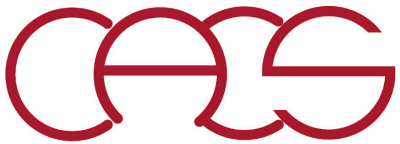
Accelerating Molecular-Dynamics Simulation on a Many-core Computing Platform

Liu Peng

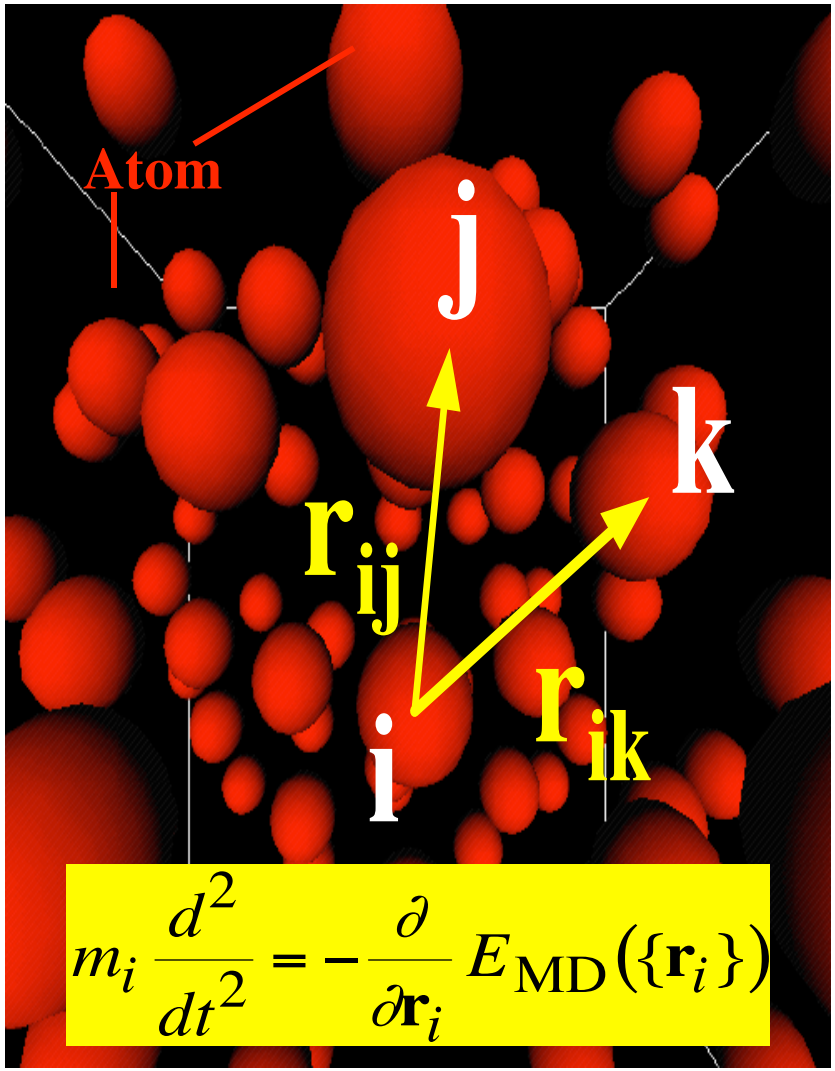
Collaboratory for Advanced Computing & Simulations
Department of Computer Science
University of Southern California

- "Scalability study of molecular dynamics simulation on Godson-T many-core architecture," L. Peng, G. Tan, R. K. Kalia, A. Nakano, P. Vashishta, D Fan, H. Zhang, and F. Song, *J. Par. Distrib. Comput.* **73**, 1469 ('13)
- "Performance analysis and optimization of molecular dynamics simulation on Godson-T many-core processor," L. Peng, G. Tan, D. Fan, R. K. Kalia, A. Nakano, and P. Vashishta, in *Proc. Int'l Conf. Computing Frontiers, CF'11* (ACM, Ischia, Italy, '11)
- "Preliminary investigation of optimizing molecular dynamics simulation on Godson-T many-core processor," L. Peng, G. Tan, R. K. Kalia, A. Nakano, P. Vashishta, D. Fan, & N. Sun, in *Proc. Workshop on Unconventional High Performance Comput., UCHPC 2010* (Naples, Italy, '10)

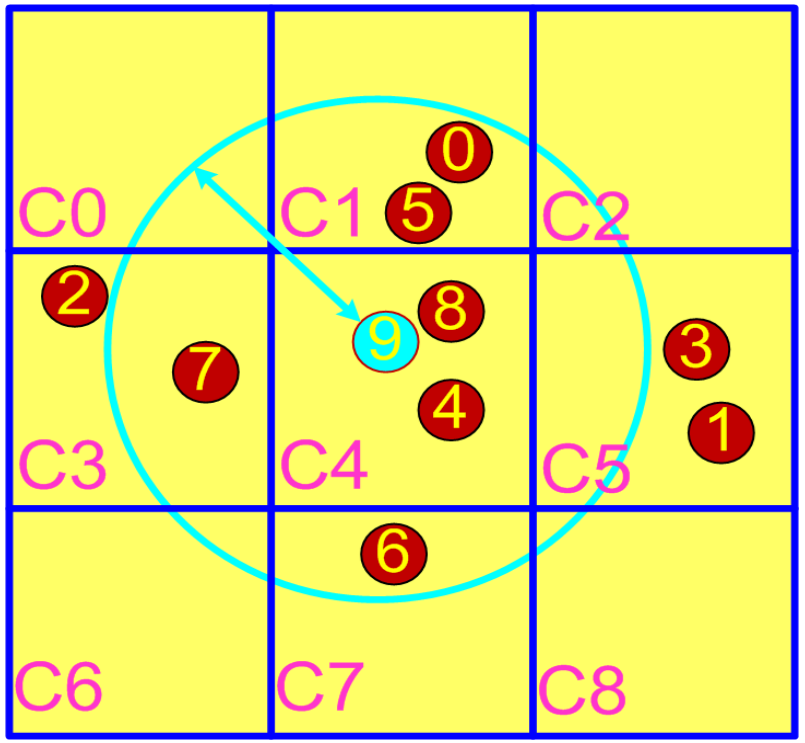


Molecular-Dynamics Simulation

Molecular Dynamics (MD)



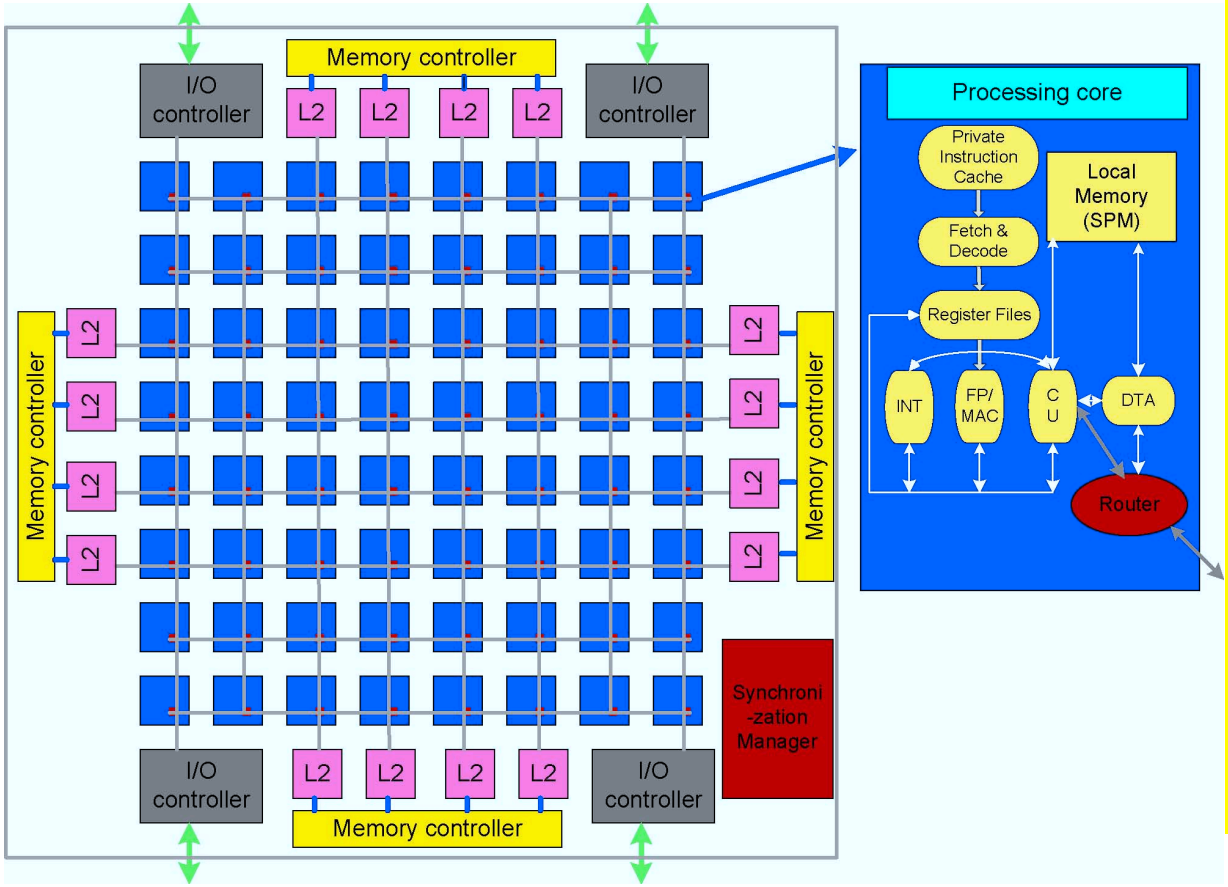
Linked-list cell method for MD



Irregular memory access
Frequent communication

GodsonT Many-core Computing Platform

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

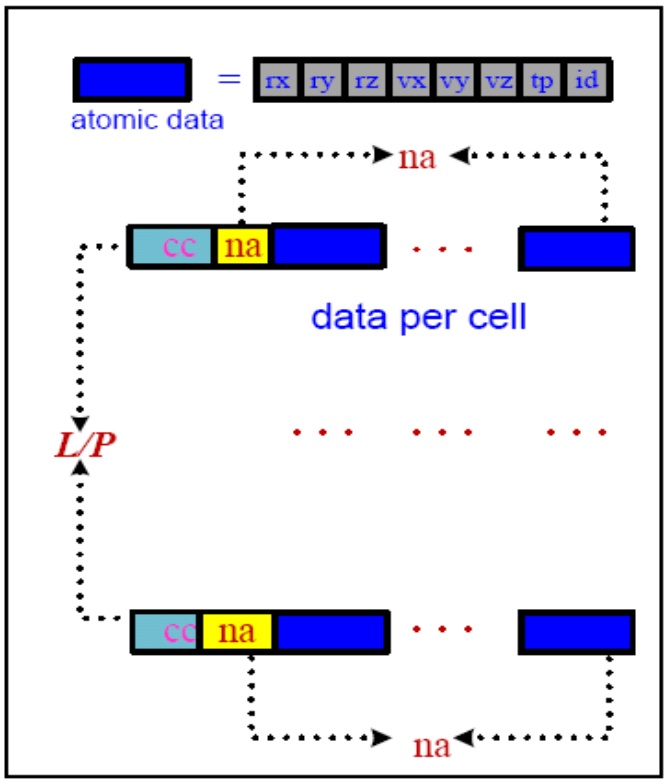
$$\left(\frac{L}{P}+N_b\right)\times qR_c^3\times B \le C_{pm} \Rightarrow R_c \le \sqrt[3]{\frac{PC_{pm}}{(PN_b+L)Bq}}$$

Estimation of the size of all data in a cell with cell size of *Rc*

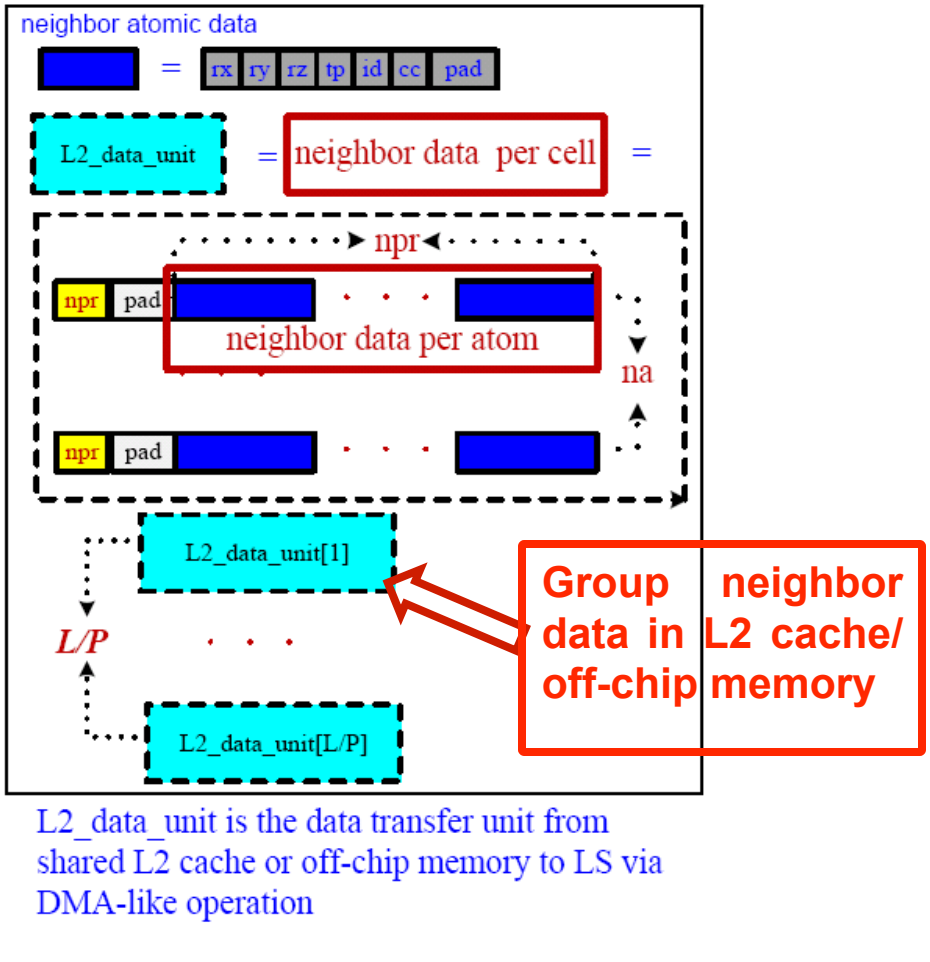
ADC + software controlled memory (decide when and where the data reside in SPM) to enhance the data usage

Optimization Strategy II: Data Layout Optimization

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



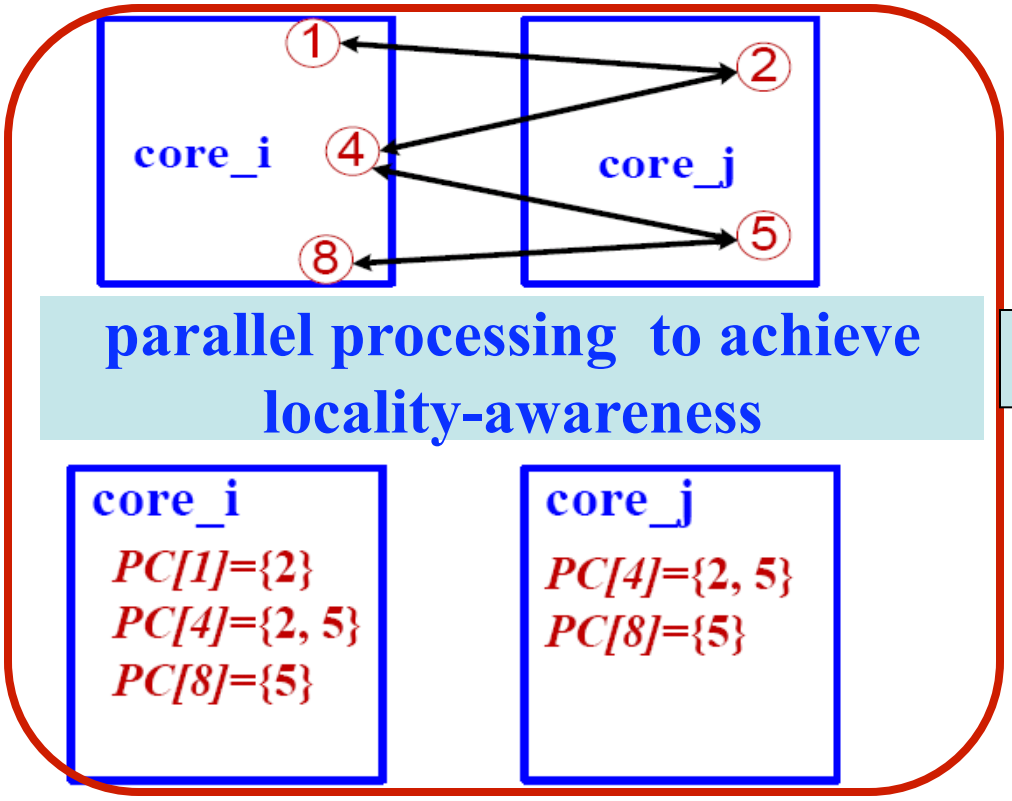
na: the number of atoms in one cell
cc: local-ID of each cell at one core



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



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

Maximize data reuse

If the interactive cell *j* is not in the same core, Issue memory transfer

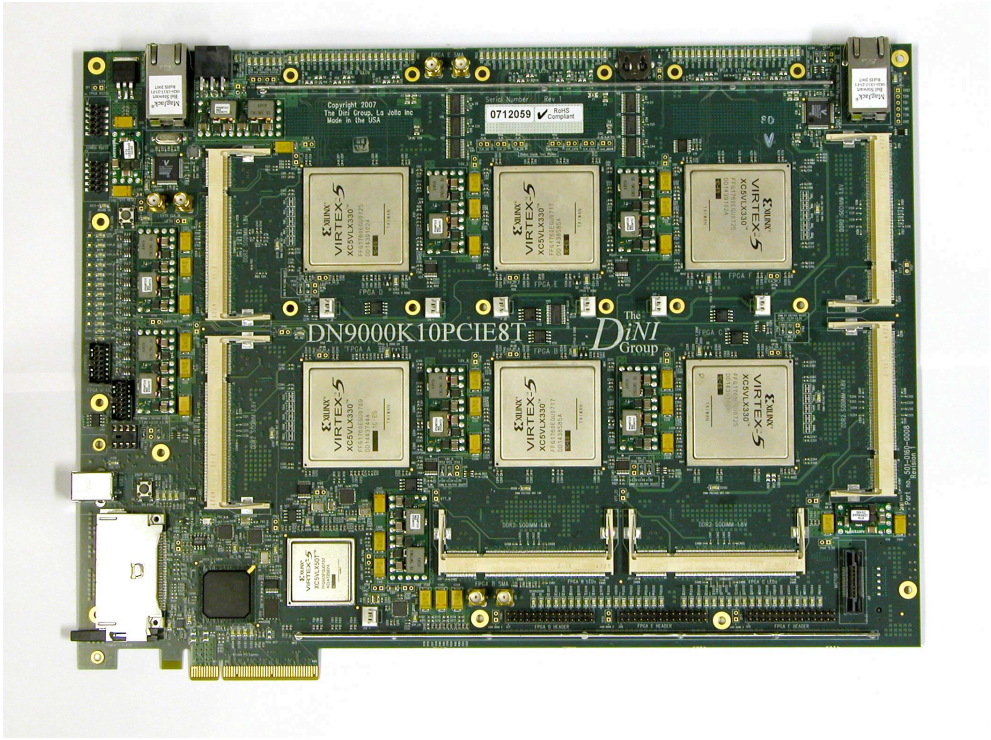
If the interactive cell *j* is already in the same core, Do computation

pipeline

```
1.  tag1 = tag2 = 0
2.  for each cell c_core_i [k] listed in PC[cj]
3.    if (tag1 ≠ tag2)
4.        DTA_ASYNC(spm_buf[1- tag2],
        l2_dta_unit[c_core_i [k]])
5.        tag2 = 1- tag2
6.    endif
7.    calculate atomic interactions between
        c_core_i [k] and cj
8.    spm_buf[tag1] ← cell c_core_i [k] 's
        neighbor atomic data
9.    tag1= 1- tag1
10.  endfor
11.  if (tag1 ≠ tag2)
12.     DTA_ASYNC(spm_buf[1- tag2],
          l2_dta_unit[c_core_i [k]])
13.    tag2 = 1- tag2
14.  endif
```

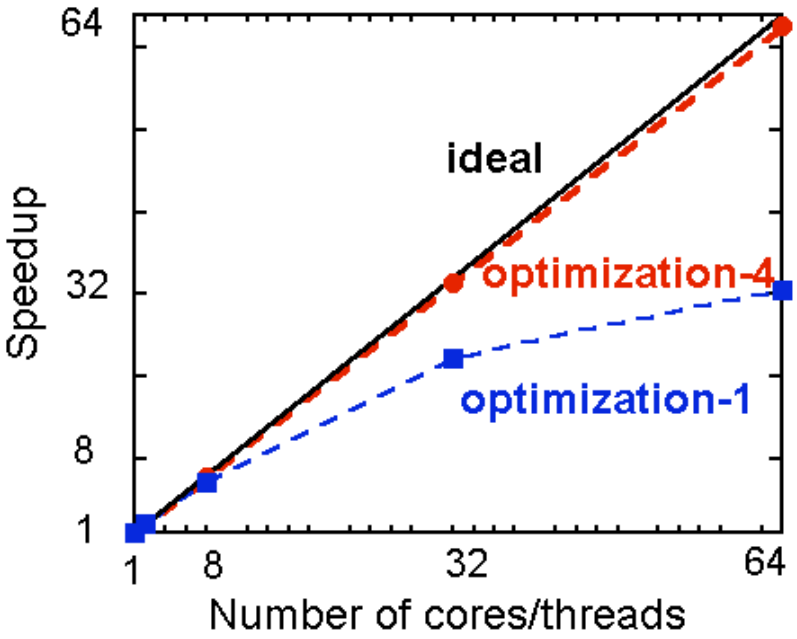
Performance Tests

FPGA emulator for 64 core *GodsonT*



On-chip strong scalability

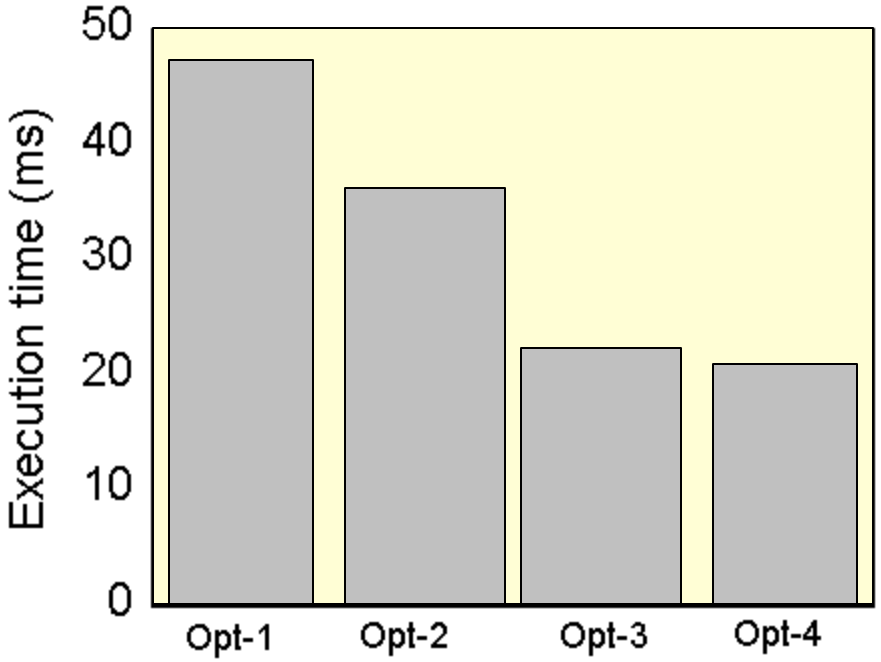
optimization-1:only ADC
optimization-4: all 4 optimizations



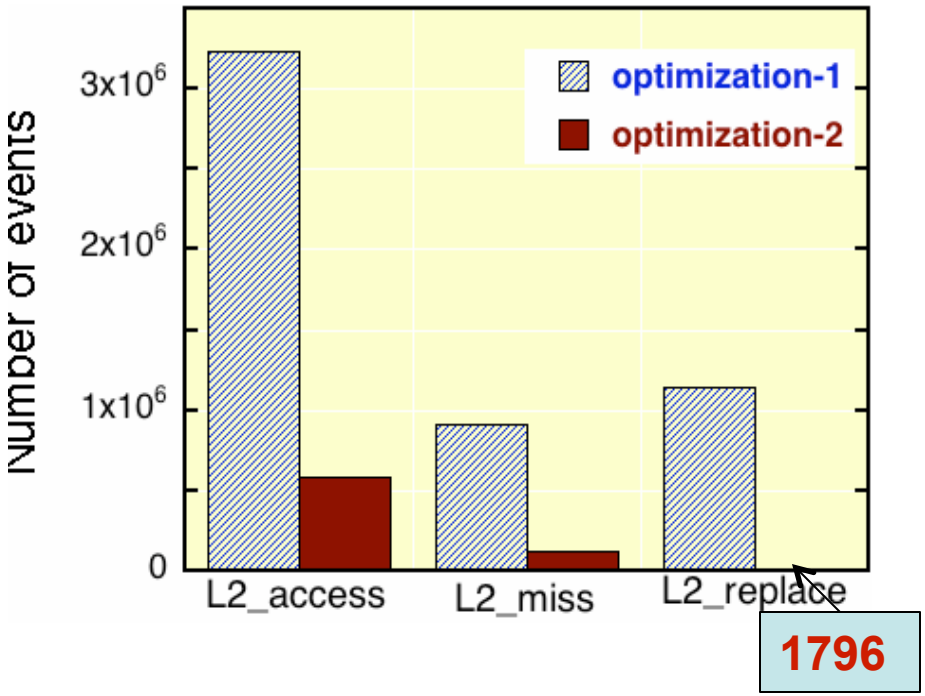
Excellent strong-scaling multithreading parallel efficiency of 0.99 on 64 cores with 24,000 atoms (0.65 on 8-core multi-core)

Performance Analysis

Running time



L2 cache performance

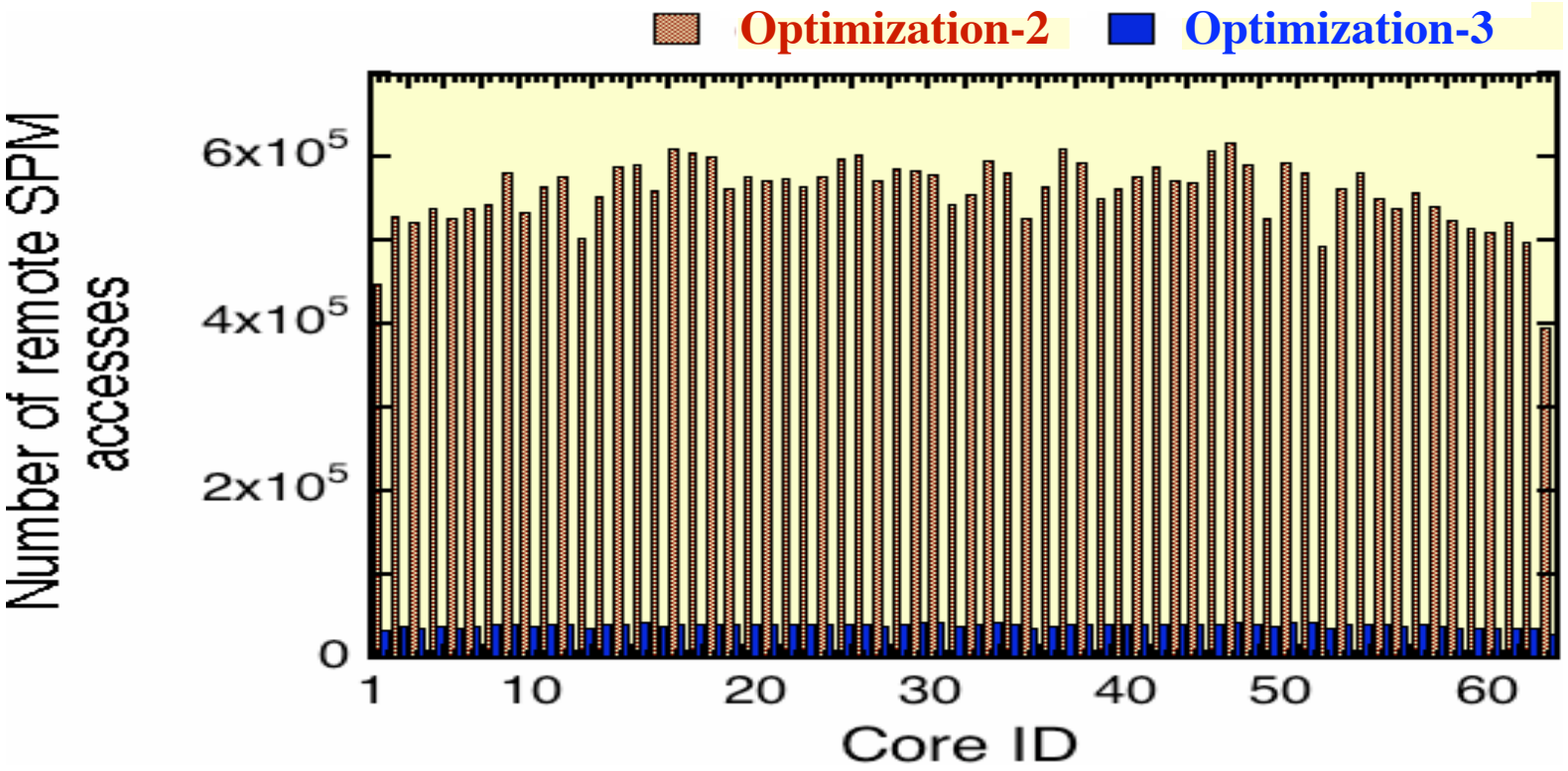


Running time is reduced to half

L2 cache events are greatly reduced

Performance Analysis

Remote memory access performance



Number of remote memory accesses is reduced to 7%