Taming the Long Tail in Apache Hadoop®: Storage Tiering for Big Data

This white paper explores how tiered storage can help organizations store and analyze more data to glean deeper insights—at a realistic total cost of ownership (TCO).

Contents

Introduction

More data analyzed over longer time horizons can lead to breakthrough insights. However, keeping everything on traditional Hadoop clusters leads to massive server sprawl and high operating expenses (OpEx). For this reason, older data sets are often discarded or sometimes archived to tape, then reloaded later for analytics. But in this scenario, reloading all archived data for processing is an impossible task if the Hadoop cluster has not grown with the size of the archive. So what ends up happening is most historical analytics tasks just take a few data sets from a few tapes, hoping that the analytics will be "good enough."

This white paper explores how tiered storage can help organizations store and analyze more data in order to glean deeper insights—at a realistic total cost of ownership (TCO).

Hadoop Then and Now

Hadoop was developed over 10 years ago, when performance was limited to small capacity HDDs that rarely exceeded 2TB. Furthermore, every DataNode had to match its counterpart in number and type of HDD. Recent, game-changing innovations like HelioSeal® technology HDDs and enterprise-grade SSDs enable management of the three major stages of Hadoop workloads (Load, Analytics and Store) using different types of drives, replication factors and data management practices.

With the traditional symmetrical Hadoop design, server sprawl is nearly instantaneous. To make matters worse, OpEx to power all of the DataNodes grows in direct proportion to the daily load.

By leveraging Hadoop's storage policies, DataNodes can be optimized with SSD and HDD tiers for different workload requirements like Throughput and IOPS, while increasing the HDD sizes to store more data for longer periods of time.

By adding tiering, organizations can now **process more data and keep it longer**, at a lower cost than ever before.

Keeping and Analyzing More with Tiered Storage

Comparing a traditional symmetrical design to a tiered storage architecture shows just how far the Hadoop community has come in 10 years. By adding tiering, organizations can now process more data and keep it longer, at a lower cost than ever before. Consider the following architectures in Figure 1.

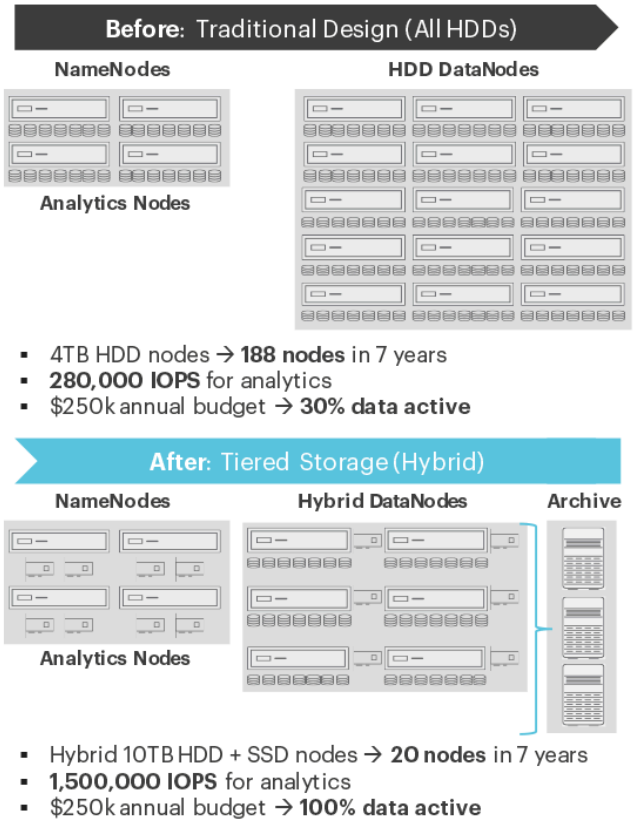

Figure 1: Traditional design vs. tiered storage architecture

The "After" architecture uses NVMe™ SSDs on NameNodes to accelerate all command and control functions.

Hadoop now defines an SSD tier for DataNodes, where SSDs can be allocated for Load and Analytics functions using a replication factor of 1. Depending on the workload, different mixes of SSDs and HDDs can be selected to optimize performance, capacity and cost.

There are commands to move data from DataNodes to Archive. This means that the number of DataNodes can be sized to match processing requirements, not storage requirements.

With the Archive function, a 3-way stripe of the data is no longer needed. Erasure Coding and HDD-based Object Storage, like HGST ActiveScale™, have a data durability of up to 17 9s so the replication factor can be set to 1.

An SSD/HDD Hybrid DataNode does more work while using less space, power and compute resources. With high capacity HDDs, the tiering approach allows for

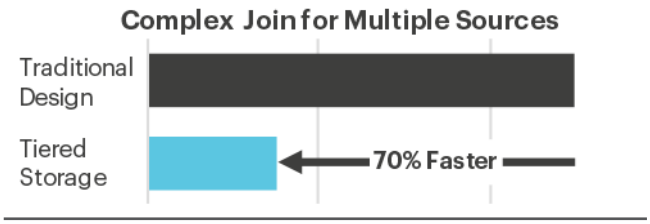

Figure 2: Comparison of complex join analytics function [1]

large data sets to be kept online longer, for a lower cost. The SSD tier accelerates Load and Analytics jobs. Processing data faster means users can increase the size and frequency of workloads as well. Figure 2 above provides an example of how SSDs in a Hybrid DataNode can help process workloads much faster than the traditional design.[1]

Hadoop Workflow: Load, Analytics and Store

Given the rapid innovation in storage technologies, the three major stages of Hadoop workloads (Load, Analytics and Store) can be managed using different types of drives, replication factors and data management practices.

Load Data

By striping across many DataNodes, Hadoop can achieve massively parallel loading. Lots of striped HDDs deliver excellent throughput, and adding DataNodes scales out in a near-linear fashion. But an all-HDD DataNode is impacted by mechanical seek time which can increase a large data set's load time as much as 40% based on the number of drives, type of drive and block size of the chunks to be loaded and especially any other concurrent operations on the Hadoop cluster.

While SSDs are known to shine in I/O-centric applications, they also deliver for sequential write/read workloads. A single NVM Express™ (NVMe) compliant SSD can deliver up to 2,200MB/s of sequential write throughput, or the equivalent of around 12 HDDs. For reads, it takes nearly 30 HDDs to match the 6GB/s performance of an NVMe SSD.

Now consider the rapid ingest of daily content that can be processed, then moved to HDD tiers, freeing up space to do other processing against a variety of batch loads on that same SSD tier simultaneously. Suddenly

1 https://www.usenix.org/system/files/conference/lisa14/lisa14-paper-kambatla.pdf

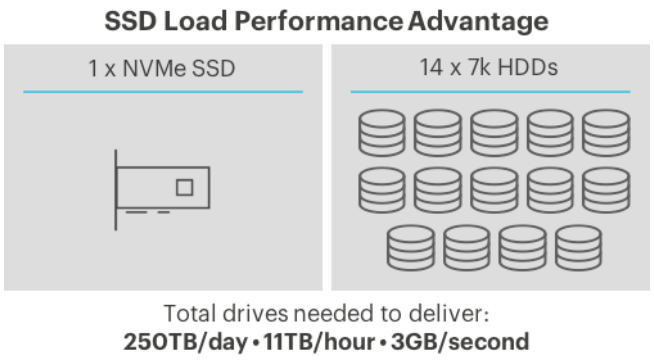

Figure 3: Number of drives needed to load 250TB in 24 hours or less

your infrastructure becomes much more efficient to load and process without compromise for data volume, variety, or velocity.

With Hadoop batch-based loading, users can now load to an SSD tier with Replication Factor_1, run processing and then de-stage to HDDs or ActiveScale at a later time. Figure 3 above shows an example of how efficient an SSD is for loading data.

Analytics to Get Insights

Harnessing the parallel nature of multiple DataNodes to process large amounts of data, analytics like Apache Spark™ or MapReduce are used to find the needle in the haystack. These tools generate two kinds of I/O patterns—large sequential reads and writes as well as small random reads and writes.

The small block size and random nature of shuffle is an ideal use-case for SSDs. According to Cloudera[2], SSDs can accelerate Analytics workloads by as much as 70%.

If one had an unlimited budget, hundreds of machines could be used to process 300TB in under an hour. In fact, Google has proven that 1PB can be sorted in 33 minutes using 8,000 computers.[3]

However, this is not a reality for most. With many different input sources and data types all needing to be compared, SSDs can have a tremendous impact for the majority. A single NVMe SSD can perform up to 1,200,000 random 4K reads, or ten times the IOPS that can be achieved from about 100 fully-loaded 2TB HDD-based DataNodes.

Deeper Insights with Active Archives

Large internet search, social, and cloud service providers collectively have Exabytes of data in Hadoop clusters. But what about the rest of the world that wish to manage Big Data in Hadoop but on a fixed budget?

By leveraging storage policies, I/O can be managed on SSDs and HDD sizes can be increased to keep more data for longer periods of time.

The Long Tail[4] is a theory on how large amounts of data, kept for long periods of time, analyzed with Bayesian classification can lead to disruptive marketing with lower competition, higher prices/margins with lower marketing costs. This is a perfect problem for the Hadoop framework as users must iterate against a large population to find the specific insight.

If the project is budget constrained and all of the available data cannot be stored, the accuracy of Long Tail analytics becomes questionable.

The larger the sample size (p) of a given population (n), the better the odds of accurate results. Margin of Error describes the odds.

At 1TB/day over 7 years and with a budget of $250K/yr, 2TB DataNodes will only be able to store 30% of the statistical sample. 4TB DataNodes will store 50% of the sample, compared to 100% with an ActiveScale system and the storage tiering approach. At 100% of sample and 100% of population, there would theoretically be a zero margin of error, yielding the deepest and most accurate insights (see Figure 5).

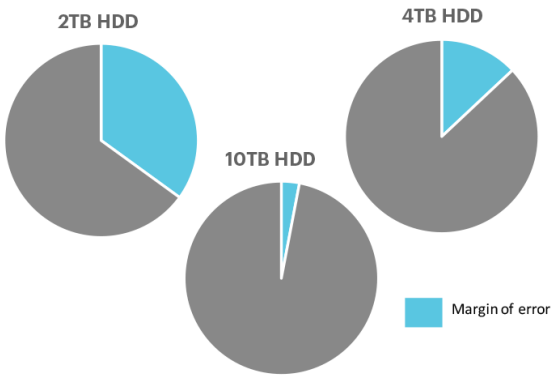

Figure 5: Statistical accuracy (margin of error) based on volume of data stored

2 http://blog.cloudera.com/blog/2014/03/the-truth-about-mapreduce-performance-on-ssds/
3 https://research.googleblog.com/2011/09/sorting-petabytes-with-mapreduce-next.html
4 http://www.longtail.com/about.html

Total Cost of Ownership

Sprawling DataNodes, CapEx and OpEx Implications

A small daily Load of 1TB yields 1.2PB a year with Hadoop striping and space allocation. Without proper planning, DataNodes quickly sprawl into an operational nightmare.

Assuming a $164/device list price for 2TB HDDs[5], storage spend is $98,400 in Year 1. Adding an average "commodity" server for every 10 HDDs at $5,925/unit[6] yields another $355,500 for capital expenses (CapEx) for a total spend of $453,900 in Year 1.

Alongside CapEx, there is also OpEx to consider, which includes maintenance, admin costs, power and cooling. Bottom line, the combined cost of CapEx and OpEx means really big data problems.

"Store" is the most expensive component of the Hadoop framework. Adding more DataNodes as the volume of data increases may not be the best solution for budget-conscious Hadoop customers. But with SSDs that can deliver very high I/O and multi-core servers that can handle CPU problems, a symmetrical architecture may not be necessary.

Hadoop has a set of commands to move data from DataNodes to an Archive tier. This means that we could potentially limit the number of DataNodes to the size of the maximum expected Analytics function (perhaps six months of data). This would cap the total DataNodes (and associated OpEx) moving the long-term storage cost to a more appropriate platform.

What is an Archive Tier?

An Archive tier is simply an administrator-specified HDFS directory a shown in figure 6. Batch or interactive jobs move their completed results to the directory and the HDFS mover process will transparently migrate data from SSD or HDD to the archive.

Hadoop Code Snippet

```
hadoop archive -archiveName foo.har
-p /user/hadoop -r 1 dir1 dir2 /user/zoo
```

Figure 6: Code snippet to create a Hadoop archive using /user/hadoop as the relative archive directory

5 Source = Newegg HGST 7K4000 2TB as of 2/6/2016
6 Source = Dell.com (R730 w/ 12 HDD Slots, Dual Xeon 2.6GHz with 18GB RAM) as of 2/6/2016

Seven Year View of Data Retention

Data retention for seven years matches up with many regulatory requirements. Table 1 below outlines how much data is accumulated each year, with 1TB of daily Load and where the replication factor to Archive = 1 unless otherwise specified.

Year	Data Growth (TB)	Kept on DataNodes (TB)	Moved to Archive (TB)
1	1,095	1,095	--
2	2,190	2,190	--
3	3,285	3,285	--
4	4,380	3,285	**1,095**
5	5,475	3,285	**2,190**
6	6,570	3,285	**3,285**
7	7,665	3,285	**4,380**

Table 1: 1TB of Daily Load Accumulated Over Seven Years

Depending on the Hadoop system architecture that is used, the TCO to store and use all of this data varies significantly, as shown in Figure 7.

Across the board for both CapEx and OpEx, a Hybrid tiered storage system with SSDs and HDDs minimizes out-of-pocket costs, delivering to users a smarter storage solution for Big Data.

High Fidelity Big Data with a Superior TCO

While the concept of storage tiering is not new, it brings new life to struggling Big Data deployments and is the blueprint for future implementations. This is especially true for organizations with fixed budgets. As illustrated in the TCO analysis in Figure 7, the ability for a Hybrid DataNode to Load and run Analytics faster with fewer

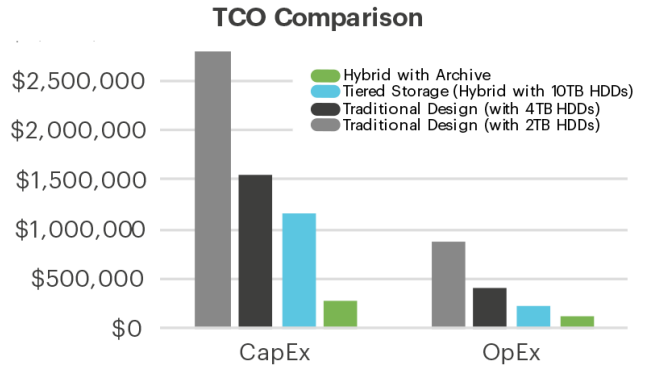

Figure 7: TCO analysis comparison

nodes, and the addition of an ActiveScale system for online archive storage is far superior.

The numbers are magnified by the following factors:

1. Fewer DataNodes in the traditional symmetrical approach yields less throughput and IOPS
2. SSDs in the Hybrid speed up certain tasks and complete many others faster than multiple HDDs
3. A significant portion of the traditional approach is dedicated to capacity storage, leaving less room for throughput and IOPS

Load, Analytics and Store deliver amazing results when properly designed. By applying storage tiering, users can optimize ingest from many sources, stage it for MapReduce or Spark, move it to an ActiveScale system for rapid access and keep more data than ever before, enabling faster, more accurate analytics and greater business insights.

To learn more, visit our website: www.hgst.com.