Programming and Optimization for Intel® Architecture

The Hands-On Workshop (HOW) Series

Colfax International — @colfaxintl

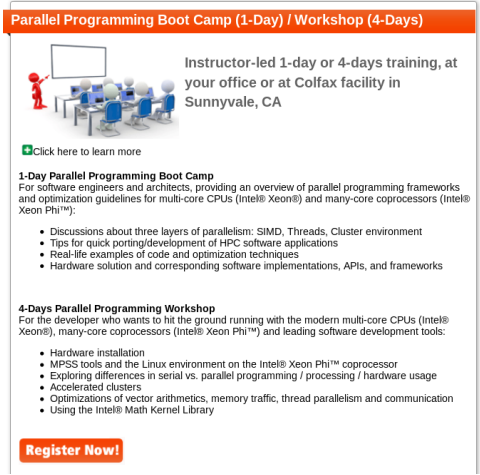
August 2016 , Rev. 03b

 © Colfax International, 2013–2016

About This Document

This document represents the materials of a Web-based training "Programming and Optimization with Intel Architecture" developed and run by Colfax International.

© Colfax International, 2013–2016



Parallel Programming Boot Camp (1-Day) / Workshop (4-Days)

Instructor-led 1-day or 4-days training, at your office or at Colfax facility in Sunnyvale, CA

Click here to learn more

1-Day Parallel Programming Boot Camp
For software engineers and architects, providing an overview of parallel programming frameworks and optimization guidelines for multi-core CPUs (Intel® Xeon®) and many-core coprocessors (Intel® Xeon Phi™):

- Discussions about three layers of parallelism: SIMD, Threads, Cluster environment
- Tips for quick porting/development of HPC software applications
- Real-life examples of code and optimization techniques
- Hardware solution and corresponding software implementations, APIs, and frameworks

4-Days Parallel Programming Workshop
For the developer who wants to hit the ground running with the modern multi-core CPUs (Intel® Xeon®), many-core coprocessors (Intel® Xeon Phi™) and leading software development tools:

- Hardware installation
- MPSS tools and the Linux environment on the Intel® Xeon Phi™ coprocessor
- Exploring differences in serial vs. parallel programming / processing / hardware usage
- Accelerated clusters
- Optimizations of vector arithmetics, memory traffic, thread parallelism and communication
- Using the Intel® Math Kernel Library

Register Now!

colfaxresearch.com/how-series

Disclaimer

While best efforts have been used in preparing this training, Colfax International makes no representations or warranties of any kind and assumes no liabilities of any kind with respect to the accuracy or completeness of the contents and specifically disclaims any implied warranties of merchantability or fitness of use for a particular purpose. The publisher shall not be held liable or responsible to any person or entity with respect to any loss or incidental or consequential damages caused, or alleged to have been caused, directly or indirectly, by the information or programs contained herein. No warranty may be created or extended by sales representatives or written sales materials.

© Colfax International, 2013–2016

Course Roadmap

- HOW to Program Intel Architecture
  - 01. Parallelism, specialization, guided tour – Aug 29
  - 02. Programming Intel Xeon Phi (KNC, KNL) – Aug 30
- HOW to Express Parallelism
  - 03. Automatic vectorization – Aug 31
  - 04. Multi-threading with OpenMP – Sep 1
- HOW to Get Performance
  - 05. Comprehensive demo – Sep 2
  - 06. Scalar & vectorization tuning – Sep 5
  - 07. Multi-threading I – Sep 6
  - 08. Multi-threading II – Sep 7
  - 09. Memory traffic – Sep 8
- HOW to Scale
  - 10. Distributed Computing: MPI – Sep 9

August 2016

S	M	T	W	H	F	S
	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			

September 2016

S	M	T	W	H	F	S
				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	

— Webinar+remote access

© Colfax International, 2013–2016

HOW Online

Course page: colfaxresearch.com/how-16-08

- Slides (including this one), code downloads
- Video of recorded sessions
- Chat (during webinars or offline)

Additional resources:

- More workshops like this one: colfaxresearch.com/training
- Video courses: colfaxresearch.com/video-courses

 © Colfax International, 2013–2016

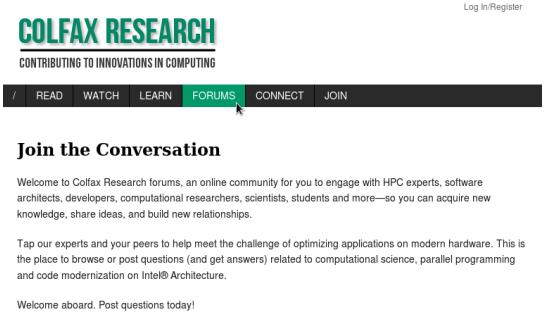
Get Your Questions Answered

Chat (current):

colfaxresearch.com/how-16-08

Forums (technical):

colfaxresearch.com/discussion



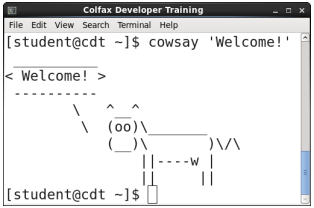
Email (organizational):

`training@colfax-intl.com`

 © Colfax International, 2013–2016

Hands-On Exercises and Remote Access

- 96 people receive a remote access token
- Virtualized Intel Xeon CPU, real Intel Xeon Phi coprocessor (1st gen, KNC), SW tools
- Can access the system the entire 2 weeks of the workshop



- Not among the 96? Stay tuned: follow along with instructor, use own system, or wait for a seat
- Use it or lose it: if you do not log in for a while, remote access token goes to next student on the list

 © Colfax International, 2013–2016

Learn More

© Colfax International, 2013–2016

HOW Series: Knights Landing

colfaxresearch.com/how-knl/

 © Colfax International, 2013–2016

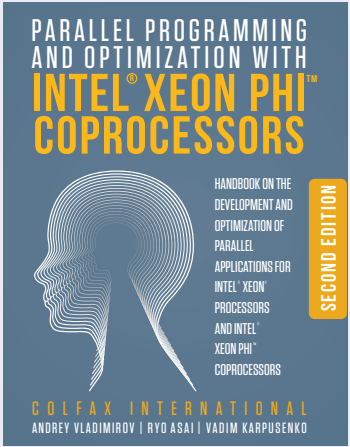
Textbook

ISBN: 978-0-9885234-0-1 (508 pages, Electronic or Print)

Parallel Programming
and Optimization with
Intel® Xeon Phi™
Coprocessors

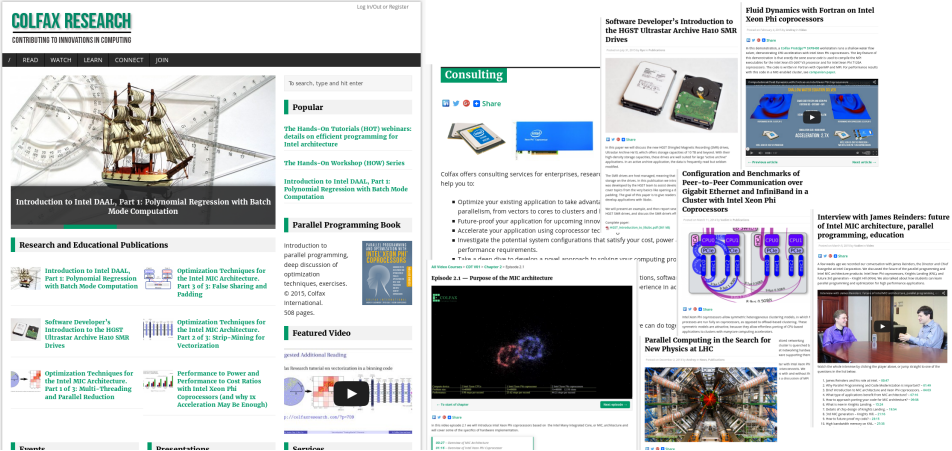
Handbook on the Development and
Optimization of Parallel Applications
for Intel® Xeon® Processors
and Intel® Xeon Phi™ Coprocessors

© Colfax International, 2015



http://xeonphi.com/book

Colfax Research



http://colfaxresearch.com/

 © Colfax International, 2013–2016

§2. Intel Architecture

 © Colfax International, 2013–2016

Computing Platforms

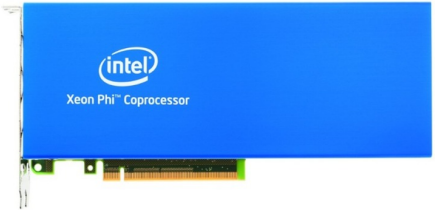
 © Colfax International, 2013–2016

Computing Platforms

Intel Xeon Processor

Current: Broadwell
Upcoming: Skylake

Intel Xeon Phi Coprocessor, 1st generation



Knights Corner (KNC)

Intel Xeon Phi Processor, 2nd generation*

* socket and coprocessor versions

Knights Landing (KNL)

Multi-Core Architecture

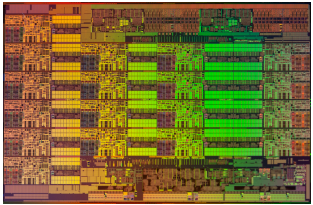
Intel Many Integrated Core (MIC) Architecture

 © Colfax International, 2013–2016

Intel Xeon CPU: Purpose and Specifications

General-purpose platform for demanding computing applications.

- Up to ~ 1 TFLOP/s in DP*
- Up to ~ 2 TFLOP/s in SP*
- Up to 3072 GiB DDR4 RAM*
- ~ 154 GB/s bandwidth*
- Hardware-rich: forgiving of sub-optimal code



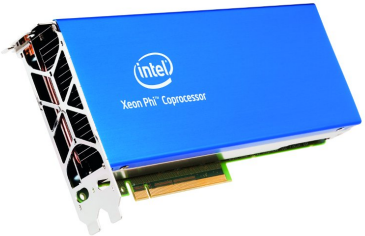
* 2-way Intel Xeon processor, Skylake architecture, top-of-the-line (e.g., E5-2699 V4)

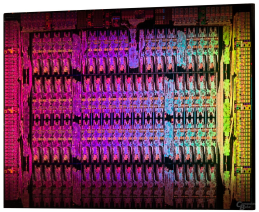
 © Colfax International, 2013–2016

Intel Xeon Phi Processors (1st Gen)

Specialized platform for demanding computing applications.

- PCIe add-in card
- ~ 1.2 TFLOP/s in DP
- ~ 2.4 TFLOP/s in SP
- Up to 16 GiB GDDR5 RAM
- ~ 176 GB/s bandwidth
- Heterogeneous clustering
- Runs special Linux distribution



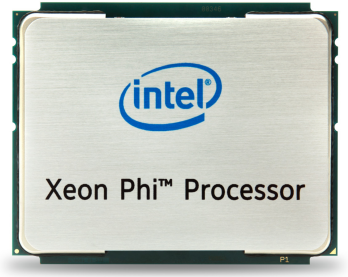


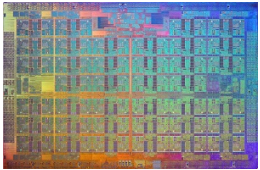
 © Colfax International, 2013–2016

Intel Xeon Phi Processors (2nd Gen)

Specialized platform for demanding computing applications.

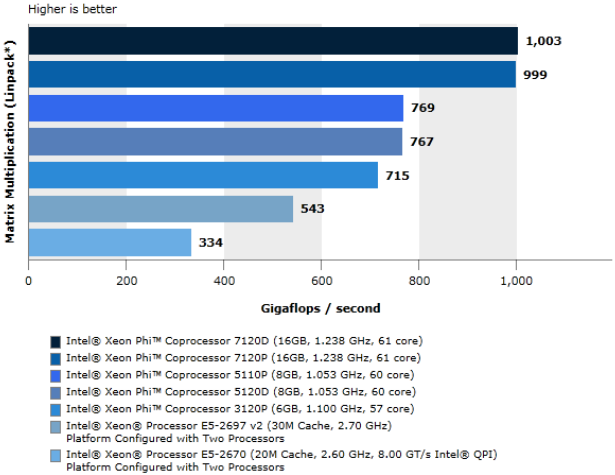
- Socket version or coprocessor
- 3+ TFLOP/s in DP
- 6+ TFLOP/s in SP
- Up to 16 GiB MCDRAM
- ~ 400 GB/s MCDRAM bandwidth
- Up to 384 GiB DDR4 RAM
- ~ 90 GB/s DDR4 bandwidth
- Supports common OS
- Public disclosures





 © Colfax International, 2013–2016

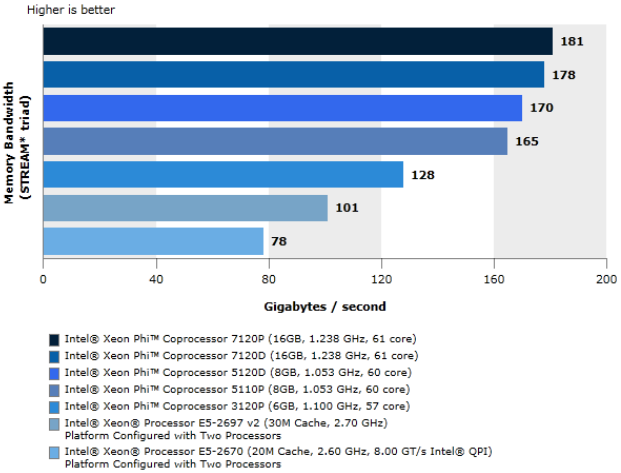
Intel Xeon Phi Coprocessors: HPC LINPACK Benchmark



Source: "Intel Xeon Phi Coprocessor LINPACK* and STREAM* Performance"

 © Colfax International, 2013–2016

Intel Xeon Phi Coprocessors: STREAM Benchmark



Source: "Intel Xeon Phi Coprocessor LINPACK* and STREAM* Performance"

 © Colfax International, 2013–2016

It Is All About Performance

© Colfax International, 2013–2016

Computing Applications

Just some examples

Computing

Finance
- Risk analysis
- Fraud prevention
- Real-time hedging

Internet
- Natural speech interaction
- Video streaming and archiving
- Facial recognition

Natural sciences
- Drug development
- Space weather
- New particle discovery

Medicine
- Medical imaging
- Radiation treatment simulation
- Human error detection

Aerospace
- Rocket engine simulation
- Radiation protection studies
- Aircraft aerodynamics

Energy
- Oil reservoir simulation
- Climate change science
- Nuclear fusion research

Manufacturing
- Simulation of material processing
- Development of new materials
- Robotic operations

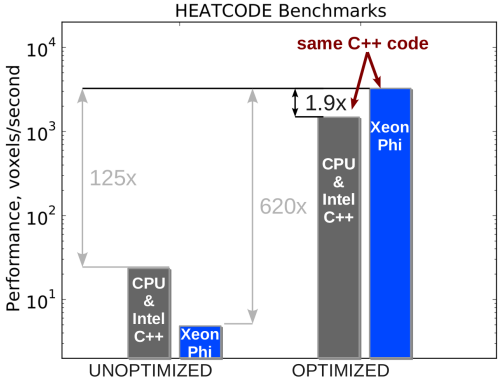
Environment
- Weather prediction
- Earthquake simulation
- Pollutant propagation

© Colfax International, 2013–2016

Will My Code Run Faster on KNL?

Performance on MIC architecture is a function of optimization Level

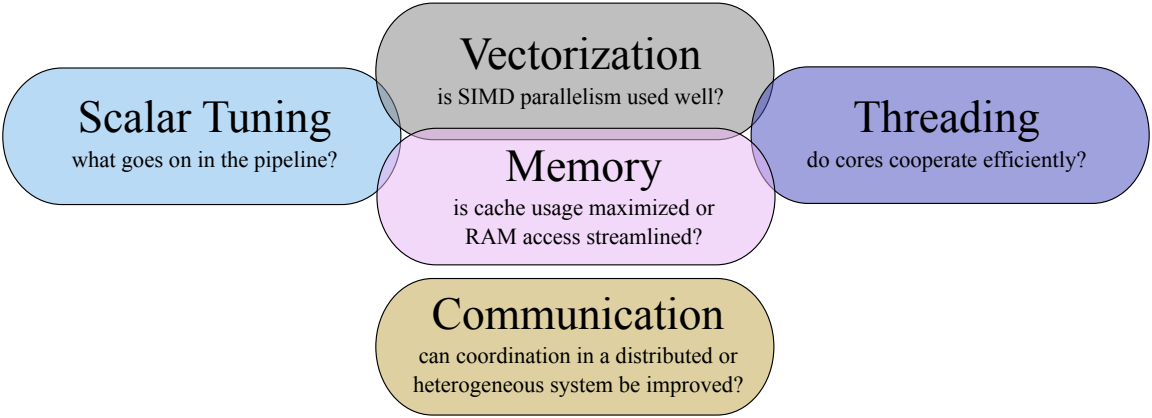
- Performance will be disappointing if code is not optimized for multi-core CPUs
- Optimized code runs better on the MIC platform *and* on the multi-core CPU
- Single code for two platforms + Ease of porting = Incremental optimization



See this case study

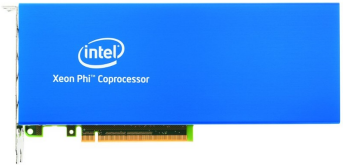
© Colfax International, 2013–2016

Optimization Areas



© Colfax International, 2013–2016

Coprocessor vs Processor Performance



vs.

One Intel Xeon Phi 7120P coprocessor

Two Intel Xeon E5-2697 v2 CPUs

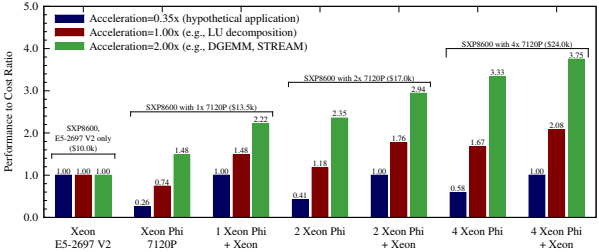
- Why compare 1 coprocessor agains 2 processors?
  Same thermal design power (TDP).

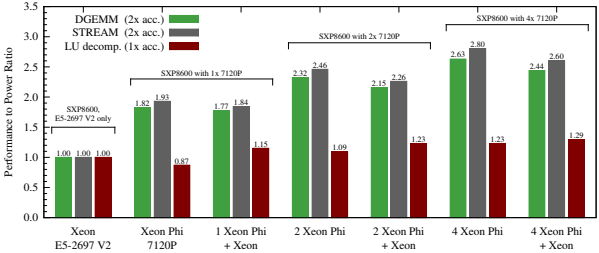
See also "Intel Xeon Product Family: Performance Brief"

 © Colfax International, 2013–2016

What is Your Performance Metric?

- Performance per System
- Performance per Watt
- Performance/Cost Ratio





See this paper for details

 © Colfax International, 2013–2016

Common Architecture

 © Colfax International, 2013–2016

Bird's Eye View

- C/C++/Fortran
- Linux/Windows
- ≤3 TiB of DDR4 RAM
- ≤44 cores/chip (2-way) ≈3 GHz
- 2 hyper-threads per core
- 256-bit AVX

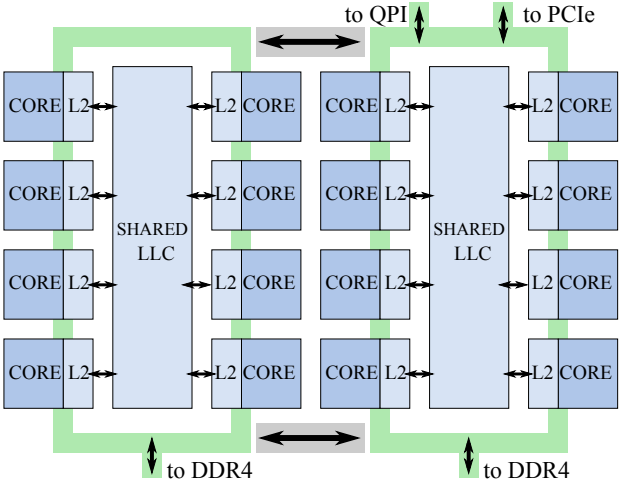
- C/C++/Fortran
- Special Linux
- 3–16 GiB cached GDDR5 RAM
- Up to 61 cores at ≈1.2 GHz
- 4 hardware threads per core
- 512-bit IMCI

- C/C++/Fortran
- Linux/Windows
- 16 GiB HBM (MCDRAM)
- Up to 72 cores at ≈1.4 GHz
- 4 hyper-threads per core
- 512-bit AVX-512

 © Colfax International, 2013–2016

Intel Xeon CPU: Die Organization

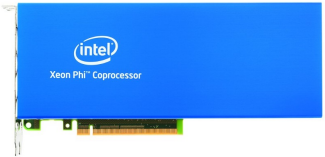
Likes data locality, but large LLC is forgiving.

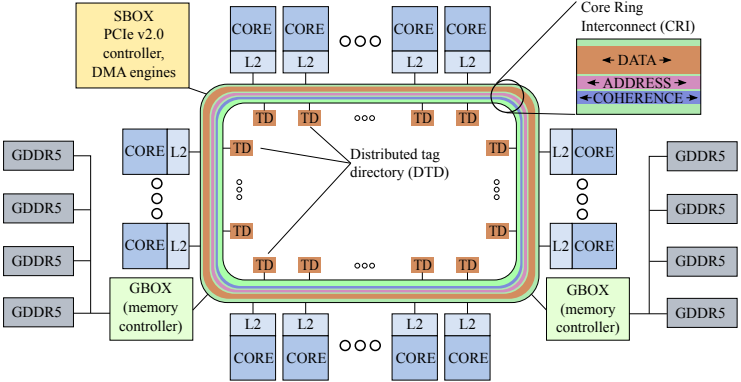


 © Colfax International, 2013–2016

KNC Die Organization

In a ring bus with distributed cache, data access locality is key.

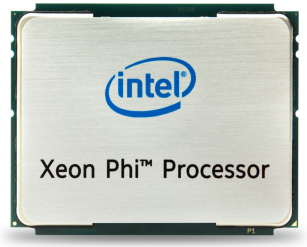


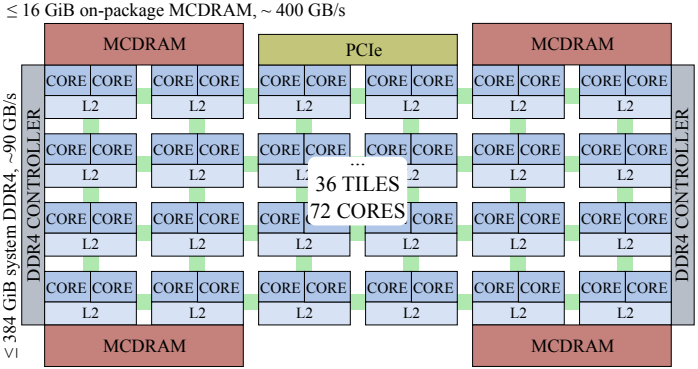


 © Colfax International, 2013–2016

KNL Die Organization

- Mesh interconnect relaxes data locality requirement [somewhat]
- All-to-all, quadrant or sub-numa domain communication in mesh

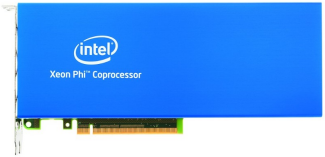


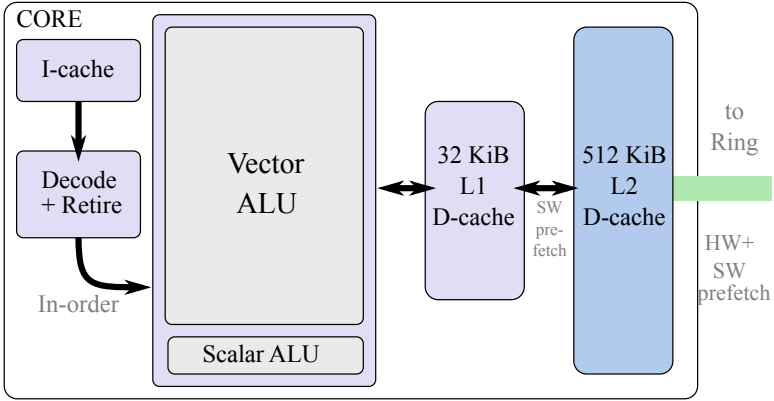


 © Colfax International, 2013–2016

KNC Cores

Computing power is in vector units. Scalar support only for legacy usage.

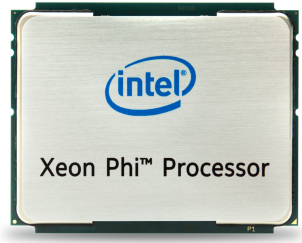


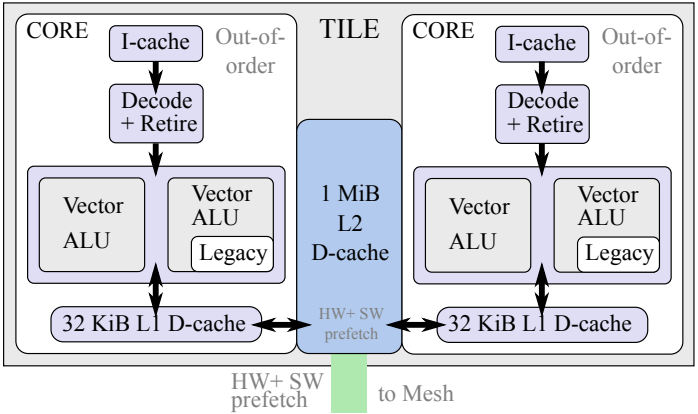


 © Colfax International, 2013–2016

KNL Cores

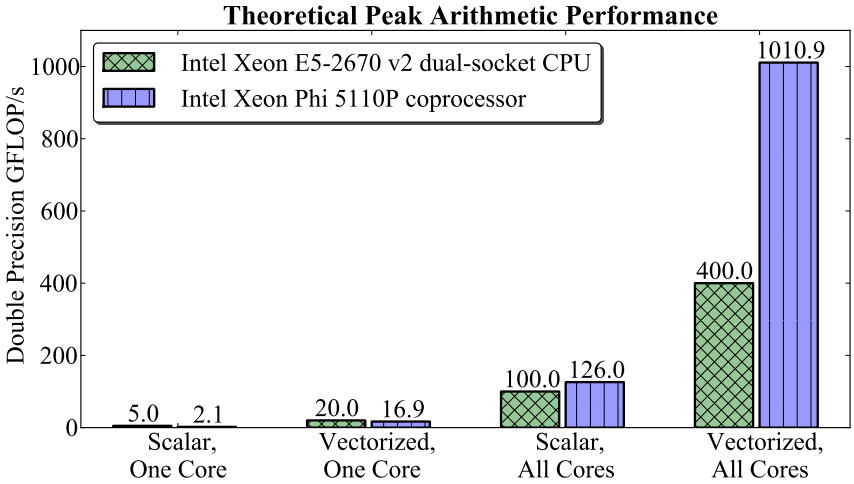
- Even more power in vector units
- Binary compatible with Xeon, but in legacy mode





 © Colfax International, 2013–2016

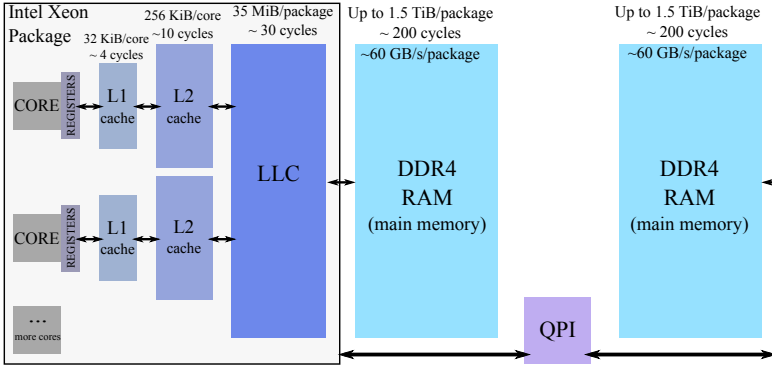
Task and Data Parallelism



 © Colfax International, 2013–2016

Intel Xeon CPU: Memory Organization

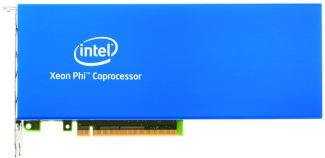
- Hierarchical cache structure
- Two-way processors have NUMA architecture

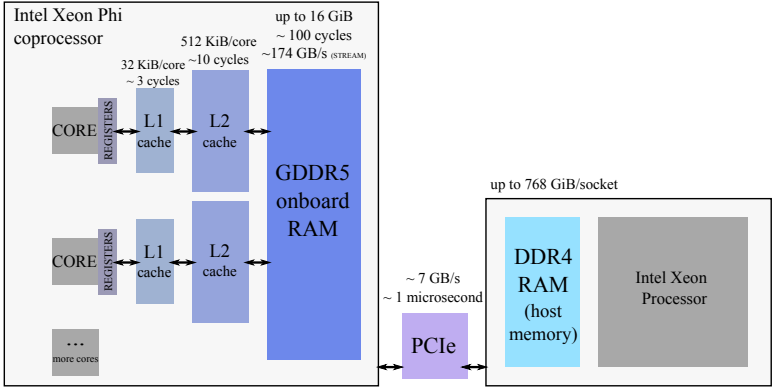


 © Colfax International, 2013–2016

KNC Memory Organization

- Direct access to ≤ 16 GiB of cached GDDR5 memory on board
- No access to system DDR4, connected to host via PCIe

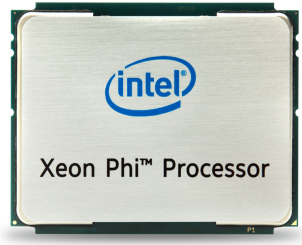


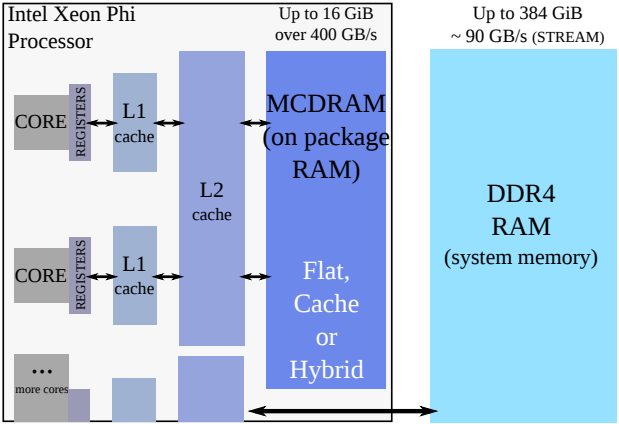


 © Colfax International, 2013–2016

KNL Memory Organization

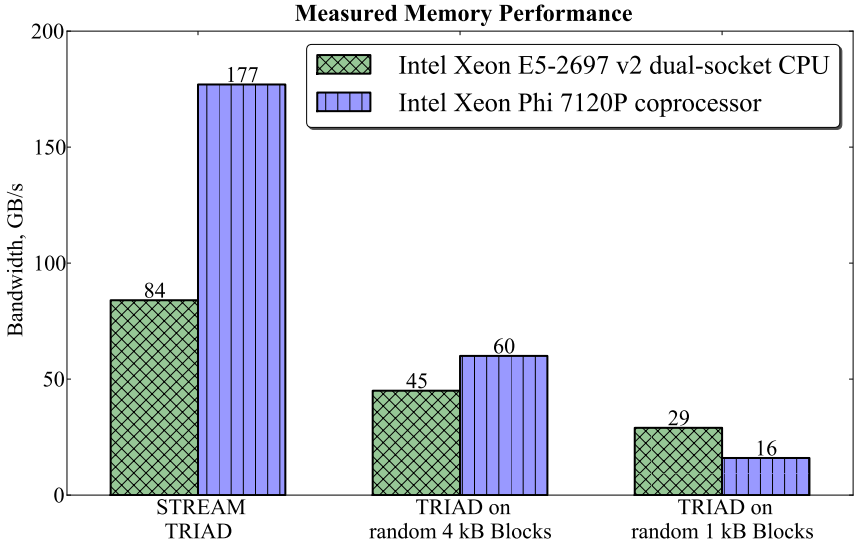
- Direct access to on-package MCDRAM *and* system DDR4 (socket)
- Use MCDRAM as cache, in flat mode, or as hybrid





 © Colfax International, 2013–2016

Memory Access Pattern



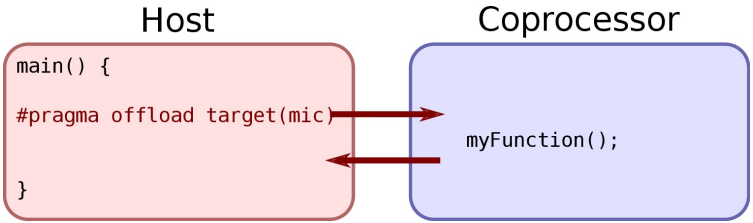
 © Colfax International, 2013–2016

Programming Coprocessors

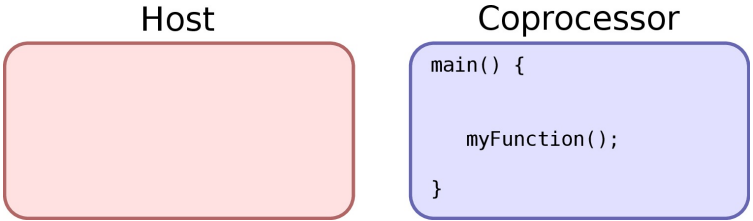
 © Colfax International, 2013–2016

Offload and Native Models

- Offload model (explicit/virtual-shared memory/OpenMP 4.0):



- Native model (standalone application/MPI process):

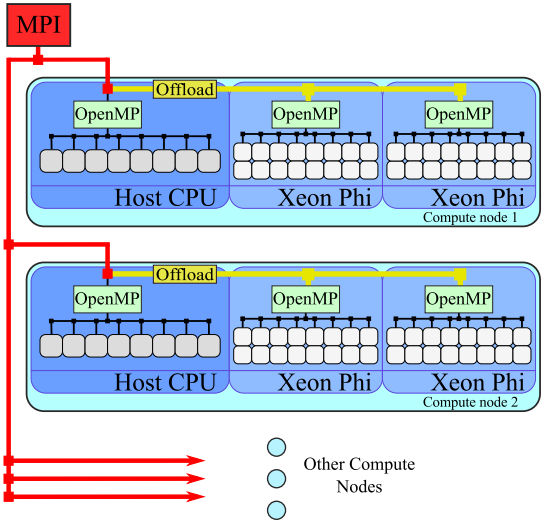


 © Colfax International, 2013–2016

Heterogeneous Distributed Computing with Xeon Phi

Option 1: MPI+OpenMP with Offload.

- MPI processes are multi-threaded with OpenMP.
- MPI runs only on CPUs.
- MPI processes offload to coprocessor(s).
- OpenMP in offload regions.

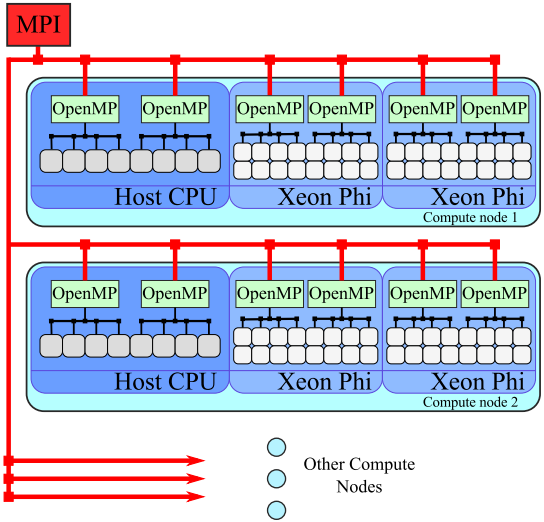


 © Colfax International, 2013–2016

Heterogeneous Distributed Computing with Xeon Phi

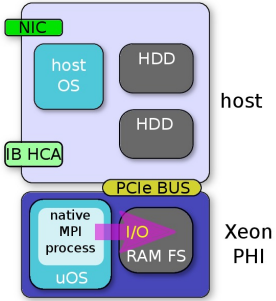
Option 2: Symmetric hybrid MPI+OpenMP.

- MPI processes on hosts
- Native MPI processes on the coprocessor.
- Multi-threading with OpenMP.

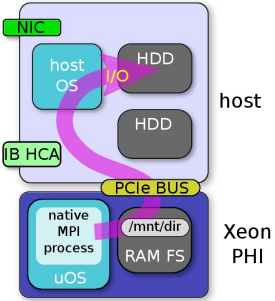


 © Colfax International, 2013–2016

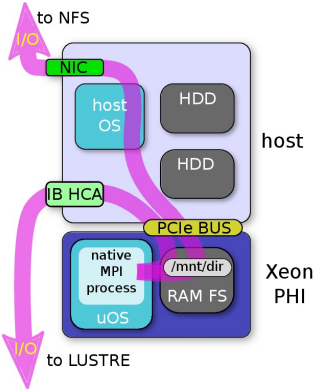
Working with Files on Coprocessors



RAM Filesystem



VirtIO Transfer

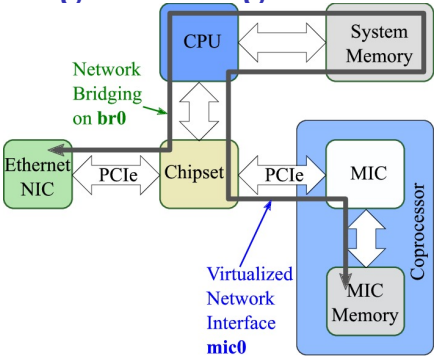


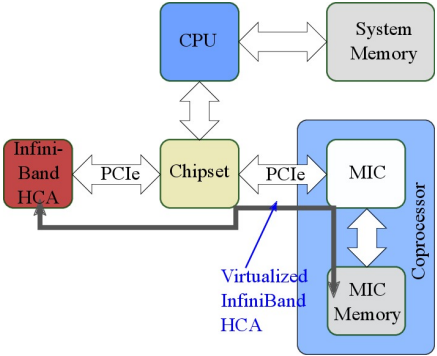
Network Storage

Details: `http://xeonphi.com/papers/io`

 © Colfax International, 2013–2016

Bridged Configuration for Peer-to-Peer Communication





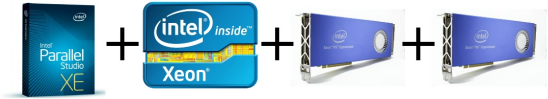
- Left: Gigabit Ethernet bridging and TCP/IP virtualization
- Right: InfiniBand on Xeon Phi with the Coprocessor Communication Link (CCL) technology
- Details: `http://xeonphi.com/papers/p2p`

 © Colfax International, 2013–2016

Colfax's Case Studies

© Colfax International, 2013–2016

Astrophysical Code HEATCODE: an Offload Story

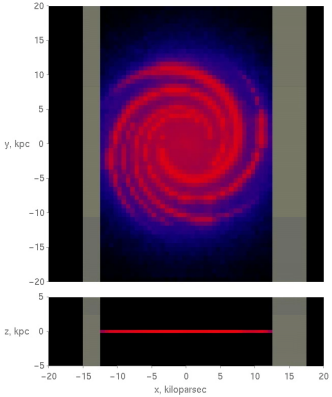


Porting to Intel® Xeon Phi™ coprocessors

- We ported Frankie code using explicit offload model
- Same code & optimization methods for Xeon Phi™
- Simultaneous calculations on CPU and coprocessors with automatic load balancing was easy to implement
- With two Intel® Xeon Phi™ coprocessors, performance for high-res calculations is 3.2x better than with two Intel® Xeon® E5 processors alone.
- RESULT: estimated target project calculation time is now 2 weeks (down from 6+ years)

Goal achieved!

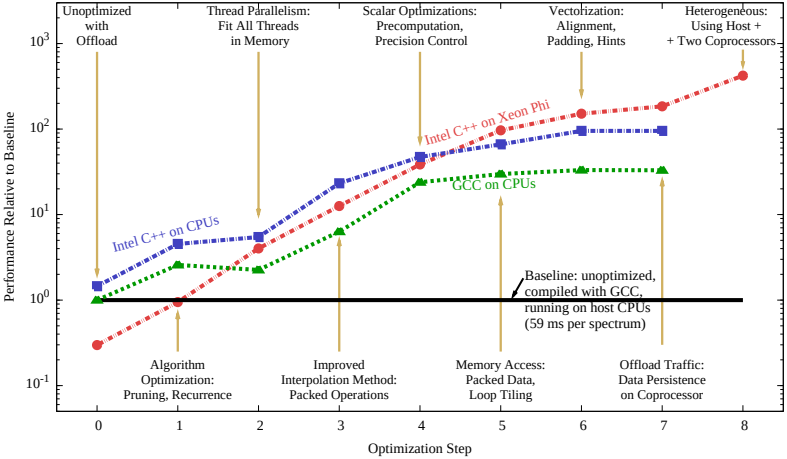
Transient Emission of Cosmic Dust Grains
in the Milky Way Galaxy,
Simulation with Frankie Code



http://xeonphi.com/papers/heatcode

© Colfax International, 2013–2016

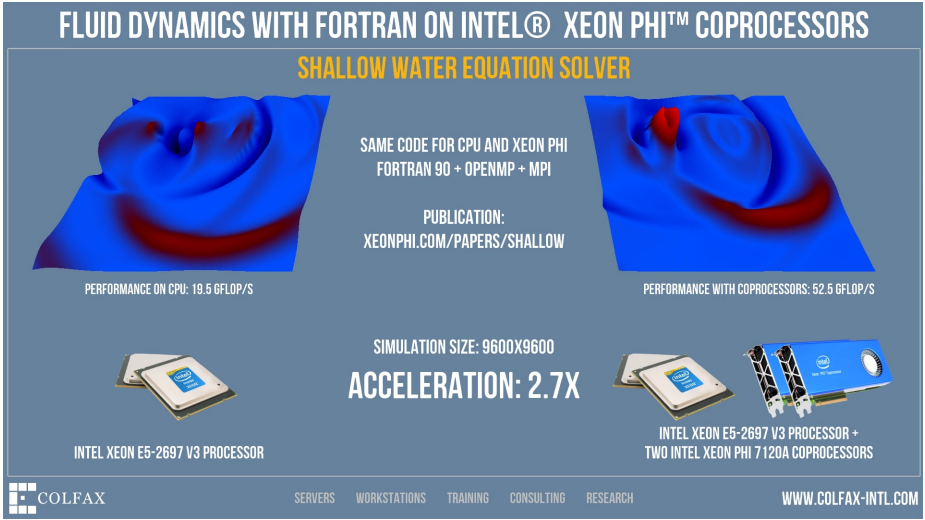
Astrophysical Code HEATCODE: an Offload Story



http://xeonphi.com/papers/heatcode

 © Colfax International, 2013–2016

Computational Fluid Dynamics: Legacy Code

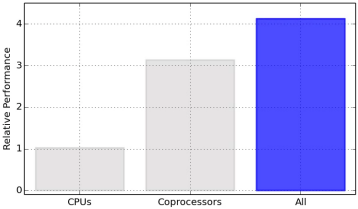


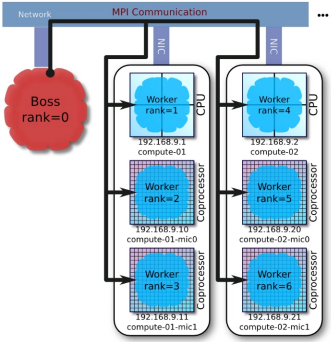
http://xeonphi.com/papers/shallow

© Colfax International, 2013–2016

Asian Option Pricing: Heterogeneous Clustering

Heterogeneous Clustering with Homogeneous Code:
Asian Option Pricing





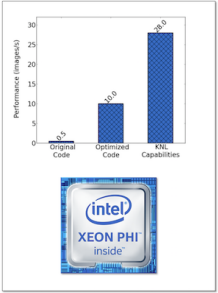
COLFAX
Customized Solutions

http://xeonphi.com/papers/heterogeneous

© Colfax International, 2013–2016

Machine Learning: Optimized Middleware

INTEL® XEON PHI™ PROCESSORS — MACHINE LEARNING



NEURALTALK2 — OPEN SOURCE IMAGE TAGGING CODE (KARPATHY & FEI-FEI, STANFORD)

COLFAX Customized Solutions

http://colfaxresearch.com/isc16-neuraltalk

© Colfax International, 2013–2016

§3. Hands-On Part: Guided Tour

© Colfax International, 2013–2016

Linux Environment on Intel Xeon Phi Coprocessors

```
vega@lyra% lspci | grep -i "co-processor"
06:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor 7120 series (rev 20)
82:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor 7120 series (rev 20)
vega@lyra% sudo service mpss status
mpss is running
vega@lyra% cat /etc/hosts | grep mic
172.31.1.1  lyra-mic0 mic0
172.31.2.1  lyra-mic1 mic1
vega@lyra% ssh mic0

vega@mic0% cat /proc/cpuinfo | grep proc | tail -n 3
processor: 241
processor: 242
processor: 243
vega@mic0% ls /
amplxe   dev    home   lib64    oldroot  proc   sbin     sys    usr
bin      etc    lib    linuxrc  opt      root   sep3.10  tmp    var
```

 © Colfax International, 2013–2016

Intel Manycore Platform Software Stack

- `micinfo` – system information
- `micsmc` – monitor and modify the physical paramaters: temperature, power modes, core utilization, etc.
- `micctrl` – configure the Intel Xeon Phi coprocessor operating system
- `miccheck` – verify the Intel Xeon Phi coprocessor configuration
- `micrasd` – log of hardware errors reported by Intel Xeon Phi coprocessors
- `micflash` – flash memory agent

Monitoring MIC activity with `micsmc` (an MPSS tool)

© Colfax International, 2013–2016

Software Necessary to Build Xeon Phi Applications:

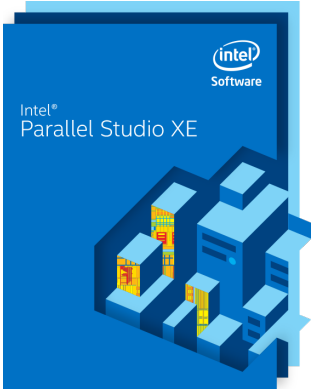
Compilers : Intel C Compiler, Intel C++ Compiler, and Intel Fortran Compiler — mandatory

Optimization tools : Intel VTune Amplifier XE and Intel Trace Analyzer and Collector (ITAC) — highly recommended

Mathematics support : Intel Math Kernel Library (MKL) — highly recommended

Cluster Development : Intel MPI — industry standard parallel framework

Development : Intel Inspector XE, Intel Advisor XE — optional



All-in-One bundle,
Intel Parallel Studio XE

 © Colfax International, 2013–2016

Intel Compilers + Intel Xeon Processor

"Hello World" application:

```
#include <cstdio>
#include <unistd.h>
int main(){
    printf("Hello world! I have %ld logical processors.\n",
    sysconf(_SC_NPROCESSORS_ONLN ));
}
```

Compile and run on host CPU:

```
vega@lyra% icpc hello.cc -xhost
vega@lyra% ./a.out
Hello world! I have 48 logical processors.
vega@lyra%
```

 © Colfax International, 2013–2016

Native Execution on an Intel Xeon Phi Coprocessor (KNC)

Compile and run the same code on the coprocessor in the native mode:

```
vega@lyra% icpc hello.cc -mmic # Cross-compile
vega@lyra% scp a.out mic0:~/ # Put executable on coprocessor
a.out 100% 10KB 10.4KB/s 00:00
vega@lyra% ssh mic0 # Log in to coprocessor
vega@mic0% pwd
/home/lyra
vega@mic0% ls
a.out
vega@mic0% ./a.out # Launch application
Hello world! I have 244 logical processors.
vega@mic0%
```

- Use `-mmic` to produce executable for MIC architecture
- Must transfer executable to coprocessor (or NFS-share) and run from shell
- Native MPI applications work the same way (need Intel MPI library)

 © Colfax International, 2013–2016

Compiling for an Intel Xeon Phi Processor (KNL)

"Hello World" application:

```
#include <cstdio>
#include <unistd.h>
int main(){
    printf("Hello world! I have %ld logical processors.\n",
    sysconf(_SC_NPROCESSORS_ONLN ));
}
```

Compile and run on host CPU:

```
vega@lyra% icpc hello.cc -xMIC-AVX512
vega@lyra% ./a.out
Hello world! I have 256 logical processors.
vega@lyra%
```

From Intel compilers 17.0 on, `-xMIC-AVX512` is not necessary when compiling on KNL.

 © Colfax International, 2013–2016

Review and What's Next

- Intel Xeon and Intel Xeon Phi – parallel processors
- Xeon Phi (MIC architecture) – specialized for highly parallel workloads without complex memory access
- Coprocessor – either offload device or an additional compute node
- Native+offload programming allow for a range of design options

Next session: details of Intel Xeon Phi processor and coprocessor programming.

 © Colfax International, 2013–2016