PROGRAMMING AND OPTIMIZATION FOR INTEL® ARCHITECTURE

Hands-On Workshop (HOW) Series "Deep Dive"

Session 1

Colfax International — *colfaxresearch.com*

© Colfax International, 2013–2017

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, 2013–2017

COURSE ROADMAP

- Module I. Programming Models
  - 01. Intel Architecture and Modern Code
  - 02. Xeon Phi, Coprocessors, Omni-Path
- Module II. Expressing Parallelism
  - 03. Automatic vectorization
  - 04. Multi-threading with OpenMP
  - 05. Distributed Computing, MPI
- Module III. Performance Optimization
  - 06. Optimization Overview: N-body
  - 07. Scalar tuning, Vectorization
  - 08. Common Multi-threading Problems
  - 09. Multi-threading, Memory Aspect
  - 10. Access to Caches and Memory

© Colfax International, 2013–2017

HOW SERIES ONLINE

Course page:
colfaxresearch.com/how-series

- Slides
- Code
- Video
- Chat

More workshops:
colfaxresearch.com/training

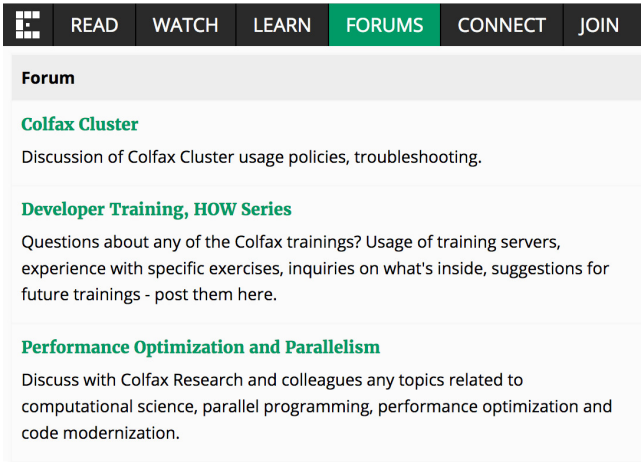
© Colfax International, 2013–2017

GET YOUR QUESTIONS ANSWERED: CHAT

colfaxresearch.com/how-series

© Colfax International, 2013–2017

GET YOUR QUESTIONS ANSWERED: FORUMS



colfaxresearch.com/forum

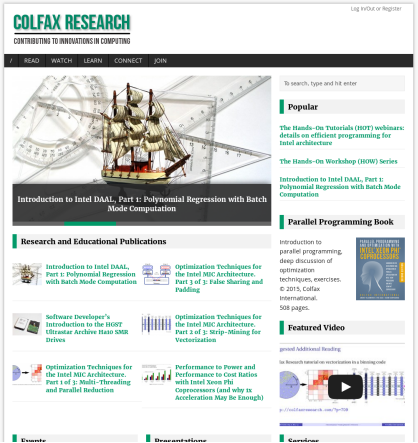
© Colfax International, 2013–2017

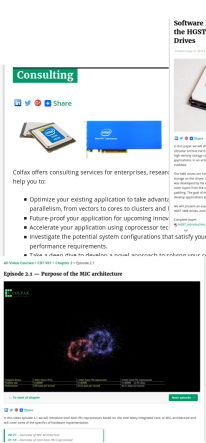
HANDS-ON EXERCISES AND REMOTE ACCESS

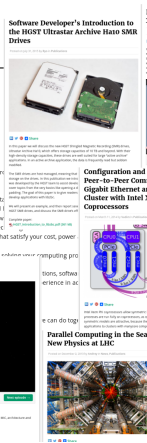
- All registrants receive an invitation from cluster@colfaxresearch.com
- Queue-based access to Intel Xeon E5, Intel Xeon Phi (KNC and KNL)
- Can access the cluster the entire 2 weeks of the workshop

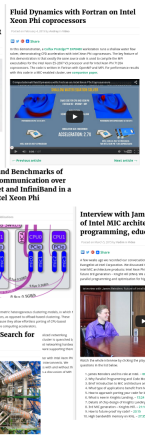
© Colfax International, 2013–2017

COLFAX RESEARCH









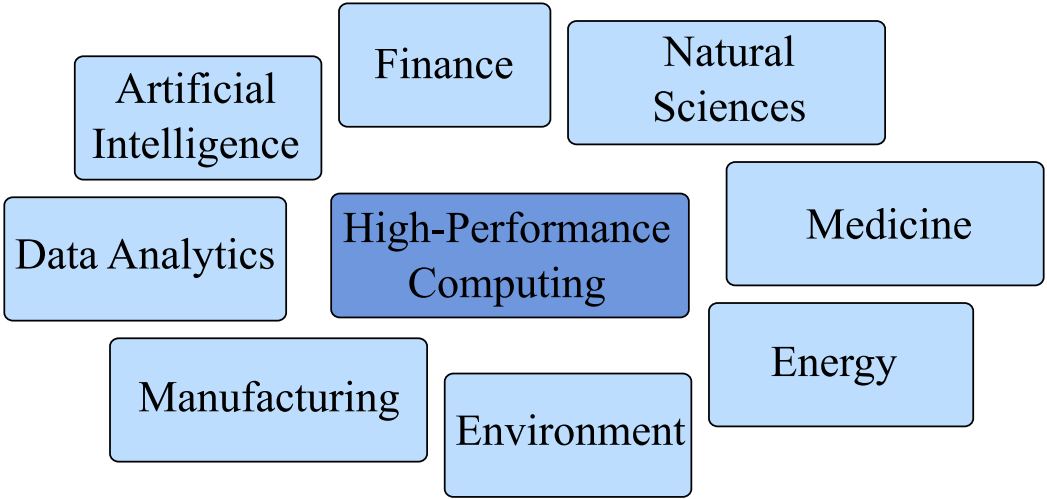
https://colfaxresearch.com/

© Colfax International, 2013–2017

§2. MODERN CHALLENGES

AREAS OF APPLICATION

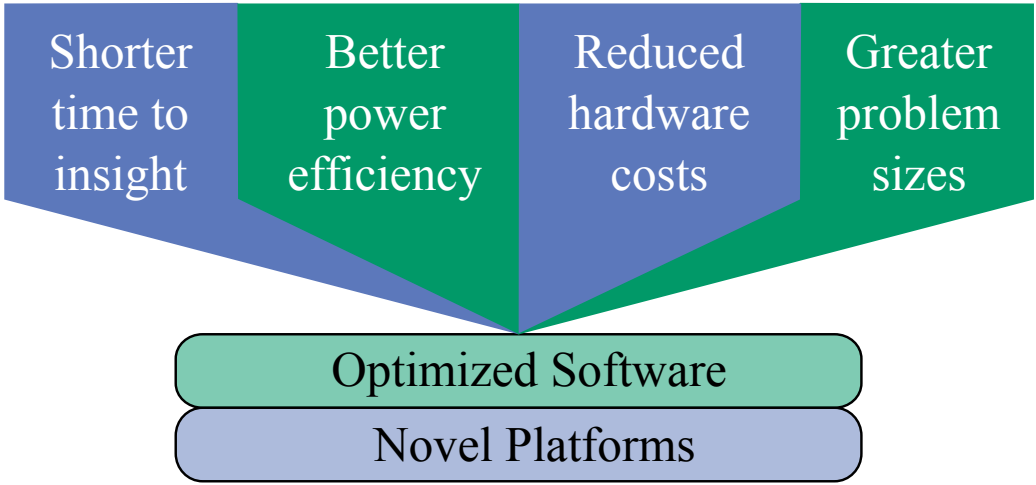
COMPUTING DOMAINS



© Colfax International, 2013–2017

NEED FOR SPEED

PERFORMANCE OPTIMIZATION OUTCOMES

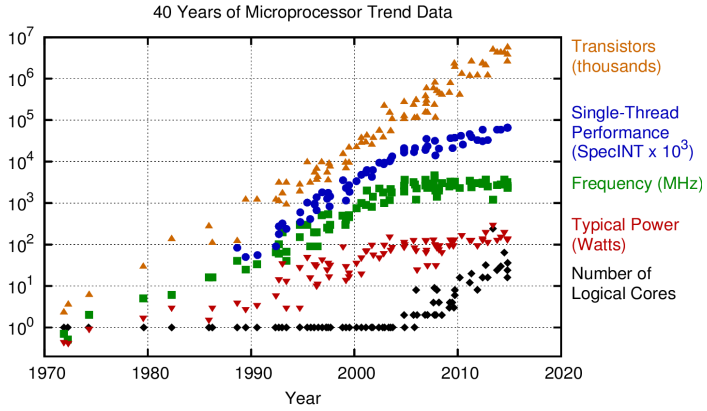


© Colfax International, 2013–2017

§3. MODERN ARCHITECTURE

HOW PROCESSORS GET FASTER

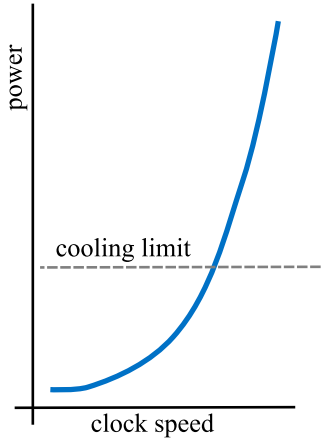
40 YEARS OF MICROPROCESSOR DATA



Source: karlrupp.net

© Colfax International, 2013–2017

POWER WALL



Liquid nitrogen for CPU overclocking

youtu.be/WZr0W_g0dqk

Microsoft's project Natick: immersed datacenter

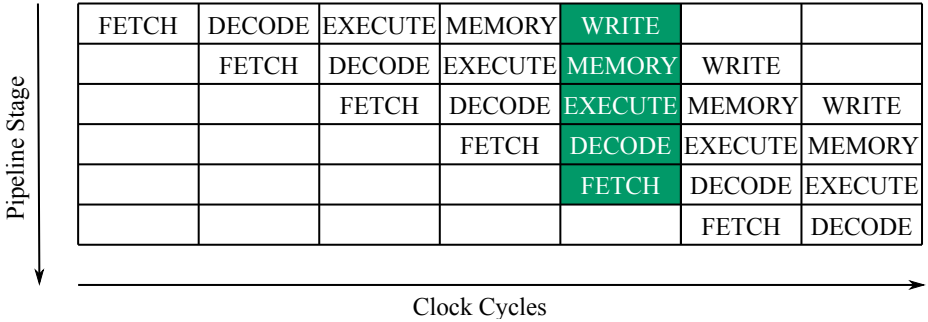
news.microsoft.com/natick

Cooling solutions for high clock speeds are not practical or expensive

© Colfax International, 2013–2017

INSTRUCTION-LEVEL PARALLELISM (ILP) WALL: PIPELINING

Pipelining – replication of hardware to run different stages of different instruction streams at the same time



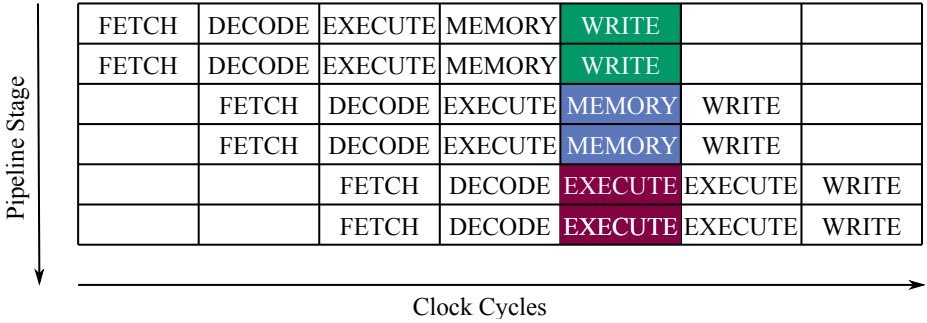
FETCH	DECODE	EXECUTE	MEMORY	WRITE		
	FETCH	DECODE	EXECUTE	MEMORY	WRITE	
		FETCH	DECODE	EXECUTE	MEMORY	WRITE
			FETCH	DECODE	EXECUTE	MEMORY
				FETCH	DECODE	EXECUTE
					FETCH	DECODE

Only so many pipeline stages, possible conflicts

© Colfax International, 2013–2017

INSTRUCTION-LEVEL PARALLELISM (ILP) WALL: SUPERSCALAR EXECUTION

Superscalar Execution – hardware checks for independence of operations, pipelines multiple instructions in a cycle

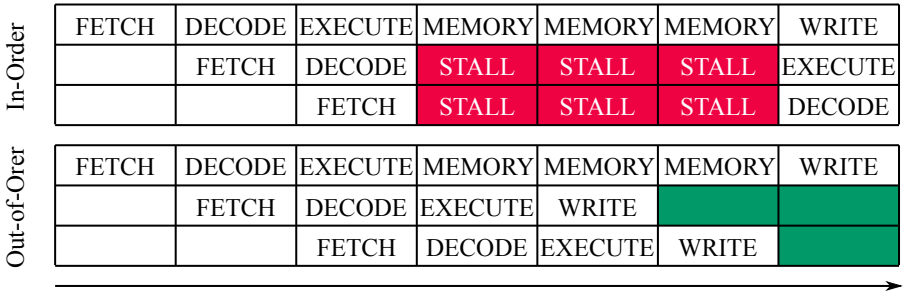


Automatic search for independent instructions reqiures extra resources

© Colfax International, 2013–2017

MEMORY WALL: OUT-OF-ORDER EXECUTION

Out-of-order Execution – hardware re-orders instructions in a stream to minimize latencies



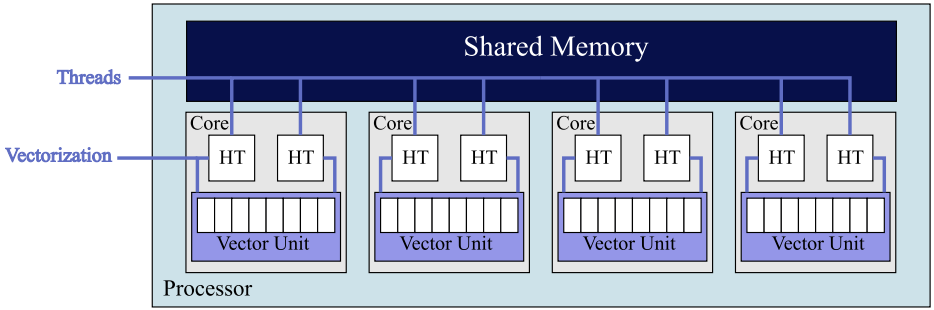
CPU performance grows faster than RAM bandwidth, OOE can't fill gap

© Colfax International, 2013–2017

PARALLELISM

CORES – multiple instructions on multiple data elements (MIMD)

VECTORS – single instruction on multiple data elements (SIMD)



Unbounded growth opportunity, but **not automatic**

© Colfax International, 2013–2017

PARALLELISM IS THE PATH FORWARD

- Clock speed has hit the power wall
- Automatic parallelism has hit the ILP wall
- Out-of-order execution cannot overcome the memory wall

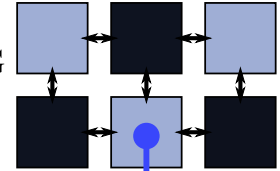
The Show Must Go On

Hardware keeps evolving through parallelism.
Software must catch up!

© Colfax International, 2013–2017

PARALLEL PROGRAMMING LAYERS

CLUSTER COMPUTING
in distributed memory

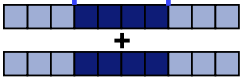


```
MPI_Sendrecv(data, k,
   MPI_DOUBLE, data2,
   ... );
```

MULTITHREADING
in shared memory

```
#pragma omp parallel for
for (j = 0; j < m; j++)
  ComputeSubset(j);
```

VECTORIZATION
of floating-point math



```
#pragma omp simd
for (i = 0; i < n; i++)
  A[i] += B[i];
```


© Colfax International, 2013–2017

INTEL ARCHITECTURE

INTEL COMPUTING PLATFORMS

General-Purpose Processors

Intel® Xeon®
Intel® Core™
Intel® Atom™, ...

Specialized Processors

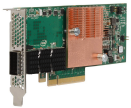
Intel® Xeon Phi™ processors and coprocessors

Computing Accelerators

Intel® VCA (x86)
Intel® Nervana™ Platform
Intel® DLIA™ (FPGAs)

Network Interconnects

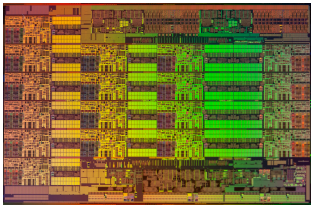
Intel® Omni-Path™ Architecture



© Colfax International, 2013–2017

INTEL XEON PROCESSORS

- 1-, 2-, 4-way
- General-purpose
- Highly parallel (44 cores*)
- Resource-rich
- Forgiving performance
- Theor. ~ 1.0 TFLOP/s in DP*
- Meas. ~ 154 GB/s bandwidth*

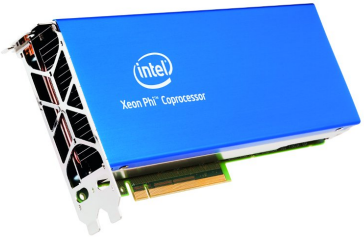


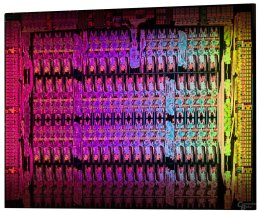
* 2-way Intel Xeon processor, Broadwell architecture (2016), top-of-the-line (e.g., E5-2699 V4)

© Colfax International, 2013–2017

INTEL XEON PHI PROCESSORS (1ST GEN)

- PCIe add-in card
- Specialized for computing
- Highly-parallel (61 cores*)
- Balanced for compute
- Less forgiving
- Theor. ~ 1.2 TFLOP/s in DP*
- Meas. ~ 176 GB/s bandwidth*



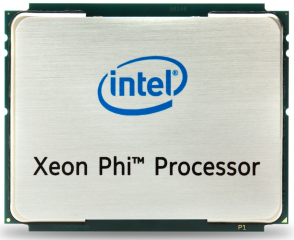


* Intel Xeon Phi coprocessor, Knighs Corner architecture (2012), top-of-the-line (e.g., 7120P)

© Colfax International, 2013–2017

INTEL XEON PHI PROCESSORS (2ND GEN)

- Bootable or PCIe add-in card
- Specialized for computing
- Highly-parallel (72 cores*)
- Balanced for compute
- Less forgiving than Xeon
- Theor. ~ 3.0 TFLOP/s in DP*
- Meas. ~ 490 GB/s bandwidth*



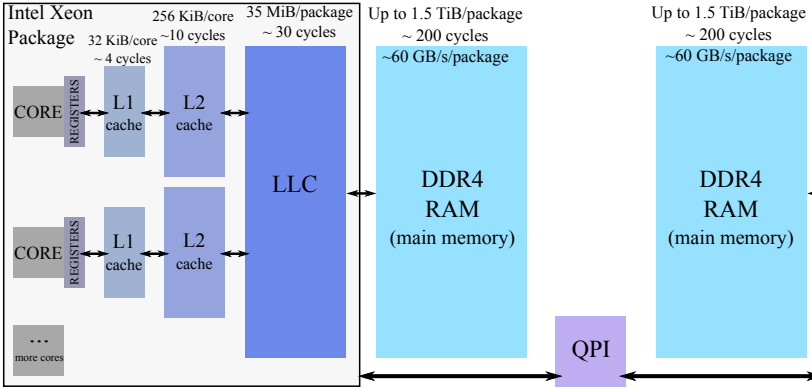
* Intel Xeon Phi processor, Knighs Landing architecture (2016), top-of-the-line (e.g., 7290P)

© Colfax International, 2013–2017

FORM-FACTORS AND MEMORY ORGANIZATION

INTEL XEON CPU: MEMORY ORGANIZATION

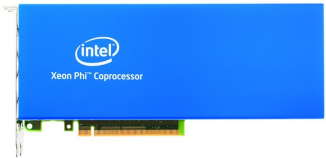
- Hierarchical cache structure
- Two-way processors have NUMA architecture

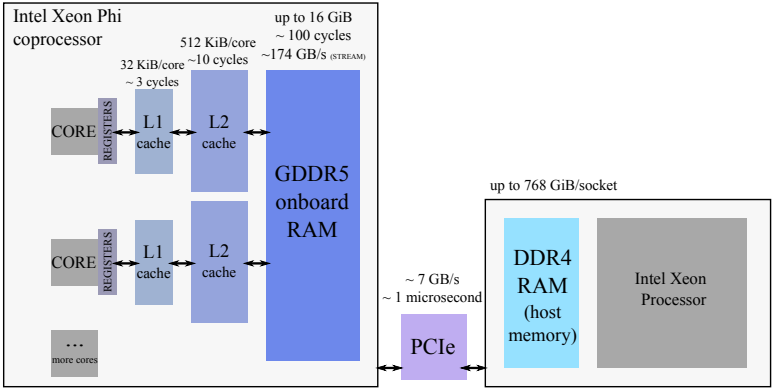


© Colfax International, 2013–2017

KNC MEMORY ORGANIZATION

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

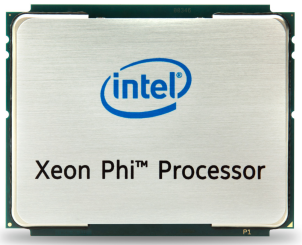


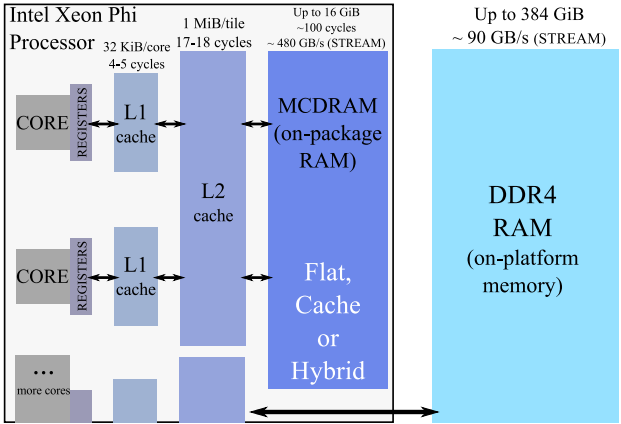


© Colfax International, 2013–2017

KNL MEMORY ORGANIZATION (BOOTABLE)

- On-package high-bandwidth memory (HBM) – MCDRAM
- Optimized for arithmetic performance and bandwidth (not latency)

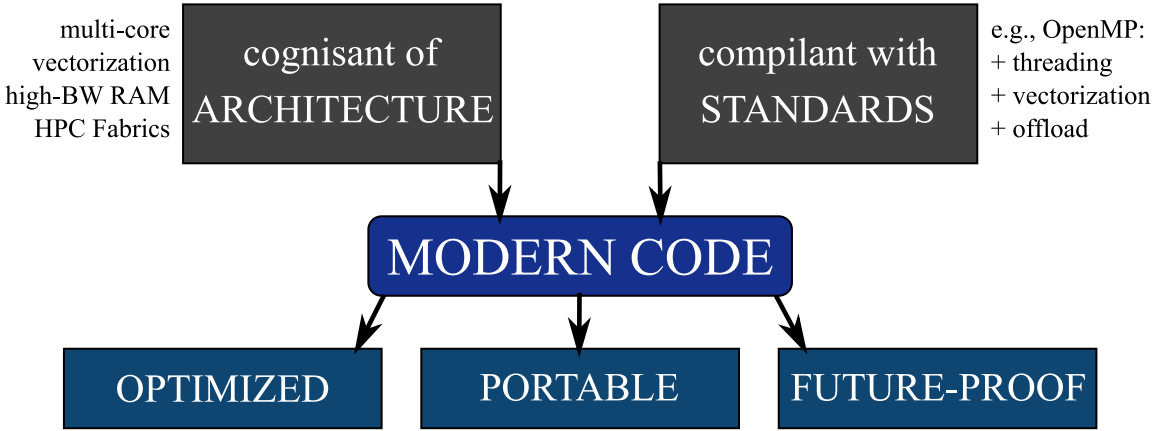




© Colfax International, 2013–2017

§4. MODERN CODE

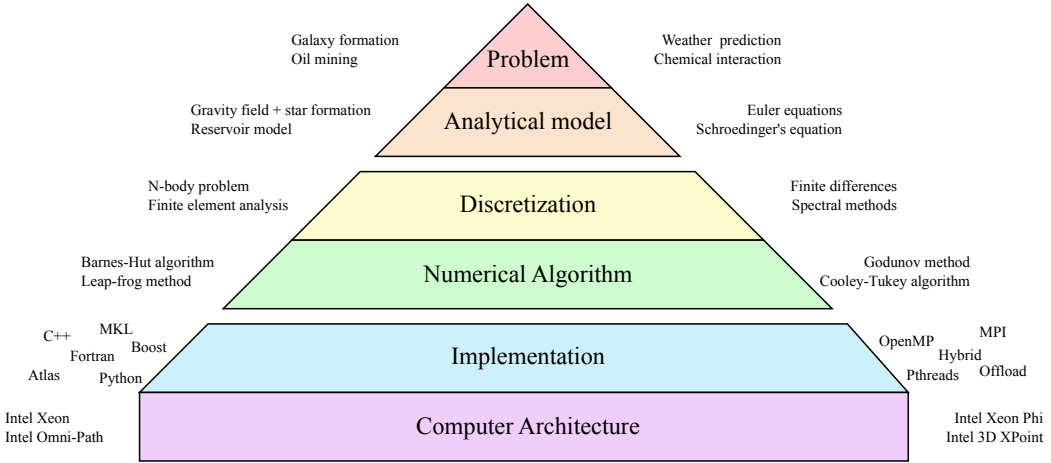
ONE CODE FOR ALL PLATFORMS



© Colfax International, 2013–2017

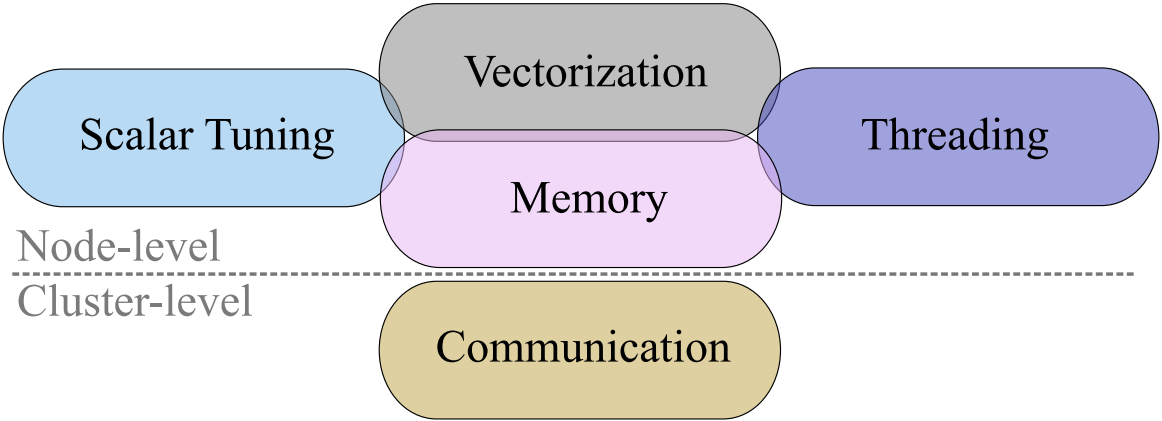
OPTIMIZATION AND FUTURE-PROOFING

COMPUTING IN SCIENCE AND ENGINEERING



© Colfax International, 2013–2017

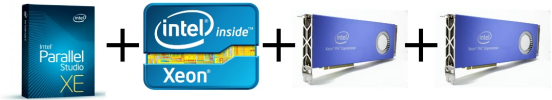
OPTIMIZATION AREAS



© Colfax International, 2013–2017

MOTIVATING EXAMPLES

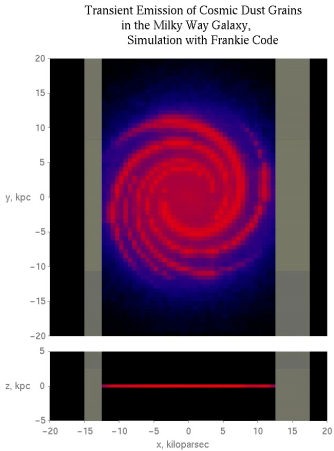
ASTROPHYSICAL CODE HEATCODE: AN OFFLOAD STORY



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)

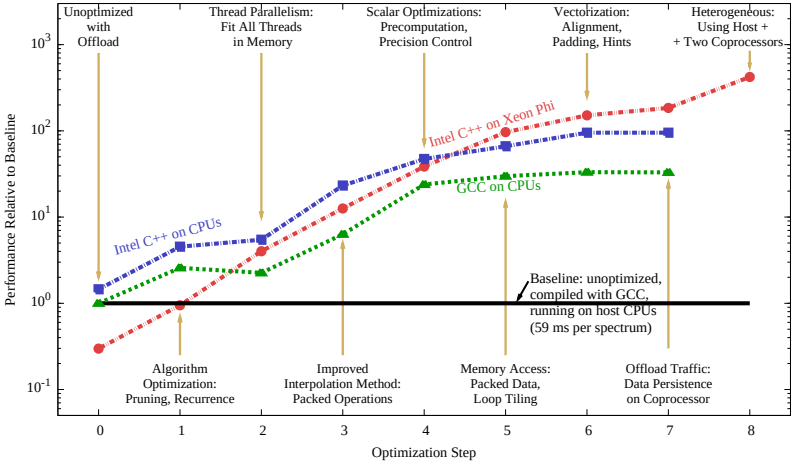
Goal achieved!



https://colfaxresearch.com/heatcode

© Colfax International, 2013–2017

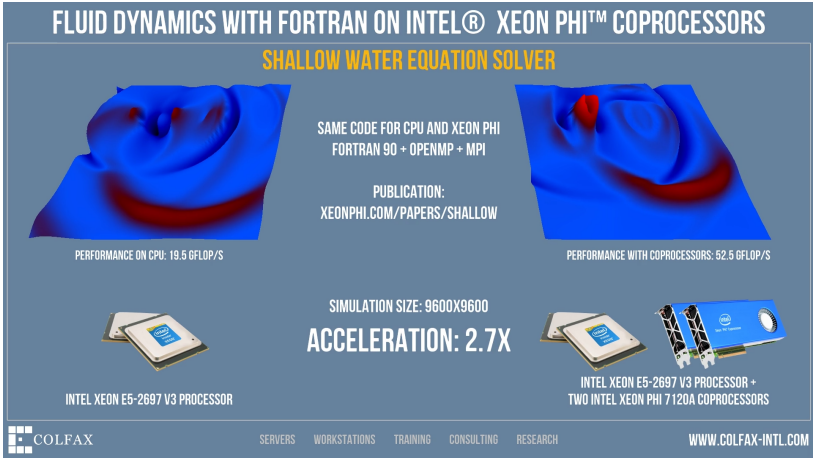
ASTROPHYSICAL CODE HEATCODE: AN OFFLOAD STORY



https://colfaxresearch.com/heatcode

© Colfax International, 2013–2017

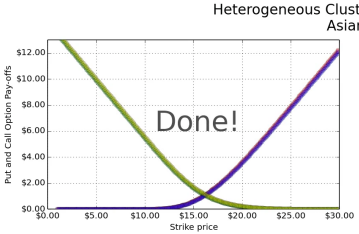
COMPUTATIONAL FLUID DYNAMICS: LEGACY CODE

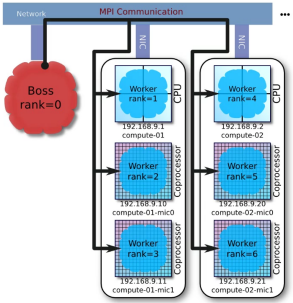


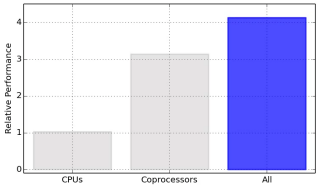
https://colfaxresearch.com/shallow-water

© Colfax International, 2013–2017

ASIAN OPTION PRICING: HETEROGENEOUS CLUSTERING







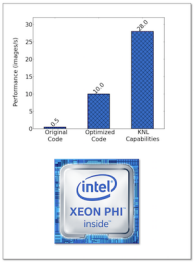
COLFAX
Customized Solutions

https://colfaxresearch.com/heterogeneous

© Colfax International, 2013–2017

MACHINE LEARNING: OPTIMIZED MIDDLEWARE

INTEL® XEON PHI™ PROCESSORS — MACHINE LEARNING



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

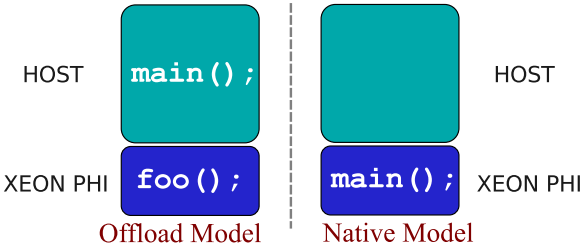
COLFAX Customized Solutions

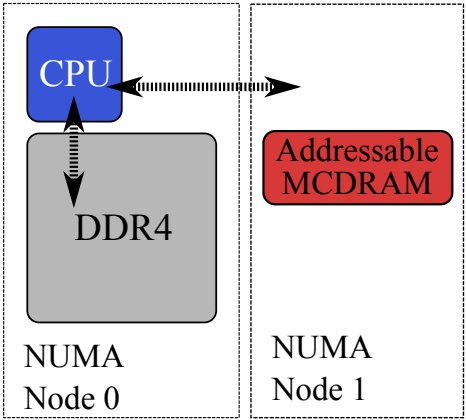
https://colfaxresearch.com/isc16-neuraltalk

© Colfax International, 2013–2017

WHAT YOU ARE GOING TO LEARN

HETEROGENEOUS AND NUMA ARCHITECTURES



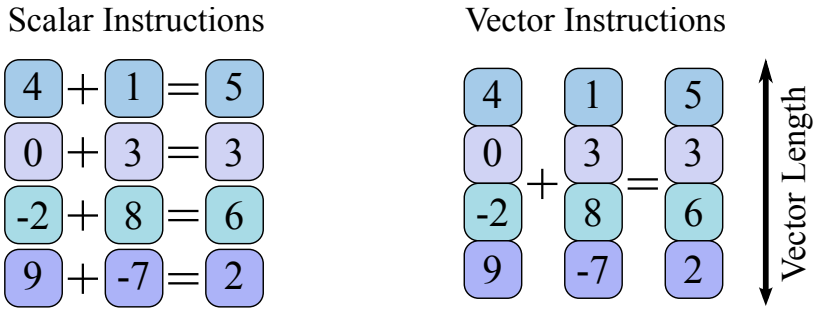


Session 2: handling memory organization in Intel Xeon Phi processors

© Colfax International, 2013–2017

DATA PARALLELISM AND VECTOR INSTRUCTIONS

Vectors – form of SIMD architecture (Single Instruction Multiple Data).

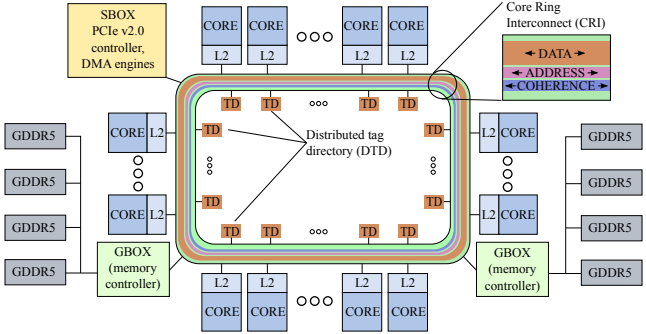


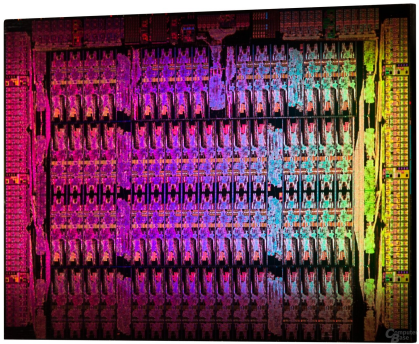
Session 3: automatic vectorization with Intel compilers

© Colfax International, 2013–2017

TASK PARALLELISM AND CORES

Cores implement MIMD (Multiple Instruction Multiple Data) arch



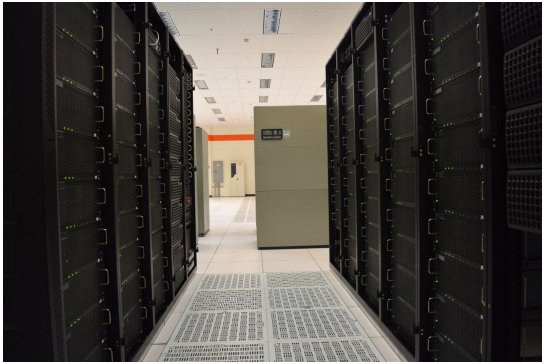


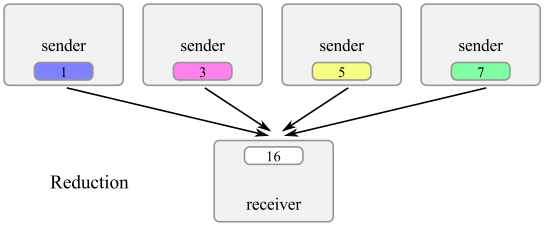
Session 4: multi-threading with OpenMP

© Colfax International, 2013–2017

DISTRIBUTED-MEMORY PROGRAMMING

Clusters form distributed-memory systems with network interconnects

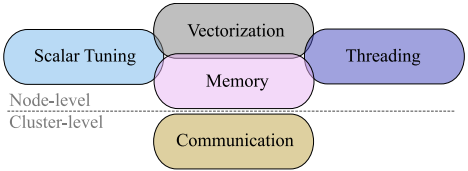


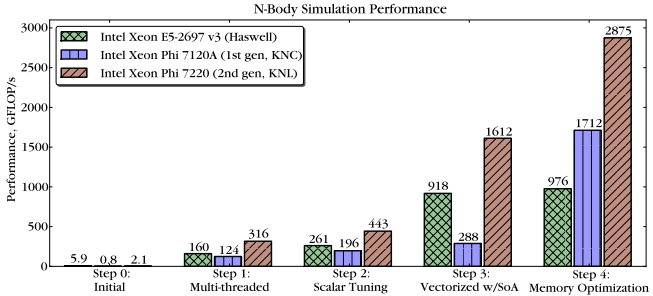


Session 5: Message Passing Interface (MPI)

© Colfax International, 2013–2017

OPTIMIZATION OVERVIEW

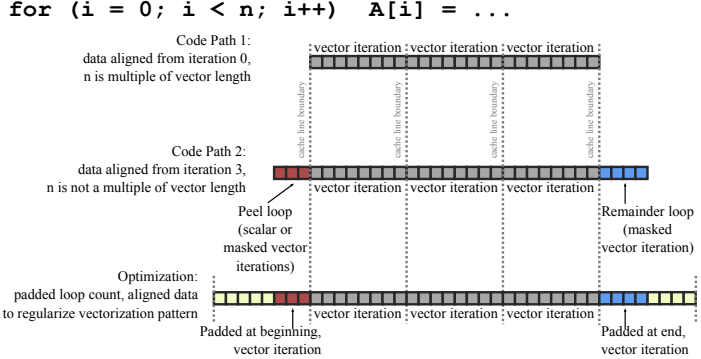




Session 6: optimization overview, case study

© Colfax International, 2013–2017

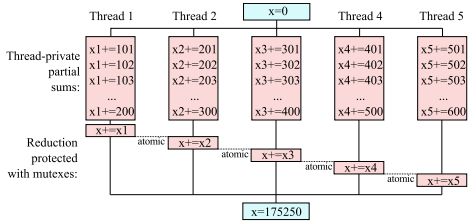
SCALAR TUNING, OPTIMIZATION OF VECTORIZATION

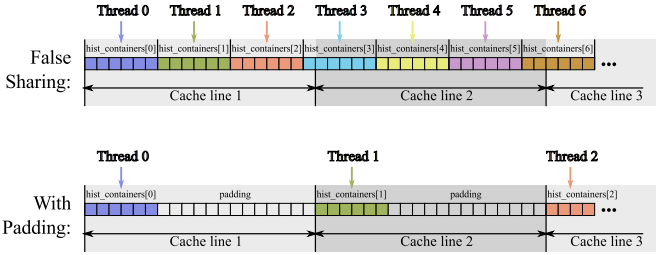


Session 7: precision control, regularizing vectorization patterns

© Colfax International, 2013–2017

COMMON ISSUES IN MULTI-THREADING

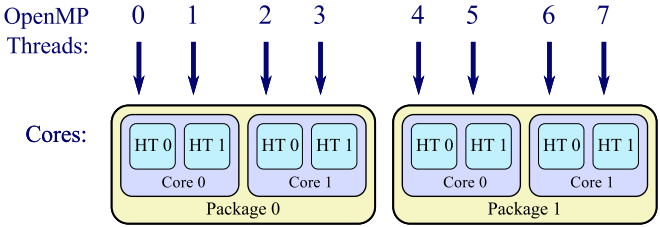


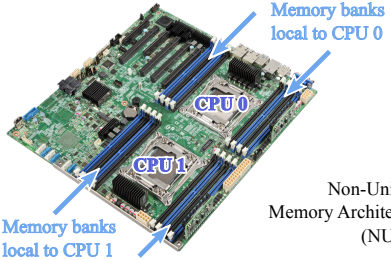


Session 8: minimizing synchronization, avoiding false sharing, strip-mining for parallelism

© Colfax International, 2013–2017

MULTI-THREADING, MEMORY ASPECT



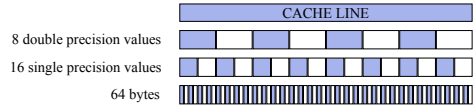


Non-Uniform Memory Architecture (NUMA)

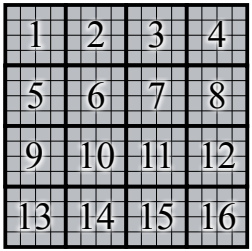
Session 9: thread affinity, NUMA locality, scheduling

© Colfax International, 2013–2017

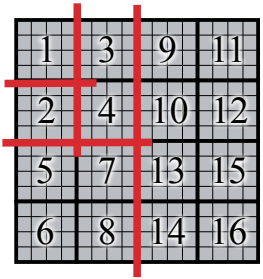
CACHE AND MEMORY ACCESS



Tiling



Cache-Oblivious Recursion



Session 10: loop transformations for locality, bandwidth secrets

© Colfax International, 2013–2017

§5. HANDS-ON DEMONSTRATION

ACCESS THE COLFAX CLUSTER

Home | Learn | Connect | Program | Compute | Benchmark | Log Out

Welcome to Colfax Cluster!

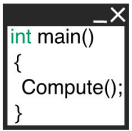
Learn

what to expect on the Colfax Cluster

Connect

from your home computer to the cluster

Program



using modern code practices

Compute

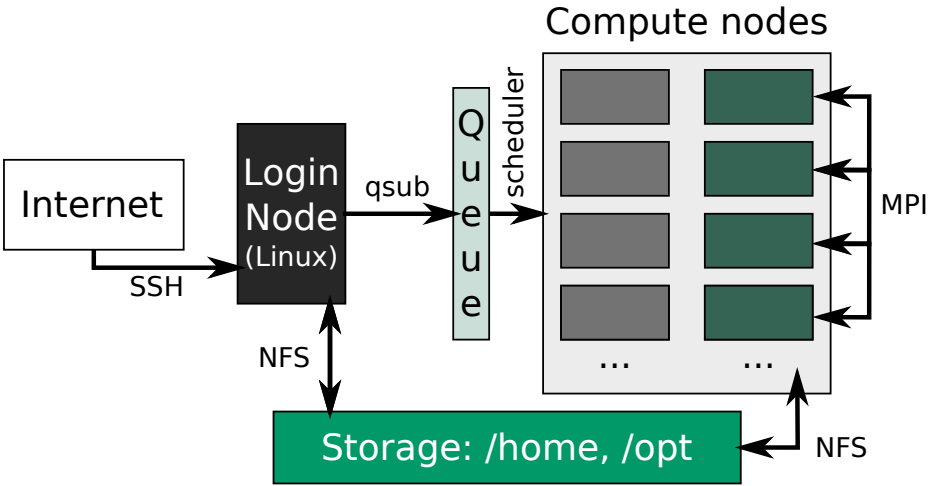
with cluster job management tools

Benchmark

applications on the cluster

© Colfax International, 2013–2017

JOB MANAGEMENT IN THE CLUSTER



© Colfax International, 2013–2017

DOWNLOAD LABS

```
[u111@c005 ~]% git clone https://github.com/ColfaxResearch/HOW-Series-Labs.git
Cloning into 'HOW-Series-Labs'...
[u111@c005 ~]% ls HOW-Series-Labs/*/
HOW-Series-Labs/2/:
2.01-native-basic  2.03-offload-basic         2.05-shared-virtual-memory-basic
2.02-native-MPI    2.04-offload-asynchronous  2.06-shared-virtual-memory-complex-

HOW-Series-Labs/3/:
3.01-vectorization  3.03-OpenMP-reduction  3.05-Cilk-Plus-basics    3.07-Cilk-Plu
3.02-OpenMP-basics  3.04-OpenMP-tasks      3.06-Cilk-Plus-reducers  3.08-MPI-basi

HOW-Series-Labs/4/:
4.01-overview-nbody                          4.07-threading-affinity
4.02-vectorization-data-structures-coulomb   4.08-memory-tiling-matrix_x_vector
4.03-vectorization-tuning-lu-decomposition   4.09-memory-loop-fusion-statistics
4.04-threading-misc-histogram                4.10-offload-double-buffering-dgem
...
```


© Colfax International, 2013–2017

REVIEW AND WHAT'S NEXT

- Computers are getting faster through parallelism and specialization
- Intel Xeon E5 product family – general-purpose parallel processors
- Intel Xeon Phi product family – specialized parallel processors
- Coprocessor – either offload device or an additional compute node

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

© Colfax International, 2013–2017