Hyper Converged Cache For Cloud Storage

02/2017

Agenda

- **Introduction**
- Hyper Converged Cache
- Hyper Converged Cache Architecture
  - Overview
  - Design details
  - Performance overview
  - Current progress and roadmap
- Hyper Converged Cache with Optane technology
- Summary

Introduction

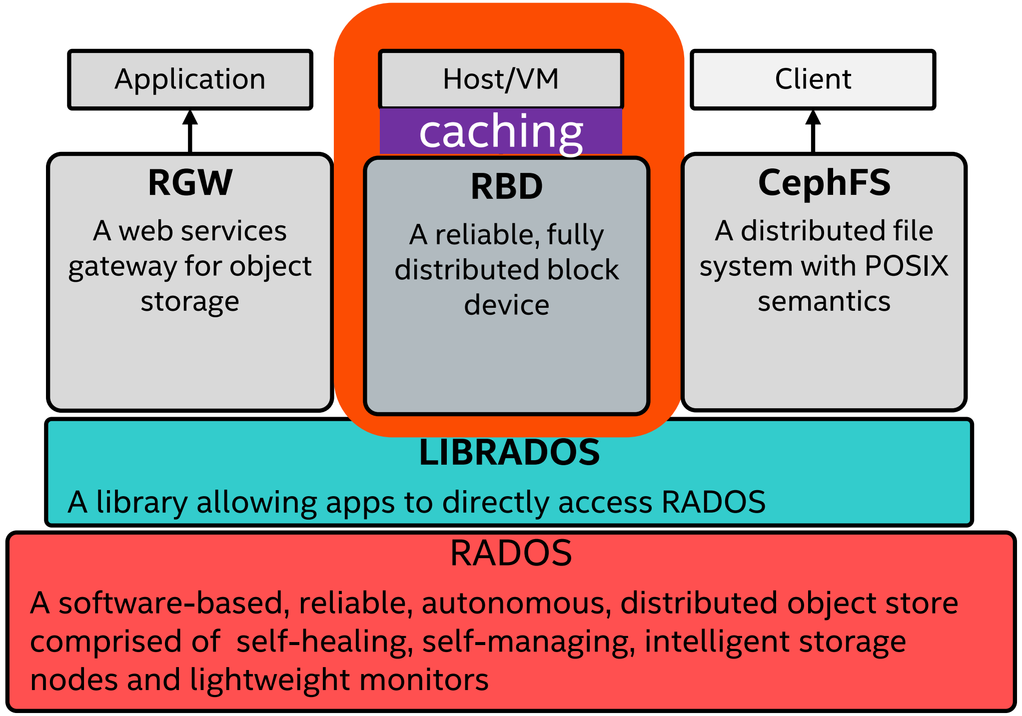
- Intel Cloud computing and Big Data Engineering Team
- Open source @ Spark, Hadoop, OpenStack, Ceph, NoSQL etc.
- Working with community and end customers closely
- Technology and Innovation oriented
  - Real-time, in-memory, complex analytics
  - Structure and unstructured data
  - Agility, Multi-tenancy, Scalability and elasticity
  - Bridging advanced research and real-world applications

Agenda

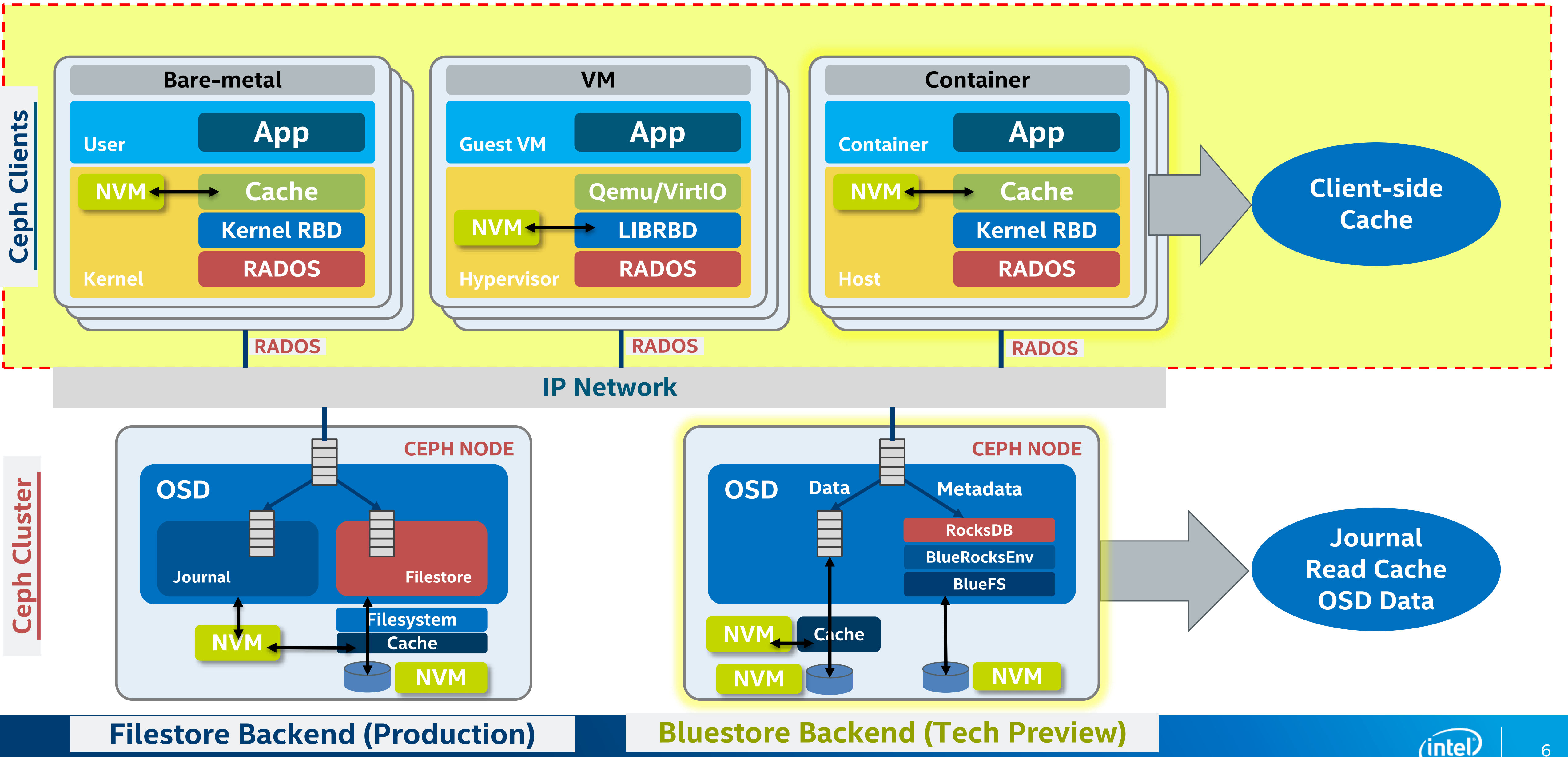
- Introduction
- **Hyper Converged Cache**
- Hyper Converged Cache Architecture
  - Overview
  - Design details
  - Performance overview
  - Current progress and roadmap
- Hyper Converged Cache with Optane technology
- Summary

Hyper Converged Cache

- A strong demands for SSD caching in Ceph cluster
- Ceph SSD caching performance has gaps
  - Journal, Cache tiering, Flashcache/bCache not work well
- Long tail latency is big issue for workloads such as OLTP
  - Need a caching layer to reduce IO path dependency on the network

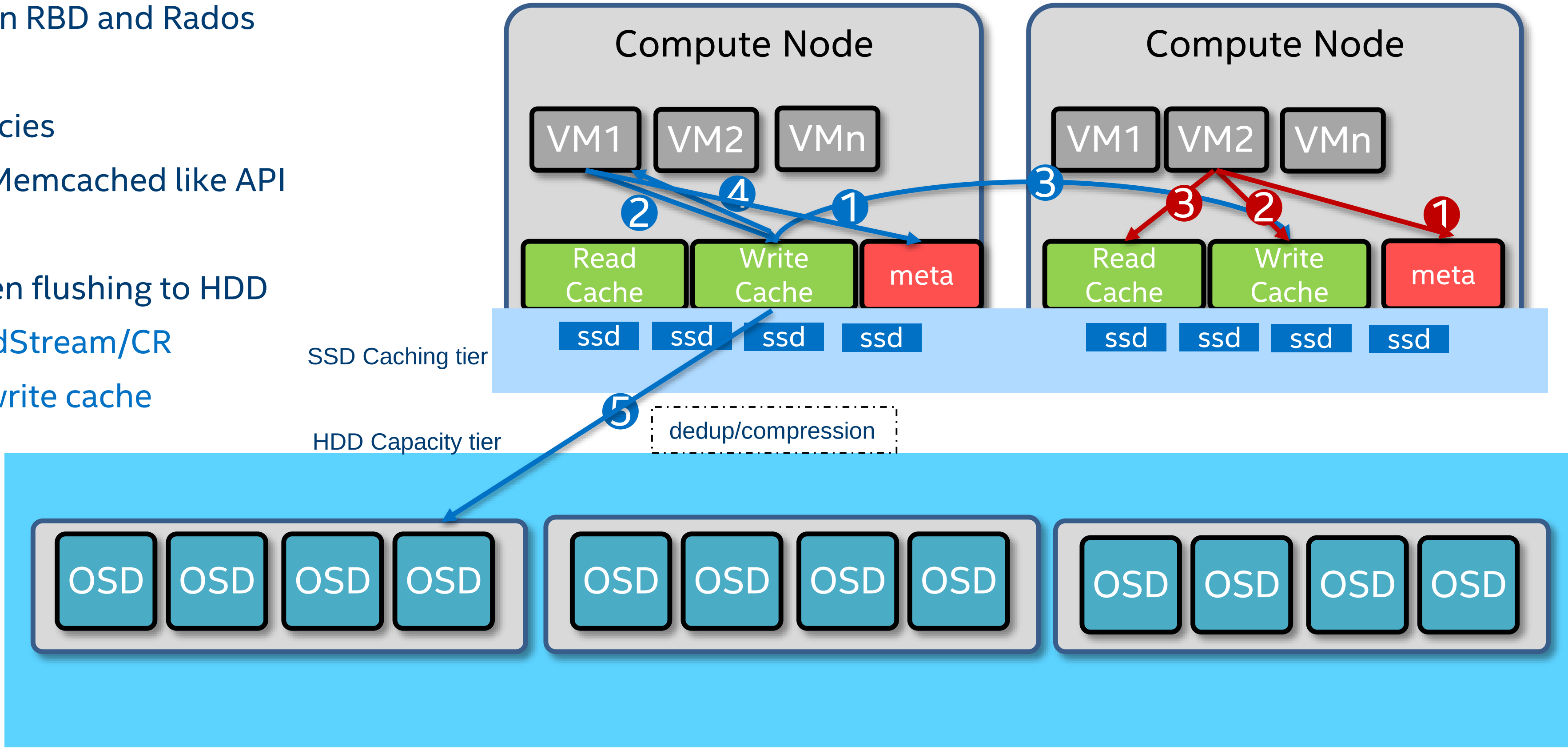


Ceph caching solutions on SSDs



Hyper Converged Cache Overview

- Client Side cache: caching on compute node
  - Local read cache and distributed write cache
  - Independent cache layer between RBD and Rados
- Extensible Framework
  - Pluggable design/cache policies
  - General caching interfaces: Memcached like API
- Data Services
  - Deduplication, Compression when flushing to HDD
- Value add feature designed for ColdStream/CR
  - Log-structure object store for write cache

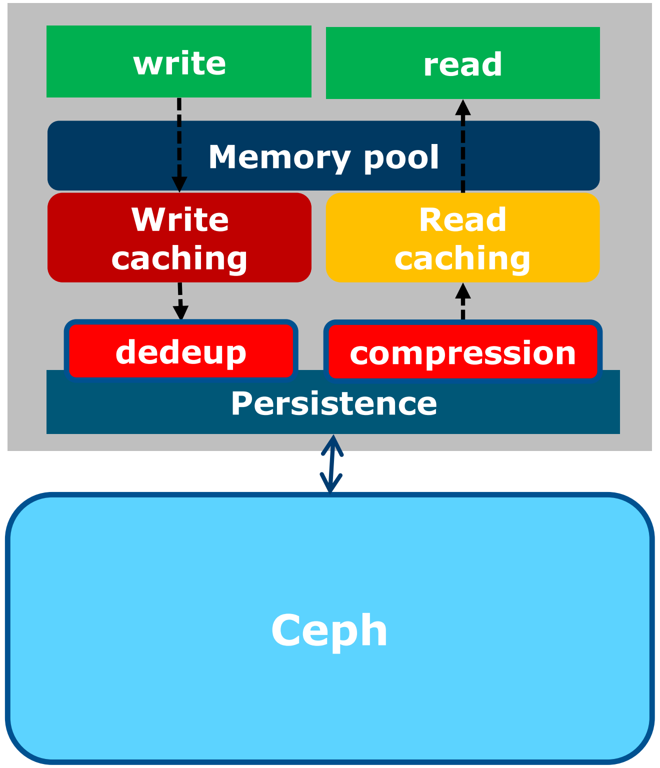


Agenda

- Introduction
- Hyper Converged Cache
- **Hyper Converged Cache Architecture**
  - Overview
  - Design details
  - Performance overview
  - Current progress and roadmap
- Hyper Converged Cache with Optane technology
- Summary

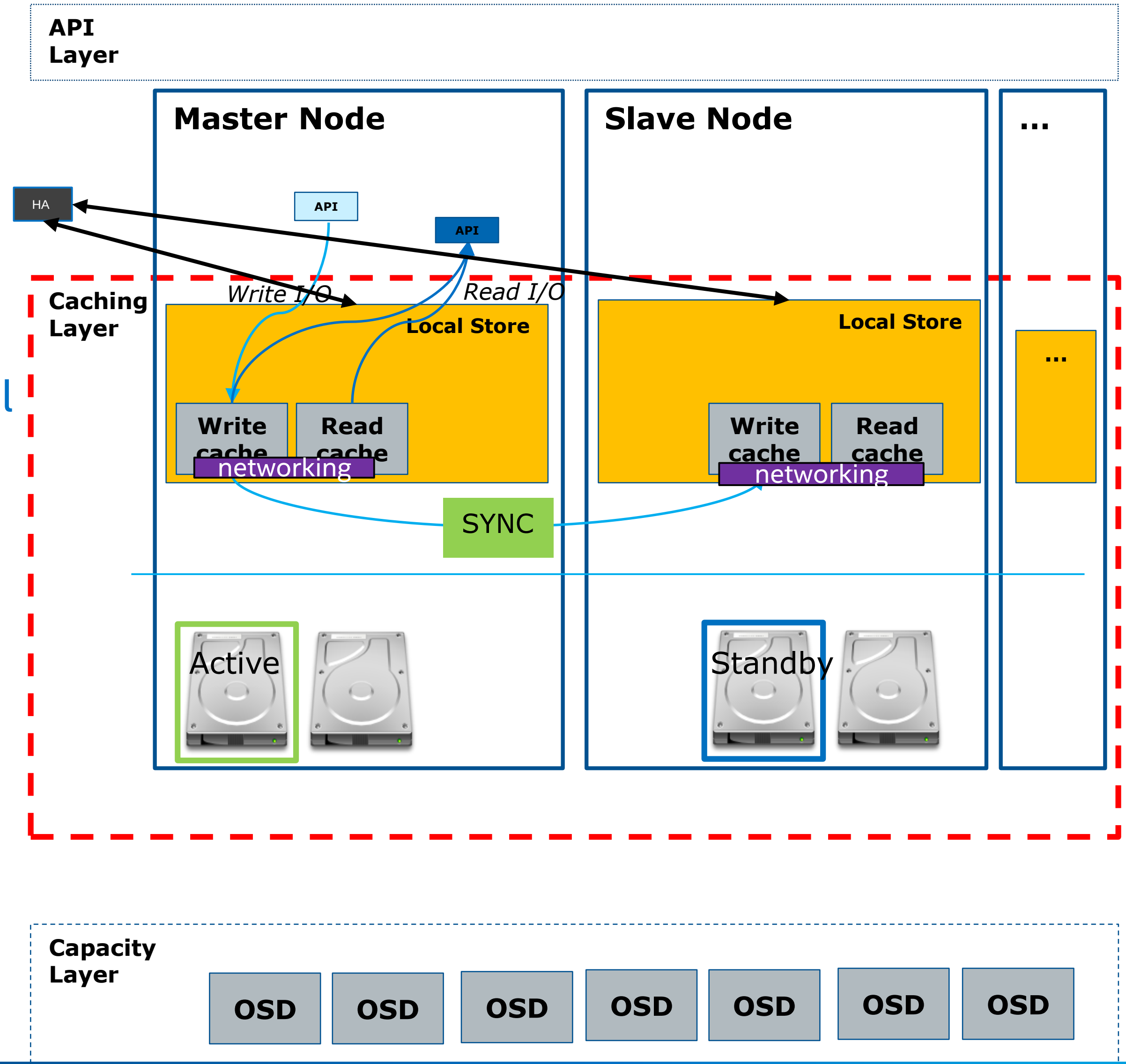
Hyper Converged Cache: General architecture

- Building a hyper-converged cache solutions for the cloud
  - Started with Ceph*
  - Block cache, object cache, file cache
  - Replication architecture
- Extensible Framework
  - Pluggable design/cache policies
  - Support third-party caching software
- Advanced data services:
  - Compression, deduplication, QOS
- Value added feature for future SCM device



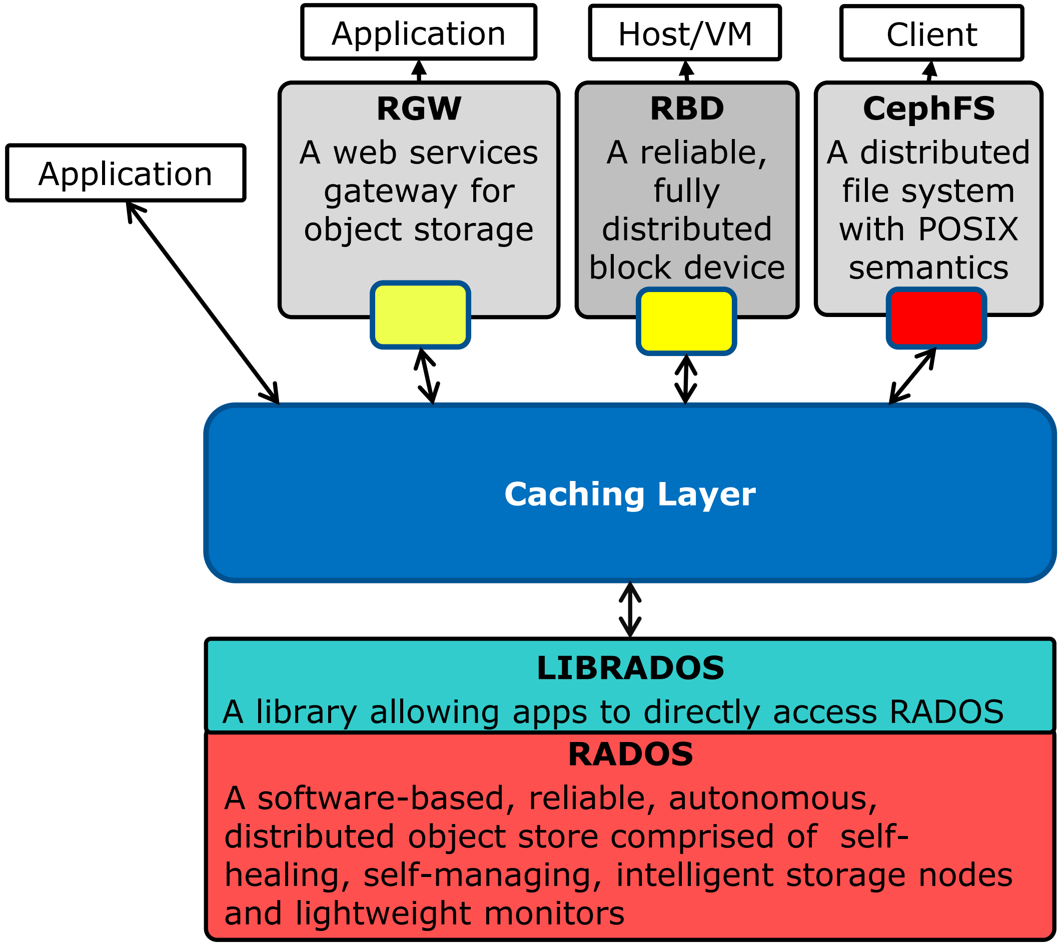
Hyper Converged Cache: Design details

- Generic interfaces:
  - **RBD**, RGW and Cephfs
- Master/Slave architecture:
  - Two hosts are required in order to provide physical redundancy
- Advanced service: dedup, compression, QoS, optimized with caching semantics



Hyper Converged Cache: API layer

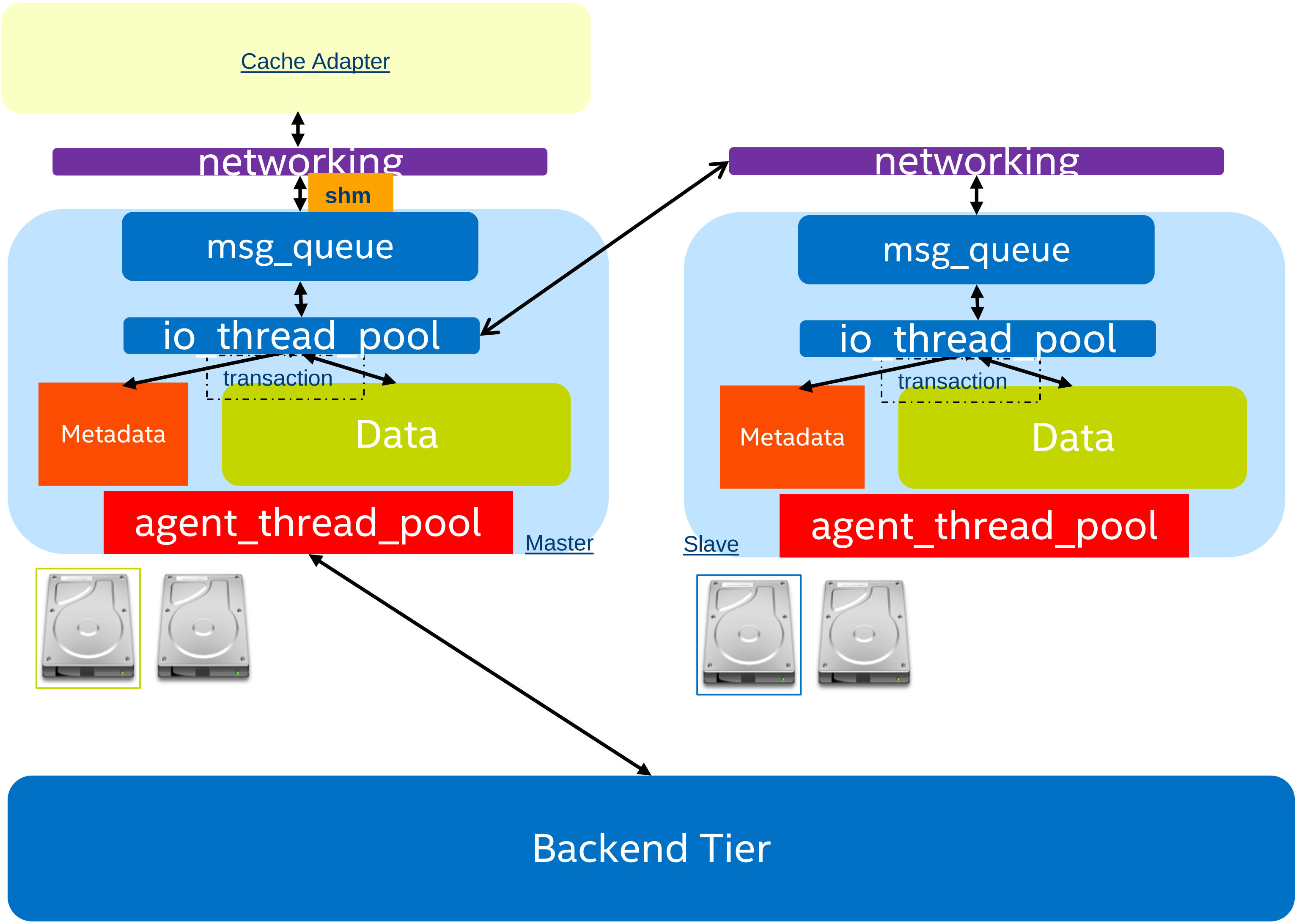
- **RBD**:
  - Hooks on librbd
  - caching for small writes
- **RGW**:
  - Caching over http
  - For metadata and small data
- **CephFS**:
  - Extend POSIX API
  - Caching for metadata and small writes



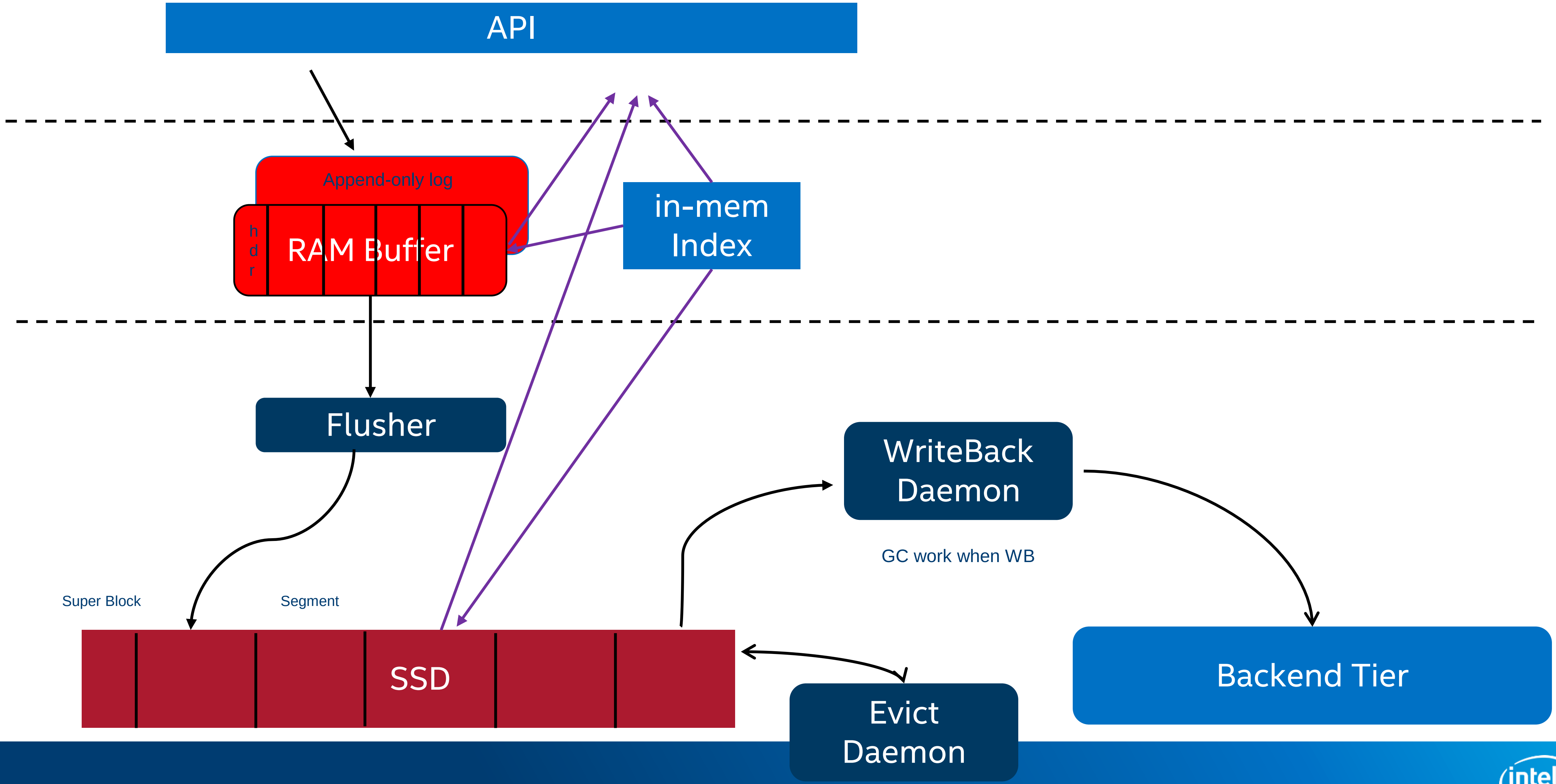
Hyper Converged Cache: Master/Slave replication

Master/Slave architecture:

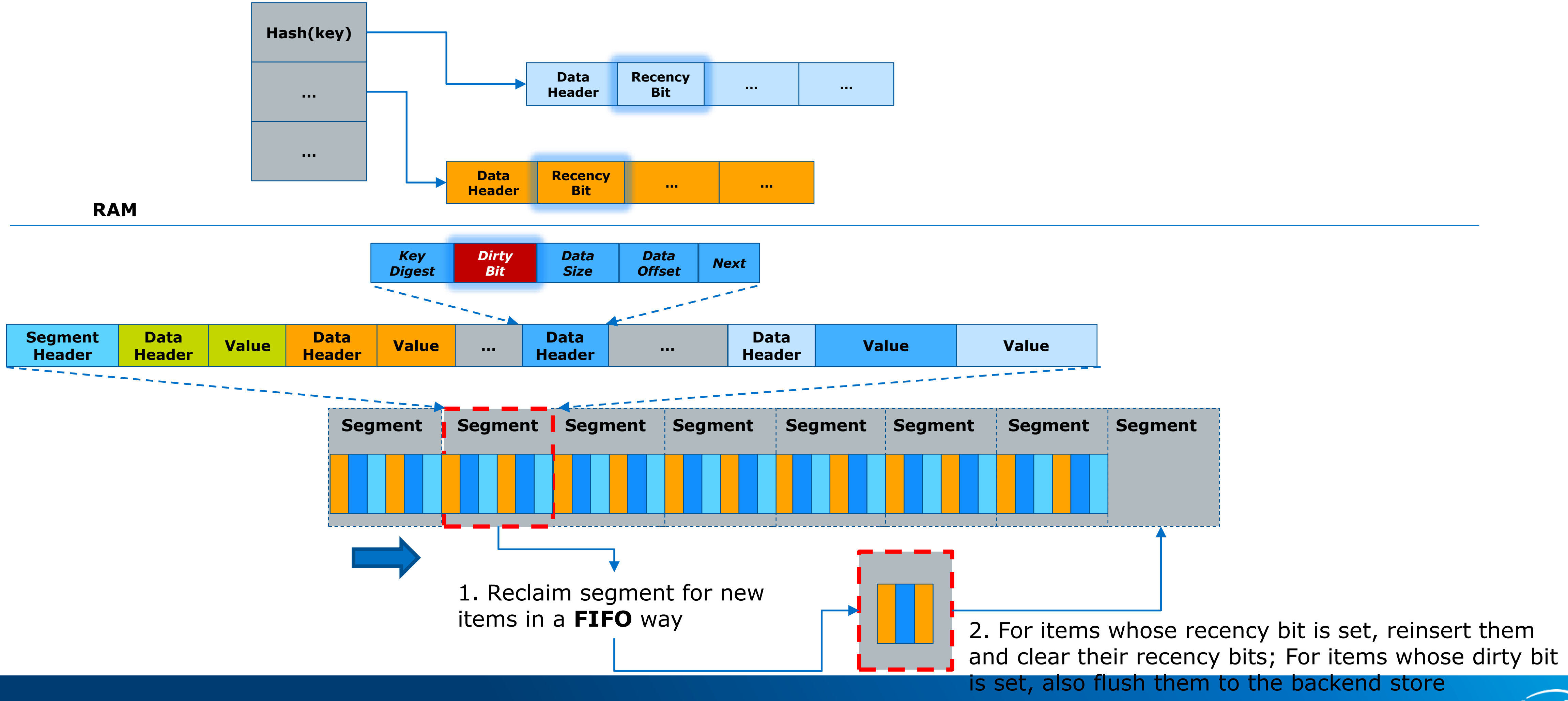
- Each host will have two process
  - Master: accept local read/writes and replicates to slave
  - Slave: accept replication writes
- Configurable master/slave pair
  - Specified in static configuration file(dynamic configuration will be in HA service later)
- Adapter sends read to master only
- Adapter sends write to master, then master replicates to slave
  - Local messaging will use shared memory
  - Client ACK on two writes finish



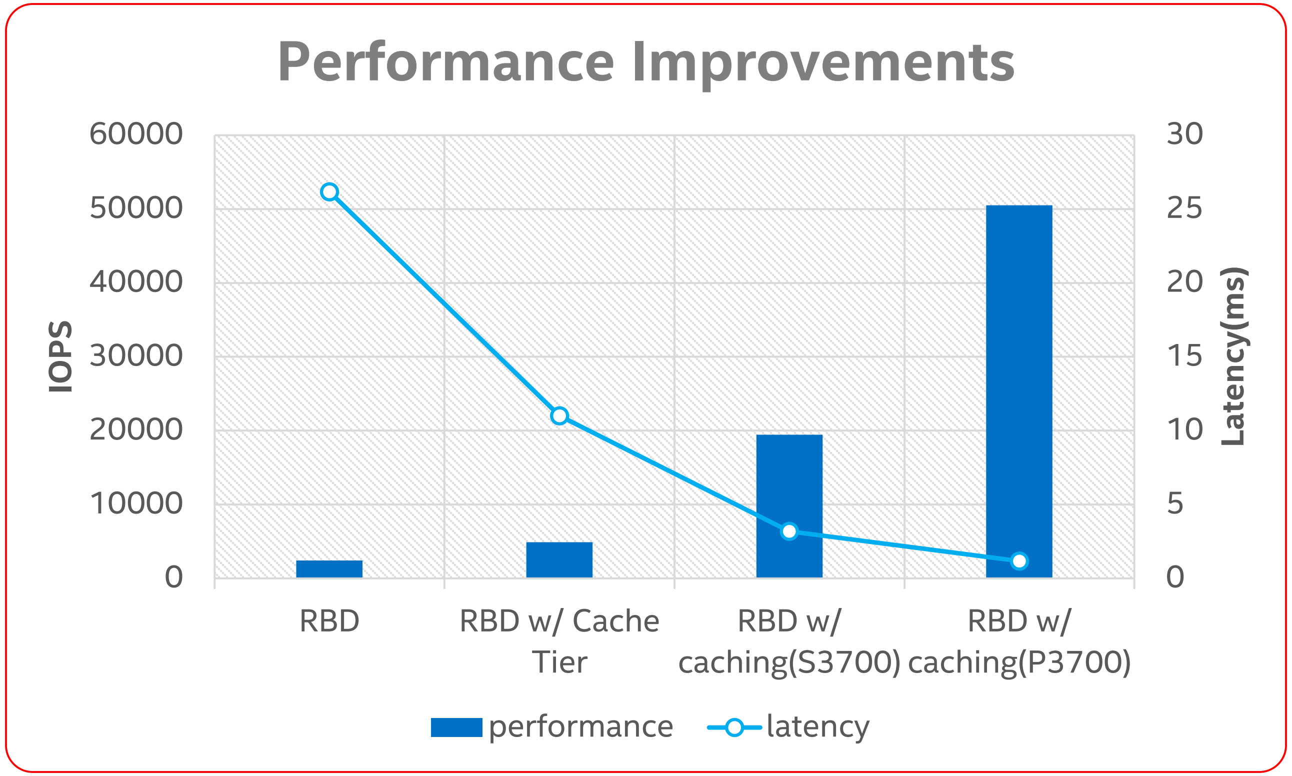
Hyper Converged Cache: Storage backend



Hyper Converged Cache: Storage backend With Caching Semantic

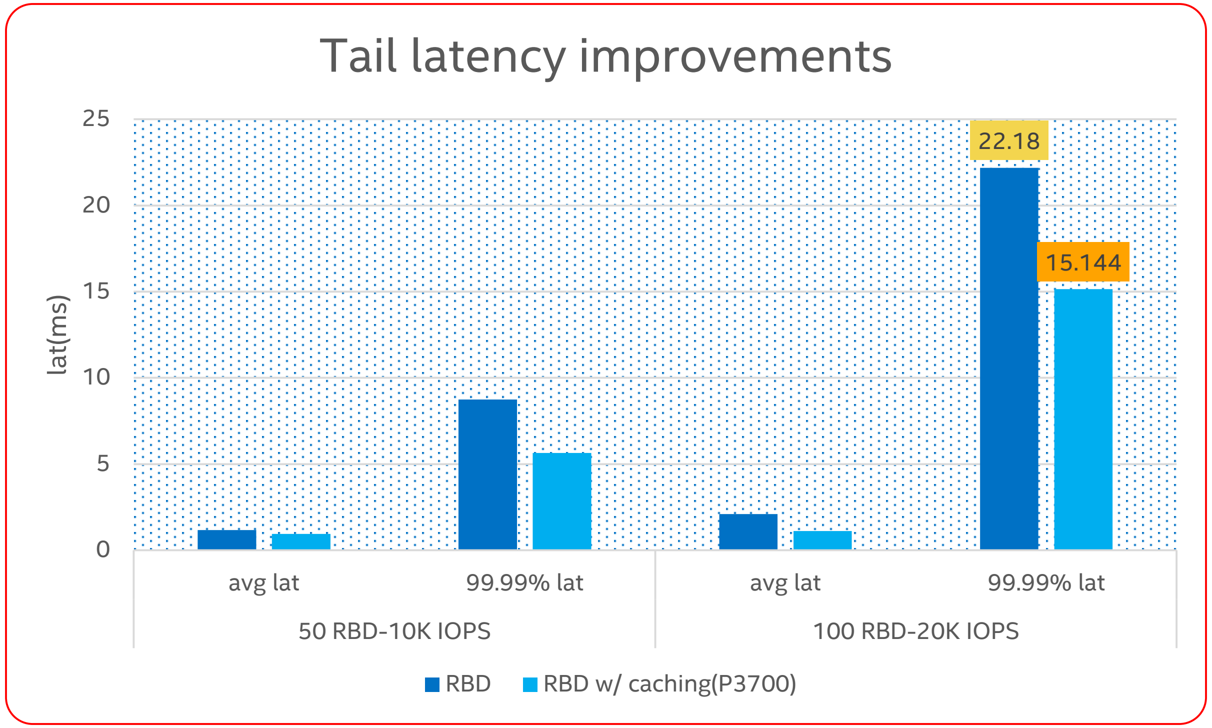


Hyper Converged Cache: Performance



- Hyper converged cache is able to provide ~7x performance improvements w/ zipf 4k randwrite, the latency also decreased ~92%.
  - With NVMe disk caching, the performance improved like 20x.
- Comparing with cache tier, the performance improved ~5x, the code path is much simpler.

Hyper Converged Cache: Tail Latency



- With SSD caching, hyper converged cache is able to reduce **~30%** tail latency under specified load.
  - Much easier to control and meet QOS/SLA requirements.

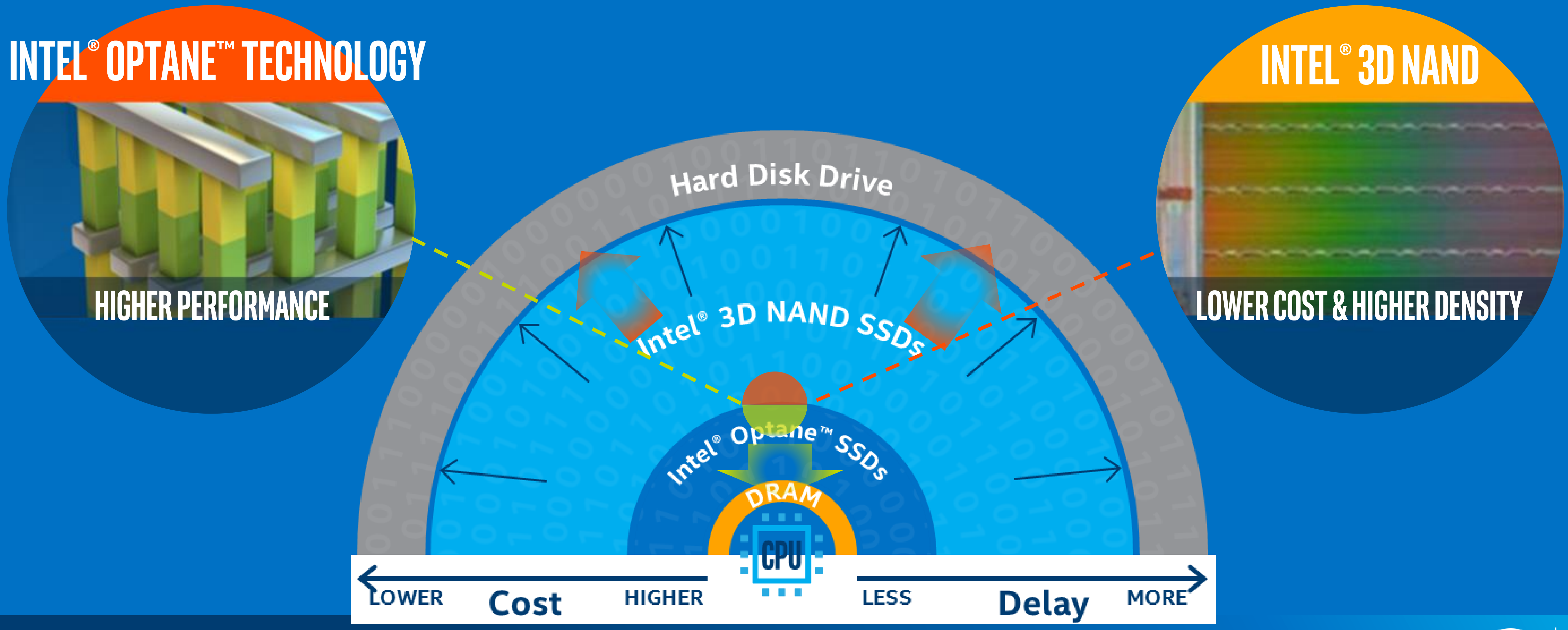
Agenda

- Introduction
- Hyper Converged Cache
- Hyper Converged Cache Architecture
  - Overview
  - Design details
  - Performance overview
  - Current progress and roadmap
- **Hyper Converged Cache with Optane technology**
- Summary

Upstream status and Roadmap

- Upstream BluePrint: **CRASH-CONSISTENT ORDERED WRITE-BACK CACHING EXTENSION**
  - A new librbd read cache to support **LBA-based** caching with DRAM/*non-volatile* storage backends
  - An **ordered write-back** cache that maintains checkpoints internally (or is structured as a data journal), such that writes that get flushed back to the cluster are always **crash consistent**. Even if one were to lose the client cache entirely, the disk image is still holding a valid file system that looks like it is just a little bit stale [1]. Should have durability characteristics similar to async replication if done right.
  - External **caching plug-in** interface – kernel and usermode
- Internal POC code done 80%. Will open source the project soon.

Intel investment: Two technologies



INTEL® OPTANE™ TECHNOLOGY

Size and Latency Specification Comparison

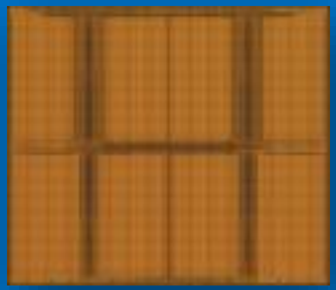
HDD
Latency: ~10 MillionX
Size of Data: ~10,000X

MEMORY

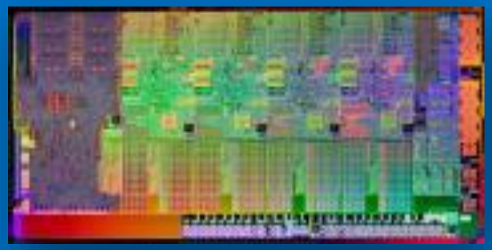
NAND SSD
Latency: ~100,000X
Size of Data: ~1,000X

Intel® Optane™ Technology
Latency: ~100X
Size of Data: ~1,000X

DRAM
Latency: ~10X
Size of Data: ~100X



SRAM
Latency: 1X
Size of Data: 1X

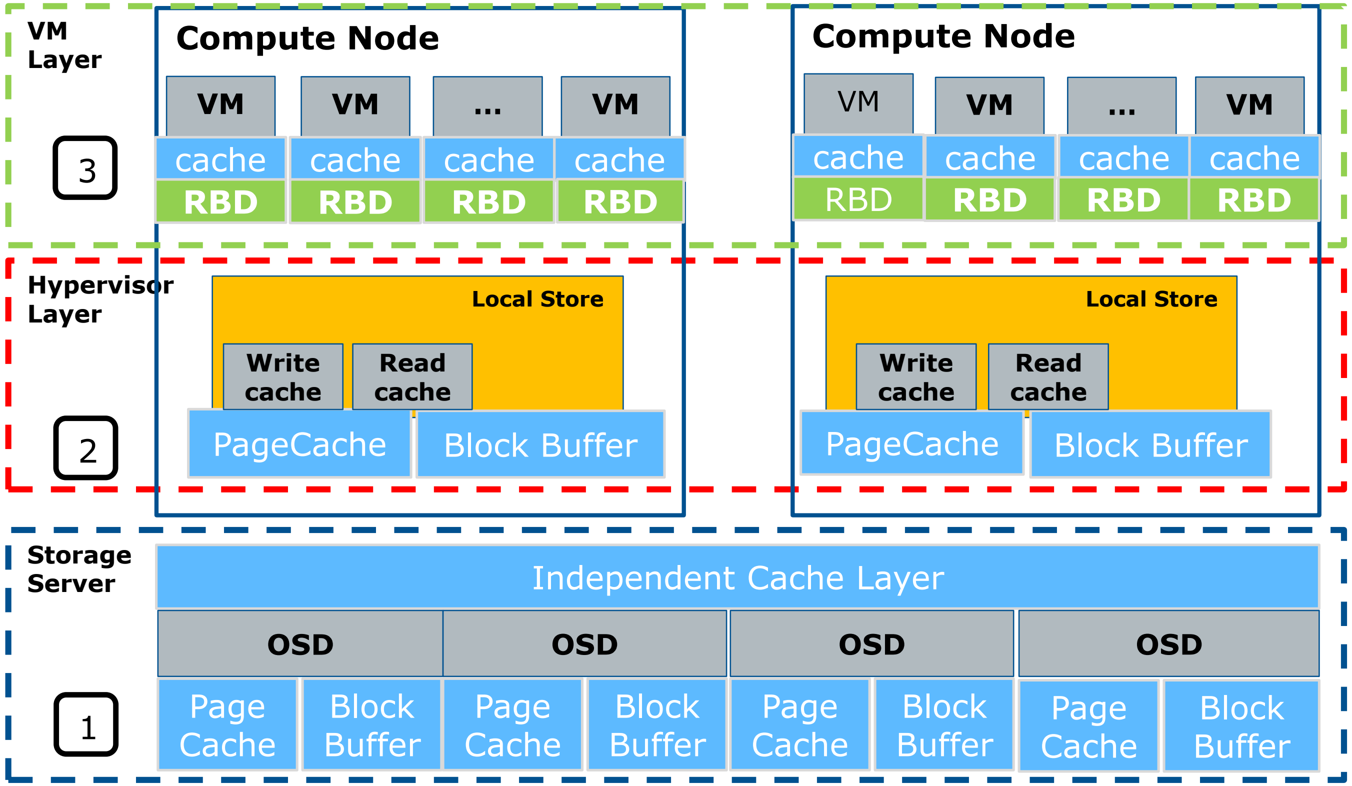


STORAGE

Technology claims are based on comparisons of latency, density and write cycling metrics amongst memory technologies recorded on published specifications of in-market memory products against internal Intel specifications.

Lower latency is faster

Hyper Converged Cache: caching on Optane?



1. Using Intel® Optane™ device as block buffer cache device.
2. Using Intel® Optane™ device as page caching device.
3. Using 3D XPointTM device as OS L2 memory?

Summary

- With client-side SSD caching, RBD randwrite improved ~5x, the latency also decreased ~92%.
- With client-side SSD caching, the long tail latency could be improved a lot.
- With the emerging new media like Optane, the caching benefit will be more higher
- Next step:
  - Finish the coding work(80% done) and open source the project
  - Tests on objects and filesystem

Q&A

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

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.

Intel, Xeon and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

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

Legal Information: Benchmark and Performance Claims 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.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase.

Test and System Configurations: See Back up for details.

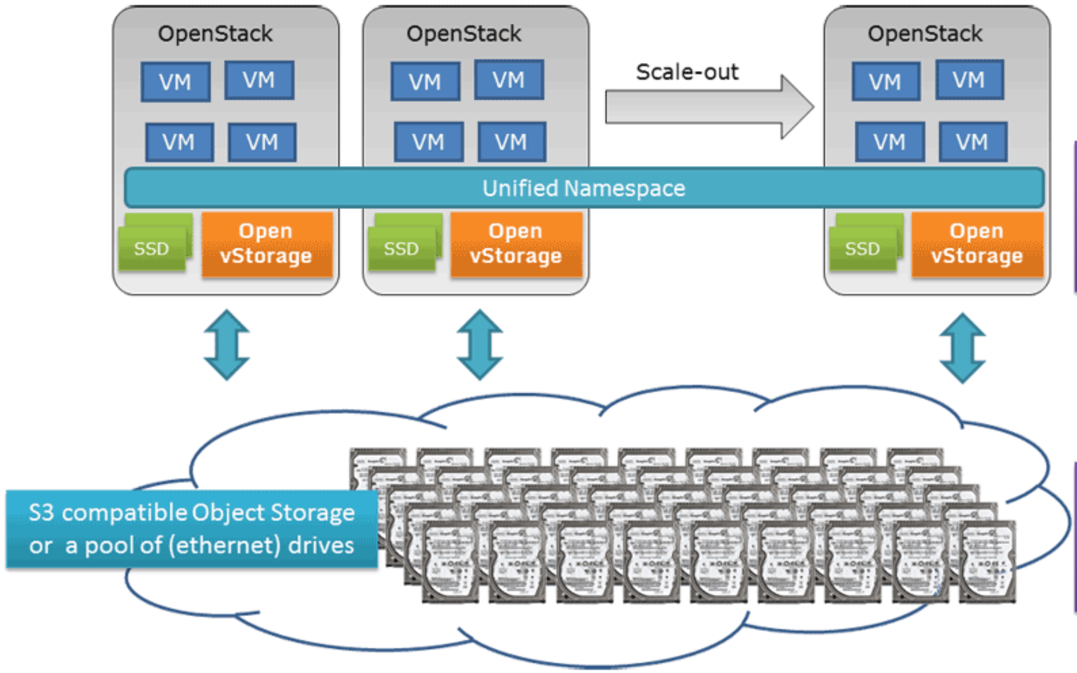
For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the first quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as "anticipates," "expects," "intends," "plans," "believes," "seeks," "estimates," "may," "will," "should" and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be important factors that could cause actual results to differ materially from the company's expectations. Demand for Intel's products is highly variable and could differ from expectations due to factors including changes in the business and economic conditions; consumer confidence or income levels; customer acceptance of Intel's and competitors' products; competitive and pricing pressures, including actions taken by competitors; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel's gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; and product manufacturing quality/yields. Variations in gross margin may also be caused by the timing of Intel product introductions and related expenses, including marketing expenses, and Intel's ability to respond quickly to technological developments and to introduce new features into existing products, which may result in restructuring and asset impairment charges. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Results may also be affected by the formal or informal imposition by countries of new or revised export and/or import and doing-business regulations, which could be changed without prior notice. Intel operates in highly competitive industries and its operations have high costs that are either fixed or difficult to reduce in the short term. The amount, timing and execution of Intel's stock repurchase program and dividend program could be affected by changes in Intel's priorities for the use of cash, such as operational spending, capital spending, acquisitions, and as a result of changes to Intel's cash flows and changes in tax laws. Product defects or errata (deviations from published specifications) may adversely impact our expenses, revenues and reputation. Intel's results could be affected by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel's ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. Intel's results may be affected by the timing of closing of acquisitions, divestitures and other significant transactions. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the company's most recent reports on Form 10-Q, Form 10-K and earnings release.

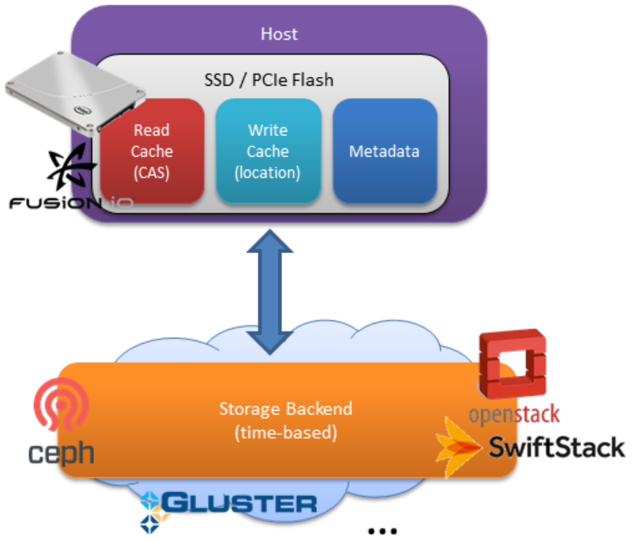
BACKUP

Open vStorage



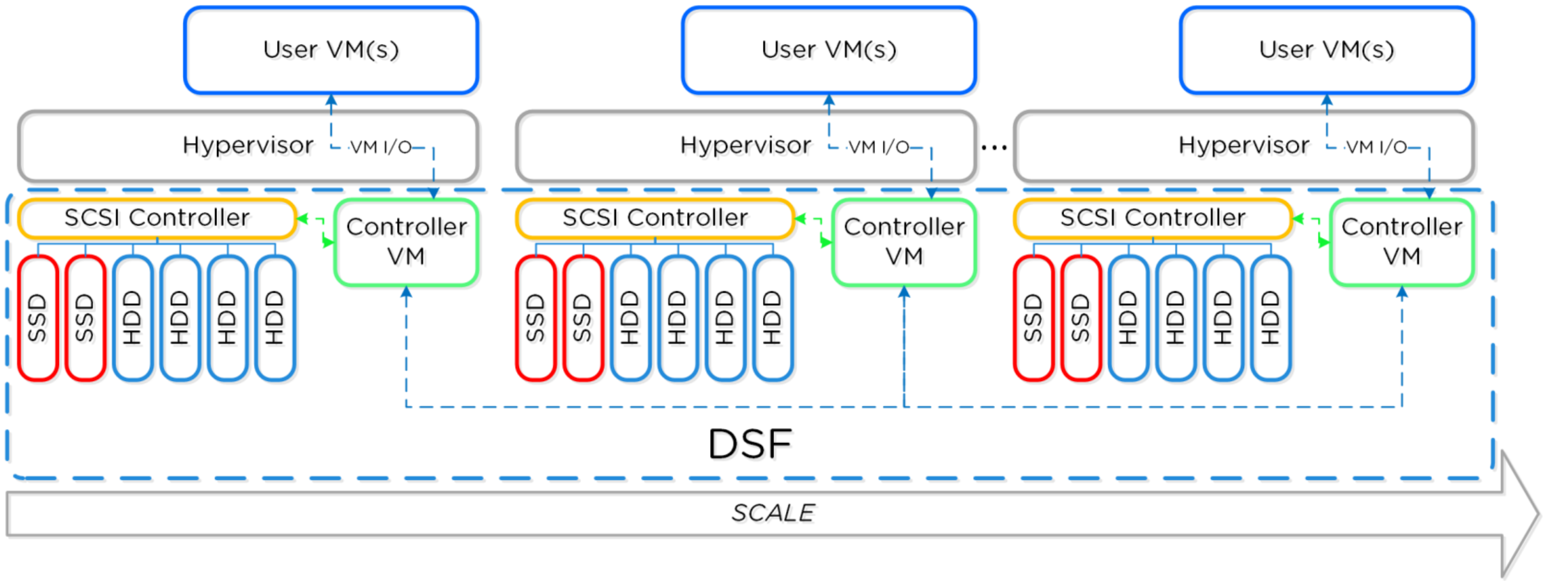
Tier 1 - Location Based
- Read/Write cache on SSD
- Block based storage
- Thin provisioning
- VM Centric
- Distributed Transaction Log

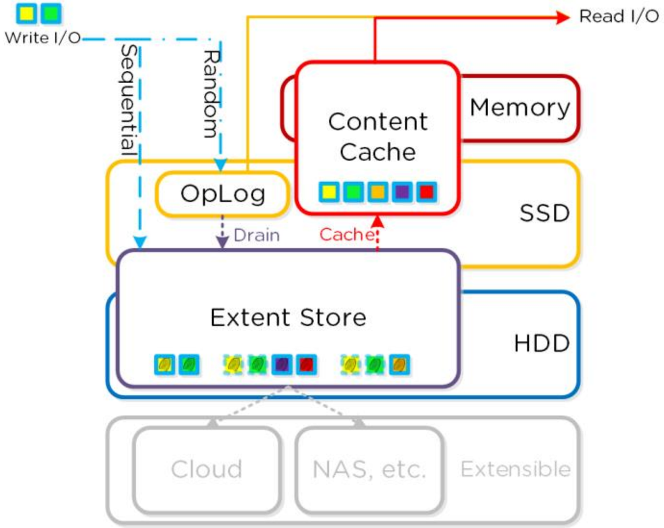
Tier 2 -Time Based
- Zero Copy Snapshot
- Zero Copy Cloning
- Continuous data protection
- Redundant storage
- Scale-out



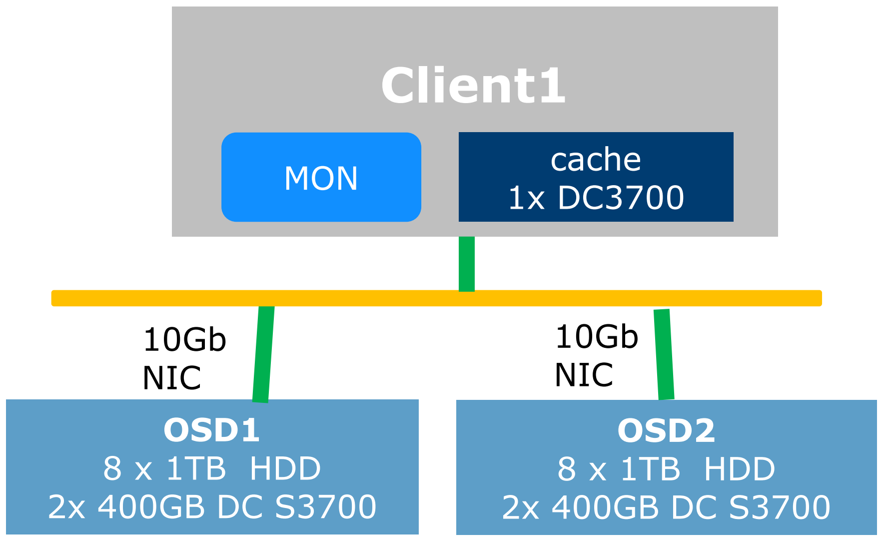
Turning Object Storage into block storage for Virtual Machine Storage

Nutanix


User VM(s)
User VM(s)
User VM(s)
Hypervisor
VM I/O
Hypervisor
VM I/O
...
Hypervisor
VM I/O
SCSI Controller
Controller VM
SCSI Controller
Controller VM
SCSI Controller
Controller VM
SSD
SSD
HDD
HDD
HDD
HDD
DSF
SCALE


Write I/O
Read I/O
Sequential
Random
Content Cache
Memory
OpLog
SSD
Drain
Cache
Extent Store
HDD
Cloud
NAS, etc.
Extensible

H/W Configuration



Client Cluster	
CPU	Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.80GHz
Memory	96 GB
NIC	10Gb
Disks	1 HDD for OS 400G SSD for cache
Ceph Cluster	
CPU	OSD: Intel(R) Xeon(R) CPU E31280 @ 3.50GHz
Memory	32 GB
NIC	10GbE
Disks	2 x 400 GB SSD (Journal) 8 x 1TB HDD (Storage)

2 hosts Ceph cluster each host has 8 x 1TB HDD as OSDs and 2x Intel® DC S3700 SSD journal

1 Client with 1x 400GB Intel® DC S3700 SSD as cache device

S/W Configuration

- Ceph* version : 10.2.2 (Jewel)
- Replica size : 2
  - Data pool : 16 OSDs. 2 SSDs for journal, 8 OSDs on each node
  - OSD Size : 1TB * 8
  - Journal Size : 40G * 8
  - Cache: 1 x 400G Intel® DC S3700
  - FIO volume size: 10G
- Cetune test benchmark
  - fio + librbd

Cetune: https://github.com/01org/cetune

Testing Configuration

Test cases:

- Operation: 4K random write with fio (zipf=1.2)

Detail case:

- Cache size < volume size (w/ zipf)
  - w/o flush & evict: cache size 10G.
  - w/ flush w/o evict: cache size 10G.
  - w/ flush & evict: cache size 10G.
- Hot data = volume size * zipf1.2(5%), runtime = 4 hours

Caching Parameters:

- object_size=4096
- cache_flush_queue_depth=256
- cache_ratio_max=0.7
- cache_ratio_health=0.5
- cache_dirty_ratio_min=0.1
- cache_dirty_ratio_max=0.95
- cache_flush_interval=3
- cache_evict_interval=5
- Runtime: Base: 200s ramp up, 14400s run
- DataStoreDev=/dev/sde
- cache_total_size=10G
- cacheservice_threads_num=128
- agent_threads_num=32