Molecular Dynamics Optimization on Intel® Many Integrated Core Architecture (Intel® MIC)

Introduction

Molecular Dynamics (MD) is a computer simulation of physical movements of atoms and molecules [1], as shown in Figure 1. The atoms and molecules interact with each other in a specified field obeying Newton’s Second law, in which the forces and potential energy are defined by molecular mechanics force fields, and the velocities and movements are determined by numerically solving the Newton's equations of motion.

Atoms and molecules can be stored in cell linked-lists [2], in which way they can easily find neighbors to interact with. In each step of simulation, atoms and molecules should follow the sequence as move, update cells, and then collide. When moving, they update velocities and movements. When updating cells, they determine the cells where they shall locate in. When colliding, they calculate the forces and potential energy with neighbors.

In chemistry, MD is used in protein structure determination and refinement using experimental tools such as X-ray crystallography and Nuclear Magnetic Resonance (NMR). It has also been applied with limited success as a method of refining protein structure predictions. In physics, MD is used to examine the dynamics of atomic-level phenomena that cannot be observed directly, such as thin film growth and ion-sub plantation [1].

Intel introduces Intel® Many Integrated Core (Intel® MIC) architecture, which consists of up to 61 cores and supports 4 hardware threads for each core. In comparison with CPU
architecture, Intel MIC has more computing power. In this report, the transportation of MD program to Intel MIC is described as well as the key optimization methods.

**MD Computation Flow**

In MD calculation, the forces, velocities and position of each molecule are recorded as its properties and are updated in each time step. To reduce the number of molecules interacted with the updated molecule, all molecules are split into a grid, composed of cells (shown in Figure 2). Then only the molecules in neighboring cells and same cell are taken into account. For example, when molecules’ properties in cell 0 are updated, they only interact with molecules in cells from 1 to 8 as neighbor cells and itself, cell 0. Since interactions between two molecules are symmetric, we can calculate in one direction and refer to only half number of the neighbors. Therefore, molecules in cells from 1 to 4 are used to update the molecules in cell 0 as neighbor cell.

In each time stamp, molecules are reassigned to cell locations as their positions are changed. Because the number of molecules in one cell is limited, few molecules may be dropped. After a given numbers of iterations, the calculated properties of all molecules are the results of molecular dynamics.

![Molecules split into grid with cells](image)

**Key Optimization Method**

In this section, we describe the key methods to optimize the MD programs in details. The original program is serial implementation. To utilize the compute resource of multi-core CPU and many-core Intel MIC, we parallel this program using Intel® Threading Building Blocks (Intel® TBB) at first.
Modify Data Structure to be Accessed Contiguously

Molecules are represented in struct. In 2-Dimension MD, each property is represented by two components in x and y directions. In the original implementation, those components are stored alternatively in one array. When we process one component, the memory accesses are not contiguous, which is not suitable for vectorization. This modification changes the Array of Struct (AOS) to Struct of Array (SOA).

When the properties are stored, we store the each property in one separated array. Then the memory access pattern can be contiguous and the operation can be vectorized.

Vectorization with CEAN and intrinsic

C/C++ Extension for Array Notation (CEAN) is an Intel-specific extension and supported by Intel® compilers. It enables the compiler to vectorize the operation with less reliance on alias and dependence analysis automatically.

Intel® Streaming SIMD Extension 4.2 (Intel® SSE 4.2) [3] is a SIMD instruction set extension to X86 architecture, which allows that more than one data can be processed simultaneously in one instruction. Now Intel has introduced a new 256-bit instruction set extension to Intel SSE --- Intel® Advanced Vector Extensions (Intel® AVX) [4]. To utilize those instruction sets to accelerate application performance, intrinsics are designed for developers in place of assembly instructions. Intrinsics are assembly-coded functions, which may consist of several SIMD instructions.

In summary, CEAN is used for implicit vectorization by the compiler and intrinsics are used explicitly by developers. In MD optimization, we use both CEAN and intrinsics for optimization. As we already store each property in an individual array, as the previous session mentions, each property can be loaded into a separated vector register. The other benefit of vectorization is reducing memory access, since more than one operand is loaded into register at one time.

Experiment Result

In this section, the optimized methods are evaluated. The platform information is given in Table 1.
The optimization result is given in Figure 3. As the result shows, the speedup of optimization methods on Intel® microarchitecture code name Westmere can reach up to 32.8X mainly derived from parallelization and vectorization.

We can disable vectorization via adding “-no-vec” option in compilation. The speedup of the program using Intel SSE 4.2 is 2.97X. If we utilize the new proposed Intel AVX in Intel® microarchitecture code name Sandy Bridge, the speedup is 2.22X in comparison with Intel SSE version. The key difference is that registers are 256 bit in Intel AVX rather than 128 bit in Intel SSE4.2. The registers in Intel MIC are 512 bits, bringing more benefit for vectorization on Intel MIC.

The optimization result is given in Figure 3. As the result shows, the speedup of optimization methods on Intel® microarchitecture code name Westmere can reach up to 32.8X mainly derived from parallelization and vectorization.

We can disable vectorization via adding “-no-vec” option in compilation. The speedup of the program using Intel SSE 4.2 is 2.97X. If we utilize the new proposed Intel AVX in Intel® microarchitecture code name Sandy Bridge, the speedup is 2.22X in comparison with Intel SSE version. The key difference is that registers are 256 bit in Intel AVX rather than 128 bit in Intel SSE4.2. The registers in Intel MIC are 512 bits, bringing more benefit for vectorization on Intel MIC.

When the hardware platform is updated, the speedup of Intel microarchitecture code name Sandy Bridge compared to Intel® microarchitecture code name Westmere is 2.98X. And the
speedup can be still increased up to 1.89X when we use Intel MIC, compared to Intel microarchitecture code name Sandy Bridge.

Summary

In this report, we optimize the molecular dynamics program in Intel platforms. The experiment results show the great performance improvement. When we parallelize and vectorize the program, the speedup reaches up to 97.74X. The computing resources of Intel Xeon CPU are explored sufficiently. After the Intel MIC is used, we can obtain an extra 1.89X speedup and the total speedup is increased to 184.74X.

References:


About Author

Xiangzheng Sun graduated from Institute of Software, Chinese Academy of Sciences. He majors in High Performance Computing (HPC) and IPDC. Now He mainly focuses on application developing/porting and performance tuning on Intel® Multi-Core CPU and Intel® Many Integrated Core (Intel® MIC).
NOTICES

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS.
NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY
INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS
PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL
ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED
WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY
OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE,
MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER
INTELLECTUAL PROPERTY RIGHT.

A “Mission Critical Application” is any application in which failure of the Intel Product
could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE
OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL
INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND
AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS
AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE
ATTORNEYS’ FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT
LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION
CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS
NEGLIGENCE IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR
ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without
notice. Designers must not rely on the absence or characteristics of any features or
instructions marked “reserved” or “undefined”. Intel reserves these for future definition
and shall have no responsibility whatsoever for conflicts or incompatibilities arising from
future changes to them. The information here is subject to change without notice. Do not
finalize a design with this information.

The products described in this document may contain design defects or errors known as
errata which may cause the product to deviate from published specifications. Current
characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications
and before placing your product order. Copies of documents which have an order number
and are referenced in this document, or other Intel literature, may be obtained by calling
1-800-548-4725, or go to:  http://www.intel.com/design/literature.htm

Intel, the Intel logo, VTune, Phi and Xeon are trademarks of Intel Corporation in the U.S.
and other countries.

*Other names and brands may be claimed as the property of others
Copyright © 2013 Intel Corporation. All rights reserved.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: Learn About Intel® Processor Numbers

Optimization Notice
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 SSE3 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