OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP

Direct Storage-class Memory Access

Using a High-Performance Networking Stack to integrate Storage Class Memory

Bernard Metzler, Blake G. Fitch, Lars Schneidenbach

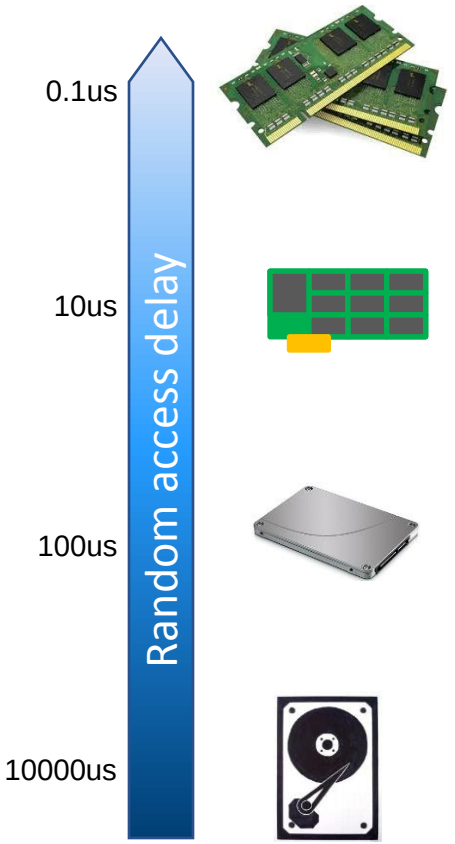
IBM Research

Outline

- DSA: What is it for?
- DSA Design: Unified OFA based I/O Stack
- DSA Prototype
- Example applications
- Summary & Outlook

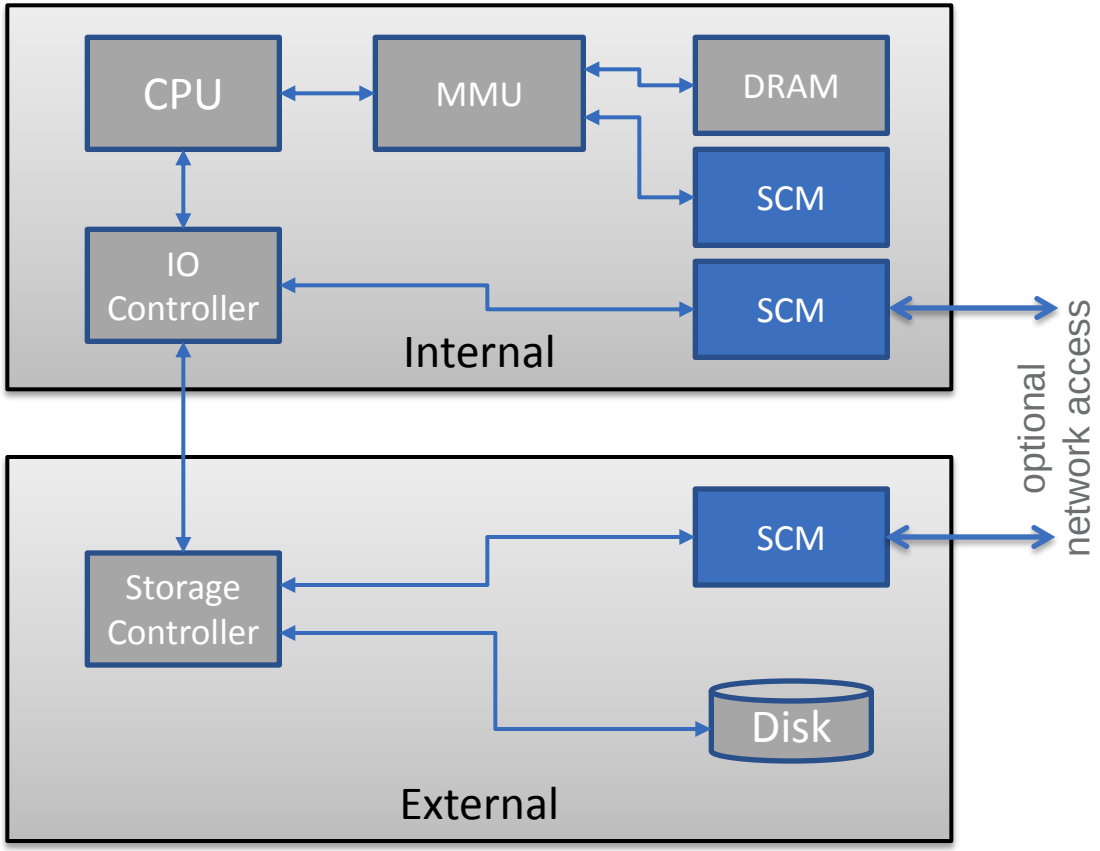
Tackling a changing Storage Landscape

- New persistent memory technologies
  - From Tape to Disc to Flash to
  - PCM, Spin/Torque, ReRAM, …
- Changing storage IO
  - From IDE/SCSI to SATA/SAS to PCI…
  - …towards IO elimination
- Direct Storage Access architecture:
  - Low level, application private storage interface
  - Can be integrated with
    - RDMA network stack, and
    - Legacy host storage stack
  - Rich semantics: Read, Write, Atomics
  - Ready for future storage technologies



Integrating Storage-class Memory

http://researcher.watson.ibm.com/researcher/files/us-gwburr/Almaden_SCM_overview_Jan2013.pdf

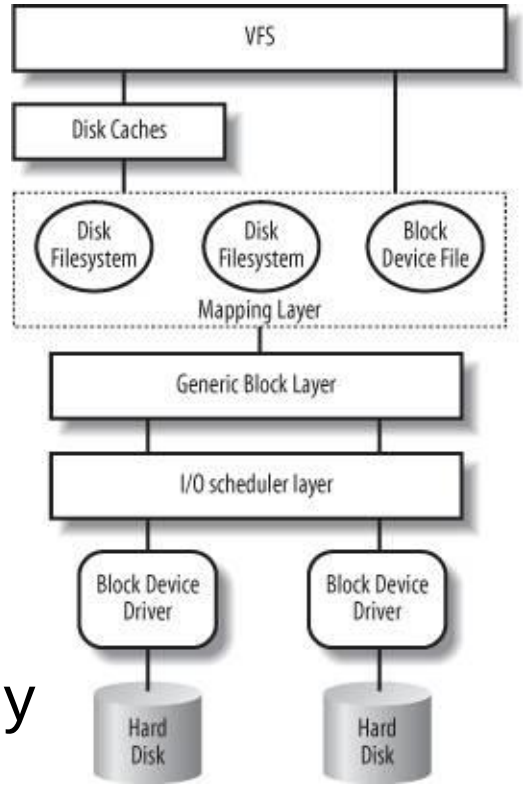


- **M-type**: Synchronous
  - Hardware managed
  - Low overhead
  - CPU waits
  - New NVM tech. (not flash)
  - Cached or pooled memory
  - Persistence requires redundancy
- **S-Type**: Asynchronous
  - Software managed
  - High overhead
  - CPU doesn't wait
  - Flash or new NVM
  - Paging or storage
  - Persistence: RAID

Goal: Define standard low level, highly efficient, potentially application private SCM Interface covering ALL types of SCM

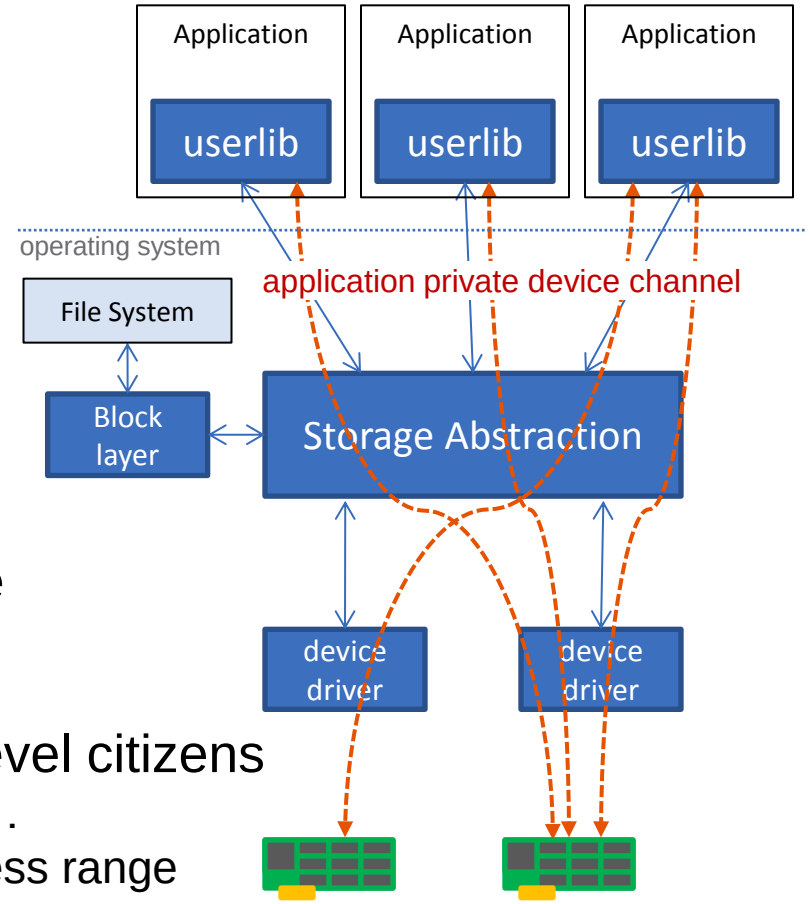
Traditional Storage I/O Stack: Bad fit for SCM

- BIO: enable efficient sequential disc access
  - Heavy CPU involvement
  - Single synchronization point for device access
    - partially relaxed with MQ BIO
- Inefficient SCM access
  - Precludes parallelism
    - NVMe + MQ BIO to improve here
  - Enforces block based device access not needed for future SCM technology



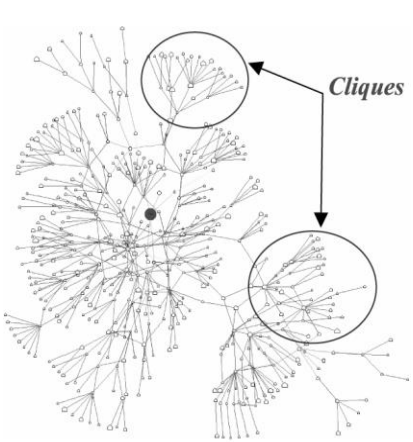
Direct Application Storage I/O

- Trusted application-device channel
- Asynchronous operation
  - Deep request and completion queue(s)
  - High level of access parallelism
- Efficient IO path
  - CPU affinity, NUMA awareness
  - Can be lock free
  - Benefits from potential HW assists
  - Ready for efficient I/O stack virtualization
- Serves as base/primary SCM interface
  - Access granularity: [address, length]
  - Optional block layer integration
- Higher level storage systems as first level citizens
  - File-systems, databases, object stores, …
  - Translation of objects to I/O device address range
  - File, database column, key/value item, …

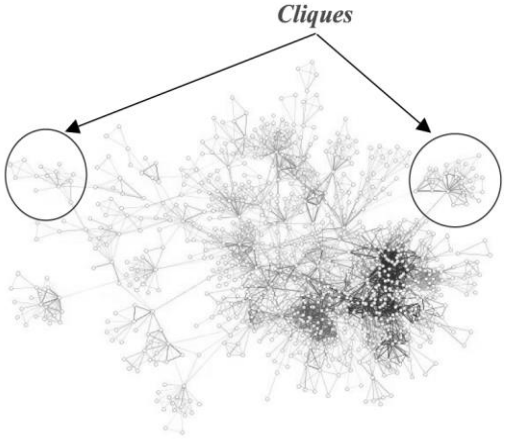


Byte addressable SCM

- Make a single interface change for Flash and Future SCM
- Lowest level of access abstraction
  - System I/O view: [PA, len]: NVM access above FTL
  - Application view: [VA, len]: most concrete object representation
  - [VA, len] to be mapped to [key, offset, len] for access
- Advantages
  - Efficient data I/O
  - Direct object reference
  - Higher levels of abstraction if needed
  - Future SCM technology proof
- Examples:
  - Byte addressable object store
  - Traversing nodes of terabyte graph, random pointer chasing
  - Terabyte Sorting



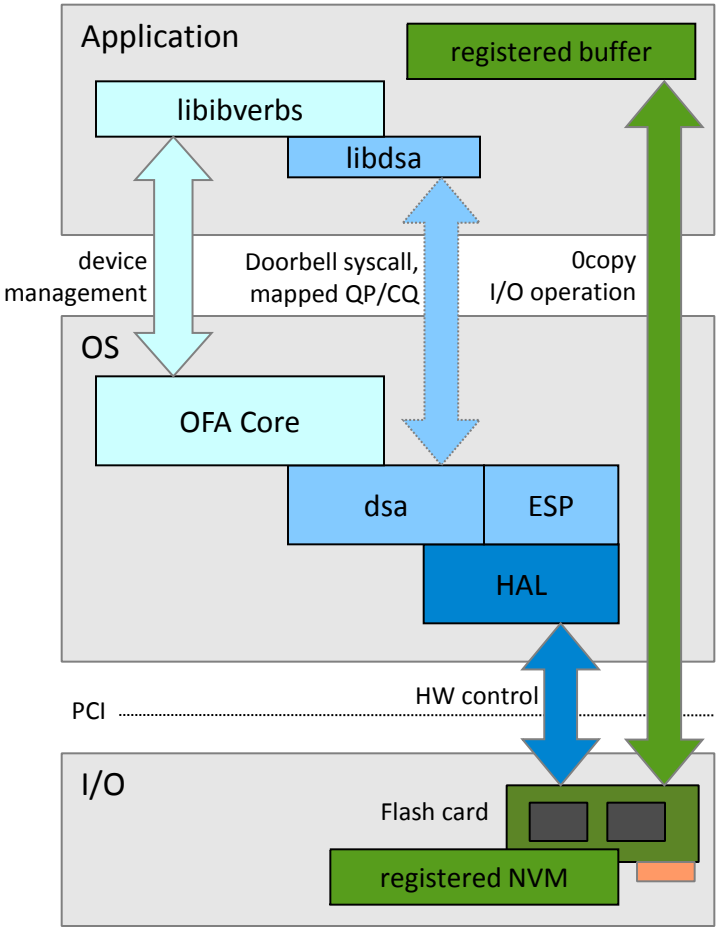
(a) Graph of a Local Gnutella P2P



(b) Graph of Research Co-authorship

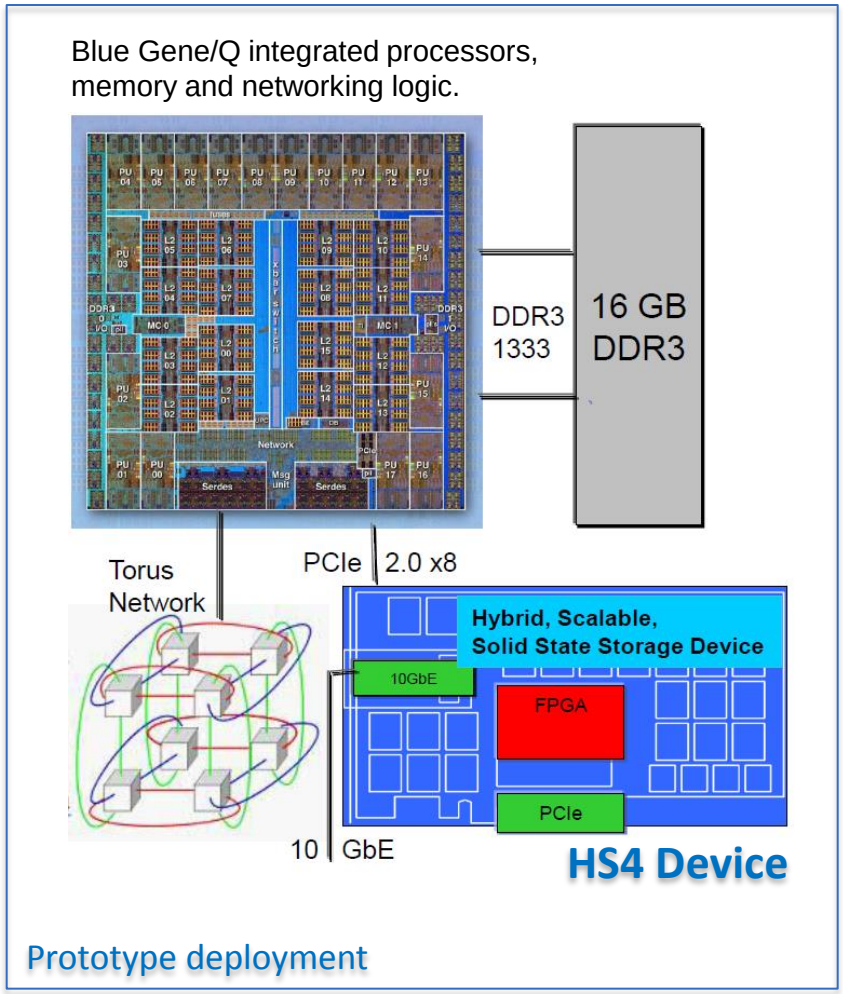
DSA: An OFED based Prototype

- Prototype PCI attached flash adapter
- ‘dsa’ OFED verbs provider and ‘libdsa’
- User mapped kernel QP/CQ
- Proprietary DB syscall (or HW capability)
- Hardware Adaptation Layer (HAL)
- DSA application operation:
  - Open dsa OFED device, create PD
  - Register local target buffers (ibv_reg_mr())
  - Create QP and move it to RTS/connect to ‘embedded storage peer’ (ESP)
  - Post Receive's + Send's executing RPC’s to ESP to learn partition parameters, register I/O memory and associated RTag's
- Post READ/WRITE to read/write IO memory into/from local registered buffer
- Atomic operations on flash TBD/next step



HS4: Prototype hybrid SCM Device

- Hybrid Scalable Solid State Storage Device
  - PCIe 2.0 x 8
  - 2 x 10Gb Ethernet
  - 2 TB SLC (raw)
  - 8 GB DRAM
  - FPGA ties it all together
- Software Stack
  - Kernel module interface fits with DSA-HAL
  - SW based GC, FTL
  - Single PCI request/response queue



Application Level Performance

Flash Performance:			BG/Q		P7+	
			Single-Thread	**Best config**	**Single-Thread**	**Best config**
DSA client	**BW (1MB)**	**Write**	1050 MB/s	==	2340 MB/s	==
		Read	1300 MB/s	2270 MB/s (2 procs)	3020 MB/s	==
	IOPS (8k)	**Write**	65k	91k (4 procs)	270k	==
		Read	70k	180k (3 procs)	360k	==
	Latency (8k)	**Write**	490µs	==	440µs	==
		Read	165µs	==	101µs	==

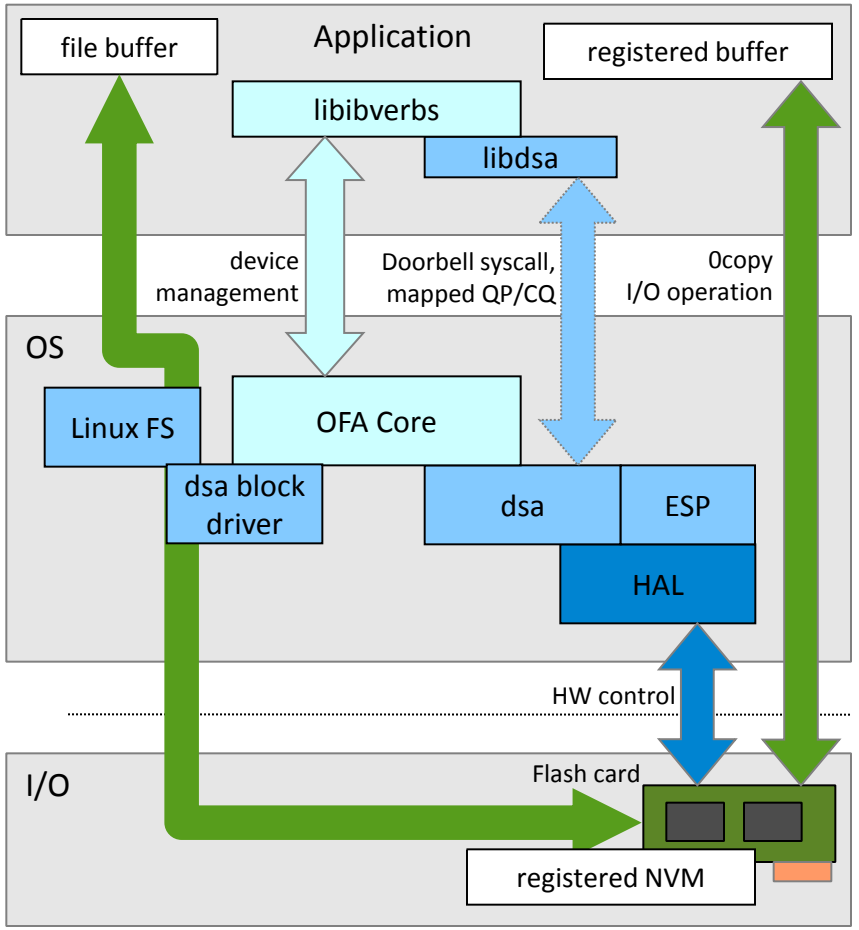
- Systems:
  - BlueGene/Q system, 1.6 GHz A2
  - P7+ system, 3.6 GHz
- Prototype
  - Single Core Write path performance
  - Working on CPU affinity, NUMA awareness, lockless

DSA Hybrid Memory Access (DRAM/MRAM)					
			BG/Q	**P7+**	**Comment**
IOPS	**DRAM**	**Write**	120k	635k	32byte, single thread
			230k		32byte, 2 threads
			340k		32byte, 3 threads
		Read	120k	740k	32byte, single thread
			235k	920k	32byte, 2 threads
			340k	1050k	32byte, 3 threads
Latency	**DRAM**	**Write**	70µs	11.5µs	
		Read	70µs	10.4µs	

- Similar results using Java/jVerbs research prototype

Example Block Layer Integration

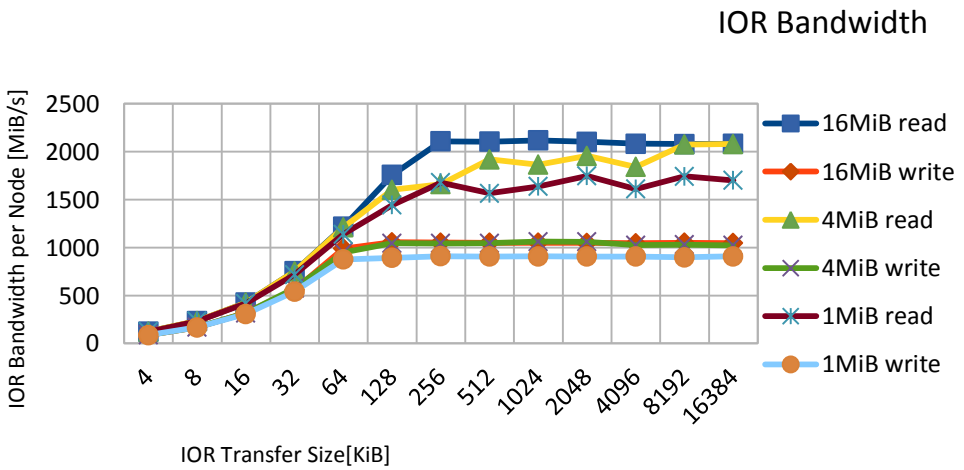
- DSA block driver
- Attaches to OFA core as a kernel level client
- Reads/Writes blocks
- Blocking/non blocking operations
- High parallelism possible (multi-QP)
  - Currently 2 QP's
  - BIO-MQ interface considered
- Prototyped
  - File system on Flash partition
  - Transparent GPFS/Flash integration on BG/Q



Block Layer Performance

Flash Performance:			BG/Q		P7+	
			Single-Thread	Best config	Single-Thread	Best config
DSA client	BW (1MB)	Write	1050 MB/s	==	2340 MB/s	==
		Read	1300 MB/s	2270 MB/s (2 procs)	3020 MB/s	==
	IOPS (8k)	Write	65k	91k (4 procs)	270k	==
		Read	70k	180k (3 procs)	360k	==
	Latency (8k)	Write	490µs	==	440µs	==
		Read	165µs	==	101µs	==
VBD (dd)	BW (1MB)	Write	920 MB/s	992 MB/s (2 procs)	1300 MB/s	2200 MB/s (2 procs)
		Read	1200 MB/s	2100 MB/s	3000 MB/s	==

- BlueGene/Q system
  - 1.6 GHz A2, 8..64 IO Nodes, 3 Dim Torus,1 HS4 card each
- Raw 'dd' I/O
  - DSA block driver
  - Read, Write
- Experimental GPFS/IOR Performance
  - 2 IOR processes per node,
  - Read ~about 2 x Write
  - POSIX and MPIO with similar results



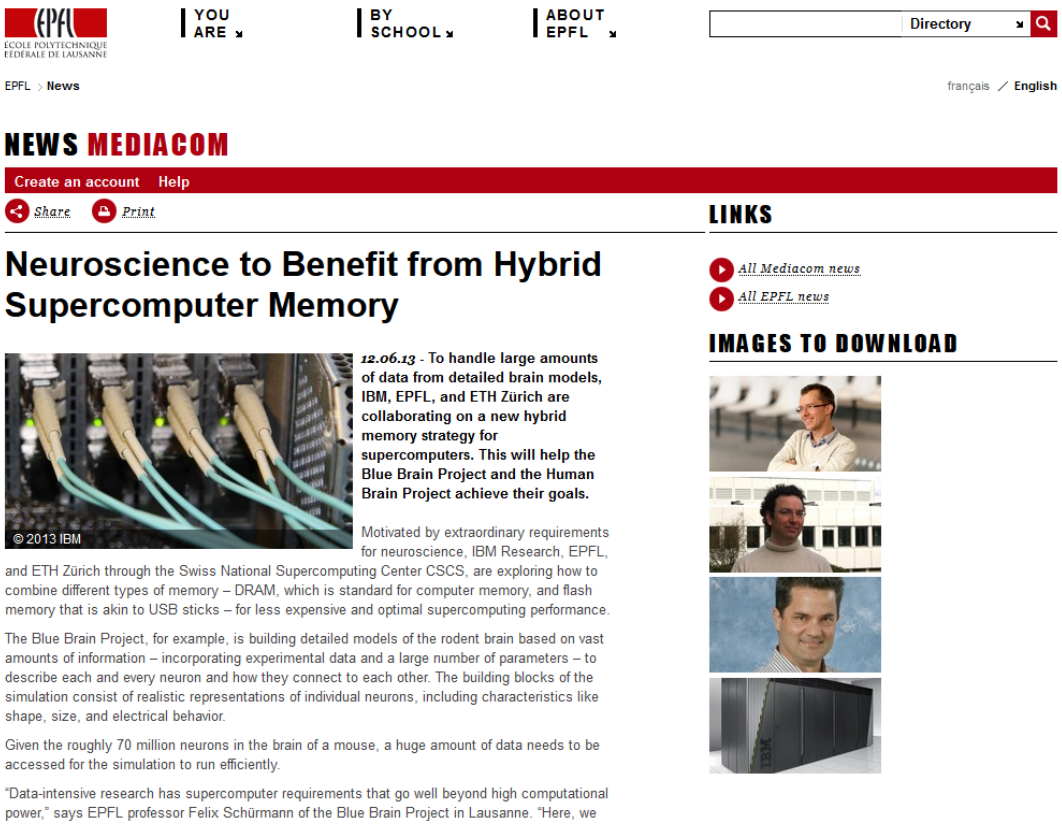
DSA and Networking

- Legacy Block Layer implies
  - Block storage access, which implies
  - Block exchange protocols for networked storage access: iSCSI, iSER, FCoE, FCoIP, NFS, …
- Further I/O consolidation possible
  - Tag, offset, length @ network address
  - There is no extra protocol (just IB, iWarp, RoCEE)
  - Explicit control over data locality (just IP address)
- Block layer remains optional upper layer abstraction, but
  - No block exchange protocol needed

Current and future DSA Usage

- BlueBrain & HumanBrain projects
  - BlueGene/Q systems running Linux
  - Equipped with HS4 NVM cards in I/O drawers

- RDFS: IBM Zurich Lab effort for HDFS compatible file system
  - Completely RDMA based
  - Java RDMA I/O via 'jVerbs' (zero copy I/O)
  - In-memory file system
  - To be integrated with DSA for local and networked storage access



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

YOU ARE | BY SCHOOL | ABOUT EPFL | Directory

EPFL › News

français / English

NEWS MEDIACOM

Create an account Help

Share Print

Neuroscience to Benefit from Hybrid Supercomputer Memory

© 2013 IBM

12.06.13 - **To handle large amounts of data from detailed brain models, IBM, EPFL, and ETH Zürich are collaborating on a new hybrid memory strategy for supercomputers. This will help the Blue Brain Project and the Human Brain Project achieve their goals.**

Motivated by extraordinary requirements for neuroscience, IBM Research, EPFL, and ETH Zürich through the Swiss National Supercomputing Center CSCS, are exploring how to combine different types of memory – DRAM, which is standard for computer memory, and flash memory that is akin to USB sticks – for less expensive and optimal supercomputing performance.

The Blue Brain Project, for example, is building detailed models of the rodent brain based on vast amounts of information – incorporating experimental data and a large number of parameters – to describe each and every neuron and how they connect to each other. The building blocks of the simulation consist of realistic representations of individual neurons, including characteristics like shape, size, and electrical behavior.

Given the roughly 70 million neurons in the brain of a mouse, a huge amount of data needs to be accessed for the simulation to run efficiently.

"Data-intensive research has supercomputer requirements that go well beyond high computational power," says EPFL professor Felix Schürmann of the Blue Brain Project in Lausanne. "Here, we

LINKS

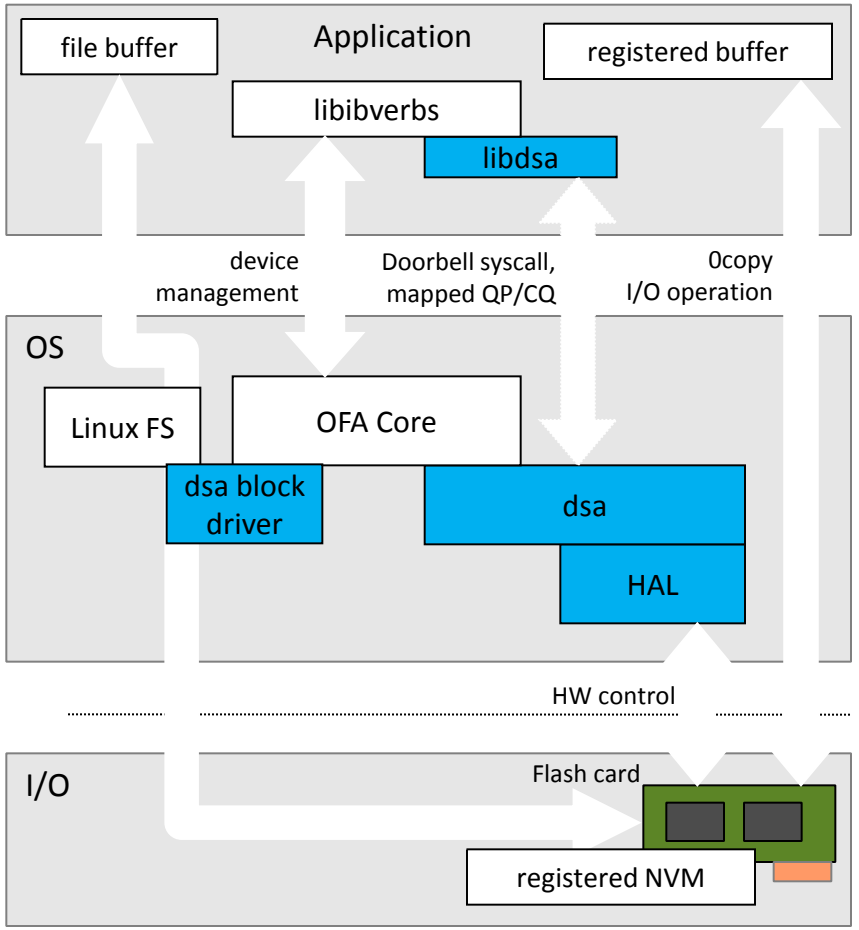
All Mediacom news

All EPFL news

IMAGES TO DOWNLOAD

DSA Code Status

- All core components implemented
  - Architecture proposed at FAST'14 Linux Summit
    - Encouraging feedback - community interested
  - dsa, libdsa, HAL to be open sourced soon
  - dsa block driver too
  - github, gitorious, … to start with
- Next steps:
  - Integration with off-the-shelf available NVM interfaces: NVMe
  - Para-virtualization support
  - SoftiWarp based kernel client for simple remote access
- Proposed as an OpenFabrics RDMA verbs provider



Summary

- Direct Storage-class Memory Access
  - Generic low-level all type SCM access
  - Application-private trusted device access
  - Rich semantics: Read, Write, Atomics @ addr, len
  - Legacy storage stack integration via block layer
  - Simplified
    - Storage/networking integration
    - High performance virtualization
- Proposed as an OpenFabrics RDMA verbs provider
  - Seamless OpenFabrics integration
  - Open source announcement soon

- Further reading

http://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/bgas-BoF/bgas-BoF-fitch.pdf?__blob=publicationFile
http://www.adms-conf.org/2013/BGAS_Overview_Fitch_VLDB_ADMS.pdf
https://www.openfabrics.org/images/docs/2013_Dev_Workshop/Wed_0424/2013_Workshop_Wed_0930_MetzlerOFA_IOAPI.pdf

Thank You

OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP