration vs write burst size

Most jobs write burst data in the range of few kilobytes for the majority of their I/O duration.

Conclusion

- Improving file system write performance is important as most jobs are write-intensive.
- Focus should be on jobs which run for short duration.
- There should be efforts to educate HPC users to develop applications which perform efficient writes to reduce I/O contention.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work is also sponsored in part by the NSF under the grants: CNS-1405697, CNS-1615411, and CNS-1565314/1838271.

LLNL-POST-784599