Unlocking Big Data infrastructure efficiency with Hadoop over disaggregated storage

Manoj Wadekar – Director Hardware Architecture, eBay Inc.
Anjaneya "Reddy" Chagam, Chief SDS Architect, Intel Corporation

Acknowledgements: QCT team (Gary Lee, Becky Lin, Marco Huang, Kido Yen)

Notices and Disclaimers

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Configurations: Ceph* storage nodes, each server: 16 Intel® Xeon® processor E5-2680 v3, 128 GB RAM, twenty-four 6 TB Seagate Enterprise* hard drives, and two 2 TB Intel® Solid-State Drive (SSD) DC P3700 NVMe* drives with 10 GbE Intel® Ethernet Converged Network Adapter X540-T2 network cards, 20 GbE public network, and 40 GbE private Ceph network.

Apache Hadoop* data nodes, each server: 16 Intel Xeon processor E5-2620 v3 single socket, 128 GB RAM, with 10 GbE Intel Ethernet Converged Network Adapter X540-T2 network cards, bonded.

The difference between the version with Intel® Cache Acceleration Software (Intel® CAS) and the baseline is that the Intel CAS version is not caching and is in pass-through mode, so software only, no hardware changes are needed. The tests used were TeraGen*, TeraSort*, TeraValidate*, and DFSIO, which are the industry-standard Hadoop performance tests. For more complete information, visit intel.com/performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

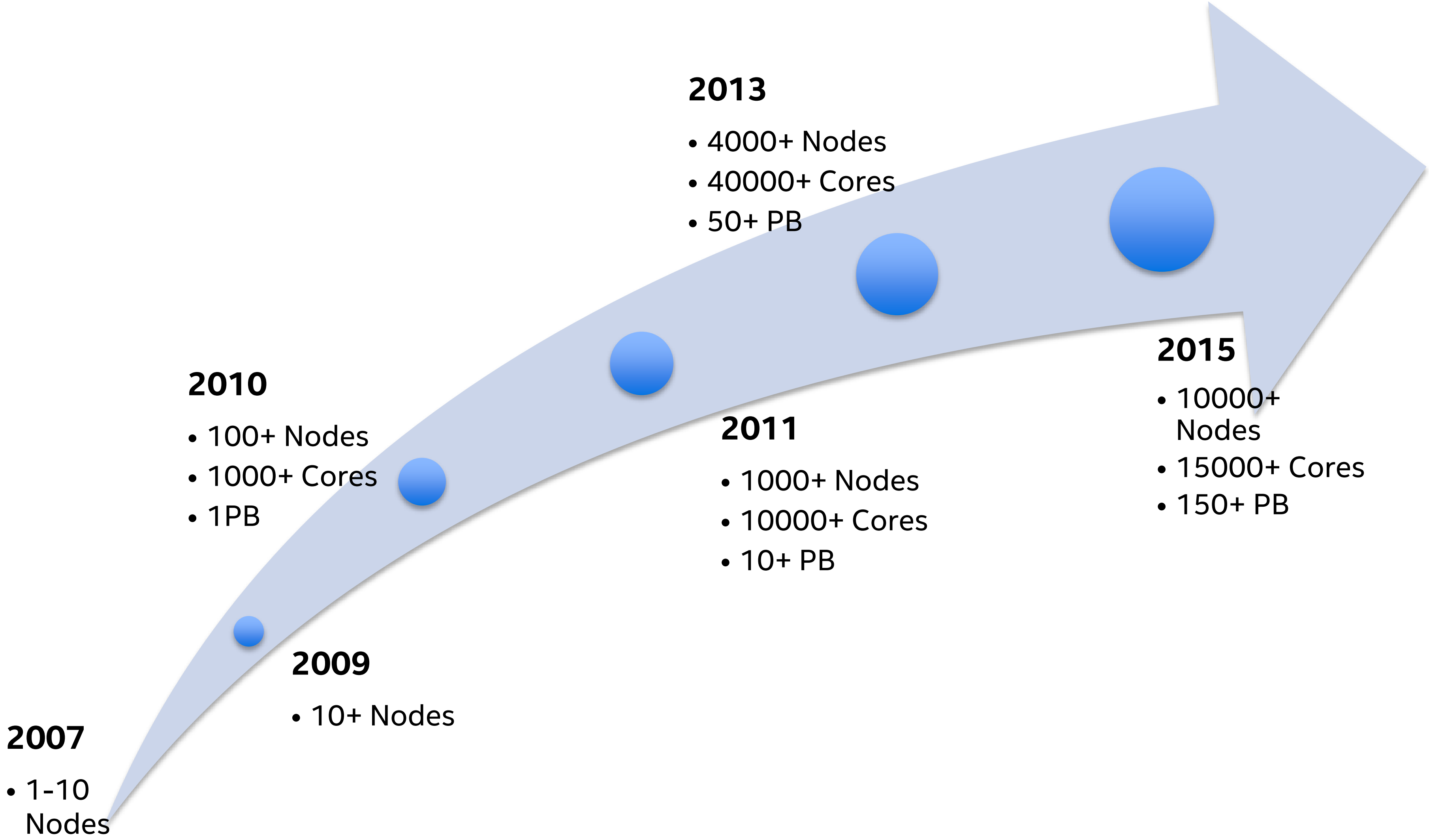
Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Agenda

- Data Growth Challenges
- Need for Storage Disaggregation
- Hadoop and Ceph Quick Tutorial
- Hadoop Over Ceph (Block)
- Summary

Hadoop @ eBay


2007
• 1-10 Nodes
2009
• 10+ Nodes
2010
• 100+ Nodes
• 1000+ Cores
• 1PB
2011
• 1000+ Nodes
• 10000+ Cores
• 10+ PB
2013
• 4000+ Nodes
• 40000+ Cores
• 50+ PB
2015
• 10000+ Nodes
• 15000+ Cores
• 150+ PB

Challenges with Scaling Apache Hadoop* Storage

Both storage and compute resources are bound to Hadoop nodes

Excess capacity: Surplus storage capacity if cluster is compute-bound. And vice versa.

Inefficiency:

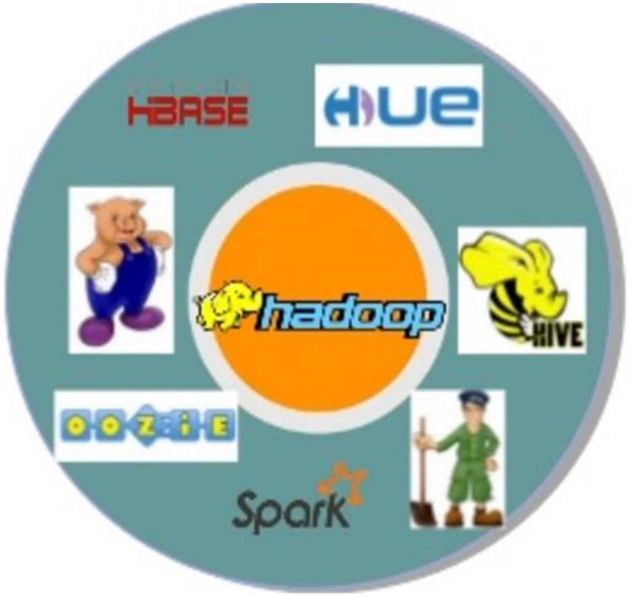
Spend, infrastructure space, power utilization

Scale Challenge:

Inefficiencies cause significant impact for large clusters

Challenges with Scaling Apache Hadoop* Storage

Native Hadoop storage and compute needs can be different for clusters



Different Cluster Needs: Compute/Storage needs can change for different clusters. Can result in under-performance or wastage.

Inefficiency:

Capacity planning and capacity utilization

Proposed Solution: Disaggregated Storage

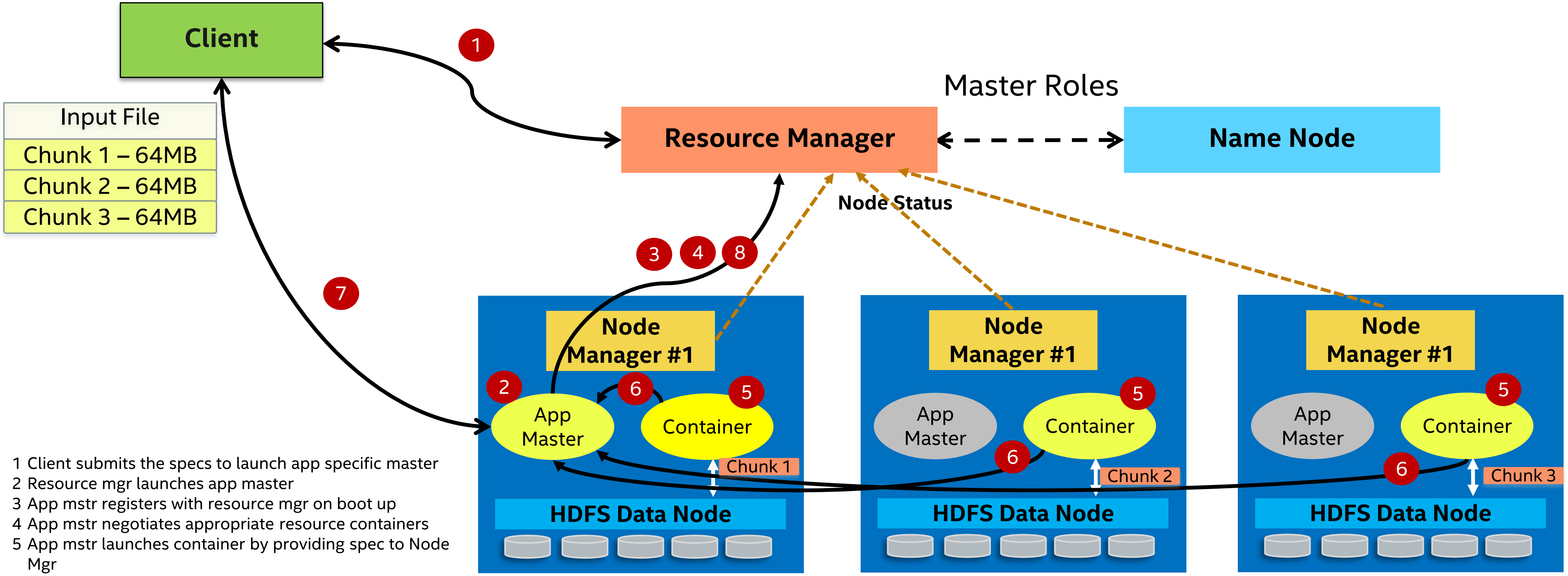
Separate out storage and compute resources

Right Sizing: Clusters can use optimized ratio of compute and storage. Should allow reducing wastage and improve performance

Independent Scaling: Compute and storage capacities can be scaled per need

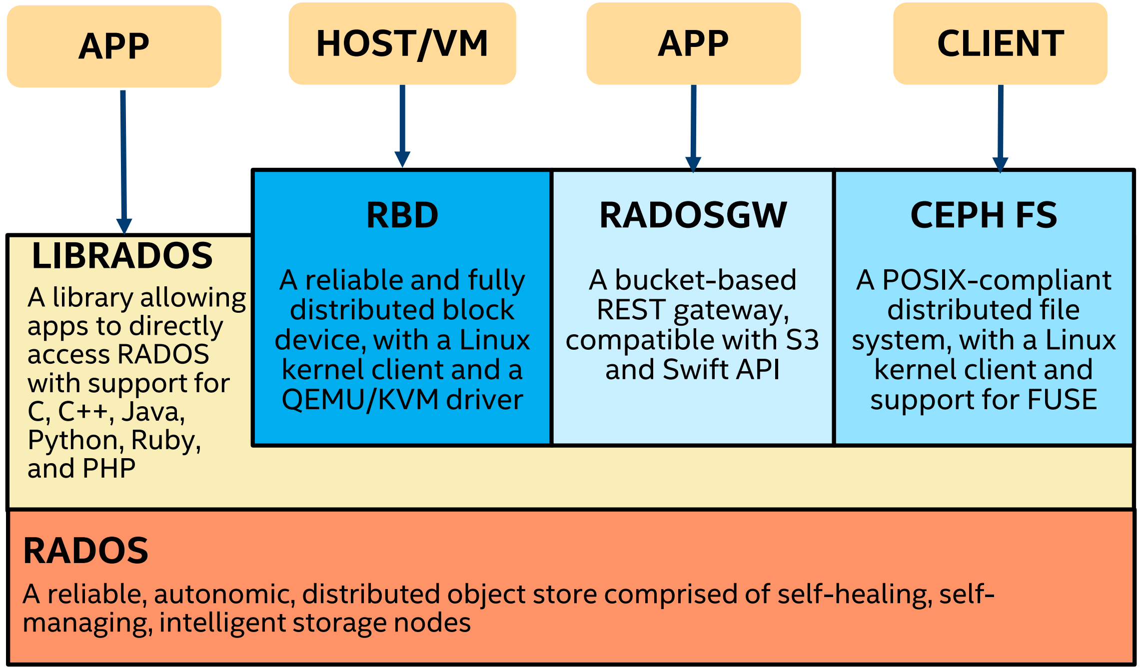
Storage Solutions: HDFS, iSCSI, Ceph etc. ***Focus of this presentation is Ceph***

Hadoop/YARN Architecture

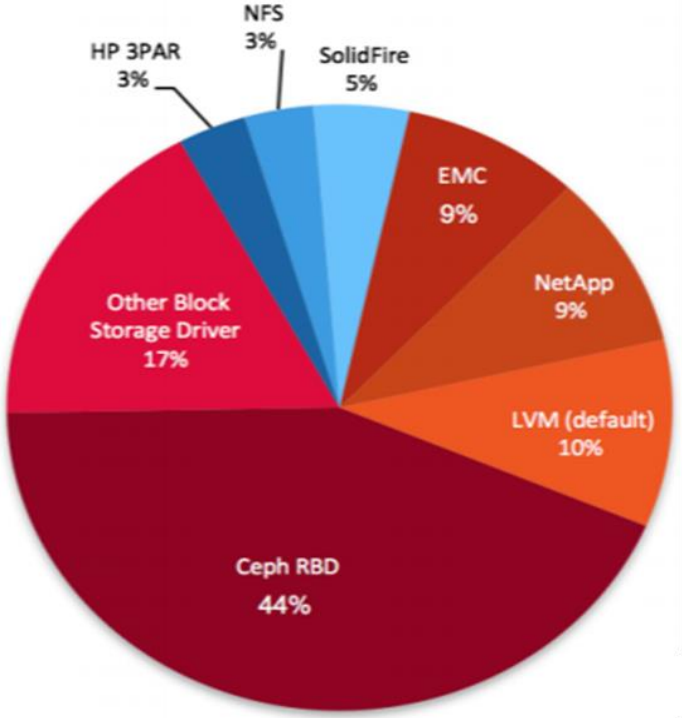


1 Client submits the specs to launch app specific master
2 Resource mgr launches app master
3 App mstr registers with resource mgr on boot up
4 App mstr negotiates appropriate resource containers
5 App mstr launches container by providing spec to Node Mgr
6 Container provides progress, status etc. to its App Mstr
7 Client communicates with App Mstr directly for progress
8 Once work finishes, App Mstr deregisters with res mgr

Ceph Overview



Which OpenStack block storage (Cinder) drivers are in use?
Source: April 2017 OpenStack User Survey

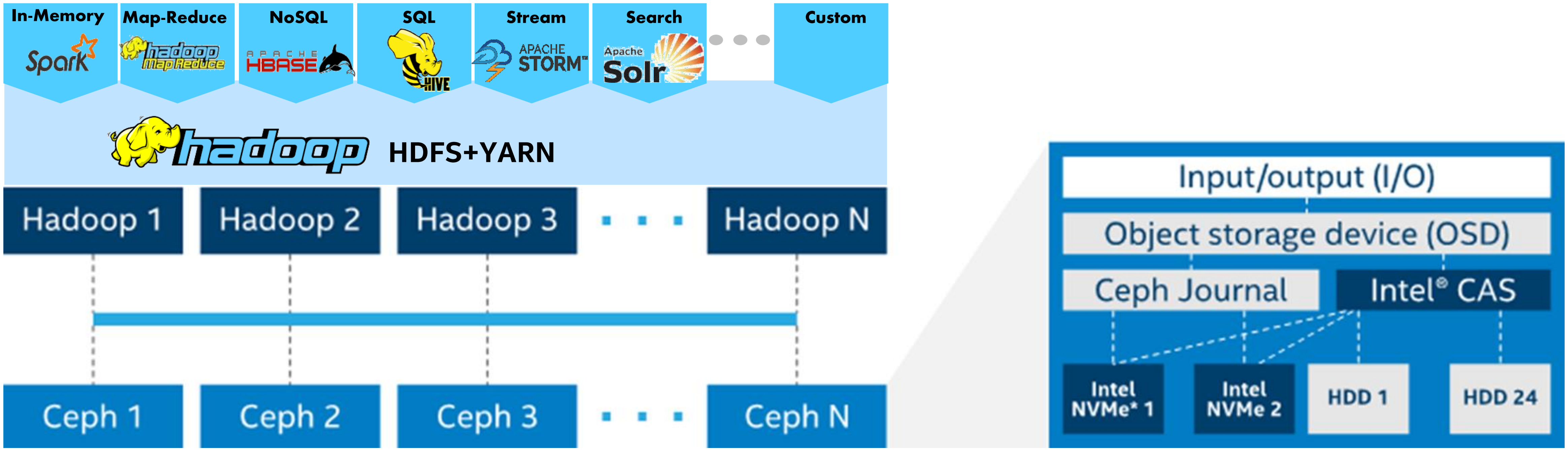


- Open-source, object-based scale-out storage
- Object, Block and File in single unified storage cluster
- Highly durable, available – replication, erasure coding
- Runs on economical commodity hardware
- 10 years of hardening, vibrant community

- Scalability – CRUSH data placement, no single POF
- Replicates and re-balances dynamically
- Enterprise features – snapshots, cloning, mirroring
- Most popular block storage for Openstack use cases
- Commercial support from Red Hat

References: https://www.openstack.org/assets/survey/April2017SurveyReport.pdf

Apache Hadoop* with Ceph* Storage: Logical Architecture



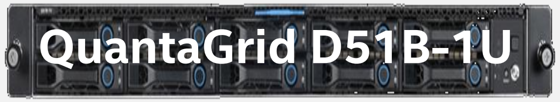
Deployment Options

- Hadoop Services: Virtual, Container or Bare Metal
- Storage Integration: Ceph Block, File or Object
- Data Protection: HDFS and/or Ceph replication or Erasure Codes
- Tiering: HDFS and/or Ceph

QCT Lab Test Environment – System Configuration

1 x Cloudera Manager

2xIntel® Xeon® E5-2670 v3
2x Intel® S3710 400G SSD
10G dual
24x16GB RAM



2 x HDFS Name Node

2x Intel® Xeon® E5-2680 v3
2x Intel® S3700 400G SSD
40G dual (Intel® XL710-QDA2), 8x16GB RAM

16 x HDFS Data Node

2x Intel® Xeon® E5-2680 v3
2x Intel® S3700 400G SSD
40G dual (Intel® XL710-QDA2), 8x16GB RAM

OS: RHEL 7.3
Hadoop version: CDH Data Hub Edition Trial 5.10.0

3 x Ceph Monitor

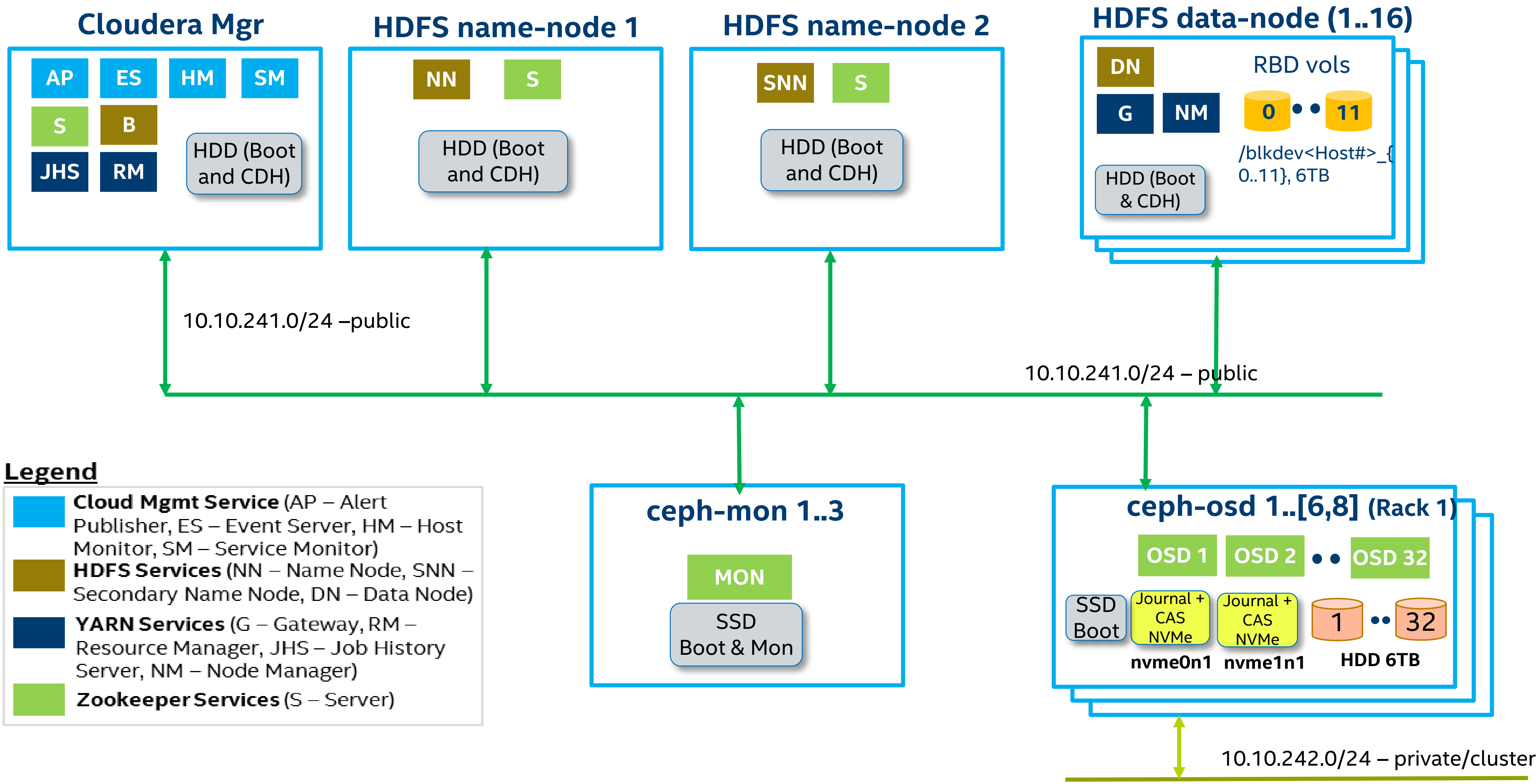
2x Intel® Xeon® E5-2620 v3
2x Intel S3700 400G SSD
40G dual (Intel® XL710-QDA2), 8x16GB RAM

10 x Ceph OSD

2x Intel® Xeon® E5-2680 v3, 2x 2TB P3700 NVMe SSDs, 1x S3700 400G
35x 6TB SATA HDD (ST6000NM0024)
40G dual (Mellanox MT27520)
8x16GB RAM

OS: RHEL 7.3
Ceph version: RedHat Ceph Storage 2.1

QCT Lab Test Setup #1 *(Cloudera Hadoop 5.10.0 & RedHat Ceph Storage 2.1/FileStore)*



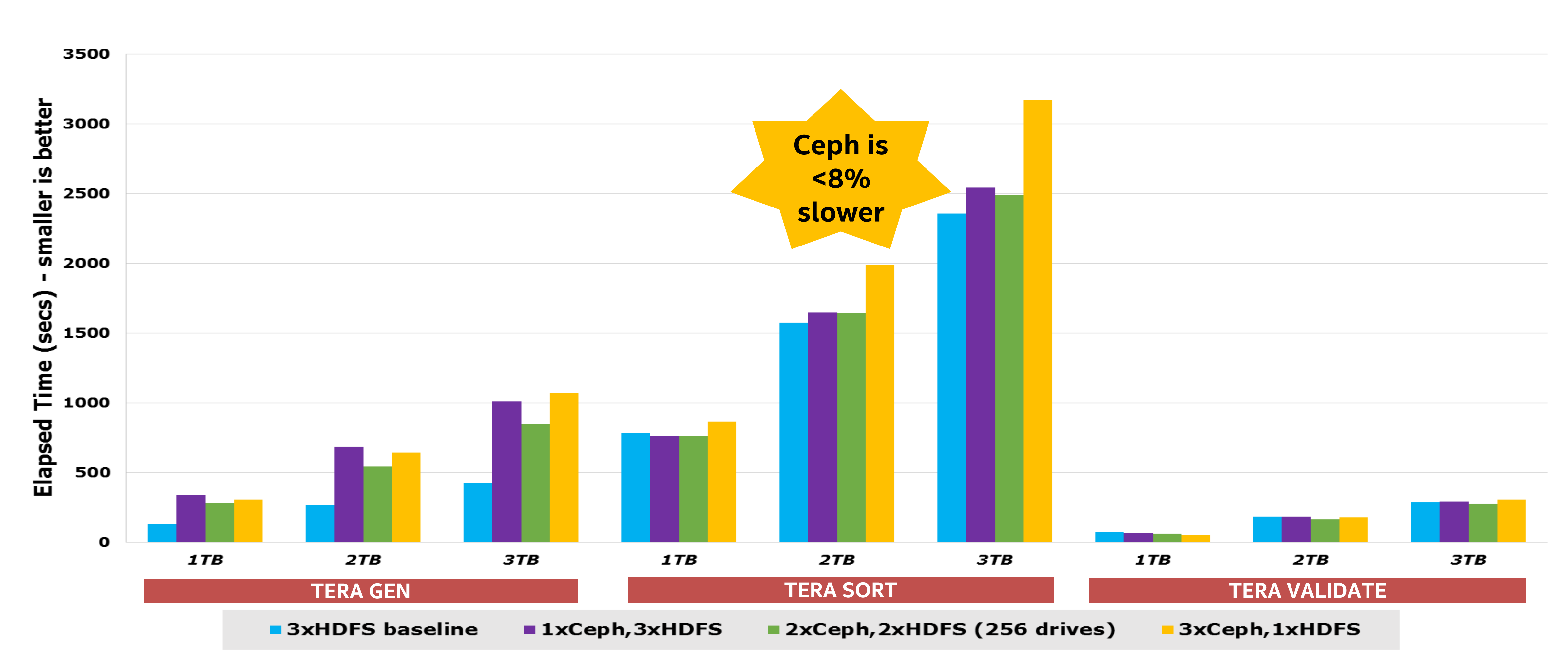
NOTE: BMC management network is not shown

Workloads – MapReduce

Tera Benchmark Suite

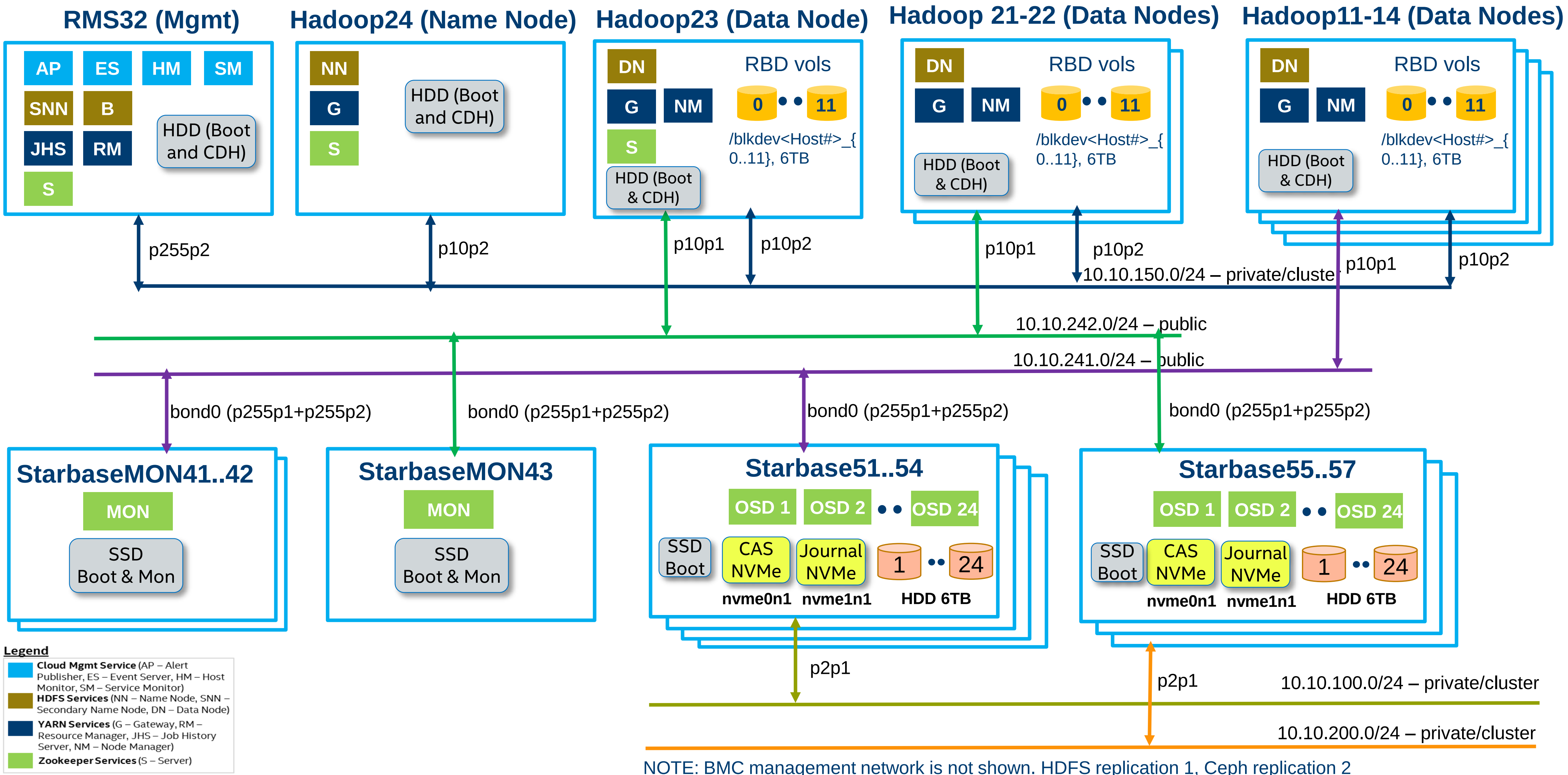
- Most popular Hadoop test, supplied with distribution, exercises CPU, memory, disk, network
- TeraGen – generates specified number of 100 byte records – 1, 2, and 3 TB used in tests
- TeraSort – sorts TeraGen output
- TeraValidate – validates TeraSort output is in sorted order

Performance Results *(journal on NVMe SSD, no read cache)*



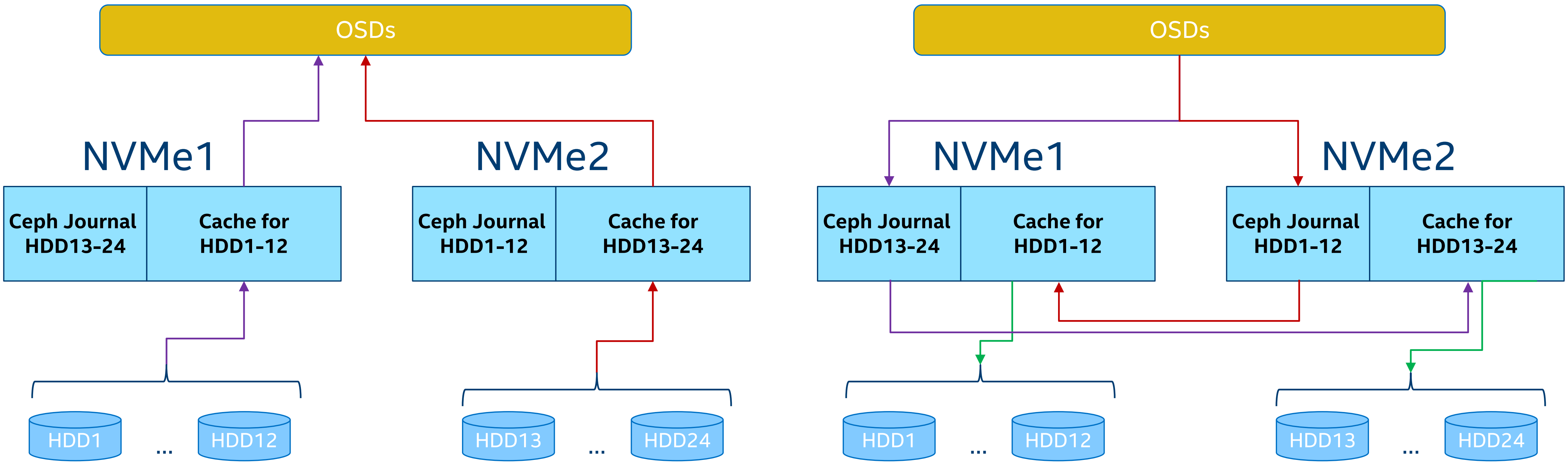
Replication in both Ceph and HDFS provides best performance with resiliency but at the expense of extra copy

QCT Lab Test Setup #2 *(Cloudera Hadoop 5.7.0 & Ceph Jewel 10.2.1/FileStore)*



NOTE: BMC management network is not shown. HDFS replication 1, Ceph replication 2

Intel CAS and Ceph Journal Configuration

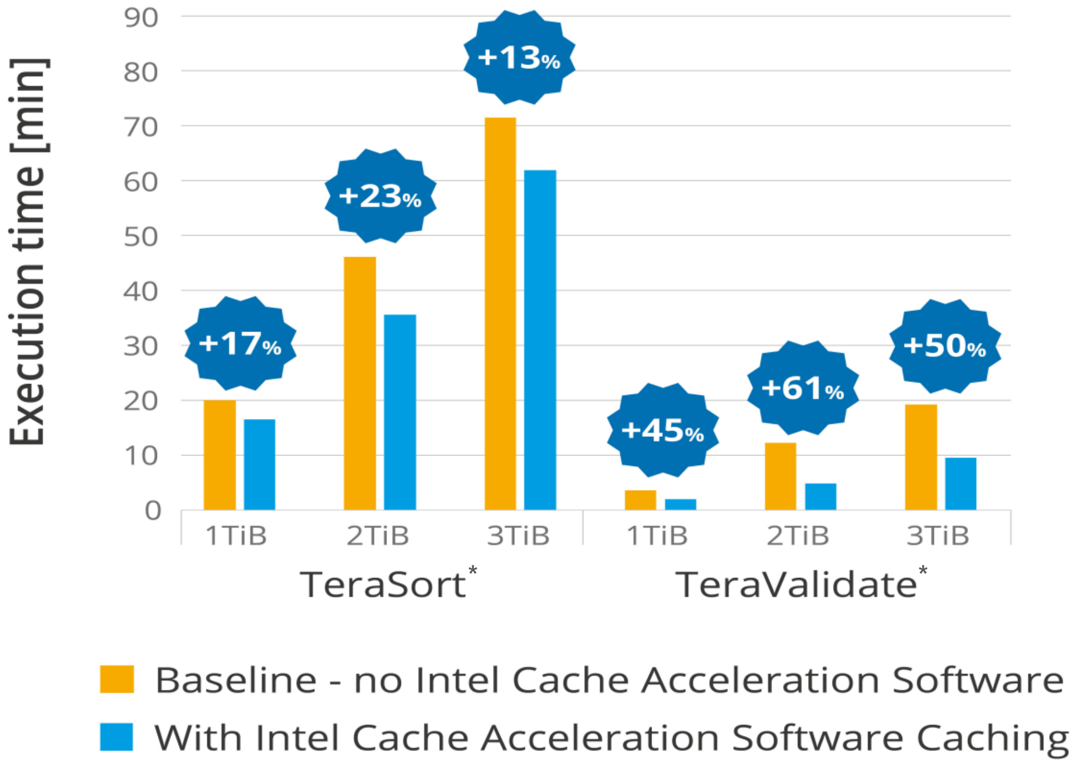


Reads

- Ceph Journal[1-24]: 20G each, 480G in Total
- Intel CAS[1-4]: 880G each, ~3520TB in Total

Writes

Performance Results #2 *(journal and read cache on NVMe SSD)*



HDFS replication 1, Ceph replication 2

Optimize performance with Intel® CAS and Intel® SSDs using NVMe*

- Resolve input/output (I/O) bottlenecks
- Provide better customer service-level-agreement (SLA) support
- Provide up to a 60-percent I/O performance improvement[2]

Summary

- Both storage and compute resources are bound to Hadoop nodes resulting in excess capacity, increased cost and scale challenges.
- Disaggregating storage from Hadoop helps in independent scaling, improves resource efficiency and performance.
- Hadoop over Ceph with Intel® technologies delivers optimum performance while achieving scalability and flexibility.

Find Out More

To learn more about Intel® CAS and request a trial copy, visit: intel.com/content/www/us/en/software/intel-cache-acceleration-software-performance.html

To find the Intel® SSD that's right for you, visit: intel.com/go/ssd

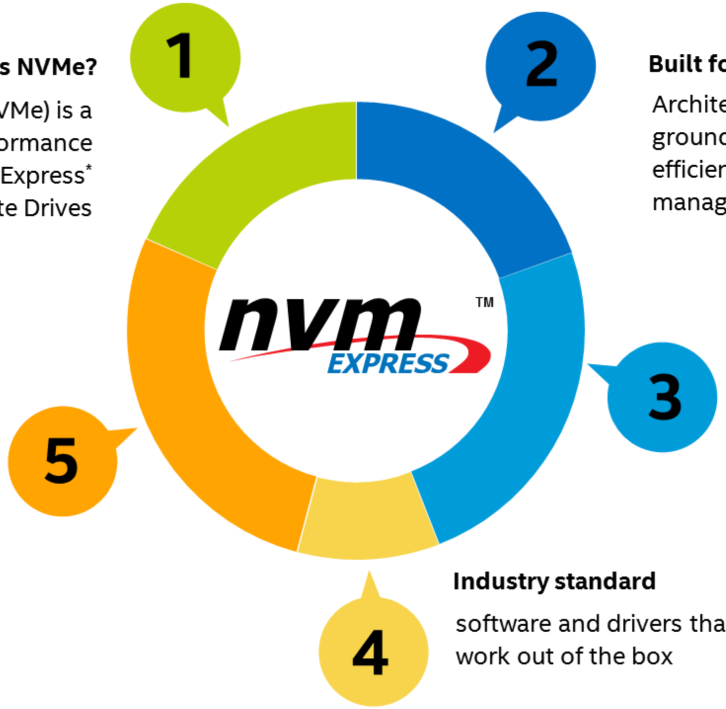
To learn about QCT QxStor* Red Hat* Ceph* Storage Edition, visit: qct.io/solution/software-defined-infrastructure/storage-virtualization/qxstor-red-hat-ceph-storage-edition-p365c225c226c230

THANK YOU!

BACKUP SLIDES

NVM Express (NVMe)

Standardized interface for non-volatile memory, http://nvmexpress.org



What is NVMe?

NVM Express* (NVMe) is a standardized high performance software interface for PCI Express* Solid State Drives

Built for SSDs

Architected from the ground up for SSDs to be efficient, scalable, and manageable

Developed to be lean

Streamlined protocol with new efficient queuing mechanism to scale for multi-core CPUs, low clock cycles per IO

Industry standard

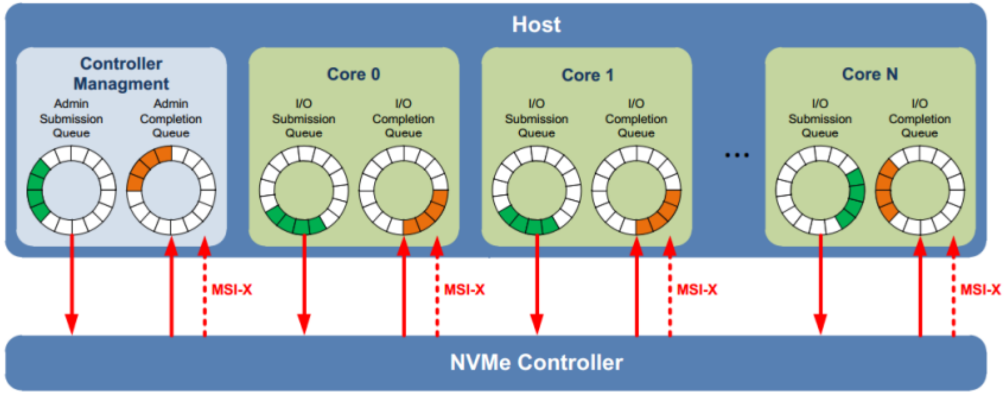
software and drivers that work out of the box

Ready for next generation SSDs

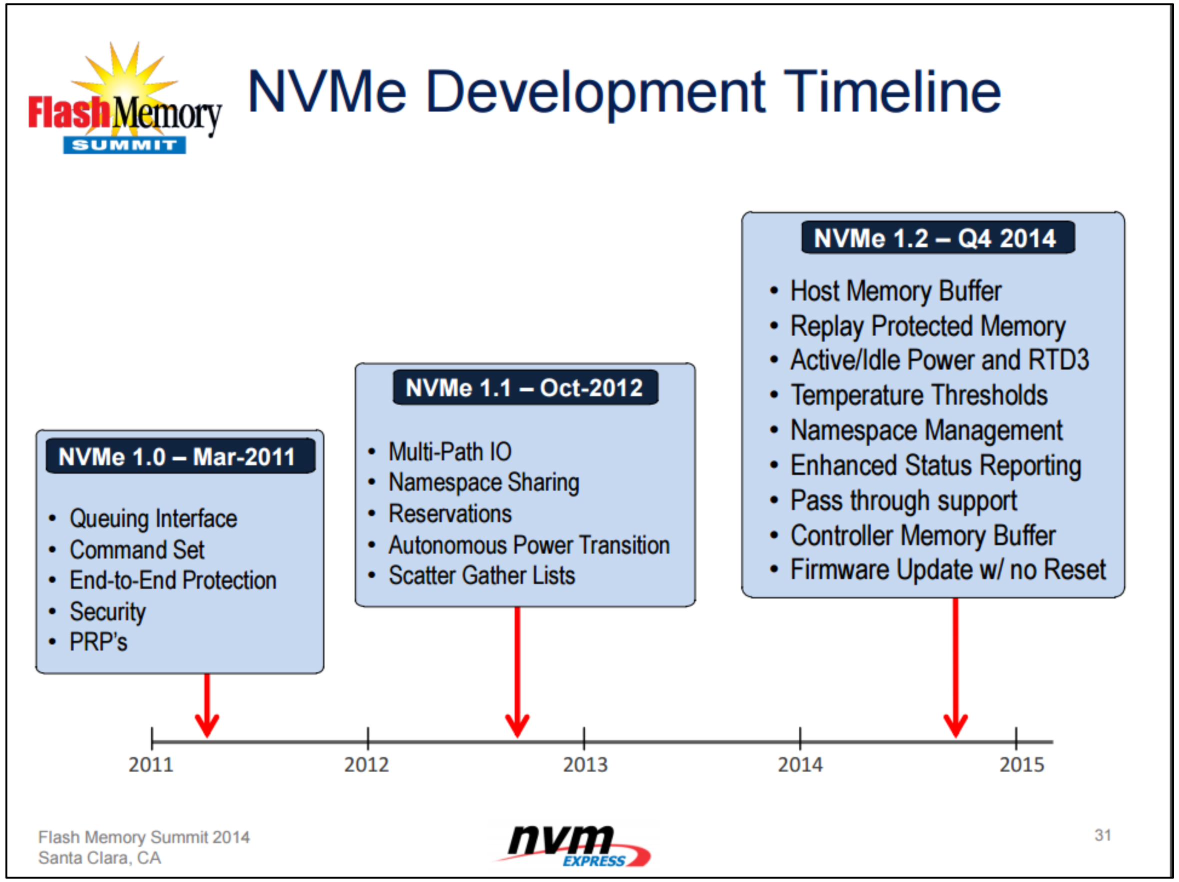
New storage stack with low latency and small overhead to take full advantage of next generation NVM

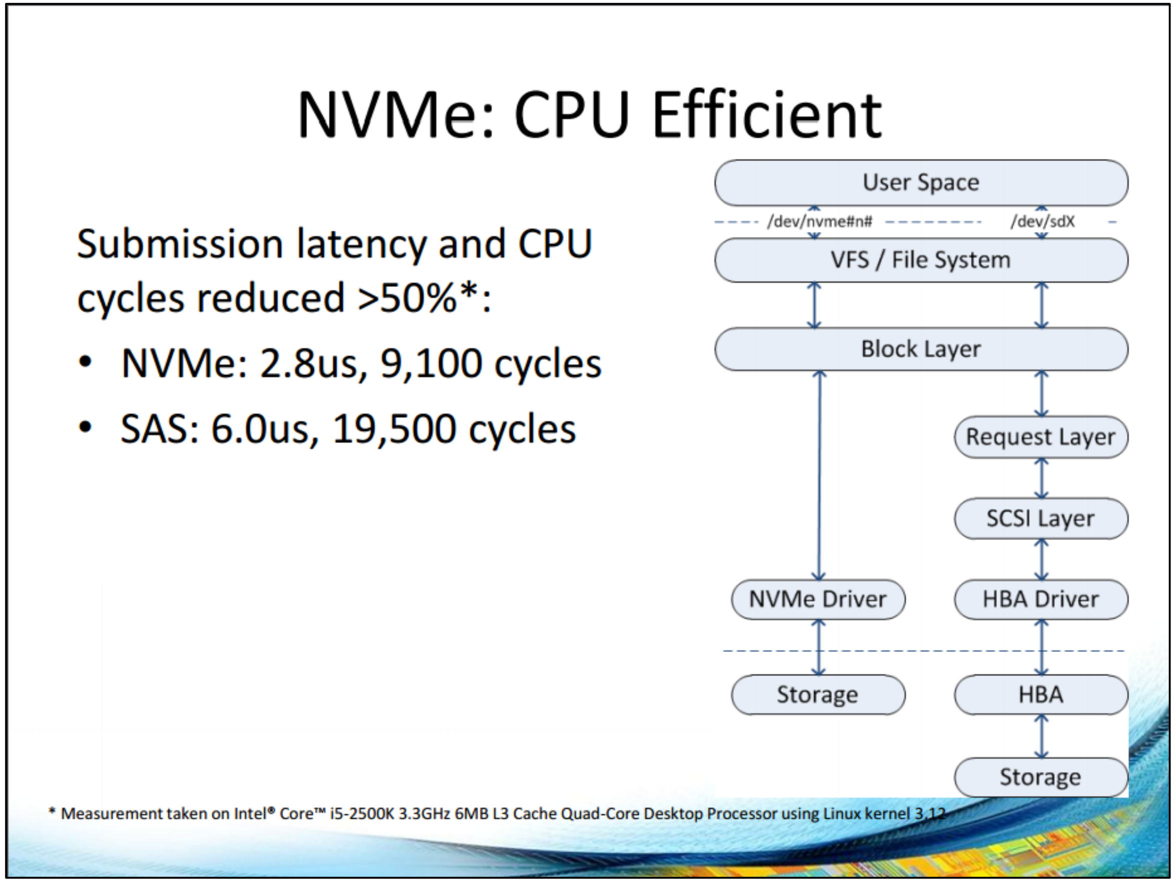
- Performance: 1 GB/s per lane.. 4 GB/s, 8 GB/s, 16 GB/s per device..
- Lower latency: Direct CPU connection
- No host bus adapter (HBA): Lower power ~ 10W and cost ~ $15
- Increased I/O opportunity: Up to 40 PCIe lanes per CPU socket
- Form factor options: PCIe add-in-card, SFF-8639, M.2, SATA Express, BGA

M.2 **U.2 2.5in (SFF-8639)** **Add-in-card**



NVM Express


Flash Memory SUMMIT
NVMe Development Timeline
NVMe 1.0 – Mar-2011
• Queuing Interface
• Command Set
• End-to-End Protection
• Security
• PRP's
NVMe 1.1 – Oct-2012
• Multi-Path IO
• Namespace Sharing
• Reservations
• Autonomous Power Transition
• Scatter Gather Lists
NVMe 1.2 – Q4 2014
• Host Memory Buffer
• Replay Protected Memory
• Active/Idle Power and RTD3
• Temperature Thresholds
• Namespace Management
• Enhanced Status Reporting
• Pass through support
• Controller Memory Buffer
• Firmware Update w/ no Reset
2011
2012
2013
2014
2015
Flash Memory Summit 2014
Santa Clara, CA
nvm EXPRESS
31


NVMe: CPU Efficient
Submission latency and CPU cycles reduced >50%*:
• NVMe: 2.8us, 9,100 cycles
• SAS: 6.0us, 19,500 cycles
User Space
/dev/nvme#n#
/dev/sdX
VFS / File System
Block Layer
Request Layer
SCSI Layer
NVMe Driver
HBA Driver
Storage
HBA
Storage
* Measurement taken on Intel® Core™ i5-2500K 3.3GHz 6MB L3 Cache Quad-Core Desktop Processor using Linux kernel 3.12

Advantages of Ceph* Storage vs. Local Storage

Free (if self-supported)	Supports all data types: file, block, and object data	Provides one centralized, standardized, and scalable storage solution for all enterprise needs
Open source	Supports many different workloads and applications	Works on commodity hardware

*Other names and brands may be claimed as the property of others.

Hadoop with Ceph* on QCT Platform

Physical architecture

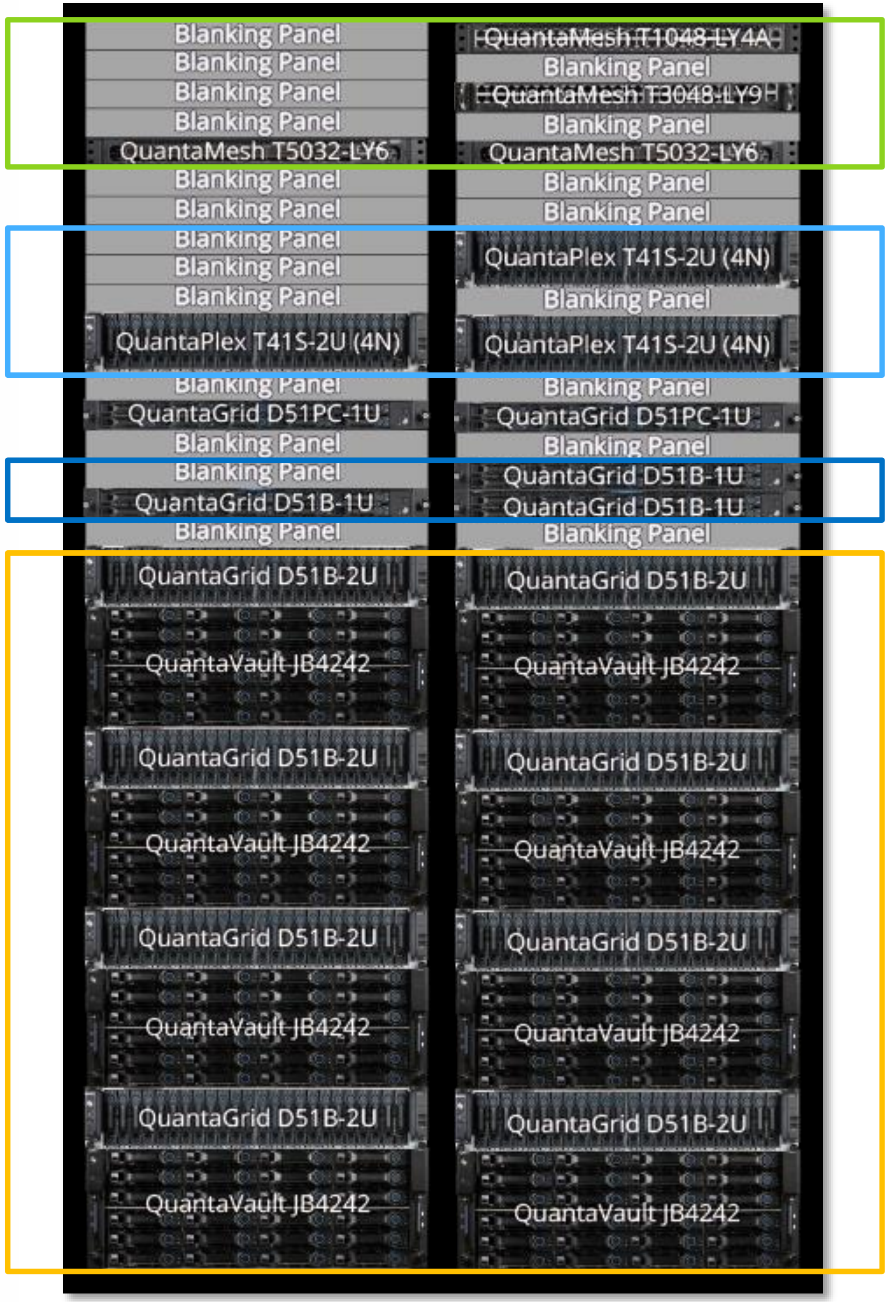
QCT Solution Center

Networking

Apache Hadoop*

Ceph Monitors

Ceph OSD x8



Test Setup (Linux OS)

/etc/sysctl.conf

```
vm.swappiness=10
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.netdev_max_backlog = 250000
```

/etc/security/limits.conf

```
* soft nofile 65536
* hard nofile 1048576
* soft nproc 65536
* hard nproc unlimited
* hard memlock unlimited
```

CPU Profile

```
echo performance> /sys/devices/system/cpu/cpu{0..n}/cpufreq/scaling_governor
```

Huge Page

```
echo never> /sys/kernel/mm/transparent_hugepage/defrag
echo never> /sys/kernel/mm/transparent_hugepage/enabled
```

Network

```
ifconfig <eth> mtu 9000
ifconfig <eth> txqueuelen 1000
```

Test Setup (Ceph)

```
[global]
fsid = f1739148-3847-424d-b262-45d5b950fa3b
mon_initial_members = starbasemon41, starbasemon42, starbasemon43
mon_host = 10.10.241.41,10.10.241.42,10.10.242.43
auth_client_required = none
auth_cluster_required = none
auth_service_required = none
filestore_xattr_use_omap = true
osd_pool_default_size = 3  # Write an object 2 times.
osd_pool_default_min_size = 3 # Allow writing one copy in a degraded state.
osd_pool_default_pg_num = 4800
osd_pool_default_pgp_num = 4800
public_network = 10.10.241.0/24, 10.10.242.0/24
cluster_network = 10.10.100.0/24, 10.10.200.0/24
debug_lockdep = 0/0
debug_context = 0/0
debug_crush = 0/0
debug_buffer = 0/0
debug_timer = 0/0
debug_filer = 0/0
debug_objecter = 0/0
debug_rados = 0/0
debug_rbd = 0/0
debug_ms = 0/0
debug_monc = 0/0
debug_tp = 0/0
debug_auth = 0/0
debug_finisher = 0/0
debug_heartbeatmap = 0/0
debug_perfcounter = 0/0
```

```
[global]
debug_asok = 0/0
debug_throttle = 0/0
debug_mon = 0/0
debug_paxos = 0/0
debug_rgw = 0/0
perf = true
mutex_perf_counter = true
throttler_perf_counter = false
rbd_cache = false
log_file = /var/log/ceph/$name.log
log_to_syslog = false
mon_compact_on_trim = false
osd_pg_bits = 8
osd_pgp_bits = 8
mon_pg_warn_max_object_skew = 100000
mon_pg_warn_min_per_osd = 0
mon_pg_warn_max_per_osd = 32768
```

Test Setup (Ceph)

```
[mon]
mon_host = starbasemon41, starbasemon42, starbasemon43
mon_data = /var/lib/ceph/mon/$cluster-$id
mon_max_pool_pg_num = 166496
mon_osd_max_split_count = 10000
mon_pg_warn_max_per_osd = 10000

[mon.a]
host = starbasemon41
mon_addr = 192.168.241.41:6789

[mon.b]
host = starbasemon42
mon_addr = 192.168.241.42:6789

[mon.c]
host = starbasemon43
mon_addr = 192.168.242.43:6789
```

```
[osd]
osd_mount_options_xfs =
rw,noatime,inode64,logbsize=256k,delaylog
osd_mkfs_options_xfs = -f -i size=2048
osd_op_threads = 32
filestore_queue_max_ops = 5000
filestore_queue_committing_max_ops = 5000
journal_max_write_entries = 1000
journal_queue_max_ops = 3000
objecter_inflight_ops = 102400
filestore_wbthrottle_enable = false
filestore_queue_max_bytes = 1048576000
filestore_queue_committing_max_bytes = 1048576000
journal_max_write_bytes = 1048576000
journal_queue_max_bytes = 1048576000
```

Test Setup (Hadoop)

Parameter	Value	Comment
Container Memory yarn.nodemanager.resource.memory-mb	80.52 GiB	Default: Amount of physical memory, in MiB, that can be allocated for containers NOTE: In a different document, it recommends
Container Virtual CPU Cores yarn.nodemanager.resource.cpu-vcores	48	Default: Number of virtual CPU cores that can be allocated for containers.
Container Memory Maximum yarn.scheduler.maximum-allocation-mb	12 GiB	The largest amount of physical memory, in MiB, that can be requested for a container.
Container Virtual CPU Cores Maximum yarn.scheduler.maximum-allocation-vcores	48	Default: The largest number of virtual CPU cores that can be requested for a container.
Container Virtual CPU Cores Minimum yarn.scheduler.minimum-allocation-vcores	2	The smallest number of virtual CPU cores that can be requested for a container. If using the Capacity or FIFO scheduler (or any scheduler, prior to CDH 5), virtual core requests will be rounded up to the nearest multiple of this number.
JobTracker MetaInfo Maxsize mapreduce.job.split.metainfo.maxsize	1000000000	The maximum permissible size of the split metainfo file. The JobTracker won't attempt to read split metainfo files bigger than the configured value. No limits if set to -1.
I/O Sort Memory Buffer (MiB) mapreduce.task.io.sort.mb	400 MiB	To enable larger blocksize without spills
yarn.scheduler.minimum-allocation-mb	2 GiB	Default: Minimum container size
mapreduce.map.memory.mb	1 GiB	Memory req'd for each type of container - may want to increase for some apps
mapreduce.reduce.memory.mb	1.5 GiB	Memory req'd for each type of container - may want to increase for some apps
mapreduce.map.cpu.vcores	1	Default: Number of vcores req'd for each type of container
mapreduce.reduce.cpu.vcores	1	Default: Number of vcores req'd for each type of container
mapreduce.job.heap.memory-mb.ratio	0.8	(Default). This sets Java heap size = 800/1200 MiB for mapreduce.{map|reduce}.memory.mb = 1/1.5 GiB

Test Setup (Hadoop)

Parameter	Value	Comment
dfs.blocksize	128 MiB	Default
dfs.replication	1	Default block replication. The number of replications to make when the file is created. The default value is used if a replication number is not specified.
Java Heap Size of NameNode in Bytes	4127MiB	Default: Maximum size in bytes for the Java Process heap memory. Passed to Java -Xmx.
Java Heap Size of Secondary NameNode in Bytes	4127MiB	Default: Maximum size in bytes for the Java Process heap memory. Passed to Java -Xmx.

Parameter	Value	Comment
Memory overcommit validation threshold	0.9	Threshold used when validating the allocation of RAM on a host. 0 means all of the memory is reserved for the system. 1 means none is reserved. Values can range from 0 to 1.

Test Setup (CAS NVMe, Journal NVMe)

NVMe0n1	NVMe1n1
Ceph journal configured for 1st 12 HDDs will be /dev/nvme0n1p1 - /dev/nvme0n1p12 Each Partition size: 20GiB	Ceph Journal configured for remaining 12 HDDs will be /dev/nvme1n1p1 - /dev/nvme1n1p12 Each Partition size: 20GiB
CAS for 12-24 HDDs will be from this SSD. Use rest of the free space and split evenly for 2 cache partitions e.g. /dev/sdo - /dev/sdz cache 1 /dev/nvme0n1p13 Running wo - ├core 1 /dev/sdo1 - - /dev/intelcas1-1 ├core 2 /dev/sdp1 - - /dev/intelcas1-2 ├core 3 /dev/sdq1 - - /dev/intelcas1-3 ├core 4 /dev/sdr1 - - /dev/intelcas1-4 ├core 5 /dev/sds1 - - /dev/intelcas1-5 └core 6 /dev/sdt1 - - /dev/intelcas1-6 cache 2 /dev/nvme0n1p14 Running wo - ├core 1 /dev/sdu1 - - /dev/intelcas2-1 ├core 2 /dev/sdv1 - - /dev/intelcas2-2 ├core 3 /dev/sdw1 - - /dev/intelcas2-3 ├core 4 /dev/sdx1 - - /dev/intelcas2-4 ├core 5 /dev/sdy1 - - /dev/intelcas2-5 └core 6 /dev/sdz1 - - /dev/intelcas2-6	CAS for 1-12 HDDs will be from this SSD. Use rest of the free space and split evenly for 2 cache partitions e.g. /dev/sdc - /dev/sdn cache 1 /dev/nvme1n1p13 Running wo - ├core 1 /dev/sdc1 - - /dev/intelcas1-1 ├core 2 /dev/sdd1 - - /dev/intelcas1-2 ├core 3 /dev/sde1 - - /dev/intelcas1-3 ├core 4 /dev/sdf1 - - /dev/intelcas1-4 ├core 5 /dev/sdg1 - - /dev/intelcas1-5 └core 6 /dev/sdh1 - - /dev/intelcas1-6 cache 2 /dev/nvme1n1p14 Running wo - ├core 1 /dev/sdi1 - - /dev/intelcas2-1 ├core 2 /dev/sdj1 - - /dev/intelcas2-2 ├core 3 /dev/sdk1 - - /dev/intelcas2-3 ├core 4 /dev/sdl1 - - /dev/intelcas2-4 ├core 5 /dev/sdm1 - - /dev/intelcas2-5 └core 6 /dev/sdn1 - - /dev/intelcas2-6