Simulating Stencil-based Application on Future Xeon-Phi Processor

PMBS workshop at SC'15

Chitra Natarajan
Intel Corporation

Carl Beckmann
Intel Corporation

Anthony Nguyen
Intel Corporation

Mauricio Araya-Polo
Shell Intl. E&P Inc.

Tryggve Fossum
Intel Corporation

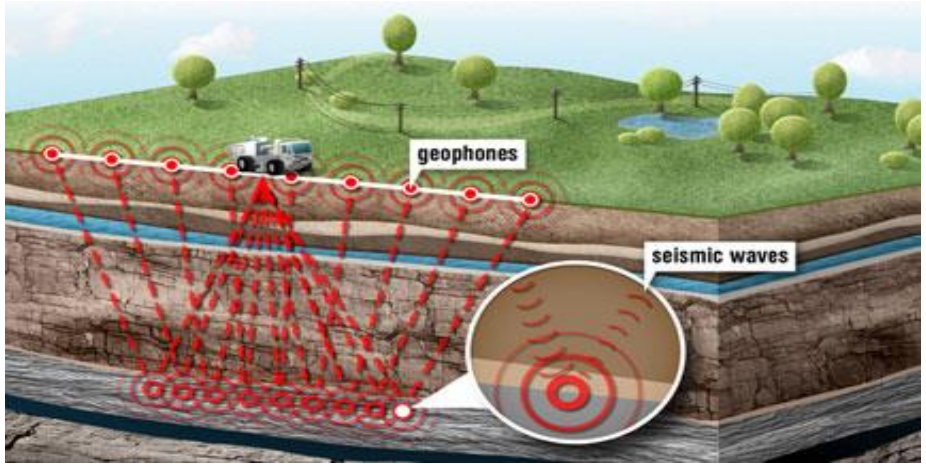
Detlef Hohl
Shell Intl. E&P Inc

Introduction

- Software/Hardware Co-design
  - Simulate high-value software portfolio ahead of hardware availability
  - Collaborative effort to influence both future software and hardware development
  - Target Software: Stencil-based O&G hydrocarbon exploration application
  - Target Hardware: Xeon Phi processor
- Outline
  - Stencil-based O&G hydrocarbon exploration application
  - Knights Landing (KNL) Xeon Phi processor
  - Cycle-Accurate Models (CAM) & Fast-Abstract Models (FAM)
  - Correlation of CAM to real system for an existing processor (Xeon SNB)
  - Correlation of FAM to CAM for KNL
  - CAM/FAM KNL simulation results

O&G Hydrocarbon Exploration Target Application

- Data acquisition, on/off shore

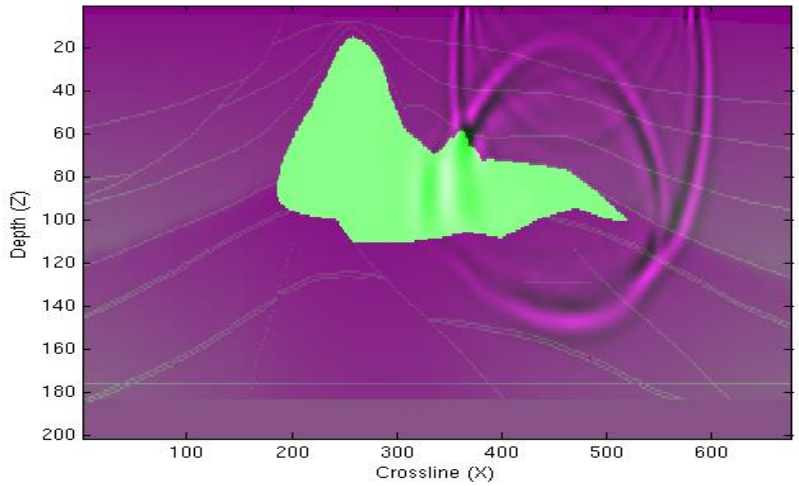


- Seismic Imaging, Wave Equations (Du, Fletcher, and Fowler, EAGE 2010) VTI assumption.

$$\frac{\partial^2 p}{\partial t^2} = V_x^2\left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}\right) + V_z \frac{\partial^2 q}{\partial z^2}$$

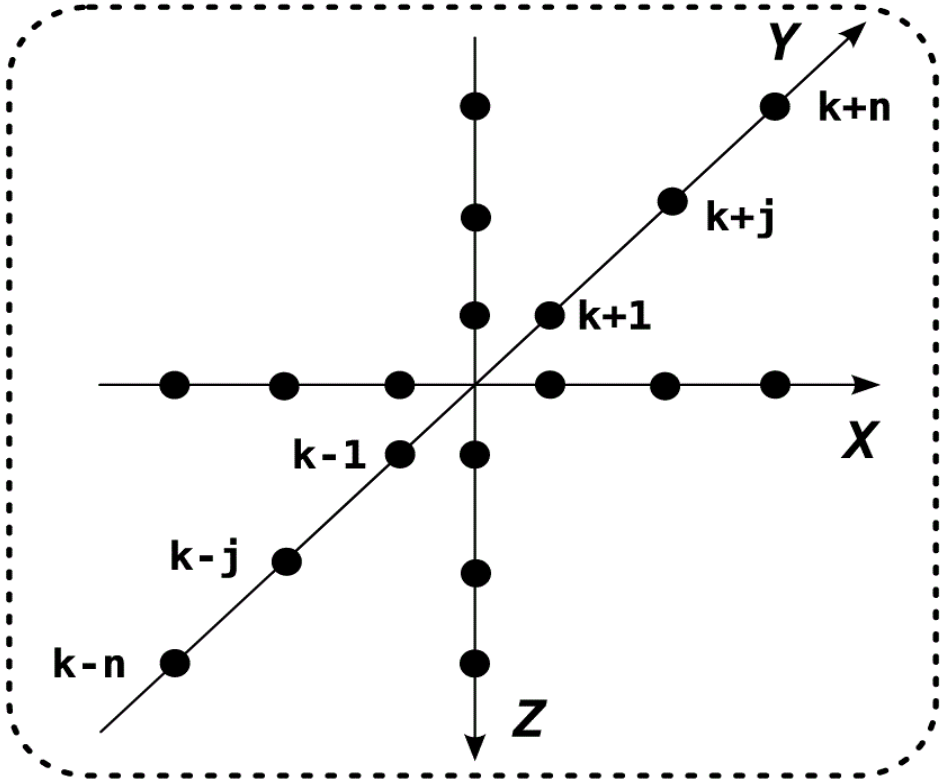
$$\frac{\partial^2 q}{\partial t^2} = V_n^2\left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}\right) + V_z \frac{\partial^2 q}{\partial z^2}$$

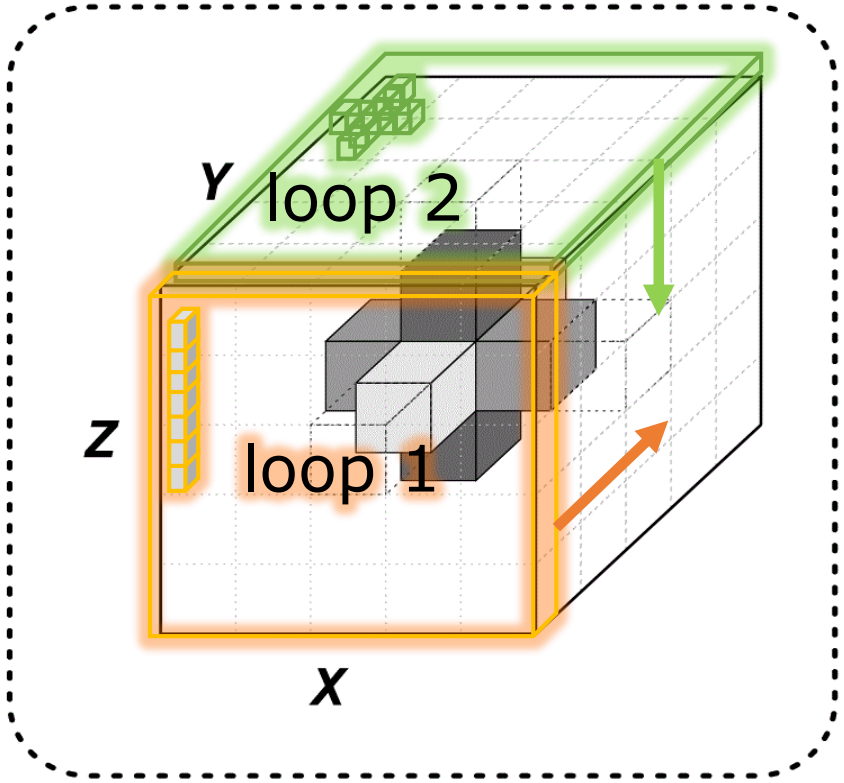
$$V_n = V_z\sqrt{1+2\delta}, \quad V_x = V_z\sqrt{1+2\varepsilon}$$



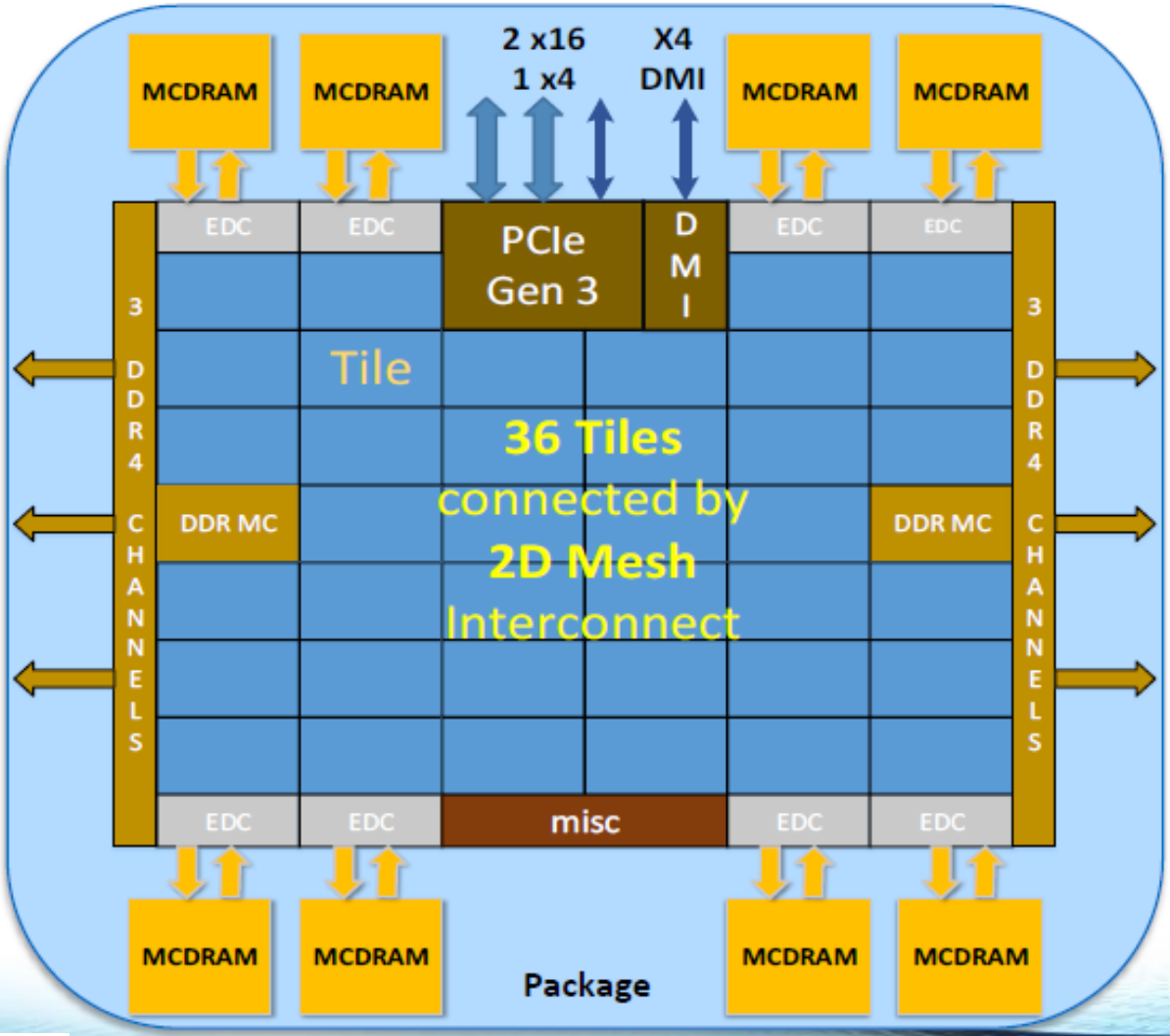
O&G Hydocarbon Exploration Target Application

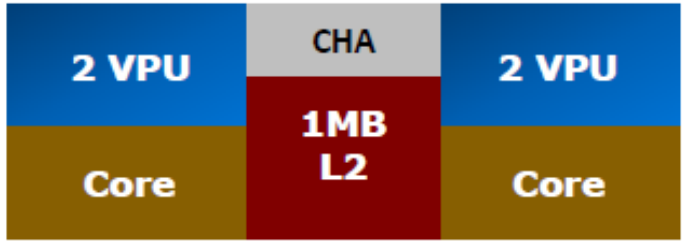
1. MPI+X model, in this work X=OpenMP, and only 1-process behavior is analyzed
2. Wave equation PDE solved explicitly, stencil-based code, high-order 24-24-16
3. Implemented as two major loops: loop1 (sweeping Z) & loop2 (sweeping X & Y)
4. Key issues: data dependency (memory bound) and low data reuse





Knights Landing Overview





Chip: 36 Tiles interconnected by **2D Mesh**

Tile: 2 Cores + 2 VPU/core + 1 MB L2

Memory: MCDRAM: 16 GB on-package; High BW

DDR4: 6 channels @ 2400 up to 384GB

IO: 36 lanes PCIe Gen3. 4 lanes of DMI for chipset

Node: 1-Socket only

Fabric: Omni-Path on-package (not shown)

Vector Peak Perf: 3+TF DP and 6+TF SP Flops

Scalar Perf: ~3x over Knights Corner

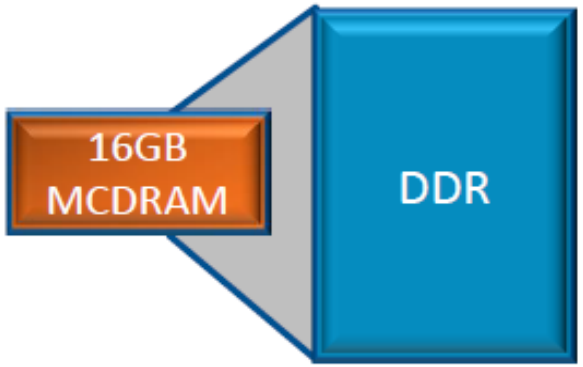
Streams Triad (GB/s): MCDRAM : 400+; DDR: 90+

Source Intel: All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. KNL data are preliminary based on current expectations and are subject to change without notice. 1Binary Compatible with Intel Xeon processors using Haswell Instruction Set (except TSX). 2Bandwidth numbers are based on STREAM-like memory access pattern when MCDRAM used as flat memory. 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.

Memory Modes

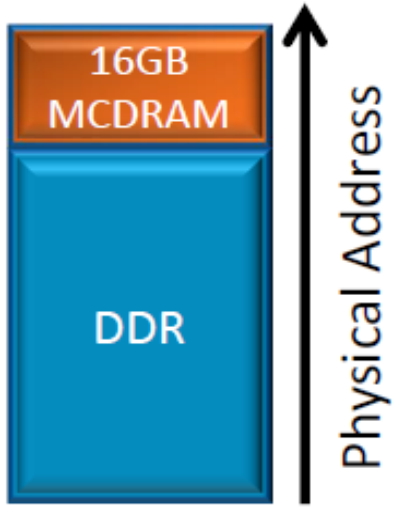
Three Modes. Selected at boot

Cache Mode



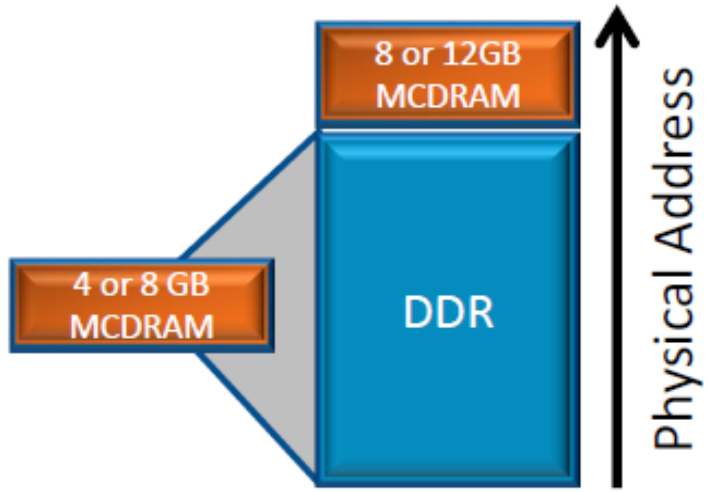
- SW-Transparent, Mem-side cache
- Direct mapped. 64B lines.
- Tags part of line
- Covers whole DDR range

Flat Mode



- MCDRAM as regular memory
- SW-Managed
- Same address space

Hybrid Mode



- Part cache, Part memory
- 25% or 50% cache
- Benefits of both

Cycle Accurate Model (CAM) vs. Fast Abstract Model (FAM)

Cycle Accurate Model (CAM)

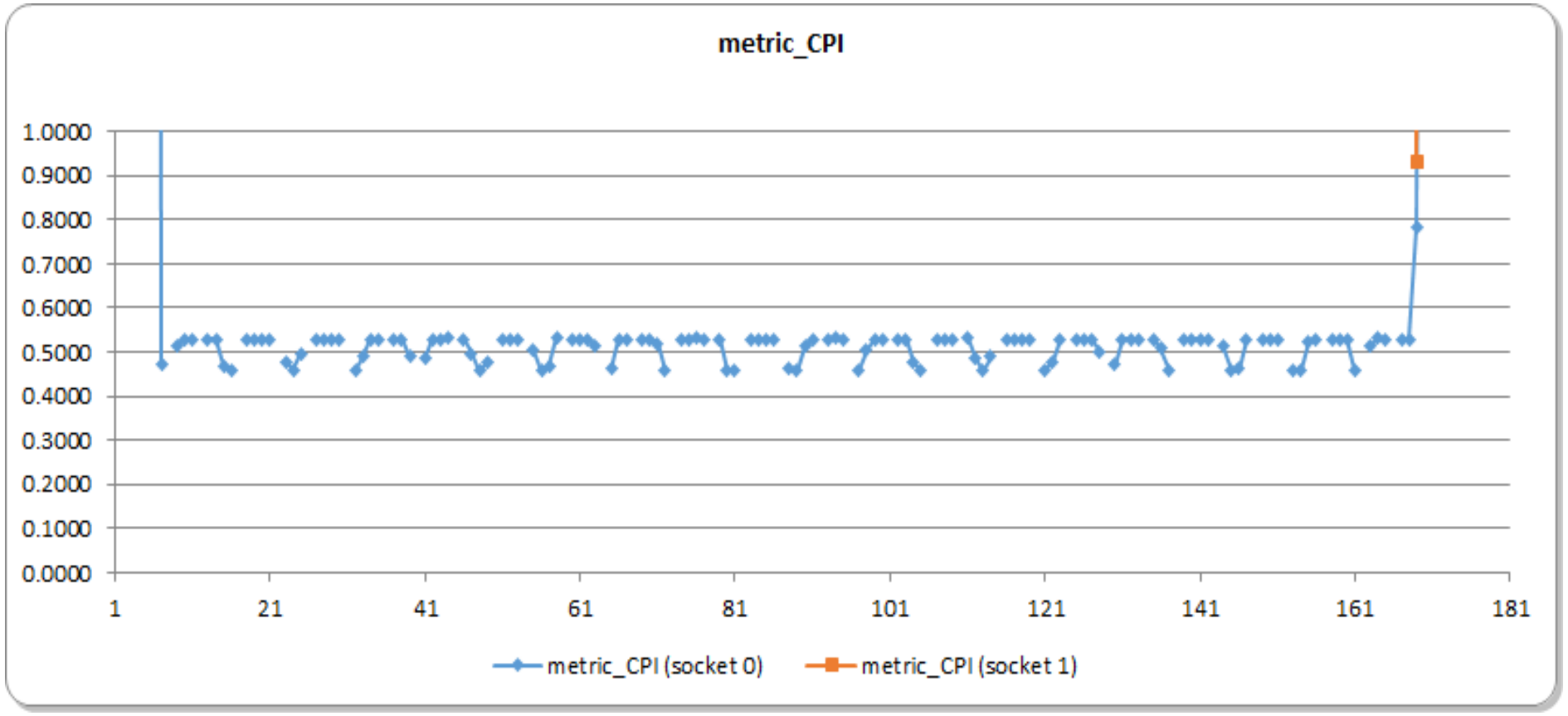
- Cycle accurate performance model
  - Validated extensively against silicon
  - Developed by product design teams across generations over many years
- Slow simulation speed
  - ~1K instructions per real second
  - Difficult to simulate more than a few 10's of million instructions per test
  - Difficult to scale to > few 10's of threads
- Primarily used trace-driven method
  - Execution-driven method added
  - Uses Intel SDE as functional emulator

Fast Abstract Model (FAM)

- Do not model in cycle accurate detail
  - Correlated against CAM
  - Accuracy vs. CAM ~ +/- 20% over a wide range of ST workloads
- Trades accuracy for speed
  - ~ 100K – 10M instructions per second
  - Can simulate 10's of billions of instructions per test
  - Simulates multiple cores and threads
- Methodologies supported
  - Trace-driven
  - Execution-driven

Xeon SNB E5-2690 EMON CPI Data for 20 Timesteps

- CPI (Cycles Per Instruction)
  - Can clearly observe the 20 time steps, with ~2/3rd of each at CPI of ~0.53x and ~1/3rd at ~0.46x
  - The 2 CPI levels reflect the 2 loops per time step



CAM Model to Real System Correlation on Xeon SNB

- Representative Simpoints-based tracing resulted in 5 regions/traces
  - As expected, 2 traces dominate corresponding to the 2 loops with ~70% and ~29% weights
  - 20 time step execution resulted in ~138.6B instructions
- Good correlation of CAM simulation data to real system measurement data
  - CPI & LLC MPI (Last-Level Cache Misses Per Instruction) within 2%, overall runtime within 3%

regions/ traces	weights	CAM sims data			
		CPI	LLC MPI	MC Rd BW	MC Wr BW
r1/t1	0.001	4.529	0.154	8277.4	7940.4
r2/t2	0.697	0.528	0.0095	4398.0	1122.1
r3/t3	0.001	3.462	0.137	9599.1	9581.0
r4/t4	0.008	0.488	0.017	8564.6	4161.4
r5/t5	0.292	0.483	0.006	3191.9	3184.8
	Sims: wtd avg	0.522	0.0090	4088.2	1765.8
	EMON	0.528	0.0088	3878.1	1291.3
	corr vs. m/s	-1.2%	1.7%	5.4%	36.7%
Projected runtime in secs		19.04	[using PL = 138.6B instr]		
Measured runtime in secs		18.5			
Runtime corr vs. m/s		3%			

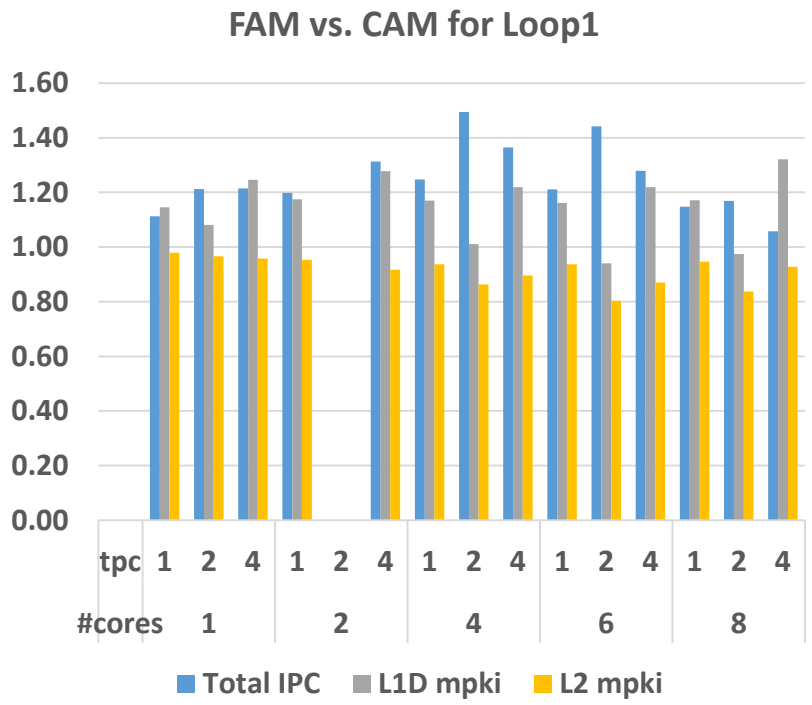
FAM vs. CAM correlation for KNL

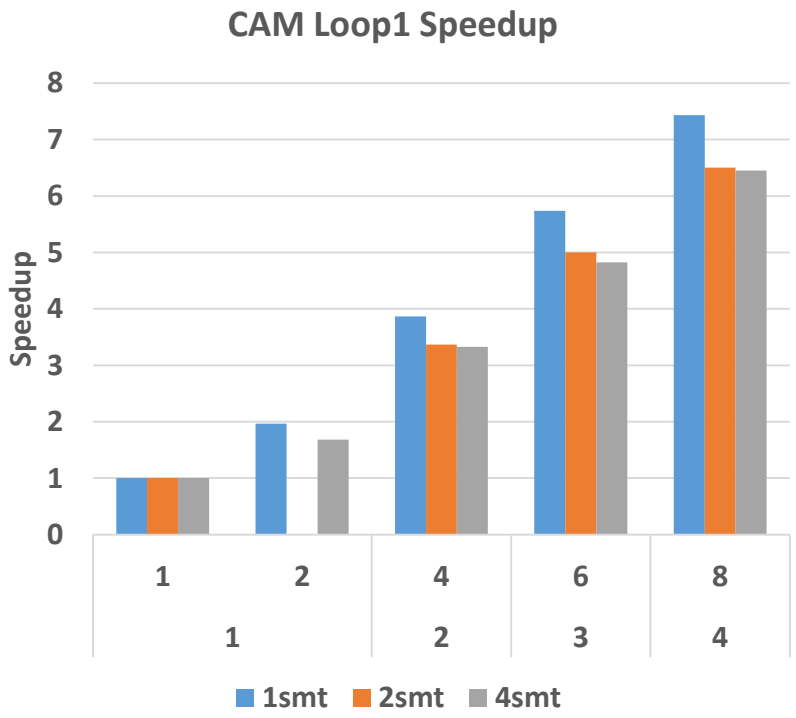
Configuration simulated:

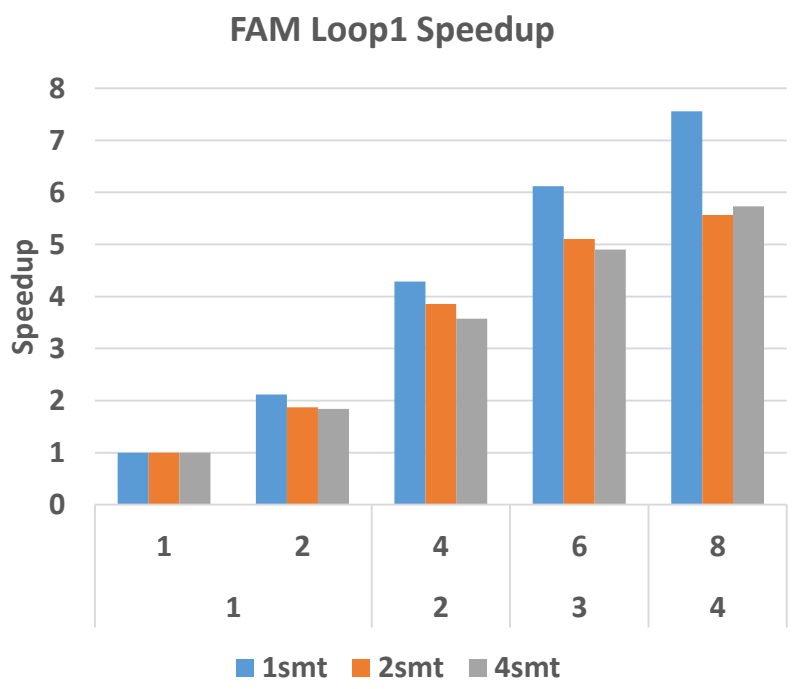
- Xeon Phi “Knights Landing” core
- 1 to 8 cores
- 2 cores per tile
- 1 to 4 SMT threads per core

Metrics compared:

- IPC
- L1 and L2 cache miss rates
- Speedup





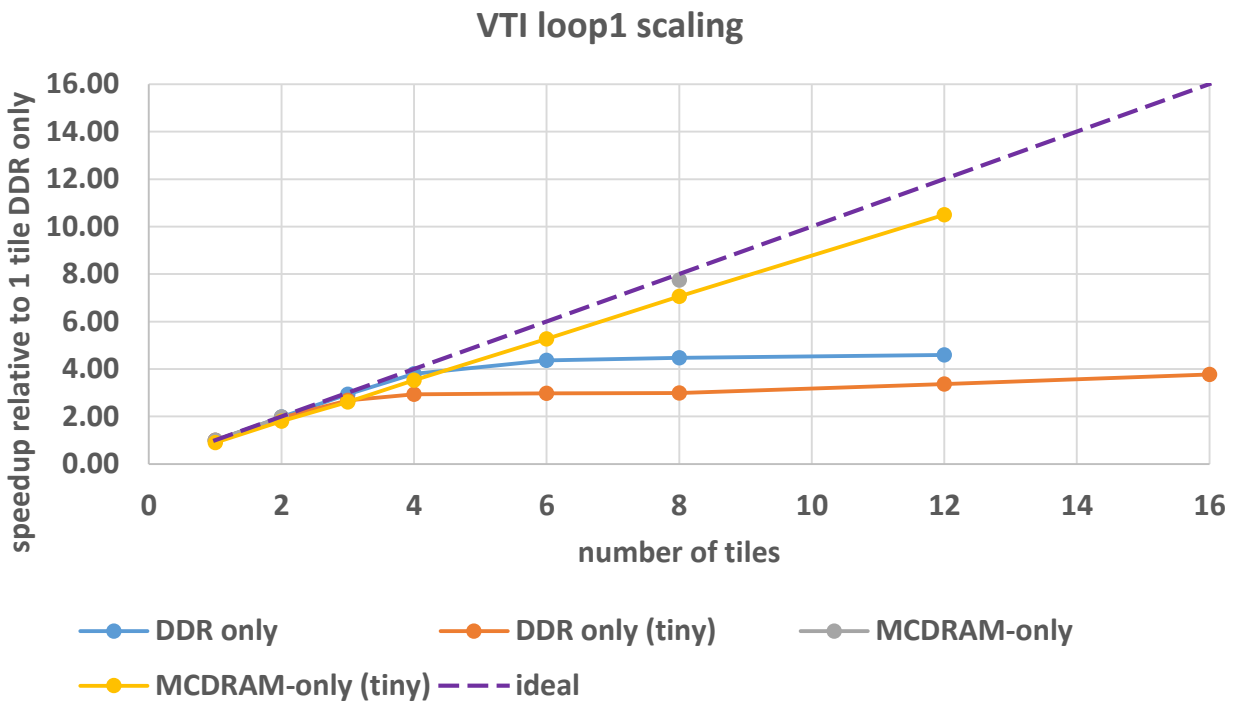


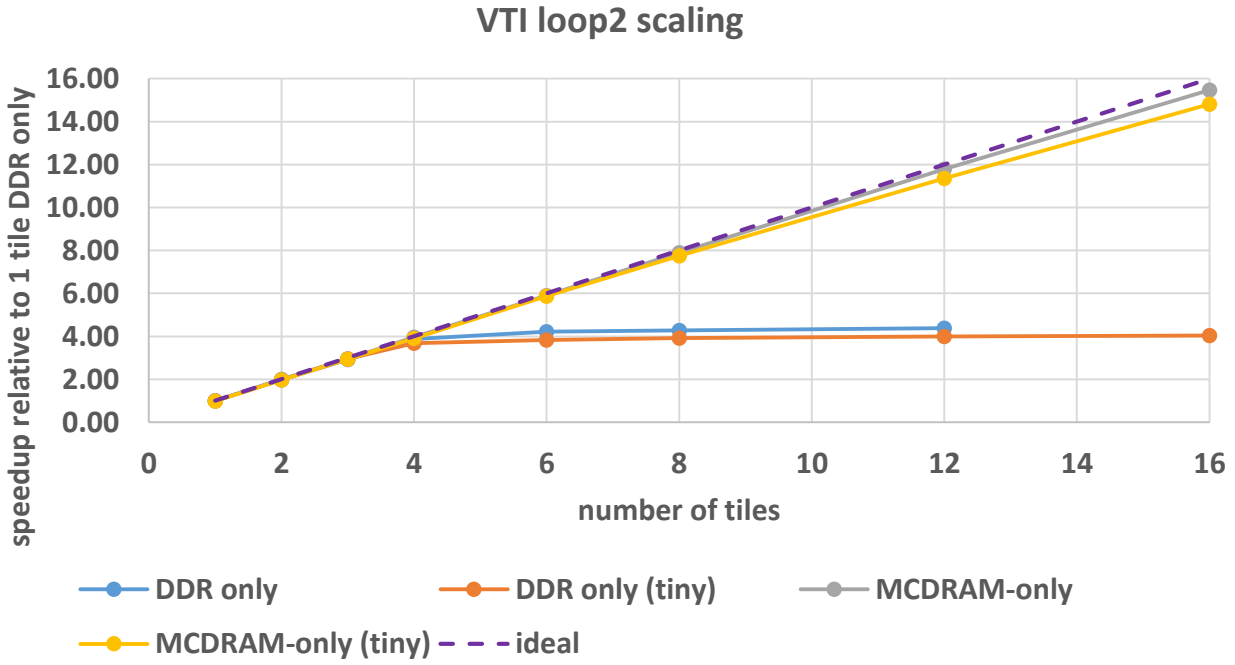
- **Correlation typically in the ~20% range for 1T, but worsens with SMT**
- **FAM vs. CAM speedup trends are similar to each other**

Tile scaling study on CAM

- Cycle accurate model experiments
  - 1 to 16 tiles (2 to 32 cores)
  - Execution driven
  - Cache sharing modeled accurately
- Two main loops simulated partially
  - Only 3 loop iterations per thread due to simulation time limits
  - More than enough to warm up L2 caches
- Stencils-per-second figure of merit
  - Measured time to complete fixed amount of work

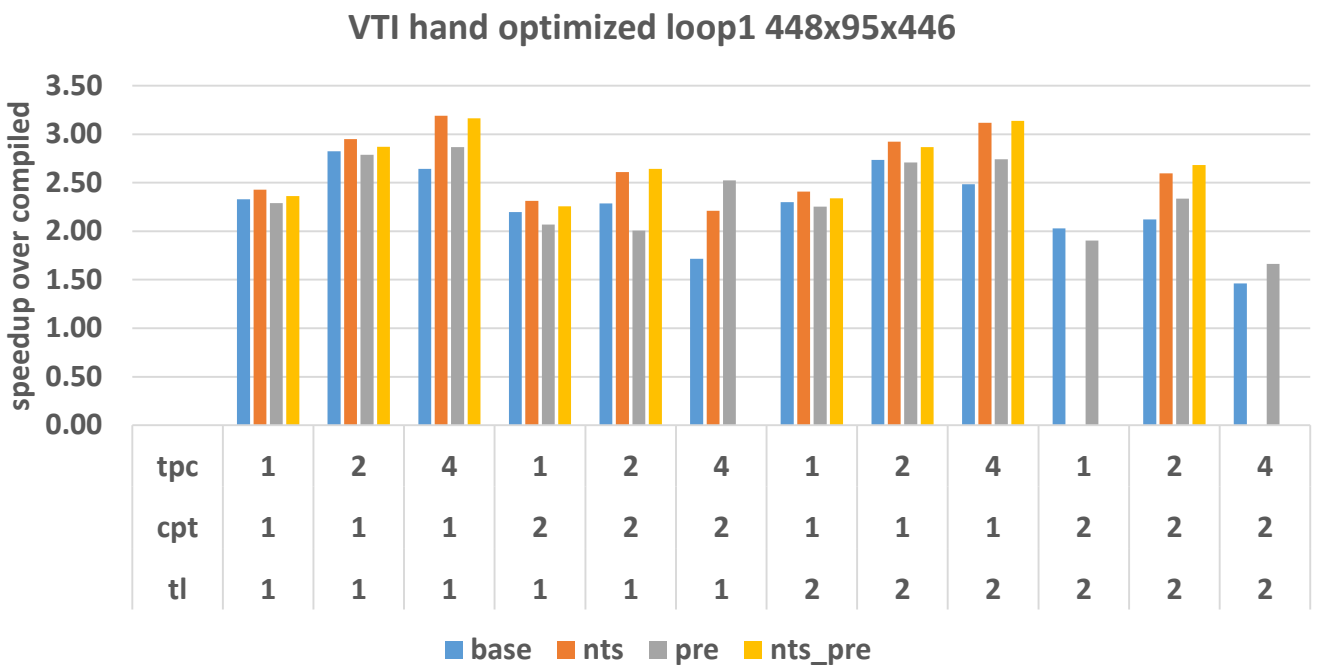
- **DDR-only: Tile scaling limited to ~4 due to BW limits**
- **MCDRAM-only: Tile scaling quite good for the full range that could be simulated**

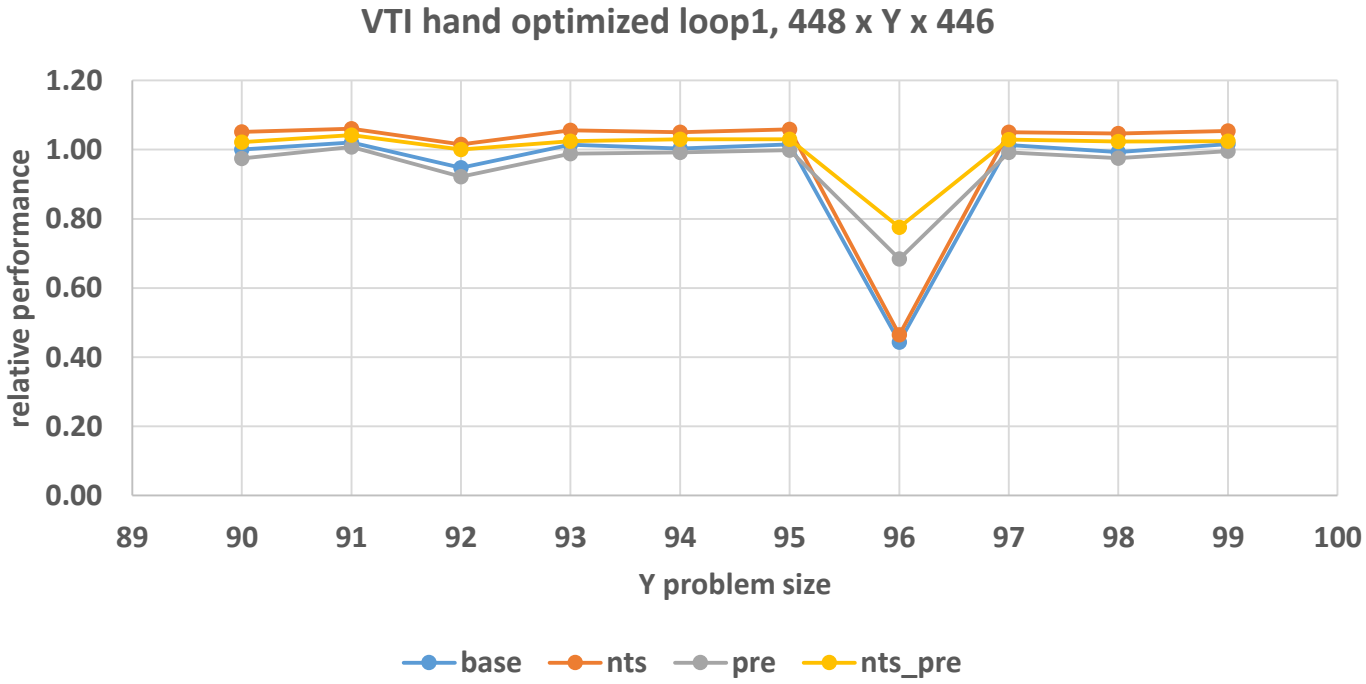




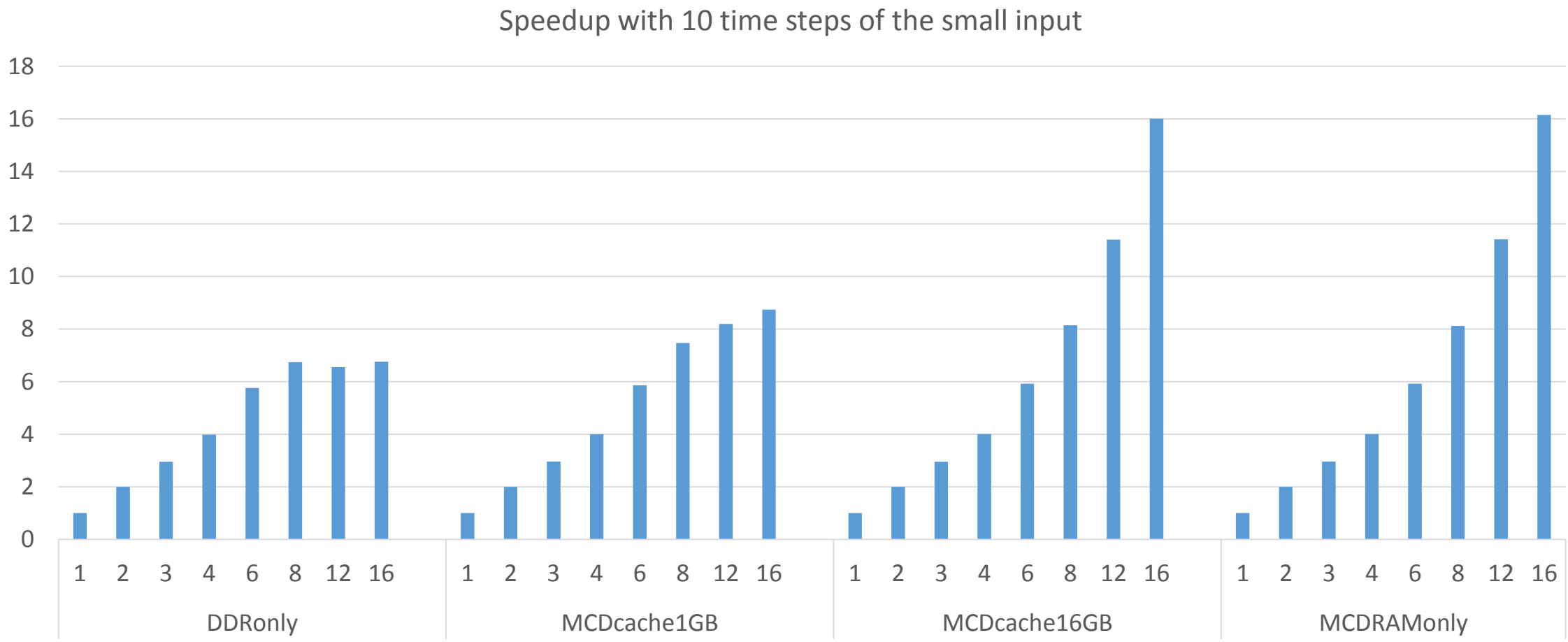
Hand optimization study on CAM

- Loop 1
  - 1-D vertical 16th-order stencil
- Compiled code performed poorly
  - 1.5B stencils/s theoretical roofline
  - Sims showed ~25% of theoretical
  - Inefficient use of cache & vectors
- Hand optimized code
  - Vectorize in x direction
  - Stripmine loop in z direction
  - Better reuse in AVX registers
  - Less L1 cache bandwidth
  - Achieved upto 3.0x speedup
  - Z array size hazard observed!



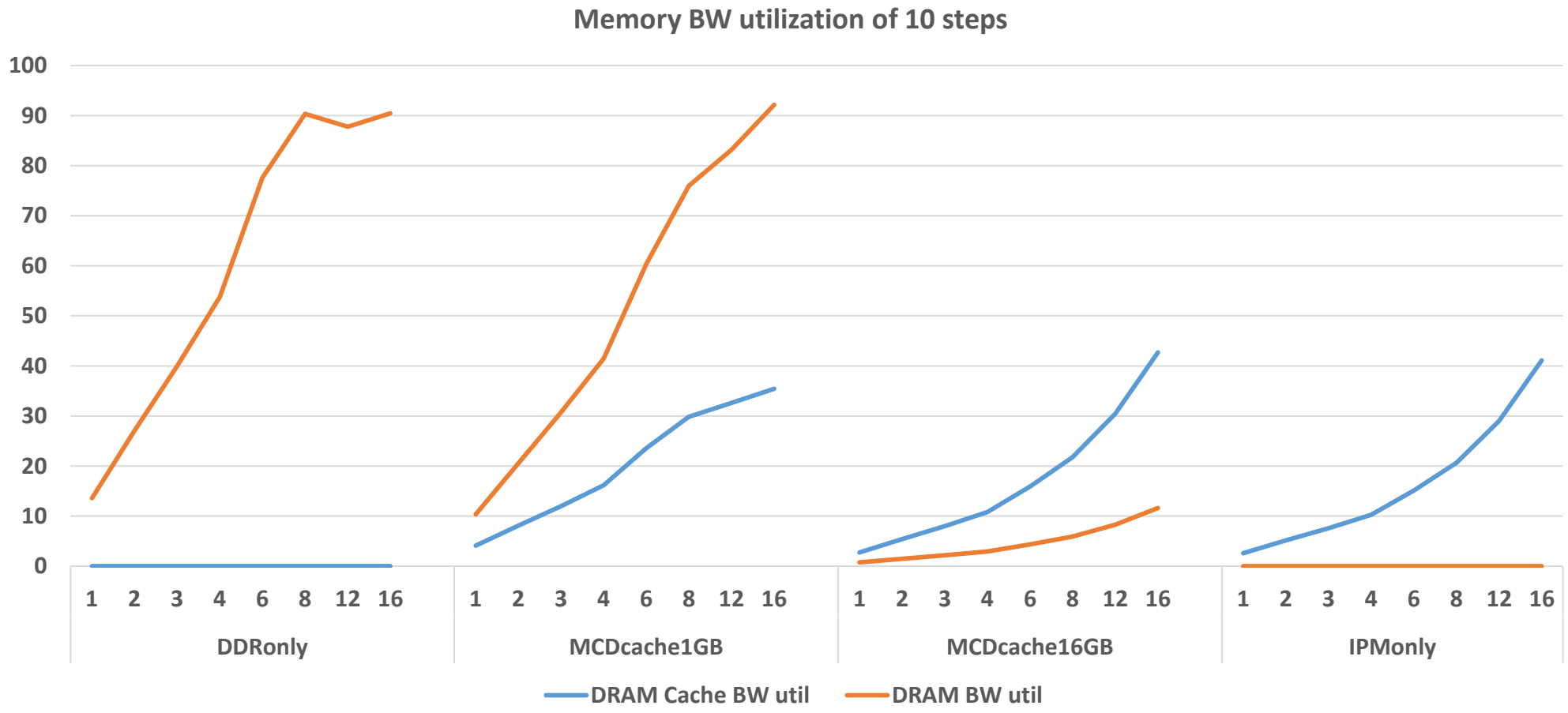


Impact of Memory Technologies Study using FAM



- **When working set (4GB) fits in MCDRAM (16GB), scaling for MCDRAM-as-cache approaches MCDRAM-only**

Memory Technology Study using FAM : Memory Bandwidth Utilization



- **When working set (4GB) fits in MCDRAM cache (16GB), DDR is accessed only once, so DDR BW not an issue**

Conclusion & Future Work

- Conclusion
  - Initial software/hardware co-design effort results presented
  - Used existing hardware for CAM model correlation & CAM/FAM models of future hardware
  - Co-design improved mutual understanding & optimization of software with hardware
    - Enabled code hand optimization performance study ahead of hardware
    - Enabled studying impact of new hardware memory features on target application ahead of hardware
- Future Work
  - Study multi-node distributed memory scenario for the target application
  - Co-design other future products – software & hardware

Acknowledgments

- Intel Corporation & Shell International
  - For allowing the work to be shared
- CAM & FAM modeling teams
  - For developing the models and supporting our use of them

Backup

Cycle Accurate Model (CAM)

- Cycle accurate performance model
  - Developed by product design teams across generations over many years
  - Validated against silicon
- Slow simulation speed
  - Approx. 1,000 simulated instructions per real second
  - Difficult to simulate more than a few tens of million instructions per experiment
  - Difficult to scale to more than a few tens of threads
- Primarily used by product teams with trace-driven methodology
  - Execution-driven methodology added in this project
  - Uses Intel SDE as functional emulator

Fast Accurate Model (FAM)

- Fast multithreaded performance model
  - Simulates multiple cores and threads
  - Simulator runs multithreaded
  - Approx. 100k – 10M instructions per second, depending on detail
- Trades accuracy for speed, correlated against CAM
  - Does not model in cycle accurate detail
  - Accuracy vs. CAM typically within +/- 20% over a wide range of ST workloads
- Methodologies supported
  - Trace-driven
  - Execution-driven