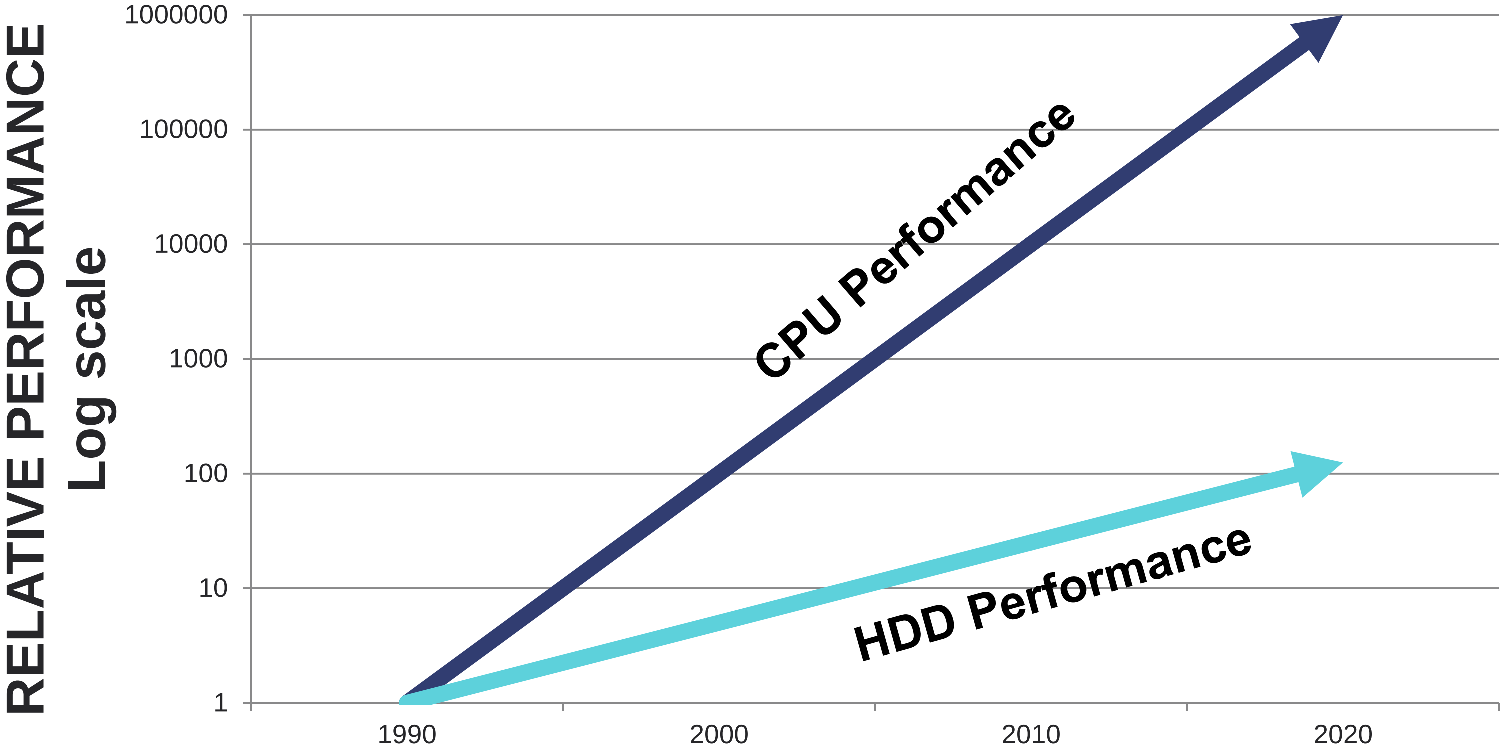
NVM EXPRESS™ IN LINUX*

THE NEXT STEP FOR STORAGE

Keith Busch (NVMe architecture and implementation)

Frank Ober (Solution Results on NVMe)

CPU vs. Storage Performance Gap



Switching to SSDs

SAS + SATA SSDs:

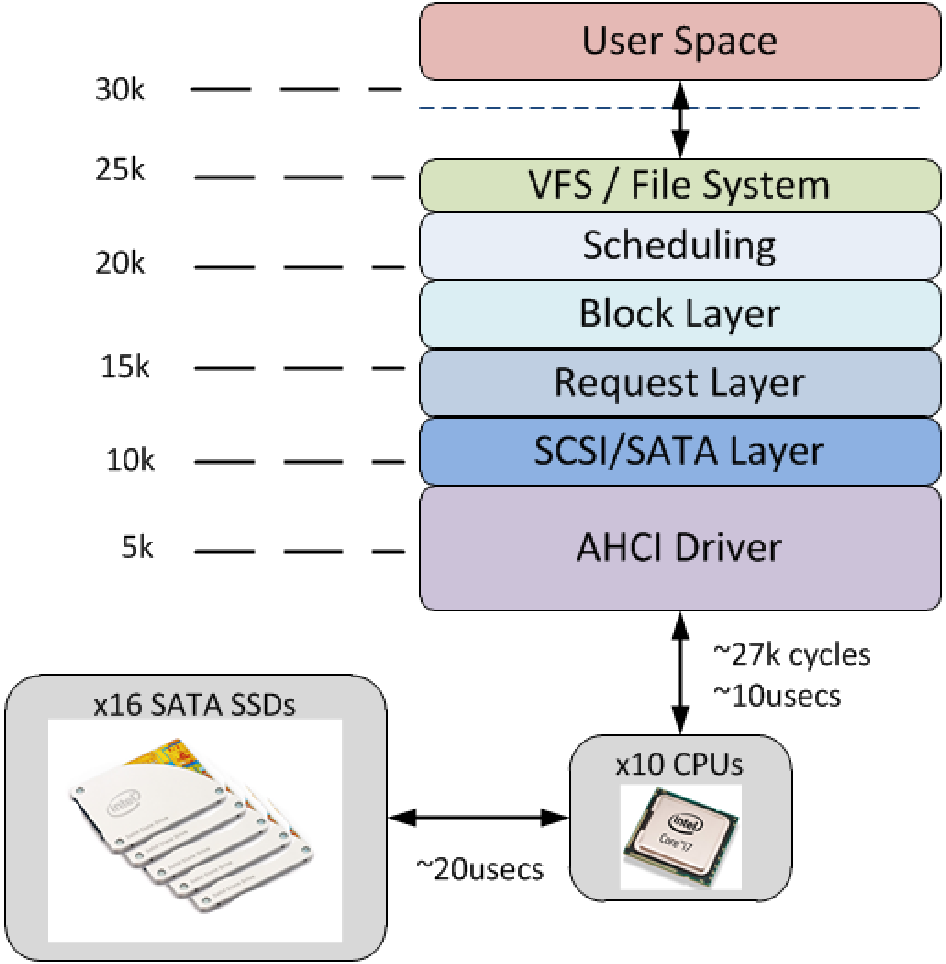
- Drop in replacement to HDDs
- Immediate latency benefit

Legacy software and transport prevent unlocking the media's true potential

To Maximize IOPS…

H/W resource intensive: software and protocol overhead

- 100% CPU utilization from 10 CPUs
- 16 SSDs



PCIe* Storage Standardization (since 2009)

nvm EXPRESS™

NVM Express™ and Linux*

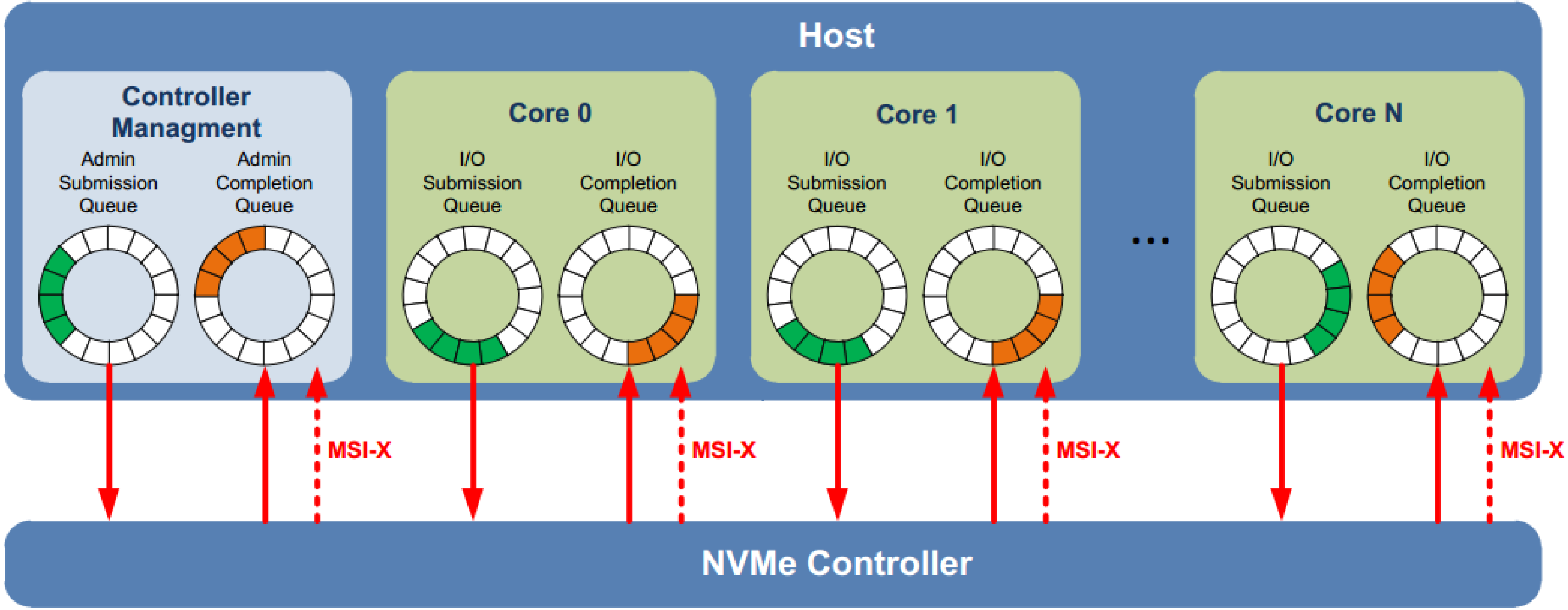
Integrated into mainline Linux* kernel since 3.3 (March 2012)

Backports to previous Linux* kernels supported by various OS vendors

What difference does a standard make?

	AHCI	NVMe
Maximum queue depth	1 command queue 32 commands	65536 queues 65536 commands per queue
MMIO	6 reads+writes/non-queued command 9 reads+writes/queued command	2 writes/command
Interrupts and steering	Single interrupt	2048 MSI-X interrupts CPU affinity
Parallelism	Single sync lock to issue command	Per-CPU lock contention free
Command Transfer Efficiency	Command requires two serialized host DRAM fetches	One 64B DMA fetch

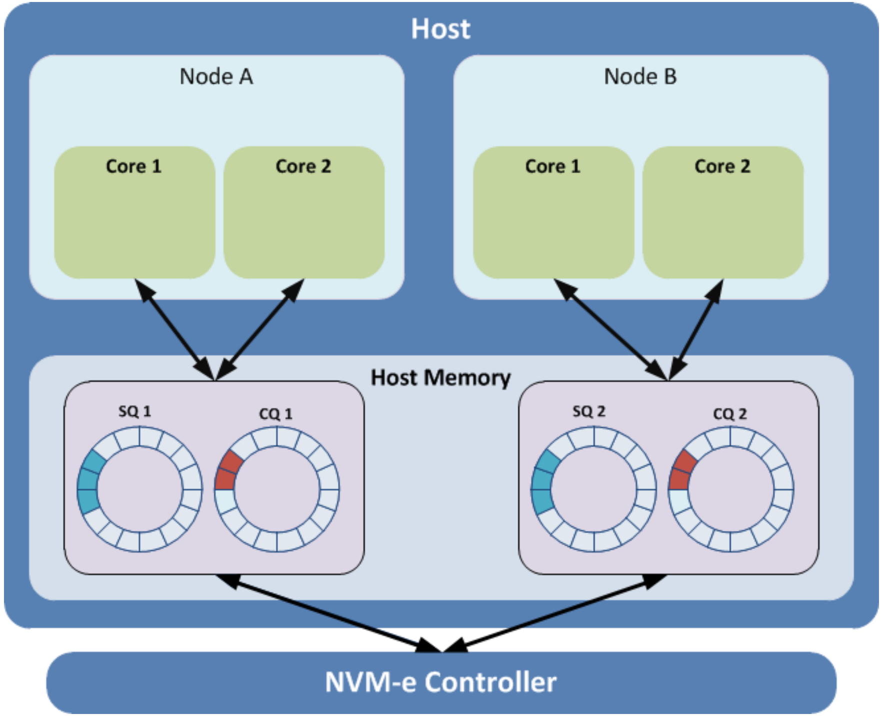
Optimized per-CPU queuing



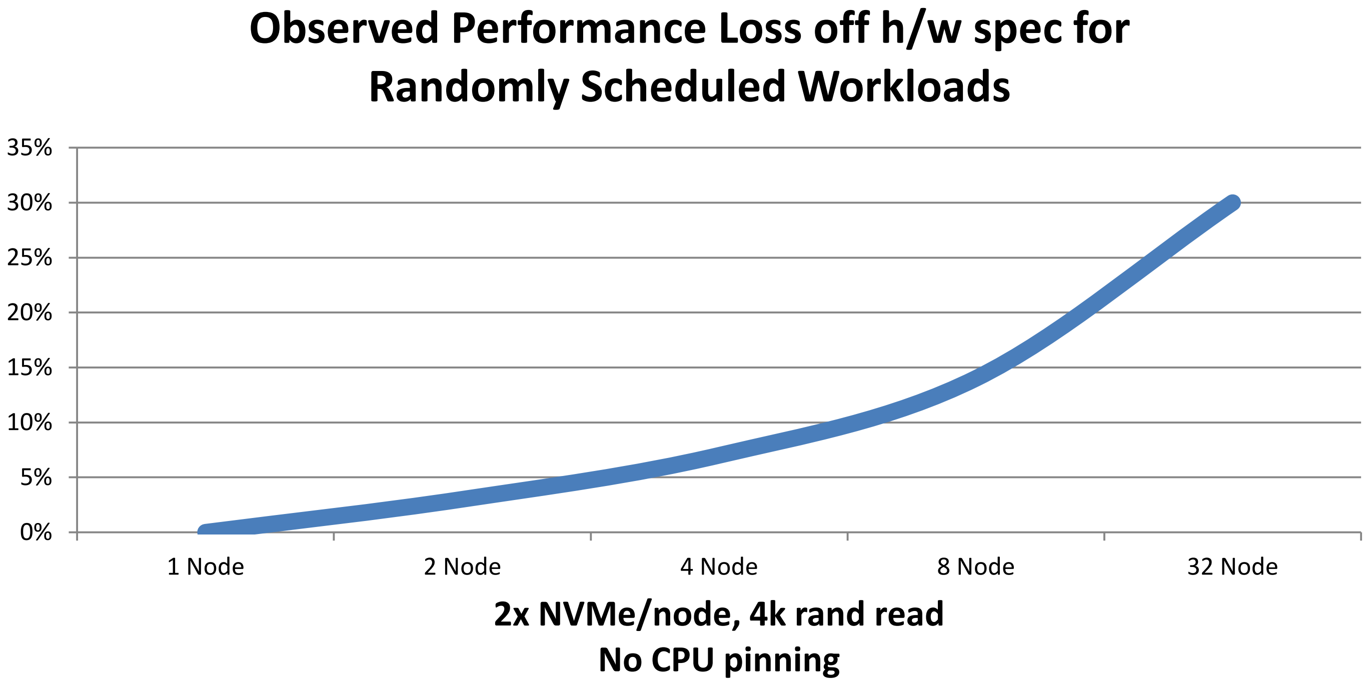
Optimizing for NUMA:

When CPUs exceed h/w queues:

Share with your neighbors



The cost of poor NUMA choices



Measured by Intel and SGI, on an SGI UV300 computer running a quantity of 32 Intel Xeon E7 v2 Processors with a quantity of 64 Intel SSD Data Center Family P3700 1.6TB using 100% 4k random reads. SGI public reference: http://blog.sgi.com/reinventing-compute-storage-landscape/

Case study: scaling upward with more h/w (SGI*)

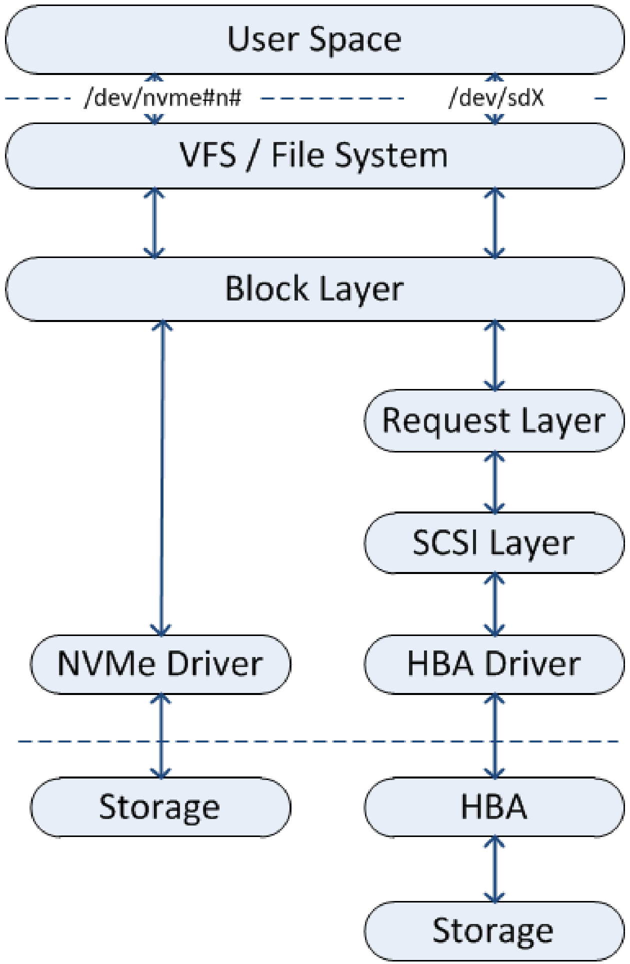
NUMA penalty: >30% performance lost

Intel and SGI solutions:

- irqbalance, numactl, libnuma, custom cpu-queue mapping
- Up to 30 **Million** IOPS (SC'14) of random read showing linear performance scaling as h/w is added

Storage Stack Comparison

- SAS vs. NVMe
- Latency and CPU utilization reduced by 50+%*:
  - NVMe: 2.8us, 9,100 cycles
  - SAS: 6.0us, 19,500 cycles

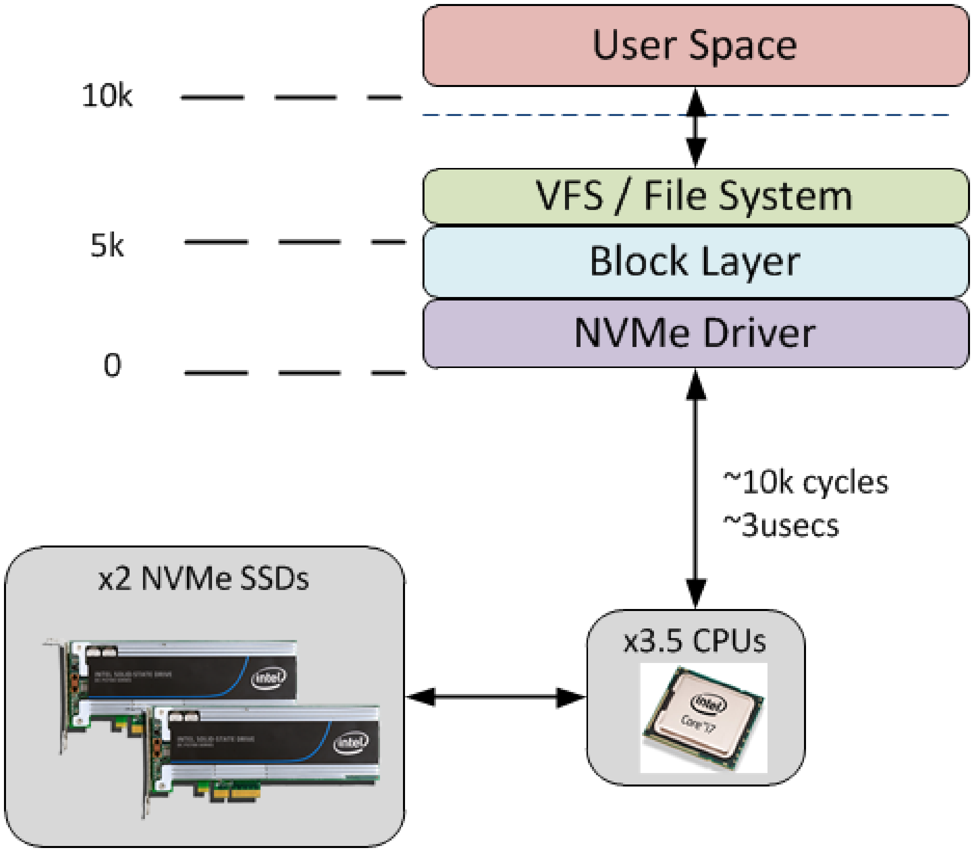


* Measured by Intel on Intel® Core™ i5-2500K 3.3GHz 6MB L3 Cache Quad-Core Desktop Processor using Linux*

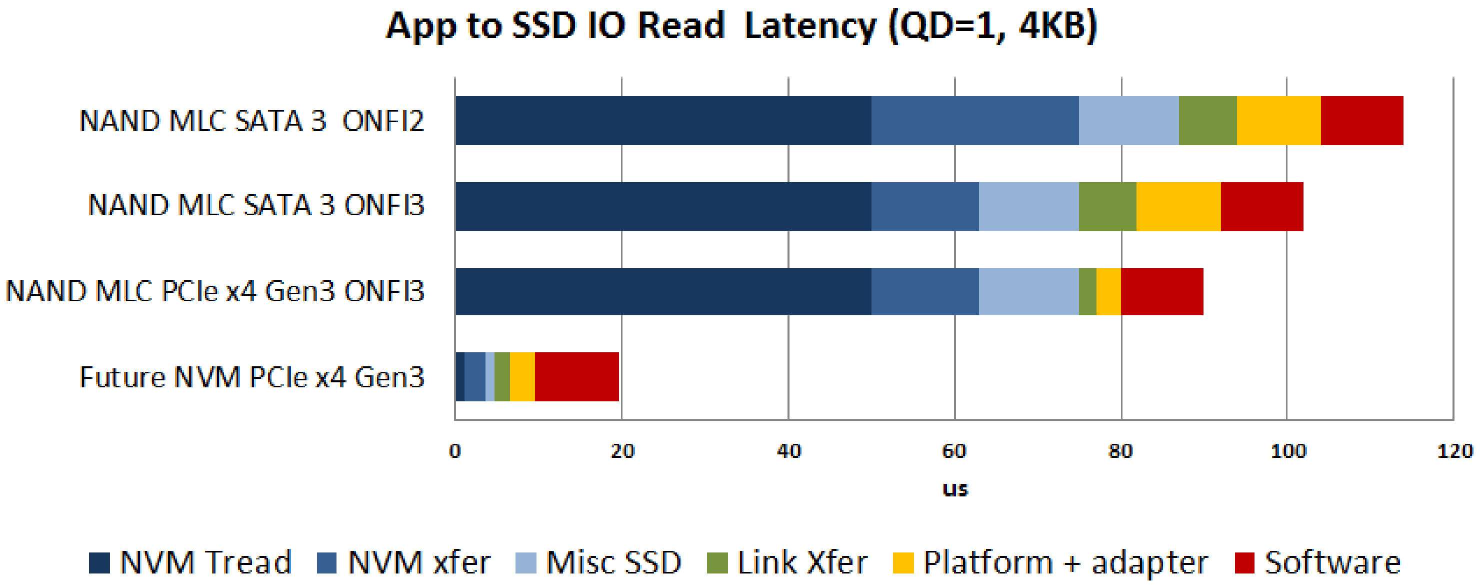
To Maximize IOPS…

Now with more efficient h/w utilization vs AHCI:

- 100% utilization from 3.5 CPUs (previously 10 CPUs)
- 2 SSDs (previously 16 SSDs)

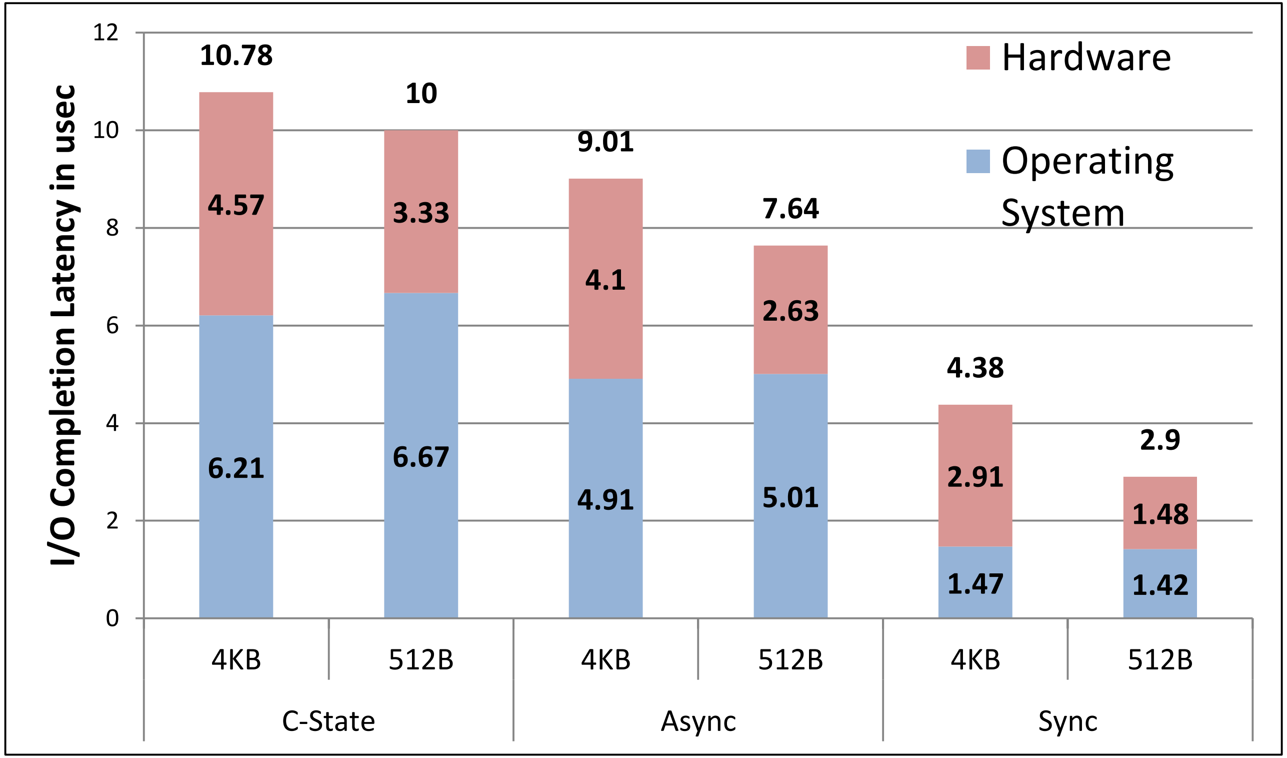


The importance of reducing S/W latency



* Measured by Intel on Intel® Core™ i5-2500K 3.3GHz 6MB L3 Cache Quad-Core Desktop Processor using Linux*

Looking ahead: removing interrupts



NVM EXPRESS™ IN LINUX*
MODERN NOSQL DATABASES FOR SSD AND FLASH

Frank Ober

http://communities.intel.com/people/FrankOber

@fxober / #IntelSSD

Are any databases truly Flash Optimized, and how do they do on NVMe™?

Glad you asked.

A short taxonomy of NoSQL Databases…

Type	Speed	Usage	Players
Key value databases	Fastest	Operational	Memcache, Redis, Aerospike Cloud guys use: DynamoDB* (Amazon). LevelDB (Google), Rocksdb* (Facebook)
Big Table , Column-based.	Faster	Analytics	Big Table*, Cassandra*, Hbase* (Hadoop)
Document databases	Faster	Web documents (JSON)	MongoDB (WiredTiger* v3.0 is released) Couchbase (ForestDB* releases 2015)
Graph databases	Fast	Social Graphs	Neo4J…

Aerospike an In-Memory, Flash Optimized NoSQL database

Environment

Aerospike
Community
Version 3.5.8

DUAL 10Gbit
networks

Dell R730xd Server System
One primary (dual system with replication testing)
Dual CPU socket, rack mountable server system
Dell A03 Board, Product Name: 0599V5
CPU Model used
2 each - Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz max frequency: 4Ghz
18 cores, 36 logical processors per CPU
36 cores, 72 logical processors total
DDR4 DRAM Memory
128GB installed
BIOS Version
Dell* 1.0.4 , 8/28/2014
Network Adapters
Intel® Ethernet Converged 10G X520 – DA2 (dual port PCIe add-in card)
1 – embedded 1G network adapter for management
2 – 10GB port for workload
Storage Adapters
None
Internal Drives and Volumes
/ (root) OS system – Intel SSD for Data Center Family S3500 – 480GB Capacity
/dev/nvme0n1 Intel SSD for Data Center Family P3700 – 1.6TB Capacity, x4 PCIe AIC
/dev/nvme1n1 Intel SSD for Data Center Family P3700 - 1.6TB Capacity, x4 PCIe AIC
/dev/nvme2n1 Intel SSD for Data Center Family P3700 - 1.6TB Capacity, x4 PCIe AIC
/dev/nvme3n1 Intel SSD for Data Center Family P3700 - 1.6TB Capacity, x4 PCIe AIC
6.4TB of raw capacity for Aerospike database namespaces

3 Clients
You need to spread the load
Here Dell 620 dual sockets are used

Aerospike results

The reason these tables on NVM are so fast is partially the small block. It also affects network usage… and costs of clusters so be careful with replication and object sizes.

Write mixes at 50/50 take the numbers down extensively.

https://communities.intel.com/community/itpeernetwork/blog/2015/02/17/reaching-one-million-database-transactions-per-second-aerospike-intel-ssd

Record Size Aerospike	Number of clients threads	Total TPS	Percent below 1ms (Reads)	Percent below 1ms (Writes)	Std Dev of Read Latency (ms)	Std Dev of Write Latency (ms)	Approx. Database size
1k	576	1,124,875	97.16	99.9	0.79	0.35	100G
2k	448	875,446	97.33	99.57	0.63	0.18	200G
4k	384	581,272	97.22	99.85	0.63	0.05	400G
1k (replication)	512	1,003,471	96.11	98.98	0.87	0.30	200G

Record Size iostat	Read MB/sec	Write MB/sec	Avg queue size on SSD	Average drive latency	CPU Busy %
1k	418	29	31	0.11	93
2k	547	43	27	0.13	81
4k	653	52	20	0.16	52
1k (replication)	396	51	30	0.13	94

Results measured by Intel and Aerospike. For tests and configurations, see slide 22.

TCO Opportunity of In Memory vs. In NVM

Storage Types	Cost per GB	1k transaction/socket	Memory Capacity
DRAM only	$10-15 + (DDR4)	Up to ~1.6 million tps (1 socket)	192GB – 768 GB
SSD Configuration	$1-3 + (PCIe SSD – retail channel)	Up to ~600k per node (1 socket)	4 x 2TB = 8TB 10# SFF NVMe servers

3x lower transactions per second, yet 5x lower price per GB with NVM.

Capacity is higher, cost is much lower allowing you to do more per unit of rack.

Costs measured by Intel from U.S. based internet retailer.

Couchbase

Now let's look at NoSQL – Web Document Store And Couchbase 4.0…

B+ Tree structured database indexing

Not suitable to index variable or fixed-length long keys

- Significant space overhead as entire key strings are indexed in non-leaf nodes

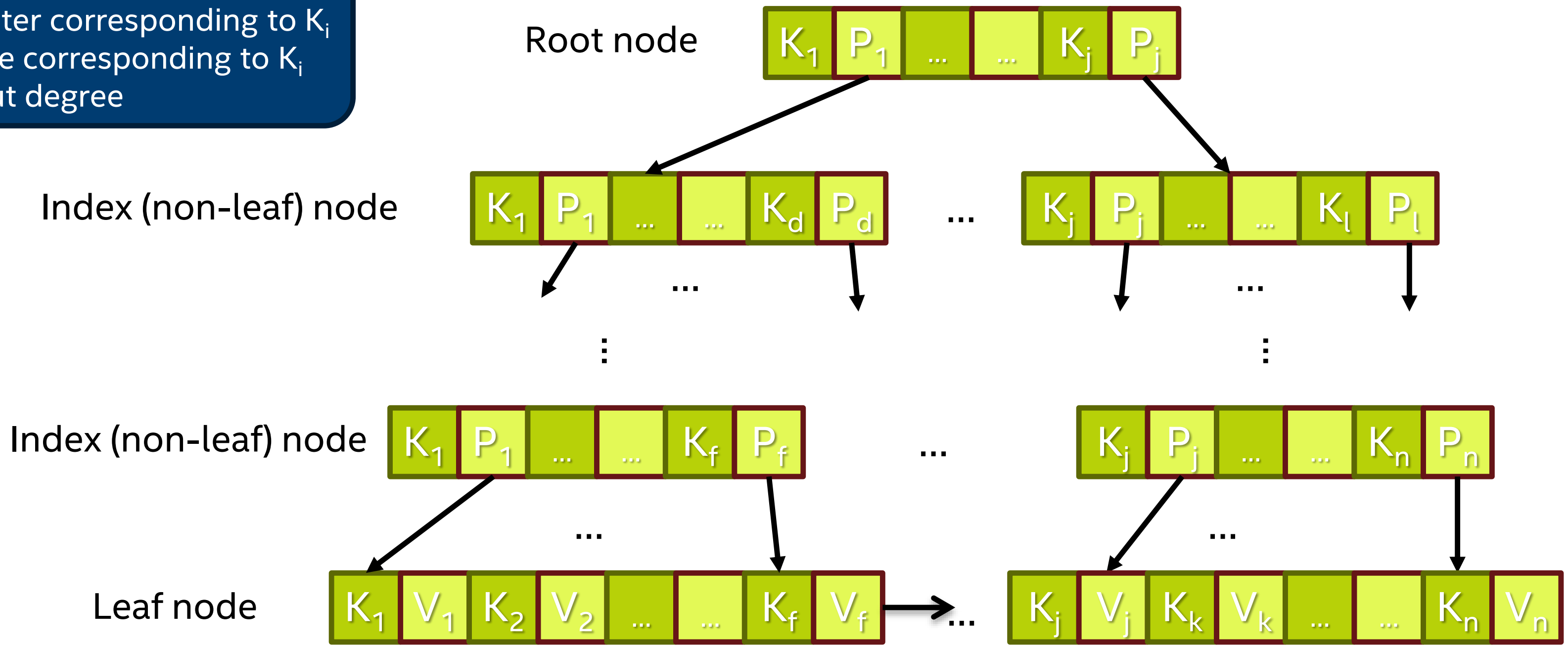
Tree depth grows quickly as more data is loaded

I/O performance is degraded significantly as the data size gets bigger


```
{ "users":[
        {
            "firstName":"Ray",
            "lastName":"Villalobos",
            "joined": {
                "month":"January",
                "day":12,
                "year":2012
            }
        },
        {
            "firstName":"John",
            "lastName":"Jones",
            "joined": {
                "month":"April",
                "day":28,
                "year":2010
            }
        }
  ]}
```

Introducing ForestDB – moving beyond B+ Tree

K_i: i[th] smallest key in the node
P_i: pointer corresponding to K_i
V_i: value corresponding to K_i
f: fanout degree



Source: ©2015 Couchbase Inc.

How ForestDB tries to achieve….

Fast, flatter, scalable index structure for variable or fixed-length long keys

- Targeting both SSD and HDD

Less storage space overhead

- Reduce write amplification

Regardless of the pattern of keys

- Efficient to keys both sharing common prefix and not sharing common prefix

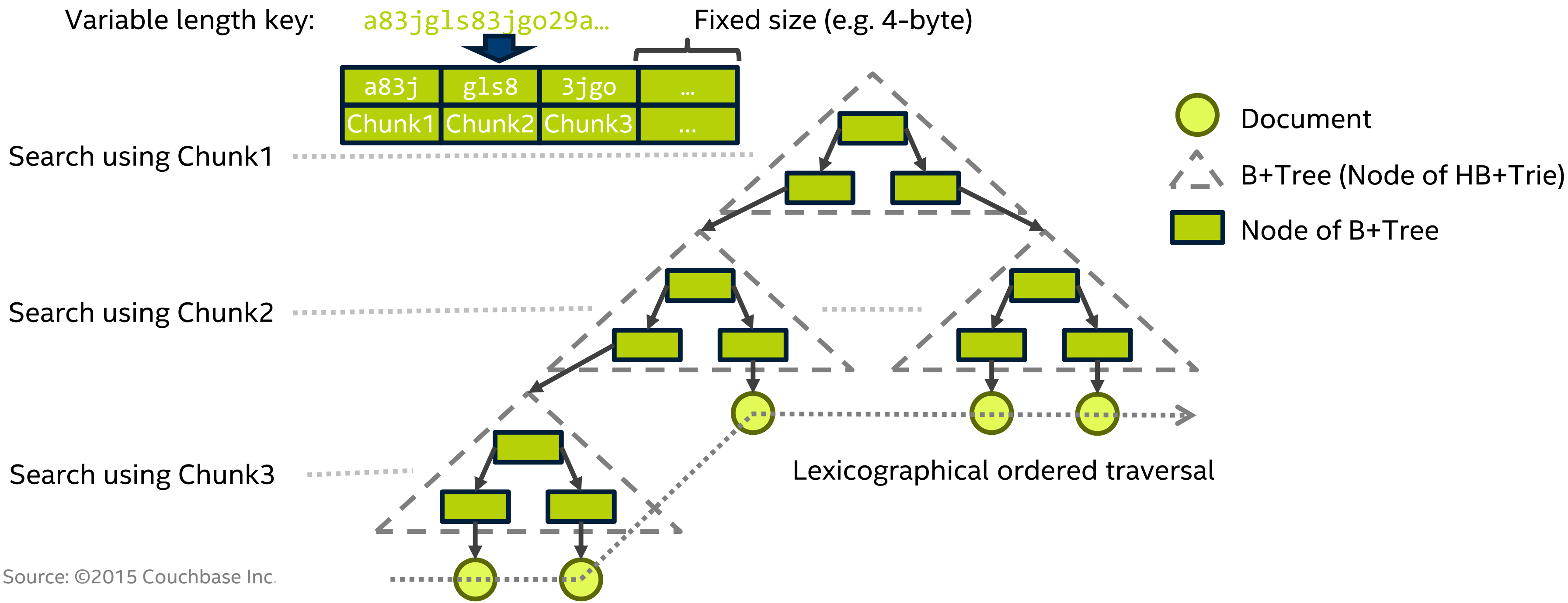
Compaction of large index or db files is still slow…

Source: ©2015 Couchbase Inc.

HB+Trie (Hierarchical B+Tree based Trie)

Trie (prefix tree) whose node is B+Tree

- A key is split into the list of fixed-size chunks (sub-string of the key)



Source: ©2015 Couchbase Inc.

Lab Configuration

- Intel® Xeon® processor E5-2697 v3 @ 2.60GHz
- Number of Cores: 28 (56 hw threads)
- RAM: 65G
- Storage:
  - SATA SSD: Intel DC S3710 1.2TB (~$1 / GB)
  - NVMe™ SSD: Intel DC P3700 1.6TB (~$2.5/ GB)
- ForestDB: https://github.com/couchbase/forestdb
- ForestDB benchmark: https://github.com/couchbaselabs/ForestDB-Benchmark

Source: ©2015 Couchbase Inc.

Testing Scenarios

- Key/Value store (used in the data server layer)
- Index Simulation (first place ForestDB will arrive)
- Throughput Testing (Parallel Benchmark)

Summary

	K/V Store		Indexing		Parallel Throughput		Benefits
	SATA	NVMe	SATA	NVMe	SATA	NVMe	
Read Throughput	16678	25302	13987	20341	30755	47345	Up to 50%
Write Throughput	4170	6325	3497	5209	7282	63946	Up 9x
95% Read Latency	1.745	1.136	1.853	1.254	4.0	7.5	Some work
95% Write Latency	264	188	276	216	1934	270	Awesome

Results measured by Intel and Couchbase Inc. For tests and configurations, see slide 30.

Legal Disclaimer

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to http://www.intel.com/design/literature.htm.

This document may contain information on products in the design phase of development.

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.

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.

Results have been simulated and are provided for informational purposes only. Results were derived using simulations run on an architecture simulator or model. Any difference in system hardware or software design or configuration may affect actual performance.

Intel does not control or audit the design or implementation of third party benchmark data or Web sites referenced in this document. Intel encourages all of its customers to visit the referenced Web sites or others where similar performance benchmark data are reported and confirm whether the referenced benchmark data are accurate and reflect performance of systems available for purchase.

Experience NVM as a complement to DRAM

experience
what's inside™