



Programming and Optimization for Intel[®] Architecture

The Hands-On Workshop (HOW) Series

Colfax International — @colfaxintl


July 2016 , Rev. 03a

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.

Course Roadmap

- HOW to Program Intel Architecture
 - ▶ 01. Parallelism, specialization, guided tour – July 25
 - ▶ 02. Programming Intel Xeon Phi (KNC, KNL) – July 26
- HOW to Express Parallelism
 - ▶ 03. Automatic vectorization – July 27
 - ▶ 04. Multi-threading with OpenMP – July 28
- HOW to Get Performance
 - ▶ 05. Comprehensive demo – July 29
 - ▶ 06. Scalar & vectorization tuning – August 1
 - ▶ 07. Multi-threading I – August 2
 - ▶ 08. Multi-threading II - August 3
 - ▶ 09. Memory traffic - August 4
- HOW to Scale
 - ▶ 10. Distributed Computing: MPI – August 5

July 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						
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			
 — 8:00am UTC Lecture+remote access						

HOW Online

Course page: colfaxresearch.com/how-16-07

- 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

Get Your Questions Answered

Chat (current):

colfaxresearch.com/how-16-07



Forums (technical):

colfaxresearch.com/discussion

Log In/Register

COLFAX RESEARCH

CONTRIBUTING TO INNOVATIONS IN COMPUTING

/ READ WATCH LEARN **FORUMS** CONNECT JOIN

Join the Conversation

Welcome to Colfax Research forums, an online community for you to engage with HPC experts, software architects, developers, computational researchers, scientists, students and more—so you can acquire new knowledge, share ideas, and build new relationships.

Tap our experts and your peers to help meet the challenge of optimizing applications on modern hardware. This is the place to browse or post questions (and get answers) related to computational science, parallel programming and code modernization on Intel® Architecture.

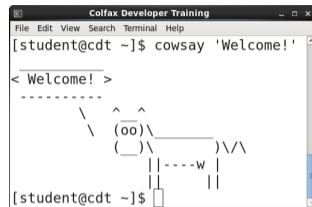
Welcome aboard. Post questions today!

Email (organizational):

training@colfax-intl.com

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 Developer Training
File Edit View Search Terminal Help
[student@cdt ~]$ cowsay 'Welcome!'
< Welcome! >
-----
      \   ^__^
         (oo)\_____)
            (_____)
                ||----w |
                ||     ||

[student@cdt ~]$
```

[Learn More](#)

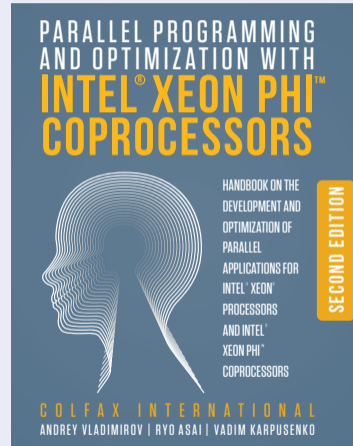
Textbook

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

Parallel Programming and Optimization with Intel® Xeon Phi™ Coproprocessors

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

COLFAX RESEARCH

CONTRIBUTING TO INNOVATIONS IN COMPUTING

[Log In/Out](#) or [Register](#)

READ
WATCH
LEARN
CONNECT
JOIN

To search, type and hit enter

Popular

The Hands-On Tutorials (HOT) webinars: details an efficient programming for Intel architecture

The Hands-On Workshop (HOW) Series

Introduction to Intel DAAL, Part 1: Polynomial Regression with Batch Mode Computation

Parallel Programming Book

Introduction to parallel programming, deep discussion of optimization techniques, exercises.

© 2015, Colfax International. 508 pages.

Research and Educational Publications



Introduction to Intel DAAL, Part 1: Polynomial Regression with Batch Mode Computation



Optimization Techniques for the Intel MIC Architecture, Part 3 of 3: Strip-Mining for Vectorization



Software Developer's Introduction to the HGST Ultrastar Archive HaaS SMR Drives



Optimization Techniques for the Intel MIC Architecture, Part 2 of 3: Strip-Mining for Vectorization



Optimization Techniques for the Intel MIC Architecture, Part 1 of 3: Multi-Threading and Parallel Reduction



Performance to Power and Performance to Cost Ratios with Intel Xeon Phi Coprocessors (and why TX Acceleration May be Enough)

Featured Video

generated Additional Reading

has Research related to vectorization like a sharing code



<http://colfaxresearch.com/?p=704>

Events

Presentations

Courses

Consulting

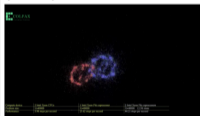


Colfax offers consulting services for enterprises, research help you to:

- Optimize your existing application to take advantage of parallelism, from vectors to cores to clusters and beyond
- Future-proof your application for upcoming innovations
- Accelerate your application using coprocessor technologies
- Investigate the potential system configurations that satisfy your cost, power performance requirements.
- Take a clean sheet to develop a novel approach to solve your computing problem

All Video Courses - COP 901 - Chapter 2 - Episode 1.1

Episode 2.1 - Purpose of the MIC architecture



For more information, visit <http://colfaxresearch.com/?p=704>

© 2015 Colfax International. All rights reserved.

Software Developer's Introduction to the HGST Ultrastar Archive HaaS SMR Drives



In this paper, we will discuss the HGST Ultrastar Archive HaaS SMR drives, software developer's introduction to the HGST Ultrastar Archive HaaS SMR drives, and how to use the HGST Ultrastar Archive HaaS SMR drives in your applications.

The HGST Ultrastar Archive HaaS SMR drives are designed to provide high performance and low cost storage solutions for your applications. The HGST Ultrastar Archive HaaS SMR drives are available in 3.5" and 2.5" form factors.

For more information, visit <http://colfaxresearch.com/?p=704>

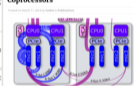
Fluid Dynamics with Fortran on Intel Xeon Phi coprocessors



This document describes a Fortran program that runs on Intel Xeon Phi coprocessors. The program simulates fluid dynamics on a grid of points. The results are shown in the image above.

For more information, visit <http://colfaxresearch.com/?p=704>

Configuration and Benchmarks of Peer-to-Peer Communication over Gigabit Ethernet and InfiniBand in a Cluster with Intel Xeon Phi Coprocessors



This document describes the configuration and benchmarks of peer-to-peer communication over Gigabit Ethernet and InfiniBand in a cluster with Intel Xeon Phi coprocessors. The results are shown in the image above.

For more information, visit <http://colfaxresearch.com/?p=704>

© 2015 Colfax International. All rights reserved.

Interview with James Reinders: future of Intel MIC architecture, parallel programming, education



In this interview, James Reinders discusses the future of Intel MIC architecture, parallel programming, and education. He also discusses the challenges of parallel programming and the importance of education in this field.

For more information, visit <http://colfaxresearch.com/?p=704>

<http://colfaxresearch.com/>

§2. Intel Architecture

Computing Platforms

Computing Platforms

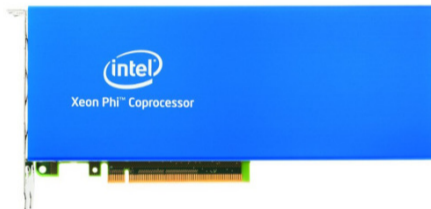
Intel Xeon Processor



Current: Broadwell
Upcoming: Skylake

Multi-Core Architecture

Intel Xeon Phi Coprocessor, 1st generation



Knights Corner (KNC)

Intel Xeon Phi Processor, 2nd generation*



* socket and coprocessor versions

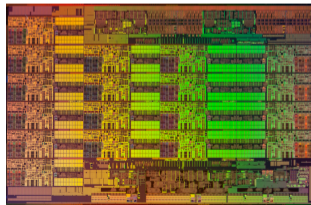
Knights Landing (KNL)

Intel Many Integrated Core (MIC) Architecture

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

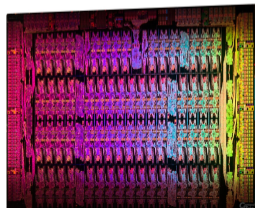
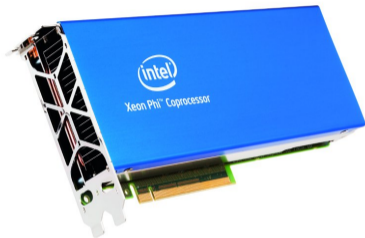


* 2-way Intel Xeon processor, Skylake architecture, top-of-the-line (e.g., E5-2699 V4)

Intel Xeon Phi Processors (1st Gen)

Specialized platform for demanding computing applications.

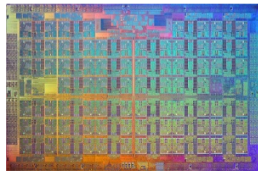
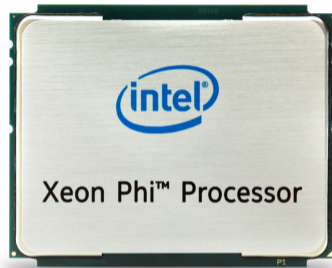
- PCIe add-in card
- ~ 1.2 TFLOP/s in DP
- ~ 2.4 TFLOP/s in SP
- Up to 16 GiB GDDR5 RAM
- ~ 176 GB/s bandwidth
- Heterogeneous clustering
- Runs special Linux distribution



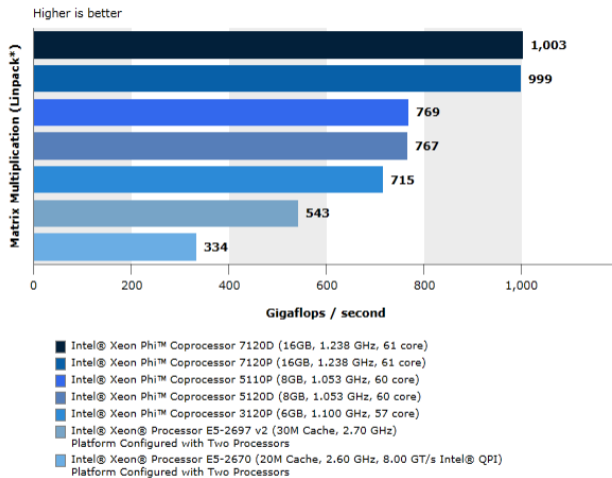
Intel Xeon Phi Processors (2nd Gen)

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**

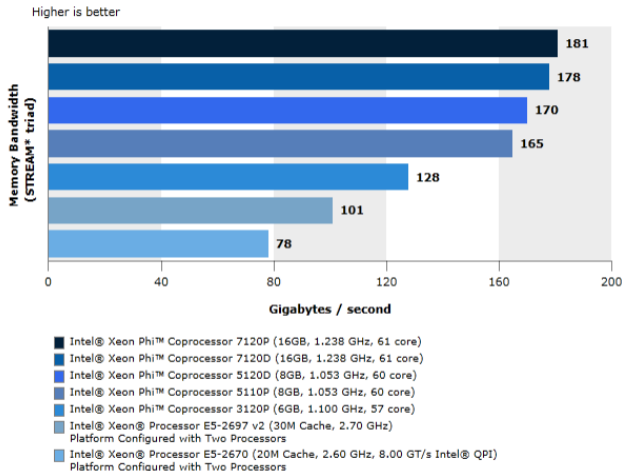


Intel Xeon Phi Coprocessors: HPC LINPACK Benchmark



Source: “Intel Xeon Phi Coprocessor LINPACK* and STREAM* Performance”

Intel Xeon Phi Coprocessors: STREAM Benchmark

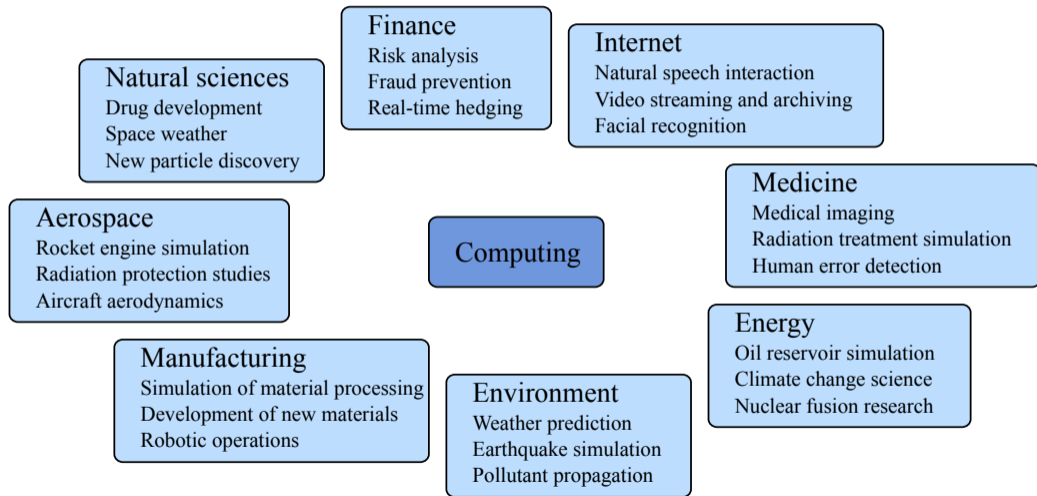


Source: “Intel Xeon Phi Coprocessor LINPACK* and STREAM* Performance”

It Is All About Performance

Computing Applications

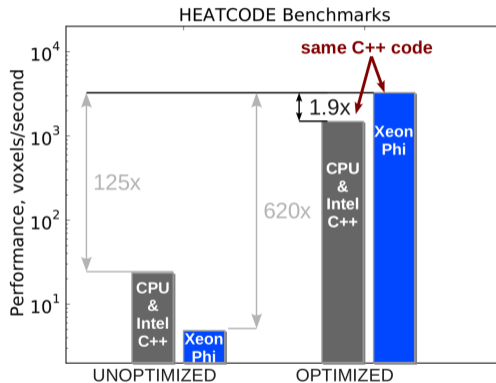
Just some examples



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

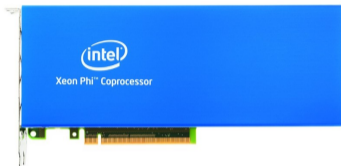


See [this case study](#)

Optimization Areas

- 1 **Scalar optimization** (compiler-friendly practices)
- 2 **Vectorization** (must use 16- or 8-wide vectors)
- 3 **Multi-threading** (must scale to 100+ threads)
- 4 **Memory access** (streaming access or tiling)
- 5 **Communication** (offload, MPI traffic control)

Coprocessor vs Processor Performance



One Intel Xeon Phi 7120P
coprocessor

vs.



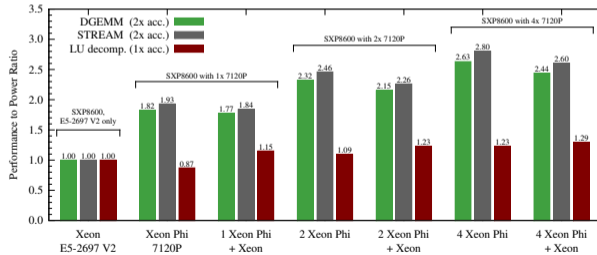
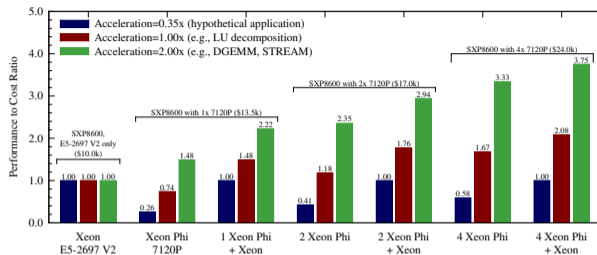
Two Intel Xeon E5-2697 v2
CPUs

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

See also [“Intel Xeon Product Family: Performance Brief”](#)

What is Your Performance Metric?

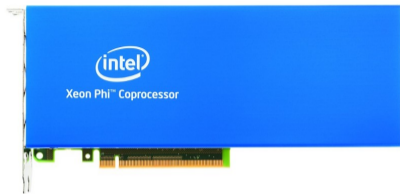
- Performance per System
- Performance per Watt
- Performance/Cost Ratio



See [this paper](#) for details

Common Architecture

Bird's Eye View

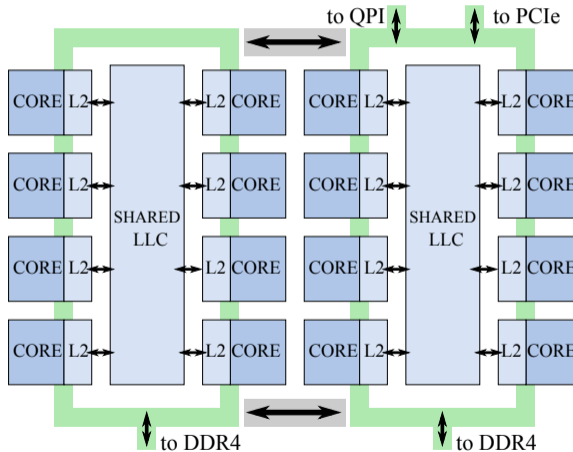


- C/C++/Fortran; OpenMP/MPI
- Standard Linux OS
- ≤ 3 TiB of DDR4 RAM
- ≤ 22 cores/chip ≈ 3 GHz
- 2 hyper-threads per core
- 256-bit AVX vectors

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

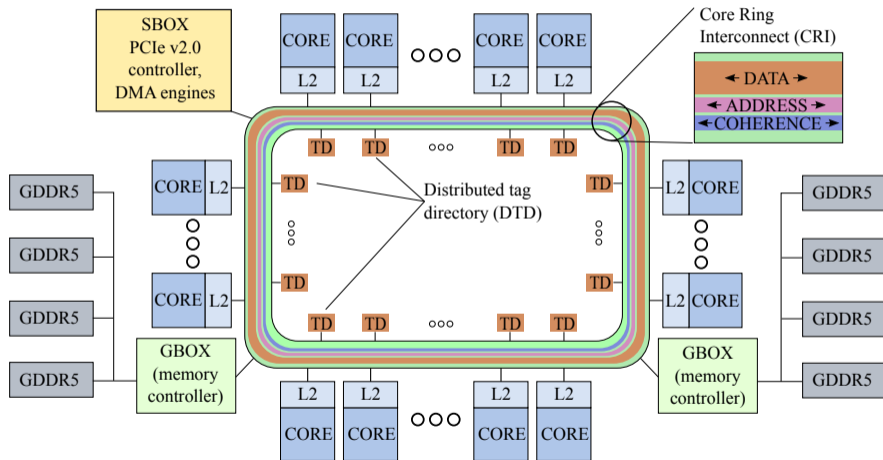
Intel Xeon CPU: Die Organization

Likes data locality, but large LLC is forgiving.



KNC Die Organization

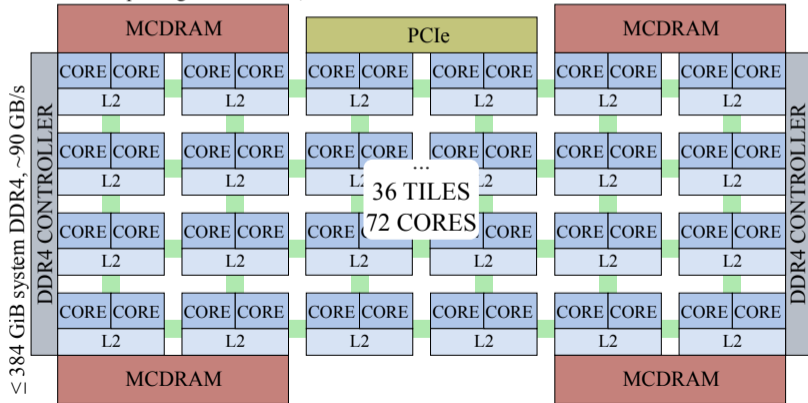
In a ring bus with distributed cache, data access locality is key.



KNL Die Organization

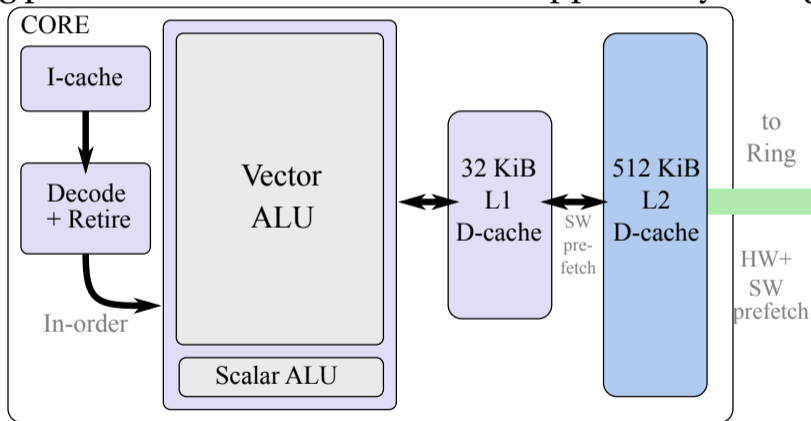
- Mesh interconnect relaxes data locality requirement [somewhat]
- All-to-all, quadrant or sub-numa domain communication in mesh

≤ 16 GiB on-package MCDRAM, ~ 400 GB/s



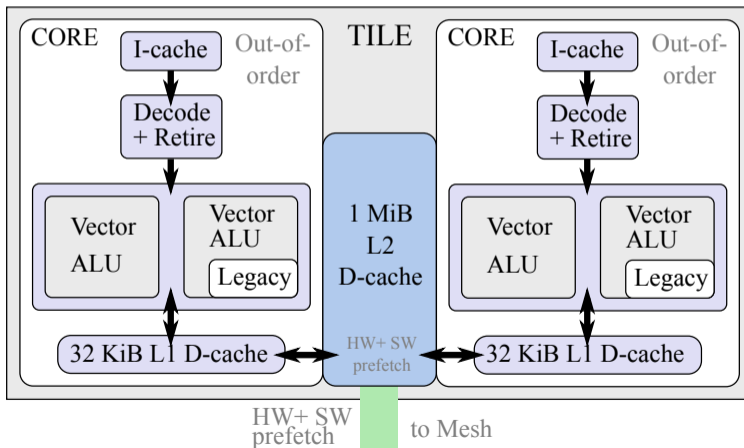
KNC Cores

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

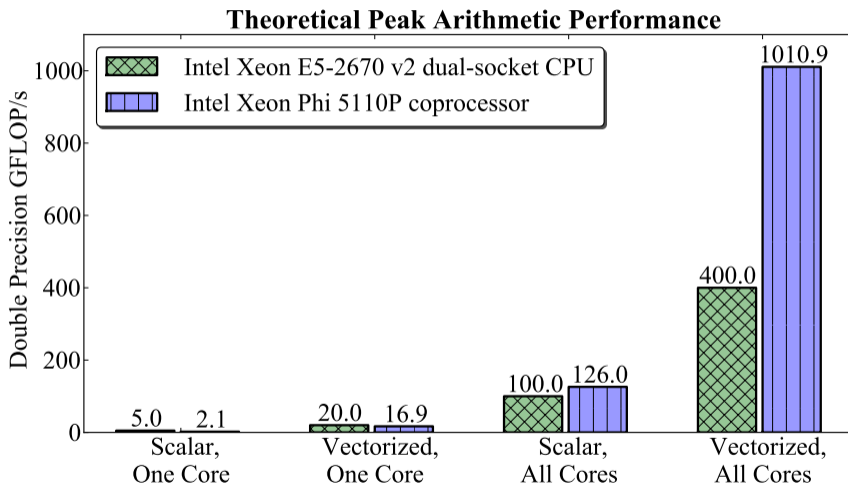


KNL Cores

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

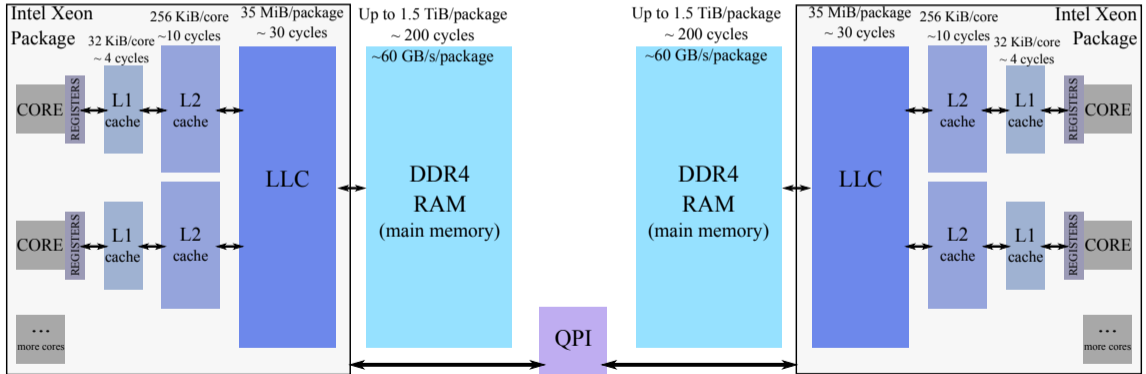


Task and Data Parallelism



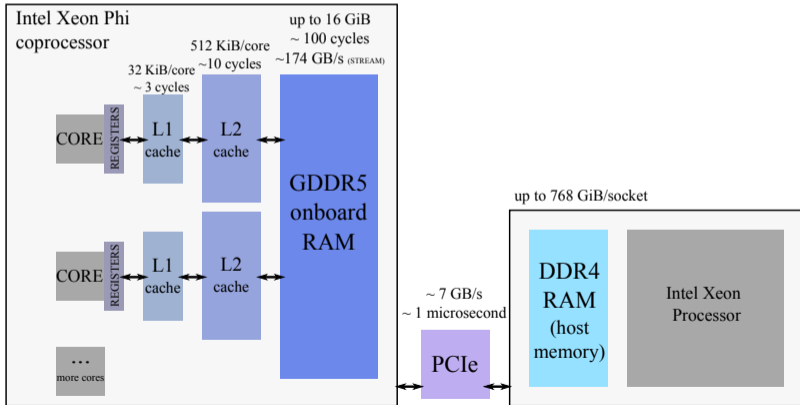
Intel Xeon CPU: Memory Organization

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



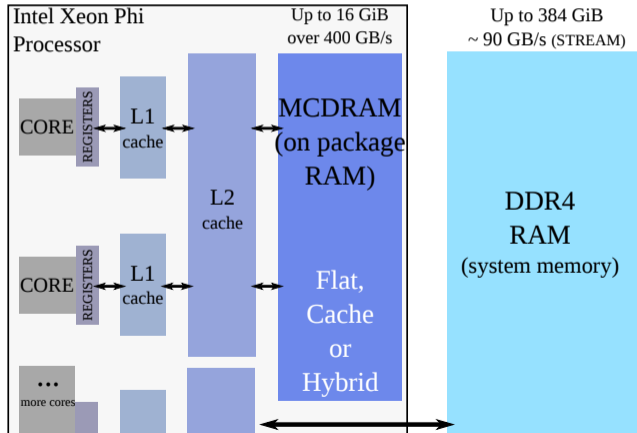
KNC Memory Organization

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

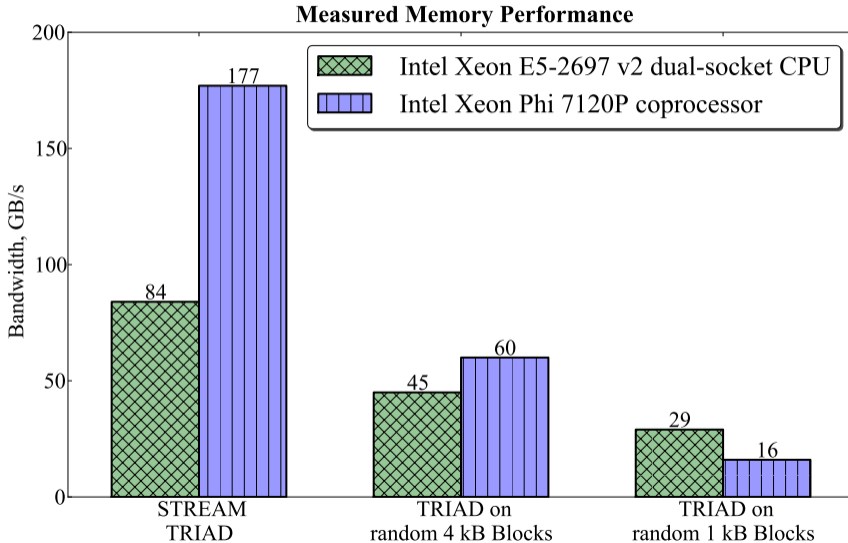


KNL Memory Organization

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



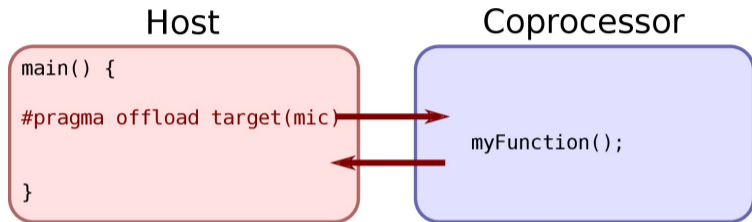
Memory Access Pattern



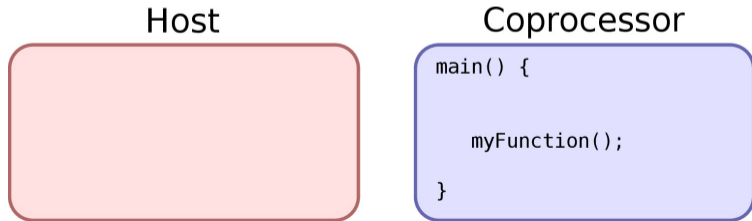
Programming Coprocessors

Offload and Native Models

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



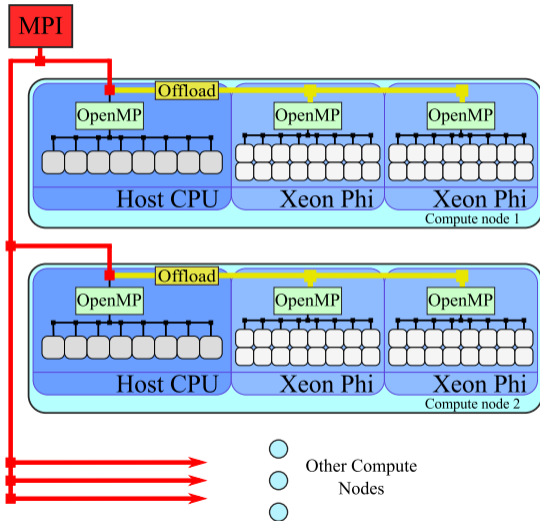
- Native model (standalone application/MPI process):



Heterogeneous Distributed Computing with Xeon Phi

Option 1: MPI+OpenMP with Offload.

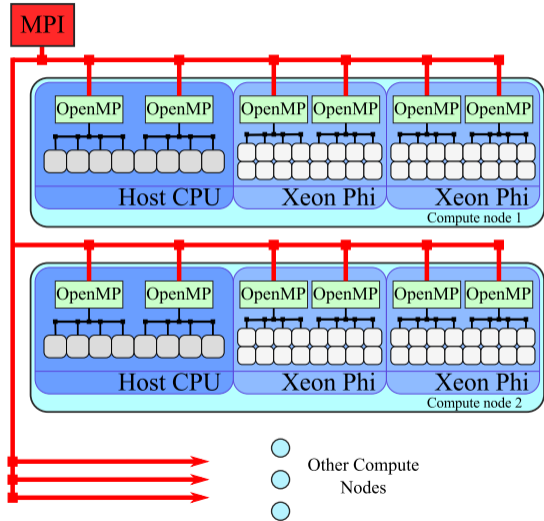
- MPI processes are multi-threaded with OpenMP.
- MPI runs only on CPUs.
- MPI processes offload to coprocessor(s).
- OpenMP in offload regions.



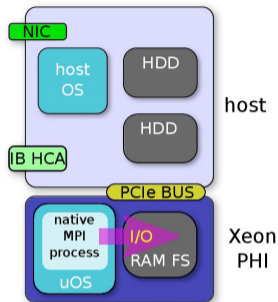
Heterogeneous Distributed Computing with Xeon Phi

Option 2: Symmetric hybrid MPI+OpenMP.

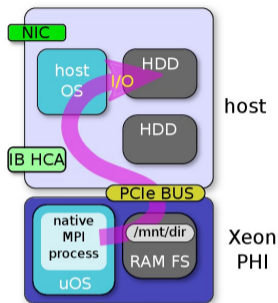
- MPI processes on hosts
- Native MPI processes on the coprocessor.
- Multi-threading with OpenMP.



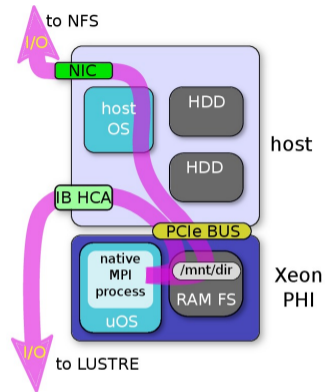
Working with Files on Coprocessors



RAM Filesystem



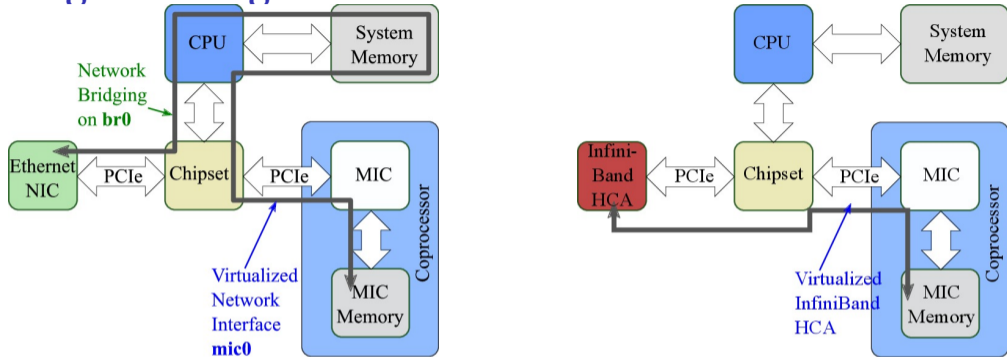
VirtIO Transfer



Network Storage

Details: <http://xeonphi.com/papers/io>

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's Case Studies

Astrophysical Code HEATCODE: an Offload Story

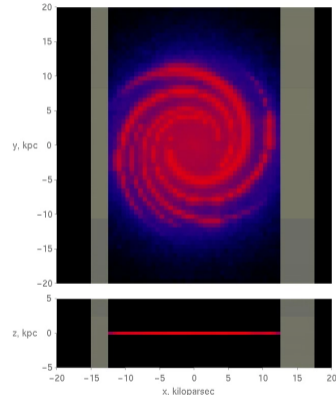


Porting to Intel® Xeon Phi™ coprocessors

- 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)

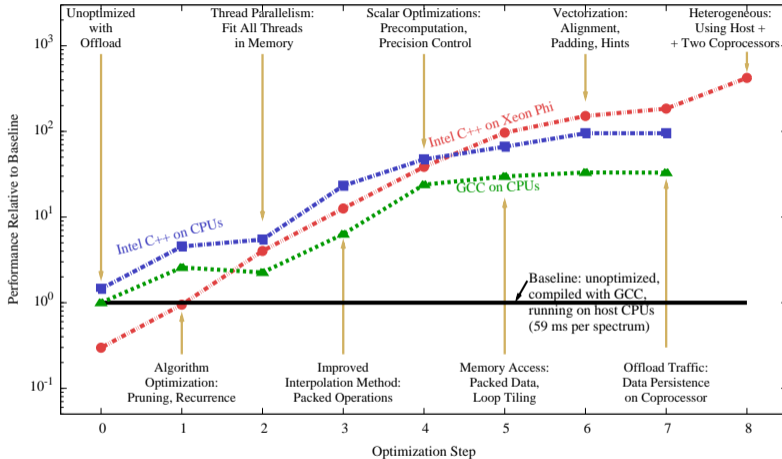
Goal achieved!

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



<http://xeonphi.com/papers/heatcode>

Astrophysical Code HEATCODE: an Offload Story

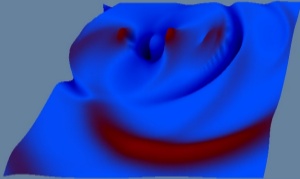


<http://xeonphi.com/papers/heatcode>

Computational Fluid Dynamics: Legacy Code

FLUID DYNAMICS WITH FORTRAN ON INTEL® XEON PHI™ COPROCESSORS

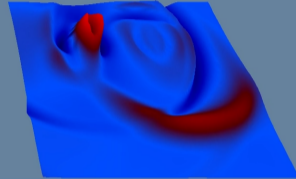
SHALLOW WATER EQUATION SOLVER



PERFORMANCE ON CPU: 19.5 GFLOP/S

SAME CODE FOR CPU AND XEON PHI
FORTRAN 90 + OPENMP + MPI

PUBLICATION:
XEONPHI.COM/PAPERS/SHALLOW



PERFORMANCE WITH COPROCESSORS: 52.5 GFLOP/S

SIMULATION SIZE: 9600X9600

ACCELERATION: 2.7X



INTEL XEON E5-2697 V3 PROCESSOR



INTEL XEON E5-2697 V3 PROCESSOR +
TWO INTEL XEON PHI 7120A COPROCESSORS



SERVERS WORKSTATIONS TRAINING CONSULTING RESEARCH

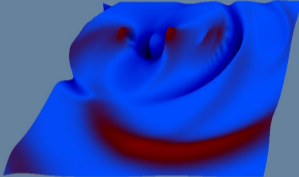
WWW.COLFAX-INTL.COM

<http://xeonphi.com/papers/shallow>

Computational Fluid Dynamics: Legacy Code

FLUID DYNAMICS WITH FORTRAN ON INTEL® XEON PHI™ COPROCESSORS

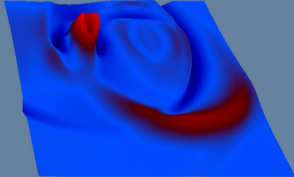
SHALLOW WATER EQUATION SOLVER



PERFORMANCE ON CPU: 19.5 GFLOP/S

SAME CODE FOR CPU AND XEON PHI
FORTRAN 90 + OPENMP + MPI


PUBLICATION:
XEONPHI.COM/PAPERS/SHALLOW



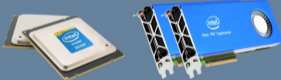
PERFORMANCE WITH COPROCESSORS: 52.5 GFLOP/S

SIMULATION SIZE: 9600X9600

ACCELERATION: 2.7X



INTEL XEON E5-2697 V3 PROCESSOR



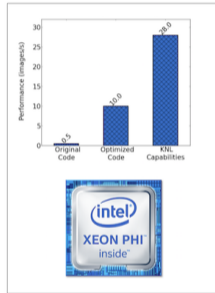
INTEL XEON E5-2697 V3 PROCESSOR +
TWO INTEL XEON PHI 7120A COPROCESSORS

COLFAX SERVERS WORKSTATIONS TRAINING CONSULTING RESEARCH [WWW.COLFAX-INTL.COM](http://www.colfax-intl.com)

<http://xeonphi.com/papers/shallow>

Machine Learning: Optimized Middleware

INTEL® XEON PHI™ PROCESSORS — MACHINE LEARNING



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



<http://colfaxresearch.com/isc16-neuraltalk>

§3. Hands-On Part: Guided Tour

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
```

Intel Manycore Platform Software Stack

- micinfo – system information
- micsmc – monitor and modify the physical parameters: 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)

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

Intel Compilers + Intel Xeon Processor

“Hello World” application:

```
1 #include <stdio>
2 #include <unistd.h>
3 int main(){
4     printf("Hello world! I have %ld logical processors.\n",
5         sysconf(_SC_NPROCESSORS_ONLN ));
6 }
```

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%
```

Compiling for an Intel Xeon Phi Processor (KNL)

“Hello World” application:

```
1 #include <stdio>
2 #include <unistd.h>
3 int main(){
4     printf("Hello world! I have %ld logical processors.\n",
5         sysconf(_SC_NPROCESSORS_ONLN ));
6 }
```

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%
```

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)

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 the offload model.