

IPTW 2025

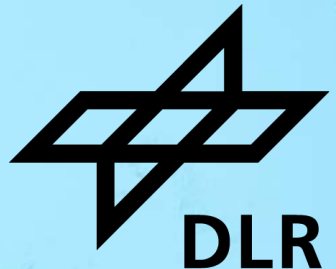
PERFORMANCE EVALUATION OF THE FLOW SOLVER TRACE – FROM FLUID FLOWS TO FLUID-STRUCTURE INTERACTION VIA THE FSTRACEINTERFACE

Martin Clemens (1), Ramandeep Jain (1), Christian Berthold (2), Immo Huisman (1), Ronny Tschüter (1)

German Aerospace Center

(1) Institute of Software Methods for Product Virtualization, Dresden

(2) Institute of Propulsion Technology, Cologne



HPC Systems operated by DLR-SP



HPC cluster CARA, Dresden



Credit: [DLR \(CC BY-NC-ND 3.0\)](#)

- 2,168 CPU-nodes with 2 AMD EPYC 7601 (2x 32 cores)
- 664 CPU-nodes with 2 AMD EPYC 7702 (2x 64 cores)
- 10 GPU-nodes with 4 Nvidia A100 and 2 AMD EPYC 7702
- 17 PB Lustre file system (0,5 PB SSD / 16,5 PB HDD)
- Operational since 2020/2023 g Replacement 2025

HPC cluster CARO, Göttingen



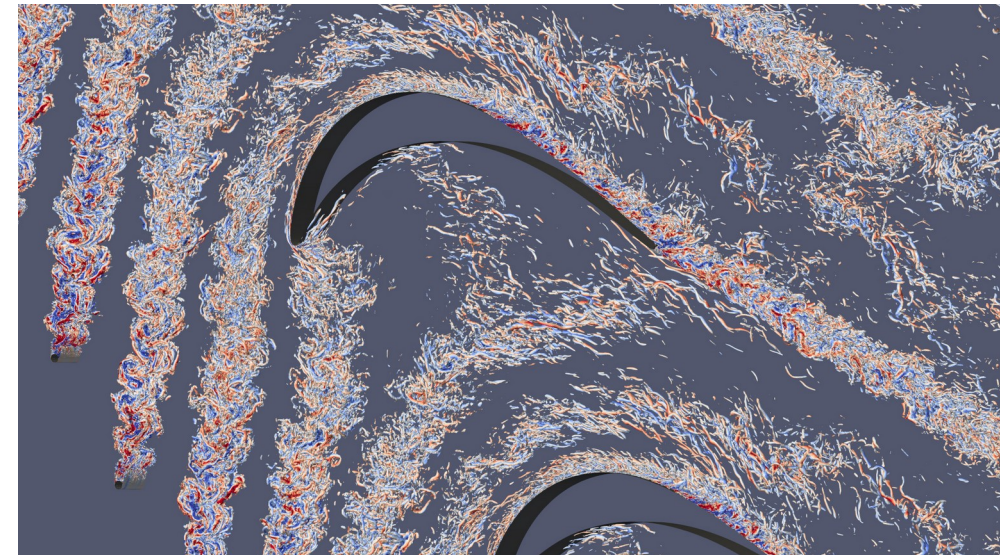
Credit: [DLR \(CC BY-NC-ND 3.0\)](#)

- 1,364 CPU-nodes with 2 AMD EPYC 7702 CPUs
- 8.4 PB Lustre file system (HDD with SSD cache)
- Operational since 2022 g Replacement 2027/28

- Instrumentation using Score-P (8.3) and Score-P python bindings
 - Instrument everything on Python side
 - Compile time filter on C++ side to obtain an overhead <10%
- Visualization with Cube (4.7.1) and Vampir (10.4.1)

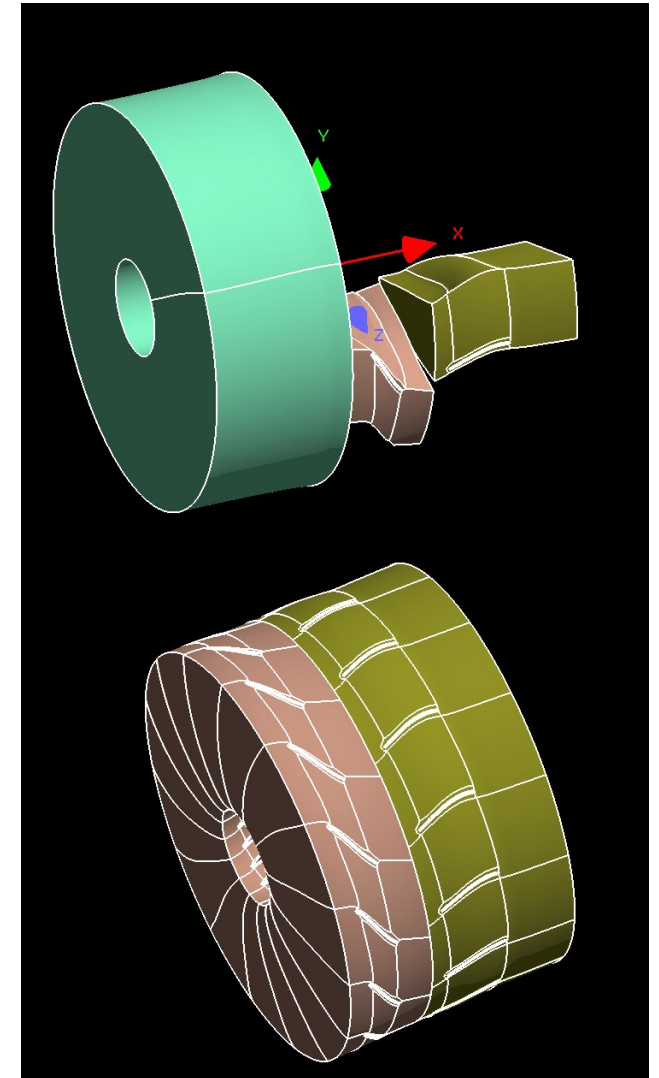


- DLR's standard CFD solver for turbomachinery flows
- Used in industrial design processes by MTU Aero Engines AG and Siemens Energy AG
- Steady and unsteady RANS solver on structured and unstructured grids
- Frequency-domain solvers for an efficient simulation of unsteady turbomachinery flows
- Originally written in C, now C++
- Hybrid parallelization using MPI and OpenMP



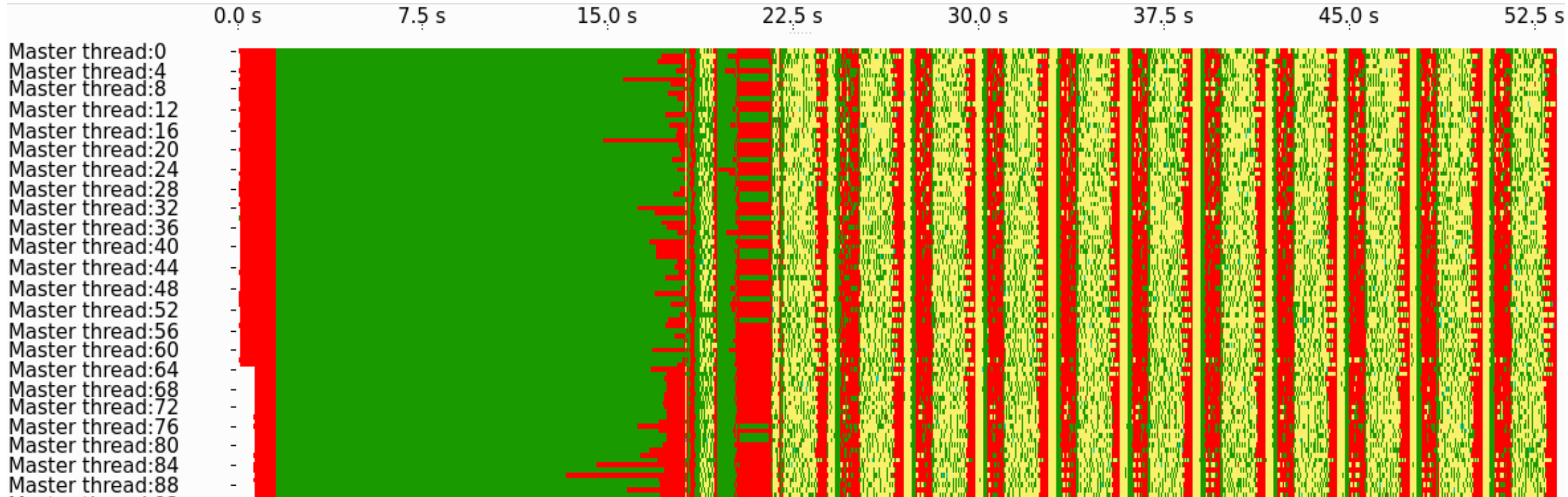
Credit: [DLR \(CC BY-NC-ND 3.0\)](https://creativecommons.org/licenses/by-nc-nd/3.0/)

- Harmonic Balance: Find periodic solution via Fourier transformation
- 1.23 million cells, 10 harmonics
 - 700.000 inflow, 300.000 rotor, 230.000 stator
- Two CARA Nodes (128 cores total)
- Pure MPI parallelization (#omp threads = 1)



Trace of TRACE

Complete Overview

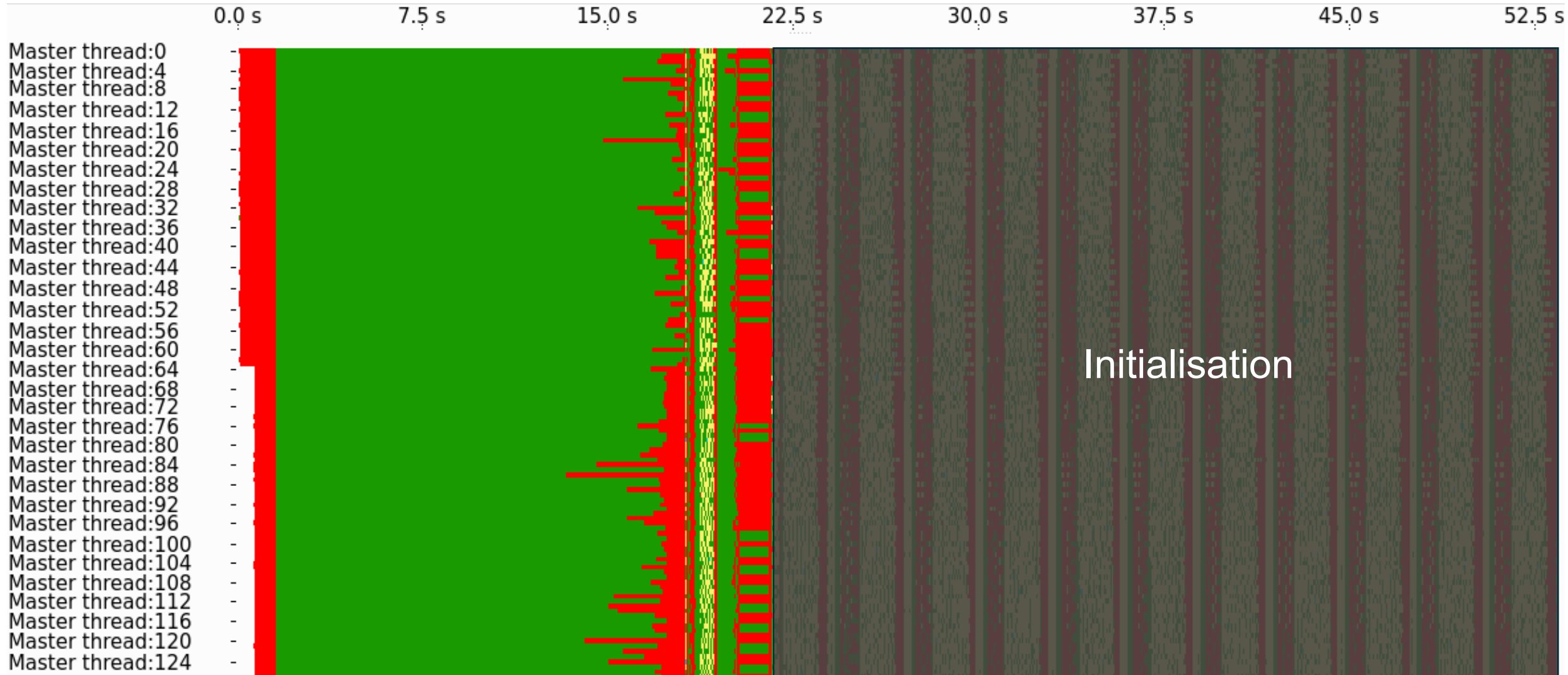


MPI, OMP, everything else

Master thread:124 -

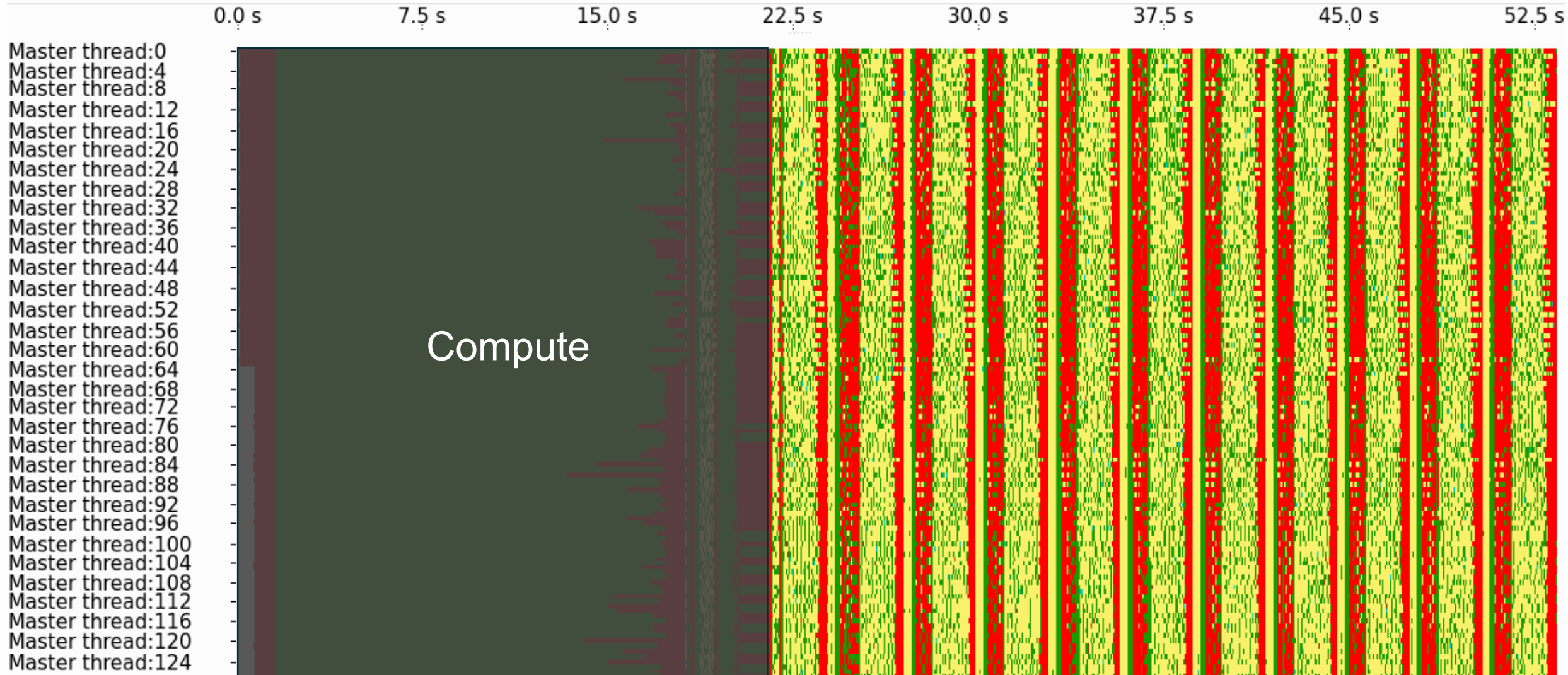
Trace of TRACE

Complete Overview



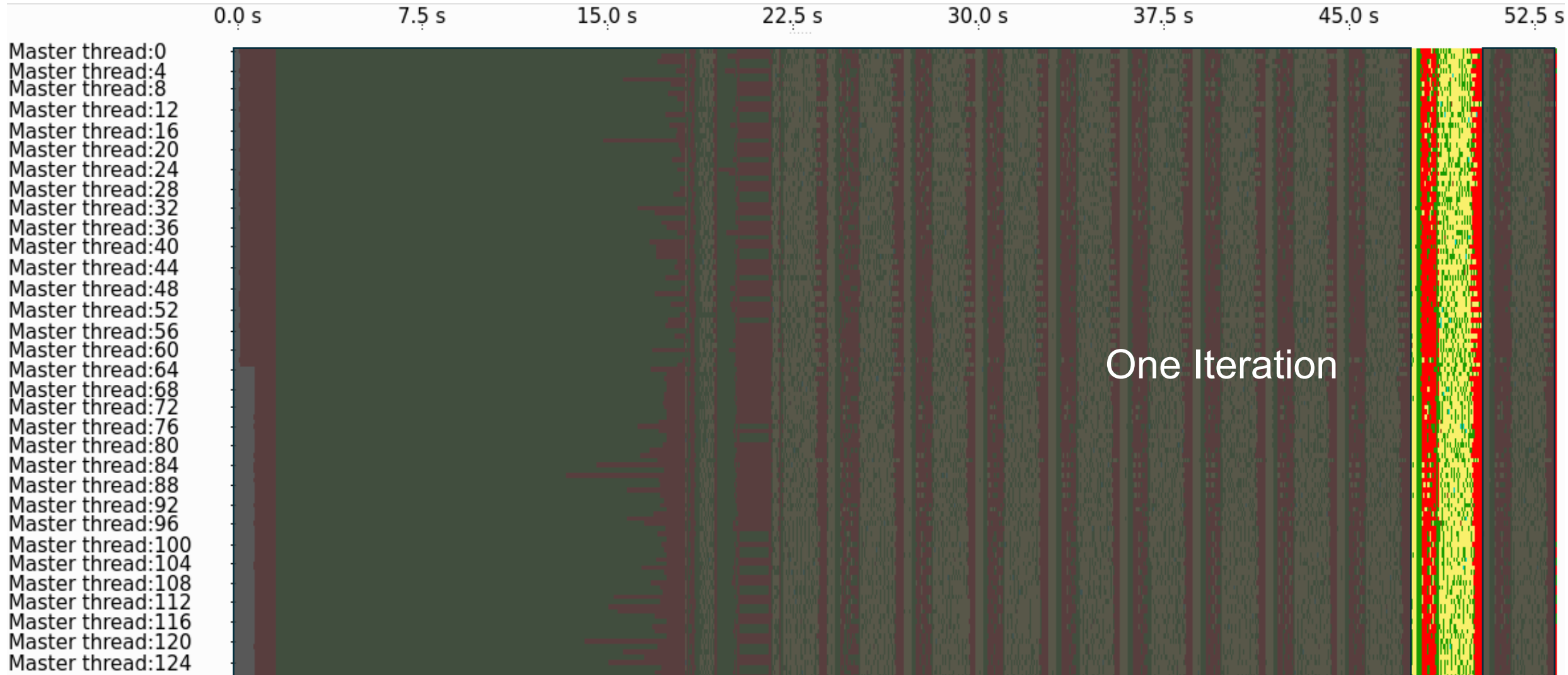
Trace of TRACE

Complete Overview



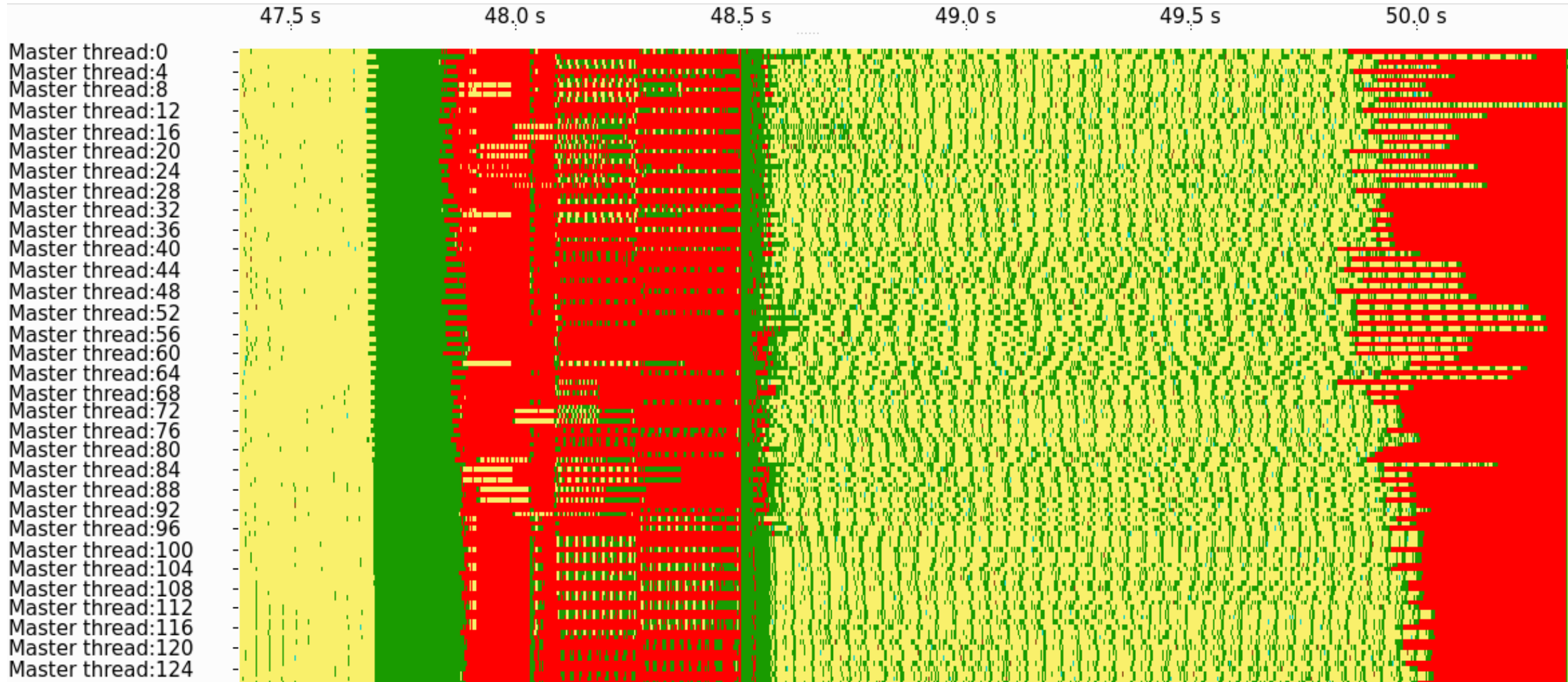
Trace of TRACE

Complete Overview



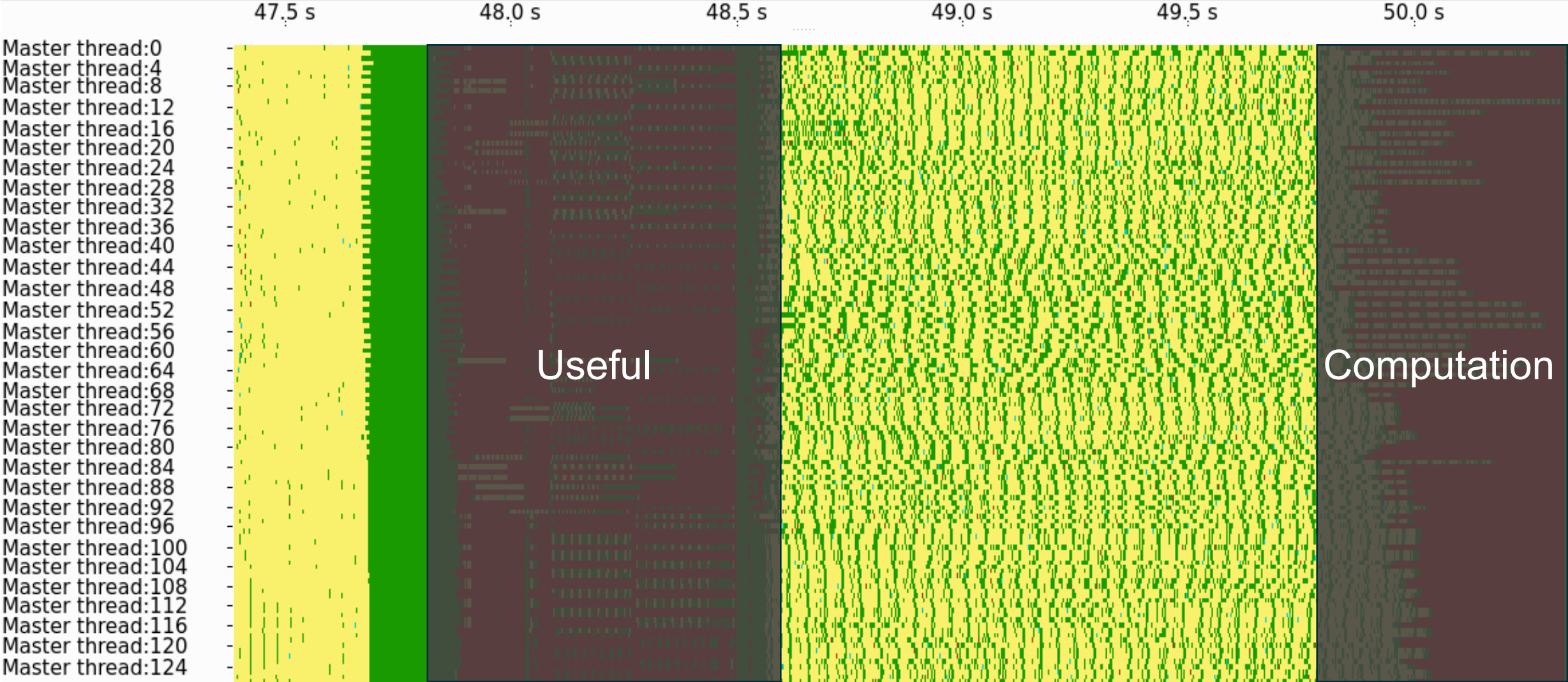
Trace of TRACE

One Iteration



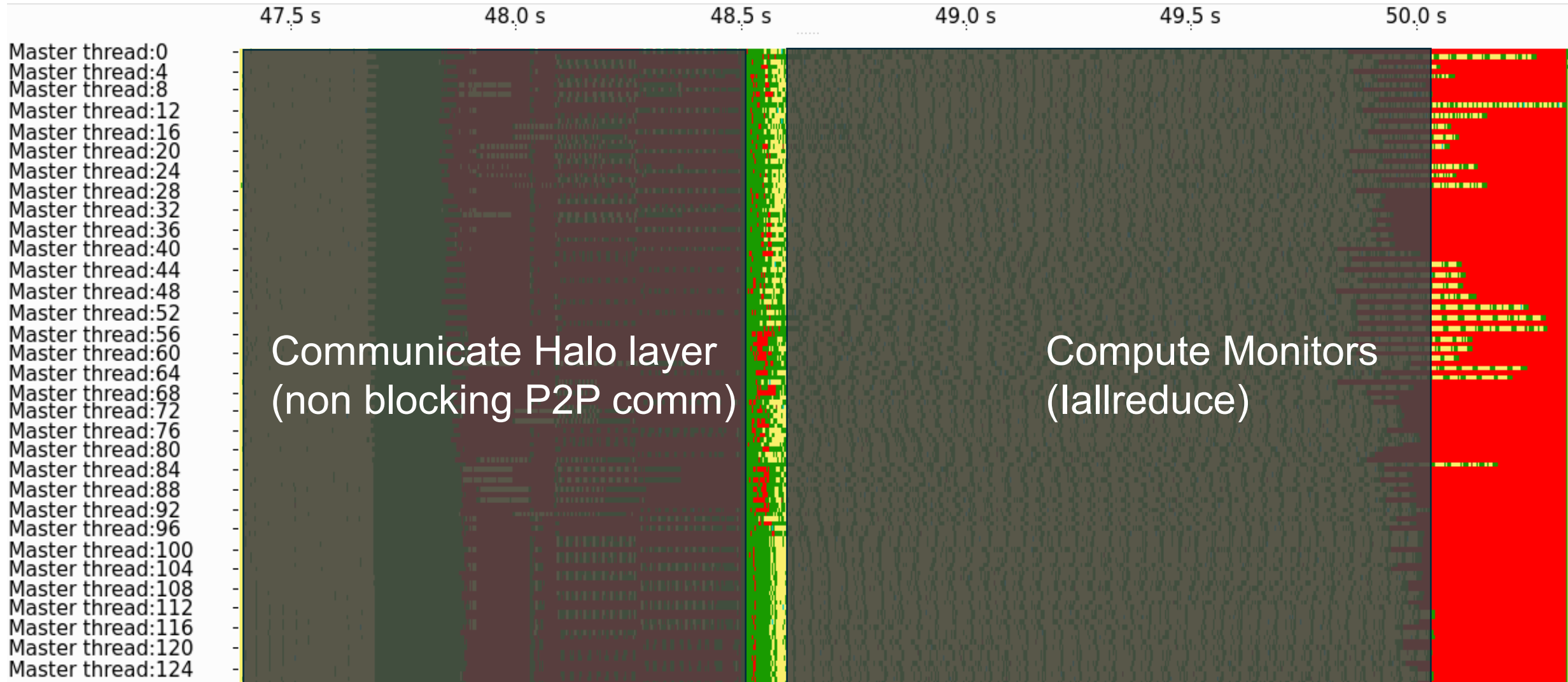
Trace of TRACE

One Iteration



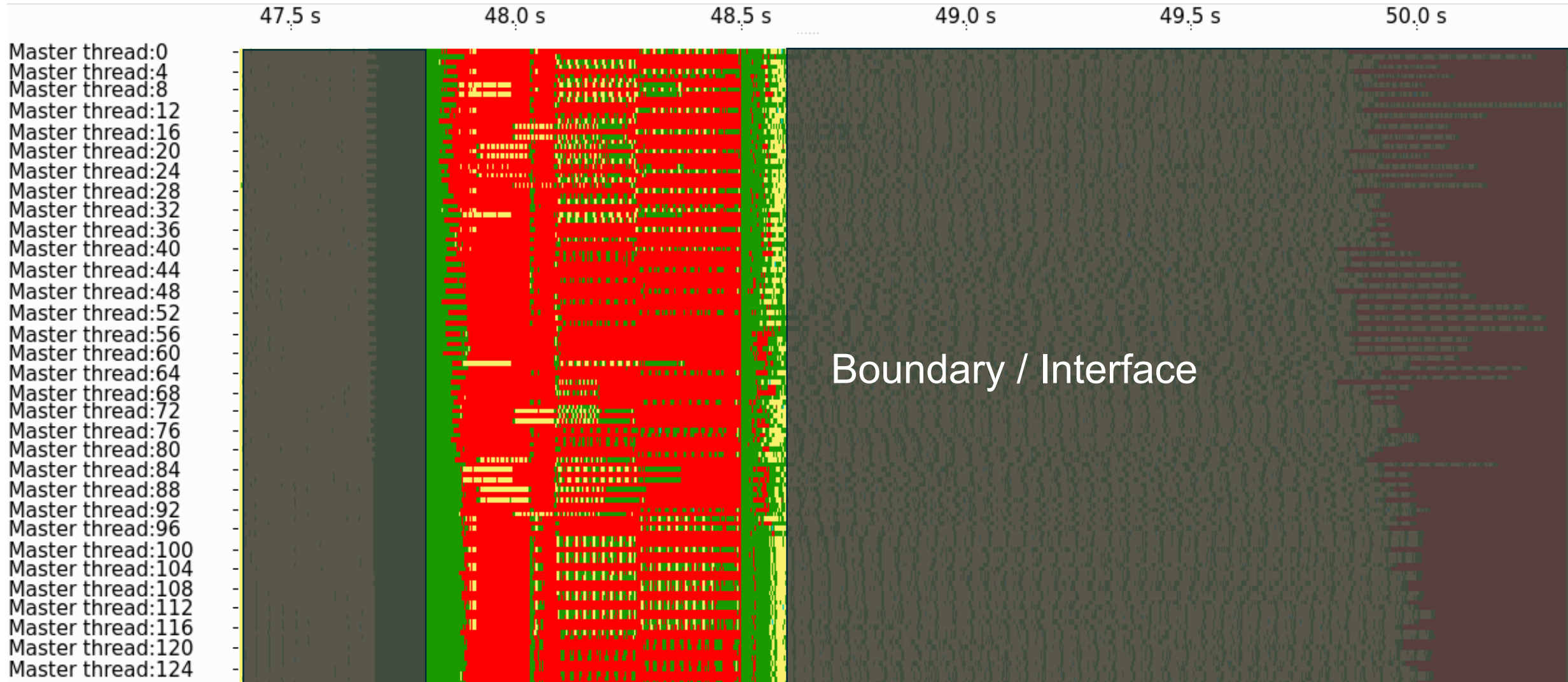
Trace of TRACE

One Iteration



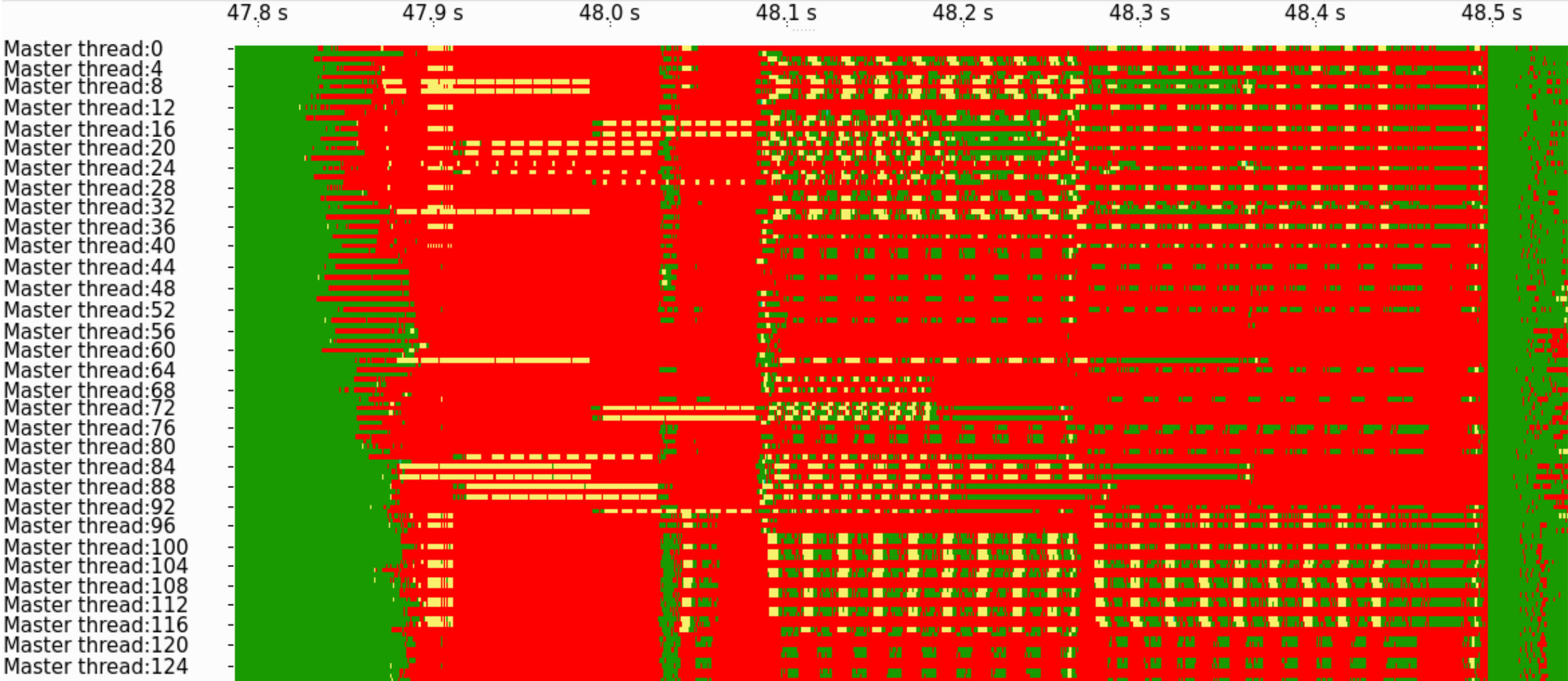
Trace of TRACE

One Iteration



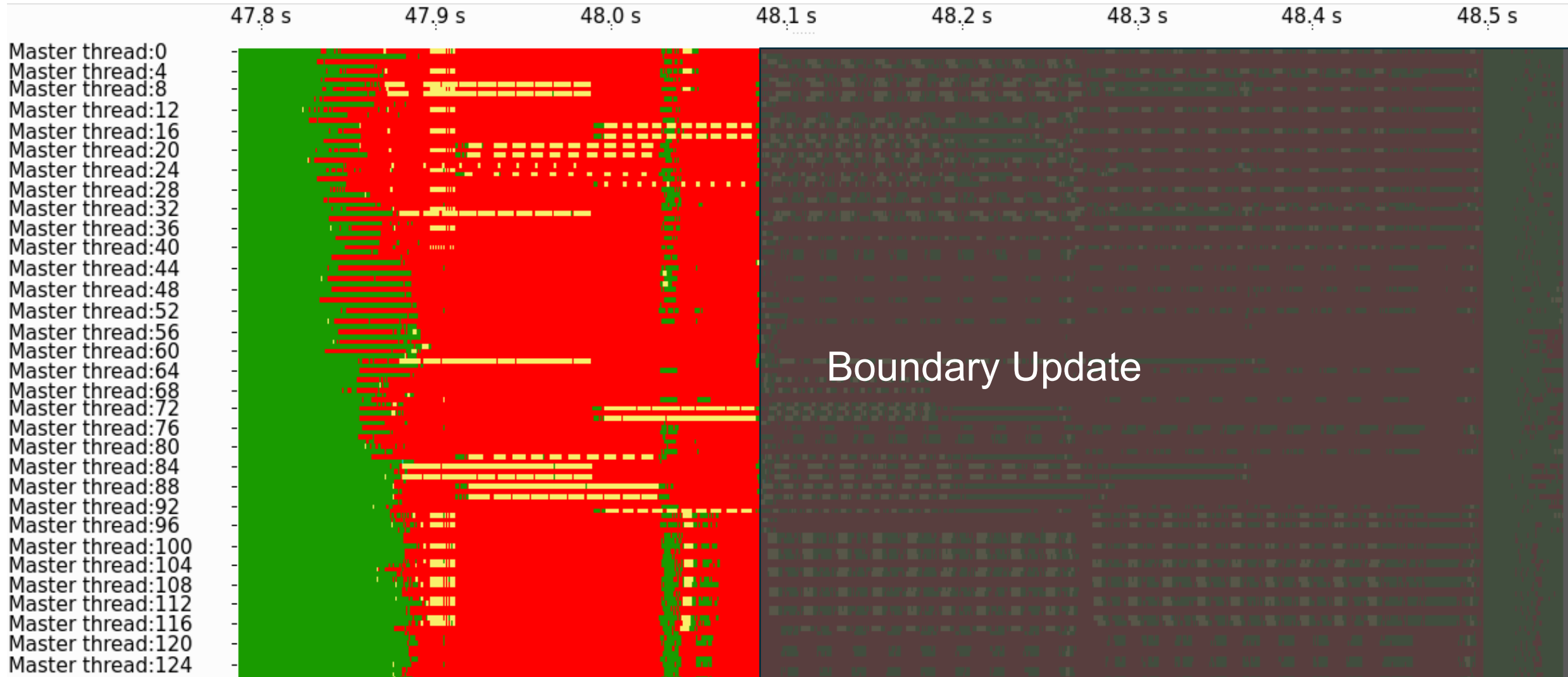
Trace of TRACE

Boundary and Interface Update



Trace of TRACE

Boundary and Interface Update



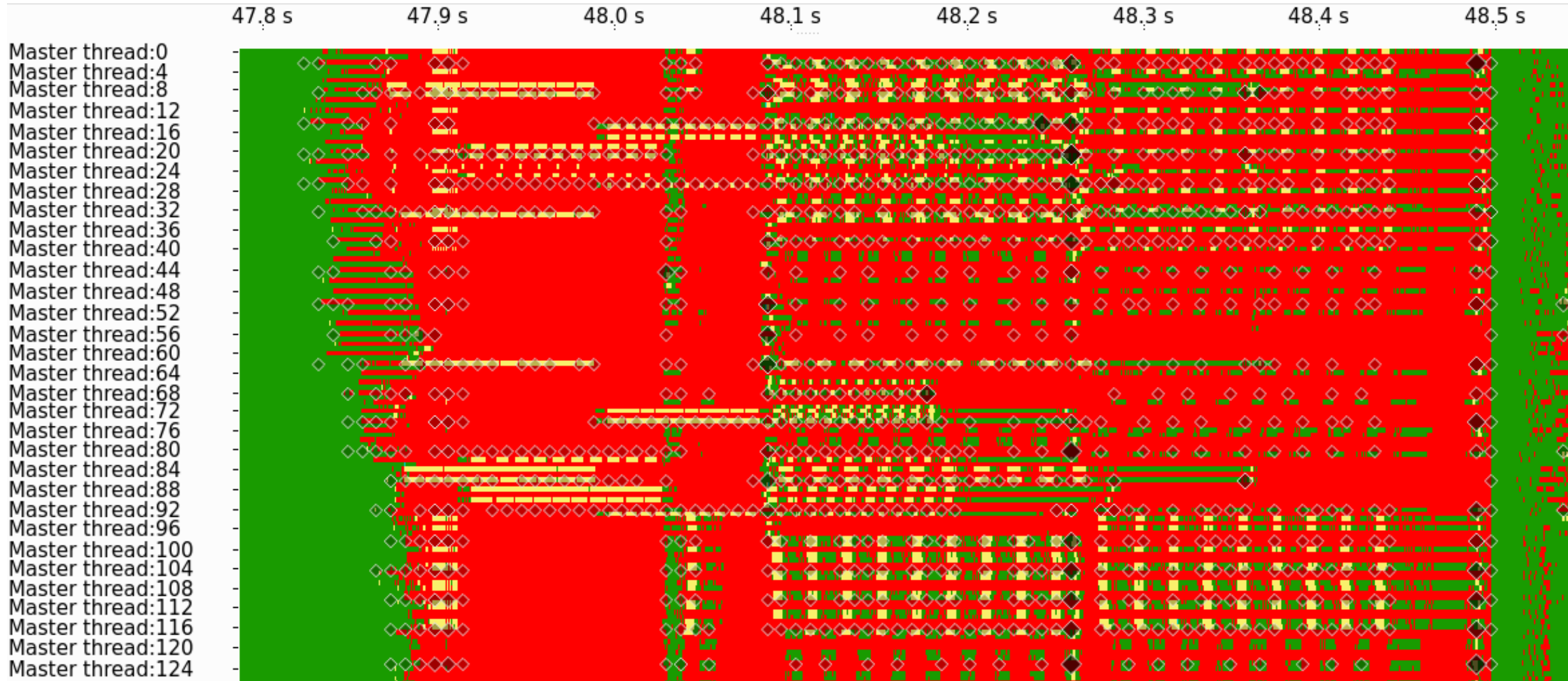
Trace of TRACE

Boundary and Interface Update



Trace of TRACE

Boundary and Interface Update, Collectives



Trace of TRACE

Boundary and Interface Update, P2P communication



Trace of TRACE

Boundary and Interface Update



- highLevelFunction

- for each object
 - foo()
 - communicate()
 - bar()

- highLevelFunction

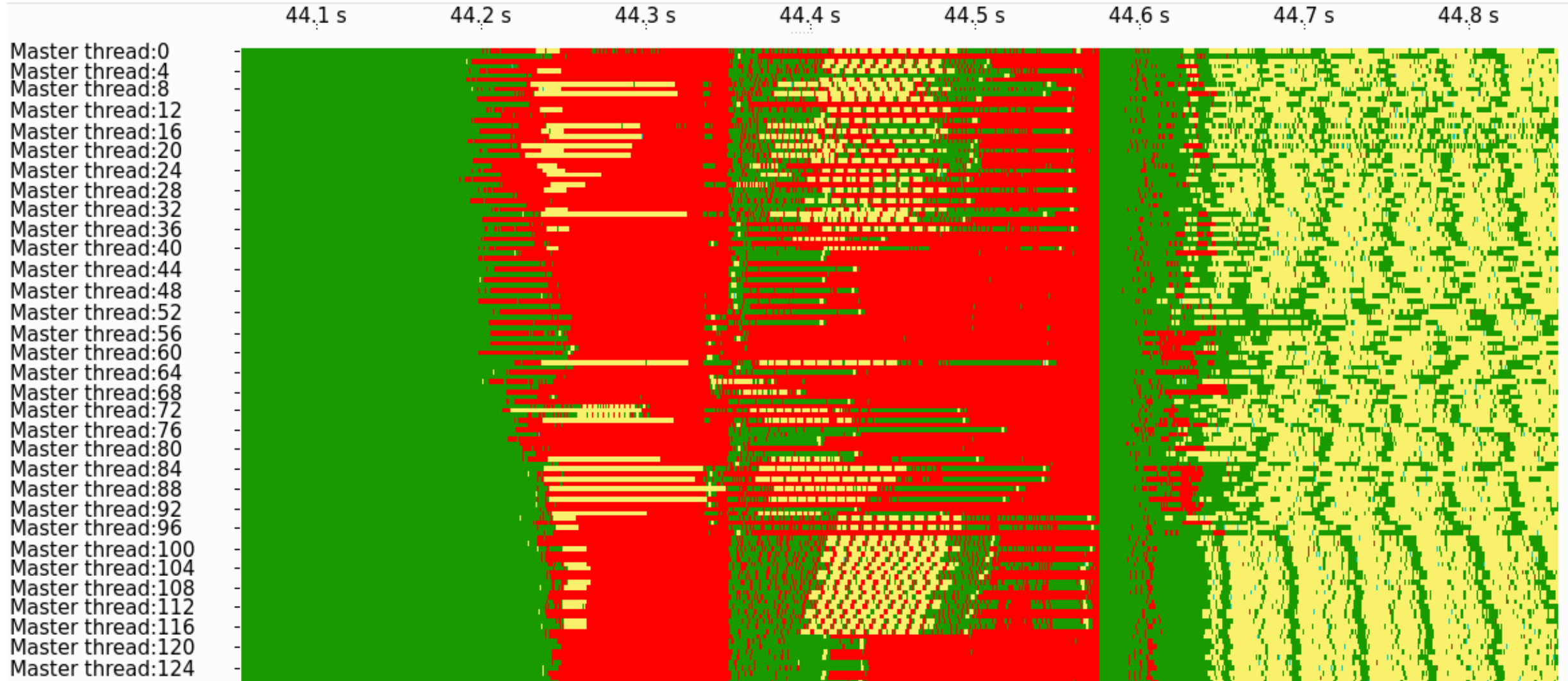
- for each object
 - foo()
 - communicateStart()
- for each object
 - communicateFinish()
 - bar()

- highLevelFunction

- commQueue queue;
- for each object
 - foo()
 - communicateEnqueue(queue, context)
- finishQueue(queue)
 - bar

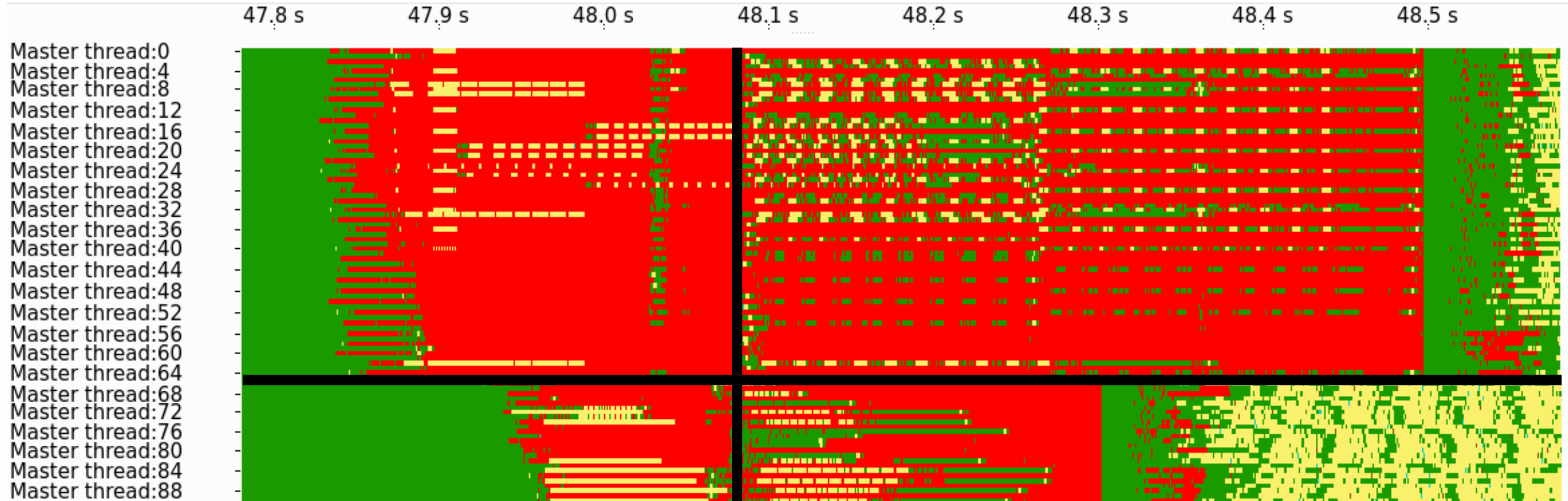
Trace of TRACE

Boundary and Interface Update



Trace of TRACE

Boundary and Interface Update



Reduction of iterate from 30.3s to 27.8s (8% speedup)
Reduction of boundary/interface block from ~0.61s to ~0.32s

Master thread:124



- Mutual interplay between fluid flow and deformable structure
 - Fluid pressure may deform structure
 - Deformed structure may lead to different fluid flows
 - e.g. flutter of airplane wing / turbine blade
- One way coupling:
 - Assume rigid structure
 - Perform complete fluid simulation
 - Compute loads on structure
- Two way coupling:
 - For every timestep iterate fluid and structure solver - until converged (strong coupling)
- once (weak coupling)

Example: An FSI Simulation

- Vortex induced vibration response of an elastic cylinder

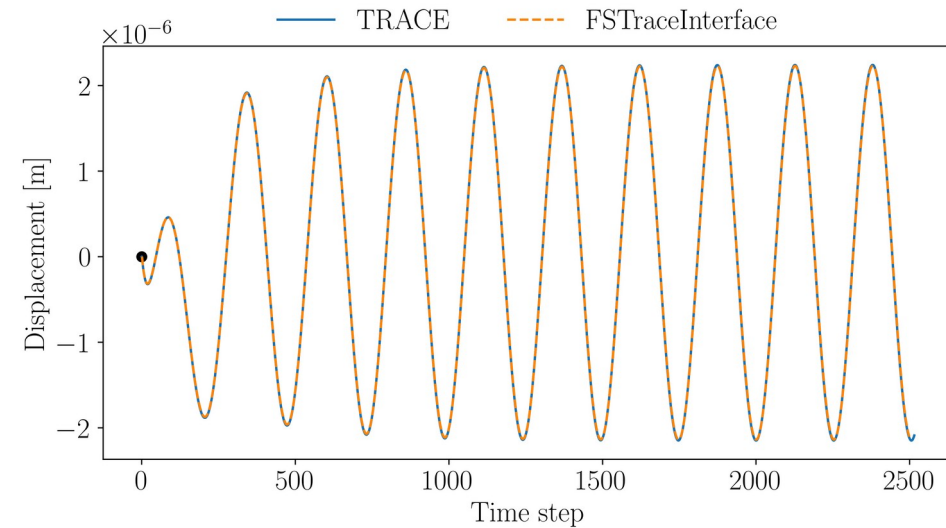
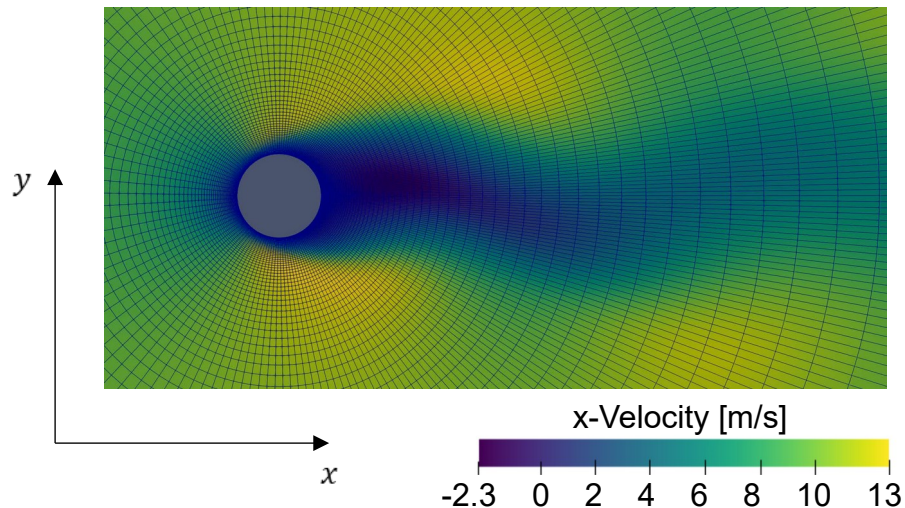
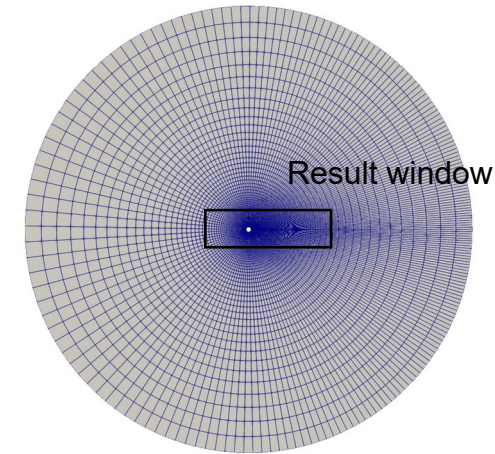
- Transient response of elastic cylinder

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = f$$

m : mass,

c : damping coefficient,

k : stiffness



Example: An FSI Simulation

- Vortex induced vibration response of an elastic cylinder

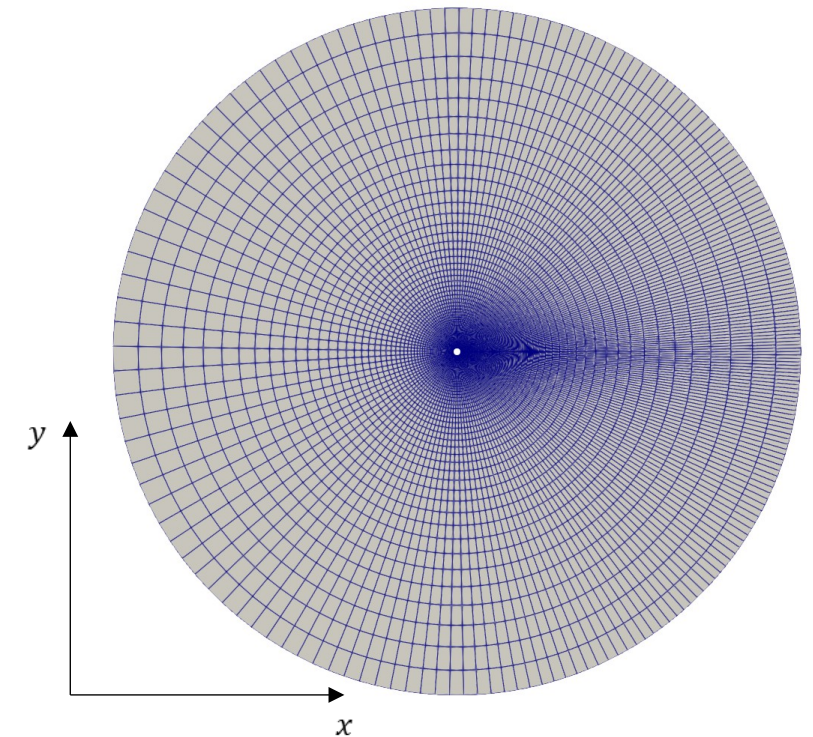
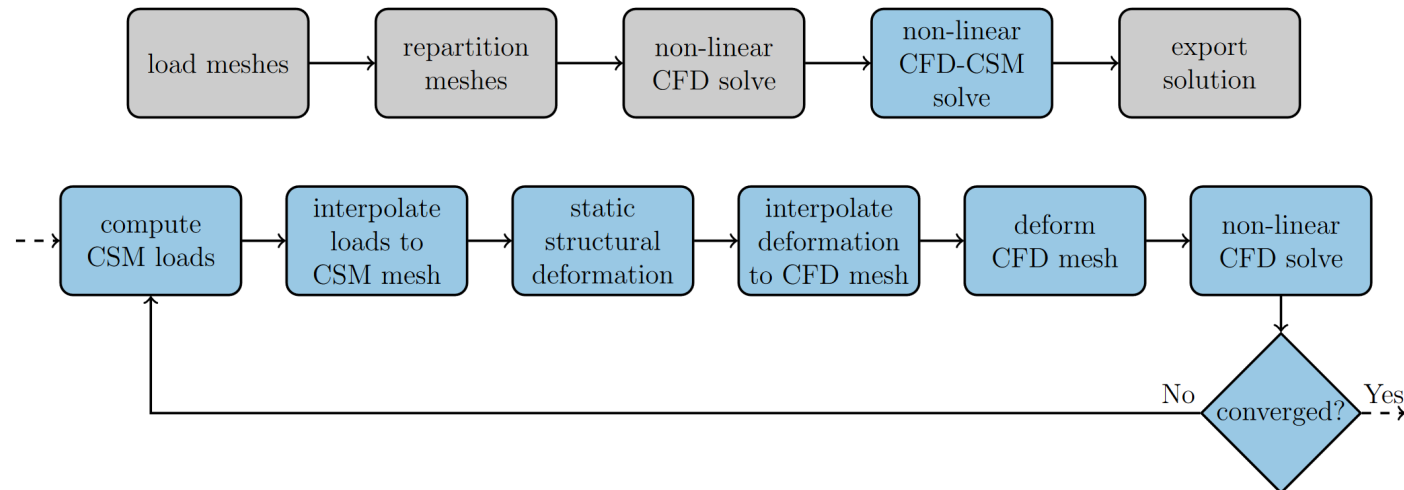
- Transient response of elastic cylinder

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = f$$

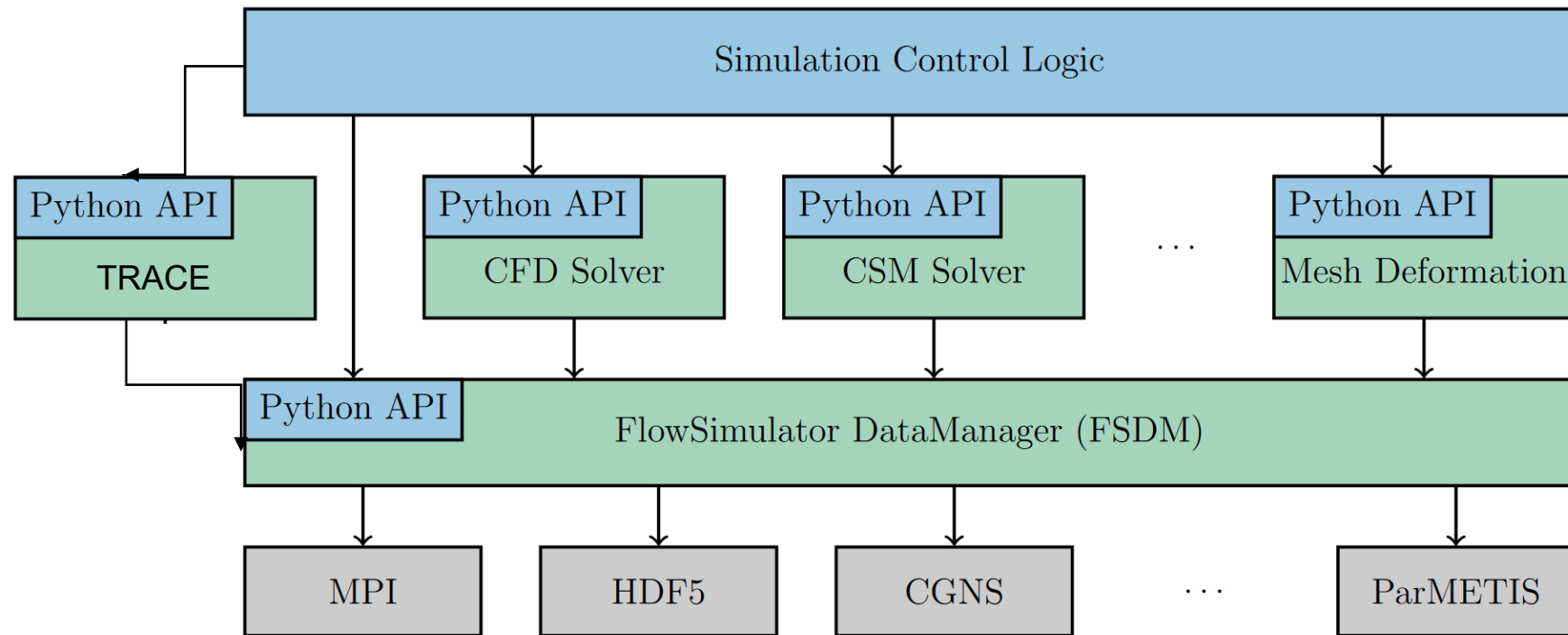
m : mass,

c : damping coefficient,

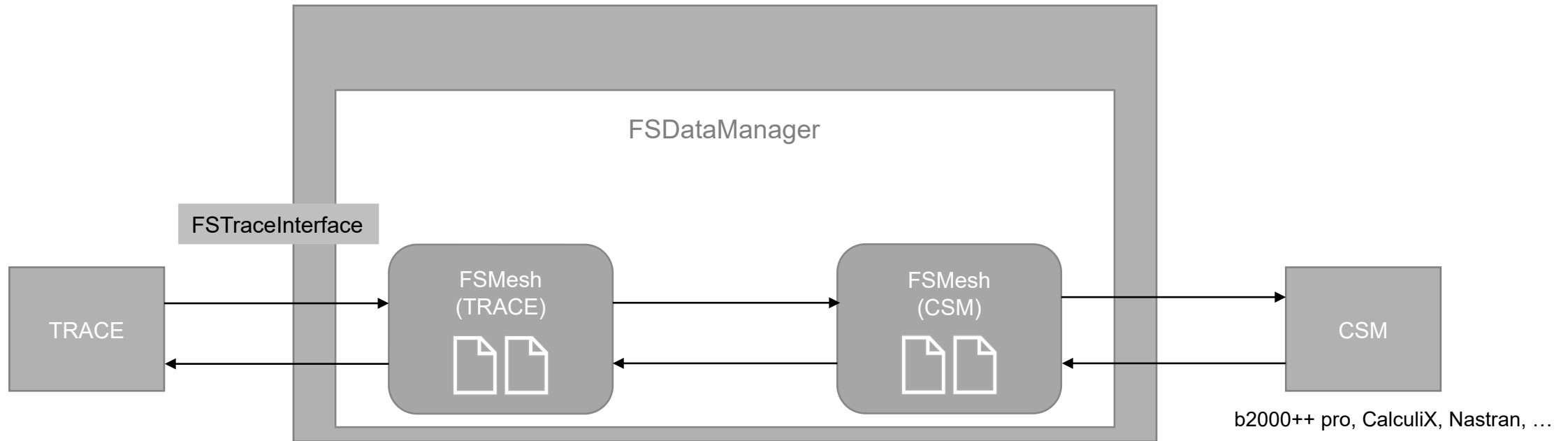
k : stiffness



- Software integration framework at DLR



How does it work?



Usage:

```
fstrace_interface = FSTraceInterface(trace, fsdm)
```

instantiate

```
for step in range(nsteps):
```

```
    trace.run()
```

run trace

```
    fs_interface.receive_dataset_from_trace()
```

receive dataset from trace

```
    CSM(fstrace_interface.cfd_coupling_mesh).solve()
```

calculate new values

```
    fs_interface.update_dataset_to_trace()
```

send dataset to trace

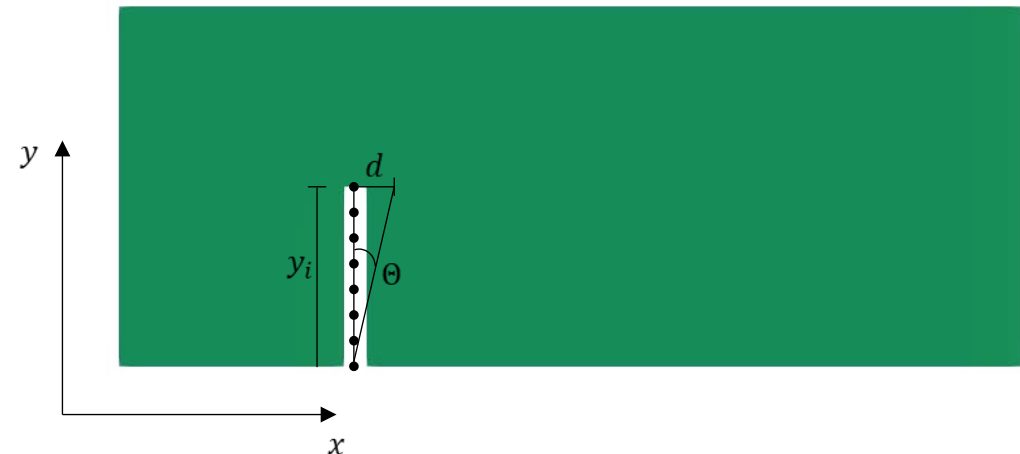
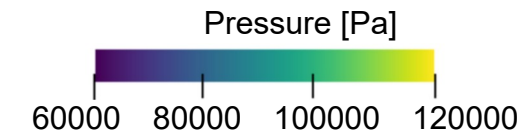
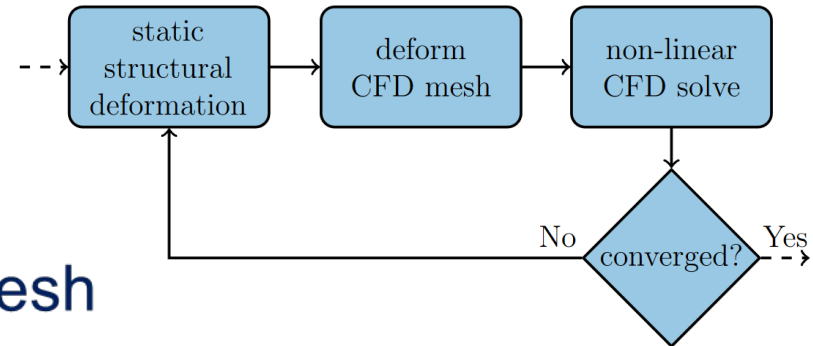
Simulation setup

- Cantilever beam in a still fluid
- 2.7M cells in 3d grid, 10 iterations
- Dummy deformations (d) given in FSMesh

$$d = y_i \sin(\theta) \quad \forall i \in N_{\text{nodes}}$$

- Mesh deformations are transferred to TRACE

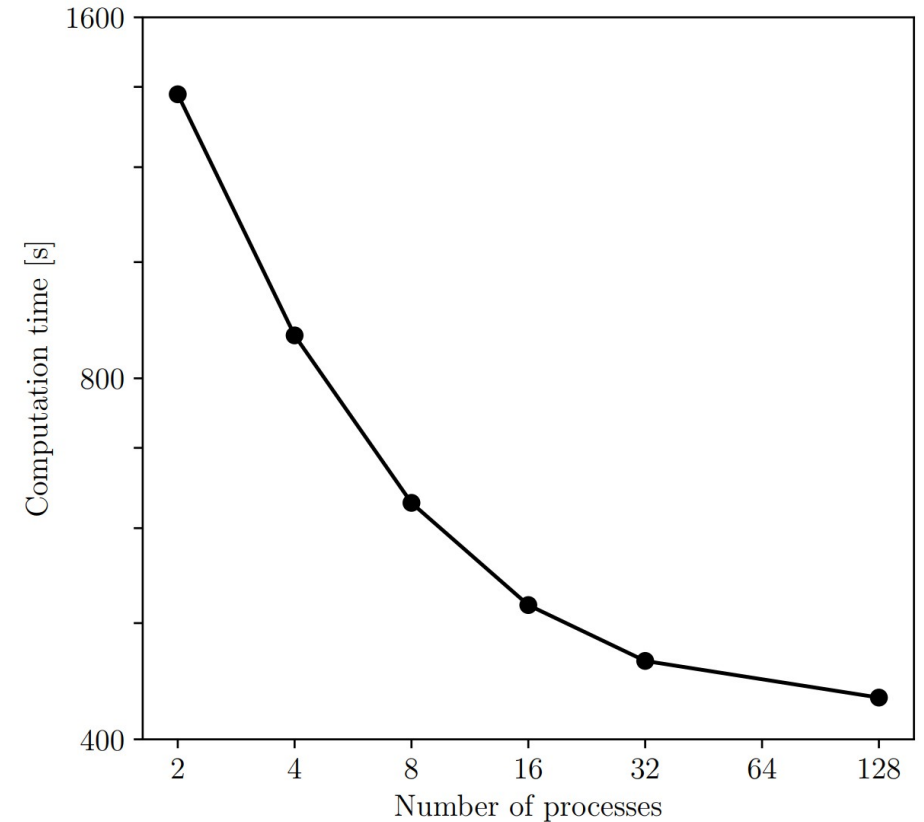
- Mesh Deformation: solve Laplace equation using GMRES



Performance analysis



- Two Goals:
 - Analysis of FSTraceInterface specifically
 - Analysis of overall FSI process chain
- Scaling run shows max speedup of less than 4



Runtime Profile for 2 Processes



time [s]	function
3839.7	tracesuite.trace.trace.TRACE:run
100.5	tracesuite.fstrace.fstrace_interface.FSTraceInterface:register_solution
90.7	tracesuite.trace.trace.TRACE:__init__
24.9	tracesuite.fstrace.fstrace_interface.FSTraceInterface:__init__
11.1	FSDDataManager.OpCallable:__call__
7.9	tracesuite.fstrace.fstrace_interface.FSTraceInterface:send_dataset_to_trace
3.8	FSDDataManager.FSMeshInterpolationGlobalNumber:Interpolate
1.0	FSDDataManager.FSUnstructDataset:ScaleToDimensional
0.6	FSDDataManager.FSMeshSelectionOpAttribute:ApplyInto
0.5	FSDDataManager.FSMeshSelection:Apply
0.4	FSDDataManager.FSDDataManager:__init__
0.4	tracesuite.fstrace.fstrace_interface.FSTraceInterface:receive_dataset_from_trace
0.3	tracesuite.fstrace.fstrace_interface.FSTraceInterface:receive_coordinates_from_trace

Runtime Profile

readable, no initialization



2 processes time [s]	128 processes time [s]	Context	Function
3839.7	59895.0	TRACE	run
11.1	519.4	FSDM	GatherMesh
7.9	14.9	FSTraceInterface	send_dataset_to_trace
3.8	25.8	FSDM	Interpolate
1.0	0.8	FSDM	ScaleToDimensional
0.6	6.8	FSDM	ApplyInto
0.5	2.1	FSDM	Apply

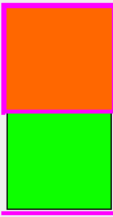
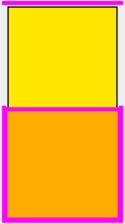
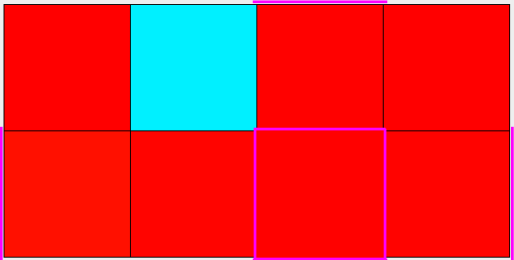
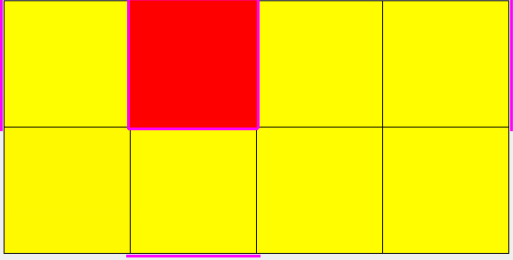
No immediate concerns for FSTraceInterface itself
Focus should be on TRACE

Runtime profile TRACE



2 processes Time[s]	128 processes Time[s]	function
2641.2	58498.9	solidMechanicsSolver
245.8	300.6	computeLHSPredictor
244.4	63.4	communicateBlockToBlock
227.2	291.9	solverControlSolvePredictorTask
129.6	159.8	computeRHSPredictor
128.2	151.2	computeRHSCorrector
79.8	113.6	updateMeshGeometry
55.4	111.6	solverControlSolveCorrectorTask



No. of Processes	Load Balance	doCommunicationBlockToBlock		solveForDeformation	
		Max time[s] Min time[s]	Peer percent CUBE	Max time[s] Min time[s]	Peer percent CUBE
2	0.8	546 139		1180 773	
8	0.4	440 40		578 176	

Focus on implementing mesh import/export from TRACE to FSDM
to enable the use of optimized solid mechanics solvers!

Acknowledgement



The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARA is partially funded by "Saxon State Ministry for Economic Affairs, Labour and Transport" and "Federal Ministry for Economic Affairs and Climate Action". The HPC system CARO is partially funded by "Ministry of Science and Culture of Lower Saxony" and "Federal Ministry for Economic Affairs and Climate Action".

Questions, Ideas, Feedback?

→ martin.clemens@dlr.de