ADURANT
Technologies

HADOOP ON ORACLE ZFS STORAGE

A TECHNICAL OVERVIEW

ADURANT Technologies
757 Maleta Lane, Suite 201
Castle Rock, CO 80108

Brett Weninger, Managing Director
brett.weninger@adurant.com

Dave Smelker, Managing Principal
dave.smelker@adurant.com

Kevin Bujold, Principal Consulting Architect
kevin.bujold@adurant.com

ORACLE Gold Partner

Background

Hadoop, as the leading Big Data Technology, is one of the most ground breaking innovations in Information Technology to date. The current explosion in growth of services, applications, analytics, and user engagement, driven by the affordable retention of data, is unprecedented and is rapidly changing industries. The impending growth is going to burden and challenge today's conventional wisdom about scalability and performance.

With 2.7 Zettabytes (ZB) of Data globally today growing to 50 ZB by 2020, ADURANT Technologies understands the complexities of managing this explosive opportunity. Currently only 3% of data has been flagged for analysis and 1% of the 2.7 ZB has actually been analyzed. This equates to 2.6 ZB of data that needs to be analyzed today and if methods are not improved by 2020 over 48 ZB will be waiting in data lakes for analysis. As companies scale out on the traditional Hadoop reference architecture, they will hit the logical limits of network bandwidth and processing capacity. Unless solutions are created to process data faster and to store higher volumes of data without impacting overall node cluster capacity, organizations will quickly hit the limits of the current Hadoop Reference Architecture.

Nature of Hadoop

Hadoop is a massively parallel processing and data retention solution. Massive, of course, is a relative term. As compared to multi-terabyte relational databases, the petabyte deployments on Hadoop seem massive. When you consider that the largest companies are converging on data lakes that are 3 factors of 10 greater than petabytes, the volume of data is staggering. Anytime we deal with logarithmic growth, the volume of that growth can seem incomprehensible. As engineers, we are compelled to deal with these issues as we lead our organizations through technology opportunities and examine choices, risks, and costs.

Hadoop is typically characterized as a massive I/O platform. It was designed to address a significant drift between the I/O access to disk versus improvements in CPU and Memory. Hadoop was incubated as a method to address this divergence between processing and I/O. This technology allows you to read and write to all drives within a cluster in parallel. However, as a Technologist looking forward to breaking the Petabyte and the Exabyte barriers, it becomes clear that Hadoop has a natural boundary

value defined by the relationship between the number of processing nodes, disks, and network bandwidth. What does this mean? It means that as you process 10's, 100's or even the low 1000's of terabytes of data, Hadoop scales quite well on the current reference architecture. However, as you start to exceed the low 1000's of terabytes, you start to hit network bandwidth as a limiting boundary.

This is seen at scale quite quickly when unfunded Hadoop POC's go from the Test and Development Stage to Production. Many organizations start with a few nodes as an exercise to prove the value of Hadoop. The business units quickly recognize value from previously untapped data and drive usage of the environment. Starting out at a few Terabytes, a 1 Gigabit network topology works quite well. Once you start to exceed several hundred Terabytes of processing, organizations quickly find that they need to upgrade their network topology from 1 Gbps to 10 Gbps. Yet again, as usage exceeds a complex mix of performance and capacity, organizations are driven to trunking multiple 10 Gbps, or using more appropriate solutions like Infiniband. As the data lake grows, the number of nodes required grows. At some point, the amount of data movement required to keep the cluster in sync exceeds the network bandwidth. Understanding the future growth and the limitations within the standard Hadoop Reference Architecture, ADURANT Technologies in conjunction with the Oracle Solutions Center worked to address these problems with the Oracle ZFS Storage Appliance.

The Oracle ZFS Storage Appliance's inherent benefits as it relates to Big Data include I/O performance, compression, large block sizes, advanced analytics, and cost. The Oracle ZFS Storage Appliance derived its name from the Zettabyte File System that was developed as part of Solaris. The ZFS filesystem in conjunction with appropriate amounts of cache, SSD, and most importantly DRAM have allowed the Oracle ZFS Storage Appliance to deliver I/O in nanoseconds. Based on the sheer speed of I/O delivery, we hypothesized that the ZFS Storage Appliance could stay ahead of the scalability curve and the inherent constraints of the current Ethernet fabrics. In addition to processing performance, the native compression of the ZFS Storage Appliance would provide inherent operational expense improvements by reducing space, power, and cooling costs for such large environments. In conjunction with the Oracle Solutions Center, ADURANT Technologies tested this theory and validated it as fact.

Reference Architecture

A control group of six-nodes was deployed under the standard Hadoop Reference Architecture of systems with JBOD (just a bunch of disk) as the control group. Subsequently, the same cluster was leveraged against the ZFS Hadoop Reference Architecture outlined below.

Hadoop Reference Architecture

Conventional wisdom related to Hadoop Reference Architecture states that commodity x86 systems with JBOD are the most appropriate platform from a cost, scalability, and performance perspective. As you need more processing power and/or capacity, you add more nodes. Typically employing high quality CPU's and large amounts of memory do not provide increased performance or significant economic benefit. Generally, the greater the spindle count, the lower the cost per Terabyte. As a rule, more "average nodes" are better than fewer "super nodes." This is a function of Hadoop being primarily I/O bound for almost all systems technologies. Therefore, it follows that the most economical way to add the greatest amount of drives results in the most optimal ROI.

ZFS Hadoop Reference Architecture

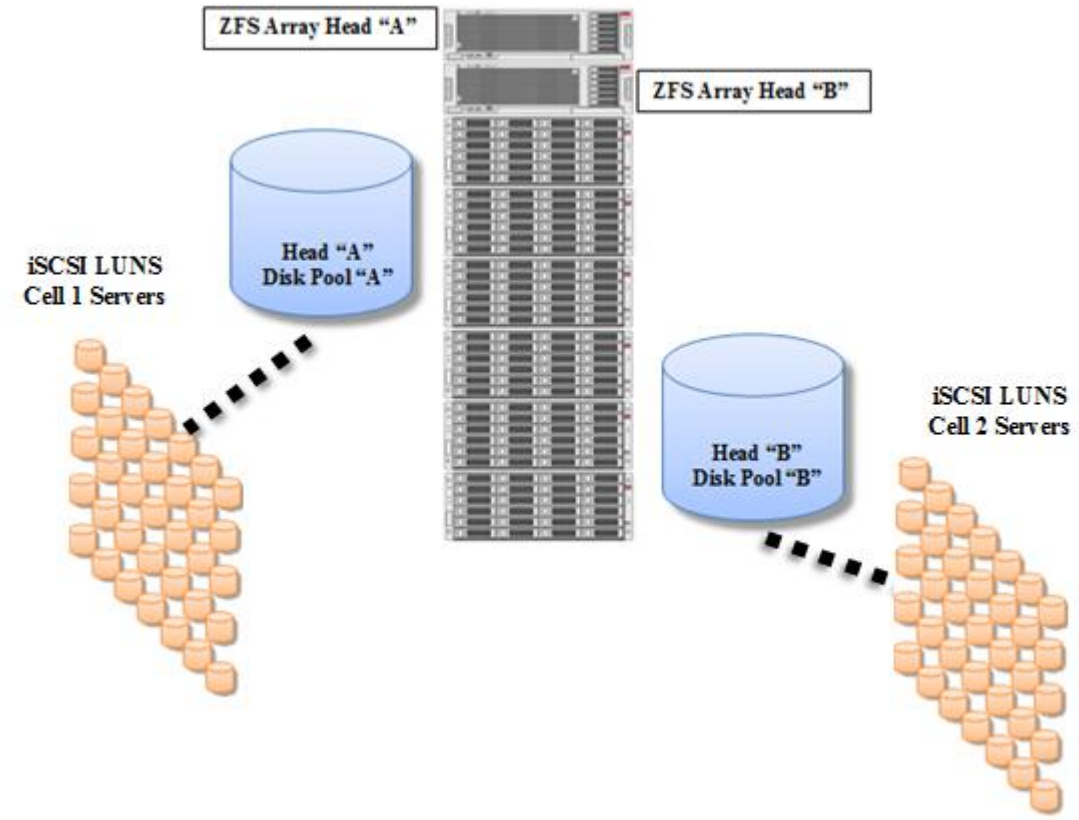
The ZFS Hadoop Reference Architecture was deployed leveraging the same six-node cluster, as well as an Oracle ZFS ZS3-4 Storage Appliance. iSCSI was used for the fabric to connect each of the server. This iSCSI SAN was designed to include multipath redundancy and load balancing leveraging the Oracle 72P 10GbE Switch.

The servers were separated into two logical cells: Cell 1 and Cell 2. Hadoop servers can be configured to be "rack aware." This rack awareness allows the Hadoop cluster to distribute replicated data blocks between only those servers in different racks. To ensure availability, we leveraged this feature to provide appropriate redundancy between the ZFS ZS3-4 storage heads. Subsequently, the LUN's were then mirrored to provide an even higher level of protection than what is seen in the Hadoop Reference Architecture, while aligning with the availability architecture of the ZS3-4. With the benefits of LZJB, this provides a higher level of compression while driving the cost per TB well below that of the Hadoop Reference Architecture.

The benefits of this configuration include:

- Reduced Hadoop cluster overhead by reducing the replication factor to 2x
- Reduced storage (disk space) requirements by reducing the replication factor to 2x
- Increased the number of copies of data to 4x via the ZFS Storage Appliance
- Added data compression via the ZFS Storage Appliance
    - Further reducing storage space requirements even in a mirrored pool configuration
- Added read and write caching via the ZFS Storage Appliance decreasing I/O response times
- Added data protection (RAID 1) with no added overhead to the Hadoop cluster
- Added fault tolerance via the ZFS Storage Appliance's clustered heads

Below is a diagram of this ZFS Hadoop Reference Architecture:



Below are the specifics of both Reference Architectures:

Reference Architecture	Standard Hadoop	ZFS Hadoop
Hadoop	Cloudera 5.1.3 Name Node 5 Data Nodes	Cloudera 5.1.3 Name Node 5 Data Nodes
Servers	(6) x86 Servers Oracle Linux 6.3 (2) Intel Xeon 10-core 3.0 GHz proc's 128GB Memory (DDR3-1600)	(6) x86 Servers Oracle Linux 6.3 (2) Intel Xeon 10-core 3.0 GHz proc's 128GB Memory (DDR3-1600)
Storage	(12) 4TB 3.5-inch SAS-2 HDD	ZS3-4 (Clustered) • 2TB DRAM • 6 Shelves • 900GB 10K RPM HDD • iSCSI Fabric

Evaluation Process

The evaluation process included a two-phase approach of synthetic testing leveraging known benchmarks and testing against real world data running MapReduce jobs of differing complexities.

Terasort Benchmark

Benchmark testing was completed using primarily Terasort, which is a suite of benchmarks including Teragen, Terasort, and Teravalidate.

- **TeraGen** is a map/reduce program to generate the data
- **TeraSort** samples the input data and uses map/reduce to sort the data into a total order
- **TeraValidate** is a map/reduce program that validates the output is sorted

Additional details about Terasort can be found under the Terasort Apache Project. This process included Terasort regression testing at 10 GB, 100 GB and 1 TB quantities.

Real Data Testing

For Real Data Testing, FASTA formatted publically available bioinformatics data was used. The FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. The format also allows for sequence names and comments to precede the sequences. This type of formatted data is the industry standard in the area of bioinformatics. This data is in a semi-structured state and does not permit sequence annotation. These files are mainly intended for use with local sequence similarity search algorithms. Each directory has a README file with a detailed description of the header line format, the file naming conventions, general file types including DNA, coding sequences (CDS), cDNA, peptides, and RNA.

This data was loaded into Hadoop under different directories based on the file type. 3 types of MapReduce processes of varying complexity were executed via Hive:

- Simple - The first process was a very straightforward process that grouped the data on the species and counted the number of distinct DNA sequences.
- Medium - The second process was a little more complex. It took 2 of the data sets and joined them. This process summarized the datasets prior to joining the data, which lead to the execution of 3 MapReduce jobs: 1 on each summation and one for the join.
- Complex - The final process joined the raw data together and then performed the same aggregation. By doing this, the MapReduce processes were forced to pull the full data sets for both data sets prior to performing the reduction. This forced a much larger initial MapReduce and the final MapReduce step brought the data down to the final output. The final process ran 5 map/reduce jobs. This Test was designed to be inefficient in order to push the limits of Hadoop and both Reference Architectures.

Regression testing was conducted with 1.6TB of data for each type of MapReduce process using Hive to compute the results.

Test Results

The Test results were positive for the Terasort tests and even more dramatic with real-world data. The I/O channels, ZFS storage and system resources were monitored and compared. The standard Hadoop Reference Architecture saw typical behavior where CPU and Memory usage were low and load was spread evenly across the servers. However, on the ZFS Hadoop Reference Architecture CPU resources were steady at 85-100% usage, Memory utilization was nominal (less than 5%), I/O on the channels was less than 5%, and cache hits were nearly 100%. The implication of this performance profile is that you can add CPU/Memory to address load requirements instead of spindles to improve performance.

Terasort Findings

The Terasort findings resulted in the ZFS Hadoop Reference Architecture consistently outperforming the standard Hadoop Reference Architecture by over 10%. Also, as the amount of data for the Terasaort increased, the ZFS performed even better which was expected behavior.

Benchmark	1TB Terasort Tests		Performance Improvement
	Local (s)	ZFS (s)	
Teragen	1148.5	1015.2	13%
Terasort	10102.9	8844.3	14%
Teravalidate	192.2	164.0	17%

Real Data

The Real Data findings resulted in the ZFS Hadoop Reference Architecture clearly outpacing the standard Hadoop Reference Architecture. Again, as the amount of data for the Real Data testing increased, the ZFS response improved which was expected behavior.

As stated previously, CPU resources were being utilized in excess of 90%, I/O channels were less than 5%, and ZFS Cache hits were nearly 100%.

MapReduce Job (1.6TB Hive)	Completion Times		Performance Improvement
	Local (s)	ZFS (s)	
Simple	7978.1	2510.9	318%
Medium	8970.6	2994.2	300%
Complex	14121.2	5854.8	240%

*ZFS Performance Statistics were completed at a **minimum of 3.5x compression**

Additional testing against the Simple MapReduce Jobs using Impala was completed with results yielding processing of 10.1TB/hour with only 5 data nodes. At 40 data nodes this would be roughly 80TB/hour or **almost 2 PB per day at 3.5x compression**!

Conclusions

The findings of the Hadoop ZFS Proof of Concept testing clearly indicate that the ZFS Storage Appliance is more than able to handle current Hadoop workloads. Data processing was CPU bound, memory utilization was nominal, I/O utilization was nominal, and data was compressed by a minimum of 3.5x. As a more tangible comparison, with a 40 Node Cluster the ZFS Hadoop Reference Architecture can process nearly one-third Petabyte per day for MapReduce jobs of Medium complexity on Hive.

The ZFS Reference Architecture provides several compelling business opportunities to not only improve performance, but significantly reduce operating expense. Servers can be reduced by approximately 66% for both processing and storage. This translates to a significant reduction in cost for server hardware, software support, space, power, cooling, networking, and administration. Additionally, because the storage is being used for storage and not data processing, you can right size a cluster by adding another array to the existing cluster to scale out storage, thereby making the ZFS Hadoop Reference architecture the first Exabyte processing platform. For additional details please contact sales@adurant.com and/or brett.weninger@adurant.com.