Analysis of Vlasiator with BSC Tools

Valentin Seitz, BSC

48th VI-HPS Tuning Workshop 09.02.2026

HORIZON-EUROHPC-JU-2023-COE

EuroHPC Joint Undertaking

1 January 2024– 31 December 2026

Grant Agreement No 101143931

Content

- The ecosystem of BSC performance tools
  - TALP
  - Extrae
  - Paraver
- Show case of a performance analysis using the BSC performance tools

TALP

Catalan for mole

A spy monitoring your applications resource usage.

TALP

- Profiler that collects POP efficiency-metrics.
- Supported programming models:
  - MPI
  - OpenMP
  - CUDA + HIP
- Provides a API to:
  - query metrics at runtime
  - annotate code regions

```
############### Monitoring Region POP Metrics ################
### Name:                                      Global
### Elapsed Time:                              10.16 s
### Host
### ----
### Parallel efficiency:                       0.99
###   - MPI Parallel efficiency:               1.00
###     - Communication efficiency:            1.00
###     - Load Balance:                        1.00
###       - In:                                1.00
###       - Out:                               1.00
###   - Device Offload efficiency:             0.99
###
### NVIDIA Device
### -------------
### Parallel efficiency:                       0.50
###   - Load Balance:                          1.00
###   - Communication efficiency:              1.00
###   - Orchestration efficiency:              0.50
```

Example output of TALP of an application on 2 GPUs with MPI

Extrae

Spanish “Extracting”

The tracer collecting the information from hardware and the programming models

Extrae

No need to recompile or relink

- Trace producer that is transparent to the application
- Parallel programming models:
  - MPI, OpenMP, pthreads, OmpSs, CUDA, HIP
- Hardware counters (PAPI)
- Link to source:
  - Callstack at MPI routines
  - OpenMP outlined routines
  - Selected user functions (Dyninst)
- Periodic sampling
- API to annotate your code with custom events

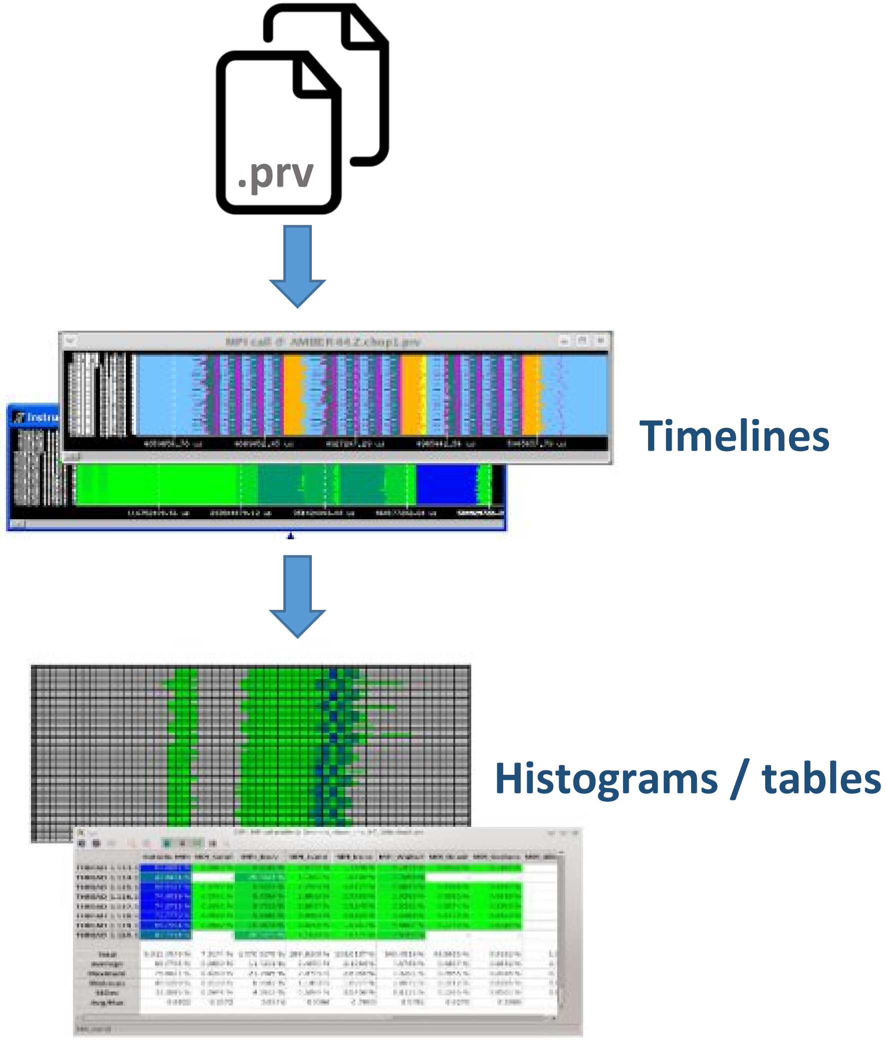
Paraver

Spanish “to see”

The flexible trace viewer visualizing the data.

Paraver

- (Performance) Data Visualizer
  - Any kind of timestamped data
  - Trace Visualization and analysis
  - Flexible
    - No pre-assumed semantics
    - Programmable
  - 2 Kind of views:
    - Timeline
    - Histograms, 2D and 3D tables



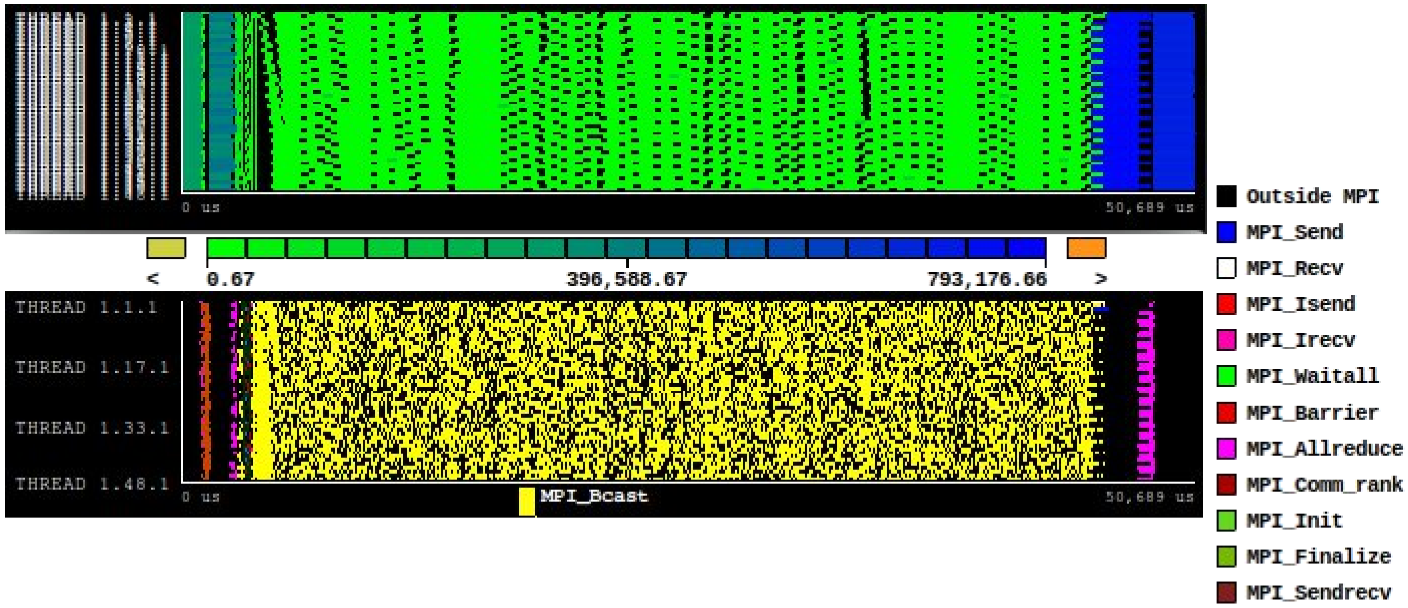
Paraver: Timelines

Every trace is a function of time of certain data (MPI calls, CPU Frequency,..)

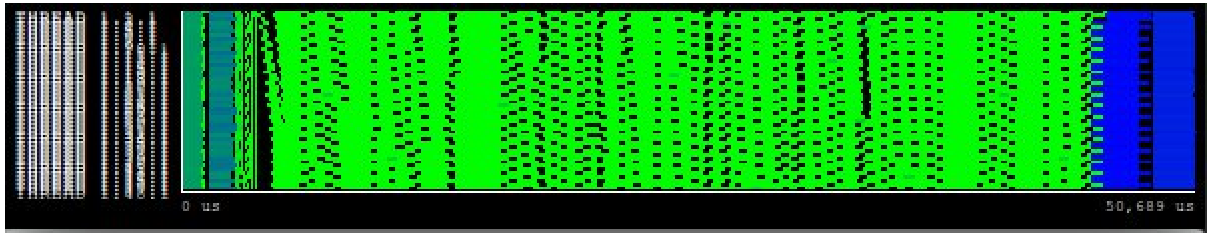
Different semantics require different visualizations.

Granularity between runtime calls;
IPC; Network Bandwidth ...

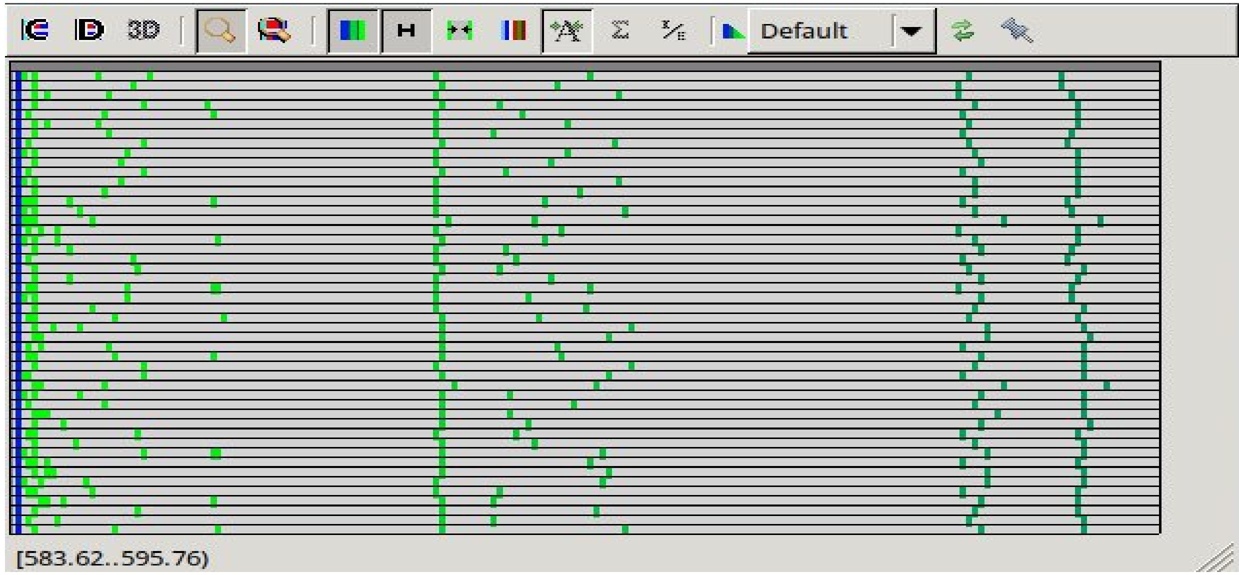
MPI Calls; Functions; CUDA kernels ...



Paraver: Histograms



	Outside MPI	MPI_Send	MPI_Recv	MPI_Bcast	MPI_Allreduce	MPI_Comm_rank	MPI_Comm_size
THREAD 1.1.1	23.74 %	0.26 %	1.40 %	72.80 %	0.48 %	0.03 %	0.03 %
THREAD 1.2.1	24.55 %	0.19 %	0.23 %	72.50 %	0.77 %	0.03 %	0.03 %
THREAD 1.3.1	22.63 %	1.47 %	0.34 %	73.42 %	0.38 %	0.03 %	0.03 %
THREAD 1.4.1	23.79 %	-	-	72.27 %	2.29 %	0.03 %	0.03 %
THREAD 1.5.1	23.47 %	-	-	73.83 %	1.06 %	0.03 %	0.02 %
THREAD 1.6.1	23.34 %	-	-	72.99 %	2.03 %	0.03 %	0.03 %
THREAD 1.7.1	22.74 %	-	-	73.19 %	2.36 %	0.04 %	0.03 %
THREAD 1.8.1	23.58 %	-	-	74.02 %	0.69 %	0.03 %	0.03 %
THREAD 1.9.1	22.66 %	-	-	73.59 %	2.05 %	0.04 %	0.03 %
THREAD 1.10.1	22.93 %	-	-	73.34 %	2.10 %	0.04 %	0.02 %
THREAD 1.11.1	23.02 %	-	-	74.22 %	1.11 %	0.03 %	0.03 %
THREAD 1.12.1	23.94 %	-	-	72.64 %	1.78 %	0.03 %	0.03 %
THREAD 1.13.1	23.04 %	-	-	73.28 %	1.99 %	0.04 %	0.03 %
THREAD 1.14.1	24.67 %	-	-	72.64 %	0.99 %	0.03 %	0.03 %
THREAD 1.15.1	22.88 %	-	-	73.61 %	1.80 %	0.03 %	0.03 %
THREAD 1.16.1	22.96 %	-	-	73.46 %	1.98 %	0.04 %	0.03 %
THREAD 1.17.1	22.96 %	-	-	74.58 %	0.88 %	0.03 %	0.03 %
THREAD 1.18.1	23.72 %	-	-	72.61 %	2.09 %	0.03 %	0.03 %
THREAD 1.19.1	23.33 %	-	-	72.71 %	2.28 %	0.04 %	0.03 %
THREAD 1.20.1	22.98 %	-	-	74.26 %	1.09 %	0.03 %	0.03 %
THREAD 1.21.1	23.03 %	-	-	73.04 %	2.27 %	0.03 %	0.03 %
THREAD 1.22.1	22.96 %	-	-	73.35 %	2.06 %	0.04 %	0.02 %



EuroHPC
Joint Undertaking

Performance analysis of Vlasiator

using BSC performance tools

Vlasiator

- Scientific field: near-earth plasma simulations
- Programming: C++, MPI+OpenMP
- Input: Restart from timestep 82848 of recent unpublished production run. 4.6TB

- Production run:
  - 6400 MPI x 16 OpenMP (SMT enabled) @Mahti (500 Nodes)
- Our runs:
  - 3500 MPI x 8 OpenMP (SMT disabled) (250 Nodes)
  - 7168 MPI x 4 OpenMP (SMT disabled) (256 Nodes)
  - 7000 MPI x 8 OpenMP (SMT disabled) (500 Nodes)

Vlasiator: Timings

- Runtime on 250 nodes: 36.8s
- Runtime on 500 nodes: 17.3s

  —> Superlinear Speed-up of 2.1

- Runtime on 1000 nodes: ~18s

Parallel efficiency 250 nodes: 0.15 (measured by TALP)
Parallel efficiency 500 nodes: 0.14 (measured by TALP)
Parallel efficiency 1000 nodes: 0.11 (measured by TALP)

Steps

Add custom events through internal profiler

↓

Gather metrics with TALP

↓

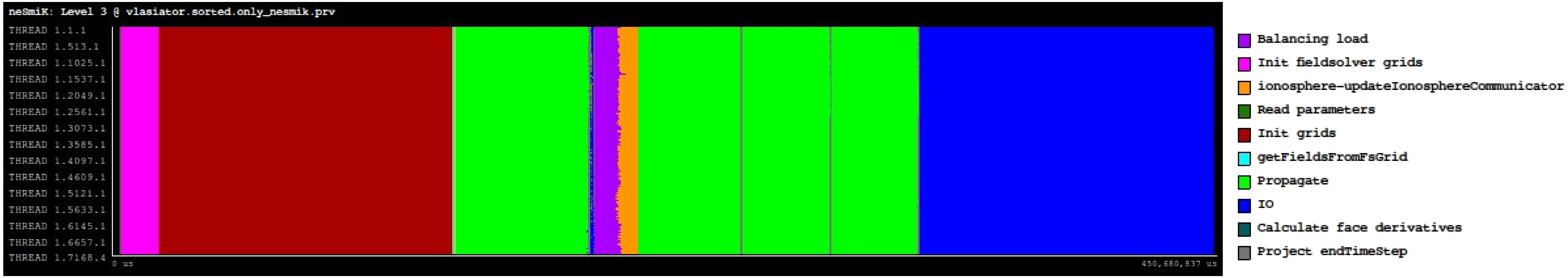
Selective tracing of interesting regions

↓

Detailed performance analysis

Custom events from internal profiler

Main regions identified in trace:



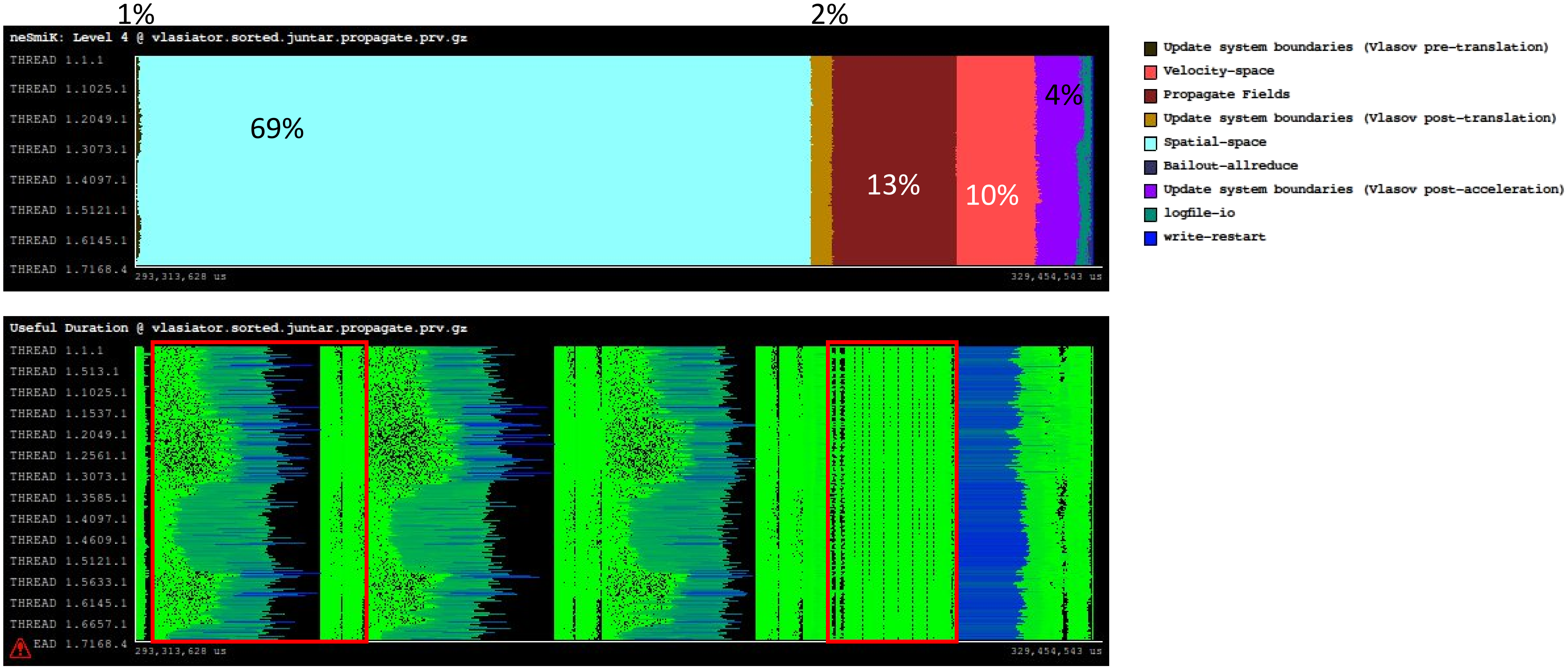
Metrics for `Propagate` slice

Metrics	250 Nodes
Global Effiency	0.15
- Parallel efficiency	0.15
-- MPI Parallel efficiency	0.25
--- MPI Communication efficiency	0.79
--- MPI Load balance	0.32
---- MPI In-node load balance	0.38
---- MPI Inter-node load balance	0.83
-- OpenMP Parallel efficiency	0.36
--- OpenMP Scheduling efficiency	1
--- OpenMP Load balance	1
--- OpenMP Serialization efficiency	0.36
- Computation Scalability	1
-- Instructions scaling	1
-- IPC scaling	1
-- Frequency scaling	1
Useful IPC	1.43
Frequency [GHz]	2.91
Elapsed time [s]	226.79

- We check the metrics obtained with TALP for the relevant region

  Main issues:
  - MPI Load imbalance
  - OpenMP serialization

Propagate



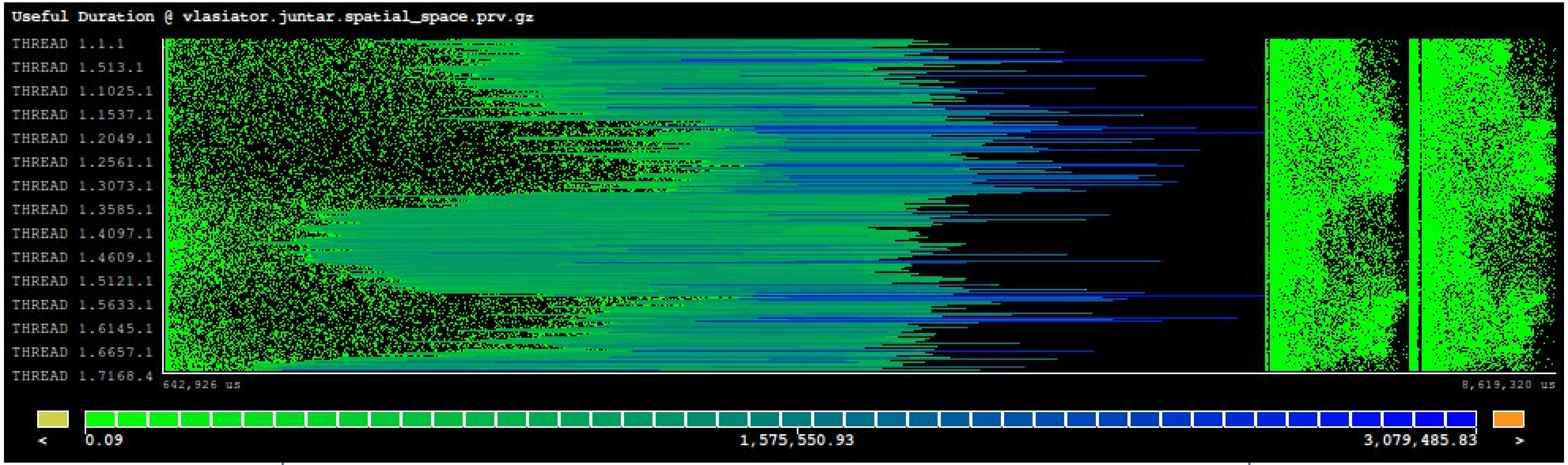
1%
2%
neSmiK: Level 4 @ vlasiator.sorted.juntar.propagate.prv.gz
THREAD 1.1.1
THREAD 1.1025.1
THREAD 1.2049.1
THREAD 1.3073.1
THREAD 1.4097.1
THREAD 1.5121.1
THREAD 1.6145.1
THREAD 1.7168.4
69%
13%
10%
4%
293,313,628 us
329,454,543 us
Update system boundaries (Vlasov pre-translation)
Velocity-space
Propagate Fields
Update system boundaries (Vlasov post-translation)
Spatial-space
Bailout-allreduce
Update system boundaries (Vlasov post-acceleration)
logfile-io
write-restart
Useful Duration @ vlasiator.sorted.juntar.propagate.prv.gz
THREAD 1.1.1
THREAD 1.513.1
THREAD 1.1025.1
THREAD 1.1537.1
THREAD 1.2049.1
THREAD 1.2561.1
THREAD 1.3073.1
THREAD 1.3585.1
THREAD 1.4097.1
THREAD 1.4609.1
THREAD 1.5121.1
THREAD 1.5633.1
THREAD 1.6145.1
THREAD 1.6657.1
EAD 1.7168.4
293,313,628 us
329,454,543 us

POP metrics Spatial-space

Metrics	Spatial-space
Global Effiency	0.11
- Parallel efficiency	0.11
-- MPI Parallel efficiency	0.15
--- MPI Communication efficiency	0.86
--- MPI Load balance	0.18
---- MPI In-node load balance	0.24
---- MPI Inter-node load balance	0.74
-- OpenMP Parallel efficiency	0.33
--- OpenMP Scheduling efficiency	1
--- OpenMP Load balance	1
--- OpenMP Serialization efficiency	0.33
- Computation Scalability	1
-- Instructions scaling	1
-- IPC scaling	1
-- Frequency scaling	1
Useful IPC	0.58
Frequency [GHz]	2.96
Elapsed time [s]	148

- Metrics for the Spatial-space region
- Main issues:
  - MPI Load imbalance
  - OpenMP serialization

Spatial-space: z-direction

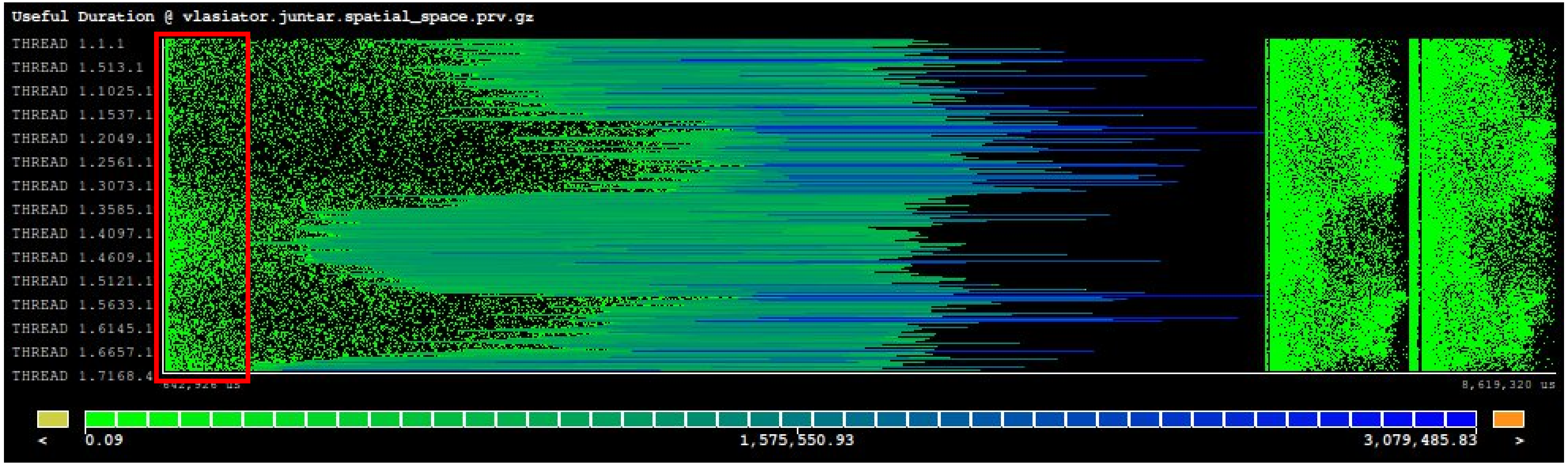


transfer-stencil-data-z

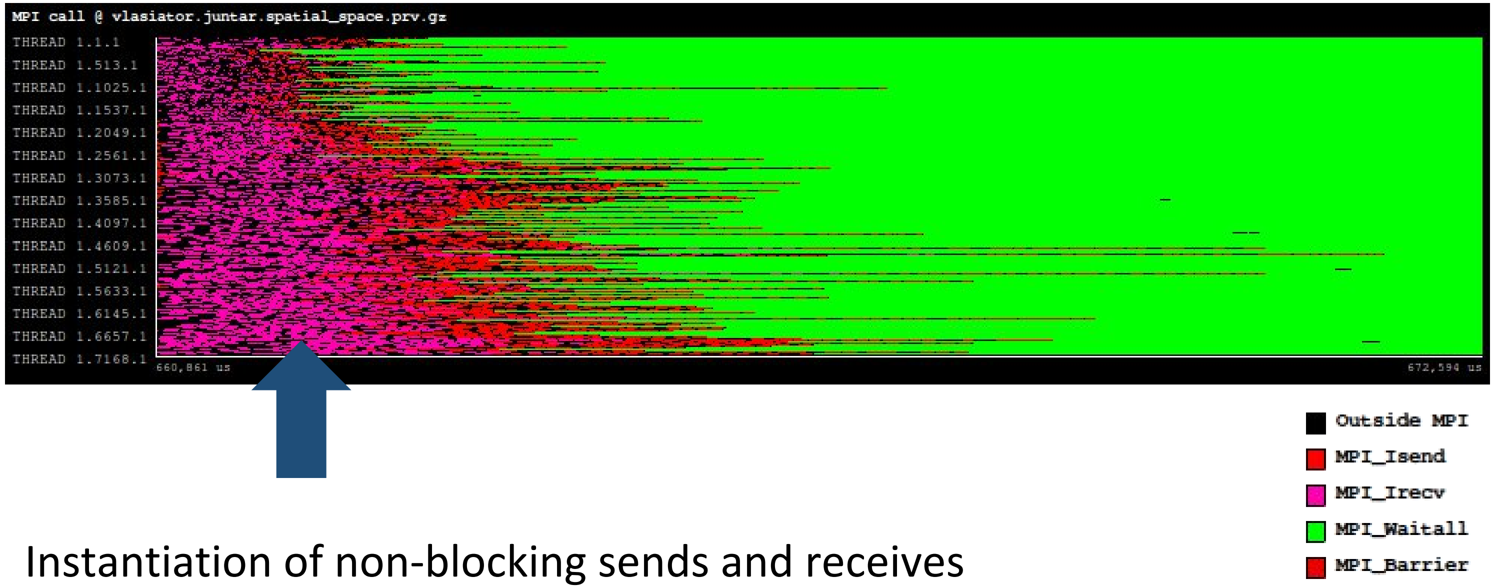
compute-mapping-z

update_remote-z

Spatial-space::transfer-stencil-data-z

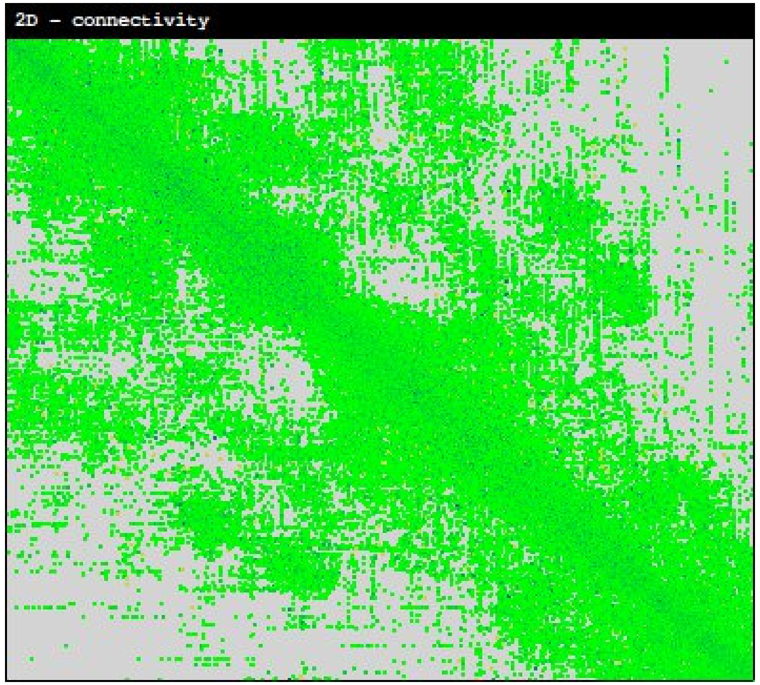


Spatial-space::transfer-stencil-data-z



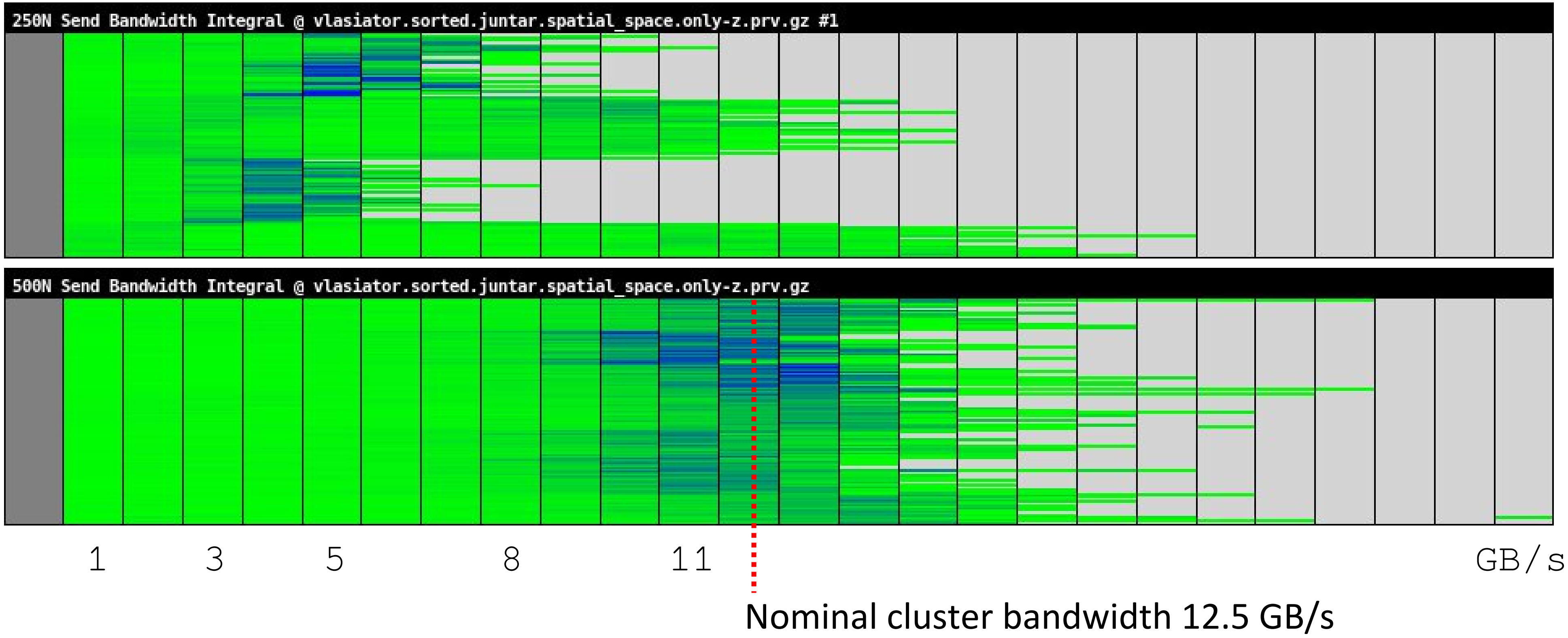
Instantiation of non-blocking sends and receives

Average Bytes transferred: 900MB
Maximum Bytes transferred: 4.3GB
Minimum Bytes transferred: 173MB
Total Bytes transferred: 6.35TB



Bytes sent(color) between the processes

Spatial-space::transfer-stencil-data-z

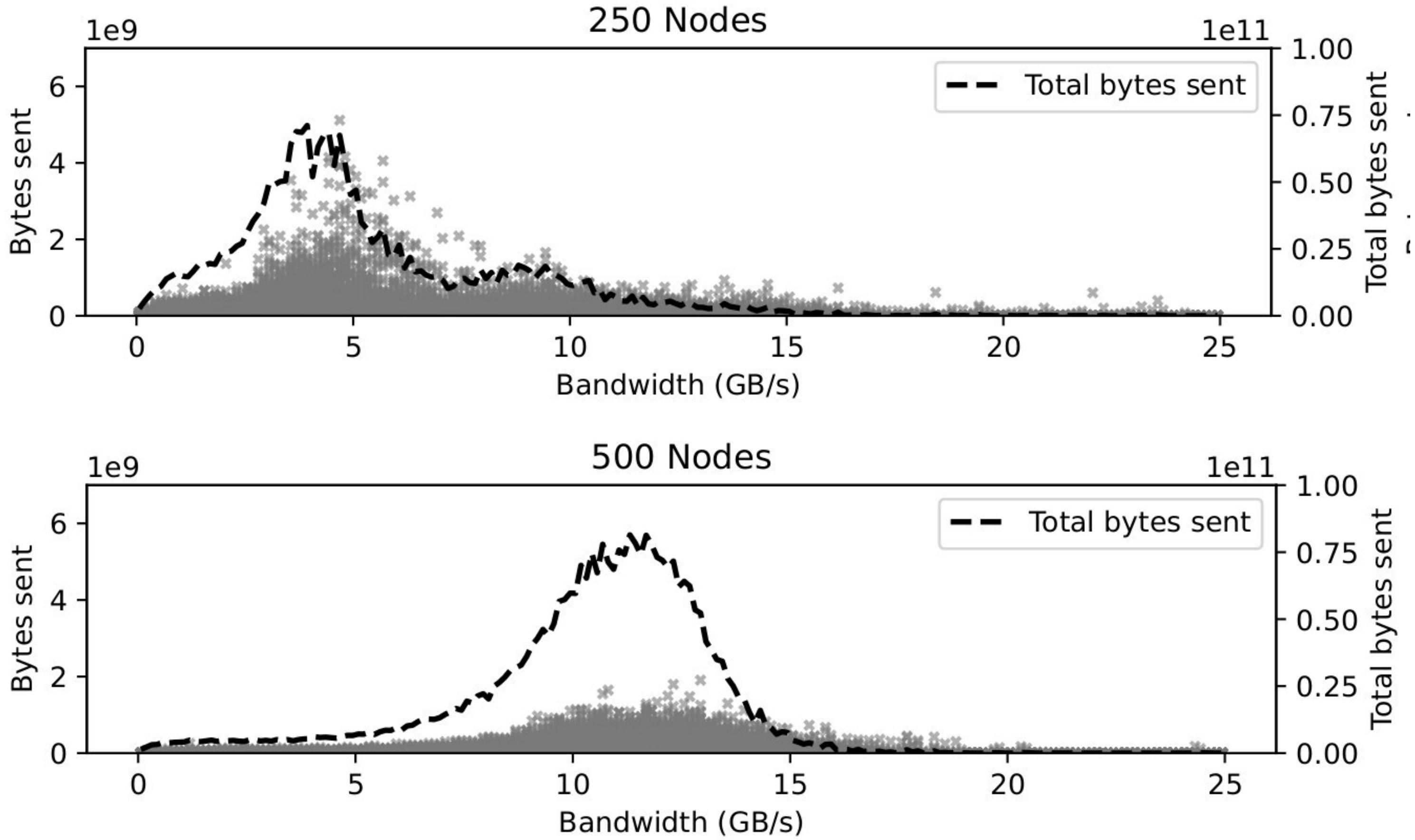


-> Under utilization of network bandwidth for 250 node execution by the MPI stack.

Send Bandwidth integral: How much data (the more blue -> more GB) is sent at which effective bandwidth.

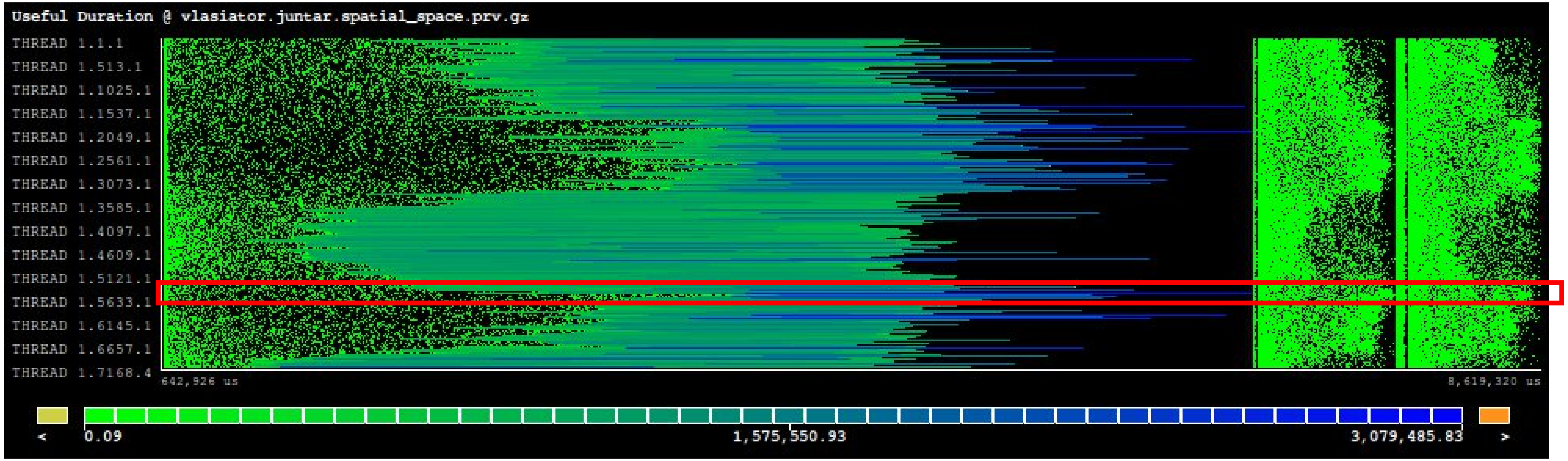
Colors scales are different for the two subplots

Spatial-space::transfer-stencil-data-z

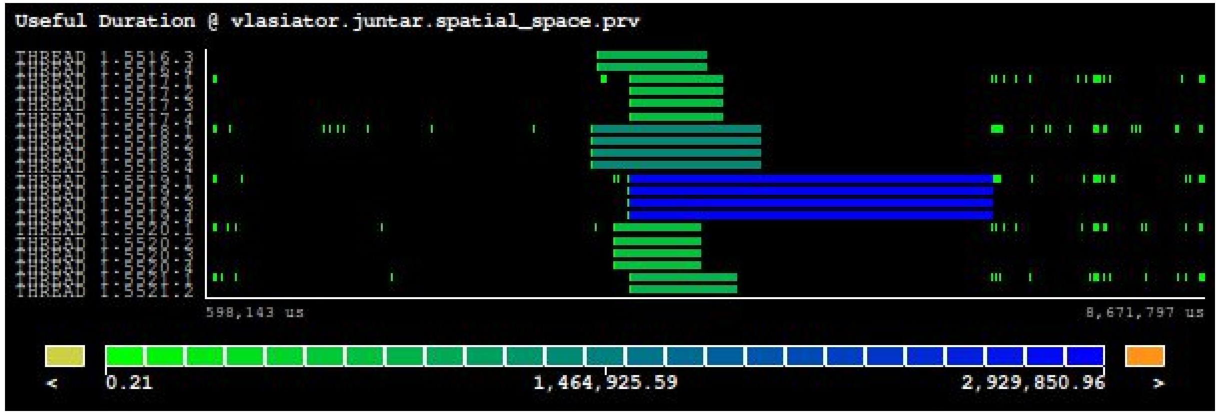


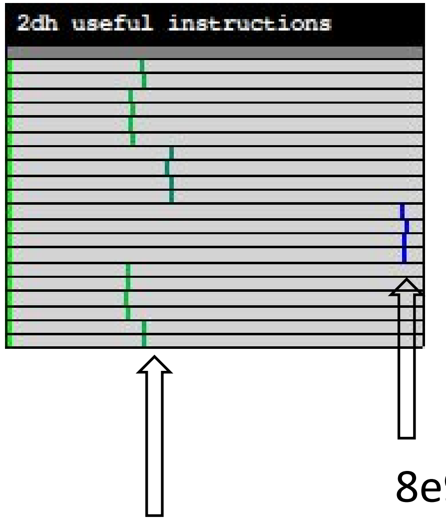
Taken from: https://doi.org/10.1016/j.procs.2025.08.237

Spatial-space



Spatial-space::compute-mapping-z



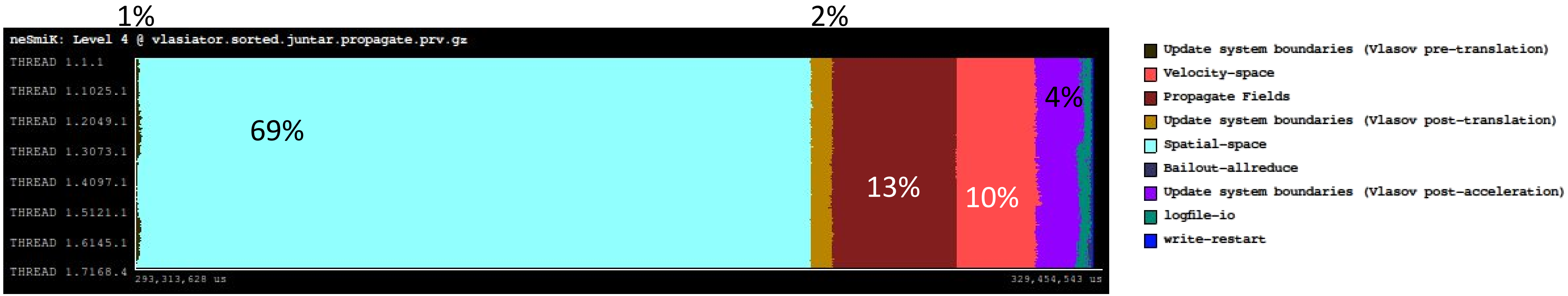


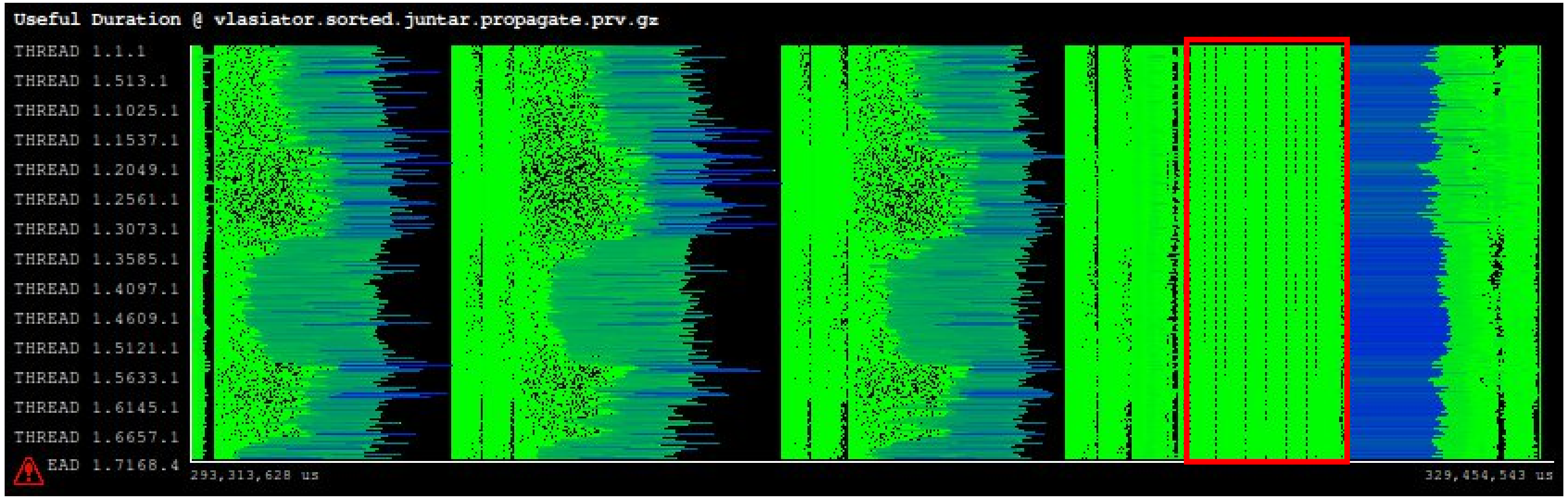
8e9 instr

2.5e9 instr

6 different processes arriving similarly after communication phase -> Then encountering load imbalance caused by instructions imbalance between the MPI processes

Propagate


1%
2%
neSmiK: Level 4 @ vlasiator.sorted.juntar.propagate.prv.gz
THREAD 1.1.1
THREAD 1.1025.1
THREAD 1.2049.1
THREAD 1.3073.1
THREAD 1.4097.1
THREAD 1.5121.1
THREAD 1.6145.1
THREAD 1.7168.4
69%
13%
10%
4%
293,313,628 us
329,454,543 us
Update system boundaries (Vlasov pre-translation)
Velocity-space
Propagate Fields
Update system boundaries (Vlasov post-translation)
Spatial-space
Bailout-allreduce
Update system boundaries (Vlasov post-acceleration)
logfile-io
write-restart

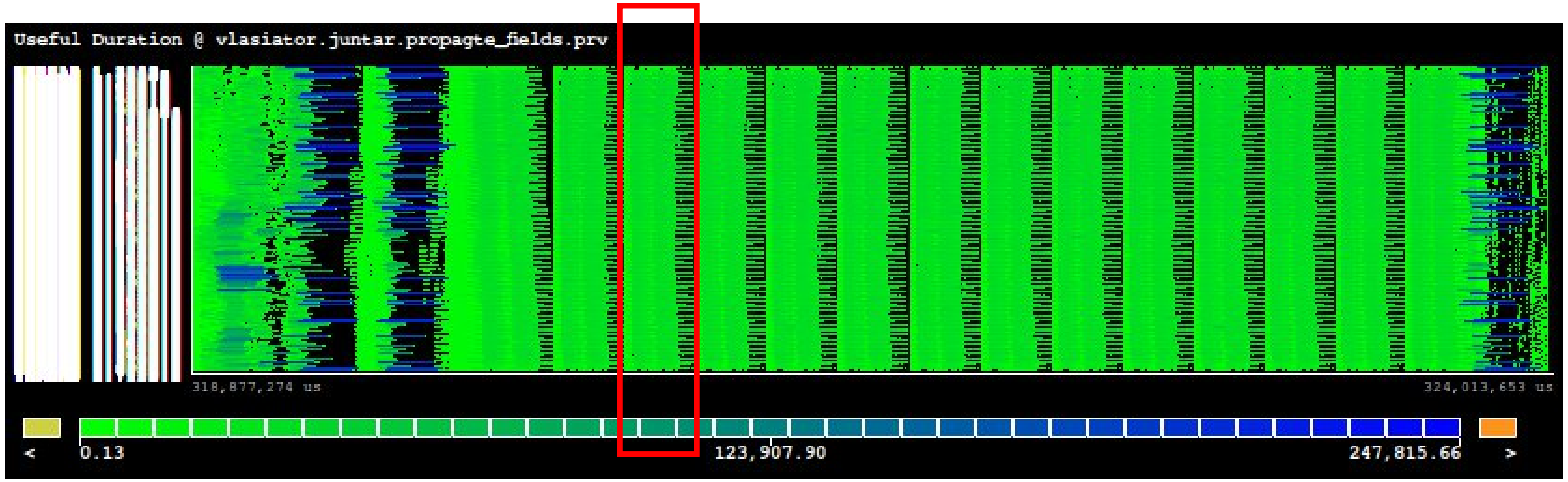

Useful Duration @ vlasiator.sorted.juntar.propagate.prv.gz
THREAD 1.1.1
THREAD 1.513.1
THREAD 1.1025.1
THREAD 1.1537.1
THREAD 1.2049.1
THREAD 1.2561.1
THREAD 1.3073.1
THREAD 1.3585.1
THREAD 1.4097.1
THREAD 1.4609.1
THREAD 1.5121.1
THREAD 1.5633.1
THREAD 1.6145.1
THREAD 1.6657.1
EAD 1.7168.4
293,313,628 us
329,454,543 us

POP metrics Propagate-fields

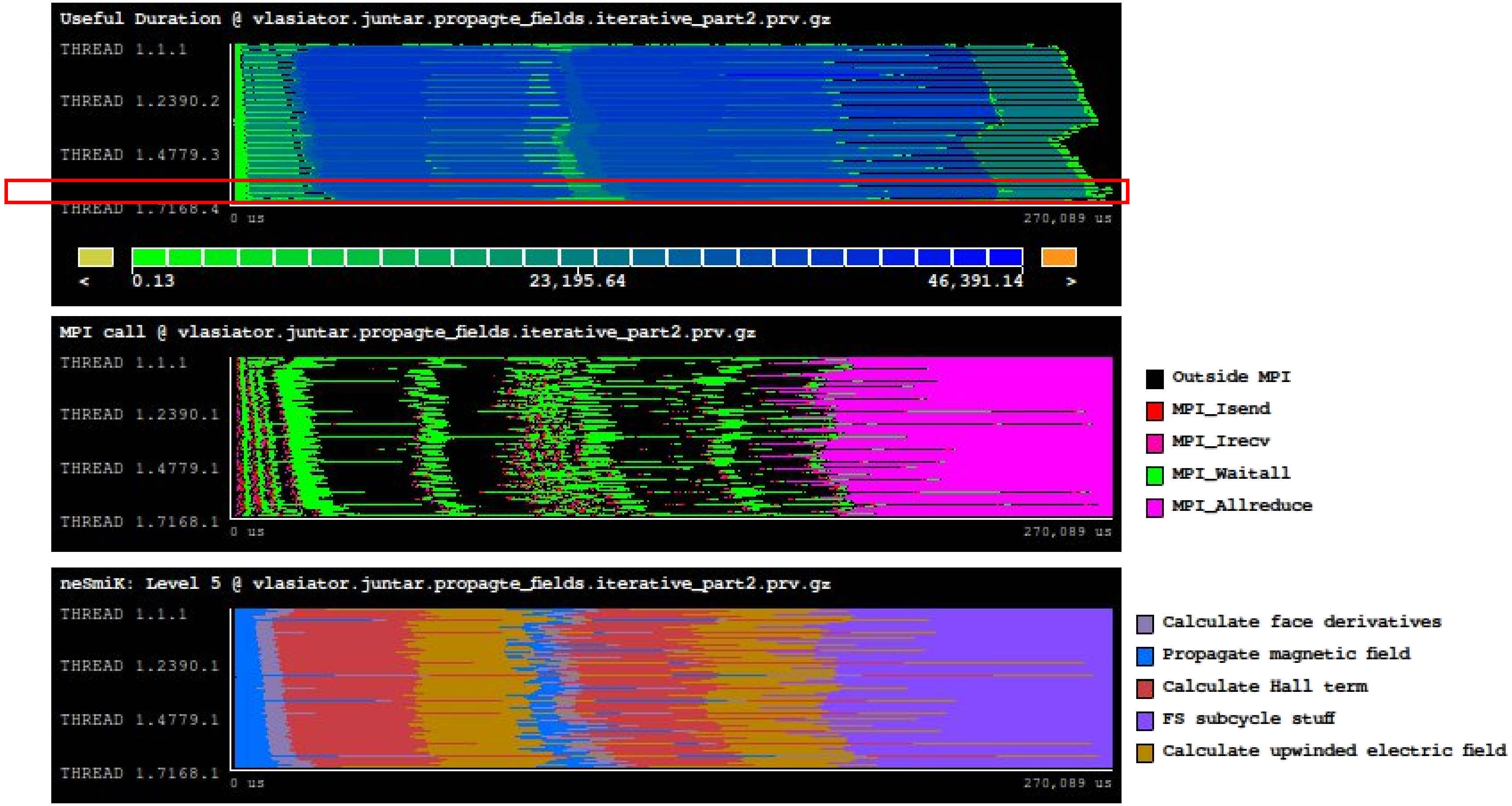
Metrics	Propagate Fields
Global Effiency	0.15
- Parallel efficiency	0.15
-- MPI Parallel efficiency	0.27
--- MPI Communication efficiency	0.85
--- MPI Load balance	0.32
---- MPI In-node load balance	0.41
---- MPI Inter-node load balance	0.77
-- OpenMP Parallel efficiency	0.36
--- OpenMP Scheduling efficiency	1
--- OpenMP Load balance	1
--- OpenMP Serialization efficiency	0.36
- Computation Scalability	1
-- Instructions scaling	1
-- IPC scaling	1
-- Frequency scaling	1
Useful IPC	2.84
Frequency [GHz]	2.93
Elapsed time [s]	56.27

- Metrics for the Propagate-fields
- Main issues:
  - MPI Load imbalance
  - OpenMP serialization

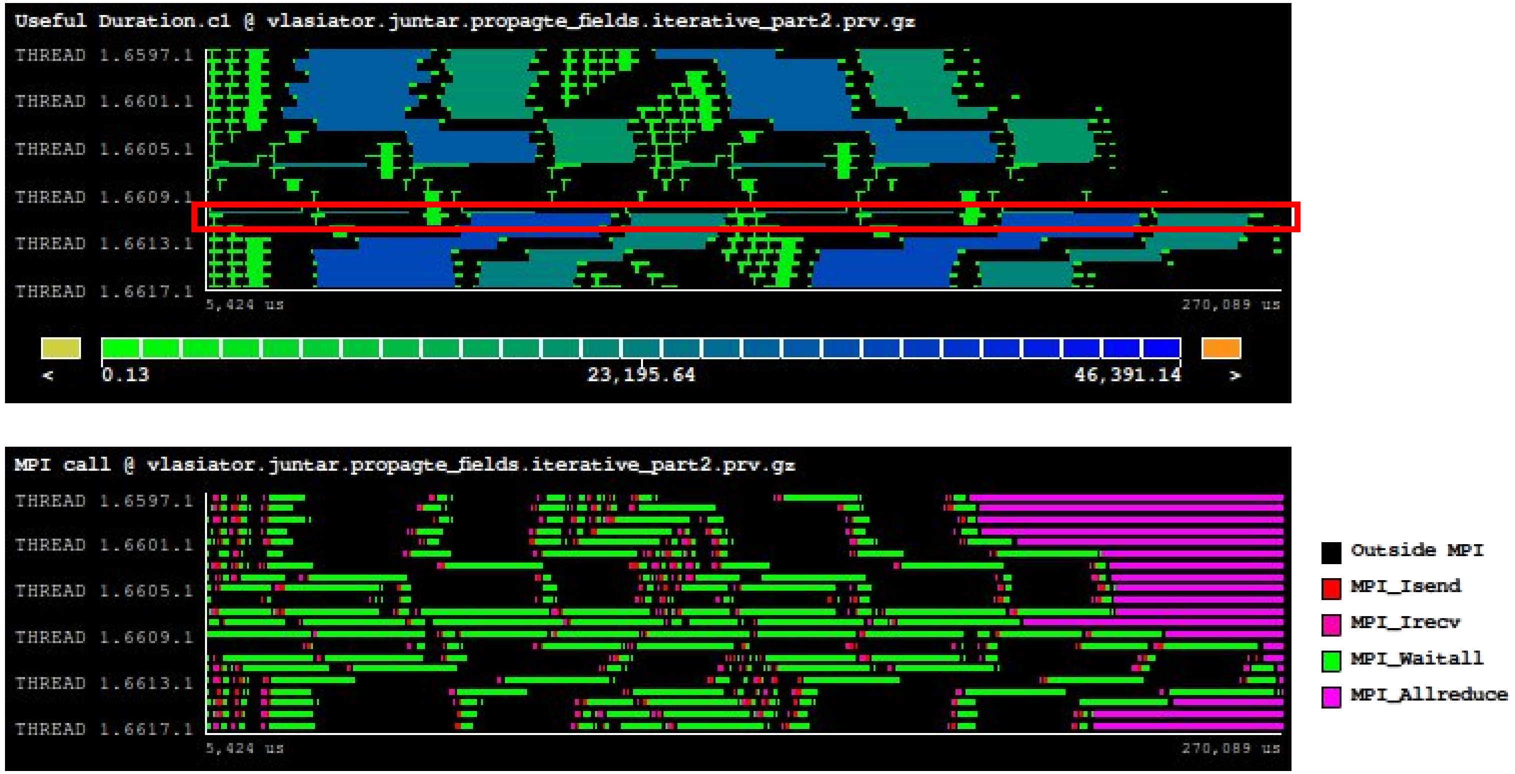
Propagate-fields



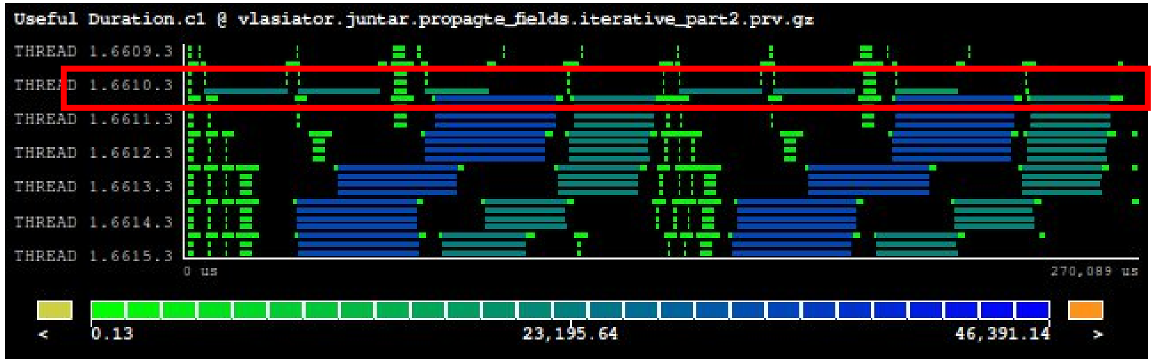
Propagate-fields

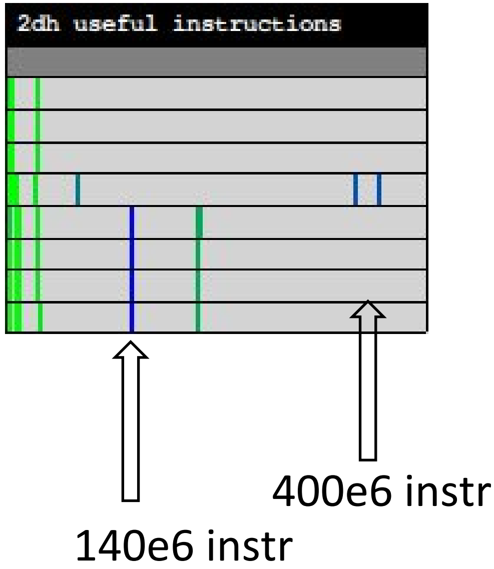


Propagate-fields



Propagate-fields





- Issues in 2 parallel regions:
  - Green/purple: MPI LB, not parallelized OpenMP
  - Blue/red: MPI and OpenMP LB

Summary

- Spatial space updates
  - Improper bandwidth utilization for 250 Node execution
  - Instruction LB problems in `compute-mapping-z`
- Field Solvers
  - LB Problems related to instructions
  - Propagate-magnetic-field: worksharing in OpenMP worsens instruction LB

Performance Optimisation and Productivity 3

A Centre of Excellence in HPC

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101143931. The JU receives support from the European Union's Horizon Europe research and innovation programme and Spain, Germany, France, Portugal and the Czech Republic.

EuroHPC
Joint Undertaking