Accelerating Ceph data services

with Intel® ISA-L and QuickAssist® Technology

Greg Tucker, Tushar Gohad, Brian Will

Credits

This work wouldn’t have been possible without contributions from –

Reddy Chagam (anjaneya.chagam@intel.com)

Weigang Li (weigang.li@intel.com)

Praveen Mosur (praveen.mosur@intel.com)

Edward Pullin (edward.j.pullin@intel.com)

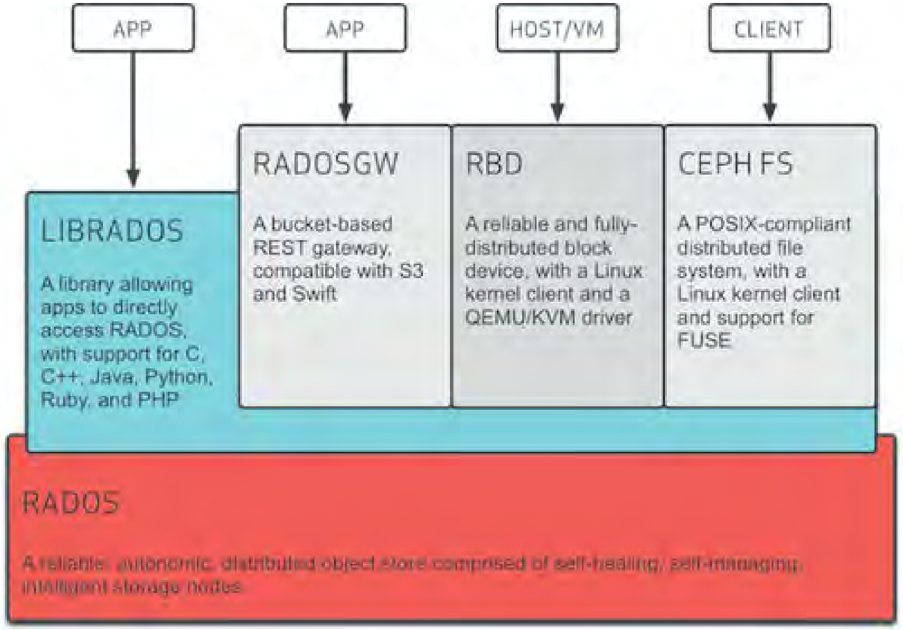
Agenda

- Ceph
  - A Quick Primer
  - Storage Efficiency and Security Features
- Storage Workload Acceleration
  - Software and Hardware Approaches
- Ceph Data Services
  - Erasure Coding and ISA-L based acceleration
  - Compression and hardware acceleration based on QAT
- Key Takeaways

Ceph scale-out storage

Ceph

- Open-source, object-based scale-out storage system
- Software-defined, hardware-agnostic – runs on commodity hardware
- Object, Block and File support in a unified storage cluster
- Highly durable, available – replication, erasure coding
- Replicates and re-balances dynamically



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

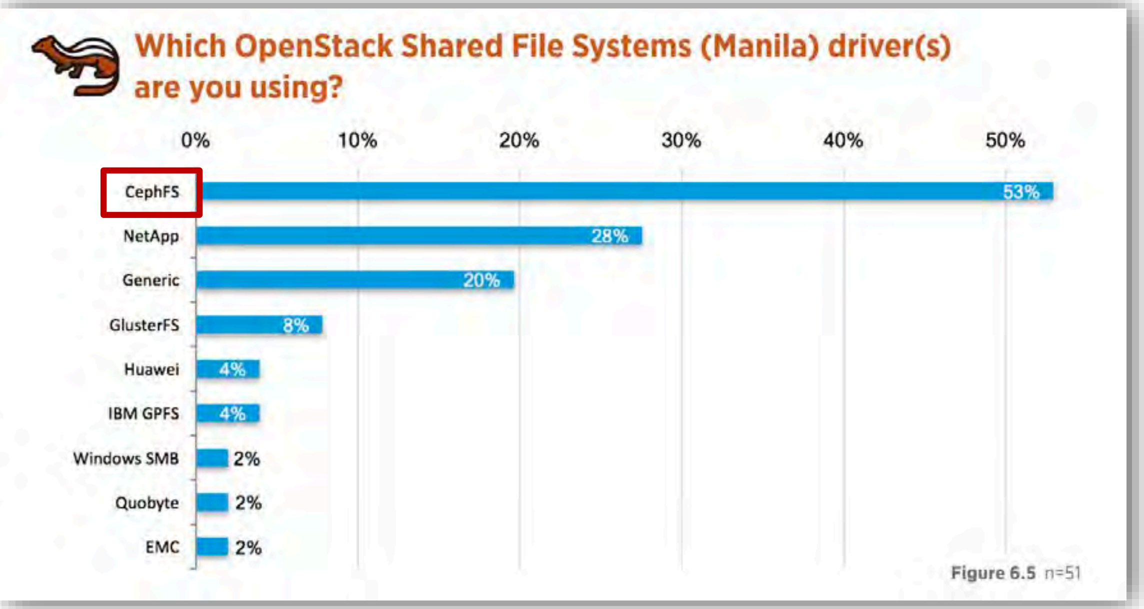
RADOS

A reliable, autonomic, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Image source: http://ceph.com/ceph-storage

Ceph

- Scalability – CRUSH data placement, no single POF
- Enterprise features – snapshots, cloning, mirroring
- Most popular block and file storage for Openstack use cases
- 10 years of hardening, vibrant community



Which OpenStack Shared File Systems (Manila) driver(s) are you using?

Figure 6.5 n=51

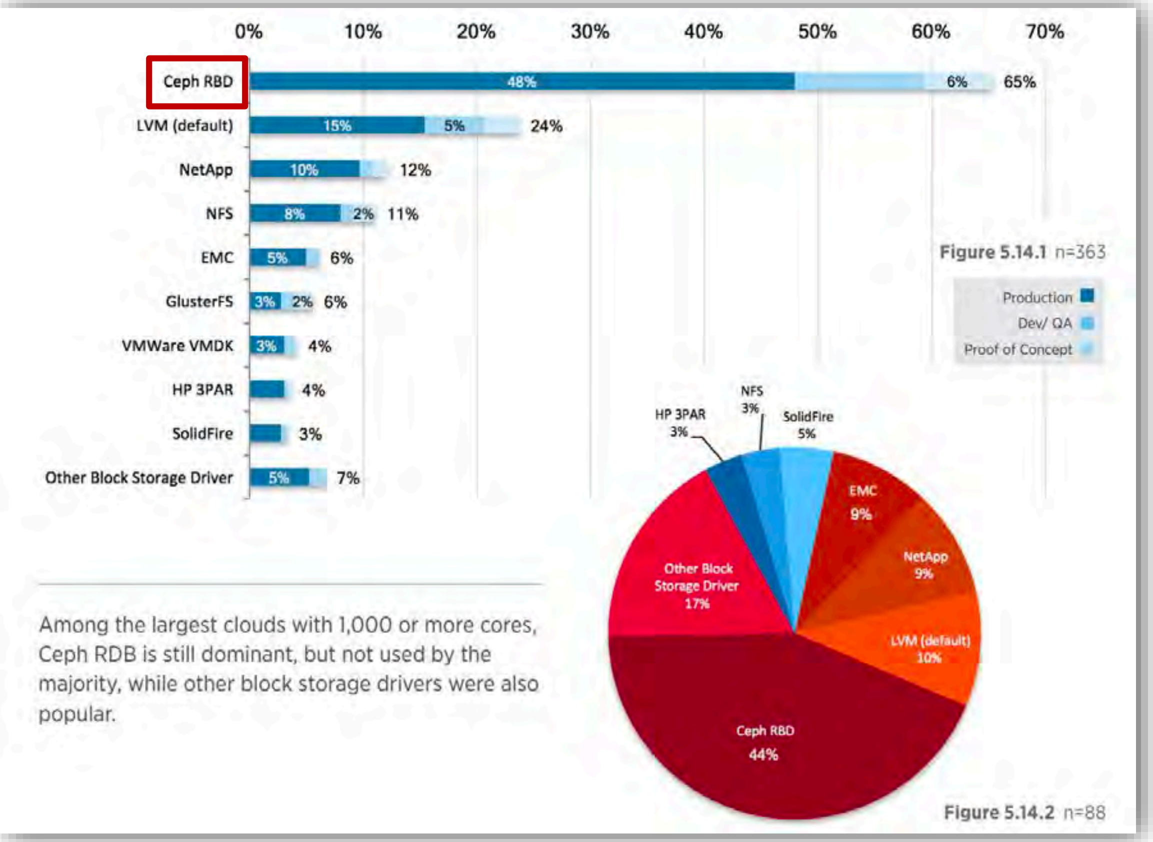


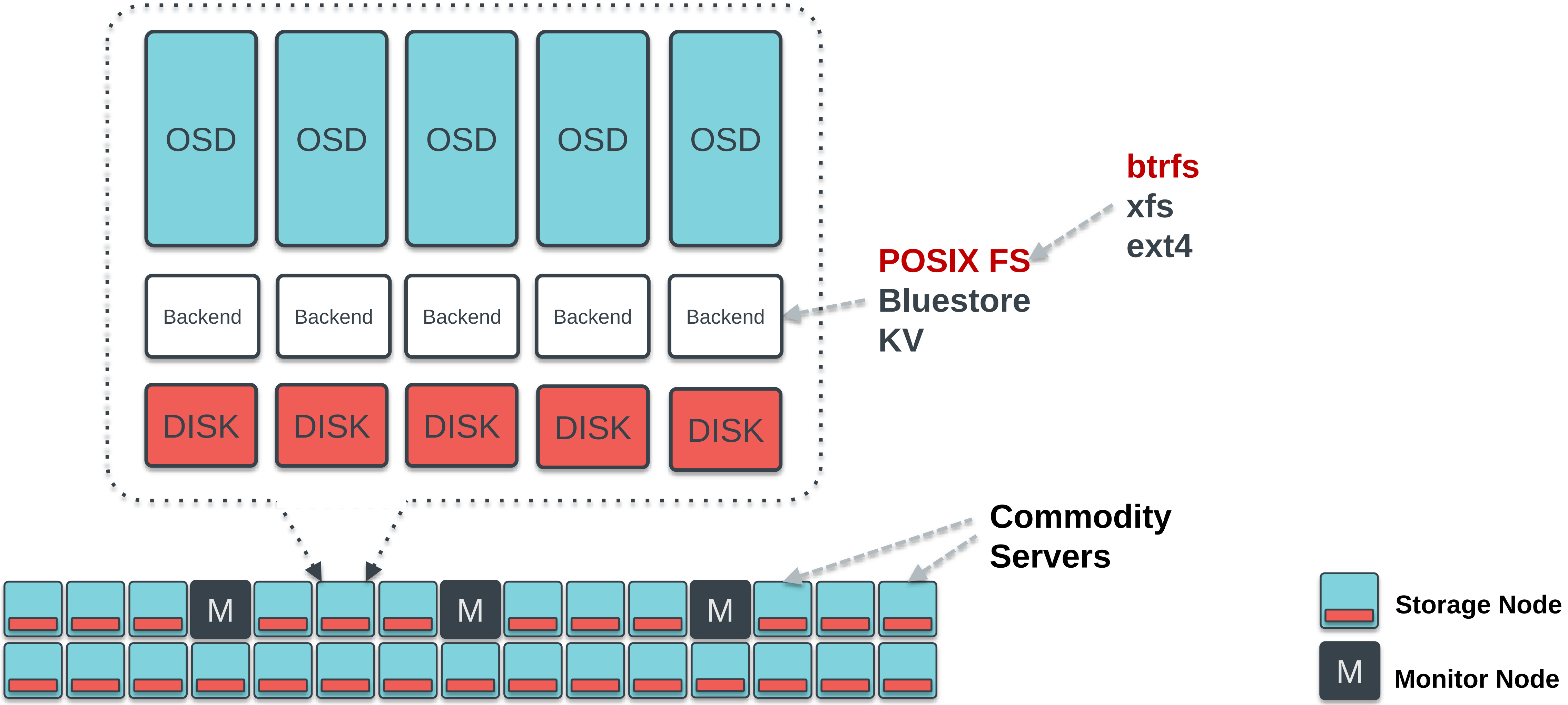
Figure 5.14.1 n=363

Among the largest clouds with 1,000 or more cores, Ceph RDB is still dominant, but not used by the majority, while other block storage drivers were also popular.

Figure 5.14.2 n=88

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

Ceph: Architecture



btrfs
xfs
ext4

POSIX FS
Bluestore
KV

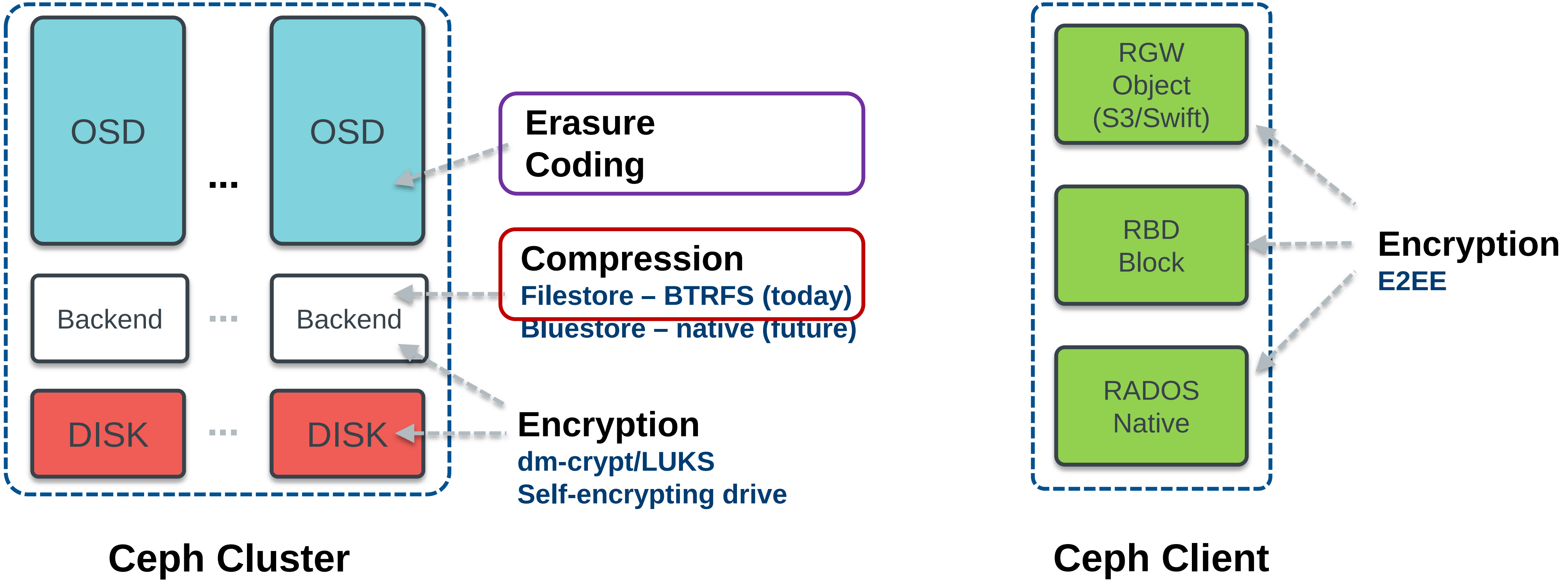
Commodity Servers

Storage Node

M Monitor Node

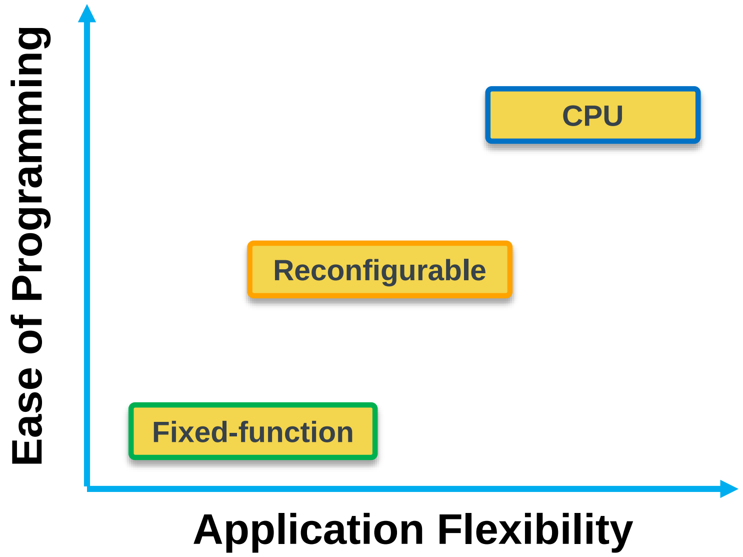
Ceph: Storage Efficiency, Security

Erasure Coding, Compression, Encryption



Storage Workload Acceleration

Software and Hardware-based Approaches and Trade-offs



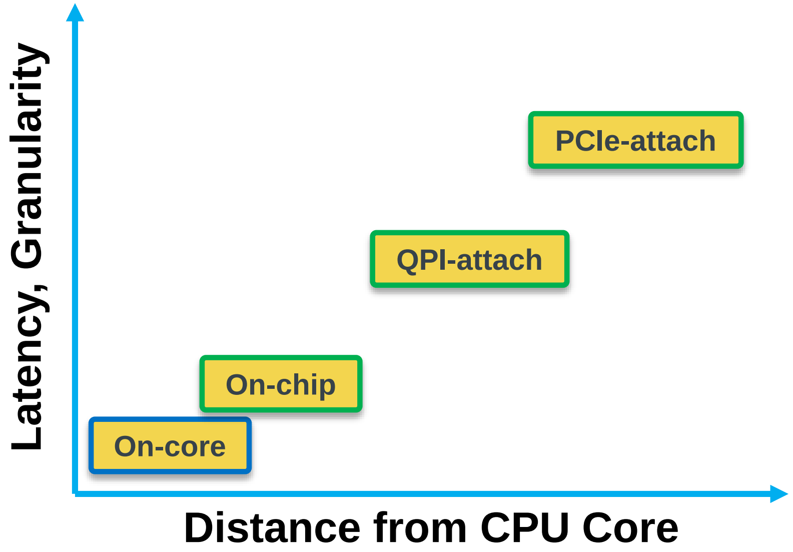
Ease of Programming

CPU

Reconfigurable

Fixed-function

Application Flexibility



Latency, Granularity

PCIe-attach

QPI-attach

On-chip

On-core

Distance from CPU Core

ISA-L (SIMD)

ASIC (QAT)

FPGA

GPGPU

Software-based Approaches

Fixed-function, Reconfigurable Approaches

Intel® ISA-L

Intel® ISA-L **Value Proposition**

Algorithmic Library

for core storage algorithms where throughput and latency are the most critical factors

Optimized Libraries for the fundamental building blocks of storage software on Intel® Architecture

Enhances Performance for data integrity, security/encryption, data protection, and compression algorithms

Single API call delivers the optimal implementation for past, present and future Intel processors

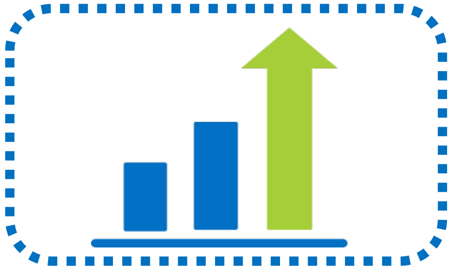
Validated on Linux*, BSD, and Windows Server* operating systems

Intel® ISA-L **Value Proposition**

Pure assembly library

hand-optimized to take advantage of each and every Intel CPU cycle

Fantastic Performance
5X faster compression, 15X faster hashing



Future Proof & Backwards Compatible single API for all platforms, delivering the best available implementation at runtime

Operating System Agnostic
optimize in Windows, Linux, FreeBSD, or any other OS environment running on x86

Free and Open Source
Licensed under BSD for maximum adoption, commercially and open source compatible

Where is ISA-L used?

Open Source Projects

- Scale-out storage (HDFS, Ceph & Swift)
- Streaming encryption (Netflix)
- Deduplication software
- File systems

Proprietary Projects

- Hyperscale object storage
- Deduplication & backup solutions
- Multi-cloud backup
- Low-latency scale-up appliances

Integration Points

Ceph: ISA-L Erasure Code Integrated 2015
http://docs.ceph.com/docs/jewel/rados/operations/erasure-code-isa/

Swift: Policies framework allows liberasure (ISA-L wrapper in Python)
http://docs.openstack.org/developer/swift/overview_erasure_code.html

HDFS: ISA-L Erasure Code Patches in 3.0.0-alpha1, Compression in progress
https://issues.apache.org/jira/browse/HADOOP-11887
https://blog.cloudera.com/blog/2016/02/progress-report-bringing-erasure-coding-to-apache-hadoop/

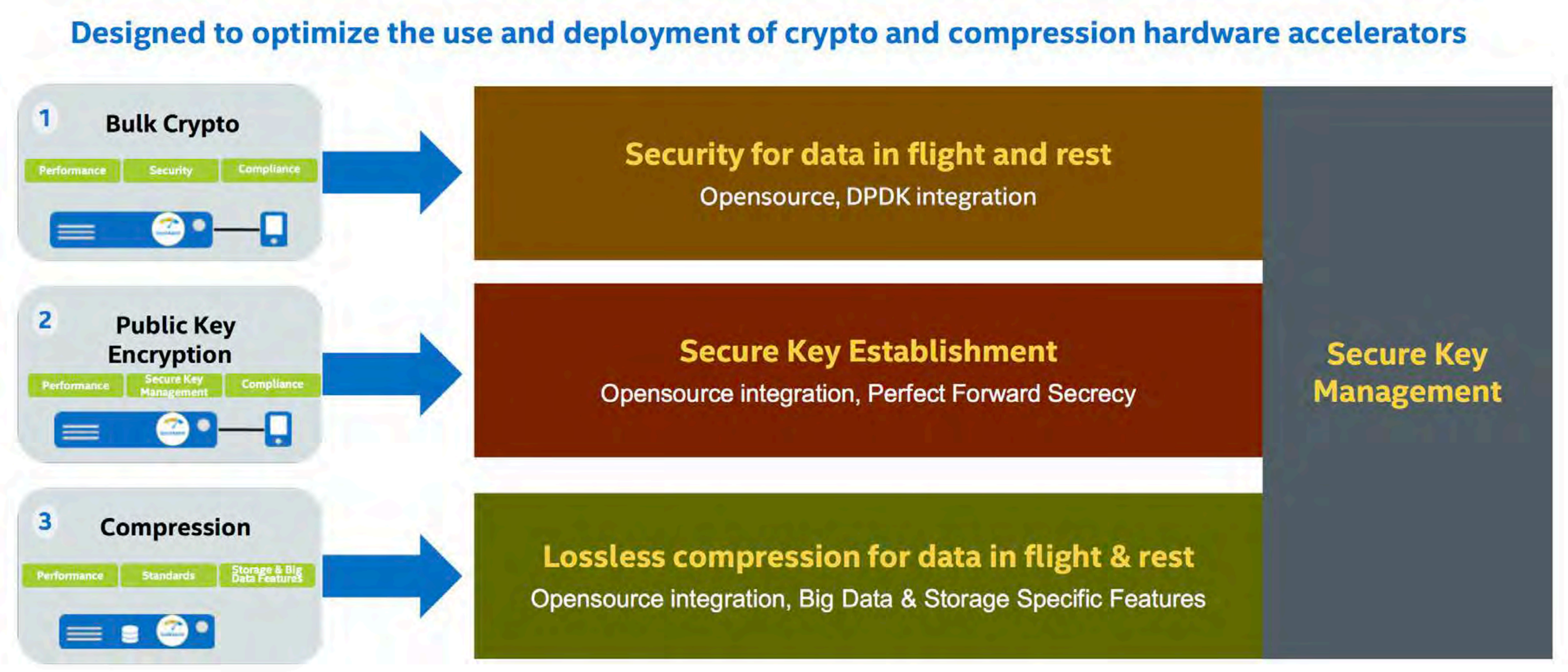
FreeBSD Netflix-Optimized Encryption Path:
http://techblog.netflix.com/2016/08/protecting-netflix-viewing-privacy-at.html

ZFS: Deduplication using ISA-L
http://www.snia.org/sites/default/files/SDC/2016/presentations/capacity_optimization/Xiadong_Qihau_Accelerate_Finger_Printing_in_Data_Deduplication.pdf

InteL® QAT

Hardware-based Acceleration

Intel® QuickAssist Technology



Designed to optimize the use and deployment of crypto and compression hardware accelerators

Security for data in flight and rest

Opensource, DPDK integration

Secure Key Establishment

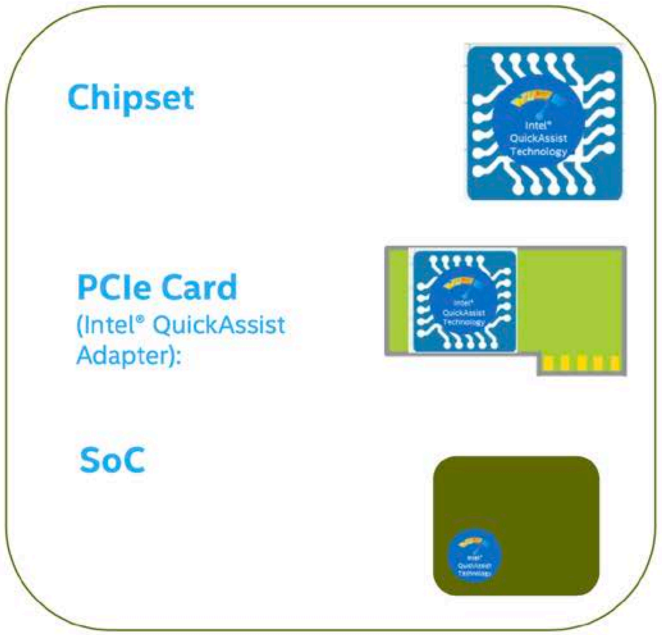
Opensource integration, Perfect Forward Secrecy

Lossless compression for data in flight & rest

Opensource integration, Big Data & Storage Specific Features

Secure Key Management

Intel® QuickAssist Technology Ingredients



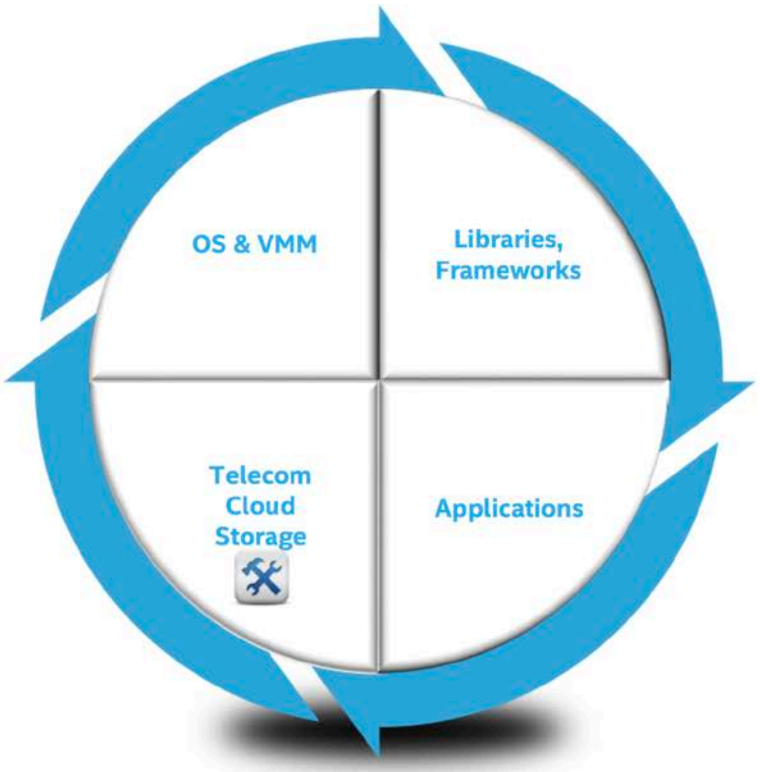
Chipset

PCIe Card

(Intel® QuickAssist Adapter):

SoC

Hardware



OS & VMM

Libraries, Frameworks

Telecom Cloud Storage

Applications

Software

Open-source Software Support

Cryptography	OpenSSL libcrypto, Linux Kernel Crypto Framework
Data Compression	zlib (user API), BTRFS/ZFS (kernel), Ceph, Hadoop, Databases

Ceph and Storage Function Offloads

Intel® ISA-L and QAT

- Erasure Coding
  - ISA-L offload support for Reed-Solomon codes
  - Supported since Hammer
- Compression
  - Filestore
    - QAT offload for BTRFS compression (kernel patch submission in progress)
  - Bluestore
    - ISA-L offload for zlib compression supported in upstream master
    - QAT offload for zlib compression (work-in-progress)
- Encryption
  - RADOS GW
    - RGW object-level encryption with ISA-L and QAT offloads (work-in-progress)

Ceph erasure coding and isa-L

ISA-L: Erasure Codes that Fly

Who is using Erasure Codes?

- “All the clouds” – distributed storage frameworks
- Hadoop HDFS, Ceph, Swift, hyperscalers...

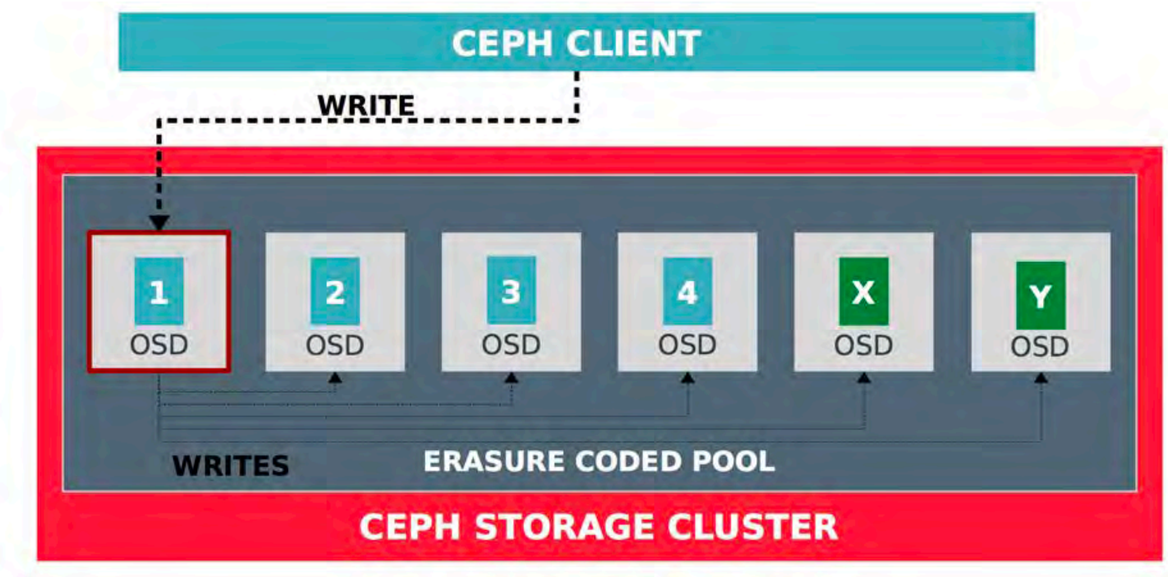
Why are they using Erasure Codes?

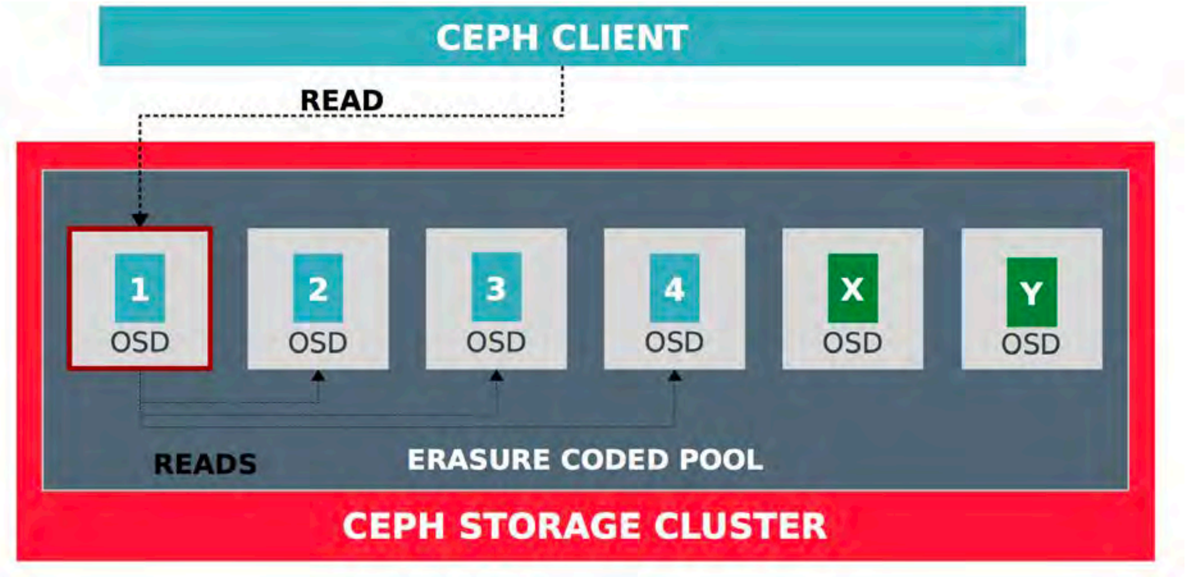
- Irresistible economics: (at least) as much redundancy as triple replication with half the raw data footprint
- Half the storage media costs = big capex and opex savings

Why wasn’t everyone using them before?

- Until ISA-L, EC was computationally prohibitive
- Now very fast

Erasure Coding in Ceph





Write – EC Encode

Read – EC Decode/Reconstruct

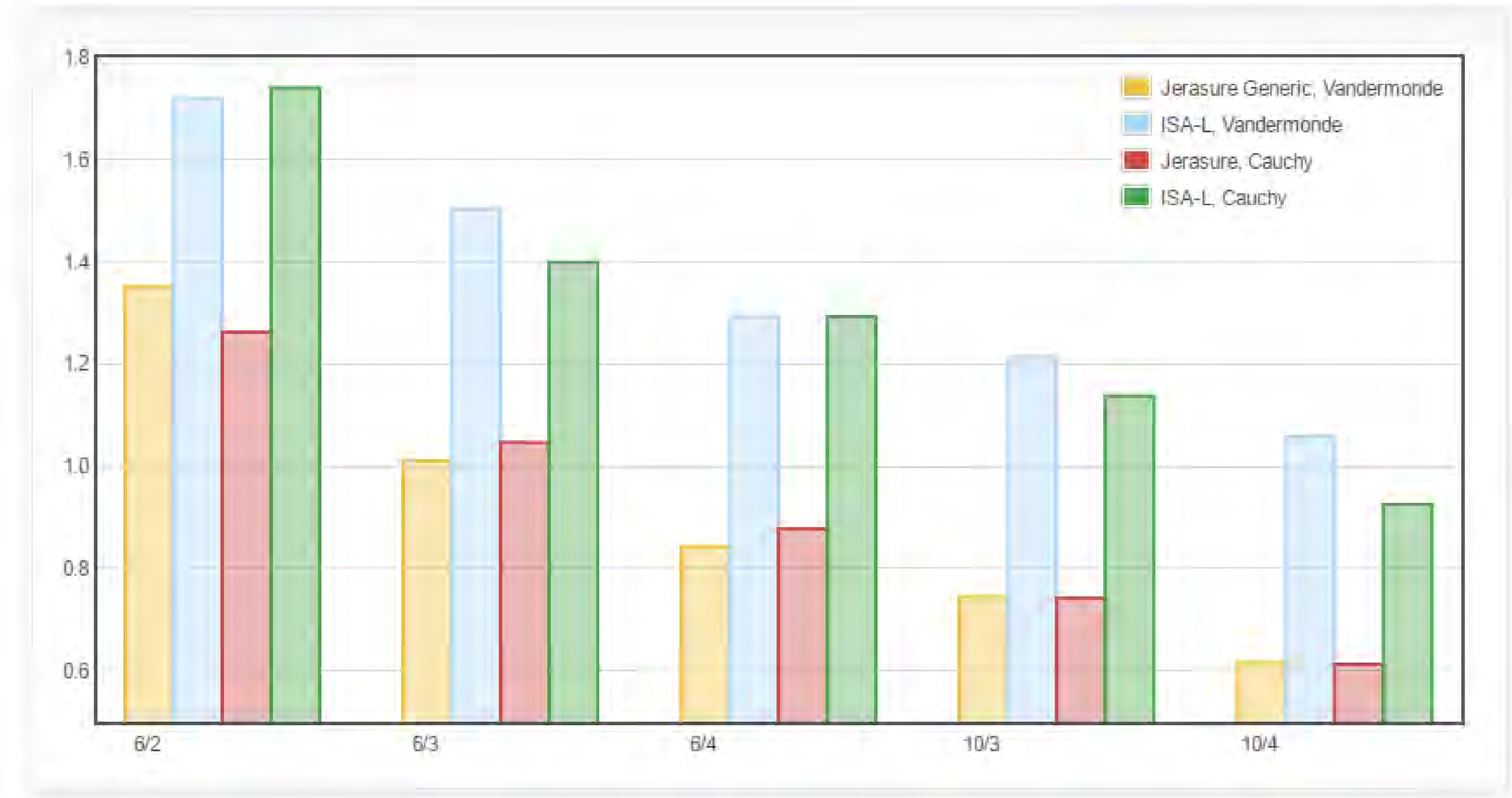
CPU Intensive

O(k*m) multiply-add operations

Credit: Sage Weil, Storage tiering and erasure coding in Ceph (SCaLE13x)

Ceph Erasure Coding Performance (Single OSD)

Encode Operation – Reed-Soloman Codes



encode: Y = GB/s, X = K/M

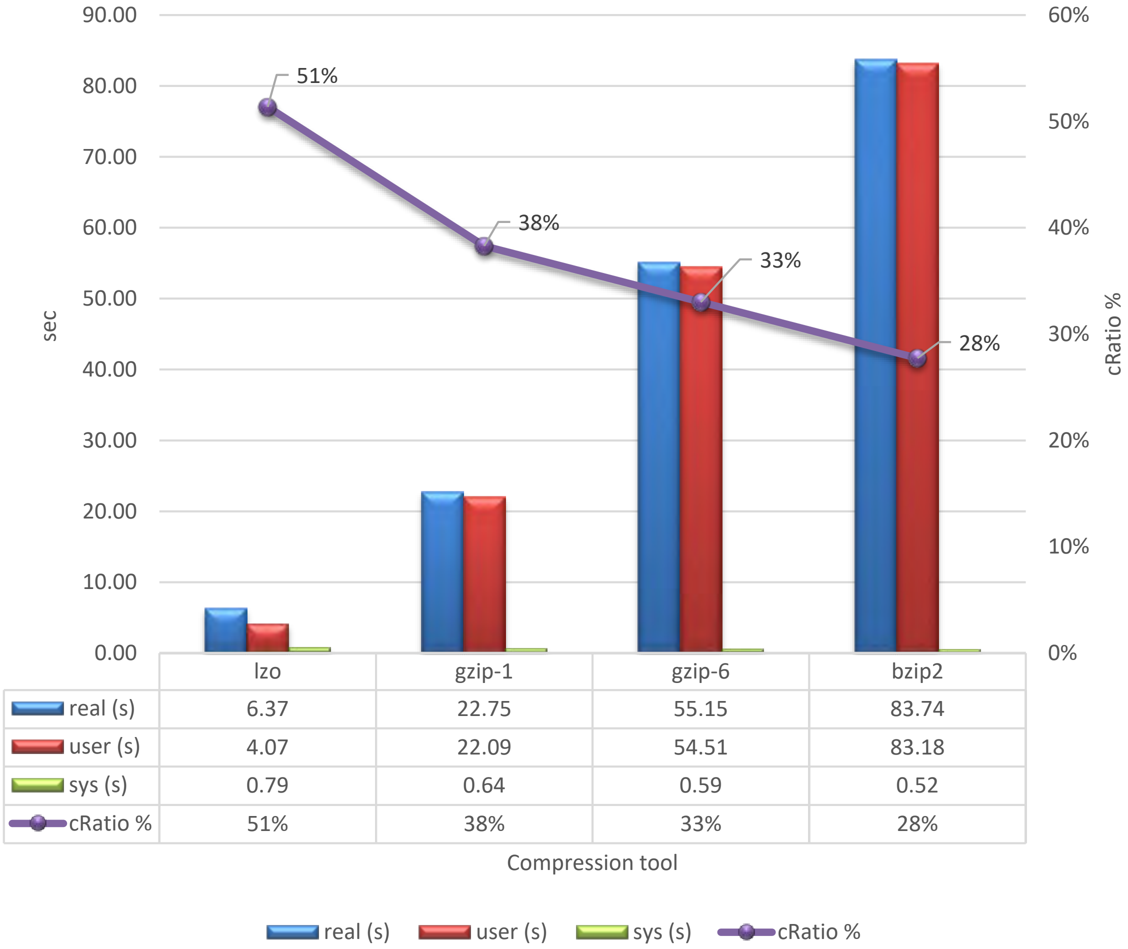
Source as of August 2016: Intel internal **measurements** with Ceph Jewel 10.2.x on dual E5-2699 v4 (22C, 2.3GHz, 145W), HT & Turbo Enabled, Fedora 22 64 bit, kernel 4.1.3, 2 x DH8955 adaptor, DDR4-128GB

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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. Any difference in system hardware or software design or configuration may affect actual performance. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance. **For more information go to http://www.intel.com/performance**

ISA-L Encode is up to 40% Faster than alternatives on Xeon-E5v4

Ceph compression and Intel® qat

Compression: Cost



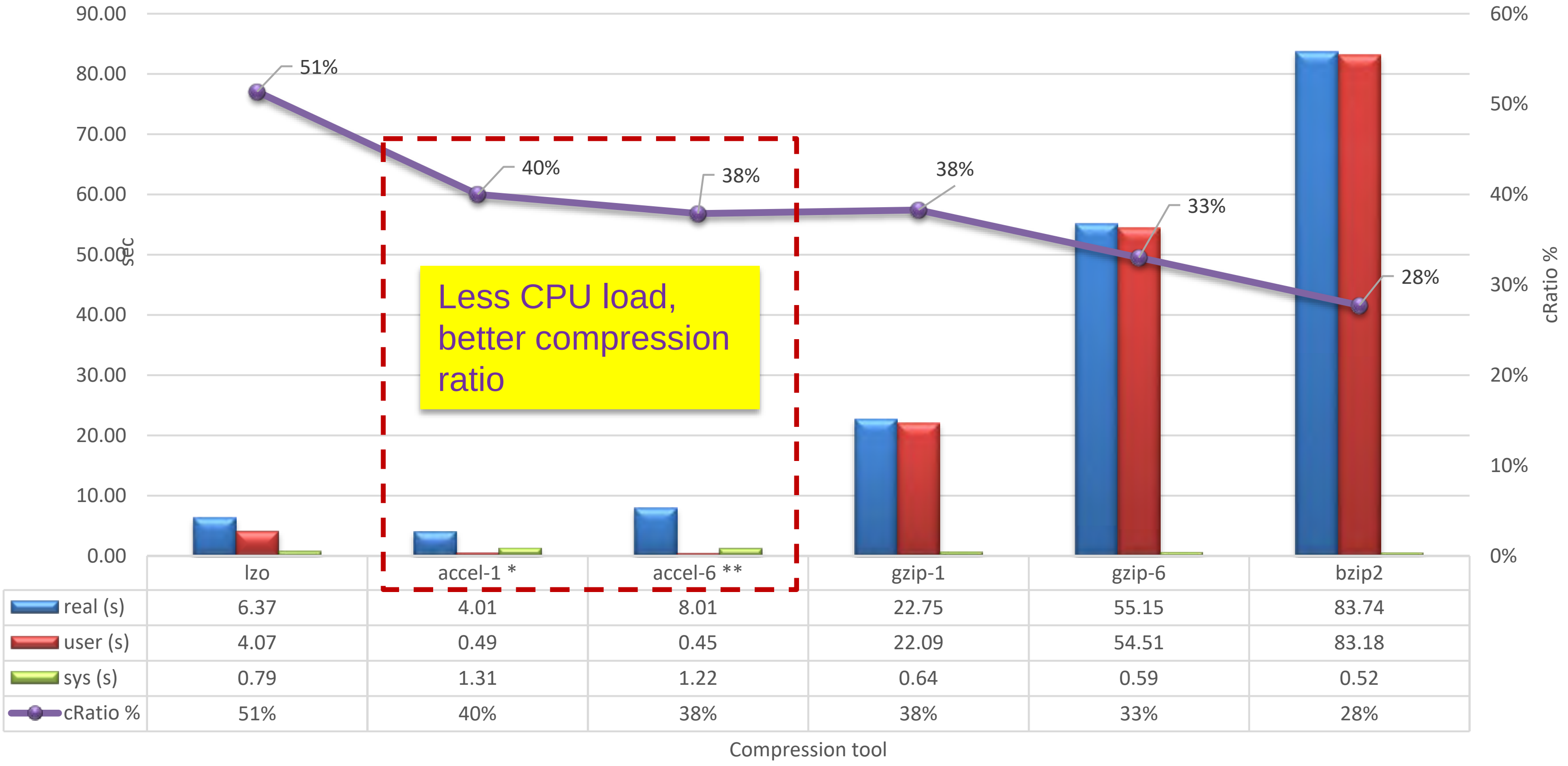
	lzo	gzip-1	gzip-6	bzip2
real (s)	6.37	22.75	55.15	83.74
user (s)	4.07	22.09	54.51	83.18
sys (s)	0.79	0.64	0.59	0.52
cRatio %	51%	38%	33%	28%

- Compress 1GB Calgary Corpus* file on one CPU core (HT).
- Compression ratio: less is better
  - *cRatio = compressed size / original size*
- CPU intensive, better compression ratio requires more CPU time.

Source as of August 2016: Intel internal **measurements** with dual E5-2699 v3 (18C, 2.3GHz, 145W), HT & Turbo Enabled, Fedora 22 64 bit, DDR4-128GB
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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. Any difference in system hardware or software design or configuration may affect actual performance. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance. **For more information go to http://www.intel.com/performance**

*The *Calgary corpus* is a collection of text and binary data files, commonly used for comparing data compression algorithms.

Benefit of Hardware Acceleration



	lzo	accel-1 *	accel-6 **	gzip-1	gzip-6	bzip2
real (s)	6.37	4.01	8.01	22.75	55.15	83.74
user (s)	4.07	0.49	0.45	22.09	54.51	83.18
sys (s)	0.79	1.31	1.22	0.64	0.59	0.52
cRatio %	51%	40%	38%	38%	33%	28%

Less CPU load, better compression ratio

Compress 1GB Calgary Corpus File

* Intel® QuickAssist Technology DH8955 level-1

** Intel® QuickAssist Technology DH8955 level-6

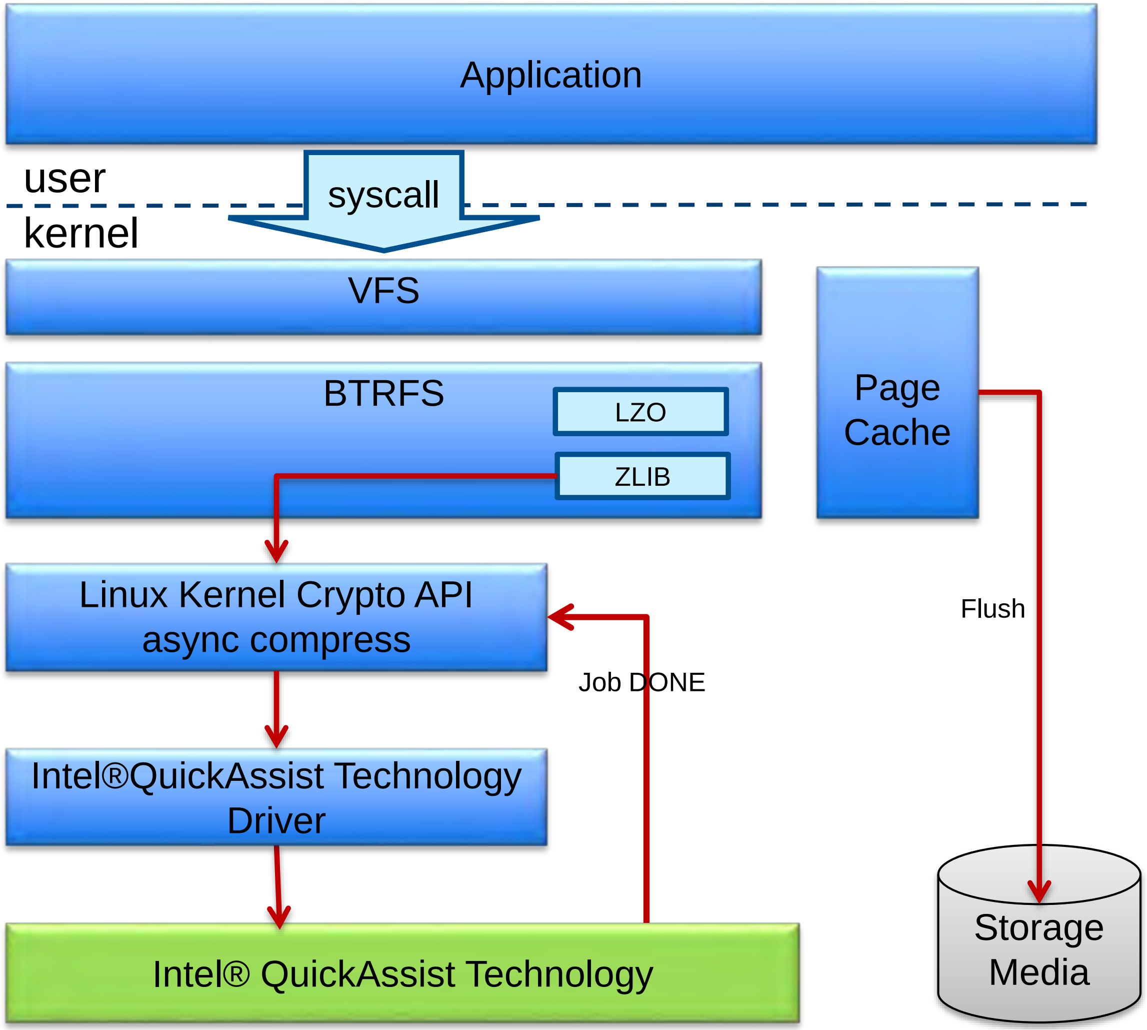
Source as of August 2016: Intel internal **measurements** with dual E5-2699 v3 (18C, 2.3GHz, 145W), HT & Turbo Enabled, Fedora 22 64 bit, 1 x DH8955 adaptor, DDR4-128GB
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Any change to 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. Any difference in system hardware or software design or configuration may affect actual performance. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance. **For more information go to http://www.intel.com/performance**

Transparent Compression in Ceph: BTRFS

- Copy on Write (CoW) filesystem for Linux
  - "Has the correct feature set and roadmap to serve Ceph in the long-term, and is recommended for testing, development, and any non-critical deployments… This compelling list of features makes btrfs the ideal choice for Ceph clusters"*
- Native compression support
  - ZLIB / LZO supported.
  - Compress up to 128KB each time
- Intel® QuickAssist Technology supports
  - DEFLATE: LZ77 compression followed by Huffman coding with GZIP or ZLIB header

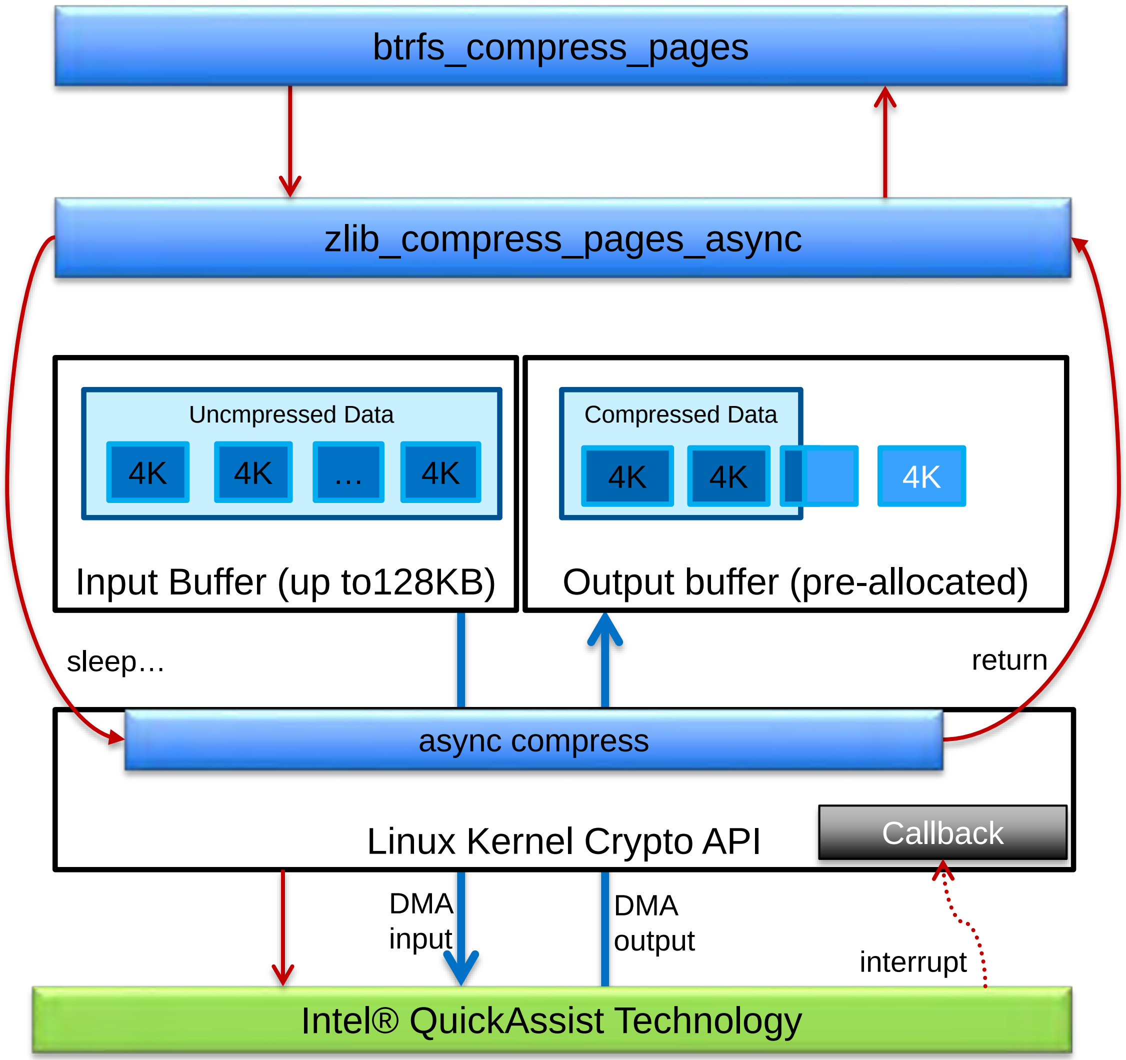
* http://docs.ceph.com/docs/hammer/rados/configuration/filesystem-recommendations/

Hardware Compression in BTRFS



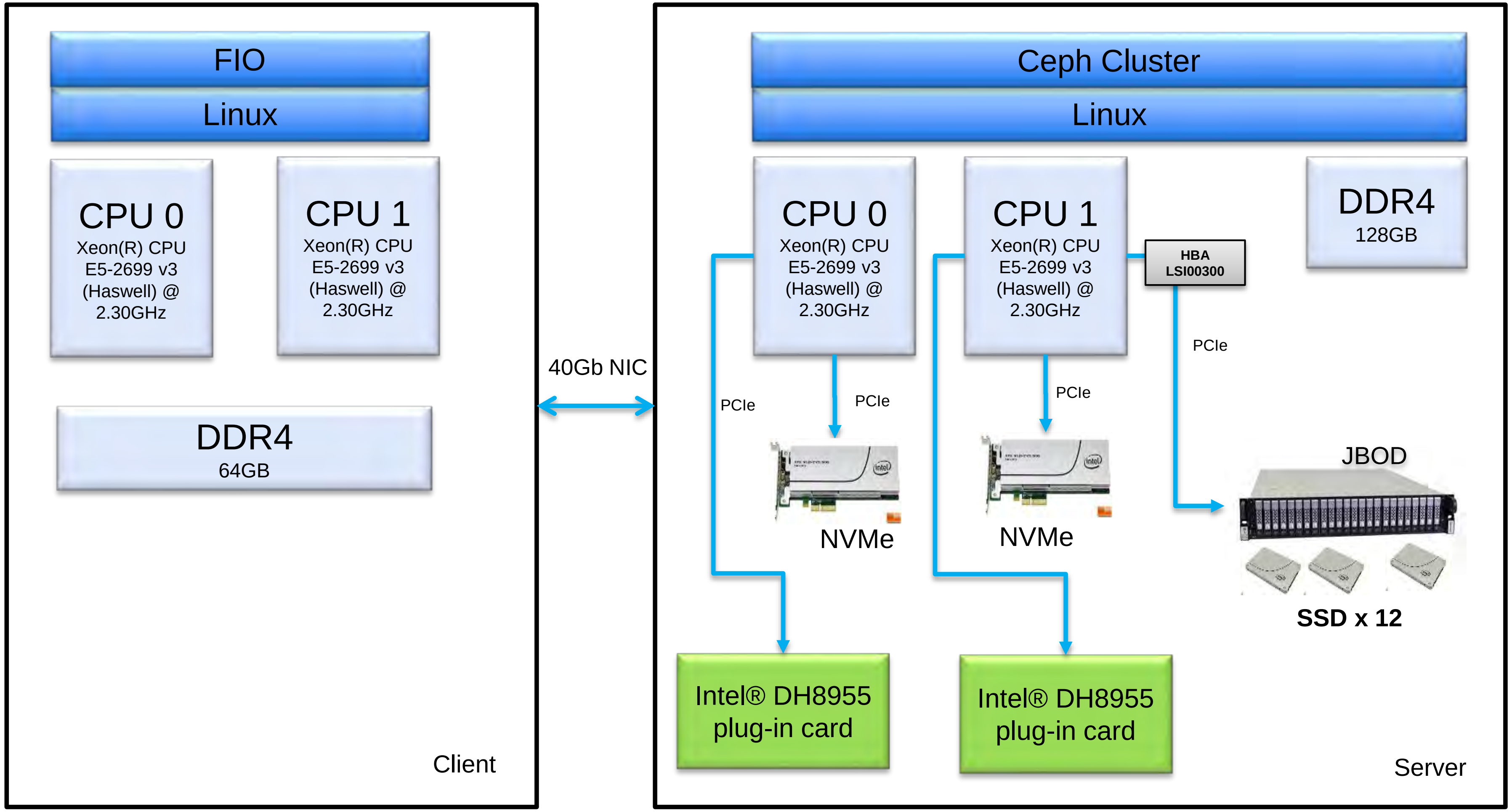
- BTRFS compress page buffers before writing to the storage media.
- LKCF selects hardware engine for compression.
- Data compressed by hardware can be de-compressed by software library, and vise versa.

Hardware Compression in BTRFS



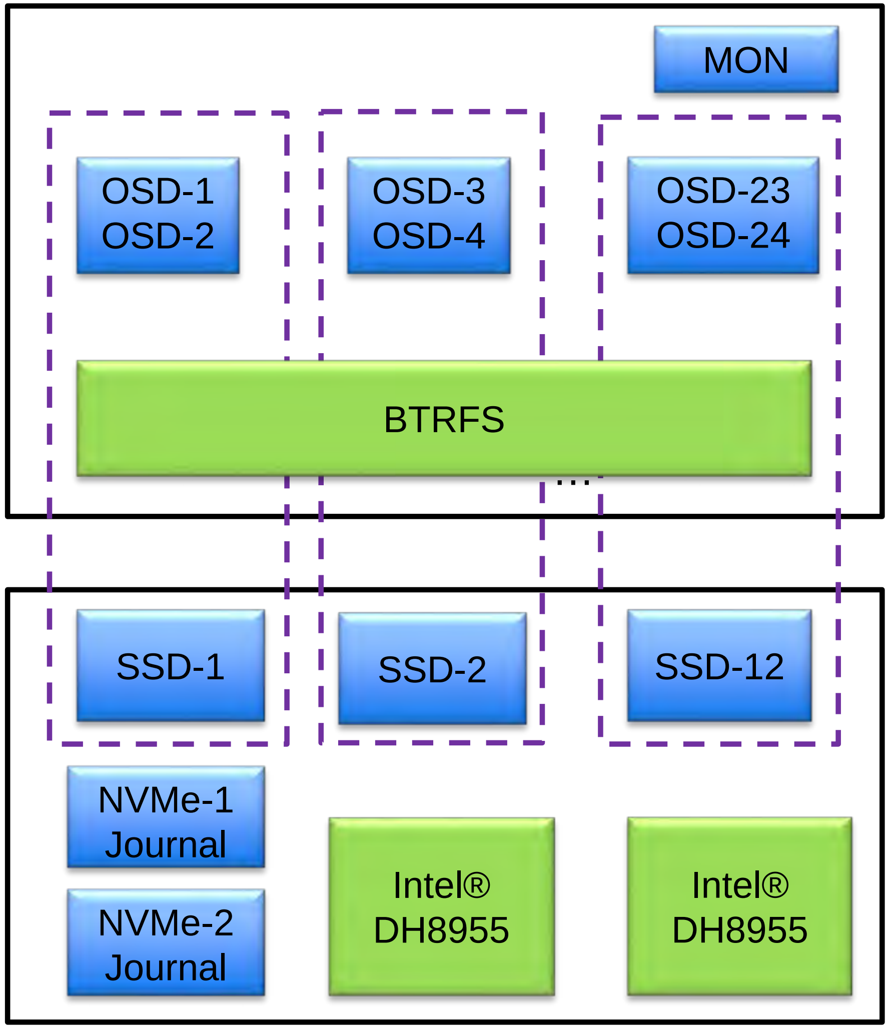
- BTRFS submits “async” compression job with sg-list containing up to 32 x 4K pages.
- BTRFS compression thread is put to sleep when the “async” compression API is called.
- BTRFS compression thread is woken up when hardware complete the compression job.
- Hardware can be fully utilized when multiple BTRFS compression threads run in-parallel.

Ceph, BTRFS, QAT Test Setup



FIO
Linux
CPU 0
Xeon(R) CPU E5-2699 v3 (Haswell) @ 2.30GHz
CPU 1
Xeon(R) CPU E5-2699 v3 (Haswell) @ 2.30GHz
DDR4
64GB
Client
40Gb NIC
Ceph Cluster
Linux
CPU 0
Xeon(R) CPU E5-2699 v3 (Haswell) @ 2.30GHz
CPU 1
Xeon(R) CPU E5-2699 v3 (Haswell) @ 2.30GHz
HBA
LSI00300
DDR4
128GB
PCIe
PCIe
PCIe
PCIe
NVMe
NVMe
JBOD
SSD x 12
Intel® DH8955 plug-in card
Intel® DH8955 plug-in card
Server

Benchmark - Ceph Configuration



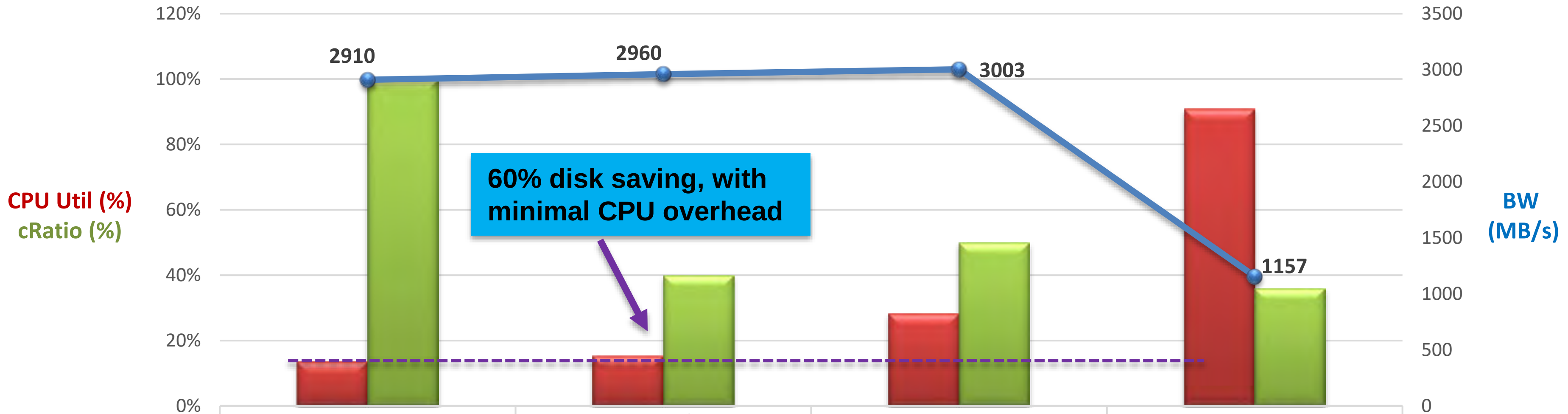
- BTRFS as Ceph Filestore backend
- 2 OSDs per SSD
- 2x NVMe for Ceph journals
- Data written to Ceph OSD is compressed by Intel® QuickAssist Technology (Intel® DH8955 PCIe Adapter)

Benchmark Configuration Details

Client	
CPU	2 x Intel® Xeon CPU E5-2699 v3 (Haswell) @ 2.30GHz (36-core 72-threads)
Memory	64GB
Network	40GbE, jumbo frame: MTU=8000
Test Tool	**FIO 2.1.2, engine=libaio, bs=64KB, 64 threads**

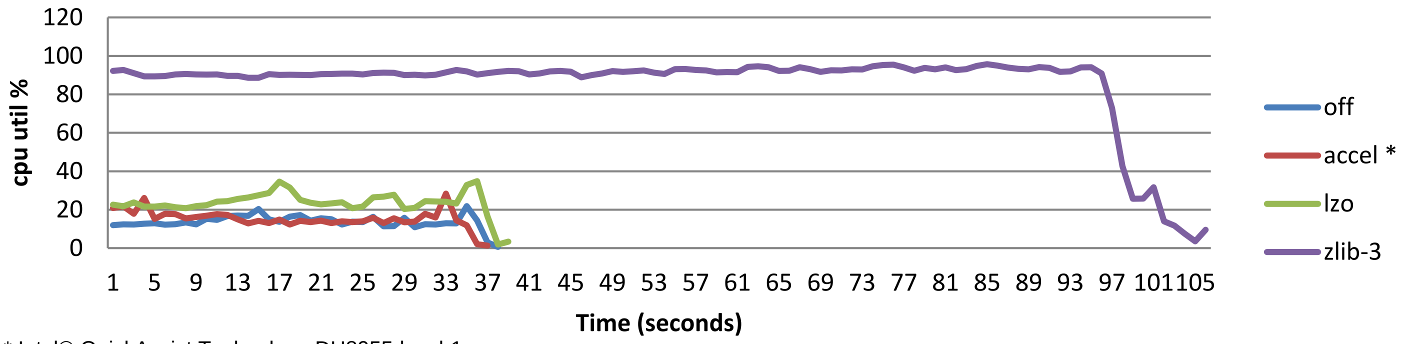
Ceph Cluster	
CPU	2 x Intel (R) Xeon CPU E5-2699 v3 (Haswell) @ 2.30GHz (36-core 72-threads)
Memory	128GB
Network	40GbE, jumbo frame: MTU=8000
HBA	HBA LSI00300
OS	Fedora 22 (Kernel 4.1.3)
OSD	24 x OSD, 2 on one SSD (S3700), no-replica 2 x NVMe (P3700) for journal 2400 PGs
Accelerator	**Intel® QuickAssist Technology, 2 x Intel® QuickAssist Adapters 8955** **Dynamic compression Level-1**
BTRFS ZLIB S/W	**ZLIB Level-3**

Sequential Write



60% disk saving, with minimal CPU overhead

	off	accel *	lzo	zlib-3
🟥 Cpu Util (%)	13.62%	15.25%	28.30%	90.95%
🟩 cRatio (%)	100%	40%	50%	36%
● Bandwidth(MB/s)	2910	2960	3003	1157

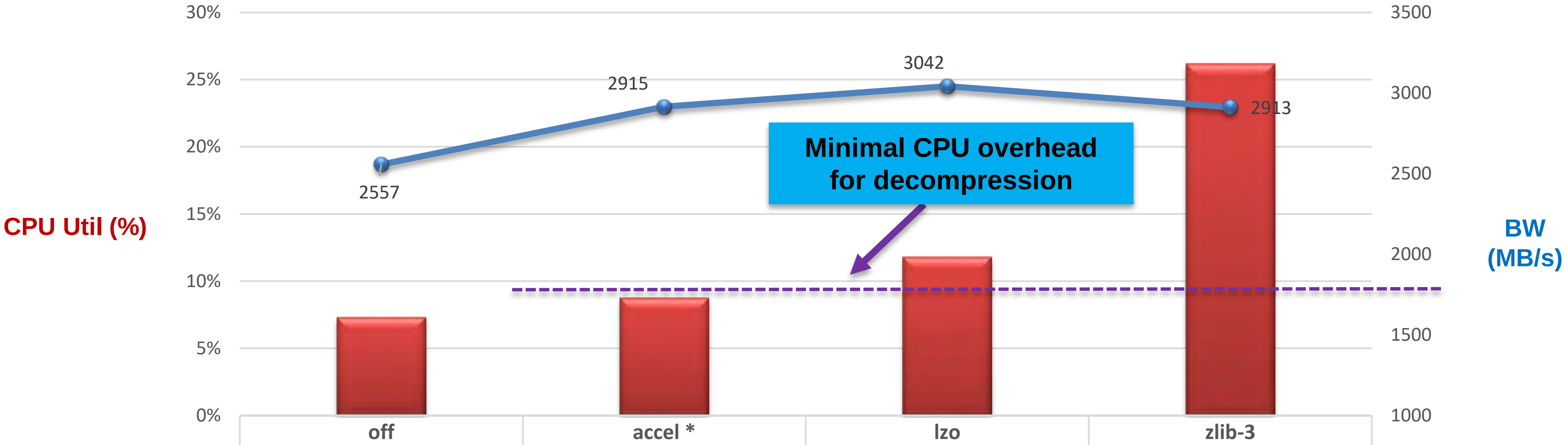


* Intel® QuickAssist Technology DH8955 level-1

** Dataset is random data generated by FIO

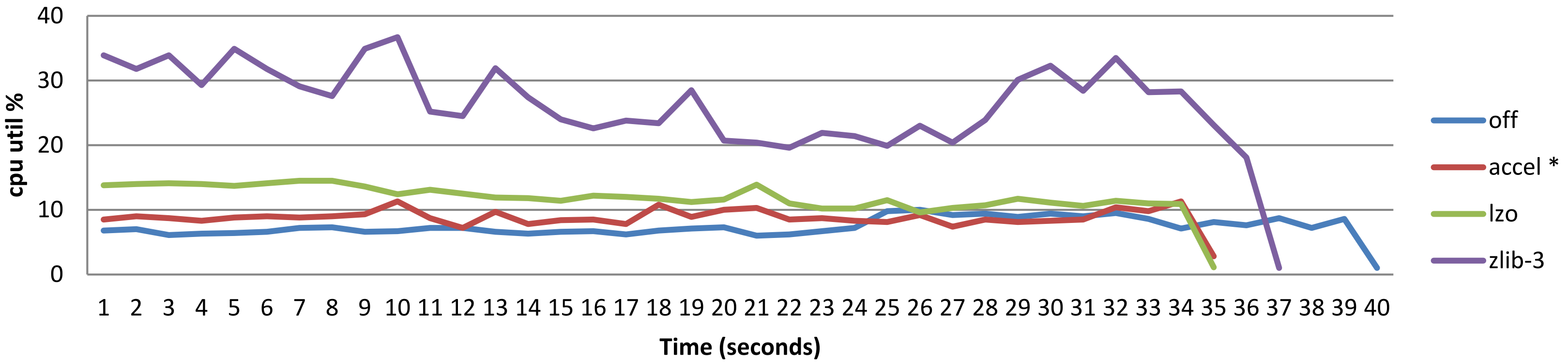
Source as of August 2016: Intel internal **measurements** with dual E5-2699 v3 (18C, 2.3GHz, 145W), HT & Turbo Enabled, Fedora 22 64 bit, kernel 4.1.3, 2 x DH8955 adaptor, DDR4-128GB Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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. Any difference in system hardware or software design or configuration may affect actual performance. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance. **For more information go to** **http://www.intel.com/performance**

Sequential Read



Minimal CPU overhead for decompression

	off	accel *	lzo	zlib-3
🟥 Cpu Util (%)	7.33%	8.76%	11.81%	26.20%
● Bandwidth(MB/s)	2557	2915	3042	2913



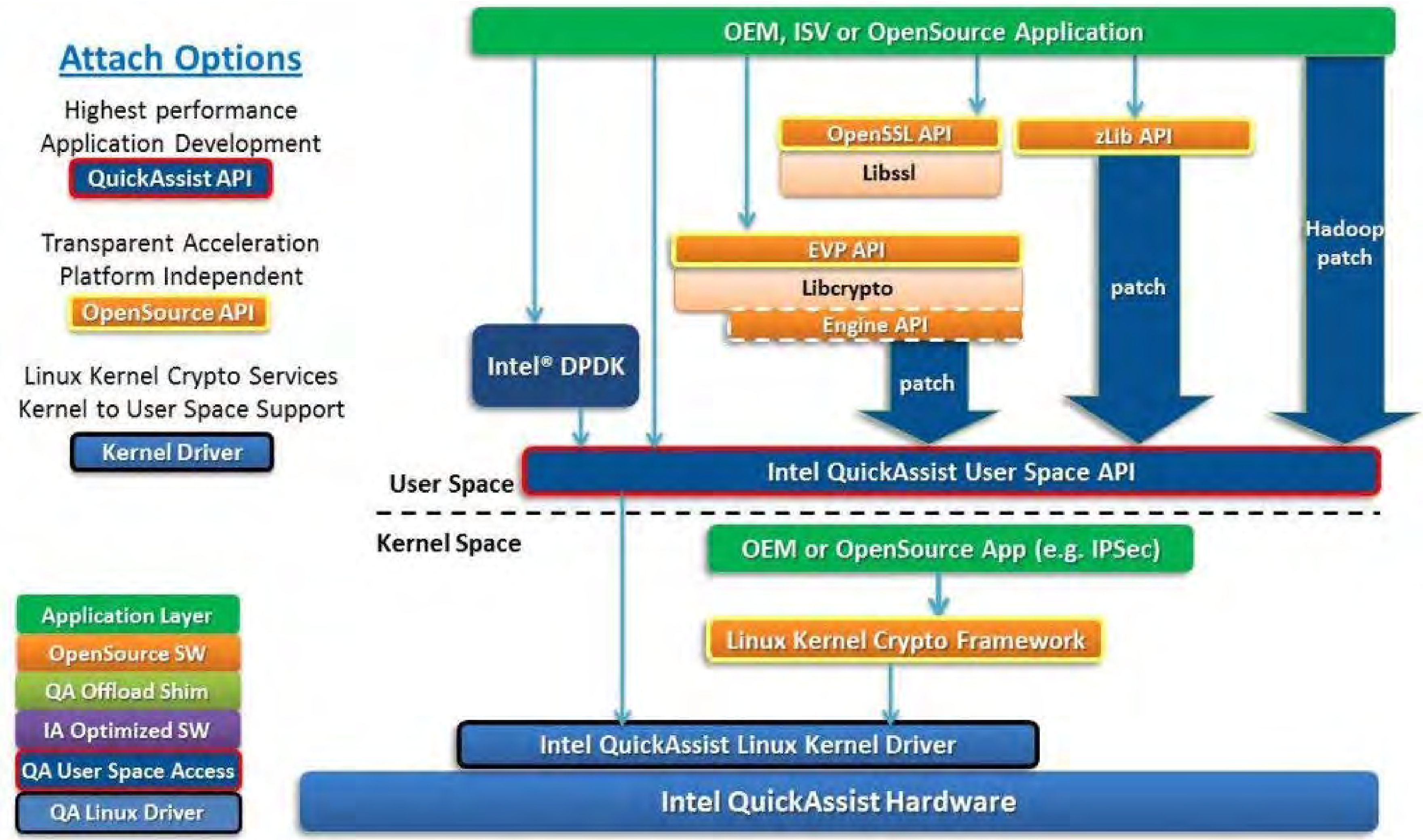
Source as of August 2016: Intel internal **measurements** with dual E5-2699 v3 (18C, 2.3GHz, 145W), HT & Turbo Enabled, Fedora 22 64 bit, kernel 4.1.3, 2 x DH8955 adaptor, DDR4-128GB Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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. Any difference in system hardware or software design or configuration may affect actual performance. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance. **For more information go to http://www.intel.com/performance**

* Intel® QuickAssist Technology DH8955 level-1

Additional Sources of Information

- For more information on Intel® QuickAssist Technology & Intel® QuickAssist Software Solutions can be found here:
  - Software Package and engine are available at 01.org: Intel QuickAssist Technology | 01.org
  - For more details on Intel® QuickAssist Technology visit: http://www.intel.com/quickassist
  - Intel Network Builders: https://networkbuilders.intel.com/ecosystem
- Intel®QuickAssist Technology Storage Testimonials
  - IBM v7000Z w/QuickAssist
    - http://www-03.ibm.com/systems/storage/disk/storwize_v7000/overview.html
    - https://builders.intel.com/docs/networkbuilders/Accelerating-data-economics-IBM-flashSystem-and-Intel-quick-assist-technology.pdf
- Intel’s QuickAssist Adapter for Servers: http://ark.intel.com/products/79483/Intel-QuickAssist-Adapter-8950
- DEFLATE Compressed Data Format Specification version 1.3 http://tools.ietf.org/html/rfc1951
- BTRFS: https://btrfs.wiki.kernel.org
- Ceph: http://ceph.com

QAT Attach Options



intel®