SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2014

Big Data Storage

Apurva Vaidya
Anshul Kosarwal
IGATE

Agenda

- Abstract
- Big Data and its characteristics
- Key requirements of Big Data Storage
- Triage and Analytics Framework
- Big Data Storage Choices
- Hyper scale, Scale Out NAS, Object Storage
- Impact of Flash memory and tiered storage technologies
- Big Data Storage Architecture
- Performance Optimization
- Conclusion

Abstract

Big Data is a key buzzword in IT.

The amount of data to be stored and processed has exploded to a mind boggling degree.

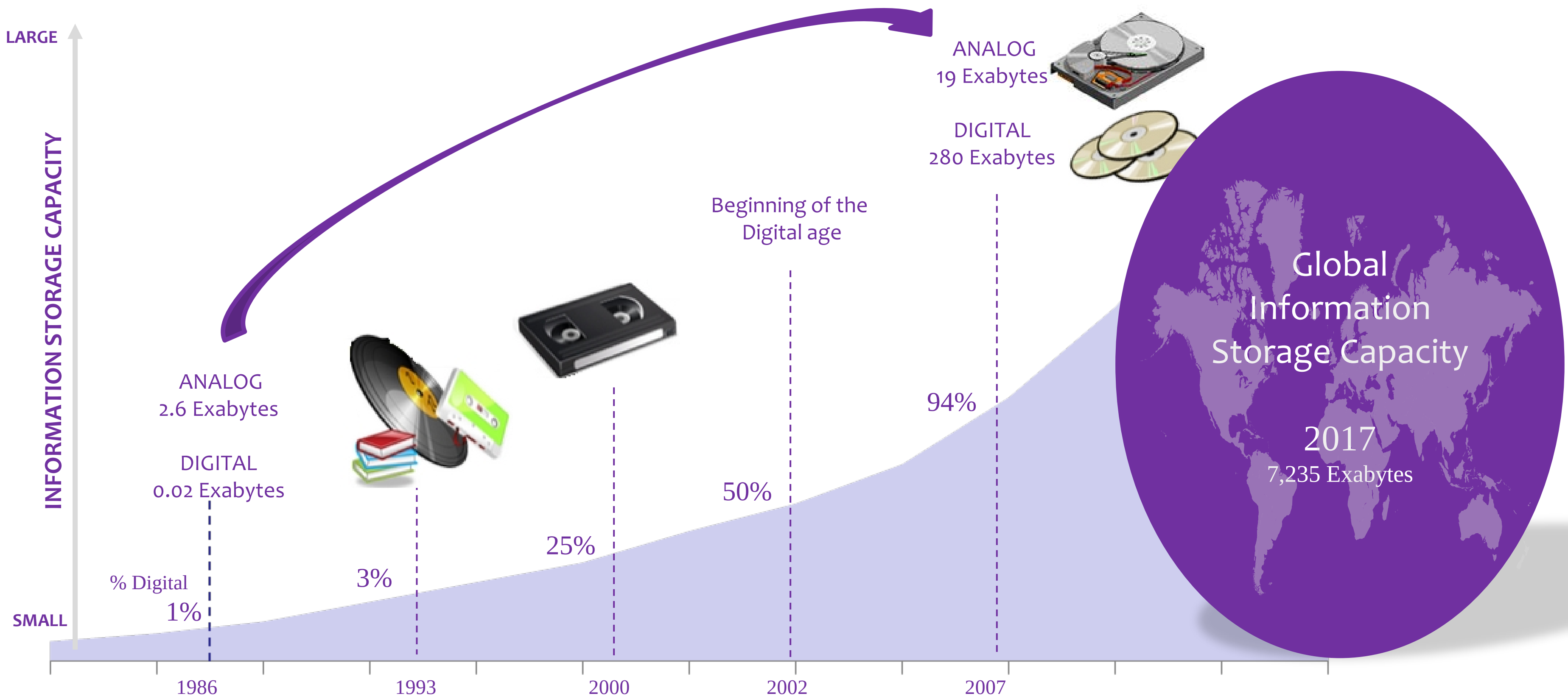
Although "Big Data Analytics" has evolved to get a handle on this information content, the data must be appropriately stored for easy and efficient retrieval.

And for this, proper storage options should be in place to handle BIG DATA


"The pessimist – complains about the wind
The optimist – expects it to change
The leader – *adjusts the sails*"
John Maxwell

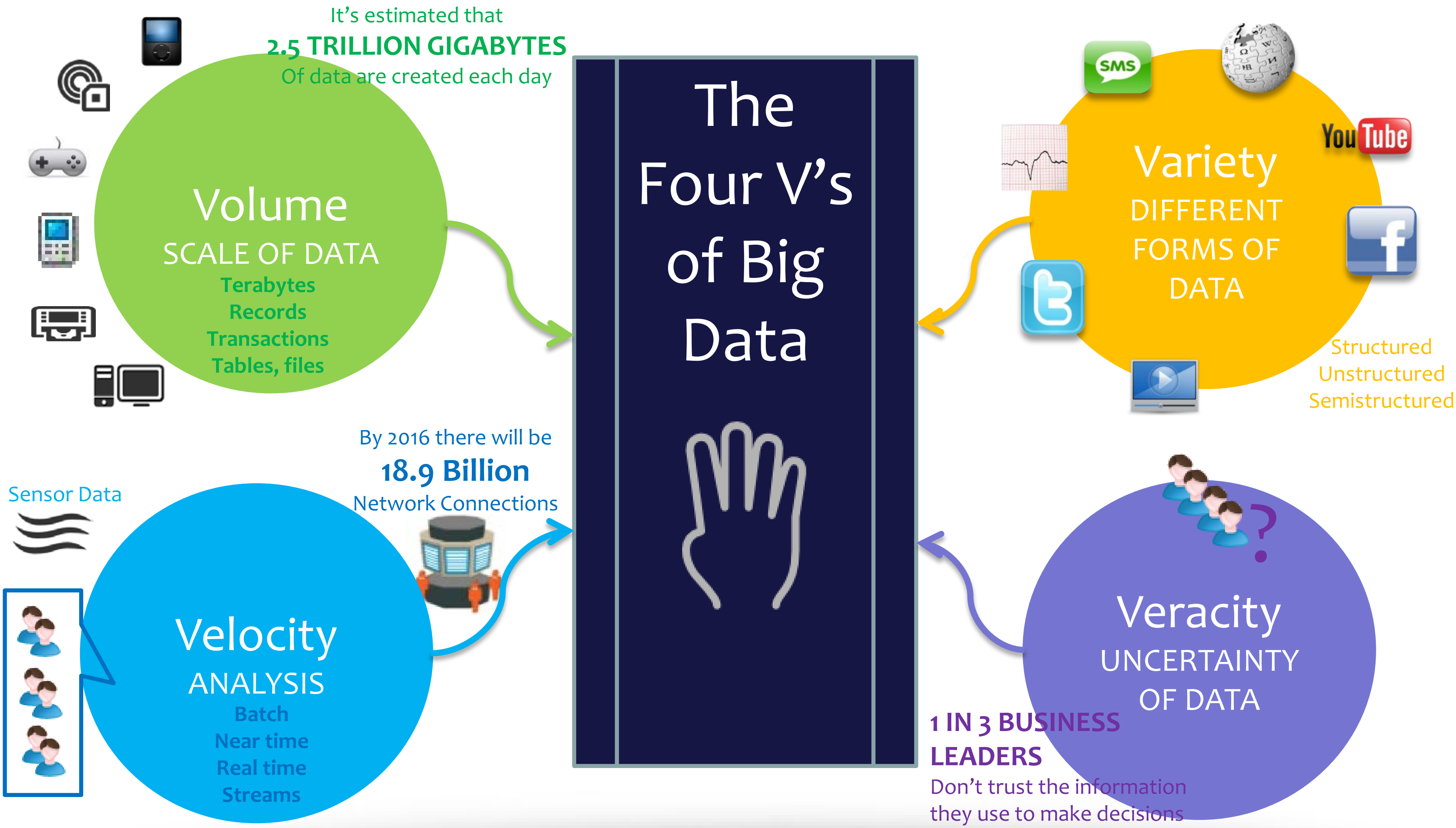
Big Data & Its Characteristics

Storage Capacity



Source: Business Computing World
(http://www.businesscomputingworld.co.uk/storage-predictions-for-2014/)

Key Characteristics of Big Data



Source: http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Prioritizing Things

Concerns

- Data growth
- System performance and scalability
- Network congestion and connectivity

Business Priorities

- Enterprise growth
- Attracting & retaining new customers
- Reducing enterprise costs

Technology Preferences

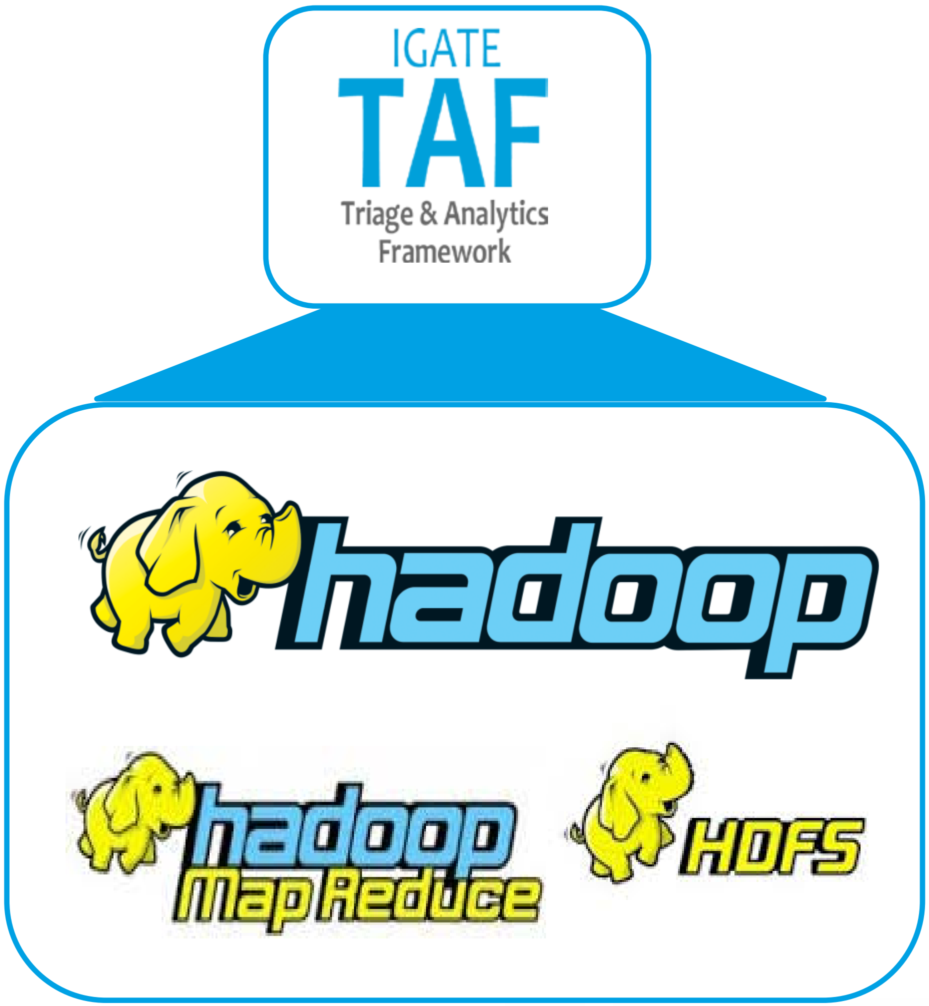
- Analytics/business Intelligence
- Mobile technologies
- Cloud computing

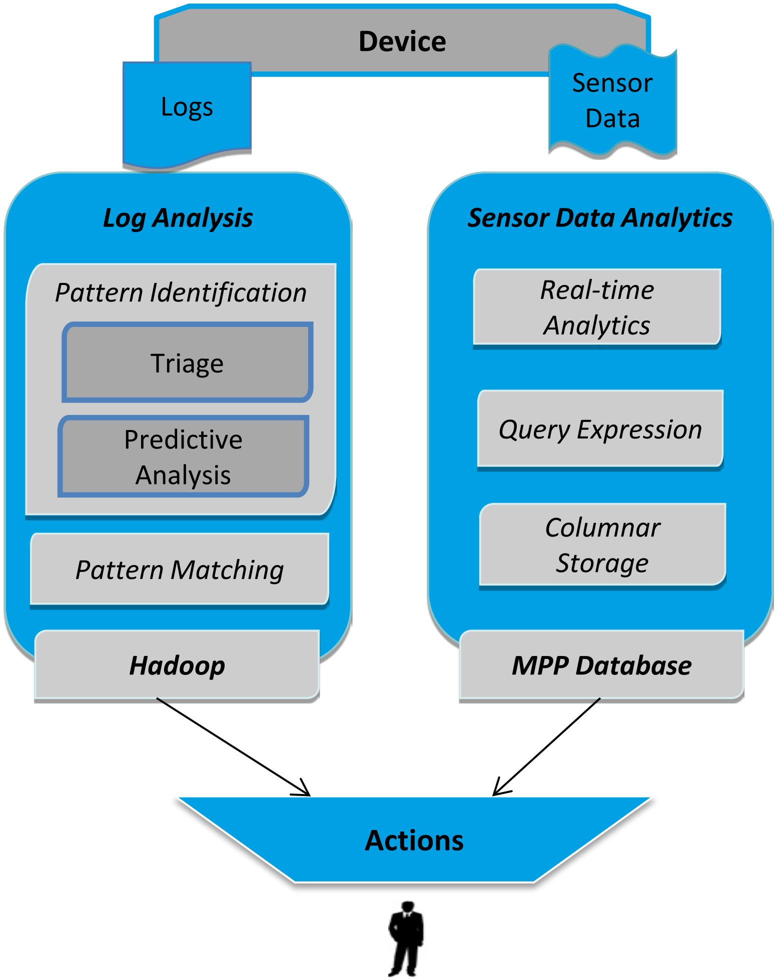
Requirements of Big Data Storage

Key Requirements of Big Data Storage

Triage and Analytics Framework

TAF Overview



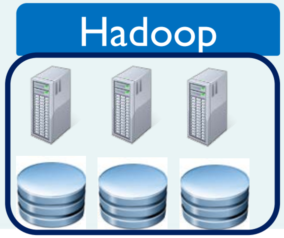


The way we discovered our needs

IGATE TAF

IGATE has its 'Triage and Analytics framework' for proactive diagnostic and predictive analysis

Working Good

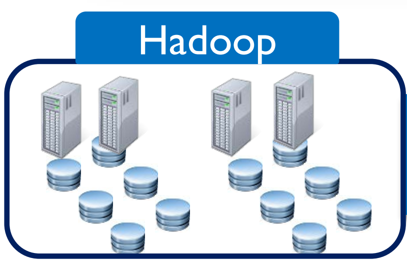


DAS Storage

But as data grew, there was performance drop

May be it was time for us to think about the **storage** that goes below and its scalability

So, in order to OPTIMIZE storage we went on to discover storage options which can **scale**



Scale-out nodes on a Hadoop framework.

Fast for computation
Leverages Individual nodes, servers, CPU, memory and RAM
RAID support

Efficient
For cost and performance

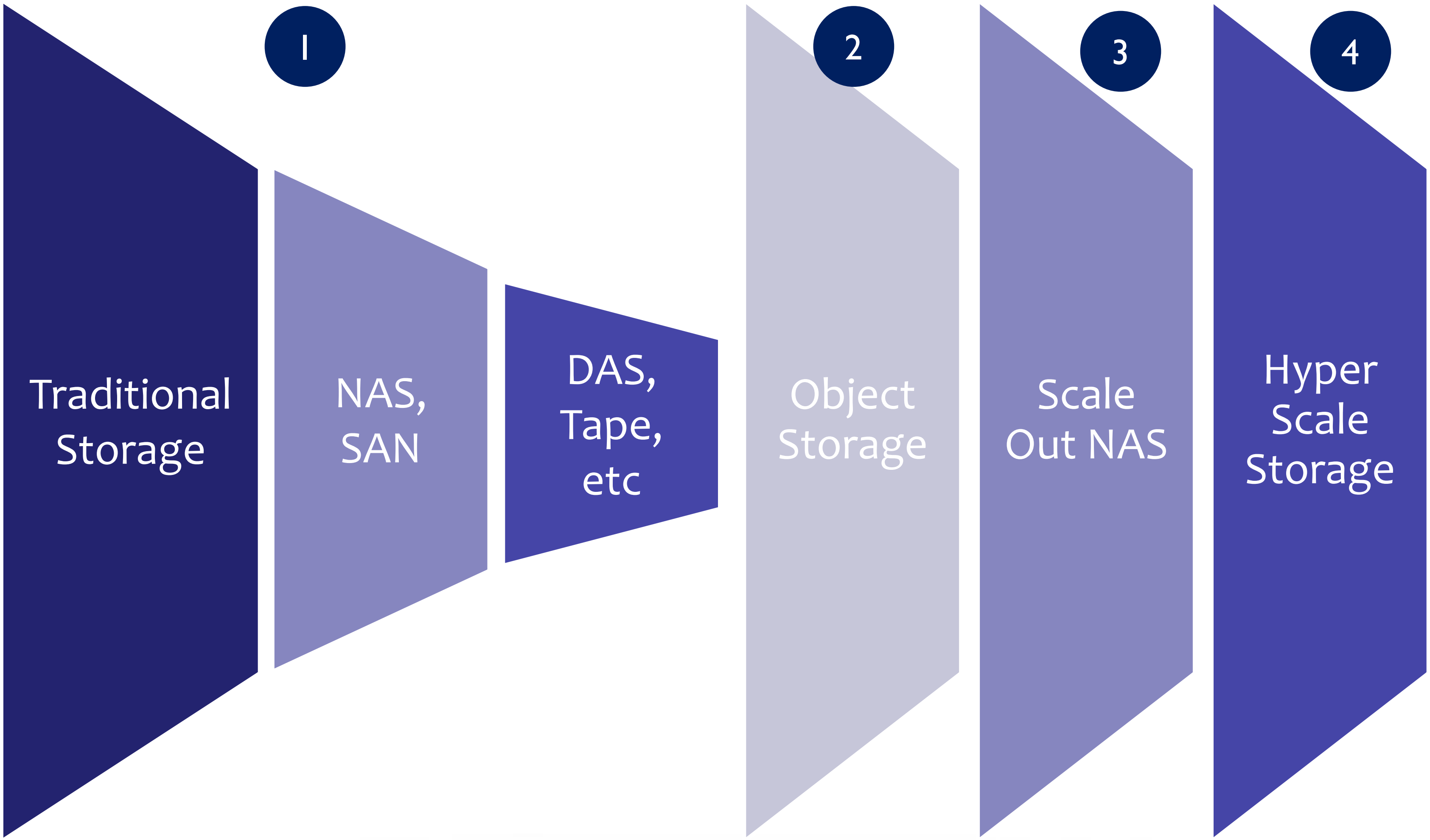
Global Namespace
Accessed and managed as a single system

High Availability

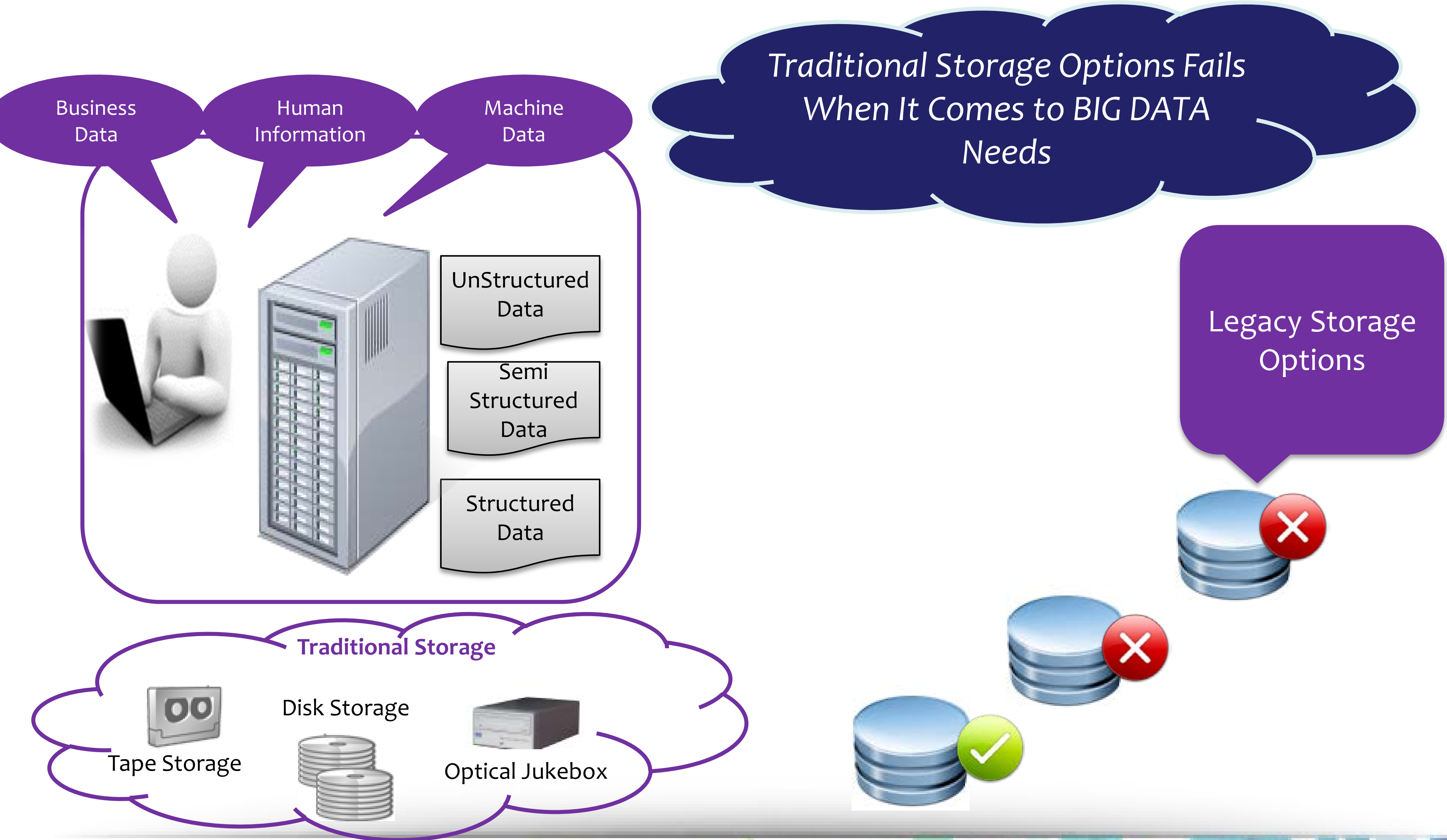
SDC 14

Big Data Storage Choices

Storage Options



Traditional Storage Options



Scale out NAS, Object Storage and HSS

Scale Out NAS

Built to

- ❑ Deliver a flexible and scalable platform for Big Data.
- ❑ Simplify management of big data storage infrastructure.
- ❑ Increase efficiency and reduce costs.

Why it can be useful?

Simple to scale	Scale Out NAS is a very simple to scale platform for Big Data.
Predictable	The performance and reliability is predictable
Efficient	It leverages all the resources in an efficient way
Available	High Availability support is within the FS and hence can be guaranteed
Enterprise-proven	Technology is matured and time tested

Object Storage

Built to

- Optimized object storage systems purpose built for petabytes scale, which also enables organizations to build cost-efficient storage pools for all their unstructured data needs.
- Provide a highly reliable, infinitely scalable, flexible pricing options, and the ability to store, retrieve, and leverage data the way you want to.

It provides?

Maximum Portability

Provide set of interfaces including C++ and Java APIs, REST for web applications and file gateways to support file-based workflows.

Scalable to Petabytes and beyond

The object storage systems are scalable to petabytes of data, billions of files and beyond.

Immutable data

Provides best option for unstructured immutable data

Lowest cost of ownership

It is delivered on commodity based platform for lowest cost of ownership.

Index and Search

Indexing of the metadata allows extremely fast filtering and search capabilities.

Cloud Capability (Multi Tenancy)

Storage can be carved up into virtual containers based on who owns the data.

Hyper Scale Storage

Built for

- Coined to represent the scale-out design of IT systems with the ability to massively expand under a controlled and efficient managed, software-defined framework.
- It enables the construction of storage facilities which caters for very high volumes of processing and manages and store petabytes (PB) or even greater quantities of data.

Key Criteria

Elasticity	It works with a notion of treating data independent from the hardware, making it easy to grow or shrink volumes as needed, with no interruption in services.
Scalability	It supports not just terabytes, but petabytes of data, right from the start.
High performance and availability	It eliminates hot spots, I/O bottlenecks and latency, hence offering very fast access to information.
Standards based and open source	It supports any and every file type, benefiting from the collective collaboration inherent in any open-source system, resulting in constant testing, feedback and improvements.
Leading-edge architecture and design	It supports or includes a complete storage OS stack, hence has a modular, stackable architecture which enables data to be stored in native formats and which also avoid metadata indexing, instead storing and locating data algorithmically.

Analytical Comparison

	Scale Out NAS	Object Storage	Hyper Scale Storage
Scalability	Scale Out NAS caters problem with software management approach., that enables virtualization layer to makes the nodes behave like a single system.	Object storage can scale seamlessly for number of objects, Exabyte capacity and distributed sites.	Hyperscale architectures typically include dozens to hundreds of generic servers running either a clustered application or a hypervisor.
Accessibility / Availability	Assuming that hardware will fail, Scale Out infrastructure is built (on commodity hardware) and designed to deal with a higher rate of hardware failures.	This type of storage supports Omni-protocol support (reconfigurable bit-stream processing in high speed network). It also supports wide choice of protocols and rich application ecosystem.	Automatic replication deliver a high level of data protection and resiliency. It also eliminates hot spots, I/O bottlenecks and latency, hence offering very fast access to information.
Efficiency	Scale-out NAS for big data is intelligent enough to leverage all the resources in storage system, regardless of where they are. By moving data around it optimizes performance /capacity.	Object storage provides highest efficiency with low overhead, no bandwidth kill and lowest management cost.	Hyperscale gives stripped storage which provides rapid and efficient expansion to handle big data in different use cases, such as Web serving or for some database applications.
Reliability	Reliability In Scale Out NAS is achieved by features like hardware RAID, redundant hot-swappable power supply modules, and full access to data even if a compute node goes down.	Provides all round data protection with smart replication.	Reliability with hyper scale can be achieved by installing number of hyper scale manager servers.
Performance	Scale out storage system expands to maintain functionality and performance as it grows in-line with data centre demands as business needs change over time.	Object storage gives high performance for throughput, IOPS & low latency.	The hyperscale auto-balance workloads (can also obtain priority), when nodes are added.

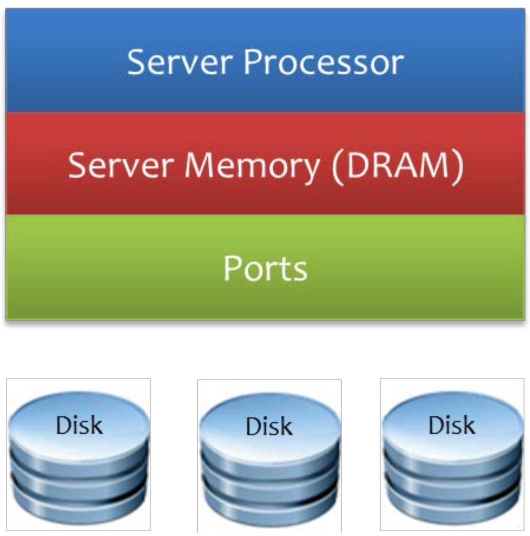
Impact of Flash memory and tiered storage technologies

Current Storage Architectures

Direct Access Storage Architecture

The problems with this type of architecture:

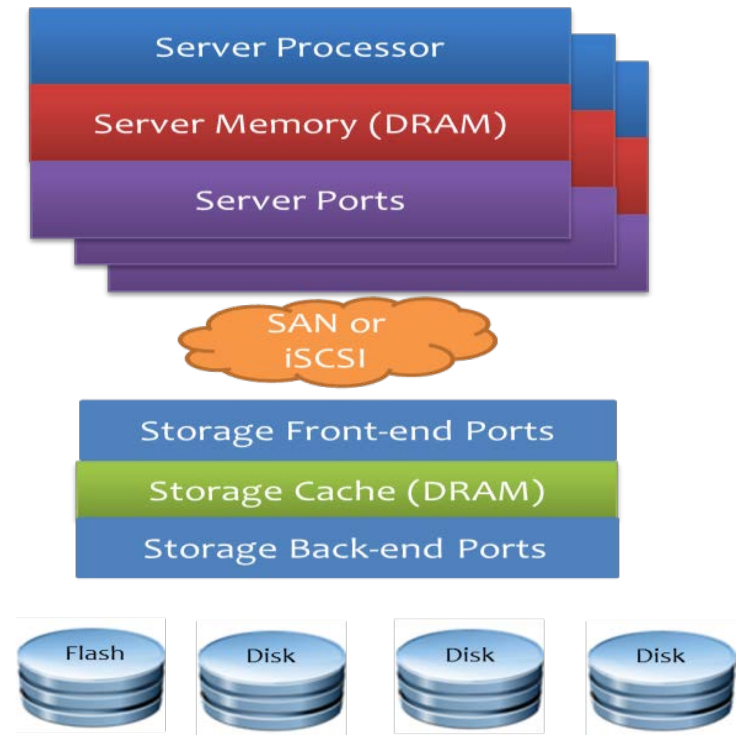
- Underutilized storage
- Localization of data
- Dependence on server OS for replication, etc.



Networked Storage Architecture

The major problems with this model is:

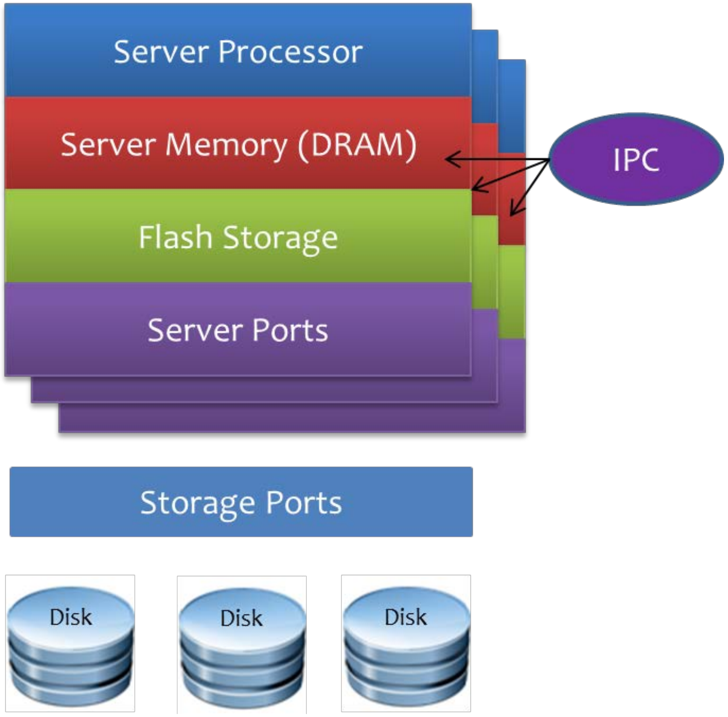
- Explosive increase in random I/O coming from high performance server systems is the least efficient for mechanical hard disk drives.



Future System and Storage Architectures

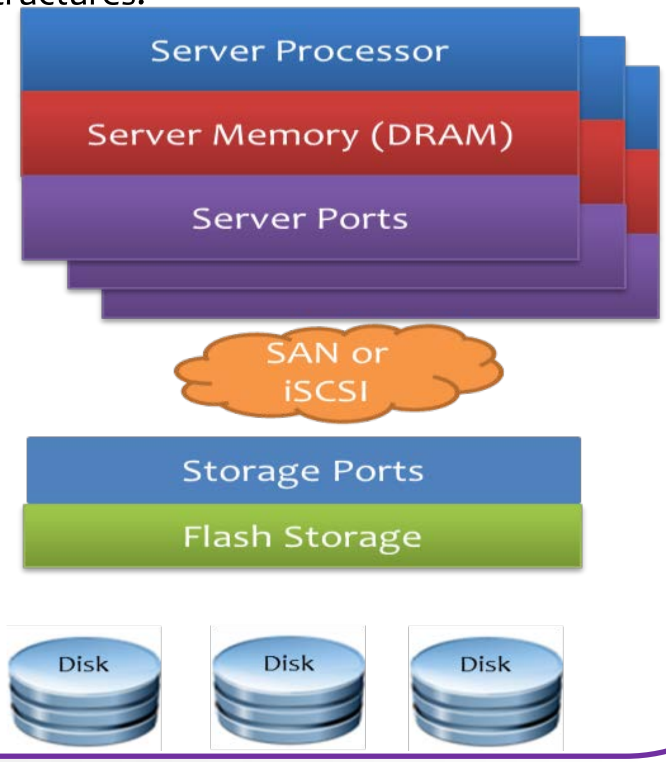
Flash on Server (FOS)

This architectural approach is revolutionary and will provide the largest benefit but will require the greatest change in infrastructure.



Flash on Storage Controller (FOSC)

This approach is more evolutionary and will be easier to incorporate into existing application infrastructures.



Big Data Storage Architecture

Considerations

Three questions every business should consider when architecting the storage component of a big data plan:

What kind of storage will you use?

- Own data center or data warehouse or go with a cloud system.
- Newer storage technologies: object storage, software-defined storage, and data-defined storage.

How much data is there and are you planning for growth?

- The amount of data produced, collected, and managed is going to grow, and it may do so exponentially.

What kind of data you have and what is the intent of storing data?

- Content driven or analytical application
- Structured/Unstructured
- In the big data era, it's not uncommon to encounter a "bigger is better" mindset.

Right storage strategy depends upon identifying and understanding the data sets

Big Data Storage Architecture Decisions

Protecting Data Sets

- RAID, HDFS and more

Capacity Intensive vs Compute Intensive

- Loads of data vs processing

Scale Out vs Object

- Global namespace file system for easy management or object storage's use of metadata

Compliance and Security needs

- Regulatory compliance(HIPAA, etc.) and security(PCI standards)

Get the right requirements to get the Architecture right.

Optimizations for Big Data

Optimizing Hadoop for MapReduce

dfs.block.size(renamed to *dfs.blocksize* in Hadoop 2.x)

- Specifies the size of data blocks in which the input data set is split. Default is 64MB.

mapred.compress.map.output

- Specifies whether to compress output of maps before sending across the network. Default is ‘false’. Uses SequenceFile compression.

mapred.map/reduce.tasks.speculative.execution

- Multiple instances of mao or reduce tasks can be run in parallel. Default is ‘true’. The output of the task which finishes first is taken and the other task is killed.

mapred.tasktracker.map/reduce.tasks.maximum

- The maximum number of map/reduce tasks that will be run simultaneously by a task tracker. Default is 2.

SDC 14

Optimizing Hadoop for MapReduce

io.sort.mb

- The size of in-memory buffer (in MBs) used by map task for sorting its output. Default is 100MB.

io.sort.factor

- The maximum number of streams to merge at once when sorting files. This property is also used in reduce phase. Default is 10.

mapred.job.reuse.jvm.num.tasks

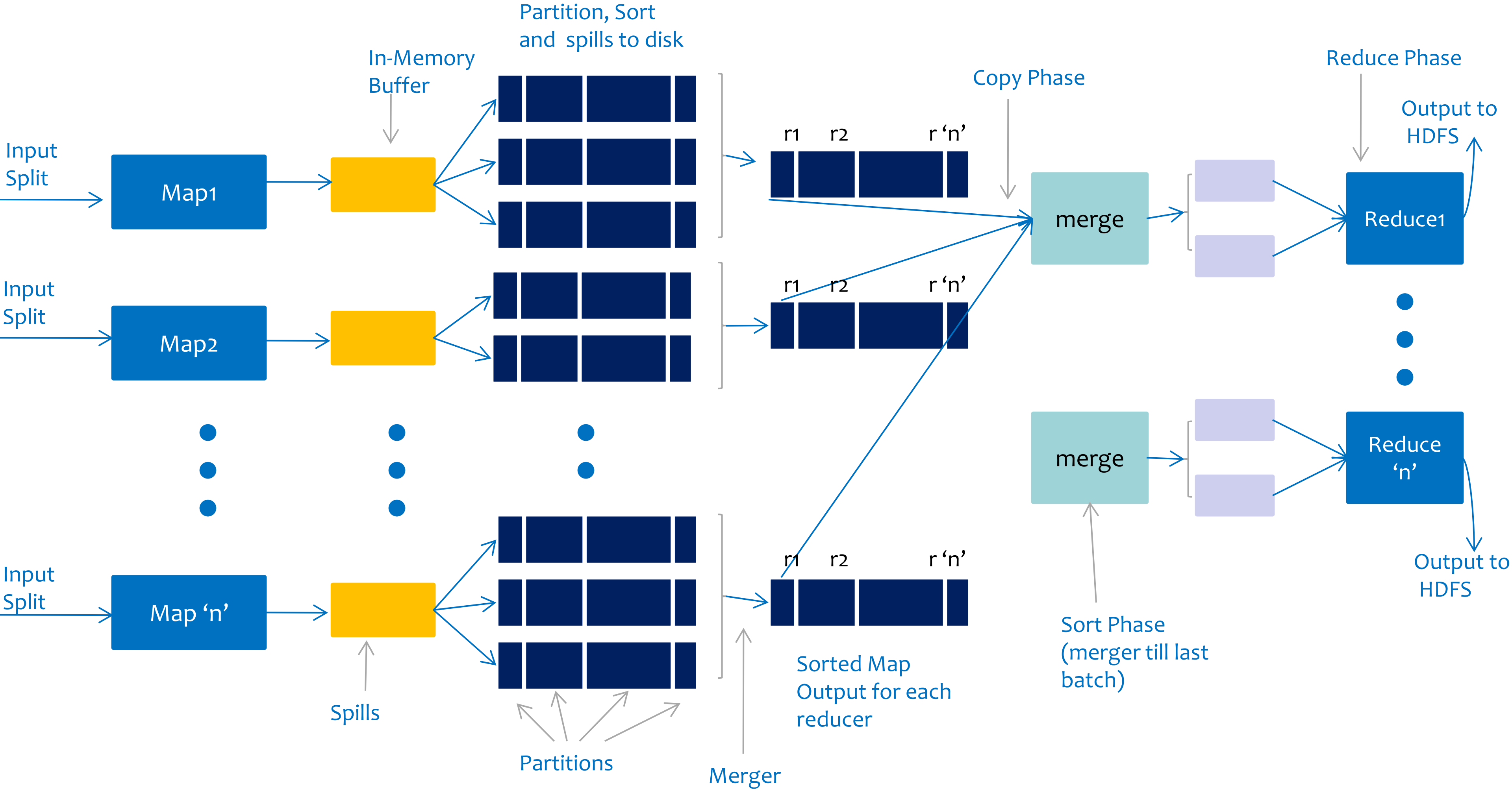
- The maximum number of tasks to run for a given job per JVM on a tasktracker. A value of –1 indicates no limit: the same JVM may be used for all tasks for a job. Default is 1.

mapred.reduce.parallel.copies

- The number of threads used to copy map outputs to the Reducer. Default is 5.

SDC 14

Optimizing Hadoop for MapReduce



Efficient MapReduce for problem solving

- MapReduce and HDFS are complex distributed systems that run arbitrary user code. There are no fixed rules to achieve optimal performance.
- MapReduce is a framework. Solution should fit in the framework.
- Good programmers can produce bad MapReduce algorithms.

Algorithmic Performance

- Algorithmic complexities, Data Structures, Join strategies

Implementation Performance

- Follow general Java Best Practices
- MapReduce Patterns(Numerical Summarizations, Filtering, Reduce Side Join)

Physical Performance

- Storage, CPU, Hadoop Environment, etc.

Conclusion

Conclusion

Data is at the heart of business and life in 2014; survival and success for both organization and individuals increasingly depends on the ability to access and act on all relevant information. The key is to craft effective strategies to distill raw data into meaningful, actionable analytics. The payoff can include better, smarter, faster business decisions- creating a truly agile organization and empowering people with the data they need.

Any analysis of Big Data will require a new kind of storage solution, one that can support and manage unstructured and semi-structured data as readily as if it were structured information.

Hyperscale computing and storage leverages vast numbers of commodity servers with direct-attached storage (DAS) to ensure scalability, redundancy and scale, and provide the input/output operations per second (IOPS) necessary to deliver data immediately to analytics tools and the end users who use them. Companies ready to embrace this high-performance yet cost-effective solution will see business benefits, including better and faster decision making improved customer relationships and retention, and a clear competitive advantage.

Also, an effective and efficient MapReduce implementation can make a lot of performance difference. Hadoop parameters can be configured and tweaked to suit the requirements of problem solving.

References

1. ***"The Impact of Flash on Future System and Storage Architectures"*** ; Author: David Floyer; http://wikibon.org/wiki/v/The_Impact_of_Flash_on_Future_System_and_Storage_Architectures; 2014.
2. ***"Hyper scale and Software-led Infrastructure"***; Author: David Floyer; http://wikibon.org/wiki/v/Hyperscale_and_Software-led_Infrastructure; 2013
3. ***"Ten Properties of the Perfect Big Data Storage Architecture";*** Author: Dan Woods; http://www.forbes.com/sites/danwoods/2012/07/23/ten-properties-of-the-perfect-big-data-storage-architecture/; 2012.
4. ***"A look at a 7,235 Exabyte world";*** Author: Larry Dignan; http://www.zdnet.com/a-look-at-a-7235-exabyte-world-7000022200/
5. ***"High Performance Storage For Hyper scale Data Centers"***; Author: George Crump; http://www.storage-switzerland.com/Articles/Entries/2013/5/28_High_Performance_Storage_For_Hyperscale_Data_Centers.html; 2013.
6. ***"What Is A Hyper scale Data Center";*** Author: George Crump; http://www.storage-switzerland.com/Articles/Entries/2013/5/20_What_Is_A_Hyperscale_Data_Center.html
7. ***"An object storage device for a data overload";*** Author: Randy Kerns; http://searchdatacenter.techtarget.com/feature/An-object-storage-device-for-a-data-overload; 2014.
8. ***"From TPC-C to Big Data Benchmarks: A Functional Workload Model"***; Author: Yanpei Chen, Francois Raab and Randy Kartz.
9. ***"TPC Benchmark B Standard Specification";*** Revision 2.0. http://www.tpc.org/tpca/spec/tpcb_current.pdf, 1994.
10. ***"Tips for improving MapReduce Performance";*** Todd Lipcon; http://blog.cloudera.com/blog/2009/12/7-tips-for-improving-mapreduce-performance/
11. "A Beginer's guide to Next Generation Object Storage"; Author: Tom Leyden; http://www.ddn.com/pdfs/BeginnersGuideToObjectStorage_whitepaper.pdf


Q&A