TECHNISCHE UNIVERSITÄT DRESDEN

RWTH AACHEN UNIVERSITY

Center for Information Services and High Performance Computing (ZIH)

Session 4: Advanced Use Cases and Discussion

Dr. Matthias S. Müller (RWTH Aachen University)
Tobias Hilbrich (Technische Universität Dresden)
Joachim Protze (RWTH Aachen University, LLNL)

Email:
mueller@itc.rwth-aachen.de
tobias.hilbrich@tu-dresden.de
protze@itc.rwth-aachen.de

ZIH
Center for Information Services & High Performance Computing

Content

- MUST
  - Frontend/Backend
  - Application Crash Handling
  - Scalability
  - Integrations
- Vampir
  - Tracing at Large Scale
  - Paradigms: Xeon Phi, OpenSHMEM
  - External Energy Counters
  - GPGPU Critical Paths

Vampir – Advanced (Select Topics)

Scalability
- Jaguar 220K processes
- SIONlib

Paradigms
- XeonPhi
- OpenSHMEM

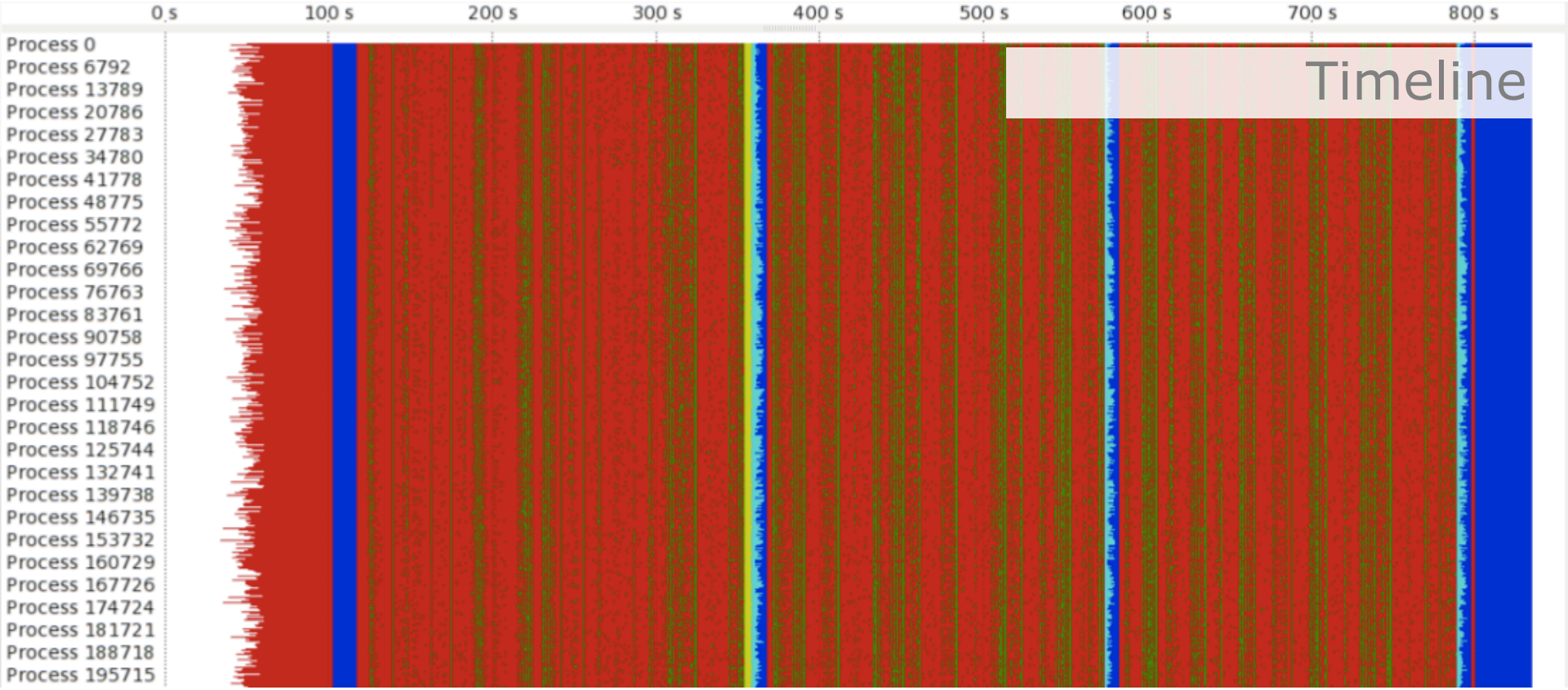
Energy
- Cray XC30 Counters

Evaluation
- GPGPU Critical paths

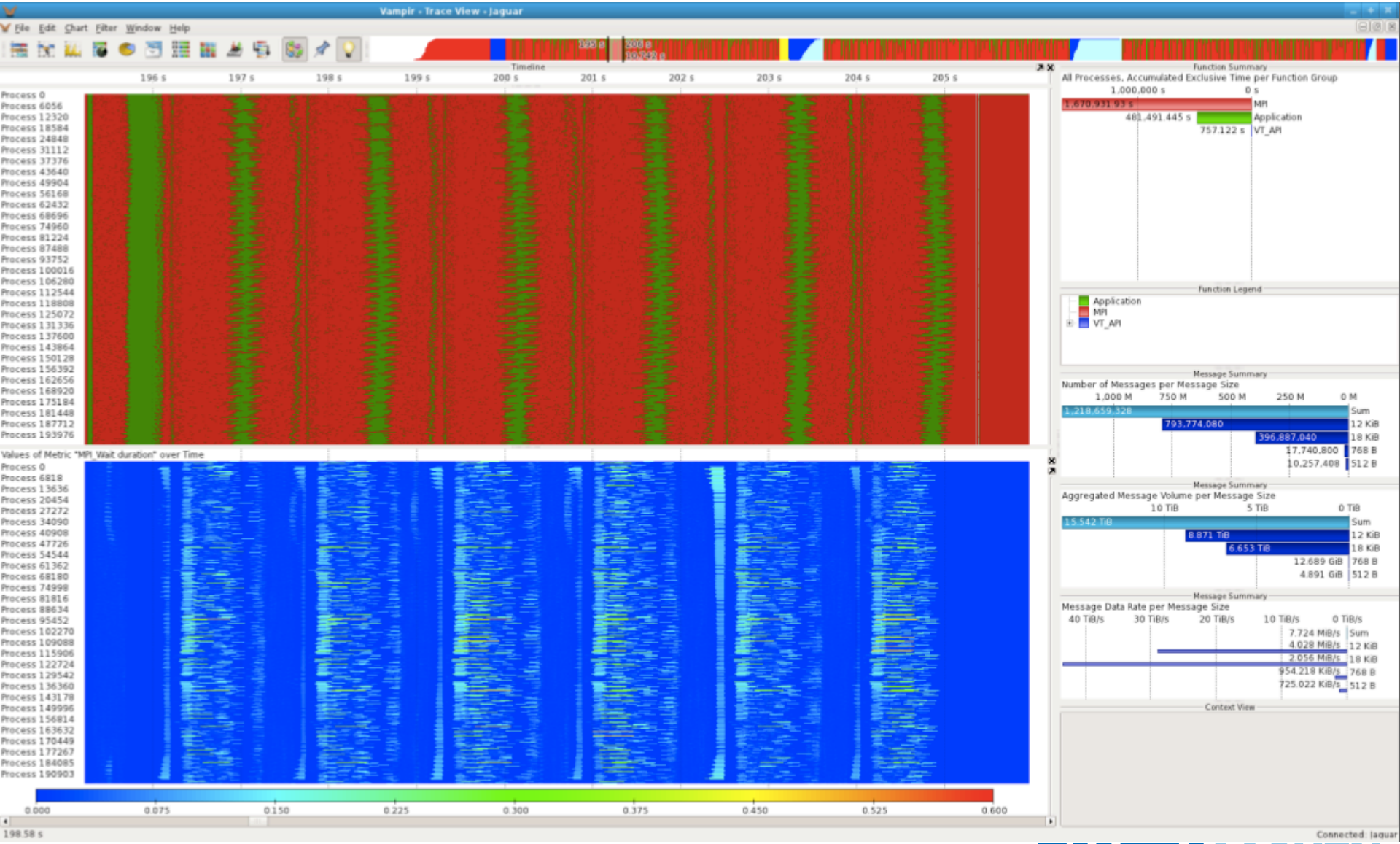
Vampir: Scalability – Collaboration with ORNL

- Goal:
  - Support hybrid system and software architecture at 10 Petascale
  - Show MPI and GPGPU programming
  - Do full system performance profiling **and** tracing
- Facts:
  - Jaguar + Successor
  - >220.000 cores
  - 200,448 monitored MPI processes
  - >20 Tera-bytes of performance data
  - 21,515 VampirServer processes

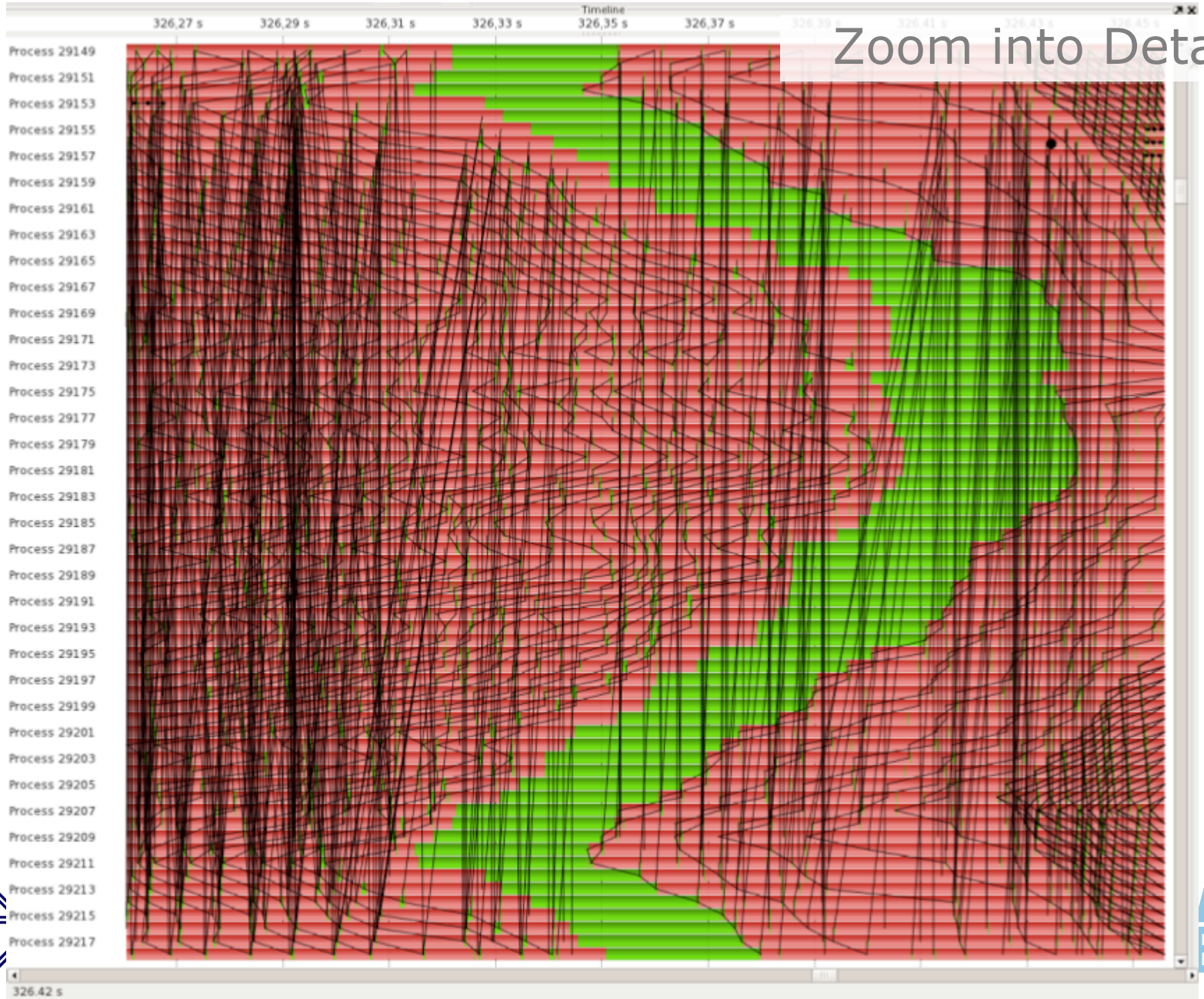
Vampir: Scalability – Collaboration with ORNL



Vampir: Scalability – Collaboration with ORNL



Vampir: Scalability – Collaboration with ORNL



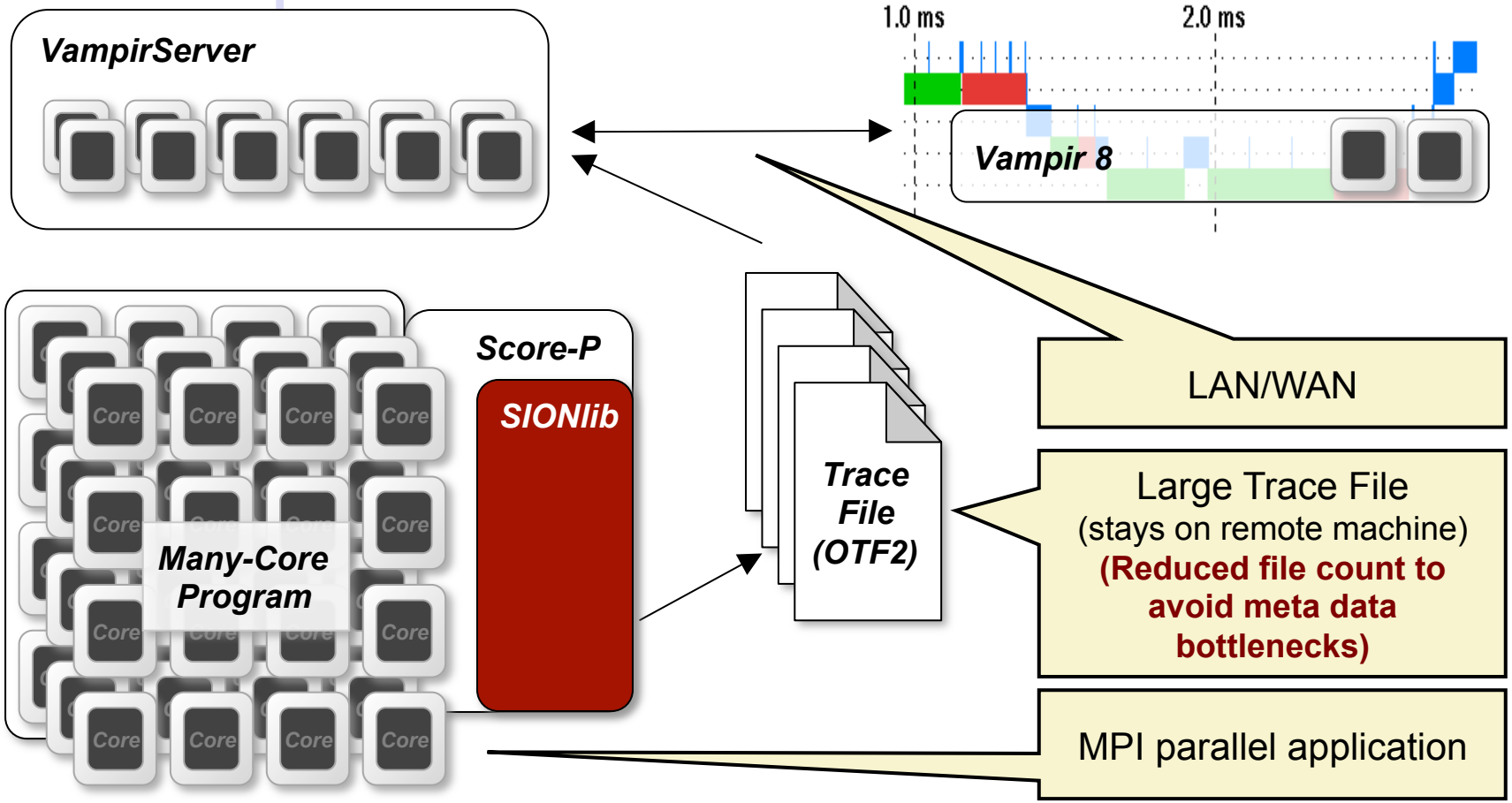
Zoom into Details

Vampir: Scalability – SIONlib with Score-P 1.3

- On local machine with remote VampirServer

```
% module load UNITE vampirserver
% vampirserver start -n 12
```

```
% module load UNITE vampir
% vampir
```



Vampir Scalability: SIONlib with Score-P 1.3

- > 10K processes: Filesystem becomes scalability bottleneck
- SIONlib merges multiple trace files into a single physical file
- Enabled via environmental variables, e.g.:

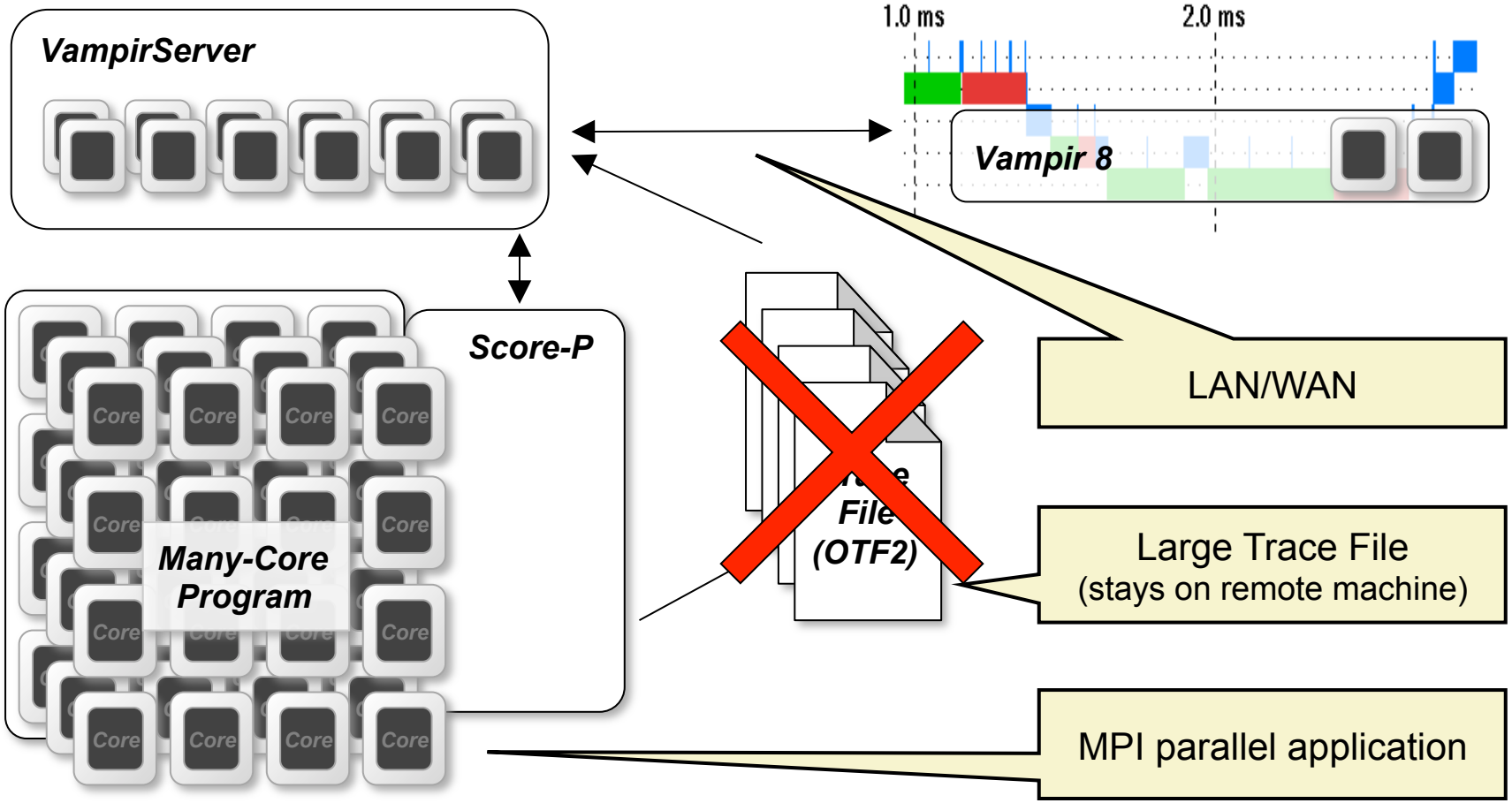
```
% export SCOREP_ENABLE_PROFILING=0
% export SCOREP_ENABLE_TRACING=1
% export SCOREP_TRACING_USE_SION=true
% export SCOREP_TRACING_NLOCATIONS_PER_SION_FILE=64
% export SCOREP_EXPERIMENT_DIRECTORY=scorep_bt_trace
% export SCOREP_FILTERING_FILE=filter.txt
% mpirun …
```

Vampir: Online Analysis Prototype

- Long running applications:
  - ⇒ Either huge traces or little detail
- Exascale:
  - ⇒ Memory/core ratio very low (Straw man, 22MiB)
  - ⇒ Very limited I/O bandwidth per core
- Online tracing workflows avoid these and allow:
  - Increased detail for short time periods (e.g. iteration)
  - Adapt event sources (e.g., add OpenMP tracing)
  - Stop on conditions (e.g., slow iteration, I/O activity)

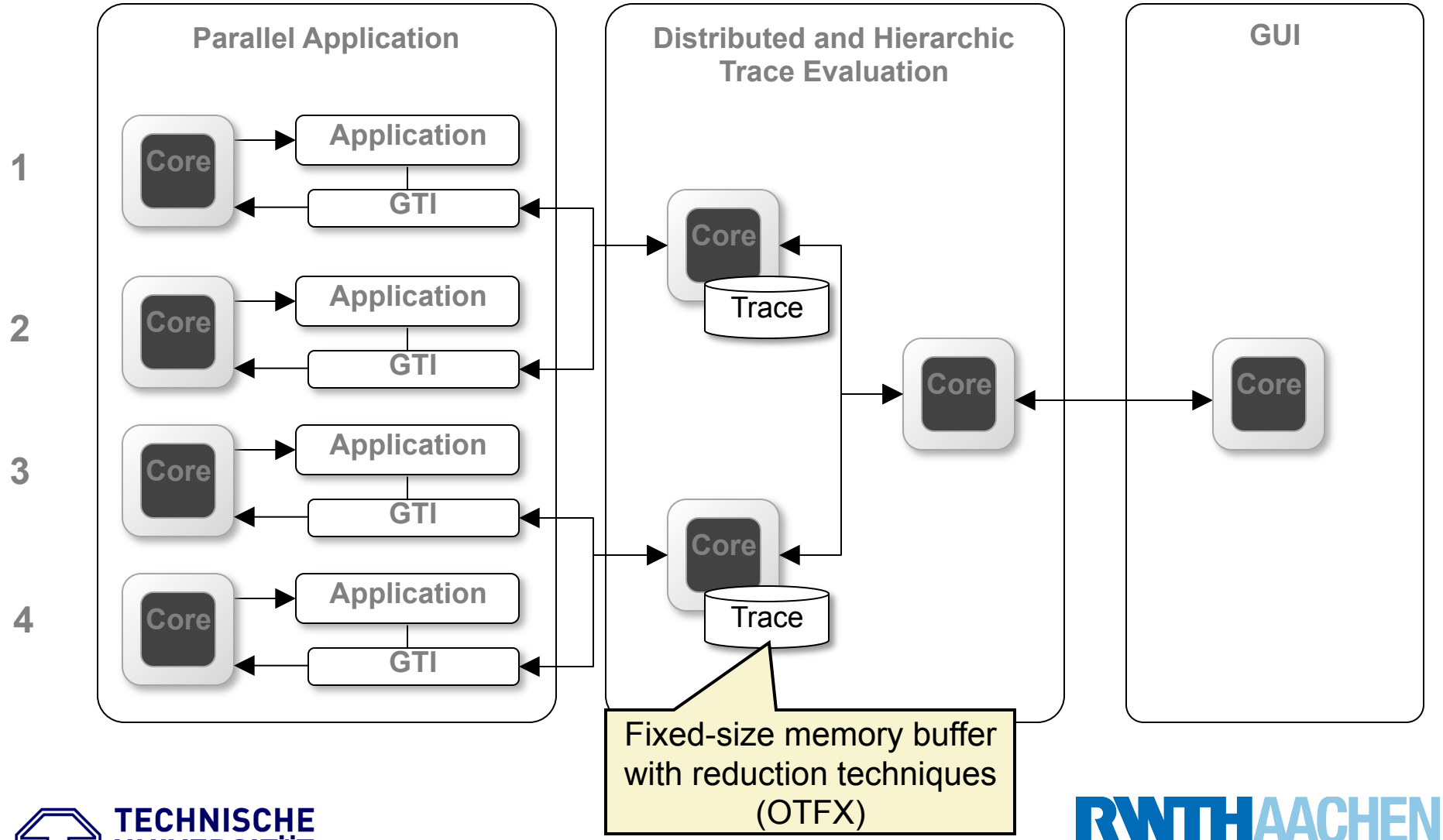
Vampir: Online Analysis Prototype (2)

- Modified workflow



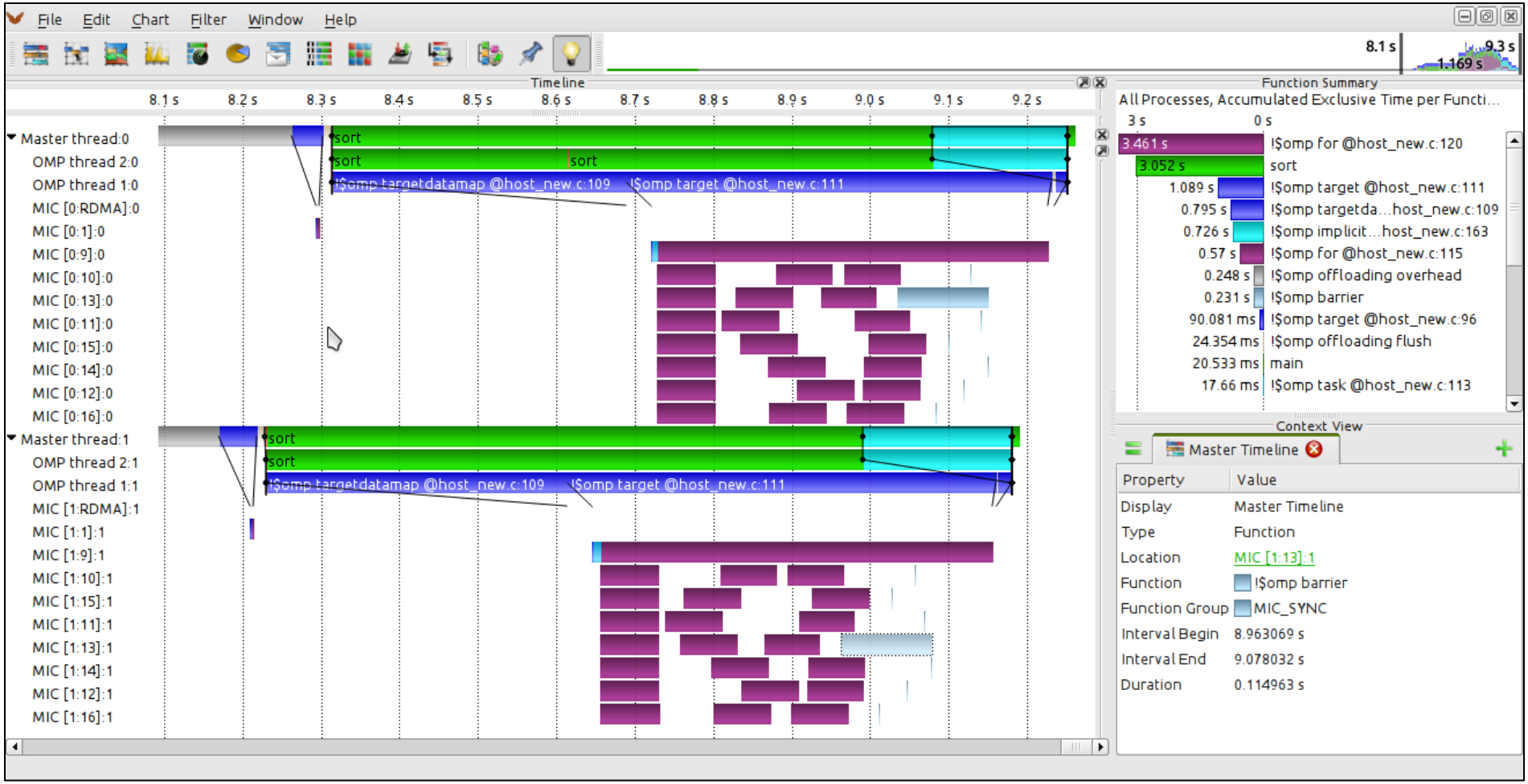
Vampir: Online Analysis Prototype (3)

- With MUST/GTI technology:



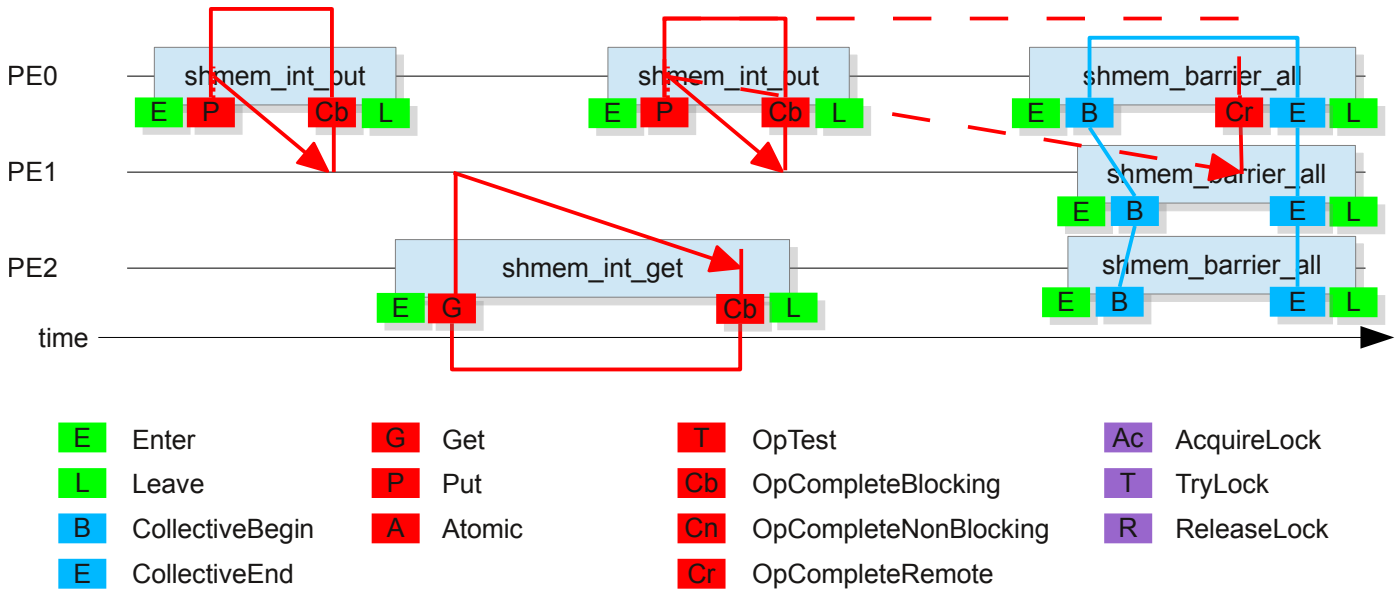
Vampir: Paradigm Support – XeonPhi

- Offloading with OpenMP 4.0 target construct:
  - Support in early Score-P prototypes
  - Employs new OpenMP tools interface OMPT

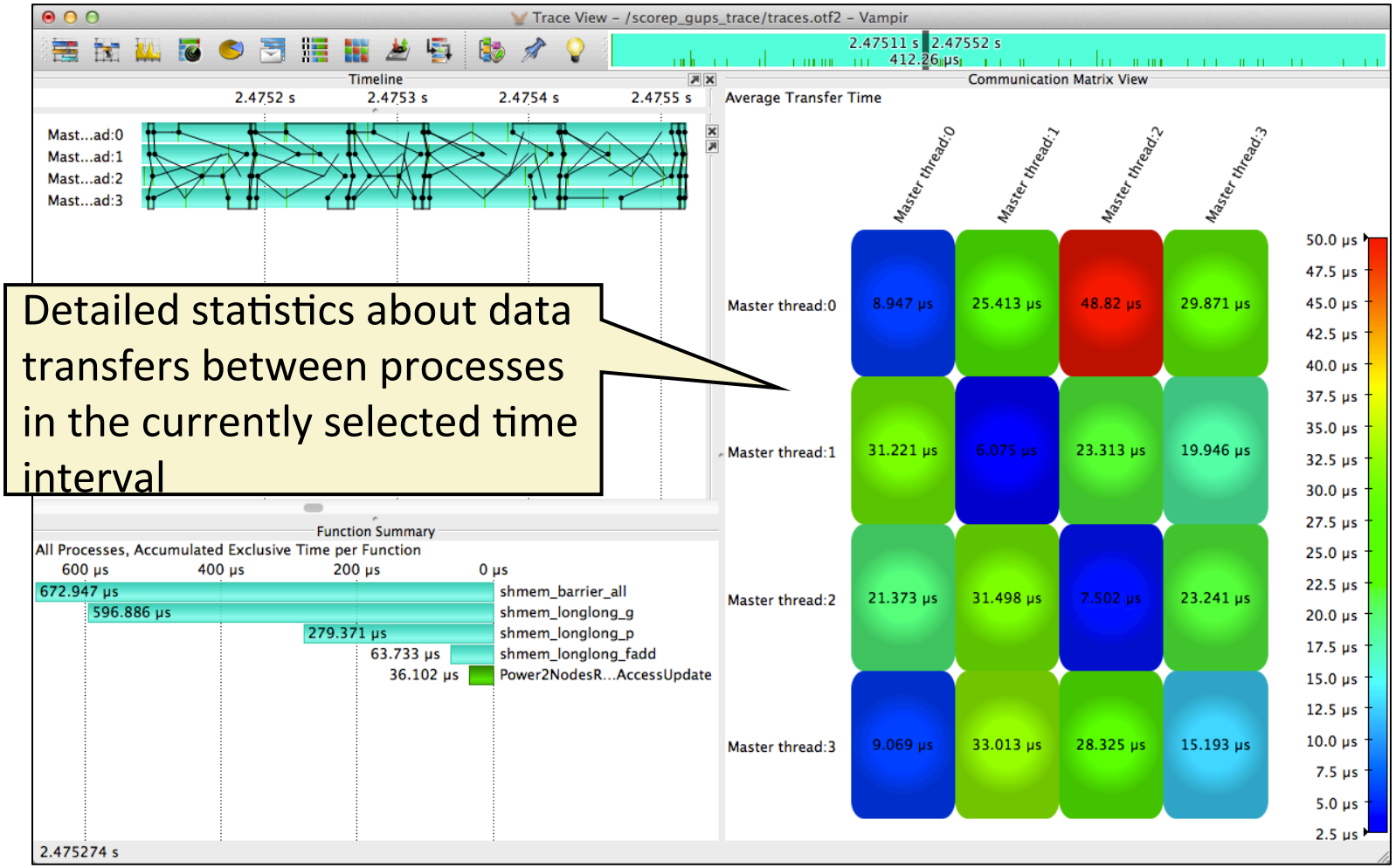


Vampir: Paradigm Support – OpenSHMEM

- One-sided operation analysis and visualization

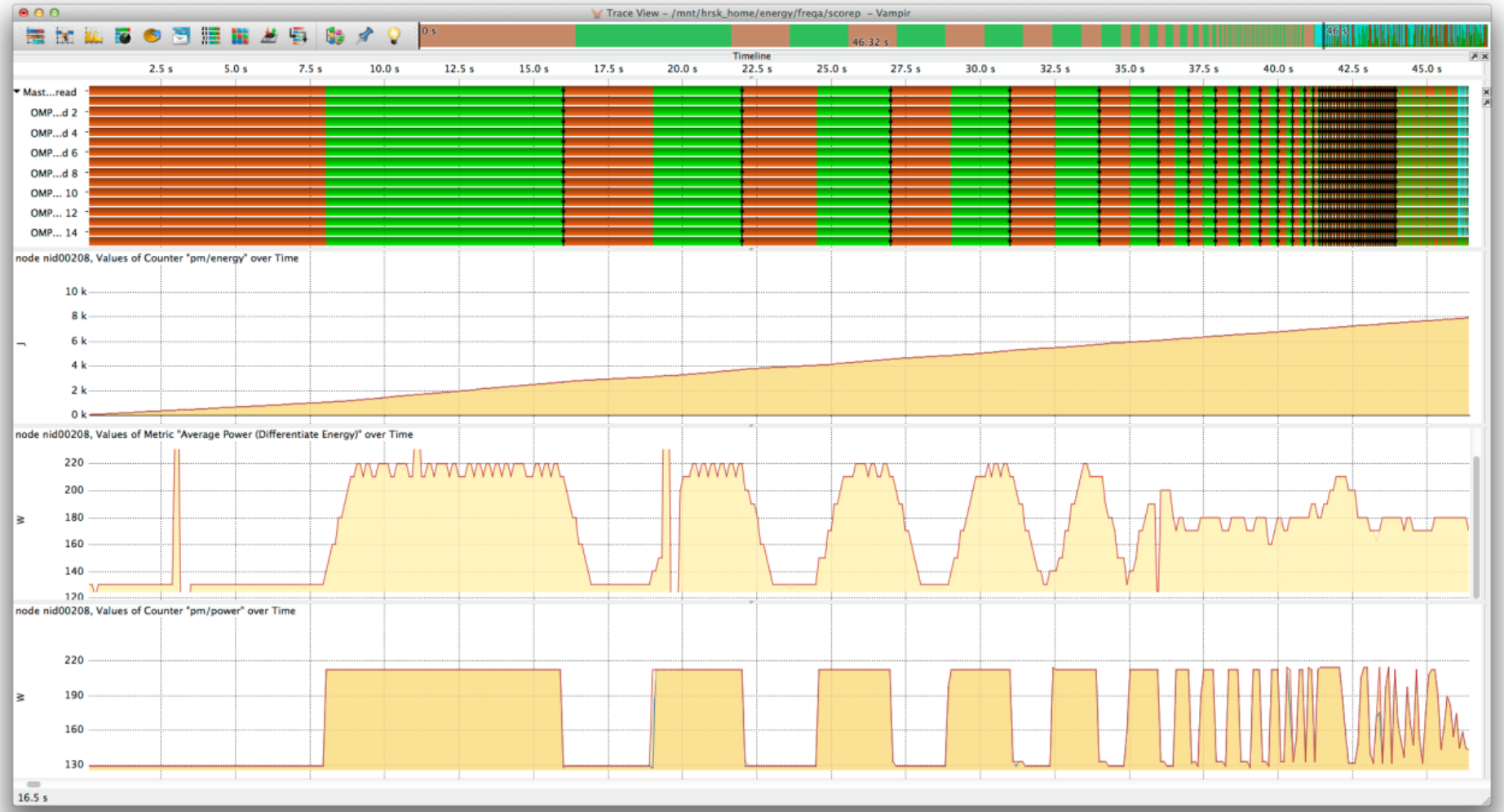


Vampir: Paradigm Support – OpenSHMEM (2)



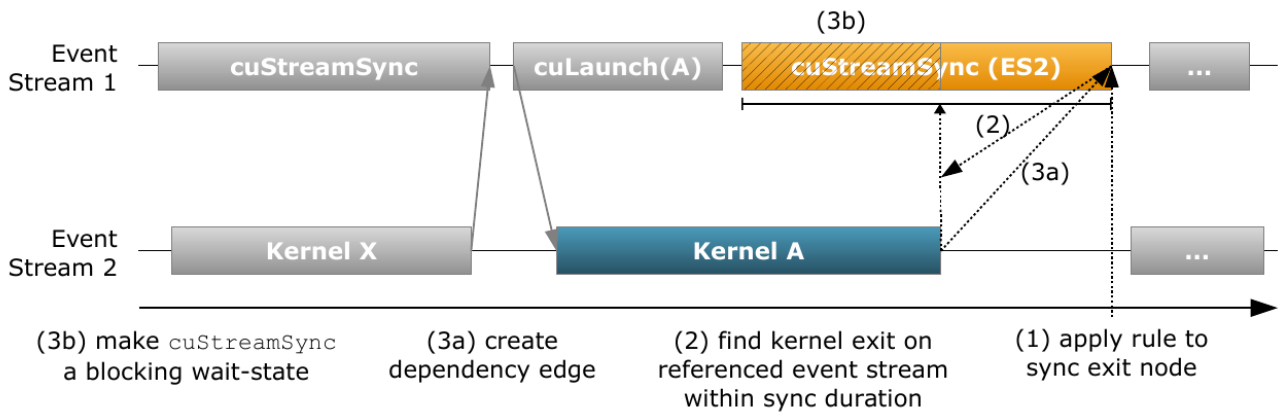
Detailed statistics about data transfers between processes in the currently selected time interval

Vampir: Energy Efficiency – XC30 Counter Support



Vampir: Research in Evaluation – CASITA

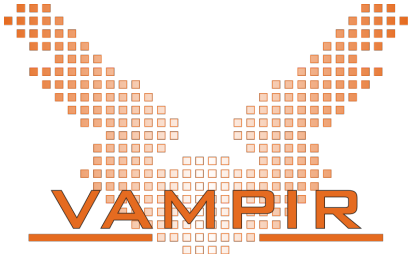
- Identifies critical optimization targets for heterogeneous applications (MPI+OpenMP+CUDA)
- Technique: Post-mortem rule-based trace analysis



BlameKernelRule
Identifies blocking synchronization that is delayed by device activities.

Vampir: Research in Evaluation – CASITA (2)

- CASITA: analysis results as trace counters
  - Waiting time
  - Critical path
  - Blame
- Enables visualization in Vampir
- Example: Hybrid n-body code (PHI-GPU) using MPI, CUDA and OpenMP



Vampir: Research in Evaluation – CASITA Use Case

Analysis results visualized in Vampir *master timeline* and *performance radar*

Critical path (red) over MPI, CUDA and OpenMP

Waiting time distribution

Blame distribution (penalty assigned to wait-state root causes)

Vampir: Research in Evaluation – CASITA Use Case (2)

Critical path (red) on GPU due to blocking cuCtxSynchronize call

Wait-state on CPU stalled waiting for the device

Blame (combined with critical path) identifies optimization targets

Roadmap

November 2014 | June 2015 | Longer term

Vampir:

- Collapsing timelines

Score-P:

- Sampling+tracing
- Xeon Phi (Native+Offload)
- Full SIONlib support (multi-flush, MPI+OpenMP)
- Memory tracing
- CPU ID
- 3rd party library wrapping
- OpenCL support

Score-P:

- Cobi (Dyninst) instrumentation
- I/O tracing
- More MPI-3 support

Vampir:

- Evaluation of traces with semantic compression

Score-P:

- Generation of traces with semantic compression