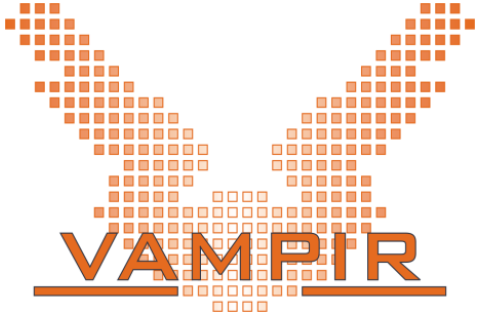
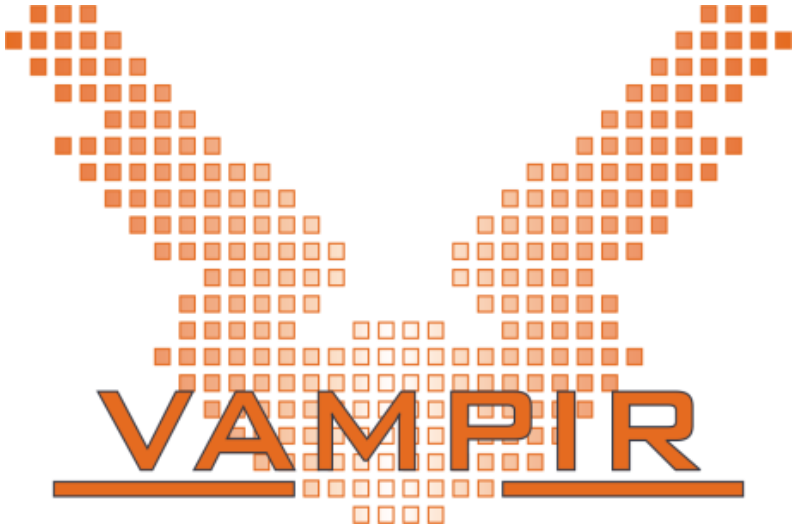
Performance Analysis with Vampir

Frank Winkler
Technische Universität Dresden



Outline

- **Part I: Welcome to the Vampir Tool Suite**
  - Event Trace Visualization
  - The Vampir Displays
  - Vampir & VampirServer
- **Part II: Vampir Hands-On**
  - Visualizing and Analyzing NPB-MZ-MPI / BT
- **Part III: Vampir Case Study**
  - Optimizing COSMO-SPECS

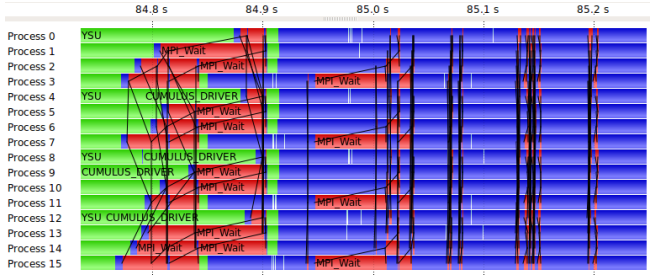


Event Trace Visualization with Vampir

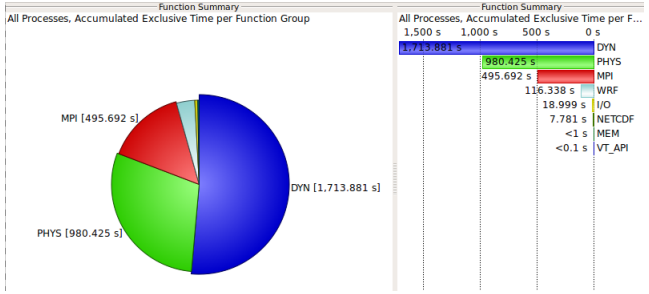
- Visualization of dynamic runtime behaviour at any level of detail along with statistics and performance metrics
- Alternative and supplement to automatic analysis

- **Typical questions that Vampir helps to answer**
  - What happens in my application execution during a given time in a given process or thread?
  - How do the communication patterns of my application execute on a real system?
  - Are there any imbalances in computation, I/O or memory usage and how do they affect the parallel execution of my application?

- **Timeline charts**
  - Application activities and communication along a time axis



- **Summary charts**
  - Quantitative results for the currently selected time interval

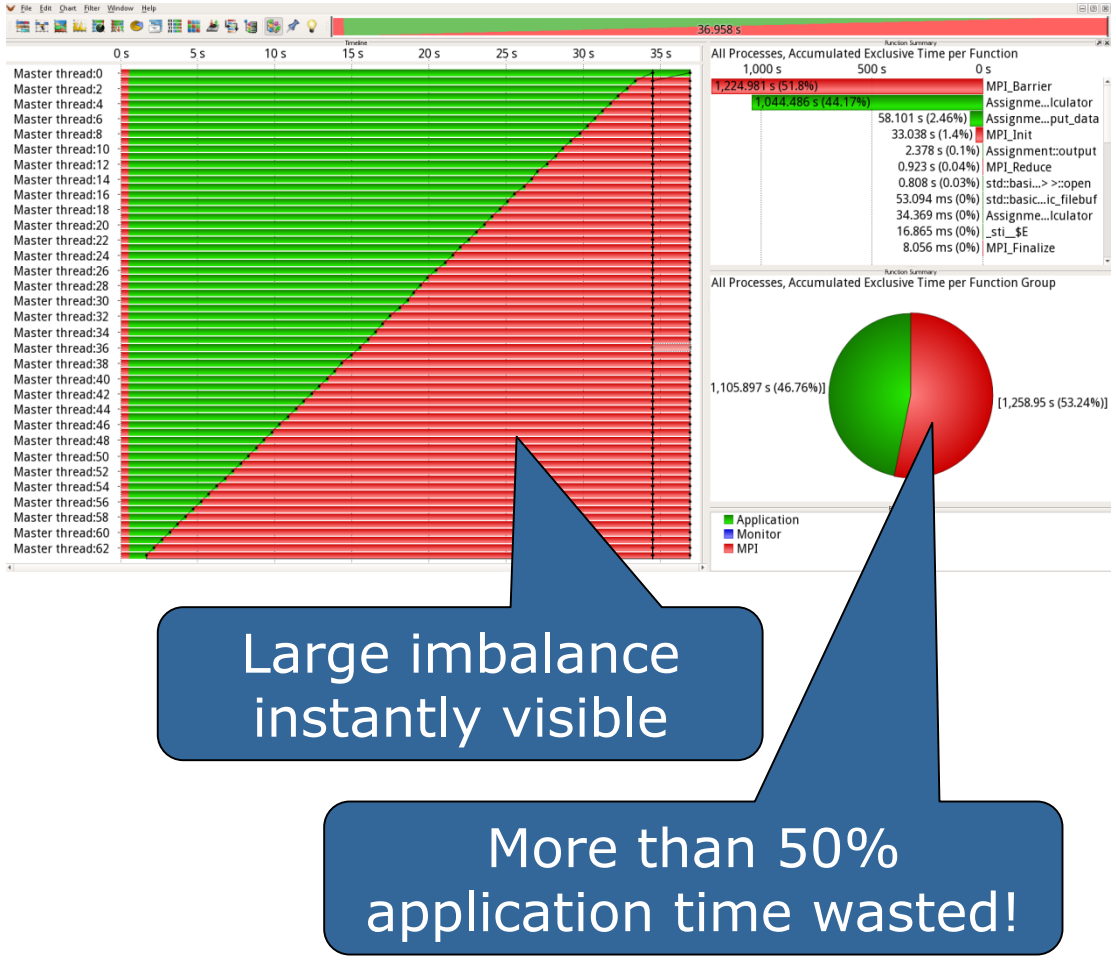


Event Trace Visualization with Vampir

The value of seeing how an application executes on the machine

- Application code computing coulomb forces
- The workload was distributed evenly across available processes
- The user expected perfect parallelized code
- However the underlying algorithm worked differently than expected

Visualization of the application execution instantly shows a problem in the parallelization approach



Main Performance Charts of Vampir

Timeline Charts


Master Timeline ➡ *all threads' activities*


Process Timeline ➡ *single thread's activities*


Summary Timeline ➡ *all threads' function call statistics*


Performance Radar ➡ *all threads' performance metrics*


Counter Data Timeline ➡ *single threads' performance metrics*


I/O Timeline ➡ *all threads' I/O activities*

Summary Charts


Function Summary


Message Summary


I/O Summary

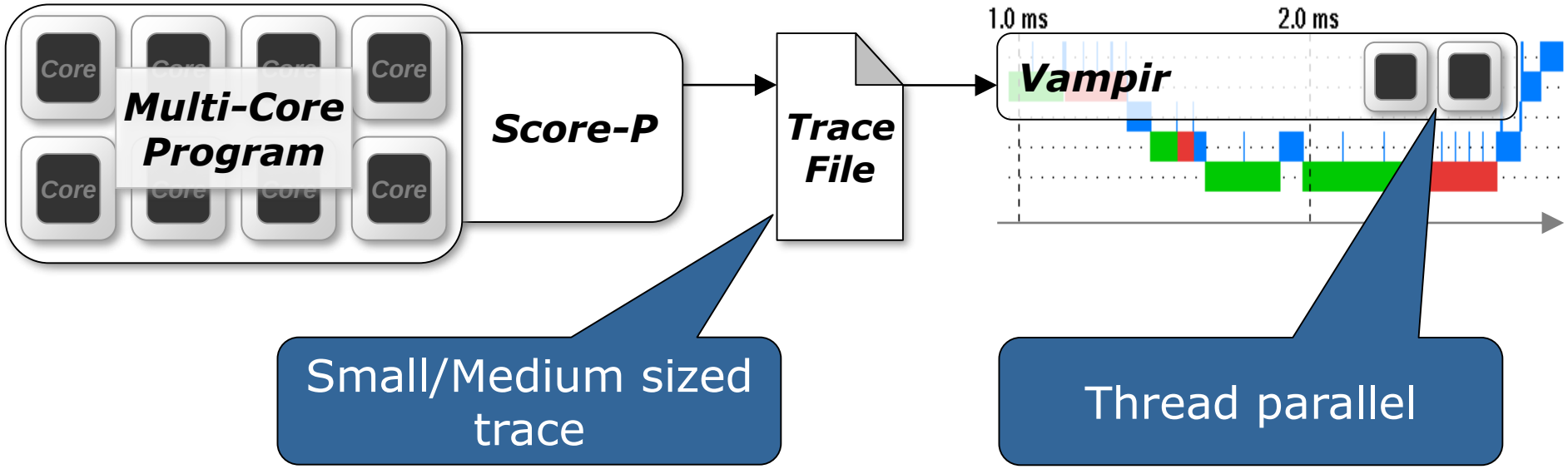

Process Summary


Communication Matrix View


Call Tree

Visualization Modes (1)
Directly on front end or local machine

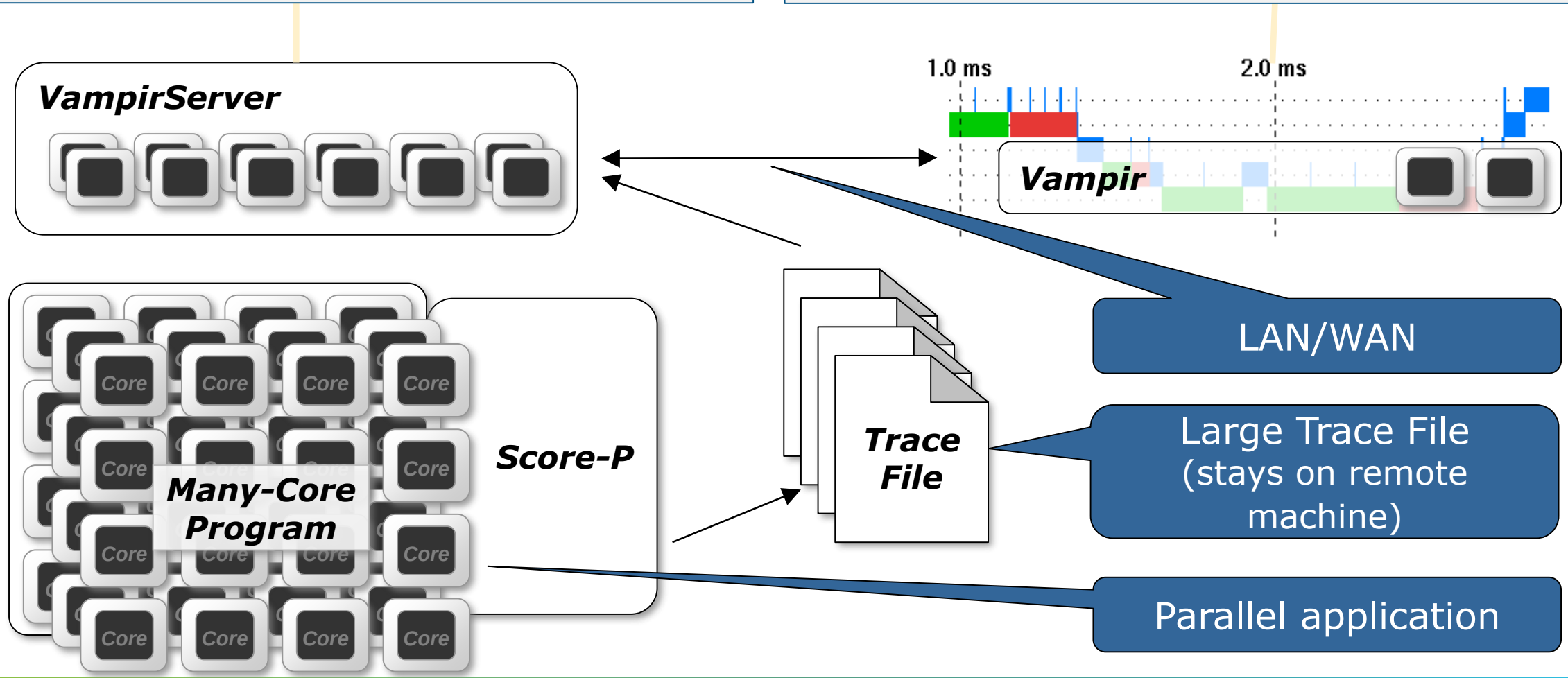
```
% vampir
```



Visualization Modes (2)
On local machine with remote VampirServer

```
% vampirserver start
```

```
% vampir
```



Hands-on:
Visualizing and Analyzing NPB-MZ-MPI / BT

Help! Where is my trace file?

```
% ls $SCRATCH/NPB3.3-MZ-MPI/bin.scorep/\
> scorep_bt-mz_C_32x4_trace
profile.cubex  scorep.cfg    traces/   traces.def  traces.otf2




% ls ~tg828282/Tutorial/Experiments/scorep_bt-mz_C_32x4_trace
profile.cubex  scorep.cfg    traces/   traces.def  traces.otf2
```

- If you followed the Score-P hands-on up to the trace experiment
- If you did not follow to that point, take a prepared trace

Installing Vampir on Local Machine

```
% scp <user>@stampede2.tacc.utexas.edu: \
  /home1/02438/winklerf/vampir/vampir-<x64|linux|mac>* .

% scp <user>@stampede2.tacc.utexas.edu: \
  /home1/02438/winklerf/vampir/vampir.license .
```

- Copy Vampir to your local machine
- Install Vampir using a demo license

Starting VampirServer on Stampede

```
% vampirserver start
Launching VampirServer...
Submitting batch job (this might take a while)...
```

- Start VampirServer on Stampede2

Starting VampirServer on Stampede

```
% vampirserver start
Launching VampirServer...
Submitting batch job (this might take a while)...

VampirServer 9.4.0  (r10676)
Licensed to ZIH, TU Dresden (@ISC 2018)
Running 4 analysis processes... (abort with \
  vampirserver stop 28974)
VampirServer <28974> listens on: \
  c401-602.stampede2.tacc.utexas.edu:30019
```

- Start VampirServer on Stampede2

Copy host:port

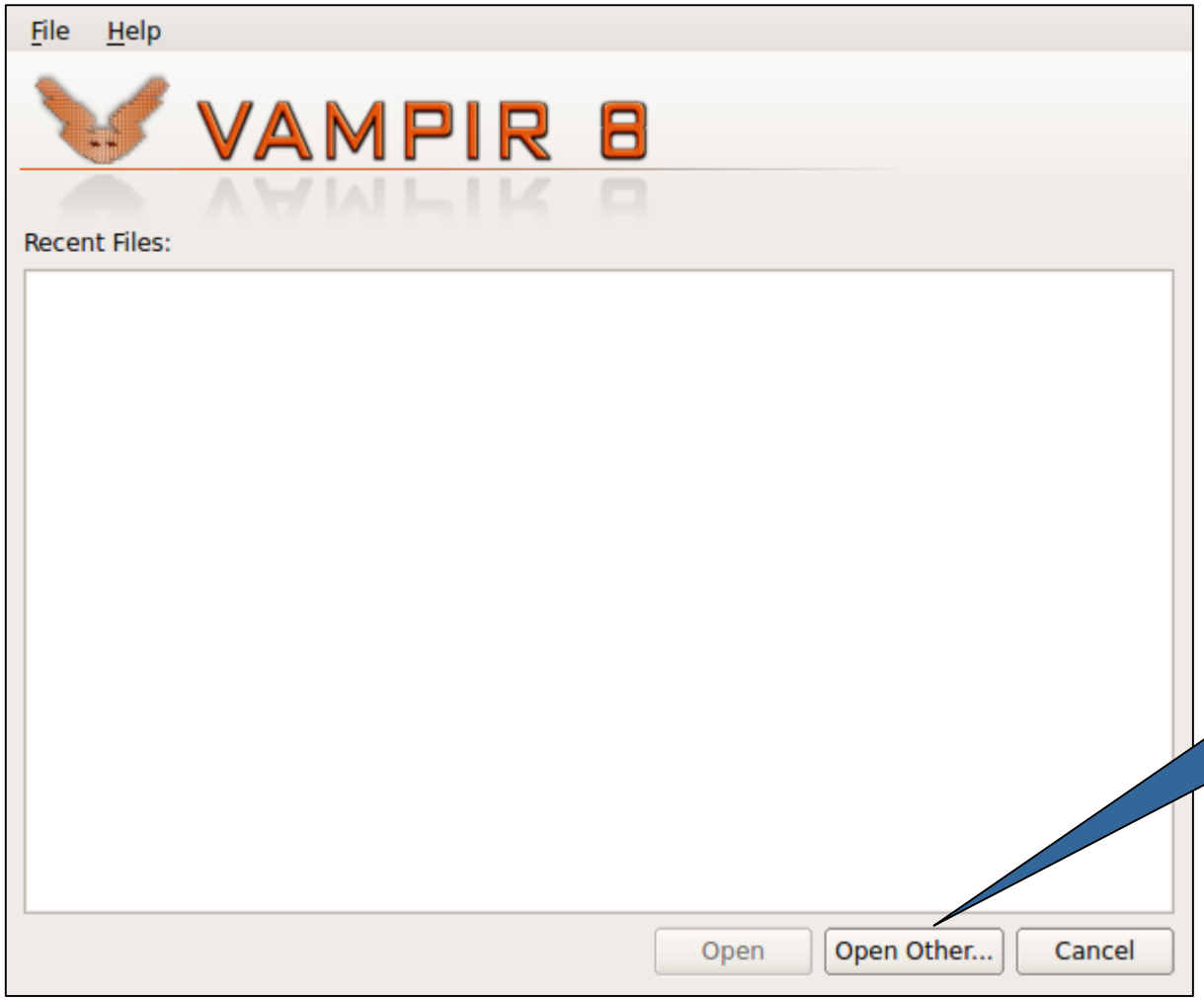
Start Vampir

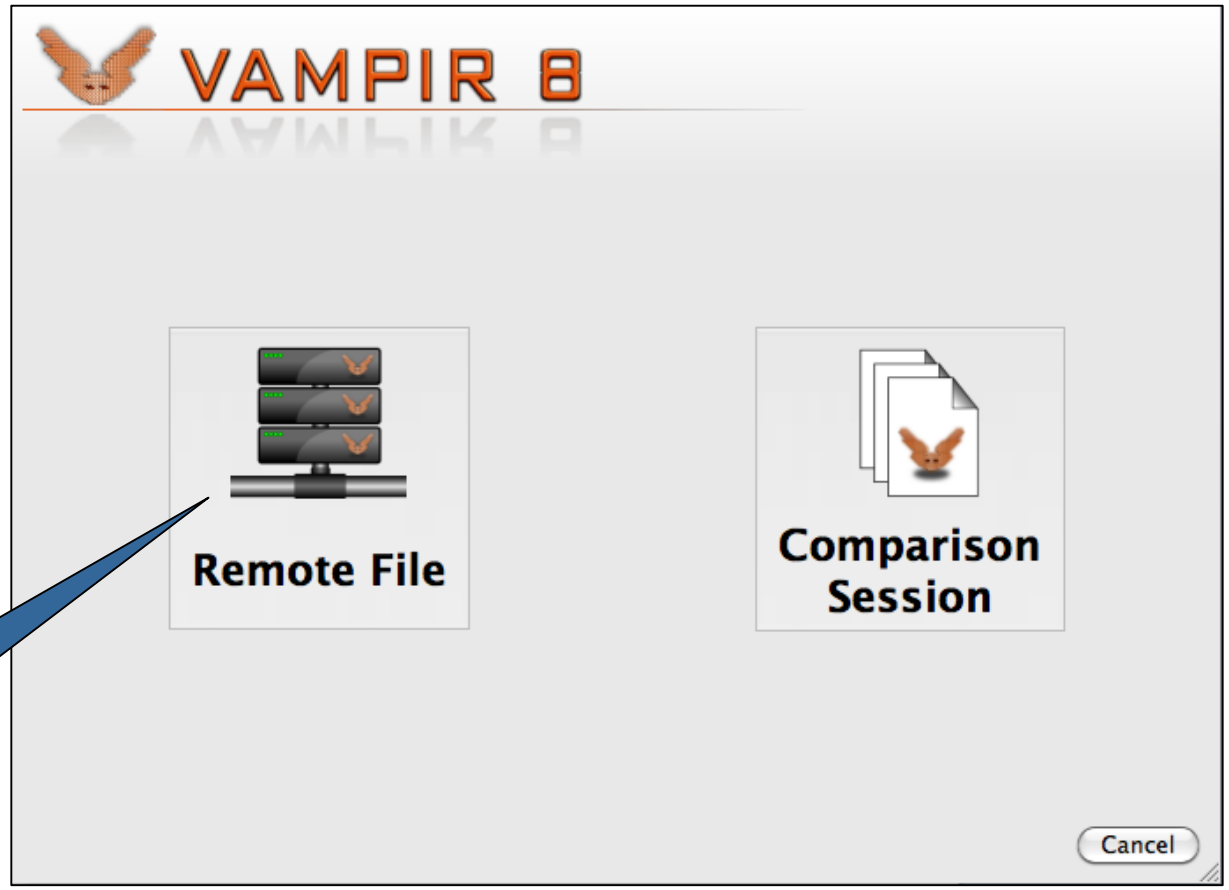
```
% ssh -N -L 30000:c401-602.stampede2.tacc.utexas.edu:30019 \
    <user>@stampede2.tacc.utexas.edu
```

- Open a port forwarding to Stampede2 to be able to access the VampirServer

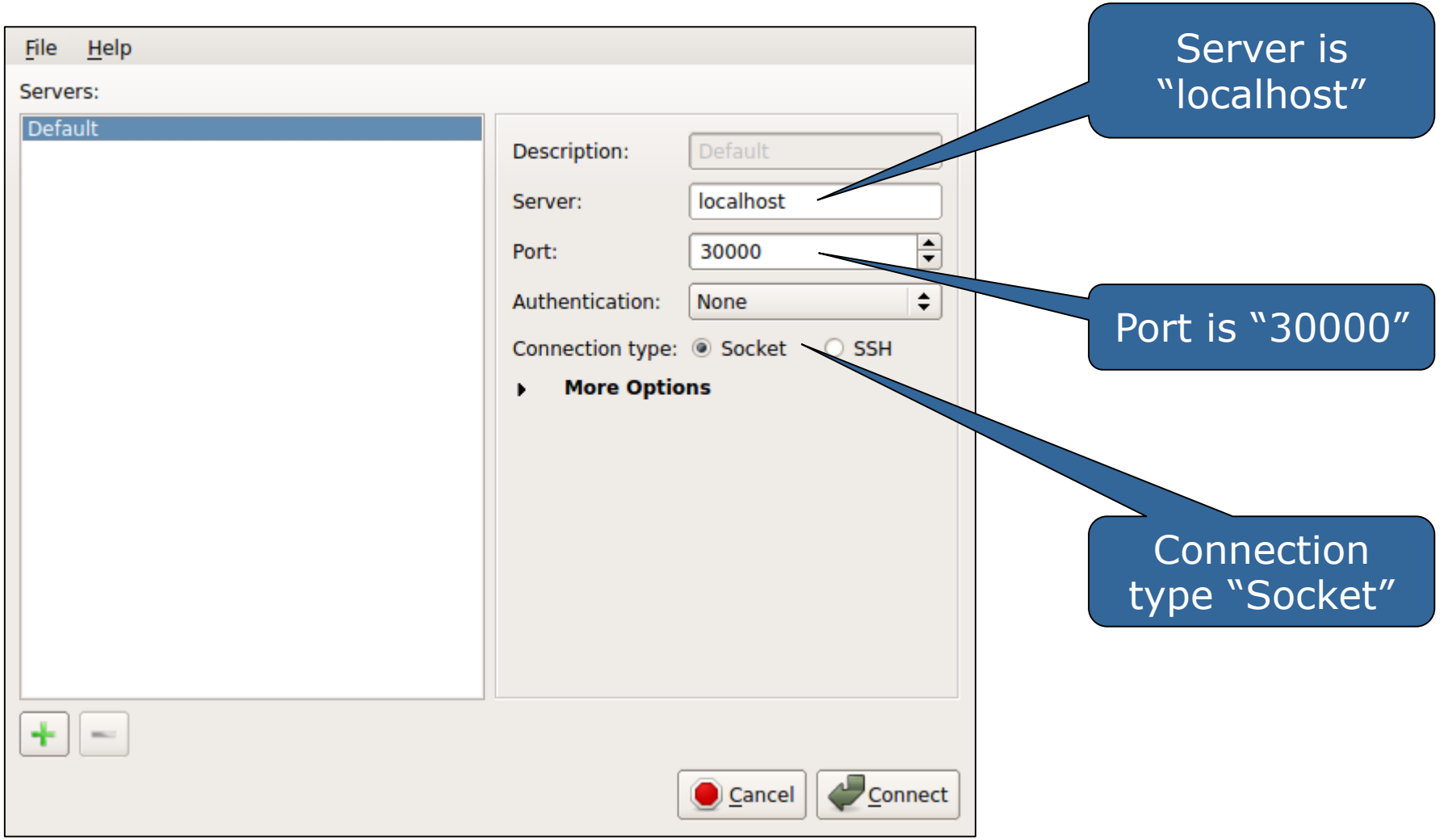
host:port from VampirServer output

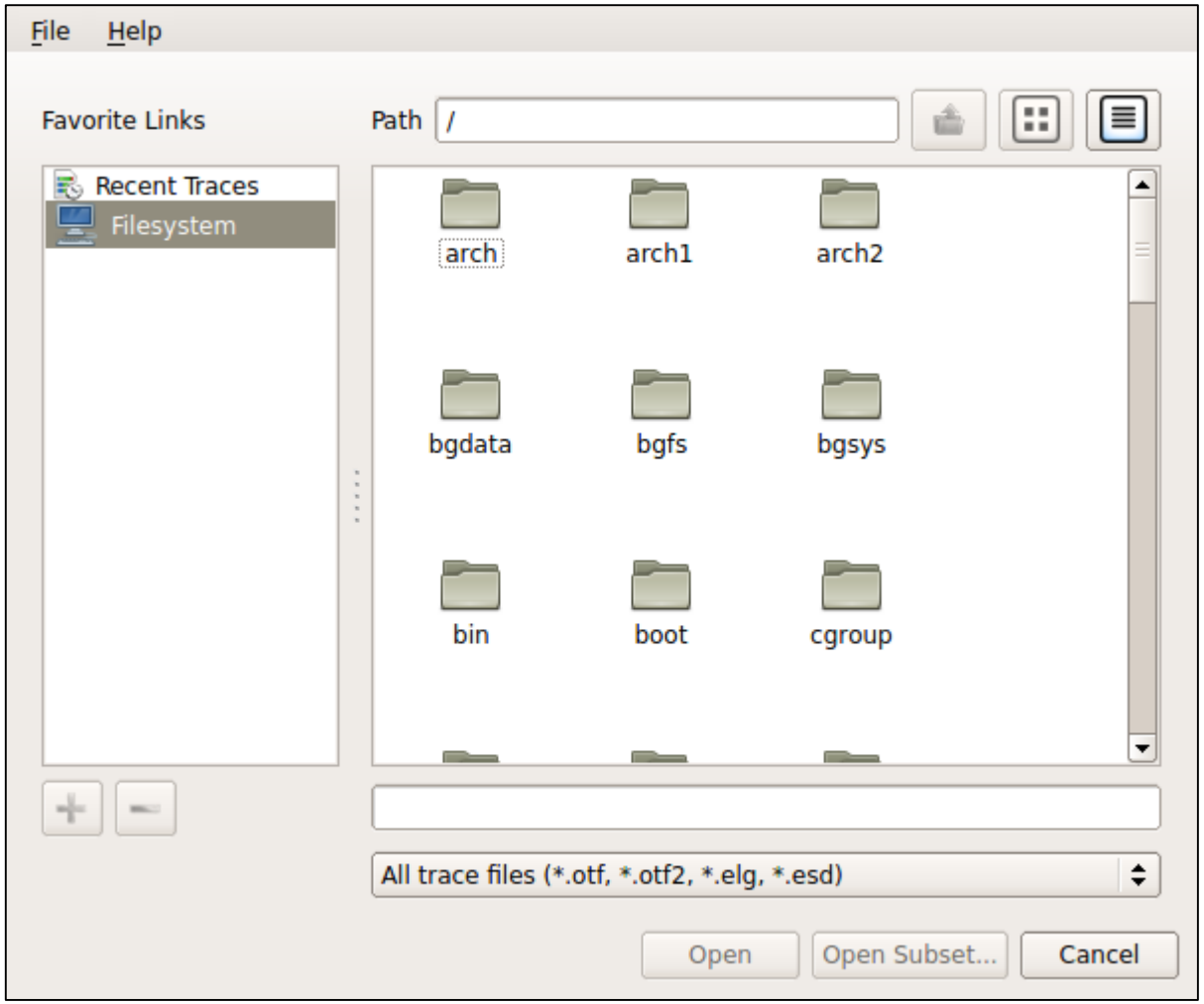
Start Vampir on local computer



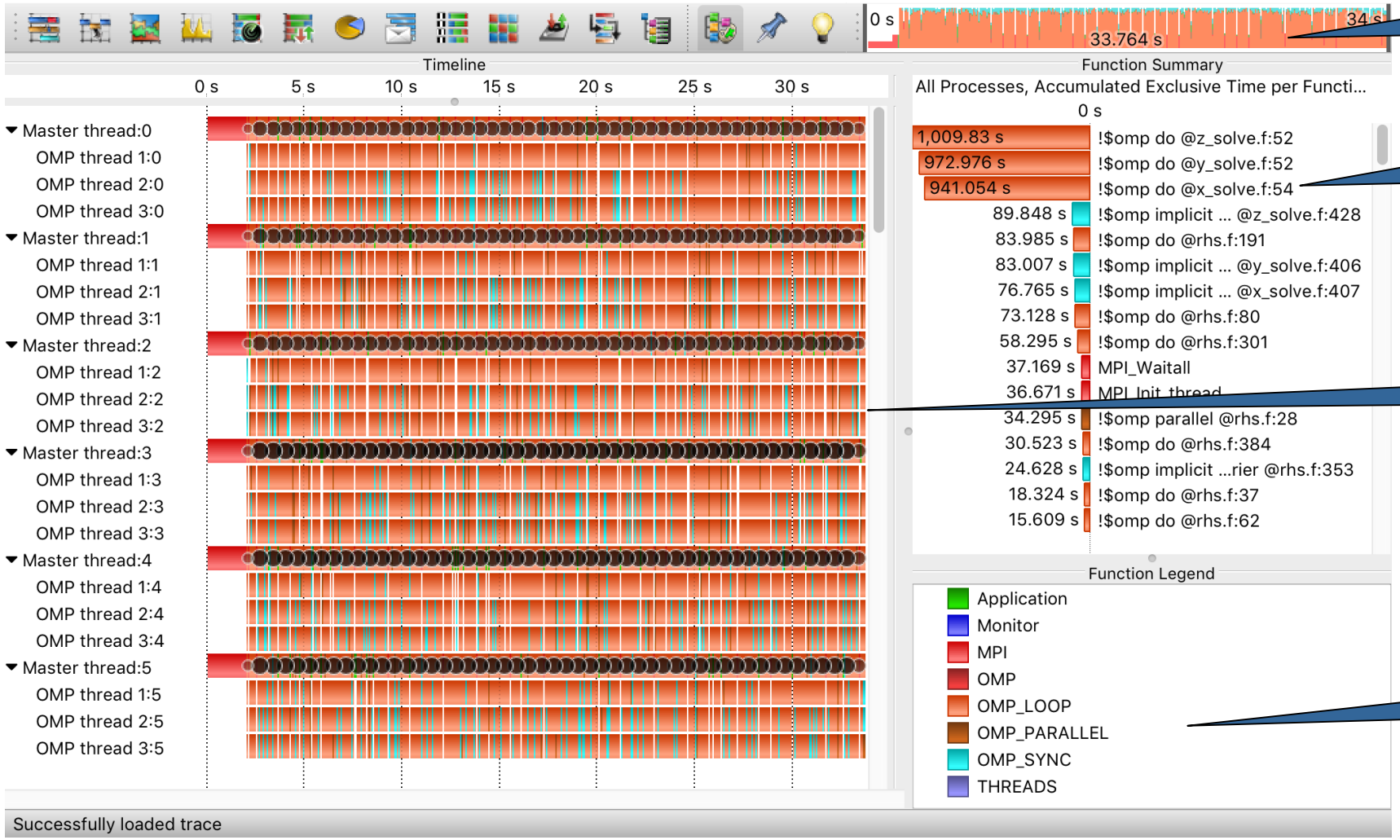


Select "Remote File"


File
Help
Servers:
Default
Description:
Default
Server:
localhost
Port:
30000
Authentication:
None
Connection type:
Socket
SSH
More Options
Cancel
Connect
Server is "localhost"
Port is "30000"
Connection type "Socket"


File
Help
Favorite Links
Path
/
Recent Traces
Filesystem
arch
arch1
arch2
bgdata
bgfs
bgsys
bin
boot
cgroup
All trace files (*.otf, *.otf2, *.elg, *.esd)
Open
Open Subset...
Cancel

Visualization of the NPB-MZ-MPI / BT trace



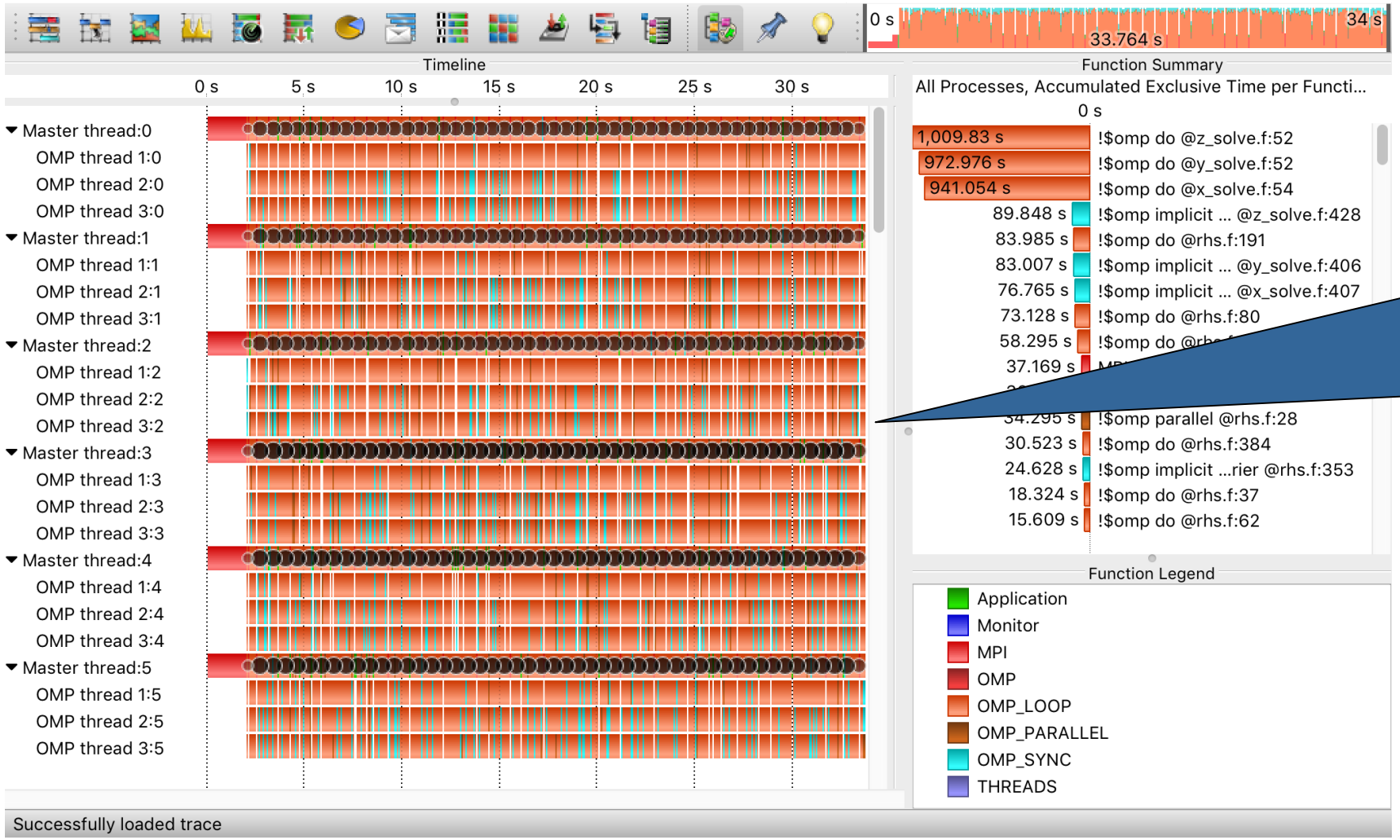
Navigation Toolbar

Function Summary

Master Timeline

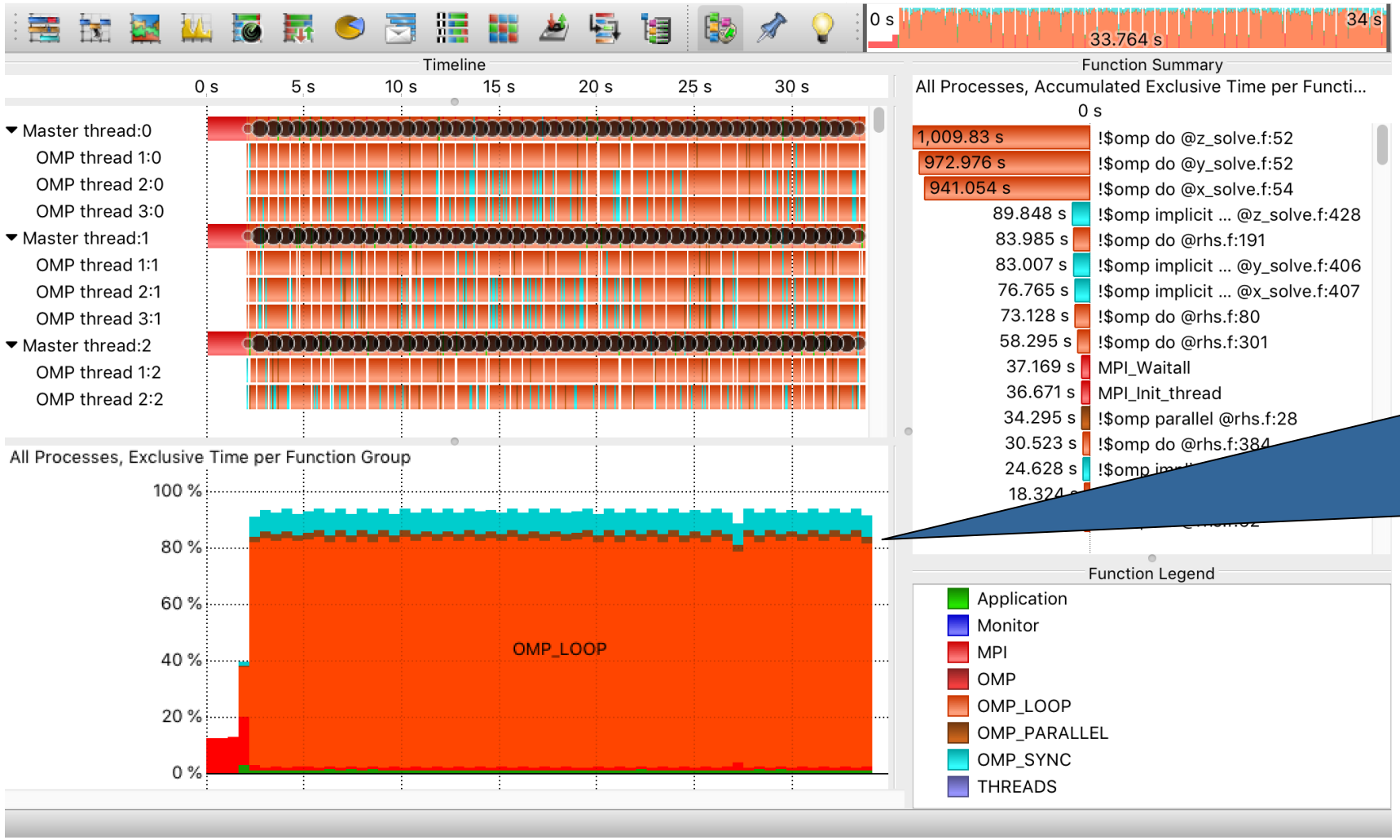
Function Legend

Visualization of the NPB-MZ-MPI / BT trace
Master Timeline



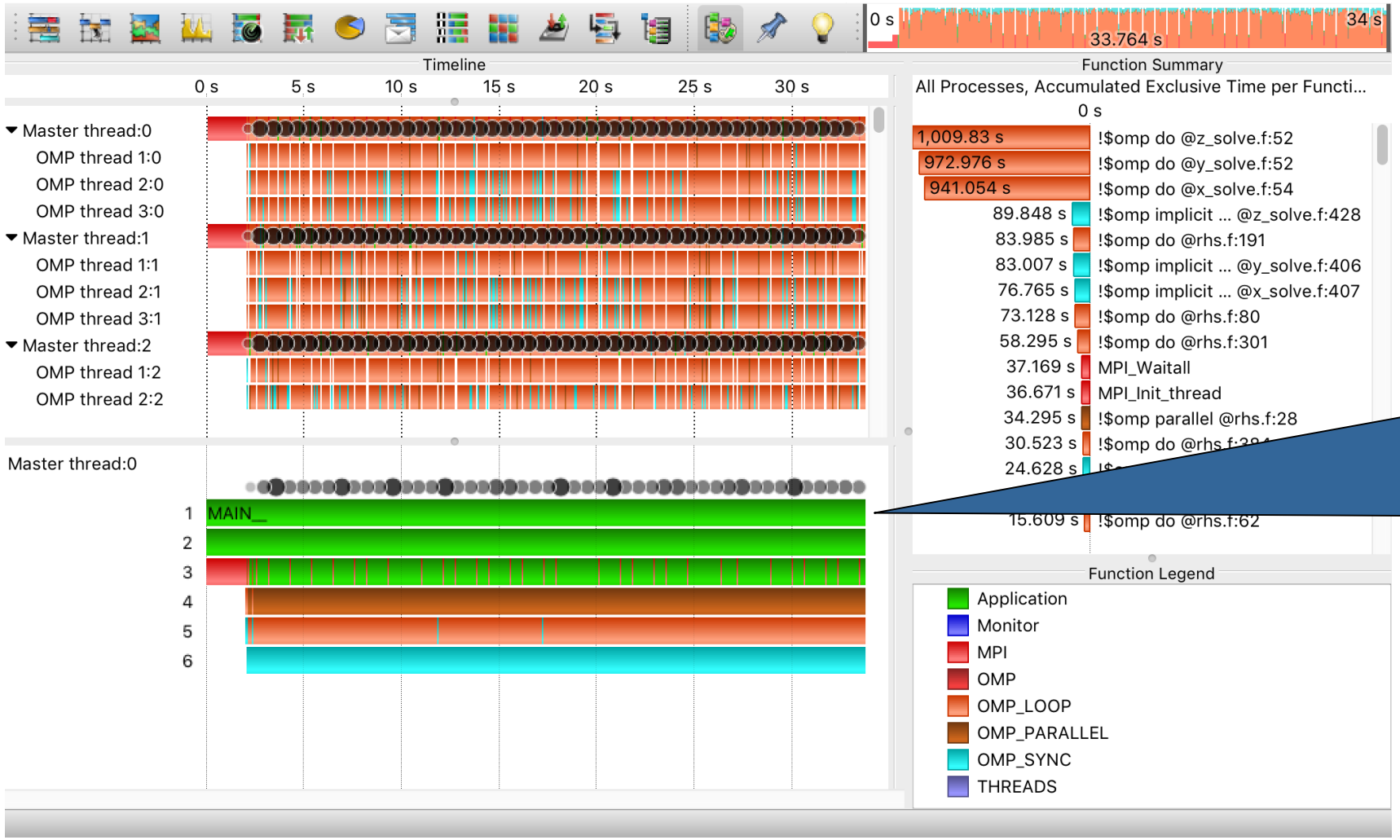
Detailed information about functions, communication and synchronization events for collection of processes.

Visualization of the NPB-MZ-MPI / BT trace
Summary Timeline



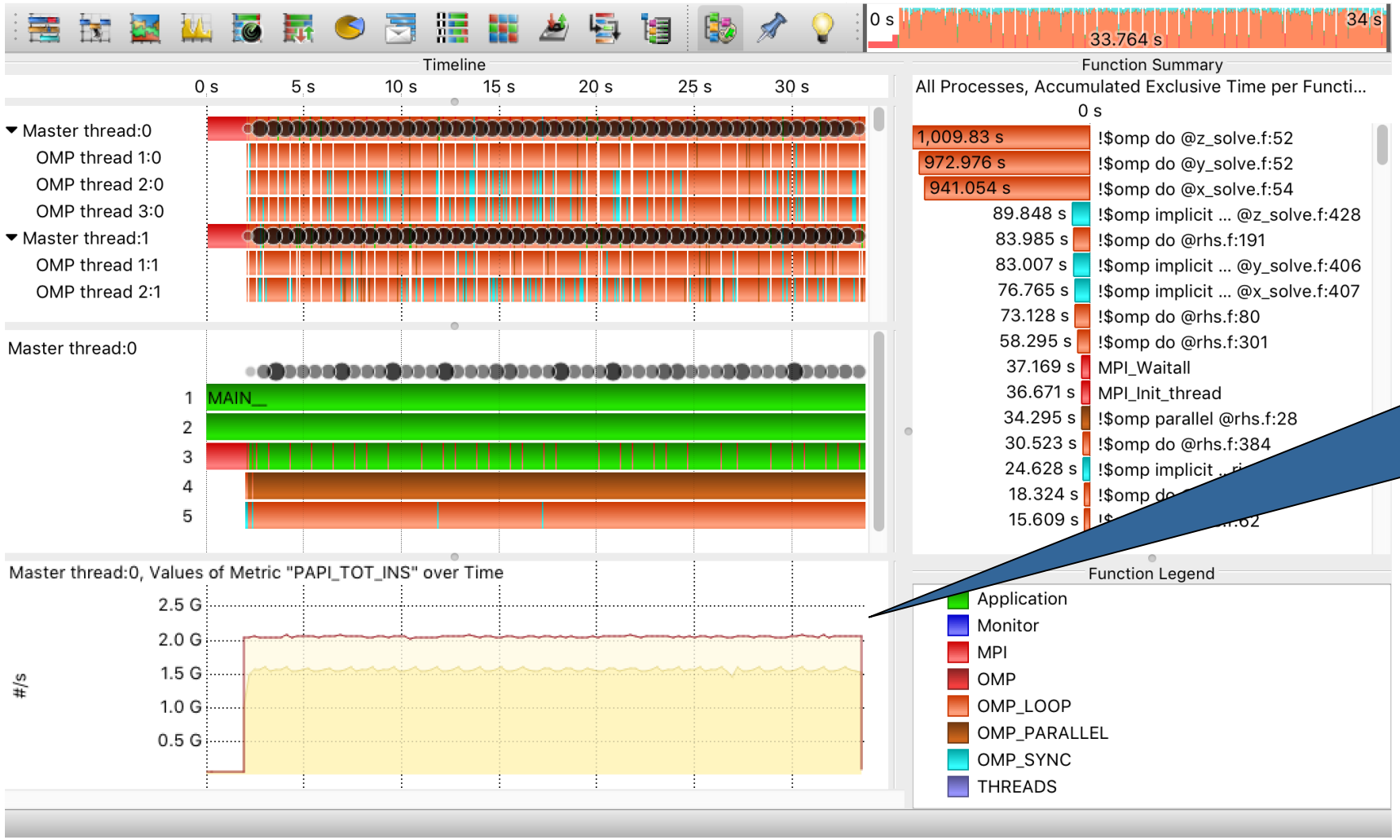
Fractions of the number of processes that are actively involved in given activities at a certain point in time.

Visualization of the NPB-MZ-MPI / BT trace
Process Timeline



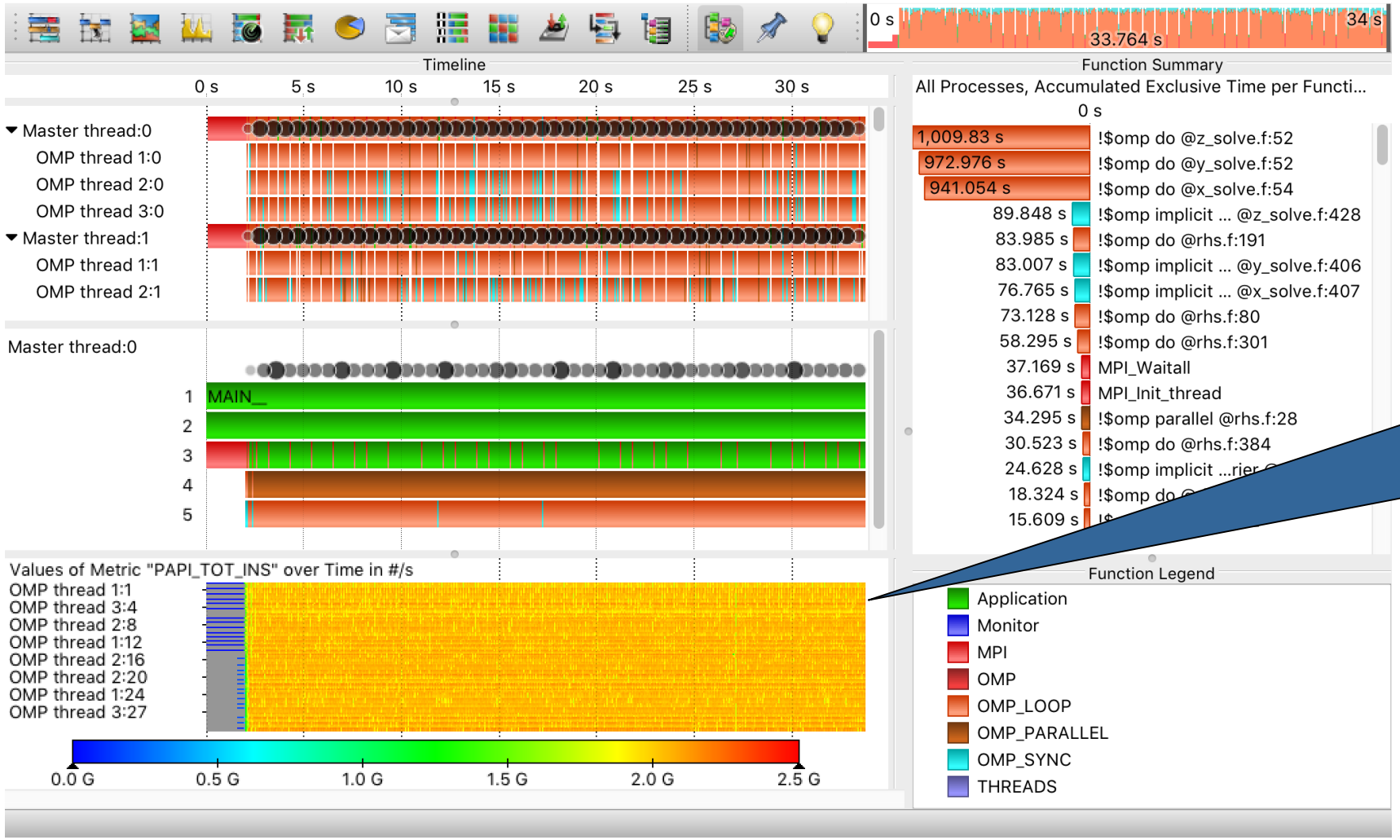
Detailed information about different levels of function calls in a stacked bar chart for an individual process.

Visualization of the NPB-MZ-MPI / BT trace
Counter Timeline



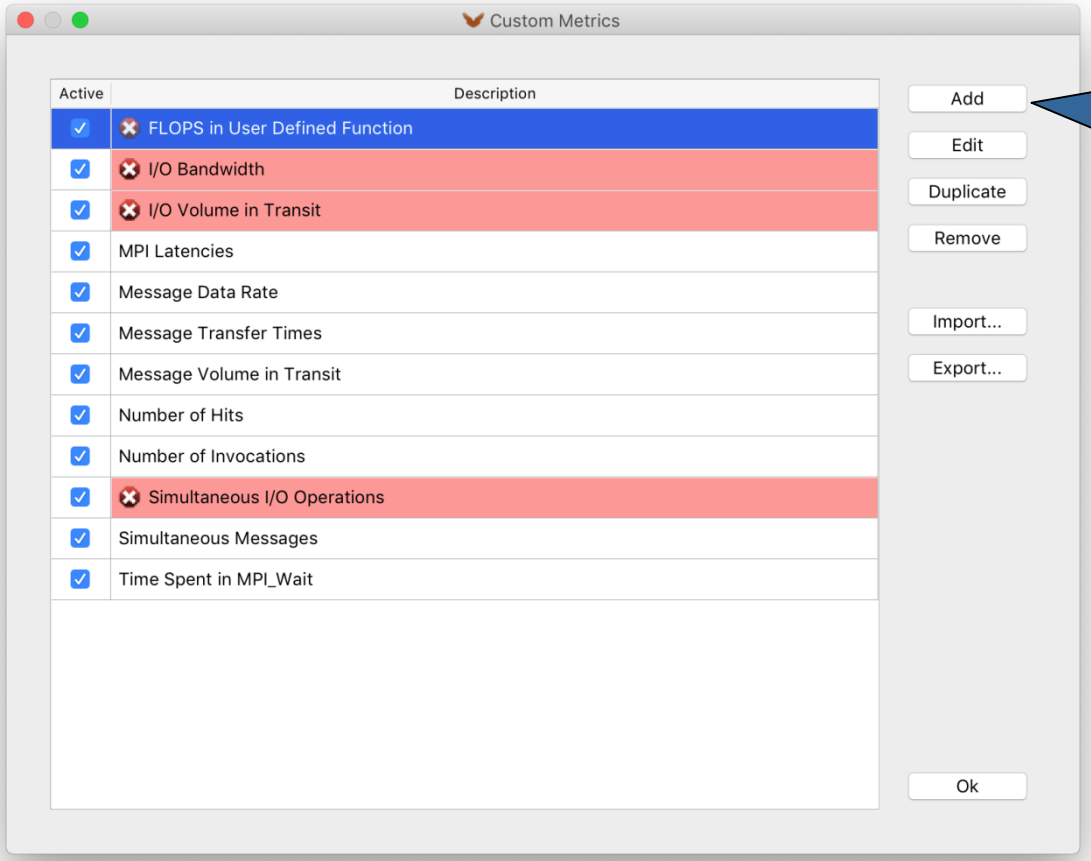
Detailed counter information over time for an individual process.

Visualization of the NPB-MZ-MPI / BT trace
Performance Radar

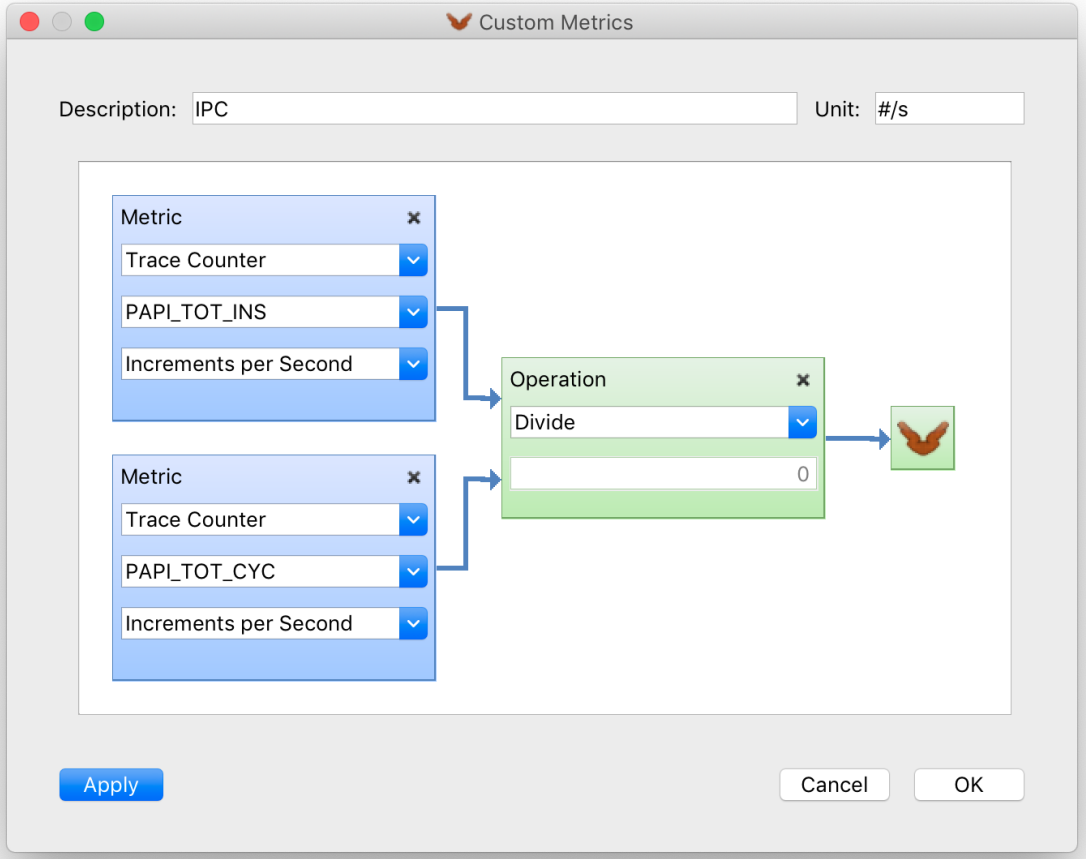


Detailed counter information over time for a collection of processes.

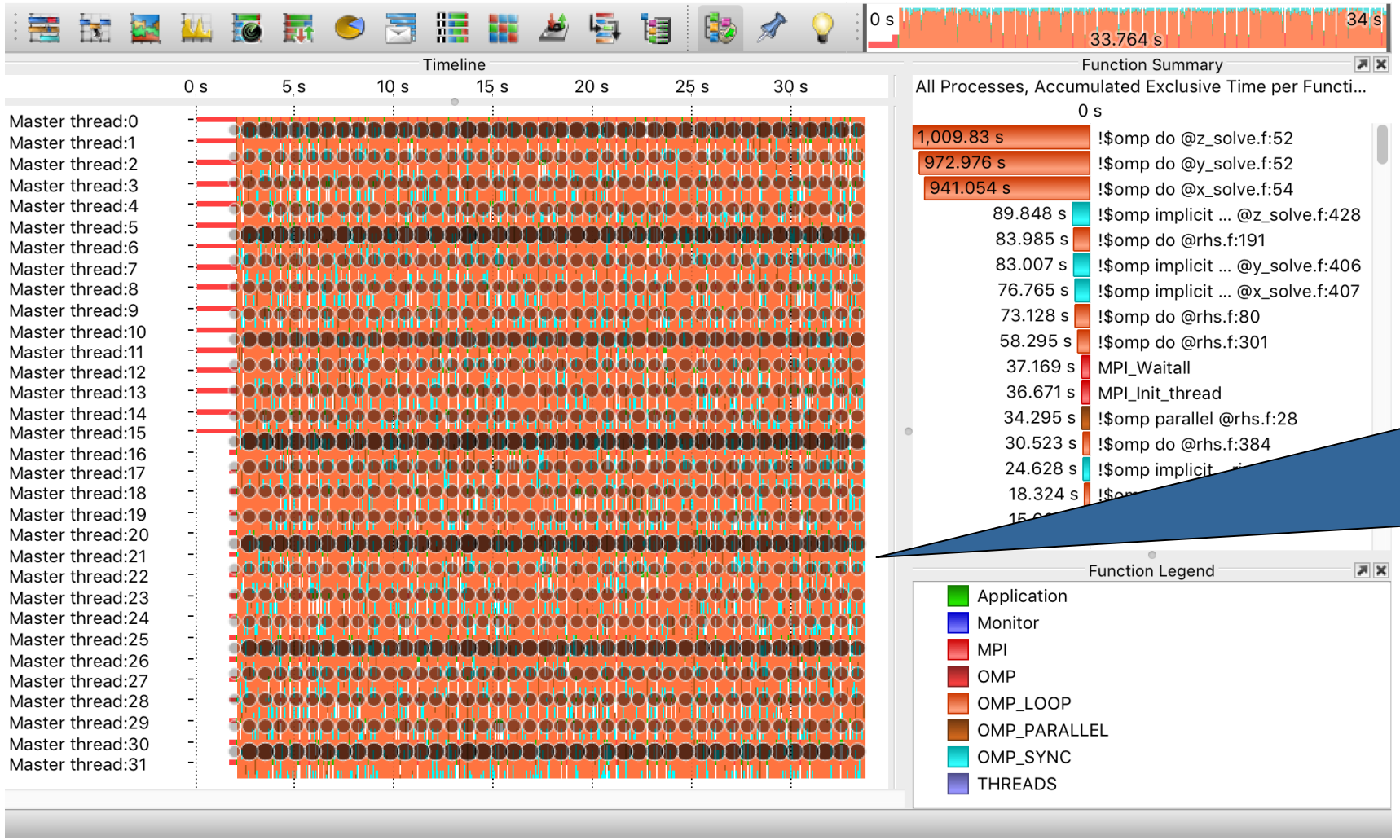
Visualization of the NPB-MZ-MPI / BT trace
Custom Metrics Build-In Editor



Add IPC as new custom metric.

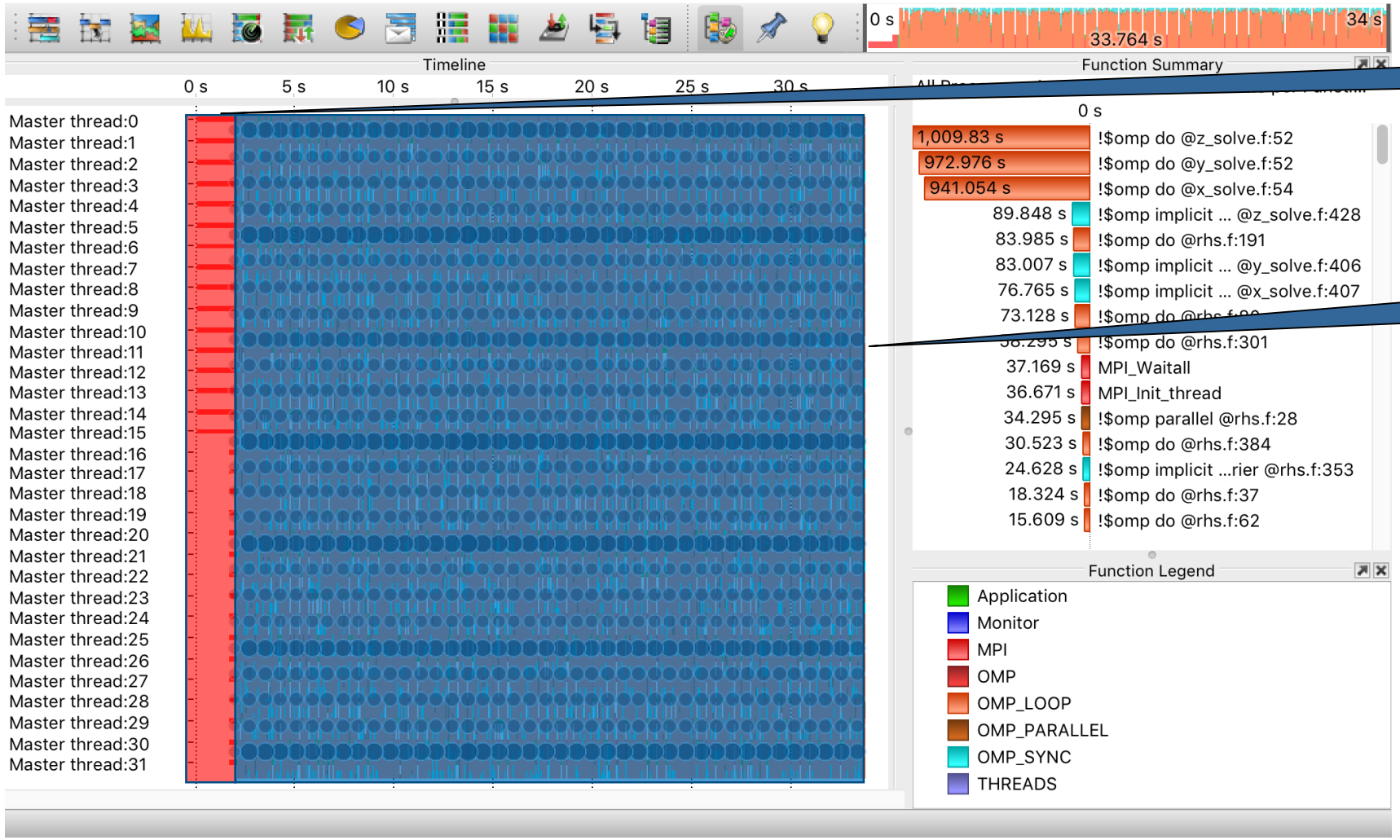


Visualization of the NPB-MZ-MPI / BT trace
Fit to chart height feature

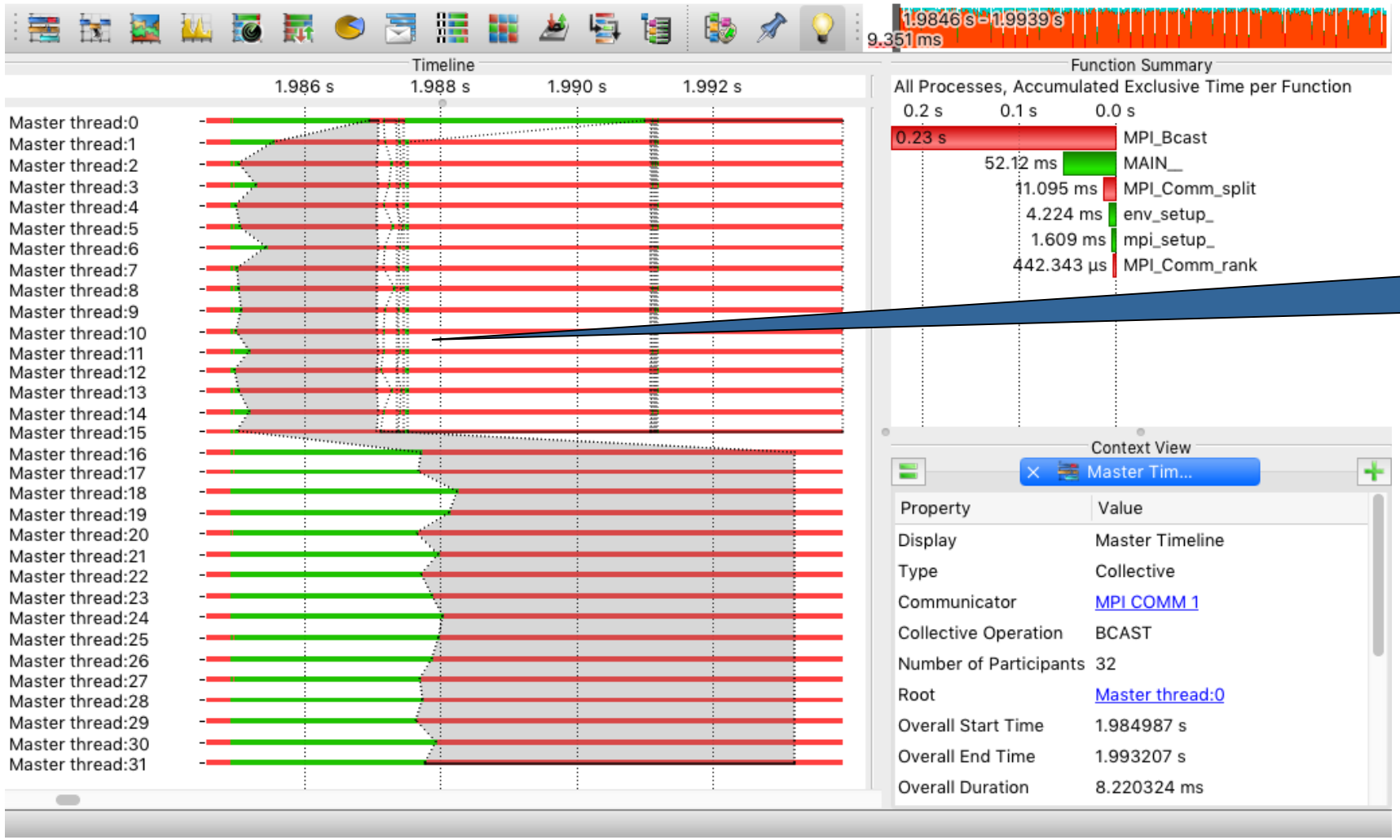


Overview of the entire application run across all processes based on available pixels on screen.

Visualization of the NPB-MZ-MPI / BT trace
Typical program phases

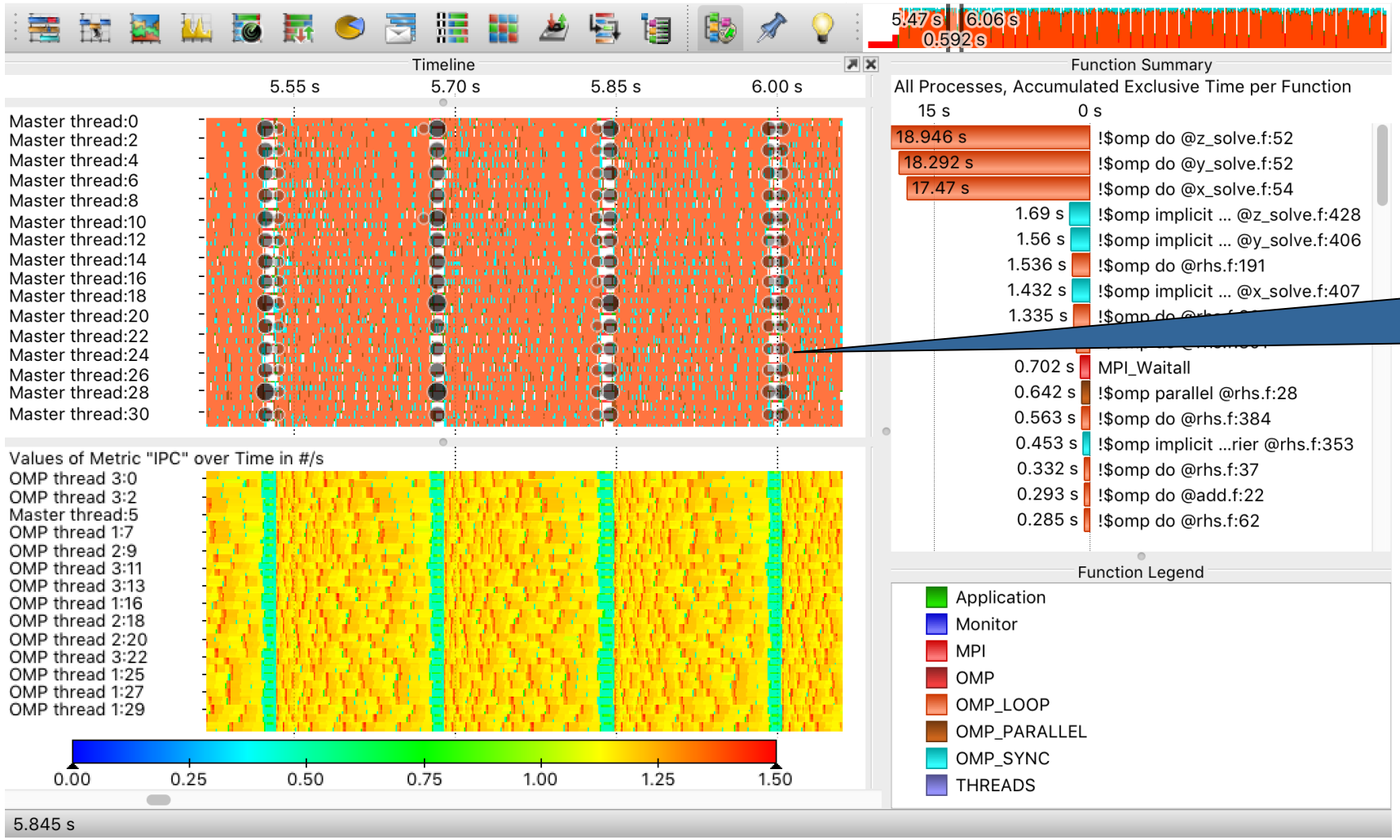


Visualization of the NPB-MZ-MPI / BT trace
Zoom in: Initialization Phase



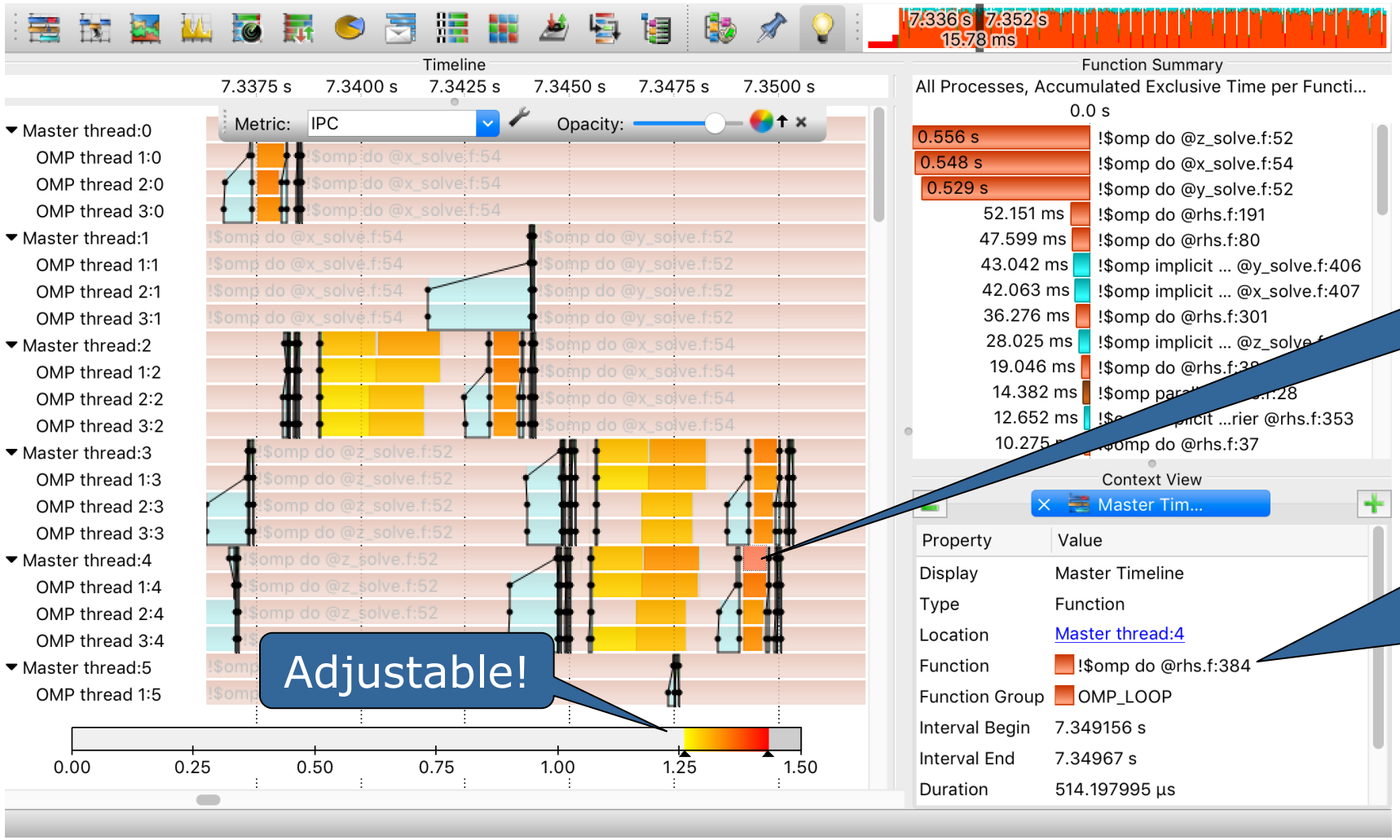
"Late broadcast" bottleneck.

Visualization of the NPB-MZ-MPI / BT trace
Zoom in: Computation Phase



MPI communication results in lower IPC.

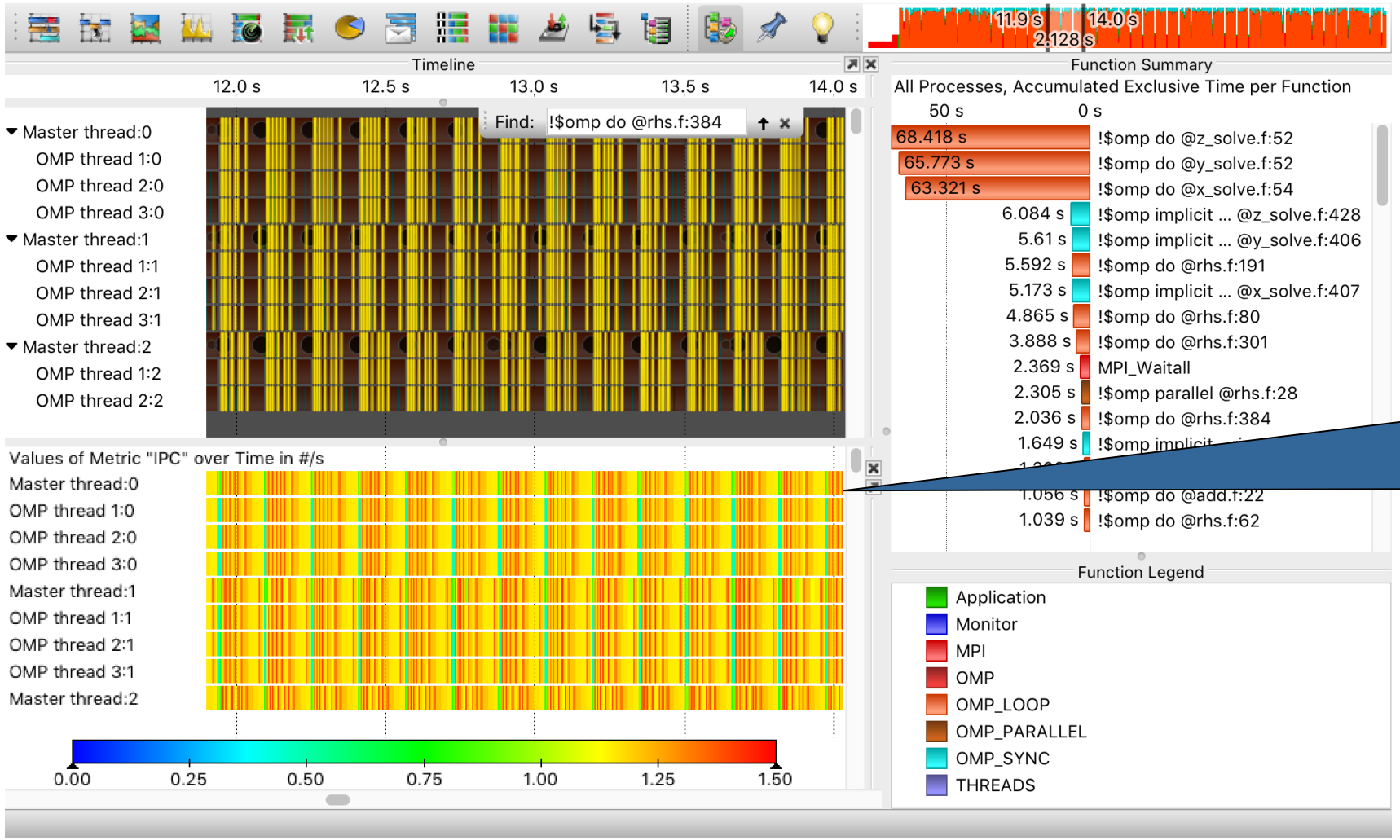
Visualization of the NPB-MZ-MPI / BT trace
Overlay Functionality of Master Timeline



Find function with highest IPC value.

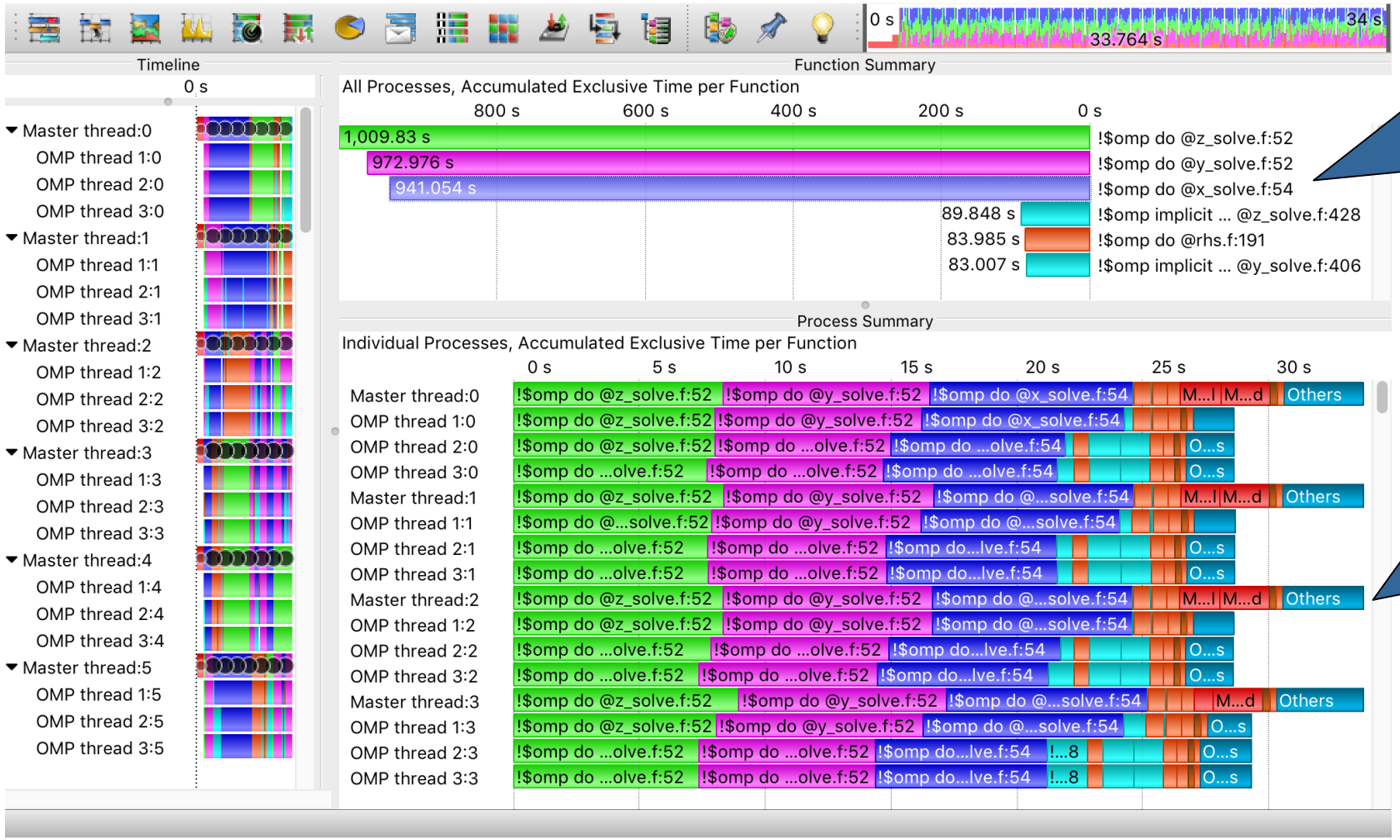
Context View shows detailed information about selected function.

Visualization of the NPB-MZ-MPI / BT trace
Find Function



Execution of function "!$omp do @rhs.f:384" results in higher IPC.

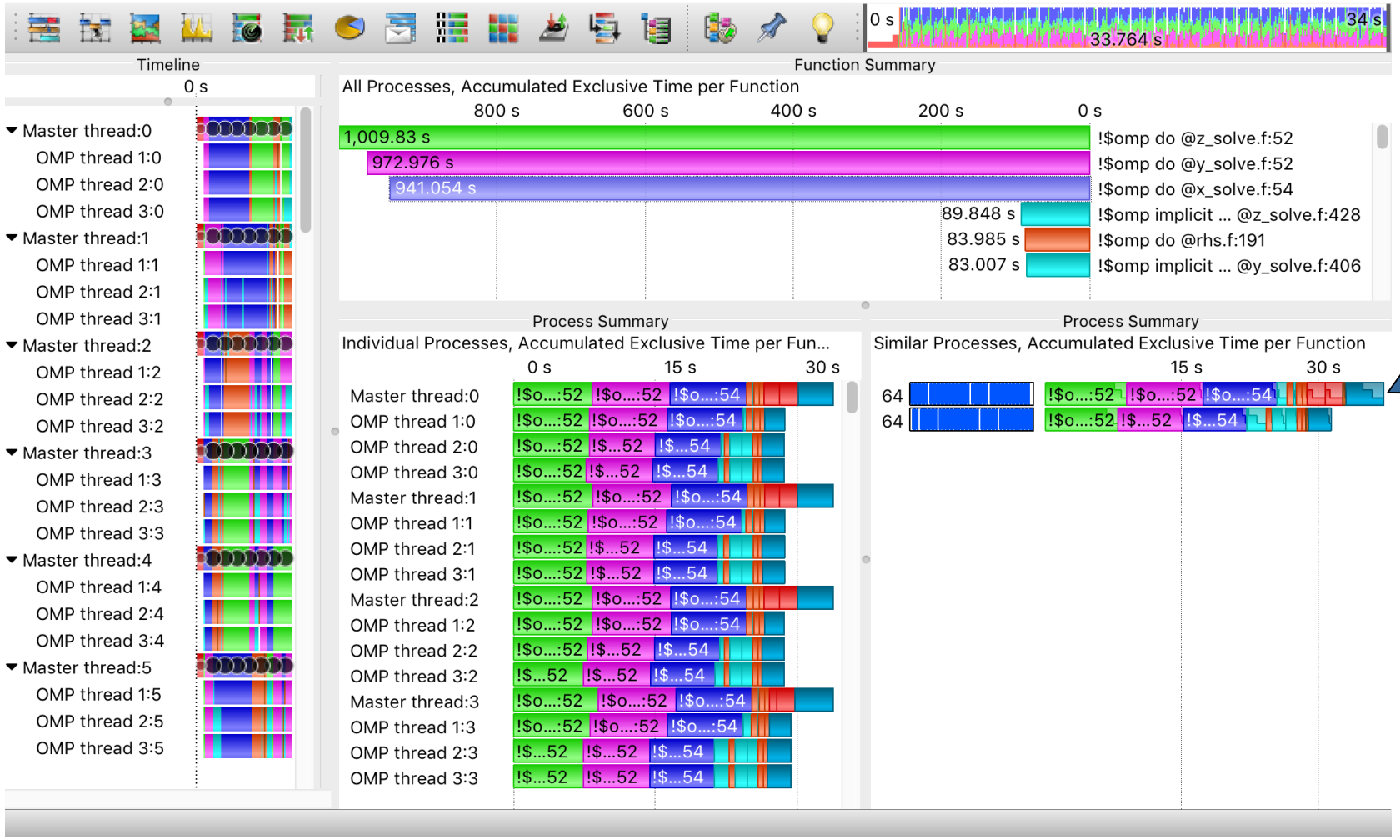
Visualization of the NPB-MZ-MPI / BT trace
Process Summary



Function Summary: Overview of the accumulated information across all functions and for a collection of processes.

Process Summary: Overview of the accumulated information across all functions and for every process independently.

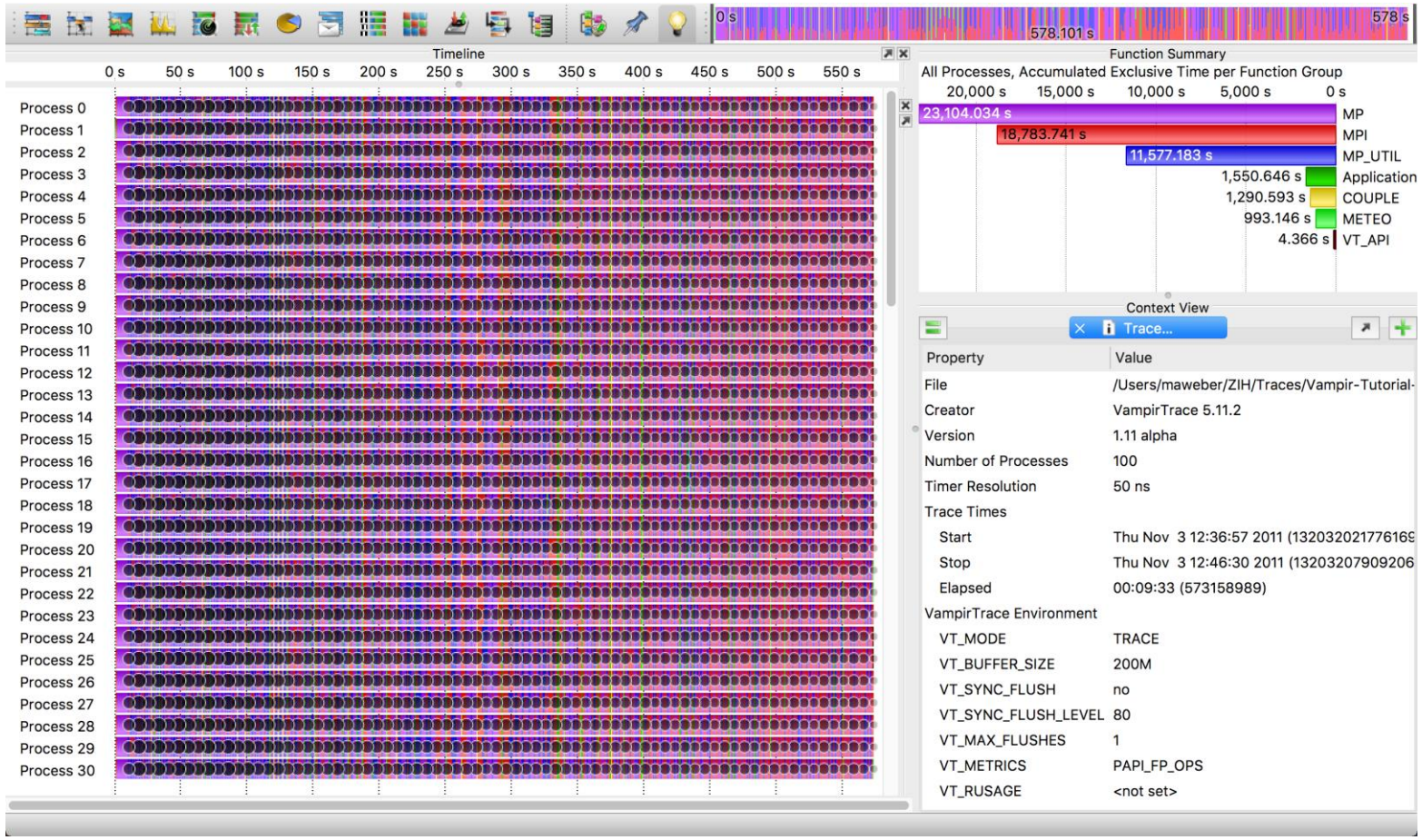
Visualization of the NPB-MZ-MPI / BT trace
Process Summary



Find groups of similar processes and threads by using summarized function information.

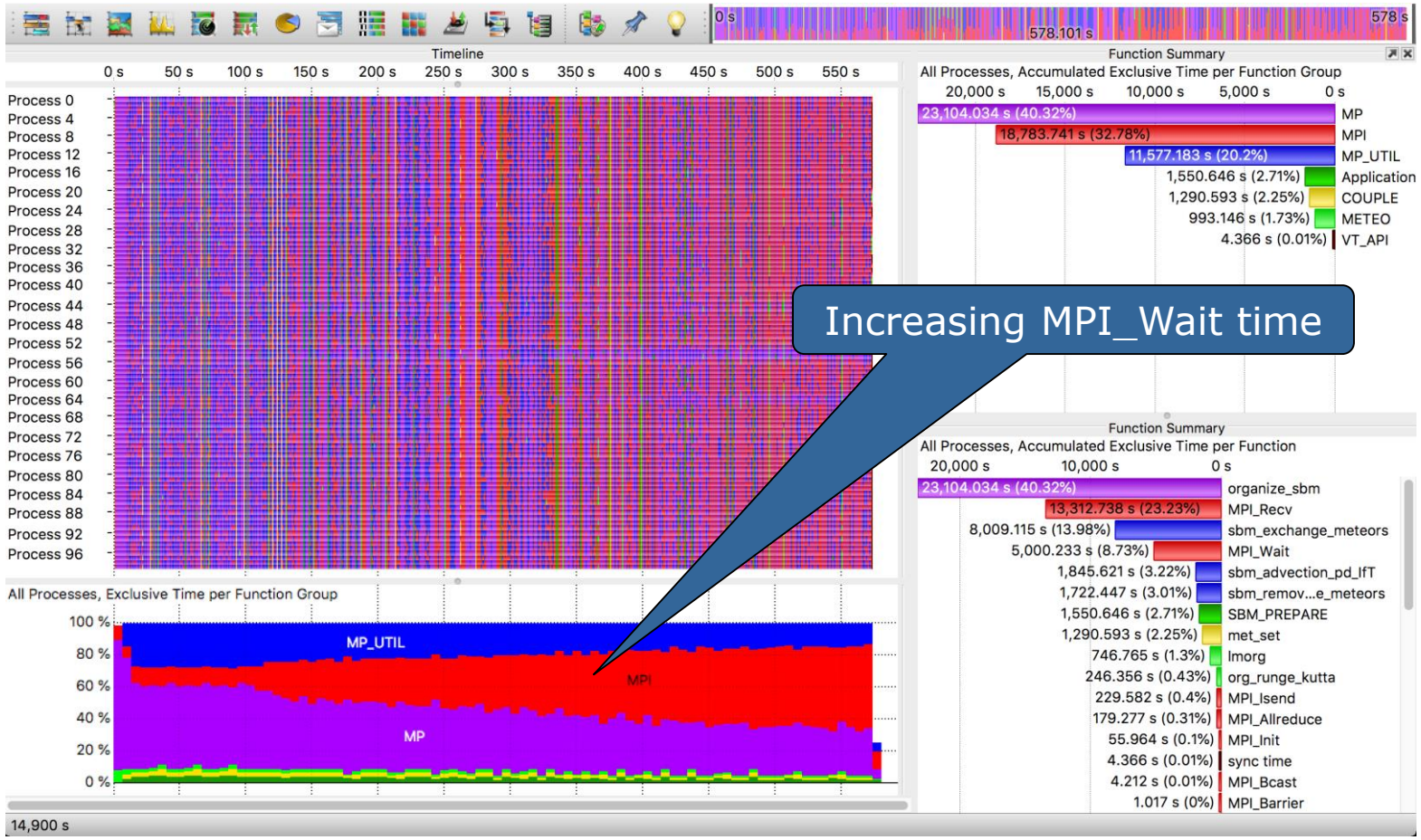
Vampir Case Study: Optimizing COSMO-SPECS

COSMO-SPECS Original



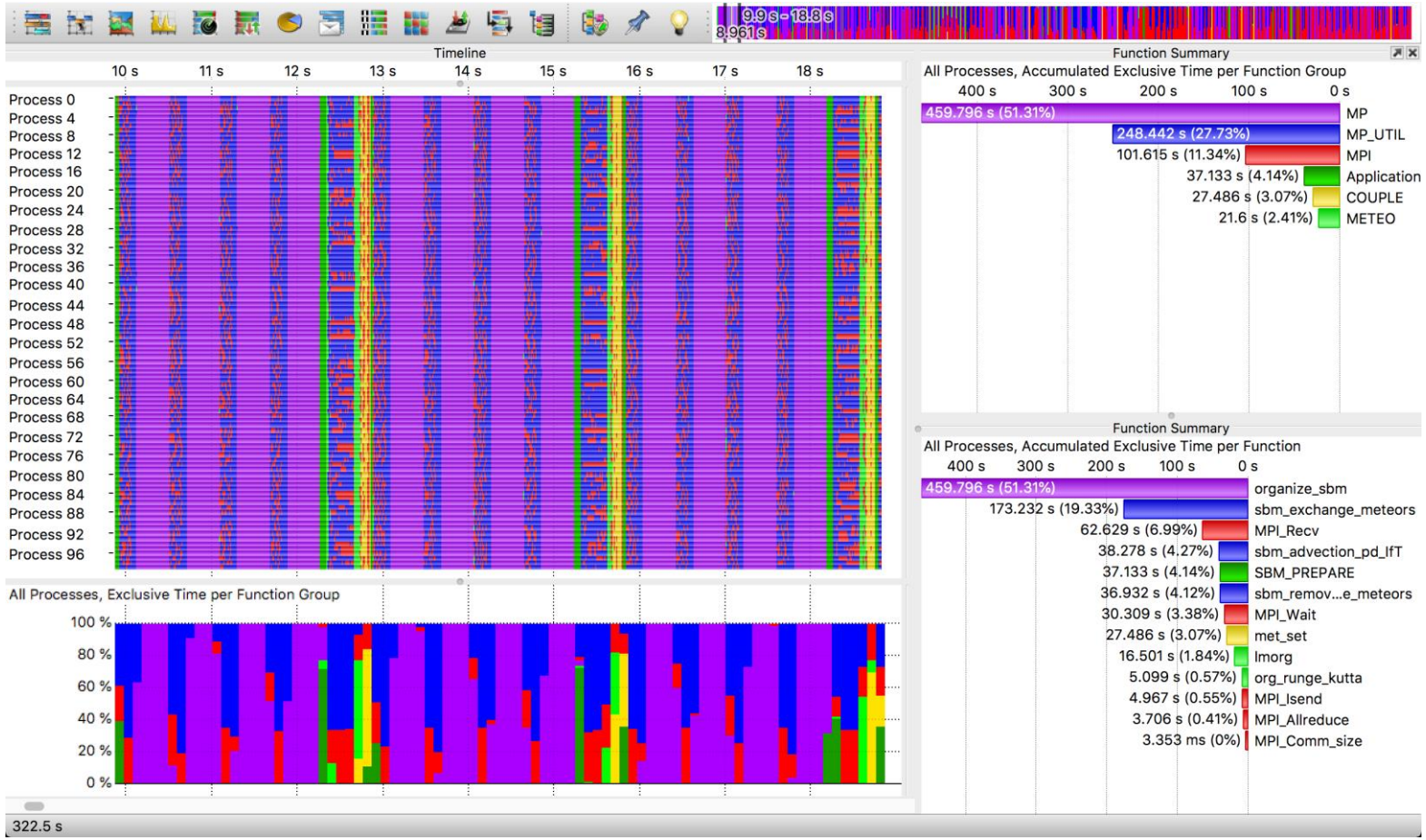
- Weather forecast code COSMO-SPECS
- Run with 100 processes
- COSMO: weather model (METEO group)
- SPECS: microphysics for accurate cloud calculation (MP and MP_UTIL group)
- Coupling of both models done in COUPLE group

COSMO-SPECS Original



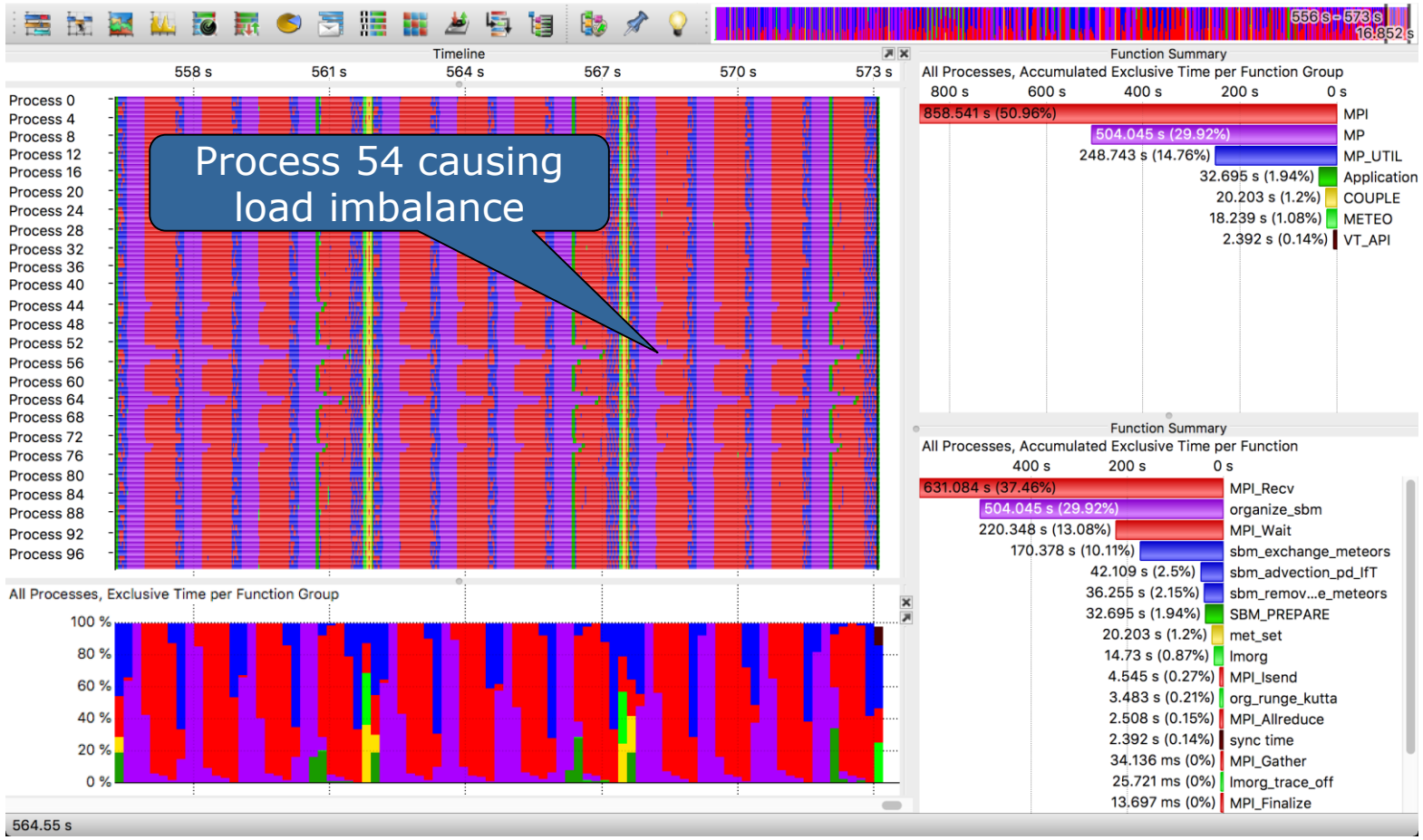
- Compared to METEO, MP and MP_UTIL are very compute intensive, however this is due to more complex calculations and no performance issue
- Problem: >32% of time spent in MPI
- MPI runtime share increases throughout the application run

COSMO-SPECS Original



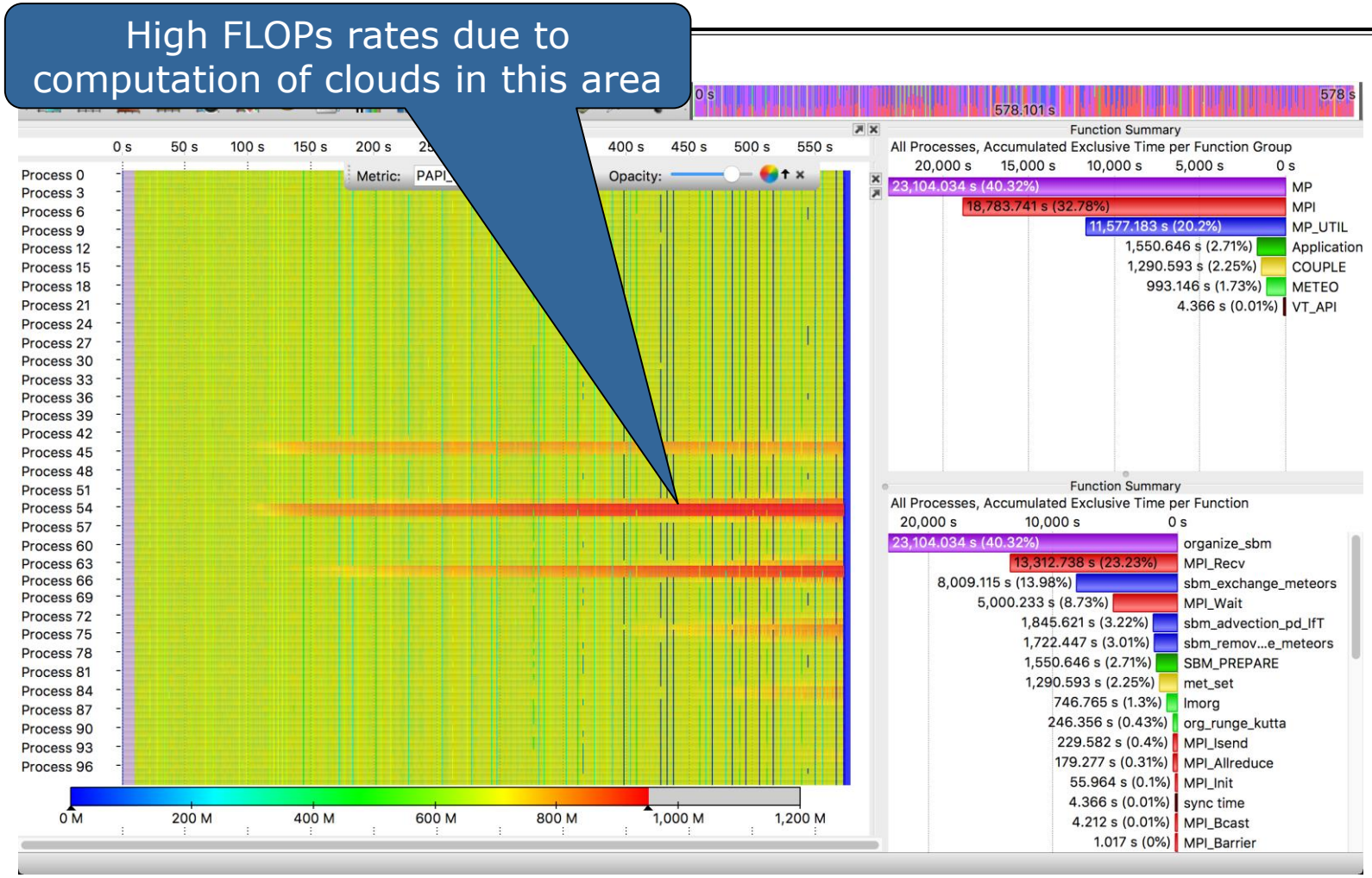
- Zoom into the first three iterations
- MP/MP_UTIL perform four sub-steps in one iteration
- Low MPI time share
- Everything is balanced and looks okay

COSMO-SPECS Original



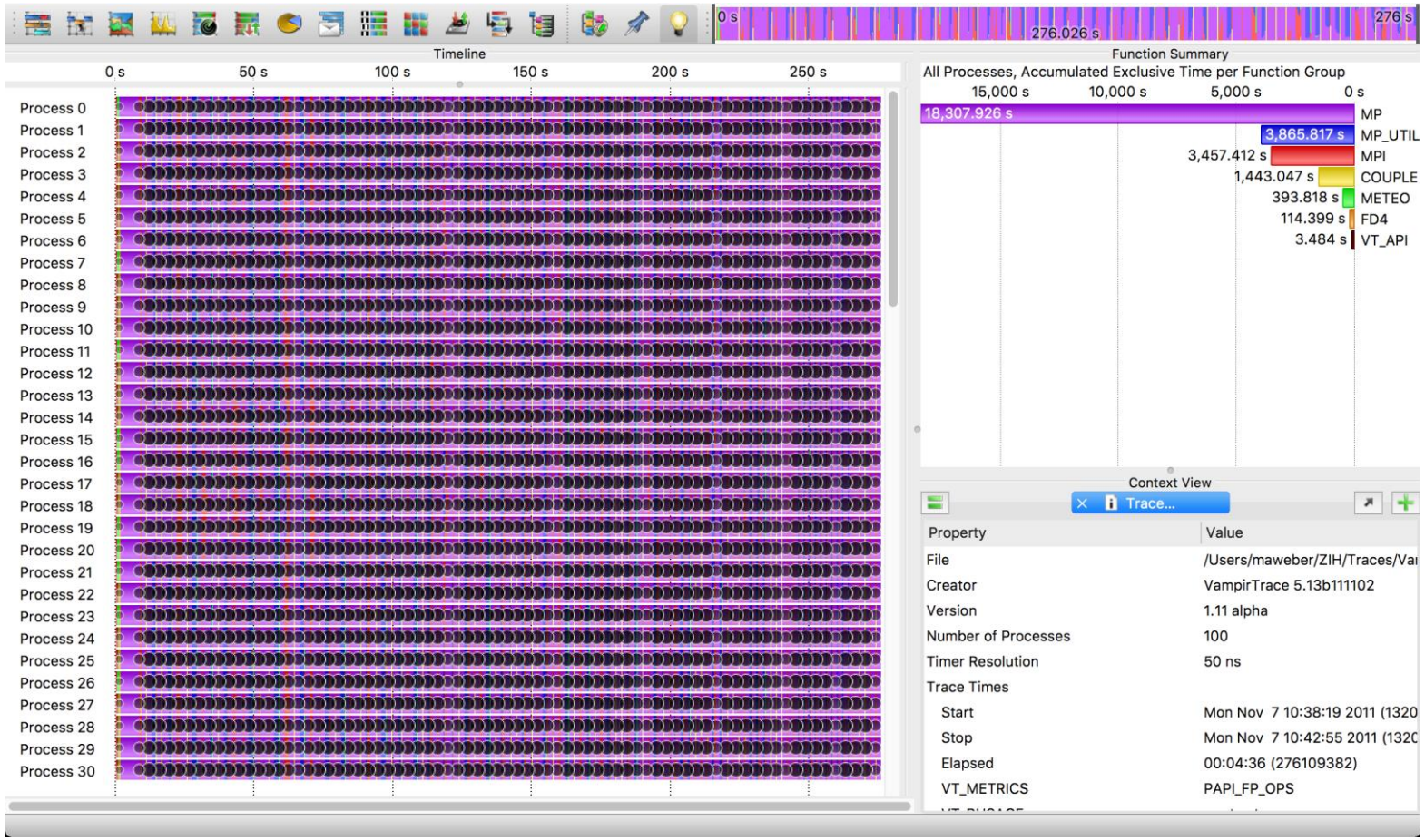
- Zoom into the last three iterations
- Very high MPI time share (>50%)
- Large load imbalance caused by MP functions around **`Process 54`** and **`Process 64`**

COSMO-SPECS Original



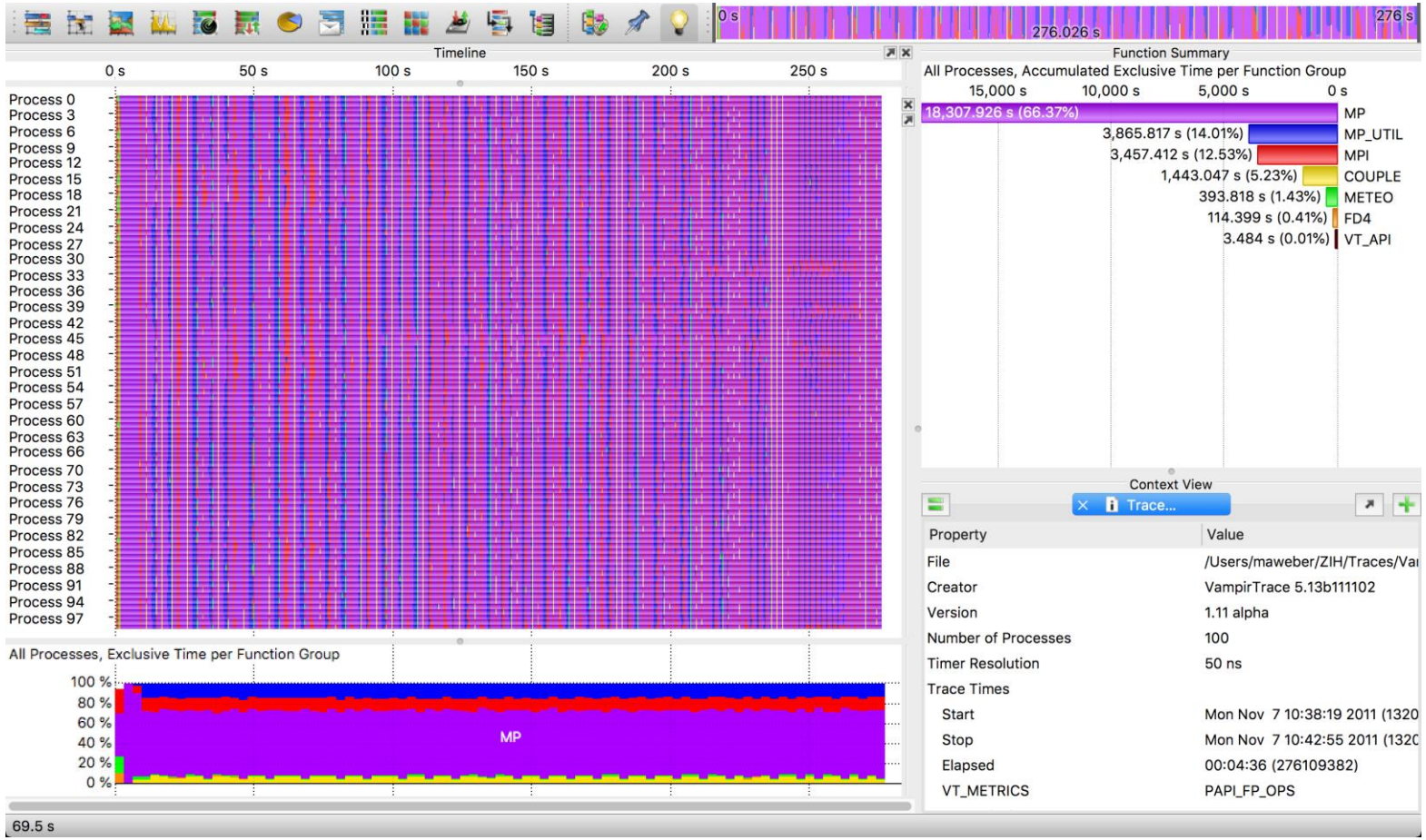
- **`PAPI_FP_OPS`** counter showing higher FLOPs rates on processes causing the imbalance
- Reason for imbalance: Static grid used for distribution of processes. Depending on the weather, expensive cloud computations (MP group) may be only necessary on some processes

COSMO-SPECS FD4



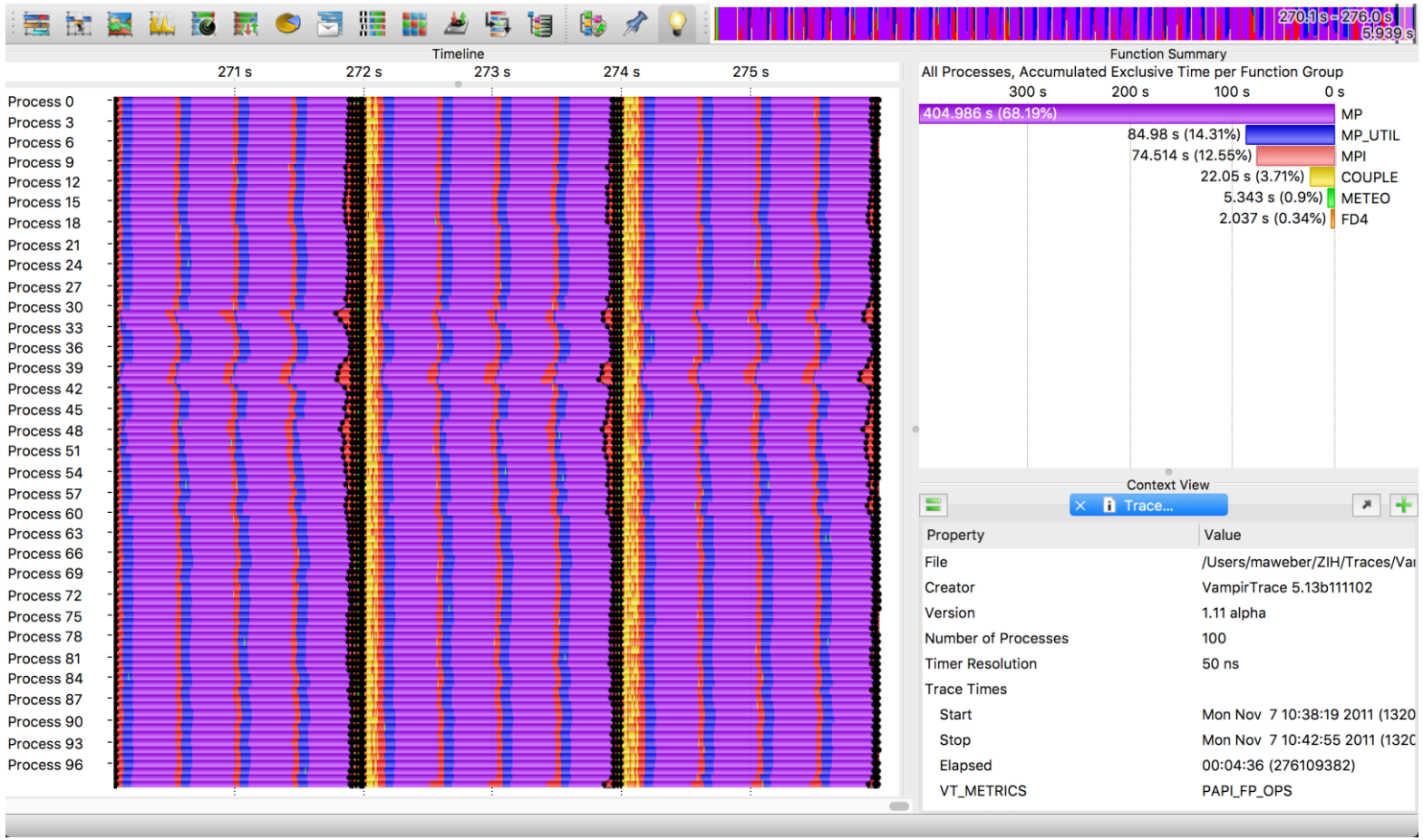
- Weather forecast code COSMO-SPECS
- Run with 100 processes
- COSMO: weather model (METEO group)
- SPECS: microphysics for accurate cloud calculation (MP and MP_UTIL group)
- Coupling of both models done in COUPLE group
- Dynamic load balancing (FD4 group)

COSMO-SPECS FD4



- Dynamic load balancing mitigates the balance problems of the original COSMO-SPECS version
- MPI time share is reduced to <13%
- MPI time share stays constant throughout the application runtime
- Runtime reduced by factor of 2.1, from initially 578s to 276s

COSMO-SPECS FD4



- Zoom into last three iterations
- FD4 balances MP load (precipitation processes in clouds) across all available processes

Summary and Conclusion

Summary

- Vampir & VampirServer
  - Interactive trace visualization and analysis
  - Intuitive browsing and zooming
  - Scalable to large trace data sizes (20 TiByte)
  - Scalable to high parallelism (200,000 processes)
- Vampir for Linux, Windows, and Mac OS X

http://www.vampir.eu

vampirsupport@zih.tu-dresden.de