ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

On Present Status of Vlasov Simulation

QuESpace Science Club 14.1.2011

I. Honkonen & A. Sandroos

Contents

Present Status

Since last April, we have:

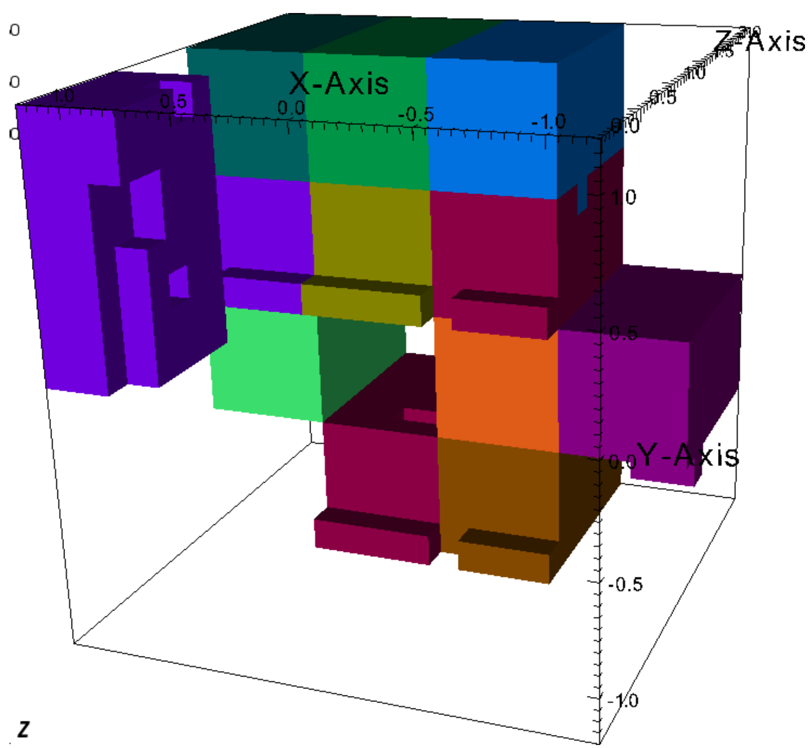
- Implementation of 6D Vlasov sim.
    - No field solver, AMR, ...
- Runs in meteo (MPI+OpenMP)
- Some global-scale tests have been made (I.H.)

Outstanding problems:

- Scalability / load balancing (?)
- *Very long* run times

MPI Partitioning

Distribution of cells to N (MPI) processes. We use Zoltan library which supports many partitioners.



16x16x16 spatial grid

"Simple" geometric partitioning into smaller cubes.

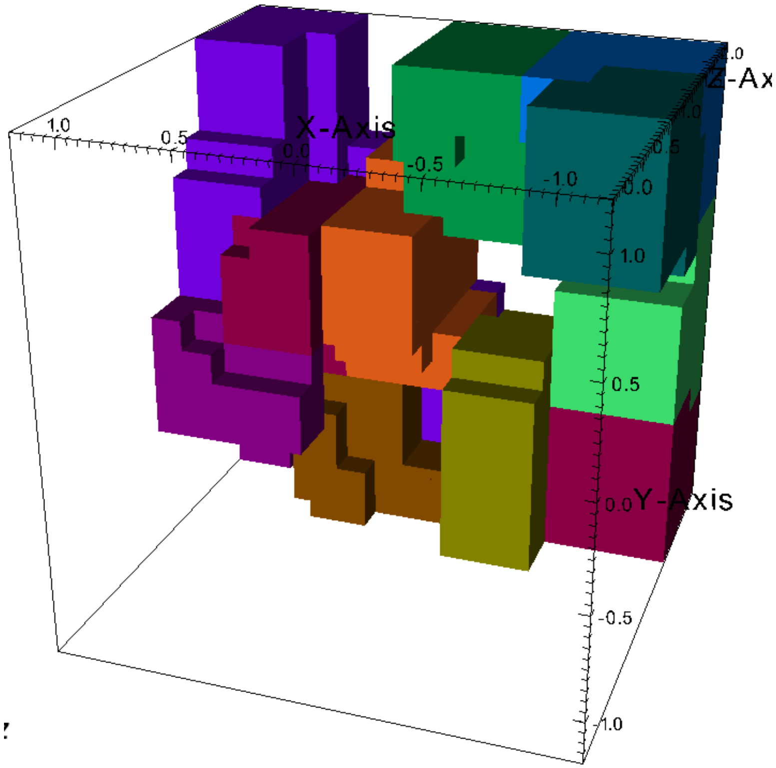
RCB in Zoltan

Usually not optimal.

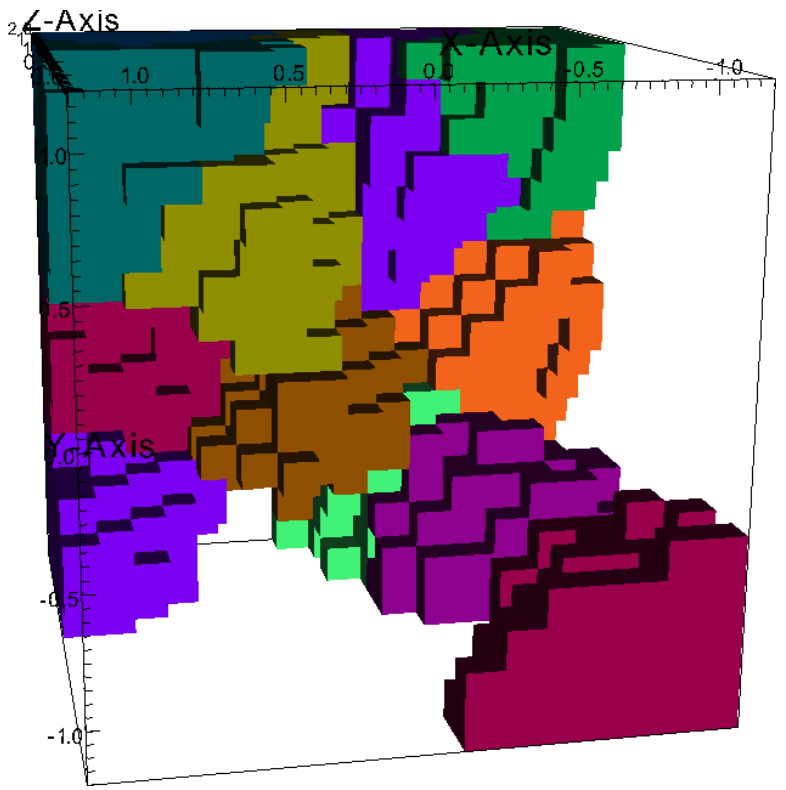
ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

MPI Partitioning

Graph

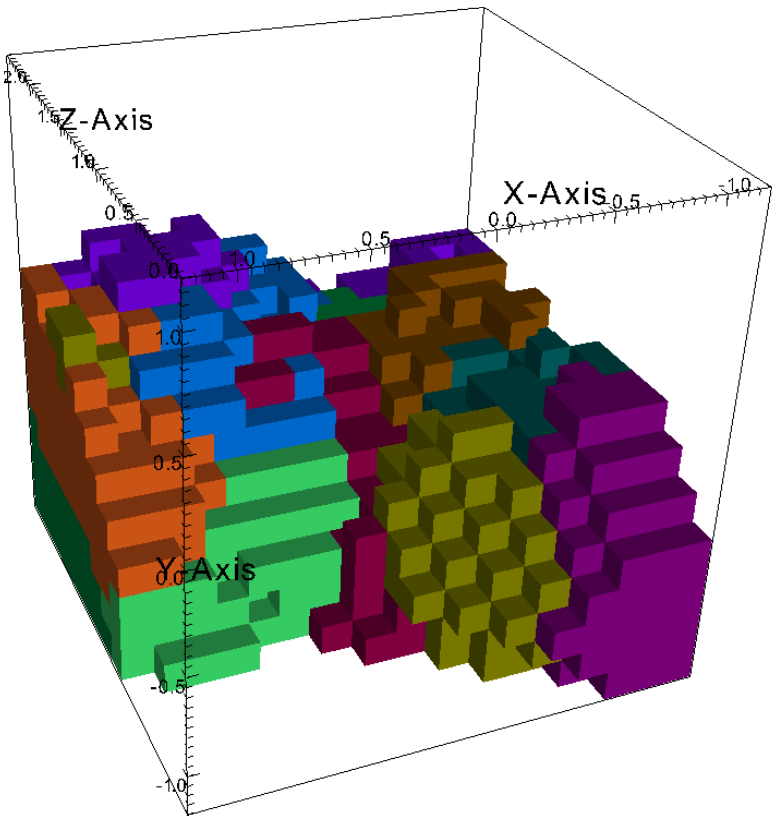


Hypergraph



MPI Partitioning

HG + HG Hierarchical



Partitioners often assume a homogeneous computing environment, which many supercomps really are not (meteo).

Computing node in meteo consists of 2 6-core CPUs which share memory.

Thus, MPI communication within the node is faster than comm. to other nodes.

Zoltan supports hierarchical partitioning.

MPI Partitioning

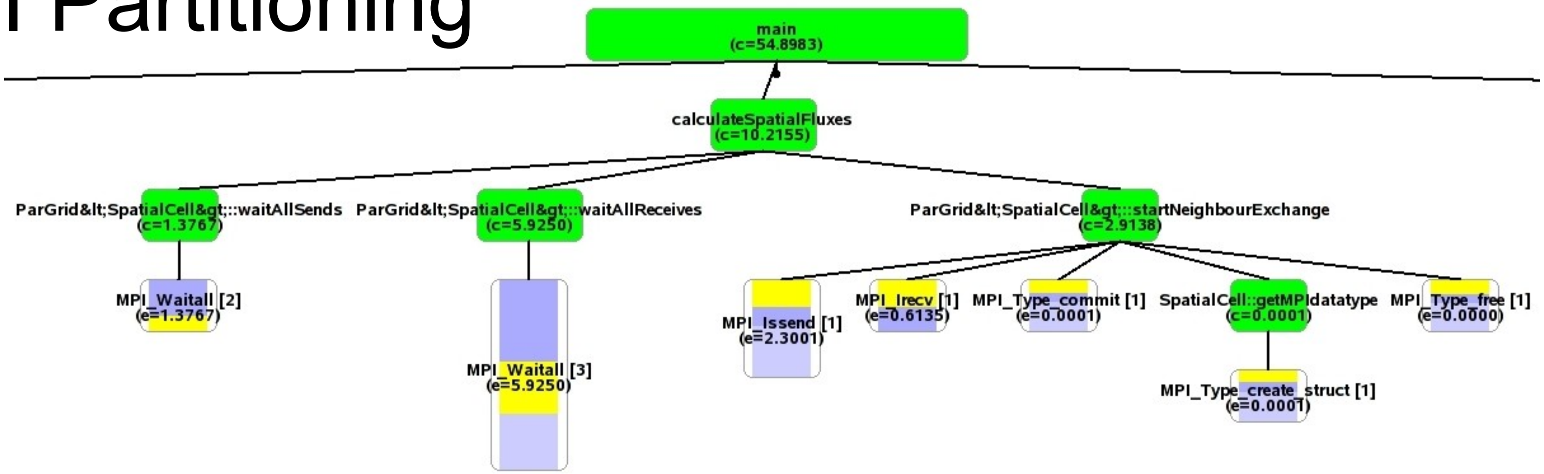
Meteo: 36 MPI procs (3 nodes)

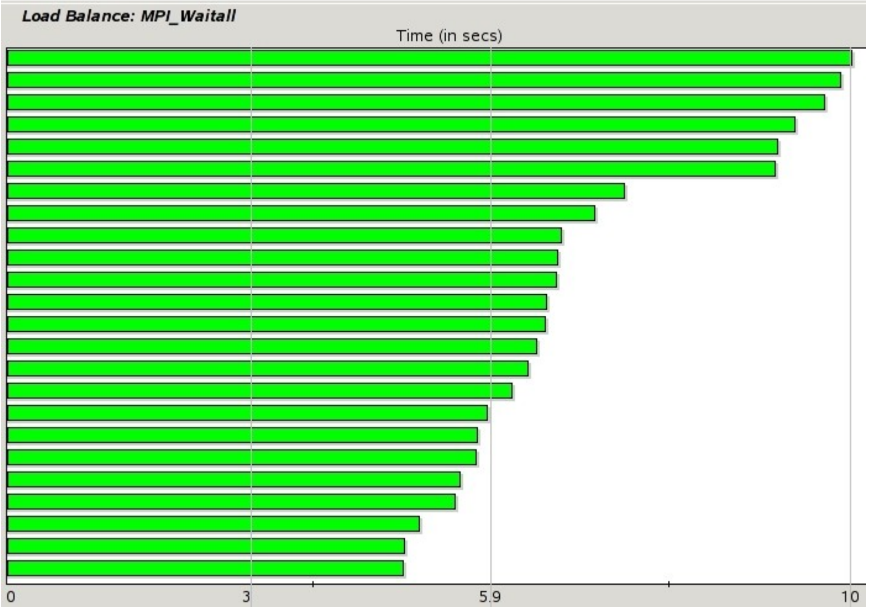
Method	Time (seconds)
RCB	213
Graph	240
Hypergraph (HG)	211
Hierarchical (HG+HG)	175 (~20% faster)

This example is not terribly good – RCB is surprisingly fast and graph is slow. HG+HG created 3 superpartitions, and each superpartition was further partitioned into 12 parts.

ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

MPI Partitioning





There are tools in meteo for profiling load balance.

Scalability

In order to get supercomputing time, code has to meet some scalability criterion.

Strong scaling: same workload, increase number of processes

$$\frac{t_n}{t_{2\mathrm{n}}} = 2$$

Weak scaling: increase no. processes & workload proportionally

$$\frac{t_n}{t_{2\mathrm{n}}} = 1$$

Morale: transfer only what you absolutely must, and *partition well*.

Scalability

Strong scaling: starting to run out of cells with 144 processes.

Processes	Time (seconds)	Scaling factor	Cells per proc.
36	175		114
72	97	1.80	57
144	67	1.45	28

Each spatial cell contains 40x40x40 (=64000) velocity grid

Scalability

Weak scaling: run times should be equal

Processes	Nodes	Time (s)	Scaling
24	2	130	1.00
36	3	171	1.32
72	6	182	1.40
360	10	199	1.53
720	20	197	1.52
1200	100	198	1.52
1440	120	201	1.55

114 spatial
cells per process

This does not look too bad.

Intro to Load Balancing Problem

- Hundreds of processes calculating stuff and sending data to other processes at every timestep
- The volume of spatial cells is well balanced between processes, e.g. all processes have about as many calculations to do
- But one or a few processes have to send / receive as much as 50 % more data from other processes

ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

Intro to Load Balancing Problem

Prosesses	Volume	Calculation time	Data transfer time
N	1 l	1 s	1 s
2N	0.5 l	0.5 s	0.63 s

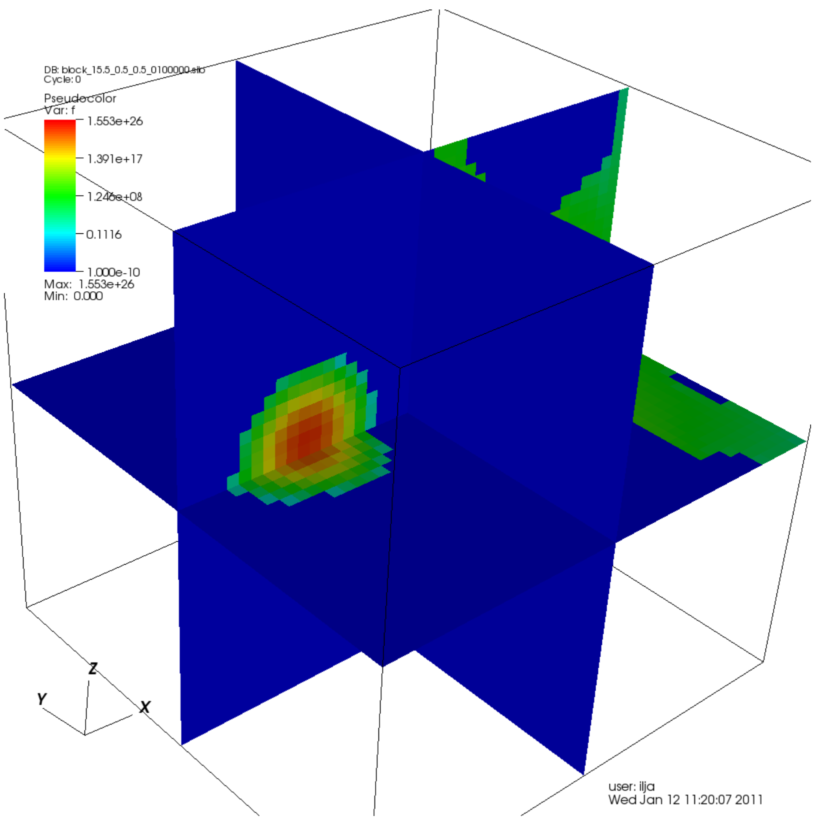
- When N -> 2N, those few processes with 50 % more data to transfer will keep twise the number of processes waiting even longer (relatively) than previously, which can't be good for scalability
- Above assumes that transfer time depends linearly on the abount of data: always sending only 2 messages / process could be more efficient for certain volumes

ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

Tests

- Rotation in velocity in constant Bx, y, z
- Harmonic ~1d oscillator
- Test particle simulation in GUMICS fields

15.5 Re



7.5 Re