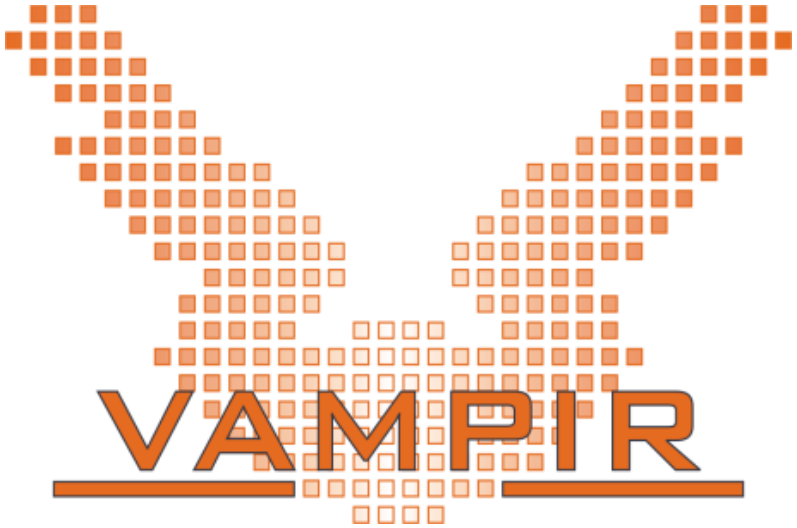
Introduction to Trace Vizualization with Vampir

Marc-André Hermanns
RWTH Aachen University

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 TeaLeaf

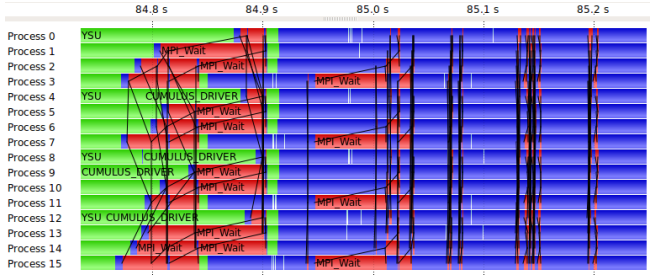


Event Trace Visualization with Vampir

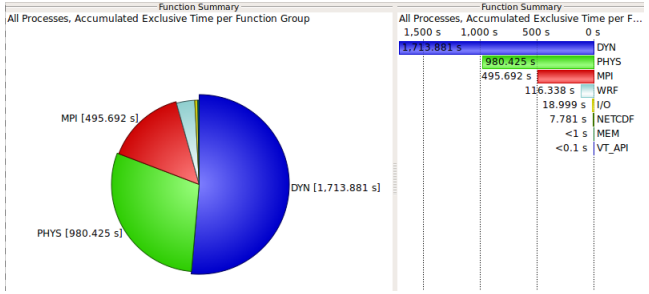
- Visualization of dynamic runtime behaviour at any level of detail along with statistics and performance metrics
- Alternative and supplement to automatic analysis

- **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

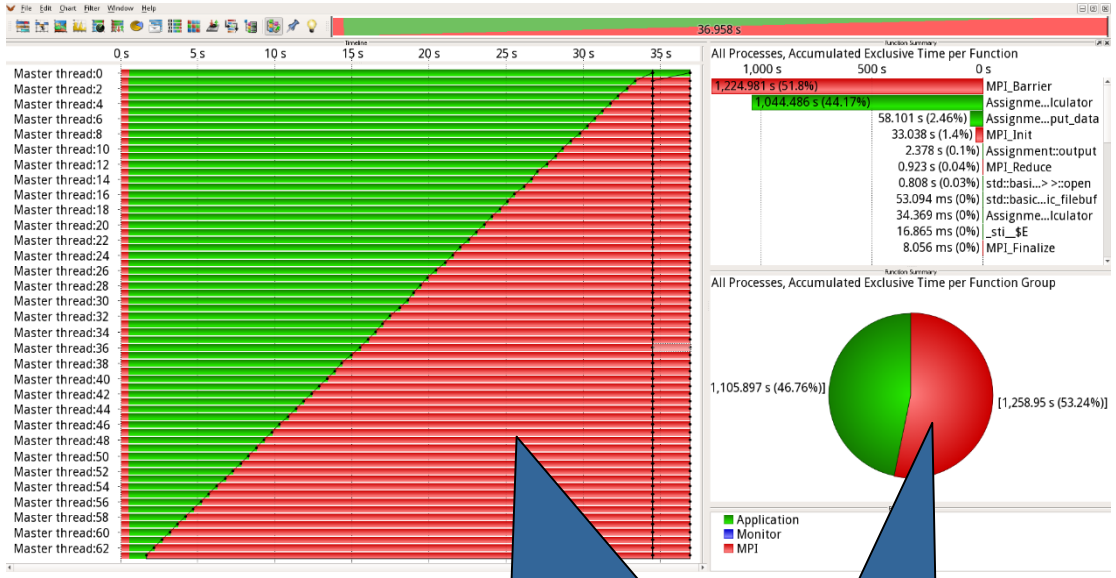


Event Trace Visualization with Vampir

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

Hands-on:
Visualizing and analyzing TeaLeaf

Help! Where is my trace file?

```
% ls <working_directory>/score-p/scorep_tea_leaf_8x12_trace
MANIFEST.md  scorep.cfg   scorep.log  scout.cubex  scout.err
scout.log    trace.cubex  traces      traces.def   traces.otf2
trace.stat


% ls <working_directory>/score-p/scorep_tea_leaf_8x12_trace/\
traces/
0.def             25769803782.def  42949672964.def
0.evt             25769803782.evt  42949672964.evt
12884901888.def   25769803783.def  42949672965.def
12884901888.evt   25769803783.evt  42949672965.evt
12884901889.def   2.def            42949672966.def
12884901889.evt   2.evt            42949672966.evt
[…]
25769803780.evt   42949672962.evt  8589934598.evt
25769803781.def   42949672963.def  8589934599.def
25769803781.evt   42949672963.evt  8589934599.evt
```

- Anchor file `traces.otf2` resides in experiment directory
- Local trace files reside in `traces/` sub-directory

Start Vampir

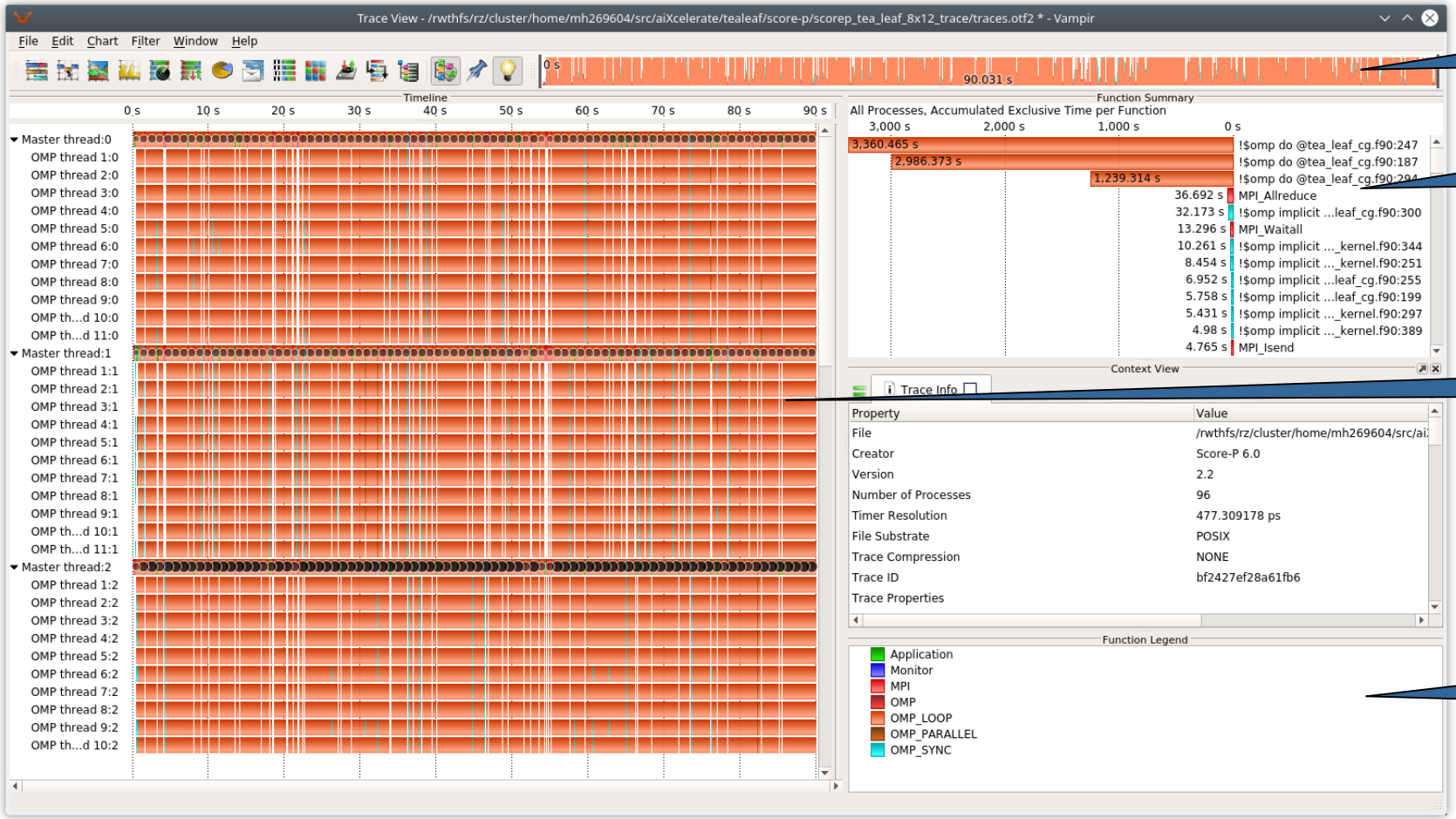
```
% module load DEV-TOOLS vampir




% vampir <working_directory>/score-p/scorep_tea_leaf_8x12_trace\
> /traces.ot2
```

- Load correct module to add local tool installations to $PATH (required for each shell session)
- Start Vampir on the current login-node
  (requires ssh X-forwarding)

Visualization of the TeaLeaf trace



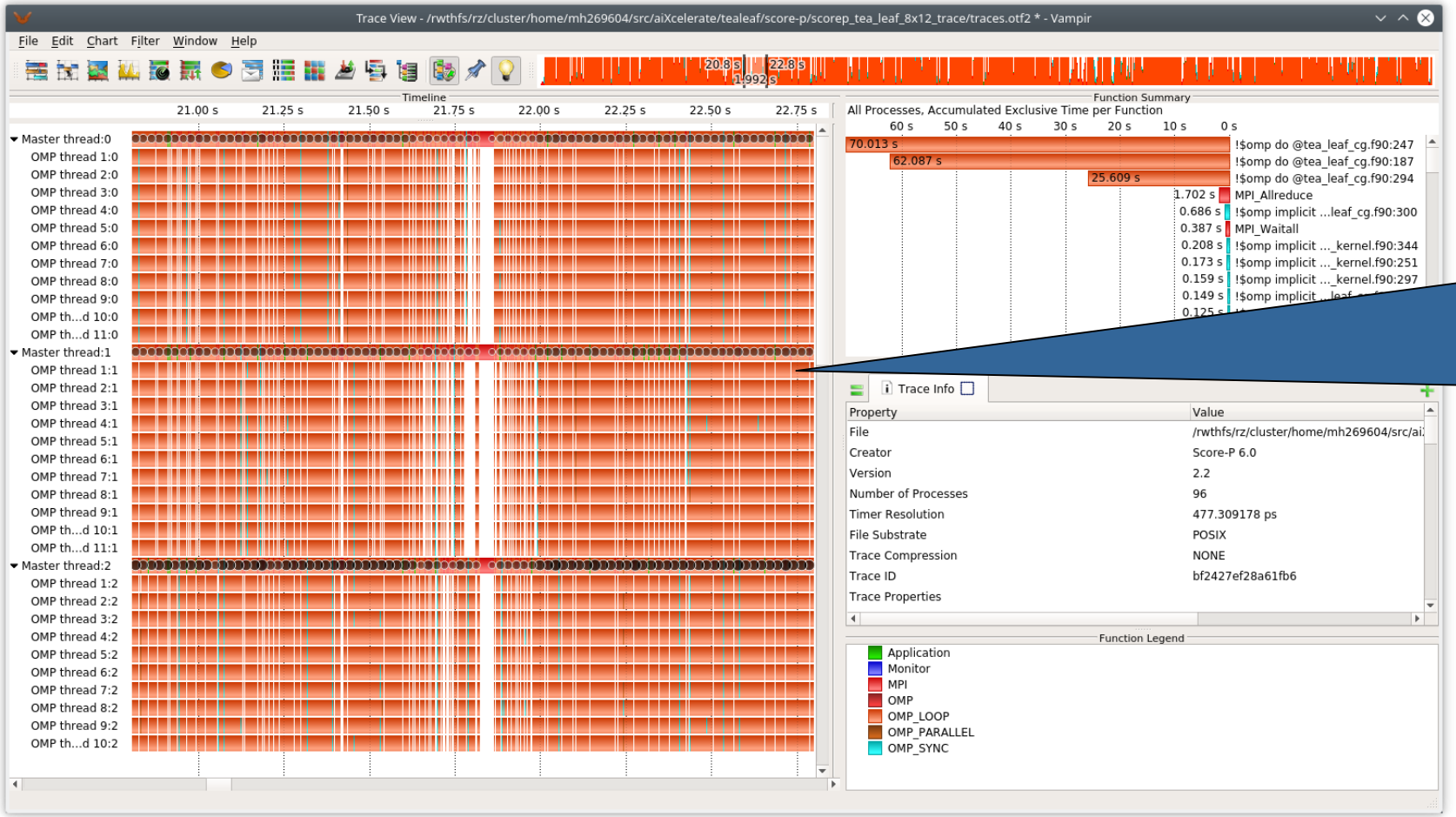
Navigation Toolbar

Function Summary

Master Timeline

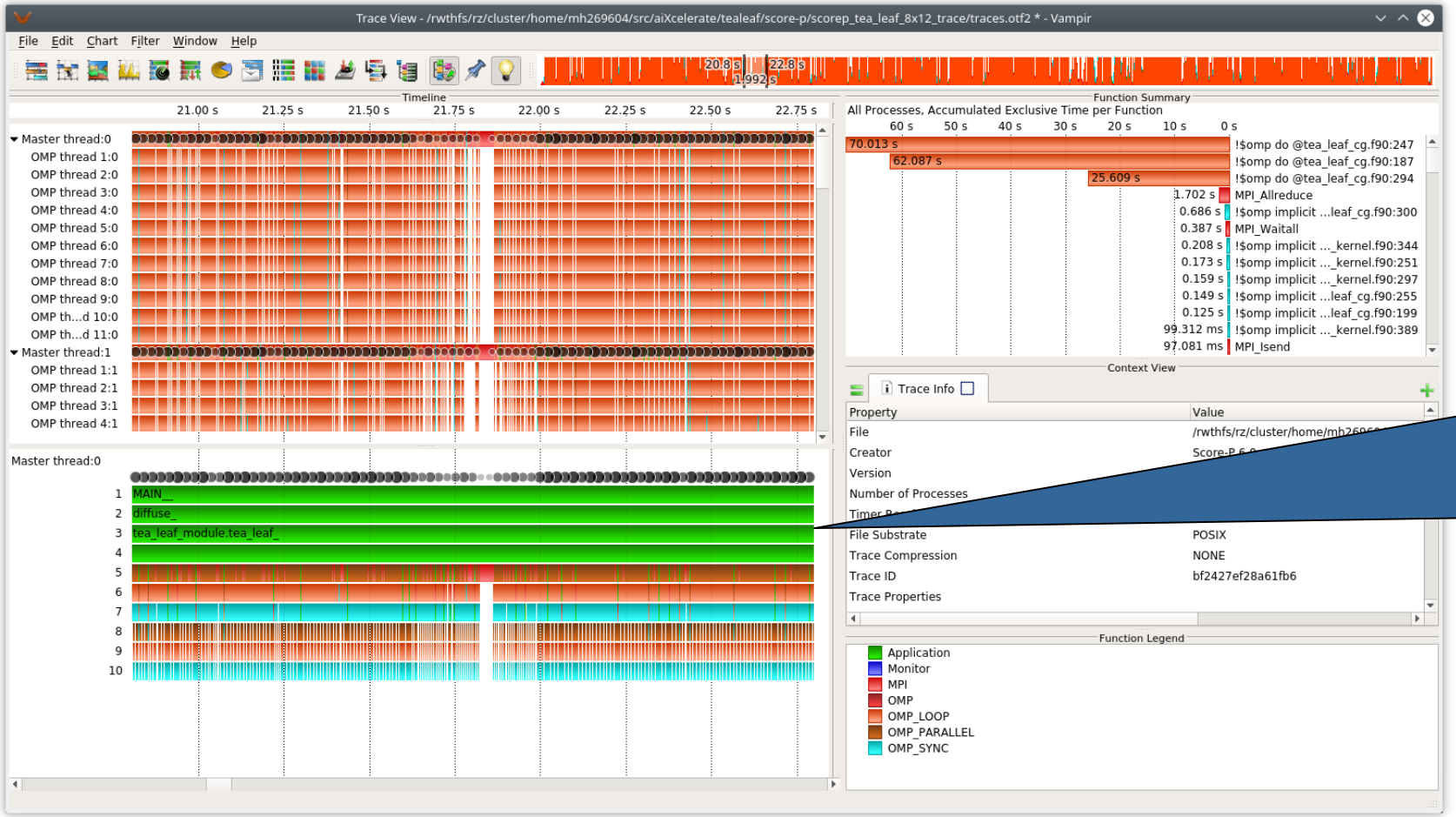
Function Legend

Visualization of the TeaLeaf trace
Master Timeline



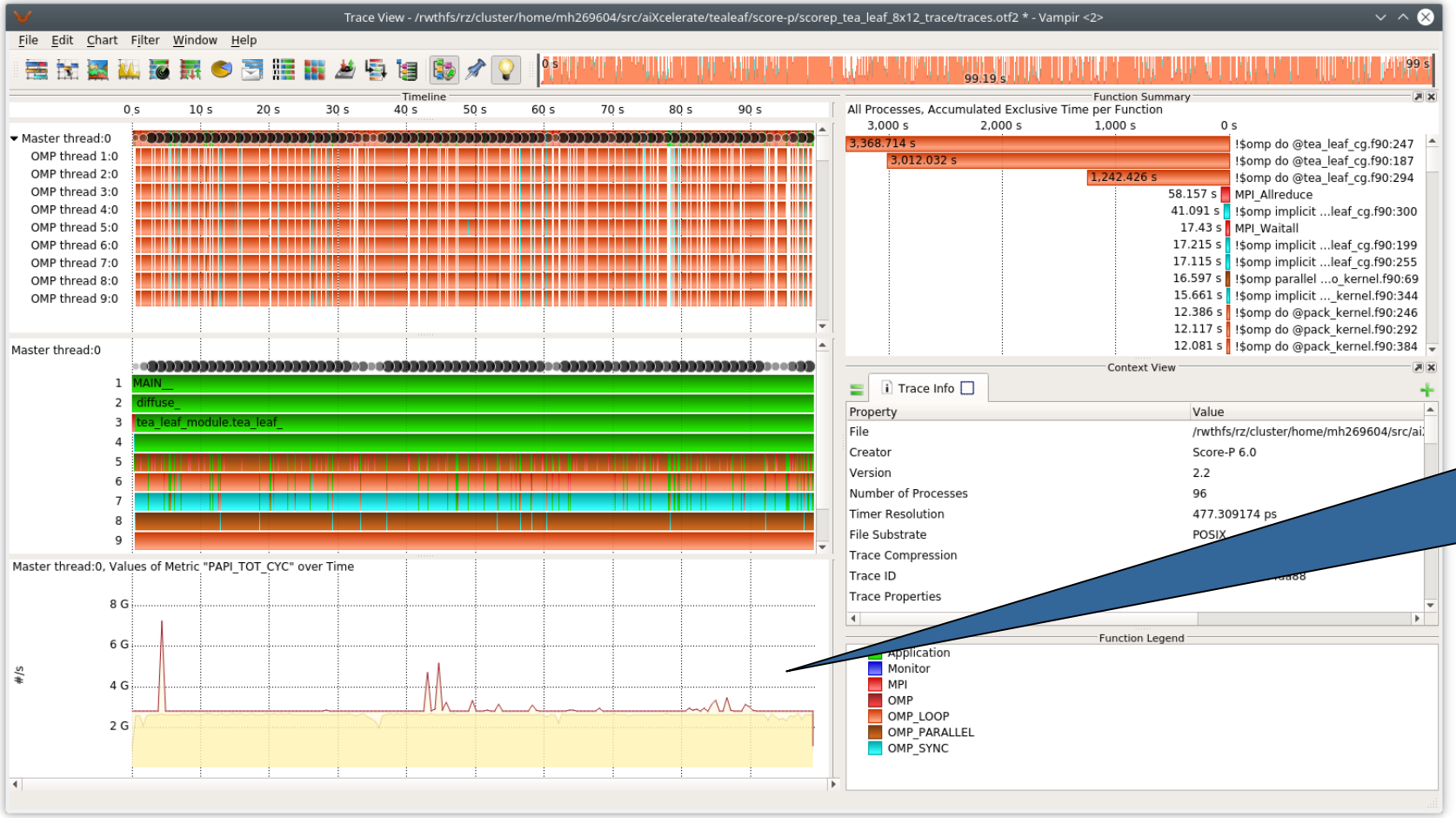
Detailed information about functions, communication and synchronization events for collection of processes.

Visualization of the TeaLeaf trace
Process Timeline



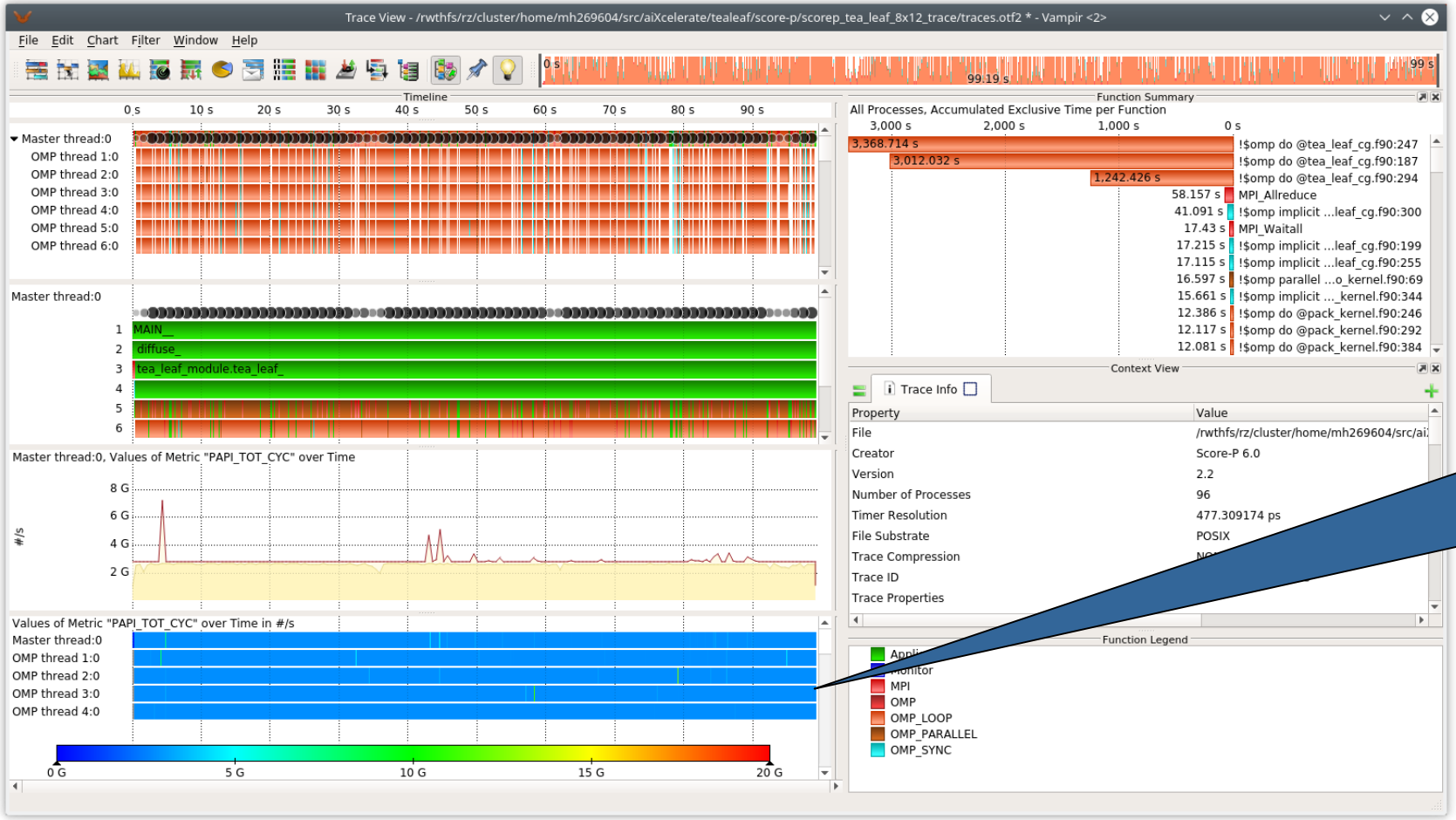
Detailed information about different levels of function calls in a stacked bar chart for an individual process.

Visualization of the TeaLeaf trace
Counter Data Timeline



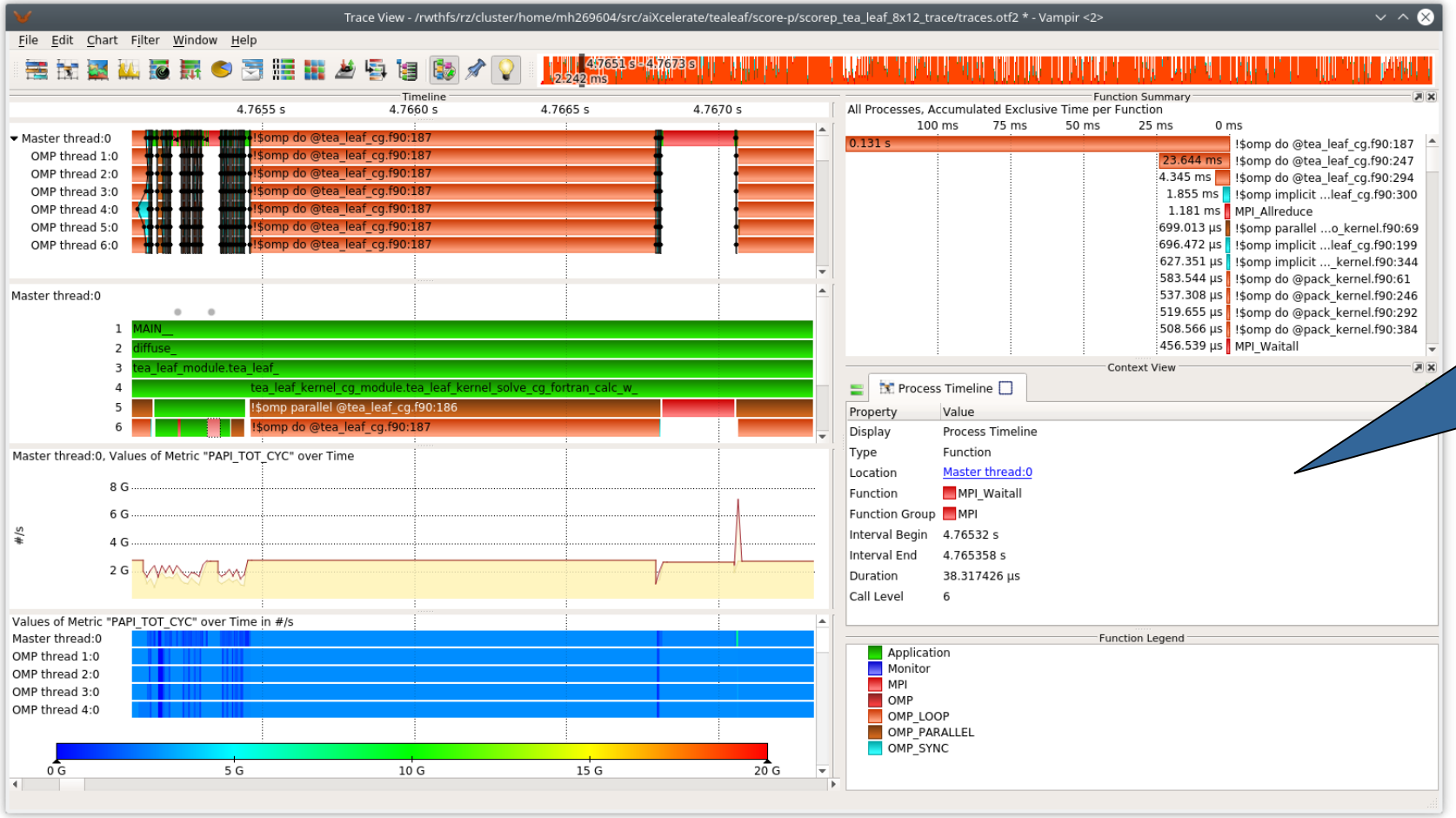
Detailed counter information over time for an individual process.

Visualization of the TeaLeaf trace
Performance Radar



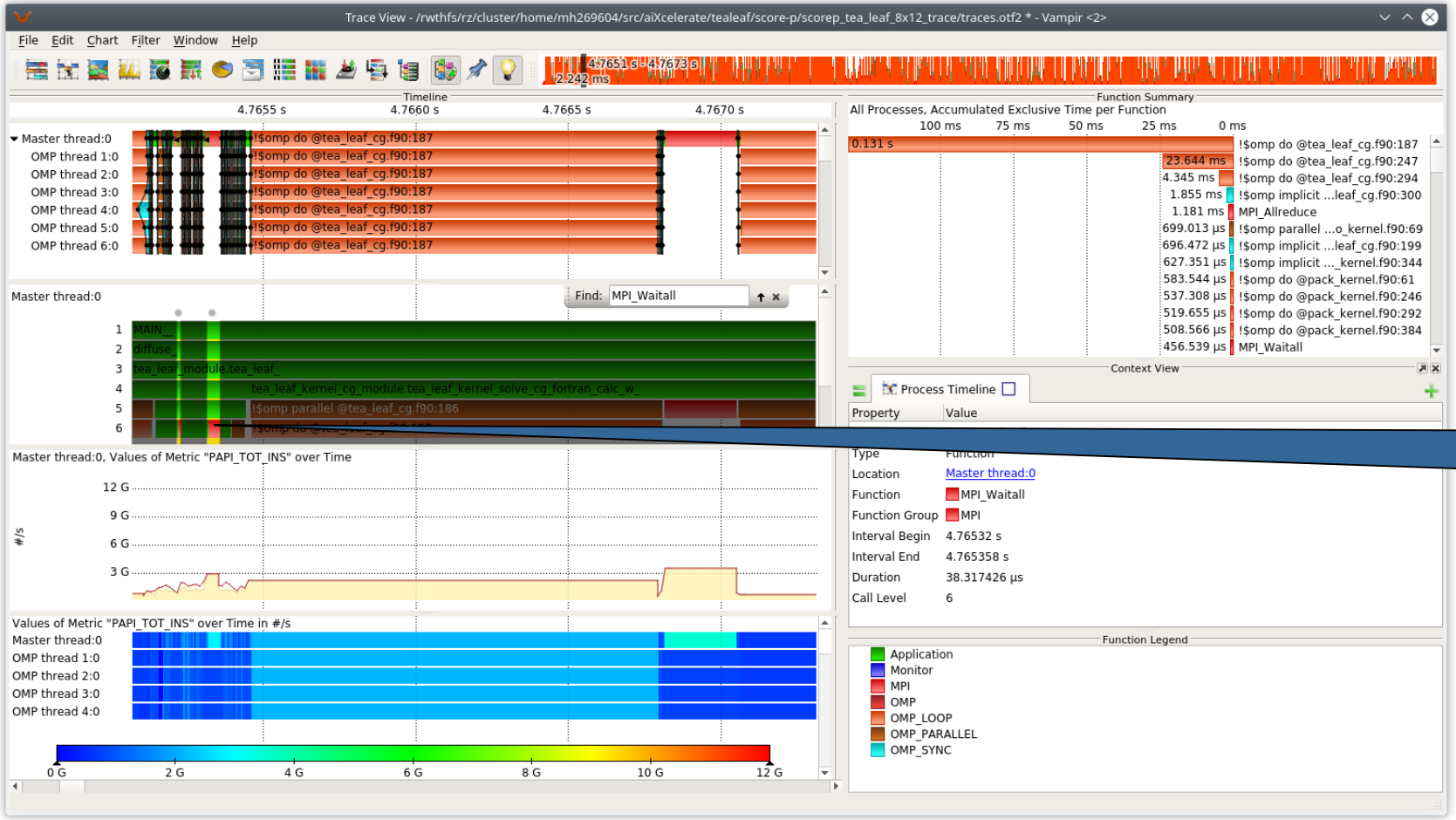
Detailed counter information over time for a collection of processes.

Visualization of the TeaLeaf trace
Zoom in: Communication Phase



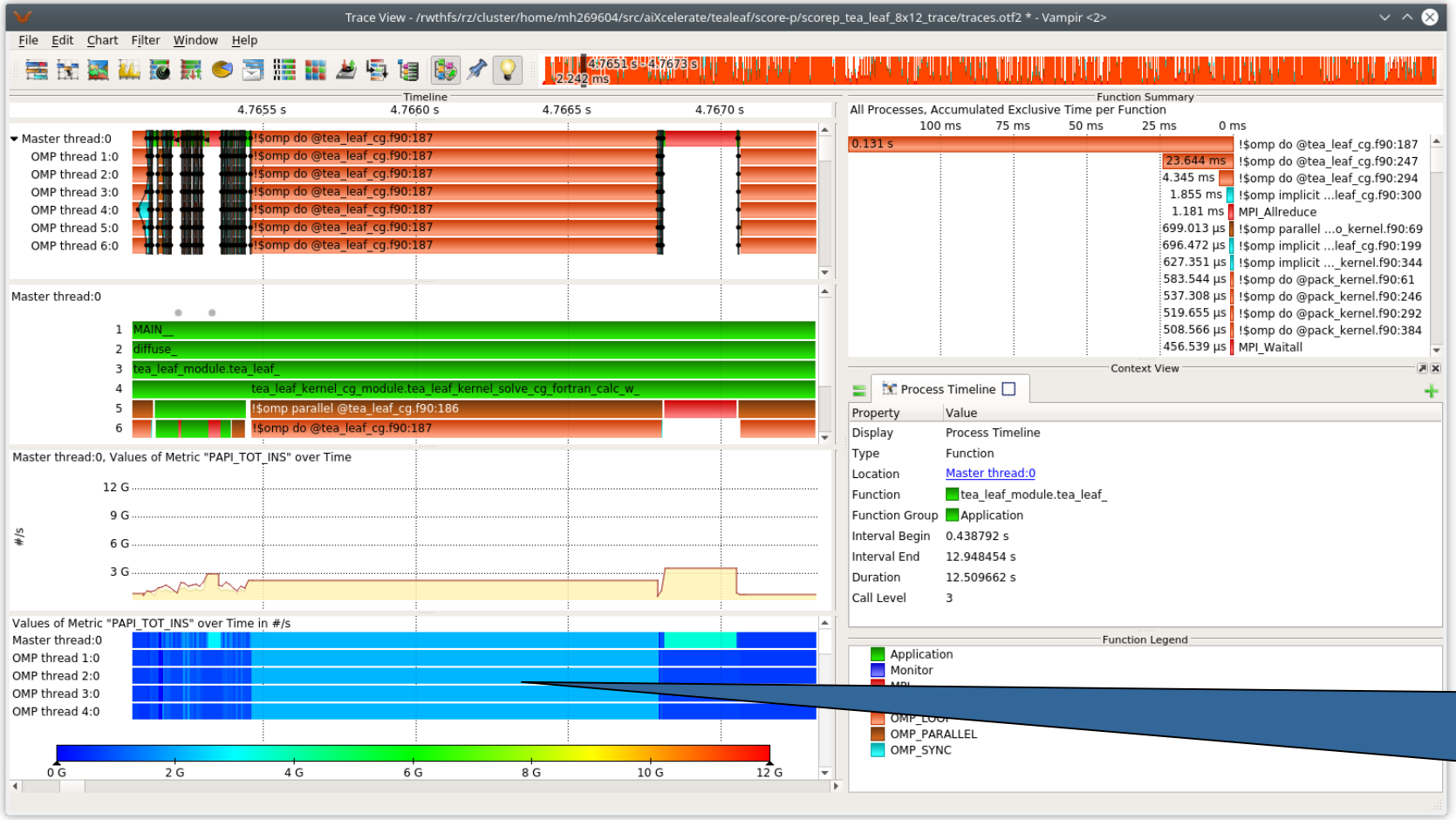
Context View: Detailed information about function “MPI_Waitall”.

Visualization of the TeaLeaf trace
Find Function



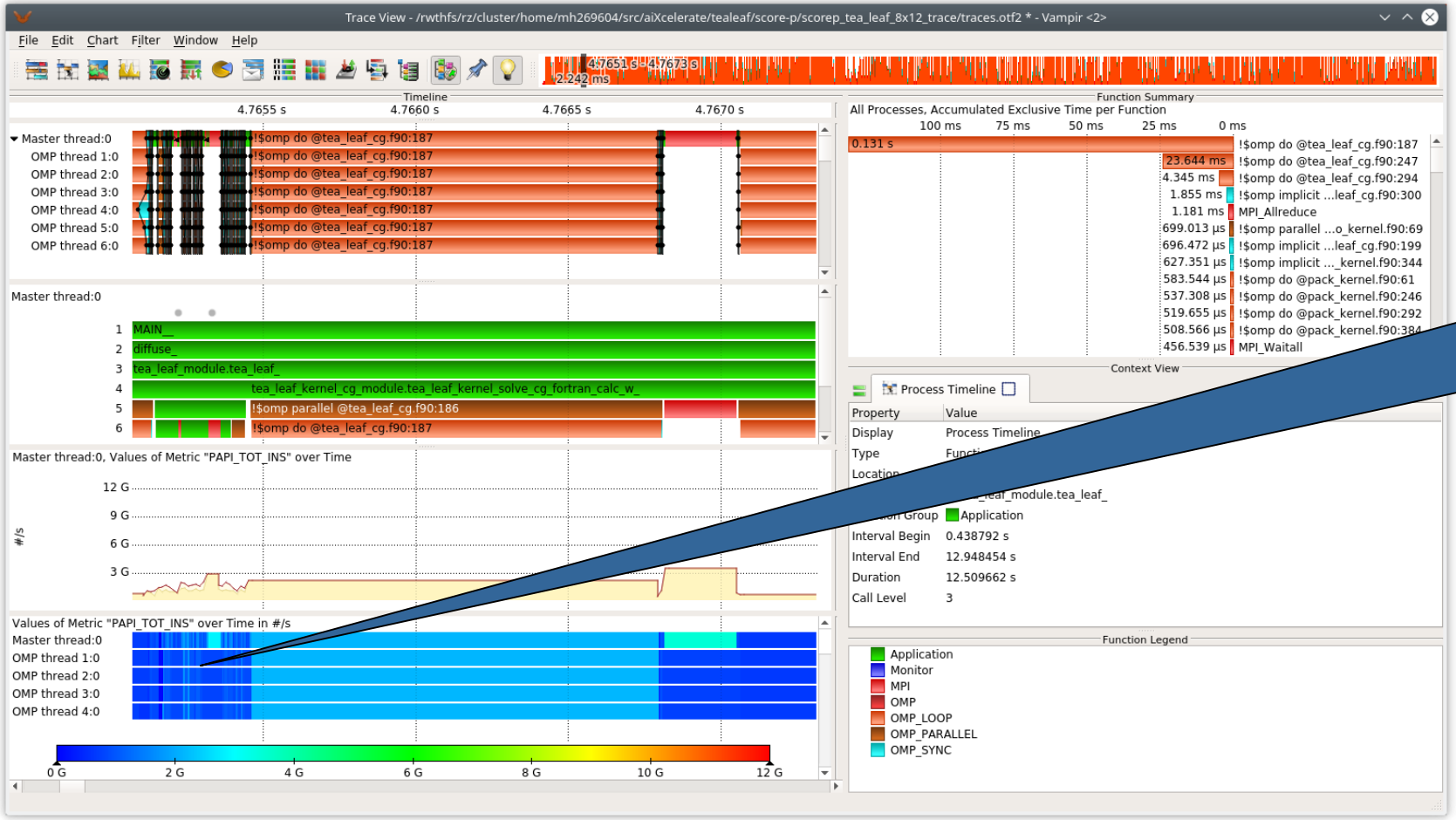
Identify “MPI_Waitall” in timeline

Visualization of the TeaLeaf trace
Zoom in: Communication Phase



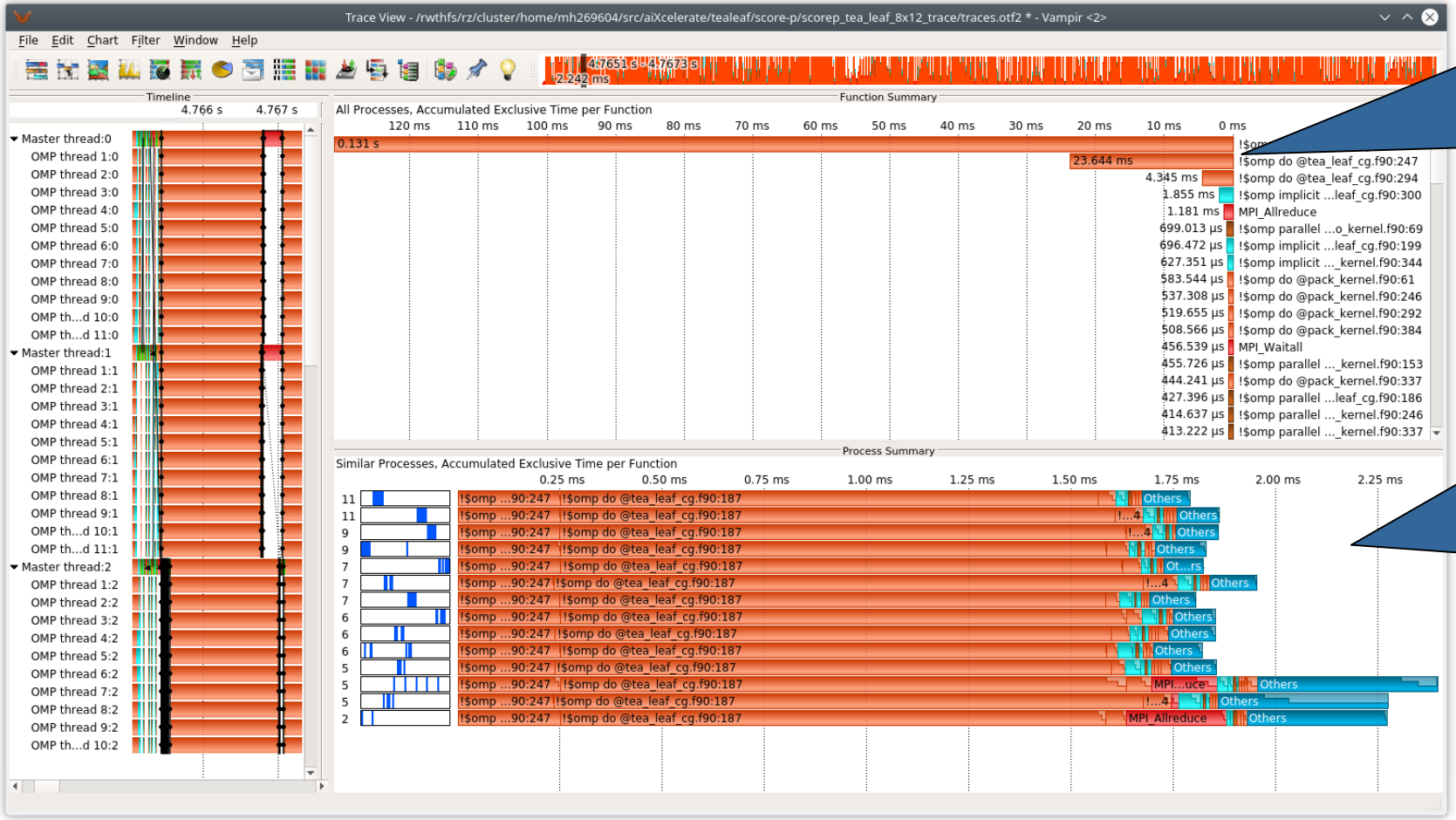
Computation phase results in higher instruction rate.

Visualization of the TeaLeaf trace
Zoom in: Communication Phase



MPI communication results in lower instruction rate.

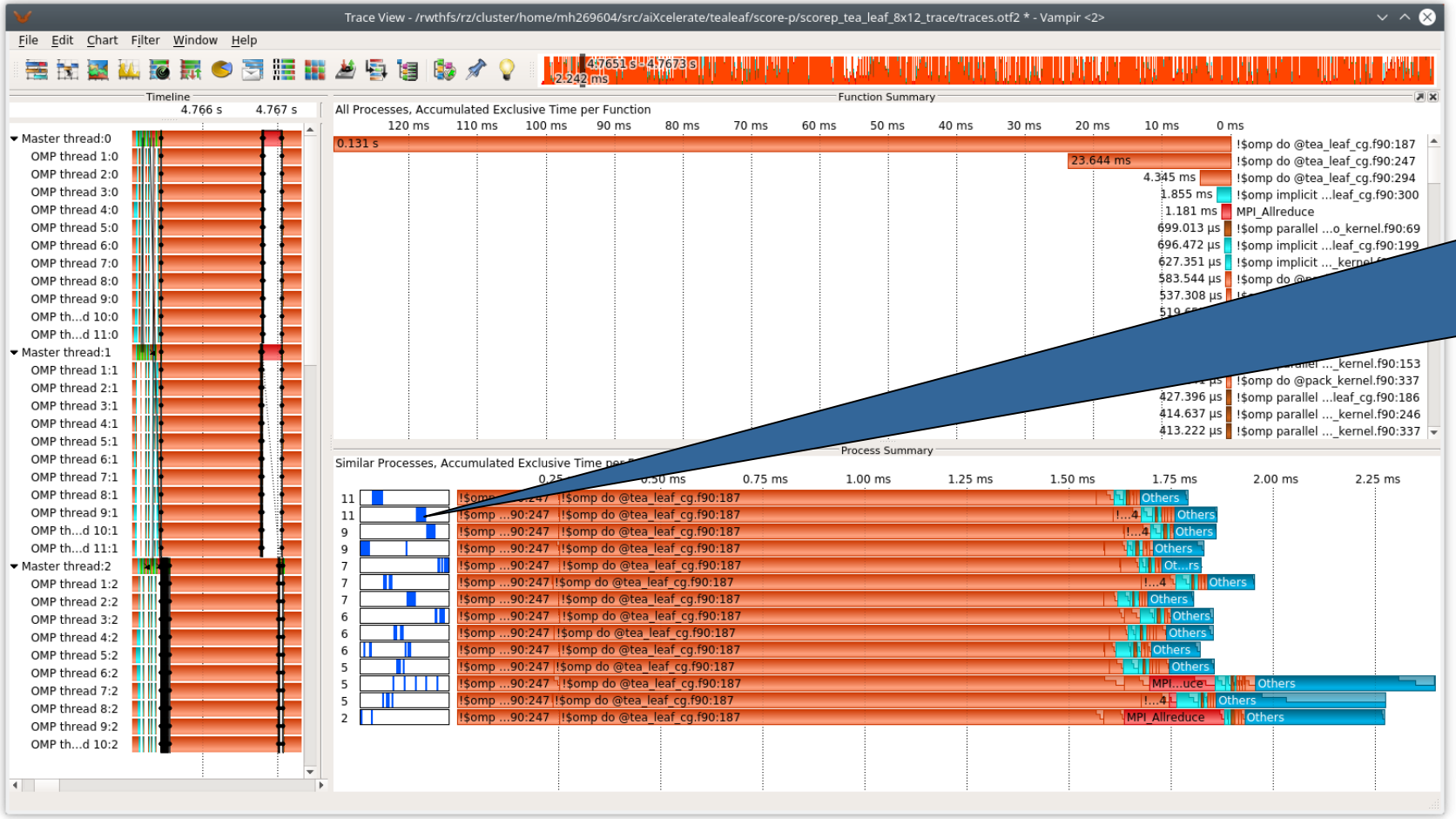
Visualization of the TeaLeaf trace
Process Summary



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 TeaLeaf trace
Process Summary



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

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