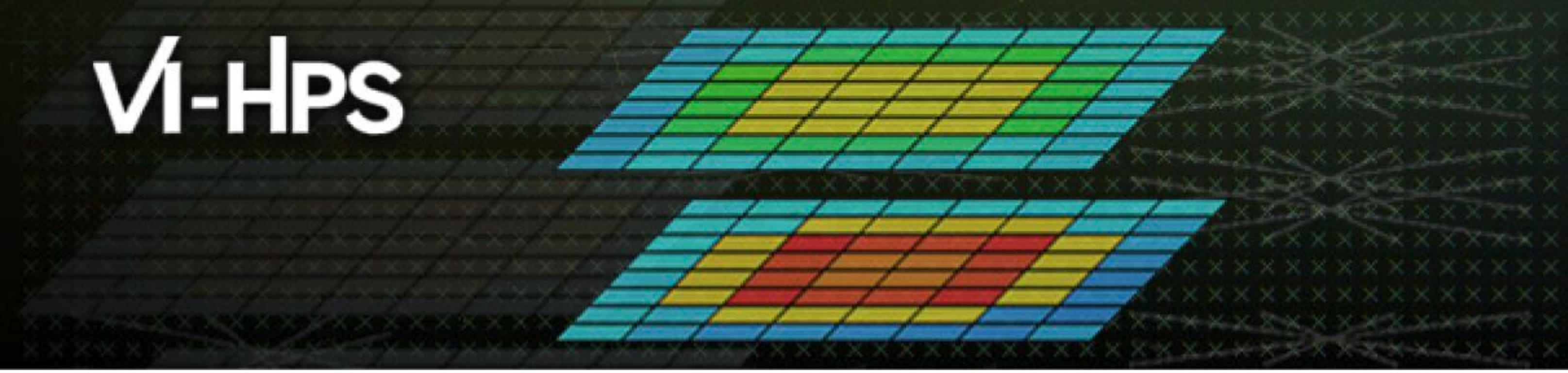


Performance Analysis with Vampir

Hartmut Mix
Technische Universität Dresden

VAMPIR

allinea · JÜLICH FORSCHUNGSZENTRUM · RWTH AACHEN UNIVERSITY · TECHNISCHE UNIVERSITÄT DRESDEN · UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES · Universität Stuttgart
BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación · Lawrence Livermore National Laboratory · TECHNISCHE UNIVERSITÄT DARMSTADT · TUM Technische Universität München · UNIVERSITY OF OREGON · THE UNIVERSITY of TENNESSEE

Event Trace Visualization with Vampir

- **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 Analysis Exercise**
  - Analysing Application Traces

Event Trace Visualization with Vampir

- **Vampir Tool Suite**

  (**V**isualization and **A**nalysis of **MPI R**esources)

  - Score-P – Instrumentation => **OTF2** traces
  - Vampir & VampirServer - Visualization

- **Post-mortem visual performance analysis**
  - Developed originally at Research Centre Jülich and since 1997 by ZHR/ZIH of **TU Dresden**
  - Commercial product: **www.vampir.eu**
  - Distribution and Support: **GWT-TUD GmbH**

VAMPIR

ZIH
Center for Information Services &
High Performance Computing

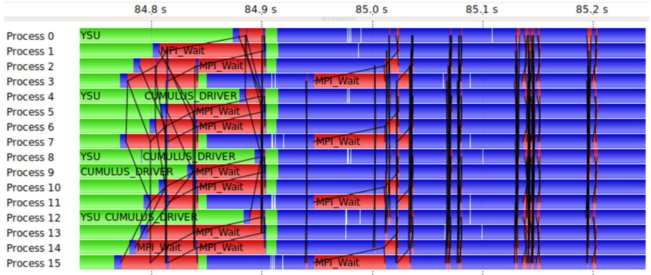
GWT forschung+innovation

Event Trace Visualization with Vampir

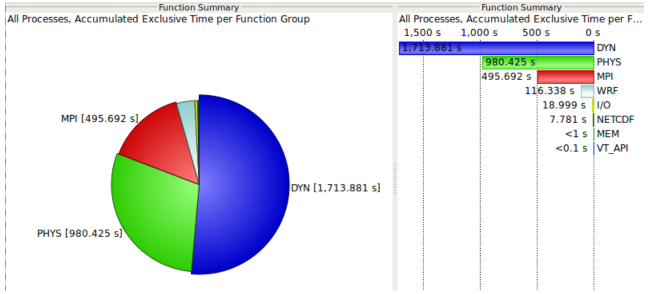
- Visualization of dynamic runtime behaviour at any level of detail along with statistics and performance metrics
- Alternative and supplement to automatic analysis

- **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



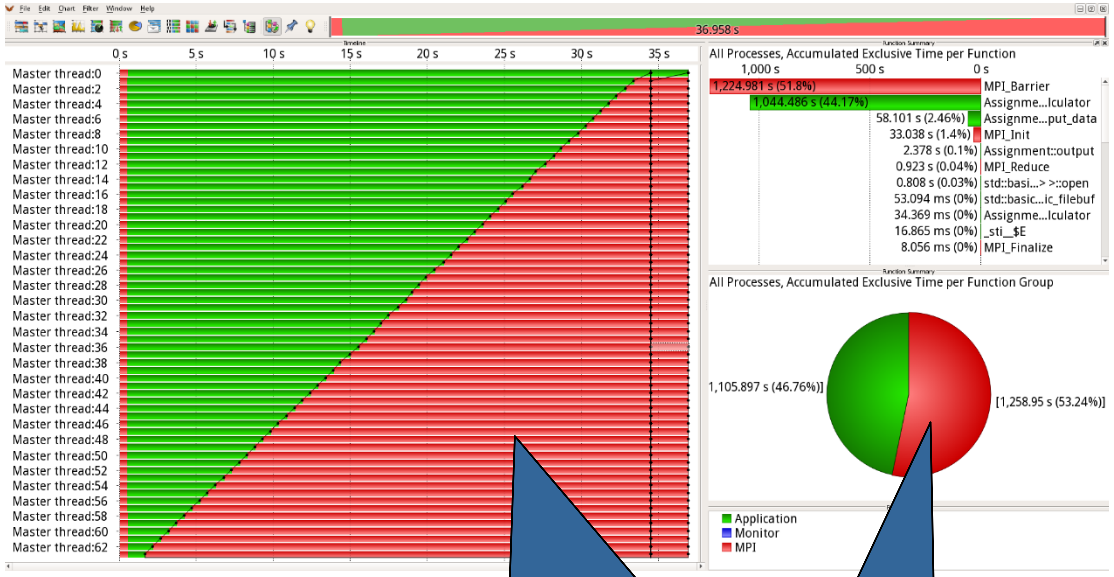
- **Summary charts**
  - Quantitative results for the currently selected time interval



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



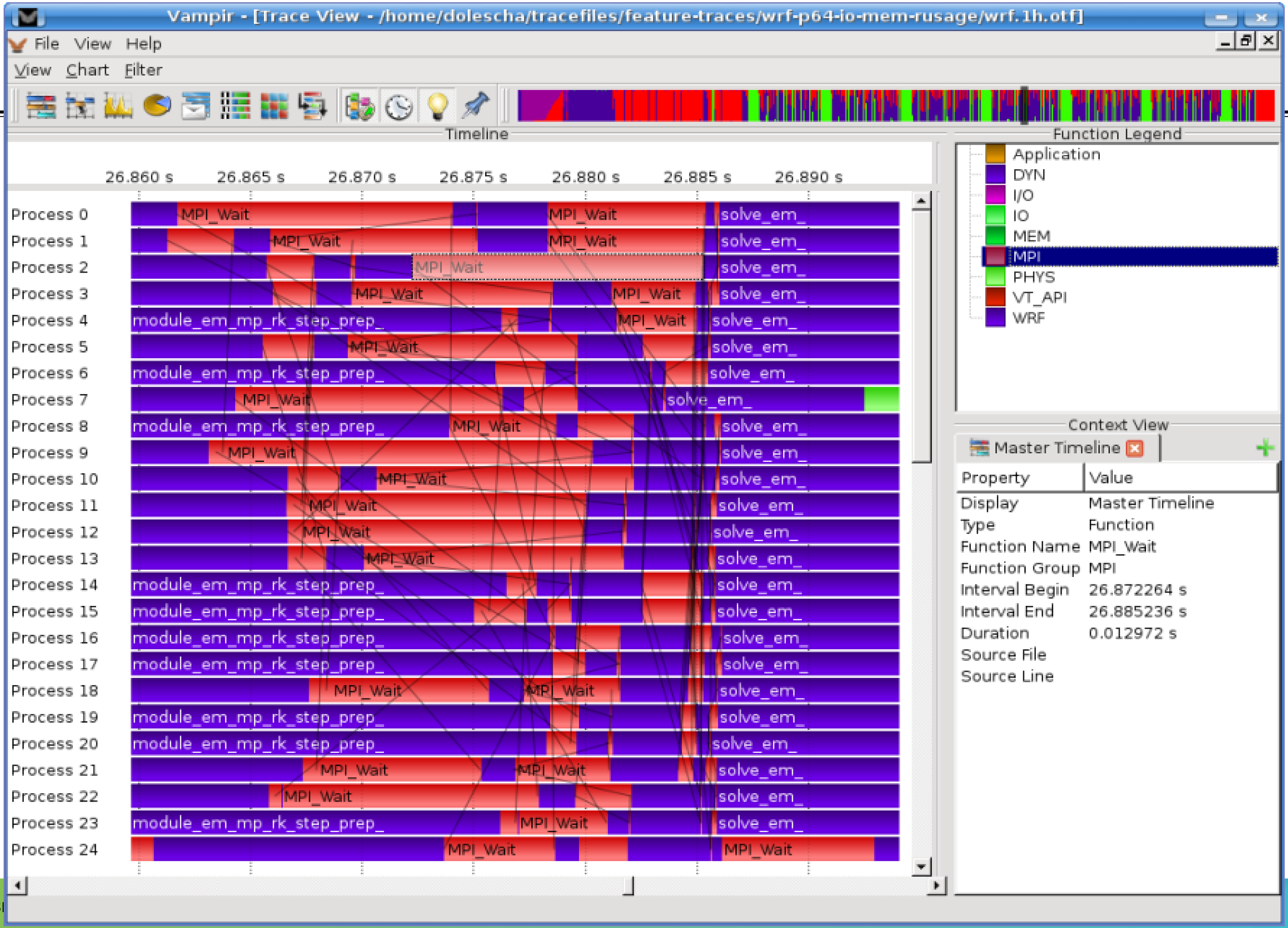
All Processes, Accumulated Exclusive Time per Function

1,224.981 s (51.8%)	MPI_Barrier
1,044.486 s (44.17%)	Assignme...lculator
58.101 s (2.46%)	Assignme...put_data
33.038 s (1.4%)	MPI_Init
2.378 s (0.1%)	Assignment::output
0.923 s (0.04%)	MPI_Reduce
0.808 s (0.03%)	std::basi...> >::open
53.094 ms (0%)	std::basic...ic_filebuf
34.369 ms (0%)	Assignme...lculator
16.865 ms (0%)	_sti__$E
8.056 ms (0%)	MPI_Finalize

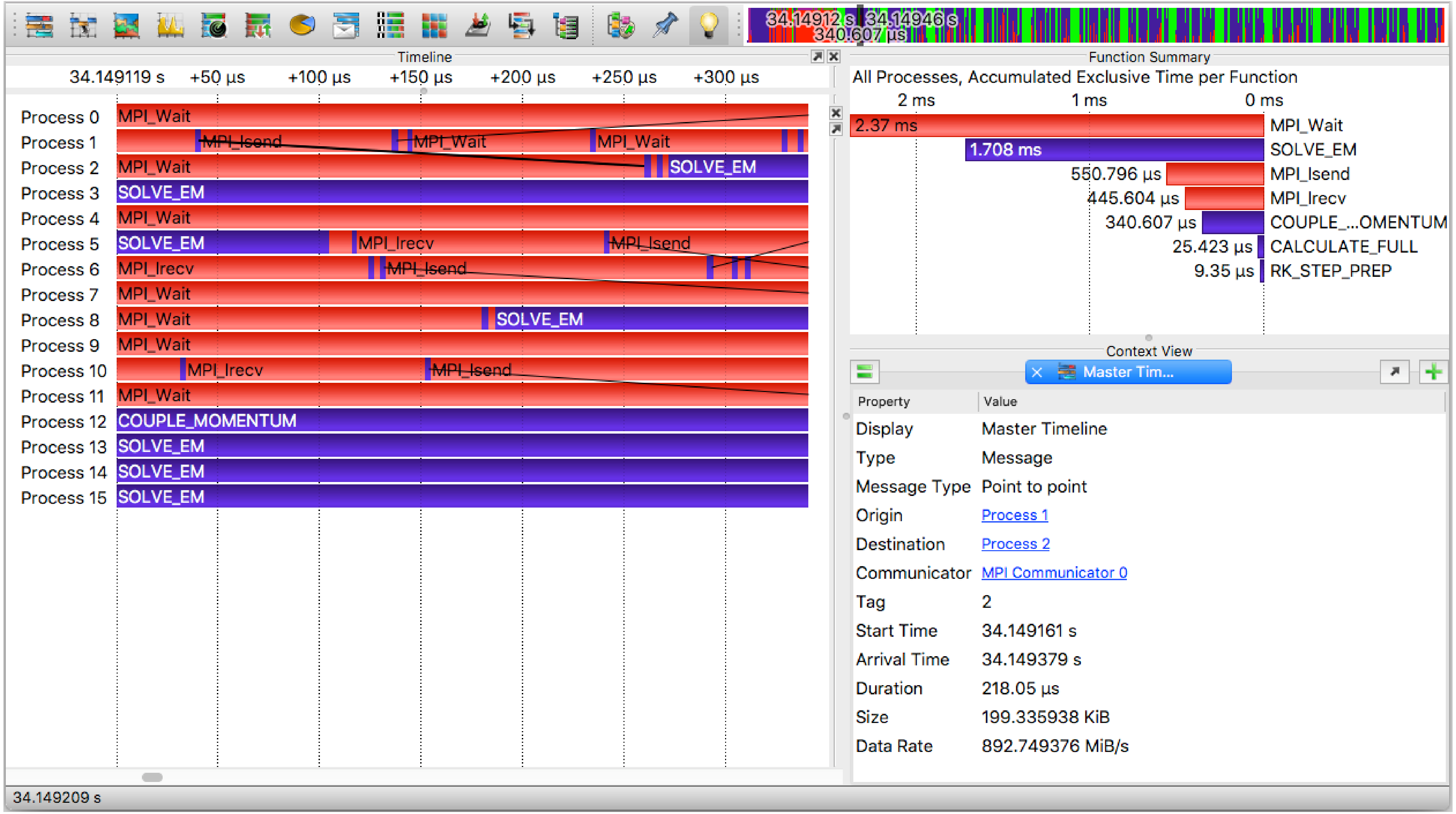
All Processes, Accumulated Exclusive Time per Function Group

Large imbalance instantly visible

More than 50% application time wasted!



Property	Value
Display	Master Timeline
Type	Function
Function Name	MPI_Wait
Function Group	MPI
Interval Begin	26.872264 s
Interval End	26.885236 s
Duration	0.012972 s
Source File	
Source Line	



Property	Value
Display	Master Timeline
Type	Message
Message Type	Point to point
Origin	Process 1
Destination	Process 2
Communicator	MPI Communicator 0
Tag	2
Start Time	34.149161 s
Arrival Time	34.149379 s
Duration	218.05 µs
Size	199.335938 KiB
Data Rate	892.749376 MiB/s

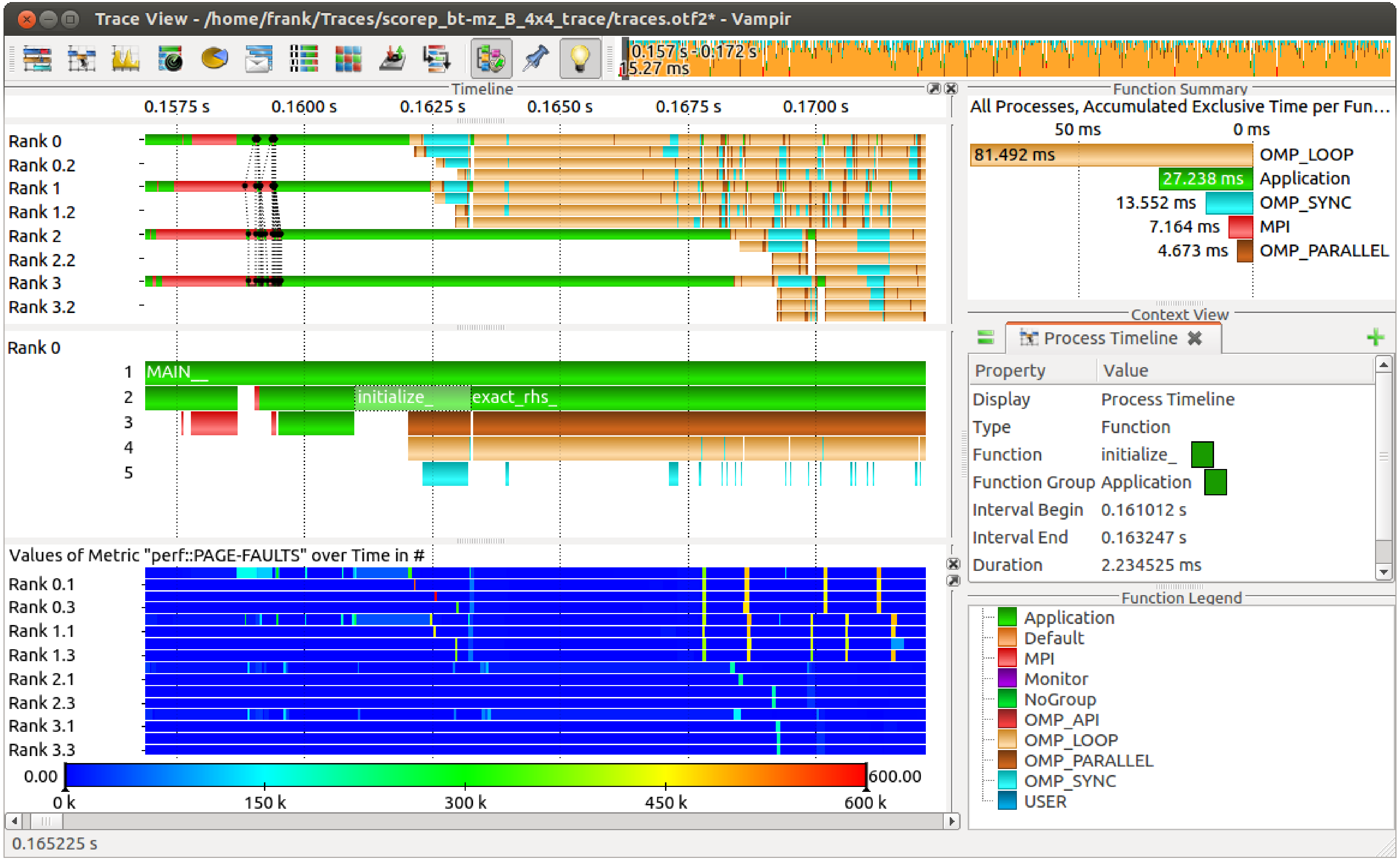
Main Performance Charts of Vampir

Timeline Charts

Master Timeline	➡	*all threads' activities*
Process Timeline	➡	*single thread's activities*
Summary Timeline	➡	*all threads' function call statistics*
Counter Data Timeline	➡	*single threads' performance metrics*
Performance Radar	➡	*all threads' performance metrics*
Shared Resource Timeline	➡	*all threads' I/O activities*

Summary Charts

- Function Summary
- Message Summary
- Process Summary
- Communication Matrix View
- Collective Operation Summary
- I/O Summary
- Call Tree
- System Tree



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

All Processes, Accumulated Exclusive Time per Fun...

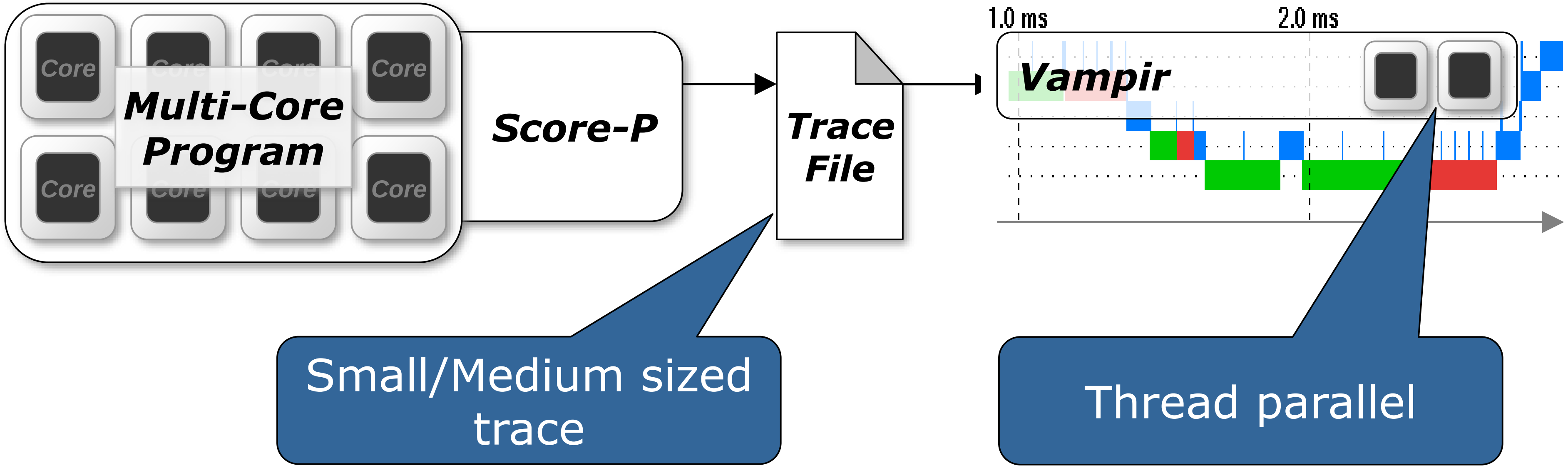
81.492 ms	OMP_LOOP
27.238 ms	Application
13.552 ms	OMP_SYNC
7.164 ms	MPI
4.673 ms	OMP_PARALLEL

Property	Value
Display	Process Timeline
Type	Function
Function	initialize_ 🟩
Function Group	Application 🟩
Interval Begin	0.161012 s
Interval End	0.163247 s
Duration	2.234525 ms

- 🟩 Application
- 🟧 Default
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

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

```
% vampir
```



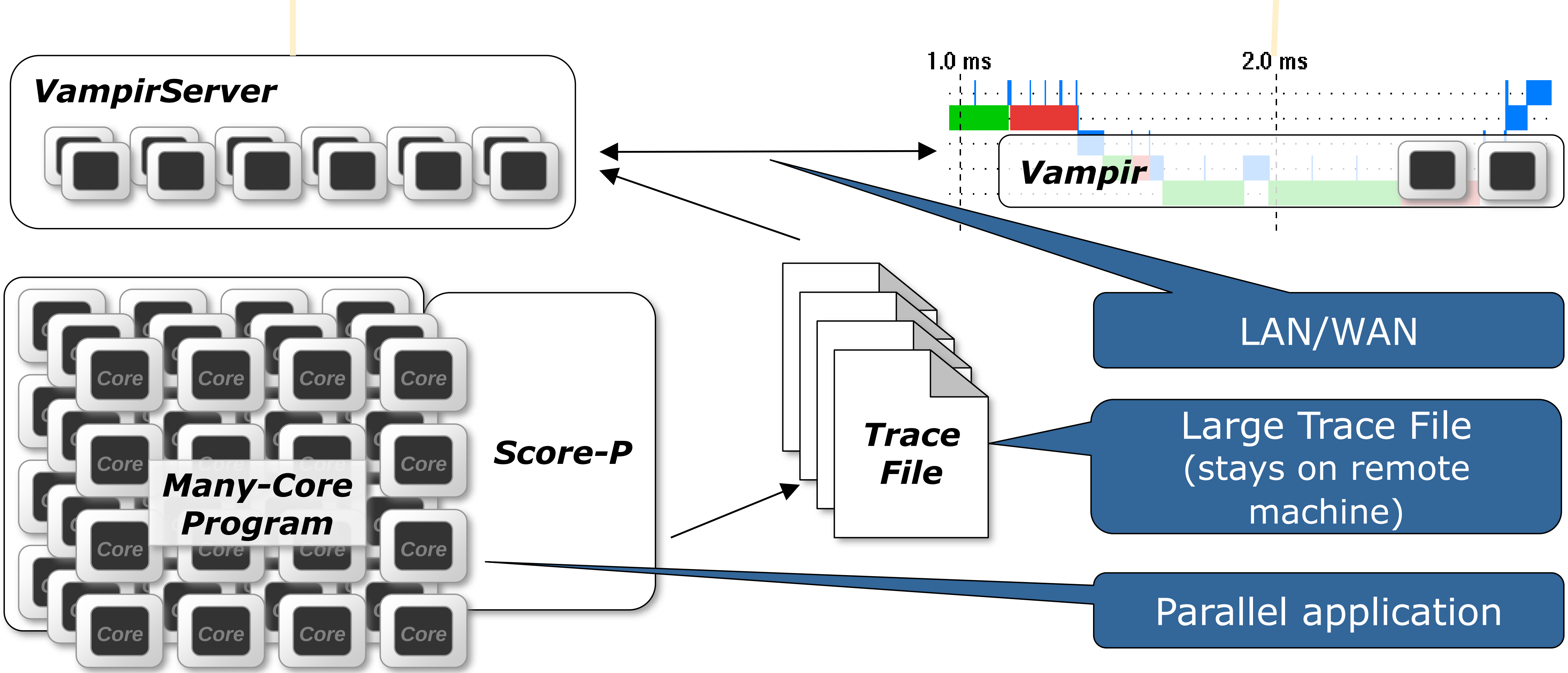
Small/Medium sized trace

Thread parallel

Visualization Modes (2)
On local machine with remote VampirServer

```
% vampirserver start
```

```
% vampir
```



LAN/WAN

Large Trace File (stays on remote machine)

Parallel application

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

Help! Where is my trace file?

```
% ls $WORK/NPB3.3-MZ-MPI/bin.scorep/scorep_bt-mz_C_16x4_trace
MANIFEST.md  scorep.cfg  scorep.log ...
traces/   traces.def  traces.otf2




% ls $TW35/scorep-examples/scorep_bt-mz_C_16x8_trace
MANIFEST.md  scorep.cfg  scorep.log ...
traces/   traces.def  traces.otf2

% ls /tmp/TW35/vampir/scorep_bt-mz_C_16x8_trace
MANIFEST.md  scorep.cfg  scorep.log ...
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

Start Vampir

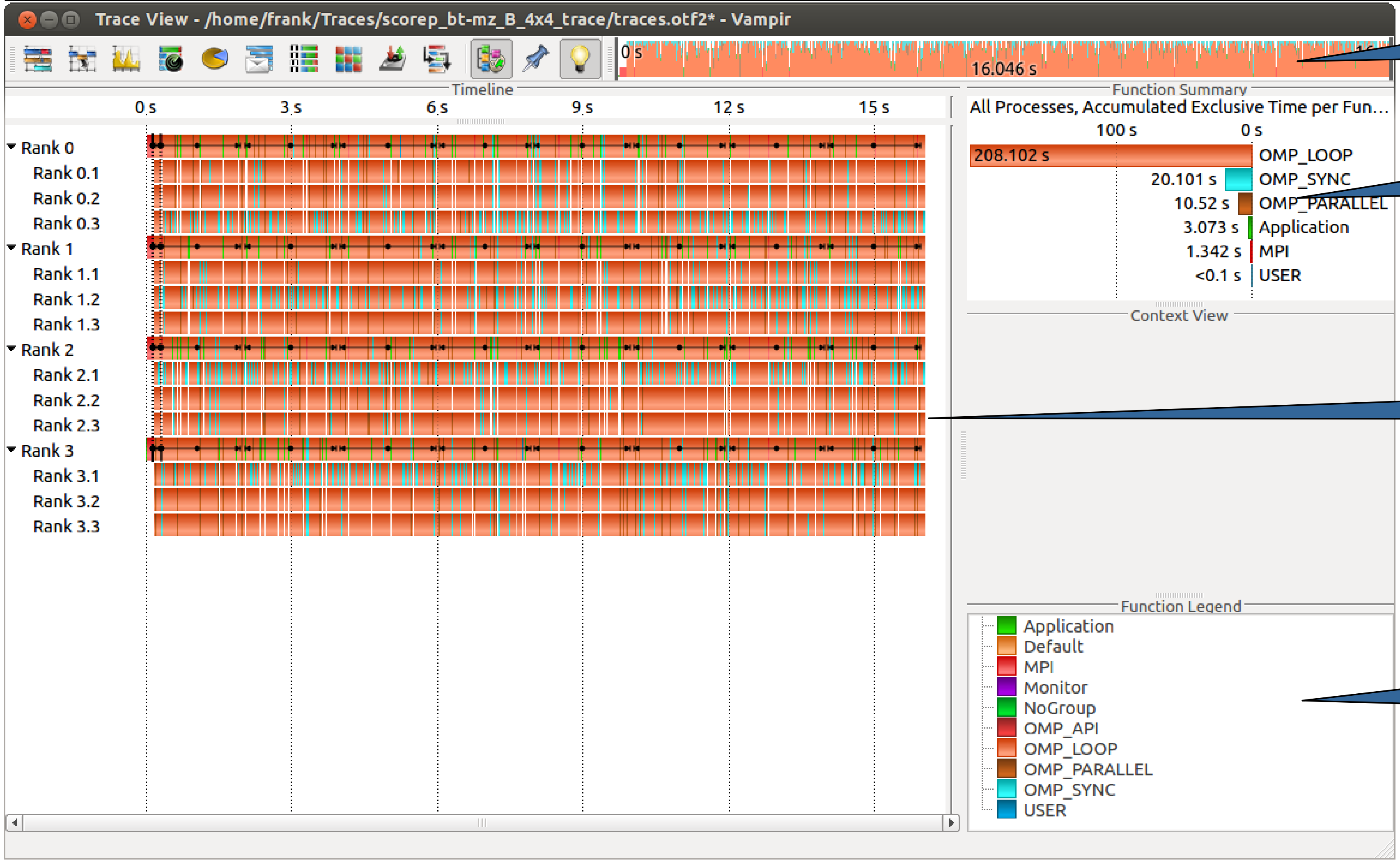
```
% qsub -I -X -q R_tw -l select=1:mpiprocs=12 -l walltime=0:30:00



% module load vampir/9.9.0





% vampir <tracefile>

% vampir $TW35/scorep-examples/\
        scorep_bt-mz_C_16x4_trace/traces.otf2
```

- Work on Login2 or request an Interactive Batch Shell

- Load correct module to add local tool installations to $PATH (required for each shell session)

- Start Vampir (requires ssh X-forwarding) and Open (local) trace file

Visualization of the NPB-MZ-MPI / BT trace



All Processes, Accumulated Exclusive Time per Fun…

100 s / 0 s	
208.102 s	OMP_LOOP
20.101 s	OMP_SYNC
10.52 s	OMP_PARALLEL
3.073 s	Application
1.342 s	MPI
<0.1 s	USER

Function Legend:

- 🟩 Application
- 🟧 Default
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

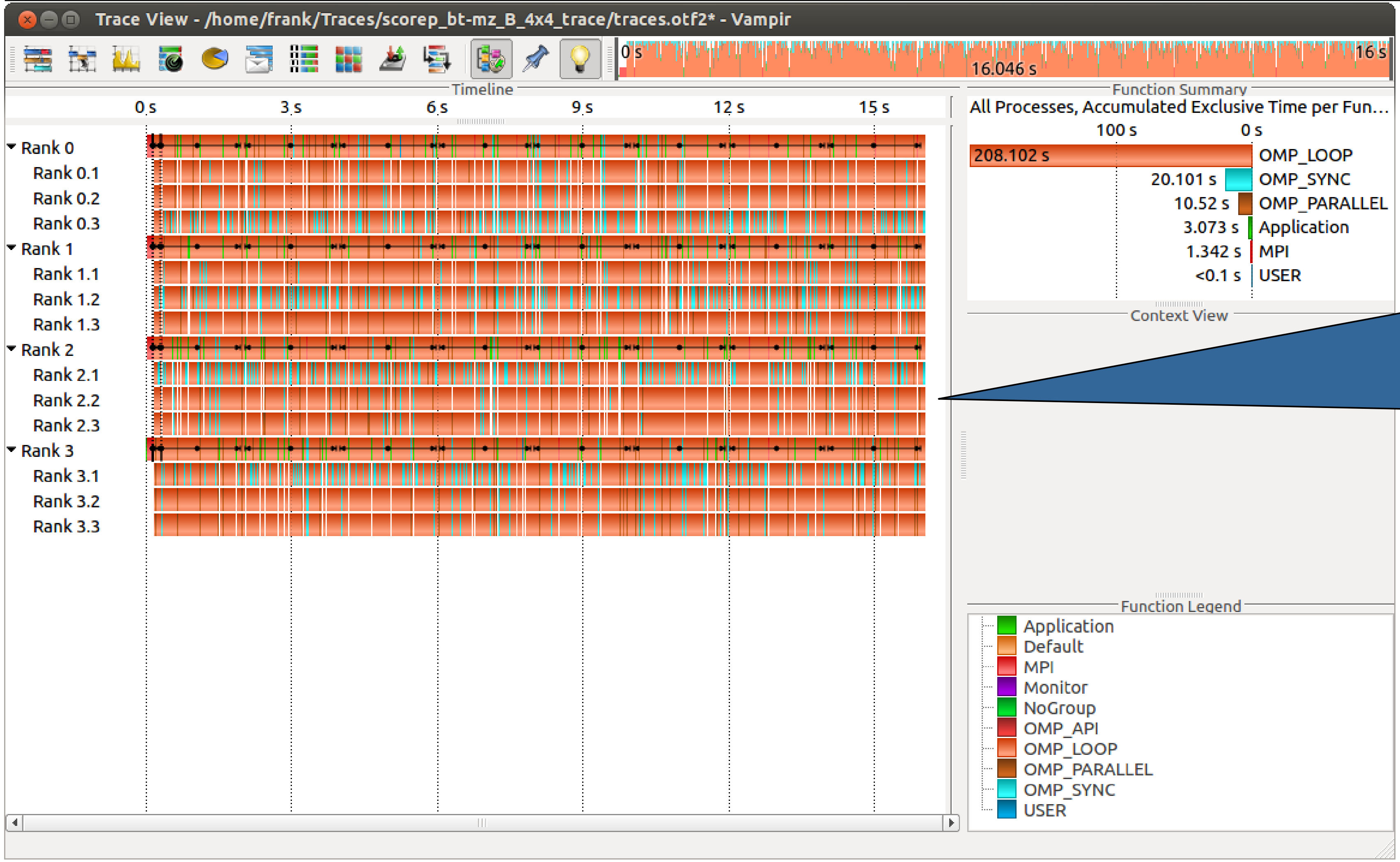
Navigation Toolbar

Function Summary

Master Timeline

Function Legend

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



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

Function Summary

All Processes, Accumulated Exclusive Time per Fun…

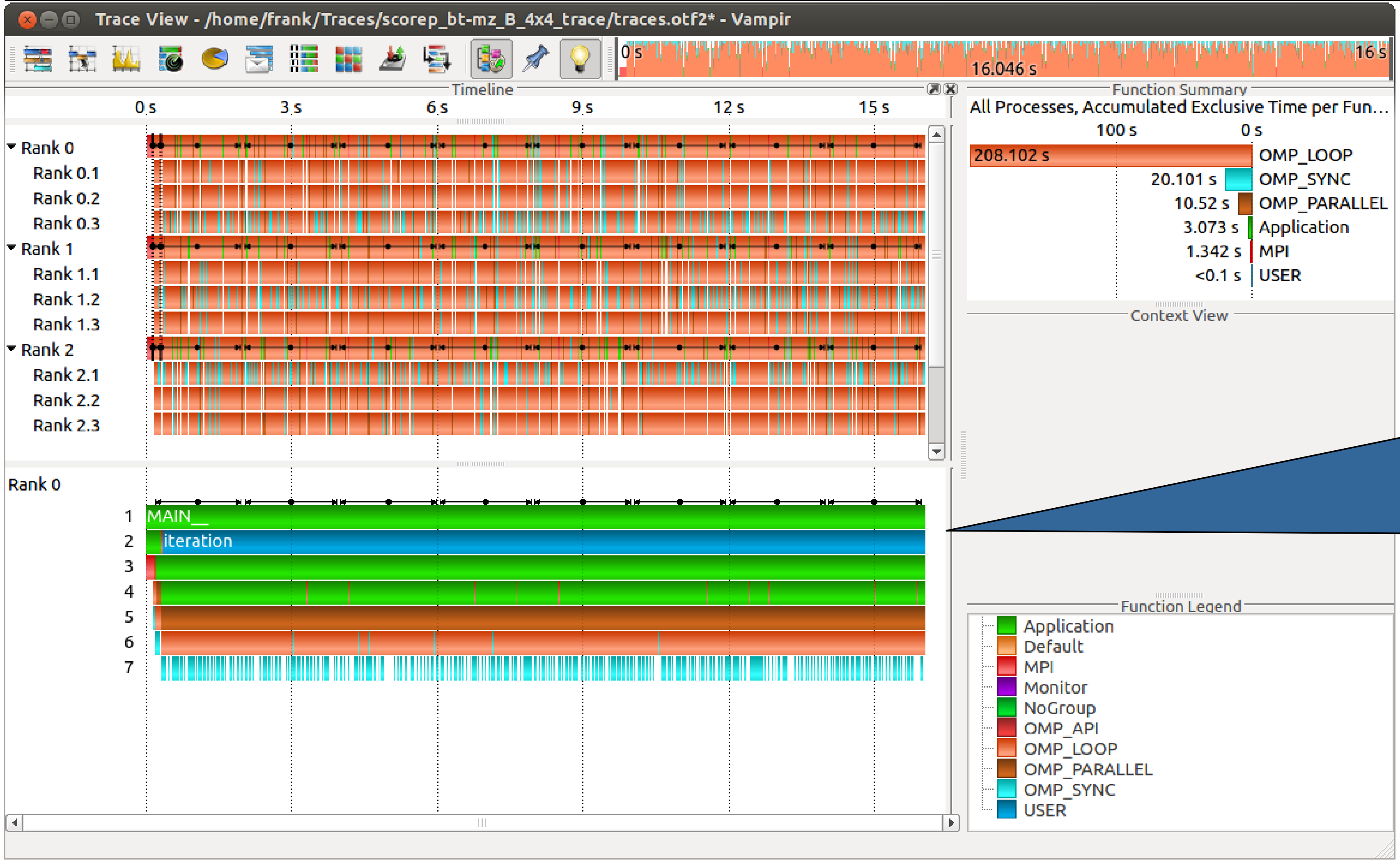
100 s	0 s
208.102 s	OMP_LOOP
20.101 s	OMP_SYNC
10.52 s	OMP_PARALLEL
3.073 s	Application
1.342 s	MPI
<0.1 s	USER

Function Legend

- 🟩 Application
- 🟧 Default
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

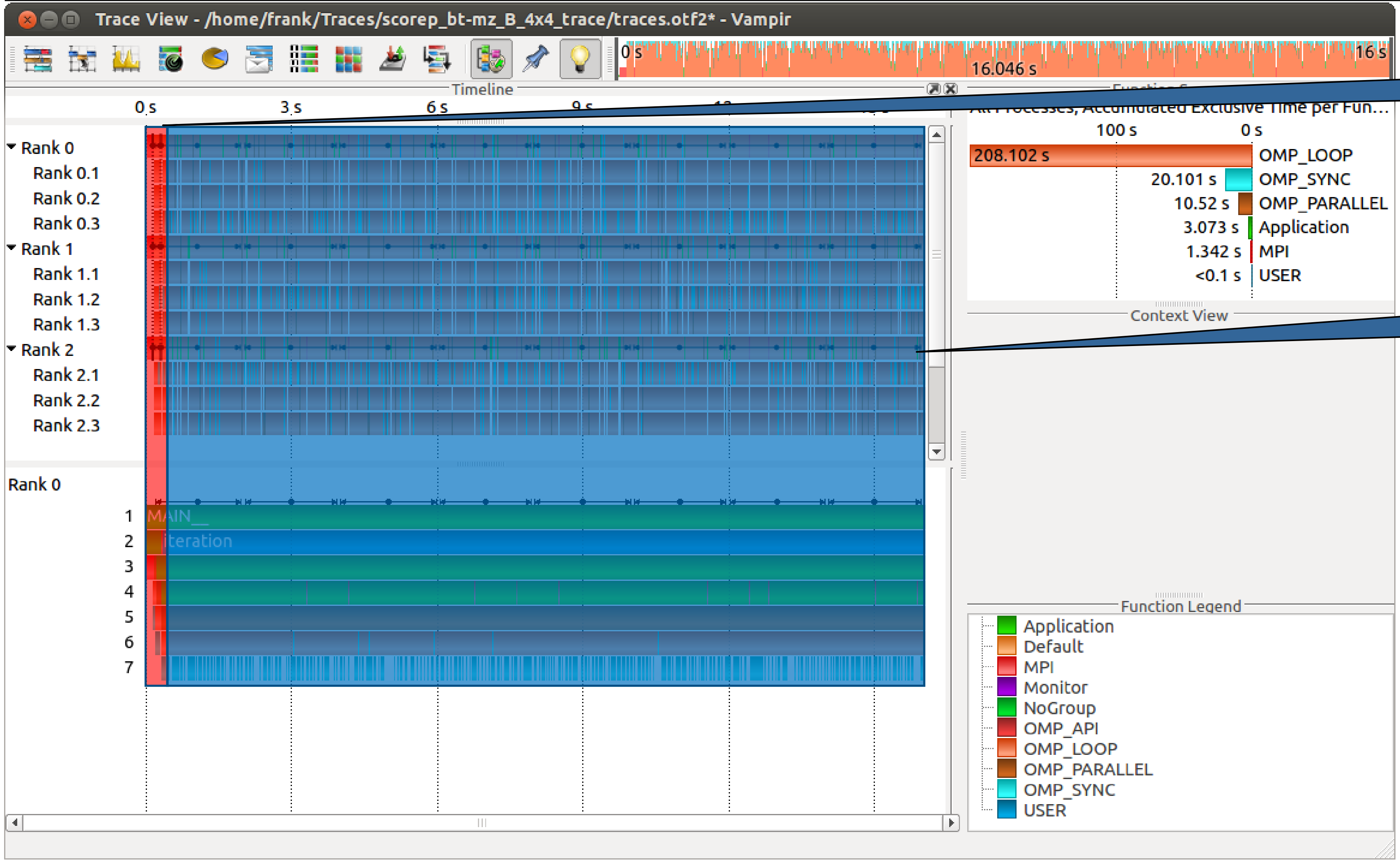
Detailed information about functions, communication and synchronization events for collection of processes.

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
Typical program phases



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

All Processes, Accumulated Exclusive Time per Fun…

100 s	0 s
208.102 s	OMP_LOOP
20.101 s	OMP_SYNC
10.52 s	OMP_PARALLEL
3.073 s	Application
1.342 s	MPI
<0.1 s	USER

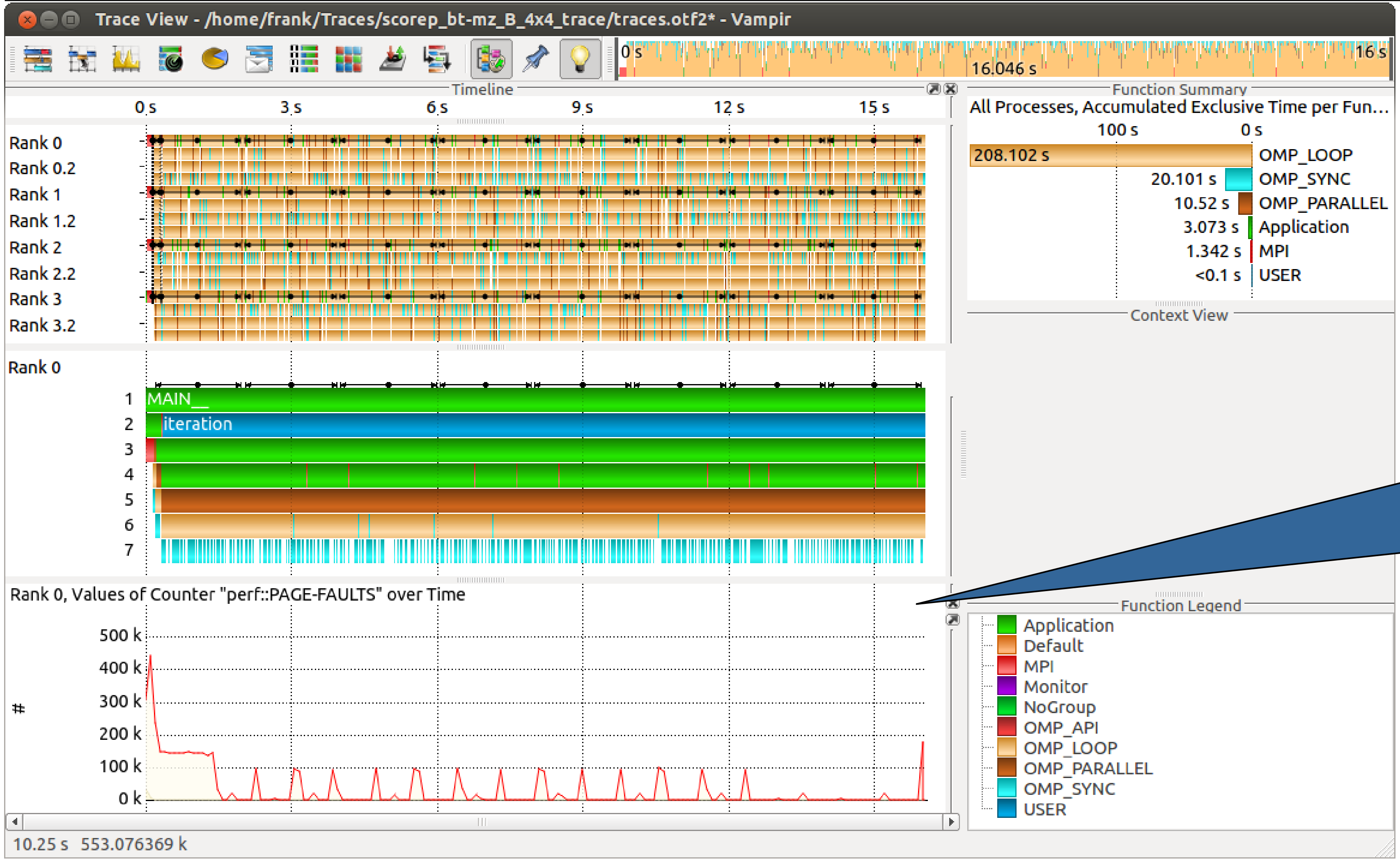
Function Legend

- 🟩 Application
- 🟧 Default
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

Initialisation Phase

Computation Phase

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


Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

All Processes, Accumulated Exclusive Time per Fun…

100 s … 0 s	
208.102 s	OMP_LOOP
20.101 s	OMP_SYNC
10.52 s	OMP_PARALLEL
3.073 s	Application
1.342 s	MPI
<0.1 s	USER

Rank 0, Values of Counter "perf::PAGE-FAULTS" over Time

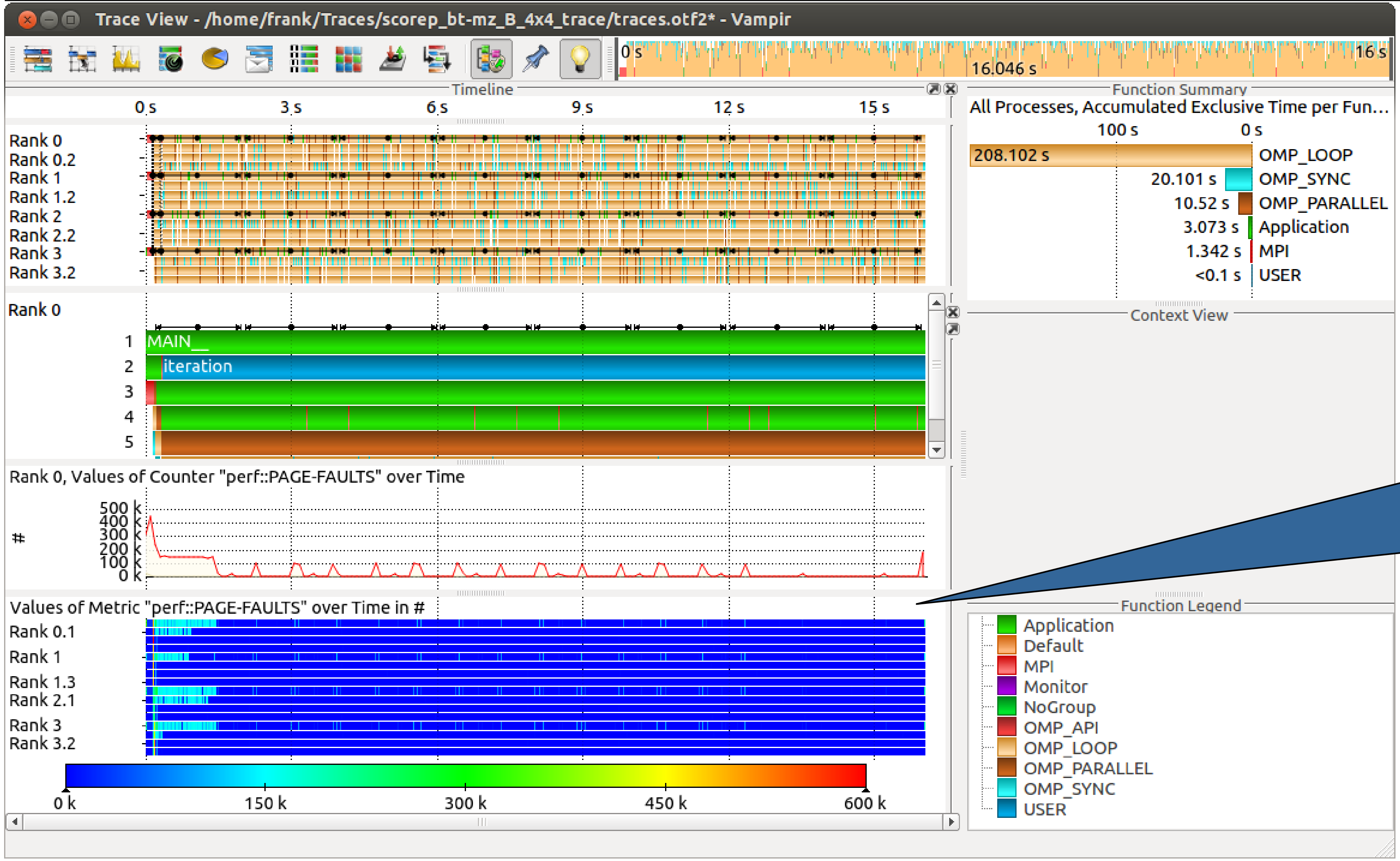
- Application
- Default
- MPI
- Monitor
- NoGroup
- OMP_API
- OMP_LOOP
- OMP_PARALLEL
- OMP_SYNC
- USER

10.25 s 553.076369 k

Detailed counter information over time for an individual process.

Visualization of the NPB-MZ-MPI / BT trace

Performance Radar



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

0 s 16.046 s 16 s

Timeline

0 s 3 s 6 s 9 s 12 s 15 s

Rank 0
Rank 0.2
Rank 1
Rank 1.2
Rank 2
Rank 2.2
Rank 3
Rank 3.2

Rank 0

1 MAIN__
2 iteration
3
4
5

Rank 0, Values of Counter "perf::PAGE-FAULTS" over Time

Values of Metric "perf::PAGE-FAULTS" over Time in #

Rank 0.1
Rank 1
Rank 1.3
Rank 2.1
Rank 3
Rank 3.2

0 k 150 k 300 k 450 k 600 k

Function Summary

All Processes, Accumulated Exclusive Time per Fun…

Time	Function
208.102 s	OMP_LOOP
20.101 s	OMP_SYNC
10.52 s	OMP_PARALLEL
3.073 s	Application
1.342 s	MPI
<0.1 s	USER

Context View

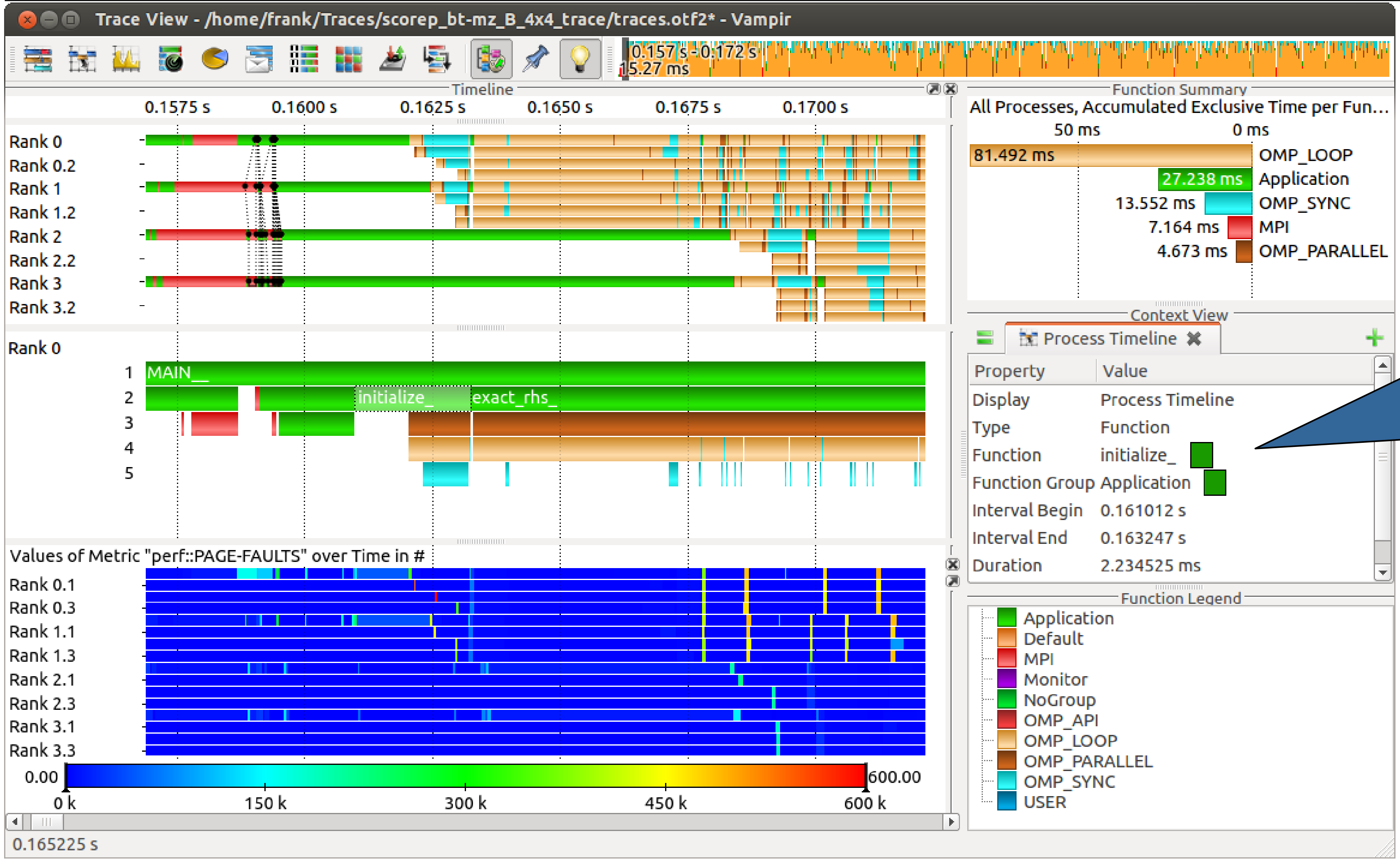
Function Legend

- Application
- Default
- MPI
- Monitor
- NoGroup
- OMP_API
- OMP_LOOP
- OMP_PARALLEL
- OMP_SYNC
- USER

Detailed counter information over time for a collection of processes.

Visualization of the NPB-MZ-MPI / BT trace

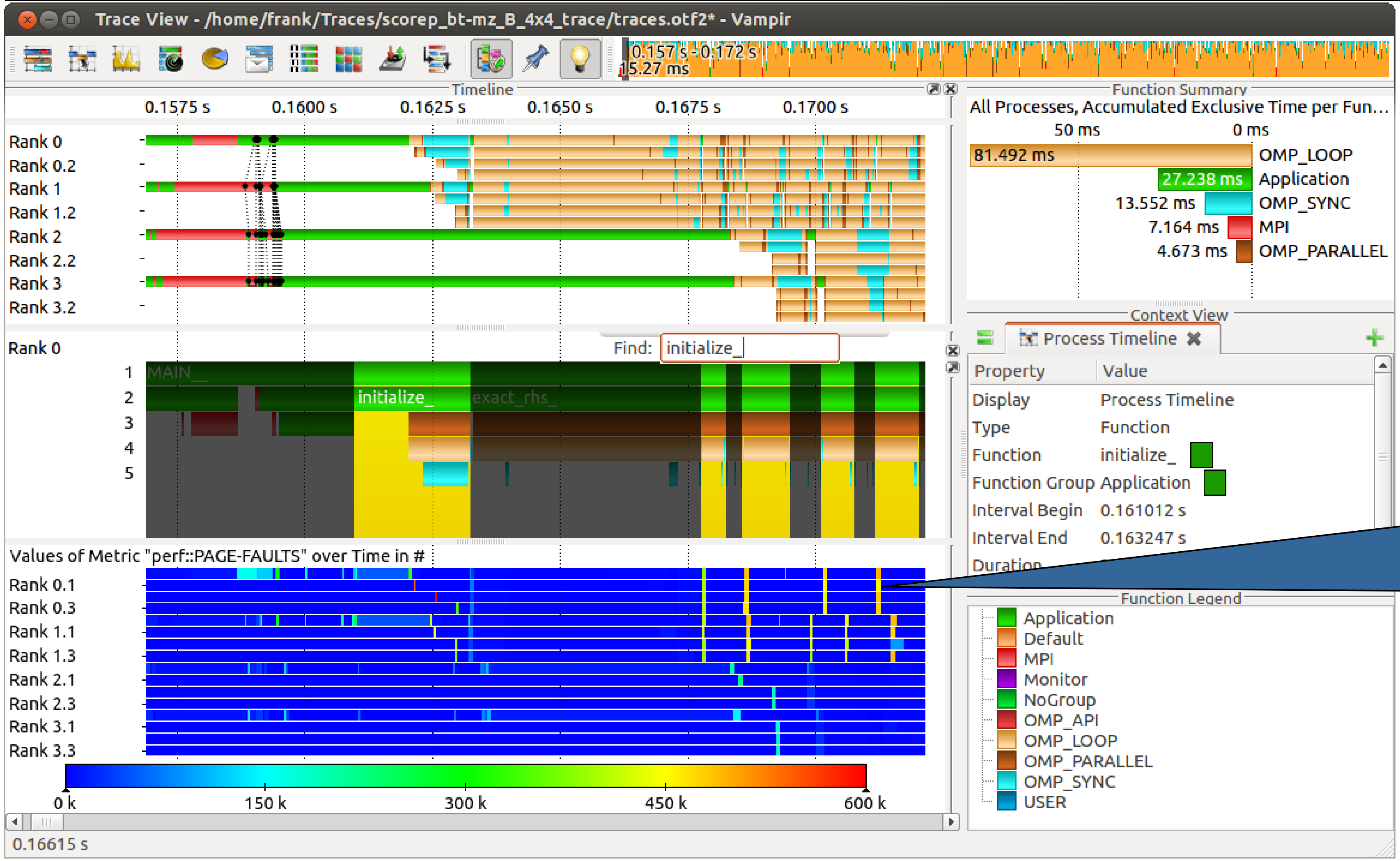
Zoom in: Inititialisation Phase



Property	Value
Display	Process Timeline
Type	Function
Function	initialize_ 🟩
Function Group	Application 🟩
Interval Begin	0.161012 s
Interval End	0.163247 s
Duration	2.234525 ms

Context View: Detailed information about function “initialize_”.

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



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

Find: initialize_

All Processes, Accumulated Exclusive Time per Fun...

81.492 ms	OMP_LOOP
27.238 ms	Application
13.552 ms	OMP_SYNC
7.164 ms	MPI
4.673 ms	OMP_PARALLEL

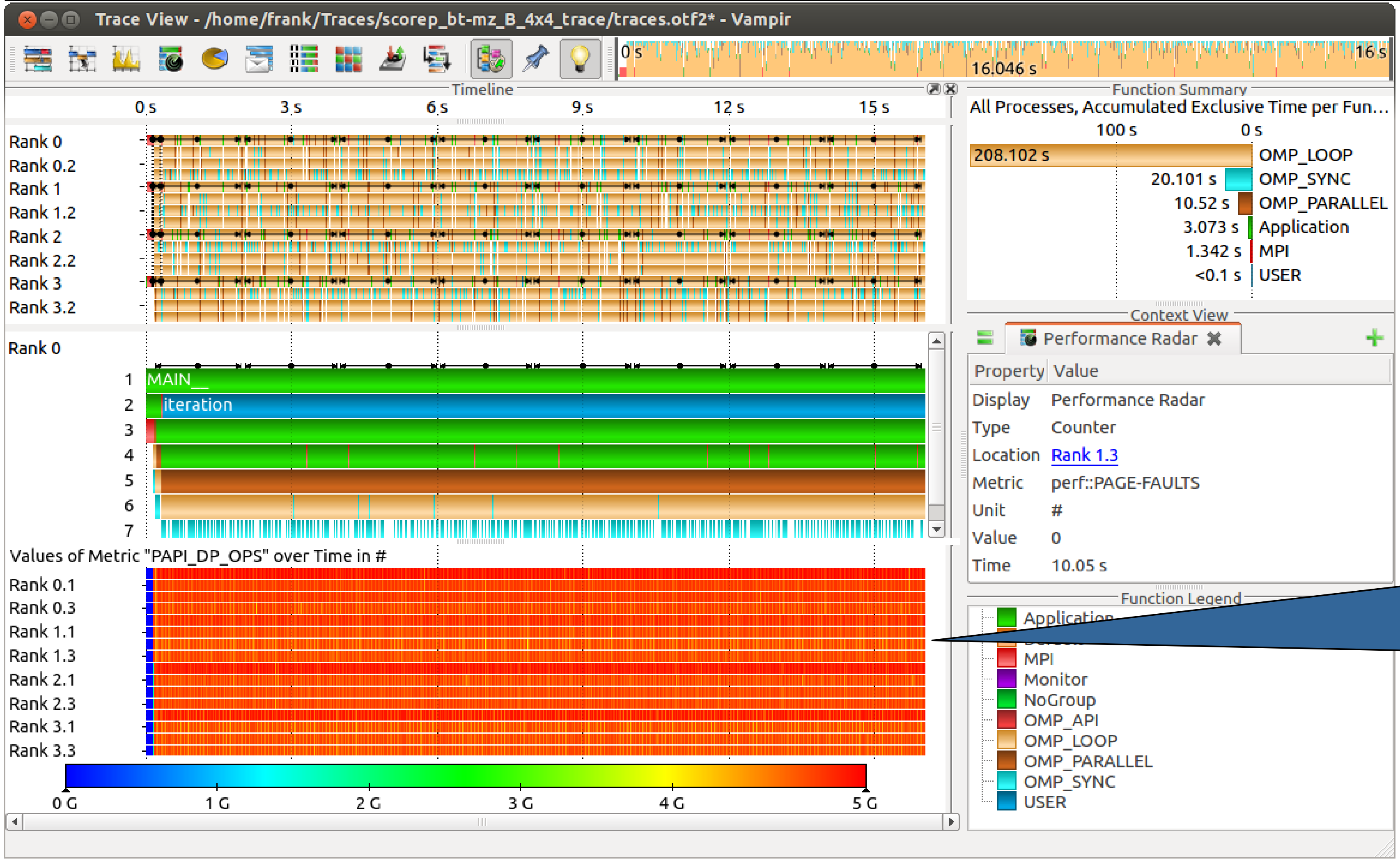
Property	Value
Display	Process Timeline
Type	Function
Function	initialize_ 🟩
Function Group	Application 🟩
Interval Begin	0.161012 s
Interval End	0.163247 s
Duration	

- 🟩 Application
- 🟧 Default
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

Execution of function “initialize_” results in higher page fault rates.

Visualization of the NPB-MZ-MPI / BT trace

Computation Phase



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

Function Summary

All Processes, Accumulated Exclusive Time per Fun...

Time	Function
208.102 s	OMP_LOOP
20.101 s	OMP_SYNC
10.52 s	OMP_PARALLEL
3.073 s	Application
1.342 s	MPI
<0.1 s	USER

Context View — Performance Radar

Property	Value
Display	Performance Radar
Type	Counter
Location	Rank 1.3
Metric	perf::PAGE-FAULTS
Unit	#
Value	0
Time	10.05 s

Function Legend

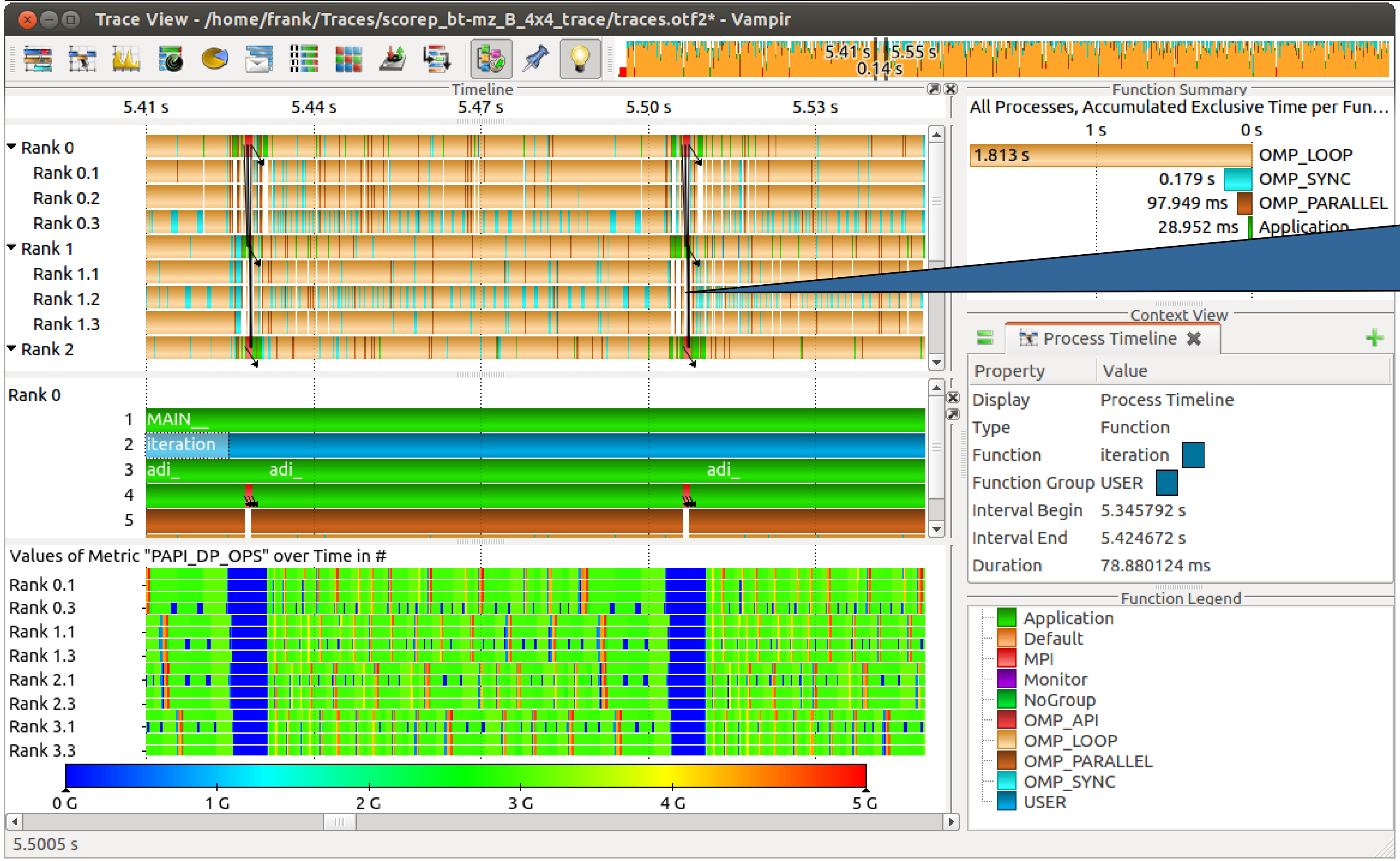
- 🟩 Application
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

Values of Metric "PAPI_DP_OPS" over Time in #

Computation phase results in higher floating point operations.

Visualization of the NPB-MZ-MPI / BT trace

Zoom in: Computation Phase



Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

All Processes, Accumulated Exclusive Time per Fun…

Time	Function
1.813 s	OMP_LOOP
0.179 s	OMP_SYNC
97.949 ms	OMP_PARALLEL
28.952 ms	Application

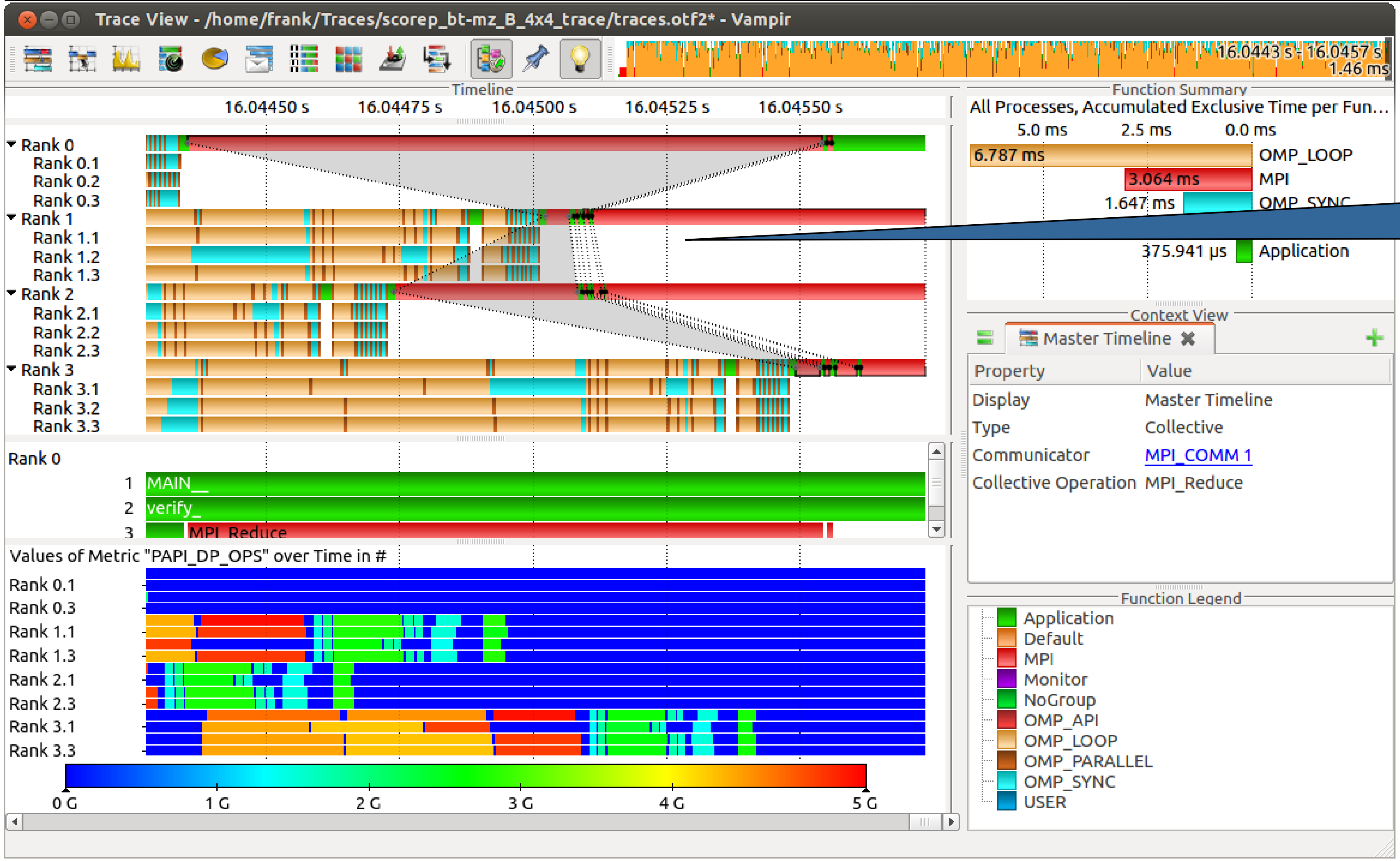
Property	Value
Display	Process Timeline
Type	Function
Function	iteration
Function Group	USER
Interval Begin	5.345792 s
Interval End	5.424672 s
Duration	78.880124 ms

- 🟩 Application
- 🟧 Default
- 🟥 MPI
- 🟪 Monitor
- 🟩 NoGroup
- 🟥 OMP_API
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC
- 🟦 USER

MPI communication results in lower floating point operations.

Visualization of the NPB-MZ-MPI / BT trace

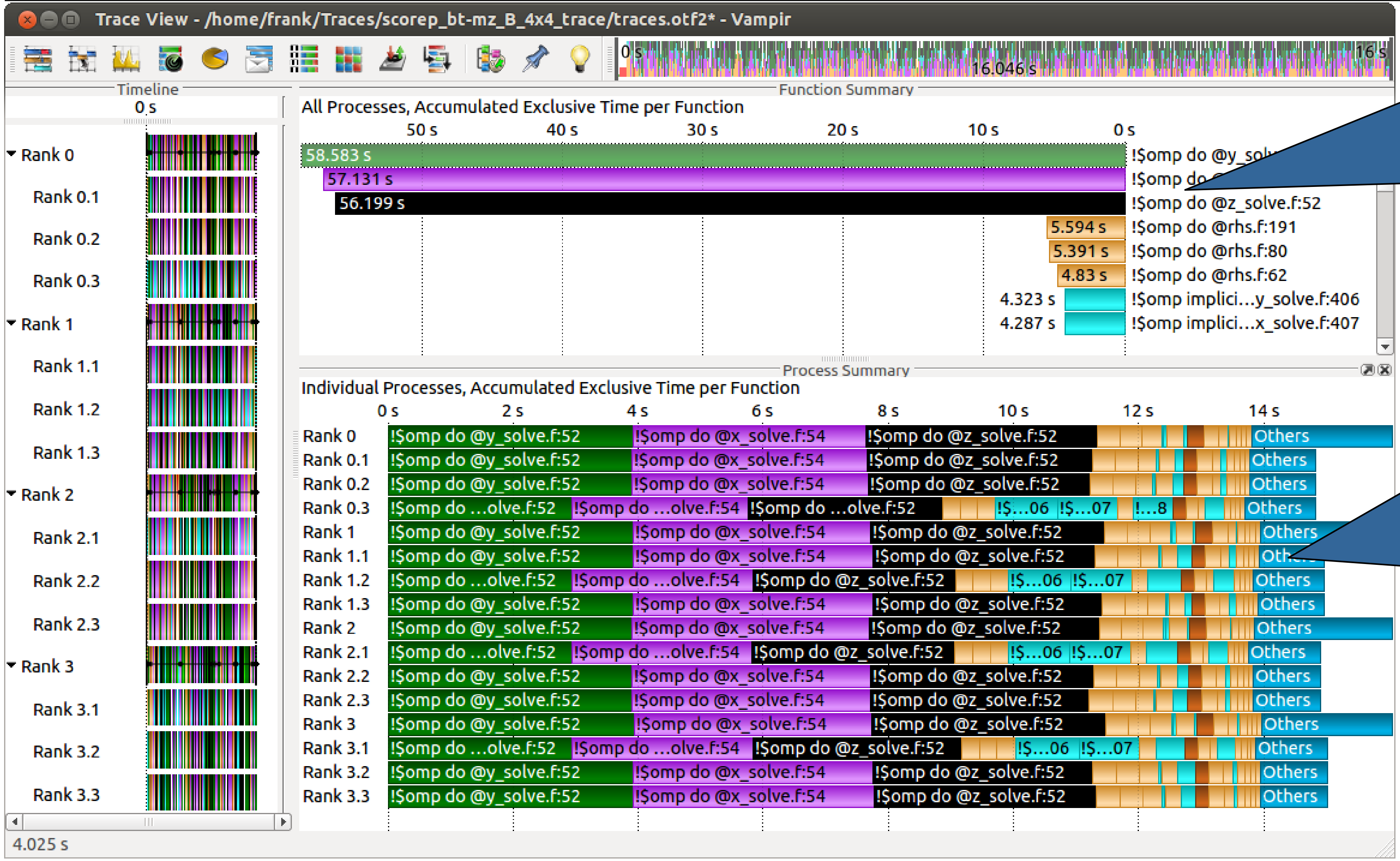
Zoom in: Finalisation Phase



Property	Value
Display	Master Timeline
Type	Collective
Communicator	MPI_COMM 1
Collective Operation	MPI_Reduce

"Early reduce" bottleneck.

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

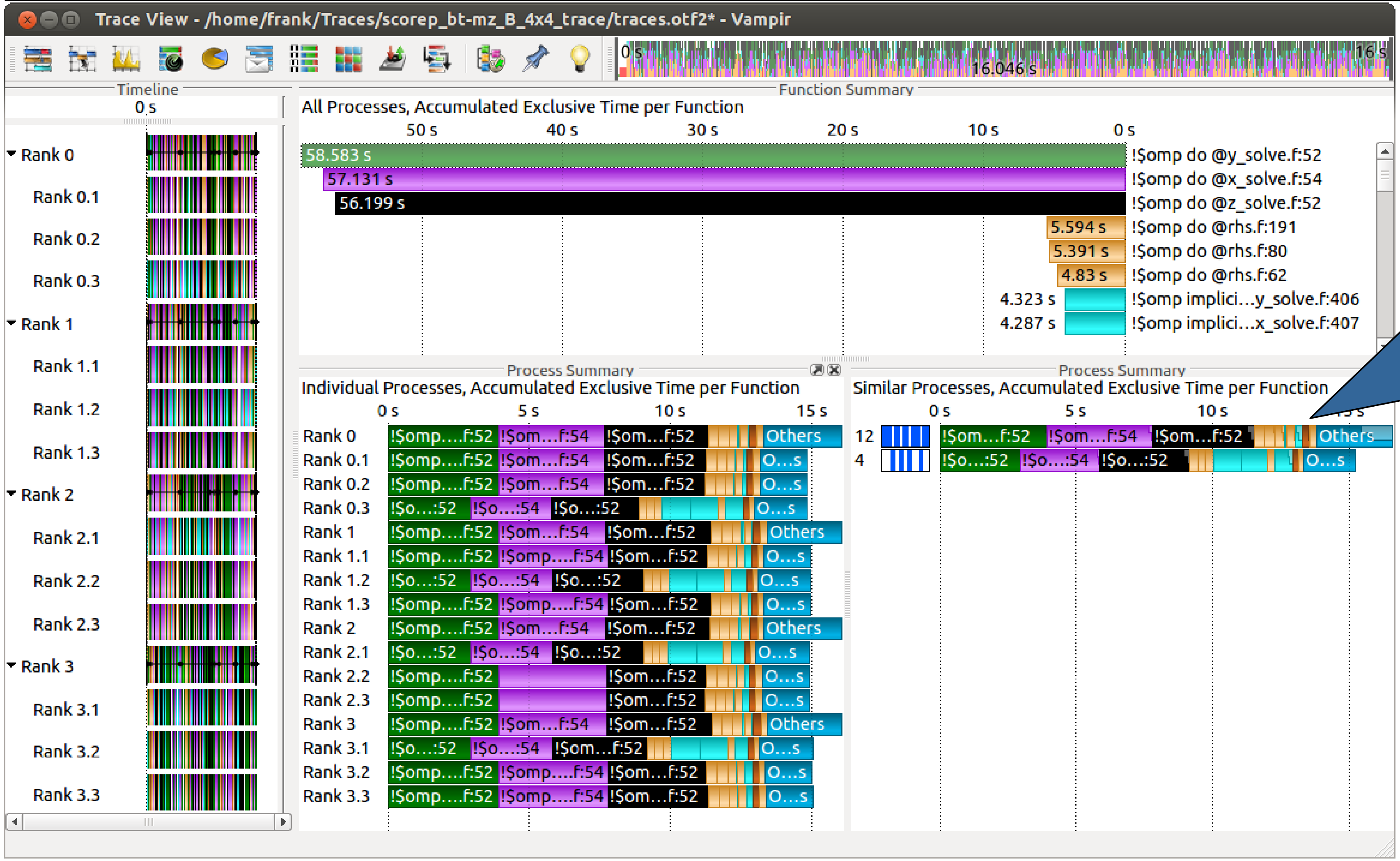

Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir
0 s
16.046 s
16 s
Timeline
0 s
Rank 0
Rank 0.1
Rank 0.2
Rank 0.3
Rank 1
Rank 1.1
Rank 1.2
Rank 1.3
Rank 2
Rank 2.1
Rank 2.2
Rank 2.3
Rank 3
Rank 3.1
Rank 3.2
Rank 3.3
4.025 s
Function Summary
All Processes, Accumulated Exclusive Time per Function
50 s
40 s
30 s
20 s
10 s
0 s
58.583 s
!$omp do @y_sol…
57.131 s
!$omp do …
56.199 s
!$omp do @z_solve.f:52
5.594 s
!$omp do @rhs.f:191
5.391 s
!$omp do @rhs.f:80
4.83 s
!$omp do @rhs.f:62
4.323 s
!$omp implici…y_solve.f:406
4.287 s
!$omp implici…x_solve.f:407
Process Summary
Individual Processes, Accumulated Exclusive Time per Function
0 s
2 s
4 s
6 s
8 s
10 s
12 s
14 s
Rank 0 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 0.1 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 0.2 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 0.3 !$omp do …olve.f:52 !$omp do …olve.f:54 !$omp do …olve.f:52 !$…06 !$…07 !…8 Others
Rank 1 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 1.1 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Oth…
Rank 1.2 !$omp do …olve.f:52 !$omp do …olve.f:54 !$omp do @z_solve.f:52 !$…06 !$…07 Others
Rank 1.3 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 2 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 2.1 !$omp do …olve.f:52 !$omp do …olve.f:54 !$omp do @z_solve.f:52 !$…06 !$…07 Others
Rank 2.2 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 2.3 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 3 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 3.1 !$omp do …olve.f:52 !$omp do …olve.f:54 !$omp do @z_solve.f:52 !$…06 !$…07 Others
Rank 3.2 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others
Rank 3.3 !$omp do @y_solve.f:52 !$omp do @x_solve.f:54 !$omp do @z_solve.f:52 Others

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


Trace View - /home/frank/Traces/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir
0 s
16.046 s
16 s
Timeline
0 s
Rank 0
Rank 0.1
Rank 0.2
Rank 0.3
Rank 1
Rank 1.1
Rank 1.2
Rank 1.3
Rank 2
Rank 2.1
Rank 2.2
Rank 2.3
Rank 3
Rank 3.1
Rank 3.2
Rank 3.3
Function Summary
All Processes, Accumulated Exclusive Time per Function
50 s
40 s
30 s
20 s
10 s
0 s
58.583 s !$omp do @y_solve.f:52
57.131 s !$omp do @x_solve.f:54
56.199 s !$omp do @z_solve.f:52
5.594 s !$omp do @rhs.f:191
5.391 s !$omp do @rhs.f:80
4.83 s !$omp do @rhs.f:62
4.323 s !$omp implici…y_solve.f:406
4.287 s !$omp implici…x_solve.f:407
Process Summary
Individual Processes, Accumulated Exclusive Time per Function
0 s
5 s
10 s
15 s
Rank 0 !$omp….f:52 !$om…f:54 !$om…f:52 Others
Rank 0.1 !$omp….f:52 !$om…f:54 !$om…f:52 O…s
Rank 0.2 !$omp….f:52 !$om…f:54 !$om…f:52 O…s
Rank 0.3 !$o…:52 !$o…:54 !$o…:52 O…s
Rank 1 !$omp….f:52 !$om…f:54 !$om…f:52 Others
Rank 1.1 !$omp….f:52 !$omp….f:54 !$om…f:52 O…s
Rank 1.2 !$o…:52 !$o…:54 !$o…:52 O…s
Rank 1.3 !$omp….f:52 !$omp….f:54 !$om…f:52 O…s
Rank 2 !$omp….f:52 !$om…f:54 !$om…f:52 Others
Rank 2.1 !$o…:52 !$o…:54 !$o…:52 O…s
Rank 2.2 !$omp….f:52 !$om…f:52 O…s
Rank 2.3 !$omp….f:52 !$om…f:52 O…s
Rank 3 !$omp….f:52 !$om…f:54 !$om…f:52 Others
Rank 3.1 !$o…:52 !$o…:54 !$om…f:52 O…s
Rank 3.2 !$omp….f:52 !$omp….f:54 !$om…f:52 O…s
Rank 3.3 !$omp….f:52 !$omp….f:54 !$om…f:52 O…s
Process Summary
Similar Processes, Accumulated Exclusive Time per Function
0 s
5 s
10 s
15 s
12 !$om…f:52 !$om…f:54 !$om…f:52 Others
4 !$o…:52 !$o…:54 !$o…:52 O…s

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

INDIANA UNIVERSITY

SPONSORED BY THE
Federal Ministry
of Education
and Research

SILC

OAK RIDGE
National Laboratory

LMAC

READEX
Runtime Exploitation of Application Dynamism
for Energy-efficient eXascale computing

eeClust
Energy-Efficient Cluster Computing

nextgenio

Score-E

VAMPIR

European Commission
Horizon 2020
European Union funding
for Research & Innovation

Microsoft®

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

SEVENTH FRAMEWORK PROGRAMME

//H4H

//ParMA

http://www.vampir.eu

vampirsupport@zih.tu-dresden.de