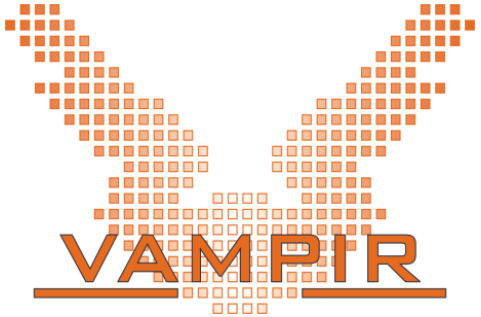
Performance Analysis with Vampir

Bert Wesarg
Technische Universität Dresden

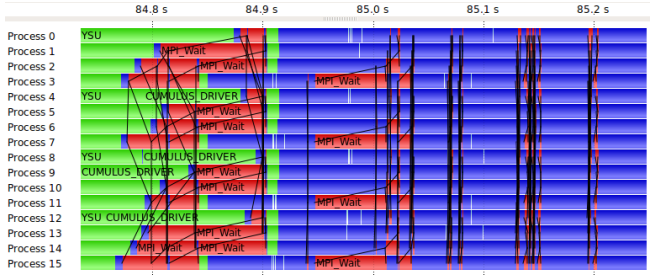


Event Trace Visualization with Vampir

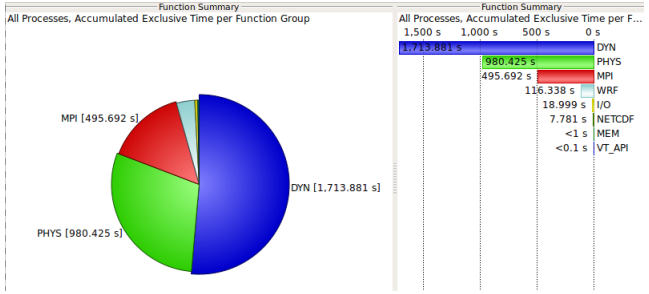
- Visualization of dynamic runtime behaviour at any level of detail along with statistics and performance metrics
- Alternative and supplement to automatic analysis

- **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

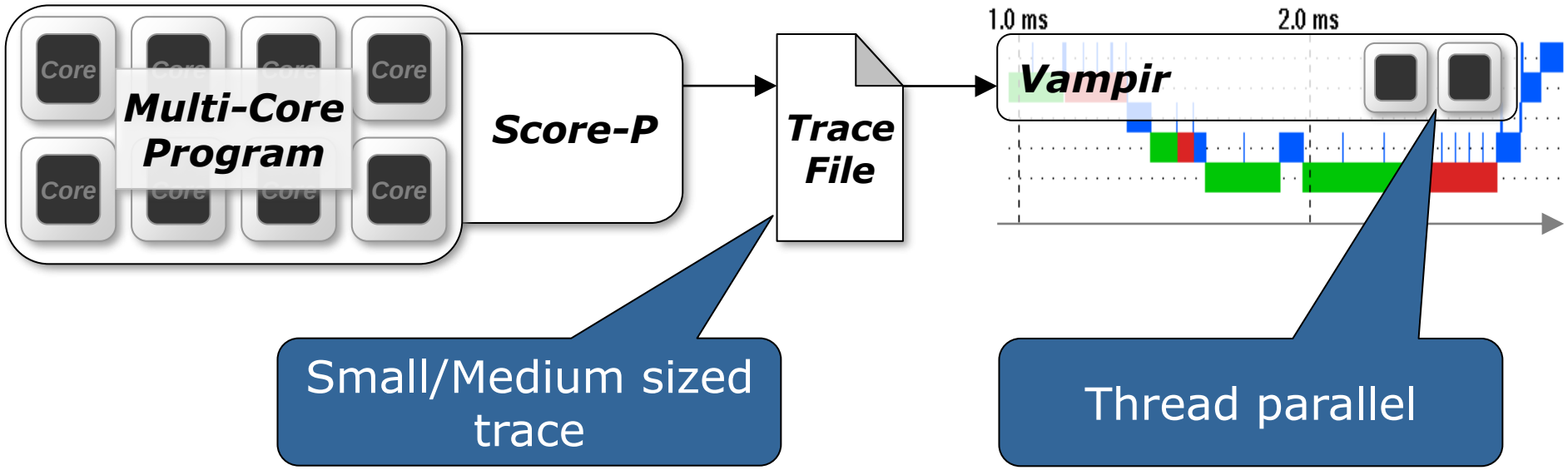


- **Summary charts**
  - Quantitative results for the currently selected time interval



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

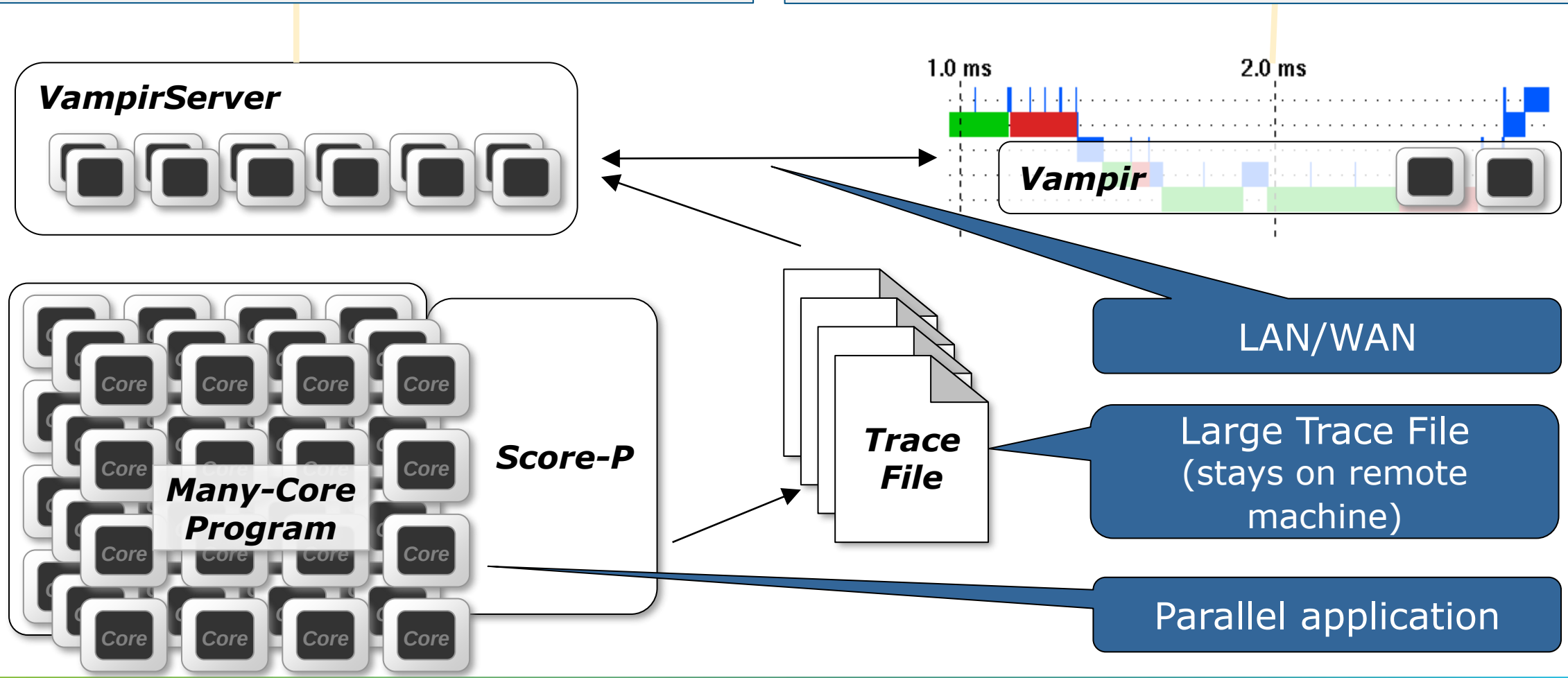
```
% vampir
```



Visualization Modes (2)
On local machine with remote VampirServer

```
% vampirserver start
```

```
% vampir
```



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

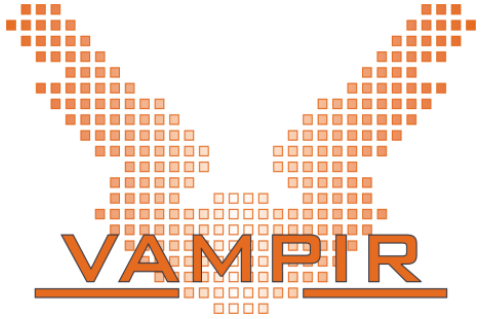

I/O Summary


Process Summary

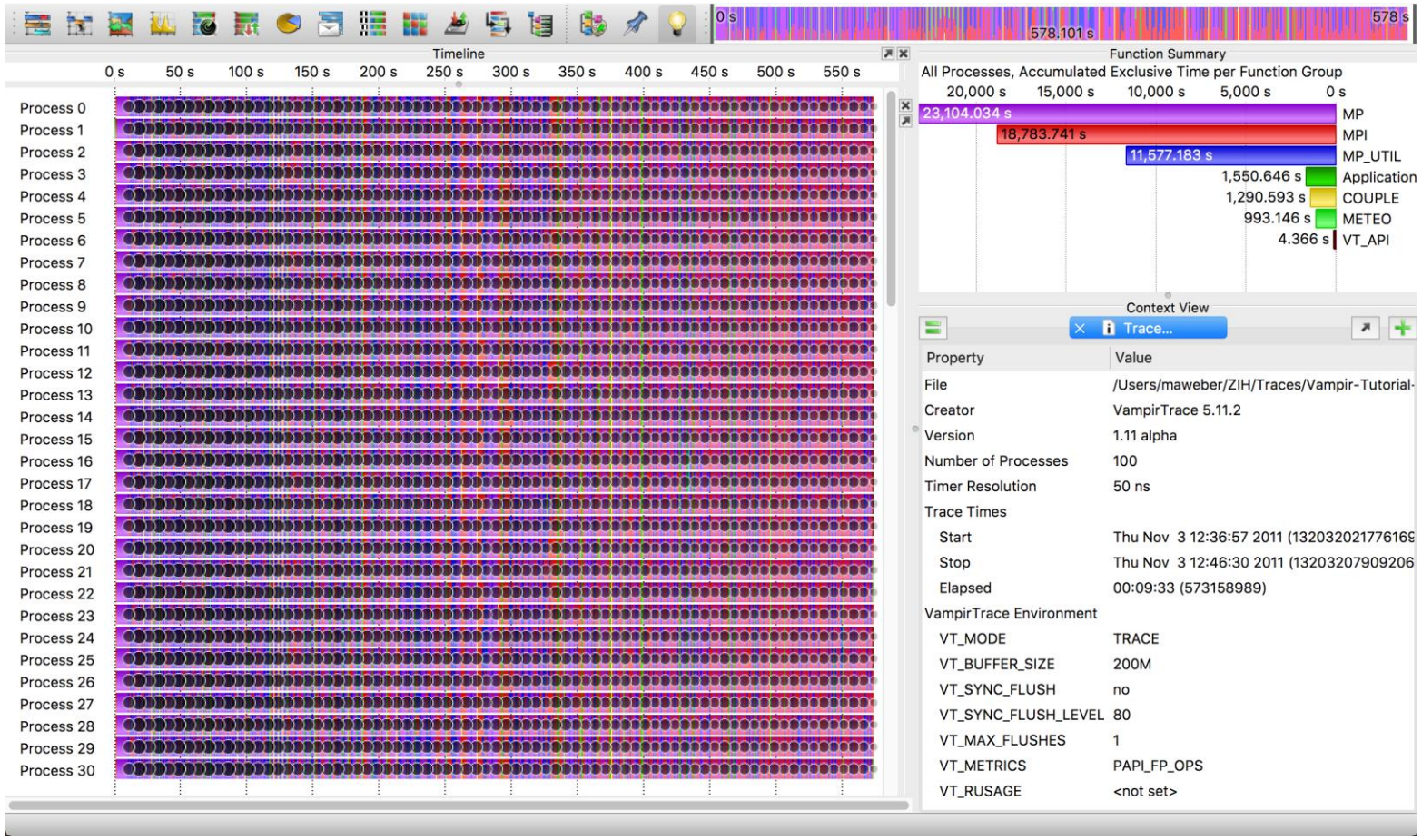

Communication Matrix View


Call Tree

Vampir Case Study: Analyzing Load Imbalance in COSMO-SPECS

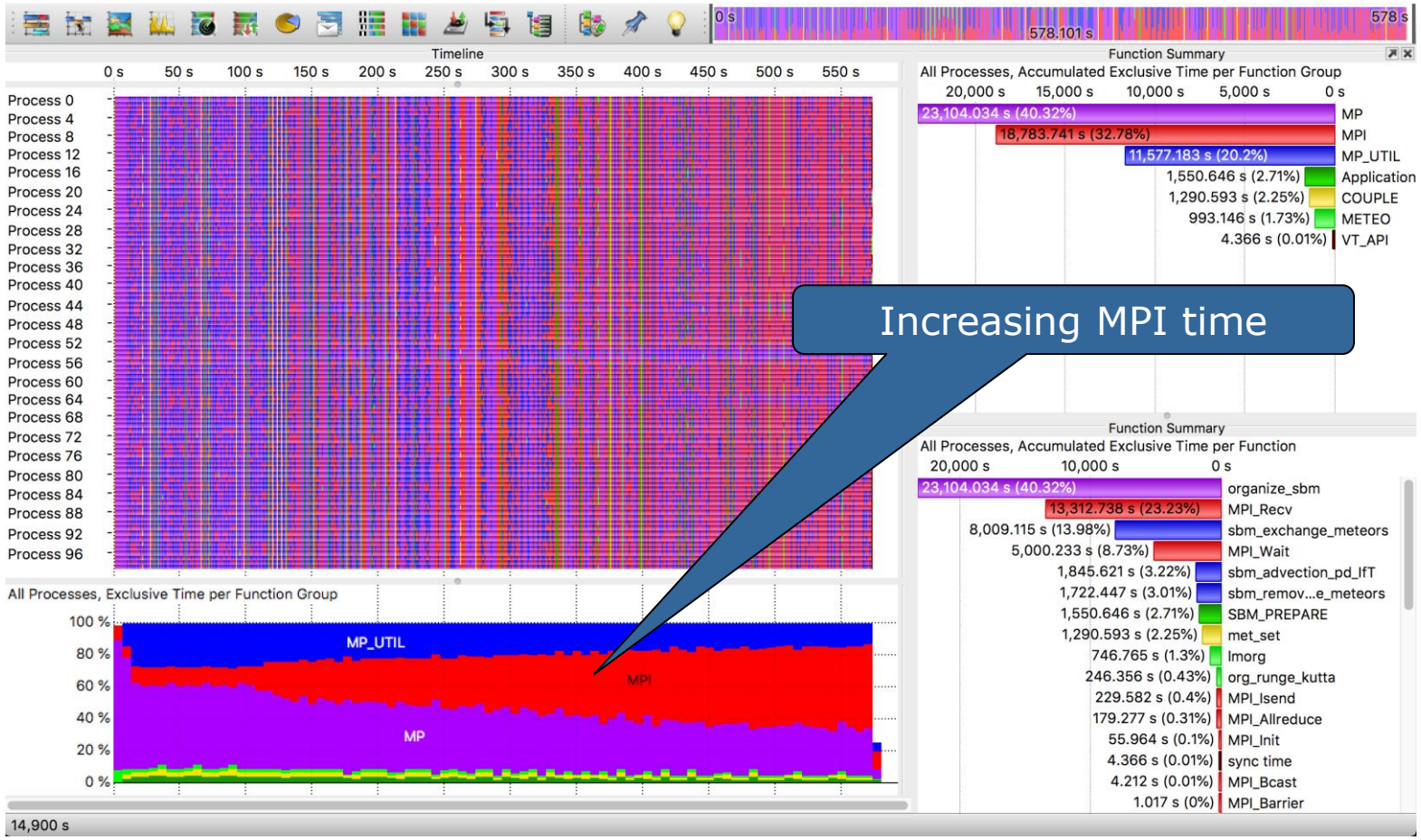


COSMO-SPECS



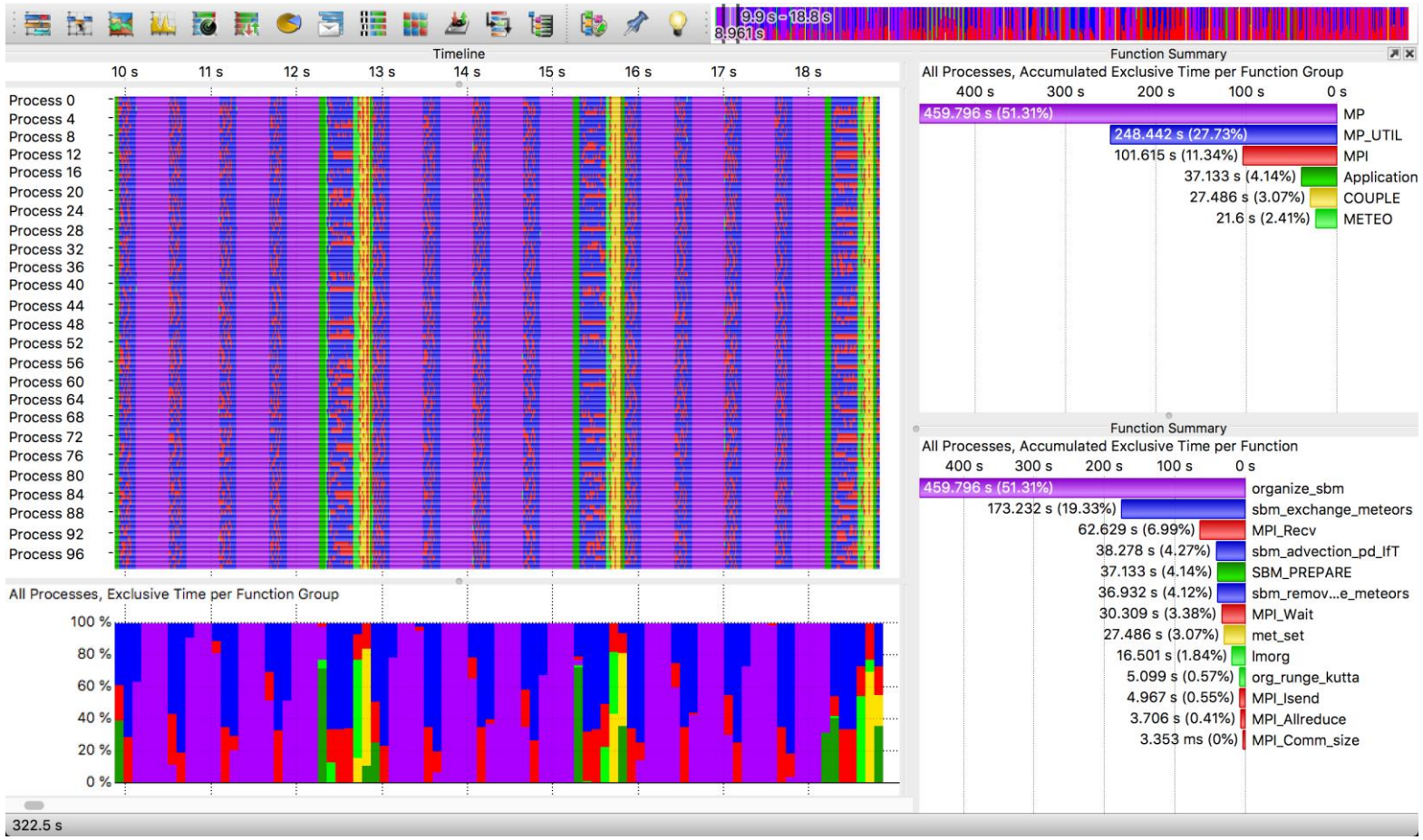
- Weather forecast code COSMO-SPECS
- Run with 100 processes
- COSMO: weather model (METEO group)
- SPECS: microphysics for accurate cloud calculation (MP and MP_UTIL group)
- Coupling of both models done in COUPLE group

COSMO-SPECS



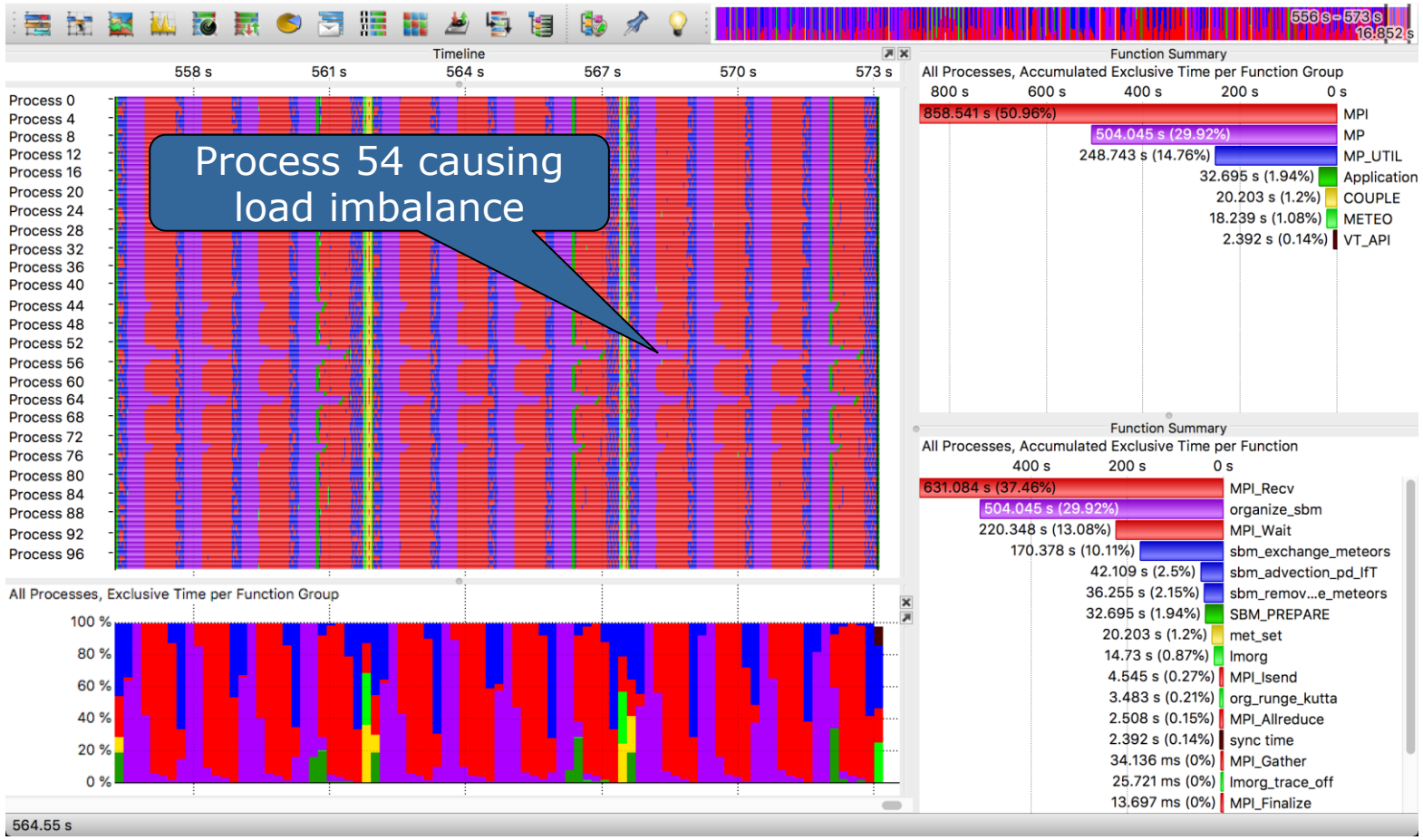
- Compared to METEO, MP and MP_UTIL are very compute intensive, however this is due to more complex calculations and no performance issue
- Problem: >32% of time spent in MPI
- MPI runtime share increases throughout the application run

COSMO-SPECS



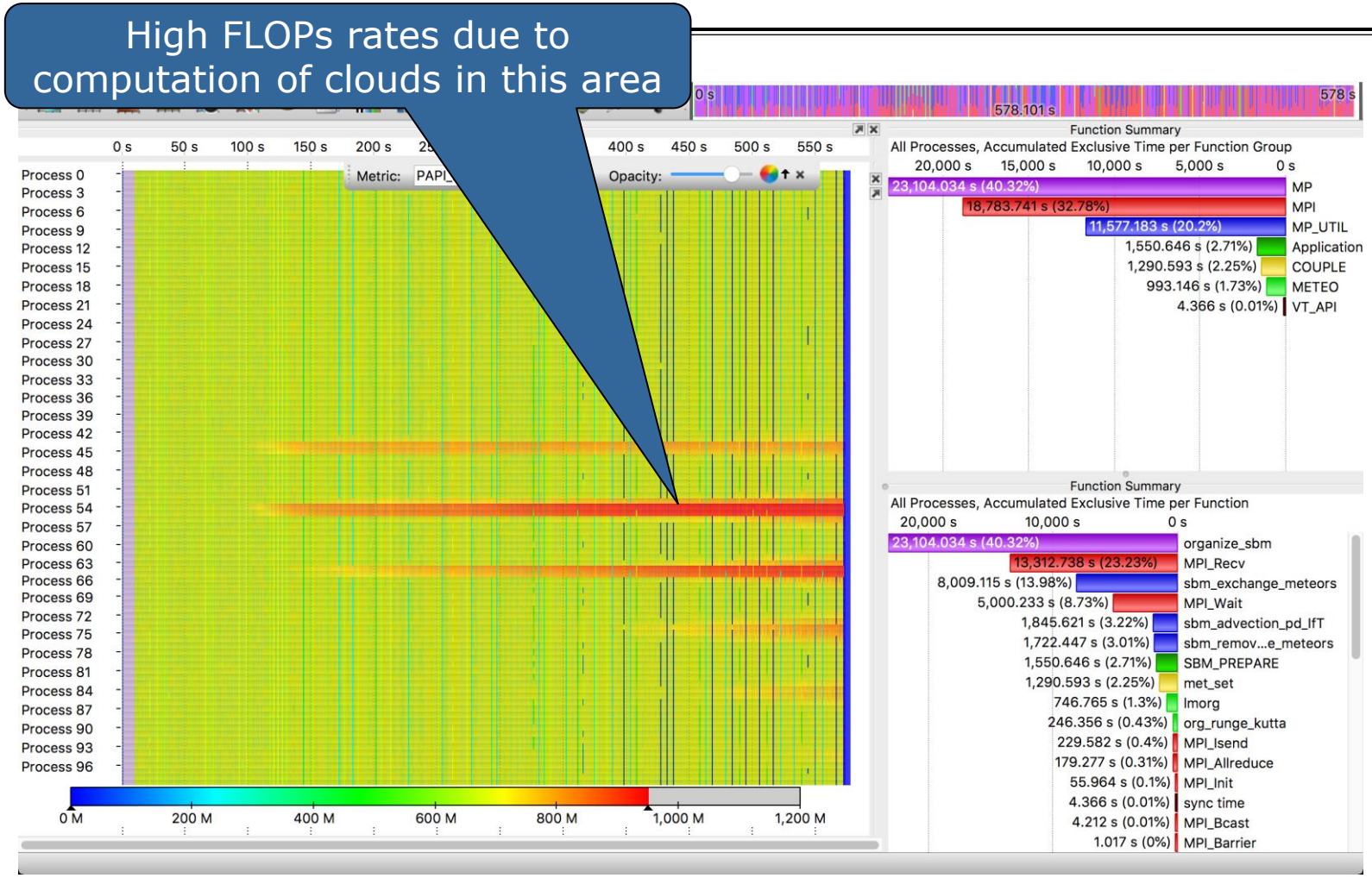
- Zoom into the first three iterations
- MP/MP_UTIL perform four sub-steps in one iteration
- Low MPI time share
- Everything is balanced and looks okay

COSMO-SPECS



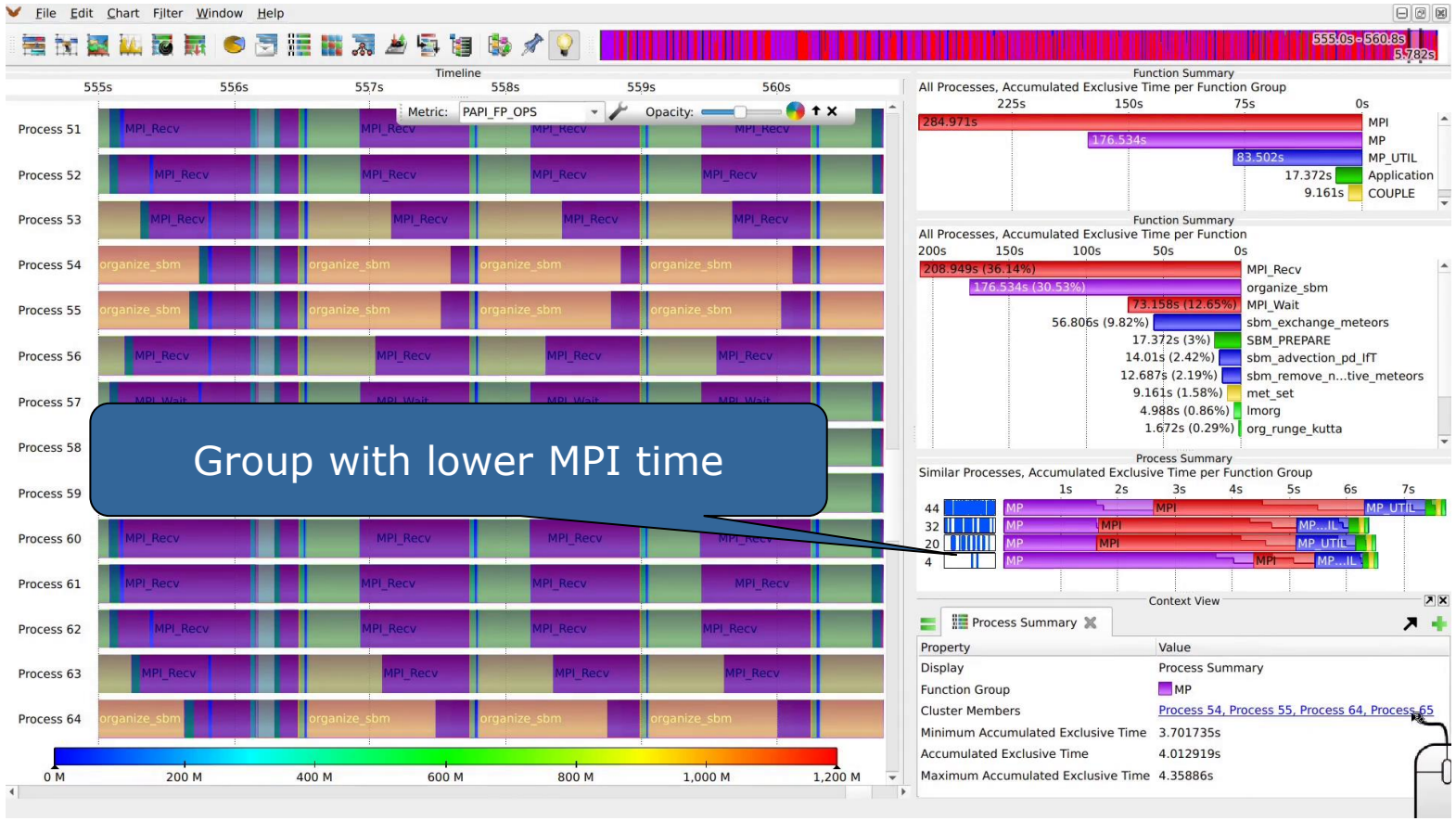
- Zoom into the last three iterations
- Very high MPI time share (>50%)
- Large load imbalance caused by MP functions around **`Process 54`** and **`Process 64`**

COSMO-SPECS



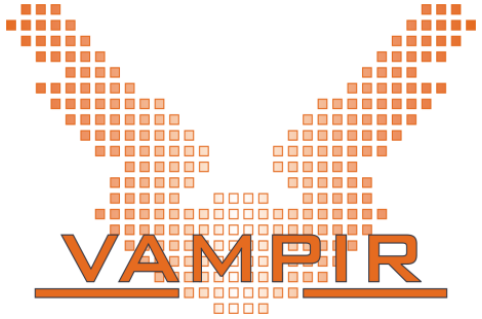
- **`PAPI_FP_OPS`** counter showing higher FLOPs rates on processes causing the imbalance
- Reason for imbalance: Static grid used for distribution of processes. Depending on the weather, expensive cloud computations (MP group) may be only necessary on some processes

COSMO-SPECS

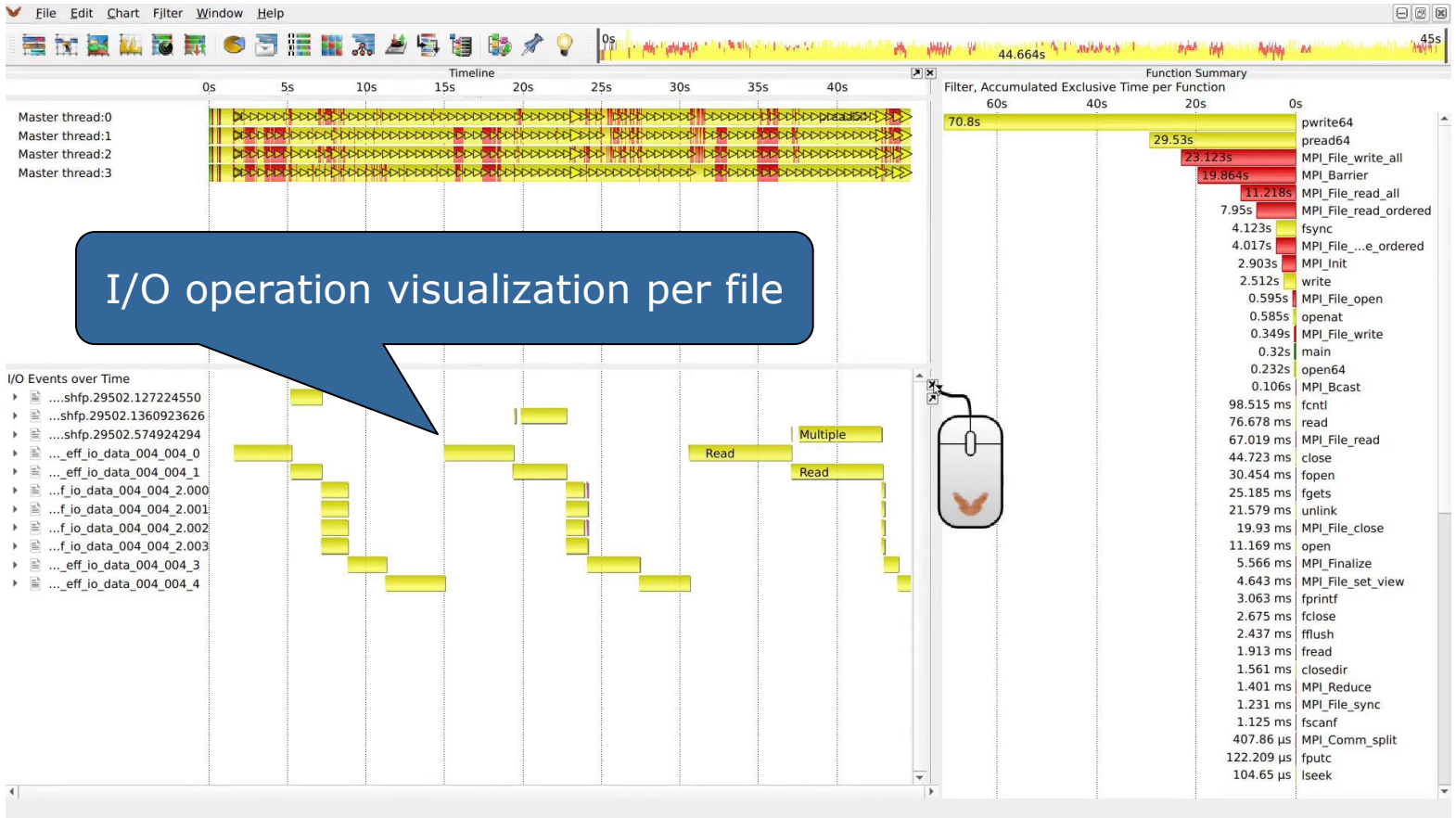


- Process Summary helps finding outliers
- Groups processes by their behavior (similar call/duration profile)
- Number of expected groups is variable
- In this case 4 yields the best results

Vampir Showcase:
Analyzing Multilayer File I/O Applications

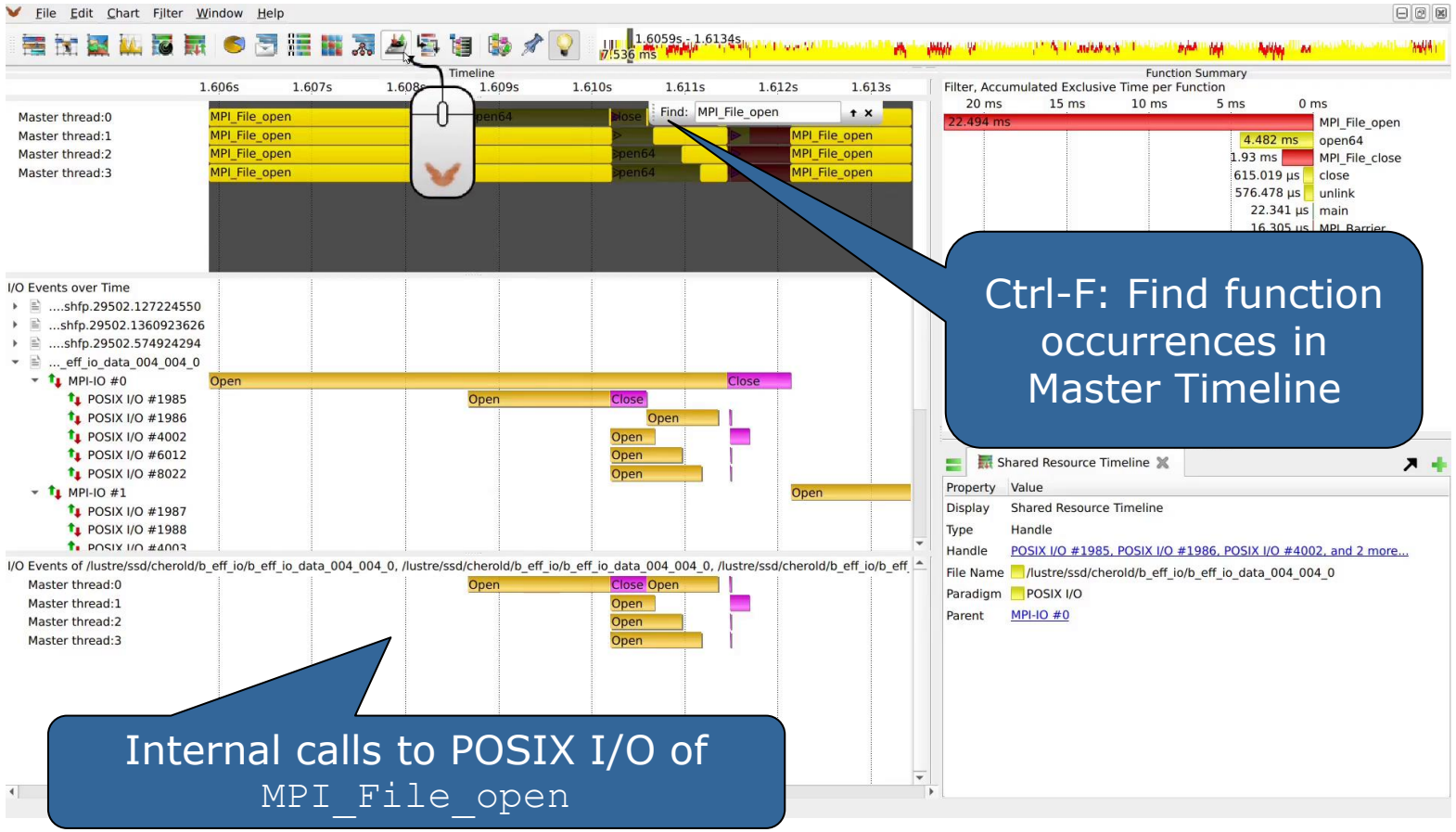


Multilayer File I/O Application



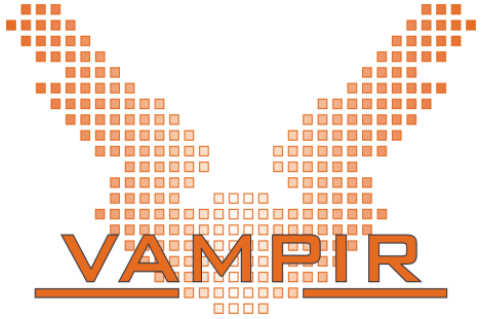
- IO bandwidth benchmark b_eff_io
- Measures achievable I/O bandwidth of parallel MPI-I/O applications
- Shared Resource Timeline offers a per file and per thread view on File I/O operations

Multilayer File I/O Application

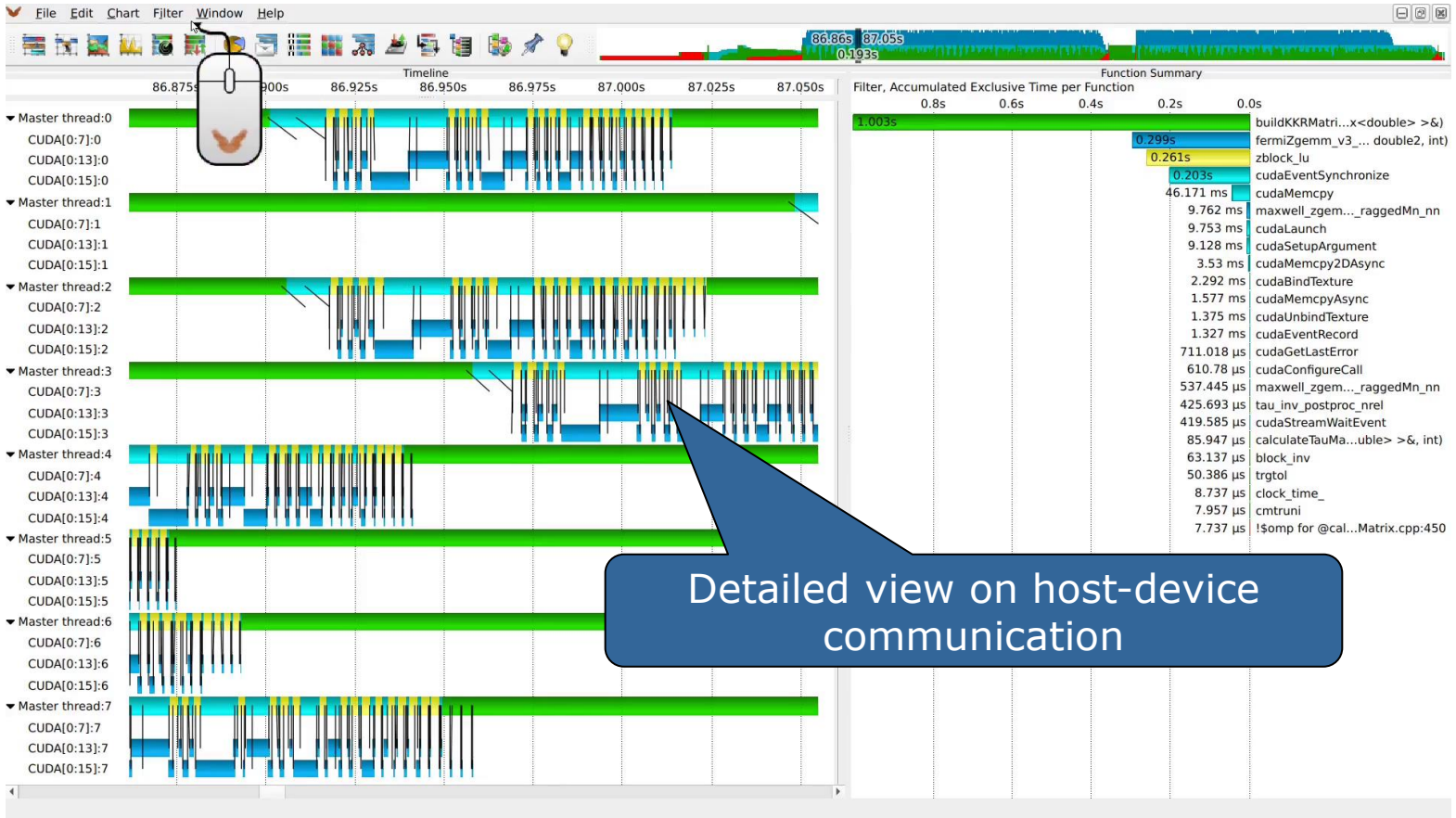


- Visualization of I/O on multiple layers (MPI & POSIX)
- Example: behavior of **`MPI_File_open`**
- Internally uses POSIX **`open`** for opening the actual file on disk
- Multiple consecutive calls to **`open`** and **`close`** on master rank

Vampir Showcase: Analyzing CUDA Applications



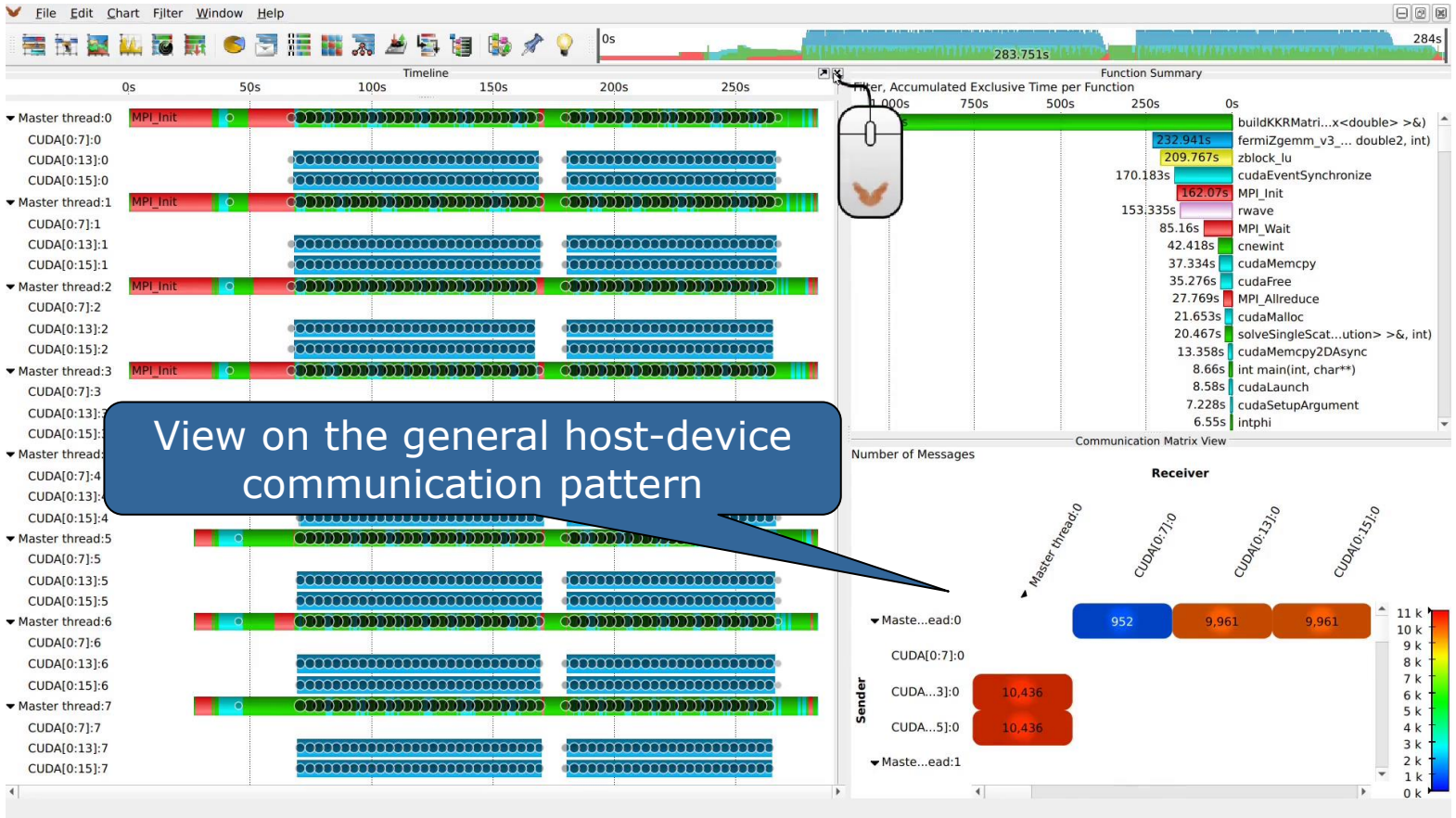
CUDA Application



- Material science code LSMS
- CUDA is utilized for heavy computations

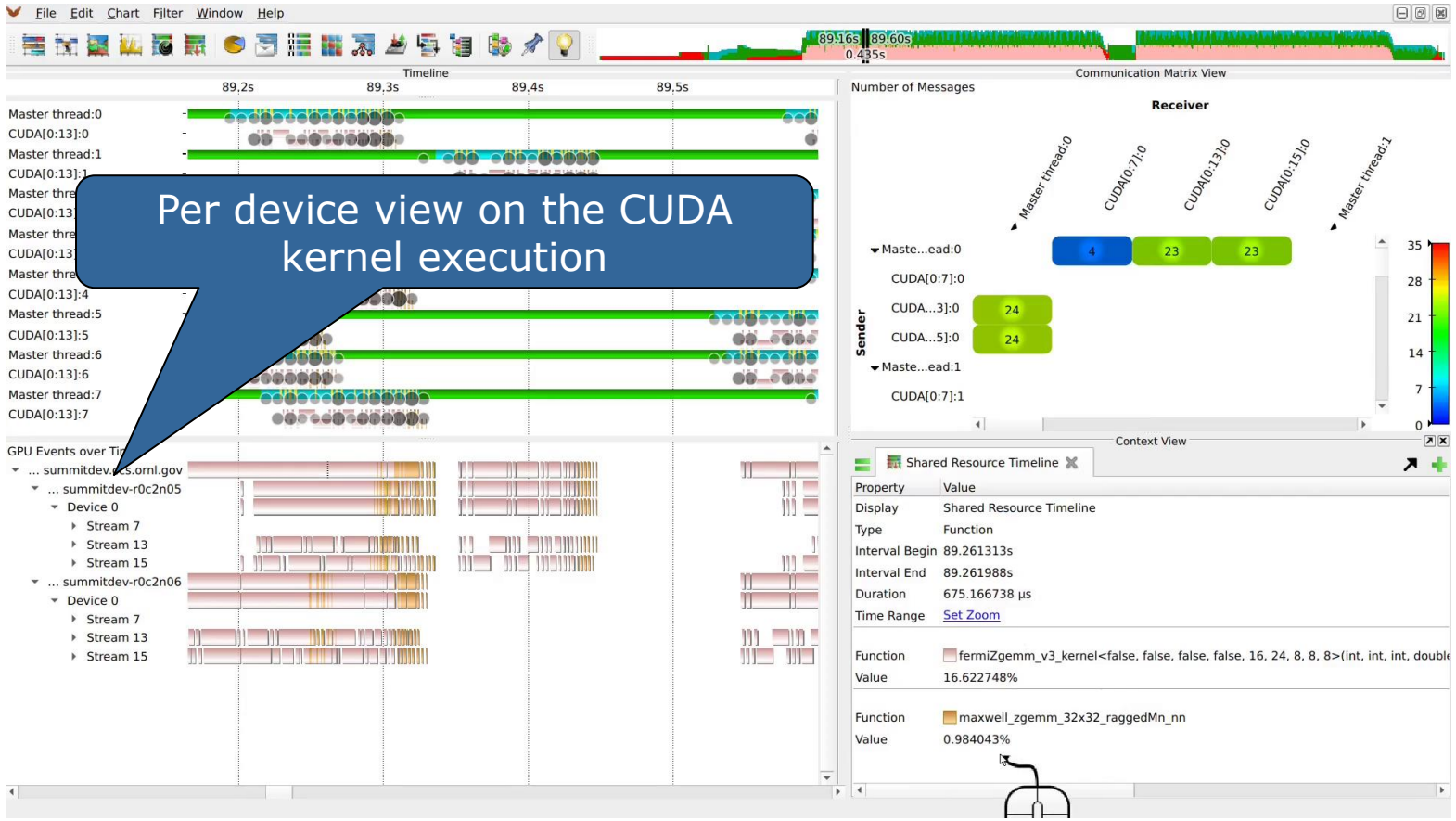
- CUDA streams are child's of the owning Process
- Allows an in-depth analysis of host-device communication

CUDA Application



- Communication Matrix best for analyzing the general communication pattern
- Expectation: balanced communication, represented by a symmetric matrix
- Problem: communication with stream 7 is different

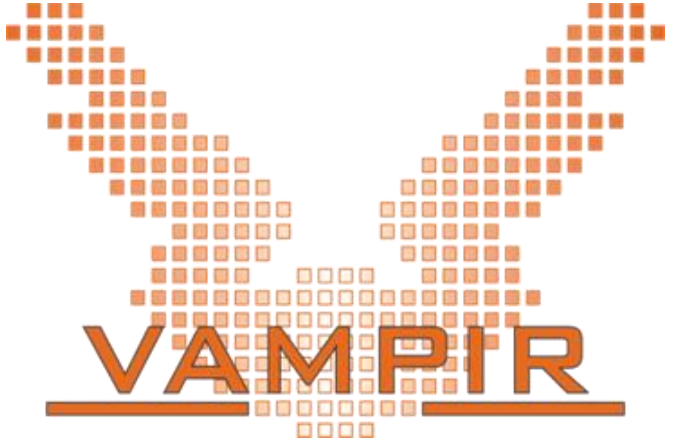
CUDA Application



- Shared Resource Timeline offers a per device view on kernel executions
- Best suited for analyzing multi GPU per node scenarios
- Allows a dedicated analysis of kernel execution patterns
- Yields insights of the actual hardware usage

Summary

- Vampir
  - Interactive trace visualization and analysis of:
    - MPI, OpenMP, CUDA applications
    - File I/O
    - Hardware performance counters
    - (Collective) communication
  - Intuitive browsing and zooming
  - Available for Linux, Windows, and macOS
- VampirServer
  - Scalable to large trace data sizes (20 TiByte)
  - Scalable to high parallelism (200,000 processes)



https://vampir.eu/