
TOMORROW starts here.
cisco
Cisco live!

Designing Big Data Clusters with Cisco UCS and Nexus

BRKAPP-2033

Sean McKeown

Technical Solution Architect

Cisco live!

Agenda

- Big Data Concepts and Overview
  - Enterprise data management and big data
  - Problems, Opportunities and Use case examples
  - Hadoop, NOSQL and MPP Architecture concepts
- Hadoop and the Network
  - Job types and their network traffic patterns
  - Network characteristics of the cluster
  - Impact of QoS
- Cisco UCS for Big Data
  - Building a big data cluster with the UCS Common Platform Architecture (CPA)
  - UCS Networking, Management, and Scaling for big data
- Q&A

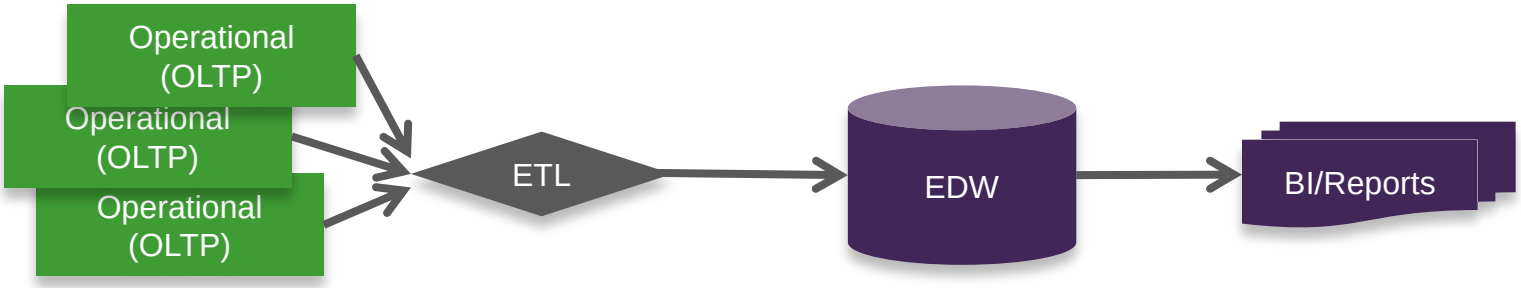
"Life is unfair, and the unfairness is distributed unfairly."

-Russian proverb

Big Data Concepts and Overview

- Enterprise data management and big data
- Problems, Opportunities and Use case examples
- Hadoop NOSQL and MPP Architecture concepts

Traditional Enterprise Data Management



Online Transactional Processing

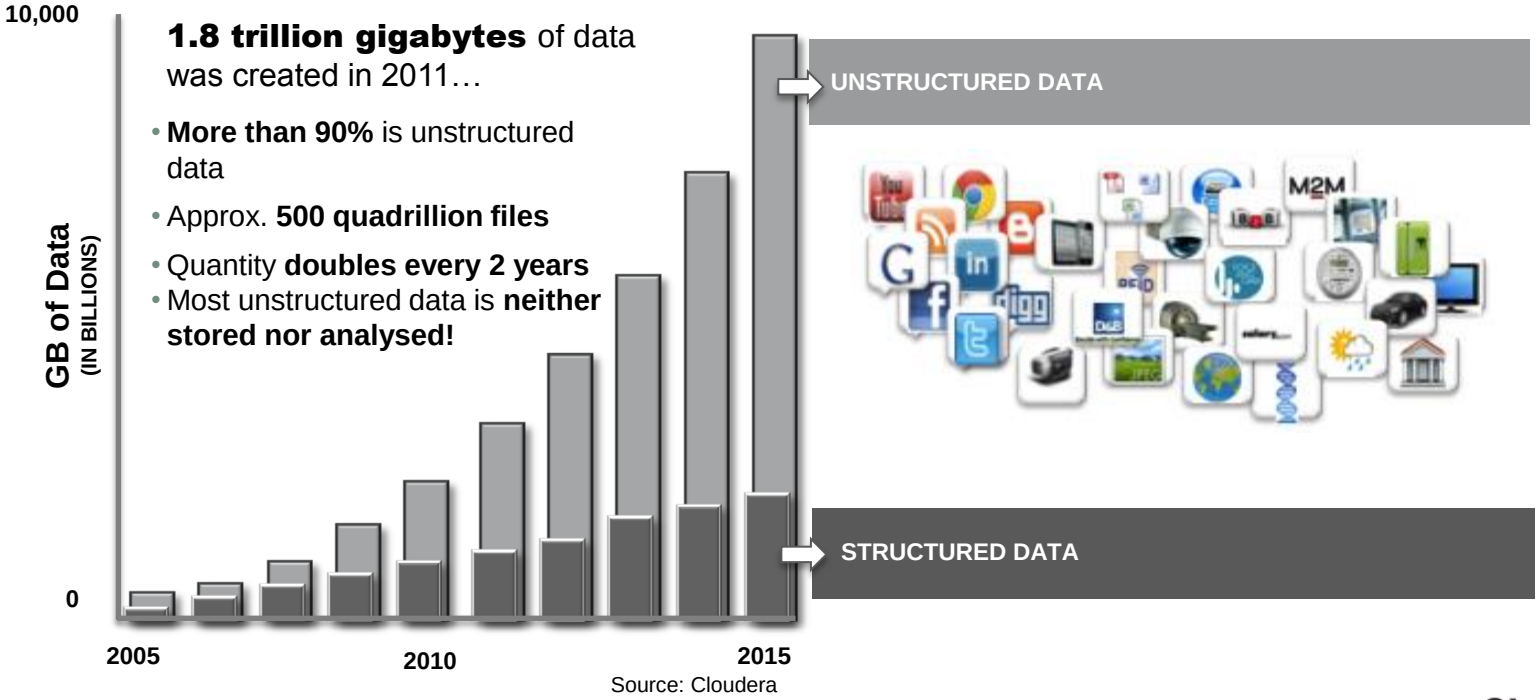
Extract, Transform, and Load (batch processing)

Enterprise Data Warehouse

Business Intelligence

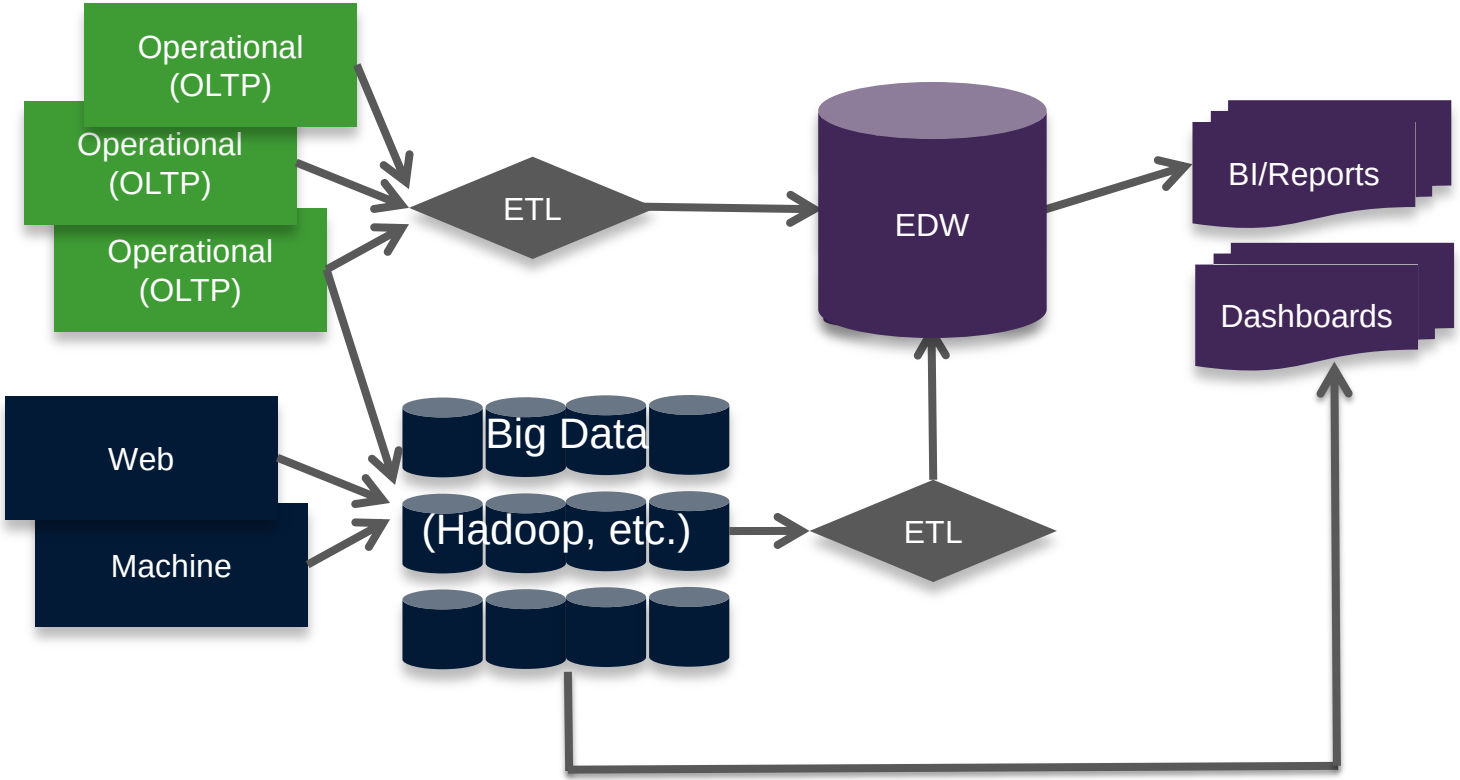
So what has changed?

The Explosion of Unstructured Data

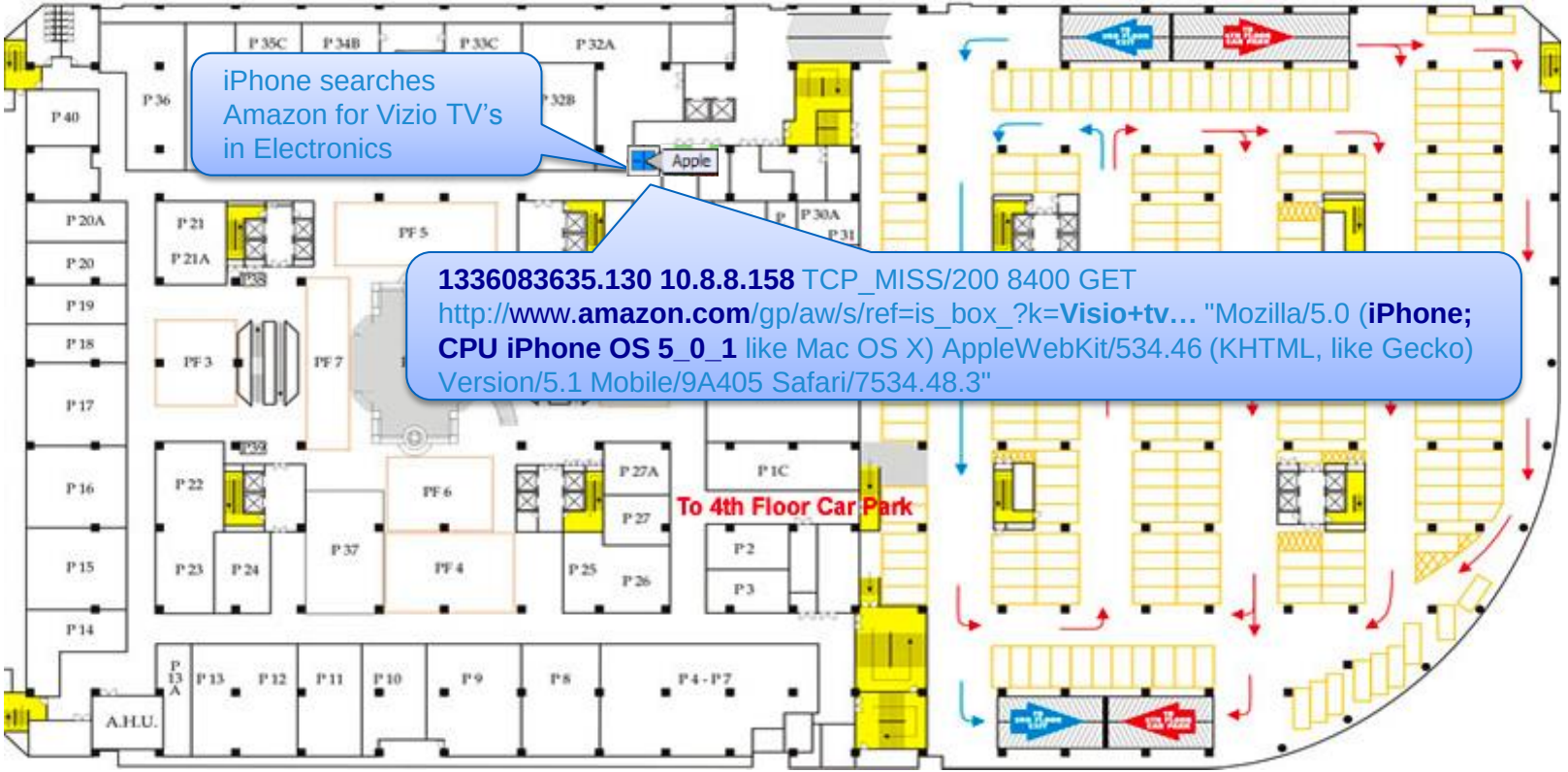


Source: Cloudera

Enterprise Data Management with Big Data

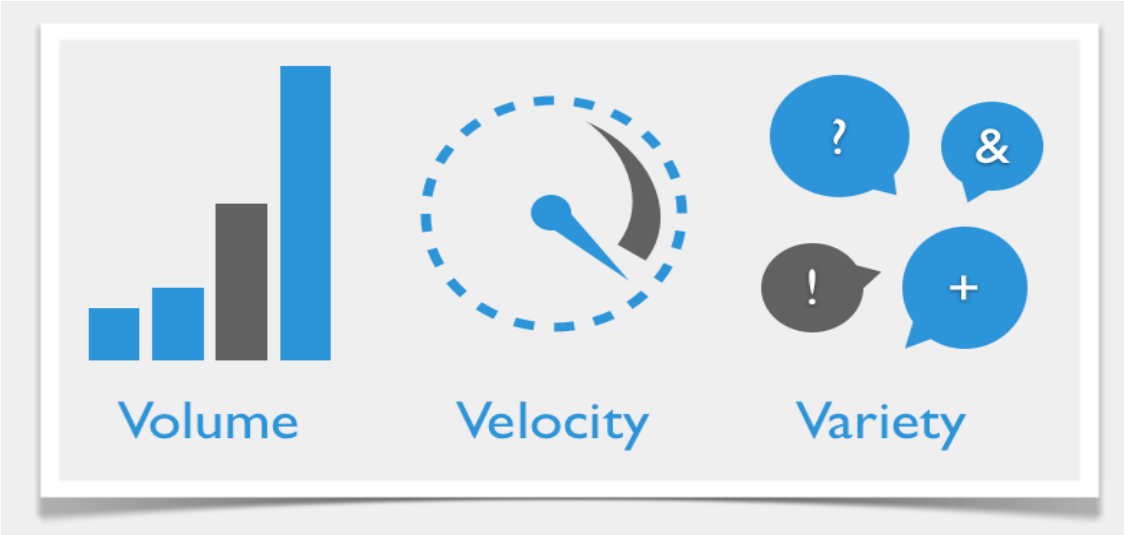


Example: Web and Location Analytics



What is Big Data?

For our purposes, big data refers to distributed computing architectures specifically aimed at the “3 V’s” of data: Volume, Velocity, and Variety



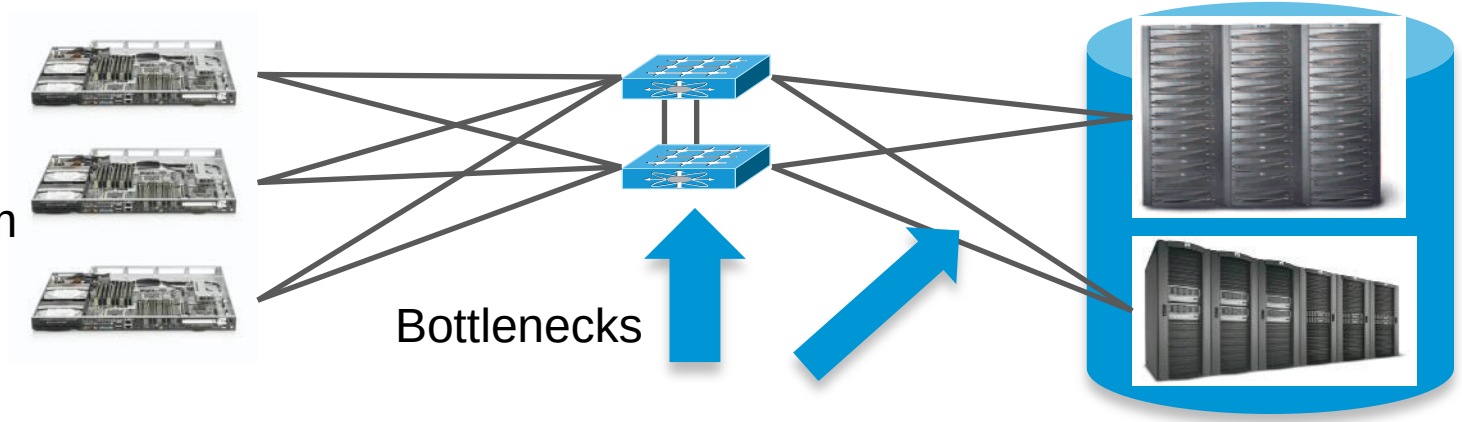
Big Data and Key Infrastructure Attributes

(What big data isn't)

- Usually not virtualised (hypervisor only adds overhead)
- Usually not blade servers (not enough local storage)
- Usually not highly oversubscribed (significant east-west traffic)
- Usually not SAN/NAS (see subsequent slides)

Classic NAS/SAN vs. new scale-out DAS

Traditional – separate compute from storage



New – move the compute to the storage

$$$

Low-cost, DAS-based, scale-out clustered filesystem

Big Data Software Architectures and Design Considerations

Three Common Big Data Architectures

DATASTAX

MarkLogic

NoSQL

Fast key-value store/retrieve in real time

HBASE

PARACCEL

Greenplum

MPP Relational Database

Scale-out BI/DW

MAPR TECHNOLOGIES

PIVOTAL HD

Hadoop

Heavy lifting, batch processing

cloudera

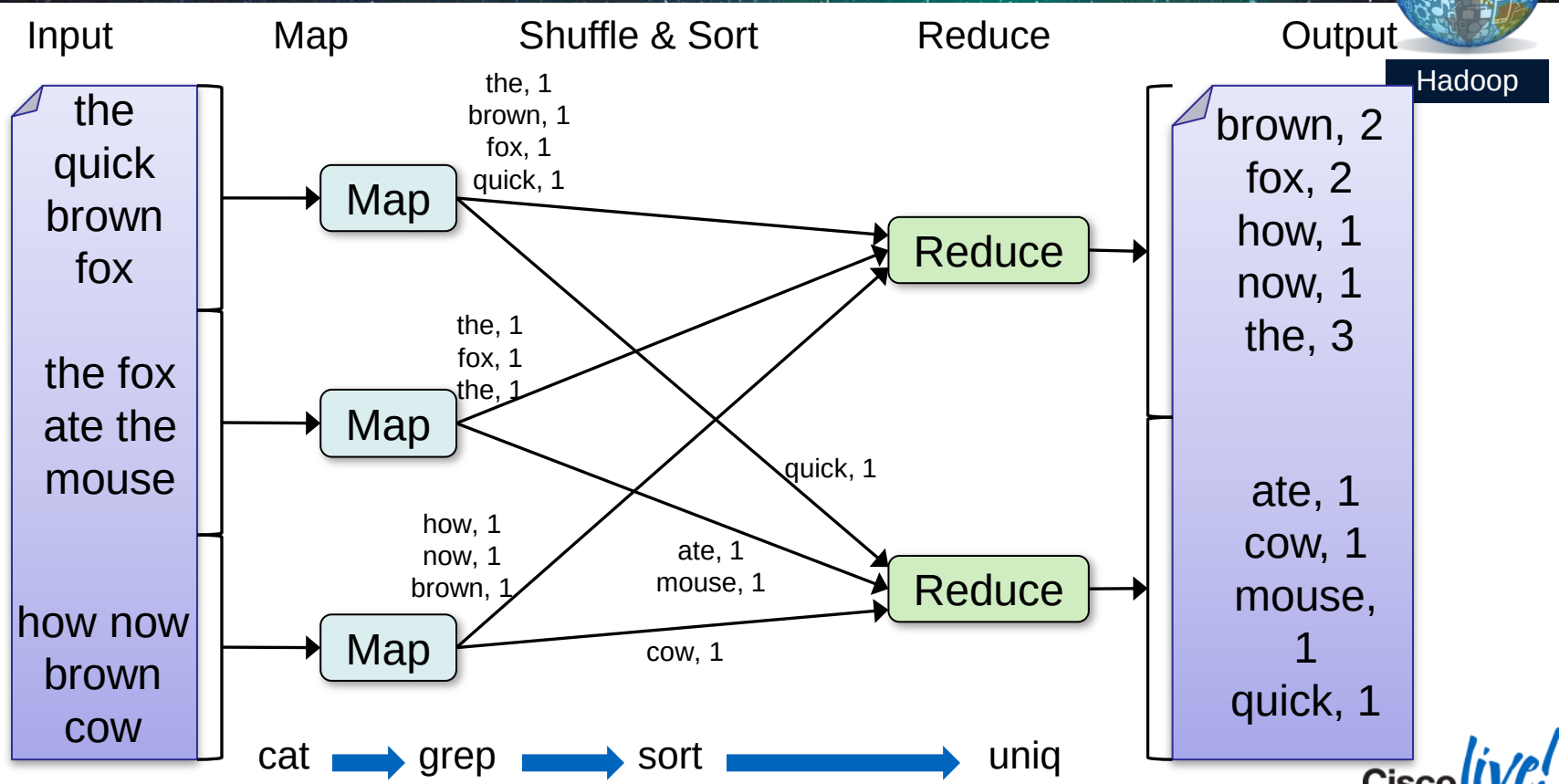
intel

hortonworks

What Is Hadoop?

- Hadoop is a distributed, fault-tolerant framework for storing and analysing data
- Its two primary components are the Hadoop Filesystem (HDFS) and the MapReduce application engine

MapReduce Example: Word Count



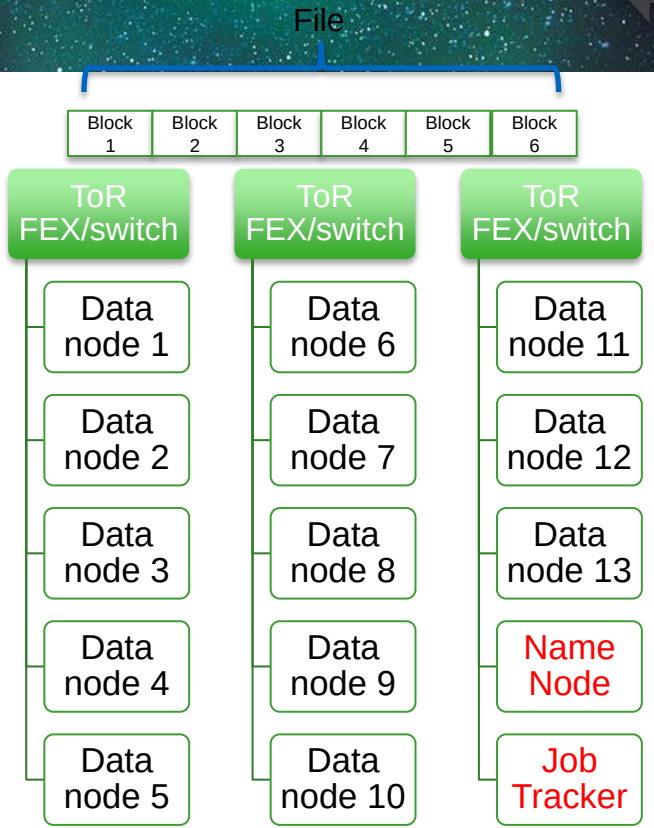
"Failure is the defining difference between distributed and local programming"

- Ken Arnold, CORBA Designer

Hadoop Components and Operations

Hadoop Distributed File System

- Scalable & Fault Tolerant
- Filesystem is distributed, stored across all data nodes in the cluster
- Files are divided into multiple **large blocks** – 64MB default, typically 128MB – 512MB
- Data **is stored reliably.** Each block is replicated 3 times by default
- Types of Nodes
  - Name Node - Manages HDFS
  - Job Tracker – Manages MapReduce Jobs
  - Data Node/Task Tracker – stores blocks/does work



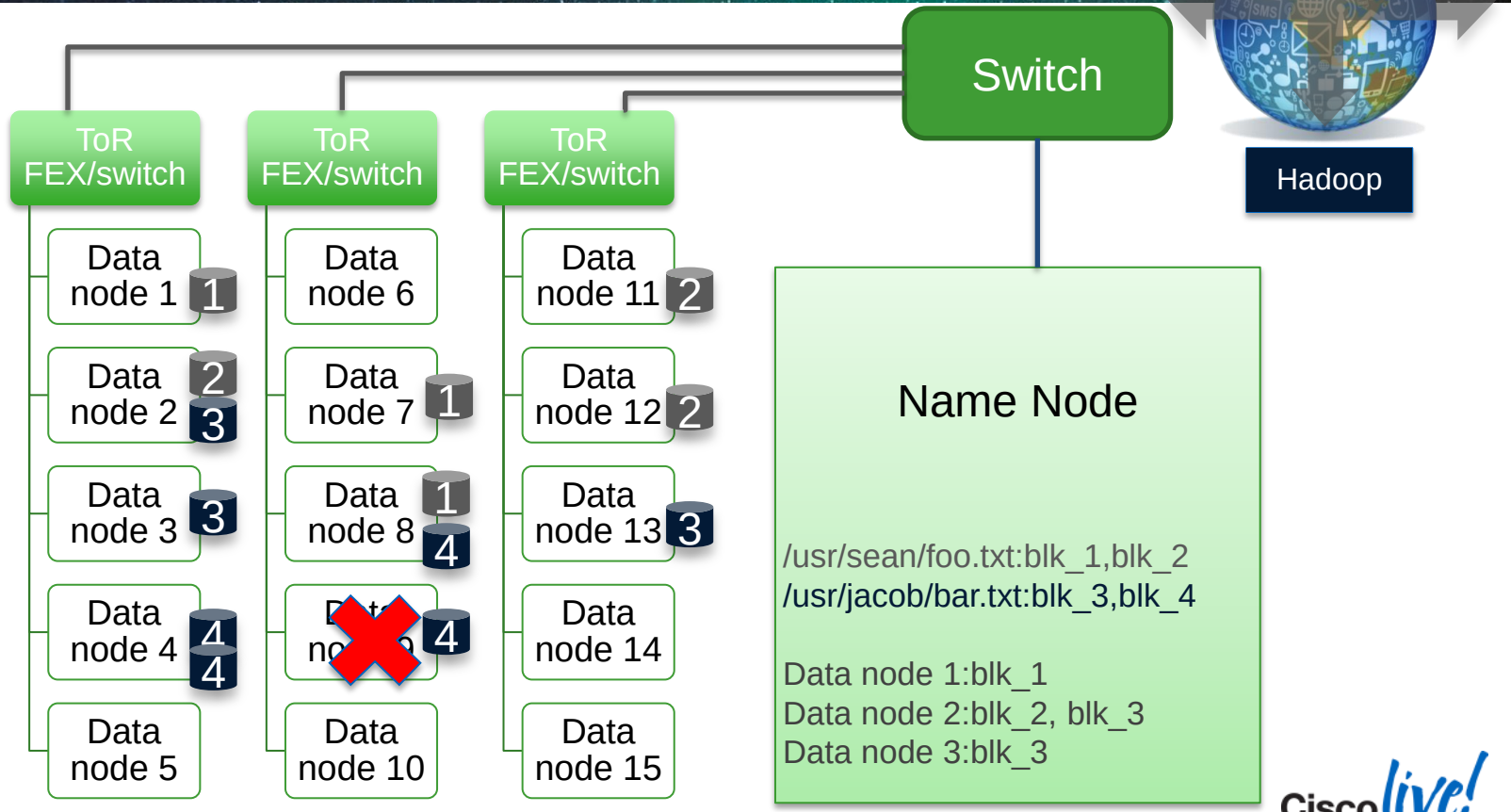
Hadoop

Why Replicate?

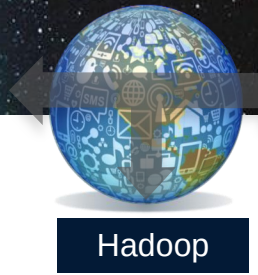
Two Key Reasons

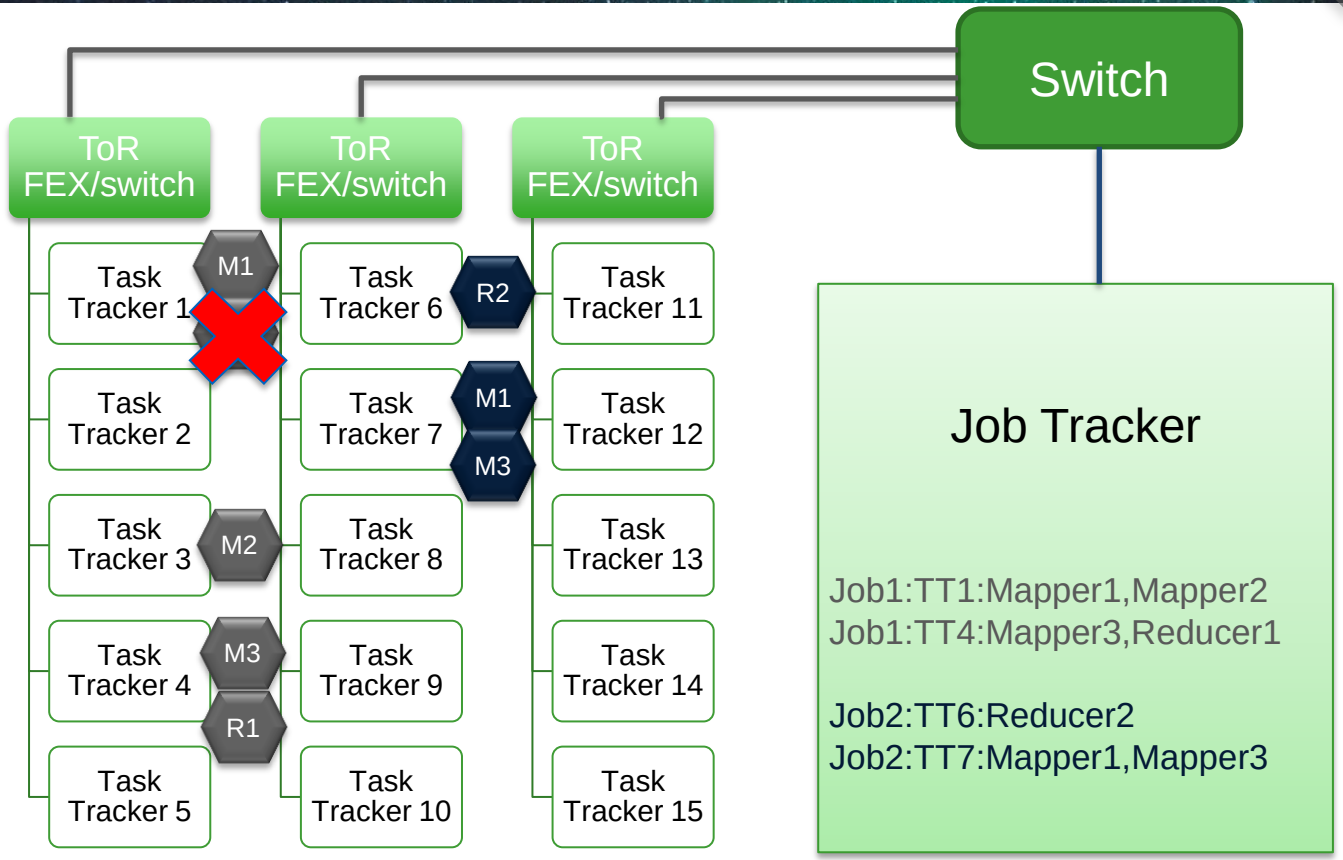
1. Fault tolerance
2. *Increase data locality* (we'll come back to this one in the network section)

HDFS Architecture



MapReduce Architecture



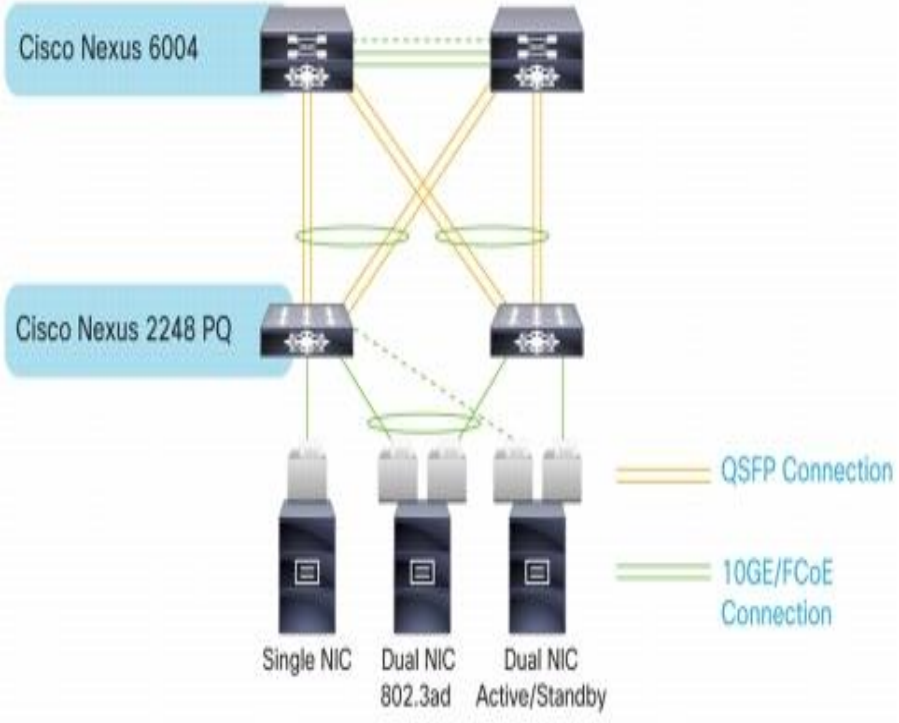


Hadoop and the Network

- Job types and their network traffic patterns
- Network characteristics of the cluster
- Impact of QoS

Hadoop Network Design

- The network is the fabric – the ‘bus’ - of the ‘supercomputer’
- Big data clusters often create high east-west any-to-any traffic flows compared to traditional transactional networks
- Homogeneity of cluster nodes helps simplify design requirements
- Over the time clusters will likely have multi-user, multi-workload behaviour
  - Security, SLA, QoS relevant



Hadoop Network Traffic Types

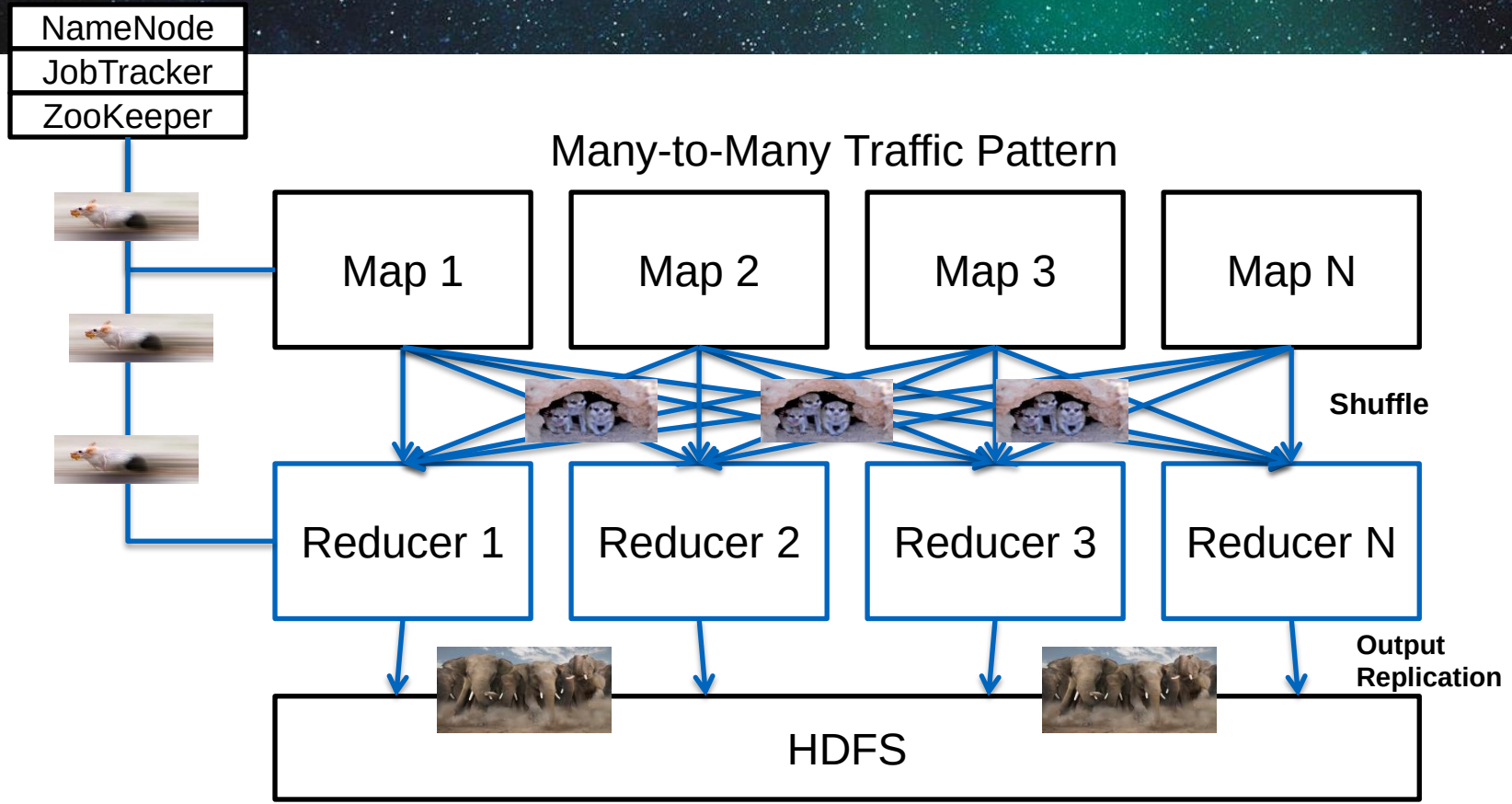
Small Flows/Messaging
(Admin Related, Heart-beats, Keep-alive, delay sensitive application messaging)

Small – Medium Incast
(Hadoop Shuffle)

Large Flows
(HDFS egress)

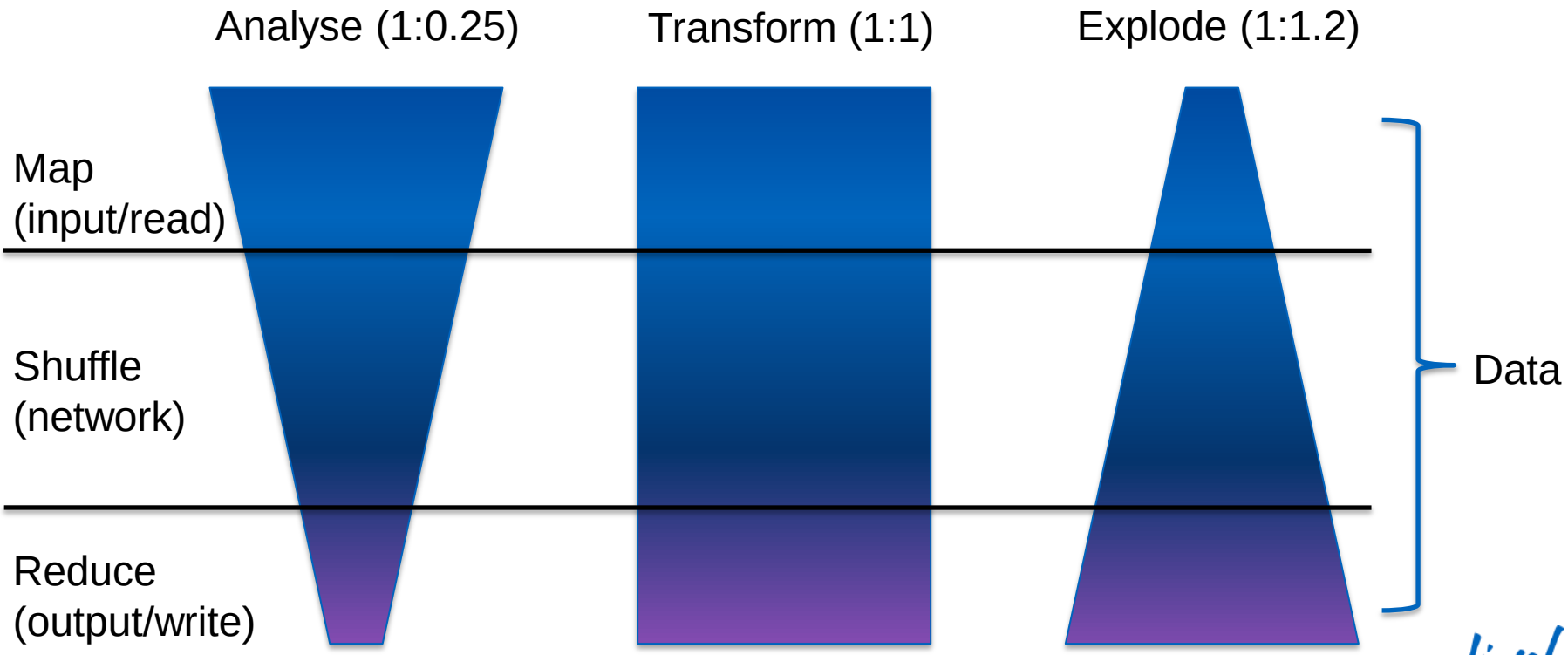
Large Pipeline
(Hadoop Replication)

Map and Reduce Traffic



Typical Hadoop Job Patterns

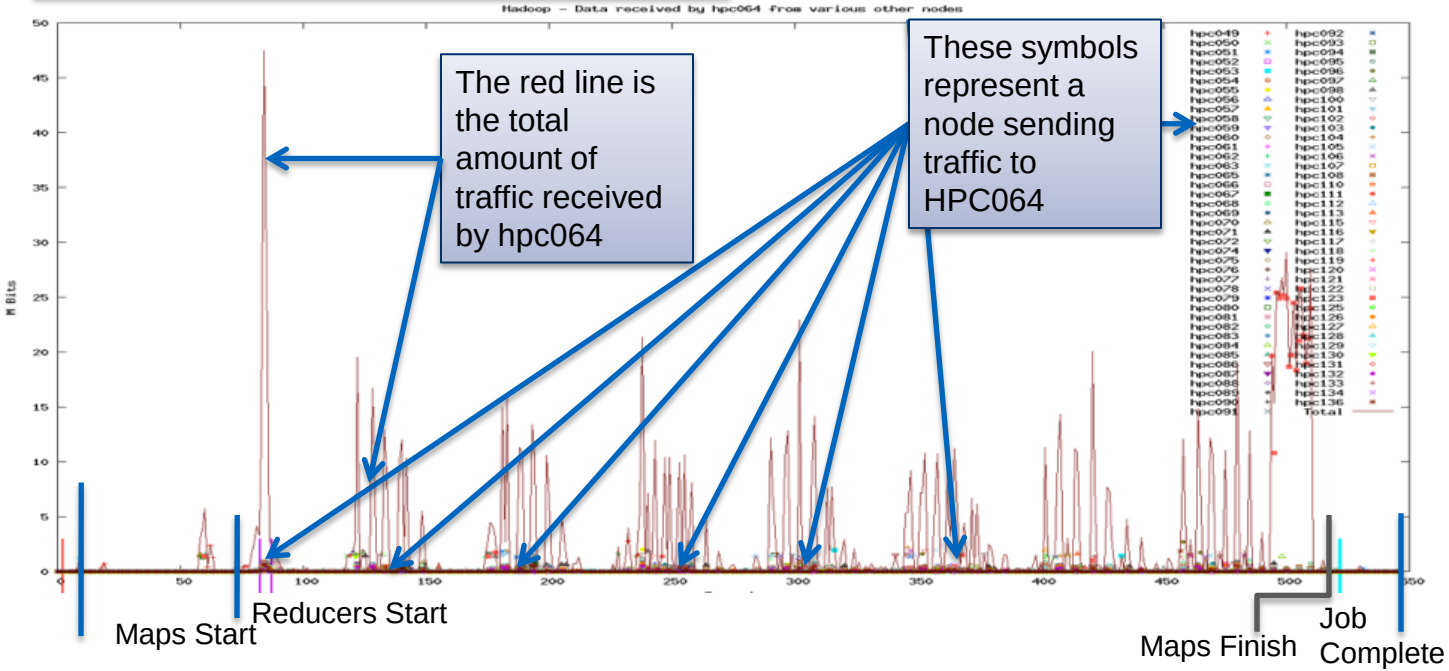
Different workloads can have widely varying network impact



Analyse Workload

Wordcount on 200K Copies of complete works of Shakespeare

Note:
Due the combination of the length of the Map phase and the reduced data set being shuffled, the network is being utilised throughout the job, but by a limited amount.

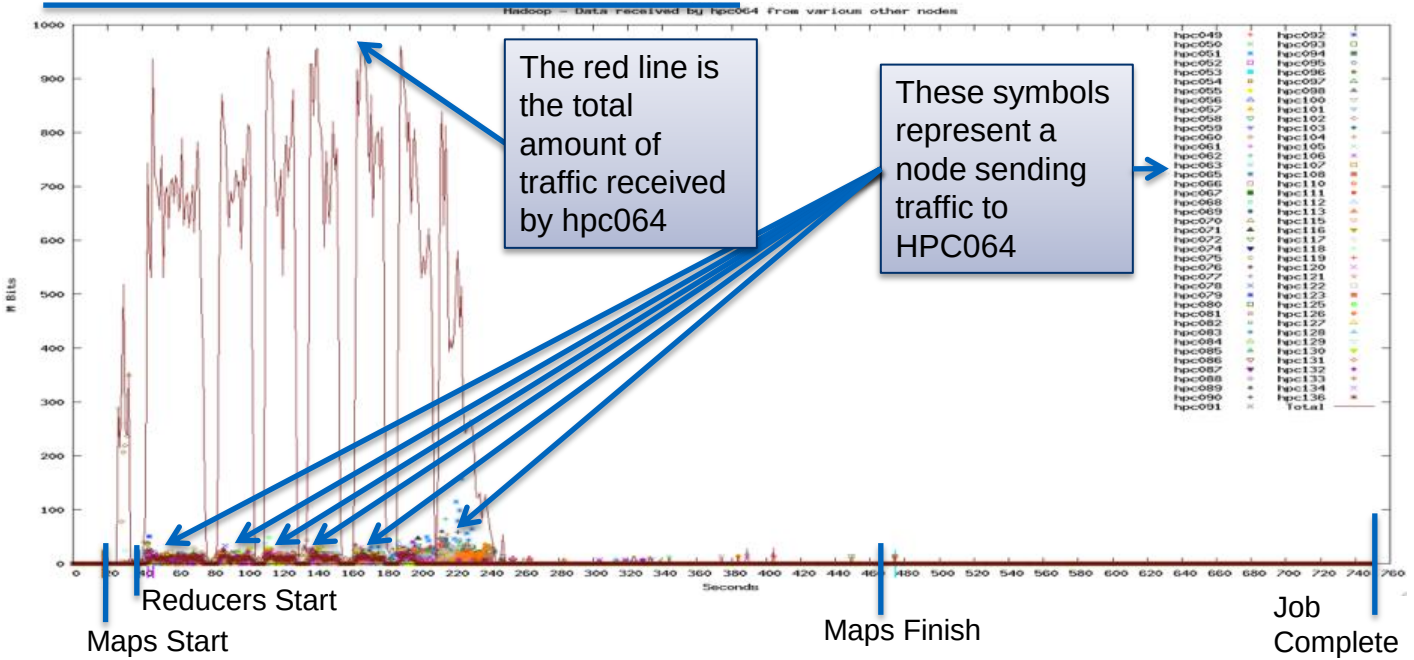


Network graph of all traffic received on a single node (80 node run)

Transform Workload (1TB Terasort)

Network graph of all traffic received on a single node (80 node run)

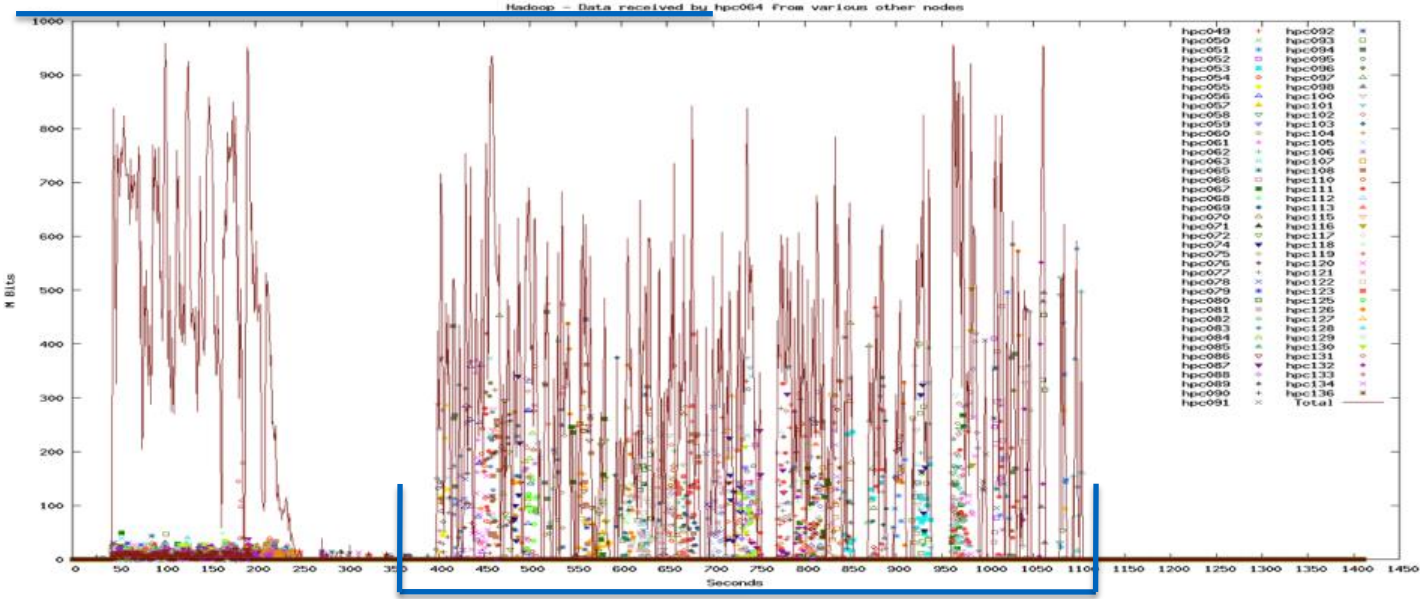
Note:
Shortly after the Reducers start Map tasks are finishing and data is being shuffled to reducers
As Maps completely finish the network is no longer used as Reducers have all the data they need to finish the job



Transform Workload (1TB Terasort with output replication)

Network graph of all traffic received on a single node (80 node run)

Note:
If output replication is enabled, then at the end of the job HDFS must store additional copies. For a 1TB sort, 2TB will need to be replicated across the network.

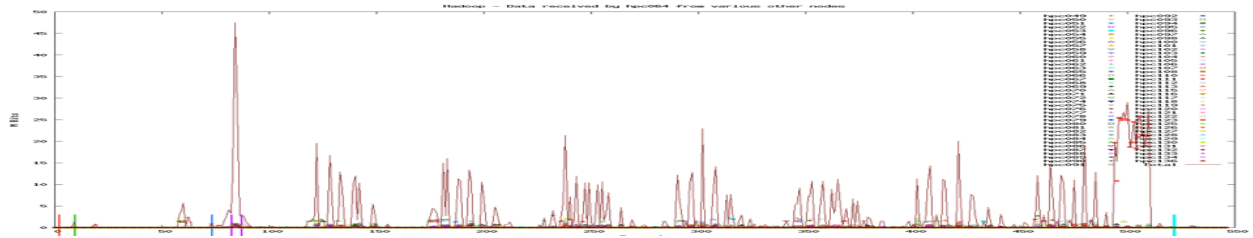


Output Data Replication Enabled

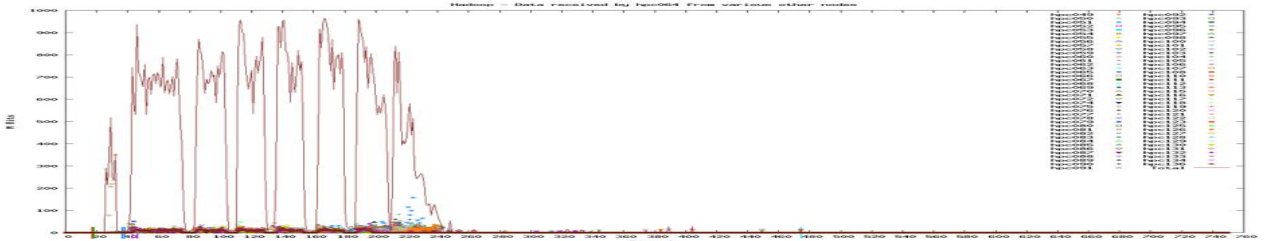
- Replication of 3 enabled (1 copy stored locally, 2 stored remotely)
- Each reduce output is replicated now, instead of just stored locally

Job Patterns - Summary

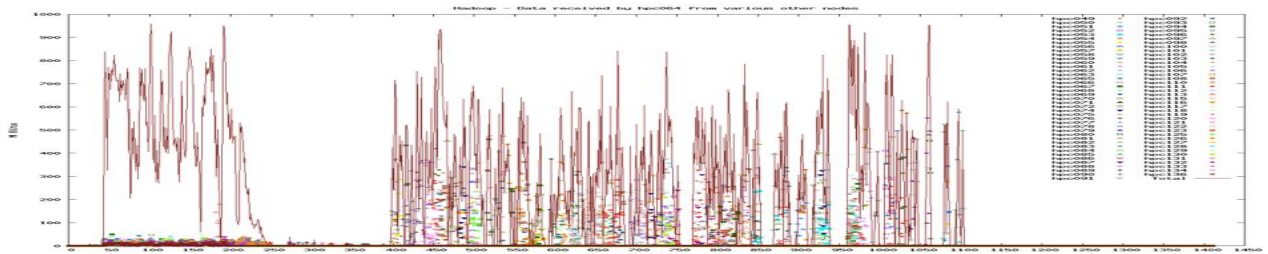
Job Patterns have varying impact on network utilisation



Analyse
Simulated with Shakespeare Wordcount



Extract Transform Load (ETL)
Simulated with Yahoo TeraSort



Extract Transform Load (ETL)
Simulated with Yahoo TeraSort with output replication

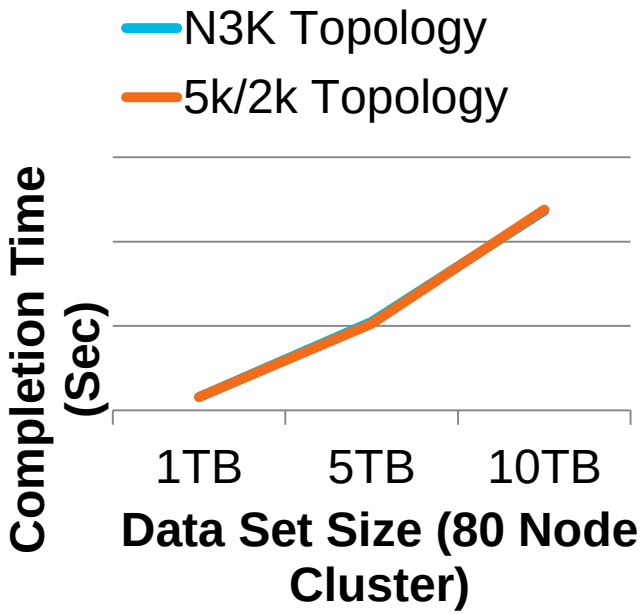



Frequently Asked Questions

How Important is Network Latency for Hadoop?

Consistent, low network latency is desirable, but ultra low latency does not represent a significant factor for typical Hadoop workloads.

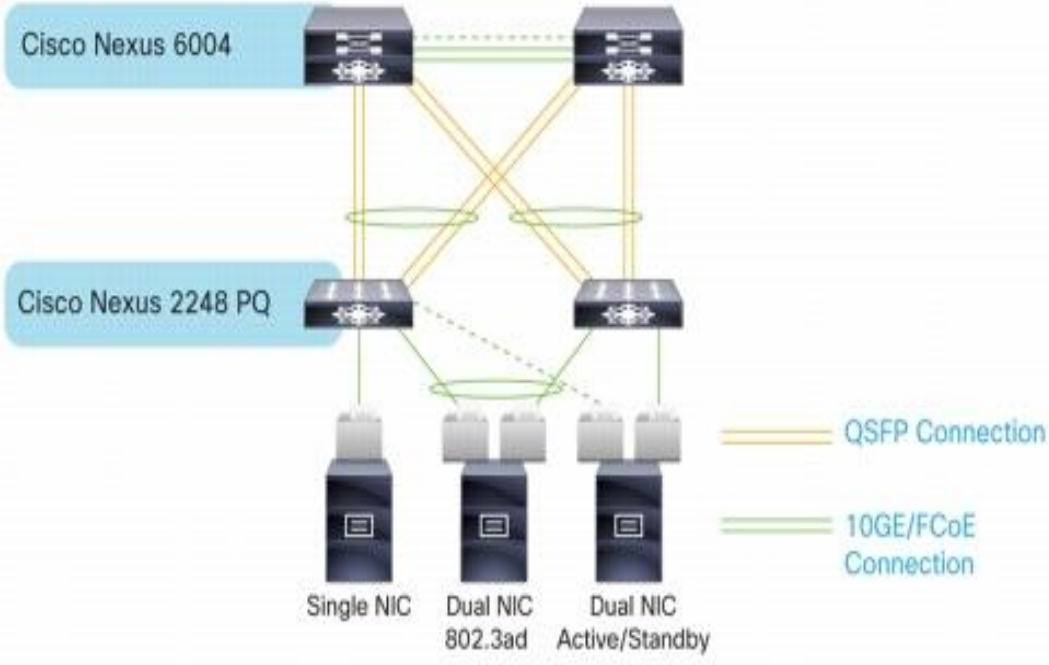
Note:
There is a difference in network latency vs. application latency. Optimisation in the application stack can decrease application latency that can potentially have a significant benefit.



“Should I dual-home my servers?”

It Depends

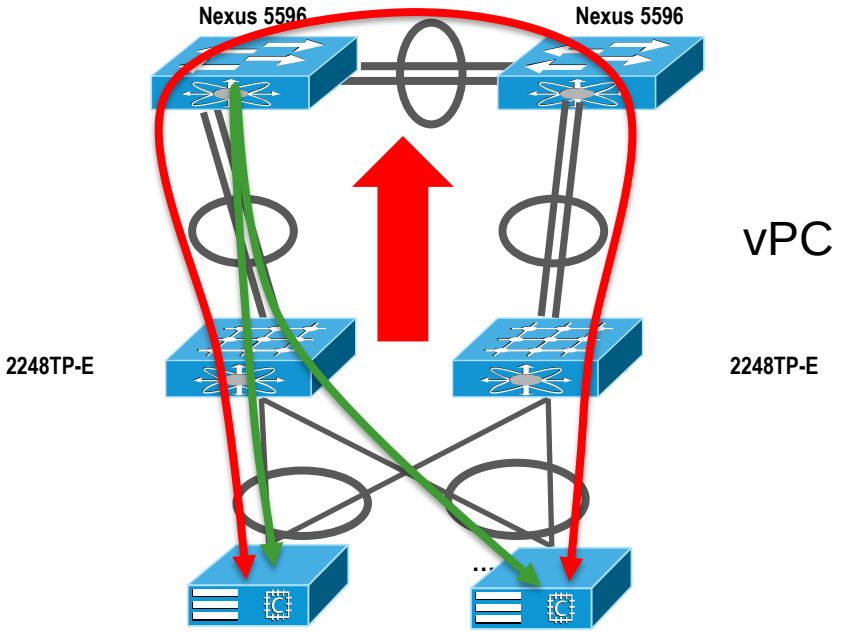
- Protects against switch/FEX failure
- Largely dependent on cluster size
  - Small clusters: yes
  - Large clusters: maybe
- FEX failure = 20+ servers (500TB or more) offline
  - How much capacity lost?
  - How long to re-replicate lost data?



“How should I dual-home my servers?”

EvPC or not? Active-standby or active-active?

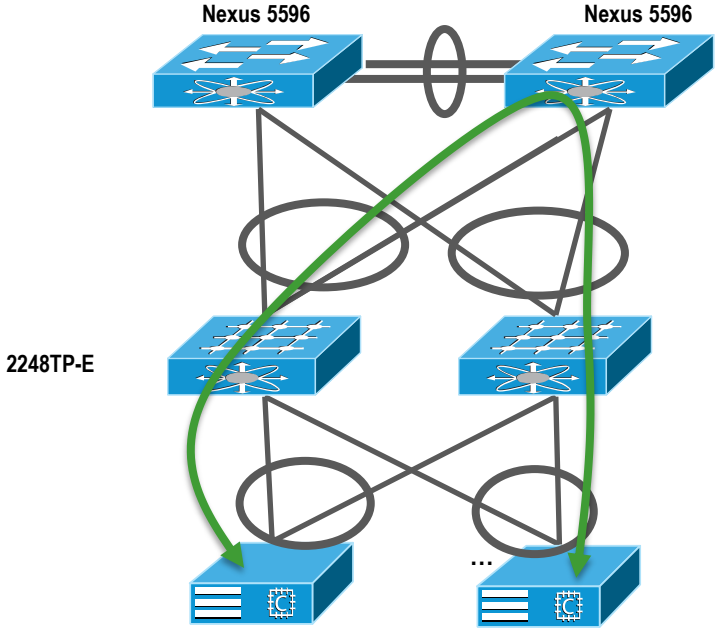
- Without EvPC, care must be taken to avoid peer-link traffic
  - Need to either use Linux bond mode 4 (LACP) or have all active links on same FEX fabric
  - Failure scenarios can be erratic



“How should I dual-home my servers?”

EvPC or not? Active-standby or active-active?

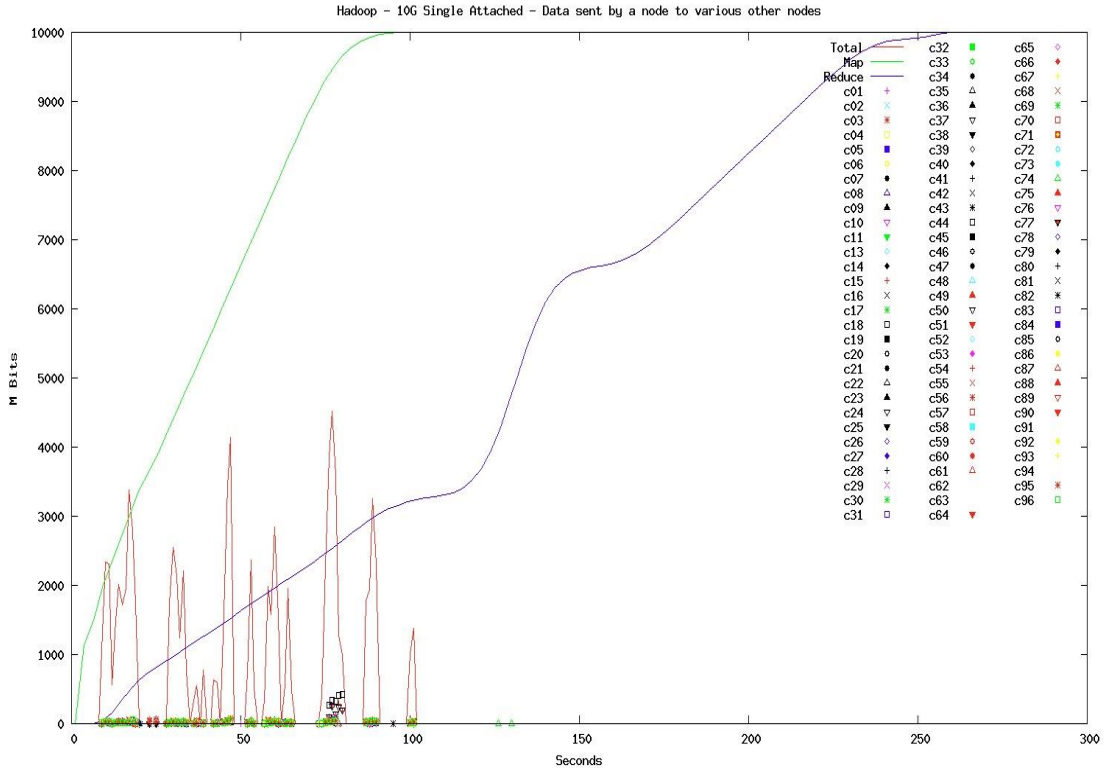
- EvPC pros and cons:
  - Avoids potential 5K/6K peer-link traffic, regardless of bond mode used
  - Cuts scalability in half
  - More complicated setup/management



EvPC

Can Hadoop really push 10GE?

It can, depending on workload, so tune for it!



- Analytic workloads tend to be lighter on the network
- Transform workloads tend to be heavier on the network
- Hadoop has numerous parameters which affect network
- Take advantage of 10GE:
  - mapred.reduce.slowstart.completed.maps
  - dfs.balance.bandwidthPerSec
  - mapred.reduce.parallel.copies
  - mapred.reduce.tasks
  - mapred.tasktracker.reduce.tasks.maximum
  - mapred.compress.map.output

“What’s the right oversubscription number?”

Put another way, how much node-to-node bandwidth should we build in?

- For 1GE attached servers, shoot for non-blocking rack-to-rack
  - Easily achieved with most 1GE ToR solutions, even with dual-homed servers
  - Most modern Hadoop nodes are capable of pushing at least 1GE
- For 10GE, focus on node-to-node bandwidth
  - Aim for at least 4 Gbit to provide clear step up from 2 x 1GE
  - With active-active dual-homed (2 x 10GE) servers, 5:1 oversub
  - With single-homed or active-passive (1 x 10GE) servers, 2.5:1 oversub
  - Consider node config – what can the disks realistically push?

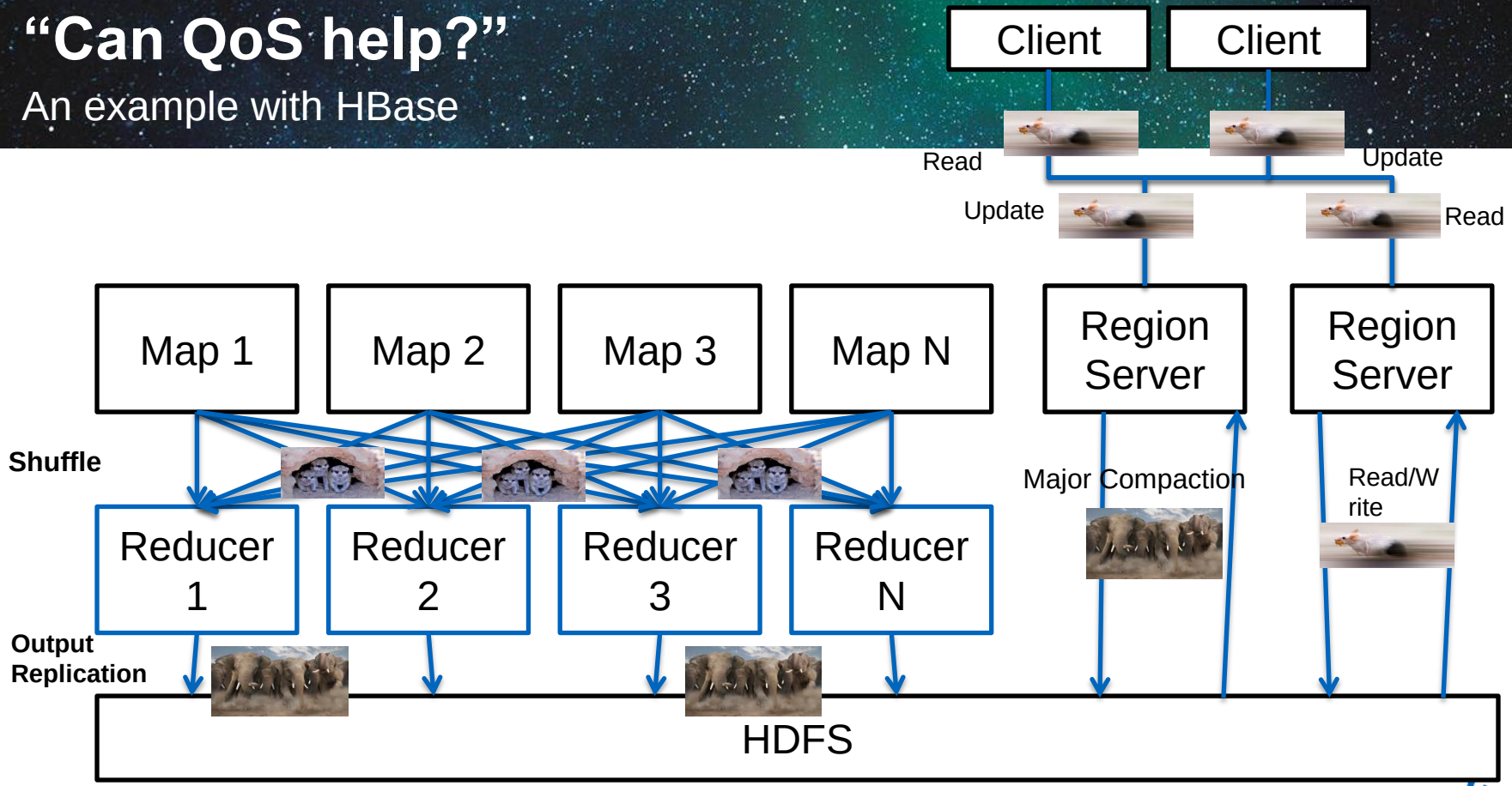
“How do I size my network?”

Don’t panic! It’s not rocket surgery…

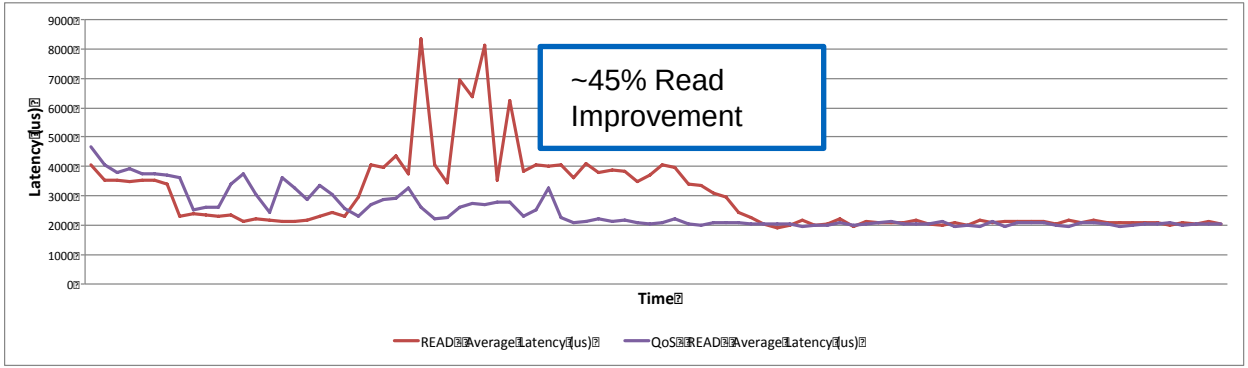
- Basic port math
- Very homogenous design exercise – same for all nodes
- Start with total node count
  - Be sure to understand desired server-side bonding (active-active, active-passive, vPC, etc.)
  - Factor in projected growth
- Keep an eye on oversubscription
- Ask about server config
  - Fat or thin node?
  - Which software distribution?

“Can QoS help?”

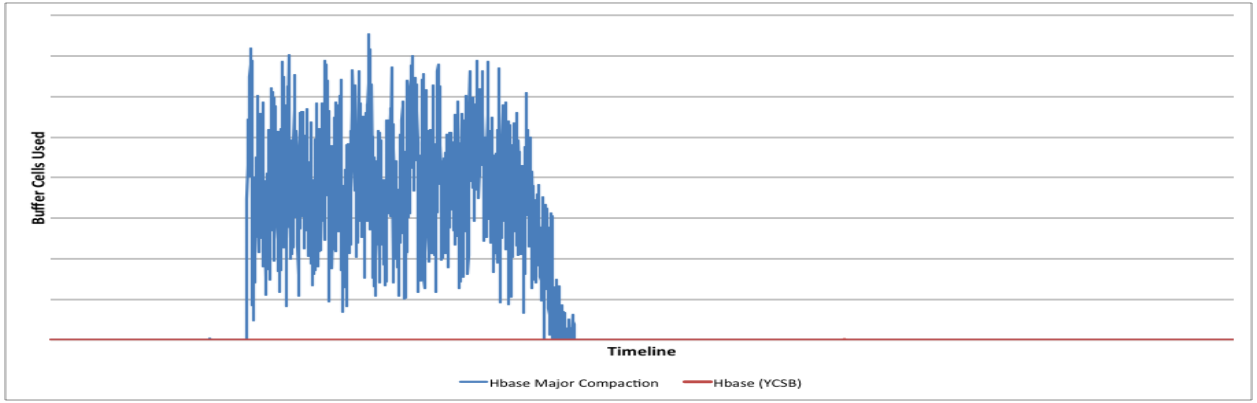
An example with HBase



HBase During Major Compaction with QoS

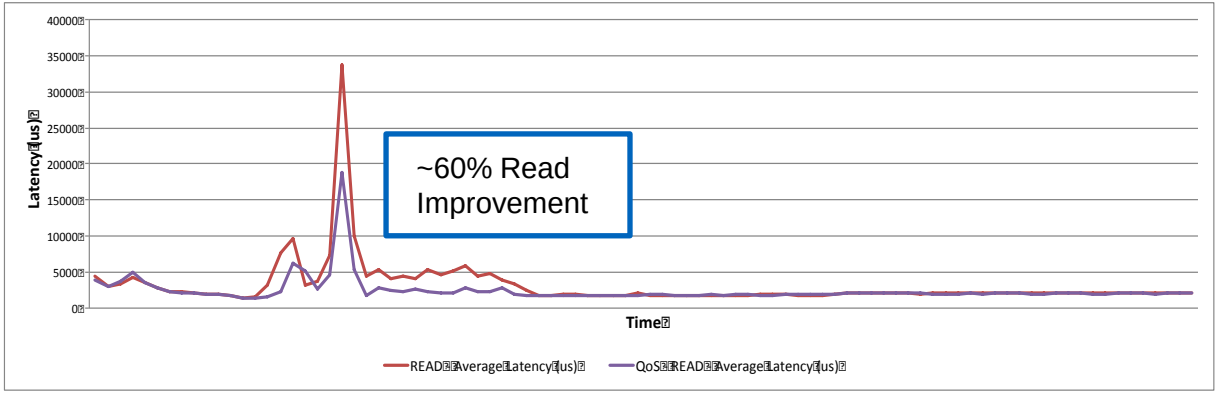


Read Latency
Comparison of Non-QoS vs. QoS Policy

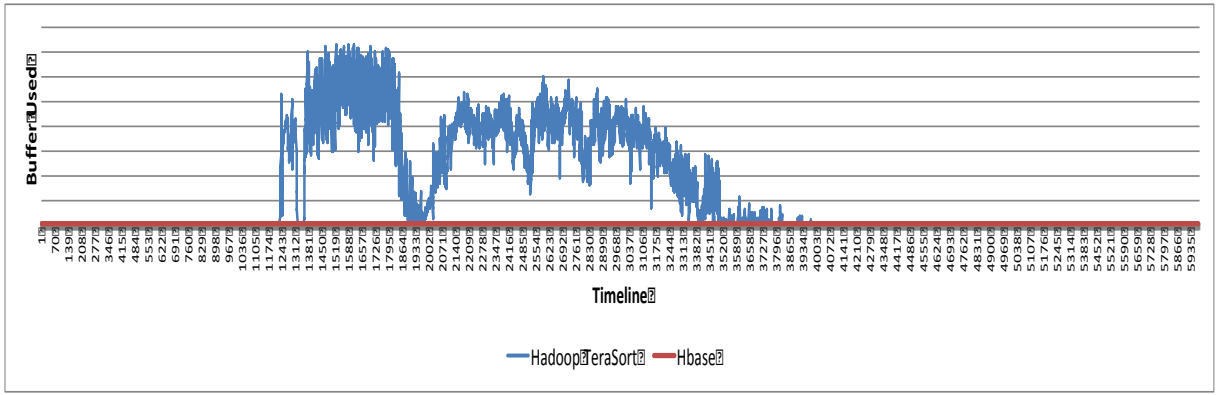


Switch Buffer Usage
With Network QoS Policy to prioritise Hbase Update/Read Operations

HBase + MapReduce with QoS



Read Latency
Comparison of Non-QoS vs. QoS Policy



Switch Buffer Usage
With Network QoS Policy to prioritise HBase Update/Read Operations

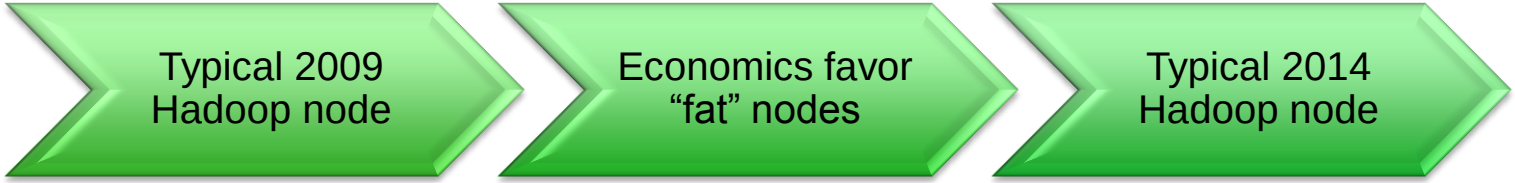
Network Summary

- The network is the “system bus” of the Hadoop “supercomputer”
- Analytic- and ETL-style workloads can behave very differently on the network
- Ultra-low latency probably not critical for typical Hadoop workloads
- Minimise oversubscription, leverage vPC and QoS, and tune Hadoop to take advantage of 10GE – *distribute fairly*

Cisco UCS and Big Data

- Building a big data cluster with the UCS Common Platform Architecture (CPA)
- CPA Networking
- CPA Sizing and Scaling

Hadoop Server Hardware Evolving in the Enterprise



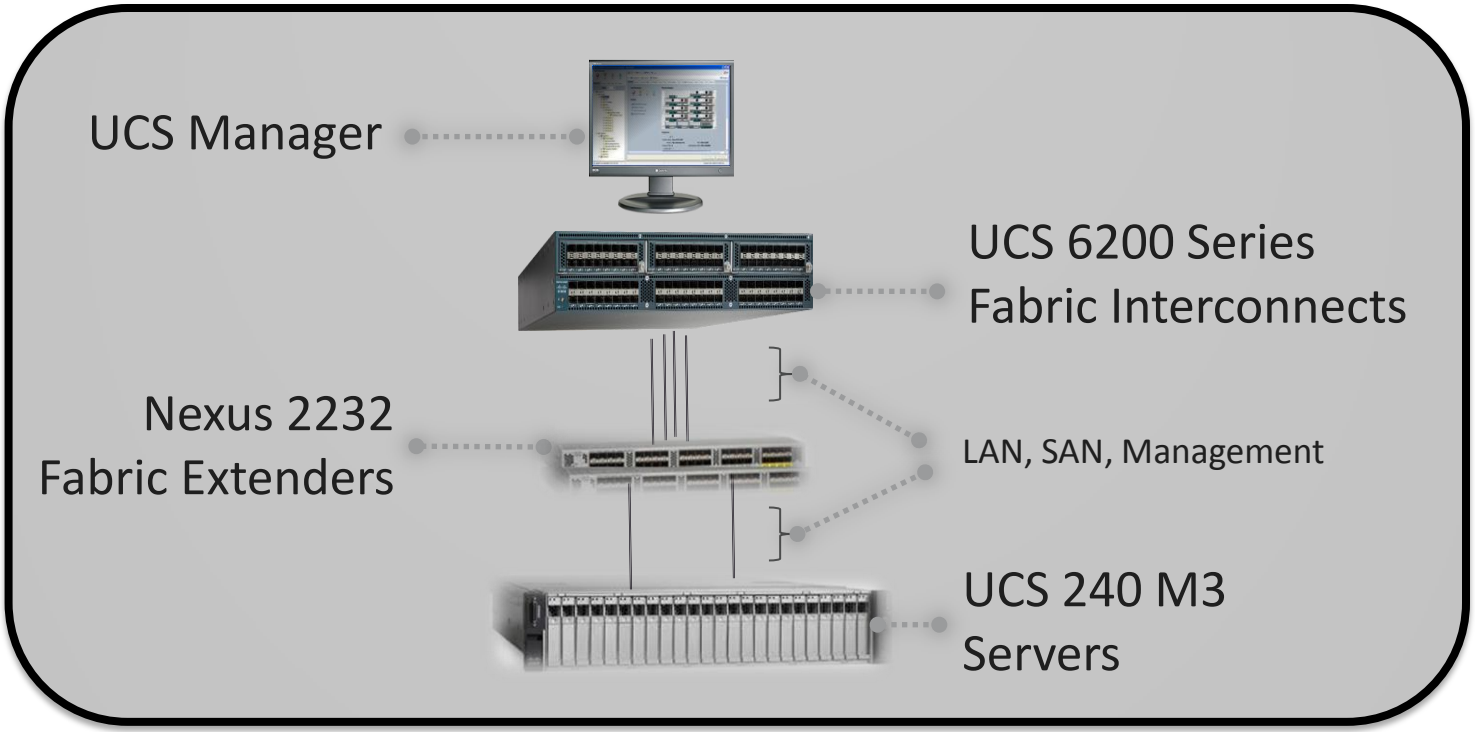
- 1RU server
- 4 x 1TB 3.5" spindles
- 2 x 4-core CPU
- 1 x GE
- 24 GB RAM
- Single PSU
- Running Apache
- $

- 6x-9x more data/node
- 3x-6x more IOPS/node
- Saturated gigabit, 10GE on the rise
- Fewer total nodes lowers licensing/support costs
- Increased significance of node and switch failure

- 2RU server
- 12 x 3TB 3.5" or 24 x 1TB 2.5" spindles
- 2 x 8-core CPU
- 1-2 x 10GE
- 128-256 GB RAM
- Dual PSU
- Running commercial/licensed distribution
- $$$

Cisco UCS Common Platform Architecture (CPA)

- Building Blocks for Big Data



Design Considerations for NoSQL Databases

NoSQL

Design Considerations

- Scale-out with Shared-Nothing
- Data Redundancy Options
  - Key-Value: JBOD + 3-Way Rep
  - D RAID or Replication

Configura tions

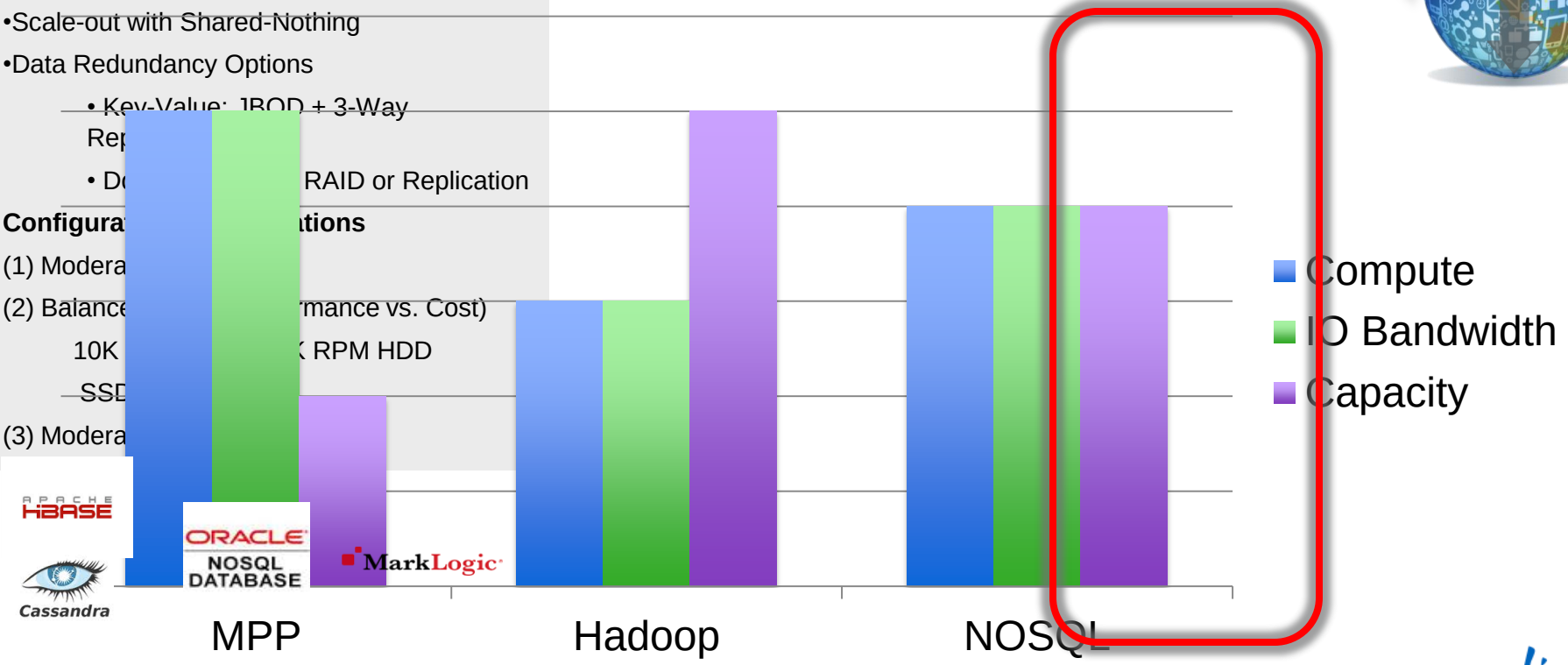
(1) Modera

(2) Balance rmance vs. Cost)

10K K RPM HDD

SSD

(3) Modera



Design Considerations for MPP Database

MPP

Design Considerations

- Scale-out with Shared nothing
- Data Redundancy with Local RAID 5

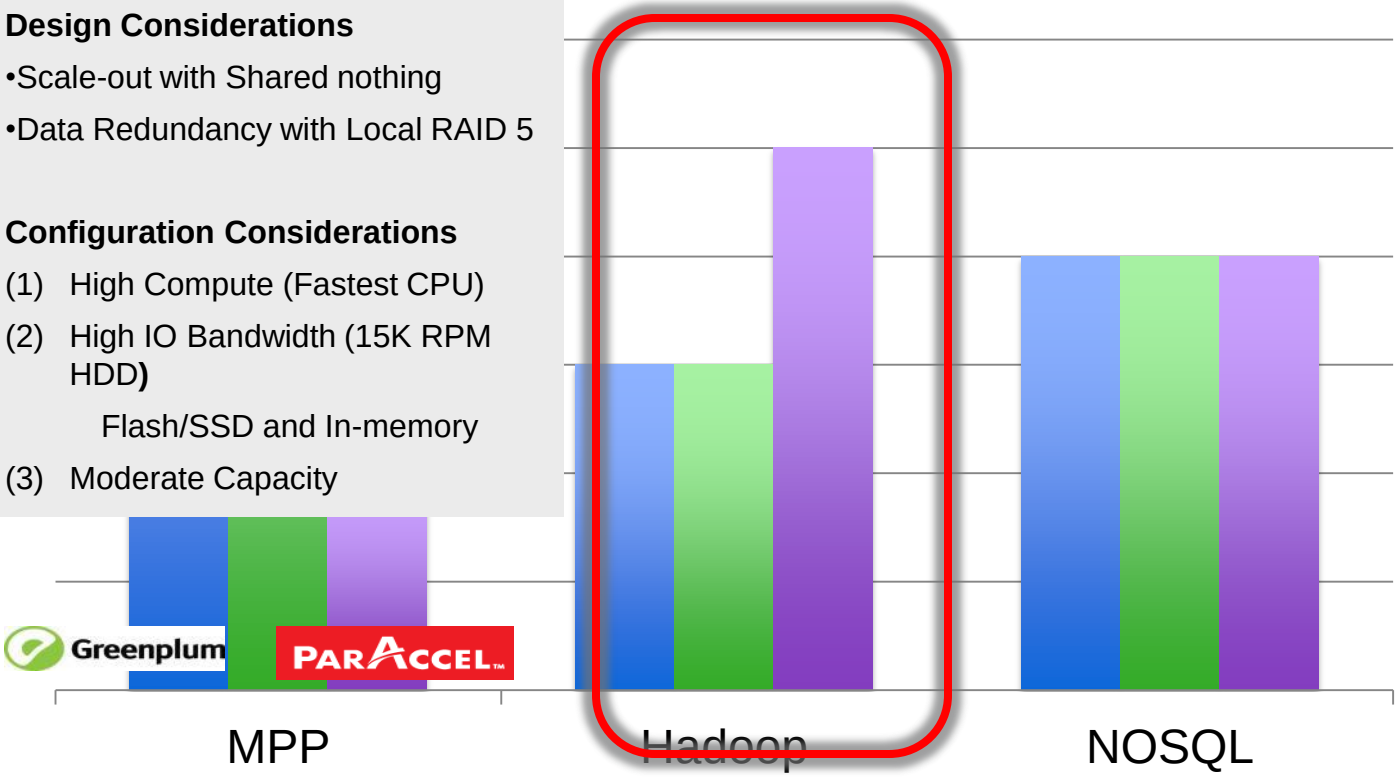
Configuration Considerations

(1) High Compute (Fastest CPU)

(2) High IO Bandwidth (15K RPM HDD**)**

Flash/SSD and In-memory

(3) Moderate Capacity



UCS Reference Configurations for Big Data

Half-Rack UCS Solution for MPP, NoSQL – Performance-Optimised

2 x UCS 6248
2 x Nexus 2232 PP
8 x C240 M3 (SFF)

2 x E5-2680v2
256GB
24x 900GB 10K SAS

Full Rack UCS Solution for Hadoop, NoSQL – Performance + Capacity

2 x UCS 6296
2 x Nexus 2232 PP
16 x C240 M3 (SFF)

2 x E5-2660v2
256GB
24 x 1TB 7.2K SAS

Full Rack UCS Solution for Hadoop Capacity-Optimised

2 x UCS 6296
2 x Nexus 2232 PP
16 x C240 M3 (LFF)

2 x E5-2640v2
128GB
12 x 4TB 7.2K SATA




FREQUENTLY ASKED QUESTIONS

Hadoop and JBOD

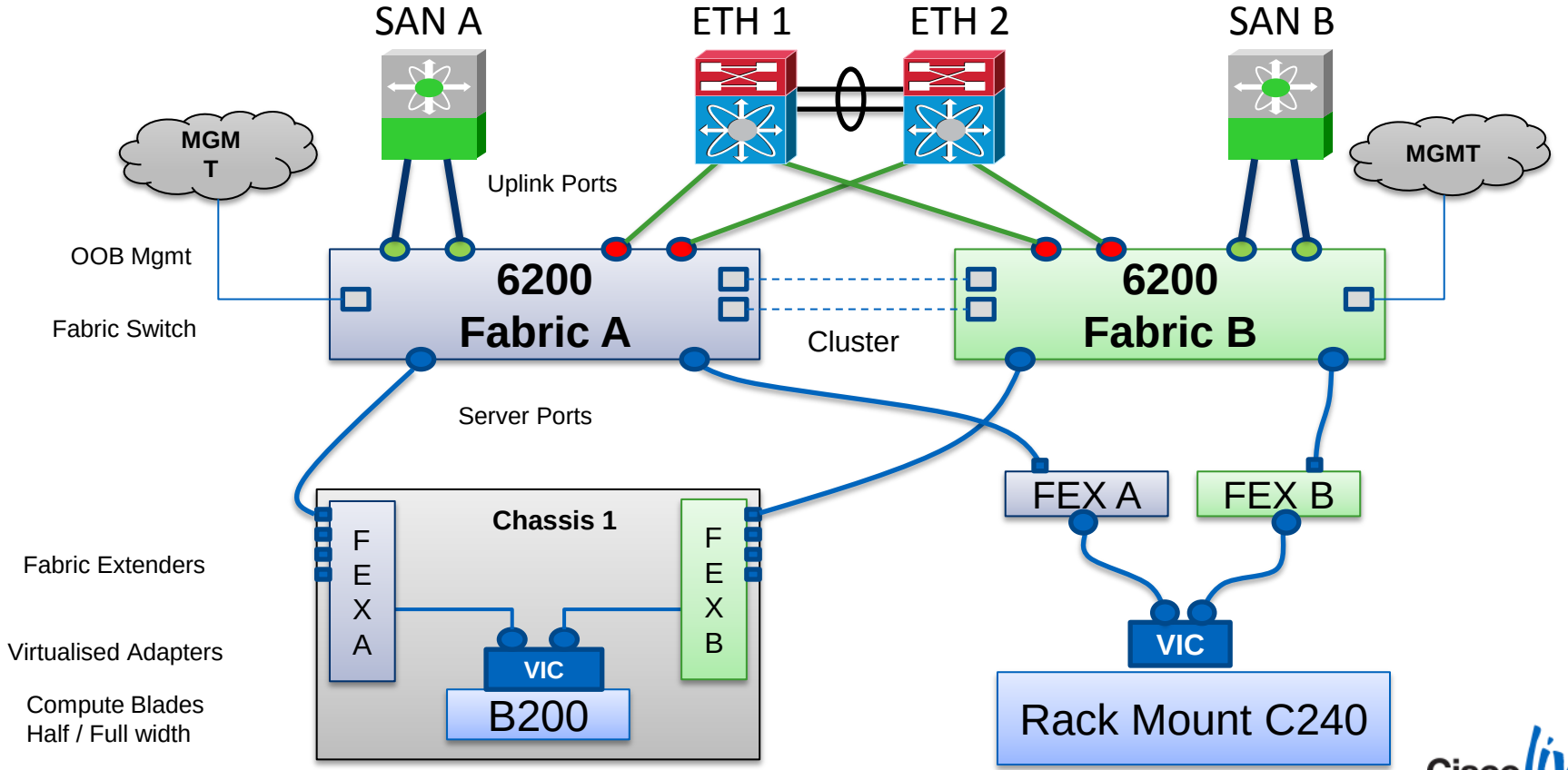
- Why not use RAID-5?
- Most big data architectures are optimised for sequential reads and writes
  - This maximizes data throughput by minimizing seeks/head movement
  - Multiple Map tasks attempting to read from the same RAID-5 device concurrently will look to the disks like random reads
- Rotational speed can vary as much as 10% amongst drives
  - RAID-5 means speed limited to the slowest device in the group
- Hadoop already replicates data, no need for redundancy
  - Multiple block copies serve two purposes: 1) redundancy and 2) performance (more copies available increases data locality % for map tasks)

Can I virtualise?

- Yes you can (easy with UCS), but should you?
- Hadoop and most big data architectures can run virtualised
- However this is typically not recommended for performance reasons
  - Virtualised data nodes will contend for storage and network I/O
  - Hypervisor adds overhead, typically without benefit
- Some customers are running master/admin nodes (e.g. Name Node, Job Tracker, Zookeeper, gateways, etc.) in VM's, but consider single point of failure
- UCS is ideal for virtualisation if you go this route

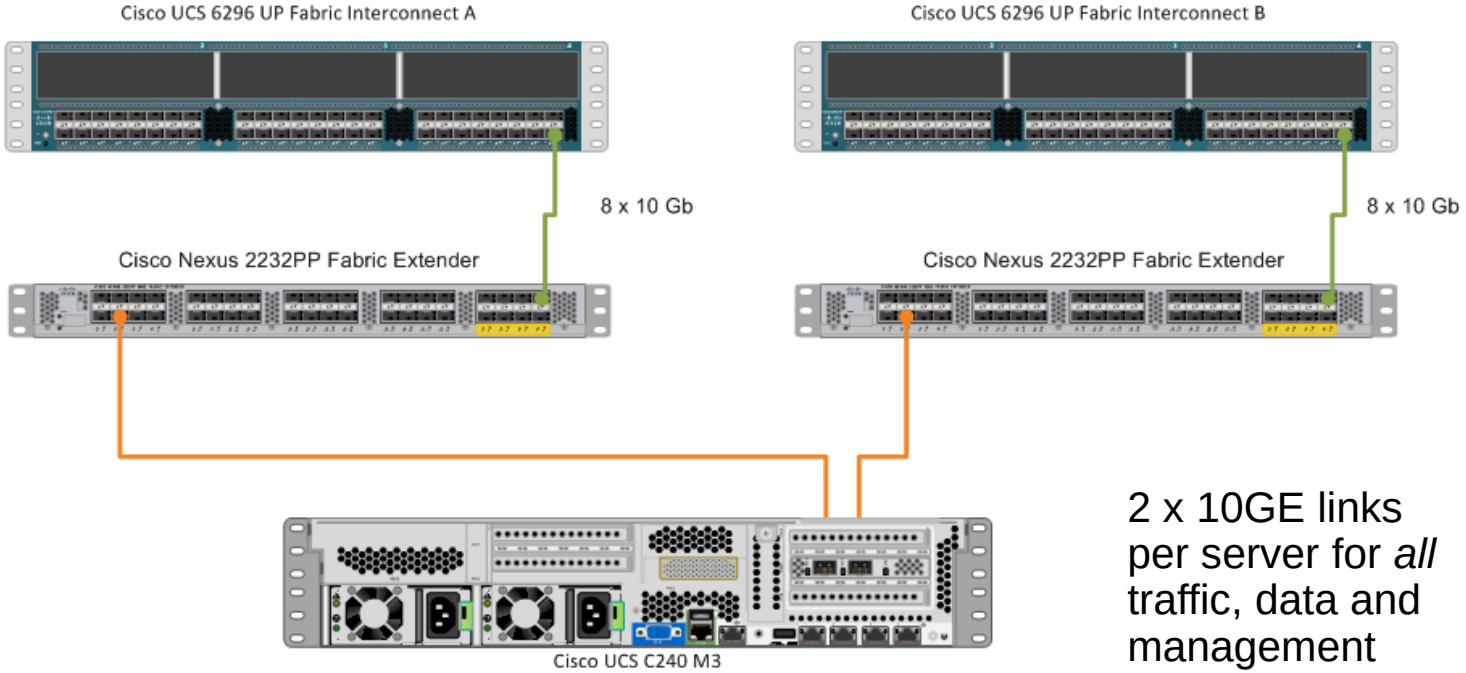
CPA Network Design for Big Data

Cisco UCS: Physical Architecture



CPA: Topology

- Single wire for data and management

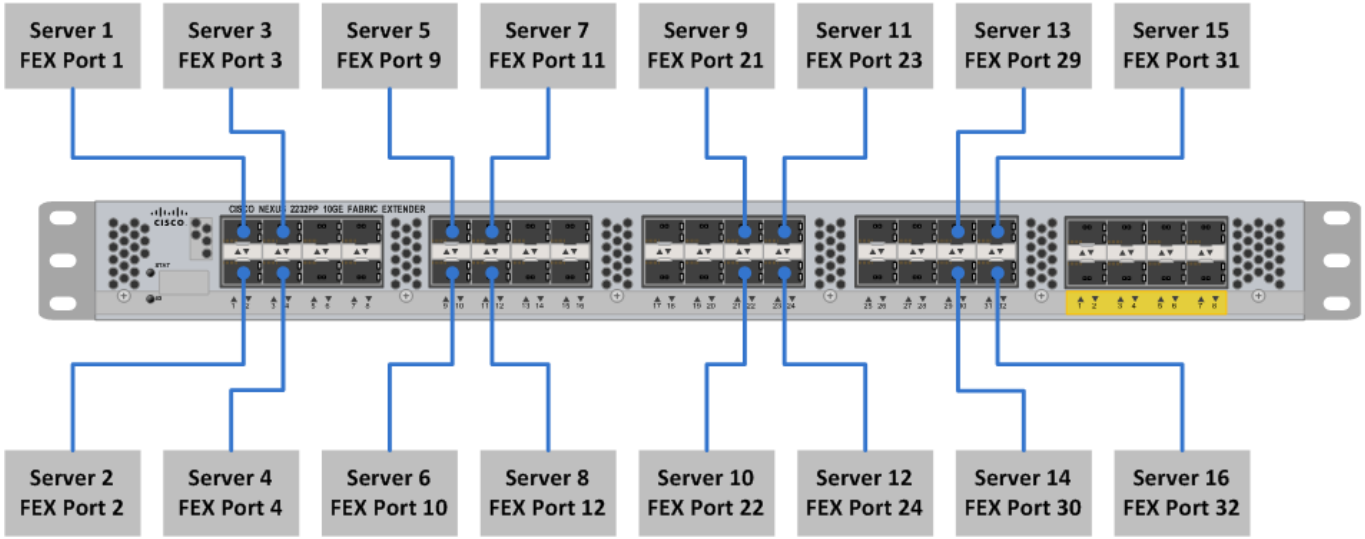


8 x 10GE uplinks per FEX= 2:1 oversub (16 servers/rack), no portchannel (static pinning)

2 x 10GE links per server for *all* traffic, data and management

CPA Recommended FEX Connectivity

- 2 FEX's and 2 FI's

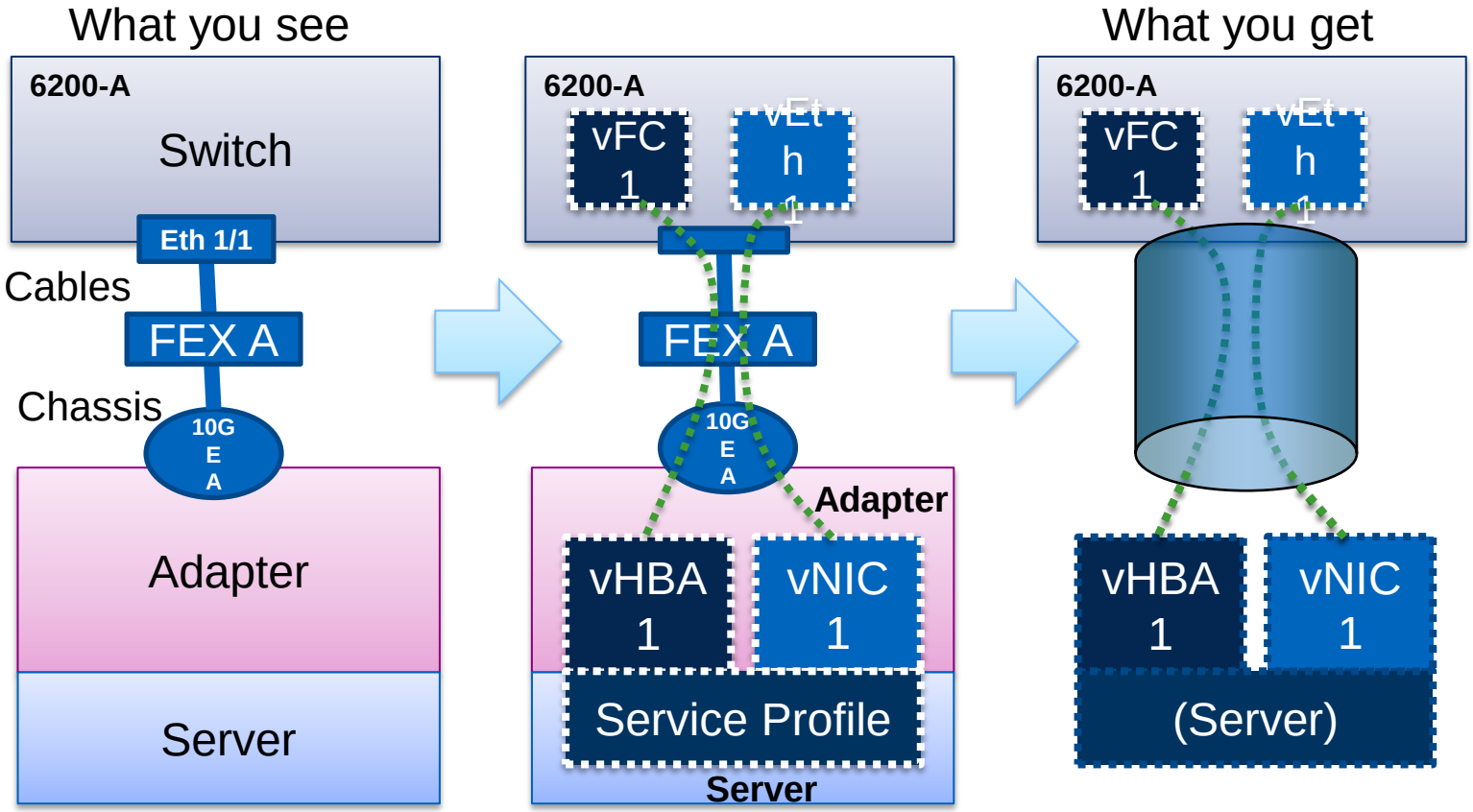


- 2232 FEX has 4 buffer groups: ports 1-8, 9-16, 17-24, 25-32
- Distribute servers across port groups to maximize buffer performance and predictably distribute static pinning on uplinks

Cisco Virtual Interface Card (VIC)

- Converged Network Adapter
  - FCoE in hardware
- Bare metal and VM deployments
  - Virtualise in hardware
  - PCIe compliant
  - vNIC Fabric Failover
- Up to 256 distinct PCIe devices
  - Ethernet vNIC and FC vHBA
- **QoS**
  - 8 queues
  - vNIC bandwidth guarantees

Virtualise the physical network pipe



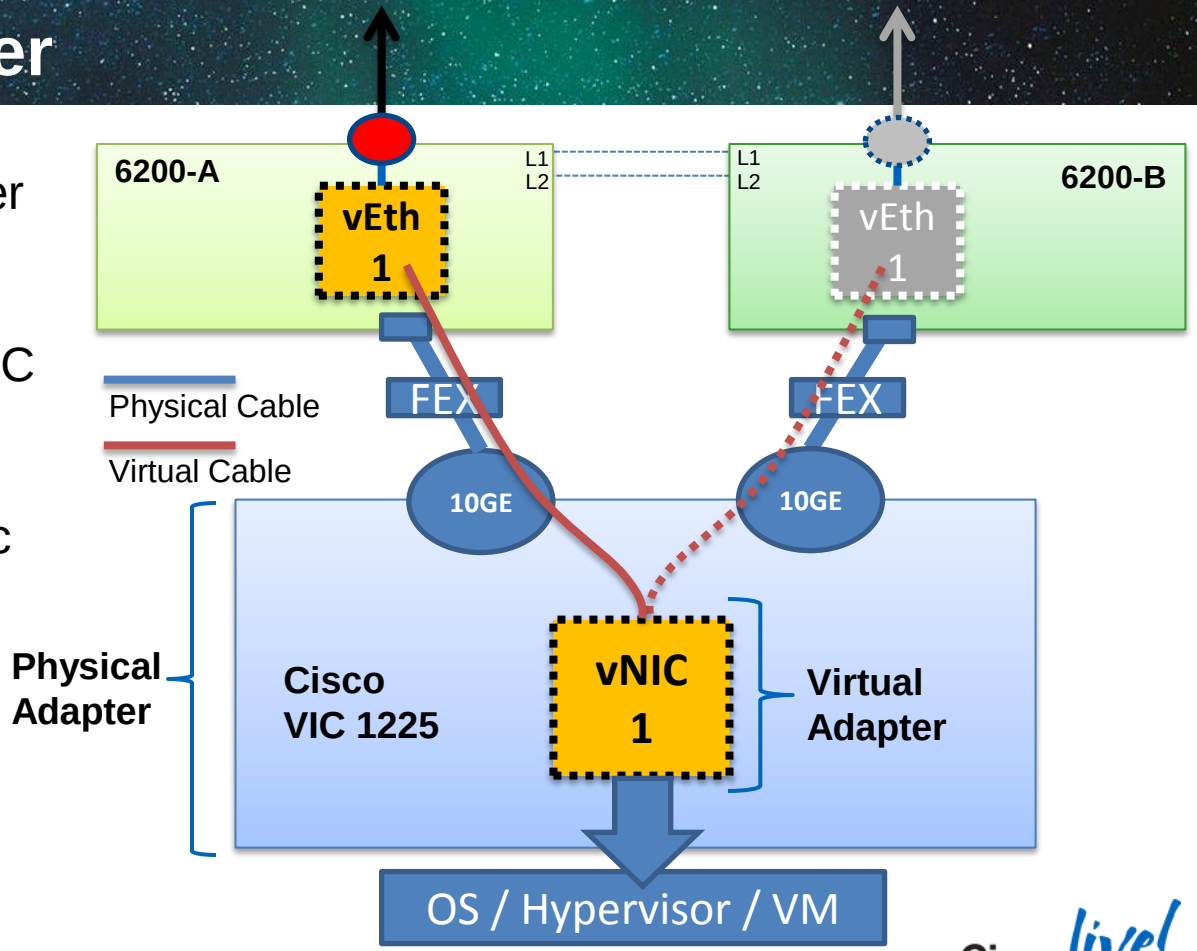
- ✓ Dynamic, Rapid Provisioning
- ✓ State abstraction
- ✓ Location Independence
- ✓ Blade or Rack

Physical Cable

Virtual Cable (VN-Tag)

UCS Fabric Failover

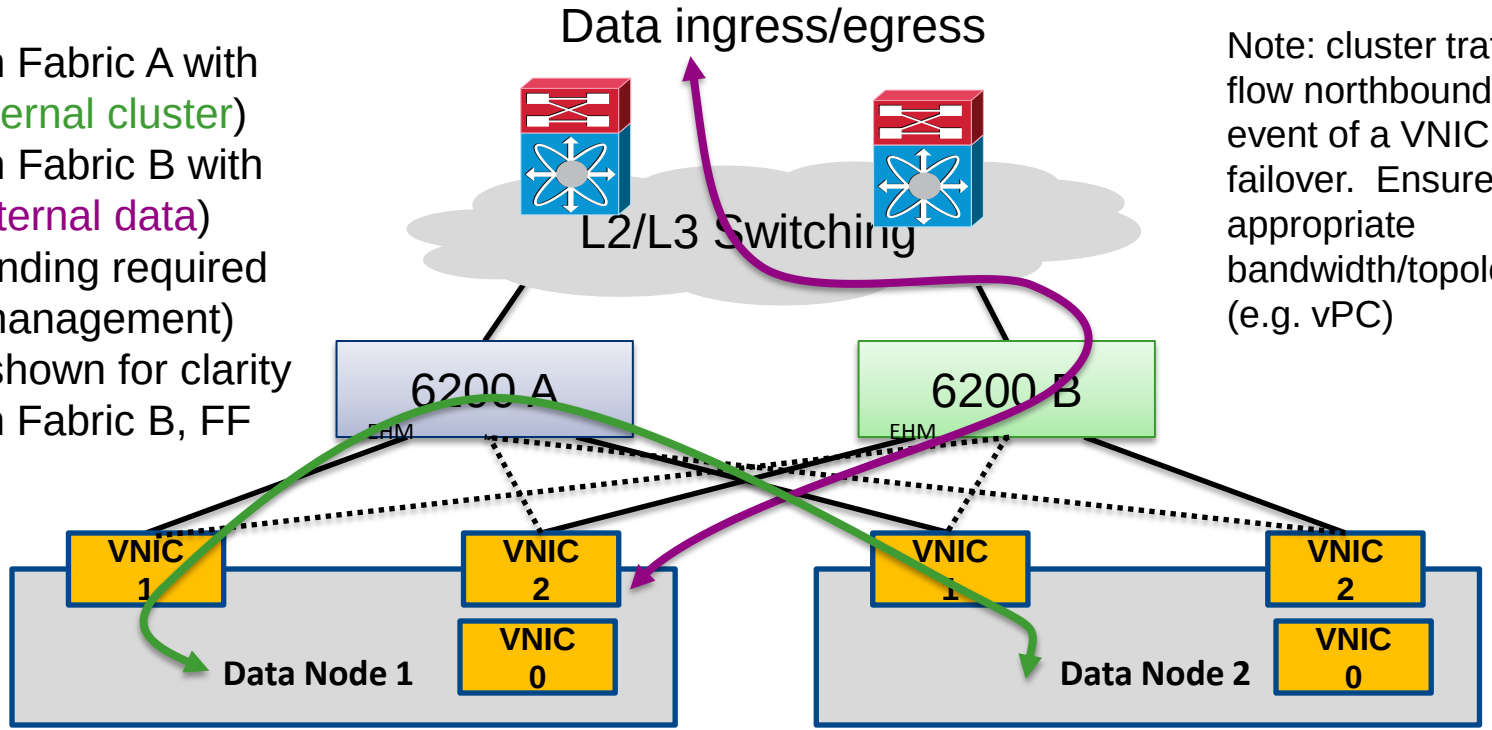
- Fabric provides NIC failover capabilities chosen when defining a service profile
- Traditionally done using NIC bonding driver in the OS
- Provides failover for both unicast and multicast traffic
- Works for any OS on bare metal
- (Also works for any hypervisor-based servers)



Recommended UCS networking with Apache Hadoop

- Use 2 VNICs with Fabric Failover on opposite fabrics for internal and external traffic

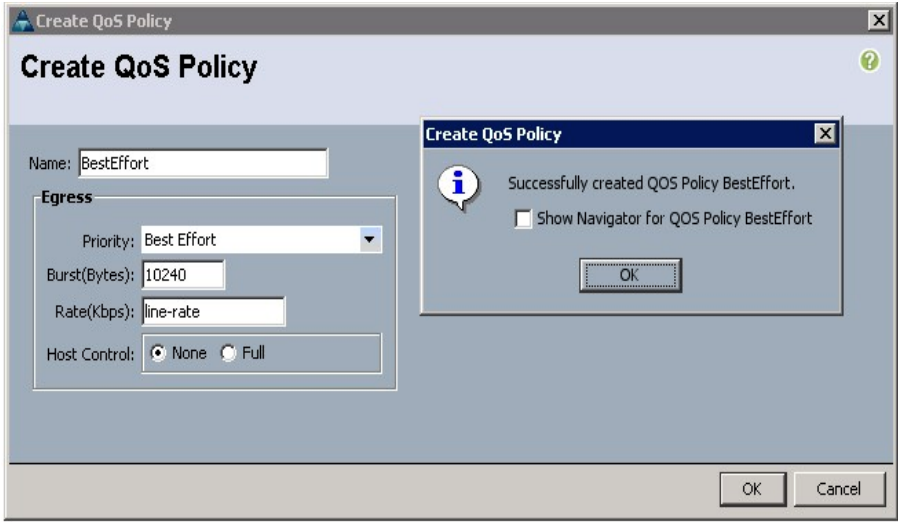
▪VNIC 1 on Fabric A with FF to B (internal cluster)
▪VNIC 2 on Fabric B with FF to A (external data)
▪No OS bonding required
▪VNIC 0 (management) wiring not shown for clarity (primary on Fabric B, FF to A)



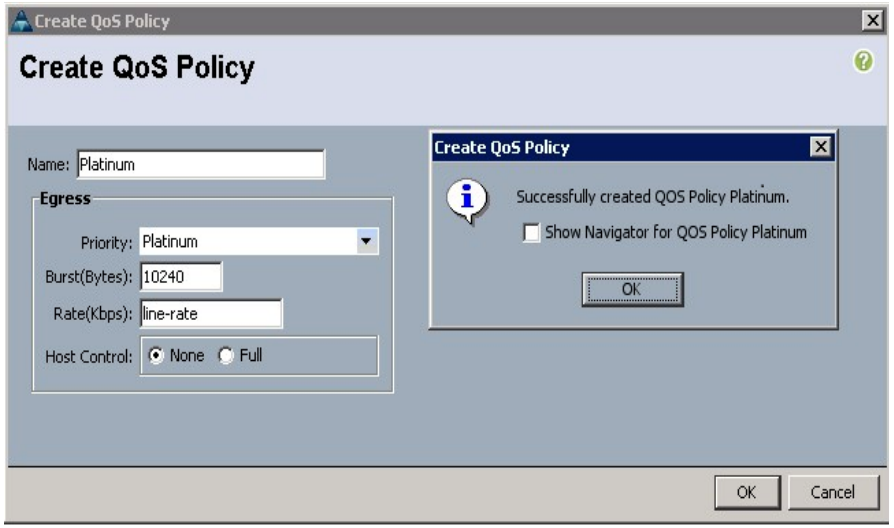
Note: cluster traffic will flow northbound in the event of a VNIC1 failover. Ensure appropriate bandwidth/topology (e.g. vPC)

Create QoS Policies

- Leverage simplicity of UCS Service Profiles

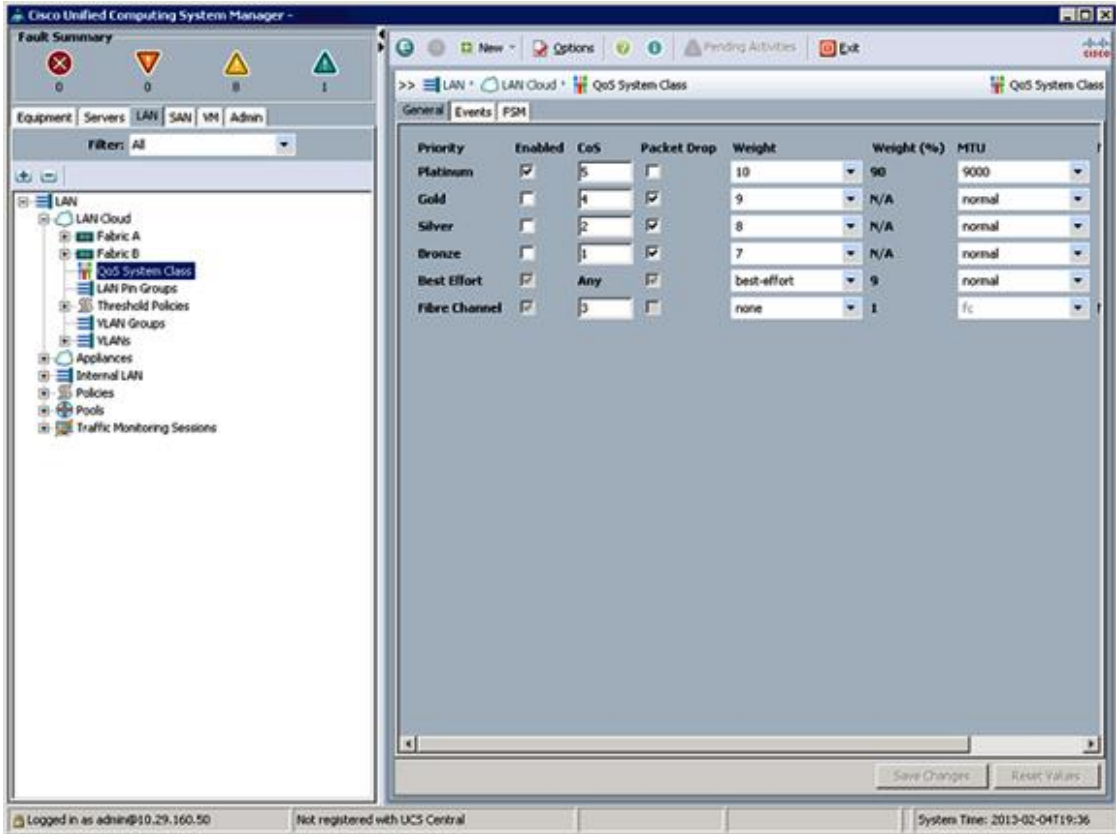


Best Effort policy for management VLAN



Platinum policy for cluster VLAN

Enable JumboFrames for Cluster VLAN

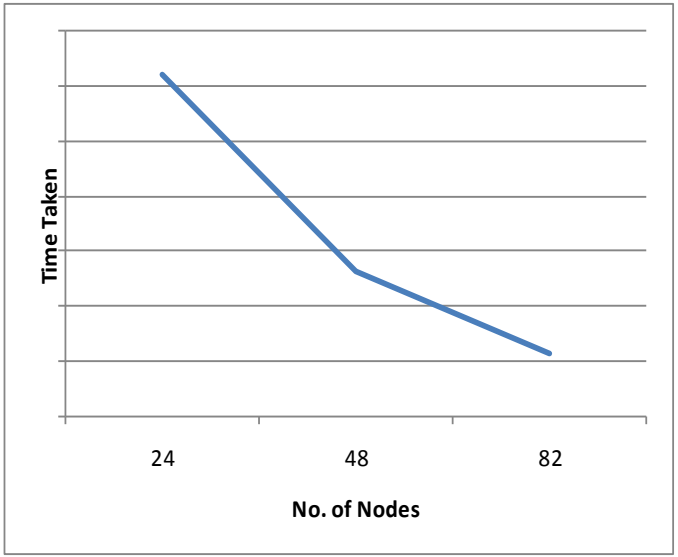


1. Select the LAN tab in the left pane in the UCSM GUI.
2. Select LAN Cloud > QoS System Class.
3. In the right pane, select the General tab
4. In the Platinum row, enter 9000 for MTU.
5. Check the Enabled Check box next to Platinum.
6. Click Save Changes.
7. Click OK.

CPA Sizing and Scaling for Big Data

Cluster Scalability

A general characteristic of an optimally configured cluster is the ability to either 1) decrease job completion time, or 2) do more work in the same amount of time, by scaling out the nodes

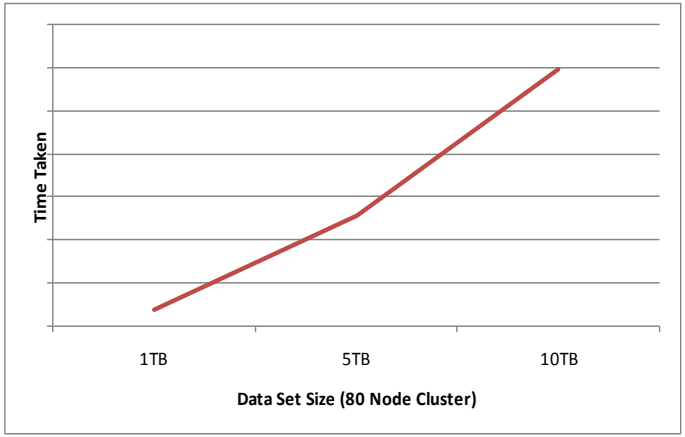


Test results from ETL-like Workload (Yahoo Terasort) using 1TB data set.

Input Data Size

Given the same MapReduce Job, the larger the input dataset, the longer the job will take.

Note:
It is important to note that as dataset sizes increase completion times may not scale linearly as many jobs can hit the ceiling of I/O and/or Compute power.



Test results from ETL-like Workload (Yahoo Terasort) using varying data set sizes.

Sizing

- Part science, part art
- Start with current storage requirement
  - Factor in replication (typically 3x) and compression (varies by data set)
  - Factor in 20-30% free space for temp (Hadoop) or up to 50% for some NoSQL systems
  - Factor in average daily/weekly data ingest rate
  - Factor in expected growth rate (i.e. increase in ingest rate over time)
- If I/O requirement known, use next table for guidance
- Most big data architectures are very linear, so more nodes = more capacity and better performance
- Strike a balance between price/performance of individual nodes vs. total # of nodes

CPA Sizing and Application Guidelines

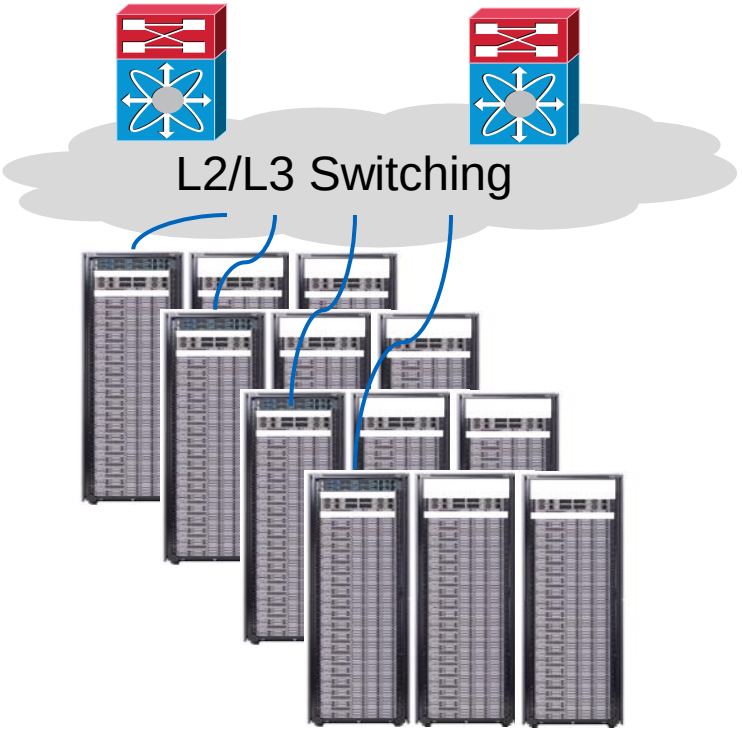
Server	CPU	2 x E5-2680v2	2 x E5-2660v2	2 x E5-2640v2
	Memory (GB)	256	256	128
	Disk Drives	24 x 900GB 10K	24 x 1TB 7.2K	12 x 4TB 7.2K
	IO Bandwidth (GB/Sec)	2.6	2.0	1.1
Rack-Level	Cores	320	320	256
	Memory (TB)	4	4	2
	Capacity (TB)	336	384	768
	IO Bandwidth (GB/Sec)	41.3	31.9	16.9
Applications		MPP DB NoSQL	Hadoop NoSQL	Hadoop

Best Performance ⟷ Best Price/TB

Scaling the CPA

Single Rack
16 servers

Single Domain
Up to 10 racks, 160 servers



Multiple Domains

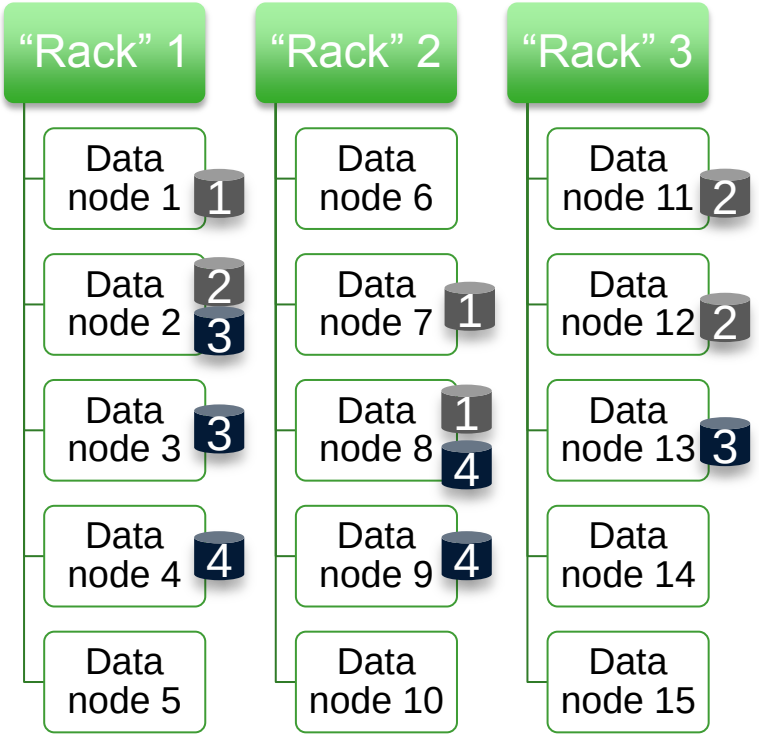
Scaling the Common Platform Architecture

- Multiple domains based on 16 servers per rack and 2 x 2232 FEXs

Consider intra- and inter-domain bandwidth:

Servers Per Domain (Pair of Fabric Interconnects)	Available North-Bound 10GE ports (per fabric)	Southbound oversubscription (per fabric)	Northbound oversubscription (per fabric)	Intra-domain server-to-server bandwidth (per fabric, Gbits/sec)	Inter-domain server-to-server bandwidth (per fabric, Gbits/sec)
160	16	2:1	5:1	5	1
144	24	2:1	3:1	5	1.67
128	32	2:1	2:1	5	2.5

Rack Awareness



- Rack Awareness provides Hadoop the optional ability to group nodes together in logical "racks"
- Logical "racks" may or may not correspond to physical data centre racks
- Distributes blocks across different "racks" to avoid failure domain of a single "rack"
- It can also lessen block movement between "racks"
- Can be useful to control block placement and movement in UCSM integrated environments

Recommendations: UCS Domains and Racks

Single Domain Recommendation
Turn off or enable at physical rack level

- For simplicity and ease of use, leave Rack Awareness off
- Consider turning it on to limit physical rack level fault domain (e.g. localised failures due to physical data centre issues – water, power, cooling, etc.)

Multi Domain Recommendation
Create one Hadoop rack per UCS Domain

- With multiple domains, enable Rack Awareness such that each UCS Domain is its own Hadoop rack
- Provides HDFS data protection across domains
- Helps minimise cross-domain traffic

Summary

Leverage UCS and Nexus to integrate big data into your data centre operations

- Think of big data clusters as a single “supercomputer”
- Think of the network as the “system bus” of the supercomputer
- Strive for consistency in your deployments
- The goal is an even distribution of load – *distribute fairly*
- Cisco Nexus and UCS Common Platform Architecture for Big Data can help!

Call to Action…

- Visit the **Cisco Stand** at the World of Solutions to experience demos/solutions in action
- Get hands-on experience with the following **Walk-in Labs**
- **Meet the Expert**
- **CL Online** -Visit us online after the event for updated PDFs and on-demand session videos. www.CiscoLiveAPAC.com




Q & A

Complete Your Online Session Evaluation

Give us your feedback and receive a Cisco Live 2014 Polo Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site www.ciscoliveaustralia.com/mobile
- Visit any Cisco Live Internet Station located throughout the venue

Polo Shirts can be collected in the World of Solutions on Friday 21 March 12:00pm - 2:00pm

Learn online with Cisco Live!

Visit us online after the conference for full access to session videos and presentations.
www.CiscoLiveAPAC.com


CISCO
TM