11. Particle Simulator Research Team

11.1. Team members

Junichiro Makino (Team Leader)
Keigo Nitadori (Research Scientist)
Masaki Iwasawa (Postdoctoral Researcher)
Yuri Iida (Assistant)

11.2. Research Activities

We are developing particle-based simulation software that can be used to solve problems of vastly different scales.

Simulation schemes for hydrodynamics and structural analysis can bedivided into grid-based and particle-based methods (see Figure 1). In grid-based methods, the computational region is mapped to regular or irregular grids. Continuous distributions of physical values are represented by discrete values at grid points, and the governing partial differential equation is approximated to a set of finite difference equations.

In the case of the particle-based methods, physical values are assigned to particles, while the partial differential equation is approximated by the interactions between particles.

Both methods are widely used, and they have their advantages and disadvantages. The computational cost of grid-based schemes is generally lower than that of particle-based methods with similar number of freedoms. Thus, if an near-uniform grid structure is appropriate for the problem to be solved, grid-based methods perform better.

The advantage of the particle-based methods comes from the fact that they use "Lagrangian" schemes, in which the particles move following the motion of the fluid in the case of the CFD calculation. In the case of grid-based methods, we generally use "Eulerian" schemes, in which the grid points do not move.

There are three points in which the Lagrangian schemes are better than Eulerian schemes.
One is that the Lagrangian schemes are, to some extent, adaptive to the requirement of the accuracy, since when a low-density region is compressed to become high density, Second one is that the timestep criteria are quite different. In the case of the Lagrangian schemes, the timestep is determined basically by local sound velocity, while in the Eulerian scheme by global velocity. Thus,

if a relatively cold fluid is moving very fast, the timestep for the Eulerian schemes can be many orders of magnitude shorter than that for Lagrangian schemes. Finally, in the case of fast-moving low-temperature fluid, the required accuracy would be very high for Eulerian scheme, since the error comes from the high velocity, while that error would be transferred to internal energy of the fluid element which is much smaller than that of the kinetic motion.

Of course, there are disadvantages of Lagrangian schemes. The primary one is the difficulty of construction of such schemes in two or higher dimensions. In the case of one-dimensional calculation, it is easy to move grid points following the motion of the fluid, but in two or higher dimensions, the grid structure would severely deform if we let the grid points follow the flow. Thus, we have to reconstruct the grid structure every so often. This requirement causes the program to become complex. Moreover, reconstruction of the grid structure (so called remeshing) means we lose numerical accuracy.

Particle-based methods "solve" this difficulty by not requiring any mesh. In particle-based methods, particles interact with its neighboring particles, not through some connection through grid, but through distance-dependent kernel functions. Thus, there is no need of remeshing. As a result, particle-based schemes are simple to implement, and can give reasonable results even when the deformation is very large. Another important advantage is that it is relatively easy to achieve high efficiency with large-scale particle-based simulation.

In the case of grid-based schemes, in order achieve some adaptivity to the solution, we have to use either irregular grid or regular grid with adaptive mesh refinment. In both cases, adaptivity breaks the regularity of the mesh structure, resulting in non-contiguous access to the main memory. In the case of the particle-based schemes, it does require some irregular memory access, but it is relatively straightforward to make good use of spacial locality, and thereby achieving high efficiency. Similarly, very high parallel performance can be achieved.

However, it has its own problems. In the case of the SPH method, it has been known that the standard scheme cannot handle the contact discontinuity well. It also require rather strong artificial viscosity, which results in very low effective Reynolds number.

Thus, in many fields of computational sciences, many groups are working on implementation of high-performance particle-based simulation codes for their specific problem.

One serious problem here is that, high-performance, highly-parallel simulation codes for

particle-based simulations are becoming more and more complex, in order to make full use of modern supercomputers. We need to distribute particles to many computing nodes in an appropriate way, so that the communication between nodes is minimized and at the same time near-optimal load balance is achieved. Within each nodes, we need to write an efficient code to find neighbor particles, rearrange data structure so that we can make good use of the locality, make good use of multiple cores and SIMD units within each core.

Even for the case of very simple particle-particle interaction such as the Lenard-Jones potential or Coulomb potential, the calculation code tends to be very large, and since the large fraction of the code is written to achieve a high efficiency on a specific architecture, it becomes very hard to port a code which is highly optimized to one architecture to another architecture.

Our goal is to develop a "universal" software that can be applied to a variety of problems whose scales are vastly different.
In designing such universal software, it is important to ensure that it runs efficiently on highly parallel computers such as the K computer. Achieving a good load balance with particle-based simulation is a difficult task, since using a regular spatial decomposition method causes severe load imbalance, though this works well for grid-based software. Consequently, we have developed an adaptive decomposition method that is designed to work in a way that the calculation time on each node is almost the same, resulting in near-optimal load balance.

The strategy to develop such a universal software is as follows.

We first construct an highly parallel and very efficient implementation of the TreePM algorithm for gravitational N-body problem. This is actually not a completely new implementation, but the GreeM code developed by researchers of the Strategic Program for Innovative Research (SPIRE) Field 5 “The origin of matter and the universe. In collaboration with the Field 5 researchers, we improve the efficiency of the code and study the issues of the data structure, domain decomposition, load balance strategy etc.

In the second stage, we will develop a prototype of the parallel particle simulation platform. We will design the platform so that it can be used for multiple physical systems. In practice, we consider the following three applications as the initial targets.

1. Gravitational N-body simulation
2. Smoothed Particle Hydrodynamics

3. Molecular Dynamics

In the meantime, we will also investigate the way to improve the performance and accuracy of the current particle-based algorithms for hydrodynamics.

11.3. Research Results and Achievements

As we stated in section 2, we are working on the three major subtopics, in order to develop the universal platform for particle simulations.

In the following, we briefly describe the status of our research in each subtopic.

11.3.1. High-performance gravitational N-body solver.

In collaboration with the researchers of researchers of the Strategic Program for Innovative Research (SPIRE) Field 5 “The origin of matter and the universe, we have developed an extremely high performance gravitational N-body solver, GreeM, for the K computer. It achieved, as of November 2012, the sustained performance of 5.67 petaflops (55% of the theoretical peak performance of the K computer). Even more important is its performance measured in the unit of the number of particles updated per second. GreeM on K integrates 4×10^{11} particles per second. Researchers in the US developed a similar calculation code on the BG/Q, and its measured speed was 1.6×10^{11} particles per second, on the BG/Q machine with the peak speed of 20Pflops. Thus GreeM on K is about 2.4 times faster than the best competing code on a machine nearly two times faster than the K computer. In other words, GreeM on K is about five times more efficient than the best competing code. The numerical accuracy was similar. In the following, we briefly describe the method used and the possible reason for the performance difference between GreeM on K and the calculation on BG/Q. The full detail of the GreeM code is discussed in Ishiyama et al (2012).

We use the TreePM algorithm as the basic method for the evaluation of gravitational interaction between particles. TreePM is a combination of the tree method and the P^3M (particle-particle particle-mesh) scheme. Figure 1 shows the basic idea of the tree algorithm. The space is divided into a hierarchical octree structure (quadtree in the figure). Division is stopped when a cell contains one or no particle. When we calculate the force on a particle, we evaluate the force from a group of particles, with size larger for more distant particles. In this way, we can reduce the calculation cost from $O(N^2)$ to $O(N \log N)$.

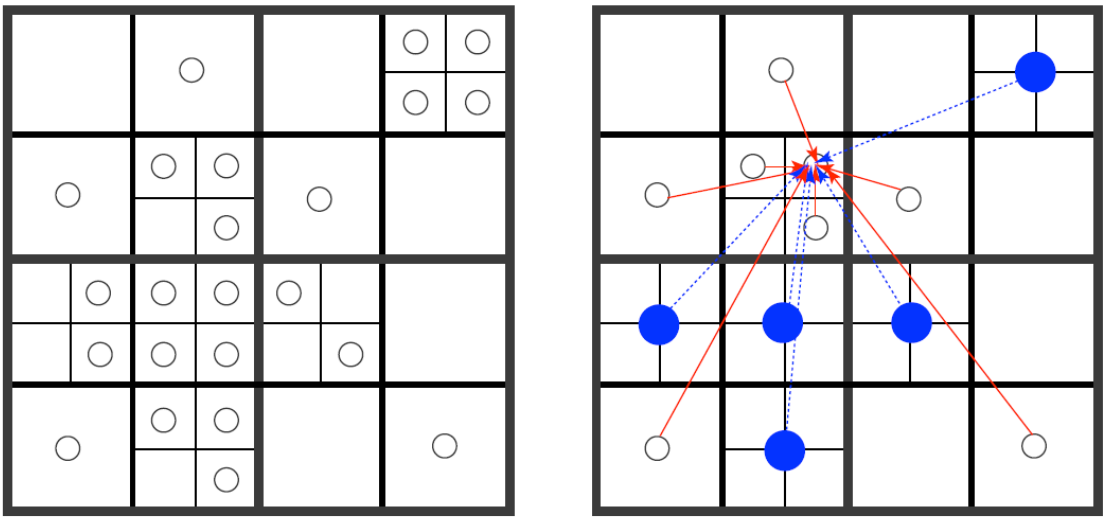

Figure 1. Basic idea of the tree algorithm

The tree algorithm is widely used, but when the periodic boundary condition is applied, we can actually use a more efficient efficient scheme, since we can calculate the long-range, periodic term using FFT. The P^3M scheme has been used for such problem, but it has the serious problem that when the density contrast becomes high, the calculation cost increases very quickly. The TreePM scheme solves this difficulty by using the tree algorithm to evaluate the forces from nearby particles. Even when there are very large number of neighbor particles, the calculation cost does not increase much, since the calculation cost of the neighbor force is proportional to the logarithm of the number of neighbors.

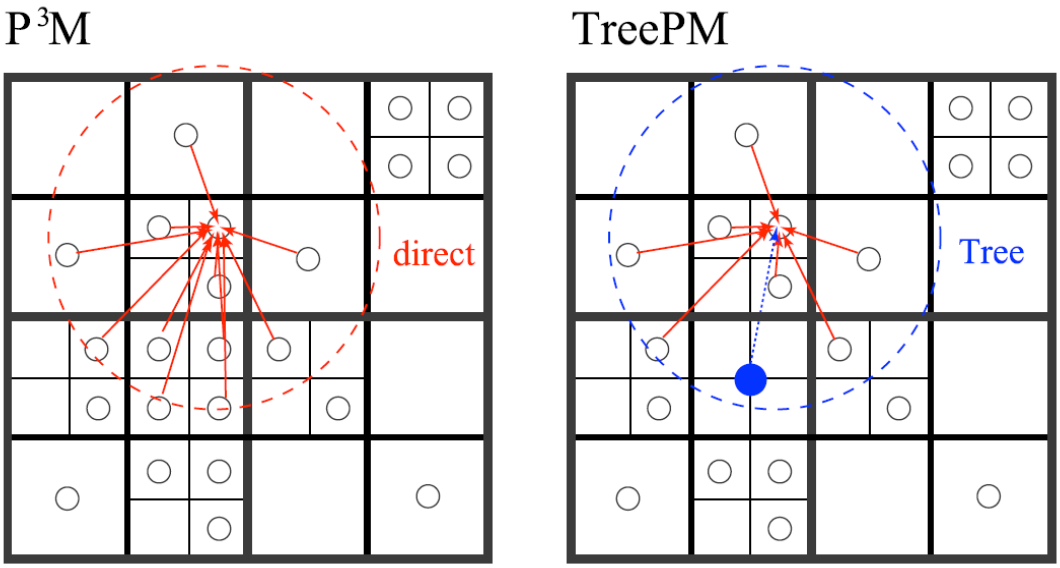

Figure 2. P^3M and TreePM

In order to map the problem to the distributed-memory parallel computer such as the K computer, we adopted the approach to divide the space into domains and assign particles in one domain to one calculation node. We used the orthogonal recursive multisection method developed by the team leader some years ago. It is the generalization of the orthogonal recursive bisection (ORB), which has been widely used in many parallel implementations of the tree algorithm.

With ORB, we recursively divide space into two halves, each with the same number of particles. An obvious disadvantage of the ORB approach is that it can utilize the computing nodes of integral powers of two. Thus, in the worst case we can use only half of the available nodes.

The difference between the multisection method and the ORB is that with the multisection method we allow the divisions to arbitrary number of domains, instead of bisection. This would allow too many possible divisions. In our current implementation, we limit the number of levels to three, and make the numbers of divisions at all levels as close as possible. Thus, our domain decomposition is topologically a simple three-dimension grid. This fact makes the multisection method well suited to the machines with the 3D torus network like the K computer.

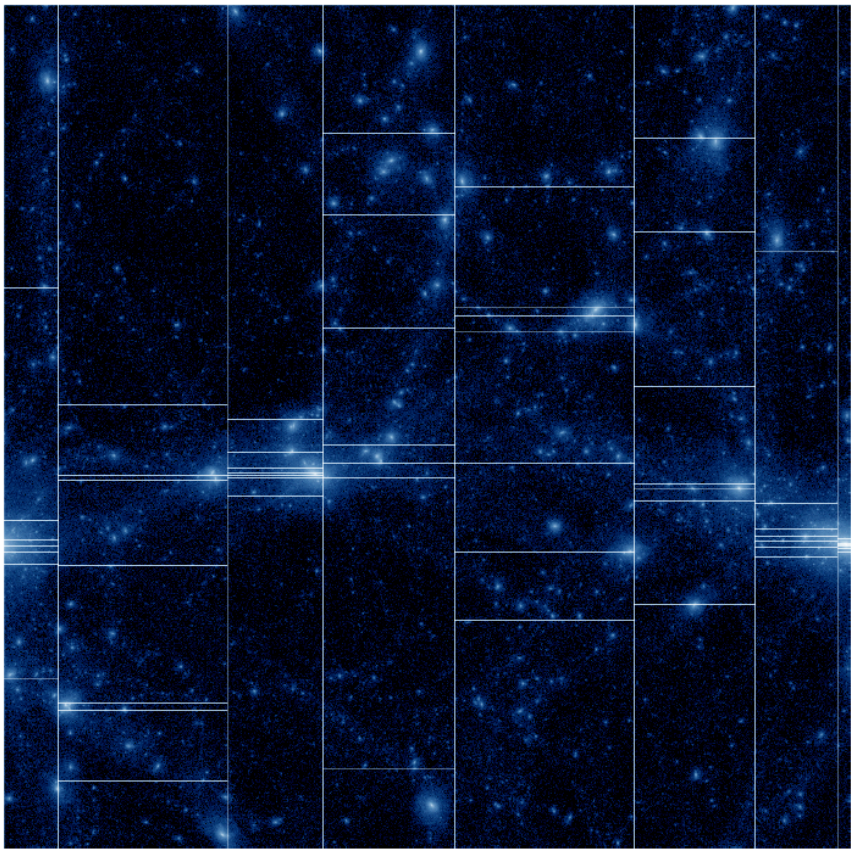

Figure 3. Recursive multisection in two dimension

Originally, with the ORB method the domains contain the same number of particles. It was soon realized that this division did not provide the best load balancing, and people started to use the number of interaction calculations as the measure for the calculation cost. We found that even that is not ideal, and have adopted a much better approach. We simply measure the cost in terms of the CPU seconds, and assign particles the average CPU time. Then we divide the space so that each node should require the same CPU time. This approach turned out to be able to achieve the near-ideal load balance.

Finally, in order to achieve the high efficiency, the efficiency of the force calculation kernel is extremely important. In the case of the K computer, we need to make use of the two-way SIMD unit, and the fact that two units are there. In order to achieve this goal, we developed a new expression for the spline kernel for the force cutoff, which requires only one masked operation. For the force kernel, we have achieved the performance of 72.8% of the theoretical peak or actually 97% of the theoretical limit when we take into account the fact not all floating-point operations are mapped to FMA operations.

Our current implementation is fairly straightforward, and there is nothing unusual. Thus, we have some difficulty in understanding why the competing code is much slower. The most likely reason is that the competing code is the modification of the P^3M code developed by the same group for the Roadrunner supercomputer, which has the IBM Cell processor. Either their code is not yet optimized for the BG/Q, or the original structure of the P^M code resulted in some intrinsic limitation of the performance.

11.3.2. Particle Simulation Platform.

Currently, we are in the design phase for the particle simulation platform. We have determined what algorithms we should use for the domain decomposition and load balance, but have not made the detailed specification yet.

11.3.3. Improvements on SPH.

SPH (Smoothed Particle Hydrodynamics) has been used in many field, including astrophysics, mechanical engineering and civil engineering. Recently, however, it was pointed out that the standard formulation of SPH has numerical difficulty at the contact discontinuity.

We have been working on the possible solution on this problem, and have made two significant steps in this year. The first one is the generalization of the density-independent SPH to an arbitrary

equation of state, and the second one is its further generalization which requires the continuity of neither density nor pressure.

The density-independent SPH is a new formulation of SPH we proposed in 2011. It uses the pressure, instead of the density, as the basic variable using which we evaluate the gradient of other quantities. With hydrodynamics, the pressure is continuous everywhere, except at the shock front. In the case of SPH, we use he artificial viscosity so that the physical variables are all continuous and differentiable even at the shock front. Thus, by using pressure as the basic variable, we can avoid the numerical difficulty associated with the contact discontinuity.

In the case of an ideal gas, we can calculate the pressure easily from the internal energy of particles, but if the equation of state is non-ideal, we cannot calculate the pressure explicitly. We can obtain the pressure by solving an implicit equation, and found that the additional cost of solving the equation is actually small. The reason is that we can also integrate the time evolution of the pressure, and therefore can obtain very good initial guess. Iteration with simple direct substitution is stable and fast enough.

11.4. Schedule and Future Plan

We plan to release the first prototype of the platform by FY 2014. It will have the basic abilities to run on large-scale parallel computers with reasonable load-balancing, for multiple forms of the interparticle interaction formula. We will extend this to fully user-specifiable interface to interparticle interactions in the future release.

References

Ishiyama, T. Nitadori, K, and Makino, J., 2012, 4.45 Pflops astrophysical N-body simulation on K computer: the gravitational trillion-body problem, SC '12 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis Article No. 5 IEEE Computer Society Press Los Alamitos, CA, USA ©2012 ISBN: 978-1-4673-0804-5

11.5. Publication, Presentation and Deliverables

(1) Journal Papers

1. Prog. Theor. Exp. Phys. (2012) 01A303 doi: 10.1093/ptep/pts029 Astrophysics with GRAPE *Junichiro Makino and Takayuki Saitoh
2. 10.1016/j.newast.2012.01.003 Farr, W. M., Ames, J., Hut, P., *Makino, J., McMillan, S., Muranushi, T., Nakamura, K., *Nitadori, K., and Portegies Zwart, S., "PSDF: Particle Stream Data Format for N-body simulations", NewA, 17, 520-523(2012)

3. 10.1016/j.newast.2011.09.001 Tanikawa, A., Hut, P., and *Makino, J., "Unexpected formation modes of the first hard binary in core collapse", NewA, 17, 272-280(2012)
4. 10.1016/j.newast.2011.07.011 *Saitoh, T. R. and *Makino, J., "A natural symmetrization for the plummer potential", NewA, 17, 76-81(2012)
5. 10.1088/0004-637X/746/1/26 Matsui, H., *Saitoh, T. R., *Makino, J., Wada, K., Tomisaka, K., Kokubo, E., Daisaka, H., Okamoto, T., and Yoshida, N., "Origin of Multiple Nuclei in Ultraluminous Infrared Galaxies", ApJ, 746,

(2) Conference Papers

-None

(3) Invited Talks

1. Junichiro Makino, Application and Architecture of Exascale Computing --- To make sure that we will not cry "This is not what I want!" ,RIRONKON Symposium 2012, "New developments of Computational Astrophysics", 2012/12/23
2. Junichiro Makino, Possibility of Galactic Paleoclimatology --- Global climate change of the Earth and the dynamics of the galactic disk, Symposium of GCOE program From The Earth To “Earths” Interdisciplinary Study On Habitable Planets, 2012/11/10
3. Junichiro Makino, From supercomputer development to science --- Lessons learned in GRAPE project. Meeting of JSPS committee No. 165 on Silicon VLSI system
4. Junichiro Makino, Exaflops and Beyond --- The view of the HPCI WG and my view Meeting on the Acceleration Technology --- Can we make computers equivalent to the human brain? 2012/9/6-7
5. Junichiro Makino, From the point of view of applications --- To make sure that we will not cry "This is not what I want!" HPC Forum, 2012/8/20
6. Junichiro Makino, Research on the future HPCI system with accelerators --- Basic concepts of the architecture 7th SDHPC workshop, 2012/7/31
7. Junichiro Makino, From the K computer to Exascale --- Development of supercomputers andthe role of National projects. Software Symposium SS2012, 2012/6/14

(4) Posters and presentations

1. Keigo Nitadori, “SIMD Class Library for HPC-ACE Instructions and its Application for N-body Kernel”, The 3rd AICS international Symposium, 28 Feb – 1 Mar 2013
2. Masaki Iwasawa, “Particle-Particle Particle-Tree scheme for dense stellar systems”, The 3rd AICS international Symposium, 28 Feb – 1 Mar 2013
3. Masaki Iwasawa, “Particle-Particle Particle-Tree scheme for dense stellar systems” HPCI

Strategic Program Field5 Symposium, 5-6 Mar 2013

(5) Patents and Deliverables

-None

(6) Awards

Gordon Bell Prize, Tomoaki Ishiyama, *Keigo Nitadori, *Junichiro Makino, SC12, Salt Lake City USA, 15th Nov. 2012