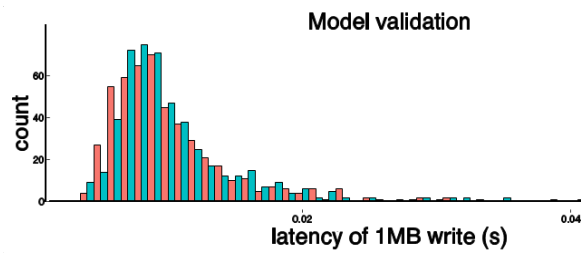
Analysis and Modeling of the End-to-End I/O Performance on Titan

Achievement:

- Created a hidden Markov model of the I/O performance on Titan
- The model was based on observed properties of the distribution of I/O latencies.
- This was used to predict the I/O performance of applications running on Titan



Our model is validated against future write latencies, and is shown to be highly accurate

Significance and Impact: Having the ability to predict I/O performance allows developers to either transfer I/O on different OSTs or delay I/O for more optimal times

Research Details

- Conduct test on 7 days of the week to get write latencies to different Object Storage Targets (OST) shown in the top figure
- Build a hidden Markov model based on the statistical properties of the observed I/O traces
- Validate the model by comparing the distribution of the predicted I/O latencies against the distribution of the real latency numbers in the traces which were not part of the model

Facility: Work was performed at Oak Ridge National Laboratory

PI and affiliation: Scott Klasky, ORNL

Team: L. Wan, M. Wolf, F. Wang, J. Choi, G. Ostrouchov, S. Klasky, Carlos Maltzahn (UC Santa Cruz), Jay Lofstead (Sandia), Manish Parashar (Rutgers), Mark Ainsworth (Brown), Qing (Gary) Liu (NJIT)

Publication: L. Wan, M. Wolf, F. Wang, J. Choi, G. Ostrouchov, S. Klasky, Analysis and Modeling of the End-to-End I/O Performance in OLCF's Titan Supercomputer, IEEE International Conference on High Performance computing and Communications, 2017

Sponsor: DOE

Overview:

By continuously sending 1MB probing write requests to multiple OSTs (object storage targets) of Titan's Lustre file system, we collected the end-to-end access latencies of those OSTs for seven consecutive days. Since different performance states are observed from the I/O latency data we collected and the transition between performance states shows strong Markov property, we build a hidden Markov model to characterize the I/O latency on Titan. Here is how we use our model to predict the I/O latency in an adaptive manner:

1) Gather 12 hours' I/O latency data
2) Use the data to train the hidden Markov model
3) Use the model predict the I/O latency in the next 12 hours
4) Train the model again with the newly collected data every one or two hours