Parallelization of PIC code by OhHelp

2013.11.22
JIFT workshop

D.Nii¹, H.Sakagami²,
K.A.Tanaka¹, H.Habara¹

¹ Faculty of Engineering, Osaka University, 2-6 Yamada-oka,
Suita Osaka 565-0871, Japan

²Fundamental Physics Simulation Research Division, National Institute for Fusion
Sciences, 322-6 Oroshi-cho, Toki, Gifu 509-5292, Japan
Outline

• Fast Ignition (FI)
• Parallelizing simulation reproducing FI
• OhHelp
• Simulations
• Future work
Fast Ignition (FI) on Laser fusion

Concept of Fast Ignition
Heating the fuel by 2 steps
1. Compressing the fuel by high energy lasers
2. Heating the core with ultra-intense short pulse laser

Approaches to heat the core
Cone guiding:
Attached to the shell target to guide the short pulse laser
Super penetration:
Irradiating ultra-intense laser to high-density regions directly

1. compression
2. ignition
Analyze the laser propagation
-PIC simulation-

To achieve FI
We must understand the mechanism of laser propagation, relativistic nonlinear processes.

PIC simulation is the useful way to analyze

Simulation of high dense plasma requires much computer memory and CPU time.

→ Parallelization is needed.
Parallelization

Particle decomposition
• Advantage
  Perfect load balance can be achieved among the processors.
• Disadvantage
  Calculating $\mathbf{E}, \mathbf{B}$ cannot be parallelized.
  Increasing the processors, parallel efficiency becomes worse.

Domain decomposition
• Advantage
  It is possible to calculate efficiently in all step.

Is there any disadvantage?

Parallelization for FI simulation

Due to the nonuniform distribution of particles, degradation of parallel efficiency is expected. Dynamic load balance is needed. “OhHelp” is a promising way.

**Concept**

The little populated subdomains help the densely populated subdomain to calculate particles.

\[ P_d > (P / N)(1 + \alpha) \]

- \( P \) : total number of particles
- \( N \) : the number of nodes
- \( P_d \) : total number of particles in a subdomain \( d \)
- \( \alpha \) : tolerance factor

**Space Partitioning**

Black number : primary subdomain
Blue number : secondary subdomain

Load balance cycle
Choosing particles in heaviest-first subdomain to supply its particles to the lightest one

Every node, except for one node, has both primary particles and secondary particles.
After load balancing, OhHelp makes Helpand-Helper Tree. E and B is broadcasted through the branches.
load balancing & particle transfer
  exchange the histograms of particle population
all-reduce
  sum up the current density in helpand-helper family
broadcast
  send \mathbf{E}, \mathbf{B} from the helpand to its helpers
Implement OhHelp to

• field solve part
• particle push and load balancing part
• current scattering part
Purpose of the simulations

Simulation 1
Searching the relation between execution time and the number of subdomains

Simulation 2
Discovering the most efficient tolerance factor $\alpha$ depending on the mobility of particles

$$P_d > \left( \frac{P}{N} \right)(1 + \alpha)$$
Set-up - simulation 1 -

machine: Fujitsu PRIMERGY RX200 S7
CPU: Xeon processor E5-2650 (2GHz, 8 cores, 20MB) × 2
grid points: 1200 × 600
number of particles: 5068800
simulation time: 50 fs (2809 steps)
Processes: 2, 4, 9, 16
Tolerance factor $\alpha$: 5%
Laser intensity: $1.0 \times 10^{20} \text{[W/cm}^2\text{]}$
    wavelength: 1.06 [\mu m]
Almost linear increase of simulation speed is observed.
The ratios of communicating time are almost same among the number of subdomains.
machine: Fujitsu PRIMERGY RX200 S7
CPU: Xeon processor E5-2650 (2GHz, 8 cores, 20MB) \times 2
grid points: 1200 \times 600
number of particles: 5,068,800
simulation time: 50 fs (2809 steps)
Processes: 16

Laser 1
Laser intensity: \(1.0 \times 10^{20} [W / cm^2]\)
wavelength: 1.06 [\mu m]
Tolerance factor \(\alpha\) [%]: 1, 3, 5, 10, 15, 20

Laser 2
Laser intensity: \(1.0 \times 10^{25} [W / cm^2]\)
wavelength: 1.06 [\mu m]
Tolerance factor \(\alpha\) [%]: 5, 10, 15, 20

low mobility of particles
high mobility of particles
Result

-simulation2-

\[ \alpha_{\text{max}} : \alpha \text{ at the maximum point of PPS} \]

<table>
<thead>
<tr>
<th>tolerance factor ( \alpha )</th>
<th>PPS ([10^4 \text{ particles}])</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1908</td>
</tr>
<tr>
<td>3</td>
<td>1989</td>
</tr>
<tr>
<td>5</td>
<td>1898</td>
</tr>
<tr>
<td>10</td>
<td>1799</td>
</tr>
<tr>
<td>15</td>
<td>1704</td>
</tr>
<tr>
<td>20</td>
<td>1798</td>
</tr>
</tbody>
</table>

Laser 1: \( \alpha_{\text{max}} = 3 \)

Laser 2: \( \alpha_{\text{max}} = 10 \)

Mobility of Particles is higher, \( \alpha_{\text{max}} \) becomes larger.
Summary

Simulation 1

- Linear increase of simulation speed is observed with less than 16 processes.
- Significant increase of communicating time is not observed from 2 processes to 16 processes.

It is expected that simulation speed increases linearly with more than 16 processes.

Simulation 2

It is important to change $\alpha$ depending on the mobility of particles.
Future work

• Implement OhHelp to current scattering part
• Calculating the simulation time by a server, having more than 16 nodes
• Simulating high-dense plasma and reproducing FI

15000×8000 grids
256 processes
Thank You!