INTEL® PERFORMANCE LIBRARIES OVERVIEW

Agenda

- **Intel® Math Kernel Library**
- **Intel® Data Analytics Acceleration Library**
- **Intel® Integrated Performance Primitives**

Intel® Math Kernel Library (Intel® MKL)

Software Solutions Group

Intel Corporation

Agenda

Introduction and Overview

MKL Components

Some Special Features

- Conditional Numerical Reproducibility
- Batch mode processing
- Small Matrices
- New enhancements in MKL 11.3

References

Motivation

How and where to optimize?

1. Appropriate algorithm
2. **Performance Library**
3. Multicore
4. SIMD

Delivered Values

- Easy access to high perf.
- Rich functionality
- Support

```
for (int i = 0; i < M; ++i) {
 for (int j = 0; j < N; ++j) {
  c[i*K+j] = 0;
  for (int k = 0; k < K; ++k) {
   c[i*K+j] += a[i*N+k]
             * b[k*K+j];
  }
 }
}
```

Intel® Math Kernel Library

Intel® MKL is industry's leading math library *

Linear Algebra
- BLAS
- LAPACK
- Sparse solvers
- ScaLAPACK

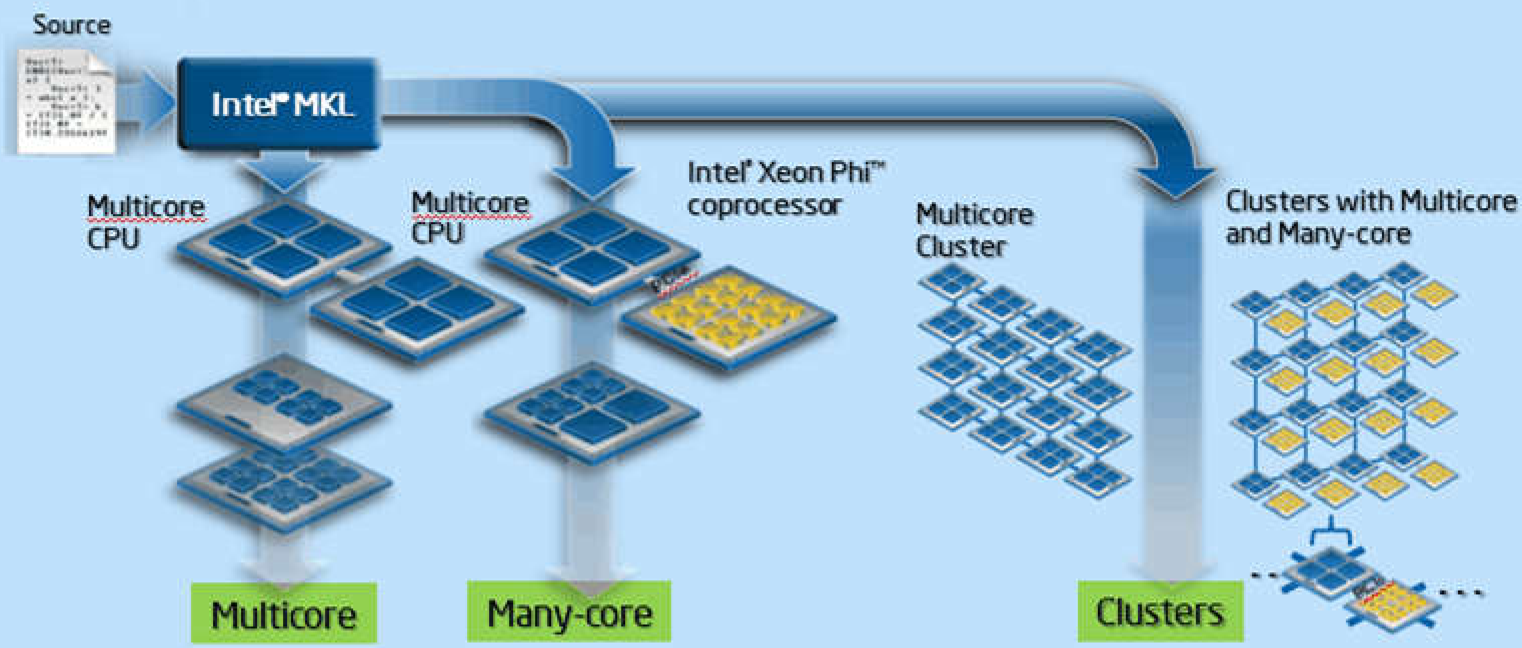
Fast Fourier Transforms
- Multidimensional (up to 7D)
- FFTW interfaces
- Cluster FFT

Vector Math
- Trigonometric
- Hyperbolic
- Exponential, Logarithmic
- Power / Root
- Rounding

Vector Random Number Generators
- Congruential
- Recursive
- Wichmann-Hill
- Mersenne Twister
- Sobol
- Neiderreiter
- Non-deterministic

Summary Statistics
- Kurtosis
- Variation coefficient
- Quantiles, order statistics
- Min/max
- Variance-covariance
- ...

Data Fitting
- Splines
- Interpolation
- Cell search



Intel® Math Kernel Library: BLAS

Basic Linear Algebra Subroutines (BLAS)	Level 1: Vector operations	Dot products, swap, min, max, scaling, rotation, etc.
	Level 2: Matrix-vector operations	Matrix-vector products, rank-1/rank-2 updates, triangular solvers, etc.
	Level 3: Matrix-matrix operations	Matrix-matrix products, rank-k/rank-2k updates, triangular solvers, etc.
	Sparse BLAS	BLAS level 1, 2 and 3 for sparse vectors and matrices
Matrix storage schemes	BLAS: Full, packed, and banded storage	
	Sparse BLAS: CSR, CSC, coordinate, diagonal, skyline, BSR, etc.	

Original BLAS available at
http://netlib.org/blas/

Intel® Math Kernel Library: LAPACK

Linear Algebra Package (LAPACK)	Solving systems of linear equations, factoring and inverting matrices.
	Solving linear least squares problems, Eigenvalues, singular value problems, and Sylvester's equations.
	Many auxiliary and utility functions.
	LAPACK driver routines: Combines several routines in one call to solve a particular problem.
ScaLAPACK	LAPACK for distributed memory architectures.
	Using MPI, BLACS (basic linear algebra communication subprograms), and BLAS.

Original LAPACK is available at:
http://netlib.org/lapack/

Intel® Math Kernel Library: Fast Fourier Transforms (FFT)

- Mixed radix, multidimensional FFTs
- Supports user-defined scaling and transform sign
- Multiple transforms in a single call (batch)
- Supports data stride in input
- Supports FFTW* interfaces via wrappers
- Cluster FFTs
  - FFTs for distributed memory systems
  - Works with MPI
  - FFTW* support

Intel® Math Kernel Library: Sparse Solvers

PARDISO – Parallel Direct Sparse Solver	Support a wide range of matrix types.
	Based on BLAS level 3 update and pipelining parallelism.
	Supports out-of-core execution for huge problem sizes.
	Supports C-style 0-based indexing.
DSS – Direct Sparse Solver Interface for PARDISO	An alternative, simplified interface to PARDISO.
ISS – Iterative Sparse Solver	Symmetric positive definite: CG solver.
	Non-symmetric indefinite: Flexible generalized minimal residual solver.
	Based on Reverse Communication Interface (RCI).

Intel® MKL Vector Math Library (VML)

- Collection of vector math functions
- Real/Complex
- Double precision(DP), Single precision(SP)
- 3 accuracy modes
  - **H**igh **A**ccuracy, HA (correct rounding in >99% cases, behave according to C99; slowest, default mode)
  - **L**ow **A**ccuracy, LA (≤2 lsb incorrect, behave according to C99;30-50% faster than HA)
  - **E**nhanced **P**erformance, EP (~1/2 incorrect bits, is not guaranteed on entire domain;30-50% faster than LA)

MKL_VML_MODE

Real functions

Arithmetic	Power and Root	Exponential & Logarithmic	Trigonometric	Hyperbolic	Special	Rounding
Add	Inv	Exp	Sin	Sinh	Erf	Floor
Sub	Div	Expm1	Cos	Cosh	Erfc	Ceil
Sqr	Sqrt	Ln	Tan	Tanh	ErfInv	Trunc
Mul	InvSqrt	Log10	Asin	Asinh	ErfcInv	Round
Abs	Cbrt	Log1p	Acos	Acosh	CdfNorm	NearbyInt
LinearFrac	InvCbrt		Atan	Atanh	CdfNormInv	Rint
	Pow2o3		Atan2		LGamma	Modf
	Pow3o2		SinCos		TGamma	
	Pow					
	Powx					
	Hypot					

Complex functions

Arithmetic	Power and Root	Exponential & Logarithmic	Trigonometric	Hyperbolic
Add	Div	Exp	Sin	Sinh
Sub	Sqrt	Ln	Cos	Cosh
Mul	Pow	Log10	Tan	Tanh
MulByConj	Powx		Asin	Asinh
Conj			Acos	Acosh
Abs			Atan	Atanh
Arg			CIS	

Intel® Vector Statistical Library (VSL)

Random Number Generators (RNGs)	Pseudo-random, quasi-random, and non-deterministic generators
	Continuous and discrete distributions of various common distribution types
Summary Statistics (SS)	Parallelized algorithms for computation of statistical estimates for raw multi-dimensional datasets.
Convolution/correlation	A set of routines intended to perform linear convolution and correlation transformations for single and double precision real and complex data.

More Intel® Math Kernel Library Components

Data Fitting

- 1D linear, quadratic, cubic, step-wise const, and user-defined splines
- Spline based interpolation/extrapolation

PDEs (Partial Differential Equations)

- Solving Helmholtz, Poisson, and Laplace problems.

Optimization Solvers

- Solvers for nonlinear least square problems with/without constraints

Support Functions

- Memory management
- Threading control
- …

Using and Linking Intel MKL

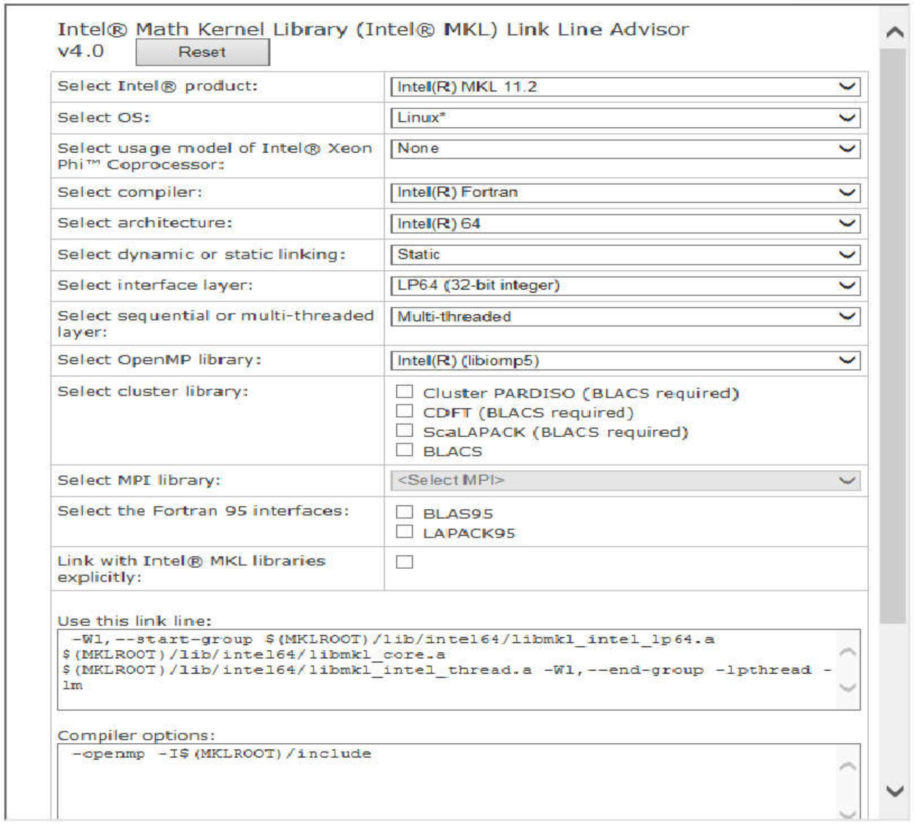
With the Intel Fortran or C/C++ compiler on Linux*, simply

ifort prog.f –mkl[:lib]

or

icc prog.c –mkl[:lib]

Linking with other compilers requires additional steps …



http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/

Parallelization in Intel® MKL

Domain	SIMD	Open MP	MPI
BLAS 1, 2, 3	X	X	
FFTs	X	X	
LAPACK **(dense LA solvers)**	X (relies on BLAS 3)	X	
ScaLAPACK **(cluster dense LA solvers)**	X	X (hybrid)	X
PARDISO **(sparse solver)**	X (relies on BLAS 3)	X	X
VML/VSL	X	X	
Cluster FFT	X	X	X

Intel® Math Kernel Library Environment

	Windows*	Linux*	Mac OS*
Compiler	Intel, CVF, Microsoft, PGI	Intel, GNU, PGI	Intel, GNU, PGI
Libraries	.lib, .dll	.a, .so	.a, .dylib

Language Support			
Domain	Fortran 77	Fortran 95/99	C/C++
BLAS	X	X	Via CBLAS
Sparse BLAS Level 1	X	X	Via CBLAS
Sparse BLAS level 1&2	X	X	X
LAPACK	X	X	X
ScaLAPACK	X		
PARDISO	X	X	X
DSS & ISS	X	X	X
VML/VSL/DF	X	X	X
FFT/Cluster FFT		X	X
PDEs		X	X
Optimization (TR) Solvers	X	X	X
SSL	X	X	X

Top New Features in Intel® MKL 11.x

- Support for Intel® Xeon Phi™ Coprocessor
  - Linux* hosted (11.0) and Windows* hosted (11.1)
- Conditional Numerical Reproducibility (CNR)
  - Support for unaligned data (11.1)
- Optimizations for Intel® AVX2 including FMA3
- Small Matrix Multiply enhancements (11.2)
  - MKL_DIRECT_CALL
  - Link with –DMKL_DIRECT_CALL or –DMKL_DIRECT_CALL_SEQ
- Early optimizations for Intel® AVX-512

- Extended Eigensolvers based on and compatible with FEAST[1]
- Parallel Direct Sparse Solver for Clusters (11.3)

Motivation for Conditional Reproducibility

Egineered to address issues that previously seemed to be unrelated or diffuse:

- Different results in consecutive runs, across different platforms, multi-threaded run

Ingredients and requirements:

Memory alignment	• Align memory — try Intel MKL memory allocation functions • 64-byte alignment for processors in the next few years
Number of threads	• Set the number of threads to a constant number • Use sequential libraries
Deterministic task scheduling	• Ensures that FP operations occur in order to ensure reproducible results
Code path control	• Maintains consistent code paths across processors • Will often mean lower performance on the latest processors

* Conditional (if possible, relaxed in future versions): across OS / bits / versions, varying # of threads, …

Conditional Numerical Reproducibility

- Deterministic multi-threading: MKL_CBWR_AUTO
  - Run-to-run reproducible results on the same processor; result is currently specific to the number of threads
  - Ordered execution and work-division; deterministic scheduling/reductions require Intel® OpenMP* runtime
  - Bitwise reproducible results
  - Auto-dispatched code path
- Across different processors: MKL_CBWR_[]
  - Code path needs to be selected according to instruction set extension that is commonly available to pool of systems
  - Number of threads is currently required to be the same across the pool of systems
  - Enables deterministic multi-threading

Intel® MKL 11.3 – Notable Enhancements

Optimized for the latest Intel® Xeon® processors and for Intel® Xeon Phi™ x200 coprocessor (KNL)

MKL memory manager detects and supports usage of MCDRAM via mkl_malloc

- mkl_set_memory_limit; export MKL_FASTMEMORY_LIMIT

Batch GEMM functions

- Improve the performance of multiple, simultaneous matrix multiply operations

Sparse BLAS inspector-executor API

- 2-stage API for Sparse BLAS (analyze and execute)

New counter-based pseudorandom number generator

– ARS-5 based on the Intel AES-NI instruction set and Philox4x32-10

Improved Intel® MKL PARDISO scalability

Cluster components extension

– MPI wrappers provide compatibility with most MPI implementations including custom ones
– Cluster components support on OS X*

New TBB threading Layer

Inspector-Executor Sparse BLAS API

Two-step API provides advanced sparse optimizations

1. Inspect step – analyze matrix to choose best strategy
   - Computational kernels for portrait
   - Balancing strategy for parallel execution
2. Execute step – use analysis data to get better performance
   - Optimization applied to get better performance
   - Level chosen based on expected number of iterations

Intel® Math Kernel Library and C++ Libraries

Several C++ template libraries available*

- **Armadillo**, **Eigen**, etc.

Typical criterions when deciding

- Use of expression templates to enable lazy evaluation and to avoid intermediate temporaries
- Data containers able to allocate aligned buffers and able to wrap existing memory layouts (user-allocated)
- Simple configuration (preprocessor symbols preferred) and compiler-agnostic (OS portable)

* http://software.intel.com/en-us/articles/intelr-mkl-and-c-template-libraries

Third-party Tools Powered by Intel® Math Kernel Library

IMSL* Fortran Numerical Libraries (Rogue Wave)

NAG* Libraries

MATLAB* (MathWorks)

GNU Octave*

NumPy* / SciPy*

PETSc* (Portable Extensible Toolkit for Scientific Computation)

WRF* (Weather Research & Forecasting run-time environment)

The HPCC* benchmark

And more …

Documentation

https://software.intel.com/en-us/intel-mkl-support/documentation

Reference Manual

http://software.intel.com/en-us/mkl_11.3_ref

User's Guide

http://software.intel.com/en-us/mkl_11.3_ug_lin

http://software.intel.com/en-us/mkl_11.3_ug_win

http://software.intel.com/en-us/mkl_11.3_ug_osx

Release Notes (good source of what's new)

https://software.intel.com/en-us/articles/intel-mkl-113-release-notes

https://software.intel.com/en-us/articles/intel-mkl-112-release-notes

http://software.intel.com/en-us/articles/intel-mkl-111-release-notes

http://software.intel.com/en-us/articles/intel-mkl-110-release-notes

Additional Resources

Intel® MKL product page: performance charts, licensing options, …

- http://software.intel.com/en-us/articles/intel-mkl

Documentation:

- https://software.intel.com/en-us/articles/intel-math-kernel-library-documentation

User forum:

- http://software.intel.com/en-us/forums/intel-math-kernel-library

MKL Link Line Adviser:

- http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor

Legal Disclaimer & Optimization Notice