21PA-037

Acceleration of plasma Particle-In-Cell simulations using GPU by hash

GPU を用いたプラズマ PIC シミュレーションのハッシュ法による高速化

Hiromi Matsunaga, Yasunori Mizuno, Hiroshi Inuzuka
松永寛海　　水野保則　犬塚博

Shizuoka University
3 - 5 - 1, Johoku, Nakaku, Hamamatsu 432 - 8561, Japan
静岡大学　〒432-8561　浜松市中区城北3−5-1

It is commonly thought that if the large scale plasma PIC simulation generally processed with the supercomputer can be processed by GPU, time and monetary cost is greatly reducible. In order to realize it, there is two methods using sorting and hash. However, it remains to be elucidated whether the hash is suitable for the large-scale plasma PIC simulation by GPU rather than sorting. To determine it, we compare hash with sorting, with the algorithm which imitated a part of PIC. We found that hash method is high speed than sorting in the large-scale plasma PIC simulation.

1. Introduction

Recently, computing power of GPU is dramatically improved. Then, the technology of applying GPU to large scale plasma simulations, attracts attention.

PIC is the typical technique of the particle simulation of plasma. But, its amount of operations is immense. Therefore, in a large scale simulation with many particles, a super computer is used in many cases.

Thus, if it can be calculated at high speed by GPU, a large scale simulation will become possible with a cheap and easy computing unit like a common personal computer. Then, the method of making PIC calculate at high speed by GPU is studied all over the world. We are studying the method of accelerating PIC using hash.

2. Hash method

PIC is the technique of dividing space in a cell and representing the charged particle in a cell as an electric charge of a cell. By the random access which is dependent on particle distribution when it is going to parallelize this processing simply, the problem that memory access speed is dependent on particle distribution arises.

We try to solve this problem using hash. The structure of hash is the mechanism of changing a certain keyword into the hash value which expresses the keyword by a hash function, as shown in Fig.1.

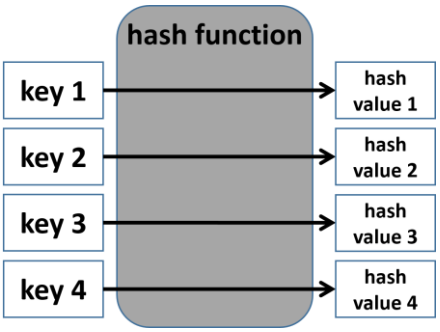

Fig.1 Structure of hash

That is, the same hash value is given to the particles which exist in the cell of a certain range, the particles of the same hash value are stored in 128 bytes of array, the array is connected by a chained list, and each chained list is processed in parallel. In this way, the dependence to the particle distribution of memory access is avoidable by narrowing the range of the cell which a parallel processing unit treats.

3. Comparison of hash and sorting method

There is an algorithm using sorting as a method of solving the problem of the memory access in PIC [1]. We examined disadvantages of the algorithm using hash and sorting in large scale operation, and the high speed technique by hash [2]. As sorting for comparison, we use bitonic sort suitable for parallel operation [3]. As an operation model to examine, we use the algorithm

which makes a random sequence an ascending order, and examine the calculation time. In this case, the length of sequence and the range of the random number are equivalent to the number of particles and the number of cells in PIC. Fig.2 shows the relation between the length of a sequence and the calculation time. Fig.3 shows the relation between the range of the random number and the calculation time. Since we assume large scale operation, we set the length of a sequence of number to 2^{24} which is the maximum of this environment.

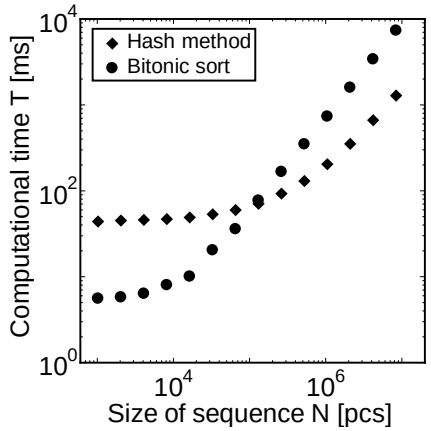

Fig.2 Effect of size of sequence.

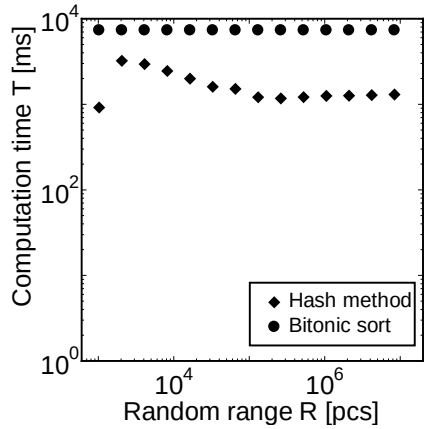

Fig.3 Effect of size of random range.

Fig.2 shows that sorting is more high speed than hash when the length of a sequence is short, and that hash is more high speed than sorting when the length of a sequence is long. This shows that hash is superior to sorting in large scale operation. Fig.3 also shows same, because hash is more high speed than sorting in Fig.3 whose length of a sequence is the maximum.

In a plasma simulation, the number of cells can be called parameter which can be set comparatively freely as compared with the number of particles. Therefore, when accelerating hash, the range of a random number corresponding to the number of cells is important value. If the range of a random number is too wide and it is too narrow, computing speed downs. Fig.4 is taken up about the variation of the computational speed.

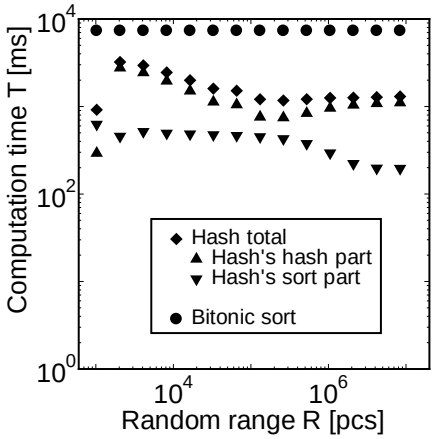

Fig.4 Effect of random range.

When rearranging a sequence by hash, the algorithm consists of a portion which gives a hash value to a sequence, and a portion which rearranges a sequence of numbers based on a hash value. Fig.4. shows that the portion which gives a hash value occupies a great portion of computational time. It happens easily that this portion is not desirable in respect of memory access compared with the portion which rearranges a sequence of numbers. The valley of this computation time is also depended on memory access.

References

[1]C. K. Birdsall, et al.: “Plasma physics via computer simulation”, McGraw-Hill Book Company, 1985.
[2]K. Sawamoto et al.: “Efficient algorithm of plasma PIC simulation for GPU”, Abstract of US-Japan JIFT Workshop 2011, 2011.
[3]K. Okada et al.: “A guide to CUDA-accelerated GPU computing architecture and programming”, Shuwasystem, 2010.