UN/Japan Workshop on Space Weather, 2015

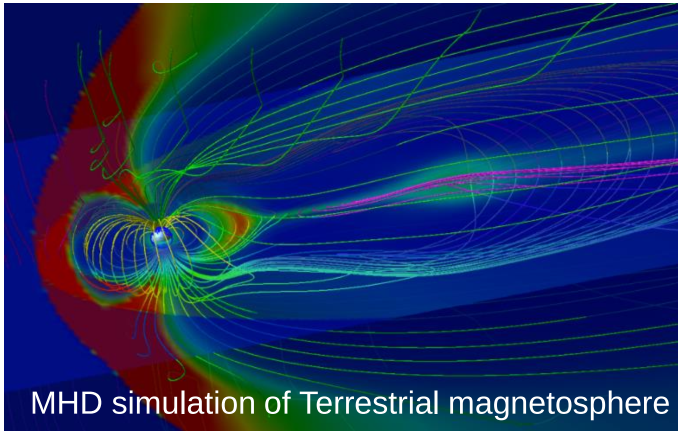
Optimization of MHD Simulation for Space Weather to Manycore Processor

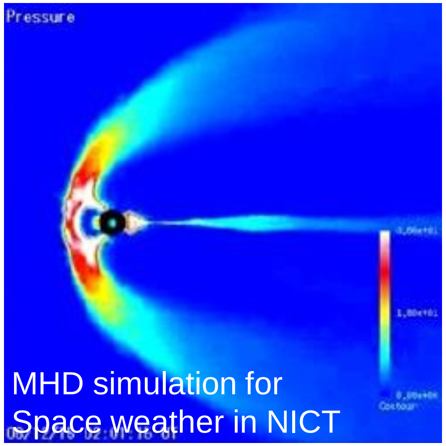
Keiichiro FUKAZAWA[1, 2], Akimasa YOSHIKAWA[3, 4]

1. Academic Center for Computing and Media Studies, Kyoto University, Japan
2. CREST, JST, Japan
3. Department of Earth and Planetary Sciences, Faculty of Sciences, Kyushu University, Japan
4. International Center for Space Weather Science and Education, Kyushu University, Japan

MHD Simulation of magnetosphere

- MHD simulation is often used to simulate the global configuration and dynamics of planetary magnetosphere.
- Recently the combination simulations of magnetosphere and ionosphere, radiation belt, etc. are performed to forecast the space weather.
- Thus the important of global MHD simulation of magnetosphere plays more important roles.


MHD simulation of Terrestrial magnetosphere

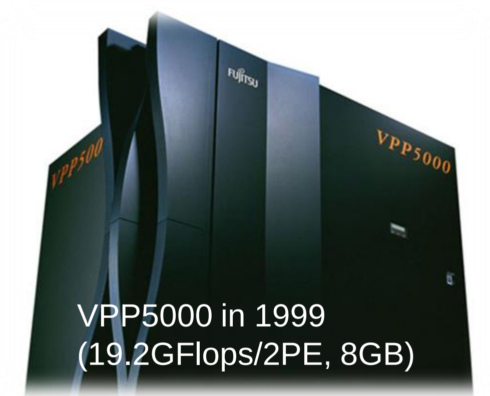


MHD simulation for Space weather in NICT

Introduction | Supercomputer

Transition of supercomputer

- In 1990’s the vector-type supercomputer systems were popular.
- Then in 2000’s the scalar-type supercomputer became the major architecture such as Xeon, POWER, SPARC.
- Recently GPU and MIC which are the accelerators and manycore system have achieved the top 1 performance in the world.



VPP5000 in 1999
(19.2GFlops/2PE, 8GB)

Tesla K20X in 2012
(dp:1.31TFlops, sp:3.95TFlops)


Tianhe-2 in 2013 Xeon + Xeon Phi
Top 1 supercomputer (33.86PFlops)

Motivation

We want to run the space weather simulation faster!!

For application developers we do not have likes and dislikes if the simulation will finish quickly

- Either CPU（x86, POWER,SPARC, etc) or accelerator (GPU, MIC) is welcome.

Recently and to the exa-scale computing era, there are the many core systems which include GPU and MIC.

- Really high performance???
- How hard to optimize the simulation code to those systems???

Evaluate the performance of MHD code using MIC and GPU and compare the results with performance of CPU!

MHD Code | governing equation

MHD (Magnetohydrodynamics) equation

Derived from the moment of Vlasov equation

- From 0th-order, 1st-order and 2nd-order moment, we obtain

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\mathbf{v}\rho)$$

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v}\cdot\nabla)\mathbf{v} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\mathbf{J}\times\mathbf{B}$$

$$\frac{\partial p}{\partial t} = -(\mathbf{v}\cdot\nabla)p - \gamma p \nabla\cdot\mathbf{v}$$

Combination of these equations and the following Induction equation

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla\times(\mathbf{v}\times\mathbf{B})$$ is MHD equation.

→ without the velocity space, we can calculate the MHD equation using the present supercomputer.

MHD Code | numerical method

Numerical simulation code

- Our three-dimensional MHD code uses the “Modified Leap frog (MLF)” method [*Ogino et al.*, 1992].
- Using MLF method, partial difference equations are solved by the two-step Lax-Wendroff method for one time step and then by the LF method for $(l-1)$ time steps and the procedure is repeated.
- MLF method is a kind of combination technique which balances numerical stability of the two step Lax-Wendroff method and dissipationlessness of the LF method.

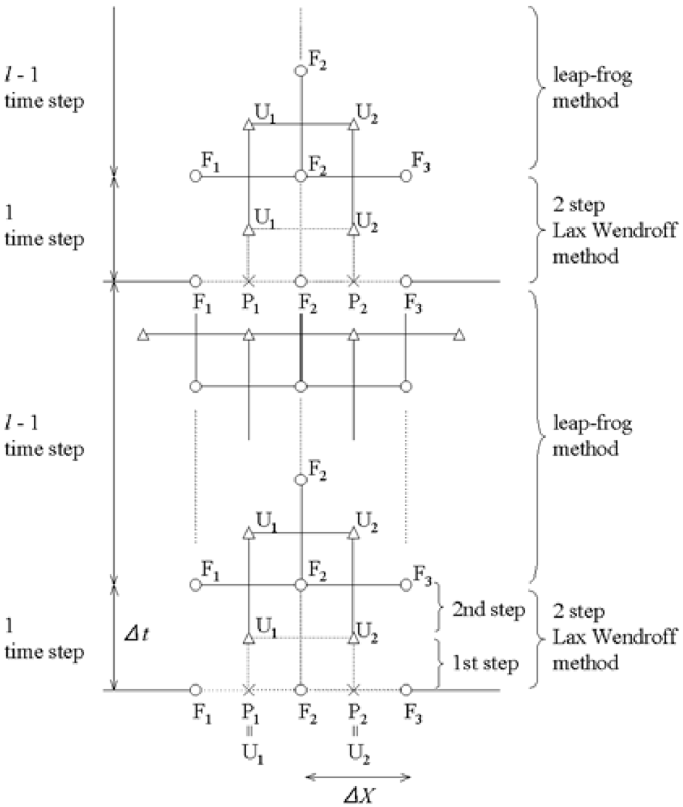


Fig. 1. Diagram of Modified Leap frog method

MIC Evaluation | Simulation setting

Basic simulation size

- 10MB/core is used
- Considering the workspace, 50MB/core is used additionally
- Weak scaling

Parallelization

- Evaluate 1D, 2D and 3D domain decompositions
- MPI_sendrecv (blocking comunicxation) is used for halo communication

Array order

- Type A: $f(i, j, k, m)$, Type B: $f(m, i, j, k)$

How to control the MIC

- Native mode (use as the CPU not accelerator)

MIC Evaluation | environment

Character of Xeon Phi

Xeon Phi is the first product of MIC (many integrated core) by Intel.

Table 1. Characters of Xeon Phi

MIC	Architecture	Xeon Phi 5110P (Knights Corner)
	Number of core	60 cores (240 threads)
	Frequency	1.053 GHz
	Cache	L2: 30 MB/CPU
	Rmax	1.01 TFlops
	Memory size	8 GB (GDDR5)
	Bandwidth	320 GB/s
	B/F	0.32
CPU	Architecture	Xeon E5 2643 3.3 GHz
	Memory size	64 GB
System	Number of nodes	2
	MIC per node	1

MIC Evaluation | results

Performance of Multi Xeon Phi

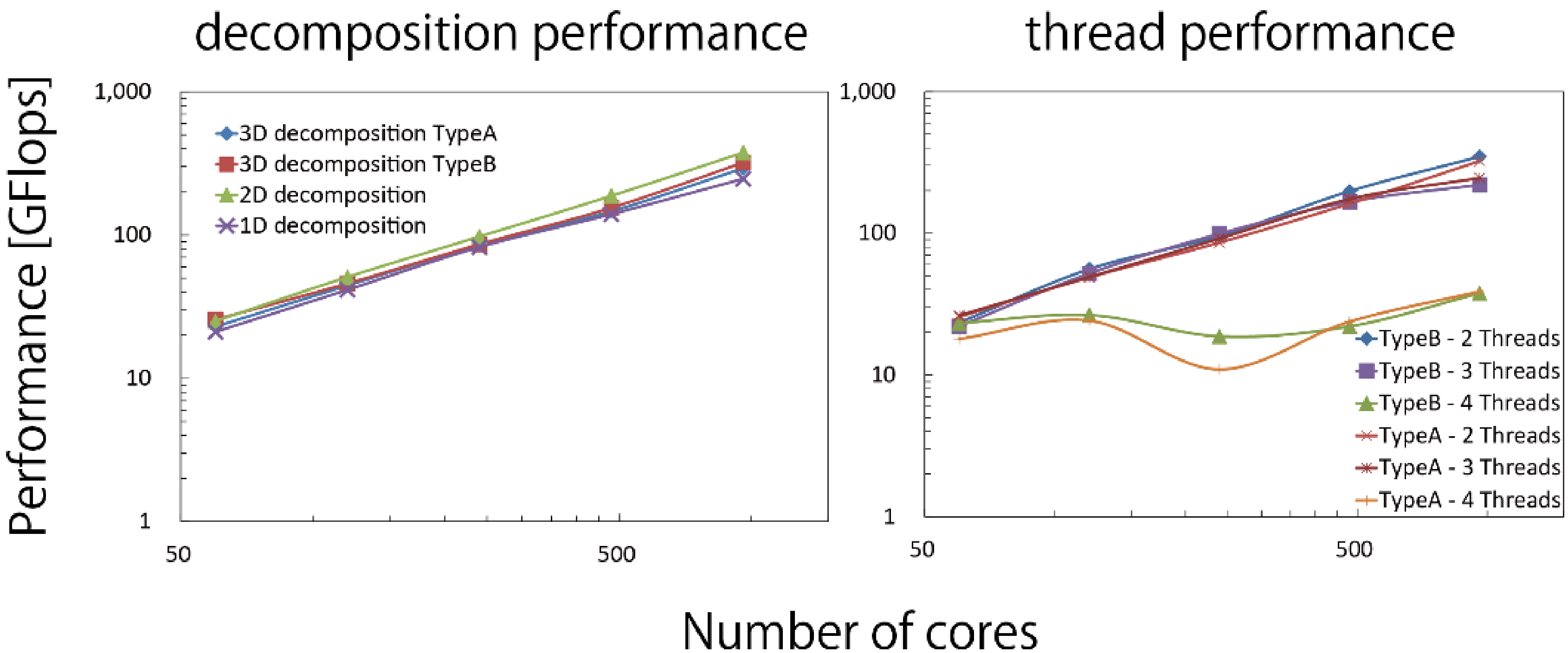


Fig. 5. Performance results of MHD simulation on Xeon Phi 5120D（CRAY XC30）

We obtained 48.97 GFlops with 3D typeA
on one Xeon Phi at maximum.

MIC Evaluation | optimization

Optimization for Xeon Phi

64 byte alignment of array

- 64 byte aligned load/store is used high efficient operation in Xeon Phi

  ex)Fortran version

  ```
  real*4 A(1024,100)
  !DEC$ATTRIBUTES ALIGN: 64:: A
  ```
  ※A is $4\times1024\times100=64^2\times2^2\times5^2$

Compile option

- `-opt-prefetch-distance=64,8`

  prefetch control option

- `-opt-streaming-stores always`

  use the streaming store order as regular store order

- `-opt-streaming-cahce-evict=0`

  do not use the cache clear order to streaming store order

MIC Evaluation | optimization results

Tuning results for Xeon Phi

- Alignment of array
  48.97 GFlops → 79.27 GFlops
- Compile option
  79.27 GFlops → 83.93 GFlops

From these optimizations we obtain almost double performance than the non optimized performance!!

GPU Evaluation | Simulation setting

Basic simulation size

- 200MB/core is used
- Considering the workspace, 1.0GB/core is used additionally
- Weak scaling

Parallelization

- Evaluate 3D domain decomposition
- MPI_sendrecv (blocking comunicxation) is used for halo communication

Array order

- Type A: $f(i, j, k, m)$

GPU coding

- Use OpenACC (directive base not CUDA)

```
!$acc kernels pcopy(f)
!$acc loop collapse(3) gang vector(128)
```

GPU Evaluation | environment

Character of GPU

This GPU is installed in the supercomputer system CX400 at Kyushu University.

Table 2. Characters of Tesla K20m

GPU	Architecture	Kepler
	Number of core	832 (double precision)
	Frequency	706 MHz
	Rmax	1.17 TFlops (double precision)
	Memory size	5 GB (GDDR5)
	Bandwidth	208 GB/s
	B/F	0.18
CPU	Architecture	SB Xeon E5 2.7 GHz
	Memory size	128 GB
System	Number of nodes	1476
	GPU per node	1 (in 386 nodes)

GPU Evaluation | results

Performance of GPU

- The scalability of GPU is little bit worse than that of CPU.
- The efficiency is 14.7% at one node and 13.4% at 8 nodes.
- The performance of CX400 at 1 node is 69.12GFlops with 20 % efficiency thus the performance is half of GPU performance in this case.

Table 3. Performance results of MHD simulation on Tesla K20m

Number of GPUs	1	2	4	8
Performance [GFlops]	153.34	299.80	581.25	1121.86
Scalability		0.97	0.95	0.91

Comparison of CPU and MIC・GPU

Comparison of MIC and GPU with other computer systems

Table 4. Performance evaluation of MHD code on various computer systems.

	Core/CPU	Rpeak [TFlops]	Rmax [TFlops]	Rpeak /CPU [GFlops]	Efficiency [%]	Suitable domain decomposition	CPU architecture
SX-9	64/64	2.19	6.55	34.2	33	2D	Vector
HX600	1024/256	2.17	10.24	8.5	21	3D_A	Opteron (Shanghai)
XE6	8192/512	14.16	81.92	27.7	17	1D or 2D	Opteron (Interlagos)
RX200S6	864/144	3.51	10.13	24.4	35	3D_A	Xeon (Westmere)
CX400	23616/2952	104.23	510.11	35.3	20	3D_A	Xeon (SB)
HA8000	23160/1930	83.42	500.26	43.2	17	2D	Xeon (IB)
XC30-HSW	448/32	1.37	16.49	42.8	8	2D	Xeon (HSW)
FX1	1024/256	2.08	10.24	8.1	21	3D_B	SPARC64VII
K	26144/32768	914.12	4194.30	27.9	22	3D_B	SPARC64 VIIIfx
FX10	76800/4800	234.59	1135.41	48.9	21	3D_B	SPARC64 IXfx
SR16000/L2	1344/672	5.38	25.27	8.0	21	3D_B	POWER6
Xeon Phi	60/1	0.08	1.01	83.9	8	3D_A	Knights Corner
CX400/GPU	8/8	1.12	8.34	140.2	13	3D_A	Tesla (Kepler)

KYOTO UNIVERSITY

Estimation for Real-time Simulation

To perform the simulation in real-time

From this study if the array size of 1GB is used and 64 times calculated then it takes 32.228 sec and reaches 83.93GFlops.

$90R_E \times 60R_E \times 60R_E$ with $0.1R_E$ grid interval

- $900 \times 600 \times 600 \rightarrow 9.66$GB calculation size
- To proceed 1 sec in the simulation, it is necessary to calculate the simulation with 43 times

→it takes 209.1 sec to proceed 1 sec with 83.93GFlops

→210 nodes of Xeon Phi, 125 nodes of GPU and 16 nodes of Haswell can calculate the magnetosphere in real-time

$90R_E \times 60R_E \times 60R_E$ with $0.2R_E$

- It is required 1/16 resources of above simulation size

  →14 nodes of Xeon Phi, 8 nodes of GPU, 1 node of Haswell

Summary

Performance evaluation of MHD code with GPU and MIC

- ✓ Using Xeon Phi 3D domain decomposition with Hybrid MPI (60 processes + 4 threads) achieve the best performance (48.97 GFlops) without tuning.
- ✓ Optimization for Xeon Phi (64 byte alignment of array and compiler option) makes the performance 83.9GFlops.
- ✓ Using 4 nodes of GPU we obtain the performance of 1TFlops with OpenACC.
- ✓ Calculation efficiency of Xeon Phi is 1/3 of CPU however the effective performance becomes twice of SPARC64 Ixfx which is the CPU of FX10.
- ✓ Performance of GPU becomes double performance of Sandy Bridge Xeon.
- ✓ It is possible for an individual researcher to perform the real-time space weather simulation with $0.2R_E$ resolution.