Running NoSQL natively on flash

Thomas Rochner

trochner@fusionio.com

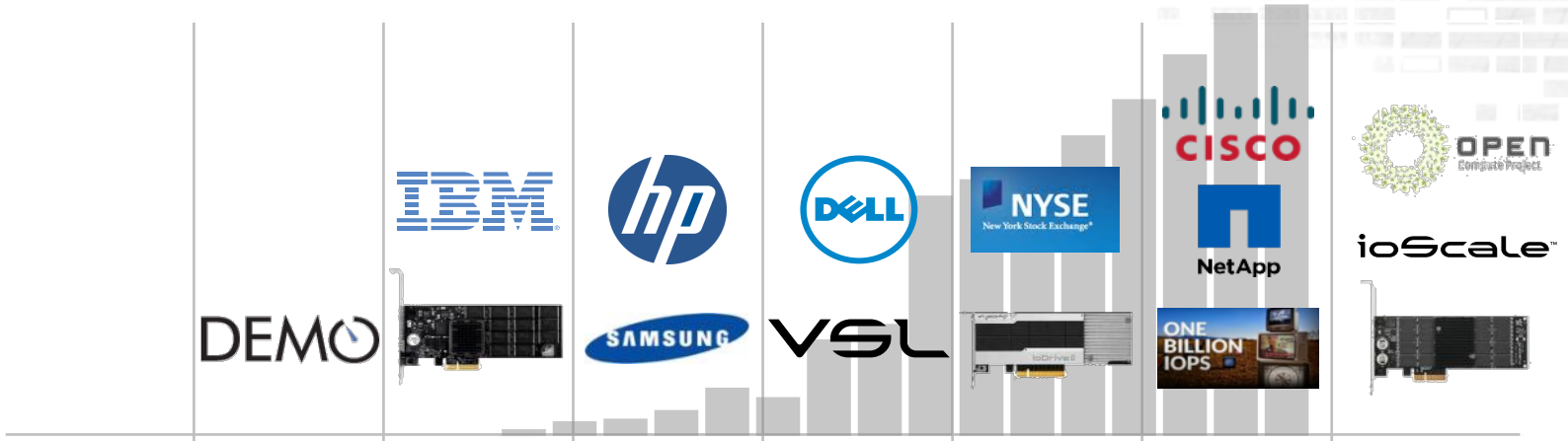
Topics – NoSQL Munich 2013

FUSION-IO

1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

Fusion-io First Mover Milestones

FUSION-IO



2006

Mission to consolidate memory and storage

2007

ioMemory technology unveiled

2008

First products launched

1 million IOPS IBM Quicksilver

Dell strategic investment

2009

HP OEMs products

IBM OEMs products

Samsung strategic investment

2010

Dell OEMs products

VSL introduced

2011

IPO on NYSE

ioTurbine acquired

ioDrive2 announced

Supermicro OEMs products

2012

1 Billion IOPS

2,500 customers

>120 channel and alliance partners

ioFX announced

ioMemory SDK introduced

Cisco OEMs products

2013

ioScale announced at Open Compute Summit

Fusion-io Accelerates

FUSION-IO

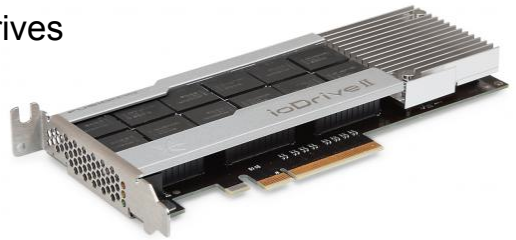
Databases	Virtualization	Search	Analytics	Big Data	Collaboration
ORACLE, MySQL, SQL Server, IBM DB2, INFORMIX, SAP, SYBASE, INGRES, PostgreSQL	vmware, Windows Server Hyper-V, XenDesktop 5, KVM	fast, Autonomy, Lucene, ORACLE Text	AccessData, MarkLogic, LexisNexis	hadoop, mongoDB	Microsoft Exchange, Microsoft SharePoint 2010, IBM Lotus

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, Adobe	PERFORCE	Powered by Solid, VARNISH SOFTWARE	ArcSight, splunk>	LAMP, Microsoft .NET

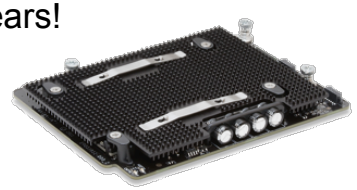
What is Fusion-io?

FUSION-IO

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

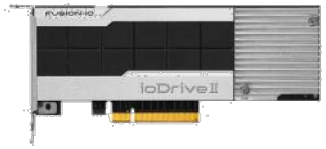


- **PCIe based NAND Flash storage**
  - Micro-second level Disk Access Latency - 15µs
  - Very high data throughput - 1,5GB/s
  - Very high IOPS – 535.000 – 800.000 random write/s
  - Scalable – stay ahead of data / performance demands
  - Advanced wear-leveling algorithm
  - N+1 Chip level redundancy (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!



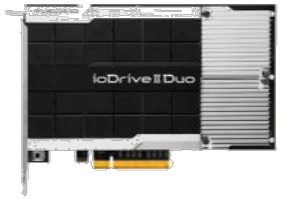
Direct Acceleration

FUSION-IO



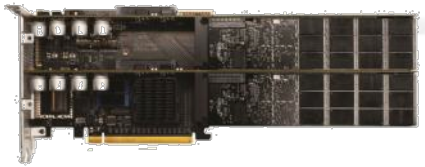
ioDrive II

Up to 3 TB of capacity
per x4 PCI Express slot



ioDrive II Duo

Up to 2.4TB of capacity
per x8 PCI Express slot



ioDrive Octal

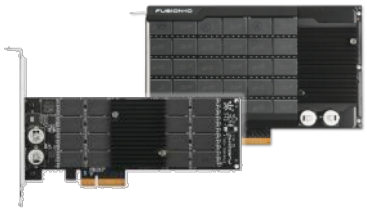
Up to 10.24TB to maximize
performance for large data sets

ioFX

1650 GB of workstation
acceleration

ioDrive II
MEZZANINE

Up to 1.2TB for maximum
performance density



ioScale™

Up to 3.2 of Hyperscale
Acceleration

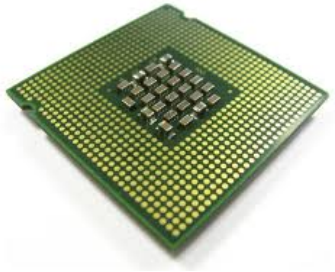
Topics – NoSQL Munich 2013

FUSION-IO

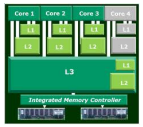
1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

The landscape of sub second Timings.
How fast do you get data to the factory?

FUSION-IO



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 Coffee
2,5 minutes

Fly to Australia
11,1 hours

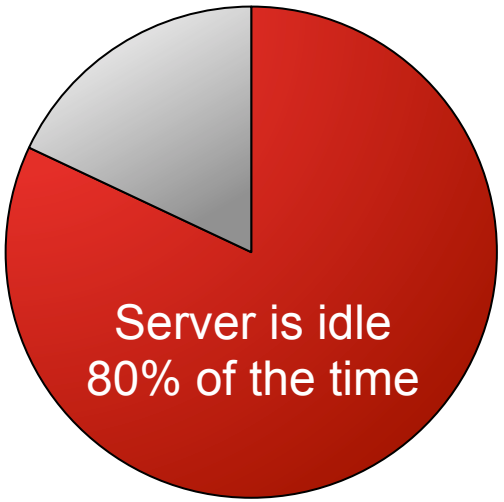
Multiplier is 10m - 10.000.000

SLOW STORAGE LEADS TO IDLE CAPACITY

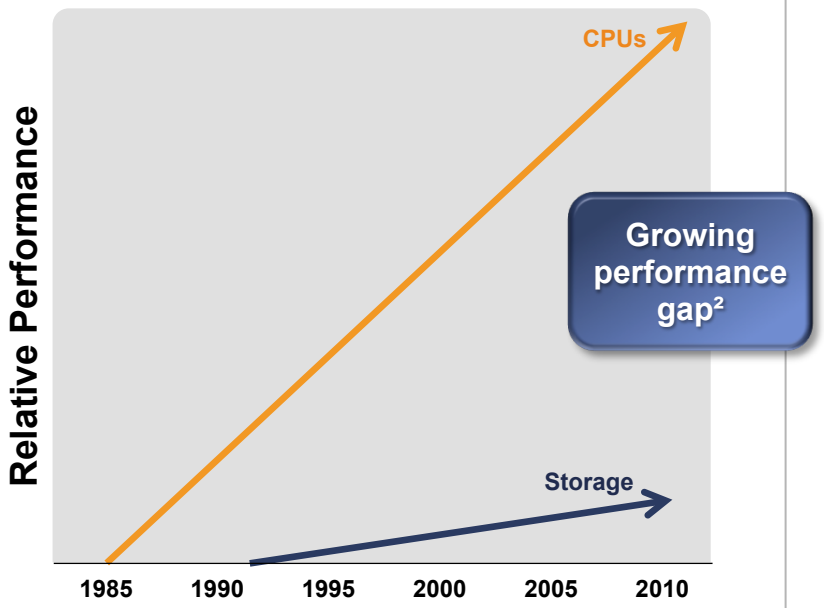
FUSION-IO

According to Moore's Law, processing performance doubles every 18 months

37% of servers are massively underutilized[1]…



…because the performance gap continues to grow



[1] *Source: IDC's Server Workloads 2010, July 2010*
[2] *Source: Taming the Power Hungry Data Center, Fusion-io White Paper*


SPINNING MEDIA
OVER 150 YEARS OLD

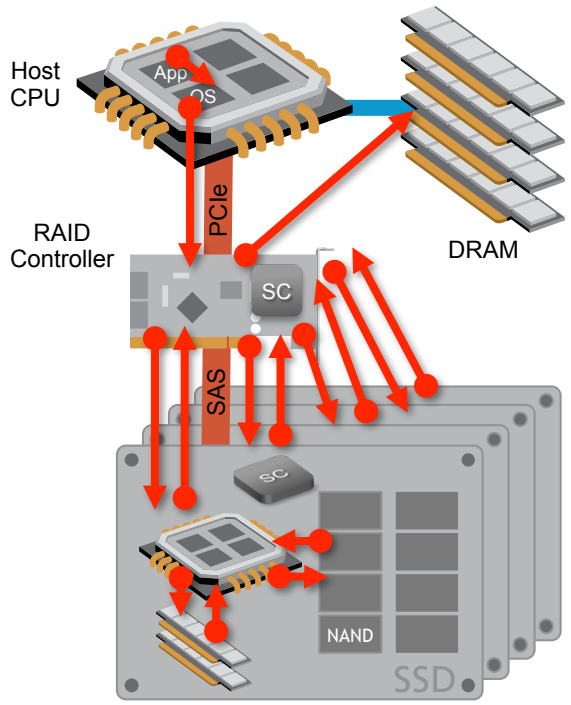
SSD treats memory like disk

FUSION-IO

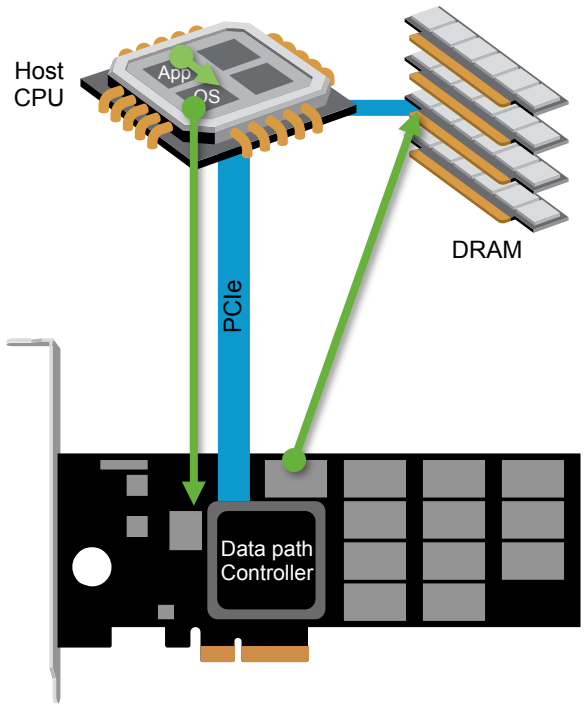
Flash Architectures

FUSION-IO

FLASH AS DISK



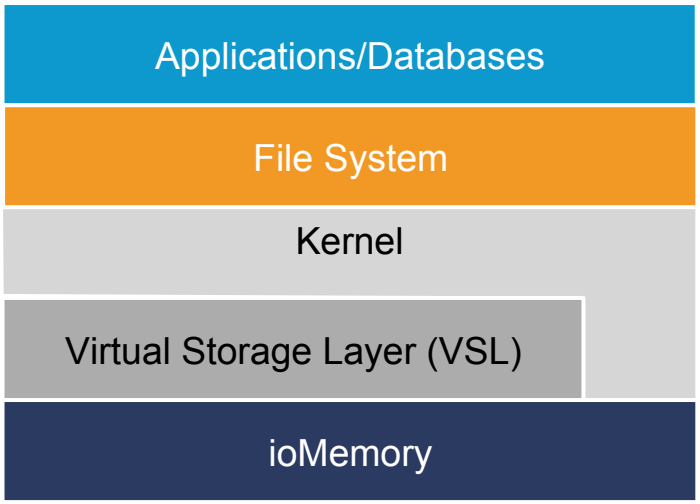
FLASH AS MEMORY

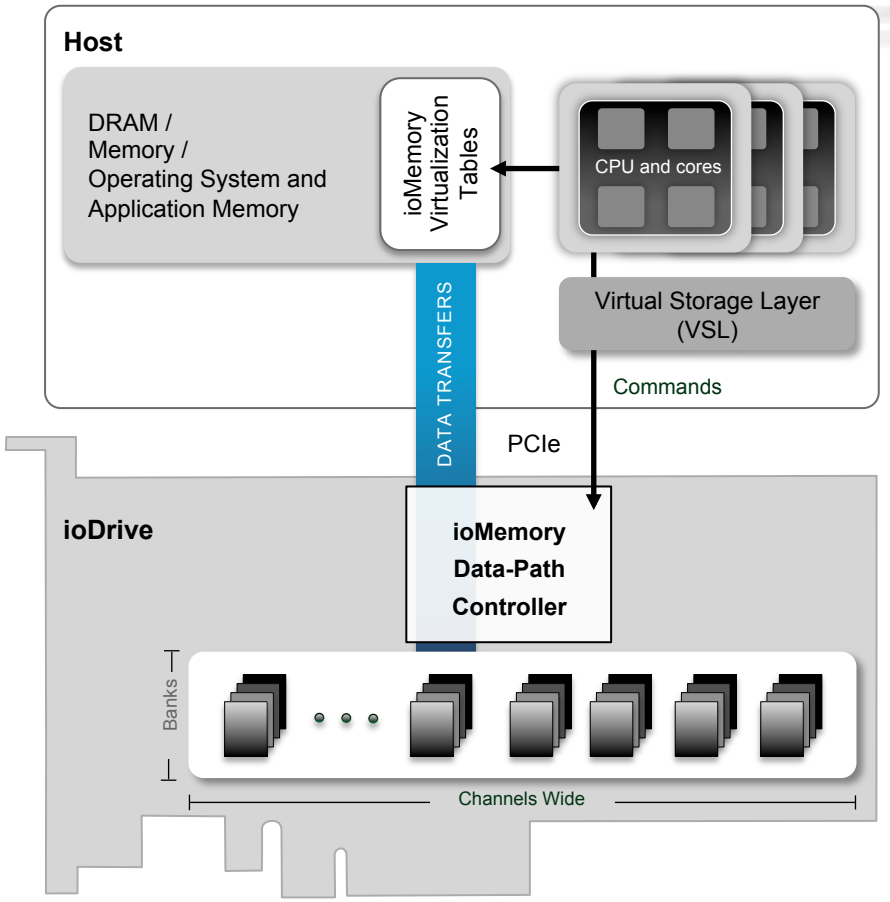


Cut-through Architecture and VSL

FUSION-IO

- Sophisticated architecture
  - maximum performance
- Intelligent software
  - advanced features

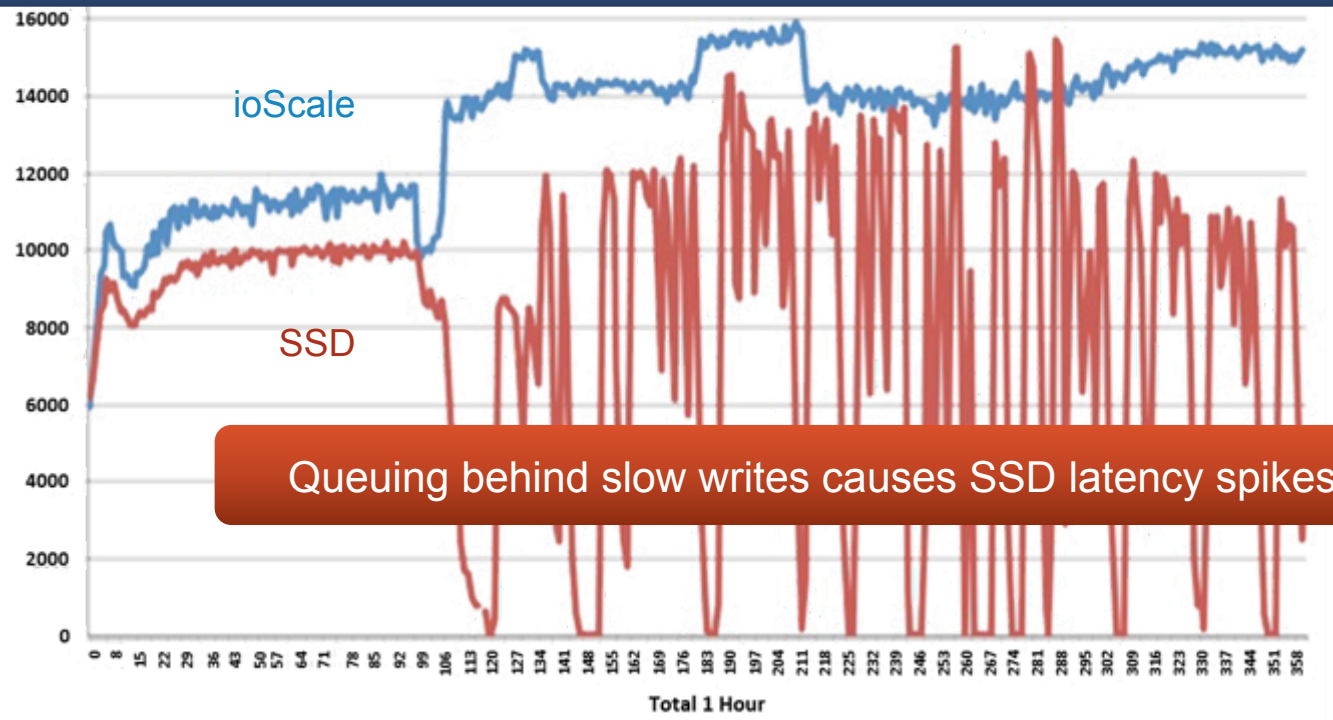




Balanced Performance Affects Throughput

FUSION-IO

ioMemory balances read/write performance for consistent throughput



Queuing behind slow writes causes SSD latency spikes

Topics – NoSQL Munich 2013

FUSION-IO

1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

NoSQL Software challenges

FUSION-IO

- Keeping NoSQL software simplicity with data persistence
- Transforming in-memory structures to block I/O
- Tiering data between DRAM and persistent storage
- Keeping latency low with data persistence
- Scaling up

ioMemory Software Development Kit

FUSION-IO

- Native programming interfaces
- Access flash as a memory
- Eliminate legacy software layers
- Simplify application authoring
- Accelerate time-to-market

NVM Software interfaces

FUSION-IO

- Industry-first, direct API access to non-volatile memory's unique characteristics.

- The ioMemory SDK was introduced to help developers:

- **Write less code** to create high-performing apps
- **Tap into performance** not available with conventional I/O access to SSDs
- **Reduce operating costs** by decreasing RAM while increasing NVM

Direct-Access to non-volatile memory is now emerging

FUSION-IO

- Developers are beginning to manipulate data

directly in Non-Volatile Memory (NVM)

without converting to basic block I/O.

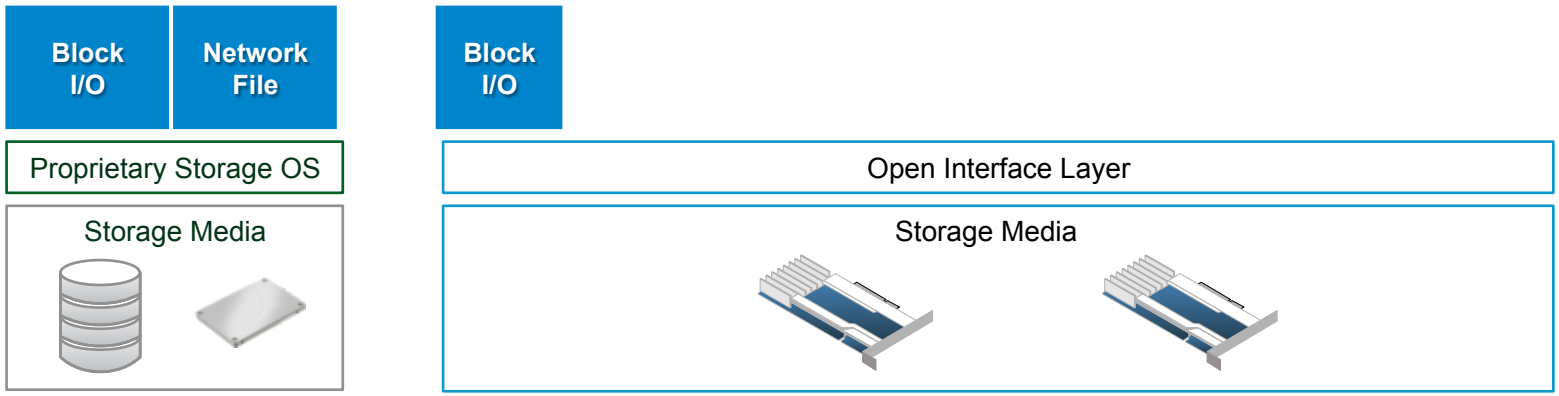
Conventional I/O access

FUSION-IO

APPLICATION

Application source code translates native data structures into block I/O

Conventional I/O access

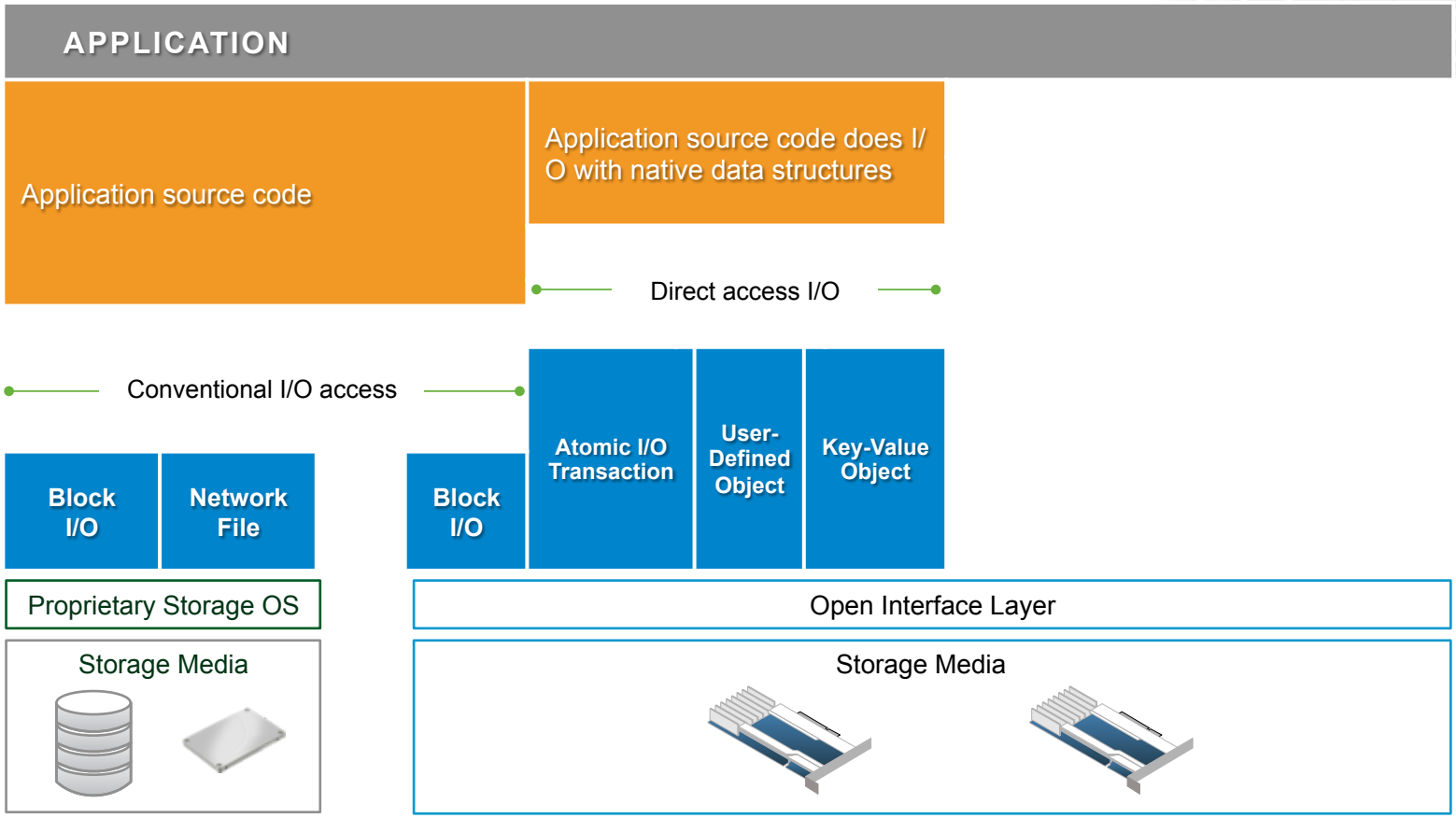


Traditional Storage

Software Defined Storage

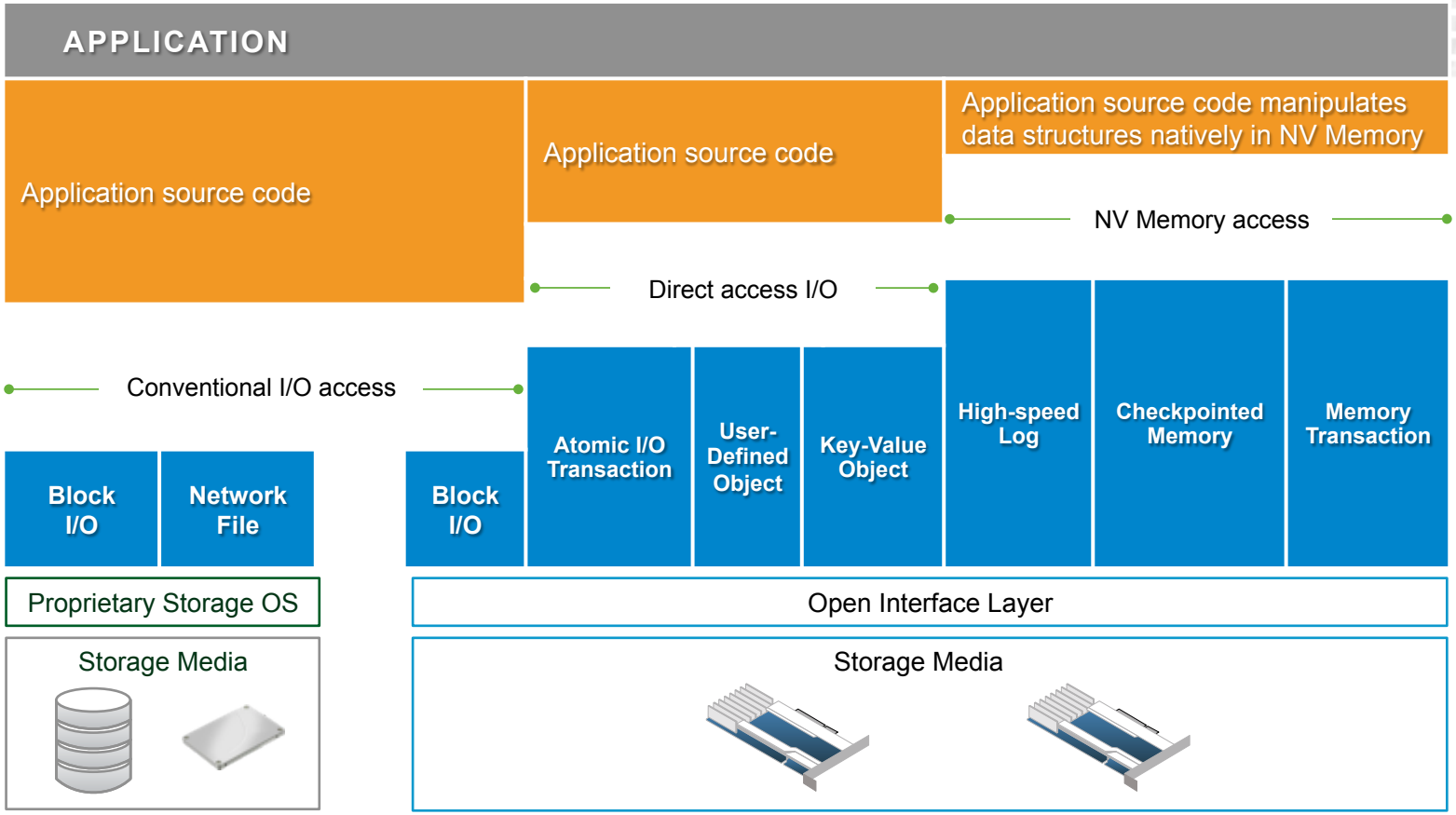
native nvm access: I/O

FUSION-IO



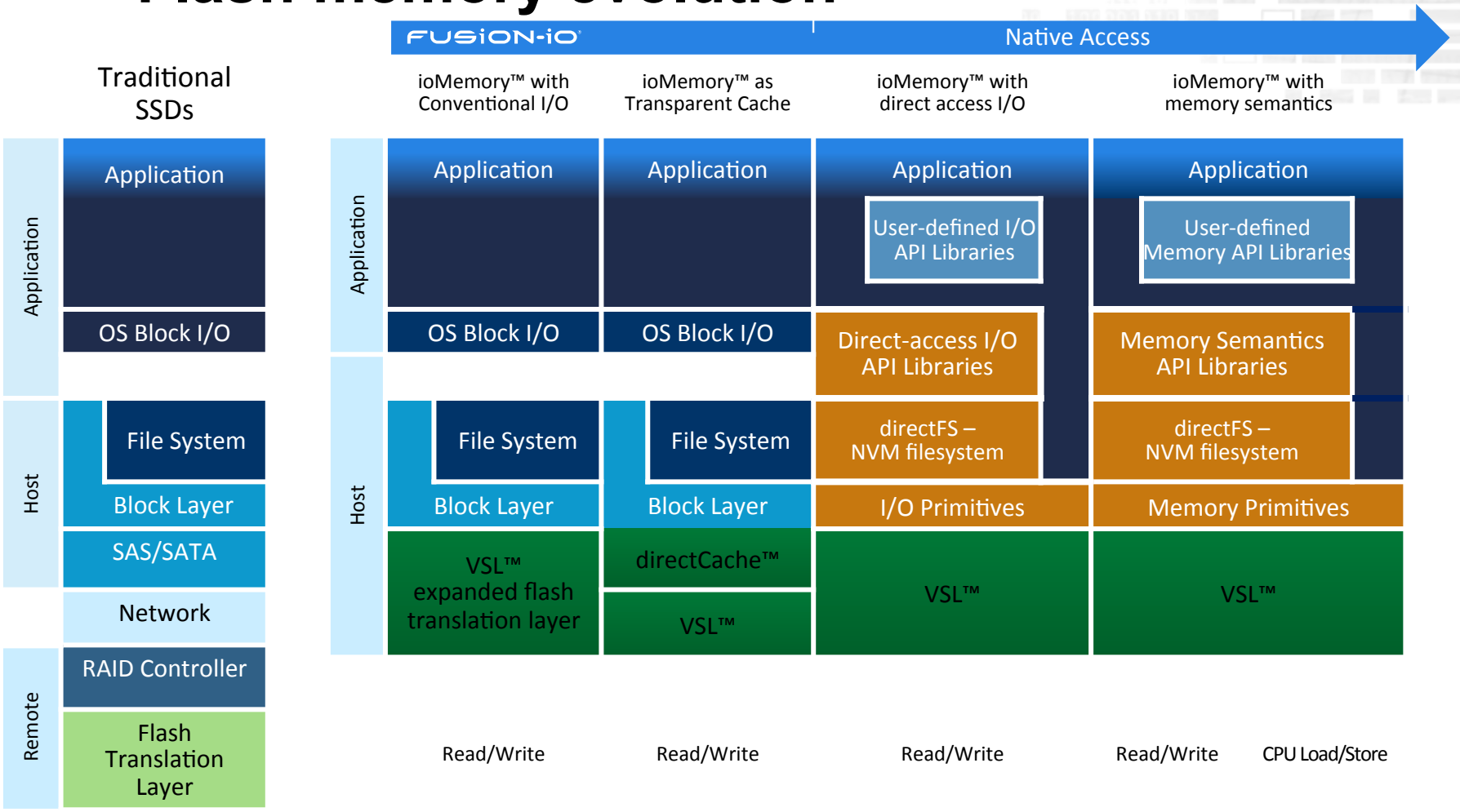
native nvm access: persistent memory

FUSION-IO



Flash memory evolution

FUSION-IO

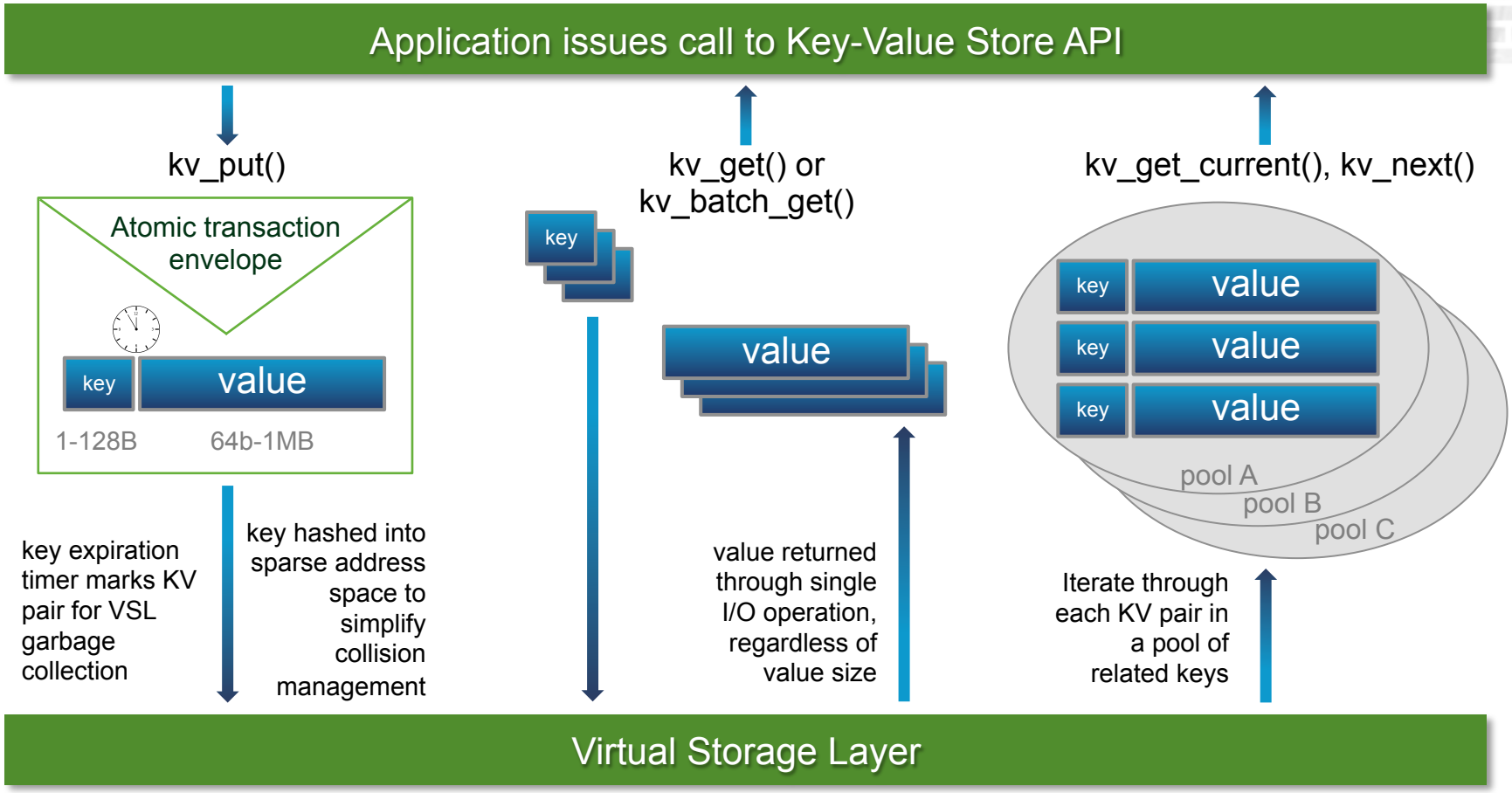


Topics – NoSQL Munich 2013

FUSION-IO

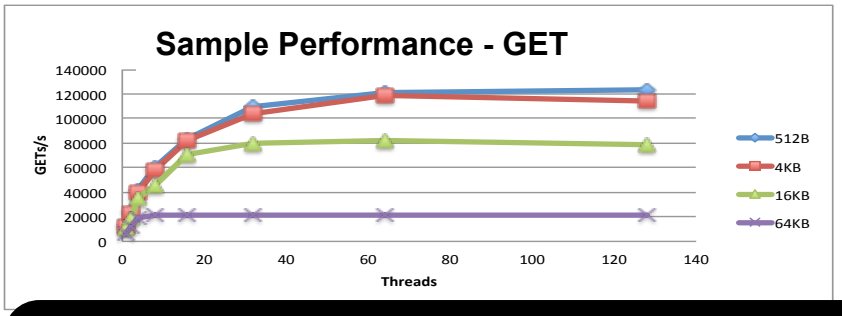
1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

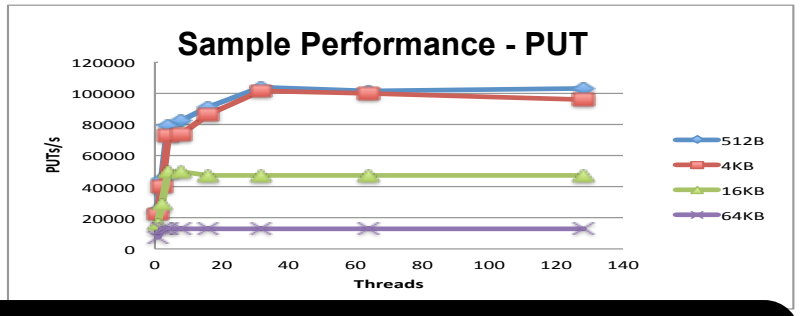
Example: Key-Value Store API Library



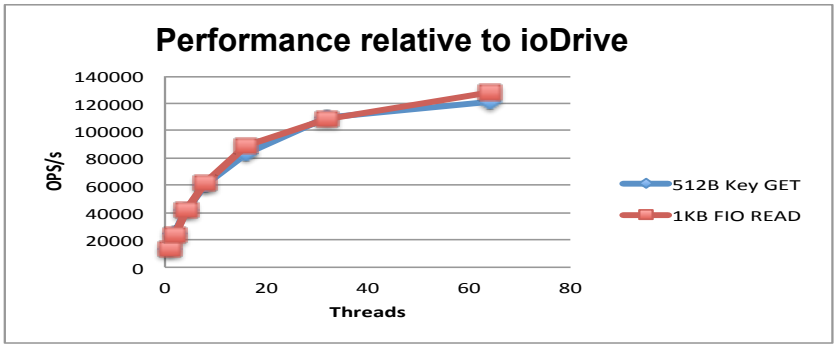
Key-Value Store API Library Benchmarks
(native KV Get/Put vs. raw reads/writes)

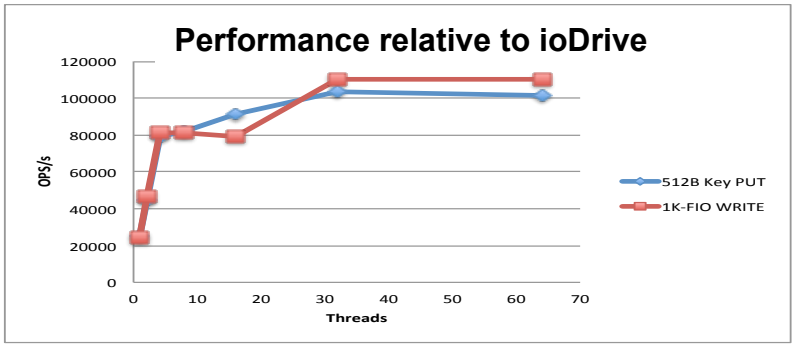
FUSION-IO





SIGNIFICANTLY MORE FUNCTIONALITY WITH NEGLIGIBLE PERFORMANCE COST





1U HP blade server with 16 GB RAM, 8 CPU cores - Intel(R) Xeon(R) CPU X5472 @ 3.00GHz with single 1.2 TB ioDrive2 mono

Key-Value Store API library Benchmarks: (vs memcachedb)

FUSION-IO

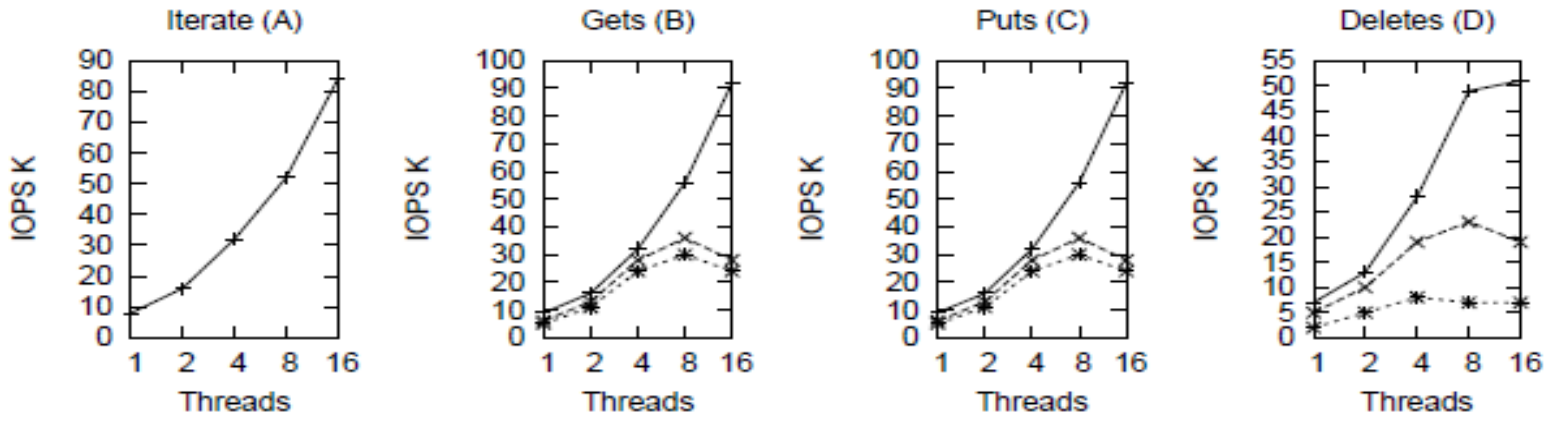
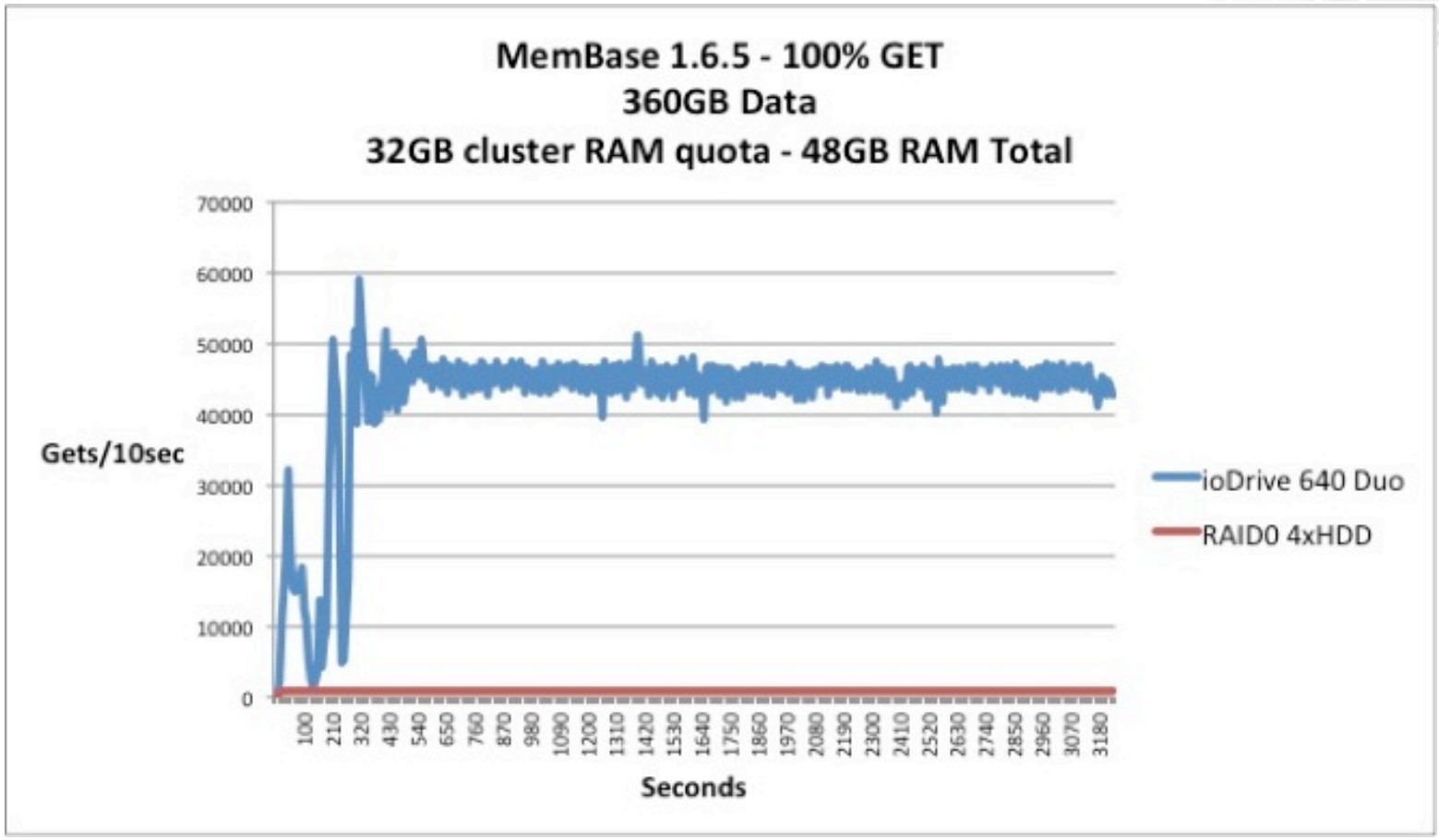


Figure 5: Performance comparison of basic operations between DirectKV and Memcachedb.
DKV: DirectKV, MDB-A: Memcachedb Async, MDB-S: Memcachedb Sync

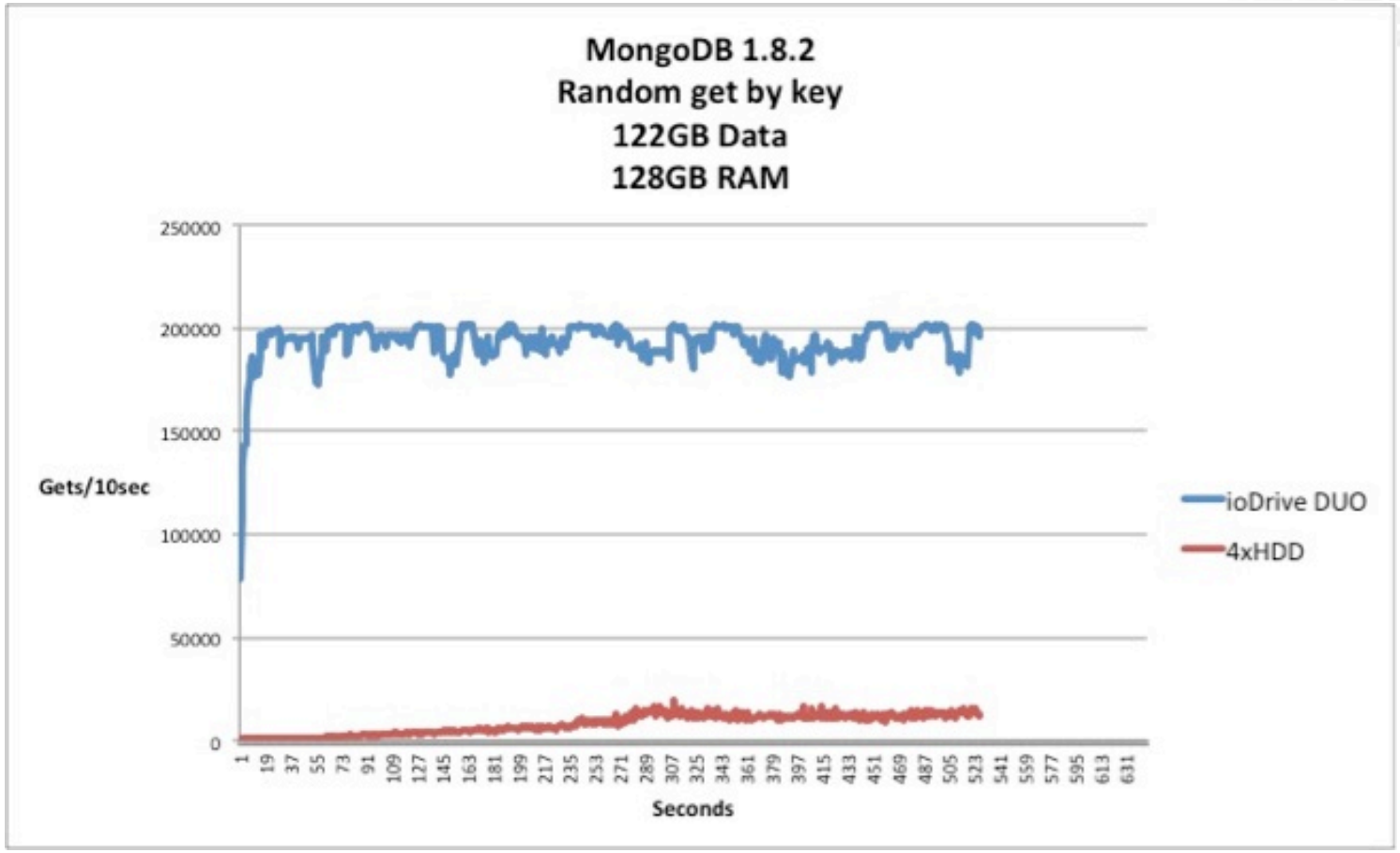
Membase ioDrive vs. HDD

FUSION-IO



MongoDB cache warmup

FUSION-IO



Key-value store API Library: Sample Uses and Benefits

FUSION-IO

NoSQL Applications

Increase performance by eliminating packing and unpacking blocks, defragmentation, and duplicate metadata at app layer.

Reduce application I/O through batched put and get operations.

Reduce overprovisioning due to lack of coordination between two-layers of garbage collection (application-layer and flash-layer). Some top NoSQL applications recommend over-provisioning by 3x due to this.

- **95% performance of raw device**
  Smarter media now natively understands a key-value I/O interface with lock-free updates, crash recovery, and no additional metadata overhead.
- **Up to 3x capacity increase**
  Dramatically reduces over-provisioning with coordinated garbage collection and automated key expiry.
- **3x throughput on same SSD**
  Early benchmarks comparing against memcached with BerkeleyDB persistence show up to 3x improvement.

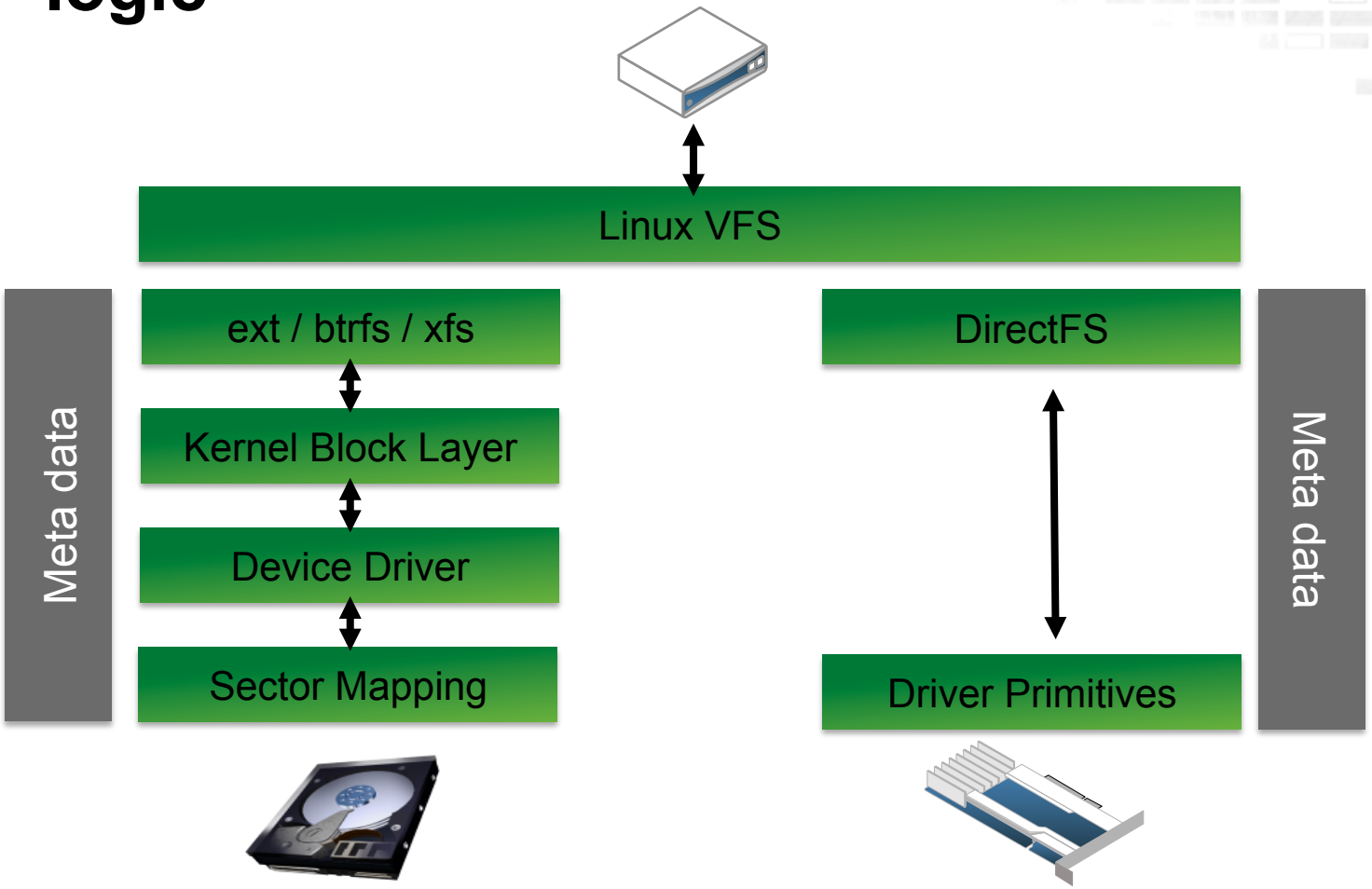
Topics – NoSQL Munich 2013

FUSION-IO

1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

directFS – Eliminating duplicate logic

FUSION-IO



DIRECTFS – Benefits in Eliminating Duplicate logic

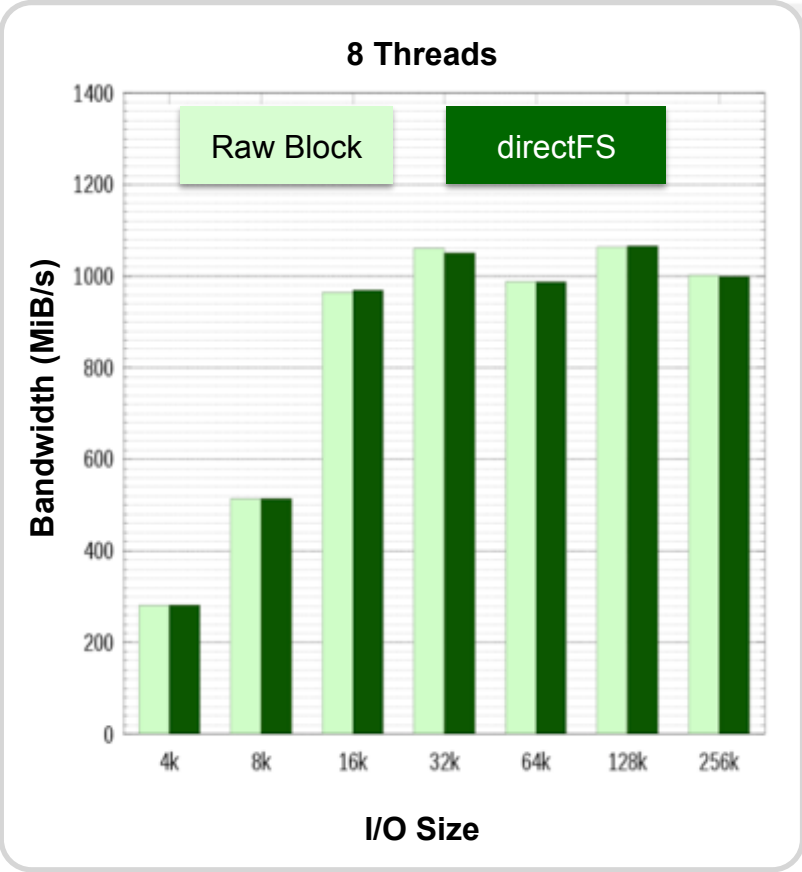
FUSION-IO

File System	Lines of Code
directFS	6879
ReiserFS	19996
ext4	25837
btrfs	51925
XFS	63230

directFS: Native Flash Filesystem

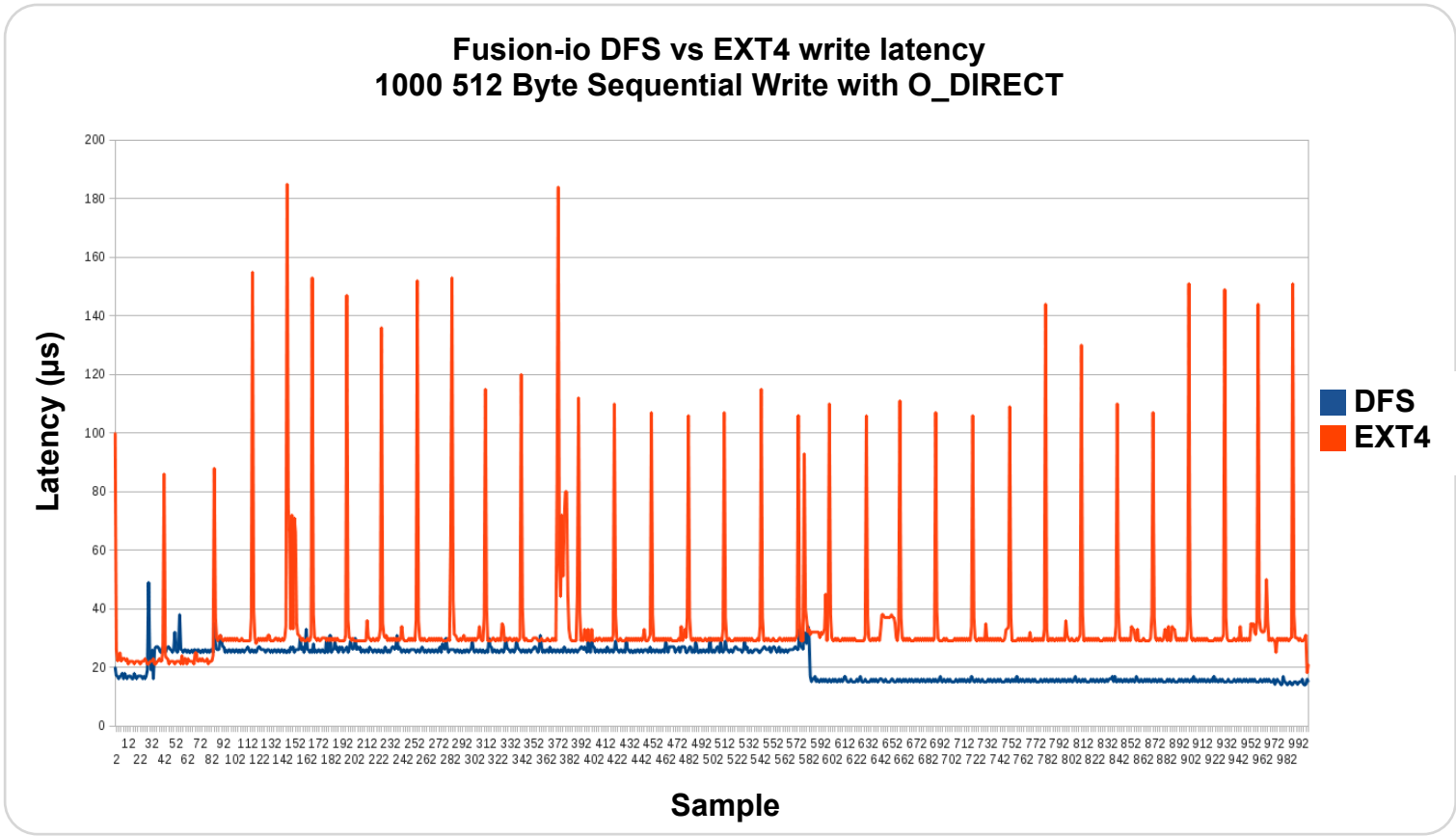
FUSION-IO

- File system convenience
- Raw block performance
- No compromise necessary



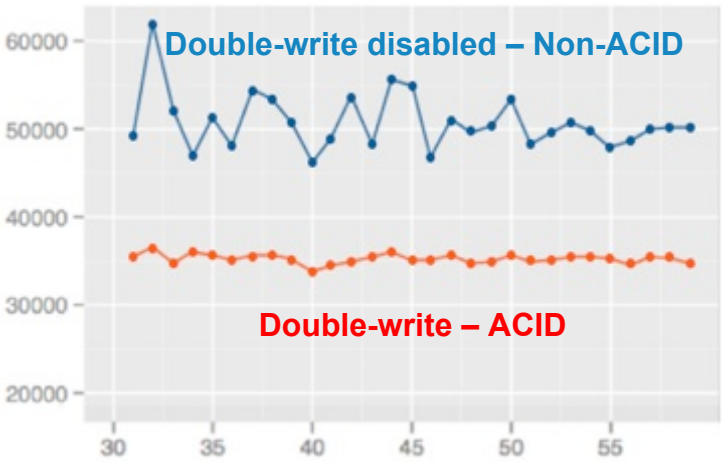
directFS: Consistent Low Latency

FUSION-IO

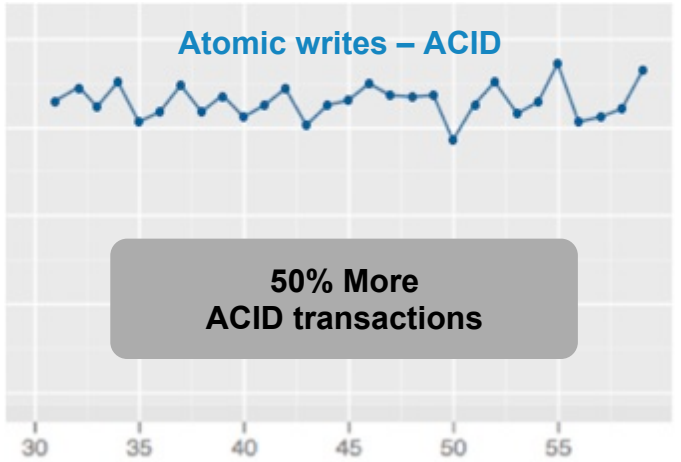


MySQL: directFS and Atomic Writes

FUSION-IO



XFS



directFS with Atomic I/O

Case Study: Percona SERVER (MySQL)

FUSION-IO

- Percona has added atomics support to Percona Server 5.5
  - Removes the need of the MySQL double write buffer
  - Ensures data integrity in case of system crashes
  - Writes 50% less, great for flash
  - Removes complexity from the software stack
  - Improves both transaction bandwidth and latency
  - Works though the DirectFS filesystem or on RAW devices

Topics – NoSQL Munich 2013

FUSION-IO

1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

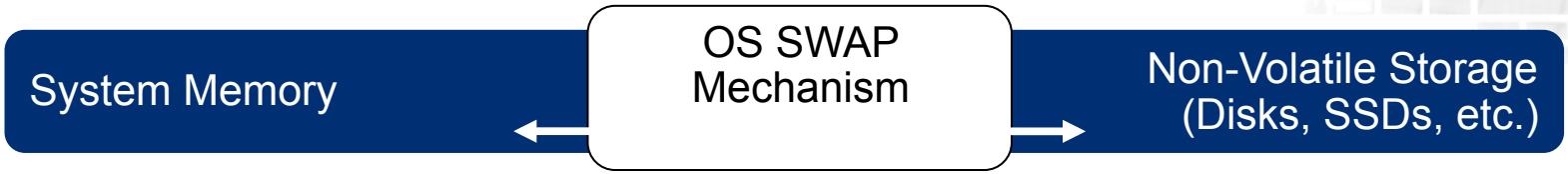
Range of memory-Access Semantics

FUSION-IO

Extended Memory	**Volatile**	Transparently extends DRAM onto flash, extending application virtual memory
Checkpointed Memory	Volatile with non-volatile checkpoints	Region of application virtual memory with ability to preserve snapshots to flash namespace
Auto-Commit Memory™	**Non-volatile**	Region of application memory automatically persisted to non-volatile memory and recoverable post-system failure

OS Swap vs. Extended Memory

FUSION-IO



- Originally designed as a last resort to prevent OOM (out-of-memory) failures
- Never tuned for high-performance demand-paging
- Never tuned for multi-threaded apps
- Poor performance, ex. < 30 MB/sec throughput

- No application code changes required
- Designed to migrate hot pages to DRAM and cold pages to ioMemory
- Tuned to run natively on flash (leverages native characteristics)
- Tuned for multi-threaded apps
- 10-15x throughput improvement over standard OS Swap

Example: Memory transaction primitives

FUSION-IO

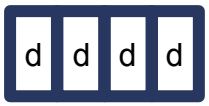
Application issues calls to transactional memory interface

1. Allocate pages
app virtual address space

Process allocate ACM™ pages and maps to virtual address space

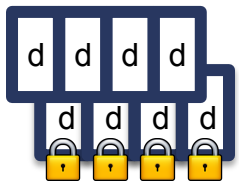
2. memcpy()

Process writes and reads to pages through standard memory-access semantics



CPU loads/stores data to ACM™ pages

3. Checkpoint pages



working pages

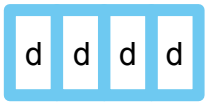
snapshot pages

Application-consistent snapshot pages are created or updated

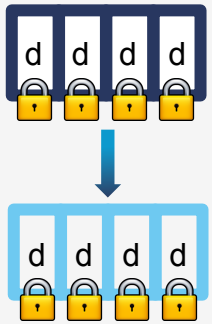
4. Destage pages

Completed pages

Pages persisted to flash namespace

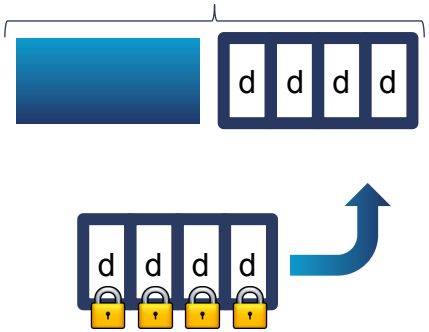


System failure event



System failure automatically preserves snapshot pages

5. Restore state
app virtual address space



Re-started process maps restored snapshot pages into virtual address space

Virtual Storage Layer

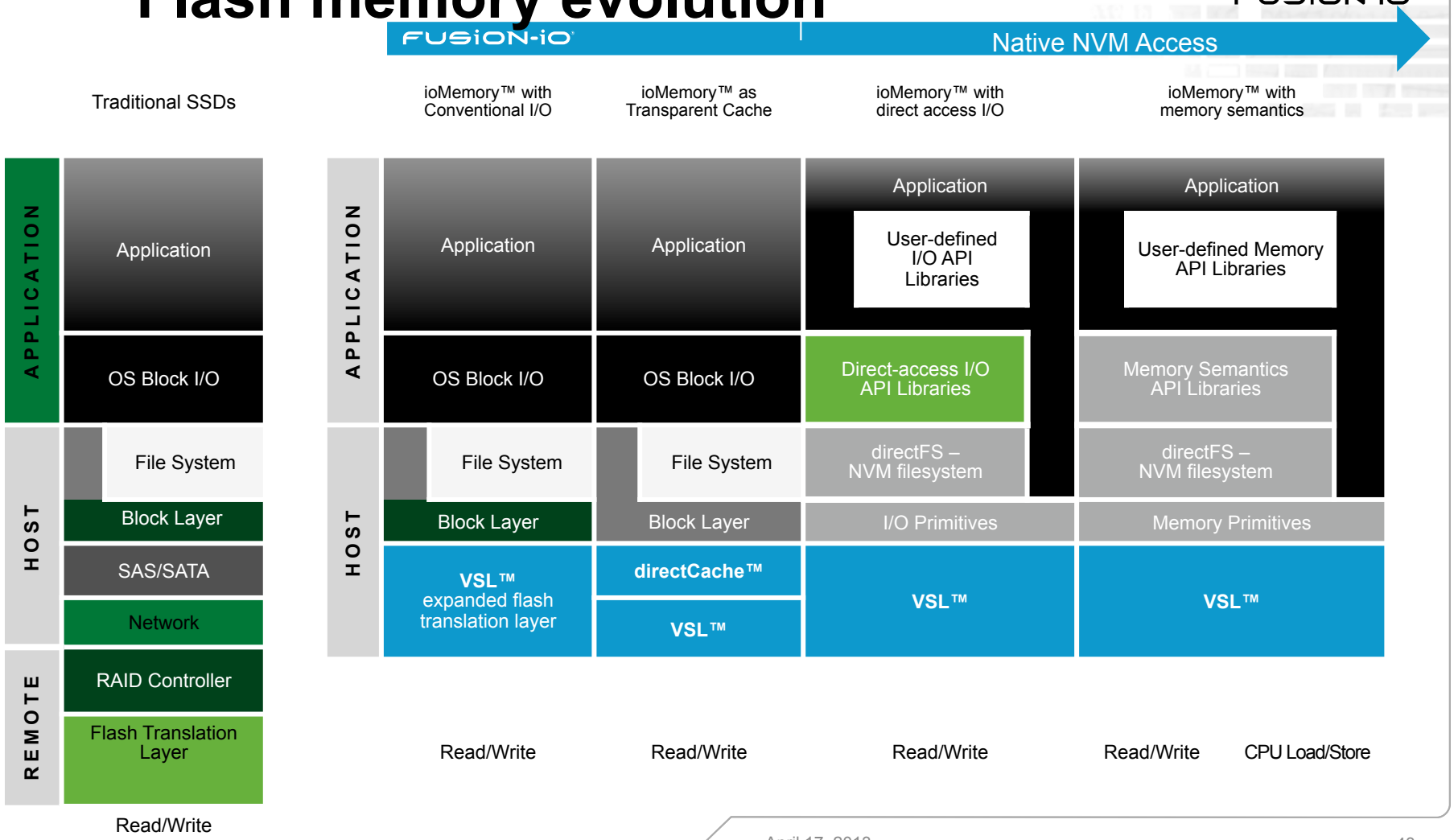
Comparing i/o and memory access semantics

FUSION-IO

I/O	I/O semantics examples: • Open file descriptor – open(), read(), write(), seek(), close() • (New) Write multiple data blocks atomically, nvm_vectored_write() • (New) Open key-value store – nvm_kv_open(), kv_put(), kv_get(), kv_batch_*()
Memory Access **(Volatile)**	Volatile memory semantics example: • Allocate virtual memory, e.g. malloc() • memcpy/pointer dereference writes (or reads) to memory address • (Improved) Page-faulting transparently loads data from NVM into memory
Memory Access **(Non-Volatile)**	Non-volatile memory semantics example: • (New) Allocate and map Auto-Commit Memory™ (ACM) virtual memory pages • memcpy/pointer dereference writes (or reads) to memory address • (New) Call checkpoint() to create application-consistent ACM page snapshots • (New) After system failure, remap ACM snapshot pages to recover memory state • (New) De-stage completed ACM pages to NVM namespace • (New) Remap and access ACM pages from NVM namespace at any time

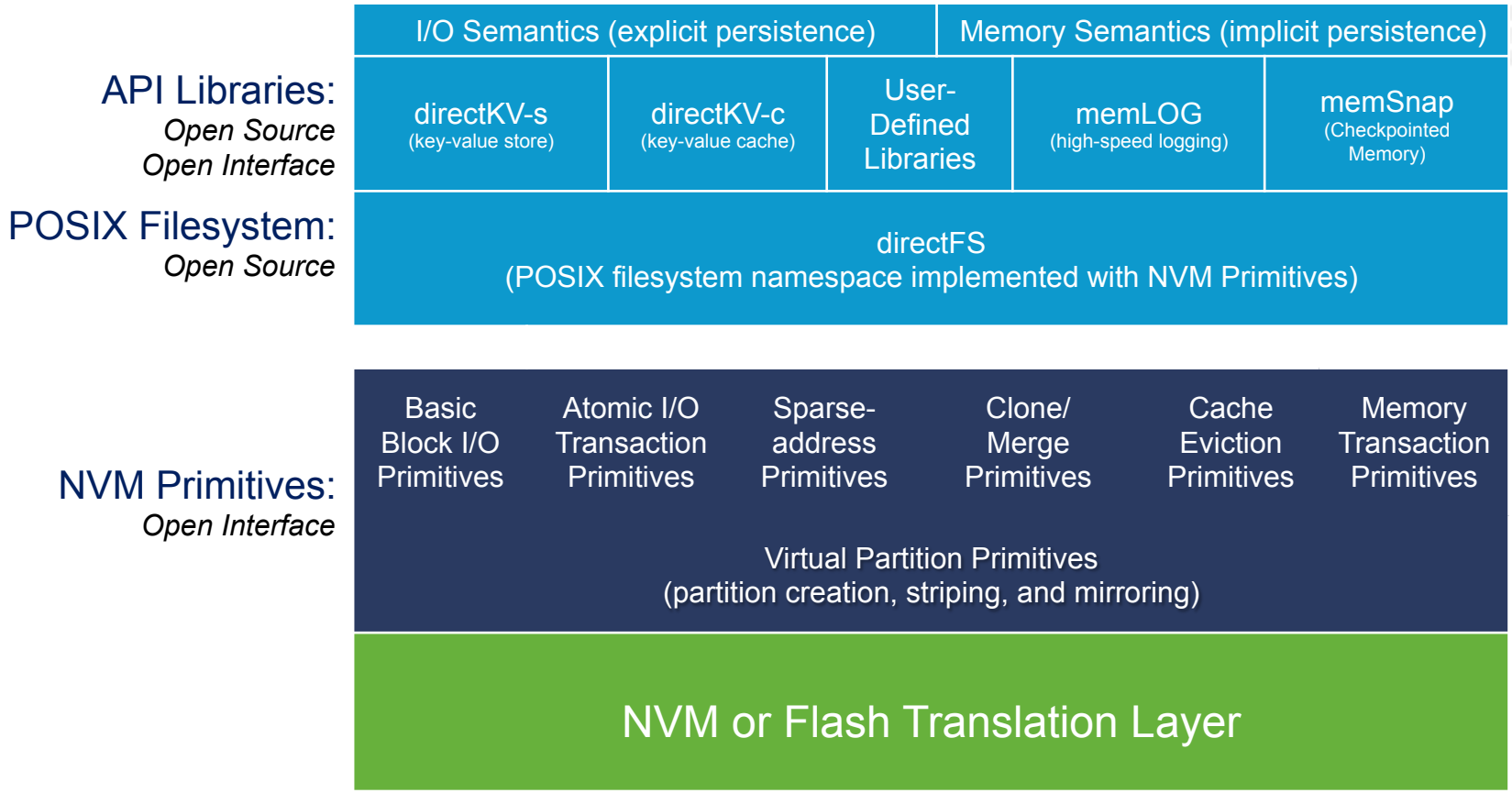
Flash memory evolution

FUSION-IO



Open NVM development Interfaces

FUSION-IO



Topics – NoSQL Munich 2013

FUSION-IO

1. What are we building ?
2. Why are we building it?
3. ioMemory SDK
4. KV-API
5. Direct FS
6. Memory Access Semantics
7. Where are we headed?

API Specs posted at developer.fusionio.com

FUSION-IO

Direct-access to NVM is for developers whose software retrieves and stores data.

- Early-access to ioMemory SDK API specs and technical documentation (limited enrollment during early-access phase) http://developer.fusionio.com

- **Write less code** to create high-performing apps
- **Tap into performance** not available with conventional I/O access to SSDs
- **Reduce operating costs** by decreasing RAM while increasing NVM

Open Interfaces and Open Source

FUSION-IO

- NVM Primitives: Open Interface
- directFS: Open Source, POSIX Interface
- NVM API Libraries: Open Source, Open Interface
- INCITS SCSI (T10) active standards proposals:
  - SBC-4 SPC-5 Atomic-Write
    http://www.t10.org/cgi-bin/ac.pl?t=d&f=11-229r6.pdf
  - SBC-4 SPC-5 Scattered writes, optionally atomic
    http://www.t10.org/cgi-bin/ac.pl?t=d&f=12-086r3.pdf
  - SBC-4 SPC-5 Gathered reads, optionally atomic
    http://www.t10.org/cgi-bin/ac.pl?t=d&f=12-087r3.pdf
- SNIA NVM-Programming TWG active member

Catalyst for top industry players to Accelerate pursuit of NVM programming

FUSION-IO

SNIA™
Advancing storage & information technology

SNIA Links:

- Webcasts
- Videos
- Certification
- Tutorials
- Multimedia
- e-Courses
- Standards
- Events
- News
- Membership
- Solid State Storage

A Message from SNIA Technical Council

SNIA CALL FOR PARTICIPATION
NVM Programming Technical Work Group (TWG)

The SNIA Technical Council has recently approved a new technical work group. The NVM Programming TWG was created for the purpose of accelerating availability of software enabling NVM (Non-Volatile Memory) hardware. The TWG creates specifications which provide guidance to operating system, device driver, and application developers. These specifications are vendor agnostic and support all the NVM technologies of member companies. The NVM Programming TWG:

Dell, EMC, Fujitsu, HP, Intel, NetApp, Oracle, and QLogic have all communicated their support for this activity. Development teams at several other SNIA member companies have expressed support and are waiting for official company approval to state support.

…And Resonating through the Industry

FUSION-IO

The Register®

Three questions Fusion-io's rivals face after flash API bombshell

Apps bypassing OS and disk to store hot data - chaos or breakthrough?

By **Chris Mellor** • **Get more from this author**

Posted in Blocks and Files, 20th April 2012 07:29 GMT

Storage array vendors are at a disadvantage here. They need three things to play in this area:

- To remain strategically important to their customers they need to get server-connected flash hardware, or shared flash array hardware connected to servers across links fast enough to provide a memory tier, meaning PCIe-class speed.
- Then they need to get cut-through software capability equivalent to that of Fusion-io.
- They would also require software to hook up their existing arrays to the server flash, bleeding off cooling data and loading up hotter data to keep app software direct disk I/O to a minimum.

These are the table stakes I think are necessary for storage array vendors to play in the server flash application speed-up game. Getting the ability to accelerate applications by factors of 5X to 20X is going to place storage vendors in a whole new pecking order. Application acceleration glory days are there for the taking.

Comprehensive Customer Success

FUSION-IO

FINANCIALS	WEB	TECHNOLOGY	RETAIL	MANUFACTURING/ GOVERNMENT
NYSE Euronext, STARKEY MORTGAGE, Callcredit Information Group, First Rate, TransUnion	facebook, LOCKERZ, HOST EUROPE, myspace, craigslist, NING, Kontera, HOST EUROPE, dwango, mixi, skyscanner	salesforce.com, V3, SevOne, PIPKINS, CLOUDMARK, PRIME FOCUS, zendesk, MCX, ak, FRONTSAFE, PANDORA, BrainPad	alibris, Rakuten, Zappos.com, wine.com, XBOX LIVE, CyberAgent, BETONSOFT, DRECOM, Jointown	LLOG, Lawrence Livermore National Laboratory, REVERE ELECTRIC SUPPLY CO., Syntricity, Esbjerg Kommune
CREDIT SUISSE	Answers.com The world's leading Q&A site	datalogix	EQUINOX	tabulex
5x FASTER DATA ANALYSIS	**30x** FASTER DATABASE REPLICATION	**40x** FASTER DATA WAREHOUSE QUERIES	**15x** QUERY PROCESSING THROUGHPUT	**15x** FASTER QUERIES

30+ case studies at http://fusionio.com/casestudies


THANK YOU!
fusionio.com | REDEFINE WHAT'S POSSIBLE