

Yours Truly

Sunil Template



Contents

1 The Convergence of Enterprise, Internet Scale, and High Performance Computing Storage Infrastructures	1
<i>Jay Lofstead, Eric Barton, Matthew Curry, Carlos Maltzahn, Robert Ross, and Craig Ulmer</i>	
1.1 Introduction	2
1.2 Existing File Systems	3
1.3 Object-Based Stores	4
1.3.1 Big-Data/General Computing Optimized Object/Key-Value Stores	5
1.3.2 HPC Oriented Object Stores	5
1.4 Next Generation HPC Storage Systems	5
1.4.1 Lustre/DAOS	5
1.4.2 Kelpie/Data Warehousing	5
1.4.3 Hybrid Models	5
1.5 Conclusions	6
1.6 Glossary	6
Bibliography	7



Chapter 1

The Convergence of Enterprise, Internet Scale, and High Performance Computing Storage Infrastructures

Jay Lofstead
Sandia National Laboratories

Eric Barton
Intel

Matthew Curry
Sandia National Laboratories

Carlos Maltzahn
University of California, Santa Cruz

Robert Ross
Argonne National Laboratory

Craig Ulmer
Sandia National Laboratories

Abstract	2
1.1 Introduction	2
1.2 Existing File Systems	3
1.3 Object-Based Stores	4
1.3.1 Big-Data/General Computing Optimized Object/Key-Value Stores	5
1.3.2 HPC Oriented Object Stores	5
1.4 Next Generation HPC Storage Systems	5
1.4.1 Lustre/DAOS	5
1.4.2 Kelpie/Data Warehousing	5
1.4.3 Hybrid Models	5
1.5 Conclusions	6
Acknowledgements	6

Abstract

Large scale storage infrastructures have been significantly impacted by the growth in data analytics applications. High Performance Computing storage infrastructures, once the extreme end of the storage scale spectrum, must now adapt to technologies optimized for large scale data analytics applications. Hardware changes, such as storage class memory, are also affecting how the exascale storage stack will be constructed. We examine use cases, trends, convergent technologies, and new opportunities generated by this technology blending.

1.1 Introduction

HPC infrastructures have grown around the requirement to handle large, decomposed data structures for parallel computation. Single data objects may be 100s of TB spread across the entire machine. Parallel storage systems have grown trying to address performance and storage requirements while maintaining backwards compatibility with the standard POSIX interface. Unfortunately, this is proving increasingly difficult. Big data application, on the other hand, focus on searching through immense volumes of tiny items looking for patterns or correlations that may lead to insights. Some science applications, such as genomics, have a workload pattern similar to these big data applications. For these applications, parallel file systems are not well suited because of the broad, relatively continuous read demand of independent items does not benefit from the overheads of parallel file systems. Instead, distributing storage across the compute infrastructure makes more sense. With pressures to effectively leverage a single platform for these disparate workloads and the shifting storage market, new considerations for how to design storage resources for extreme scale compute systems must be made.

Traditionally, the HPC market has focused on supporting coherent and consistent output methods from parallel sources to parallel targets. This requirement is driven from validating that the output of a single item is complete and correct. Largely, the workload is write-intensive during the expensive, at scale computation process with a read-intensive phase lasting months on cheaper machines or at small scale with low priority. File systems like the dominant Lustre [?] and GPFS [?] systems have been carefully optimized to address these workloads.

The big data market has opposite priorities. The big computation phase requires reading in large data quantities for processing at scale. The output from this process can be handled at much smaller scale later and is orders of magnitude smaller. Given the small item focus, the overhead inherent in coherent and consistent storage for write intensive workloads is both unnecessary and a heavy cost. Instead, systems like HDFS [?] dominate. These work by storing files that will be read for processing throughout the compute area including assumption that failures are common prompting replication.

The output from initial stream or file processing for the big data workloads use distributed object-based storage technology. It offers independent, uncoordinated data access with a simplistic key search space for subsequent analysis. The profit potential for this market has caused an explosion in specialized products aimed at accelerating this processing style. For small enough data sets, in memory object stores like memcached and Radius dominate. For larger data sets, approaches like Google's Big Table are accomplishing the same function. There are also hardware products targeting this market segment, such as Kinect [?], offering a native object interface for the devices connected directly to a network.

Adding complexity to this storage environment is the relentless performance improvements and cost reductions for solid state storage, like NAND-based flash memory. These devices have already rendered 15,000 RPM disk drives obsolete. The 10,000 RPM disk drives will not survive for more than a few more years. New disk technology like shingled drives [?] offer a path for disks to survive longer. The enormous capacities for read intensive, write infrequently workloads is very attractive for many communities. For example, storing images created sequentially for later read-intensive processing can yield a better cost/performance balance.

This chapter investigates these new technologies and how they affect extreme scale computing. We evaluate how the HPC environment can and must adapt to this new storage environment to address future computing needs and to take advantage of the different kinds of technology being developed. We will also consider the planned reintegration of large scale computing from the split of big data applications from simulation-based computing with both the necessary and forced integration of these large, expensive platforms for multi-use.

1.2 Existing File Systems

HDFS developed to support the Hadoop implementation of the MapReduce system. It offers a distributed, replicated file store optimized to support the MapReduce processing configuration. Each file is small enough to fit on a single device. Ceph also addresses this distributed computing infrastruc-

ture, but with a different emphasis. It seeks to offer scale out performance for objects across a storage infrastructure. With the rise of cloud systems using object-based storage, such as Amazon's S3, the interaction style offered by Ceph has been adopted for similar workloads. Ceph has features to handle storage devices failing and new ones joining a running system without interruption. Ceph offers a complete file system including metadata management support as well as an object system for users. GlusterFS focuses on providing a network attached storage interface to storage distributed throughout a cluster. Rather than providing metadata services, GlusterFS relies on the underlying storage file system for most basic capabilities, such as security.

Parallel file systems are optimized to support large files that must be spread across multiple devices. For example, a 100 TB file cannot fit on any current storage device and cannot be stored with any performance. Parallel file systems solve this problem by using multiple devices spread across multiple servers together as if they are a single device. Data is striped across devices, all of which can be written to or read from simultaneously. This parallel access offers aggregate performance enabling manipulating very large files with reasonable performance. Because of these characteristics, parallel file systems are deployed on most simulation intensive large scale compute systems to handle the large single object output characteristic of these applications. Lustre is arguably the most popular parallel file system appearing on a majority of the Top500 machines. IBM's GPFS is increasing in popularity as optimizations focusing on addressing big data workloads are incorporated. PVFS offers a rethinking of some of Lustre's earlier limitations to give better scalability characteristics.

Discussion

The different optimizations distributed vs. parallel file systems offer are at the cost of supporting the other kind of workload. As was mentioned above, parallel file systems aim to support very large objects and aggregate simultaneous writing and reading for a single object across the array. For workloads consisting of entirely small objects, this functionality and overhead is a cost. Similarly, the inability of the distributed file systems to handle arbitrarily large objects and massive parallel simultaneous access to a single object make them unsuitable for simulation workloads.

For both workloads, a more flexible object-based interface are being considered. This is discussed more below.

1.3 Object-Based Stores

placeholder cite [1].

Here we want to talk about how object-based key-value stores are used

for big data applications summarizing the specific features that identify this market segment.

1.3.1 Big-Data/General Computing Optimized Object/Key-Value Stores

Wisconsin [Chou, et. al] is the granddaddy 1985. Talk about others and how they differ.

memcached [2003] from LiveJournal.

1.3.2 HPC Oriented Object Stores

Parallel file systems inherently have an object-like layer beneath the surface. The requirement to spread a single file across multiple devices for capacity reasons alone prompts this approach. The actual implementation may vary, such as using individual files within a local file system, each representing part of parallel file. Popular examples include Lustre [?], ...,

1.4 Next Generation HPC Storage Systems

Here we want to talk about, at a proposal sort of level, what we think needs to be done.

Talk about the disconnect between metadata and storage and the complications it introduces and some ideas on what we plan to do about it.

Talk about the major efforts

1.4.1 Lustre/DAOS

The FFSIO phase II sort of info to start. Keep it acceptable. This seems to me to be just a pure object store.

1.4.2 Kelpie/Data Warehousing

Native multi-level key-value store

1.4.3 Hybrid Models

SSIO project from ORNL/Sandia

1.5 Conclusions

This is our overall view on things

Acknowledgements

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

1.6 Glossary

Adaptable: An adaptable process is designed to maintain effectiveness and efficiency as requirements change. The process is deemed adaptable when there is agreement among suppliers, owners, and customers that the process will meet requirements throughout the strategic period.

Bibliography

- [1] M. Ilyas, I. Mahgoub, and L. Kelly. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, Inc. Boca Raton, FL, USA, 2004.