Running NoSQL Natively on Flash Fusion-io SDK

Torben Mathiasen & Salvatore Buccoliero

The Future of High Performance Storage?

FUSION-IO

Everyone seems to agree.

ioMemory

FUSION-IO

- A New Memory tier called ioMemory
  - Leverages the best advantages of DRAM and rotating drives
    - High Speed like DRAM
    - Persistence and Large capacity of Spinning Hard Drives

- PCIe based NAND Flash storage
  - Micro-second level Disk Access Latency - 15µs
  - Very high data throughput - 1,5GB/s
  - Very high IOPS – 400.000 random write/s
  - Scalable – stay ahead of data / performance demanc
  - Advanced wear-leveling algorithm

  - N+1 Chip level redundancy (think RAID protection on card)
  - 100% data integrity protection in case of power loss
  - Endurance is PBW – TB's written daily for more than 8 years!

Manufactured by Fusion-io - OEM'ed by

IBM hp DELL NetApp CISCO

ioMemory vs Disk

800,000 IOPs

=

4,000 x

150-200 IOPs

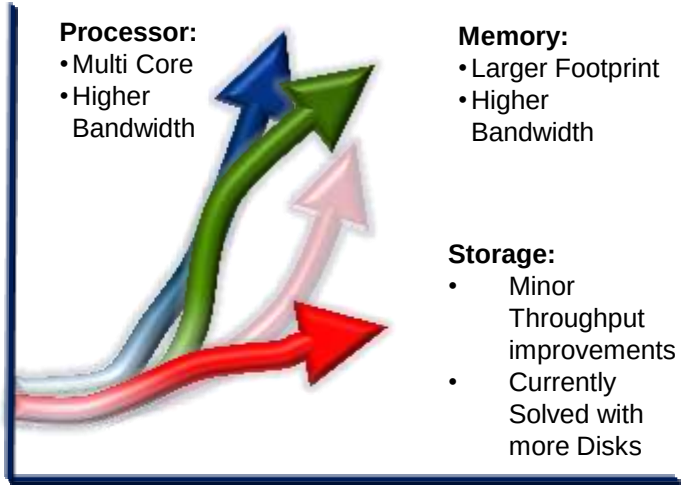
Where to use ioMemory

FUSION-IO

Databases	Virtualization	Search	Analytics	Big Data	Backup
ORACLE, MySQL, SQL Server, IBM DB2, INFORMIX, SAP, SYBASE, INGRES, PostgreSQL	vmware, Windows Server Hyper-V, XenDesktop 5, KVM	fast, Autonomy, lucene, ORACLE Text	AccessData, MarkLogic, LexisNexis	hadoop, mongoDB, HDInsight, Cassandra	commvault, Tivoli, Tivoli Storage Manager

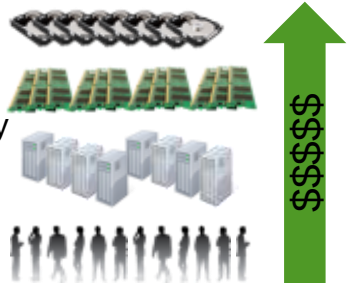
HPC	Messaging	Workstation	Development	Caching	Security/Logging	Web
FLUENT, MagmaSoft, NX NASTRAN, lustre, IBM GPFS	IBM MQ, TIBCO The Power of Now, software AG	Autodesk, SolidWorks, Ps	PERFORCE	Powered by Spymemcached, VARNISH SOFTWARE	ArcSight, splunk>	LAMP, Microsoft .NET

The compute performance problem



Legacy solution to the data supply problem:

- Add more disk
- Add more memory
- Add more servers
- Optimize the application

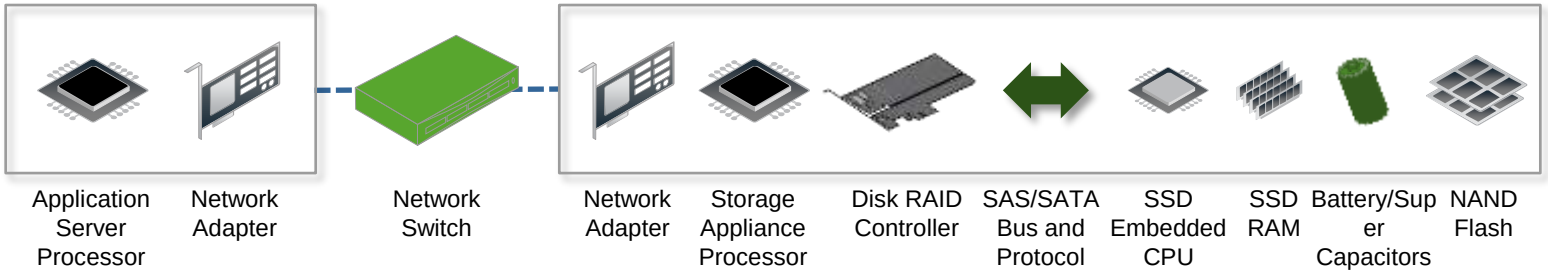


Each option requires significant increase in CAPEX and OPEX, and does not fully address the problem.

- "Compute power continues to outpace performance delivered by Storage."
  - "Problem is not getting better, its getting worse."

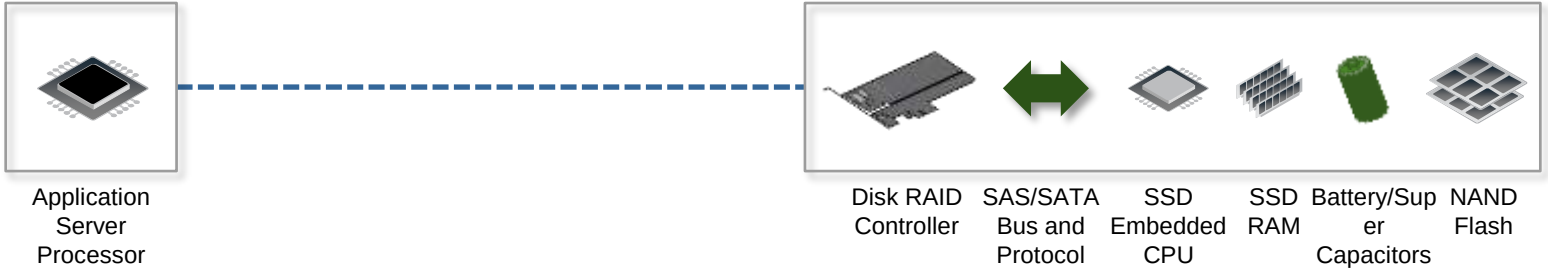
Networked storage data supply chain from application to flash

FUSION-IO



- **9 Intermediary components required**
- All adding access delay, cost, complexity, and lowering reliability (especially the super capacitors)
- Requests must do a round trip touching everything TWICE…

SSD data supply chain from application to flash

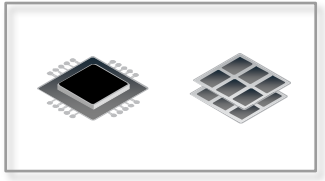


- **5 Intermediary components required**
- All adding access delay, cost, complexity, and lowering reliability (especially the super capacitors)

A horse in front of a Ferrari?

FUSION-IO

Fusion-io Approach
From Application…. to Flash

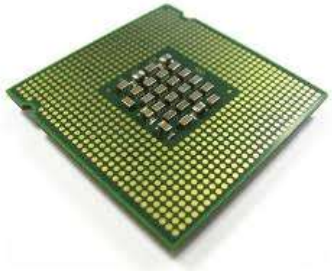


- **0 Intermediary components required**
- No need for super capacitors because data is not "buffered" in DRAM

The landscape of sub second Timings

FUSION-IO

HOW FAST DO YOU GET DATA TO THE FACTORY?



Very Small - Kb

L1-L3
Cache 10 ns

Very expensive
Volatile

DRAM
100 ns

Non volatile

Fusion-io
15 µs

SAN
4 ms

Blink of an eye
1/10 second

Heartbeat
1 second

Get cup of Coffee
2,5 minutes

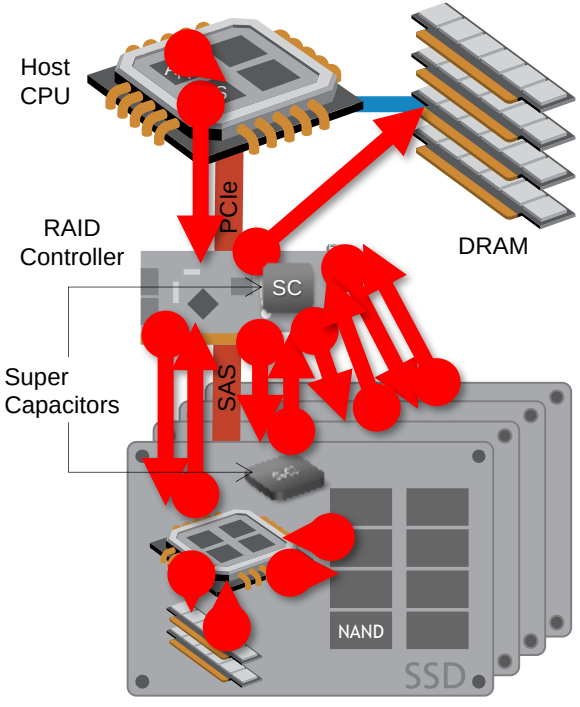
Fly to Thailand
11,1 hours

Multiplier is 10m - 10.000.000

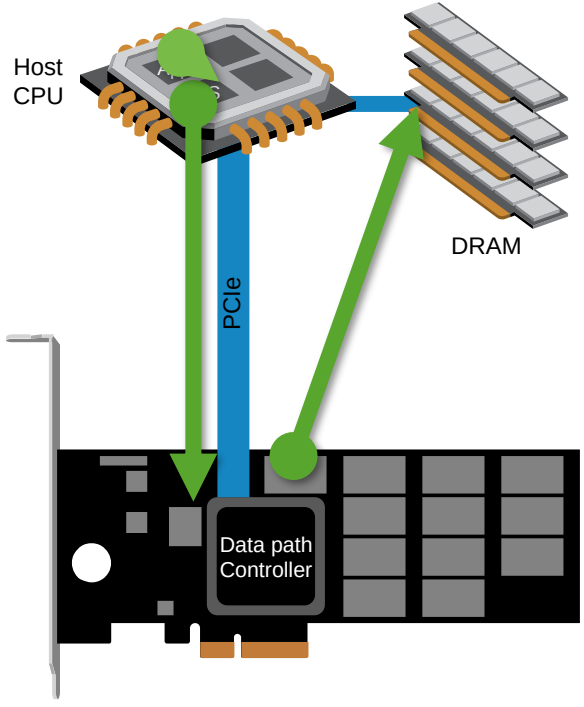
Direct Cut Through Architecture

FUSION-IO

LEGACY APPROACH



FUSION DIRECT APPROACH



The goal of every I/O operation is to move data to and from DRAM and the device

Fusion-io is not a SSD device

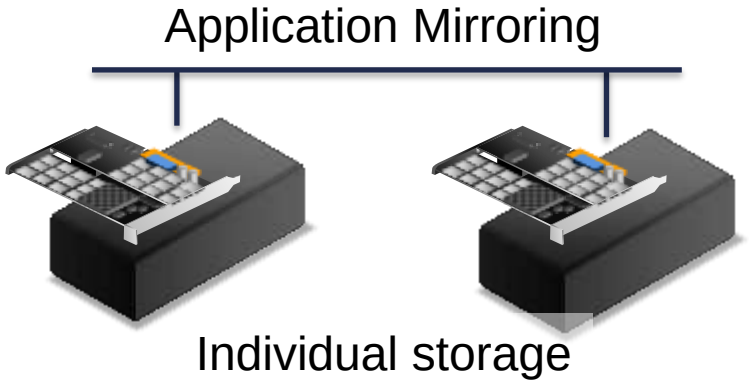
Usage Models – Baby Steps

- Moving specific components of the database to the ioDrives:
  - Tempdb database
  - Indexes
  - Frequently accessed tables
  - Transaction logs
  - Partition tables

All In

- If database size permits, placing entire database system on Fusion-ioDrives provides maximum performance benefit

n Node Cluster



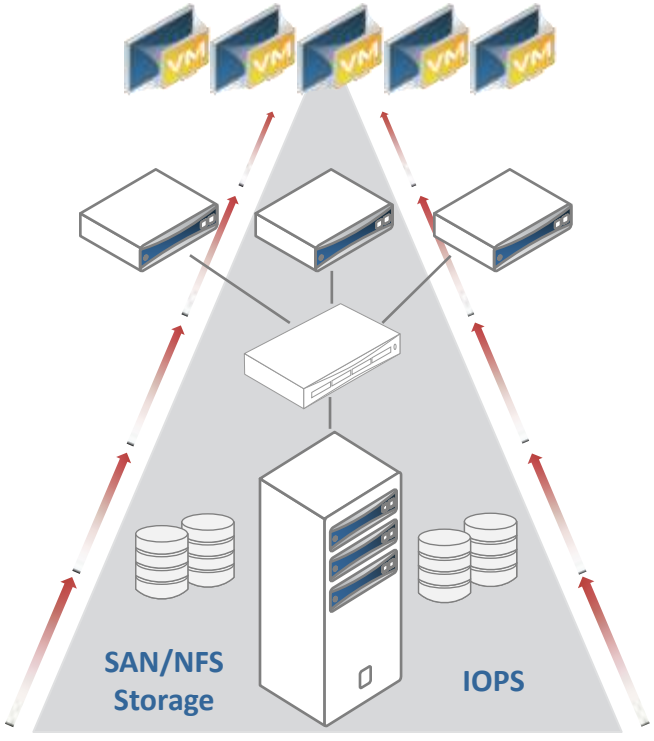
- Perfect NoSQL Model
- MSFT SQL Server Always On
- Oracle Dataguard
- SIOS DataKeeper
- Advantages
  - Fast replication
  - Just another block storage device

Clustering / HA with No Shared Storage!

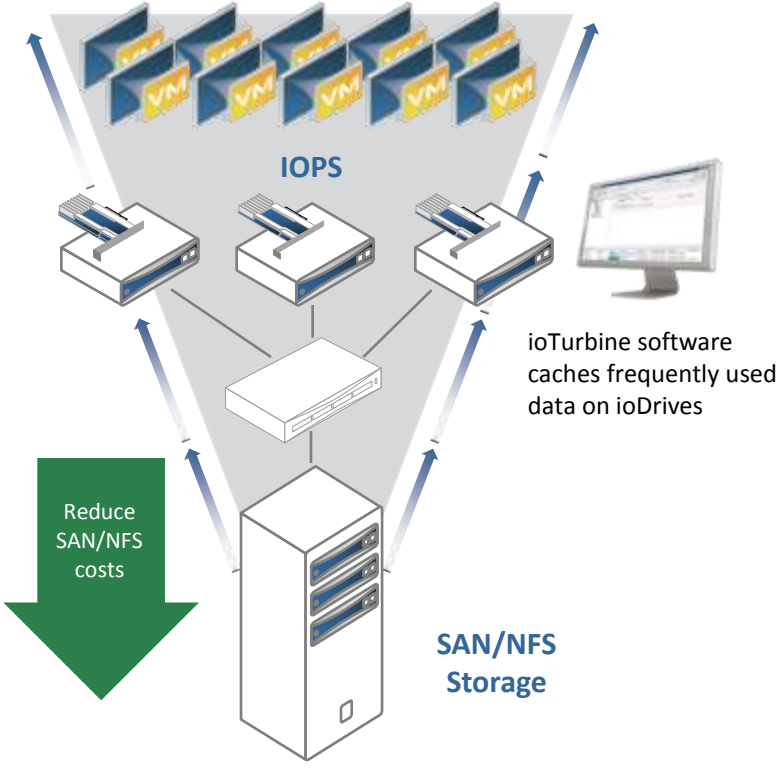
Getting Performance To ESXi

FUSION-IO

- External storage for virtual machines too costly

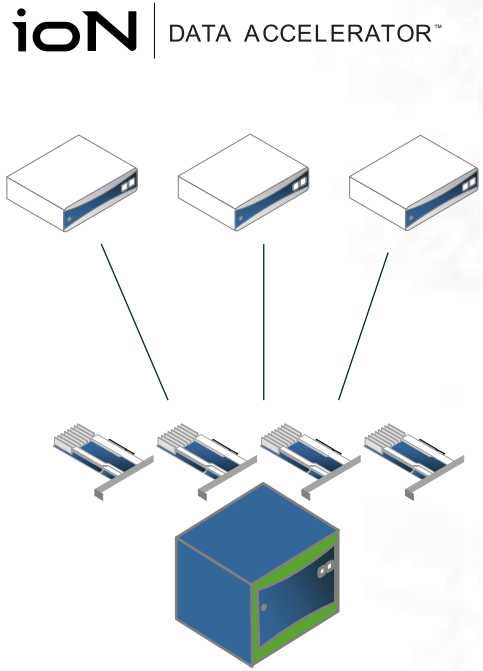


- Fusion-io delivers IOPS to hosts and virtual machines



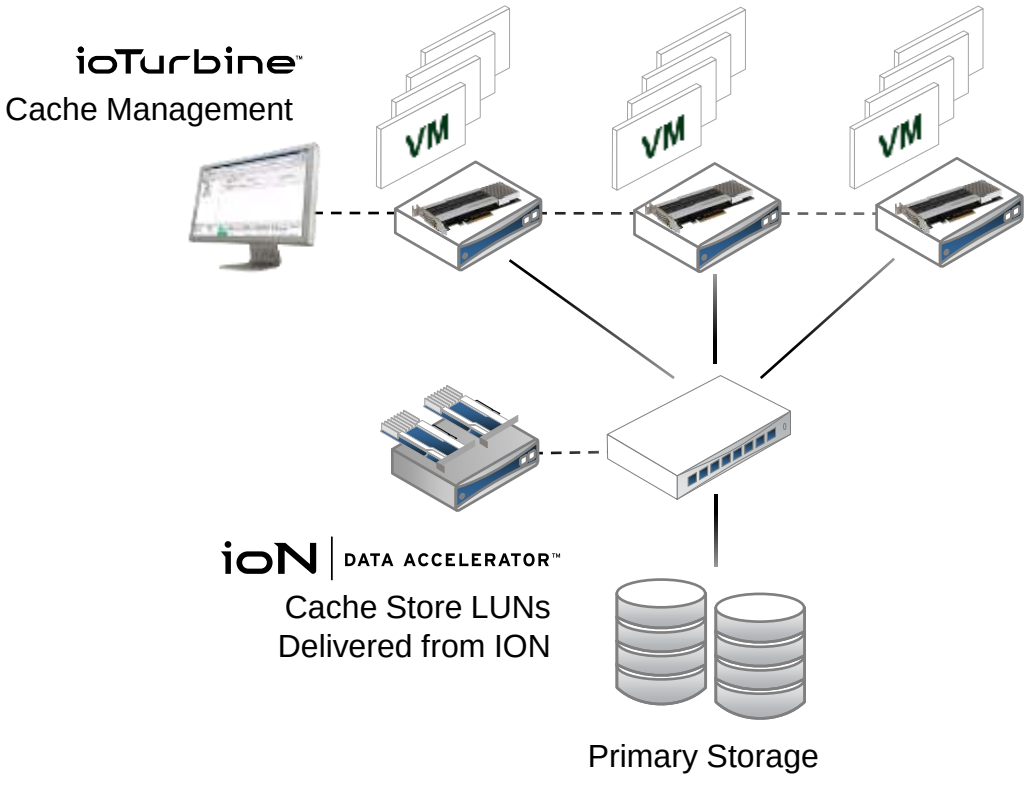
Fusion-io as a storage appliance server

- Standard HP, IBM, DELL Servers
- Rack or Blades…
- SHARED STORAGE
- FC
- ISCSI
- HA MIRROR



IoTurbine with ION cache store (distributed cache)

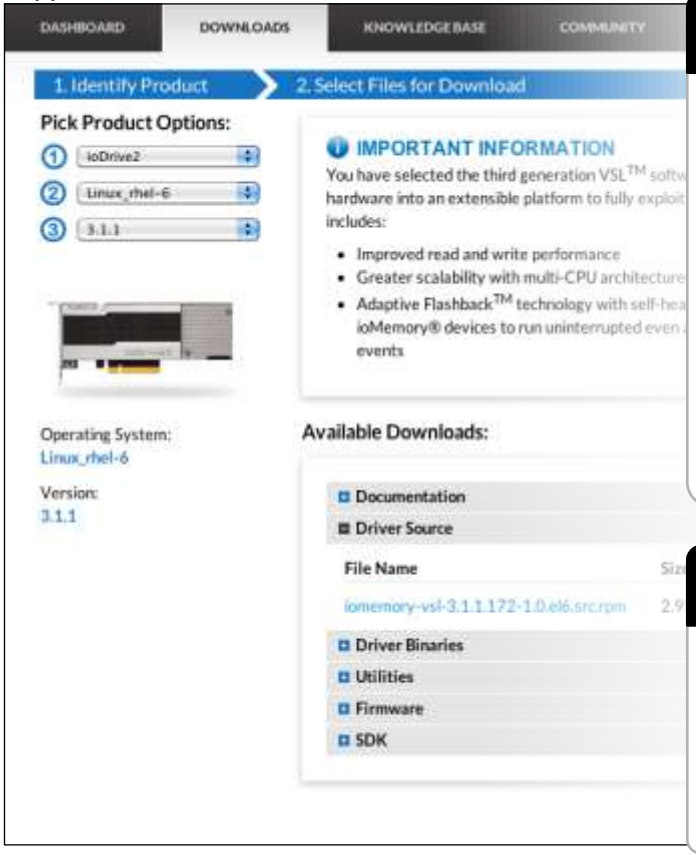
FUSION-IO



- Best For
  - Enterprise class shared caching
  - Large scale server farms
  - Ideal where servers cannot accommodate local ioMemory
  - Each ESX(i) Host will have a unique ION LUN presented

OS Support

support.fusionio.com



Windows

Windows Server 2003 SP2
Windows 7 64 bit
Windows 8 64 bit (in Oct)
Windows Server 2008 R1 SP2
Windows Server 2008 R2
Windows Server 2012 (in Oct)

Linux

RHEL 5.6, 5.7, 5.8, 6.0, 6.1, 6.2
SLES 10.4, 11, 11.1
OEL 5.7, 6.0, 6.1, 6.2
CentOS 5.6, 5.7, 6.0, 6.1, 6.2
Debian Squeeze
Fedora 15, 16
openSUSE 12.1
Ubuntu 10.04, 11.10

Hypervisors

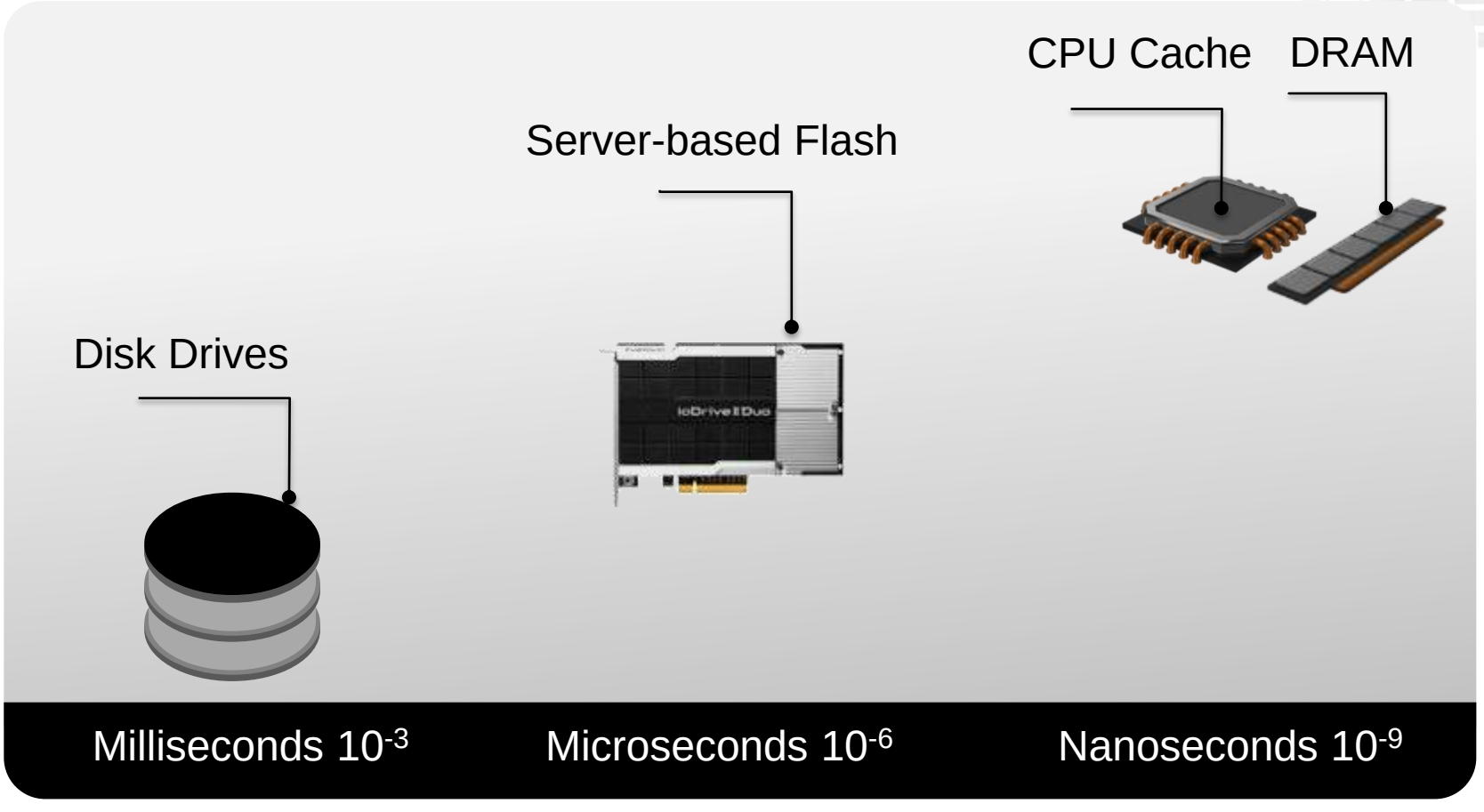
VMware ESX 4.0, 4.1
VMware ESXi 4.0, 4.1
VMware ESXi 5.0, 5.1
Windows 2008 R2 with Hyper-V

Unix

Solaris 10 x64 U8, U9, U10
OpenSolaris 2009.06 x64
OSX 10.6 and later
FreeBSD 8,9

Flash Offers A New Architectural Choice

FUSION-IO



Evolution of Flash Performance

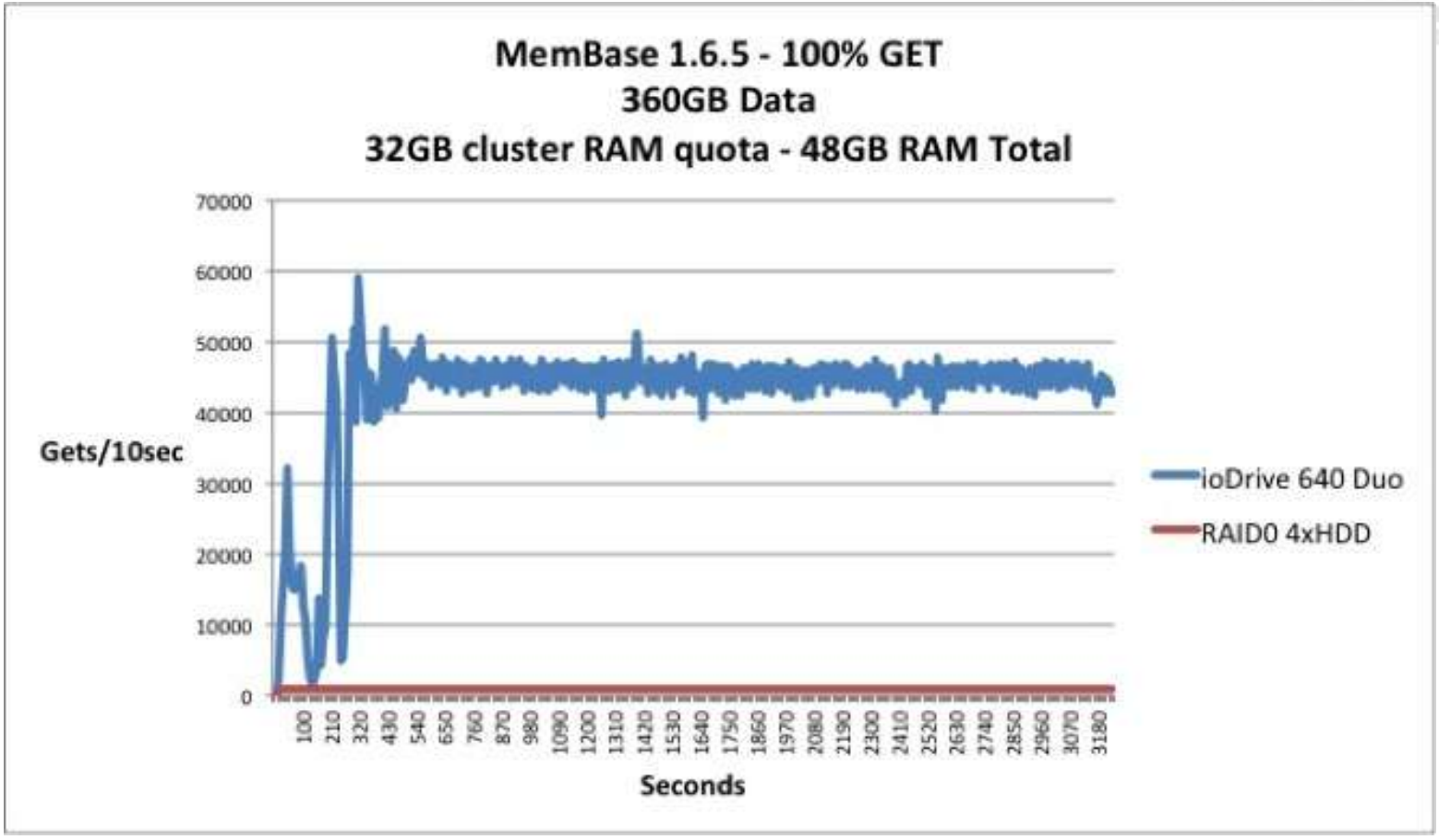
FUSION-IO

FLASH AS DISK

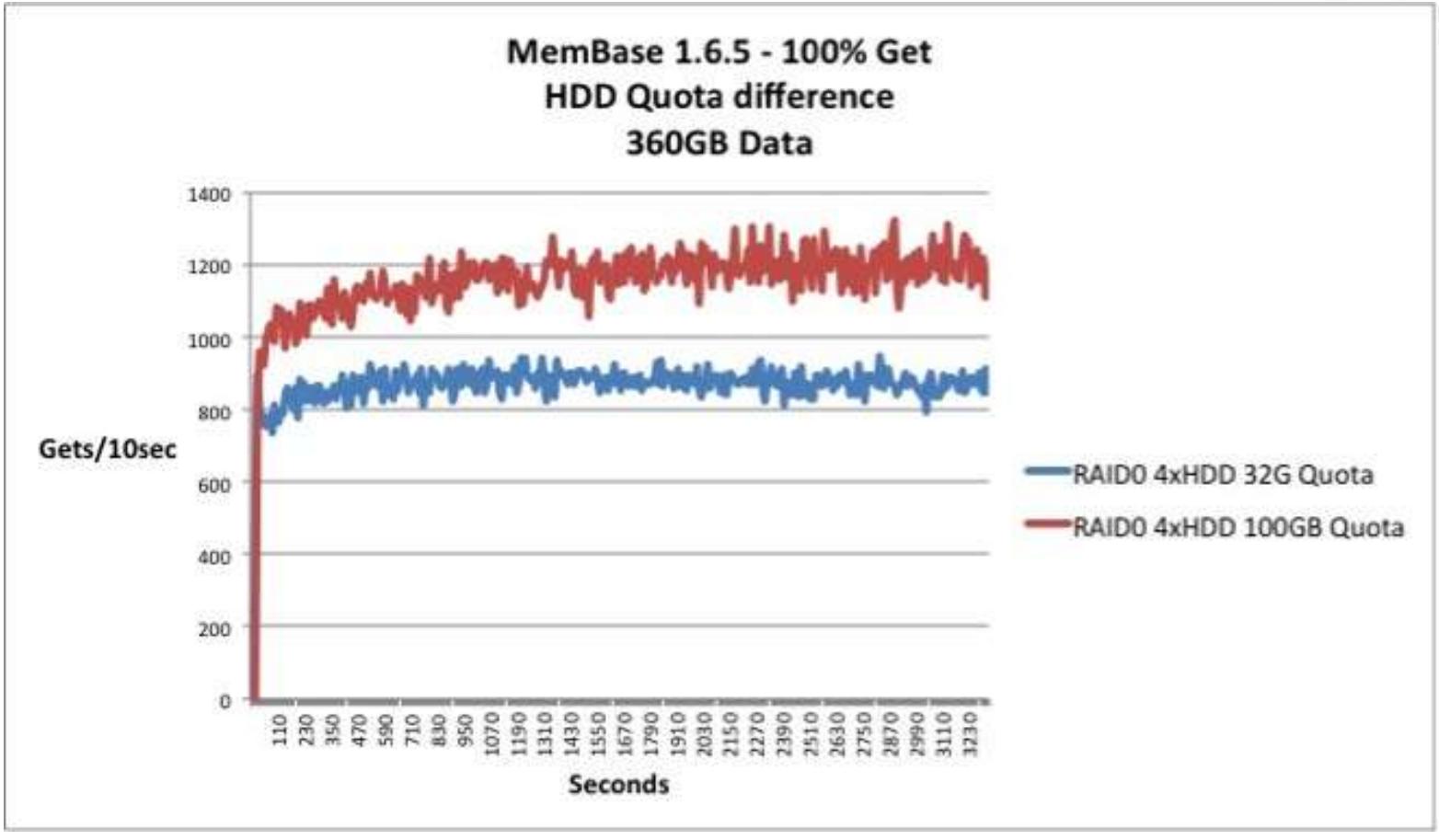
FLASH AS MEMORY

Performance

Lets look at some charts



Adding 3x the DRAM does not really improve things

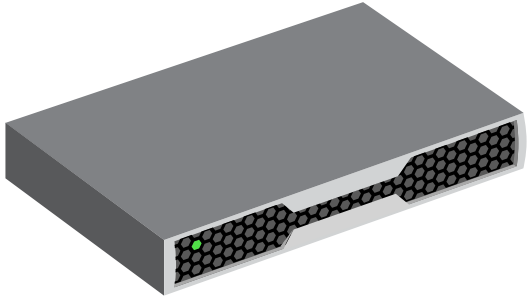


HBase Server

- A typical server…

CPU Cores: 32 with HT

Memory: 128 GB



Is your working set larger than 128GB?

HBase Cluster

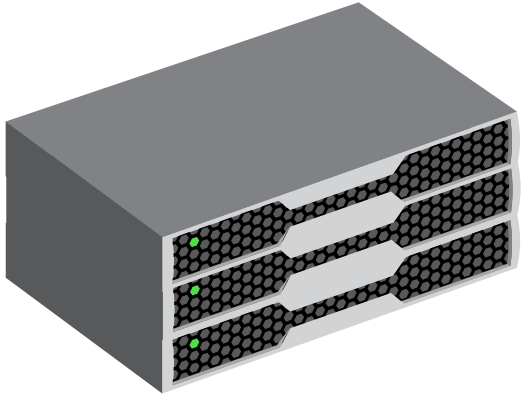
FUSION-IO

- With NoSQL Databases, we tend to scale out for DRAM

Combined Resources

CPU Cores: 96

Memory: 384 GB



More cores than needed to serve reads and writes.

The HBase BucketCache (HBase-7404)

Committed to HBase trunk. Will be in 0.96 release, backport patch for 0.94 available.`

Victim cache for LRUBlockCache – Move fast ioMemory close to DRAM cache

+

https://issues.apache.org/jira/browse/HBASE-7404

BucketCache Configuration

FUSION-IO

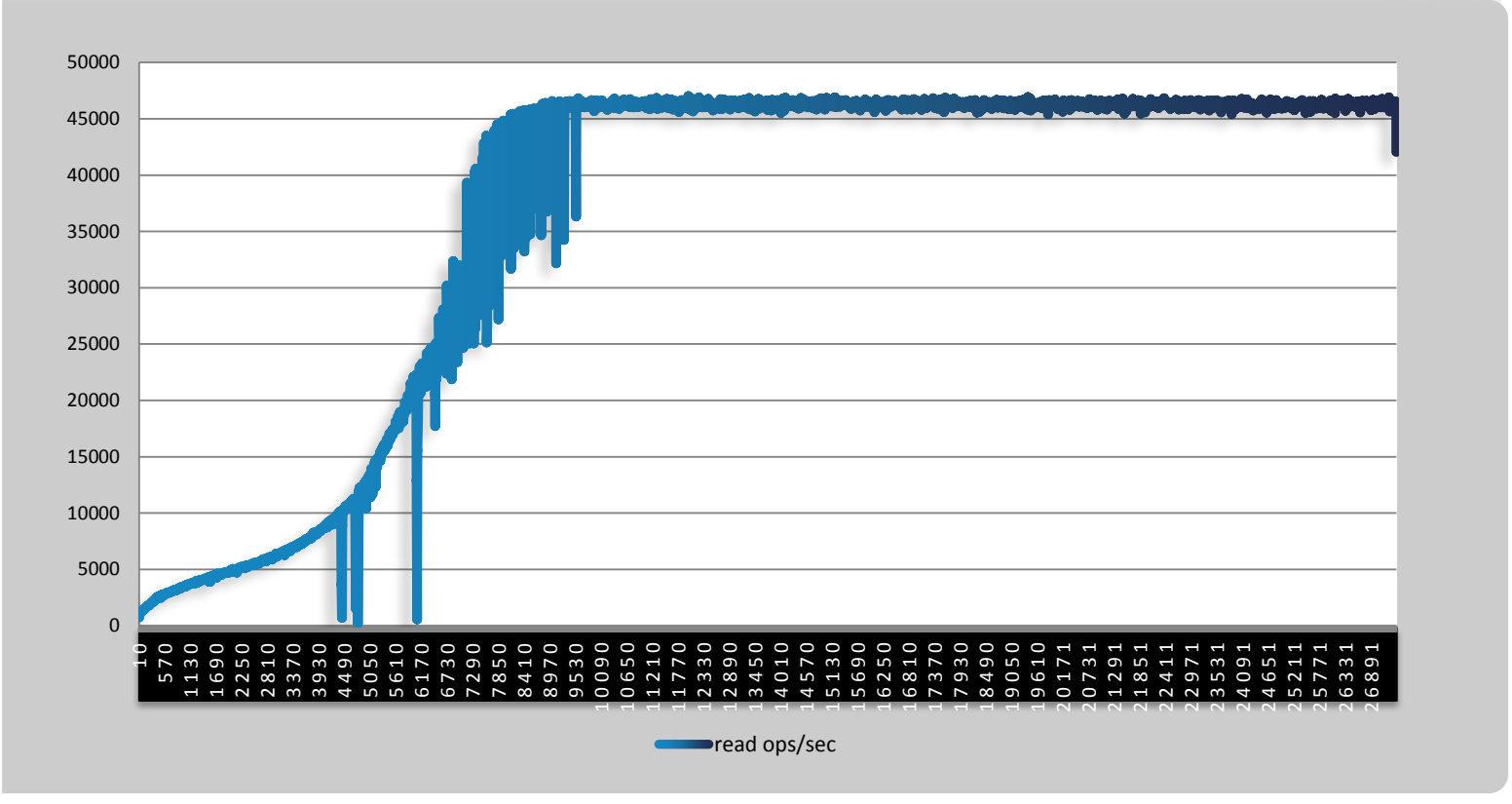
- In hbase-site.xml

```
<property>  <name>hbase.bucketcache.ioengine</name>  <value>file:/path
/to/bucketcache.dat</value>  </property>
 <property>  <name>hbase.bucketcache.size</name>  <!-- 2TB: unit is MB
-->
   <value>2097152</value>
 </property>
```

BucketCache Warm-up

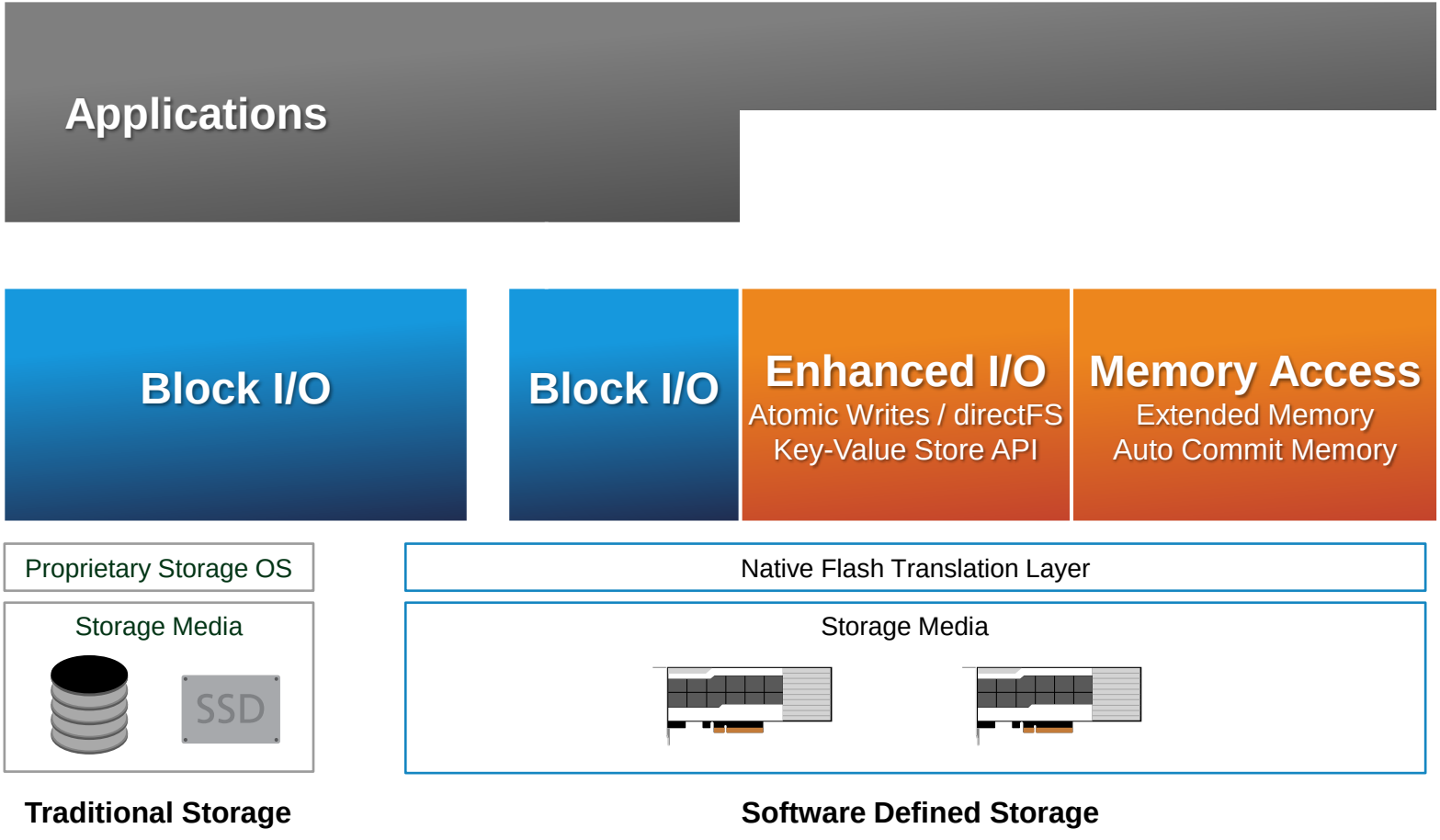
FUSION-IO

READ OPS DURING CACHE WARM-UP



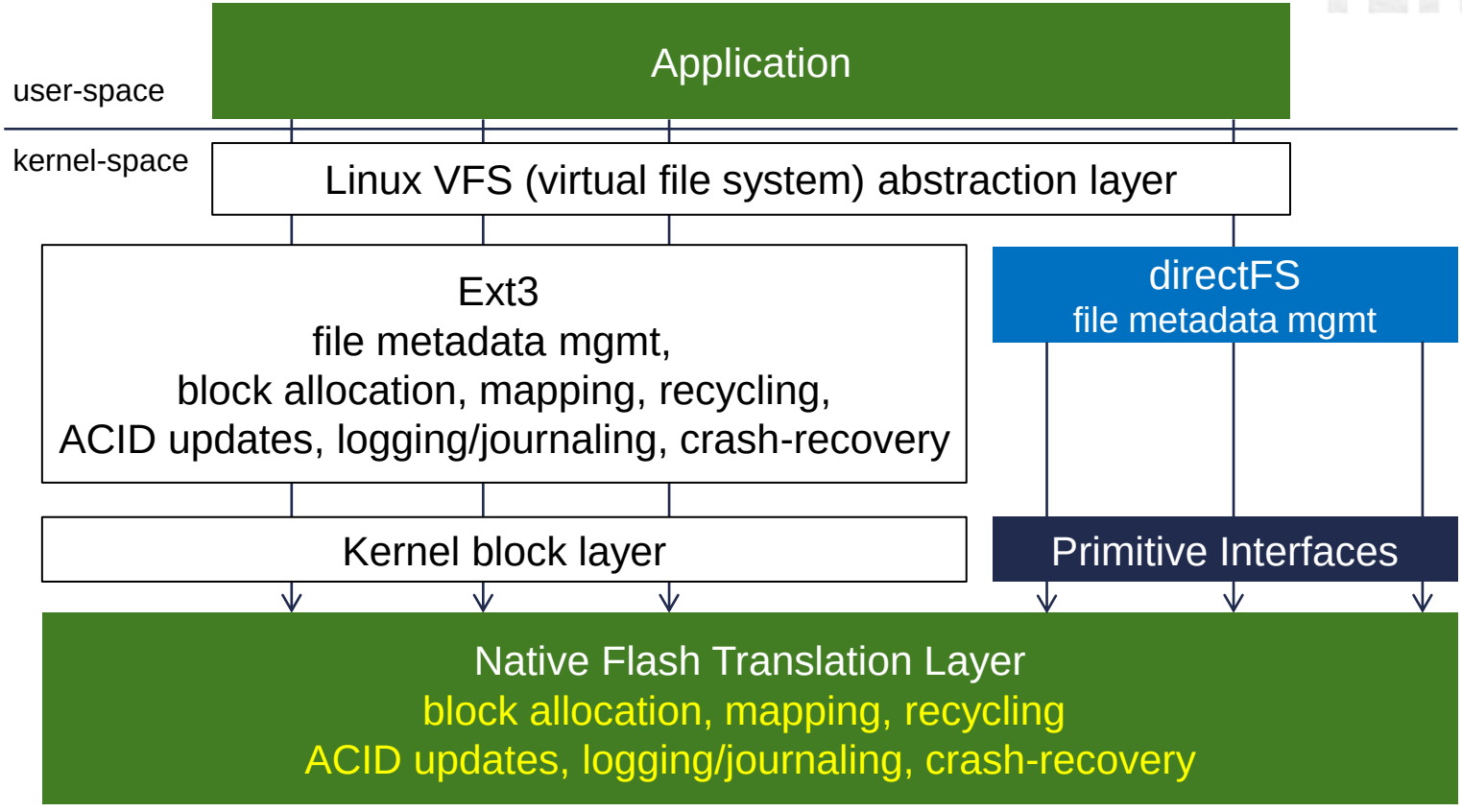
Fusion-io Software Development Kit

FUSION-IO



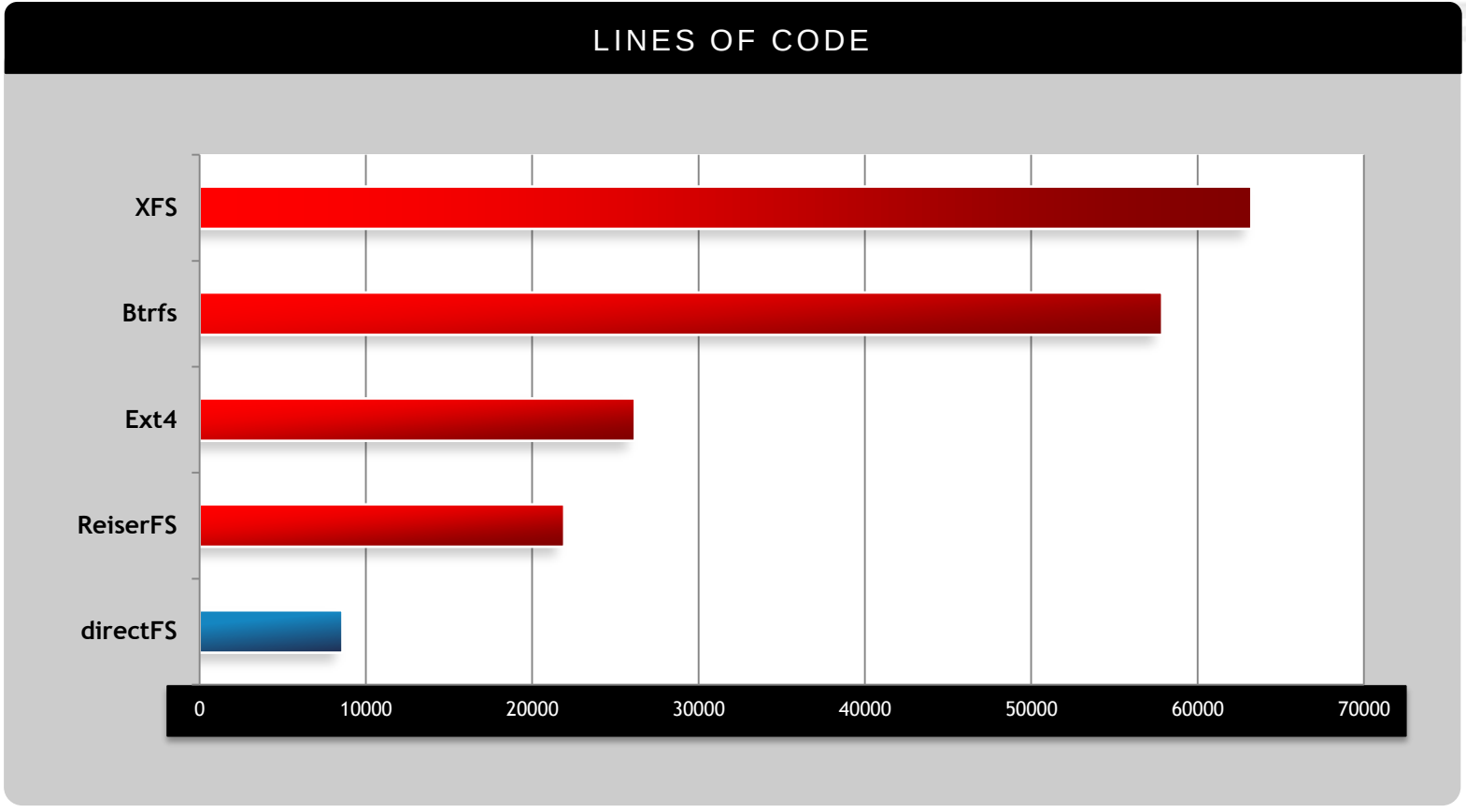
DirectFS Linux file system

FUSION-IO



directFS: Speed Through Simplicity

FUSION-IO



Atomic writes – Transactional I/O

FUSION-IO

- System call tells DirectFS that all I/O to this file should be treated as atomic
- Avoids the partial page write problem
- Accepted by T10 technical committee for SCSI standard
- Minimal application changes required

Percona Server, MariaDB, MySQL 5.6

FUSION-IO

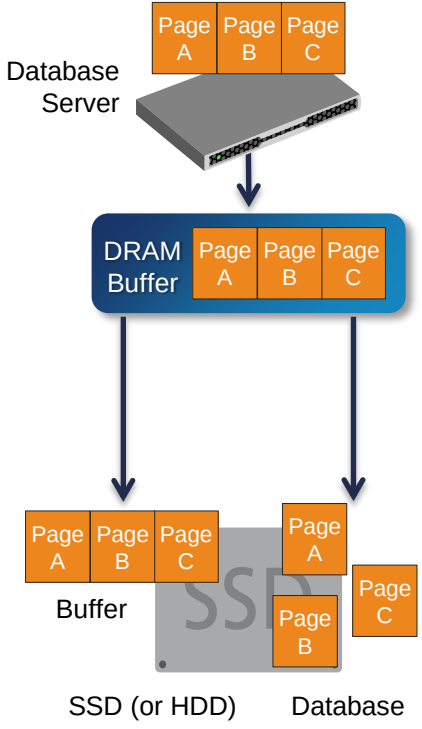
- Efficient XtraDB/InnoDB storage engine
- Well optimized for seek-less storage like flash
- Many config parameters to fine-tune performance

- What else can be done?
  - Lock contention can still be improved as seen by using multiple instances with the same storage device
  - Tapping into the native performance of flash by exposing key FTL features to the application

MySQL Writes Comparison

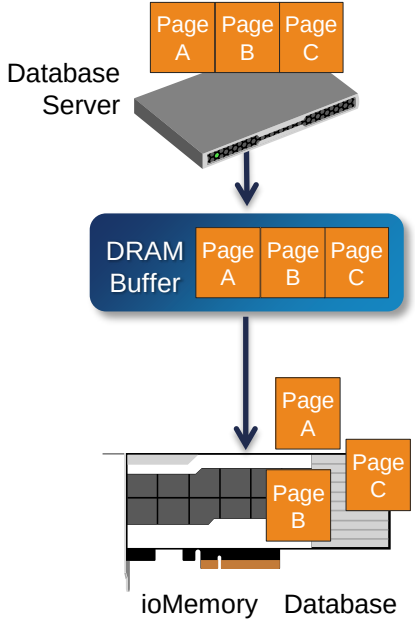
FUSION-IO

Traditional MySQL Writes



1. Application initiates updates to pages A, B, and C.
2. MySQL copies updated pages to memory buffer.
3. MySQL writes to double-write buffer on the media.
4. Once step 3 is acknowledged, MySQL writes the updates to the actual tablespace.

MySQL with Atomic Writes



1. Application initiates updates to pages A, B, and C.
2. MySQL copies updated pages to memory buffer.
3. MySQL writes to actual tablespace, bypassing the double-write buffer step due to inherent atomicity guaranteed by the intelligent device.

Atomic benchmarks

FUSION-IO

First, lets sum up the MySQL benefits here:

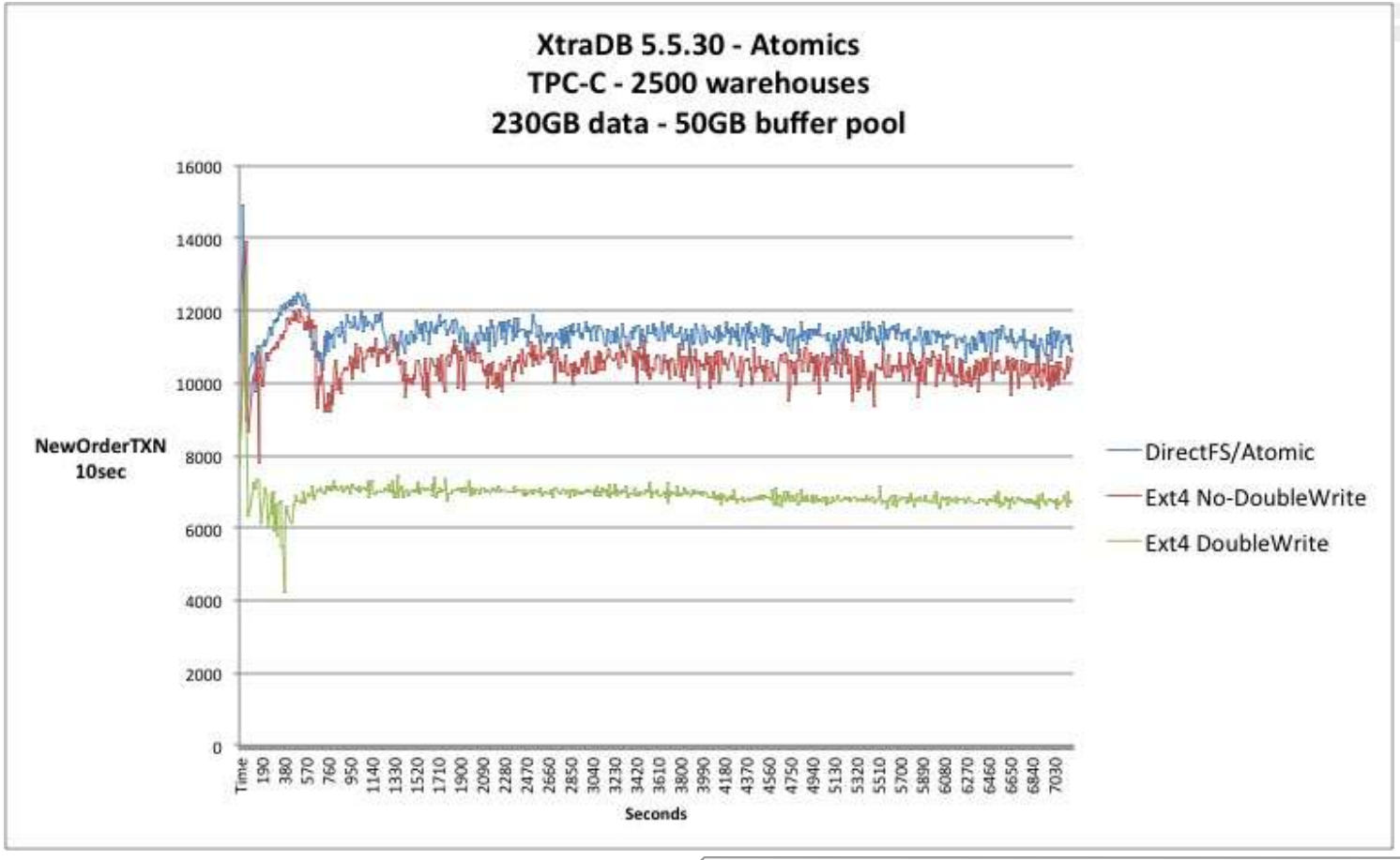
- Writing only 50% of the data otherwise required for ACID compliance

That's pretty much it…but it gives us

- Twice the flash endurance
- Much better latency because of fewer syscalls
- Much better application throughput due to less I/O
- Better concurrency due to fewer locks

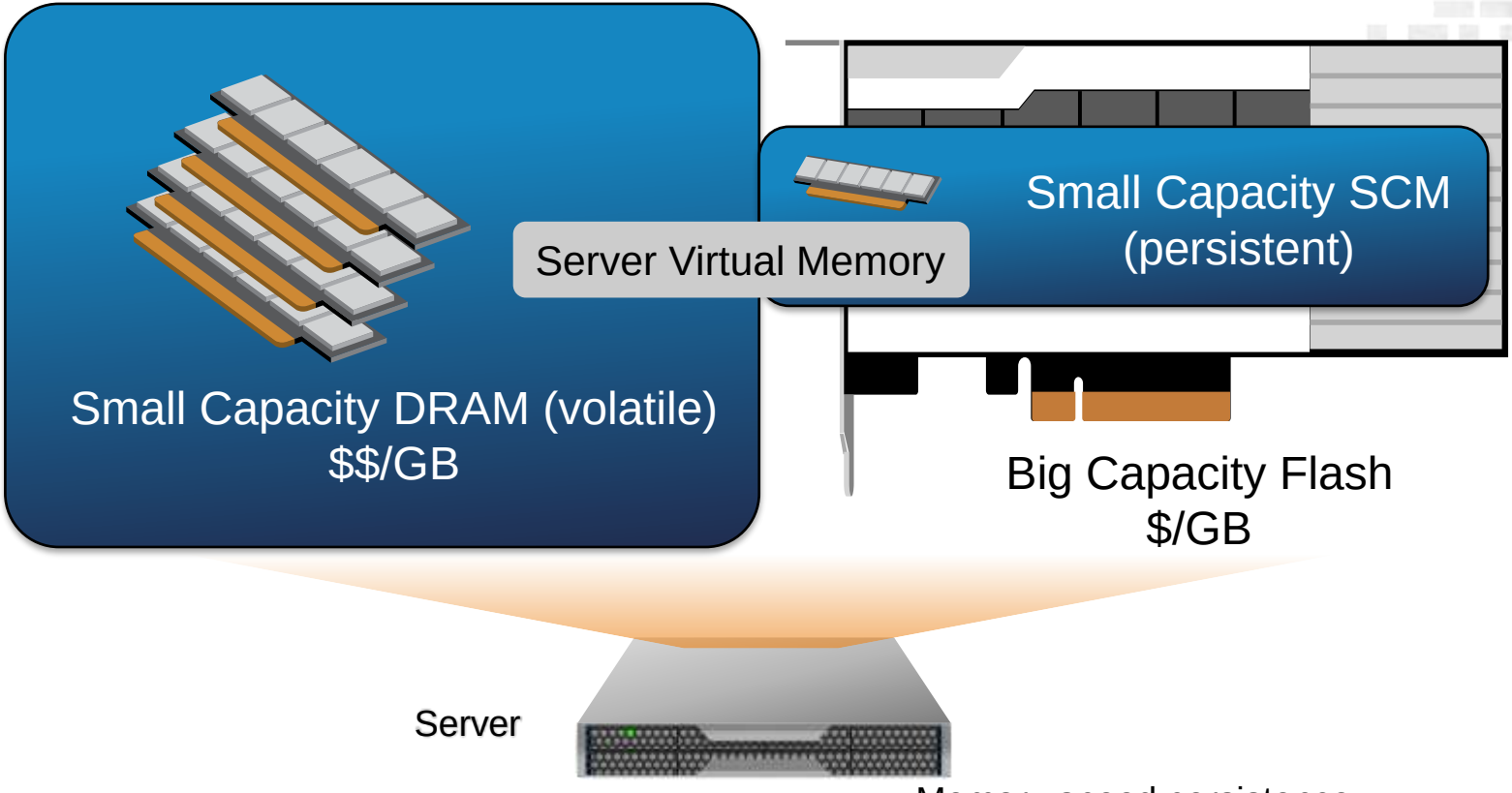
Atomics
50% more TPC-C throughput

FUSION-IO



Fusion-io advanced development
Storage Class Memory

FUSION-IO



Memory-speed persistence
Byte-addressable vs. block addressable

SCM research

- Lets look at keeping a database log using memory semantics
- Goal is to reduce latency, cost of flushing data to a persistent state and further minimize writes
- SCM testing using modified Innosim toool

SCM logger interface

- ***logger_open()***

  Open and initialize logging infrastructure within the FTL

- ***logger_close()***

  Clean-up

- ***logger_append()***

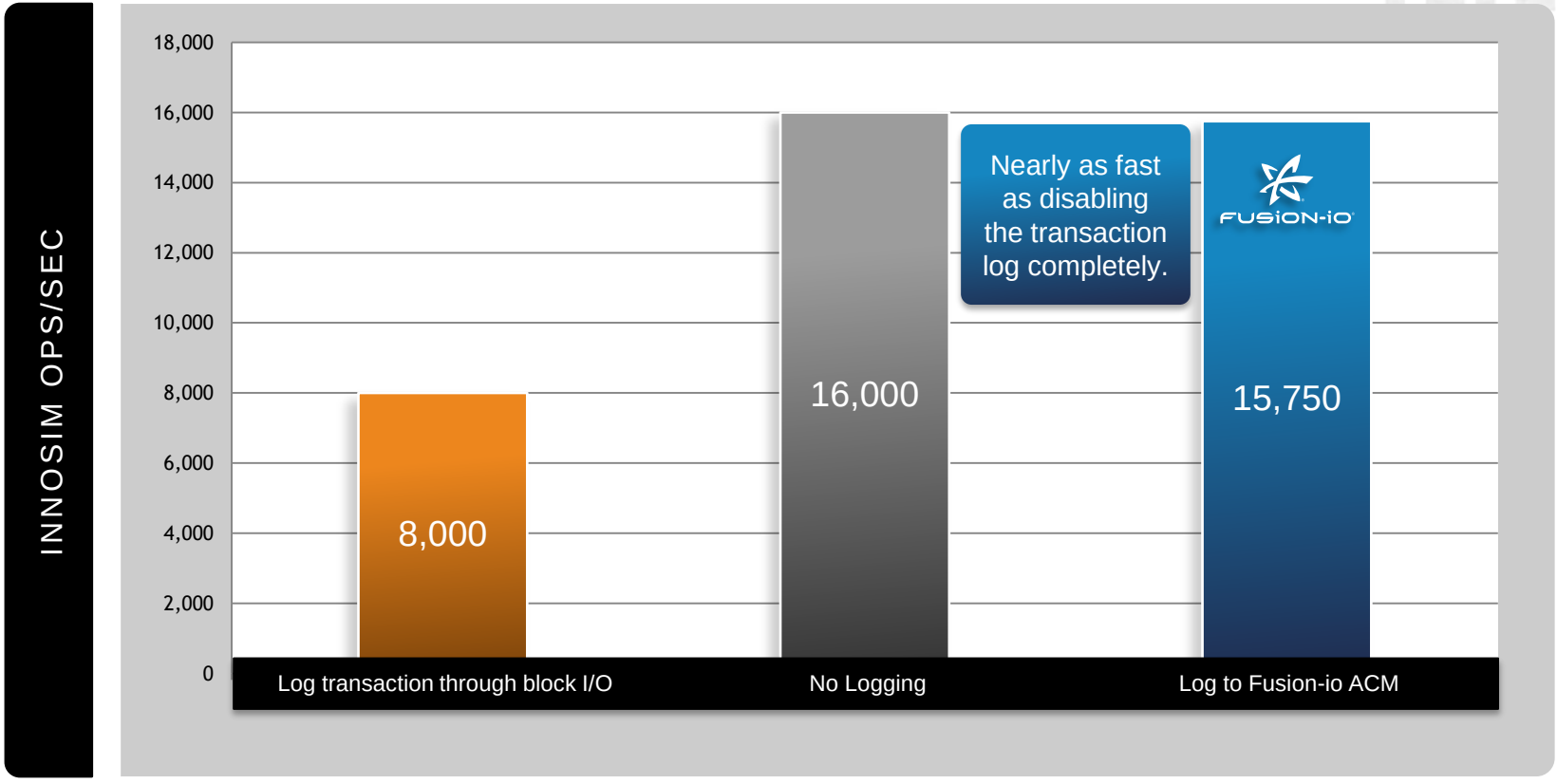
  Append to head of log at memory speeds. This basically translates to a memcpy()

- ***logger_sync()***

  Serialize data using assembler ‘mfence’ instruction

Practical Database Use Case: MySQL

FUSION-IO



The coming shift in software development

FUSION-IO

- As an SSD, flash accelerates applications.

At full maturity, Non-Volatile Memory will transform software development.

Native Flash API availability

- Percona Server: 5.5.31
- MariaDB mainline: 5.5.31
- Oracle MySQL:
  - https://code.launchpad.net/~tmathiasen/mysql-server/mysql-5.5-fio
- Cassandra atomics implementation in progress
- DirectFS public beta expected July 22nd


THANK YOU
fusionio.com | REDEFINE WHAT'S POSSIBLE