Accelerating Molecular-Dynamics Simulation on a Many-core Computing Platform

Liu Peng

Collaboratory for Advanced Computing & Simulations
Department of Computer Science
University of Southern California

Molecular Dynamics Simulation

Molecular Dynamics (MD)

Linked-list cell method for MD

Irregular memory access
Frequent communication

\[ m_i \frac{d^2}{dt^2} r_i = -\frac{\partial}{\partial r_i} E_{MD}(\{r_i\}) \]
64-core GodsonT many-core architecture

- 64 homogenous, dual-issue core 1GHz, 128Gflops in total
- Lightweight hardware thread
- Explicit memory hierarchy
- 16 shared L2 cache banks, 256KB each
- High bandwidth on-chip network: 2TB/s
Optimization Strategy I: Adaptive Divide-and-Conquer (ADC)

- **Purpose:** Estimate the upper bound of decomposition cell size where all data can fit into each core’s local storage (SPM)
- **Solution:** Recursively do cellular decomposition until the following equation (adaptive to the size of each core’s SPM) is satisfied

\[
\left( \frac{L}{P} + N_b \right) \times qR_c^3 \times B \leq C_{pm} \Rightarrow R_c \leq \sqrt[3]{\frac{PC_{pm}}{(PN_b + L)Bq}}
\]

**Estimation of the size of all data in a cell with cell size of } R_c \]}

ADC + software controlled memory (decide when and where the data reside in SPM) to enhance the data usage
Optimization Strategy II: Data Layout Optimization

- **Purpose:** Ensure contiguous touching of data in each cell
- **Solution:** Data grouping/reordering + local-ID centered addressing

**Diagram Explanation:**
- **Na:** the number of atoms in one cell
- **Cc:** local-ID of each cell at one core
- **L2_data_unit** is the data transfer unit from shared L2 cache or off-chip memory to LS via DMA-like operation
Optimization Strategy III: On-chip Locality Optimization

• **Purpose:** Maximize data reuse for each cell
• **Solution:** Pre-processing to achieve locality-awareness, and further use locality-awareness to maximize data reuse

**Parallel processing to achieve locality-awareness**

- **core_i**
  - $PC[1]=\{2\}$
  - $PC[4]=\{2, 5\}$
  - $PC[8]=\{5\}$

- **core_j**
  - $PC[4]=\{2, 5\}$
  - $PC[8]=\{5\}$

**Maximize Data reuse**

If cell $k$ in core $i$, then use $PC$ to get all interactive cells, exhaust all the inter-computation

**Architecture mechanism support for high-bandwidth core-core communication**
Optimization Strategy IV: Pipelining Algorithm

**Purpose:** Hide latency to access off-chip memory

**Solution:** Pipelining implemented via double buffered, asynchronous DTA operations

1. $tag_1 = tag_2 = 0$
2. for each cell $c_{core_i}[k]$ listed in PC[$cj$]
3. if ($tag_1 \neq tag_2$)
4. DTA_ASYNC(spm_buf[1- $tag_2$], l2_dta_unit[$c_{core_i}[k]$])
5. $tag_2 = 1 - tag_2$
6. endif
7. calculate atomic interactions between $c_{core_i}[k]$ and $cj$
8. spm_buf[$tag_1$] ← cell $c_{core_i}[k]$’s neighbor atomic data
9. $tag_1 = 1 - tag_1$
10. endfor
11. if ($tag_1 \neq tag_2$)
12. DTA_ASYNC(spm_buf[1- $tag_2$], l2_dta_unit[$c_{core_i}[k]$])
13. $tag_2 = 1 - tag_2$
14. endif
Performance Tests

FPGA emulator for 64 core *GodsonT*

On-chip strong scalability

optimization-1: only ADC
optimization-4: all 4 optimizations

Excellent strong-scaling multithreading parallel efficiency of 0.99 on 64 cores with 24,000 atoms (0.65 on 8-core multi-core)
Performance Analysis

**Running time**

- Running time is reduced to half

**L2 cache performance**

- L2 cache events are greatly reduced

---

**Bar Chart:**
- **Execution time (ms):**
  - Opt-1: 50
  - Opt-2: 40
  - Opt-3: 20
  - Opt-4: 10

**Graph:**
- **Number of events:**
  - L2_access: $3 \times 10^6$
  - L2_miss: $1 \times 10^6$
  - L2_replace: $1796$

- **Legend:**
  - optimization-1
  - optimization-2
Remote memory access performance

Number of remote memory accesses is reduced to 7%