Early Application Experiences with the Intel® MIC Architecture

R. Glenn Brook

glenn-brook@tennessee.edu

Electronic Structure Calculation Methods on Accelerators

Workshop – Oak Ridge, TN – February 6 - 8, 2012
Application Acceleration Center of Excellence (AACE)
Joint Institute for Computational Sciences
University of Tennessee & ORNL

- Established early in 2011 to investigate the application of future computing technologies to simulation in science and engineering
- An essential element of a sustainable software infrastructure for scientific computing
- Director: Glenn Brook
AACE — Mission

• To prepare the national supercomputing community to effectively and efficiently utilize future supercomputing architectures
  – To optimize applications for current and future compute systems
  – To develop expertise in the expression and exploitation of fine-grain and medium-grain parallelism
  – To conduct research and education programs focused on developing and transferring knowledge related to emerging computing technologies
  – To provide expert feedback to HPC vendors to guide the development of future supercomputing architectures and programming models
NICS-Intel Strategic Engagement

• Multi-year agreement with Intel to jointly pursue:
  – Development of next-generation, HPC solutions based on the Intel Many Integrated Core (MIC) architecture
  – Design of scientific applications emphasizing a sustainable approach for both performance and productivity

• NICS receives early access to Intel technologies and provides application testing, performance results, and expert feedback
  – Help guide further development efforts by Intel
  – Help prepare the scientific community to use future HPC technologies immediately upon their deployment
Hardware Resources

• Rook
  – Intel MIC “Knights Ferry” Software Development Platform
  – Workstation – 2 Westmere CPUs & 2 KNF MICs

• Bishop
  – Cray CX1 cluster – 6 Westmere CPUs & 2 KNF MICs
    • 1 Head node – 2 Westmere CPUs
    • 2 Compute nodes – 2 Westmere CPUs & 1 KNF MIC (per node)

• Knight
  – Appro cluster – 8 Westmere CPUs & 8 KNF MICs
    • 4 Compute nodes - 2 Westmere CPUs & 2 KNF MIC (per node)

• Beacon – coming soon
  – Appro cluster – 2 service nodes & up to 32 compute nodes
    • up to 64 Sandybridge CPUs & up to 64 KNF MICs
Chemistry on the Intel® MIC Architecture

W. Scott Thornton & Robert J. Harrison
Joint Institute for Computational Sciences
University of Tennessee
NWChem – www.nwchem-sw.org

- Premier massively parallel chemistry code
- Molecules, biomol., nanostructures, and solid-state
- From quantum to classical, and all combinations
- Used on supercomputers world wide
- Scaling to 100K processors
- One-sided communication using Global array library
- Several million lines of code (Fortran, C), open source
NWChem Runs on Intel® MIC

- In stand-alone, native mode
  - MIC as independent 64-bit Linux system
  - Internal to MIC using MPI and Global Arrays

- Initial successes and functionality
  - No significant compilation issues
  - Configured as standard x86-64 target
  - All code seems to be working
    - MM/QM/MD, 2-integrals, Gaussian DFT, many-body, plane-wave, Python interface, ...

- Only a tiny fraction of NWChem capability is running on any other accelerator
Status and work in progress

• Status
  – Linked against MKL
  – Initial tests perfect, full QA suite not yet run
  – Global Arrays not yet tested in parallel

• In progress
  – Run full QA suite
  – Test Global Arrays in parallel
    • Within MIC, between multiple MICs, between MIC and host
  – Optimize performance
    • Vectorize two-electron integrals using compiler & intrinsics
    • Vectorize DFT quadrature using vector math library
    • Many-body methods using parallel BLAS
Electronic Structure

Paramount to extracting functionality from advanced materials is having a detailed understanding of their electronic, magnetic, vibrational, and optical properties.

**ELK** is a software platform which allows for the understanding of these properties from a first principles approach. It employs electronic structure techniques such as *density functional theory*, *Hartree-Fock theory*, and *Green’s function theory* to calculate relevant properties from first principles.

*Antiferromagnetic structure of Sr$_2$CuO$_3$*
Elk FP-LAPW Code

Background

A electronic structure package with an international user base that includes researchers in electronic, magnetic, and optical properties of materials.

- Fortran 90
- OpenMP parallelization
- Message-passing parallelization
- Efficient hybrid MPI + OpenMP parallelization

Porting

1. Compiled with an additional flag to the Intel Fortran compiler (ifort).
2. Copied executable, OpenMP shared library, and data files to the Intel MIC.
3. Executed binary natively on the Intel MIC.

No code changes were necessary.
Elk uses master-slave parallelism where orbitals for different momenta are computed semi-independently. In this test 27 different crystal momenta were used to model bulk silicon with parallel efficiency over 80%.
ENZO Cosmology on the Intel® MIC Architecture

Robert Harkness
National Institute for Computational Sciences
The ENZO Astrophysics Code

- ENZO is a community code for computational astrophysics and cosmology
  - Versatile code for many applications
  - Powerful adaptive mesh refinement
  - Hydrodynamics, chemistry, RHD and MHD
  - Hybrid mesh and particle code (for dark matter cosmology)
  - Large user base and Active development

- UCSD Lab for Computational Astrophysics runs extremely large-scale ENZO applications focused on structure formation in the Early Universe.
  - NSF and NICS support for runs on NICS Cray XT5 ‘Kraken’
  - DOE 2011 INCITE support for runs on NCCS Cray XT5 ‘Jaguar’
  - DOE 2012 INCITE support for runs on NCCS Cray XK6 ‘Titan’

- Current projects aim to understand the ‘re-ionization’ era
Ultra-large-scale Cosmology

• Genesis
  – $6400^3$ mesh, 268B dark matter particles
  – Star formation with energy and momentum feedback
  – 93,750 cores on NICS Kraken
  – More than 1 petabyte of output
  – World’s largest hydrodynamic cosmology simulation

• Cosmic Re-ionization
  – $3200^3$ mesh, 32B dark matter particles
  – Star formation with energy and momentum feedback
  – Grey flux-limited radiation diffusion
  – 46,875 cores on NCCS Jaguar
  – First full RHD simulation of cosmic re-ionization
ENZO-R

- ENZO-R is a scalable hybrid MPI + OpenMP code
  - Standard C, C++ and Fortran90
  - Approximately 250,000 lines of code
  - ENZO I/O is managed with HDF5
  - ENZO RHD uses LLNL Hypre preconditioners and solvers
  - ENZO uses 3D FFTs, Multigrid, PPM hydro
  - Much of the code is cleanly vectorizable for heritage systems
ENZO on the Intel MIC

- ENZO is now running on the Intel MIC
  - More than 1M lines of code ported to the MIC
  - MPICH, HDF5 and HYPRE as well as ENZO
  - Relatively modest effort to achieve fully-functional code
  - MIC-resident approach is preferred for ENZO
  - Multiple MPI tasks per MIC and many threads per MPI task
  - ENZO is already highly vectorized

- ENZO tools are also running on the Intel MIC
  - Initial conditions generator runs in several modes
    - Natively with MPI + OpenMP in hybrid mode
    - Natively with pure OpenMP
    - In offload mode from the host
    - Reverse offload potential for I/O and visualization
Preliminary Scaling Study

- ENZO-C
- $128^3$ mesh (non-AMR)
- pure MPI
- native mode

<table>
<thead>
<tr>
<th>Cores</th>
<th>Wall Time (s)</th>
<th>Decomp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>85.1</td>
<td>2x1x1</td>
</tr>
<tr>
<td>4</td>
<td>45.2</td>
<td>2x2x1</td>
</tr>
<tr>
<td>8</td>
<td>24.3</td>
<td>2x2x2</td>
</tr>
<tr>
<td>16</td>
<td>13.6</td>
<td>4x2x2</td>
</tr>
<tr>
<td>32</td>
<td>8.6</td>
<td>4x4x2</td>
</tr>
</tbody>
</table>
Current Status

- ENZO-R is operational in with all physics modes
  - Non-AMR cosmology with radiation
  - AMR cosmology
  - Applications are only constrained by available hardware

- MIC development
  - Improving scalability at all levels
  - MPI vs. OpenMP trade-off
  - Hand-tuning of vectorization and threading
  - Reverse offload for I/O and visualization
Computational Fluid Dynamics on the Intel® MIC Architecture

R. Glenn Brook & Ryan C. Hulguin
Application Acceleration Center of Excellence
Computational Fluid Dynamics

From the ocean, to the skies, and beyond, Computational Fluid Dynamics (CFD) has helped design and refine advanced transportation vehicles. CFD simulations of fluid flows around vehicles are often used to reduce the costs of experimental materials and testing equipment. The use of CFD also prevents unnecessary human exposure to danger, especially during the early stages of testing.
Two separate CFD solvers were developed to showcase the capabilities of the Intel MIC coprocessors:

1. The first employs the Euler equations to simulate inviscid fluid flows on unstructured grids with five state variables at each physical grid point.

2. The second employs a discrete velocity model of the Boltzmann equation with the BGK collision approximation to simulate fluid flows on structured grids with thousands to hundreds of thousands of state variables per physical grid point.

Both solvers use a novel point-iterative Newton algorithm to solve the nonlinear equations with a near-minimal memory footprint. Data parallelism across the cores of the MIC coprocessor is achieved using OpenMP threads.

<table>
<thead>
<tr>
<th></th>
<th>Euler Solver</th>
<th>Boltzmann Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of equations per</td>
<td>5</td>
<td>Hundreds of thousands</td>
</tr>
<tr>
<td>physical grid point</td>
<td>Inviscid fluid flow</td>
<td>Rarefied gas flow</td>
</tr>
</tbody>
</table>
First test problem uses the Euler solver to simulate a Sod Shock. In a shock wave, the properties of a fluid change almost instantaneously. The standard Sod Shock starts off with a fluid at rest with initial conditions shown to the left. The Sod Shock is a popular test case for verifying a solver’s ability to appropriately capture shocks and contact discontinuities in unsteady fluid flows.

Sod Shock
Initial conditions:

| $\rho = 1.0$ | $\rho = 0.125$ |
| $u = 0.0$ | $u = 0.0$ |
| $P = 1.0$ | $P = 0.1$ |

Unsteady flow problem using the Euler equations
The second test problem uses the Boltzmann solver to simulate a Couette flow. In Couette flows, gas is initially at rest between two infinitely long parallel plates. For this problem, the left plate is stationary while the right plate moves. Over time, the gas settles into a solution that does not change. Couette flow makes a great test problem to verify a solver’s ability to handle solid surfaces and moving boundary conditions.
Solutions Generated by the Intel MIC

Unsteady solution of a Sod Shock using the Euler equations

Steady-state solution of a Couette flow using the Boltzmann equation with BGK collision approximation
Scaling Performance on the Intel MIC

Speedup from the Sod shock test problem run in native mode on the MIC using both single and double precision.

Speedup from the Couette test problem run in native mode on the MIC using single precision.

92% – 96% of ideal Speedup

73% – 99% of ideal Speedup

63% of ideal Speedup

75% of ideal Speedup

Graphs showing speedup vs. number of threads for single and double precision.
Observations from CFD Applications Developed for Intel MIC

- Running the CFD solvers on the Intel MIC coprocessors is a very easy process:
  1. Compile on the host with the new compiler flag option ‘- mmic’.
  2. Copy the binary, any input files, and any shared libraries to the MIC card.
  3. Execute the binary on the MIC card directly.

- Performance results indicate that iterative algorithms using OpenMP threads can effectively use the Intel MIC cores.

- The CFD solver based on the Euler equations scales to 99% of the maximum expected speedup on the MIC.

- While the cards have 32 physical cores, the fastest execution times are achieved with 96 OpenMP threads.
Summary of Accomplishments

• Migration of important libraries to the Intel MIC
  – mpich 1.2.7p1
  – HDF5 1.8.5
  – HYPRE 2.6.0b (with BLAS and LAPACK)

• Ported millions of lines of code to the Intel MIC architecture in weeks
  – Full applications from a variety of scientific and engineering disciplines — MPI and/or OpenMP

• Implemented MIC-to-MIC communications both on node and off node
  – Demonstrated MPI and Hybrid-MPI/OpenMP communications within a single MIC, between two MICs on a node, and across multiple MICs on multiple nodes
Enabling features

- Standard programming model and languages
  - MPI, OpenMP, pThreads (also Intel Cilk Plus & TBB)
  - Existing code just runs
  - Massively parallel code runs fast
  - Scalar-vector programming model facilitates high-performance for complex control flows
  - Threading enables diverse paths to high performance

- Optimizations for MIC benefit Xeon & vice versa

- Fully functional MKL — fast and heavily tested

- Rich vector intrinsics that extend AVX & offer diverse operations and math functions
Contact

R. Glenn Brook, Ph.D.
Director, Application Acceleration Center of Excellence
Joint Institute for Computational Sciences
University of Tennessee / Oak Ridge National Laboratory
glenn-brook@tennessee.edu