



# Practical Usage of the Intel Math Kernel Library (MKL)

The Hands-On Workshop (HOW) Series “Tools”

Andrey Vladimirov, PhD — Colfax International  
[colfaxresearch.com](http://colfaxresearch.com)

# 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.

## About the Series

Hands-On Workshop (HOW “Tools” Series): webinars on efficient programming for the Intel architecture with the help of dedicated software development tools



**GOT THE TOOLS - NOW WHAT?**

Learn workflows and methodology with the “HOW” tools\* training and hands-on demos

\* Intel MKL | Intel Advisor | Intel VTune Amplifier

[colfaxresearch.com/how-tools-16-06](http://colfaxresearch.com/how-tools-16-06)

Learn More



THE "HOW" SERIES

**DEEP DIVE**  
WITH CODE MODERNIZATION EXPERTS

\*10x 2-hour sessions | 24-hour 2-weeks remote access to a system | Filling up fast, register now!

Interested? Sign-up at:

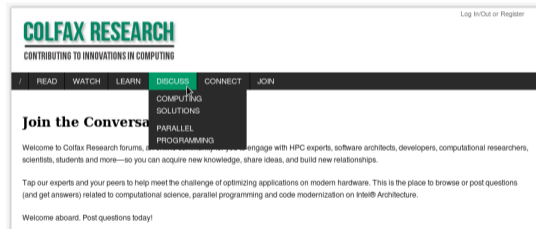
[colfaxresearch.com/how-series](https://colfaxresearch.com/how-series)

# Get Your Questions Answered

**Chat** (for this course):  
[colfaxresearch.com/how-tools-16-06](http://colfaxresearch.com/how-tools-16-06)



**Forums** (general):  
[colfaxresearch.com/discussion](http://colfaxresearch.com/discussion)



## §2. Intel Architecture

# Computing Platforms

## Intel Xeon Processor



Current: Broadwell  
Upcoming: Skylake

Multi-Core Architecture

## Intel Xeon Phi Coprocessor, 1st generation



Current: Knights Corner (KNC)

## Intel Xeon Phi Processor, 2nd generation\*



\* socket and coprocessor versions

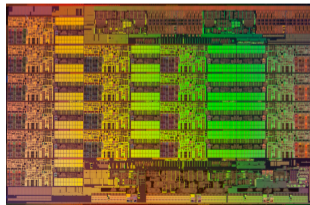
Upcoming: Knights Landing (KNL)

Intel Many Integrated Core (MIC) Architecture

# Intel Xeon CPU: Purpose and Specifications

General-purpose platform for demanding computing applications.

- Up to  $\sim 1$  TFLOP/s in DP
- Up to  $\sim 2$  TFLOP/s in SP
- Up to 768 GiB DDR4 RAM
- $\sim 126$  GB/s bandwidth
- Hardware-rich: forgiving of sub-optimal code

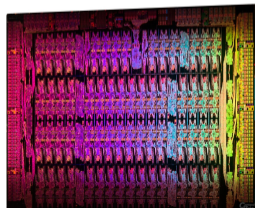
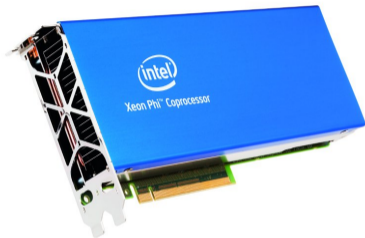




# Intel Xeon Phi Processors (1st Gen)

Specialized platform for demanding computing applications.

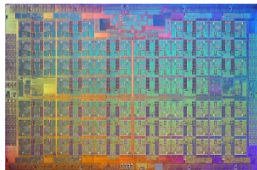
- PCIe end-point device
- ~ 1.2 TFLOP/s in DP
- ~ 2.4 TFLOP/s in SP
- Up to 16 GiB GDDR5 RAM
- ~ 176 GB/s bandwidth
- Heterogeneous clustering
- Runs special Linux distribution



# 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**



## §3. Intel Math Kernel Library

# Performance Tuning Methodology

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

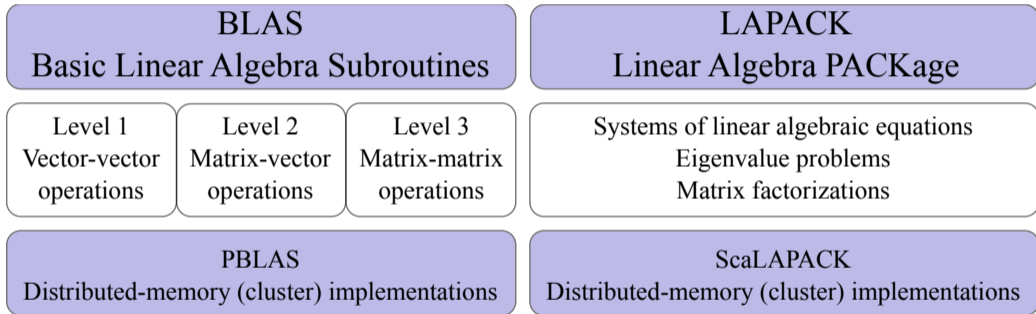
Details in the [HOW Series](#) and [our book](#).

# Scope of MKL

Intel Math Kernel Library (MKL) — standard mathematical functions optimized for Intel architecture.

Linear Algebra	Fast Fourier Transform	Vector Math	Vector Random Number Generators	Summary Statistics	Data Fitting
BLAS LAPACK Sparse solvers ScaLAPACK	Multidimensional (up to 7D) FFTW interfaces Cluster FFT	Trigonometric Hyperbolic Exponential Logarithmic Power/Root Rounding	Congruential Recursive Wichmann-Hill Mersenne Twister Sobol Neiderreiter Non-deterministic	Kurtosis Variation coefficient Quantiles, order statistics Min/max Variance-covariance	Splines Interpolation Cell search

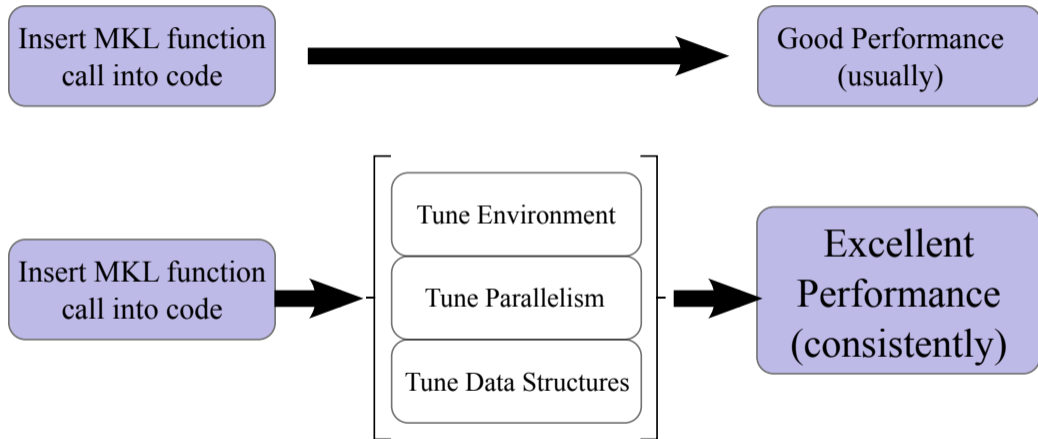
# Linear Algebra Support



# Using MKL with Intel Xeon Phi Coprocessors

	Native	Compiler-Assisted Offload (CAO)	Automatic Offload (AO)
Programming:	Call MKL functions like from CPU code	Call MKL functions from #pragma offload	Call MKL functions from CPU code
Compilation:	With argument -mmic	Conventional for CPU	Conventional for CPUs
Running	On Xeon Phi	On host CPU	On host CPU; MKL_MIC_ENABLE=1
Data movement	Data already on coprocessor	Managed by programmer	Managed by MKL
Functions	All	All	Some; problem size > threshold
Multiple Copr.	Use cluster functions	DIY	Supported out-of-box

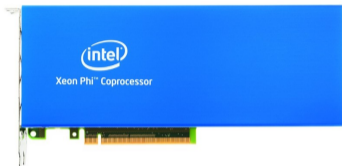
# Performance Tuning with MKL





## §4. Hands-On Labs

# “Standard Candle” Testbench



One Intel Xeon Phi 7120P  
coprocessor (2012)  
TDP: 300 W, RCP: \$4129

*vs.*



Two Intel Xeon E5-2697 v3  
CPUs (2014)  
TDP: 290 W, RCP: \$5404

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

# Batch Processing

- Feed multiple signals to MKL
- Let the library decide on the parallel strategy.

```
1 MKL_Complex16* data =  
2     (MKL_Complex16*) malloc(sizeof(MKL_Complex16)*fft_size*num_fft);  
3  
4 DFTI_DESCRIPTOR_HANDLE handle;  
5 DftiCreateDescriptor(&handle, DFTI_DOUBLE, DFTI_COMPLEX, 1, (MKL_LONG)fft_size);  
6 DftiSetValue(handle, DFTI_NUMBER_OF_TRANSFORMS, num_fft);  
7 DftiSetValue(handle, DFTI_INPUT_DISTANCE, fft_size);  
8 DftiSetValue(handle, DFTI_OUTPUT_DISTANCE, fft_size);  
9 DftiSetValue(handle, DFTI_PLACEMENT, DFTI_INPLACE);  
10 DftiCommitDescriptor(handle);
```

# Thread Affinity Tuning

---

---

## **Xeon**

MKL uses 1 thread/core

OMP\_NUM\_THREADS=#phys. cores

KMP\_AFFINITY=scatter

KMP\_AFFINITY=compact (HT off)

KMP\_AFFINITY=compact,1 (HT on)

---

## **Xeon Phi**

MKL uses 4 threads/core

OMP\_NUM\_THREADS=4×#cores

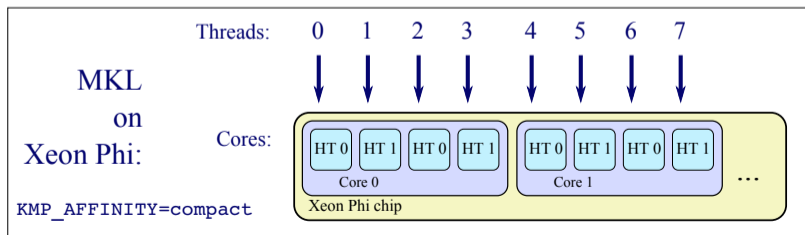
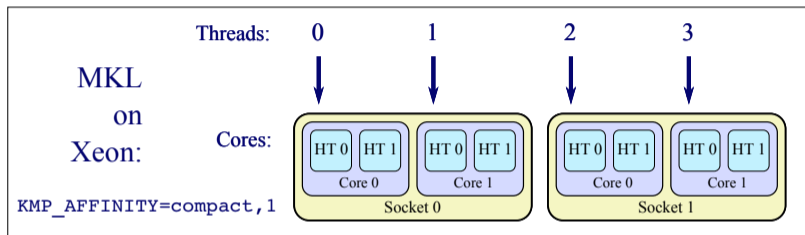
KMP\_AFFINITY=scatter

KMP\_AFFINITY=compact

See also HOW series [Session 8](#).

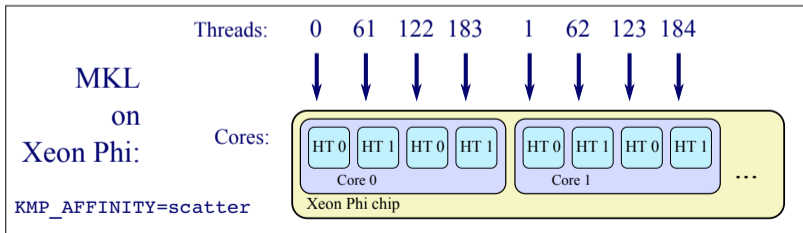
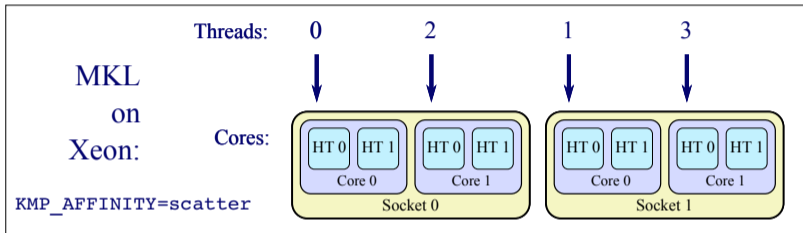
# Thread Affinity: Compact Pattern

Generally beneficial for compute-bound applications.



# Thread Affinity: Scatter Pattern

Generally beneficial for bandwidth-bound applications.



# Data Container Tweaks

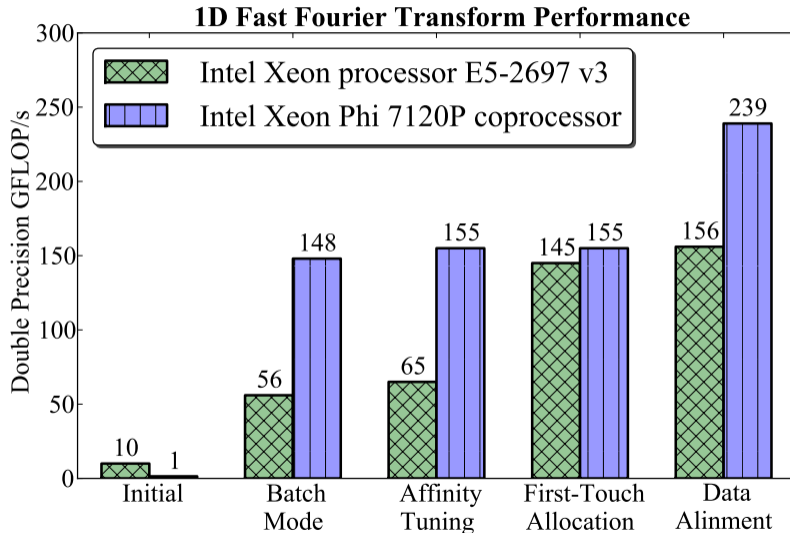
Parallel first-touch (see also HOW series [Session 8](#)):

```
1  #pragma omp parallel for
2  for(int i = 0; i < num_fft; i++)
3      for(int j = 0; j < fft_size; j++) {
4          data[i*fft_size+j].real = cos(k*j);
5          data[i*fft_size+j].imag = sin(k*j);
6      }
```

Data alignment (see also HOW series [Session 6](#))

```
1  MKL_Complex16* data =
2      (MKL_Complex16*) mkl_malloc(sizeof(MKL_Complex16)*fft_size*num_fft, 64);
```

# Complex-to-Complex 1D FFT Performance

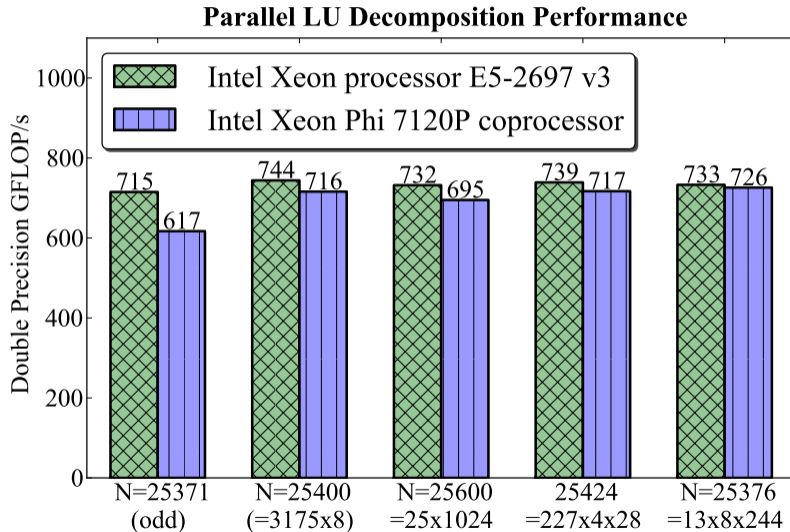




# Problem Size Tuning

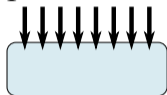
- Some functions require “good” problem sizes
- Good practice #1: array length a multiple of vector length
- Good practice #2: array length a multiple of number of threads
- Good practice #3: array length a multiple of a power of 2
- BLAS and LAPACK have “leading array dimension” for padding rows

# LU Decomposition Size Tuning



# Nested Parallelism

Fine-grained  
parallelism



...

throwing all threads  
on one MKL function

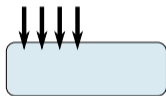
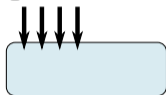
Coarse-grained  
parallelism



...

using one thread  
per MKL function

Nested  
parallelism



...

putting teams of threads  
on several MKL functions

# OpenMP Hot Teams

---

---

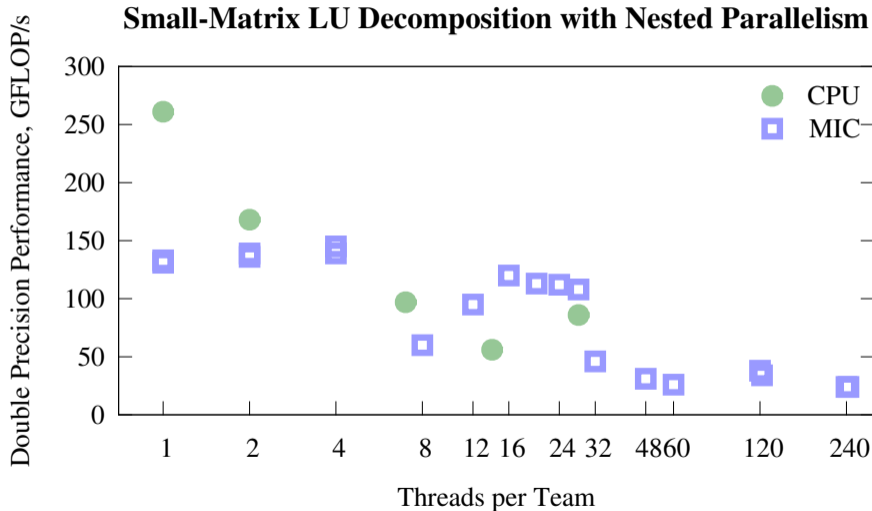
## Xeon

- `OMP_NUM_THREADS=2,14`
- `OMP_NESTED=1`  
`OMP_PROC_BIND=spread,close`  
`OMP_PLACES=cores`
- `KMP_HOT_TEAMS_MODE=1`  
`KMP_HOT_TEAMS_MAX_LEVEL=2`  
`OMP_MAX_ACTIVE_LEVELS=2`
- `MKL_DYNAMIC=false`

## Xeon Phi

- `OMP_NUM_THREADS=60,4`
- `OMP_NESTED=1`  
`OMP_PROC_BIND=spread,close`  
`OMP_PLACES=threads`
- `KMP_HOT_TEAMS_MODE=1`  
`KMP_HOT_TEAMS_MAX_LEVEL=2`  
`OMP_MAX_ACTIVE_LEVELS=2`
- `MKL_DYNAMIC=false`

# Small Matrix LU Decomposition with Nested Parallelism



# Automatic Offload

```
MKL_MIC_ENABLE=1

OMP_NUM_THREADS=... # CPU thread setting
KMP_AFFINITY=... # CPU affinity setting

MIC_KMP_PLACE_THREADS=4t # MIC thread setting
MIC_KMP_AFFINITY=... # MIC affinity setting
```

# Compilation of R with MKL

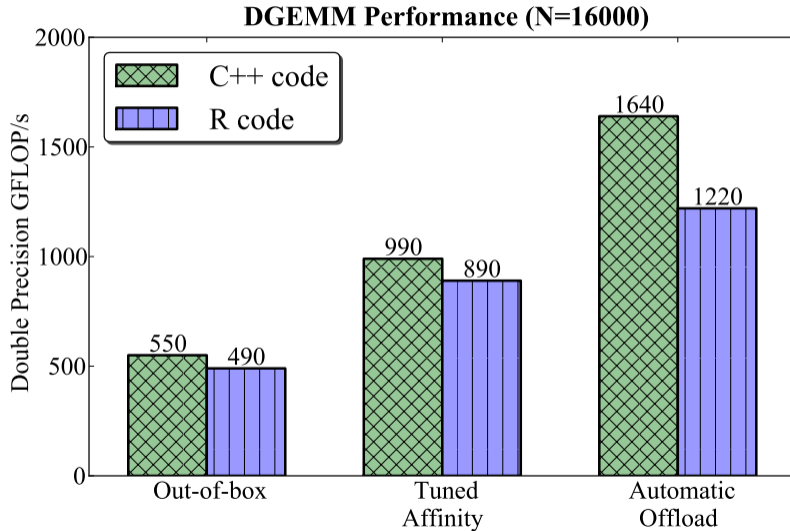
```
./configure \  
  --with-blas="-L/opt/intel/mkl/lib/intel64 \  
              -lmkl_intel_lp64 -lmkl_core -lmkl_intel_thread -lpthread -lm" \  
  --with-lapack \  
  CC=icc CFLAGS="-O2 -qopenmp -I/opt/intel/mkl/include" \  
  CXX=icpc CXXFLAGS="-O2 -qopenmp -I/opt/intel/mkl/include" \  
  F77=ifort FFLAGS="-O2 -qopenmp -I/opt/intel/mkl/include" \  
  FC=ifort FCFLAGS="-O2 -qopenmp -I/opt/intel/mkl/include" \  
  --prefix=/opt/R
```

```
user@host% make
```

```
user@host% su
```

```
root@host% make install
```

# Performance with C++ Code and in R





# §5. Additional Information

# Acquiring MKL

	Community License	Commercial License
Cost	Free	Per developer
Support	Intel Premier Support	Community forum
Use in products	Yes	Yes
Royalty-free	Yes	Yes

# Compilation with MKL

Intel® Math Kernel Library Link Line Advisor

July 20, 2012

Share Tweet + Share

Forums  
Intel® Math Kernel Library

Introduction

The Intel® Math Kernel Library (Intel® MKL) is designed to run on multiple processors and operating systems. It is also compatible with several compilers and third party libraries, and provides different interfaces to the functionality. To support these different environments, tools, and interfaces Intel MKL provides multiple libraries from which to choose.

To see what libraries are recommended for a particular use case, specify the parameters in the drop down lists below.

Intel® Math Kernel Library (Intel® MKL) Link Line Advisor v4.5

Select Intel® product: Intel(R) MKL 11.3.1

Select OS: Linux\*

Select usage model of Intel® Xeon Phi™ Coprocessor: Automatic Offload

Select compiler: GNU C/C++

Select architecture: Intel(R) 64

Use this link line:

```
-Wl,--no-as-needed -L${MKLROOT}/lib/intel64
-lmkl_intel_lp64 -lmkl_core -lmkl_intel_thread
-liomp5 -ldl -lpthread -ln
```

Compiler options:

```
-m64 -I${MKLROOT}/include
```

Rate Us: ⭐ ⭐ ⭐ ⭐ ⭐ Look for us on: f t g+ y+ English >

<https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

## §6. Resources

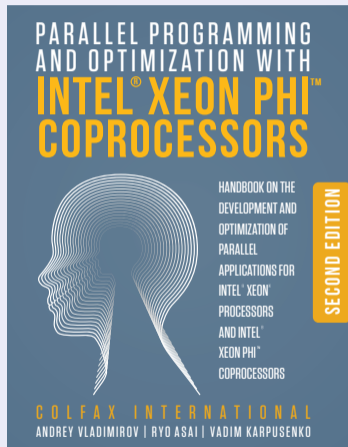
# Supplementary Materials: Textbook

ISBN: 978-0-9885234-0-1 (2nd edition, 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



CONTRIBUTING TO INNOVATIONS IN COMPUTING

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

---

READ WATCH LEARN CONNECT JOIN

To search, type and hit enter

---



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

**Popular**

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

**The Hands-On Workshop (HOW) Series**

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

---

**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: False Sharing and Padding**

**Software Developer's Introduction to the HGST Ultrastar Archive H800 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)**

**Parallel Programming Book**

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

© 2015, Colfax International.

508 pages.



---

**Featured Video**

See Research material on vectorization in a streaming code generated Additional Reading

See Research material on vectorization in a streaming 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 tech
- Investigate the potential system configurations that satisfy your cost, power performance requirements.
- Take a clean sheet to evaluate a novel approach to collaborate your computing platform

All Video Courses - COP 991 - Chapter 2 - Episode 1.1

**Episode 2.1 – Purpose of the MIC architecture**



[View](#)

In this episode (2) we will introduce how the MIC coprocessor based on the Intel Many Integrated Core, or MIC, architecture and software stack of the specific Intel-based Coprocessors.

00:07 Introduction of MIC Architecture  
01:48 Overview of Intel-based MIC Coprocessor

## Software Developer's Introduction to the HGST Ultrastar Archive H800 SMR Drives



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

In this paper we will discuss the new HGST Ultrastar Archive H800 SMR drives. Software Developer Help to guide offers storage capabilities of 10 Tb per drive, 500 Mb/s high density storage capacities. These drives are well suited for large "data archive" applications. In an other archive applications, the data is frequently read but seldom modified.

The H800 drives are high capacity, meaning the storage on the drive is 10 times the capacity of the previous generation. This means that you can store 10 times more data, meaning the need for more drives, more space, more power, more cooling. The goal of this paper is to give you a clean sheet to evaluate a novel approach to collaborate your computing platform

We will present an example, and then report our next H800 drives, and describe how drive life

## Fluid Dynamics with Fortran on Intel Xeon Phi coprocessors



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

In this demonstration a Colfax Productivity ENGINE installation on a dual-socket Intel Xeon server performing 100 million calculations with the application, the time to complete this demonstration is just under 10 minutes and the server is not occupied by the application. The Colfax Productivity ENGINE coprocessor can be used to accelerate the application. The Colfax Productivity ENGINE can be used to accelerate the application.

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



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

The Intel Xeon Phi coprocessors are designed to support peer-to-peer communication over Gigabit Ethernet and InfiniBand. This paper describes the configuration and benchmarks of peer-to-peer communication over Gigabit Ethernet and InfiniBand in a cluster with Intel Xeon Phi coprocessors.

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



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

In this video you can see an interview with James Reinders, the Director and Chief Architect of Intel Coprocessors. We discuss the future of the parallel programming and what the current challenges are for the parallel programming. Right now Intel has a lot of work to do in the area of parallel programming. We will discuss the future of parallel programming and education for high performance applications.

http://colfaxresearch.com/

colfaxresearch.com/how-tools-16-06

Resources

© Colfax International, 2013–

Learn More



THE "HOW" SERIES

**DEEP DIVE**

WITH CODE MODERNIZATION EXPERTS

\*10x 2-hour sessions | 24-hour 2-weeks remote access to a system | Filling up fast, register now!

Interested? Sign-up at:

[colfaxresearch.com/how-series](https://colfaxresearch.com/how-series)

## Slides, Code, Video

You can download slides, code and watch the video recording of this webinar here (requires registration for a free Colfax Research account):

[colfaxresearch.com/how-tools-16-06](http://colfaxresearch.com/how-tools-16-06)

Next webinar on June 17, 2016: “Guided Code Vectorization with Intel Advisor XE”:

Register