Intel® Many Integrated Core Architecture

Klaus-Dieter Oertel
CERN, July 8th, 2011

intel

Legal Disclaimer

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information. Contact your local Intel sales office or your distributor to obtain the latest specification before placing your product order.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All products, dates, and figures are preliminary for planning purposes and are subject to change without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The Intel products discussed herein may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Knights Corner, Knights Ferry, Aubrey Isle and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.
Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's website at http://www.intel.com.

Intel, Xeon, Xeon Inside, Pentium and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Optimization Notice – Please Read

Intel® Compiler includes compiler options that optimize for instruction sets that are available in both Intel® and non-Intel microprocessors (for example SIMD instruction sets), but do not optimize equally for non-Intel microprocessors. In addition, certain compiler options for Intel® Compiler are reserved for Intel microprocessors. For a detailed description of these compiler options, including the instruction sets they implicate, please refer to "Intel® Compiler User and Reference Guides > Compiler Options." Many library routines that are part of Intel® Compiler are more highly optimized for Intel microprocessors than for other microprocessors. While the compilers and libraries in Intel® Compiler offer optimizations for both Intel and Intel-compatible microprocessors, depending on the options you select, your code and other factors, you likely will get extra performance on Intel microprocessors.

While the paragraph above describes the basic optimization approach for Intel® Compiler, with respect to Intel's compilers and associated libraries as a whole, Intel® Compiler may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (Intel® SSSE3) instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors.

Intel recommends that you evaluate other compilers to determine which best meet your requirements.

Intel® MIC Customer Value

Combine :

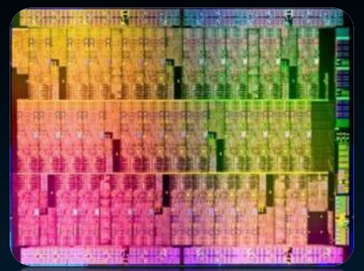
The many benefits of broad Intel CPU programming models, techniques, and familiar developer tools

+

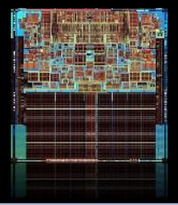
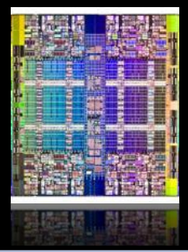
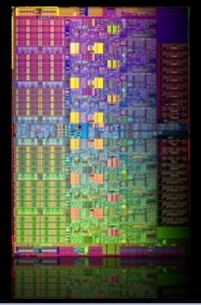
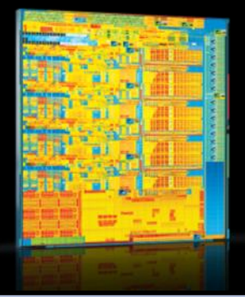
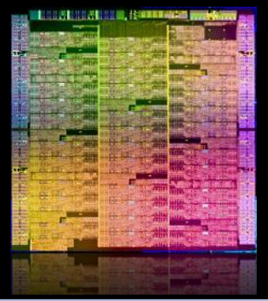
The compute density associated with specialty accelerators for parallel workloads

=

Intel® Many Integrated Core Products



Intel and Parallelism

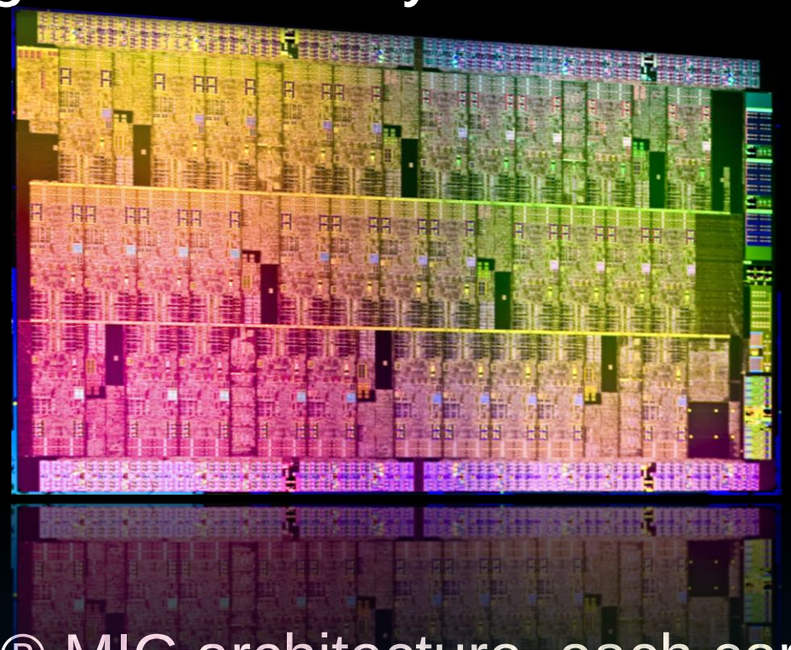
Images not intended to reflect actual die sizes

	64-bit Intel® Xeon® processor	Intel® Xeon® processor 5100 series	Intel® Xeon® processor 5500 series	Intel® Xeon® processor 5600 series	Sandy Bridge	Aubrey Isle (in Knights Ferry)
Frequency	3.6GHz	3.0GHz	3.2GHz	3.3GHz	Not Announced	1.2GHz
Core(s)	1	2	4	6	8	32
Thread(s)	2	2	8	12	16	128
SIMD Width	128 (2 clock)	128 (1 clock)	128 (1 clock)	128 (1 clock)	256 (1 clock)	512 (1 clock)

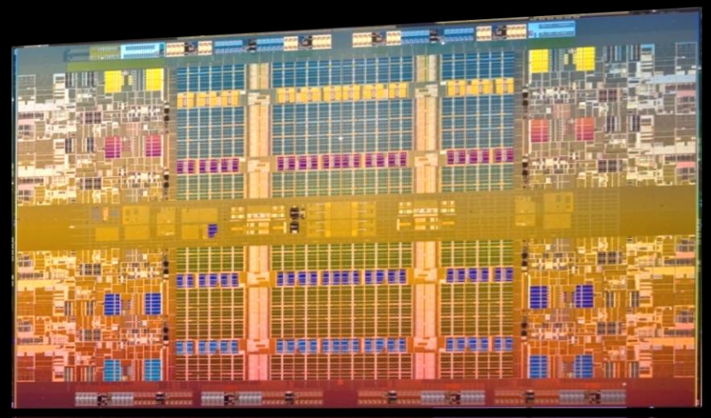
Intel® MIC builds on established CPU architecture and programming concepts - providing the benefits of code re-use to developers of highly parallel applications

Many Core and Multi-Core

Many Integrated Core Aubrey Isle at 1-1.2 GHz



Multi-core Intel® Xeon® processor at 2.26-3.5 GHz



Die Size not to scale

In Intel® MIC architecture, each core is smaller, has lower power limit, has lower single thread performance, but higher aggregate performance

Many core relies on a high degree of parallelism to compensate for the lower speed of each individual core

Relatively few specialized applications today are highly parallel, but those applications can benefit from Intel® MIC architecture

The “Knights” Family

Future Knights Products

Knights Corner

1st Intel® MIC product

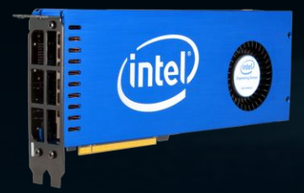
22nm process

>50 Intel Architecture Cores

PCIe

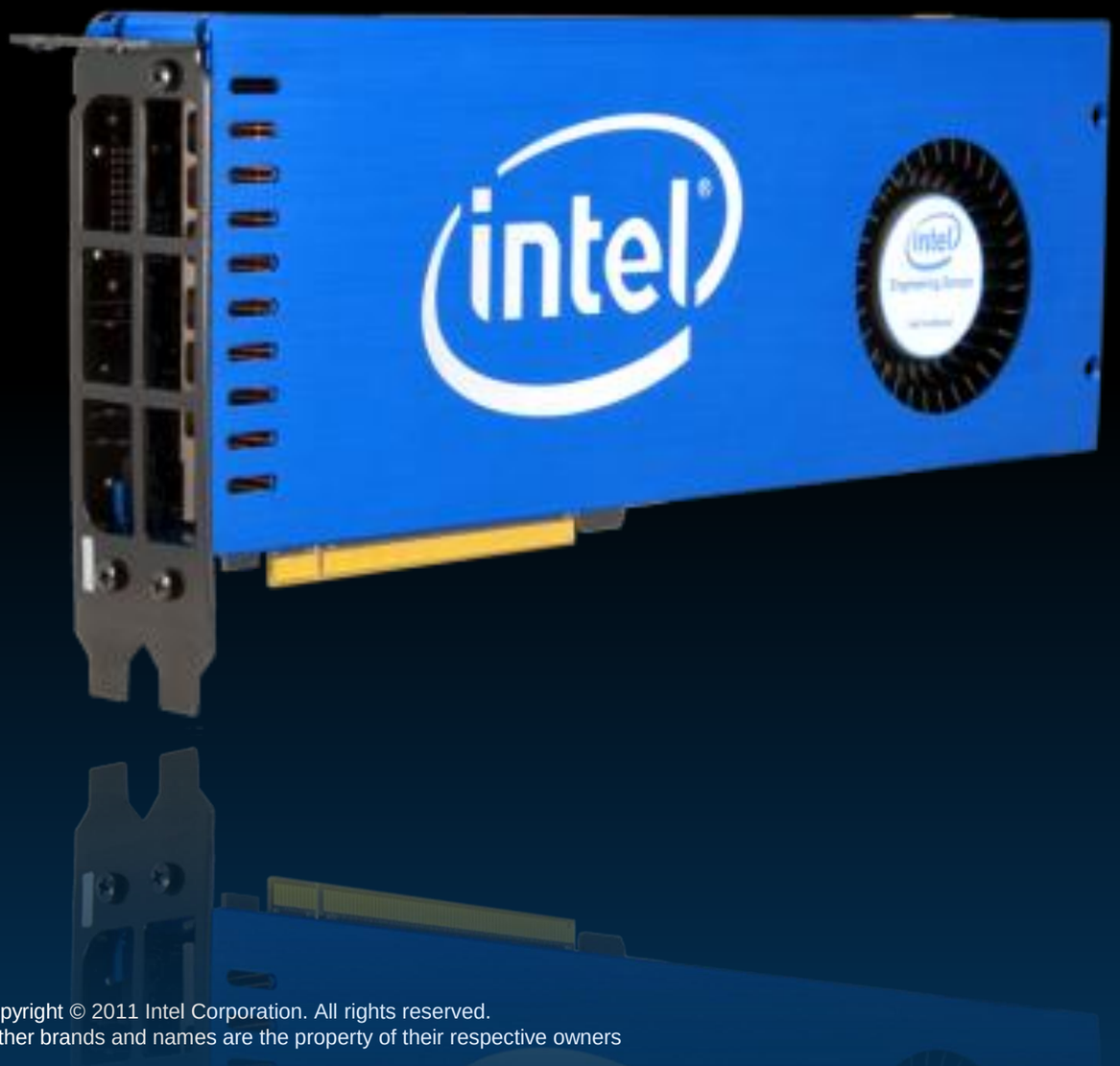
Knights Ferry

Software Development Platform



Future options subject to change without notice.

“Knights Ferry” Software Development Platform



Software Development Platform

Growing availability through 2011

Aubrey Isle Co-Processor

Up to 32 cores, up to 1.2 GHz

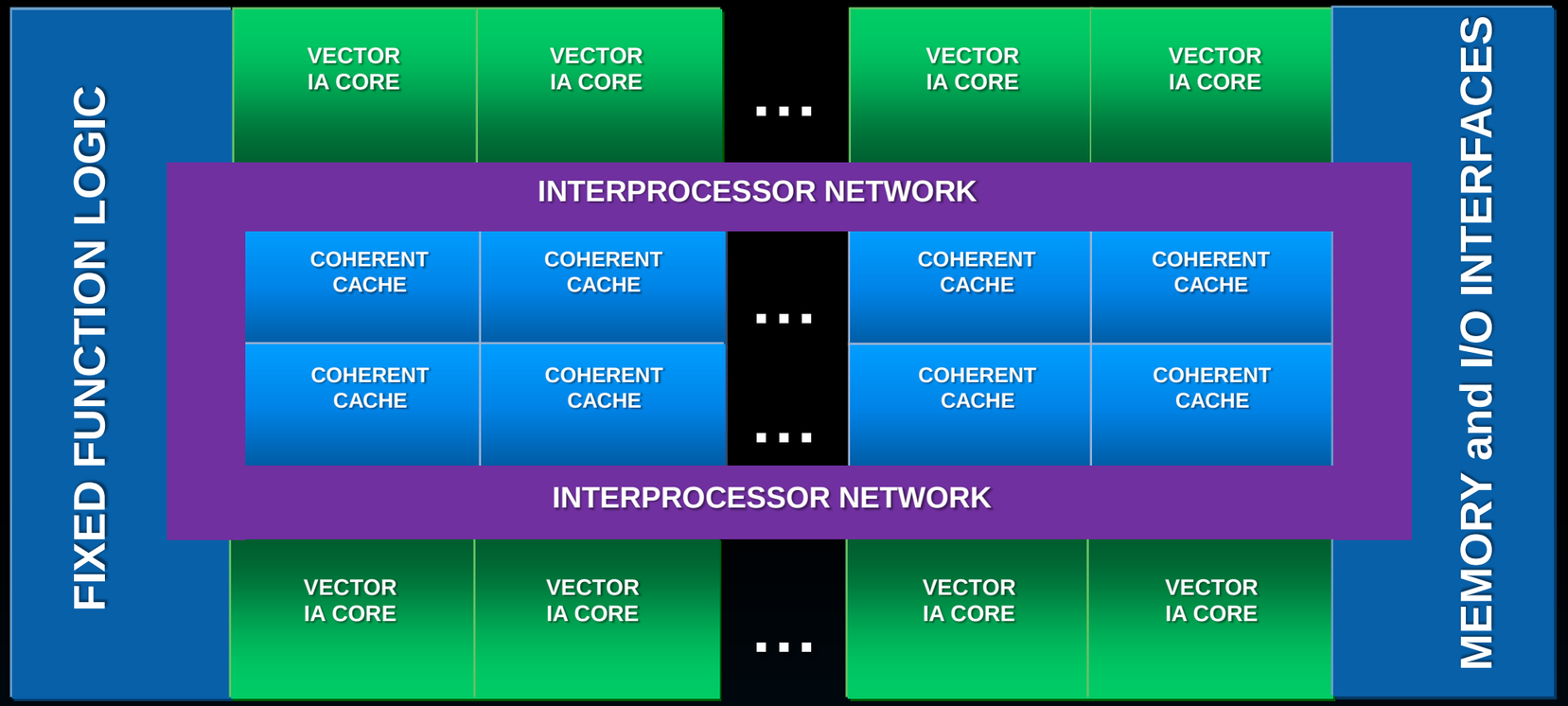
Up to 128 threads at 4 threads / core

Up to 8MB shared coherent cache

Up to 2 GB GDDR5

Bundled with Intel HPC SW tools

Aubrey Isle Co-Processor Architecture



Multiple x86 cores
- In-order, short pipeline
- Multi-thread support

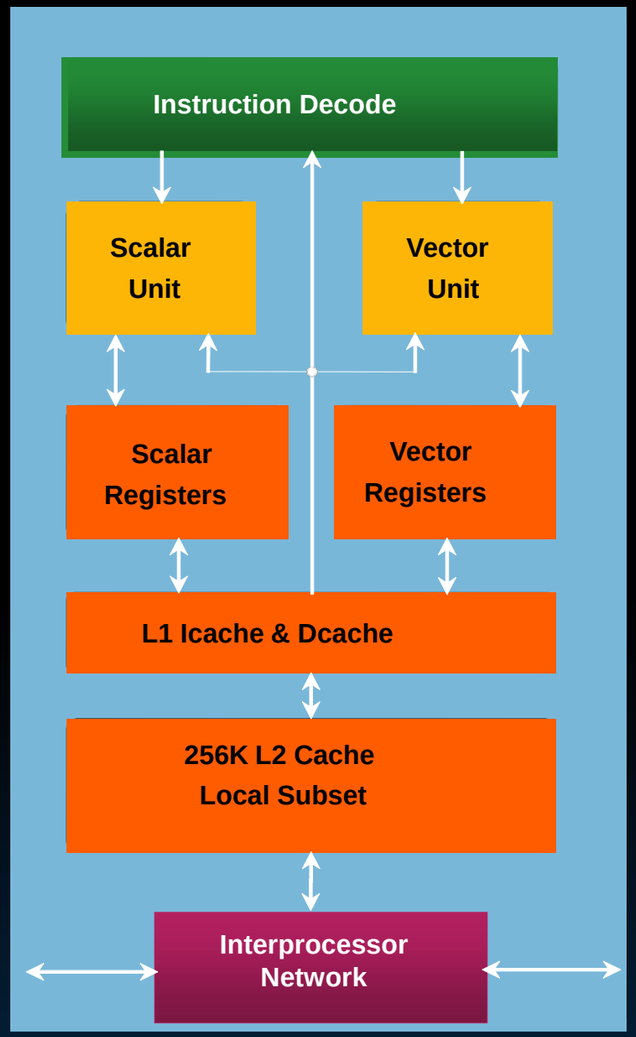
16-wide vector units (512b)
Extended instruction set
Fully coherent caches

1024-bit ring bus
GDDR5 memory
Supports virtual memory

Standard Intel Architecture Programming and Memory Model

For illustration only.
Future options subject to change without notice.

Aubrey Isle Core



The Aubrey Isle co-processor core:

- Scalar pipeline derived from the dual-issue Intel® Pentium® processor
- Short execution pipeline
- Significant modern enhancements such as multi-threading, 64-bit extensions, and sophisticated pre-fetching.
- 4 execution threads per core
- Separate register sets per thread
- Supports IEEE standards for floating point arithmetic
- Fully coherent cache structure
- Fast access to its 256KB local subset of a coherent L2 cache.
- 32KB instruction cache per core, 32KB data cache for each core.

Enhanced x86 instructions set with:

- Over 100 new instructions,
- Wide vector processing operations
- 3-operand, 16-wide vector processing unit (VPU)
- VPU executes integer, single-precision float, and double precision float instructions

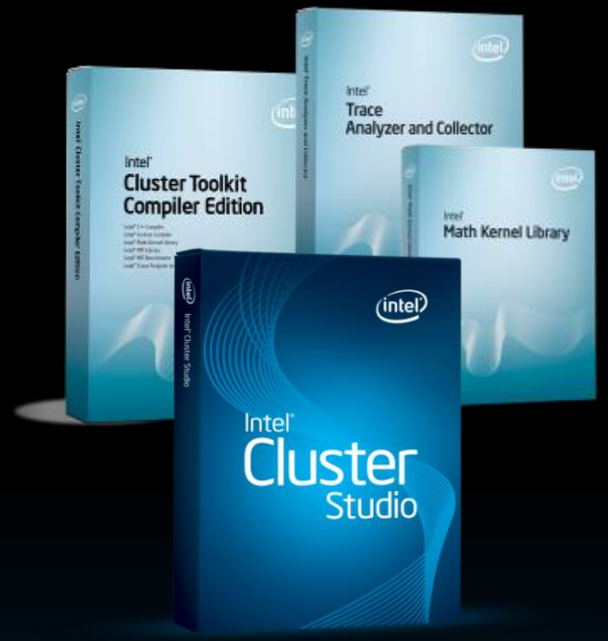
Interprocessor Network

1024 bits wide, bi-directional (512 bits in each direction)

Each Co-Processor core executes its own instructions enabling complex programs including branches and recursion

Intel Development Tools extend to Intel® MIC

Leading developer tools for performance on nodes and clusters



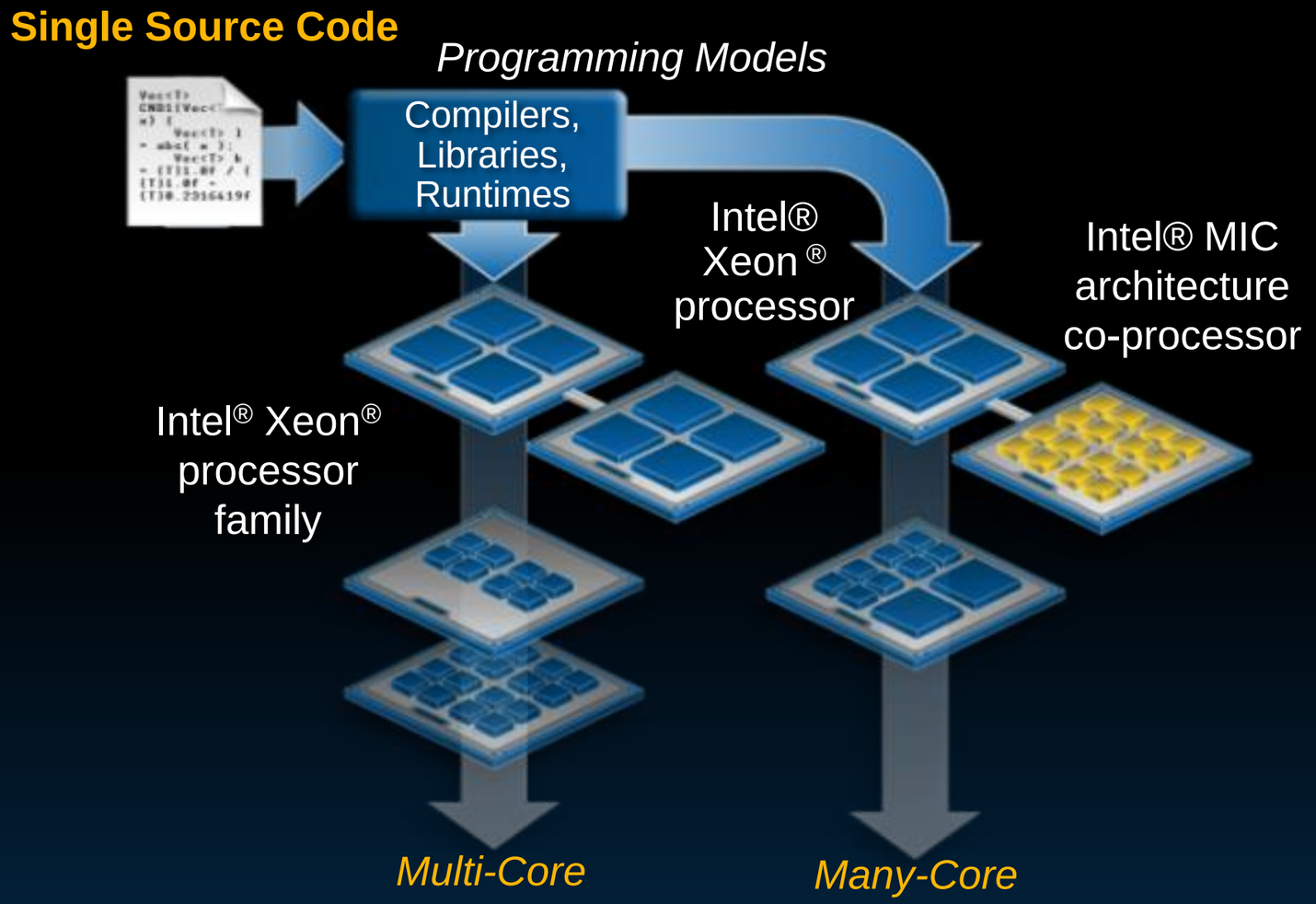
Advanced Performance

C++ and Fortran Compilers, MKL/IPP Libraries & Analysis Tools for Windows*, Linux* developers on
IA based multi-core node

Distributed Performance

MPI Cluster Tools with C++ and Fortran Compiler, MKL Libraries and Analysis Tools for Windows*, Linux* developers on
IA based clusters

Intel® MIC Architecture Programming



Common with Intel® Xeon® processors

- Programming Models
- C/C++, Fortran compilers
- Intel SW developer tools and libraries (MKL, IPP, TBB, ArBB, …)
- Coding and optimization techniques and SW tools
- Ecosystem support

Eliminates much of the need for Dual Programming Architecture

For illustration only, potential future options subject to change without notice.

Example: Computing PI

```
# define NSET 1000000
int main ( int argc, const char** argv )
{  long int i;
   float num_inside, Pi;
   num_inside = 0.0f;
#pragma offload target (MIC)
#pragma omp parallel for reduction(+:num_inside)
   for( i = 0; i < NSET; i++ )
           {         float x, y, distance_from_zero;
                     // Generate x, y random numbers in [0,1)
                     x = float(rand()) / float(RAND_MAX + 1);
                     y = float(rand()) / float(RAND_MAX + 1);
                     distance_from_zero = sqrt(x*x + y*y);
                     if ( distance_from_zero <= 1.0f )
                     num_inside += 1.0f;
            }
    Pi = 4.0f * ( num_inside / NSET );
    printf("Value of Pi = %f \n",Pi);
}
```

One additional line from the CPU version

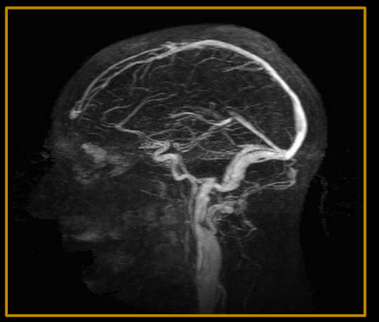
(For illustration only)

Progress to date

Committed and announced roadmap. Demonstrated ability to meet or beat graphics accelerator performance

May'10

Publicly demonstrated first complex parallel applications running on Intel® MIC



September'10

November'10

```
double option_price_call_black_scholes(
                  double S,       // spot (underlying) price
                  double K,       // strike (exercise) price,
                  double r,       // interest rate
                  double sigma,   // volatility
                  double time)    // time to maturity
{
    double time_sqrt = sqrt(time);
    double d1 = (log(S/K)+r*time)/(sigma*time_sqrt)+0.5*sigma*time_sqrt;
    double d2 = d1-(sigma*time_sqrt);
    return S*N(d1) - K*exp(-r*time)*N(d2);
}
```

Demonstrated C++ source code for both Intel® Xeon® and Intel® MIC (no hand code)

ISC 2011: Optimized SDP Performance

Hybrid LU Factorizarion

Leverages compute power of both Intel® Xeon® CPUs and Intel® MIC
Delivers optimal performance by dynamically balancing large and small matrix Computations between Intel® Xeon® and Intel® MIC

Up to 772 GFLOP

Hybrid Computing – SGEMM with Intel® MKL

High performing SGEMM with just 18 lines of code – common between Intel® Xeon® CPUs and Knights Ferry
Uses Intel® MKL in current version of Alpha stack/tools on Knights Ferry

1+ TFLOP

7.4 TFLOP SGEMM in a node

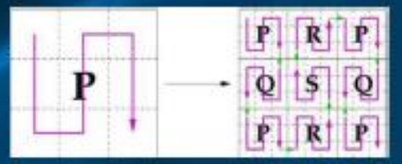
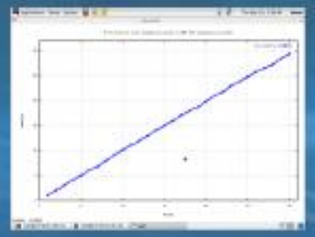
Simultaneous execution of SGEMM on 8 Knights Ferry cards to deliver 7.4 TFLOPS in 1 4U server

7.4 TFLOP

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. Source: Intel measured results as of March 2011. See backup for details.
For more information go to http://www.intel.com/performance

[1] Refer to backup material for system configurations

ISC 2011: Programmability For HPC Applications

Forschungszentrum Juelich	SMMP Protein Folding		Simulates the folding process of proteins to reach their final shape after they are produced by a cell
KISTI	Molecular Dynamics		Empirical-potential molecular dynamics, widely used for simulating nano-materials including carbon nanotube, graphene, fullerene, and silicon surfaces
LRZ	TifaMMy Matrix Multiplication		Cache-oblivious implementation of matrix-matrix multiply which uses a recursive scheme to partition input data for computation and parallelization
CERN	Core Scaling of Intel® MIC Architecture		Benchmark kernel extracted from the CBM/ALICE HLT software development for collider experiments. It estimates real trajectories from imprecise measurements

What customers are saying

"We see the Intel MIC processor line as an exciting leap forward, and we are ecstatic about working with Intel to explore application performance on this new platform" *(4/21/11)*

"Moving code to MIC might involve sitting down and adding a couple of lines of directives that takes a few minutes. Moving a code to a GPU is a project" *(4/21/11)*

Dan Stanzione, Deputy Director at TACC

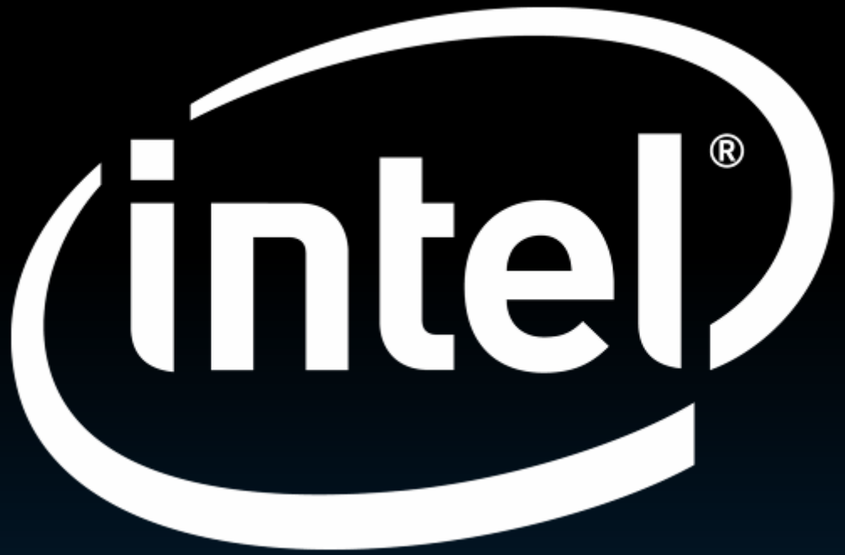
"The CERN openlab team was able to migrate a complex C++ parallel benchmark to the Intel MIC software development platform in just a few days" *(5/31/10)*

Sverre Jarp, CTO of the CERN openlab

Intel is engaged with a wide variety of ecosystem partners

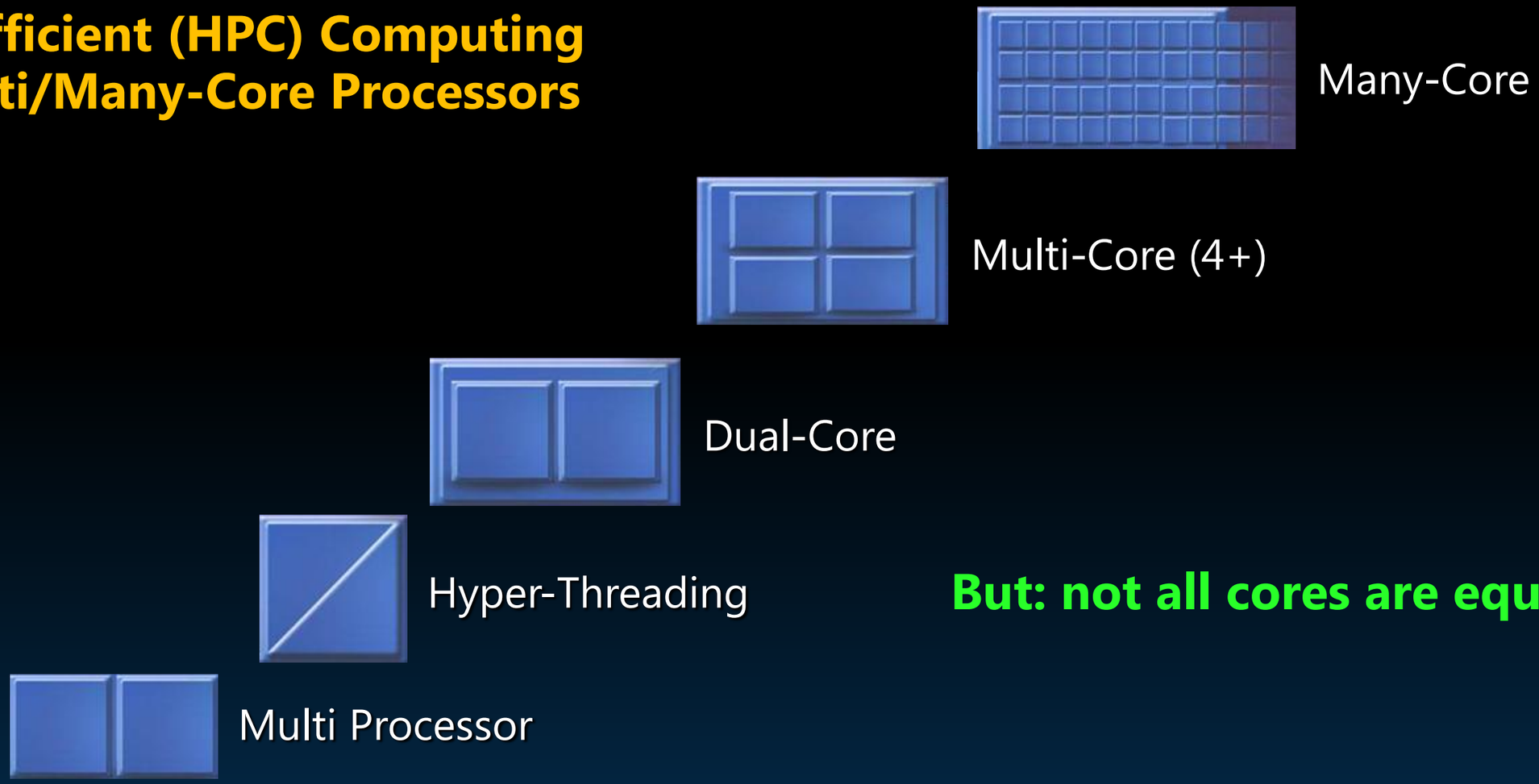
Call to Action

- Optimize for Multi-Core today
  - Use Intel's industry leading tools – C/C++/Fortran compilers, performance libraries, threading and performance analysis tools, cluster tools with Intel® MPI
  - Scale with increasing number of cores – 4S x 8 cores, 8S x 8 cores
  - Use vectorization to exploit benefits of SIMD
- Extend to Intel® Many Integrated Core Architecture


intel

Industry Trend to Multi/Many-Core

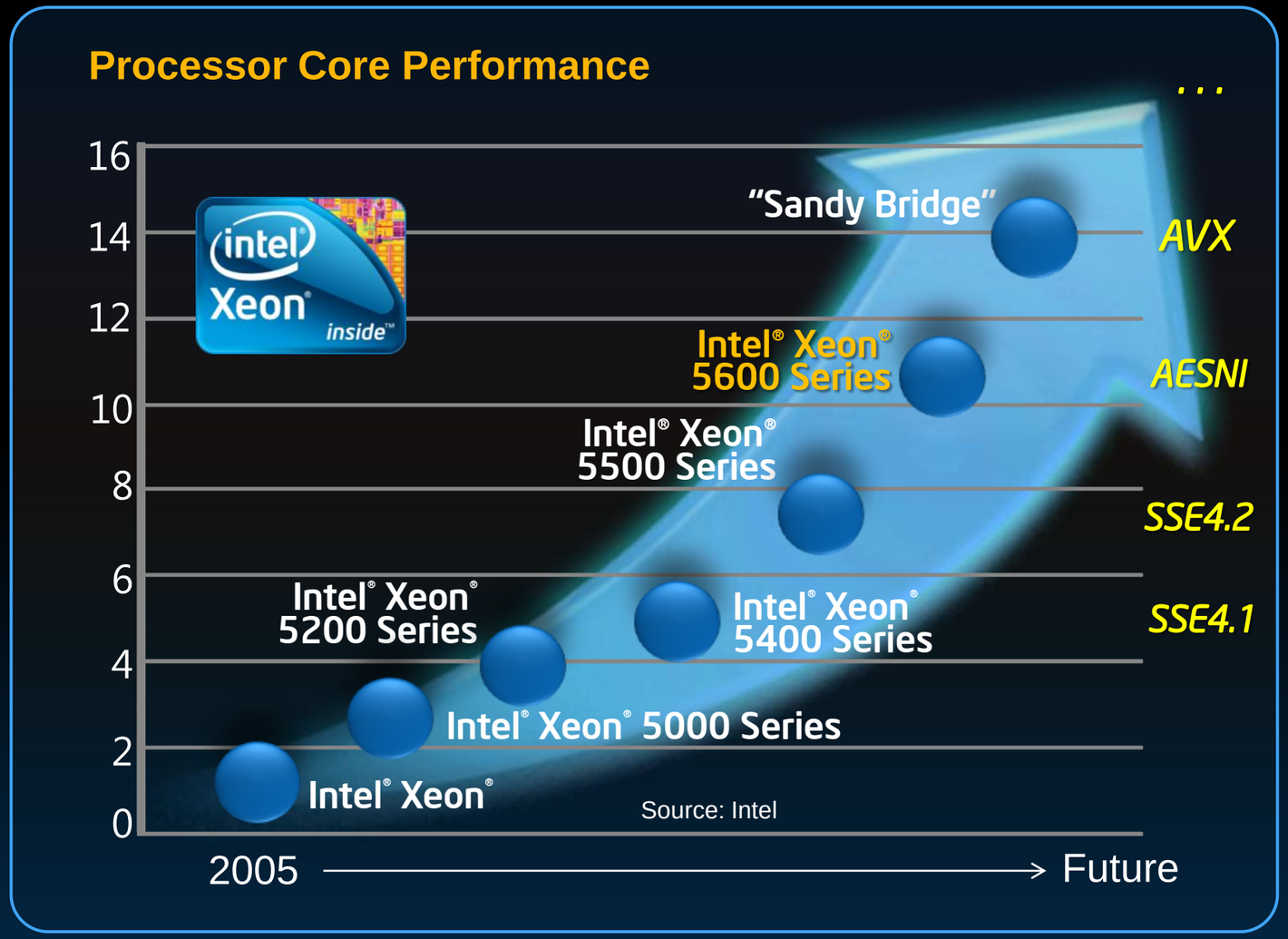
Energy Efficient (HPC) Computing with Multi/Many-Core Processors



But: not all cores are equal !

(for illustration only)

Intelligent Processor Performance Scaling Forward



Faster Time To Productivity

Total Application Performance

Increased Single Thread Performance

Increased Floating Point Performance and Bandwidth

Irregular Data-Access

Balanced Processor and System Architecture

Less Complex Software Development and Support

Potential future options, subject to change without notice.