PROGRAMMING AND OPTIMIZATION FOR INTEL® ARCHITECTURE

The Hands-On Workshop (HOW) Series

Session 1

Colfax International — *colfaxresearch.com*

September 2016

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 – Sep 26
  - 02. Programming Intel Xeon Phi (KNC, KNL) – Sep 27
- HOW to Express Parallelism
  - 03. Automatic vectorization – Sep 28
  - 04. Multi-threading with OpenMP – Sep 29
- HOW to Get Performance
  - 05. Comprehensive demo – Sep 30
  - 06. Scalar & vectorization tuning – Oct 3
  - 07. Multi-threading: common issues – Oct 4
  - 08. Multi-threading: memory aspect – Oct 5
  - 09. Memory traffic – Oct 6
- HOW to Scale
  - 10. Distributed Computing: MPI – Oct 7

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	

October 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					

— Webinar+remote access

HOW ONLINE

Course page: colfaxresearch.com/how-16-09

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

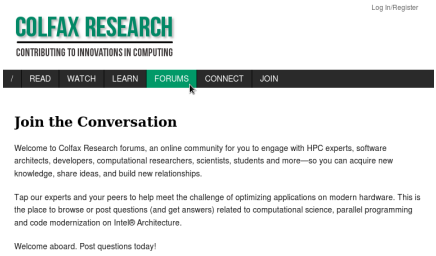
Additional resources:

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

GET YOUR QUESTIONS ANSWERED

Chat (current):
colfaxresearch.com/how-16-09

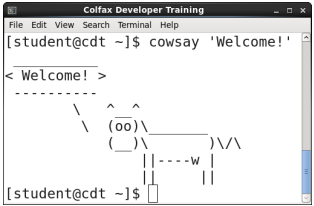
Forums (technical):
colfaxresearch.com/discussion



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

LEARN MORE

HOW SERIES: KNIGHTS LANDING

colfaxresearch.com/how-knl/

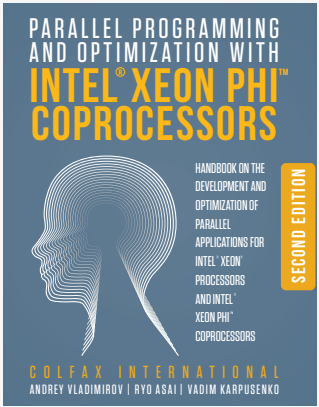
LEARN MORE

TEXTBOOK

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

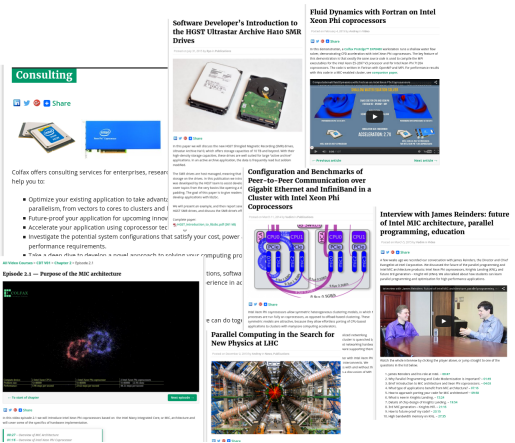
Parallel Programming
and Optimization with
Intel® Xeon Phi™
Coprocessors

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



http://xeonphi.com/book

COLFAX RESEARCH



http://colfaxresearch.com/

LEARN MORE

§2. INTEL ARCHITECTURE

COMPUTING PLATFORMS

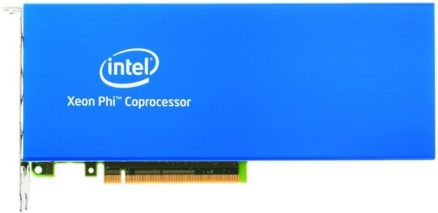
COMPUTING PLATFORMS

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

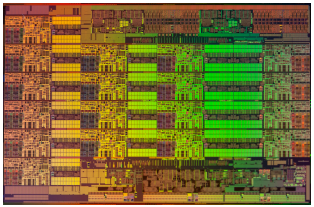
Knights Landing (KNL)

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

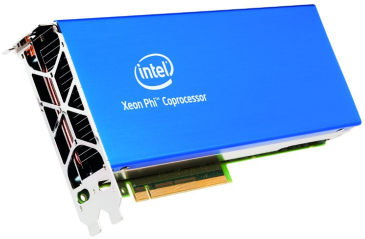


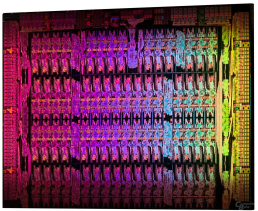
* 2-way Intel Xeon processor, Skylake architecture, top-of-the-line (e.g., E5-2699 V4)

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

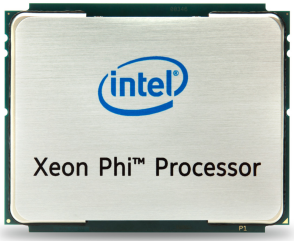




INTEL XEON PHI PROCESSORS (2ND GEN)

Specialized platform for demanding computing applications.

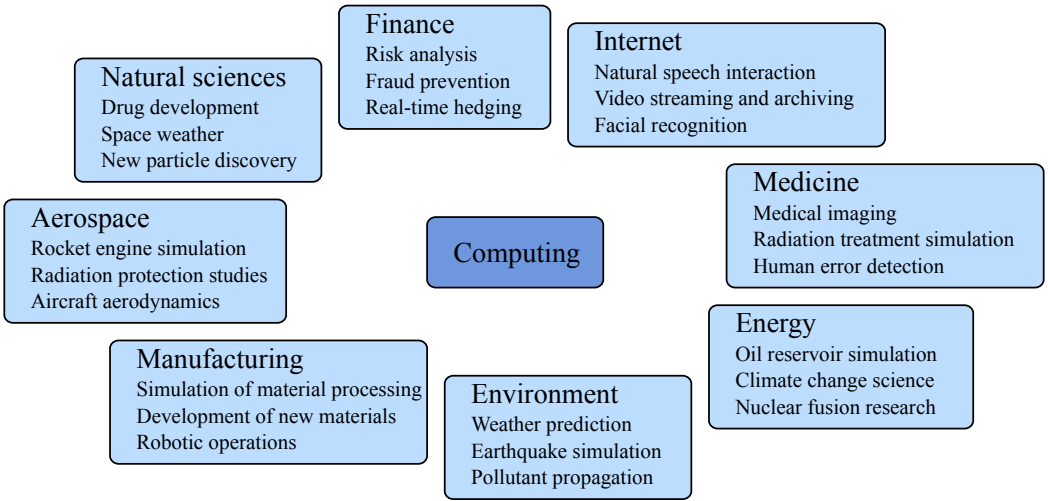
- Socket version or coprocessor
- 64-72 cores × 4 HT at 1.3-1.5 GHz
- 3+ TFLOP/s in DP (FMA)
- 6+ TFLOP/s in SP (FMA)
- ≤ 384 GiB DDR4 (> 90 GB/s)
- 16 GiB HBM (MCDRAM, > 400 GB/s)
- Binary-compatible with Xeon
- Common OS (RHEL/CentOS/Fedora/SUSE/Windows)



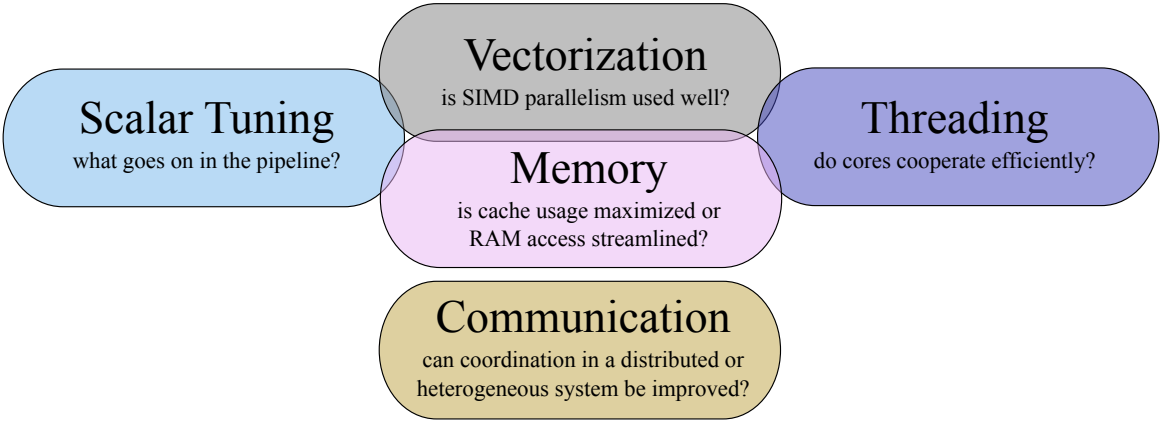
IT IS ALL ABOUT PERFORMANCE

COMPUTING APPLICATIONS

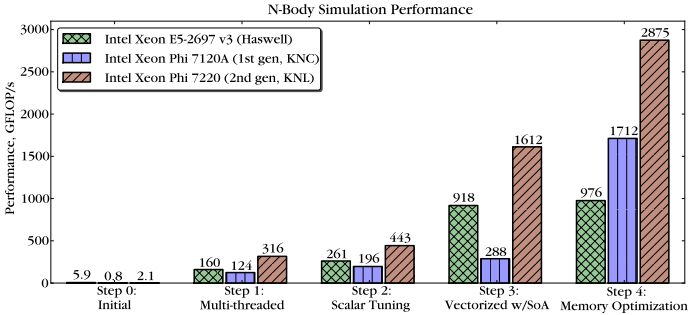
Just some examples

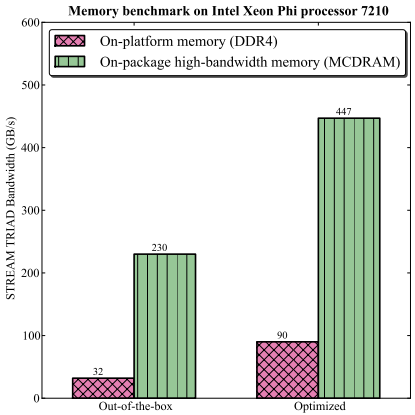


OPTIMIZATION AREAS



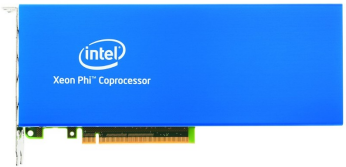
IT TAKES GOOD SOFTWARE TO UNLOCK THE PERFORMANCE!





Details on N-body simulation in Chapter 23 of this book: lotsofcores.com/KNLbook

COPROCESSOR VS PROCESSOR PERFORMANCE



vs.

One Intel Xeon Phi 7120P coprocessor

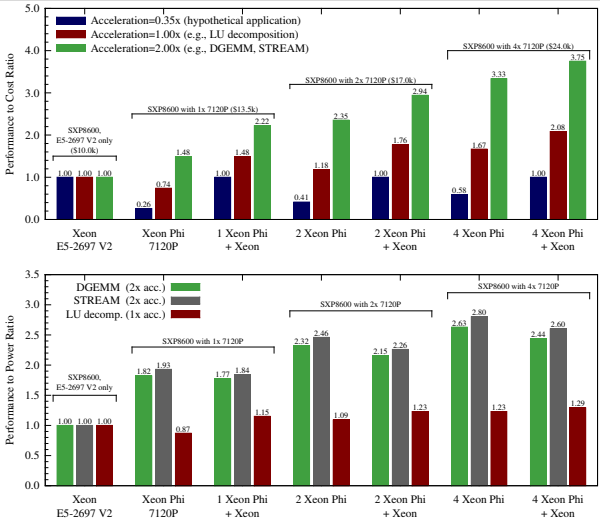
Two Intel Xeon E5-2697 v2 CPUs

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

See also “Intel Xeon Product Family: Performance Brief”

WHAT IS YOUR PERFORMANCE METRIC?

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

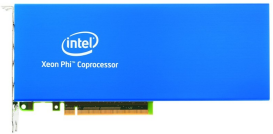


See this paper for details

COMMON ARCHITECTURE

BIRD'S EYE VIEW

- C/C++/Fortran
- Linux/Windows
- ≤3 TiB DDR4
- ≤44 cores (2-way)
- ≈3 GHz
- 2 HT/core
- 256-bit AVX

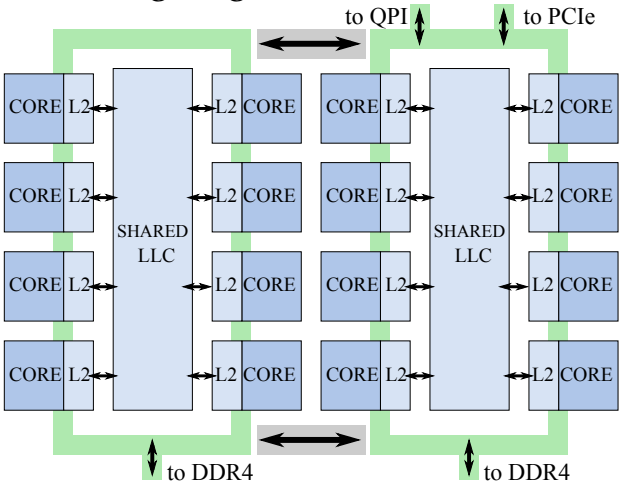


- C/C++/Fortran
- Special Linux
- 3–16 GiB GDDR
- 57-61 cores
- ≈1.2 GHz
- 4 HW THR/core
- 512-bit IMCI

- C/C++/Fortran
- Linux/Windows
- 16 GiB MCDRAM
- 64-72 cores
- 1.3-1.5 GHz
- 4 HT/core
- 512-bit AVX-512

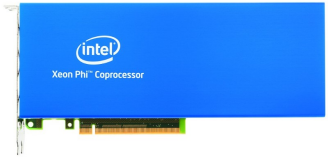
INTEL XEON CPU: DIE ORGANIZATION

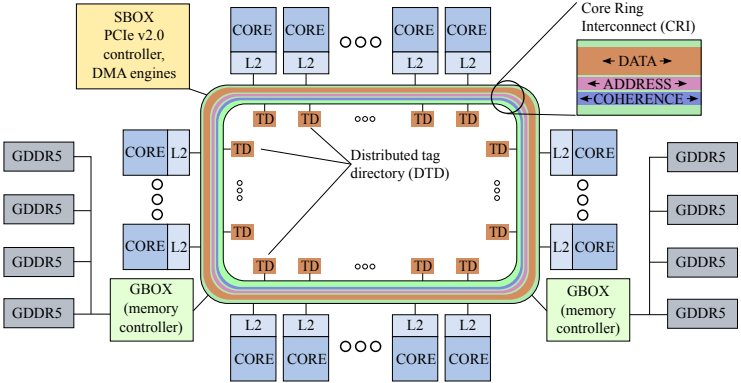
Likes data locality, but large LLC is forgiving.



KNC DIE ORGANIZATION

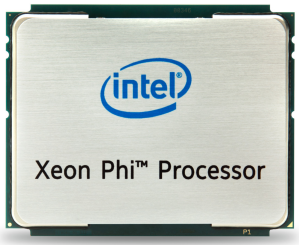
In a ring bus with distributed cache, data access locality is key.

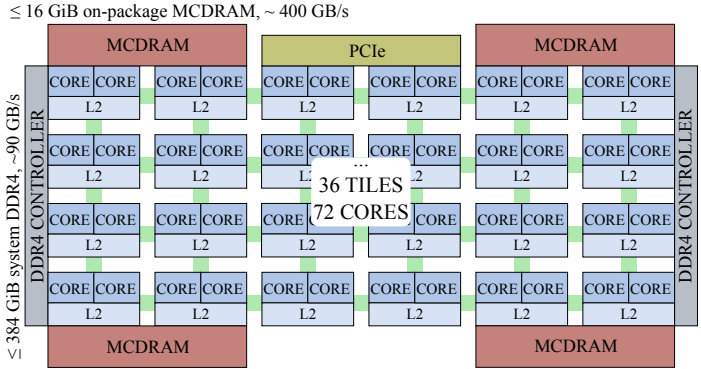




KNL DIE ORGANIZATION

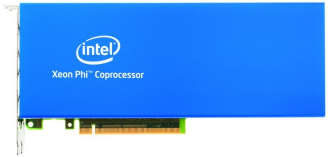
- Mesh interconnect relaxes data locality requirement [somewhat]
- All-to-all, quadrant or sub-numa domain communication in mesh

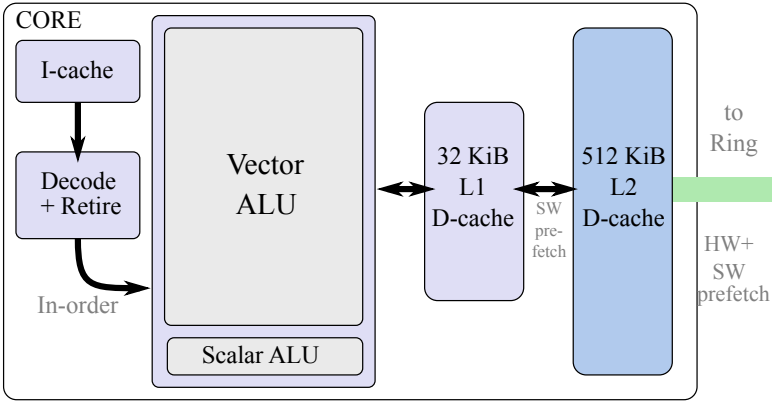




KNC CORES

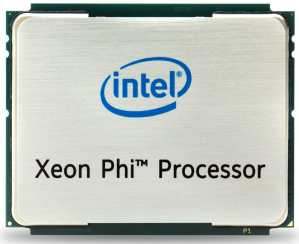
Computing power is in vector units. Scalar support only for legacy usage.

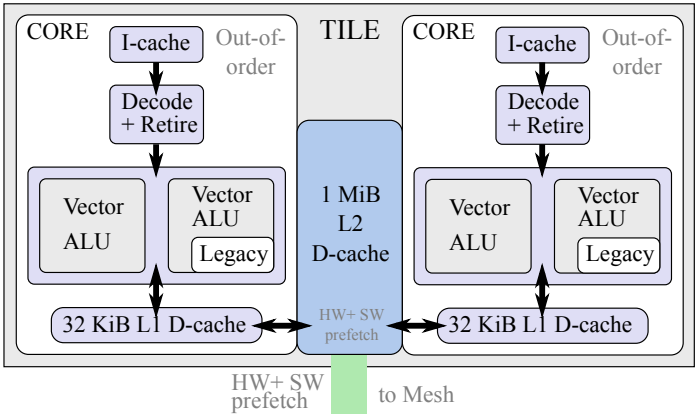




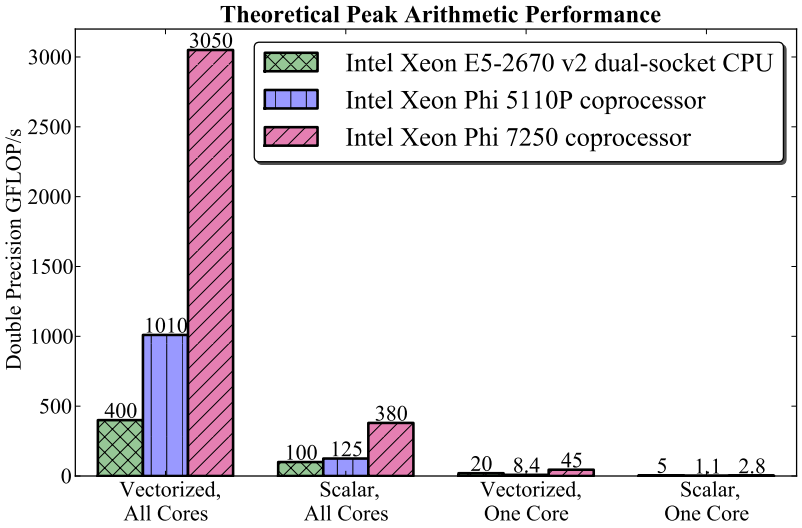
KNL CORES

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



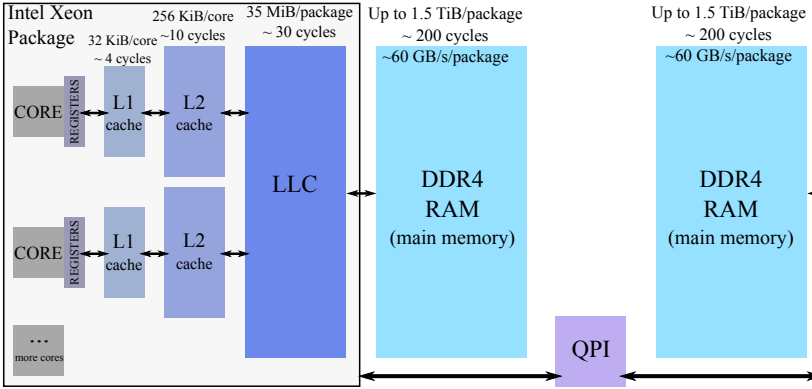


TASK AND DATA PARALLELISM



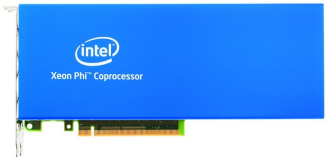
INTEL XEON CPU: MEMORY ORGANIZATION

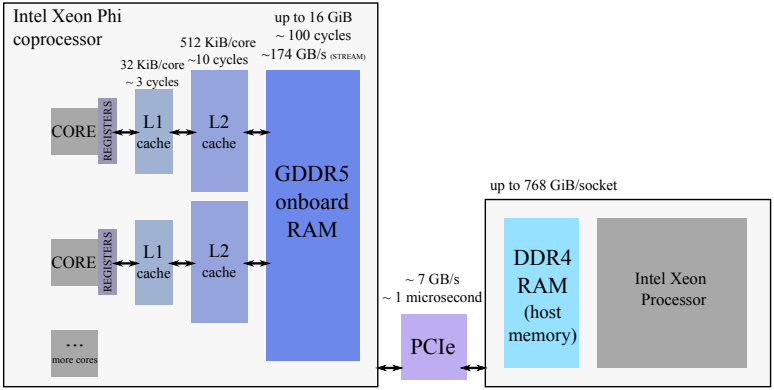
- Hierarchical cache structure
- Two-way processors have NUMA architecture



KNC MEMORY ORGANIZATION

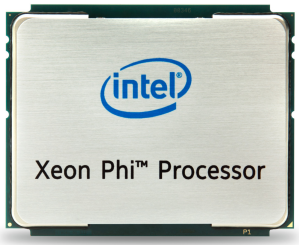
- Direct access to ≤ 16 GiB of cached GDDR5 memory on board
- No access to system DDR4, connected to host via PCIe

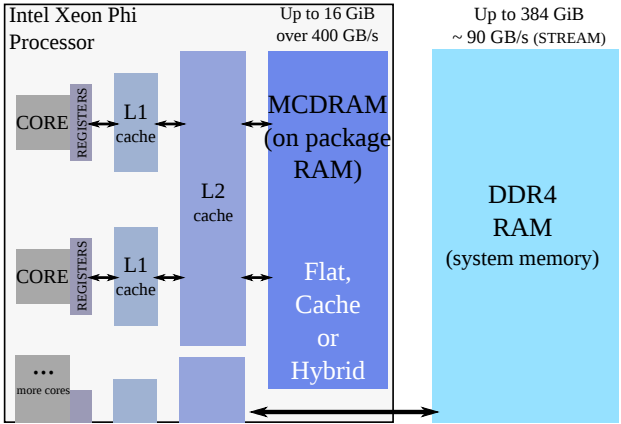




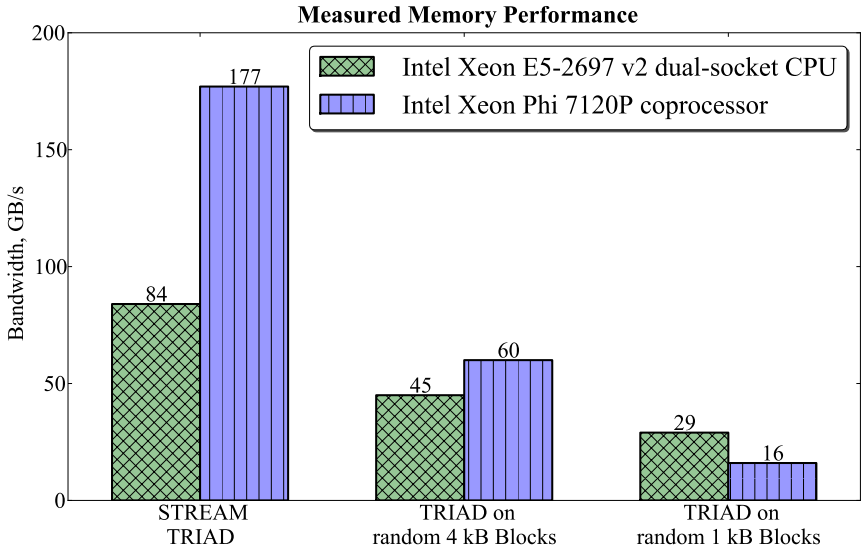
KNL MEMORY ORGANIZATION

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





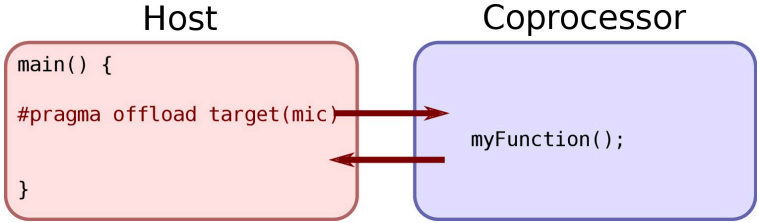
MEMORY ACCESS PATTERN



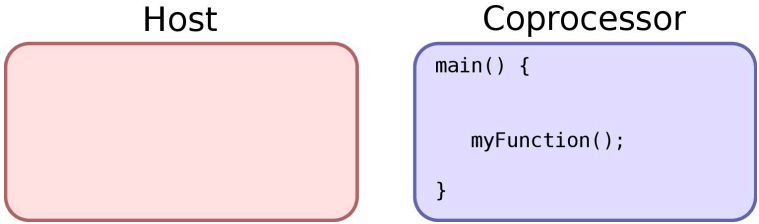
PROGRAMMING COPROCESSORS

OFFLOAD AND NATIVE MODELS

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



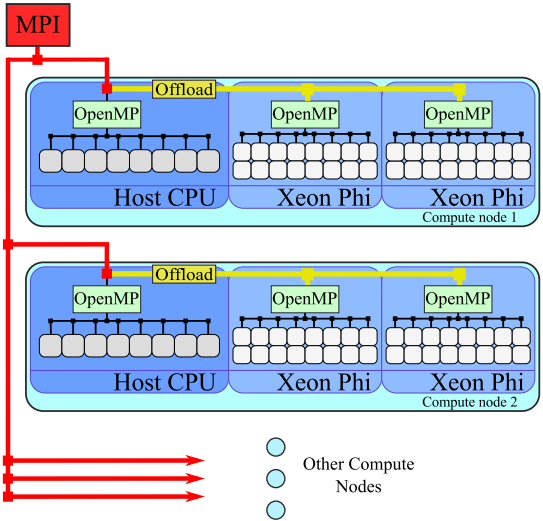
- Native model (standalone application/MPI process):



HETEROGENEOUS DISTRIBUTED COMPUTING WITH XEON PHI

Option 1: MPI+OpenMP with Offload.

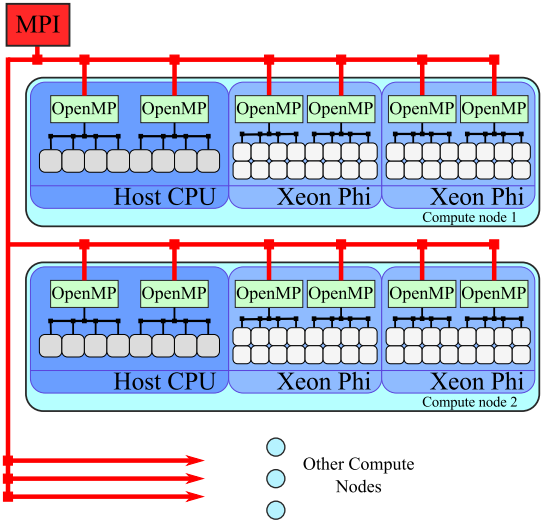
- MPI processes are multi-threaded with OpenMP.
- MPI runs only on CPUs.
- MPI processes offload to coprocessor(s).
- OpenMP in offload regions.



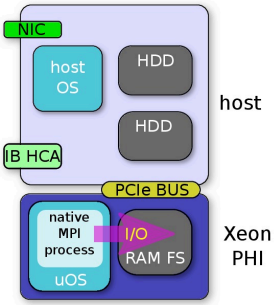
HETEROGENEOUS DISTRIBUTED COMPUTING WITH XEON PHI

Option 2: Symmetric hybrid MPI+OpenMP.

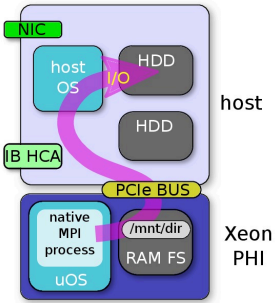
- MPI processes on hosts
- Native MPI processes on the coprocessor.
- Multi-threading with OpenMP.



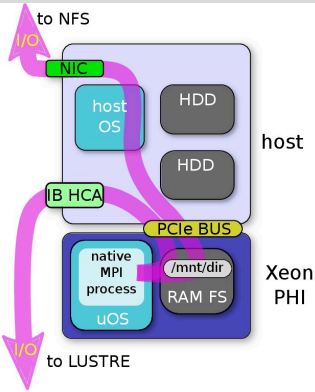
WORKING WITH FILES ON COPROCESSORS



RAM Filesystem



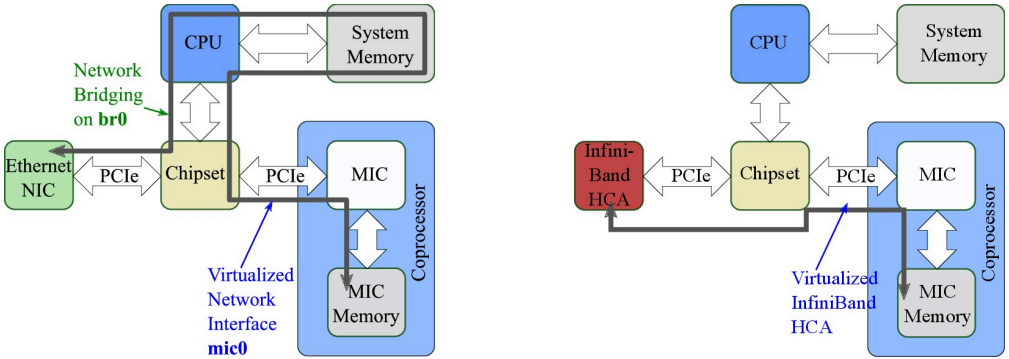
VirtIO Transfer



Network Storage

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

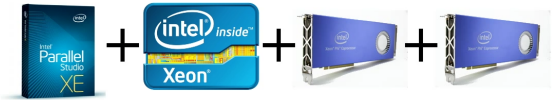
BRIDGED CONFIGURATION FOR PEER-TO-PEER COMMUNICATION



- Left: Gigabit Ethernet bridging and TCP/IP virtualization
- Right: InfiniBand + Coprocessor Communication Link (CCL)
- Details: `http://xeonphi.com/papers/p2p`

COLFAX'S CASE STUDIES

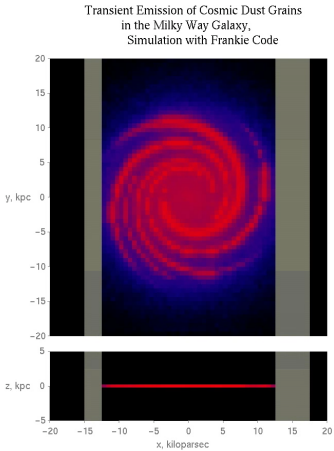
ASTROPHYSICAL CODE HEATCODE: AN OFFLOAD STORY



Porting to Intel® Xeon Phi™ coprocessors

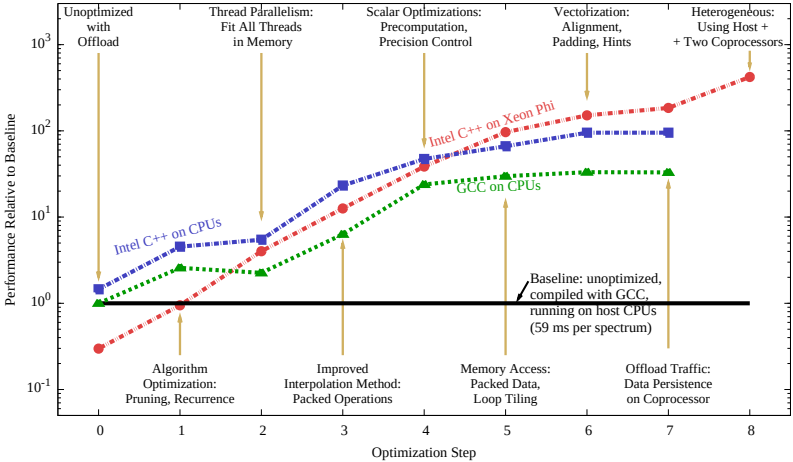
- We ported Frankie code using explicit offload model
- Same code & optimization methods for Xeon Phi™
- Simultaneous calculations on CPU and coprocessors with automatic load balancing was easy to implement
- With two Intel® Xeon Phi™ coprocessors, performance for high-res calculations is 3.2x better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

Goal achieved!



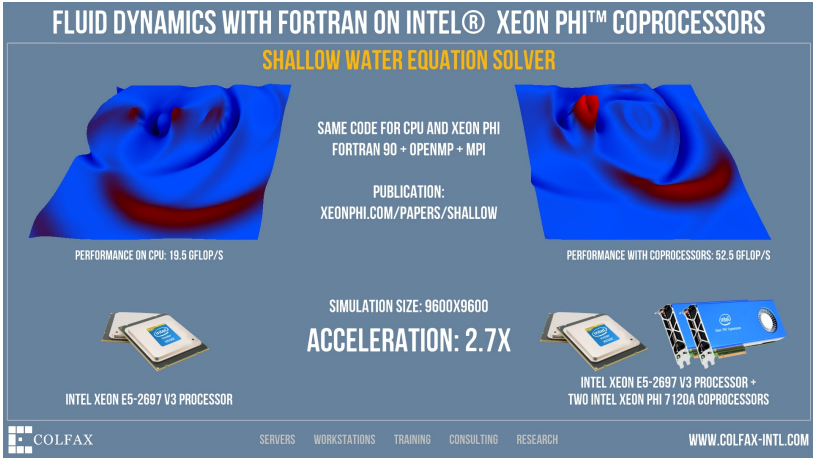
http://xeonphi.com/papers/heatcode

ASTROPHYSICAL CODE HEATCODE: AN OFFLOAD STORY



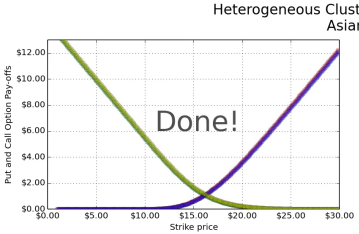
http://xeonphi.com/papers/heatcode

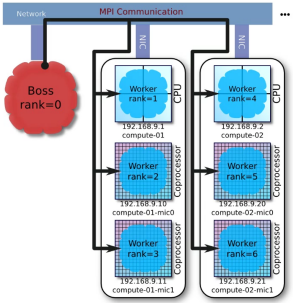
COMPUTATIONAL FLUID DYNAMICS: LEGACY CODE

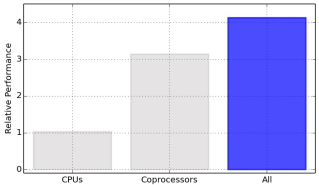


http://xeonphi.com/papers/shallow

ASIAN OPTION PRICING: HETEROGENEOUS CLUSTERING





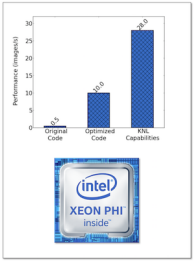


COLFAX
Customized Solutions

http://xeonphi.com/papers/heterogeneous

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

§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 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)

SOFTWARE NECESSARY TO BUILD XEON PHI APPLICATIONS:

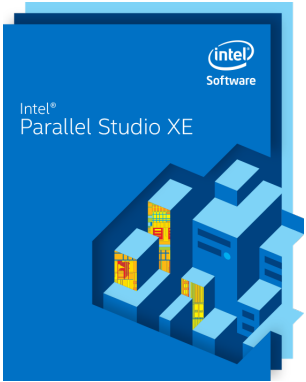
Compilers : Intel C Compiler, Intel C++ Compiler, and Intel Fortran Compiler — mandatory

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 <cstdio>
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%
```

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)

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.

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.