

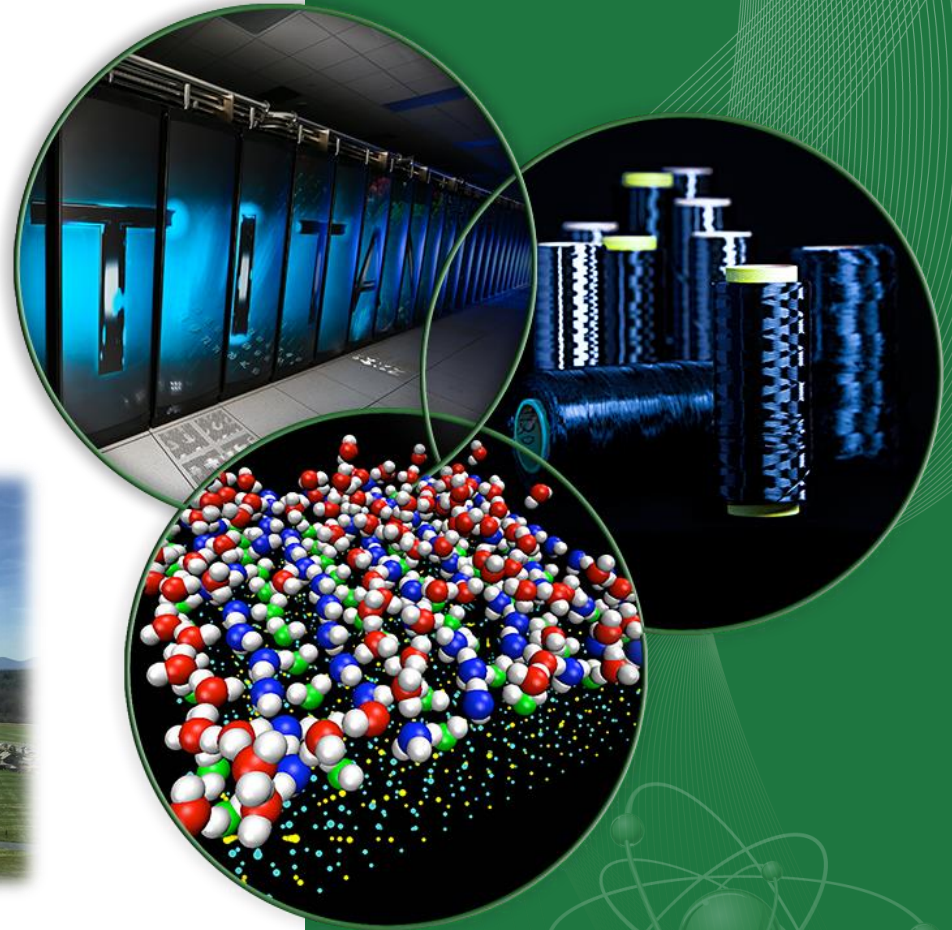
Managing the Memory Hierarchy

Jeffrey S. Vetter

Sparsh Mittal, Joel Denny,
Seyong Lee

Presented to
SOS20
Asheville

24 Mar 2016



Exascale architecture targets circa 2009

2009 Exascale Challenges Workshop in San Diego

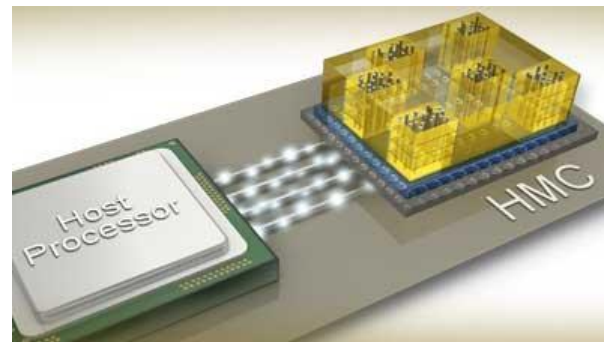
Attendees envisioned two possible architectural swim lanes:

1. Homogeneous many-core thin-node system
2. Heterogeneous (accelerator + CPU) fat-node system

System attributes	2009	“Pre-Exascale”		“Exascale”	
System peak	2 PF	100-200 PF/s		1 Exaflop/s	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32–64 PB	
Storage	15 PB	150 PB		500 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/s	1 TB/s	0.4 TB/s	4 TB/s
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	500,000	50,000	1,000,000	100,000
Node interconnect BW	1.5 GB/s	150 GB/s	1 TB/s	250 GB/s	2 TB/s
IO Bandwidth	0.2 TB/s	10 TB/s		30-60 TB/s	
MTTI	day	O(1 day)		O(0.1 day)	

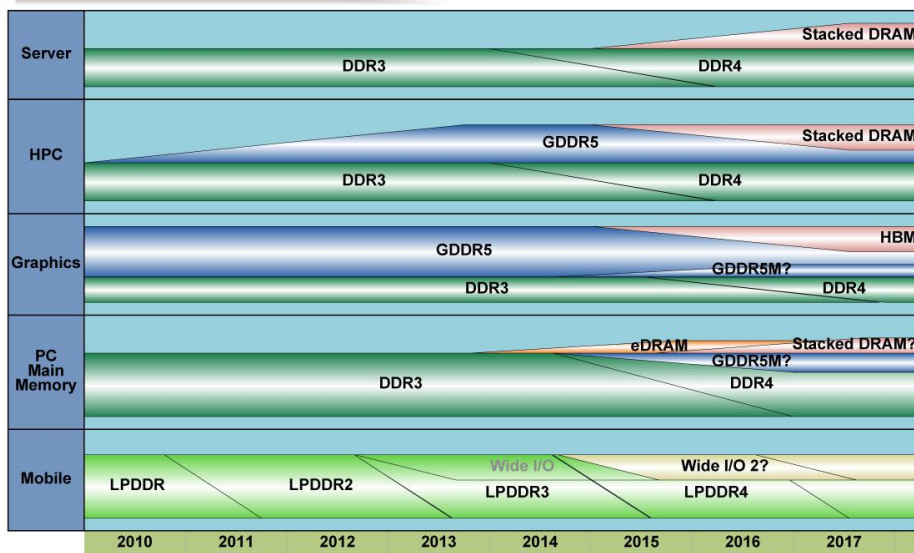
Memory Systems

- Multimode memories
 - Fused, shared memory
 - Scratchpads
 - Write through, write back, etc
 - Virtual v. Physical, paging strategies
 - Consistency and coherence protocols
- 2.5D, 3D Stacking
- HMC, HBM/2/3, LPDDR4, GDDR5, WIDEIO2, etc
- New devices (ReRAM, PCRAM, Xpoint)



https://www.micron.com/~media/track-2-images/content-images/content_image_hmc.jpg?la=en

DRAM Transition



	SRAM	DRAM	eDRAM	2D NAND Flash	3D NAND Flash	PCRAM	STTRAM	2D ReRAM	3D ReRAM
Data Retention	N	N	N	Y	Y	Y	Y	Y	Y
Cell Size (F ²)	50-200	4-6	19-26	2-5	<1	4-10	8-40	4	<1
Minimum F demonstrated (nm)	14	25	22	16	64	20	28	27	24
Read Time (ns)	<1	30	5	10 ³	10 ³	10 ⁻⁵⁰	3-10	10-50	10-50
Write Time (ns)	<1	50	5	10 ³	10 ³	10 ³	3-10	10-50	10-50
Number of Rewrites	10 ¹⁶	10 ¹⁶	10 ¹⁶	10 ³ -10 ⁴	10 ³ -10 ⁴	10 ³ -10 ¹⁰	10 ¹⁵	10 ³ -10 ¹²	10 ³ -10 ¹²
Read Power	Low	Low	Low	High	High	Low	Medium	Medium	Medium
Write Power	Low	Low	Low	High	High	High	Medium	Medium	Medium
Power (other than R/W)	Leakage	Refresh	Refresh	None	None	None	None	Stoak	Stoak
Maturity	High	High	High	Low	Low	Low	Low	Low	Low

J.S. Vetter and S. Mittal, "Opportunities for Nonvolatile Memory Systems in Extreme-Scale High Performance Computing," CISE, 17(2):73-82, 2015.

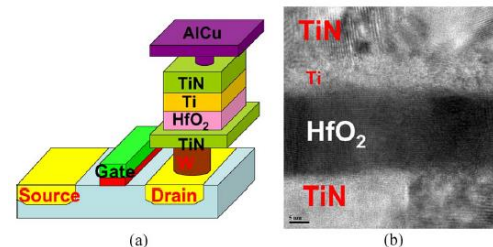


Fig. 4. (a) A typical 1T1R structure of RRAM with HfO₂; (b) HR-TEM image of the TiN/Ti/HfO₂/TiN stacked layer; the thickness of the HfO₂ is 20 nm.

H.S.P. Wong, H.Y. Lee, S. Yu et al., "Metal-oxide RRAM," Proceedings of the IEEE, 100(6):1951-70, 2012.

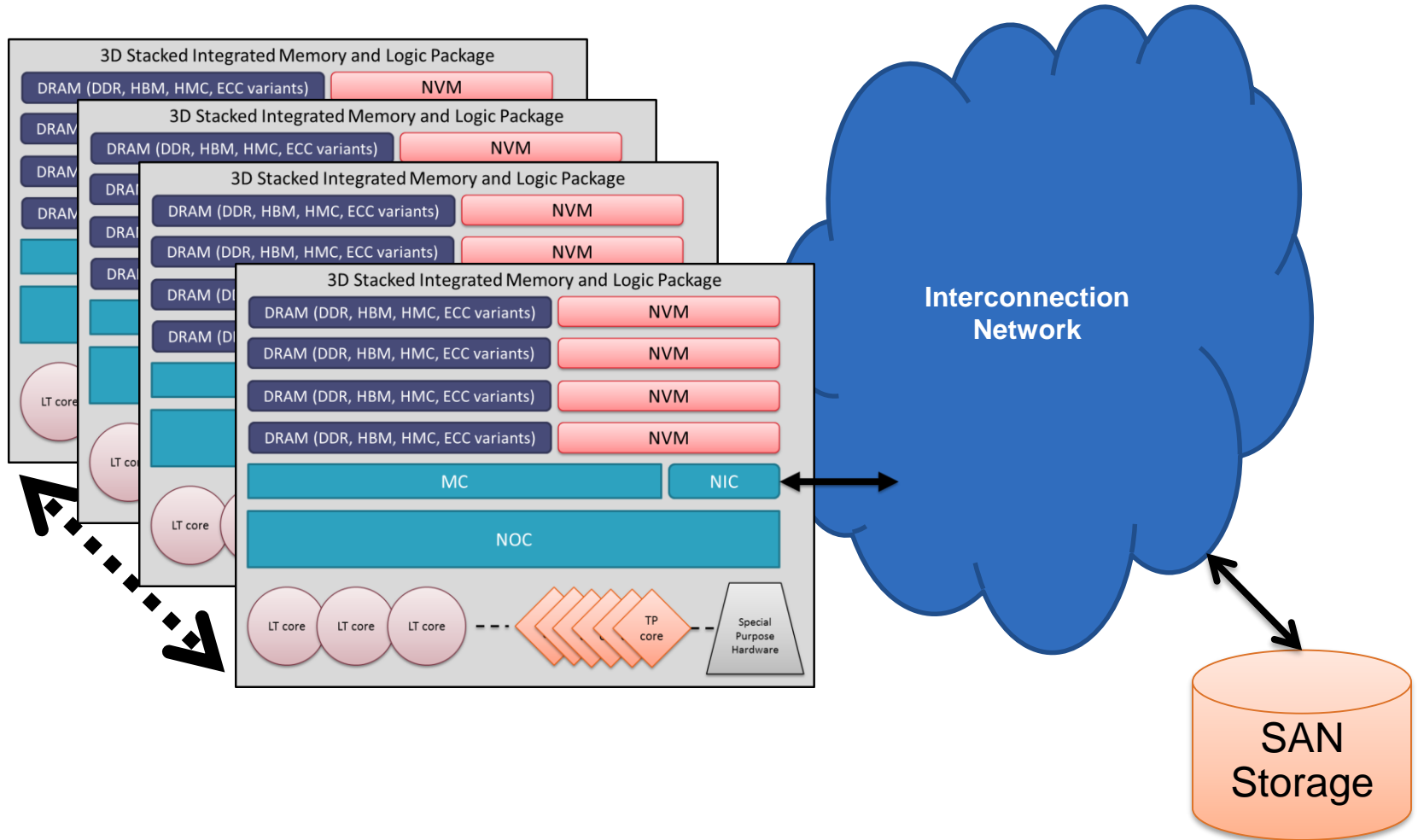
Current ASCR Computing At a Glance

System attributes	NERSC Now	OLCF Now	ALCF Now	NERSC Upgrade	OLCF Upgrade	ALCF Upgrades	
Planned Installation	Edison	TITAN	MIRA	Cori 2016	Summit 2017-2018	Theta 2016	Aurora 2018-2019
System peak (PF)	2.6	27	10	> 30	150	>8.5	180
Peak Power (MW)	2	9	4.8	< 3.7	10	1.7	13
Total system memory	357 TB	710TB	768TB	~1 PB DDR4 + High Bandwidth Memory (HBM)+1.5PB persistent memory	> 1.74 PB DDR4 + HBM + 2.8 PB persistent memory	>480 TB DDR4 + High Bandwidth Memory (HBM)	> 7 PB High Bandwidth On-Package Memory Local Memory and Persistent Memory
Node performance (TF)	0.460	1.452	0.204	> 3	> 40	> 3	> 17 times Mira
Node processors	Intel Ivy Bridge	AMD Opteron Nvidia Kepler	64-bit PowerPC A2	Intel Knights Landing many core CPUs Intel Haswell CPU in data partition	Multiple IBM Power9 CPUs & multiple Nvidia Voltas GPUS	Intel Knights Landing Xeon Phi many core CPUs	Knights Hill Xeon Phi many core CPUs
System size (nodes)	5,600 nodes	18,688 nodes	49,152	9,300 nodes 1,900 nodes in data partition	~3,500 nodes	>2,500 nodes	>50,000 nodes
System Interconnect	Aries	Gemini	5D Torus	Aries	Dual Rail EDR-IB	Aries	2 nd Generation Intel Omni-Path Architecture
File System	7.6 PB 168 GB/s, Lustre®	32 PB 1 TB/s, Lustre®	26 PB 300 GB/s GPFS™	28 PB 744 GB/s Lustre®	120 PB 1 TB/s GPFS™	10PB, 210 GB/s Lustre initial	150 PB 1 TB/s Lustre®

Steve Binkley, Dec 2015

Complexity $\propto T$

Notional Future Architecture



GPU Users: we don't want no stinking ECC!

An Investigation of the Effects of Error Correcting Code on GPU-accelerated Molecular Dynamics Simulations

Ross C. Walker
San Diego Supercomputer Center
Department of Chemistry and Biochemistry
UC San Diego
La Jolla, CA 92093
ross@rosswalker.co.uk

Robin M. Betz
San Diego Supercomputer Center
La Jolla, CA 92093
rbetz@ucsd.edu

ABSTRACT

Molecular dynamics (MD) simulations rely on the accurate evaluation and integration of Newton's equations of motion to propagate the positions of atoms in proteins during a simulation. As such, one can expect them to be sensitive to any form of numerical error that may occur during a simulation. Increasingly graphics processing units (GPUs) are

Keywords

XSEDE 2013, GPU-acceleration, ECC error

1. INTRODUCTION

The field of computational sciences uses the power of modern computers to gain insight into scientific systems. Re-

Blackcomb: Hardware-Software Co-design for Non-Volatile Memory in Exascale Systems (since 2010)

Jeffrey Vetter, ORNL
Robert Schreiber, HP Labs
Trevor Mudge, University of Michigan
Yuan Xie, Penn State University

<http://ft.ornl.gov/trac/blackcomb>

FWP #ERKJU59

Objectives

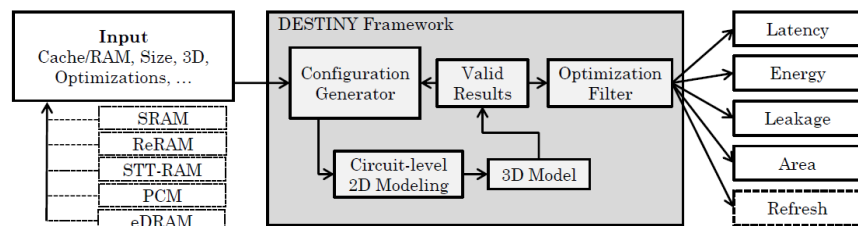
- Re-architect servers and clusters, using nonvolatile memory (NVM) to overcome resilience, energy, and performance walls in exascale computing:
 - Ultrafast checkpointing to nearby NVM
 - Redesign the memory hierarchy for exascale, using new memory technologies
 - Replace disk with fast, low-power NVM
 - Enhance resilience and energy efficiency
 - Provide added memory capacity

Approach

- Identify and evaluate the most promising (NVM) technologies – STT, PCRAM, memristor.
- Explore assembly of NVM and CMOS into a storage + memory stack.
- Propose an exascale HPC system architecture that builds on our new memory architecture.
- New resilience strategies in software.
- Test and simulate, driven by proxy applications.

NVSim, Destiny

- A comprehensive tool which models both 2D and 3D caches designed with five prominent memory technologies: SRAM, eDRAM, PCM, STT-RAM and ReRAM
- Covers both conventional and emerging memory technologies
- Models 22nm to 180nm and facilitates design-space exploration



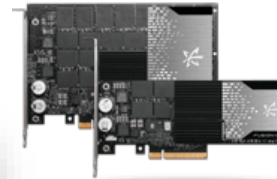
NVL-C

- Familiar and portable programming interfaces
- Provide checks for correctness and efficiency
- Understand application requirements

```
#include <nvl.h>
struct list {
    int value;
    nvl struct list *next;
};

void remove(int k) {
    nvl_heap_t *heap
        = nvl_open("foo.nvl");
    nvl struct list *a
        = nvl_get_root(heap, struct list);
    #pragma nvl atomic
    while (a->next != NULL) {
        if (a->next->value == k)
            a->next = a->next->next;
        else
            a = a->next;
    }
    nvl_close(heap);
}
```

NVRAM Technology Continues to Improve – Driven by Market Forces



designlines MEMORY

News & Analysis

3D NAND Production Starts at Samsung

Peter Clarke

8/6/2013 08:05 AM EDT
16 comments

NO RATINGS
1 saves
LOGIN TO RATE

Like 17 Tweet 7 Share 10 +1 3

LONDON — Samsung Electronics Co. Ltd. has begun mass production of a 128 Gbit NAND flash memory that is integrated in multiple layers, and claims that it is the first company to do so.

The memory is based on a charge-trap cell rather than the conventional floating gate non-volatile cell used in 2D NAND flash. In the vertical arrangement this charge-trap cell shows increased reliability between a factor of 2 and a factor of 10 over conventional floating-gate NAND flash memory, Samsung claimed in a [press release](#).

The technology is capable of stacking up 24 layers, but Samsung did not disclose whether it had in 2D memory.

The company improvements is suitable for a applications in drives.

The V-NAND co

designlines MEMORY

News & Analysis

3D NAND Transition: 15nm Process Technology Takes Shape

Gary Hilson

5/13/2014 08:15 AM EDT
5 comments

NO RATINGS
LOGIN TO RATE

Like 15 Tweet 6 Share 6 +1 1

TORONTO — With 3D NAND unlikely to make economic sense until at least 2015, SanDisk and its flash foundry partner Toshiba both recently announced 15nm process technologies to produce NAND flash.

SanDisk's 1Z-nm technology will be applied to 1 and 3-bit-per-cell NAND flash memory architecture. Production ramp to begin in the second half of 2015. The technology scales chips along both axes, and a broad range of SanDisk offerings, from remote enterprise SSDs.

Toshiba's new process replaces its 19nm process. It is aimed at providing a transitional step to 3D NAND.

Forbes / Tech

JUL 28, 2015 @ 2:46 PM 7,391 VIEWS

Intel And Micron Jointly Announce Game-Changing 3D XPoint Memory Technology

Electronic Components
A new process works in
circuitry technology to create
ed as chips formed with
nology, but boost the data
d — 1.3 times faster — by

floating gate

China's Tsinghua Unigroup plans \$23B bid for Micron Technology

CNBC.com staff | @CNBC
Monday, 13 Jul 2015 | 8:41 PM ET
CNBC



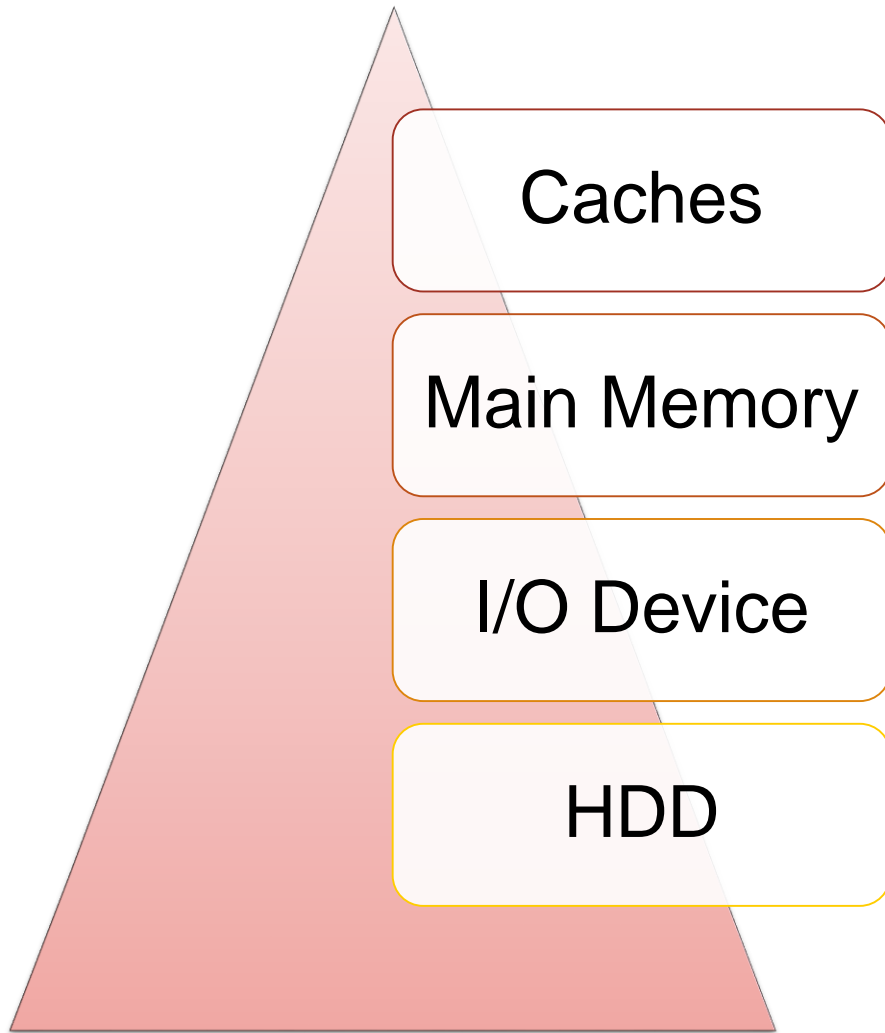
http://www.eetasia.com/STATIC/ARTICLE_IMAGES/201212/EEOL_2012DEC28_STOR_MFG_NT_01.jpg

Comparison of Emerging Memory Technologies

	SRAM	DRAM	eDRAM	2D NAND Flash	3D NAND Flash	PCRAM	STTRAM	2D ReRAM	3D ReRAM
Data Retention	N	N	N	Y	Y	Y	Y	Y	Y
Cell Size (F ²)	50-200	4-6	19-26	2-5	<1	4-10	8-40	4	<1
Minimum F demonstrated (nm)	14	25	22	16	64	20	28	27	24
Read Time (ns)	< 1	30	5	10 ⁴	10 ⁴	10-50	3-10	10-50	10-50
Write Time (ns)	< 1	50	5	10 ⁵	10 ⁵	100-300	3-10	10-50	10-50
Number of Rewrites	10 ¹⁶	10 ¹⁶	10 ¹⁶	10 ⁴ -10 ⁵	10 ⁴ -10 ⁵	10 ⁸ -10 ¹⁰	10 ¹⁵	10 ⁸ -10 ¹²	10 ⁸ -10 ¹²
Read Power	Low	Low	Low	High	High	Low	Medium	Medium	Medium
Write Power	Low	Low	Low	High	High	High	Medium	Medium	Medium
Power (other than R/W)	Leakage	Refresh	Refresh	None	None	None	None	Sneak	Sneak
Maturity									

Intel/Micron Xpoint?

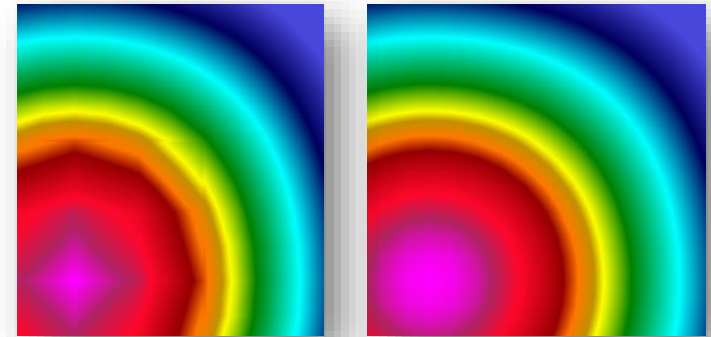
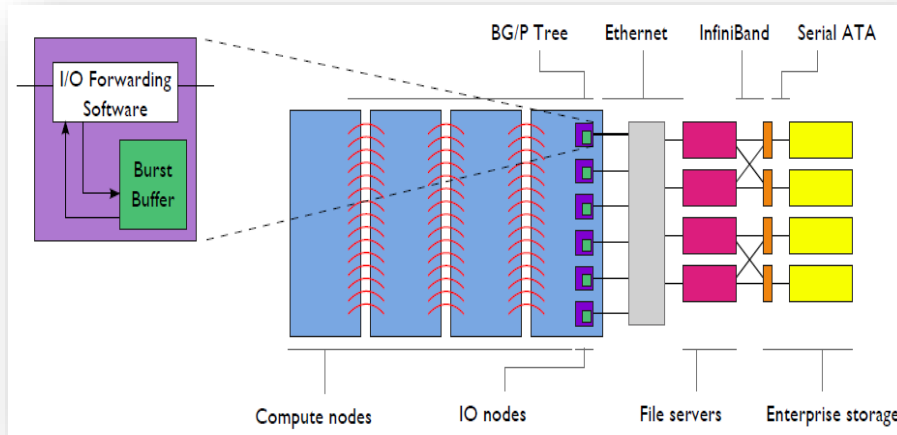
As NVM improves, it is working its way toward the processor core



- **Newer technologies improve**
 - density,
 - power usage,
 - durability
 - r/w performance
- **In scalable systems, a variety of architectures exist**
 - NVM in the SAN
 - NVM nodes in system
 - NVM in each node

Opportunities for NVM in Emerging Systems

- Burst Buffers, C/R_[Liu, et al., MSST 2012]
- In situ visualization



<http://ft.ornl.gov/eavl>

- In-mem tables

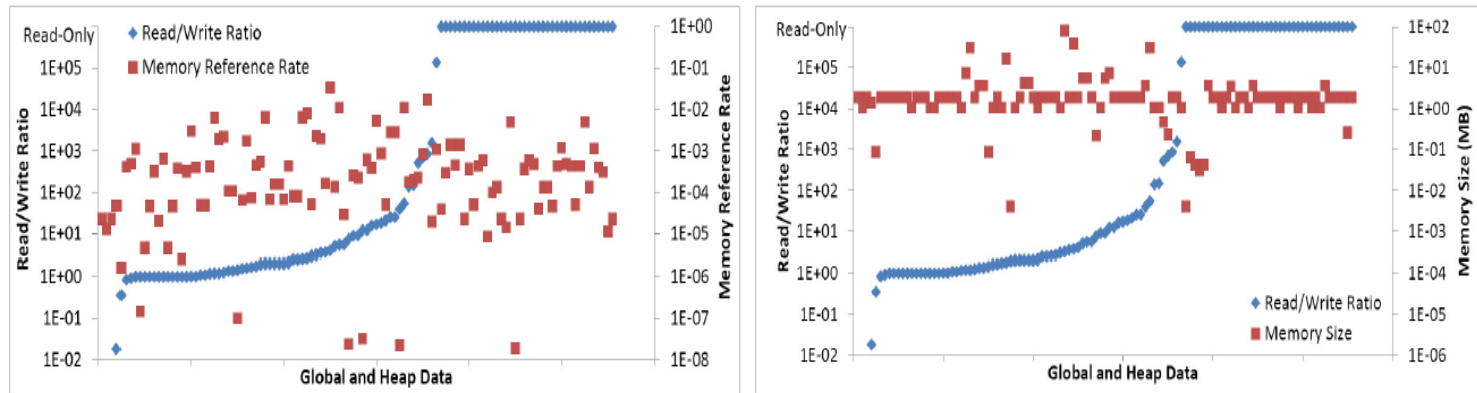


Figure 3: Read/write ratios, memory reference rates and memory object sizes for memory objects in Nek5000

Programming NVM Systems

Design Goals for NVM Programming Design

- **Active area of research**
 - See survey
- **Architectures will vary dramatically**
 - How should we design the node?
 - Portable across various NVM architectures
- **Performance for HPC scenarios**
 - Allow user or compiler/runtime/os to exploit NVM
 - Asymmetric R/W
 - Remote/Local
- **Security**
- **Assume lower power costs under normal usage**
- **Correctness and durability**
 - Enhanced ECC for NVM devices
 - A crash or erroneous program could corrupt the NVM data structures
 - Programming system needs to provide support for this model
- **ACID**
 - Atomicity: A transaction is “all or nothing”
 - Consistency: Takes data from one consistent state to another
 - Isolation: Concurrent transactions appears to be one after another
 - Durability: Changes to data will remain across system boots

MPI and OpenMP do not solve this problem.

10.1109/TPDS.2015.2442980

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTING SYSTEMS

A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems

Sparsh Mittal, Member, IEEE, and Jeffrey S. Vetter, Senior Member, IEEE

Abstract—Non-volatile memory (NVM) devices, such as Flash, phase change RAM, spin transfer torque RAM, and resistive RAM, offer several advantages and challenges when compared to conventional memory technologies, such as DRAM and magnetic hard disk drives (HDDs). In this paper, we present a survey of software techniques that have been proposed to exploit the advantages and mitigate the disadvantages of NVMs when used for designing memory systems, and, in particular, secondary storage (e.g., solid state drive) and main memory. We classify these software techniques along several dimensions to highlight their similarities and differences. Given that NVMs are growing in popularity, we believe that this survey will motivate further research in the field of software technology for NVMs.

Index Terms—Review, classification, non-volatile memory (NVM) (NVRAM), flash memory, phase change RAM (PCRAM), spin transfer torque RAM (STT-RAM) (STT-MRAM), resistive RAM (ReRAM) (RRAM), storage class memory (SCM), Solid State Drive (SSD).

♦

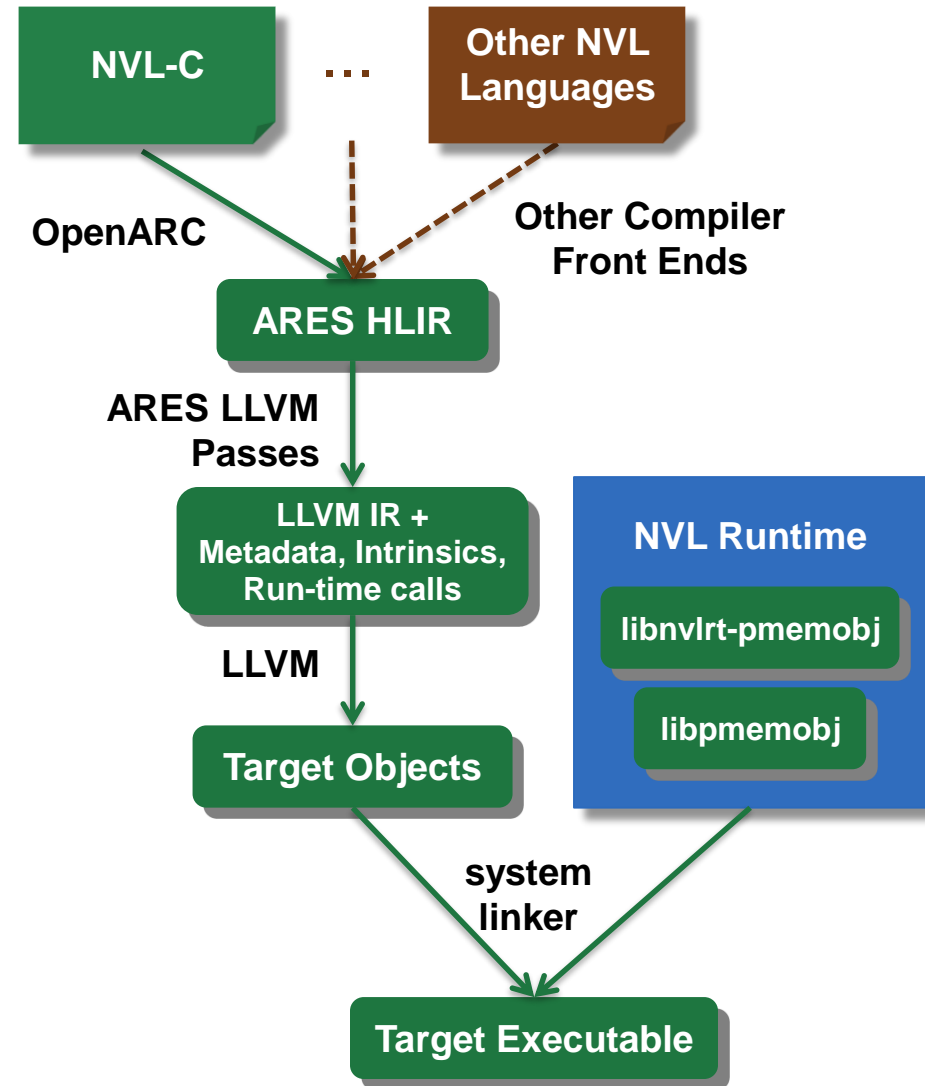
NVL-C: Portable Programming for NVMM

- Minimal, familiar, programming interface:
 - Minimal C language extensions.
 - App can still use DRAM.
- Pointer safety:
 - Persistence creates new categories of pointer bugs.
 - Best to enforce pointer safety constraints at compile time rather than run time.
- Transactions:
 - Prevent corruption of persistent memory in case of application or system failure.
- Language extensions enable:
 - Compile-time safety constraints.
 - NVM-related compiler analyses and optimizations.
- LLVM-based:
 - Core of compiler can be reused for other front ends and languages.
 - Can take advantage of LLVM ecosystem.

```
#include <nvl.h>
struct list {
    int value;
    nvl struct list *next;
};
void remove(int k) {
    nvl_heap_t *heap
        = nvl_open("foo.nvl");
    nvl struct list *a
        = nvl_get_root(heap, struct list);
    #pragma nvl atomic
    while (a->next != NULL) {
        if (a->next->value == k)
            a->next = a->next->next;
        else
            a = a->next;
    }
    nvl_close(heap);
}
```

NVL-C: Reliable Programming for NVM

- NVL-C is a novel NVM programming system that extends C.
- Currently supports multiple namespaces, dynamic allocations, and transactions.
- Critical compiler components are implemented as reusable LLVM extensions.
- Future work:
 - NVL-Fortran, NVL-C++, etc.
 - Target other persistent memory libraries.
 - Contribute components to LLVM project.



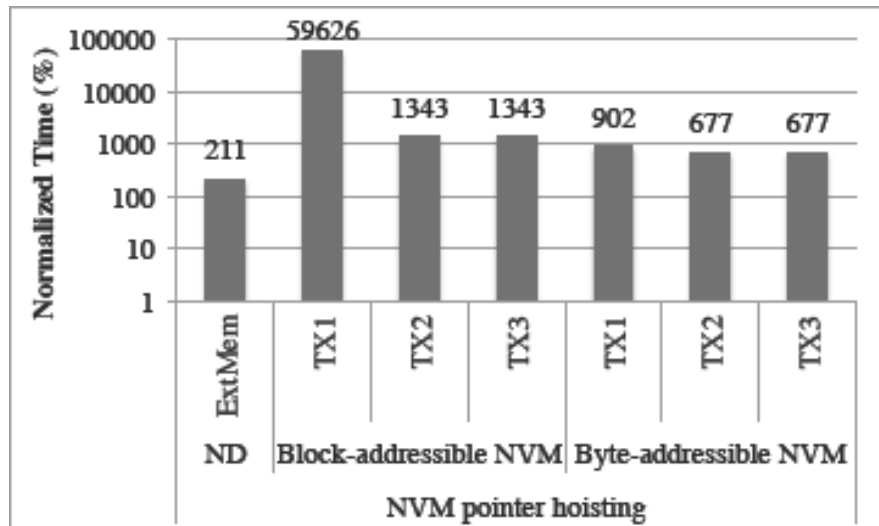
Preliminary Results

- Applications extended with NVL-C
- Compiled with NVL-C
- Executed on Fusion ioScale
- Compared to DRAM
- Various levels of optimization

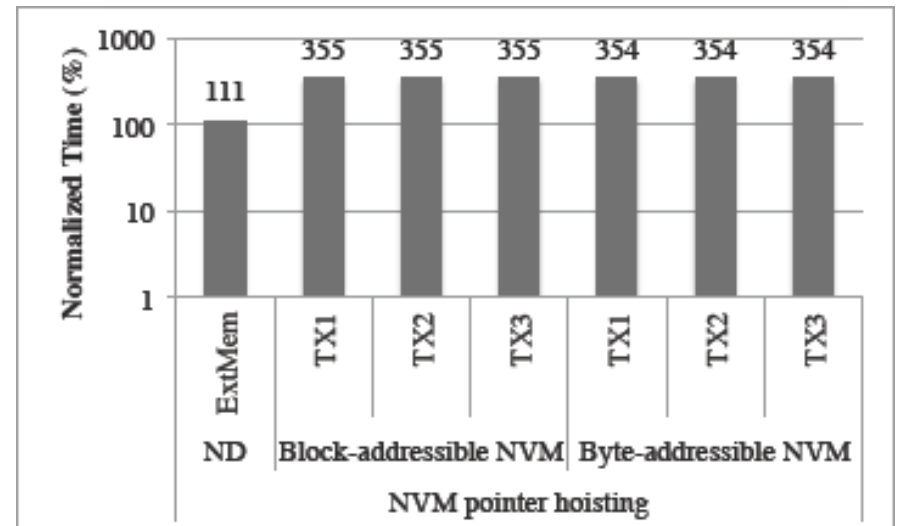
Table 3: Symbols Used in the Result Figures

Symbol	Description
ExtMem or ExM	Use persistent storage as if extended DRAM
No Durability or ND	Skip runtime operations for durability
Base or B	Basic NVL-C version w/o Safety, RefCnt, and transaction (TX0, TX1, ...)
Safety or S	Automatic pointer-safety checking
RefCnt or R	Automatic reference counting
TX0	B+S+R + Enforce only durability of each NVM write
TX1	B+S+R + Enforce ACID properties of each transaction
TX2	TX1 + aggregated transaction using backup clauses
TX3	TX2 + skipping unnecessary backup using clobber clauses
TX4	TX3 at the granularity of each loop
CLFlush	Flush cache line to memory
MSync	Synchronize memory map with persistent storage

LULESH

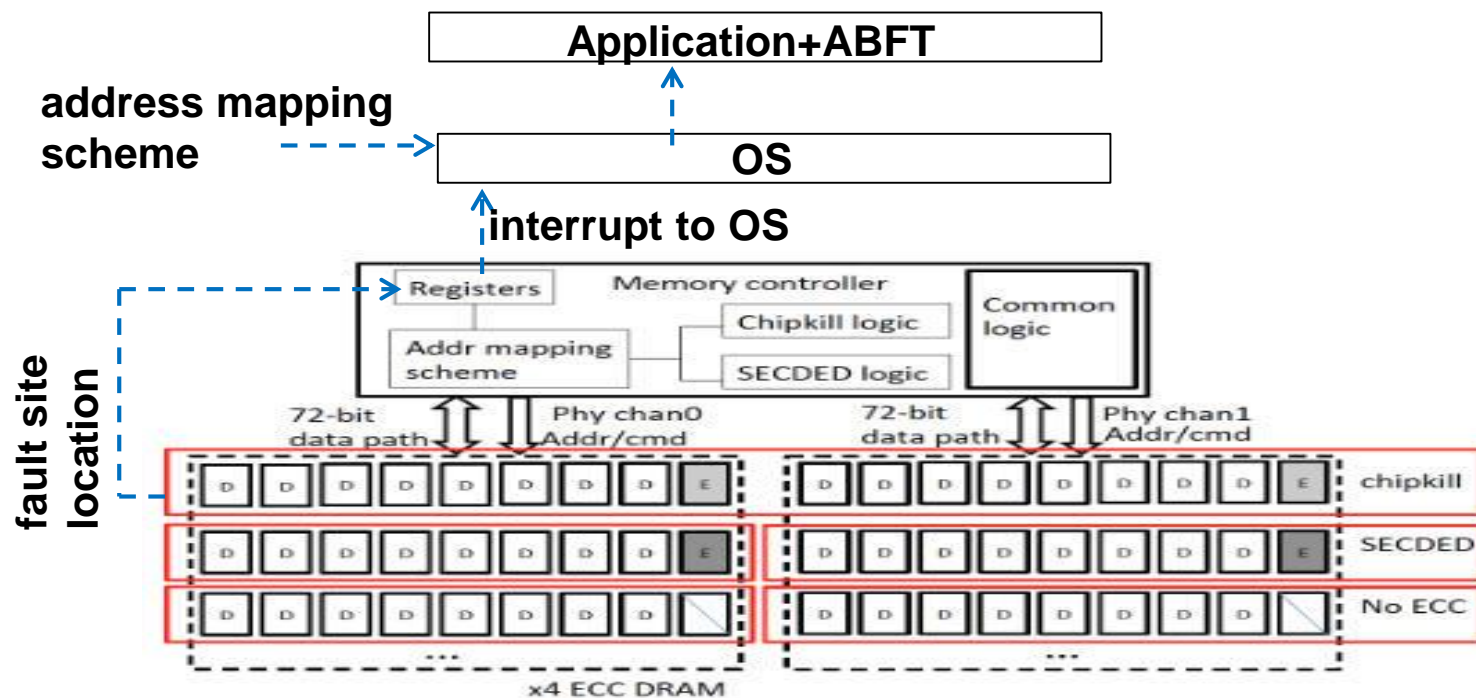


XSBENCH



A Word on ECC

Mixed Mode Memories Require User Control



- ABFT algorithms guard important data structures in no-ecc area
- Normal/extended ECC guards critical data structures
- Potential power, performance, cost improvement with different memories
- User places data in appropriate location

Overall Observations and Implications

- **“Exciting” times in computer architecture**
 - Heterogeneous cores
 - Multimode memory systems
 - Fused memory systems
 - I/O architectures
 - Error correction
 - Changing system balance
- **Uncertainty, Ambiguity**
 - How do we design future systems so that they are faster than current systems on mission applications?
 - *Entirely possible that the new system will be slower than the old system!*
 - How do we provide some level of performance portability for applications teams?
 - How do we understand reliability and performance problems?
- ***Managing complexity is our main challenge!***

Session Questions

- - What's the role of the OS and runtime system(s) in managing the memory hierarchy?
- - What application interfaces are needed to help manage the memory hierarchy?
- - What level of detail should the OS expose about the memory hierarchy?
- - To what level of the software stack should the OS expose details of the memory hierarchy?
- - What memory management functions should the runtime system contain?
- - How flexible or adaptable do memory management policies need to be?



Acknowledgements

- Contributors and Sponsors

- Future Technologies Group:
<http://ft.ornl.gov>
- US Department of Energy Office of Science
 - DOE Vancouver Project:
<https://ft.ornl.gov/trac/vancouver>
 - DOE Blackcomb Project:
<https://ft.ornl.gov/trac/blackcomb>
 - DOE ExMatEx Codesign Center:
<http://codesign.lanl.gov>
 - DOE Cesar Codesign Center:
<http://cesar.mcs.anl.gov/>
 - DOE Exascale Efforts:
<http://science.energy.gov/ascr/research/computer-science/>
- Scalable Heterogeneous Computing Benchmark team: <http://bit.ly/shocmarx>
- US National Science Foundation Keeneland Project: <http://keeneland.gatech.edu>
- US DARPA
- NVIDIA CUDA Center of Excellence

