Programming Models for Intel® Xeon® processors and Intel® Many Integrated Core (Intel MIC) Architecture

Scott McMillan
Senior Software Engineer
Software & Services Group

April 11, 2012
TACC-Intel Highly Parallel Computing Symposium
Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

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. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/software/products.


*Other names and brands may be claimed as the property of others.

Copyright © 2012. Intel Corporation.

http://intel.com/software/products
Intel’s compilers 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 SSE2®, SSE3, and 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. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804
Intel® Xeon® processor

Foundation of HPC Performance
Suited for full scope of workloads

Industry leading performance and performance/watt for serial & parallel workloads

Focus on fast single core/thread performance with “moderate” number of cores

Intel® MIC Co-processor

Performance and performance/watt optimized for highly parallelized compute intensive workloads

Common software tools with Xeon enabling efficient application readiness and performance tuning

IA extension to Many-Core

Lots of cores/threads with wide SIMD

[die sizes not to scale]
C/C++, FORTRAN  
OpenMP, MPI, ...

Same Comprehensive Set of SW Tools  
Established HPC Operating System

Application Source Code Builds with a Compiler Switch

Intel® Xeon® Processor

Intel® MIC Co-processor
Single-source approach to Multi- and Many-Core

Eliminates Need to Fork Application Code
The “Knights” Family

Knights Corner

1st Intel® MIC product
22nm process
>50 Intel Architecture cores
TFLOPS of Performance
Energy Efficient
Offload Co-Processor and
Native Linux® Node Programming

“Programmed like a computer”

All dates, product descriptions, availability, and plans are forecasts and subject to change without notice.
Operating Environment View

Intel® Xeon® processor

Knights Corner

- Linux Standard Base
- IP
- SSH
- NFS

A flexible, familiar, compatible operating environment
Intel® MIC Co-processor Becomes a Network Node

Intel® Xeon® Processor

Intel® MIC Co-processor

Virtual Network Connection

Intel® MIC Architecture + Linux enables IP addressability
Spectrum of Programming Models and Mindsets

Multi-Core Centric
- Xeon
  - Multi-Core Hosted
    - General purpose serial and parallel computing
      - Offload
        - Codes with highly-parallel phases
          - Main()
            - Foo()
            - MPI_*( )
          - Foo()
            - Main()
            - MPI_*( )
          - Main()
            - Foo()
            - MPI_*( )

Symmetric
- Codes with balanced needs
  - Main()
    - Foo()
    - MPI_*( )

Many-Core Centric
- MIC
  - Many Core Hosted
    - Highly-parallel codes
      - Main()
        - Foo()
        - MPI_*( )

Range of models to meet application needs
Programming Intel® MIC-based Systems

**MPI+Offload**

- MPI ranks on Intel® Xeon® processors (only)
- All messages into/out of processors
- Offload models used to accelerate MPI ranks
- Intel® Cilk™ Plus, OpenMP*, Intel® Threading Building Blocks, Pthreads* within Intel® MIC
- Homogenous network of hybrid nodes:
Offload Code Examples

- **C/C++ OffloadPragma**
  
  ```c
  #pragma offload target (mic)
  #pragma omp parallel for reduction(+:pi)
  for (i=0; i<count; i++) {
    float t = (float)((i+0.5)/count);
    pi += 4.0/(1.0+t*t);
  }
  pi /= count;
  ```

- **Function Offload Example**
  
  ```c
  #pragma offload target(mic)
  in(transa, transb, N, alpha, beta) \ 
  in(A:length(matrix_elements)) \ 
  in(B:length(matrix_elements)) \ 
  inout(C:length(matrix_elements))
  sgemm(&transa, &transb, &N, &N, &N, &alpha, A, &N, B, &N, &beta, C, &N);
  ```

- **Fortran Offload Directive**
  
  ```fortran
  !dir$omp offload target(mic)
  !$omp parallel do
    do i=1,10
      A(i) = B(i) * C(i)
    enddo
  ```

- **C/C++ Language Extension**
  
  ```c
  class _Cilk_Shared common {
    int data1;
    char *data2;
    class common *next;
    void process();
  };

  _Cilk_Shared class common obj1, obj2;
  _Cilk_spawn _Offload obj1.process();
  _Cilk_spawn obj2.process();
  ```
Programming Intel® MIC-based Systems

Many-core Hosted

- MPI ranks on Intel® MIC (only)
- All messages into/out of Intel® MIC
- Intel® Cilk™ Plus, OpenMP*, Intel® Threading Building Blocks, Pthreads used directly within MPI processes
- Programmed as homogenous network of many-core CPUs:
Programming Intel® MIC-based Systems

Symmetric

- MPI ranks on Intel® MIC and Intel® Xeon® processors
- Messages to/from any core
- Intel® Cilk™ Plus, OpenMP*, Intel® Threading Building Blocks, Pthreads* used directly within MPI processes
- Programmed as heterogeneous network of homogeneous nodes:
Keys to Productive Performance on Intel® MIC Architecture

• Choose the right Multi-core centric or Many-core centric model for your application

• Vectorize your application (today)
  – Use the Intel vectorizing compiler

• Parallelize your application (today)
  – With MPI (or other multi-process model)
  – With threads (via Intel® Cilk™ Plus, OpenMP®, Intel® Threading Building Blocks, Pthreads, etc.)

• Go asynchronous to overlap computation and communication
Options for Thread Parallelism

- Intel® Math Kernel Library
- Intel® Threading Building Blocks
  - Intel® Cilk™ Plus
- OpenMP*
- Pthreads* and other threading libraries

Ease of use / code maintainability

Programmer control
Options for Vectorization

Intel® Math Kernel Library

Array Notation: Intel® Cilk™ Plus

Automatic vectorization

Semiautomatic vectorization with annotation:
#pragma vector, #pragma ivdep, and #pragma simd

C/C++ Vector Classes (F32vec16, F64vec8)

Vector intrinsics (mm_add_ps, addps)

Ease of use / code maintainability (depends on problem)

Programmer control
Invest in Common Tools and Programming Models

Multicore

Intel® Xeon® processors are designed for intelligent performance and smart energy efficiency

Continuing to advance Intel® Xeon® processor family and instruction set (e.g., Intel® AVX, etc.)

Your Application

Many-core

Intel® MIC Architecture - co-processors are ideal for highly parallel computing applications

Software development platforms ramping now

Use One Software Architecture Today. Scale Forward Tomorrow.

Use One Software Architecture Today. Scale Forward Tomorrow.
Summary

• Intel® MIC Architecture offers familiar and flexible programming models

• Hybrid MPI/threading is becoming increasingly important as core counts grow

• Intel tools support hybrid programming today, exploiting existing standards

• Hybrid parallelism on Intel® Xeon® processors + Intel® MIC delivers superior productivity through code reuse

• Hybrid programming today on Intel® Xeon® processors readies you for Intel® MIC