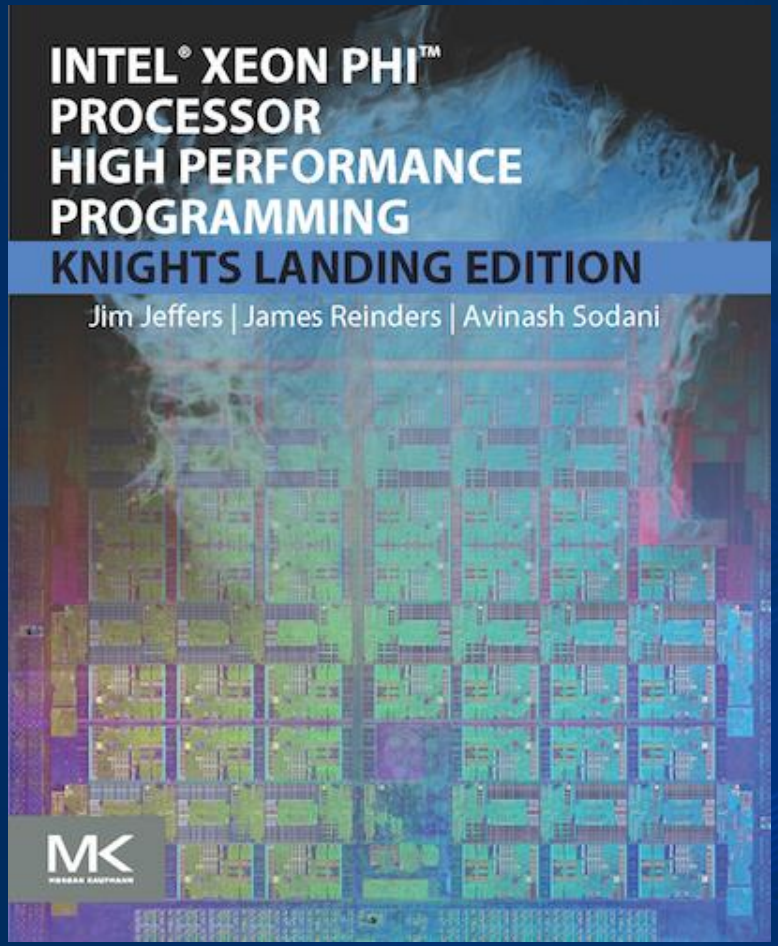
Knights Landing – An Overview for Developers

James Reinders, Intel
June 7, 2016



Knights Landing

2nd Generation Intel® Xeon Phi™ products

ALL ABOUT PARALEL PROGRAMMING

Threading, Vectorization, Data Locality

Fortran, C, C++ (plus a little Python)

OpenMPI, MPI, Intel® TBB

New with Knights Landing:

AVX-512,

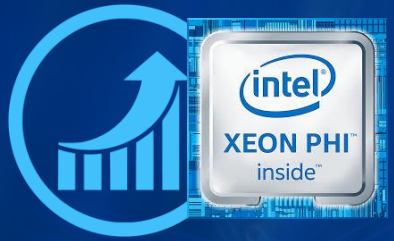
High Bandwidth Memory (MCDRAM),

Cluster Mode,

Omni-Path

XeonPhiDeveloper.com

You can buy you very own Knight Landing development system today!



Highly-Parallel Performance
to develop on

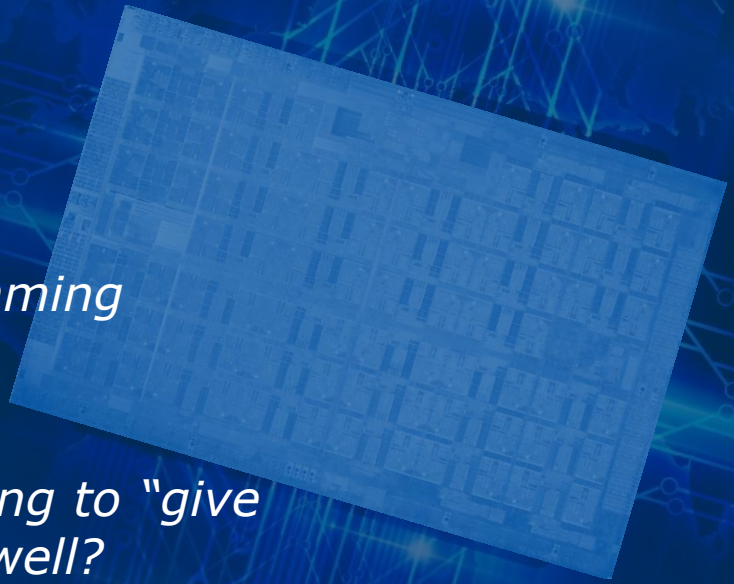
All the Software Tools & Libraries
you need

Support & Training
to help you succeed

Leading edge platform capabilities, performance to deliver multi-threaded, vectorized software for today's HPC workloads !

Why Intel® Xeon Phi™ Processors?

Imagine you could design anything



What is you wanted:

- *High performance*
- *Preserve your investment in code*
- *Reuse your knowledge of programming languages, tools, techniques*

What if the ONLY thing we were willing to "give up" was running non-parallel codes well?

In other words:
what if we assumed ONLY parallel applications would be run?

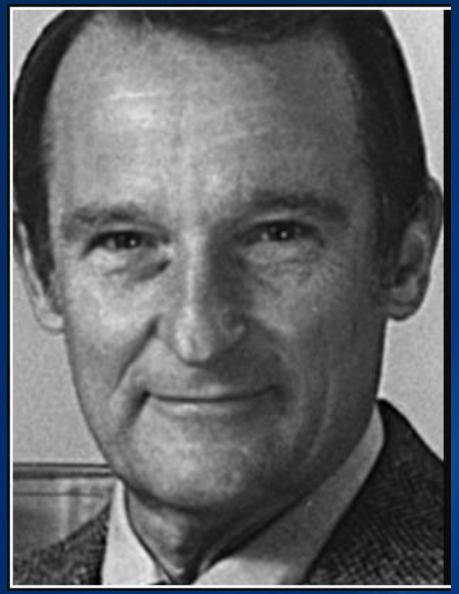
Imagine you could design anything

What is you wanted:

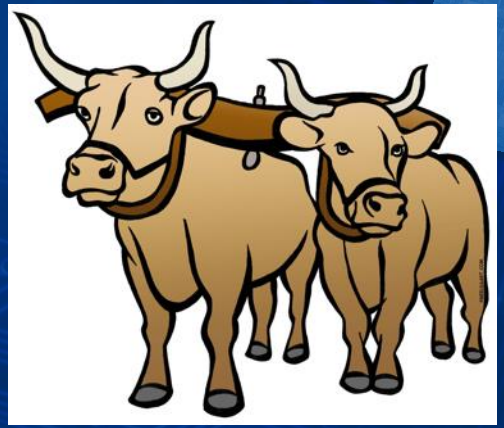
- *High performance*
- *Preserve your investment in code*
- *Reuse your knowledge of programming languages, tools, techniques*

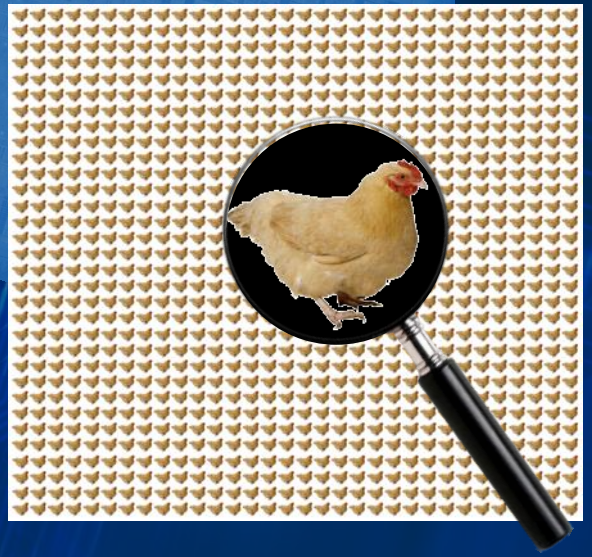
What if the ONLY thing we were willing to "give up" was running non-parallel codes well?

In other words:
what if we assumed ONLY parallel applications would be run?

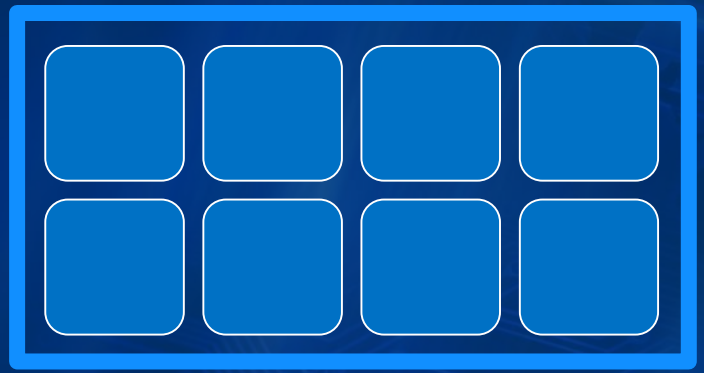


If you were plowing a field,
which would you rather use…
two strong oxen, or
1024 chickens?



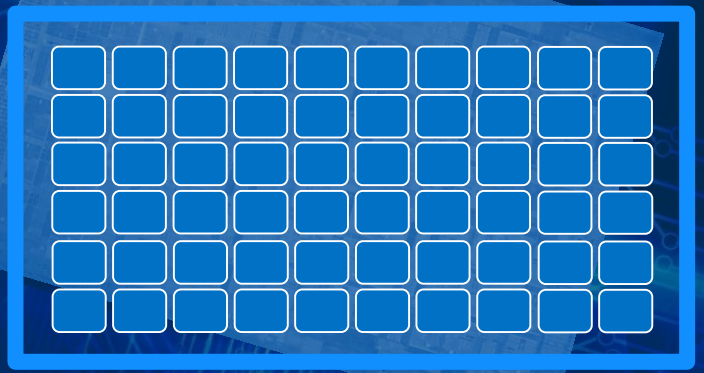


Design Question – Best for Computing?



A few powerful

vs.



Many less powerful.

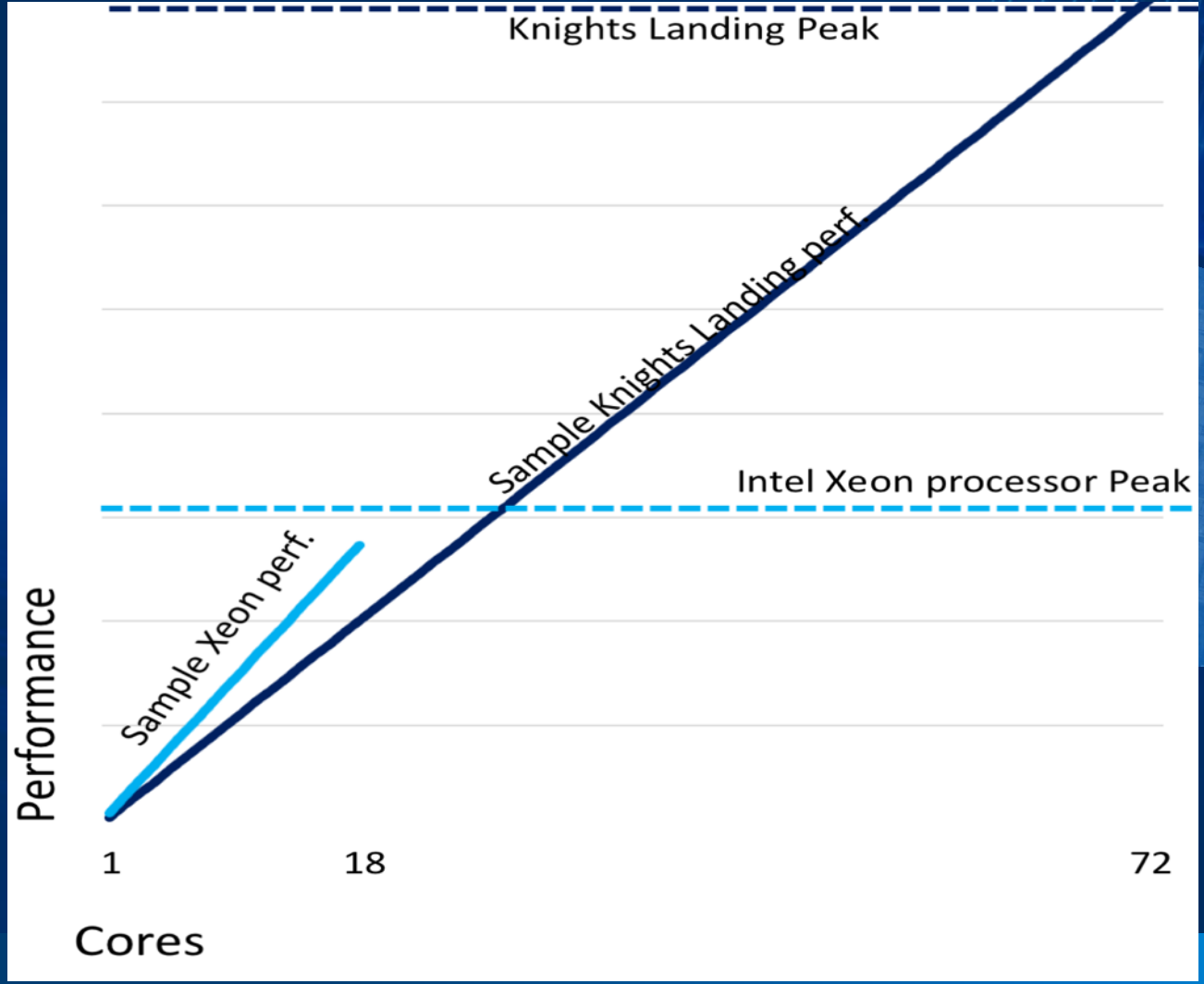
Design Question – *Our difference: Same programming models, languages, optimizations and tools*

A few powerful

vs.

Many less powerful.

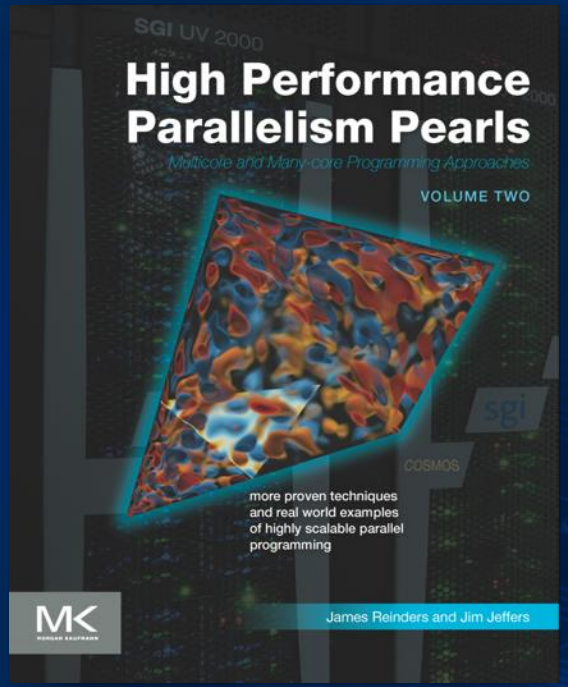
Same programming models, languages, optimizations and tools.

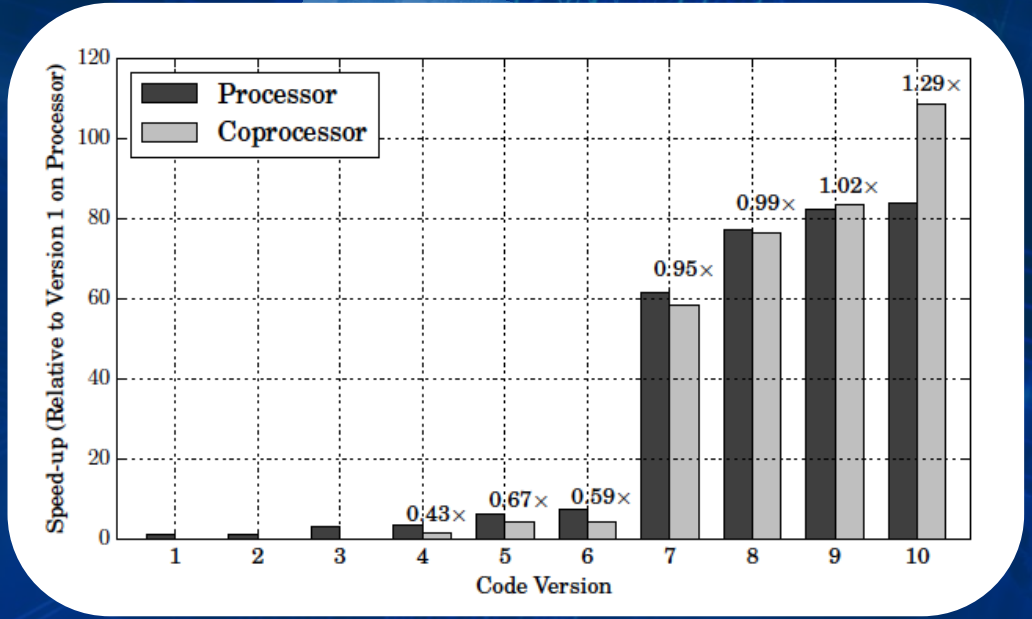

Knights Landing Peak
Sample Knights Landing perf.
Intel Xeon processor Peak
Sample Xeon perf.
Performance
1
18
72
Cores

vision

span from *few cores* to *many cores*
with consistent models,
languages, tools, and techniques

#Moderncode: COSMOS

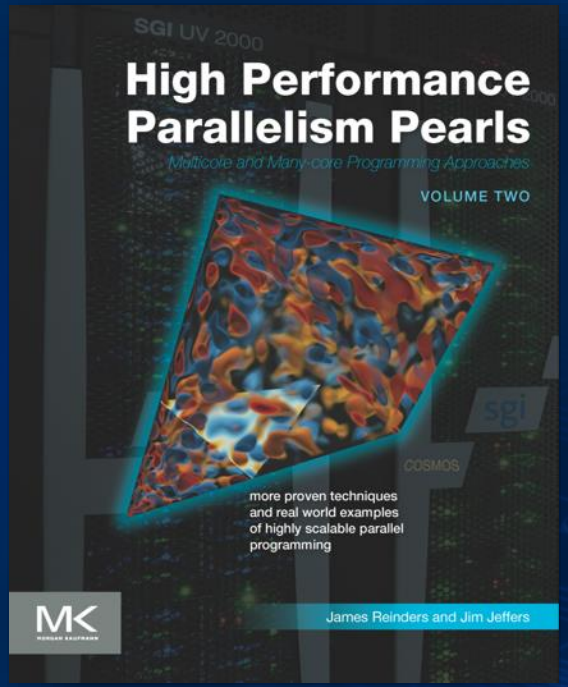


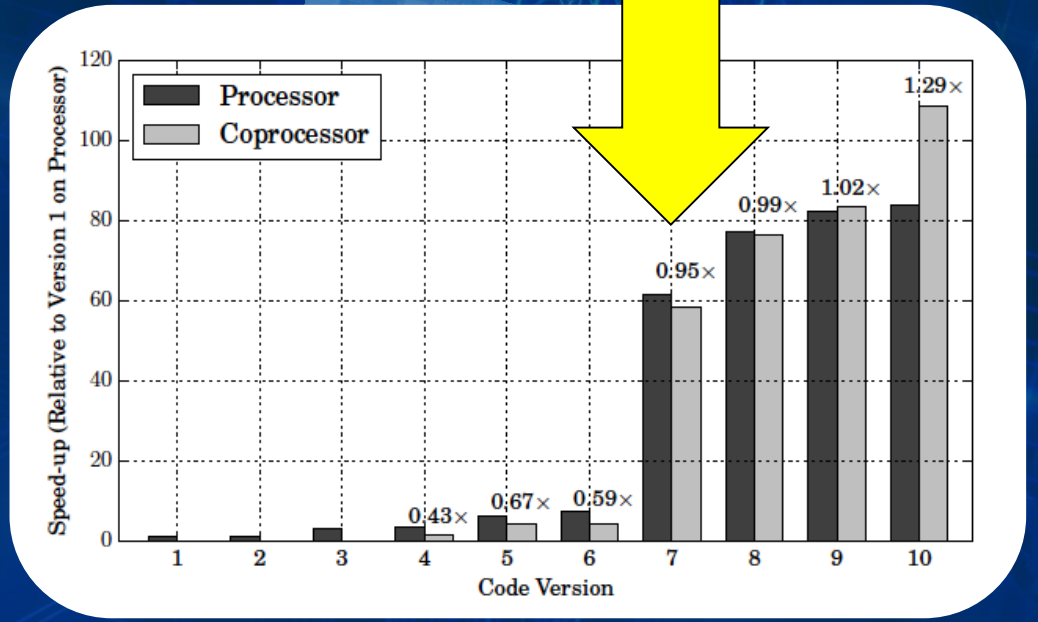


Book Cover Background: Photo of the COSMOS@DiRAC SGI UV2000 based Supercomputer manufactured by SGI, Inc and operated by the Stephen Hawking Centre for Theoretical Cosmology, University of Cambridge. Photo courtesy of Philip Mynott. Book Cover Foreground: 3D visualization of statistical fluctuations in the Cosmic Microwave Background, the remnant of the first measurable light after the Big Bang. CMB data is from the Planck satellite and is the topic of Chapter 10 providing insights into new physics and how the universe evolved. Visualization rendered with Intel's OSPRay ray tracing open source software by Gregory P. Johnson and Timothy Rowley, Intel Corporation.

#Moderncode: COSMOS

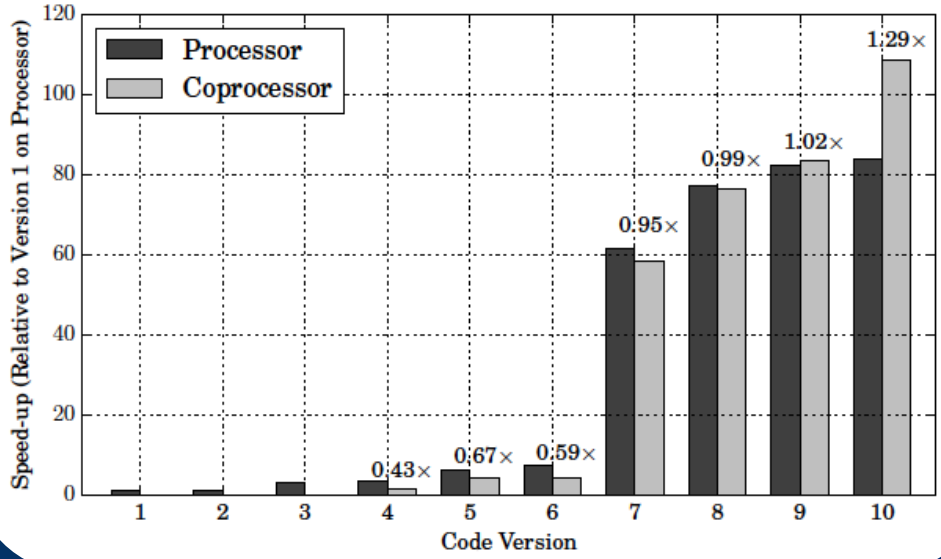
What?





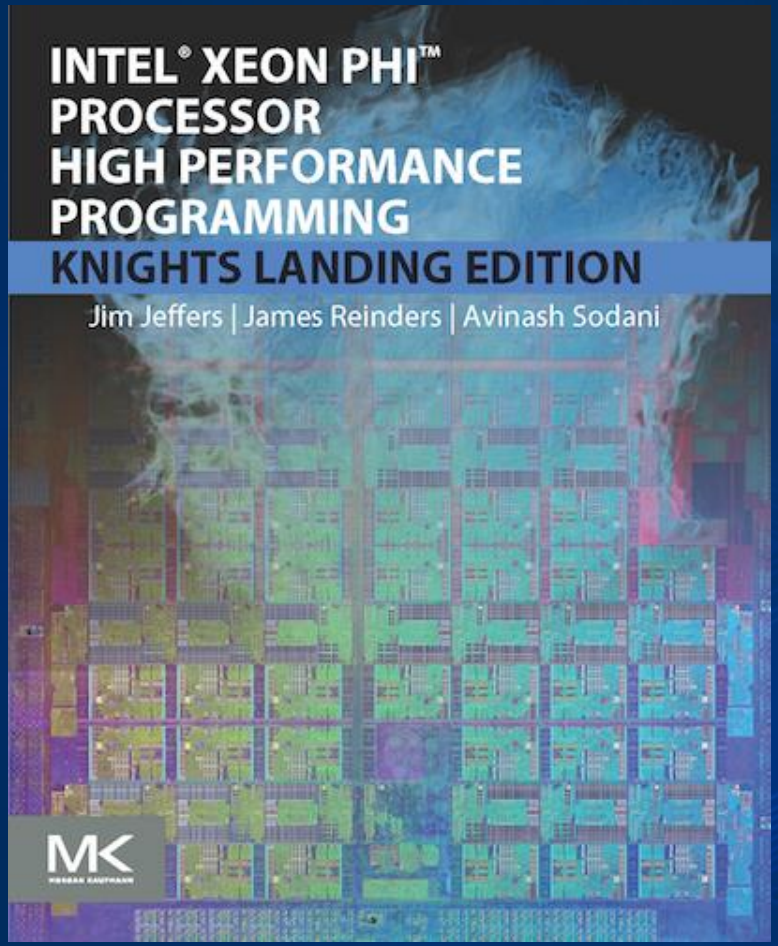
Book Cover Background: Photo of the COSMOS@DiRAC SGI UV2000 based Supercomputer manufactured by SGI, Inc and operated by the Stephen Hawking Centre for Theoretical Cosmology, University of Cambridge. Photo courtesy of Philip Mynott. Book Cover Foreground: 3D visualization of statistical fluctuations in the Cosmic Microwave Background, the remnant of the first measurable light after the Big Bang. CMB data is from the Planck satellite and is the topic of Chapter 10 providing insights into new physics and how the universe evolved. Visualization rendered with Intel's OSPRay ray tracing open source software by Gregory P. Johnson and Timothy Rowley, Intel Corporation.

High Performance Parallelism Pearls
Cosmic Microwave Background Analysis: Nested Parallelism in Practice
Volume 2, Chapter 10



We find that using a simple trapezium rule integrator combined with hand-selected sampling points (to improve accuracy in areas of interest) provides sufficient numerical accuracy to obtain a physically meaningful result, and the reduced space and time requirements of this simplified method give a speed-up of O(10x).

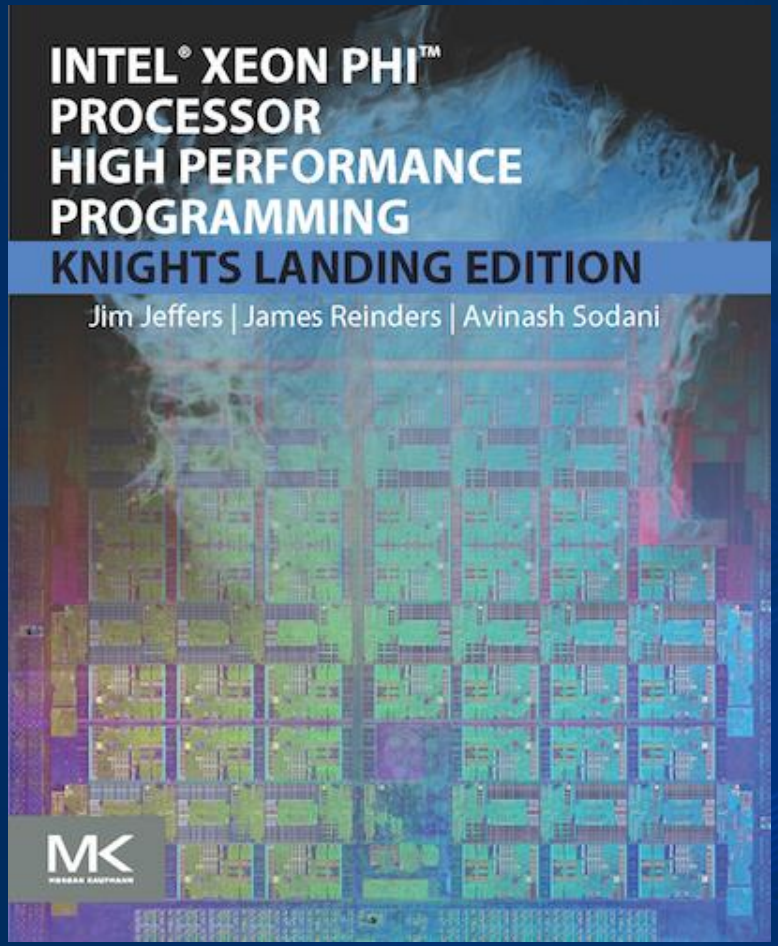
Version	Processor (s)	Coprocessor (s)	Comment
1	2887.0	-	Original code.
2	2610.0	-	Loop simplification.
3	882.0	-	Intel® MKL integration routines and function inlining.
4	865.9	1991.6	Flattened loops and introduced OpenMP threads.
5	450.6	667.9	Loop reordering and manual nested threading.
6	385.6	655.0	Blocked version of the loop (for cache).
7	46.9	49.5	Numerical integration routine (Trapezium Rule).
8	37.4	37.7	Reduction with DGEMM.
9	35.1	34.5	Data alignment (for vectorization).
10	34.3	26.6	Tuning of software prefetching distances.

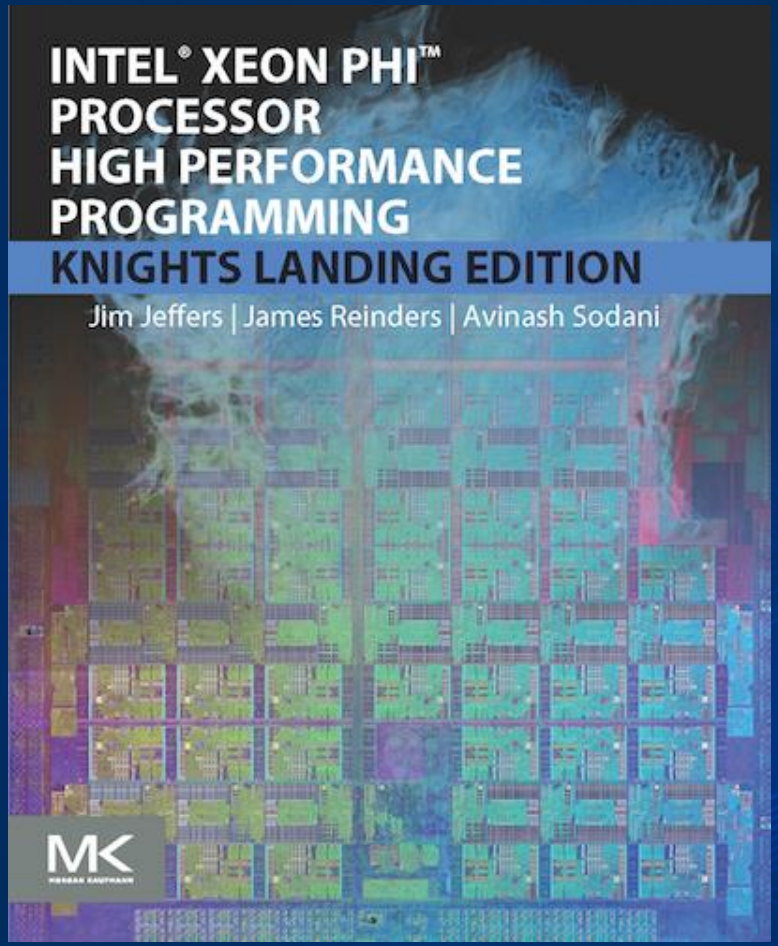


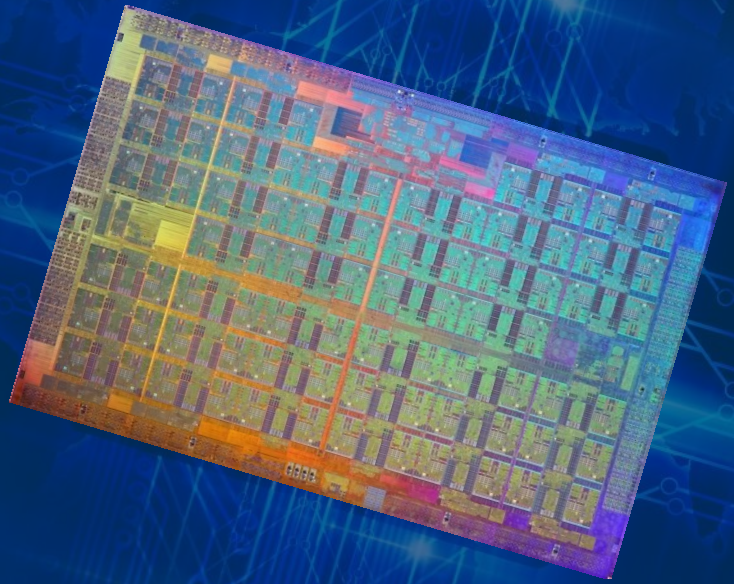
Knights Landing
2nd Generation Intel® Xeon Phi™

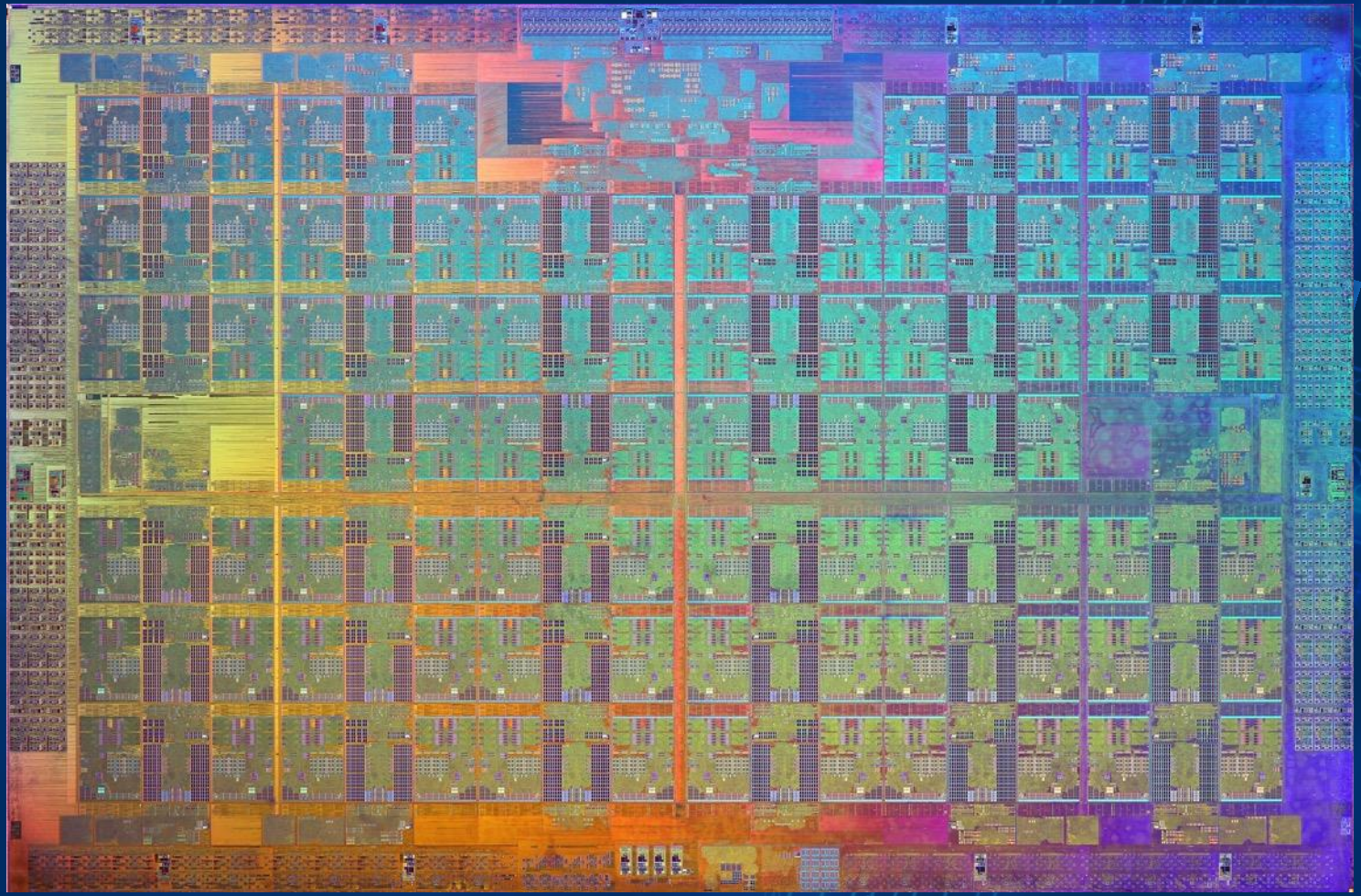
ALL ABOUT PARALEL PROGRAMMING
Threading, Vectorization, Data Locality
Fortran, C, C++ (plus a little Python)
OpenMPI, MPI, TBB

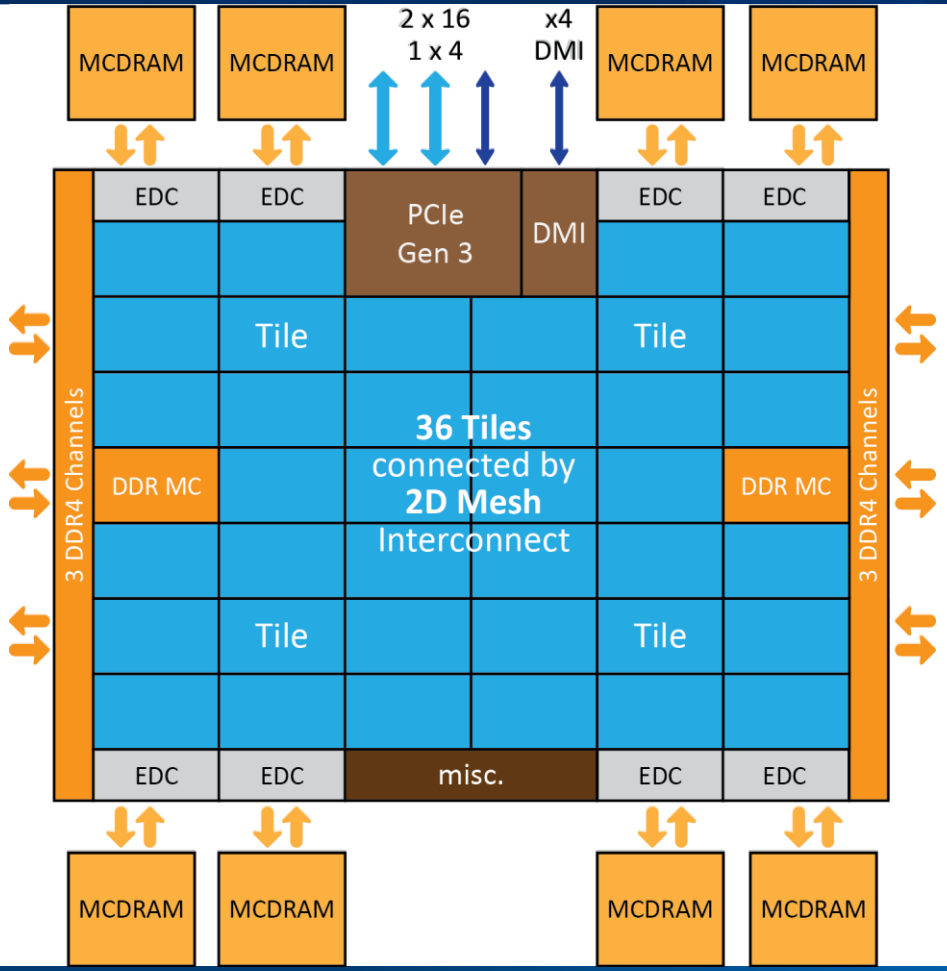
New with Knights Landing:
AVX-512,
High Bandwidth Memory (MCDRAM),
Cluster Mode,
Omni-Path


INTEL® XEON PHI™
PROCESSOR
HIGH PERFORMANCE
PROGRAMMING
KNIGHTS LANDING EDITION
Jim Jeffers | James Reinders | Avinash Sodani
MK


INTEL® XEON PHI™
PROCESSOR
HIGH PERFORMANCE
PROGRAMMING
KNIGHTS LANDING EDITION
Jim Jeffers | James Reinders | Avinash Sodani
MK



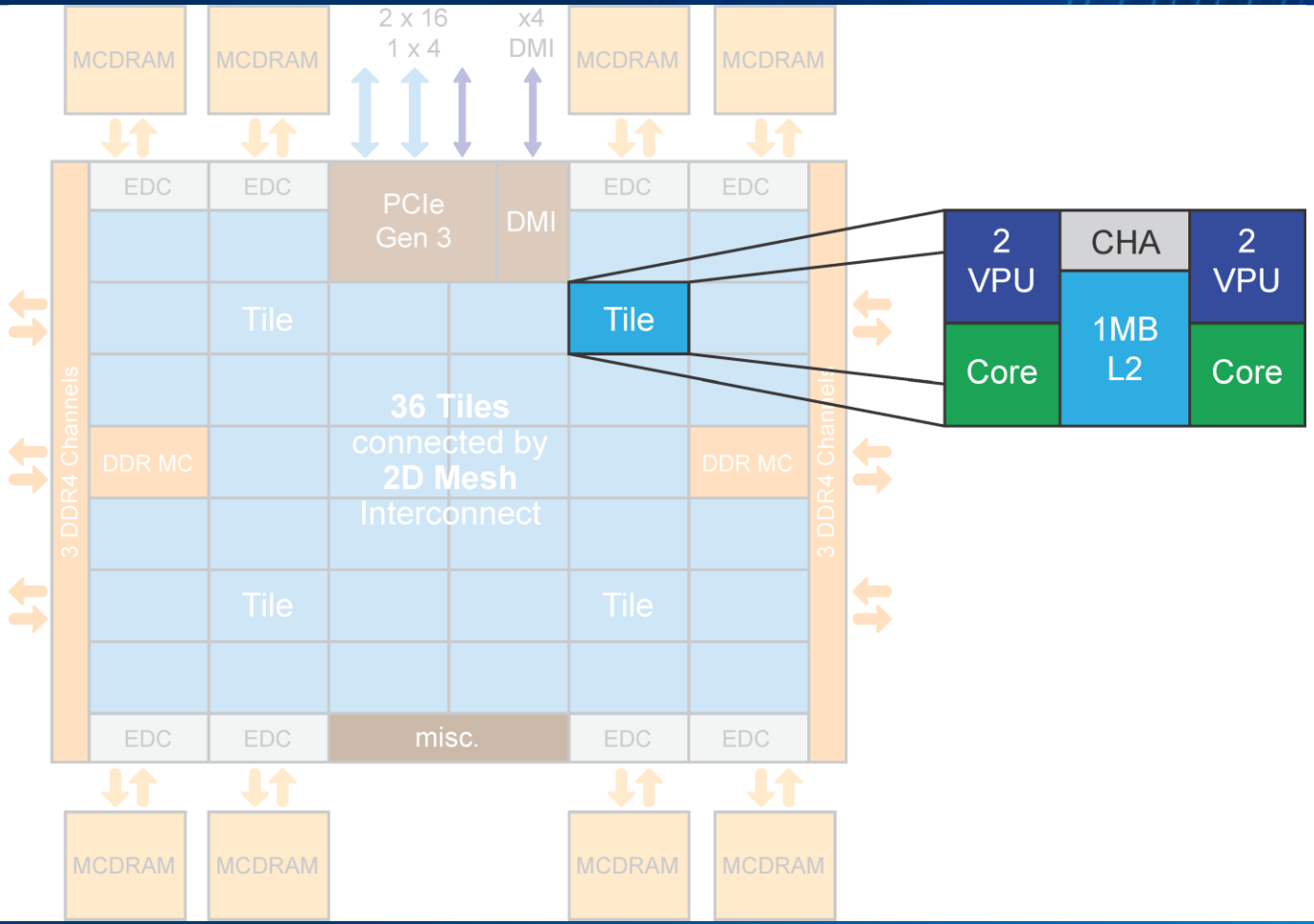


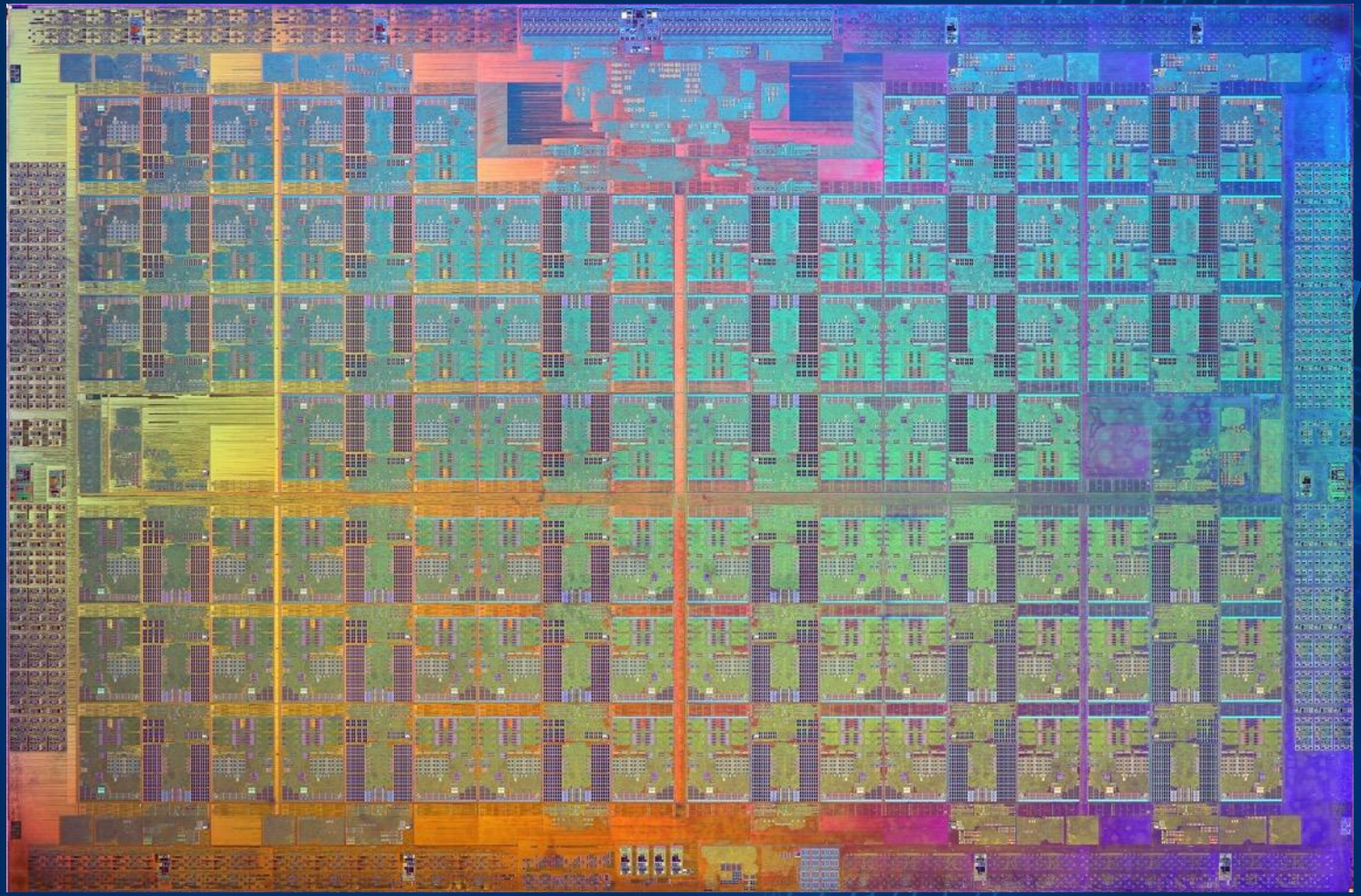


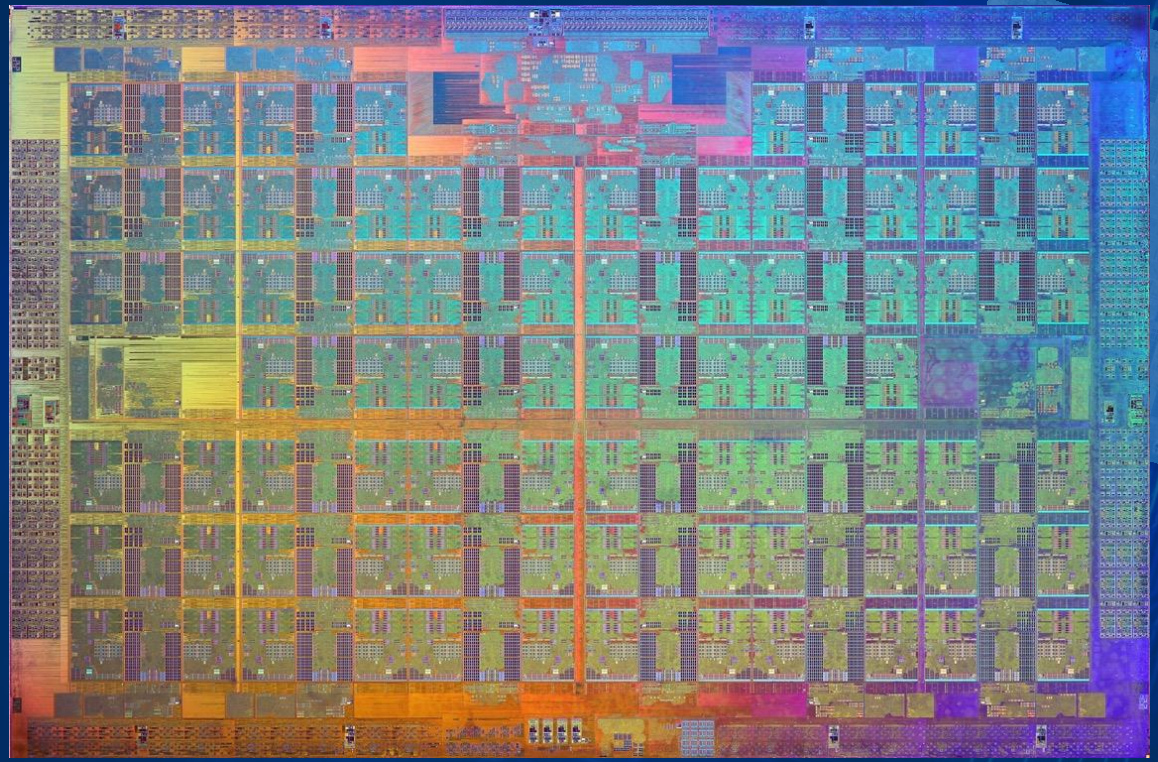
2nd Generation Intel® Xeon Phi™ Products

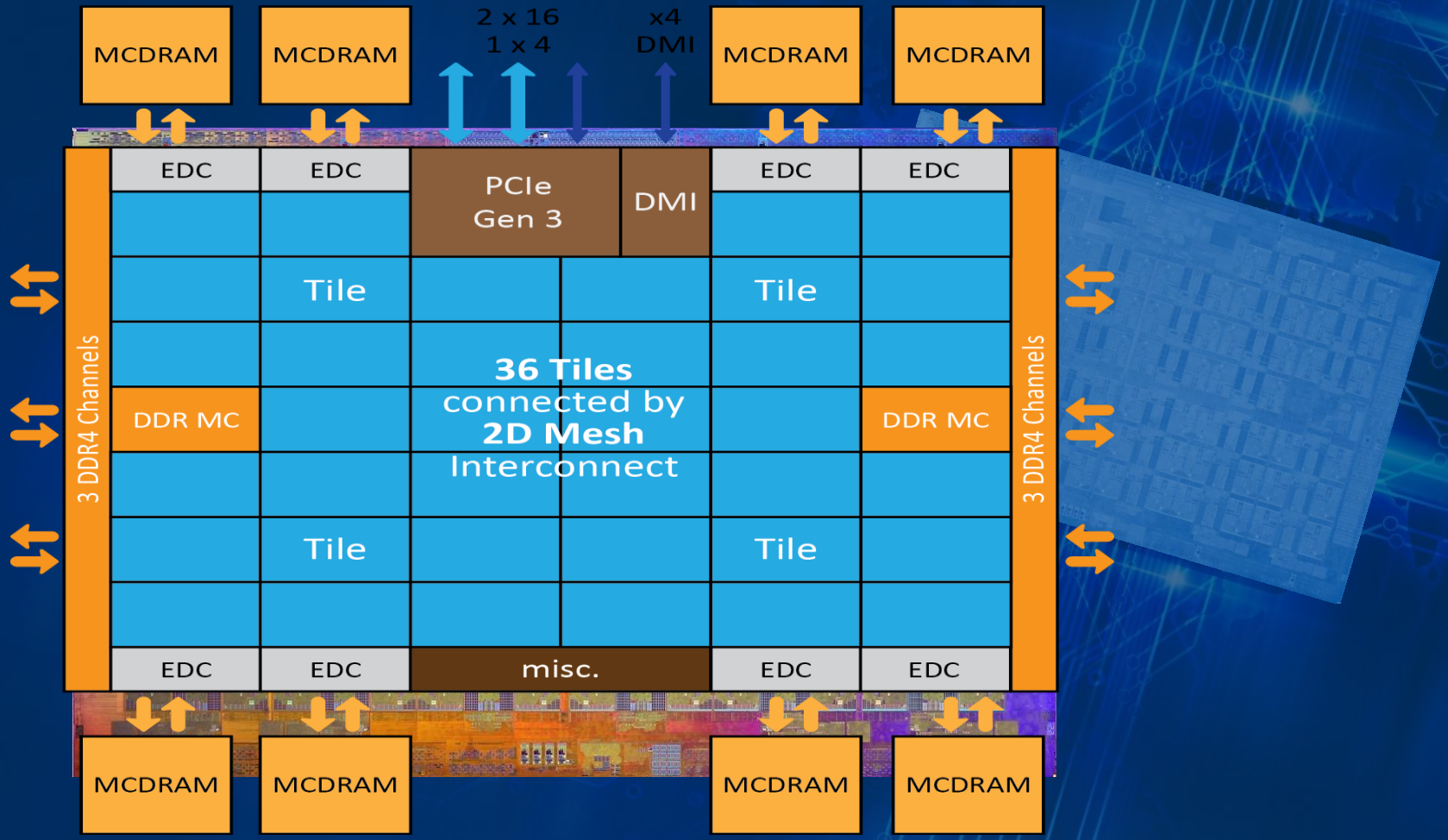
Codename:

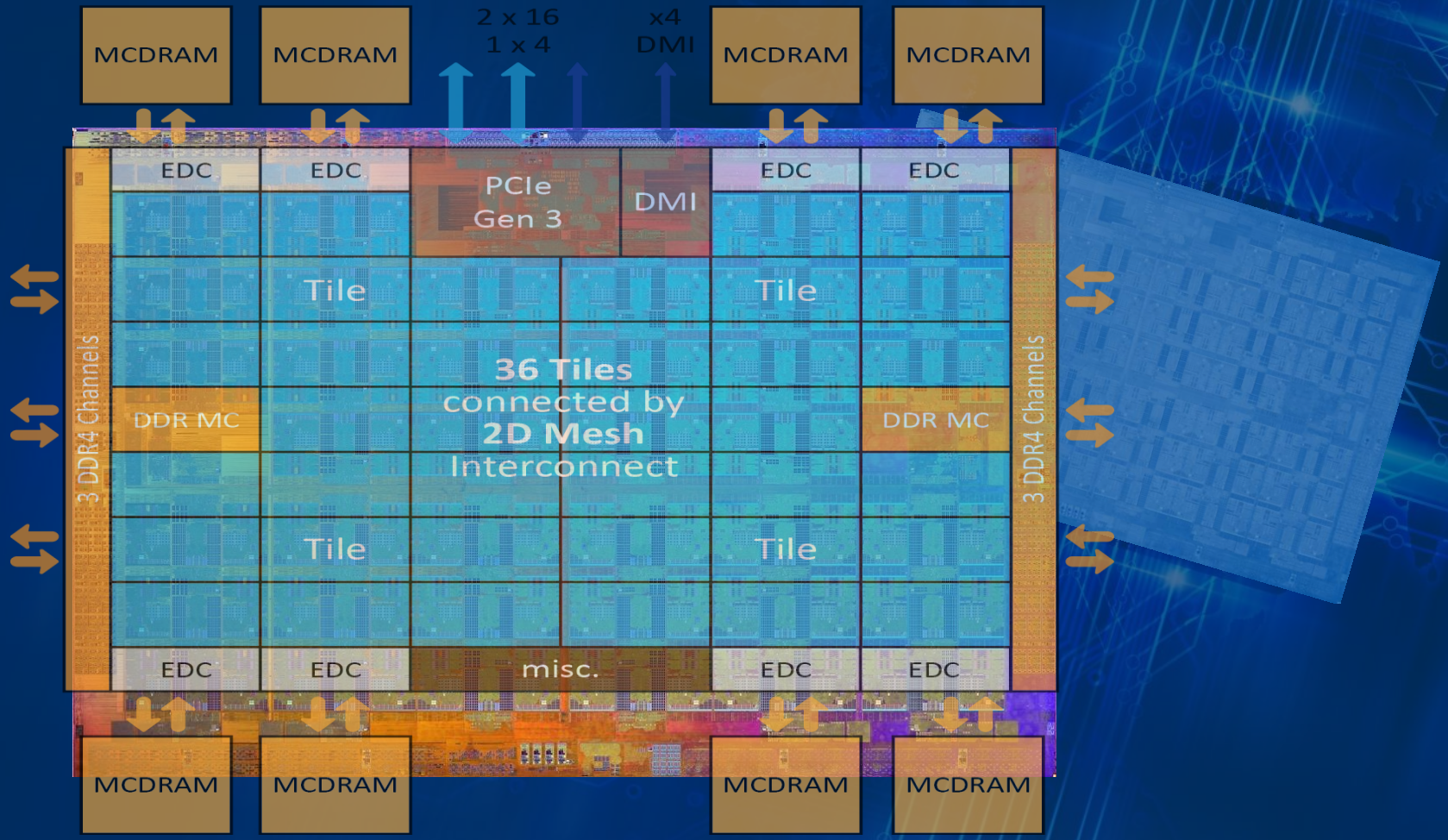
Knights Landing


MCDRAM
MCDRAM
2 x 16
1 x 4
x4
DMI
MCDRAM
MCDRAM
EDC
EDC
PCIe
Gen 3
DMI
EDC
EDC
Tile
Tile
36 Tiles
connected by
2D Mesh
Interconnect
DDR MC
DDR MC
3 DDR4 Channels
3 DDR4 Channels
Tile
Tile
EDC
EDC
misc.
EDC
EDC
MCDRAM
MCDRAM
MCDRAM
MCDRAM
2
VPU
CHA
2
VPU
1MB
L2
Core
Core






MCDRAM
MCDRAM
2 x 16
1 x 4
x4
DMI
MCDRAM
MCDRAM
EDC
EDC
PCIe
Gen 3
DMI
EDC
EDC
Tile
Tile
36 Tiles
connected by
2D Mesh
Interconnect
DDR MC
DDR MC
Tile
Tile
3 DDR4 Channels
3 DDR4 Channels
EDC
EDC
misc.
EDC
EDC
MCDRAM
MCDRAM
MCDRAM
MCDRAM


MCDRAM
MCDRAM
2 x 16
1 x 4
x4
DMI
MCDRAM
MCDRAM
EDC
EDC
PCIe
Gen 3
DMI
EDC
EDC
Tile
Tile
36 Tiles
connected by
2D Mesh
Interconnect
DDR MC
DDR MC
3 DDR4 Channels
3 DDR4 Channels
Tile
Tile
EDC
EDC
misc.
EDC
EDC
MCDRAM
MCDRAM
MCDRAM
MCDRAM

AVX-512, High Bandwidth Memory, Cluster Mode, Omni-Path

Knights Landing: Next Intel® Xeon Phi™ Processor

Intel® Many-Core Processor targeted for HPC and Supercomputing

First **self-boot** Intel® Xeon Phi™ processor that is **binary compatible** with main line IA. Boots standard OS.

Significant improvement in scalar and **vector** performance

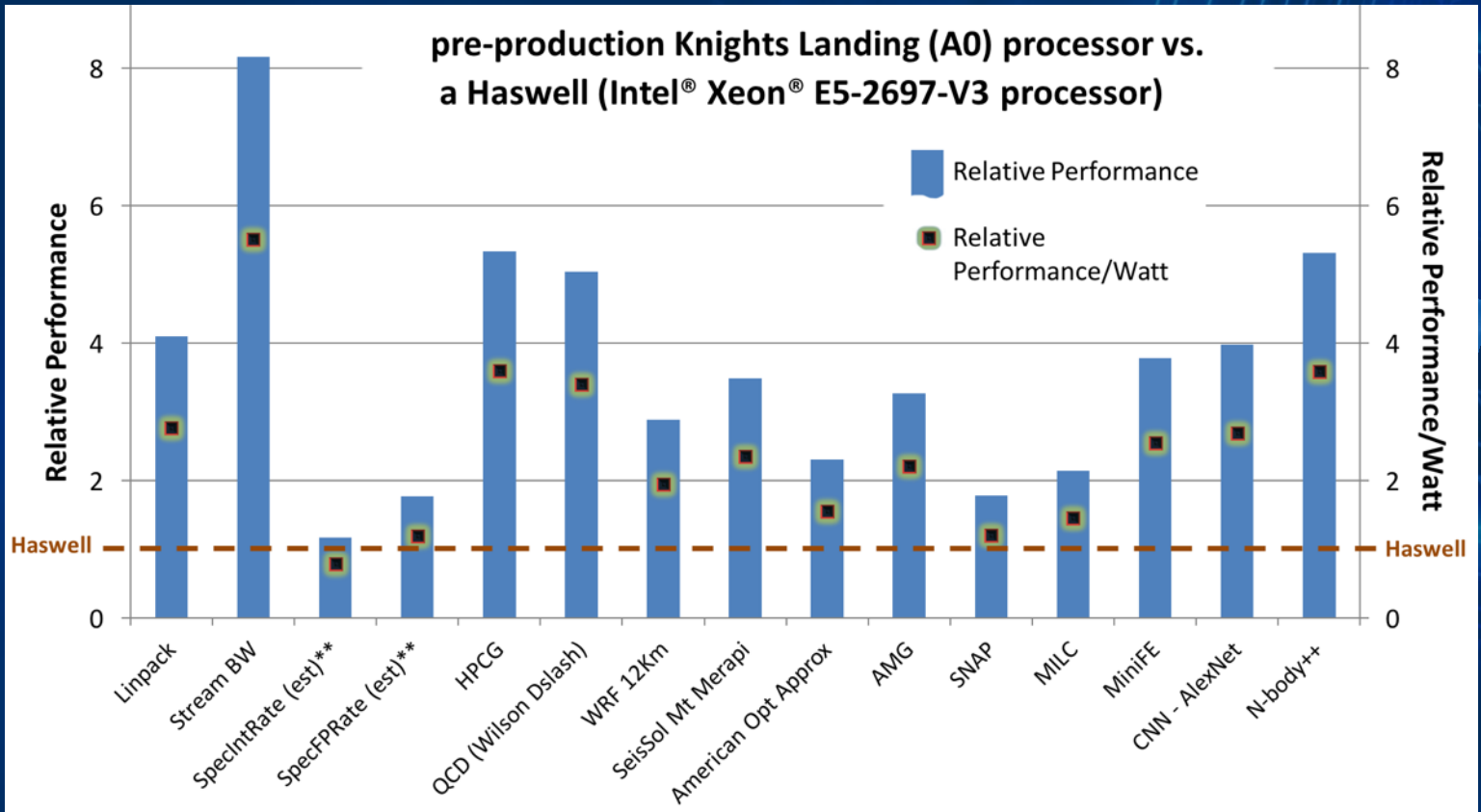
Integration of **Memory on package**: innovative memory architecture for high bandwidth and high capacity

Integration of **Fabric on package**

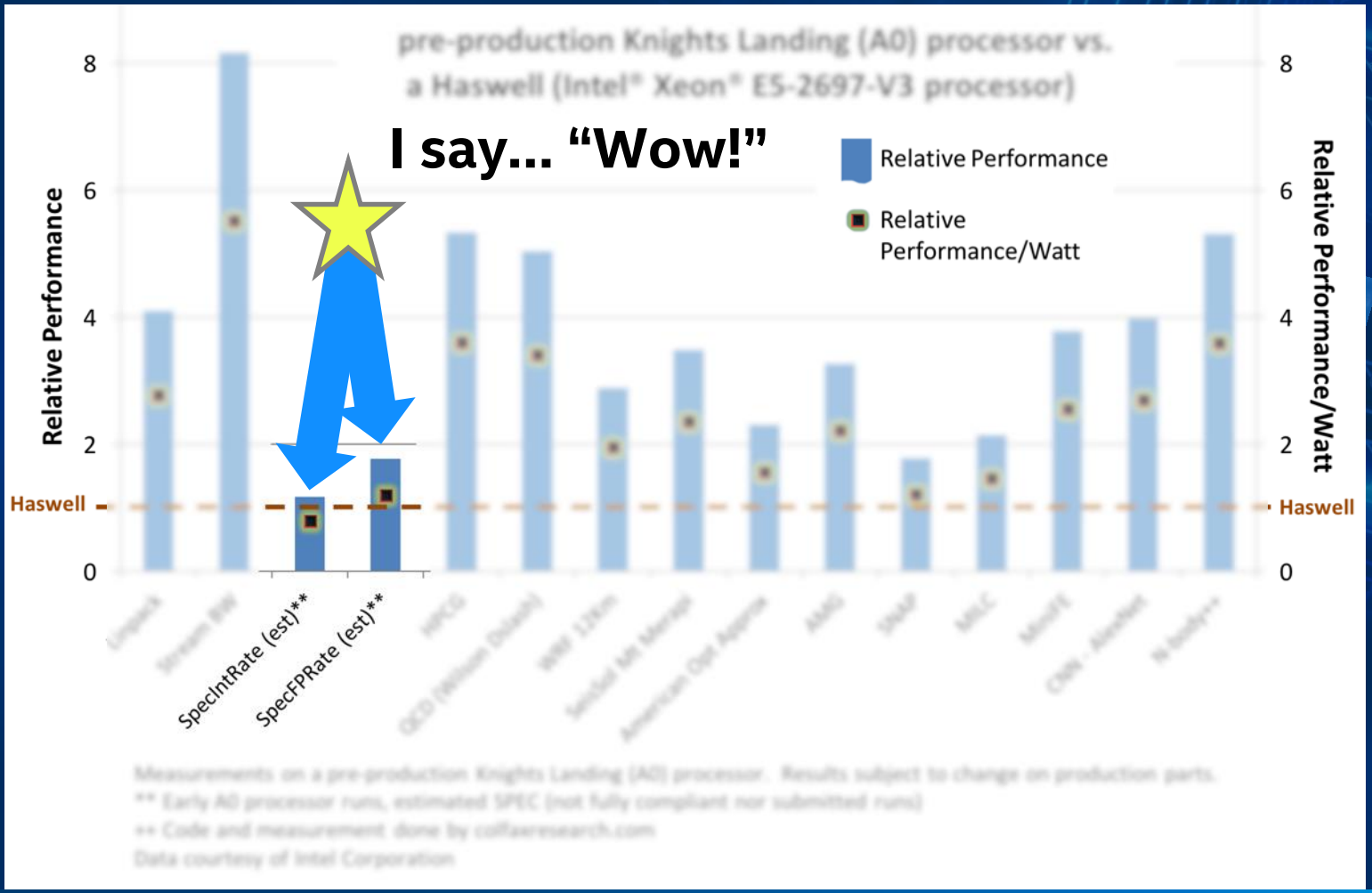
Three products

KNL Self-Boot	KNL Self-Boot w/ Fabric	KNL Card
(Baseline)	(Fabric Integrated)	(PCIe-Card)

Potential future options subject to change without notice.
All timeframes, features, products and dates are preliminary forecasts and subject to change without further notification.



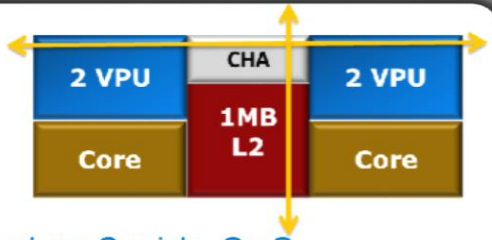
Measurements on a pre-production Knights Landing (A0) processor. Results subject to change on production parts.
** Early A0 processor runs, estimated SPEC (not fully compliant nor submitted runs)
++ Code and measurement done by colfaxresearch.com
Data courtesy of Intel Corporation


pre-production Knights Landing (A0) processor vs.
a Haswell (Intel® Xeon® E5-2697-V3 processor)
I say… "Wow!"
Relative Performance
Relative Performance/Watt
Haswell
SpecIntRate (est)**
SpecFPRate (est)**
Measurements on a pre-production Knights Landing (A0) processor. Results subject to change on production parts.
** Early A0 processor runs, estimated SPEC (not fully compliant nor submitted runs)
++ Code and measurement done by colfaxresearch.com
Data courtesy of Intel Corporation

***AVX-512**, High Bandwidth Memory, Cluster Mode, Omni-Path*

512-bit vectors via **AVX-512** means: **High performance** *and* **Binary compatibility**

KNL Tile: 2 Cores, each with 2 VPU
1M L2 shared between two Cores



Core: Changed from Knights Corner (KNC) to KNL. Based on 2-wide OoO Silvermont™ Microarchitecture, but with *many* changes for HPC.

4 thread/core. Deeper OoO. Better RAS. Higher bandwidth. Larger TLBs.

2 VPU: 2x AVX512 units. 32SP/16DP per unit. X87, SSE, AVX1, AVX2 and EMU

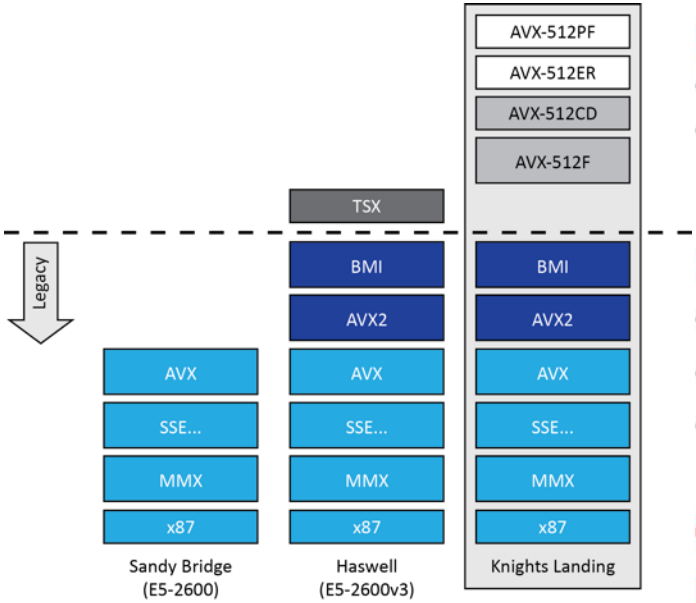
L2: 1MB 16-way. 1 Line Read and ½ Line Write per cycle. Coherent across all Tiles

CHA: **C**aching/**H**ome **A**gent. Distributed Tag Directory to keep L2s coherent. MESIF protocol. 2D-Mesh connections for Tile

AVX-512, *High Bandwidth Memory, Cluster Mode, Omni-Path*

Support of vectors under 512-bits in size means:

Binary compatibility



KNL implements all legacy instructions

- Legacy binary runs w/o recompilation
- KNC binary requires recompilation

KNL introduces AVX-512 Extensions

- 512-bit FP/Integer Vectors
- 32 registers, & 8 mask registers
- Gather/Scatter

Conflict Detection: Improves Vectorization

Prefetch: Gather and Scatter Prefetch

Exponential and Reciprocal Instructions

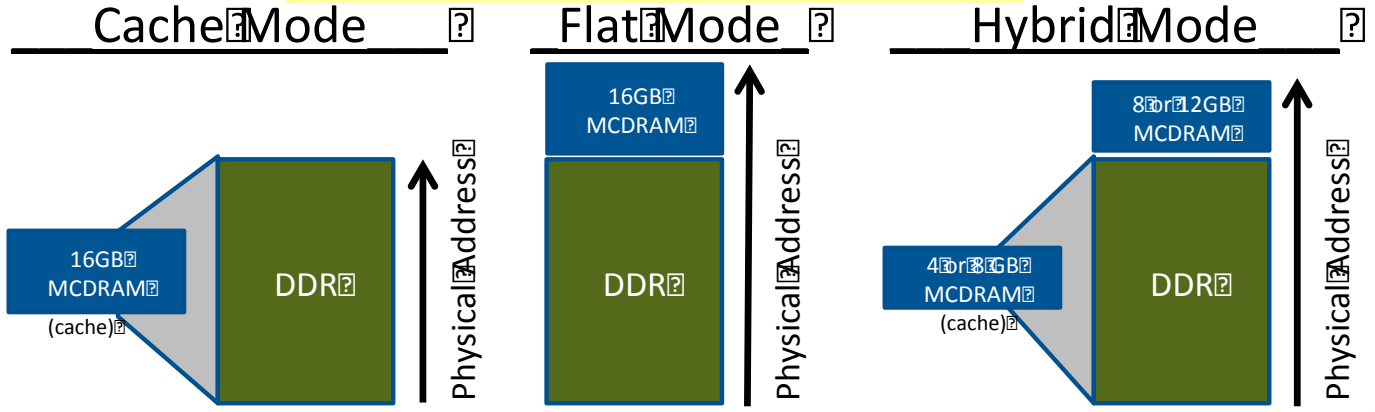
AVX-512, **High Bandwidth Memory,** Cluster Mode, Omni-Path

16MB on package DRAM means:

Performance *with options*

Memory Modes

Three Modes. Selected at boot



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

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

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

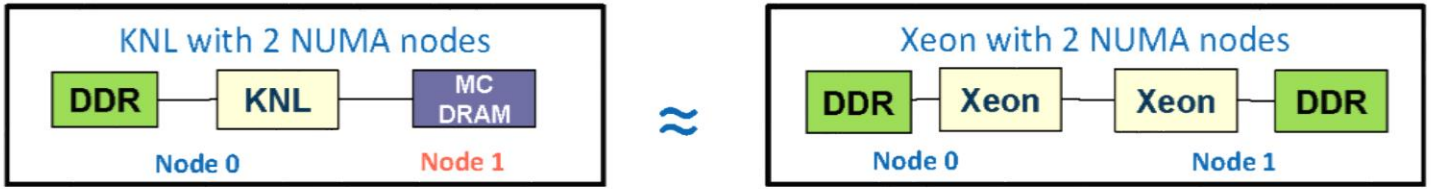
10

Standard *"NUMA" node recognition by BIOS, OS, and applications.*

Systems will eventually have this as a "norm."

Flat MCDRAM: SW Architecture

MCDRAM exposed as a separate NUMA node



Memory allocated in DDR by default → Keeps non-critical data out of MCDRAM.

Apps explicitly allocate critical data in MCDRAM. Using two methods:

- **"Fast Malloc"** functions in High BW library (https://github.com/memkind)
  - Built on top to existing *libnuma* API
- **"FASTMEM"** Compiler Annotation for Intel Fortran

Flat MCDRAM with existing NUMA support in Legacy OS

C/C++ "high bandwidth" malloc or new

Fortran "high bandwidth" allocatables

Flat MCDRAM SW Usage: Code Snippets

C/C++ (*https://github.com/memkind)

Allocate into DDR

```
float   *fv;
fv = (float *)malloc(sizeof(float)*100);
```

Allocate into MCDRAM

```
float   *fv;
fv = (float *)hbw_malloc(sizeof(float) * 100);
```

Intel Fortran

Allocate into MCDRAM

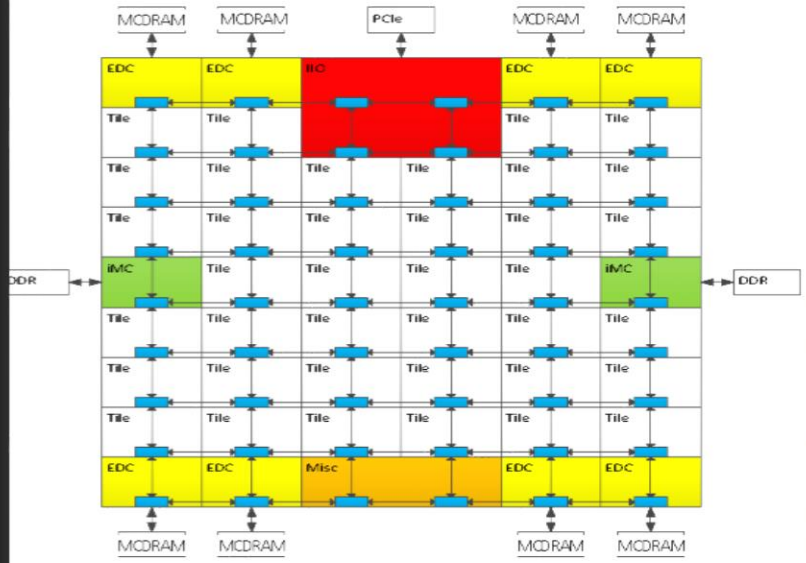
```
c      Declare arrays to be dynamic
       REAL, ALLOCATABLE :: A(:)

!DEC$ ATTRIBUTES, FASTMEM :: A

       NSIZE=1024
c      allocate array 'A' from MCDRAM
c
       ALLOCATE (A(1:NSIZE))
```

*AVX-512, High Bandwidth Memory, **Cluster Mode,** Omni-Path*

Mesh interconnect means:

Higher Performance *with options*

KNL Mesh Interconnect



Mesh of Rings

- Every row and column is a (half) ring
- YX routing: Go in Y → Turn → Go in X
- Messages arbitrate at injection and on turn

Cache Coherent Interconnect

- MESIF protocol (F = Forward)
- Distributed directory to filter snoops

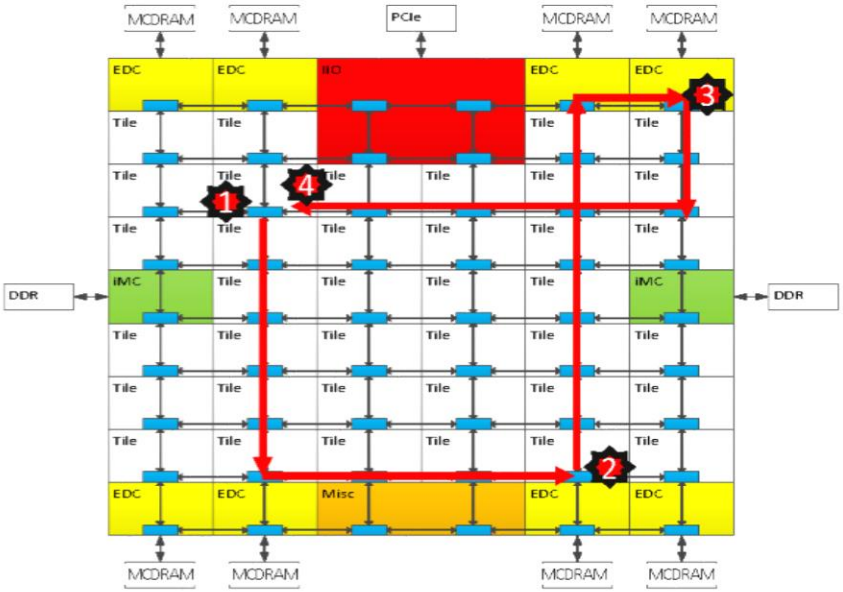
Three Cluster Modes

(1) All-to-All (2) Quadrant (3) Sub-NUMA Clustering

All-to-all
to
use all
cores
together
uniformly

for
instance –
OpenMP
across
all of KNL

Cluster Mode: All-to-All



Address uniformly hashed across all distributed directories

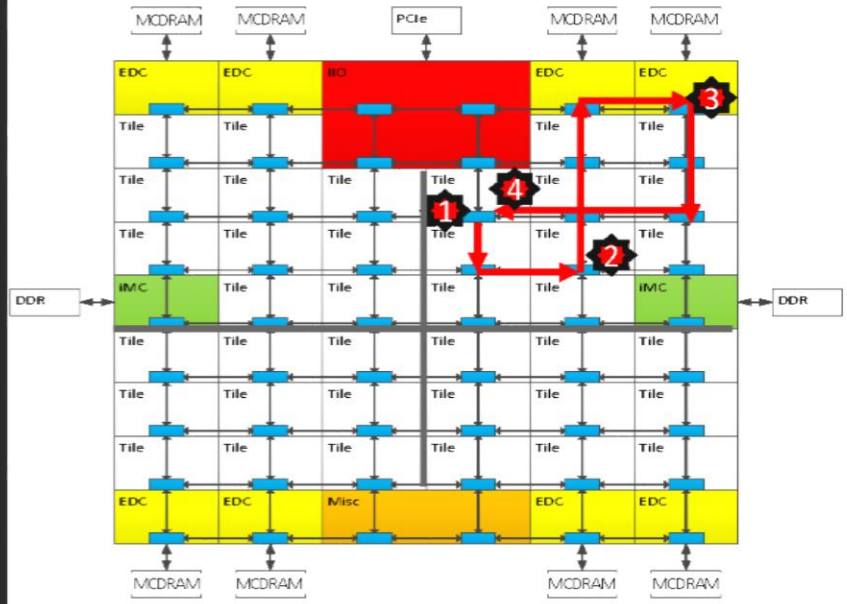
No affinity between Tile, Directory and Memory

Most general mode. Lower performance than other modes.

Typical Read L2 miss

1. L2 miss encountered
2. Send request to the distributed directory
3. Miss in the directory. Forward to memory
4. Memory sends the data to the requestor

Cluster Mode: Sub-NUMA Clustering (SNC)



1) L2 miss, 2) Directory access, 3) Memory access, 4) Data return

Each Quadrant (Cluster) exposed as a separate NUMA domain to OS.

Looks analogous to 4-Socket Xeon

Affinity between Tile, Directory and Memory

Local communication. Lowest latency of all modes.

SW needs to NUMA optimize to get benefit.

AVX-512, High Bandwidth Memory, Cluster Mode, **Omni-Path**

Integrated Fabric means Lower Cost, power; Higher Density; Plus the Advancements of Omni-Path in latency, bandwidth & scaling

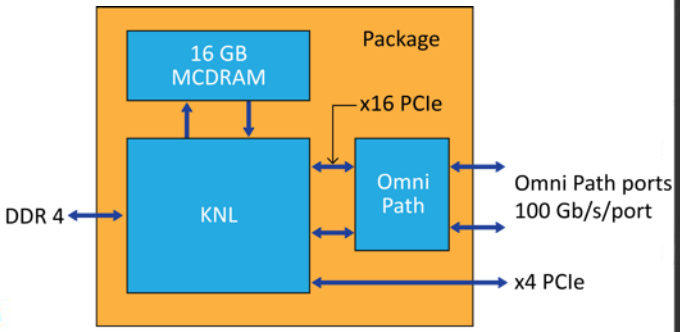
KNL with Omni-Path™

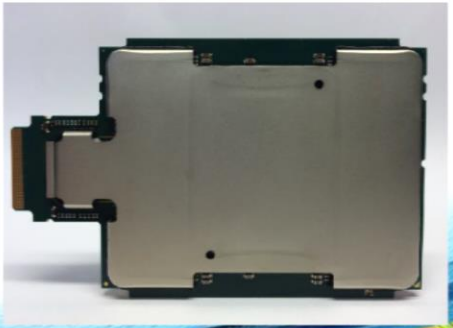
Omni-Path™ Fabric integrated *on package*

First product with integrated fabric

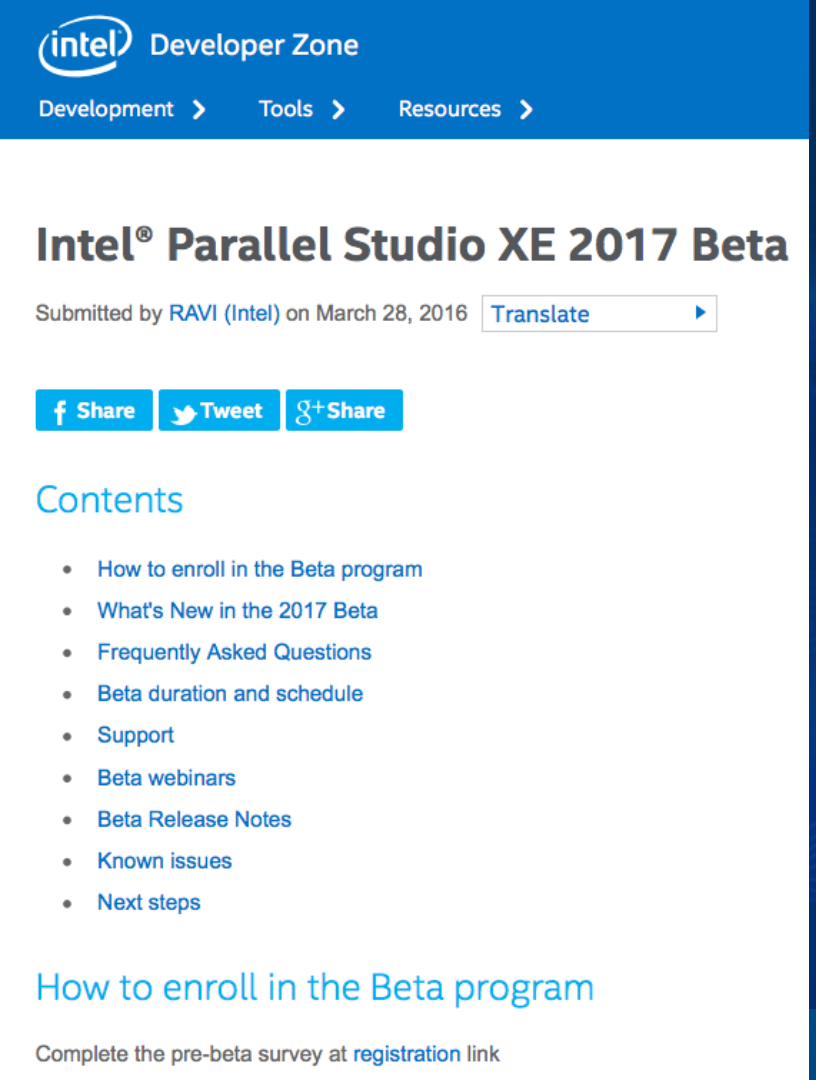
Connected to KNL die via 2 x16 PCIe* ports
Output: 2 Omni-Path ports
- 25 GB/s/port (bi-dir)

Benefits
- Lower cost, latency and power
- Higher density and bandwidth
- Higher scalability





*On package connect with PCIe semantics, with MCP optimizations for physical layer



http://
software.intel.com/
articles/
intel-parallel-studio-xe-2017-beta

BETA for “2017” Product – NOW

CHAPTER

Vectorization advisor 10

Many factors impact achieving good vectorization for our applications. The Vectorization Advisor directly analyzes an application and provides feedback on the extent of current vectorization and on possible steps to achieve more effective vectorization. Vectorization Advisor works with any compiler although some features in the Intel® compilers will increase the effectiveness of advice from the Vectorization Advisor tool. It is like having an expert sitting next to us who never tires of digging into an application to analyze what is really happening.

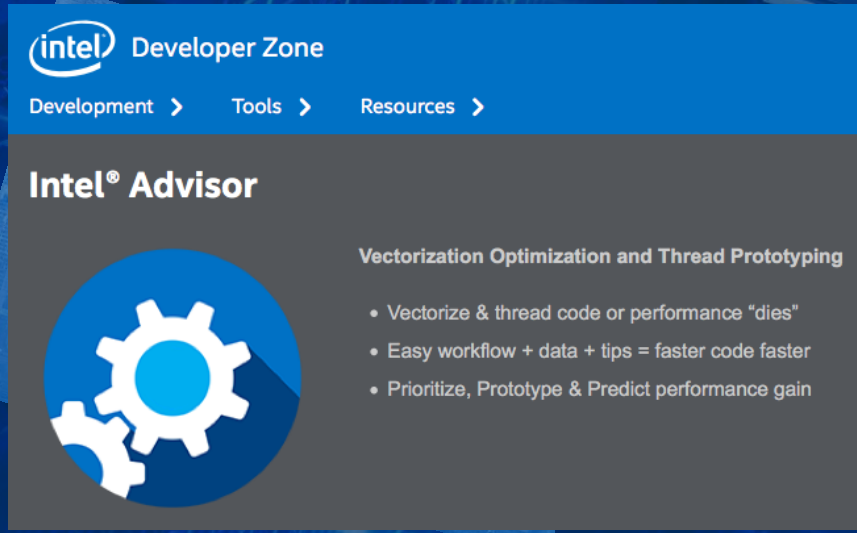
What is new with Knights Landing in this chapter?

AVX-512 and the Vectorization Advisor within the Intel® Advisor tool.

The Vectorization Advisor is one of the two major *workflows* (feature sets) available in the Intel® Advisor "2016" and later versions. The Intel Advisor also includes a thread prototyping feature set which can be useful for analysis of scaling for threads. In this chapter, we focus on using the Vectorization Advisor to help us maximize our vectorization performance.

How close is my application to maximum performance? Insight into this is helped by a "roofline model" analysis, in the Advisor Roofline Report section.

Intel® Advisor



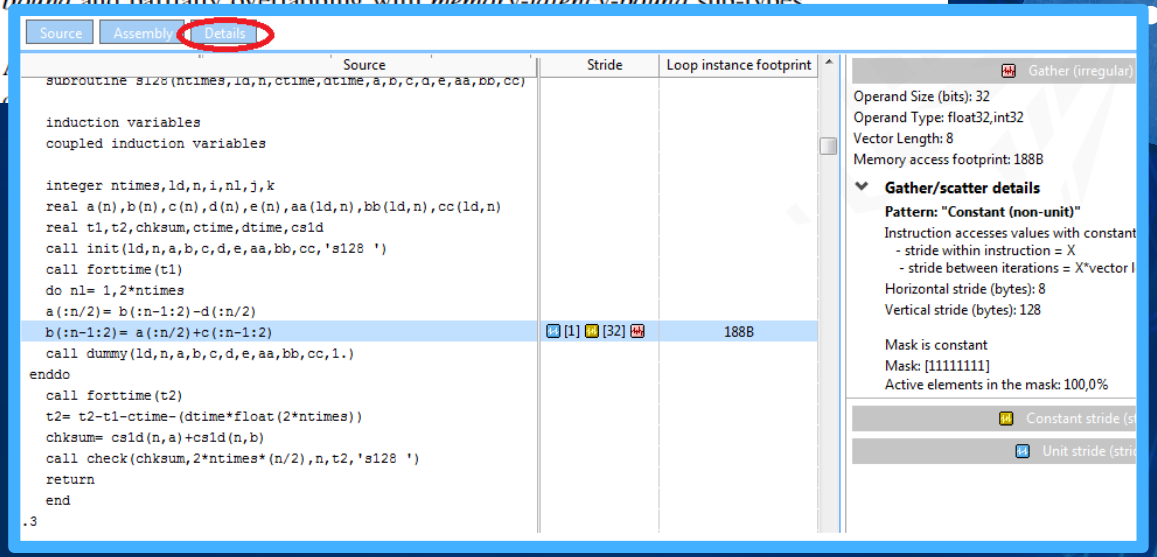
I will talk about some NEW "2017" features – which help Intel Xeon processors tuning and Intel Xeon Phi processor tuning BOTH – Of Course!

Memory Access Pattern Report

MEMORY ACCESS PATTERN REPORT

An initial **Survey** analysis of hot loops often identifies inefficient memory access patterns as a main bottleneck. Memory access patterns issues are the toughest and most frequent performance problem in code not yet modernized for vector SIMD parallelism.

Applying *straightforward* SIMD and threading optimizations often does not provide desirable speedups because some parts of applications (including vectorized hot loops) become *memory bound*. Memory-access-patterns-bound code is just one subtype of a larger memory-bound class of problems, along with *memory-bandwidth-bound* and partially overlapping with *memory-latency-bound* sub-types.



Can recommend:

- AoS to SoA
- AoSoA
- Use of SDLT
- Use of MCDRAM

Gather/Scatter Report

Mask Utilization and FLOPS Profiler

MASK UTILIZATION AND FLOPS PROFILER

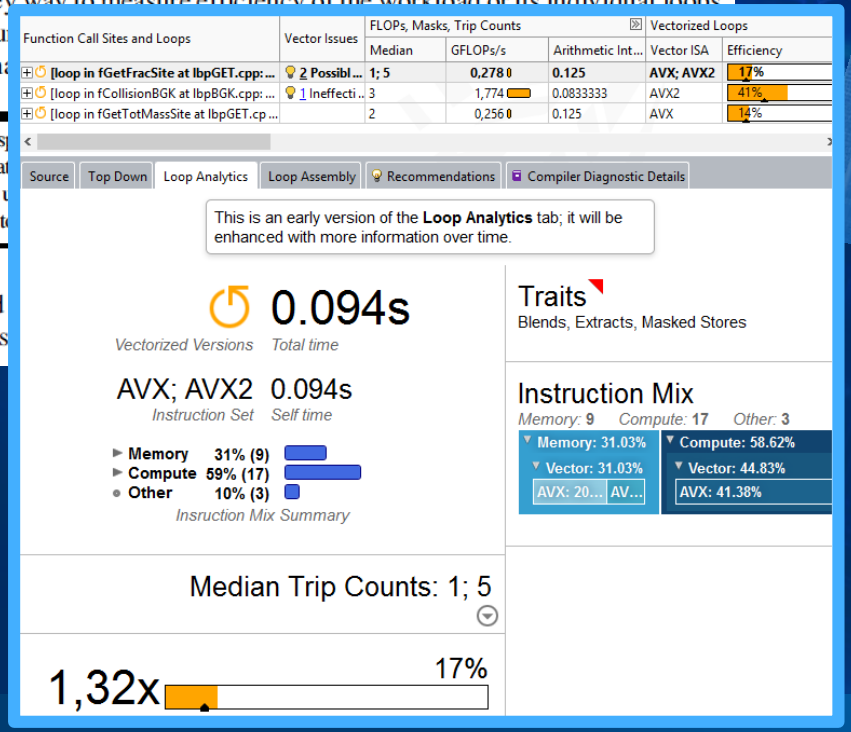
Counting FLOPs on Knights Landing is not ~~directly~~ supported by the hardware because there is no accounting for the values in mask registers when AVX-512 instructions are counted. ~~Certain~~ capabilities of the Advisor ~~tools~~ can make up for this lack of ~~direct~~ hardware support.

FLOP/s is a key way to measure efficiency of the workload or its individual loops or kernels. Measu

mance of target ha

In this book, and es
floating point operat
Elsewhere, it is not u
FLOP/s. We need t

While masked
torization, they als



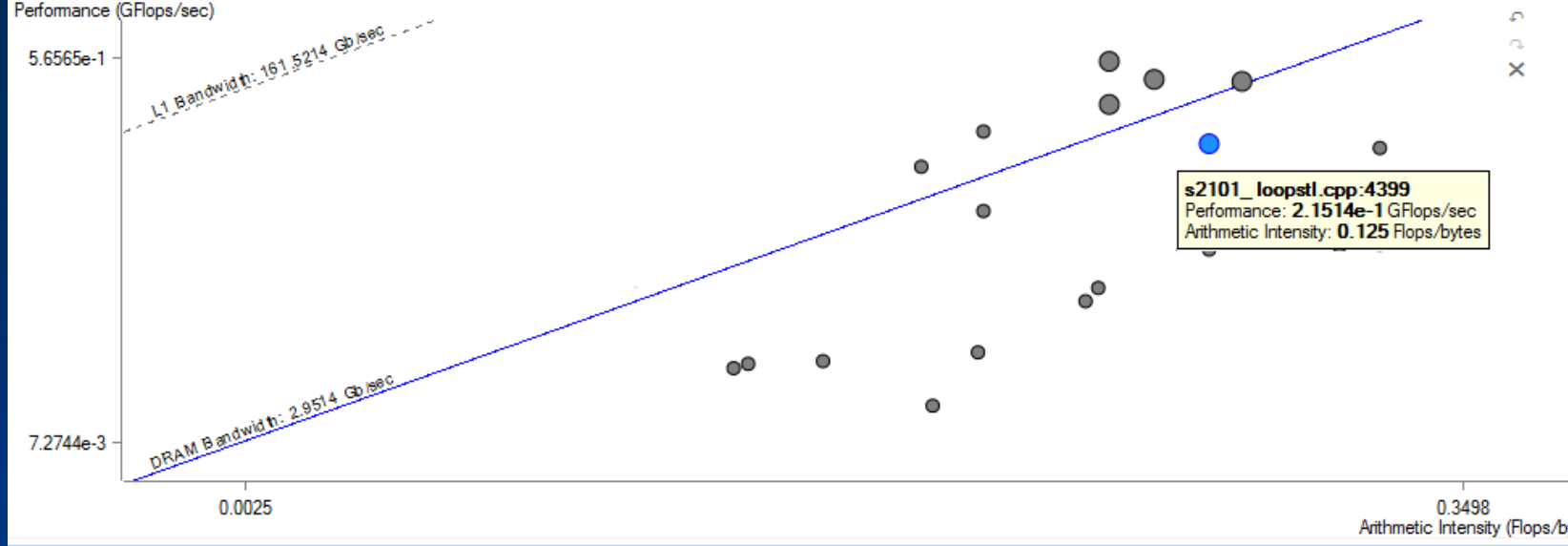
Mask-aware:

- FLOPs Report
- Vector Efficiency
- Memory Access Pattern,
- (coming soon): Roofline Analysis Graph

Vectorization efficiency and FLOP/s in Survey Report and Loop Analytics.

Roofline Report

in Intel® Advisor 2017 (soon)



Source | Top Down | Loop Assembly | Recommendations | Compiler Diagnostic Details

File: loopstl.cpp:4399 s2101_

Line	Source	Total Time	%	Loop Time	%	Traits
4399	for (i__ = 1; i__ <= i__2; ++i__)	0,031s		1,859s		
4400	aa[i__ + i__ * aa_dim1] += bb[i__ + i__ * bb_dim1] * cc[i__ + i__	1,828s				FMA
4401	* cc_dim1];					

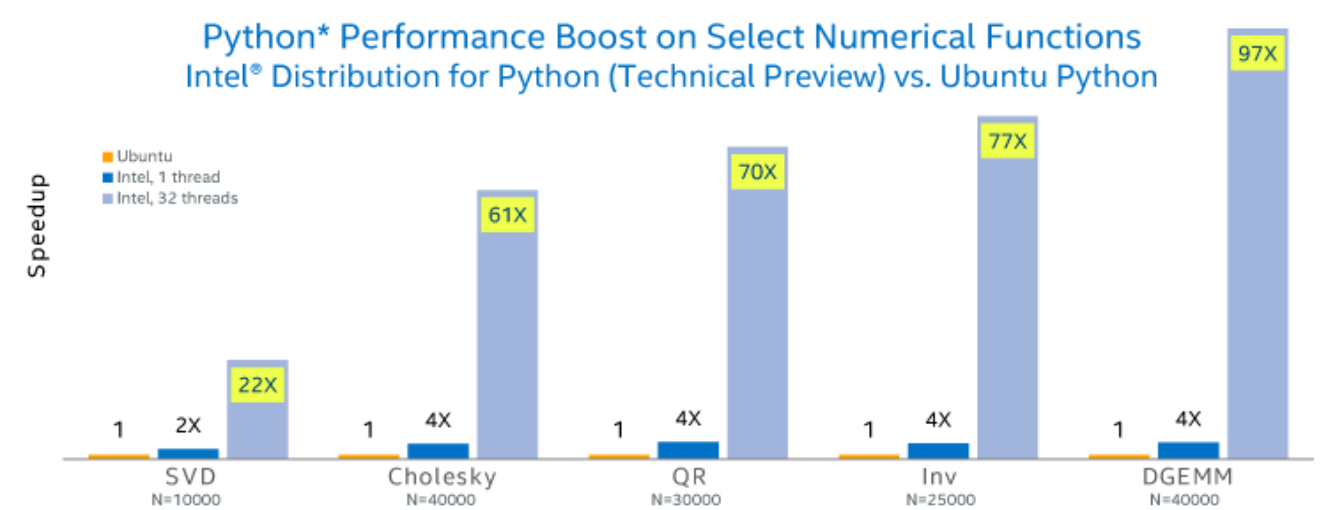
...supplements AI-based analysis with a dynamic FLOP/s profile and peak FLOPs and memory sub-system throughput levels providing enlightening “bounds and bottlenecks” analysis for complex workloads.

Intel® Distribution for Python*

- Faster NumPy/SciPy performance
  powered by Intel® Math Kernel Library
  and Intel® Threading Building Blocks
  and Intel® Data Analytics Acceleration Library
- ~3X speedups on single thread
- Easy installation
- Python 2.7 & 3.5
- Windows & Linux & OS X

http://bit.ly/intel-python

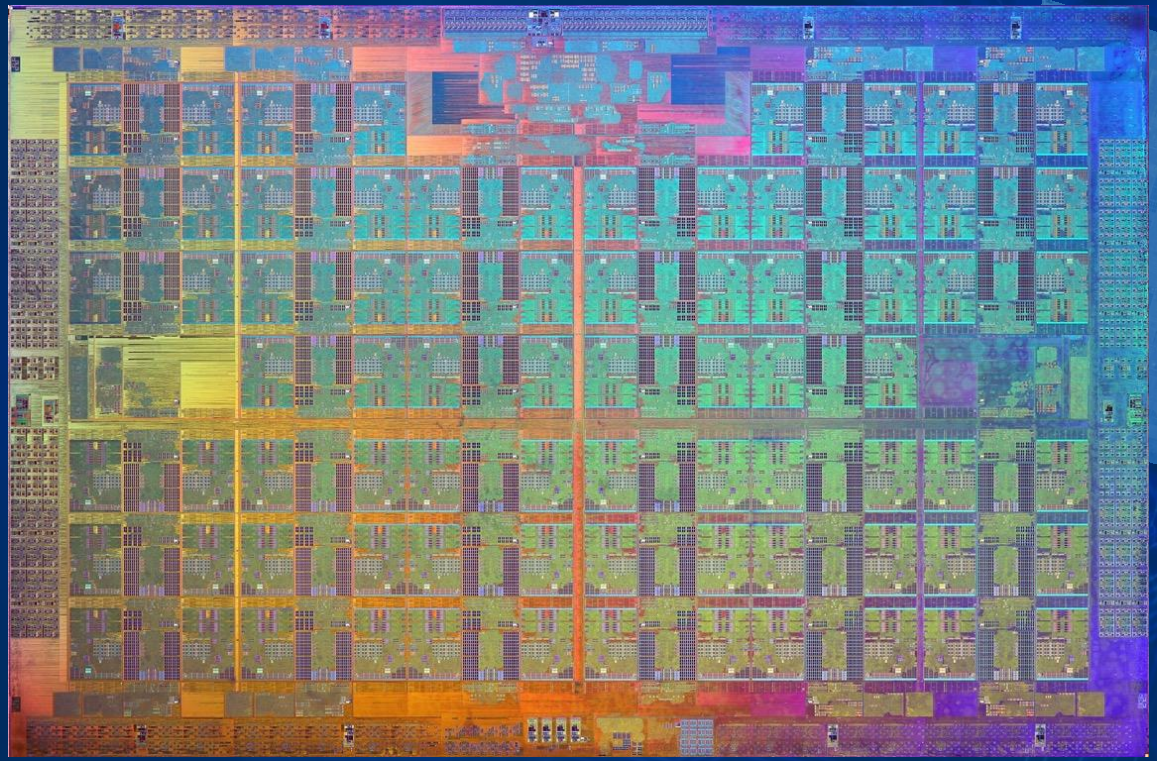
Some Performance Benchmarks
*Intel® Distribution for Python**

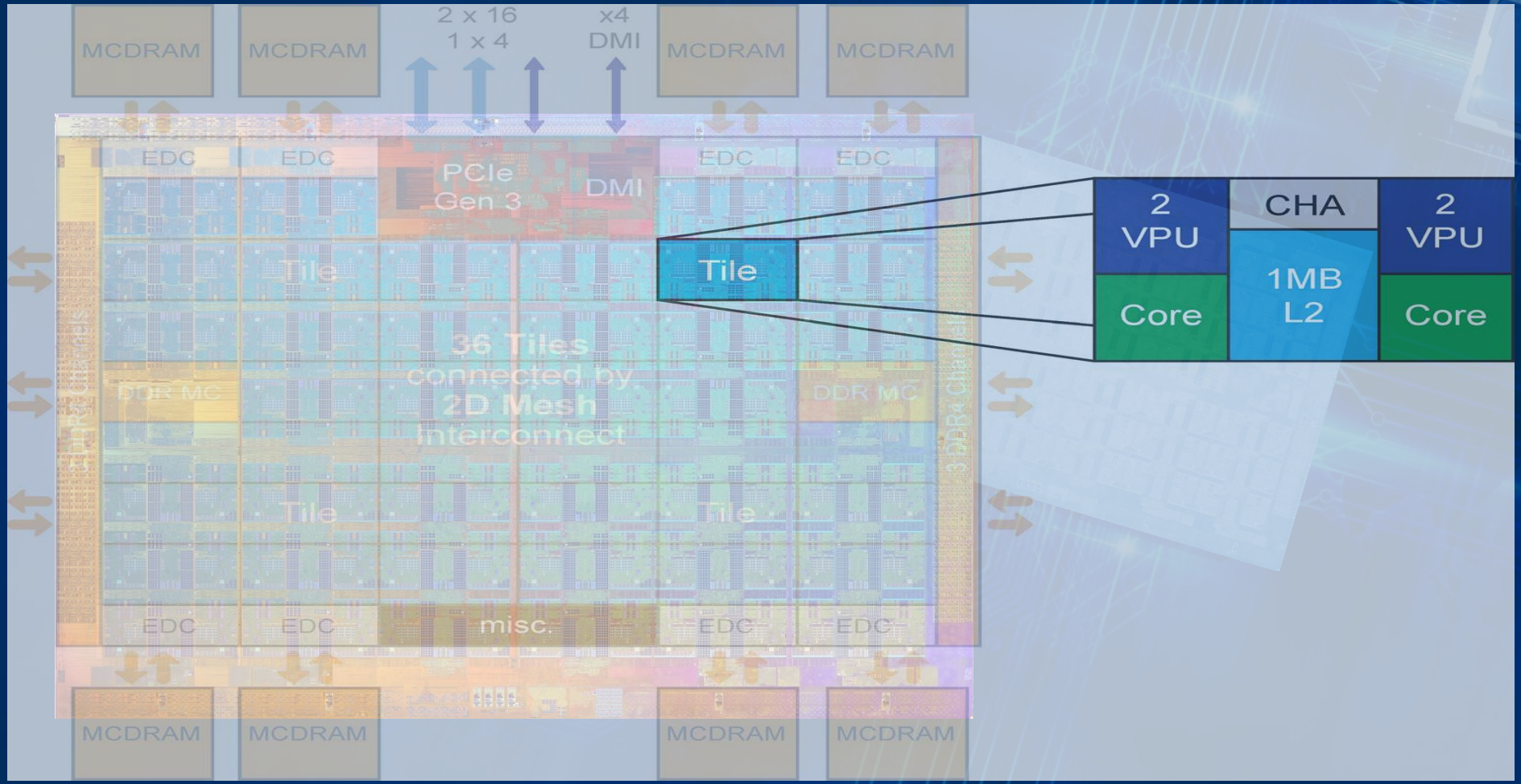


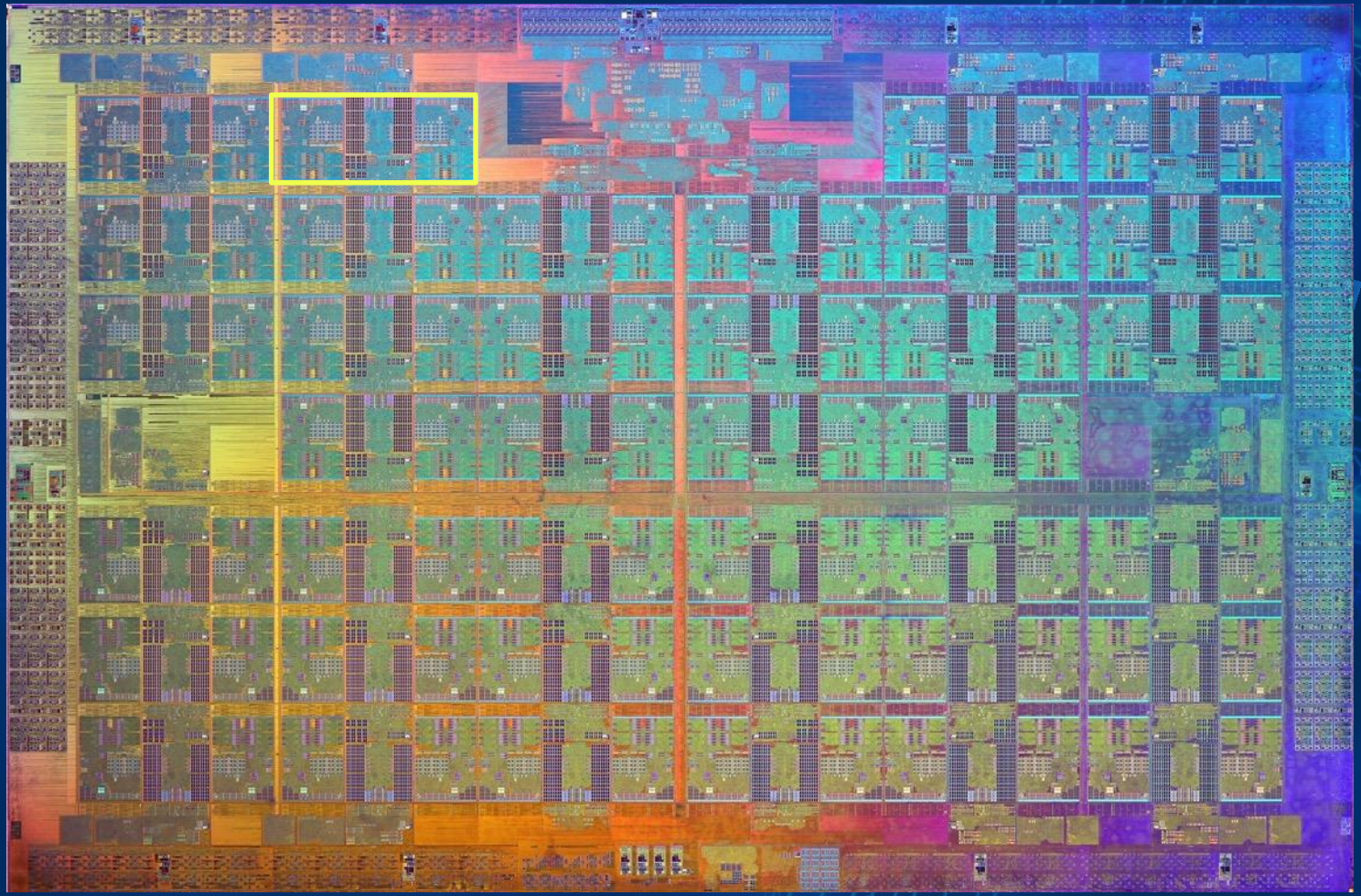
Configuration info: - Versions: Intel® Distribution for Python 2.7.10 Technical Preview 1 (Aug 03, 2015), Ubuntu* built Python*: Python 2.7.10, NumPy 1.9.2 built with gcc 4.8.4; Hardware: Intel® Xeon® CPU E5-2698 v3 @ 2.30GHz (2 sockets, 16 cores each, HT=OFF), 64 GB of RAM, 8 DIMMS of 8GB@2133MHz; Operating System: Ubuntu 14.04 LTS.

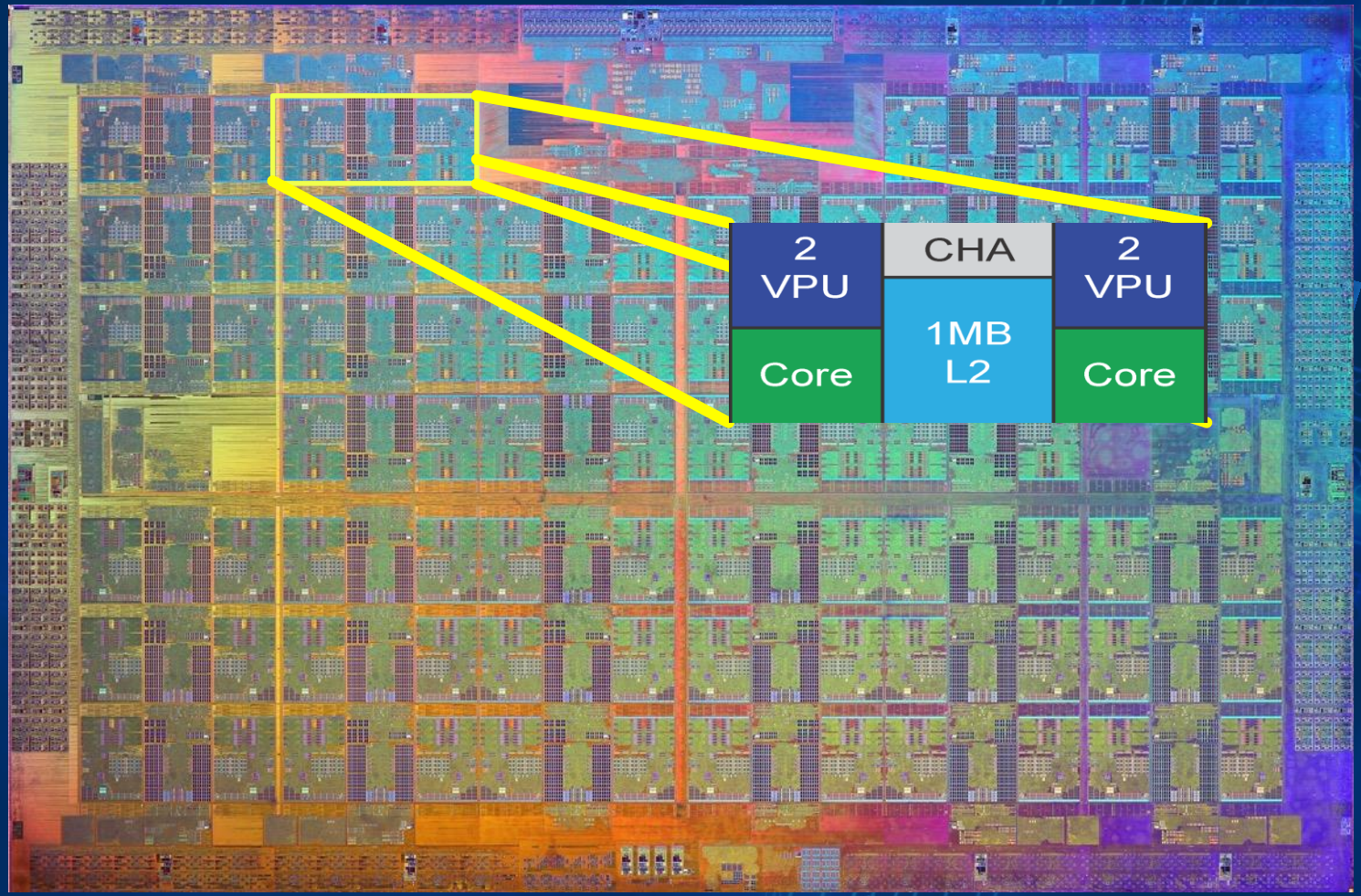
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. * Other brands and names are the property of their respective owners. Benchmark Source: Intel Corporation

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804 .




MCDRAM
MCDRAM
2 x 16
1 x 4
x4
DMI
MCDRAM
MCDRAM
EDC
EDC
PCIe
Gen 3
DMI
EDC
EDC
Tile
Tile
36 Tiles
connected by
2D Mesh
Interconnect
DDR MC
DDR MC
Tile
Tile
EDC
EDC
misc.
EDC
EDC
MCDRAM
MCDRAM
MCDRAM
MCDRAM
2
VPU
CHA
2
VPU
Core
1MB
L2
Core




2
VPU
CHA
2
VPU
Core
1MB
L2
Core


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

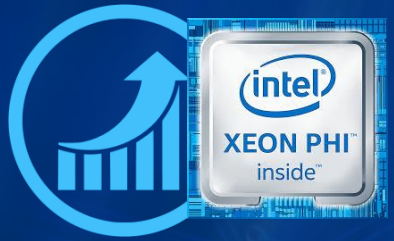
#ModernCode

Optimizations for Intel® Xeon® and Intel® Xeon Phi™ products share the same:

- ✓ Languages
- ✓ Directives
- ✓ Libraries
- ✓ Tools

XeonPhiDeveloper.com

You can buy you very own Knight Landing development system today!



Highly-Parallel Performance
to develop on

All the Software Tools & Libraries
you need

Support & Training
to help you succeed

Leading edge platform capabilities, performance to deliver multi-threaded, vectorized software for today's HPC workloads !

Q&A

Thank you!