Accelerating 3D Elastic Wave Equations on Knights Landing based Intel Xeon Phi systems

Mohammed Sourouri[1]
Espen Birger Raknes[2,3]

Norwegian University of Science and Technology (NTNU)

April 25, 2017

[nvidia.com]

1 Department of Electronic Systems

2 AkerBP

3 Work conducted when part of Department of Geoscience and Petroleum

NTNU

Accelerating 3D Elastic Wave Equations on Knights Landing based Intel Xeon Phi systems

Mohammed Sourouri[1]
Espen Birger Raknes[2,3]

Norwegian University of Science and Technology (NTNU)

April 25, 2017

[nvidia.com]

[openacc.org]

[opencl.org]

1 Department of Electronic Systems

2 AkerBP

3 Work conducted when part of Department of Geoscience and Petroleum

NTNU

Accelerating 3D Elastic Wave Equations on Knights Landing based Intel Xeon Phi systems

Mohammed Sourouri[1]
Espen Birger Raknes[2,3]

Norwegian University of Science and Technology (NTNU)

April 25, 2017

1 Department of Electronic Systems

2 AkerBP

3 Work conducted when part of Department of Geoscience and Petroleum

[newsroom.intel.com]

NTNU

Accelerating 3D Elastic Wave Equations on Knights Landing based Intel Xeon Phi systems

Mohammed Sourouri[1]
Espen Birger Raknes[2,3]

Norwegian University of Science and Technology (NTNU)

April 25, 2017

1 Department of Electronic Systems

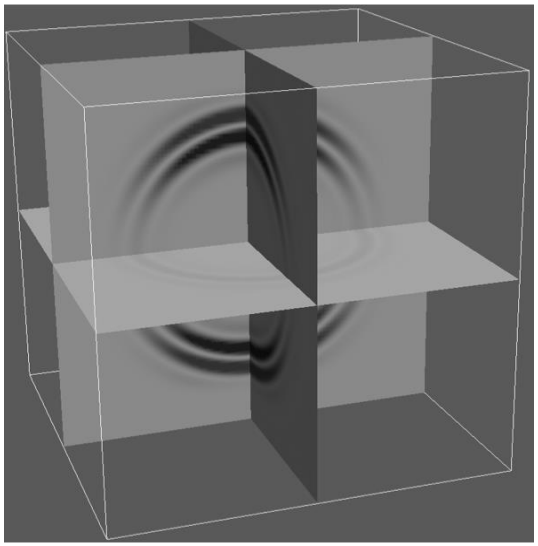
2 AkerBP

3 Work conducted when part of Department of Geoscience and Petroleum

	Tesla P100	Xeon Phi
FP32 SP	9.3 TFLOP/s	6 TFLOP/s
FP64 DP	4.7 TFLOP/s	3 TFLOP/s
Memory bandwidth	549 GB/s	475 GB/s 90 GB/s (DDR4)
Memory capacity	16 GB	16 GB 384 GB (DDR4)

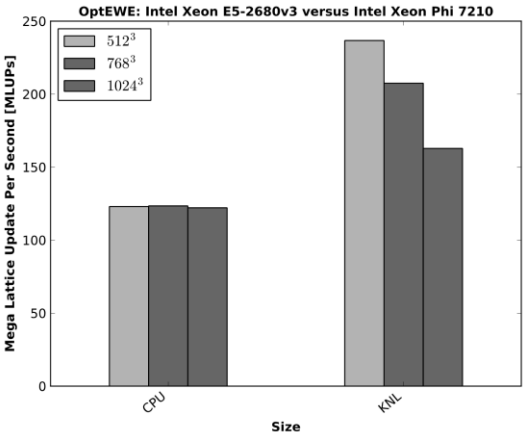
NTNU

This presentation focuses on exploring the use of KNL based Xeon Phis for accelerating 3D Seismic Wave Simulations



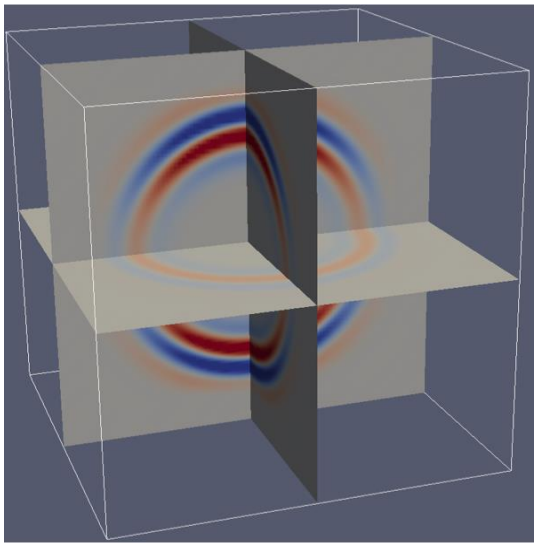
3D Seismic Wave Simulator

Xeon Phi Architecture



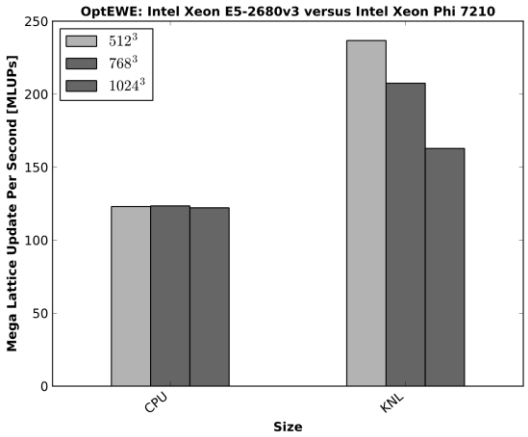
Results

This presentation focuses on exploring the use of KNL based Xeon Phis for accelerating 3D Seismic Wave Simulations



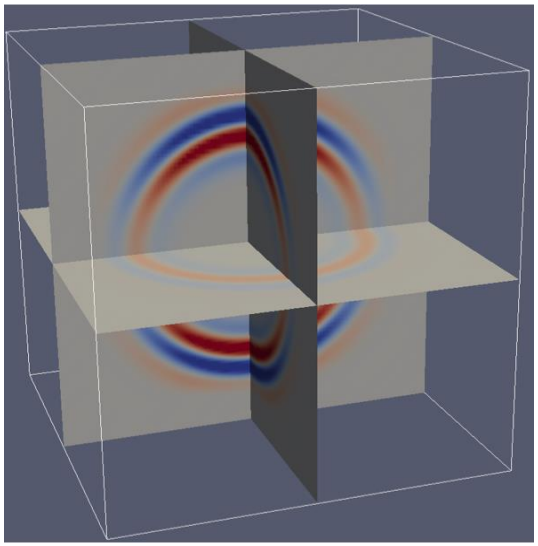
3D Seismic Wave Simulator

Xeon Phi Architecture



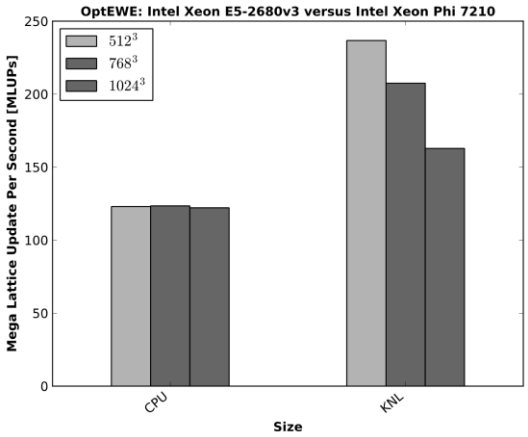
Results

This presentation focuses on exploring the use of KNL based Xeon Phis for accelerating 3D Seismic Wave Simulations



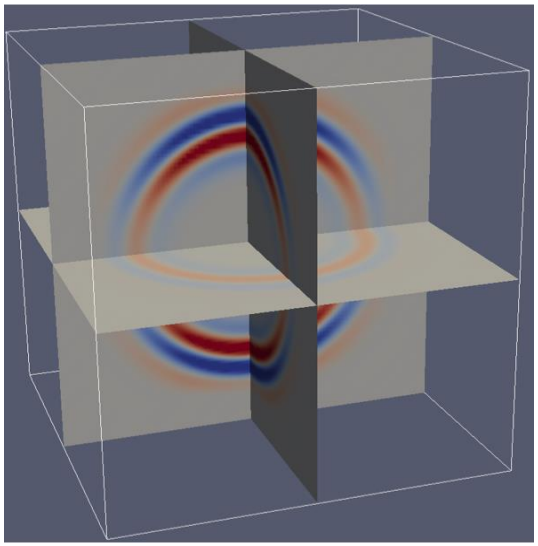
3D Seismic Wave Simulator

Xeon Phi Architecture



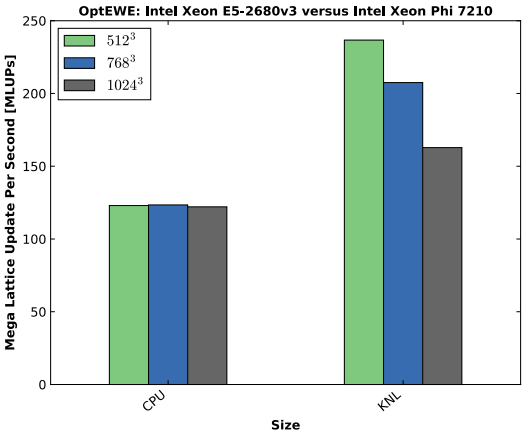
Results

This presentation focuses on exploring the use of KNL based Xeon Phis for accelerating 3D Seismic Wave Simulations



3D Seismic Wave Simulator

Xeon Phi Architecture

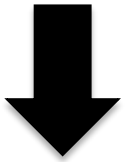


Results

Target application: A 3D Seismic Wave Simulator developed from ground-up at NTNU

Numerical framework[1]

Staggered-grid explicit FD
8th order in space
2nd order in time



Memory-bound stencil application parallelized using OpenMP

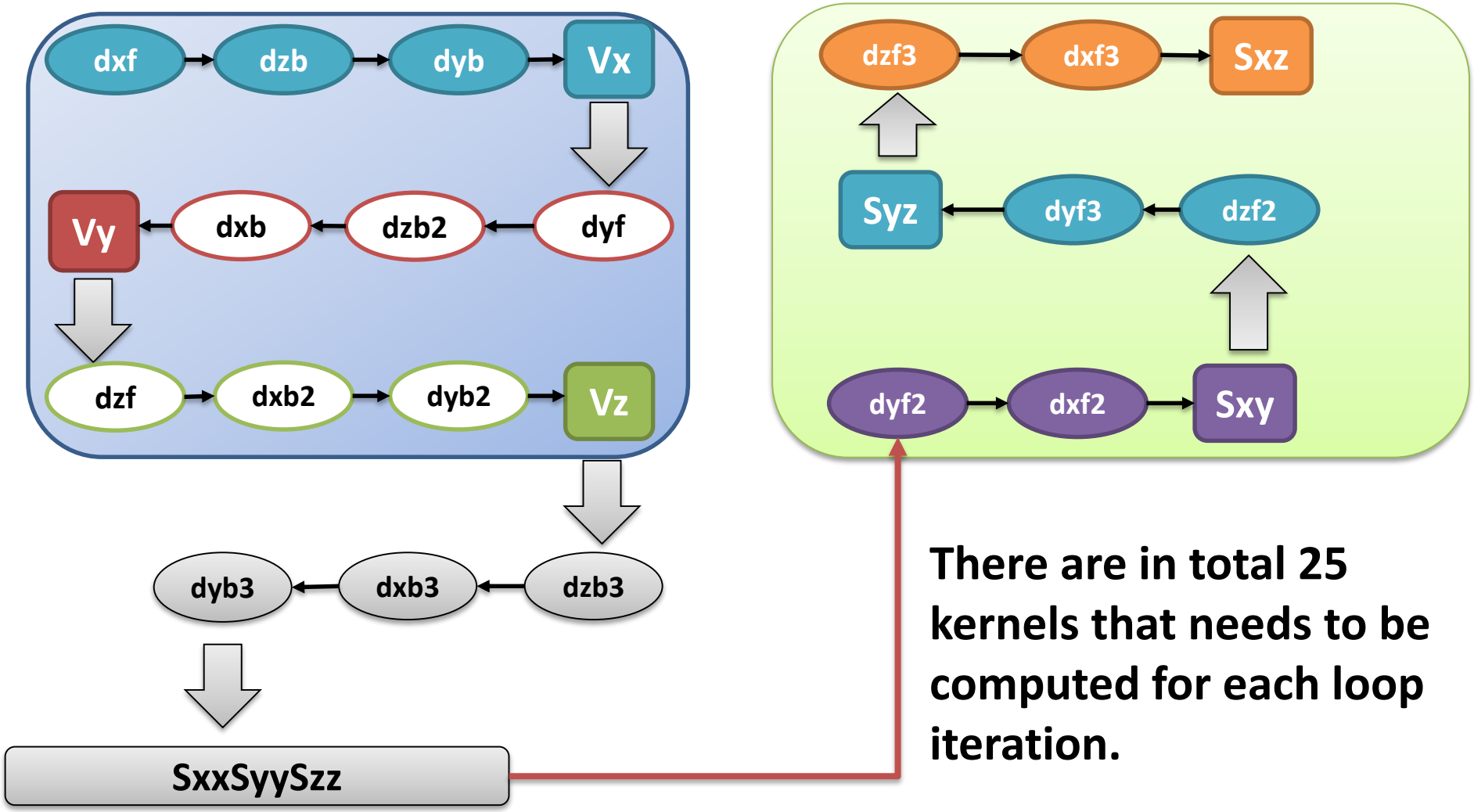
3D elastodynamic wave equation

$$\begin{cases} \rho \dot{v}_x = \partial_x \sigma_{xx} + \partial_y \sigma_{xy} + \partial_z \sigma_{xz} + f_x \\ \rho \dot{v}_y = \partial_x \sigma_{xy} + \partial_y \sigma_{yy} + \partial_z \sigma_{yz} + f_y \\ \rho \dot{v}_z = \partial_x \sigma_{xz} + \partial_y \sigma_{yz} + \partial_z \sigma_{zz} + f_z \end{cases}$$

$$\begin{cases} \dot{\sigma}_{xx} = (\lambda + 2\mu)\partial_x v_x + \lambda(\partial_y v_y + \partial_z v_z) \\ \dot{\sigma}_{yy} = (\lambda + 2\mu)\partial_y v_y + \lambda(\partial_x v_x + \partial_z v_z) \\ \dot{\sigma}_{zz} = (\lambda + 2\mu)\partial_z v_z + \lambda(\partial_x v_x + \partial_y v_y) \\ \dot{\sigma}_{yz} = \mu(\partial_y v_z + \partial_z v_y) \\ \dot{\sigma}_{xz} = \mu(\partial_x v_z + \partial_z v_x) \\ \dot{\sigma}_{xy} = \mu(\partial_x v_y + \partial_y v_x) \end{cases}$$

[1] Espen Birger Raknes, Børge Arntsen, and Wiktor Weibull. 2015. Three- dimensional elastic full waveform inversion using seismic data from the Sleipner area. Geophysical Journal International 202, 3 (2015), 1877–1894. DOI: https://doi.org/10.1093/gji/ggv258

Target application: A 3D Seismic Wave Simulator developed from ground-up at NTNU



There are in total 25 kernels that needs to be computed for each loop iteration.

KNL based Xeon Phi: A cluster on chip

Tile

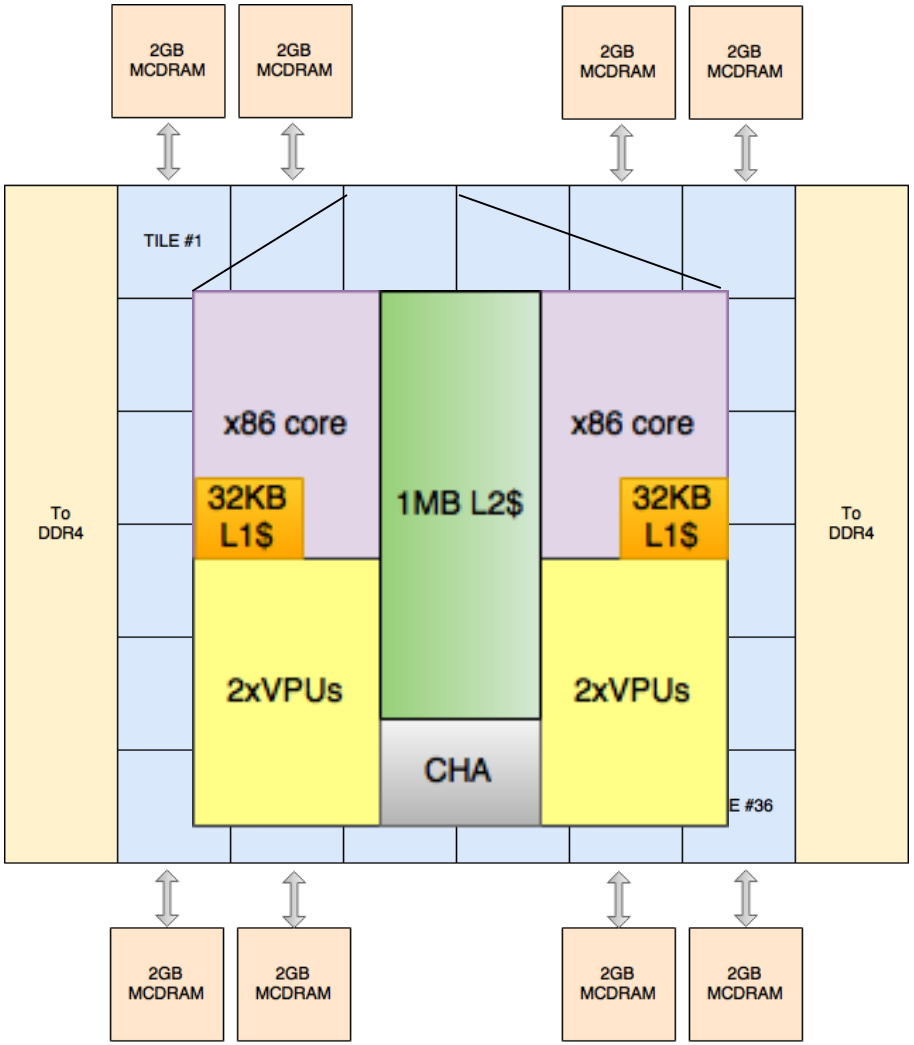
- Each tile consists of two x86 cores
- Communication via a 2D mesh interconnect

Core

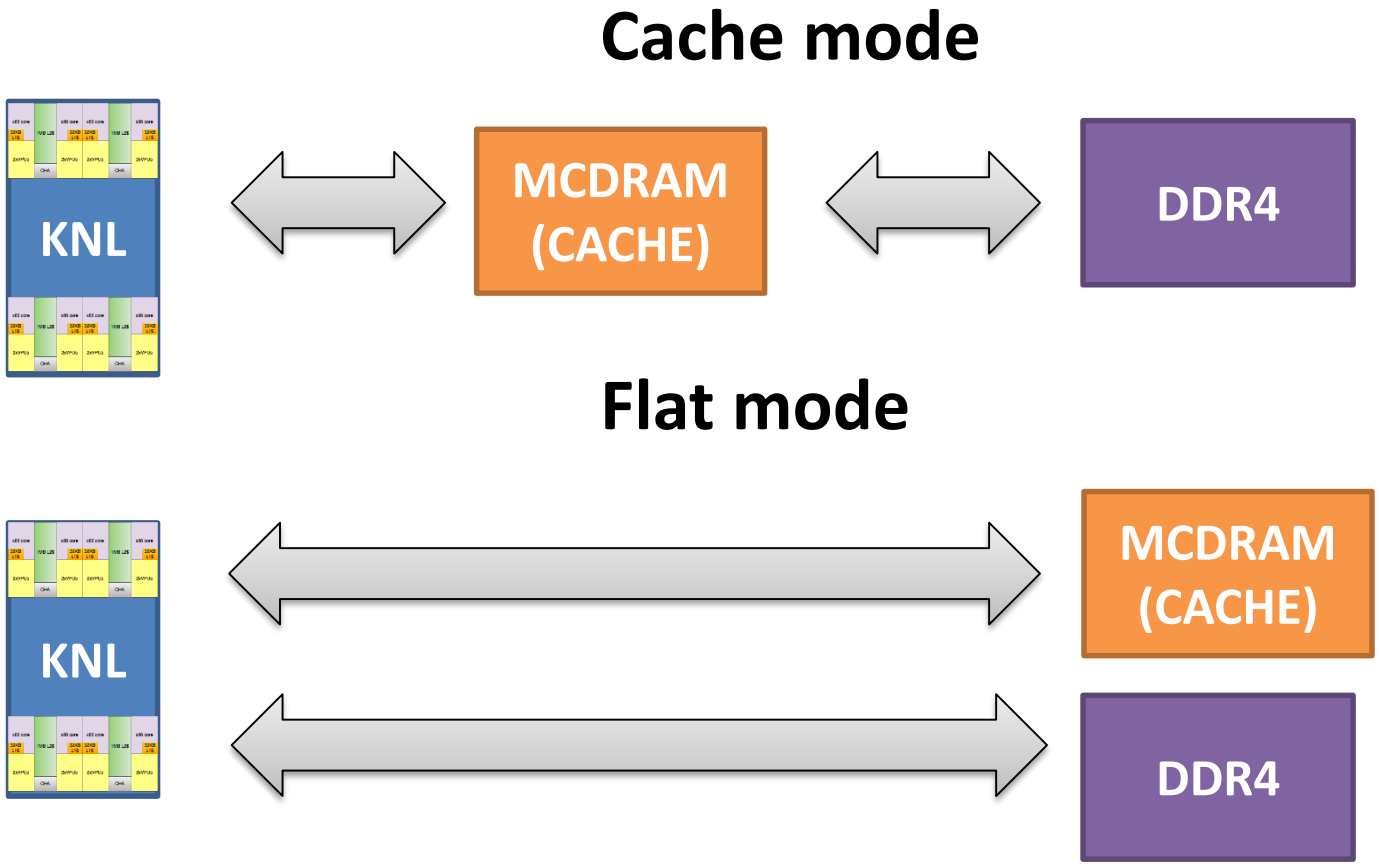
- Each core has two 512 bit vector processing units (VPUs)
- Private 32KB L1 cache
- 1MB shared L2 cache

Memory

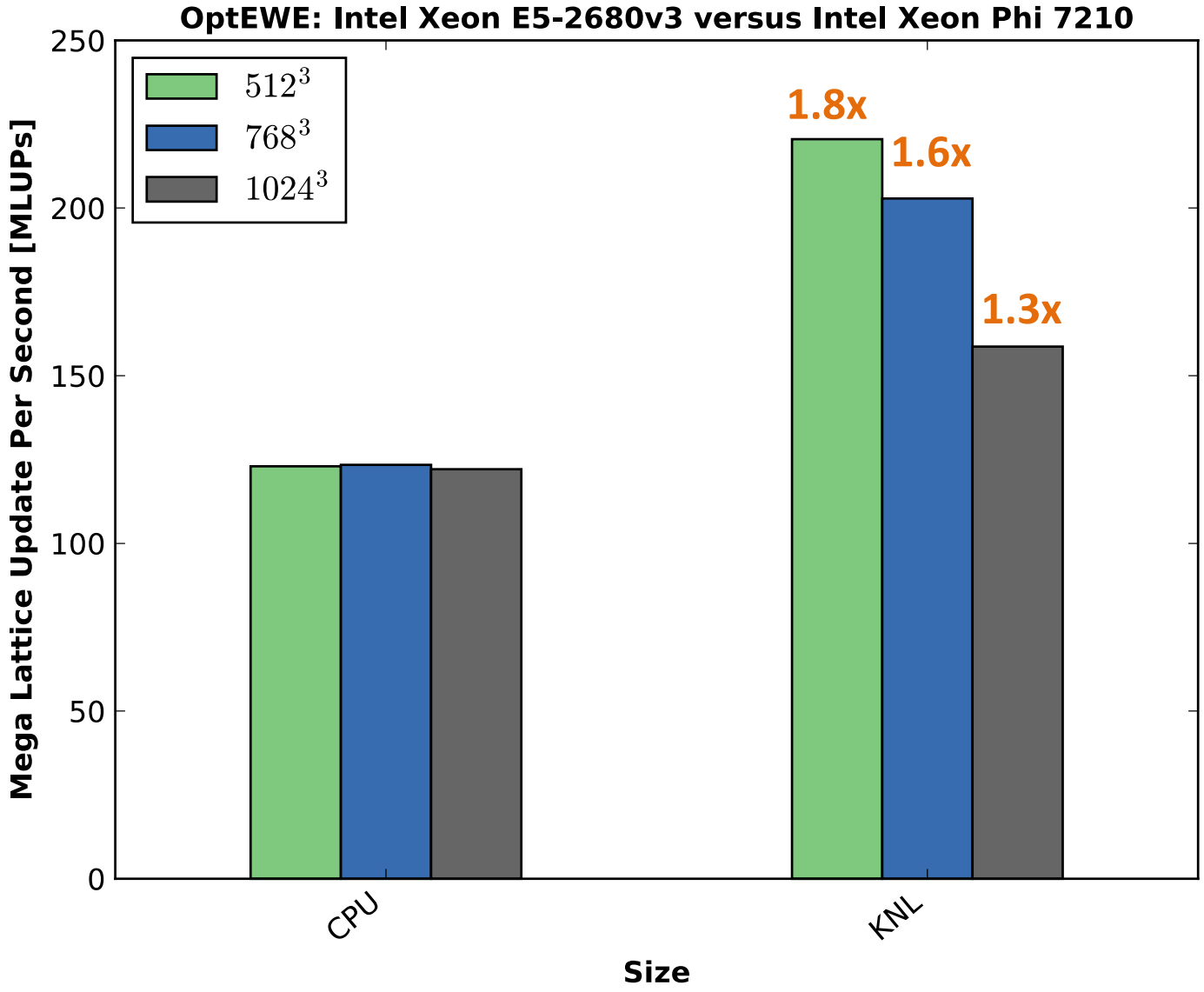
- 8x2GB MCDRAM
- Up to 384 GB DDR4 memory (three channels)



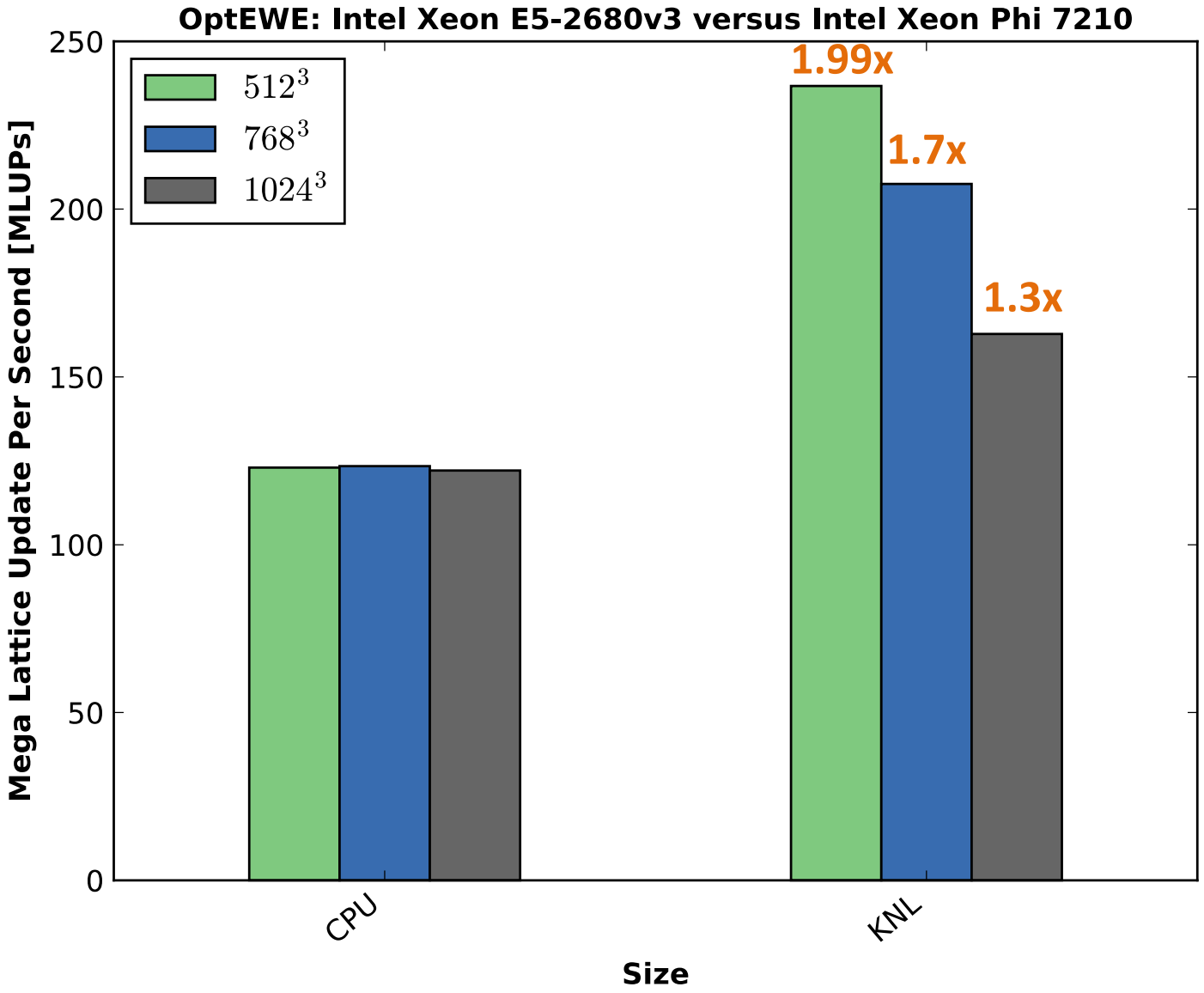
The memory architecture of KNL based Xeon Phis is user-configurable.



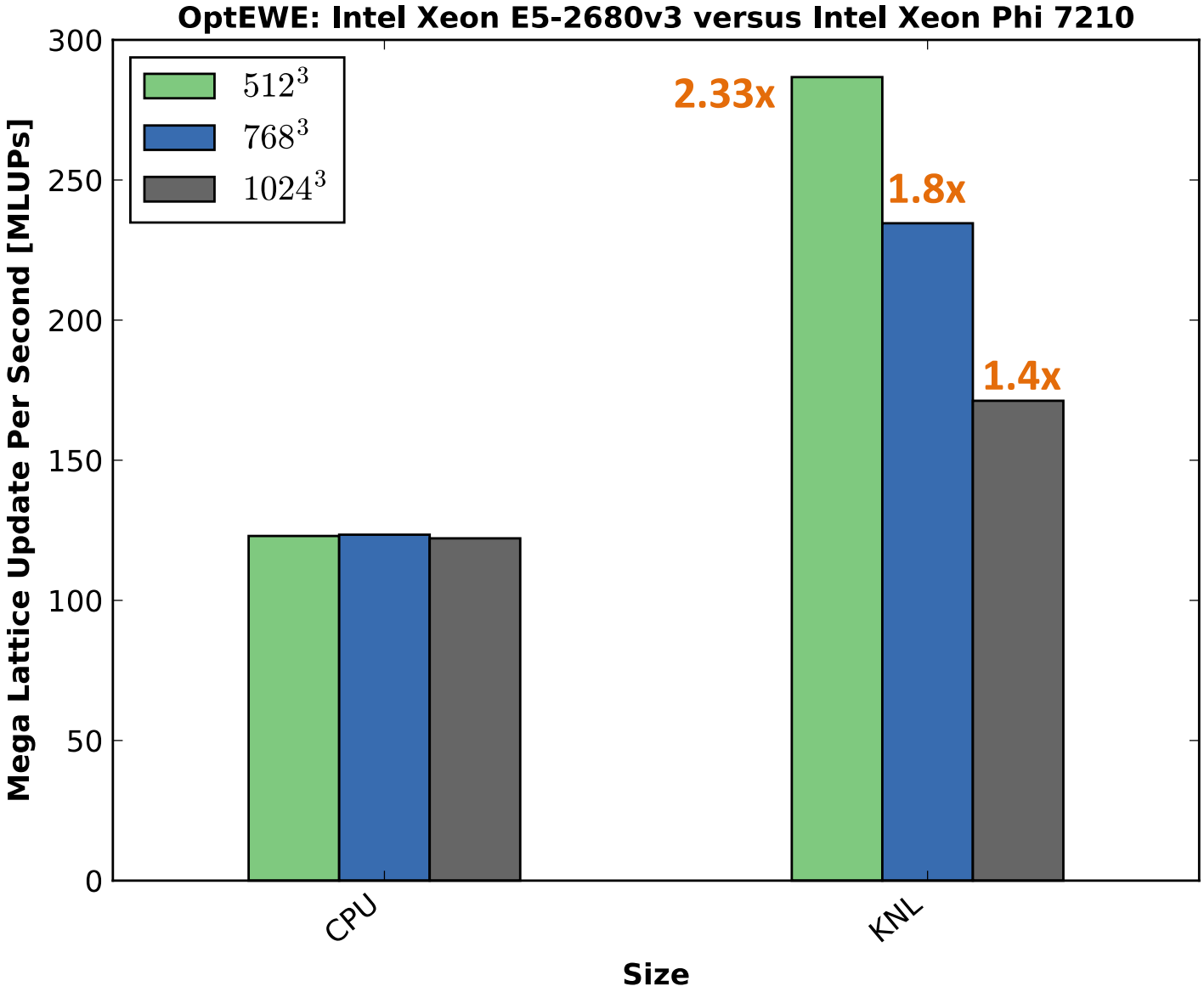
Baseline Xeon Phi results compared to a highly-tuned multi-core implementation.



By tuning the thread affinity, we can can further increase the performance.



Changing the memory policy (cache->flat) can have a dramatic impact on the performance.

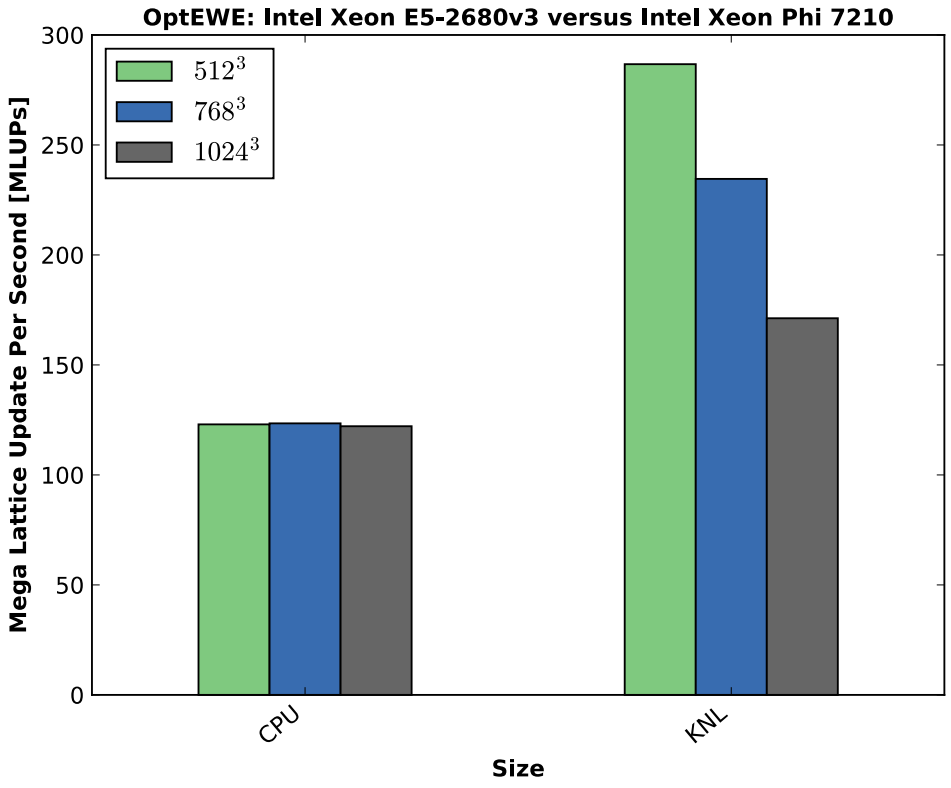


In summary, we have shown that KNL can increase performance compared to multi-core x86 CPUs significantly

KNL provides a good blend of ease-of-use and performance

If the entire workload fits into MCDRAM, use flat mode, else cache mode is always a safe choice.

We have demonstrated that high performance can be achieved with only few steps.



NTNU