Particle Simulator Research Team

Team Leader
Junichiro Makino
jmakino@riken.jp

We have developed **FDPS** (Framework for Developing Particle Simulator), a universal software application that can be used to write high-performance, particle-based simulations.

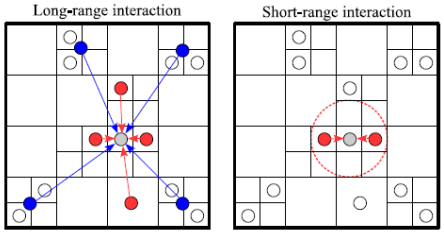
https://github.com/FDPS/FDPS

The Design of FDPS

FDPS is a C++ header library. A user of FDPS can develop the simulation code in the following three steps:

1. Define the data structure for the single particle, as a class in C++ language.
2. Define the interaction function f, that receives arrays of interacting particles, and calculates and accumulates the force on the particles.
3. Write the main program using the data class and functions provided by FDPS.

The Implementation of FDPS



FDPS handles long-range interactions by Burnes-Hut algorithm, and short-range interactions by tree-search based neighbor list. FDPS takes care of inter-node (MPI) and intra-node (OpenMP) communications, and load balancing. FDPS adopts three-dimensional multi-section domain decomposition, that is used in Gordon-Bell-prize-winning particle simulator GreeM.

The user code are passed small subset of particles delivered by FDPS. The user still have to SIMD-ize their interaction code for performance.

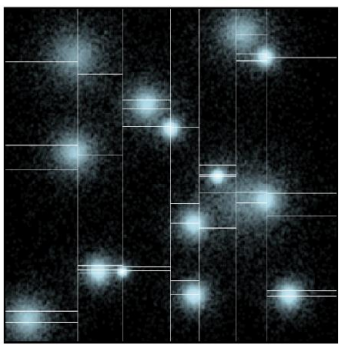
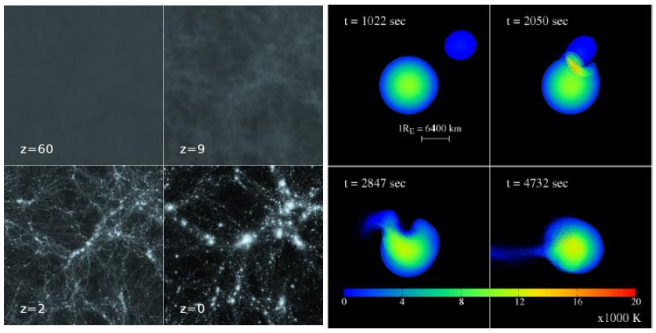


Figure: An Example of 7x6 domain decomposition.

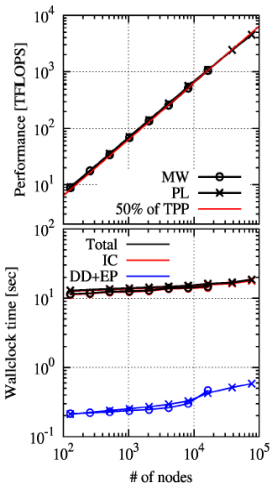
Applications

Simulations of cosmological structure formation with gravity code (left) and planet collision with smoothed-particle hydrodynamics code (right), both implemented using FDPS:



Performance

The left figure shows the performance of the gravity code, measured in the flops (top) and wall-clock time per timestep (bottom), as functions of the number of nodes. The code keeps 50% of the theoretical peak performance and scales up to 76,544 nodes, on the K computer.



Getting Started

FDPS is open-sourced under MIT license at
https://github.com/FDPS/FDPS
Tutorials, specifications, and ready-to-run example programs are found in the above URL.

With FDPS, our dream of letting one-core programmers to utilize large-scale supercomputers had come true, for particle simulation applications.

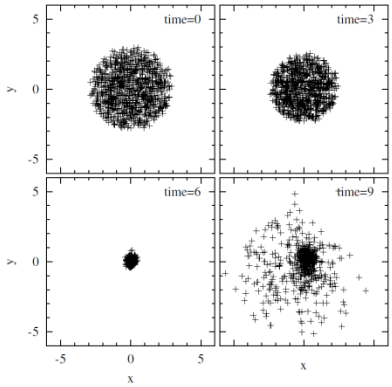
K computer

Getting Started with FDPS

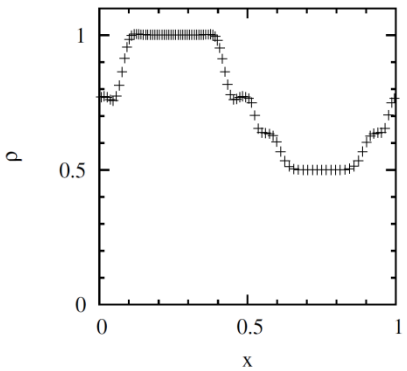
c.f. The FDPS tutorial available from **https://github.com/FDPS/FDPS**:
https://github.com/FDPS/FDPS/blob/master/doc/doc_tutorial_e.pdf

```
# Download FDPS via SVN
$ svn co --depth empty https://github.com/FDPS/FDPS
$ cd FDPS
$ svn up trunk
  -- or --
# Download FDPS via Git
$ git clone git://github.com/FDPS/FDPS.git
```

```
# Build and Run the N-body example
$ cd $(FDPS).sample/nbody/
$ make
$ ./nbody.out
$ gnuplot
> plot “result/0009.dat” using 3:4
```




```
# Build and Run the Smoothed Particle Hydro example
$ cd $(FDPS).sample/sph/
$ make
$ ./sph.out
$ gnuplot
> plot “result/0040.dat” using 3:9
```



N-body example with OpenMP & MPI

- Set the variable CC to your MPI C++ compiler
- Uncomment the line "CFLAGS += -DPARTICLE_SIMULATOR_THREAD_PARALLEL -fopenmp"
- Uncomment the line "CFLAGS += -DPARTICLE_SIMULATOR_MPI_PARALLEL“

Then

```
$ make
$ MPIRUN –np NPROC ./nbody.out
```

Here, "MPIRUN" should be mpirun or mpiexec depending on your MPI conguration, and "NPROC" is the number of processes you will use.