NVM EXPRESS™ IN LINUX*
THE NEXT STEP FOR STORAGE

Keith Busch (NVMe architecture and implementation)
Frank Ober (Solution Results on NVMe)
RELATIVE PERFORMANCE
Log scale

CPU vs. Storage Performance Gap

CPU Performance
HDD Performance

1990 2000 2010 2020
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
PCle* Storage Standardization (since 2009)
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?

<table>
<thead>
<tr>
<th></th>
<th>AHCI</th>
<th>NVMe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum queue depth</strong></td>
<td>1 command queue</td>
<td>65536 queues</td>
</tr>
<tr>
<td></td>
<td>32 commands</td>
<td>65536 commands per queue</td>
</tr>
<tr>
<td><strong>MMIO</strong></td>
<td>6 reads+writes/non-queued command</td>
<td>2 writes/command</td>
</tr>
<tr>
<td></td>
<td>9 reads+writes/queued command</td>
<td></td>
</tr>
<tr>
<td><strong>Interrupts and steering</strong></td>
<td>Single interrupt</td>
<td>2048 MSI-X interrupts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPU affinity</td>
</tr>
<tr>
<td><strong>Parallelism</strong></td>
<td>Single sync lock to issue command</td>
<td>Per-CPU lock contention free</td>
</tr>
<tr>
<td><strong>Command Transfer Efficiency</strong></td>
<td>Command requires two serialized host</td>
<td>One 64B DMA fetch</td>
</tr>
<tr>
<td></td>
<td>DRAM fetch</td>
<td></td>
</tr>
</tbody>
</table>
Optimized per-CPU queuing
Optimizing for NUMA:

When CPUs exceed h/w queues:
Share with your neighbors
The cost of poor NUMA choices

Observed Performance Loss off h/w spec for Randomly Scheduled Workloads

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

I/O Completion Latency in usec

<table>
<thead>
<tr>
<th></th>
<th>4KB C-State</th>
<th>512B C-State</th>
<th>4KB Async</th>
<th>512B Async</th>
<th>4KB Sync</th>
<th>512B Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>10.78</td>
<td>10</td>
<td>9.01</td>
<td>7.64</td>
<td>4.38</td>
<td>2.9</td>
</tr>
<tr>
<td>Operating System</td>
<td>6.21</td>
<td>6.67</td>
<td>4.91</td>
<td>5.01</td>
<td>2.91</td>
<td>1.47</td>
</tr>
<tr>
<td></td>
<td>4.57</td>
<td>3.33</td>
<td>2.63</td>
<td>1.48</td>
<td>1.42</td>
<td></td>
</tr>
</tbody>
</table>
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...

<table>
<thead>
<tr>
<th>Type</th>
<th>Speed</th>
<th>Usage</th>
<th>Players</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key value databases</td>
<td>Fastest</td>
<td>Operational</td>
<td>Memcache, Redis, Aerospike</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cloud guys use: DynamoDB* (Amazon). LevelDB (Google), Rocksdb* (Facebook)</td>
</tr>
<tr>
<td>Big Table, Column-based.</td>
<td>Faster</td>
<td>Analytics</td>
<td>Big Table*, Cassandra*, Hbase* (Hadoop)</td>
</tr>
<tr>
<td>Document databases</td>
<td>Faster</td>
<td>Web documents (JSON)</td>
<td>MongoDB (WiredTiger* v3.0 is released) Couchbase (ForestDB* releases 2015)</td>
</tr>
<tr>
<td>Graph databases</td>
<td>Fast</td>
<td>Social Graphs</td>
<td>Neo4J...</td>
</tr>
</tbody>
</table>
Aerospike an In-Memory, Flash Optimized NoSQL database
Environment

Aerospike Community Version 3.5.8

DUAL 10Gbit networks

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

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
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.


---

<table>
<thead>
<tr>
<th>Record Size Aerospike</th>
<th>Number of clients threads</th>
<th>Total TPS</th>
<th>Percent below 1ms (Reads)</th>
<th>Percent below 1ms (Writes)</th>
<th>Std Dev of Read Latency (ms)</th>
<th>Std Dev of Write Latency (ms)</th>
<th>Approx. Database size</th>
<th>Record Size iostat</th>
<th>Read MB/sec</th>
<th>Write MB/sec</th>
<th>Avg queue size on SSD</th>
<th>Average drive latency</th>
<th>CPU Busy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1k</td>
<td>576</td>
<td>1,124,875</td>
<td>97.16</td>
<td>99.9</td>
<td>0.79</td>
<td>0.35</td>
<td>100G</td>
<td></td>
<td>418</td>
<td>29</td>
<td>31</td>
<td>0.11</td>
<td>93</td>
</tr>
<tr>
<td>2k</td>
<td>448</td>
<td>875,446</td>
<td>97.33</td>
<td>99.57</td>
<td>0.63</td>
<td>0.18</td>
<td>200G</td>
<td></td>
<td>547</td>
<td>43</td>
<td>27</td>
<td>0.13</td>
<td>81</td>
</tr>
<tr>
<td>4k</td>
<td>384</td>
<td>581,272</td>
<td>97.22</td>
<td>99.85</td>
<td>0.63</td>
<td>0.05</td>
<td>400G</td>
<td></td>
<td>653</td>
<td>52</td>
<td>20</td>
<td>0.16</td>
<td>52</td>
</tr>
<tr>
<td>1k (replication)</td>
<td>512</td>
<td>1,003,471</td>
<td>96.11</td>
<td>98.98</td>
<td>0.87</td>
<td>0.30</td>
<td>200G</td>
<td></td>
<td>396</td>
<td>51</td>
<td>30</td>
<td>0.13</td>
<td>94</td>
</tr>
</tbody>
</table>

Results measured by Intel and Aerospike. For tests and configurations, see slide 22.
## TCO Opportunity of In Memory vs. In NVM

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

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.
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
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
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)

Variable length key: a83jgls83jgo29a...
Fixed size (e.g. 4-byte)

Search using Chunk1

Search using Chunk2

Search using Chunk3

Lexicographical ordered traversal

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](https://github.com/couchbase/forestdb)
- ForestDB benchmark: [https://github.com/couchbaselabs/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

<table>
<thead>
<tr>
<th></th>
<th>K/V Store</th>
<th>Indexing</th>
<th>Parallel Throughput</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SATA</td>
<td>NVMe</td>
<td>SATA</td>
<td>NVMe</td>
</tr>
<tr>
<td>Read Throughput</td>
<td>16678</td>
<td>25302</td>
<td>13987</td>
<td>20341</td>
</tr>
<tr>
<td>Write Throughput</td>
<td>4170</td>
<td>6325</td>
<td>3497</td>
<td>5209</td>
</tr>
<tr>
<td>95% Read Latency</td>
<td>1.745</td>
<td>1.136</td>
<td>1.853</td>
<td>1.254</td>
</tr>
<tr>
<td>95% Write Latency</td>
<td>264</td>
<td>188</td>
<td>276</td>
<td>216</td>
</tr>
</tbody>
</table>

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.


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.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2015 Intel Corporation. All rights reserved.
Experience NVM as a complement to DRAM