Early Application Experiences with Heterogeneous Computing Architecture

Hongsuk Yi
May 7 2012
KISTI Supercomputing Center (Korea)
KISTI-Intel MIC Collaboration

- Intel MIC Partner
  - 2 MIC workstation (2011)
  - 16 MIC cluster (2012)

- Increasing Parallelism
  - Core scaling (cores)
    - Nehalem (4) -> Westmere (6) -> KNF (32) -> KNC(50+)
  - Data level parallelism (SIMD) scaling
    - Earliest SSE (64-bit) -> SSE (128-bit) -> AVX (256-bit) -> KNF (5 12-bit)
  - Thread scaling/core
    - Core 2 (1) -> Nehalem (2) -> KNF(4)
Hardware Resources

• MIC “Knights Ferry” Software Development Platform
  – Up to 32 cores, 1~2 GB of GDDR5 RAM
  – 512-bit wide SIMD registers and L1/L2 caches
  – Multiple threads (up to 4) per core

• Medusa-1 (2011 ~) : KNF
  – 2 KNF MICs (1.2 GHz with 32 cores)
  – Intel Xeon 2 Westmere 3.33 GHz (12 cores+HT)
KISTI MIC Cluster (2012)

- **Medusa-2 : Intel MIC Cluster**
  - 16 x Intel Knights Ferry 1.2 GHz (32 cores)
  - Intel Xeon 16 Westmere 3.33 GHz (12 cores+hyper-threading)
KISTI Molecular Dynamics

• Empirical-potential molecular dynamics
  – Widely used for simulating nano-materials including carbon nanotube, graphene, fullerene, and silicon surfaces

• Highly parallel simulation
  – Parallelized well with standard multi-threading programming models such as OpenMP
KMD : KISTI Molecular Dynamics

- Empirical Potential MD
  - is very good algorithm
  - Newtonian EOM + Rather simple form of potentials
  - Our choice is Tersoff potential on covalent C, Si, Ge

\[
V_{ij} = f_C(r_{ij})[a_{ij}f_R(r_{ij}) + b_{ij}f_A(r_{ij})],
\]
\[
f_R(r) = A \exp(-\lambda_1 r),
\]
\[
f_A(r) = -B \exp(-\lambda_2 r),
\]
\[
b_{ij} = (1 + \beta^n \xi_{ij}^n)^{-1/2n},
\]
\[
\xi_{ij} = \sum_{k \neq i,j} f_C(r_{ik})g(\theta_{ijk}) \exp[\lambda_3^3(r_{ij} - r_{ik})^3],
\]
\[
g(\theta) = 1 + c^2/d^2 - c^2/[d^2 + (h - \cos\theta)^2],
\]

1. Cohesive energy per atom (eV), and bond length
KISTI-Intel MIC Demonstration

• KMD Simulation
  – ISC 2011 : KMD Demo
  – ISC 2011 : KMD Demo

• Using OpenMP Parallelization

fullerene  Graphene  Nanotube  Diamond
Linear speedup with the number of Cores

On Single MIC (32 Cores x 4 Threads)
Scaling with SIMD for MIC

• Vectorization
  – 512b SIMD extension in Intel MIC continues to scale MIC performance maintaining high-efficiency

\[
\begin{align*}
    b_{ij} &= \left(1 + (\beta \zeta_{ij})^n\right)^{-1/2n} \\
    V_{ij} &= f_c(r_{ij})[a_{ij} A \exp(-\lambda_1 r_{ij}) - b_{ij} B \exp(-\lambda_2 r_{ij})]
\end{align*}
\]

1. Intrinsic (SSE, AVX, MIC)
   • \texttt{v_tmp = _mm256_pow_pd(_mm256_mul_pd(beta, zij), n1)}

2. Class (SSE, AVX, MIC)
   • \texttt{F64vec4 v_tmp = pow(F64vec4(beta) * zij, F64vec4(n))};

3. OpenCL (GPU)
   • \texttt{float4 Bij = pow(1.0 + tmp, -1/(2*n))};
Hybrid Programming Model

- **MPI is essential**
- **GPU**
  - MPI + CUDA
  - MPI + OpenCL
- **MIC**
  - MPI + Offloading
  - MPI + Symmetric
Multi-GPU Scalability: Monte Carlo

• GPU implementation
  – Scalability of the CUDA implementations with increasing number of GPUs
  – MPI+CUDA
  – weak scaling

• Optimization strategy
  – Asynchronizion between Kernel and MPI comm.
  – Separation between ghost cell and inner region on GPU
  – Use a Nonblocking Send/recv
Future Work

• Compiling against MPI implementation on the MIC
  – Offload Programming model
  – OpenMP used within the co-processor

• Expecting more on future MIC products with productive programming models and expected performance in real applications
  – KMC, KMD
  – Implement MIC-to-MIC communications both on node and off node
    • Explain hybrid MPI/OpenMP communications within a same MIC, between two MICs on a node, and across multiple nodes
Contact
Hongsuk Yi, Ph.D.
KISTI Supercomputing Center
hsyi@kisti.re.kr

Thanks to
Hogyun Jeong (KISTI)
Seungmin Lee (KISTI)
Heungsik Kim (KISTI/SNU)
Byoungwon Choe (Intel)