THE GORILLA GUIDE TO...™

Storage Designs for Big Data and Real-Time Analytics

Written by
Joseph D'Antoni,
Data Platform MVP
and James Green,
VMware vExpert

Brought to you by
tegile

Helping you navigate the technology jungle

THE GORILLA GUIDE TO...

Storage Designs for Big Data and Real-Time Analytics

Written by
Joseph D'Antoni
Data Platform MVP
and James Green
VMware vExpert

ActualTech Media

The Gorilla Guide to Storage Designs for Big Data & Real-Time Analytics

Author:	Joseph D'Antoni, Data Platform MVP, James Green, VMware vExpert
Editors:	Hillary Kirchener, Dream Write Creative
Book Design:	Braeden Black, Avalon Media Productions Geordie Carswell, ActualTech Media
Layout:	Braeden Black, Avalon Media Productions
Project Manager:	Amy Short, ActualTech Media

Printed in the United States of America

First Printing, 2016

ISBN 978-1-943952-13-7

ActualTech Media
Okatie Village Ste 103-157
Bluffton, SC 29909
www.actualtechmedia.com

About the Authors

Joseph D'Antoni,
Data Platform MVP

Joseph D'Antoni is an Senior Consultant and Microsoft Data Platform MVP with over a decade of experience working in both Fortune 500 and smaller firms. He is a Principal Consultant for Denny Cherry and Associates and lives in Malvern, PA. He is a frequent speaker at major tech events like Microsoft Ignite, PASS Summit and Enterprise Data World. He blogs about all topics technology at joeydantoni.com. He believes that no single platform is the answer to all technology problems. He holds a BS in Computer Information Systems from Louisiana Tech University and an MBA from North Carolina State University, and is the co-author of the Microsoft white papers "Using Power BI in a Hybrid Environment" and "Security and Azure SQL Database"

James Green,
Partner, ActualTech Media

James writes, speaks, and consults on Enterprise IT. He has worked in the IT industry as an administrator, architect, and consultant, and has also published numerous articles, whitepapers, and books. James is a 2014 - 2016 vExpert and VCAP-DCD/DCA.

About ActualTech Media

ActualTech Media provides enterprise IT decision makers with the information they need to make informed, strategic decisions as they modernize and optimize their IT operations.

Leading 3rd party IT industry influencers Scott D. Lowe, David M. Davis and special technical partners cover hot topics from the software-defined data center to hyperconvergence and virtualization.

Cutting through the hype, noise and claims around new data center technologies isn't easy, but ActualTech Media helps find the signal in the noise. Analysis, authorship and events produced by ActualTech Media provide an essential piece of the technology evaluation puzzle.

More information available at **www.actualtechmedia.com**

About Tegile Systems

Tegile Systems is pioneering a new generation of flash-driven enterprise storage arrays that balance performance, capacity, features and price for virtualization, file services and database applications. With Tegile's line of all-flash and hybrid storage arrays, the company is redefining the traditional approach to storage by providing a family of arrays that accelerate business critical enterprise applications and allow customers to significantly consolidate mixed workloads in virtualized environments.

Tegile's patented IntelliFlash™ technology accelerates performance and enables inline deduplication and compression of data so each array has a usable capacity far greater than its raw capacity. Tegile's award-winning solutions enable customers to better address the requirements of server virtualization, virtual desktop integration and database integration than any other offerings. Featuring both NAS and SAN connectivity, Tegile arrays are easy-to-use, fully redundant and highly scalable. They come complete with built-in snapshot, remote-replication, near-instant recovery, onsite or offsite failover, and VM-aware features.

For more information, visit *www.tegile.com*

Table of Contents

Gorilla Guide Features

In the Book

These help point readers to other places in the book where a concept is explored in more depth.

The How-To Corner

These will help you master the specific tasks that it takes to be proficient in the technology jungle.

School House

This is a special place where readers can learn a bit more about ancillary topics presented in the book.

Food For Thought

In these sections, readers are served tasty morsels of important information to help you expand your thinking.

Dive Deep

Takes readers into the deep, dark depths of a particular topic.

Executive Corner

Discusses items of strategic interest to business leaders.

1

Infrastructure Designs for Big Data and Real Time Analytics

Big Data and the Internet of Things are trendy buzzwords in information technology (IT). While these complex computer systems have both displaced and augmented traditional analytics platforms, such as data warehouses, it is important to step back and think about some of the computing trends that drove systems here.

Increasing Data Volumes

Much like Moore's Law about processing power, the volume of data stored by storage arrays has expanded exponentially over time.

Moore's Law

The observation the computer processing power doubles every two years. This was a prediction from Intel co-founder Gordon Moore, in 1965. While not a mathematical law, this prediction has held true over time.

A 2012 IDC study* mentions how between 2010 and 2020 the "universe of data," or all of the known data in the world, will increase from 300 to 40,000 Exabytes (40 billion terabytes [TB]). This growth is a confluence of several trends:

- **Machine generated data (MGD)** collected from smart devices, meters, and computers.
- **Web traffic data** generated about users who are using the internet.
- **Streaming log data** used for analytics and security threats.

Did you notice that among all of these data sources, none of this data is generated directly by humans? Since humans have to sleep eight hours a day, and have limited typing capacity, there is a limit to human generated data; there is virtually no limit to machine generated data.

Computing Platforms

As these data volumes and processing power have increased, IT infrastructure has also seen a number of key trends which support and drive this growth, including:

- Virtualization
- Cloud computing
- Solid state storage

You will learn about each of these trends in detail, but it is important to note that the aforementioned increases in processing power and memory density are what have enabled each of these trends.

*http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf

Memory allows denser virtualization, caching for storage, and, in general, allows for better overall system performance.

Increases in CPU performance allow solid state storage devices to take advantage of technology, such as compression — which offers both increases in performance and storage density. Unsurprisingly, these trends drive each other from a computing perspective.

Another thing to keep in mind is that for many computer systems running on modern CPUs with higher memory density, they are still bottlenecked at the storage layer because of a reliance on legacy-based spinning disk devices. These spinning disk devices do not provide the input/output operations per second (IOPs), throughput, nor the latency to match the power of the CPU. Solid state storage remedies many of these problems.

Storage Performance

Until the advent of the solid state storage devices (SSDs), storage performance was limited by the laws of physics. Hard drives could not spin faster than 15,000 RPM, which meant that 15,000 RPM was the fastest as a computer could read data. Storage performance is measured in two dimensions:

- **IOPs.** This is how many reads or writes a given drive, or array of drives, can complete in a second. and then

- **Latency.** This is how long (measured in milliseconds) a read or write takes to complete.

  A single 15,000 RPM drive can perform approximately 200 IOPs at 4 ms of latency. A typical SSD can perform 70,000 IOPs at .1 ms latency.

Virtualization

Virtualization has a long history in computing, but x86 virtualization is the defining infrastructure trend of the 21st century.

Virtualization allows for a great deal of flexibility and its enabling of software-defined "everything" has driven the move toward cloud computing, both public and private.

Virtualization has also greatly increased pressure on storage subsystems by consolidating workloads.

Cloud Computing

Cloud computing is more than just virtualization. There is a hefty dose of automation and deployment technology, but at the end of the day, whether your cloud is public or private, your workload is running on a virtual machine in someone's data center.

For the high sustained workloads of streaming data analytics systems, public clouds may not be cost effective.

In the Book

To learn more about public and private clouds, check out Chapter 3!

Solid State Storage

The combination of increased pressure from consolidated workloads and the demand for real time analytics has also led to a need for higher storage performance. In the past, the only way to increase storage performance was to add additional spinning disk devices, which was costly from power, management, and hardware perspectives.

Luckily, there are now several drivers that have allowed solid state storage to become mainstream. The first is an overall reduction in the cost of NAND Flash. Costs have become markedly cheaper in recent years. Additionally, the increase in CPU power has allowed vendors to use technologies like deduplication and compression to allow SSD storage devices to nearly equal hard disk devices in storage density.

This increase in density and decrease in cost has made SSDs a viable option for almost all enterprise workloads, and a requirement for critical data systems.

Business Uses for Streaming Data

As data volumes increase, businesses try to gleam more and more insights from this data. Some of the initial areas of interest amongst businesses include social media data for marketing and other business campaigns.

Another area of focus is operational intelligence, combining log files from multiple systems in a common system to bring forth operational insights.

Social Media and Marketing — A Happy Marriage

Twitter turned 10 years old in 2016, and Facebook has been around slightly longer. When both of these services launched, no one expected for them to change the world the way they have. Neilsen, the TV ratings organization now collects and monitors Twitter data to get information on viewer engagement during a given show, thus offering a value added service to their customers. Likewise, even the smallest business would not dream of opening in 2016 without a social media presence and strategy.

The big difference between social media and traditional marketing approaches is that social media democratizes the ability to collect marketing analytics. In traditional marketing, in order to measure the impact of your effort, you had to conduct extensive research and surveys to come to a conclusion which may not even be accurate. With social media, the platforms offer built-in analytics modules that allow users to quickly see the impact of an ad.

Since tweets are only 140 characters, you might think that long-term storage of this data would have little impact on your overall infrastructure. However, each tweet carries with it over 150 individual metadata points (and can be up to 100 KB), meaning that just tracking tweets can cause a tremendous growth in data over time.

Operational Analytics

Real-time operational analytics is another area of streaming data. Many organizations have hundreds if not thousands of servers, switches, firewalls, and other devices running in their data centers and shop floors. Consider the Boeing 787 which generates more than 500 GB of data on a given flight. This data allows Boeing engineers to know well in advance when an operational problem arises, and possibly to have the parts in place to fix the problem before the plane ever lands at its destination.

Operational analytics systems offer a wide array of uses for many transactional systems. These systems act as real time query engines across a stream of incoming data. Whether it is looking for an unusual login pattern, or to report potentially fraudulent transactions, these systems can respond to either machine generated or data processed from human ordering systems at high levels of scale.

Being able to bring together all of these data sources across an organization can allow for much deeper insights into organizational performance.

In contrast to traditional business intelligence solutions (which looked inward), modern analytic systems give businesses the ability to look across all of their data sources and outward to social media.

Doing so, however, requires massive amounts of storage performance and effective capacity to be able to address the real time analytic needs of these modern systems.

Introducing the Internet of Things

The world is seeing more and more internet connected devices including health devices like the FitBit or Microsoft Band, smart electric meters which report back data in a more real time fashion, or a connected thermostat that can dynamically adjust based on the presence of people in a house. All of these devices comprise the Internet of Things (IoT).

While many think of the IoT's origin as being the smart phone, it really goes back to the beginning of the 21st century to supply chain logistics, particularly for large retailers. Reducing inventory costs can go right to the bottom line and help tight retail margins, so it was in the vested interest of large retail chains to manage inventory as efficiently as possible. This was further enabled by radio frequency id tags which allowed pallets of merchandise to be tracked at the individual pallet level. This required cooperation between the retailer and its wholesale vendors; however, once that was in place, the retailer could track inventory in real-time all the way though the supply chain.

As you can imagine, this resulted in vast amounts of data, especially if the retailer wanted to track any level of historical trending. Anecdotally, some retailers even had a 11 TB table in their supply chain database!

So, when you think about expanding the paradigm to a consumer driven model, such as in electricity, or even mobile devices, the numbers can grow exponentially. Storing and retrieving this data can quickly become problematic. Modern data warehouse and streaming analytic technology can support writing this data efficiently, even with some benefits to be gained around compression at the data tier. Compression techniques can allow data to be read faster and in some cases can even enhance the performance of writing large amounts of data. A good example of this is columnstore databases, which allow a high degree of data compression, and are optimized for bulk loading of data.

Since much of this data is for consumers (health devices, smart thermostats, etc.) and it needs to be tracked in real time to be valuable, it's easy to see how storage performance can impact delivery. This is especially true given the low latency needs users have come to expect in the era of broadband and 4G.

Up Next

In the next chapter you will learn about the details of modern analytical systems for both business intelligence and operational analytics.

2

IoT and Modern Big Data Systems

Traditional Business Analytics

The data warehouse and its related systems, such as online analytical processing (OLAP) cubes and reporting, have been traditionally considered as post-processing systems that offer a historical perspective on business information. The types of questions commonly asked were things like, "How many units of product did we sell in California last month?" Or "What was our most productive store?" While these are still valid business questions, many organizations are looking for more predictive analytics to allow them to shift resources in real time.

One of the reasons why data warehouses typically looked backward is technical. The online transaction processing (OLTP), such as point of sale or e-commerce servers, processed transactions in real-time; however, the analytic systems always functioned at least hours, if not a full day behind in data processing.

This was considered acceptable for many business systems, and was a scenario that was somewhat driven by technical limitations of the business. In fact, this model can still be acceptable for firms that operate in a single location during traditionally normal business hours.

For most organizations though, this won't work. As businesses trend toward globalization, being able to offer predictive analytics (as opposed to historical analysis) demands real-time data analysis. This increases the technical demands on business systems, so they can ingest data in real time.

Another trend, which you learned about in Chapter 1, is the vast increase in data sources that feed into modern business analytics. While having data from the OLTP system was enough for the data warehouse in the past, the data warehouse now has data from sources like social media, partner systems and other external sources. This means the modern data warehouse needs to be able to ingest and query large volumes of data very quickly.

What About that Cool Open-Source Project?

One major trend in "big data" has been the introduction of distributed open-source systems like Hadoop, HBase, Spark, amongst others. These systems began at web firms like Yahoo and Google, when the data volumes exceeded the ability to process on a single computer. They were also open source (and in many case authored in-house), which lowered overall costs.

Hadoop Origin

Hadoop was developed at Yahoo! in 2006, to replace a data warehouse that could no longer manage their search results.

While Hadoop and the other distributed big data systems have had success at many firms, they have not been as broadly adopted as many industry analysts had expected. The reasons for this are many, the first of which is that these systems are unfamiliar to many professionals who have spent their careers with relational databases.

Here is a comparison of open-source distributed systems and relational databases.

Relational Databases

Strengths

- Robust, mature technology
- Disaster recovery and high availability well defined and integrated
- SQL Language in broad use in both business and technology groups

Weaknesses

- High licensing costs at large scale, as most relational systems are licensed by the number of cores
- Some developers are more comfortable writing program code (C#, Java) than SQL
- Limited ability to scale beyond a single server, without a great deal of custom development

Distributed Systems

Strengths

- Ease of scale-out model. Just add servers to add more scale
- Open-source software means typically lower cost associated with licensing
- Good at dealing with data that is less structured such as JavaScript Object Notation (JSON) files

Weaknesses

- Systems lack maturity around toolsets and typical operations like backup and recovery
- IT organizations have limited experience managing these systems
- Built by developer for developers which can make data access for business analysts with limited programming skills challenging.

Early interactions with these systems were performed via Java or C languages, which most business intelligence analysts were not familiar with. It is telling that there are a number of open-source and commercial software projects to build an SQL interface layer on top of Hadoop. This is largely a signal of how popular and broadly adopted the SQL language is for interacting with data sets.

Additionally, from an infrastructure management perspective, many of these tools had not yet matured and lacked support from monitoring and backup tools.

While Hadoop and other open-source systems are seeing use, relational systems have developed strategies to process large volumes of data in near-real-time scenarios.

The Modern Data Warehouse

Data warehouses have a very specific query pattern that is intensive on both CPU and disk resources. For example, a common data warehouse query might report the average of all retail sales at a given store location for each day of the last month. In order to perform this calculation all of last month's sales data needs to be retrieved. Processing this data on a row-by-row basis will be a very expensive proposition, computationally.

One recent enhancement to CPU architecture has been the addition of Single Instruction Multiple Data (SIMD) instruction sets. This enables a concept called "Vector based processing" which, when encoded into a database system, allows the aforementioned sales data to be processed in batches of approximately 1,000 records. Processing batches is a much more efficient way to do aggregations, as opposed to operating row by row.

This type of processing benefits many systems, but has the most effect on data warehouse systems.

SIMD Instructions

SIMD instructions were added to most Intel server CPUs beginning in 2008. However, their history goes back to the Cray Supercomputer in the 1970s. The first common commercial implementation was in the Sony PlayStation 3.

Columnar Data Warehouse

Traditionally, databases have used structures known as *B-trees* for storing data.

B-Trees

The origin of the term *B-tree* has never been explained by the authors who coined the term. Some suggest that the *B* may stand for the terms *balanced*, *broad*, or *bushy*, while others have suggested it stands for *Boeing*.

This tabular structure of storing data works particularly well for systems that are primarily built around inserting and updating data (OLTP systems). OLTP systems need to be able to both read and write data relatively quickly. Their primary operations are to record business transactions, which sometimes requires looking up other values, but in general these systems are optimized for write performance. The b-tree structure allows quick lookups with a small amount of overhead for writes.

The structure of the B-Tree as shown in **Figure 2-1** shows the tree structure of a typical b-tree index. At the top we have root node, which typically stores the data in the table; the leaf nodes point back at the root node, simplifying data lookup operations.

Data warehouses have different query patterns; as opposed to loading one row at a time, a large number of rows are generally loaded. The rest of the time, these rows of data are only read. So a system design optimized for bulk loading and rapid reading of data, at the expense of small updates, inserts, and deletes is the ideal data warehouse design.

A 2008 white paper called "C-Store" lays out a framework for storing data in a purely columnar format for read optimization. Think of a

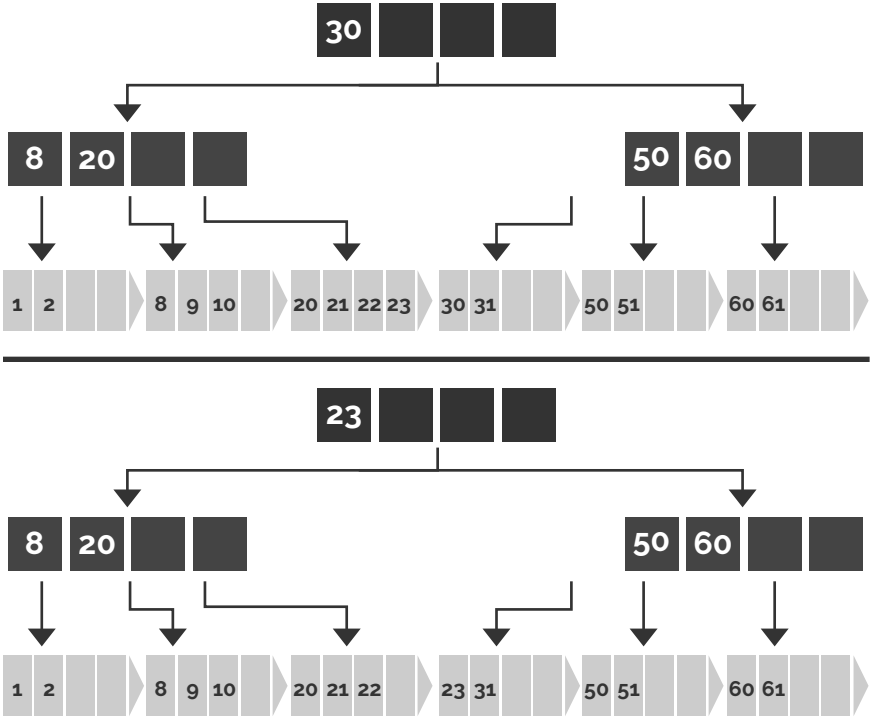


Figure 2-1: An image of a B-Tree Index in a database.

table with its data stored in columns—each of these columns is then encoded and compressed. Since the data in a column is likely to have a lot of repeated or sequential values, the compression ratios can be quite high. Also, because these columns are stored independently of each other, the database system only needs to scan the columns that are requested in the query. In **Figure 2-2** you can see a physical representation of a columnstore index the way this method is implemented in Microsoft SQL Server.

This approach to storing data allows workloads to take advantage of the SIMD instructions on the CPUs to process data in batch execution mode, meaning the data can be read and analyzed in chunks of 1,000 rows. This lowers overall CPU utilization, as well as disk reads.

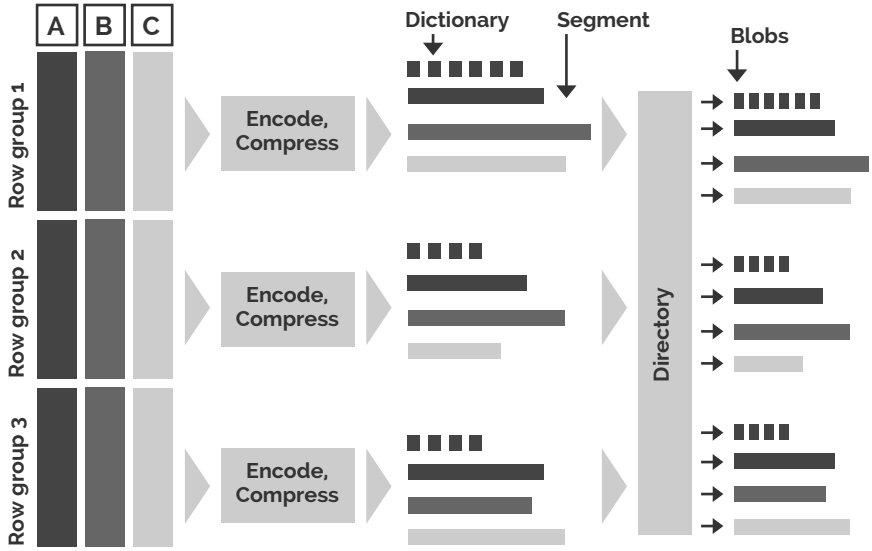


Figure 2-2: Representation of columnar data storage in SQL Server.

Keep in mind, however, that some operations may fall back to using row execution (where each row is read individually), and this will have negative impacts on system performance. It is important to understand what operators are supported in your database for batch execution mode and try to avoid allowing operations to revert to individual row execution.

A number of vendors (Microsoft, Oracle, and HP Vertica) have developed databases that take advantage of columnar technology. In fact, you may find that many relational database vendors refer to columnar technology as "in-memory technology." This is somewhat of a misnomer because while all of the major solutions do use in-memory caching similar to a normal relational database, and they are particularly efficient at memory utilization, however they are not fully in-memory. The bulk of the storage of the columnar resides on disk and not in memory.

Columnar technology has allowed for a number of changes in architecture. More data can be stored thanks to the higher degree of compression. This also means the system can support more queries that cover a wider range of data. Many vendors have begun to incorporate advanced statistical analysis into their databases, as well.

Having access to these statistical modeling packages is useful to users, but the added complexity can increase workload on memory and storage. While the database is optimized for this workload, the sheer volume of data can begin to put pressure on the storage subsystem.

To support efficient data loading, these database systems have built optimized methods for bulk inserts that stream data directly into compressed column stores.

For insert methods that are too small to take advantage of bulk loading, the databases have a B-tree structure on disk to provide storage. When these small inserts occur, overall system performance can degrade, because scanning the B-tree structures is less efficient than scanning the column stores. This means the only way for the database to traverse the B-tree is by using row execution mode.

All of these efficiencies allow for more frequent data loading, faster response times, and more complex queries, which allow business users to meet their real-time reporting requirements (as long as IT has the infrastructure to support them).

Read-Optimized Data and Storage

Most of the column store architectures across various vendor products treat storage in a very similar way. As opposed to regular databases, which typically store data in 8kb pages or blocks, the compressed columns are stored as larger binary large objects, or BLOBs. These BLOBs are accessed as needed by the query.

This is an extremely efficient method for storing data and disk access, and allows for more data to be stored and more analysis to be done. More data and more analysis can also drive high levels of load on the data warehouse and its subsystem, causing contention at the storage tier.

Analytic workloads do large sequential operations against storage devices. The key to delivering performance for these operations is having responsive storage with a great deal of bandwidth between the storage device and the server tier. Think of this as a highway with eight lanes of traffic. The key of an eight-lane highway is to have as many vehicles moving as possible. Likewise, bandwidth must allow as much total throughput as possible. This is different from an OLTP system whose I/O pattern is much more random and reliant on just having a large capacity of IOPs.

How Did Data Get This Big?

You have learned about the technology supporting these massive data volumes, but what is driving this growth? In Chapter 1, you learned about social media and machine- generated data. As mentioned in the last chapter, humans can only generate so much data, but the amount of data generated by machines and sensors is infinite.

Now you will learn about some business cases which drive big data.

The Large Hadron Collider

CERN (a French acronym for the European Organization for Nuclear Research) is an organization eponymous with its research lab in Geneva, Switzerland. The research lab has the largest particle accelerator anywhere in the world where particles of energy collide up to 600 million times a second.

At the core CERN's research is are data about its experiments. At peak research time data is generated at a rate of 10 GB per second, and data is processed at over 1 petabyte (1,024 TB) per day.

It is important to note that the data center at CERN does not collect all of the data generated at the facility because of capacity constraints in both storage and computing— even though their data center has over 45 PB of storage capacity!

The Boeing 787

As mentioned in Chapter 1, the Boeing 787 is a modern airplane built out of the latest materials like carbon fiber, and using the latest computer aided designs. In addition, the plane generates over 500 GB of data *per flight*. Some of this data is shipped in-flight to Boeing for monitoring detail.

From a business perspective, this in-flight monitoring allows Boeing to be far more efficient in their support operations. If problems are detected with a given airplane part mid-flight, the airline can move resources and parts to the destination airport, potentially even before the plane lands. This data and connectivity also supports smarter maintenance, on-time performance, and overall operational efficiency for the airline.

Big Data Closer to Home—The Smart Data Center

While CERN and Boeing's cases may not be reflective of your real world workloads, turn your consideration to another big source of data—your own data center.

Data centers are expensive to build and operate, as the associated power and cooling costs add up very quickly. The concept of a "smart data center" brings together some of the concepts you learned about in Chapter 1, such as smart power meters and grids, virtualization, and predictive analytics.

Hardware manufacturers also enable data collection by allowing data about real-time power use to be collected at the server level.

One example of how collecting real-time power use could benefit your data center is to collect data to understand when your overall resource utilization is at its lowest. Then, your virtualization hypervisor (the software that runs your virtualization layer) can power down physical machines to reduce your power utilization with no downtime for your workloads. While this may seem like a small gain, in the context of a very large data center, it can result in significant improvements in power utilization.

In this section, you learned about some real-world use cases of big data and about how businesses take advantage of real-time analytics.

Splunk

If you work in a medium-sized or larger organization, your organization likely generates a lot of log files. Each server in your data center has at least three to four log files associated with them, and may have many more depending on the applications running on them. Additionally, a lot of other devices in your organization generate log files—user PCs, machines running on computers, badge readers, and HVAC equipment.

Splunk is a software package that allows organizations to gather all of that data and search, monitor, and analyze it. The software builds indexes across the data and correlates it for generating reports, alerts, dashboards, and visualizations.

Use Cases for Splunk

IT Security

One of the major use cases in IT for Splunk is security and intrusion detection. Threats to IT security and breaches are far too common, and no CIO wants to face the board of directors to explain how their organization was hacked. By collecting data from across the organization, IT security can look for patterns and anomalies in data access patterns. Splunk allows administrators (in a large network) to quickly identify and isolate any threats to organizational security.

Customer Service

Log analysis can go far beyond IT security. For example, a software company may collect telemetry data from their applications, so they can collect data on how their users are using the applications, which in turn can help identify problem areas in code, focus new development, and identify anomalies. Ultimately, this can also help customer service by providing guidance to where customers exactly had problems to make troubleshooting easier.

Healthcare

A use case beyond IT is medical sensor data. Electronic medical records, and medical device data are quickly taking over modern medicine, providing doctors some of the same predictive analytics that IT professionals have been using for years. Splunk gives healthcare organizations the ability to tie disparate sources of data together to make better informed decisions.

Splunk and Hardware

As you can imagine, bringing together all of this data, correlating it, and querying it, is computationally expensive. Splunk has an I/O pattern that is a mix of sequential reads and sequential writes for the indexing process. Searches have phased operation of high numbers of seeks and small reads, and larger reads are mixed with decompression and computation.

Compression

Having this volume of data can lead to massive amounts of storage infrastructure. One of the ways you can reduce the costs associated with this is to use storage based compression to reduce the overall storage footprint.

This means having a high performance storage subsystem will greatly enhance the overall performance of your Splunk environment and allow your users to get more usefulness out of it.

Up Next

In the next chapter you will learn more about big data systems and how analytics have evolved to be more real time. Additionally, you will learn about public and private clouds and how the costs of the public cloud are not always cheaper than doing things yourself.

3

The Evolution of Analytics & Computing

Evolution of Big Data

In the last chapter, you learned about Hadoop, modern data warehouses built on relational databases, and a little bit about some other distributed computing systems. In this chapter you will learn about how these systems (particularly Hadoop and other open-source systems) have evolved quickly in their early days to achieve near real-time response time, especially when comparing them against a simple batch system.

You will also learn about how infrastructure design has adapted to support these real-time analytic systems.

Finally, you will learn about public and private cloud environments, including where cloud environments are suitable and where you will likely be spending too much money.

Batch Processing Versus Real Time

Hadoop was originally designed to be purely a batch-processing system. When Yahoo! designed Hadoop there were no requirements for real-time reports or small single writes and updates like a relational database management system (RDBMS) requires. Hadoop was built to take advantage of sequential data access, using very large, very slow hard disk drives in which the I/O workload is distributed over many devices.

This distributed infrastructure design is great in that it is extremely cost effective, and it scales out very well (thanks to Hadoop's distributed file system [HDFS]); however, it will never deliver low-latency reads and writes.

Hadoop's original data processing architecture used a computational paradigm called MapReduce. MapReduce employs a map preprocessing step to identify the location of the data in the cluster, and where the computation should take place. Then the reduce steps aggregate and distill the data. There is no way to apply this approach to real-time streaming data, because the reduce step requires all input data for each unique data key to be mapped and collated first.

It is important to note that Hadoop's original hardware configurations focused on medium levels of RAM (64–128 GB); CPUs with two sockets and eight or more cores; and twenty-four 4 TB disks running in a RAID0 (redundant array of independent disks) configuration, with the HDFS responsible for data protection.

Introducing YARN

Hadoop evolved quickly beyond using MapReduce as its only processing technology. Then, Hadoop 2.0 introduced YARN (Yet Another

Resource Manager) which brought forth a resource scheduler to try to perform data processing tasks in a more real-time fashion.

YARN introduced a processing model the reduced the I/O intensive, high-latency MapReduce framework.

YARN enabled a broader ecosystem of applications to be built on top of Hadoop, one of the most popular of which was Spark. Spark is a more efficient batch processing system than MapReduce, and it includes extras like transformations in ANSI SQL. This ecosystem of applications takes advantage of Hadoop and YARN as a management system.

Other more real-time use cases for Hadoop and its ecosystem include SQL engines like Impala. While SQL on Hadoop tools have existed since the early days of Hadoop (e.g., Hive is a tool that translates SQL into MapReduce jobs and was an early Hadoop sub-project), modern tools like Impala offer performance that is much closer to an RDBMS.

Trends Impacting Hadoop

Increased processing demands have led to a couple of changes in Hadoop hardware—more memory for caching and in-memory processing, and faster storage to enable real-time processing.

These increases in performance, combined with the push toward virtualization for all servers, has led to Hadoop being implemented on shared storage devices like storage area networks (SANs). While Hadoop was intentionally designed around local storage, there is a trend to move it toward shared storage.

Another trend is the use of SSDs in big data projects. When applications and business users are looking for real-time analytics the latency associated with hard disks is not acceptable.

Storage and Distributed Systems

Hadoop, as mentioned earlier, was designed around local storage; it was cheap, dense, and originally did not have a requirement for fast I/O. Additionally, since Hadoop managed its own file systems through HDFS, turning off access time and save time for files at the operating-system level are considered best practices.

While local storage was the optimal design for Hadoop, it faced a number of challenges. One is the sheer number of disk failures that will be encountered in a large cluster. Hard-disk drives are mechanical devices and will fail eventually. If you manage a small cluster, this is not an overwhelming problem; however, when you start talking about a large number of nodes (> 100) failed disk management can become a large hurdle.

The other trend that leads some Hadoop environments to use shared storage is the trend toward virtualization. This, in turn, is driven by the trend toward cloud computing, both private and public.

You will learn more about cloud computing later in this chapter, but as IT organizations move toward virtualizing all of their infrastructure, the idea of a large physical server implementation to support a Hadoop cluster is not a popular one among infrastructure teams.

One benefit to using shared storage versus local is that many SANs have added features, like compression and deduplication, which offer a higher storage density over standard storage. While this approach may be more expensive, it can be a good cost-effective solution for small- to medium-size clusters. Additionally, it gives a Hadoop cluster the ability to add storage without having to add compute.

The other thing to consider about large physical Hadoop clusters filled with local storage is the expense of power and cooling for those machines. These are not inconsequential considerations when deploying a large environment.

Cloud Computing

One trend in IT that is more pervasive than others is cloud computing. There are a number of definitions for *cloud computing*, but the National Institute for Standards and Technology (NIST) has the official one:

"Cloud computing *is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*"*

NIST also addresses a number of considerations like the service models:

- **Software as a Service (SaaS).** Providing applications to users running a cloud infrastructure

- **Platform as a Service (PaaS).** Applications such as databases, libraries, and frameworks. The consumer does not manage the underlying infrastructure, but fully manages the data within the service

- **Infrastructure as a Service (IaaS).** This is the easiest solution to explain. This is the combination of your virtual machine (VM) and networks running in a cloud vendor's data center.

*http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf Accessed on April 7th 2016

Hybrid Clouds

A *hybrid cloud* is typically defined as a bridge between your on-premises networks (whether private cloud or physical servers) that is connected to a public cloud provider.

Hybrid cloud is a fantastic solution for disaster recovery for many organizations. While many organizations have a single data center at a primary location, fewer organizations have a secondary or even tertiary disaster recovery site, which would protect their computing resources from regional disasters. Build a second (or third) data center is a very expensive capital investment, and using the public cloud for disaster recovery can be a very effective, pay-as-you-go insurance policy. Additionally, many organizations may consider eliminating the process of shipping backup tapes offsite, replacing that process with direct backups to the public cloud.

Private Clouds

Private clouds have the most varied descriptions. A lot of organizations simply think that having a virtualization farm means that they have a private cloud. However, this is not the case. A private cloud has similar characteristics of public cloud in that users can request (ideally in programmatic fashion) new compute resources to be allocated from a large pool.

Through its virtualization layer, a private cloud offers a high-availability solution and workloads are protected from hardware failures. In most cloud environments, disaster recovery (loss of an entire data center or region) is managed at the data tier, as opposed to the infrastructure tier. However, when using a private cloud in conjunction with a hypervisor and storage replication, there are options for disaster recovery.

Private Cloud Infrastructure

Building a private cloud is a rather large endeavor, which begins with estimating the amount of hardware required for your deployment. Typically, this involves a long-term assessment of your previous infrastructure utilization and applying it to a new infrastructure design.

This hardware acquisition is a large capital outlay. Typically, this will include building a large virtualization infrastructure which consists of servers and storage. Settling on the amount of hardware is challenging, as is planning for delivering consistent service and performance as the utilization increases. This all requires careful planning and an experienced infrastructure architecture team.

Building a private cloud involves a little bit of a shuffle. If the existing hardware still has life, and as workloads are moved into the private cloud, the legacy hardware can be also moved into the cloud. Balancing this transition from a dedicated physical infrastructure to a virtualized private cloud can save a lot of money through reducing the initial hardware outlay. This is particularly good for servers as the consolidation provided by virtualization will allow you to double up some workloads, however shuffling storage can be trickier because having two workloads will always require twice as much storage space.

Because of the pressure this puts on your storage subsystem (as all virtual environments do), it makes the storage the most important part of delivering adequate performance for the overall infrastructure.

This pressure of virtualization has brought forth fundamental changes in the architecture of storage arrays. Most storage vendors now use tiering, which means a combination of fast and dense storage that stripes data small (512 kilobytes-1 megabyte) across all of the storage devices (solid-state drives and hard-disk drives) in an effort to balance the I/O workload across the array. Additionally,

the array manages statistics on the individual blocks and moves them from slower to faster tiers as those blocks are more frequently accessed.

This is in contrast to the older style storage architecture, where a group of disk drives would be dedicated to a workload. This striped architecture allows more overall throughput from the storage array. While this design could be effective on physical servers, as workloads consolidated through virtualization, pressure increased on storage, and the new architectures were developed.

Private Cloud Software

After building the infrastructure, a user interface and an application programmatic interface (API) layer need to be constructed. This requires programming skills and extensive testing. While most of the virtualization engines (VMWare, Hyper-V, and Open Stack) have their own APIs, tailoring your front-end software to meet the needs of the hypervisor's API can be challenging.

You may have heard this described as "infrastructure as code," which is a fairly accurate way to describe it. By virtualizing and making all of the key components API accessible, it offers a benefit from an automation and disaster recovery perspective because new virtual hardware can be rapidly deployed.

The last component involved in a private cloud is building and maintaining templates for all of the various types of services you would like deploy. Some examples of this include regular Linux servers, Linux servers running MySQL, Windows web servers, and Windows SQL Servers. Each will require a great deal of effort to build and maintain.

Private Cloud — The Benefits

As you have read above, a private cloud is a large project that will consume a lot of resources and is not for the faint of heart (or very

small organizations). However, if you are a medium to large organization this solution can be very cost effective because of the density you can achieve, as well as the reductions in time to deployment for new projects.

A private cloud also offers you the ability to isolate priority workloads by using quality of service (QoS) software built into virtualization tiers. Using QoS, you can easily allocate the fastest storage and guaranteed server hardware resources to important VMs. Additionally, while the initial capital expense is high, over time hardware costs will reduce as your server costs become more predictable based on your usage data.

When Is the Private Cloud Right for You?

If you are a medium to large organization with hundreds (or thousands) of servers, a private cloud can be a great investment that will reap dividends over time.

Smaller organizations will have trouble reaching the payback period on this project, but it does not mean they should shy away from virtualization or relocate all of their workloads to a public cloud. However, building a true private cloud is probably not in their best interest. The costs and the complexity of the project are too large for many smaller organizations.

Managing Your Private Cloud

The goal of managing your private cloud is achieving a high virtualization density while delivering promised levels of performance.

This is one of the main differences between building a private cloud and using a public one. When you own your private cloud you can guarantee high levels of performance while maintaining a consistent cost, whereas the public cloud provider will generally be focused on density over performance.

Public Cloud

While there are number of public cloud providers, including Google, Microsoft, Amazon, IBM, and others, two of the most popular are Amazon Web Services (EC2) and Microsoft Azure. EC2 and Azure are the most adopted and most mature clouds, so it helps to focus on their features when discussing public clouds.

Public clouds are similar to private clouds in that they are both built to deliver on-demand computing and storage resources, with the additional offering of platform as a service (PaaS) that generally does not exist in most private cloud environments. Some examples of this include Azure SQL Database and Data Warehouse, and Amazon Elastic MapReduce.

These services offer a data platform on demand, with minimal configuration options beyond scale. These PaaS offerings are good for new application development which does not require much in terms of code shift to take advantage of the PaaS services. Legacy applications may require extensive code changes to deal with cloud computing paradigms.

When Is the Public Cloud Right for You?

The public cloud is good for a lot of organizations, and if you listen to the marketing from Amazon or Microsoft, you would think the right thing to do was immediately move all of your workloads directly to the public cloud. While the public cloud does offer a great deal of flexibility and nearly infinite scale, there are some limitations you need to be aware of — namely, the public cloud offers flexibility and scale, but at high costs, especially for workloads that need high levels of scale all the time.

Public cloud is ideal for smaller organizations who only have a handful of workloads, or have one or two large workloads that have large swings in utilization.

The real benefit to using the public cloud is flexibility and automation. It is easy to scale hardware and services to support the changing needs of your application. This is particularly useful when you are deploying a publically facing application that has unpredictable utilization.

If the app deploys and is wildly popular, it is easy to scale up to match its needs. Likewise, if the application is unpopular, it is just as easy to scale back the resources associated with the application.

This type of app can work in an on-premises environment, but it requires a heavy investment in hardware to provide the "burst" capabilities that public cloud offers in order to deal with large workloads.

Public Cloud Infrastructure

Microsoft has been fairly open about the infrastructure used in their public cloud by taking part in the open source hardware project called Open Compute. They use customized hardware that is not that different from what you would find in a standard server from a major vendor; however, it is optimized for the density required in a public cloud. This hardware is very similar to the blade chassis configuration that many hardware vendors have been offering for years.

From a storage perspective, both Microsoft and Amazon offer standard and premium tiers of storage.

The standard storage offers the benefit of low cost; you can allocate many terabytes of storage and only pay for what is utilized. This is likely because cloud vendors are using thin provisioning at every layer of their infrastructure.

Thin Provisioning

Thin provisioning has long been used in SANs and virtual environments to increase utilization. It requires careful management to ensure good performance and avoiding oversubscription.

From Microsoft's documentation*, you can ascertain that their standard storage is running on 3.5" hard drives. The IOPs offered with these standard devices are generally not compatible with the needs of real-time reporting applications or many database applications in general.

Both major providers (Amazon and Microsoft) offer premium storage as well. In Microsoft's case this is an SSD in the backplane of their blade chassis. The major difference between standard and premium storage is the way the storage is billed. For standard storage, if you have 3 TB of storage allocated to your VM and are only using 3 GB, you are only billed for the 3 GB. However, if you are using premium storage and have the same 3 TB allocated with 3 GB used, you will be billed for the entire 3 TB.

When Should You Avoid the Public Cloud?

Many organizations, especially larger ones, will take a hybrid approach to their cloud environments, mixing workloads amongst cloud providers and keeping a large percentage of resources on-premises.

When you are a large organization buying hardware in large quantities and owning your own data centers, it is hard to have a cloud provider beat your economies of scale. By the time you factor in the profit margin of the cloud provider, it is hard to argue with staying on-premises.

*http://www.opencompute.org/wiki/Server/SpecsAndDesigns Accessed on April 12,2016

Smaller organizations can benefit from public cloud for some of their basic workloads like e-mail or customer relationship management (CRM) systems. However, their core business systems may require heavier computing resources that are less cost effective in the public cloud.

A good example of this the GS-5 virtual machine from Microsoft. At the time of this writing (May 2016) the costs associated with that VM are approximately $89,000 per year. While the GS-5 offers 32 CPU cores and 512 GB of RAM, a similar server could be purchased for around $15,000 to $20,000.

As you can see, the public cloud offers a great deal of flexibility, but at a high cost. A smaller organization may benefit from using a public co-location data center, which allows them to choose their own hardware better suited to their needs.

Leaving the Public Cloud

A good example of a company who built its business on the public cloud but left when it reached its own economies of scale is Dropbox, the cloud based storage provider.

As Dropbox increased in both size and value, the costs associated with trying to implement their workloads within Amazon's cloud became more problematic and less cost effective.

This type of project is daunting, and it took Dropbox over two years to move off of Amazon and into their own cloud. They had to deal with the logistical challenges of moving petabytes of data out of one cloud and into another, but they made it pay off.

Summary

In this chapter you learned about the big data systems, and their changing infrastructure requirements. Additionally, you learned about public and private clouds and when they are right for your organization.

Up Next

In the next chapter, you will learn about modern all-flash and hybrid arrays, and how they can support real-time analytics systems.

4

All Flash Storage and the Modern Architecture

In earlier chapters you learned a few important things about data: data volumes are increasing across the board, businesses are demanding more real-time access to that data, and database systems are evolving to be able to manage volume and querying needs. You also learned how IT infrastructure is consolidating and when to use public and private clouds.

In this final chapter, you'll learn how all flash storage can handle these modern demands.

Pressure on the Storage Subsystem

The consolidation of workloads that virtualization has brought forth has added pressure on storage arrays. The gold standard that many database and storage administrators want in terms of disk performance is for a write to complete in less than 5 milliseconds (ms) and a read to complete in less than 10ms. Unfortunately, with the introduction of virtualization and larger data sets putting more pressure on storage

arrays, those latency numbers have crept up over time. In many virtual environments the typical numbers shifted to 20 ms for writes, and 40 ms for reads.

This increased latency has occurred because it was prohibitively expensive to get a tremendous amount of IOPs and low latency from an array of spinning disks. The only way to do it is to add more "spindles" (i.e., a hard-disk device), which comes with both a retail cost and the added cost of power, cooling, and space.

Also, in order to get maximum performance from hard disks, many storage administrators use a technique known as "short stroking," which reduced the overall capacity of storage array and effectively drives up the cost of the storage when measured in a cost-per-terabyte basis.

Short Stroking

Short stroking is a technique used in enterprise storage environments to describe an HDD that is purposely restricted in total capacity so that the actuator only has to move the heads across a smaller number of total tracks. This limits the maximum distance the heads can be from any point on the drive thereby reducing its average seek time, but also restricts the total capacity of the drive. This reduced seek time enables the HDD to increase the number of IOPs available from the drive. This performance benefit comes with a price; the cost and power per usable byte of storage rises as the maximum track range is reduced.

What About Flash Storage?

When solid state devices (SSDs) started to become mainstream in the 2000s, there were a couple of concerns. Early SSDs had a low write

limit (the total number of writes the device) and high cost, as the early devices were prohibitively expensive for all but the most mission critical workloads. In time a few things have changed to make SSDs, or flash, the storage of choice for most workloads.

Costs have dropped tremendously: just look at the consumer market for flash—at the time of this writing (mid 2016) a 1-TB internal flash drive can be purchased for less than $300. (Just two years ago the same device cost over $2,000.)

Additionally, the write limits on the devices have increased, and, in the cases of the enterprise storage arrays, the disk controller does something known as wear leveling. Wear leveling balances the writes across individual storage devices and offers reliability that in many cases is much better than hard disk drives (HDDs).

Finally, the storage arrays now have added features such as inline compression, deduplication, and encryption which offer more value. You will learn about flash storage later in the chapter, but first lets discuss how storage architecture has changed over time.

Modern Storage Architecture

Computer systems have evolved to meet the ever changing needs of workloads. A good example of a modern storage architecture is the storage area network (SAN), which developed as data volumes increased beyond the capacity of what could be stored in an individual server. SANs provide other benefits such as high availability, compression, and potentially performance, but the biggest one is just the ability to large amounts of data.

Before we get into the depth of storage architecture, review the basics of SAN architecture in **Figure 4-1.**

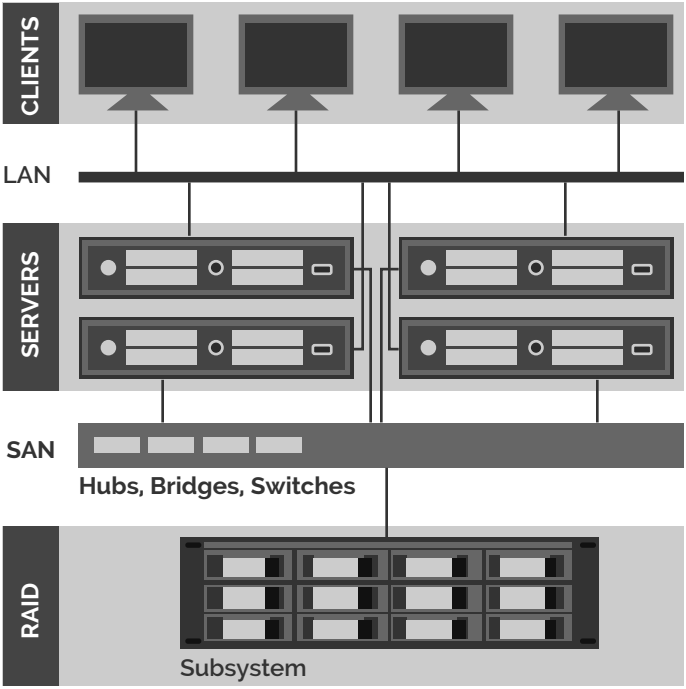


Figure 4-1: High-Level SAN Architecture

Figure 4-1 shows a very basic high-level architecture. The SAN is truly a network, which is why the top level component in the subsystem is the switch. These storage switches may be commodity network switches, or specialized storage switches, and provide connectivity to the physical servers and to the storage array. This connection needs to provide large amounts of throughput with extremely low latency. Depending on the array it may be a fibre channel connection or a high throughput Ethernet iSCSI connection.

Fiber vs Fibre

The development committee of the fibre channel protocol elected to use the British spelling of fibre as the standard. Fiber optic with the American spelling only refers to optical cabling.

Storage Controllers

Behind the network layer lies the heart of the SAN, the controllers and storage array. The controller (and there will always be two of them for redundancy) is a specialized computer with its own operating system for managing the storage and communicating the host (client) systems.

Some SAN vendors refer to *active/passive* and *active/active* controllers.

- **Active/passive.** Provides a measure of high availability to the SAN.

- **Active/active.** Allows more workload to be managed by the controller, while still providing high availability.

One of the biggest functions the controller provides in any configuration is the caching functionality. As mentioned earlier, the storage controller is just a specialized computer, which like any other machine, contains a CPU and RAM. The RAM in the controller plays the very important function of caching reads and writes from the underlying storage array, to relieve pressure on the underlying disk devices.

The storage array consists of a large number of storage devices (either hard disk, solid state, or a combination of the two), which are organized into RAID (redundant array of independent disks) to provide data protection.

Redundancy

You will notice that many layers of this infrastructure provide redundant service. That's good because a SAN is perhaps the most mission critical component in a data center, given the shared nature of the infrastructure.

The controllers and storage arrays work together to bring intelligence to the SAN. This is the real secret sauce of any SAN, starting with the automatic tiering that many SANs use.

Automated Tiering and SANs

Many vendors have incorporated automated tiering technology to provide more balanced performance in a SAN. This approach takes two or three different types of storage devices (commonly flash, 15k RPM hard drives, and slower but denser 7,200 RPM hard drives) and groups them into pools. Then the system moves small blocks of storage to higher performing or lower performing tiers as the data in those blocks is more frequently read.

This can work; however, data access patterns can be fairly unpredictable. Automatic tiering in this fashion is not always the best way to take advantage of limited flash resources.

Understanding Flash Storage

Flash storage is type of non-volatile memory storage, which is accessed electronically and can be programmed.

It was first introduced in the 1980s when Toshiba developed a new type of memory cell called E2PROM. The major limitation E2PROM had was extremely high costs. Then, in the late 1980s they developed a new type of architecture, called NAND, which had a much lower cost-per-bit, larger capacity, and many performance improvements. While still expensive at the time, NAND was cheap compared to the E2PROM cells. This NAND technology provided the footprint for the modern flash subsystems.

Flash Cells

The basic component of a flash storage device is a cell. Data is composed of 1s and 0s which are stored as electrons in a component of a flash cell called a floating gate transistor (**Figure 4-2**).

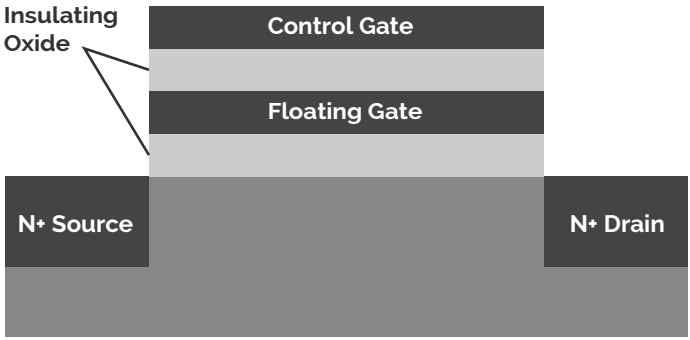


Figure 4-2: Floating Gate Transistor

For electrons to be placed on the floating gate, they must penetrate the insulating material on the cell. This involves a process called tunneling which exposes the cell to high voltage.

Tunneling is the fault of flash storage, as it causes physical damage to the insulation material. This means that the more times data is written to a cell, the more insulation is removed. This is reason that flash storage has a finite life span.

Single Level Cell Media

The traditional approach to building flash storage was to store a single bit of data in a given cell, this architecture is known as Single Level Cell (SLC). SLC offers the highest levels of performance, lowest power consumption, and the high write endurance levels. However, it also comes with a very high cost, and SLC devices are rarely seen in today's market.

Multi-Level Cell Media

The other approach to storing data on flash devices is to use a multi-level cell. As implied by its name, a multi-level cell allows double the number of bits to be stored on the same number of transistors as an SLC device.

However, this density comes at expense of endurance, since more bits are stored per cell that means more writes, and less write endurance.

Triple Level Cell Media

A triple level cell (TLC) allows for a third bit to be stored. Once again this means more storage and more writes, all at the expense of endurance.

Flash Media in the Real World

Each type of flash memory has advantages and disadvantages, which include tradeoffs between cost, density, performance, and write endurance. When choosing a flash storage architecture, you need to evaluate these tradeoffs, and see how they to fit into your environment and your needs.

Understanding Wear Leveling

With the write endurance considerations associated with flash storage, the ability to balance writes is very important. If you think about the life cycle of most data on a device, some of it is updated quite frequently, while other data is written once and then only read.

Wear leveling is a firmware method that comes in two forms:

- **Dynamic wear leveling.** Writes data being updated to a new physical location.

- **Static wear leveling.** This is a type of wear leveling in which all blocks are moved to ensure even cell wear. Static wear leveling is more complex and results in slower writes, but it does provide the maximum life for the device.

Types of I/O — Random and Sequential

All I/O can be broken down into two types of workloads: random and sequential. It's important to understand how these workloads function within HDDs and flash.

Random workloads involve searching all over a disk for data, and are less efficient than sequential workloads, particularly for HDDs. (HDDs perform at their best when doing sequential operations, which makes sense given their rotational nature.) Flash devices perform better in random operations, as there is no latency involved with searching across different storage cells to retrieve data.

This matters when considering the characteristics of your workload. Generally, the analytic workloads that have been described in this book are very sequential in nature, but do have some elements of random I/O. For example, random workloads tend to be associated with OLTP systems.

In some tests, HDDs perform almost at the same level as flash devices for sequential operations. However, these tests are typically performed on single devices, not on storage arrays.

Most flash storage vendors are aware of this performance characteristic and use firmware that randomizes the sequential I/O as it arrives to the array, this gives the flash array better performance than performing a purely sequential operation.

Flash Storage in SANs

Flash storage entered the enterprise storage market (at a high performance tier in storage arrays) as the cost came down, and the endurance problem was addressed by good wear leveling software. While this led to better performance, it did not offer the complete solution for many workloads.

A few things contributed to the more widespread adoption of the all flash storage array. The flash devices themselves became much denser, allowing companies to store data from data warehouse and other analytic systems at a per gigabyte cost that is almost the same as a storage array with HDDs. Additionally, the hyperconvergence and consolidation of workloads onto virtual machines demanded low latency for all storage.

All Flash Storage Arrays

All flash storage arrays offer a couple of benefits to the modern data center. First and most important is the high IOPs at low latency that only flash storage can offer.

Next, is a recent innovation: the high levels in density of modern flash devices allow these arrays to operate with nearly the same levels of storage density as an HDD-based array.

Another impact of density is that flash storage devices can operate in a more closely packed environment, offering higher physical density than HDDs. This means with some arrays you can deploy a petabyte or more of storage in a single rack. Given the data center footprint of traditional arrays (which are typically many racks) this is a big improvement.

Inline Compression and Deduplication

One of the benefits of many flash arrays, including Tegile, is the use of inline compression and deduplication. This allows more data to be stored on a given a storage device. It works in a multi-phase process of compression and deduplication where the data is first compressed and then deduplicated.

Starting with the basics, with most compression algorithms, compressing two identical data blocks will result in two identical compressed blocks. Deduplication then involves calculating and comparing cryptographic checksums from the data blocks, while the blocks remain unchanged as seen in **Figure 4-3**.

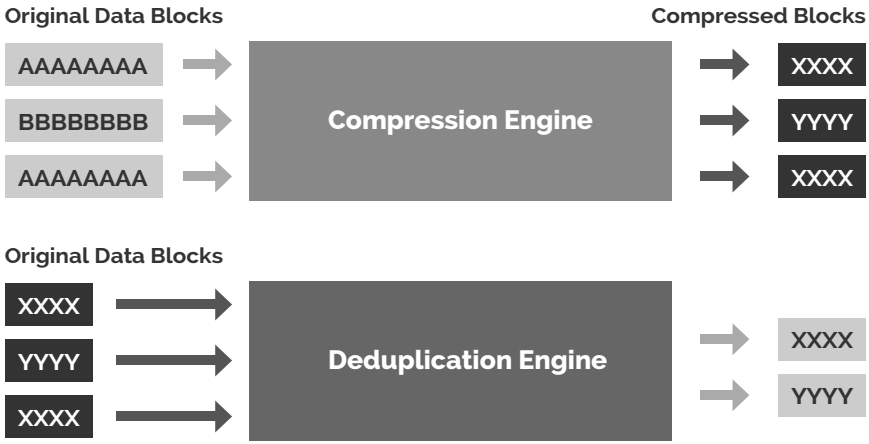


Figure 4-3: Compression and Deduplication Process

The process shown in **Figure 4-3** is CPU efficient by compressing data first and then using a strong checksum to deduplication. By compressing the data first, the process leaves fewer bytes for the more computationally expensive deduplication process.

Data Integrity

A checksum or hash sum is a small-size piece of data created from a block of digital data for the purpose of detecting errors which may have been introduced during its transmission or storage.

Compression for Density and Performance

Early computer users may have poor memories of compression, but when the HDD on your PC filled up, you had to compress folders to provide enough space. Suddenly your PC would grind to a halt.

With powerful modern CPUs this works the opposite way, while compression does require CPU cycles to compress and decompress data, the reduction in IOs to gather compressed data can greatly improve overall throughput.

Applying data reduction techniques can also offer benefits at the caching layer. A storage array can allow more hot blocks to be stored in its cache. Since the compression and deduplication take place before the data is written the underlying storage devices, the amount of writes to storage are also reduced. This can improve the overall read and write performance of storage arrays by more than a 100-fold.

Many databases offer in-database compression, which while helping I/O performance, increase CPU utilization on the database server. Also, in most commercial databases the use of compression requires additional licensing costs, and time consuming application changes. With the use of array-based compression these concerns are eliminated, as the compression is transparent to the application and database.

Compression & Deduplication — Making Flash Affordable

Storage cost is typically measured in two metrics—cost per IOPs and cost per GB/TB. In the case of cost per IOPs, there is almost no comparison between flash and HDD. A basic example of this is a PCIe SSD which costs $2,500 and delivers 400,000 IOPs, or less than a penny per IOPs. Even a high performing HDD like a 15k SAS disk, would cost $150 and deliver only about 180 IOPs. This is almost a dollar per IOPs, or over a hundred times more than flash.

When looking at costs and density, the density of HDDs has the cost-per-GB equation in their favor. The PCIe SSD in the previous example costs $2,500 and has a capacity of 1 TB, giving it a cost of $2.44/GB. The HDD costs $150 and has a capacity of 600 GB giving it a cost-per-GB of $0.25. As density in HDDs increases, this metric shifts more in their favor — dense drives like 4 TB SATA do not cost much more than the aforementioned SAS drive, bringing costs down to around $0.05/GB.

What brings these two metrics together in favor of flash is that as workloads need more IOPs (due to the demands of real-time analytics and processing applications), the cost of adding IOPs to a traditional array is very expensive based on the sheer number of HDDs required to service those IOPs. The added functionality of compression, encryption, and snapshots help tilt this equation in the favor of the all flash arrays.

Disaster Recovery

Another feature that many storage arrays provide is the ability to replicate data blocks from a primary location to a secondary array located in another data center, potentially in another location. This integrates into several solutions to provide data protection across sites, including Microsoft Server Failover Clustering or VMWare's Site Recovery Manager.

While mature database systems often provide this functionality at the database level, many of the newer data solutions, like Splunk, depend on external sources like array-based replication for disaster recovery. Additionally, this allows all resources to be failed over quickly and easily.

Modern arrays also allow this process to be configured easily and perform the initial seed for the secondary storage array to come from a snapshot. Replication may be snapshot based, or for data servers they can be real-time.

One concern in the past with SAN replication was that the second array would be completely idle (and therefore an expensive idle resource). However, many modern architectures allow a great deal of granularity and offer the potential for bidirectional replication models.

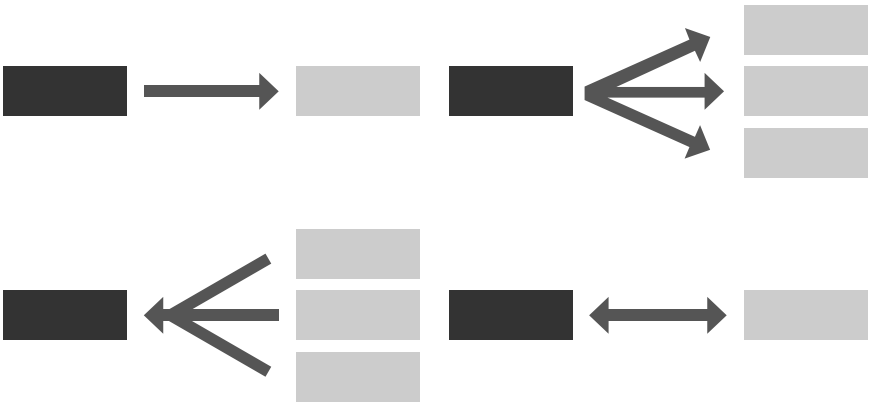


Figure 4-4: SAN Replication Topologies

As seen as **Figure 4-4**, there are potentially many replication topologies that can be configured. Multiple SANs can replicate only critical data to one SAN, or bidirectional replication can protect critical volumes on each SAN.

Data Protection

Data Protection in most organizations is typically defined by two key service level metrics recovery time objective (RTO) and recovery point objective (RPO):

- RTO represents the maximum time your systems can be down before the business is impacted.
- RPO represents how much data can your organization afford to lose before the business is impacted.

These metrics need to be evaluated for each business system in order to design a proper disaster recovery plan.

When you factor in the compression, performance, and disaster recovery options, the cost benefits of the all flash array quickly shows its benefits.

Benefits of All Flash Arrays for Splunk

Machine generated data is the biggest area of data growth in most organizations, and introduces a lot of complexity in both data and managing of the data. As you read about in Chapter 3, this workload is quite different from a traditional data warehouse or business intelligence workload.

Splunk started as an analysis system for log files, and has since evolved into a full-blown processing platform for all types of machine-generated data; Splunk is frequently described as "Google for visual analytics." This allows it to move into broader use cases like Security Incident and Event Management and the Internet of Things (IoT).

This is crucial because the amount of data will only increase. In fact:

- Cisco estimates that 50 billion devices and objects will be connected to the internet by 2020.*
- Gartner estimates that by 2020 IoT product and service suppliers will generate incremental revenue exceeding $300 billion, mostly in services.#
- IoT will transform many IT organizations by offering more detail and metrics about all aspects of operations.

All of data from these devices will need to be stored, maintained, and analyzed, which will place large demands on the IT organizations who need to support it.

Splunk Architecture

The architecture of Splunk is multi-tiered, and it is built from the following blocks (**Figure 4-5**):

- **Search Head.** Used for searching and reporting
- **Indexers.** Used for indexing and search services
- **Forwarders.** Used for data collection and forwarding
- **Data Management.**
  - Indexing Cluster Master, Search Head Cluster Deployer
  - Distributed Mismanagement/Deployment Server
  - License Master, Distributed Management Console

*https://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
#http://www.gartner.com/newsroom/id/2636073

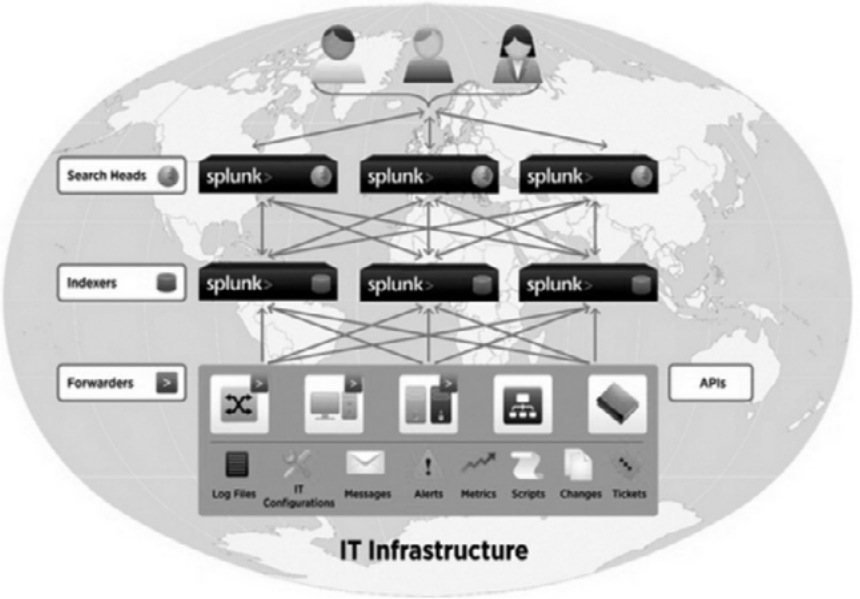


Figure 4-5: Splunk Architecture

Splunk has its own query language called SPL (Structured Programming Language) which allows querying of data sets. Additionally, other applications and APIs can be used to query Splunk data.

For the purposes of storage infrastructure, the key components of Splunk are its indexing servers. These servers provide three primary roles:

- Data storage
- Data management
- Data retrieval

Splunk actively manages its own data, processing and parsing data at the time it is indexed. The indexers also rotate data blocks in tiers (hot/warm/cold) and age and remove data. Additionally, the indexers return data to the search heads.

Making Splunk Faster

Operational analytic platforms like Splunk have significant resource requirements and drive a tremendous amount of IOPs. These workloads can benefit from flash arrays in several ways.

- **Faster ingest of data.** Inline compression reduces the number of physical rights and flash storage allows faster data loading
- **Faster query and report performance.** Latency is reduced and flash storage allows for dense queries to be processed with latency in the millisecond range
- **Indexing performance differentiation.** Tiered pipelines in Splunk tie nicely to tiered storage volumes in the array
- **Ease of scalability.** The density of IOPs provided by all flash arrays means easy scalability that does not impact the data center floor plan

The ability to perform high performance tiering that many all flash arrays support aligns perfectly with Splunk's use of hot, warm, and cold buckets, as shown in **Figure 4-6**. These tiers are aligned to the appropriate tier in the storage array based on their stage within the Splunk pipeline: Ingest, Search, Index, Query, and Visualize. Doing this within a single array can improve Splunk's overall performance.

Flash architecture provides Splunk customers with better performance by accelerating search and reporting processing. The array provides the ability to optimally handle both sequential and random I/O requirements across the Splunk data lifecycle.

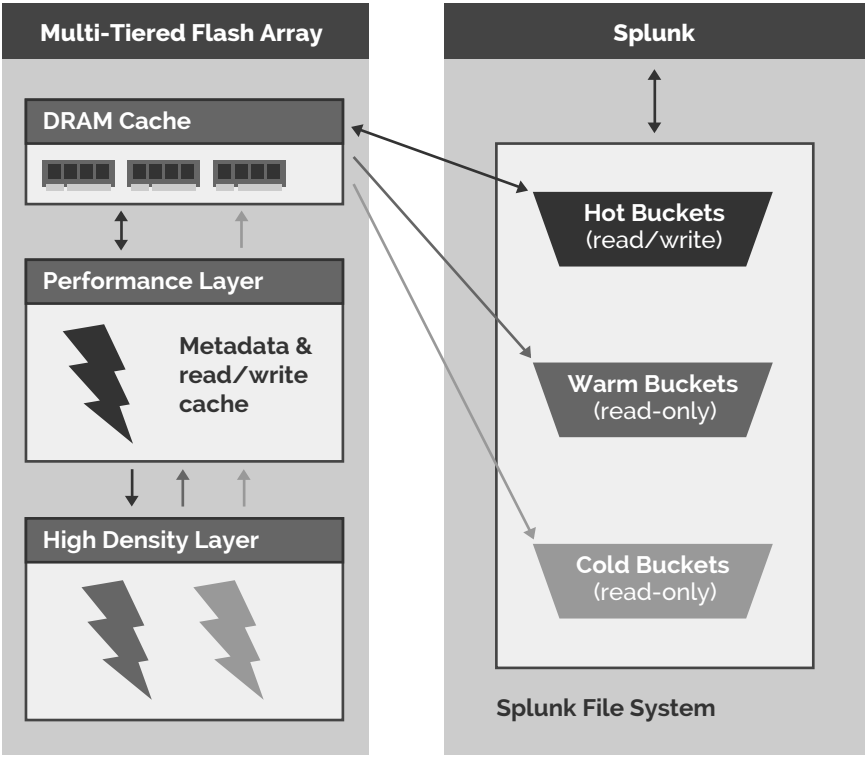


Figure 4-6: Splunk and Tiered Architecture

That's a Wrap!

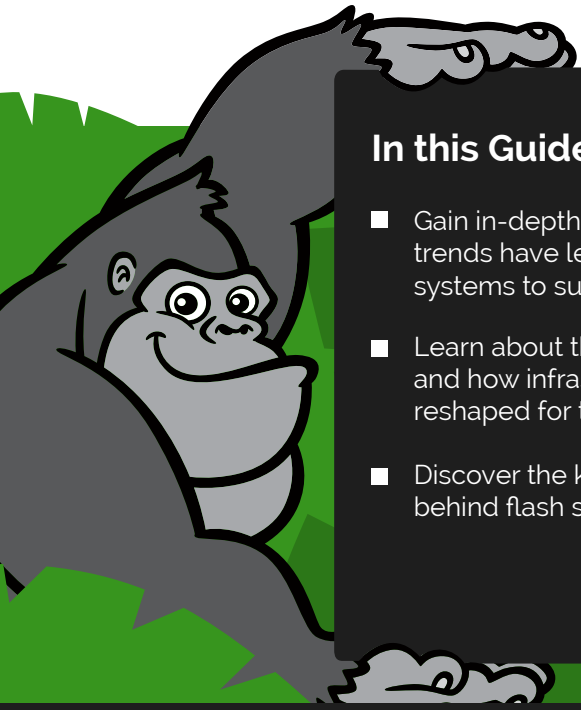
There is no question that all flash arrays offer superior performance to HDD-based arrays; however, the challenge has been meeting density and cost requirements. As the cost of flash storage has decreased, and density and compression have allowed flash density to rival HDD density, the balance has shifted strongly in favor of all flash arrays. Additionally, systems like Splunk and other real-time business analytics systems demand low latency I/O to deliver key business metrics in real-time.

Strap on your exploration gear and join us as we lead you on a journey through the flash and big data jungle!

The world of storage is undergoing a major transformation as flash storage continues to replace legacy spinning disk in the data center. Applications from simple file servers to virtualization to data-heavy workloads are being positively impacted by this renaissance in modern storage. Big Data applications are among those that are surging in the marketplace as organizations of all shapes and sizes seek to convert mountains of data into actionable intelligence.

A confluence of trends has emerged that have become the drivers behind the big data deluge. Social-media driven analytics, sensor-based infrastructure monitoring, machine-generated data, and even the electronics in your home (as a result of the appearance of the Internet-of-Things) are driving the creation of vast quantities of data elements.

Yesterday's spinning storage devices, while able to support vast quantities of data, are wholly inadequate when it comes to the sheer performance requirements for modern applications, and especially for those that support big data. In this book, you will learn about how flash-based storage systems are designed to support exactly these needs.



In this Guide You'll Learn:

- Gain in-depth understanding for how recent trends have led us to the need for flash-based systems to support big data efforts;
- Learn about the evolution of big data systems and how infrastructure architecture is being reshaped for these workloads;
- Discover the key technical characteristics behind flash storage

ISBN 978-1-943952-13-7
90000>
9 781943 952137

ActualTech Media
For bonus content & more!
Visit **www.Gorilla.Guide**