



# Tools for Parallel Programming

# A Generic Development Cycle

## Analysis

- VTune™ Performance Analyzer

## Design (Introduce Threads)

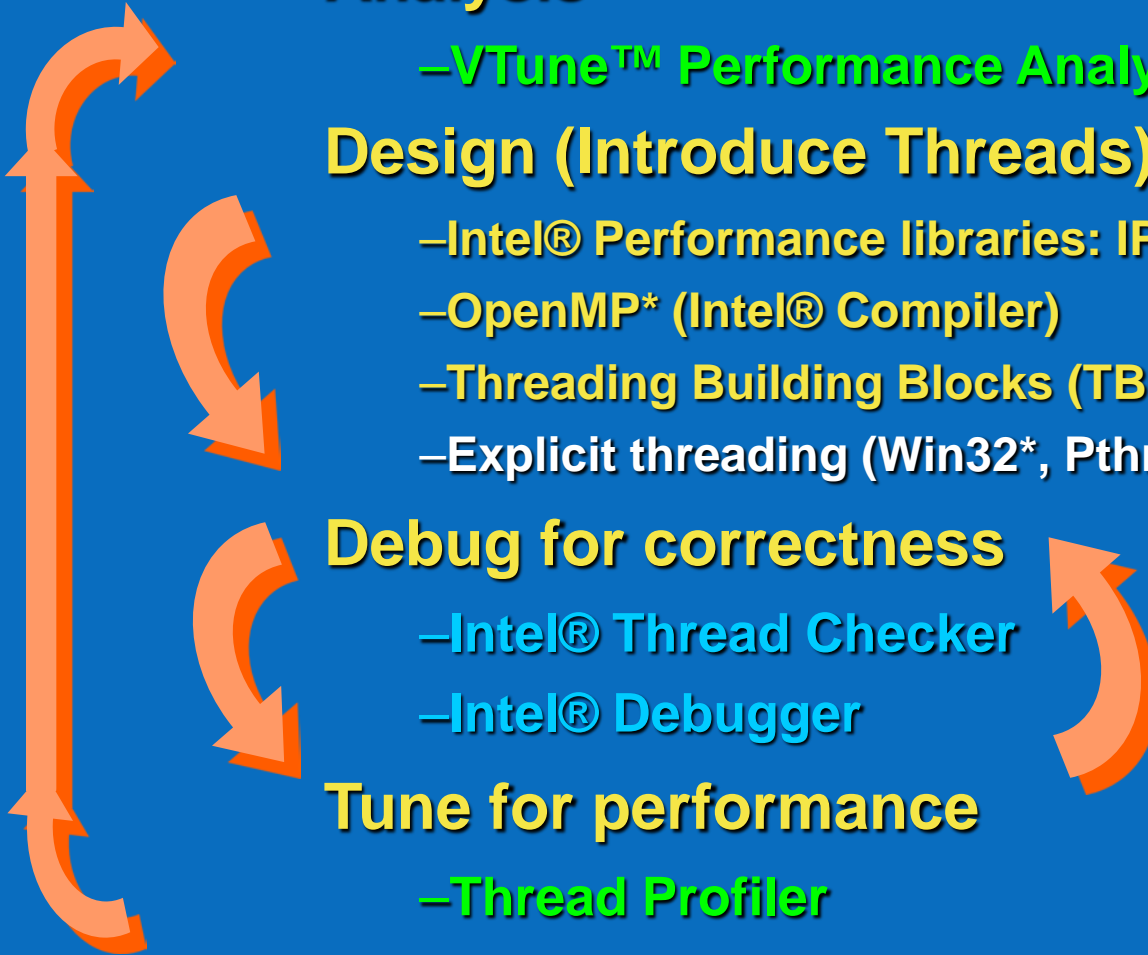
- Intel® Performance libraries: IPP and MKL
- OpenMP\* (Intel® Compiler)
- Threading Building Blocks (TBB)
- Explicit threading (Win32\*, Pthreads\*)

## Debug for correctness

- Intel® Thread Checker
- Intel® Debugger

## Tune for performance

- Thread Profiler
- VTune™ Performance Analyzer



# Where to start?



Compiler(/QParallel, ILP, Auto-Vectorization, SIMD)

Cilk, OpenMP

Ct

IPP, MKL (Optimized threaded libraries)

.NET & Java: Parallel FX, util.concurrency, TBB (C++)

Explicit threading (Win32\*, Pthreads\*)



## DESIGN

Gain insight on where parallelism will most benefit existing source code



## CODE & DEBUG

Develop effective applications with a C/C++ compiler and comprehensive threaded libraries



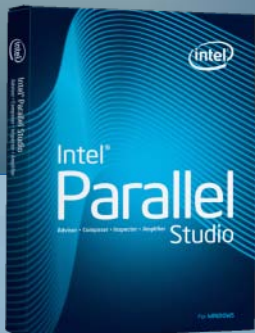
## VERIFY

Ensure application reliability with proactive parallel memory and threading error checking



## TUNE

Enhance applications with easy-to-use performance analyzer and tuner



# Intel® Threading Building Blocks

*Extend C++ for parallelism*

## Highlights

- A C++ runtime library that does thread management, letting developers focus on proven parallel patterns
- Appropriately scales to the number of HW threads available
- Supports nested parallelism
- The thread library API is portable across Linux, Windows, and Mac OS\* platforms. Open Source community extended support to FreeBSD\*, IA Solaris\* and XBox\* 360
- Run-time library provides optimal size thread pool, task granularity and performance oriented scheduling
  - Automatic load balancing through task stealing
  - Cache efficiency and memory reuse
- Committed to:
  - compiler independence
  - processor independence
  - OS independence

Both GPL and commercial licenses are available.

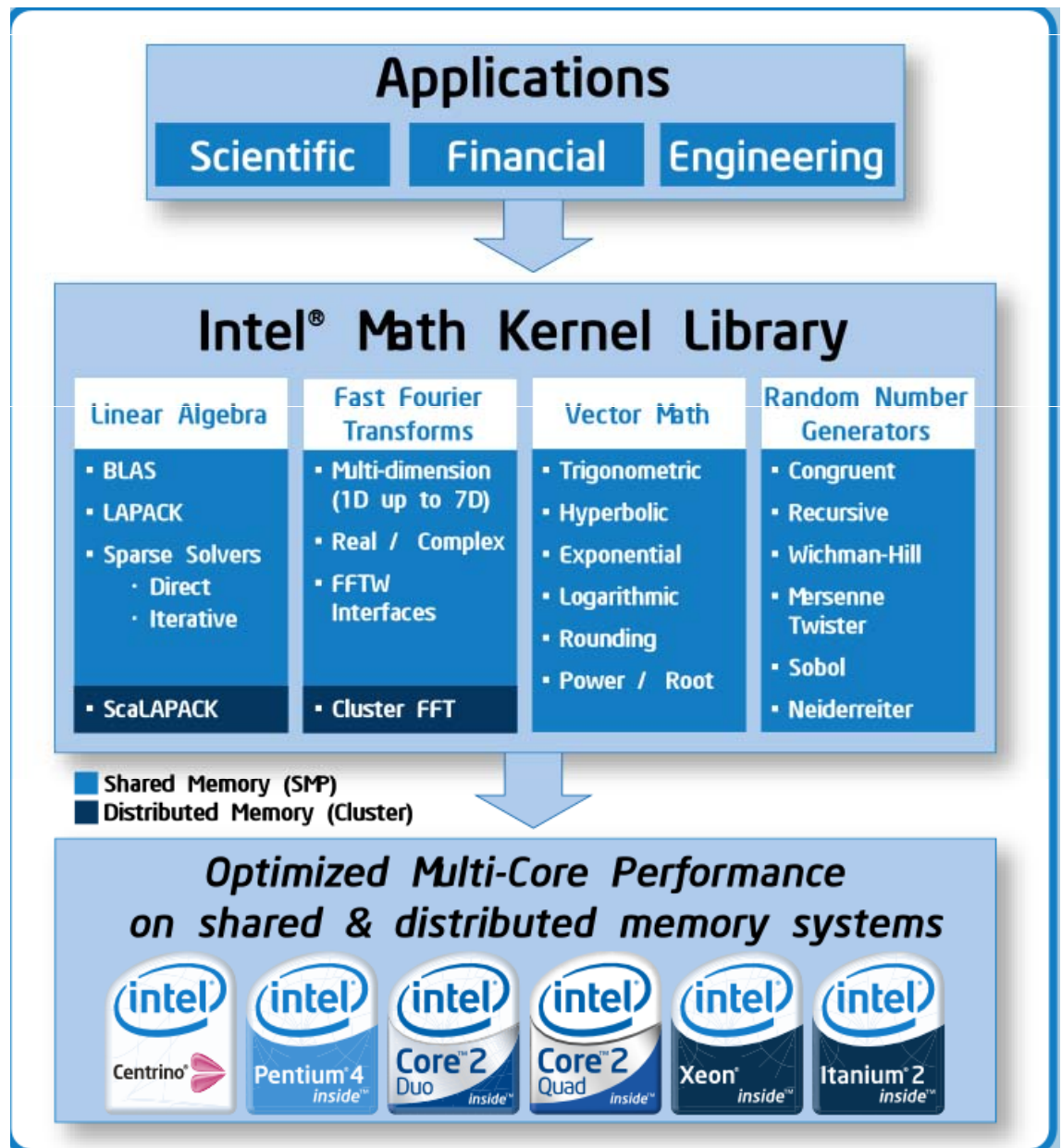
<http://threadingbuildingblocks.org>



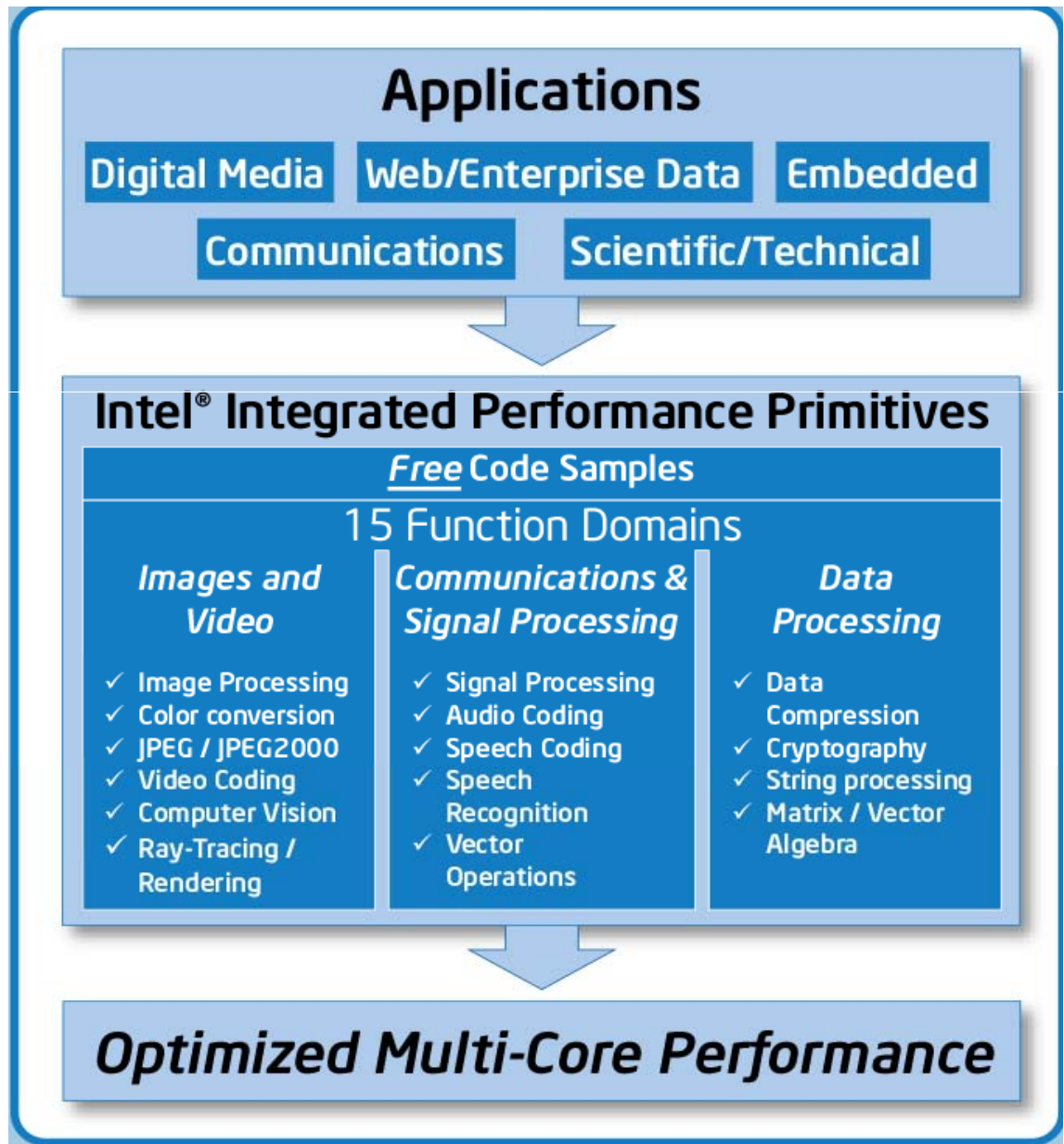
O'REILLY



# Intel® Math Kernel Library



# Intel® Integrated Performance Primitives



# Improved Cluster Performance with Intel® Cluster Toolkit 3.1 Compiler Edition

Source Code

## Intel® Cluster Toolkit 3.1 Compiler Edition

Intel® C++ & Fortran Compiler with Cluster Open MP for Intel® Compilers

*Optimization for the Node*

OpenMP\*,  
Auto-Parallelization, Vectorization,  
PGO, IPO & HPO Optimization

*Cluster OpenMP\* Intel®  
Compiler add-on*

Distributed memory version of  
OpenMP\*, known as Cluster OpenMP\*

*Intel® Debuggers*

Multi-node debug support

Intel® MPI Library

*Deployment on  
Multiple Fabrics*

Based on ANL MPICH2,  
Multiple fabric support,  
Automated fabric  
selection, Enhanced  
process pinning

Intel® Math Kernel  
Library

*Optimized Functions  
For Math Processing*

BLAS, LAPACK,  
Sparse Solvers,  
Fast Fourier Transforms,  
Vector Math,  
Statistics

ScalaPack, Cluster FFT

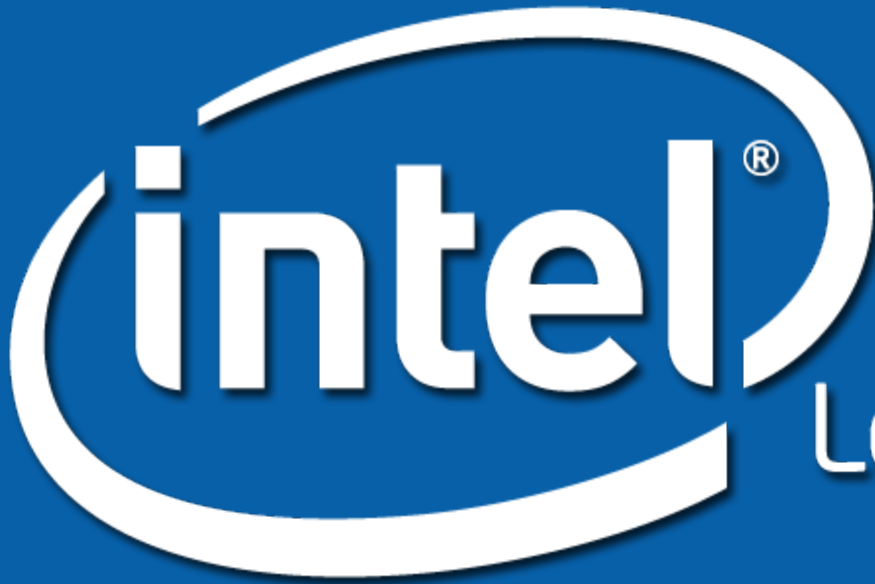
Intel® Trace Analyzer  
and Trace Collector

*The world's best  
analysis tool for  
MPI applications*

Increase productivity  
and cluster application  
performance, Very low  
impact, Excellent  
scalability on time and  
processors

*Highly Optimized  
MPI and Threaded Application Performance*





Leap ahead™

