panasas®

White Paper

Accelerating and Simplifying Apache™ Hadoop® with Panasas® ActiveStor™

November 2012

Executive Overview

The technology requirements for big data vary significantly from traditional database applications. Big data workloads strain traditional storage architectures as the data sets are unpredictable, growing at an exponential rate, while the need to keep data in a centralized repository remains constant. This requires a seamless scale-out architecture to accommodate new data sets.

Hadoop is an essential platform to support big data analytics applications. First developed by Yahoo, Hadoop is open source software inspired by the Google File System (GFS). It was designed to be highly scalable, utilizing commodity hardware, with parallel processing to achieve performance, and simple replication providing data protection. This paper will highlight the key elements of a Hadoop ecosystem focusing primarily on the Hadoop Distributed File System (HDFS).

Many of the design aspects of HDFS are fundamentally very similar to Panasas® PanFS™. Both are based on commodity storage components, leverage a modular architecture that scales linearly to thousands of nodes, employ a parallel architecture, and are optimized for big data workloads across many applications including for design and discovery. Now that Panasas scale-out NAS supports Hadoop, a single compute cluster and storage system can support many workloads including NoSQL analytics. This is particularly relevant for institutions who have already invested in compute clusters for other big data workloads as they can now run Hadoop on their existing compute infrastructure in conjunction with their Panasas ActiveStor enterprise-class storage system.

The Panasas solution does not require any additional hardware or custom software; simply follow the Panasas configuration guide to get started. There is typically no Hadoop performance degradation moving from a local disk solution to Panasas. In fact, this paper will show that analytics performance can actually be enhanced with Panasas ActiveStor storage.

Introduction to Hadoop

Apache Hadoop is an open source software framework that supports data-intensive distributed applications. Its ability to scale and support large-scale processing makes it an excellent platform for big data applications. Hadoop is based on a scale-out design that allows applications to work across large numbers of independent computers and petabytes of data. Hadoop was derived from Google's Bigtable, MapReduce, and Google File System (GFS) papers.

Hadoop Architecture

The Apache Hadoop platform consists of the Hadoop kernel, MapReduce, and HDFS, as well as a number of related projects including Apache Hive, Apache HBase, and others.[1]

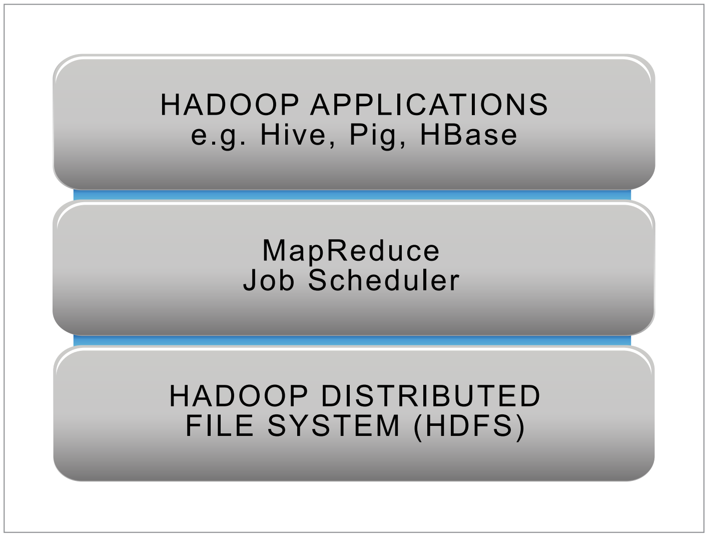

Figure 1: Hadoop framework

[1] For an in-depth overview of the Hadoop architecture, visit http://hadoop.apache.org/

Hadoop Applications

Software applications and tools that are part of the Hadoop ecosystem include:[2]

- **Avro™** - a data serialization system
- **Cassandra™** - a scalable multi-master database with no single points of failure
- **Chukwa™** - a data collection system for managing large distributed systems
- **HBase™** - a scalable, distributed database that supports structured data storage for large tables
- **Hive™** - a data warehouse infrastructure that provides data summarization and ad hoc querying
- **Mahout™** - a scalable machine learning and data mining library
- **Pig™** - a high-level data-flow language and execution framework for parallel computation
- **ZooKeeper™** - a high-performance coordination service for distributed applications

Hadoop File System (HDFS)

HDFS is a distributed, scalable, and portable file system written in Java for the Hadoop framework. Each Hadoop instance contains a NameNode and a cluster of DataNodes to form the HDFS cluster. Each DataNode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses the TCP/IP layer for communication; clients use RPC to communicate between each other. HDFS stores large files (an ideal file size is a multiple of 64MB), across multiple machines. The Hadoop framework assumes that hardware is inherently unreliable and achieves reliability by replicating the data across multiple hosts. HDFS has recently added high-availability capabilities, allowing the main metadata server (the NameNode) to be manually failed over to a backup in the event of failure. Automatic failover is being developed for future versions.

MapReduce

MapReduce is a framework for processing problems, in parallel, across huge data sets using a large number of computers (nodes), collectively referred to as a 'cluster' (if all nodes are on the same local network and use similar hardware) or a 'grid' (if the nodes are shared across geographically and administratively distributed systems).

MapReduce can take advantage of locality of data, processing data on, or near, the storage assets to decrease transmission of data.

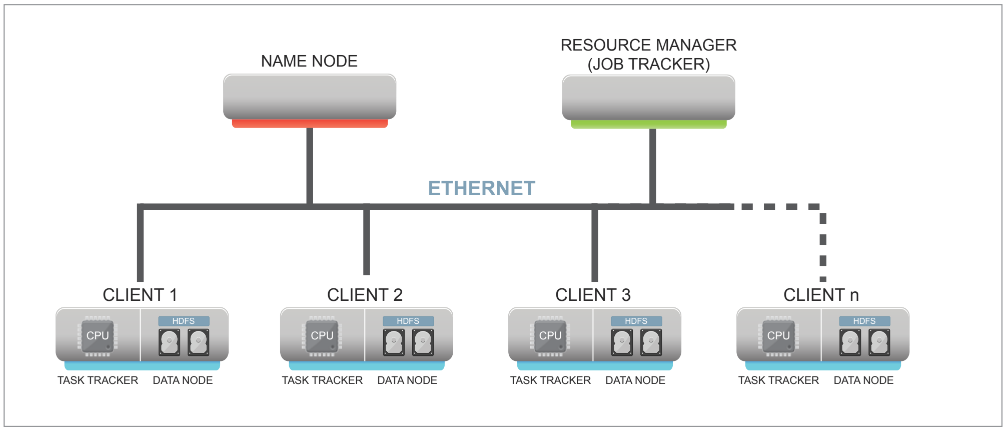

Figure 2: An architectural overview of the Hadoop hardware platform

[2] Source: http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F

Panasas PanFS™ File System

PanFS is a highly scalable, parallel, and distributed file system designed for data intensive applications. The Panasas file system shares a common heritage with modern file systems such as the Google File System (GFS)[3] and its open source counterpart Hadoop which was developed by Yahoo and handed over to the Apache Foundation.[4] PanFS is the foundation of all Panasas ActiveStor scale-out NAS solutions. It creates a single, high performance pool of storage under a global namespace, providing customers with the capability to support even the most demanding technical computing applications and big data workloads with blazing performance.

Unlike traditional NAS systems that place disks behind individual RAID controllers, PanFS uses parallel access to object storage devices (OSDs), per-file RAID, distributed metadata management, consistent client caching, file locking services, and internal cluster management to provide a scalable, fault tolerant, high performance distributed file system. The clustered design of the storage system and the use of client-driven RAID provides scalable performance to many concurrent file system clients through parallel access to file data that is striped across OSD storage nodes.

PanFS and HDFS Compared

PanFS differs from HDFS in three key ways: file access, data availability, and hardware architectural approach. For file access, HDFS implements a read-only model once a file is written to disk. While files can be mutated, they cannot be overwritten. In addition, while files follow common file system commands such as create and delete, HDFS is not POSIX compliant. Consequently, data within a Hadoop ecosystem cannot easily be shared with other applications (almost all of which assume POSIX compliance for data access), a significant problem in shared workload environments. In comparison, PanFS is POSIX compliant with data stored and accessed in a way that makes it compatible with all industry standard file systems and applications.

The second key difference between the two file systems is how data availability is maintained. HDFS relies on simple replication on commodity hardware to ensure data availability. The generally accepted level of fault tolerance for a Hadoop environment is replication level 3, which means the data can survive two drive (or DataNode) failures without data loss. This means that a reliable Hadoop infrastructure based on HDFS requires three times the physical disk capacity and three times the power and cooling to store any data set. In contrast, PanFS employs distributed Object RAID which typically occupies no more than 20% more disk capacity than the data set. Dual parity protection takes the place of triple replication. PanFS performs RAID recovery in parallel by leveraging a cluster of metadata managers, with RAID rebuild rates scaling linearly with the scale of the storage system.

The third key difference between HDFS and PanFS concerns the underlying hardware assumptions made in the file systems. HDFS was designed to use commodity servers with internal, direct-attached storage in a distributed environment. This model allows companies like Google and Yahoo to cost effectively mine data on their Hadoop clusters, while providing the scalability and performance levels required. While this approach is probably appropriate for large-scale, dedicated environments like these big Web 2.0 companies, it is accompanied by a significantly higher management burden. A bigger problem is inflexibility when it comes to varying the amount of compute vs. storage – with HDFS you have to add storage as you scale the computational capability.

[3] The Google File System Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, Google
http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/gfs-sosp2003.pdf

[4] Panasas founder and Chief Scientist, Dr. Garth Gibson is cited in the Google File System paper for his work on the NASD architecture which "GFS most closely resembles" reference# [4]

PanFS, on the other hand, is a scalable network file system that remains independent of the compute cluster, providing total flexibility for the user. In addition, the compute clusters can be utilized for many big data workloads, not just Hadoop. These advantages give IT administrators total control over their big data environments.

In summary, Panasas PanFS and Hadoop HDFS share many design goals. They both have a scalable architecture using parallel I/O to deliver high aggregate performance to many clients. This is achieved in a common way, by separating the data path from the metadata management ensuring maximum overall system performance.

However, HDFS was designed for an application-specific infrastructure with dedicated hardware, software, and IT personnel. In contrast, PanFS supports a wide range of applications, protocols, and networks managed by general purpose IT personnel. As a result, Panasas ActiveStor appliances support industry standard protocols, a POSIX compliant file system, and an easy to use management interface with enterprise-level data protection.

Hadoop with PanFS

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. Hadoop was designed to scale up from single servers to thousands of machines. Since Panasas ActiveStor storage running PanFS was also designed for that scale, it is a particularly good fit for Hadoop environments.

With Panasas, Hadoop workloads on compute clusters can seamlessly access ActiveStor appliances without the need for any additional Hadoop plug-in modules, rewriting any part of HDFS, or treating HDFS as a network storage protocol (as has been implemented by other network attached storage companies).

The following diagram provides a high level architectural view of a Hadoop environment using Panasas ActiveStor appliances for data storage:

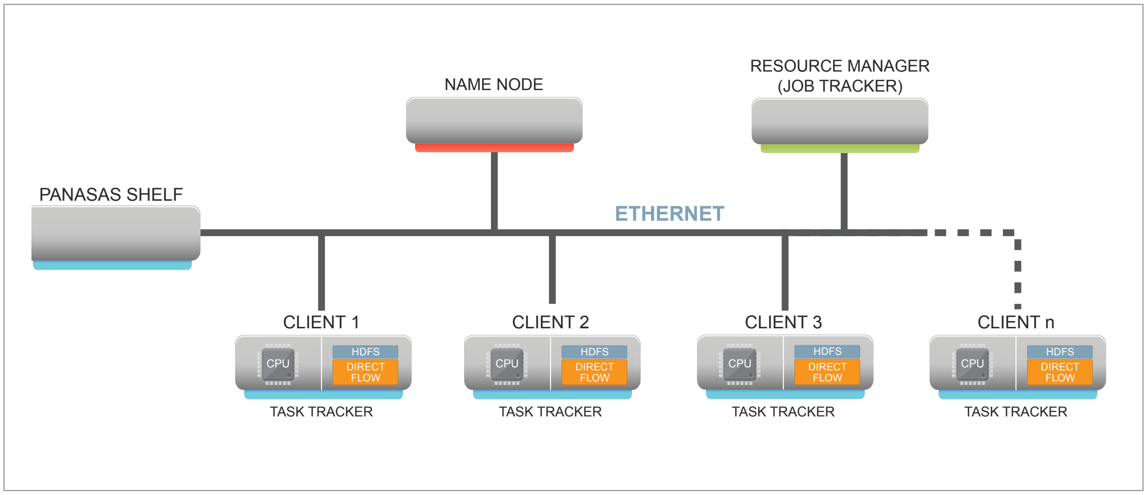

Figure 3: Hadoop cluster leveraging Panasas network attached storage

Hadoop Performance with PanFS

Panasas supports two configurations: 1) Hadoop workloads utilizing the HDFS file system to access Panasas storage and 2) Hadoop workloads running directly on the Panasas PanFS file system. To ensure consistency of comparisons, the following benchmarks utilize HDFS to communicate to disk.

This section compares the performance of Hadoop TeraSort running on a dedicated Hadoop cluster accessing local disk through HDFS, a Hadoop cluster accesses data through HDFS to an ActiveStor appliance across an Ethernet network.

Since these benchmarks were completed using Hadoop HDFS, the data was stored in a proprietary format on the Panasas ActiveStor appliances. Should a user require the ability to store data in a POSIX compliant format, the Panasas implementation can leverage the Panasas PanFS file system instead of HDFS.

TeraSort Benchmark Suite

TeraSort from the Apache Hadoop Foundation is the most commonly cited benchmark for Hadoop workloads. The benchmark suite simulates a Hadoop workload and records the total time for completion of the workload. It consists of three MapReduce applications: [5]

- TeraGen: a MapReduce program to generate the data
- TeraSort: samples the input data and uses MapReduce to sort the data
- TeraValidate: a MapReduce program that validates that the output is sorted

The sum of all three benchmarks provides an overall TeraSort result which represents the total completion time for the workload. The lower the TeraSort number, the faster the completion time.

TeraSort Benchmark Results

Panasas completed the benchmark comparing a local disk implementation with replication level 2, to Panasas ActiveStor with replication level 2. Appendix 1 provides a detailed overview of the benchmark platform.

The results show that ActiveStor performed the Hadoop workload 29% faster than a local disk implementation, completing the benchmark task in 27.3 minutes compared to 38.4 minutes for local disk.

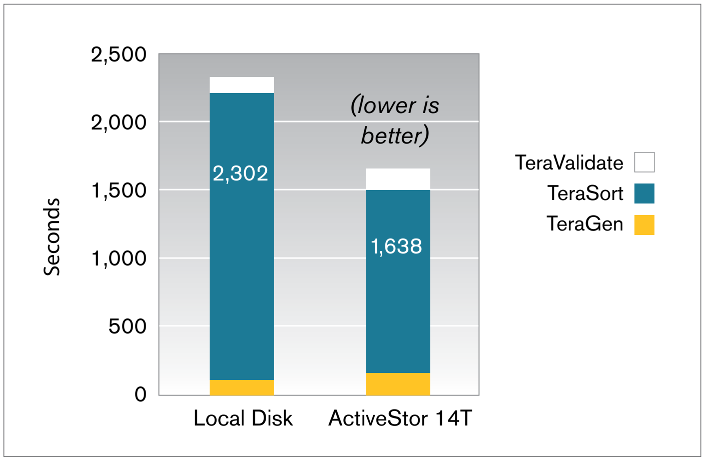

Figure 4: Comparison of local disk to Panasas ActiveStor 14T at replication level 2

As stated earlier, many Hadoop environments on dedicated clusters accessing HDFS with local disk run at replication level 3 in order to tolerate up to two DataNode failures. Using Panaas storage instead of local disks actually improves cluster reliability and availability because the local drives represent the single largest cause of node failure in the dedicated Hadoop cluster model. In contrast, PanFS protects the data with dual parity protection on top of any additional replication performed by HDFS. As a result, it is also appropriate to compare the performance of HDFS on local disk with replication level 3 to the performance of HDFS on ActiveStor with replication level 2. The Panasas solution would be expected to yield significant additional performance improvements in this case.

[5] Source: http://hadoop.apache.org/docs/r0.20.1/api/org/apache/hadoop/examples/terasort/package-summary.html

Summary

Panasas ActiveStor running the PanFS file system is an ideal storage solution for customers with varied big data workloads. Its unique, modular, scale-out architecture is ideally suited for Hadoop workloads. The Panasas solution provides significant flexibility and cost savings for IT administrators as it utilizes existing compute clusters and storage resources to run Hadoop workloads. By contrast, dedicated Hadoop appliances require additional investment in IT personnel, service contracts, and hardware. With ActiveStor, no custom Hadoop software is required and any version of Apache Hadoop can be deployed.

Panasas scale-out NAS delivers high quality enterprise-level data protection for all Hadoop data by leveraging the Object RAID functionality in PanFS. When accessing PanFS through HDFS with Hadoop, replication levels are typically set anywhere from 1 to 3 depending on customer requirements for high availability in the compute clusters. At replication level 1, the Panasas system still guarantees the data is protected, however the cluster can fail. At replication level 2 as shown in the benchmark, the compute cluster can survive a hardware failure and the data is further protected, allowing for multiple drive failures.

In addition to high performance, the Panasas solution ensures total flexibility for customers who have already invested in a high performance compute cluster. With ActiveStor, a single compute cluster can support all big data sets, no matter the application. Panasas gives the user the flexibility to use either Hadoop HDFS or PanFS as the primary Hadoop I/O path, with the latter providing a POSIX-compliant data layout for improved data flexibility. Finally, compute and storage resources can be scaled independently, allowing total hardware configuration flexibility to address big data workloads.

Finally, the Panasas Hadoop environment is extremely easy to configure—simply follow the Panasas configuration guide to get going.

Download the configuration guide HERE, or scan the QR code.

Compute Cluster Configuration	
Client Hardware	Supermicro Quad Clients X8DTT-HIBQF
Client Operating System	CentOS 5.3 (Linux ca-quad-221 2.6.18-128.el5)
Number of Clients	23 (1 NameNode, 1 ResourceManager/JobTracker and 21 DataNodes/slaves)
Memory per Client	24GB per client
CPU	Dual Socket Hex core Westmere (X5650 @ 2.67GHz)
Number of Local Disks	3 per client
Disk Write Cache [6]	Enabled

Panasas Configuration	
PanFS (3x AS14T [1+10])	5.0.0-725714.82
DirectFlow	4.1.3-702504.9
Failover	Enabled
Active Capacity Balancing	Enabled
Data Protection	Object RAID
Networking	10Gb Ethernet
Switch	Force10 S4810

Hadoop Test Environment
TeraSort Suite Hadoop 2.0.0 alpha Replication level 2 Data set size 274,887,906,900 bytes (approx 275GB)

[6] Panasas tuned the local disk environment to the best performance possible by enabling disk write cache on the compute cluster. Without this, performance on the local disk implementation would have been degraded significantly. This feature is typically disabled in a Hadoop cluster because when it is enabled power failure can cause data loss if cluster node power is interrupted for any reason. Built-in battery protection in Panasas storage means that ActiveStor is not exposed to this problem.