TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Application performance measurement towards Exascale using Score-P and Vampir

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de

ZIH
Center for Information Services & High Performance Computing

Content

- Overview– Challenges towards Exascale
- Performance Monitoring with Score-P
- Performance Analysis with Vampir
- Conclusions & Wrap-Up

Challenges towards Exascale

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de

Challenges

- HPC systems evolve; Tools need to adapt:

MPI + Accelerator + Threads

Heterogeneous / Hybrid

MPI + PGAS ?

MPI + CUDA PGAS + CUDA MPI + Threads

Paradigms

Co-Array Fortran

SHMEM

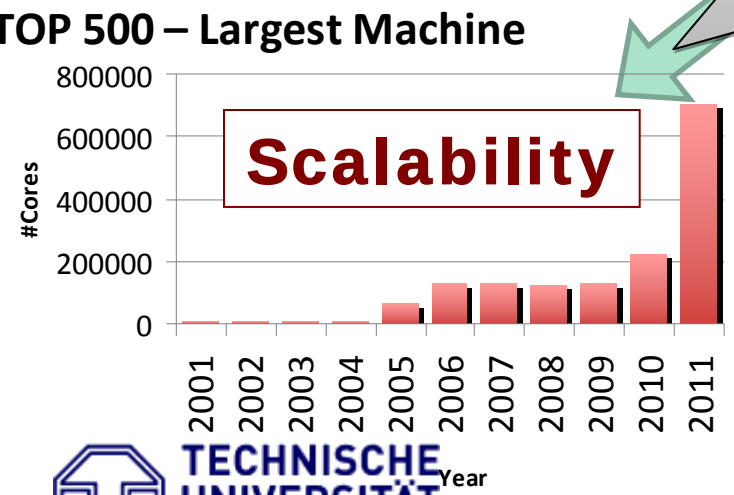
OpenCL

Challenges

Fault Tolerance

Towards Exascale ...

Scalability



Usability

Make all this accessible!

TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Performance Monitoring with Score-P

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de

ZIH
Center for Information Services & High Performance Computing

Score-P

- Community effort for a common performance tools infrastructure
  - Score-P instrumentation and measurement system
  - Common data formats OTF2 and CUBE4
- Save efforts for new features, scalability, support, training, porting, QM
- Improve tool interoperability and data exchange
- Single learning curve for users, single installation, single measurment run
- Currently works with Vampir, Scalasca, TAU, and Periscope
- Initial funding:
  - SILC and LMAC projects, German BMBF
  - PRIMA project, US DOE
- On-going Open Source project, open for new partners/tools

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

See http://www.score-p.org/

Score-P – Overview

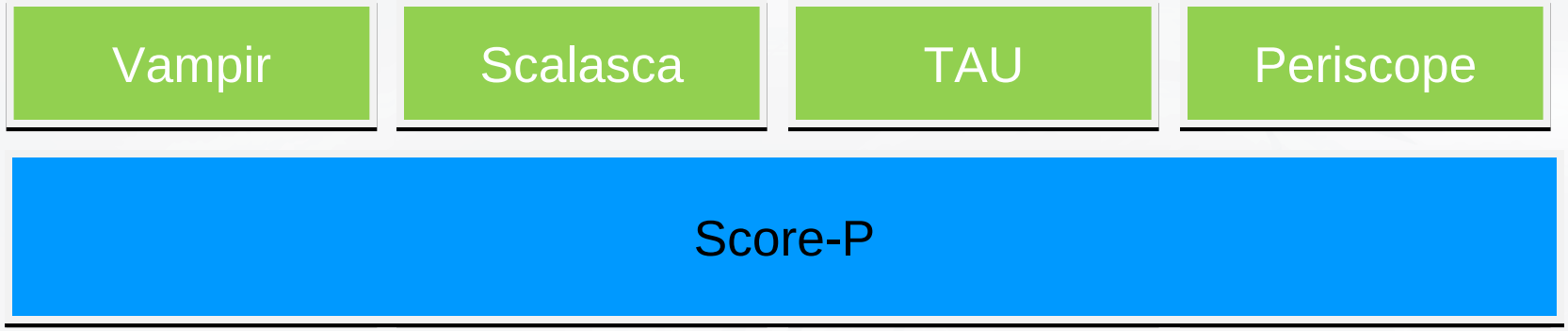
- Tool ecosystem:

Vampir	Scalasca	TAU	Periscope
VampirTrace OTF	EPILOG / CUBE	TAU native formats	Online measurement

- Consequence:
  - Each tool requires its own measurement system
  - Tool combinations limited
  - Higher tool development effort

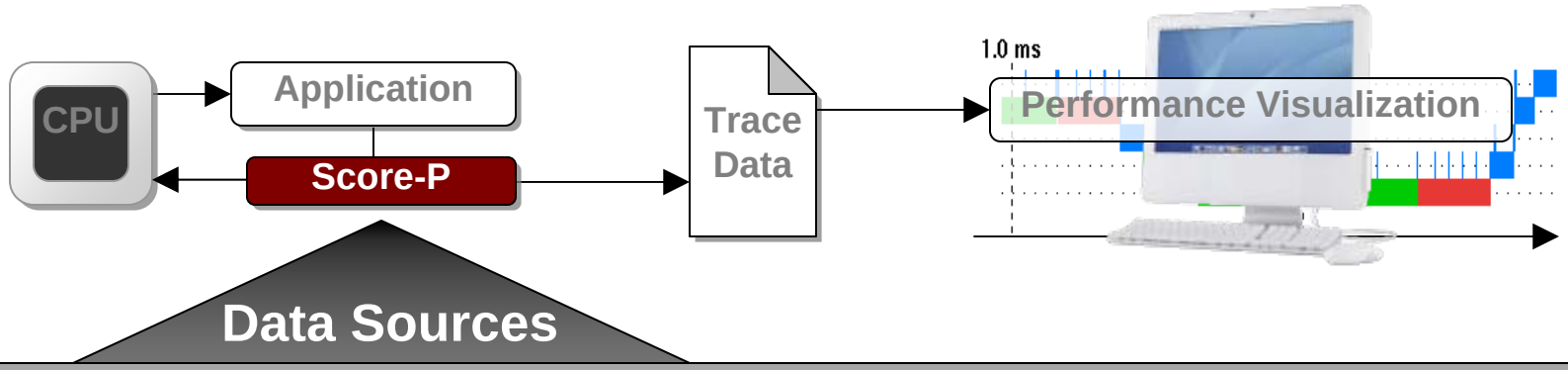
Score-P – Overview

- Tool ecosystem:



- Goals:
  - Single measurement system
  - Better learning curve
  - More tool integration
  - Reduced tool development costs

Tracing – Instrumentation



Data Sources

Source level
- Compiler
- Manual
- OpenMP with Opari
- Tau PDT

Binary level
- Dyninst

MPI Profiling Interface
- MPI

Runtime/Library
- MPI Correctness
- Pthreads
- NVIDIA CUDA
- OpenCL
- 3rd party libraries
- External counters
- Plugin counters
- Java tracing

Operating System
- Resource usage
- Memory allocation

Hardware
- Performance counters
- I/O
- CPU ID

Score-P – State

- Preview versions since 2010
- First release December 2011
- Score-P will replace VampirTrace
- Feature migrations are ongoing:
  - CUDA support on its way
  - TAU PDT support available
  - OPARI2 (in Score-P) supersedes OPARI in VT
  - …
- Vampir Tutorials will use Score-P starting mid 2012

TECHNISCHE UNIVERSITÄT DRESDEN

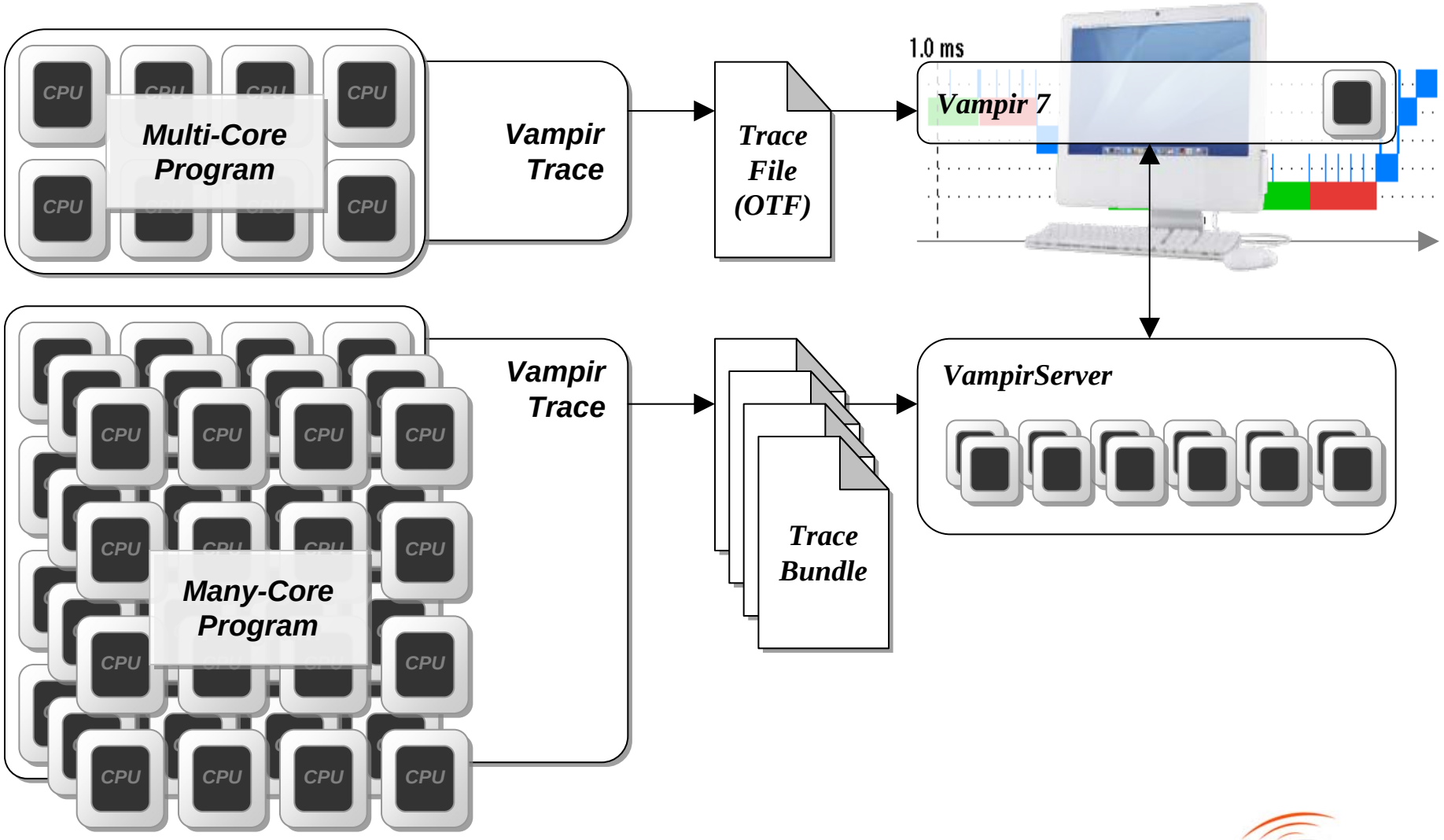
Center for Information Services and High Performance Computing (ZIH)

Performance Analysis with Vampir

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de

ZIH
Center for Information Services & High Performance Computing

Technology


CPU
Multi-Core Program
Vampir Trace
Trace File (OTF)
1.0 ms
Vampir 7
Many-Core Program
Vampir Trace
Trace Bundle
VampirServer

Meeting the Challenges

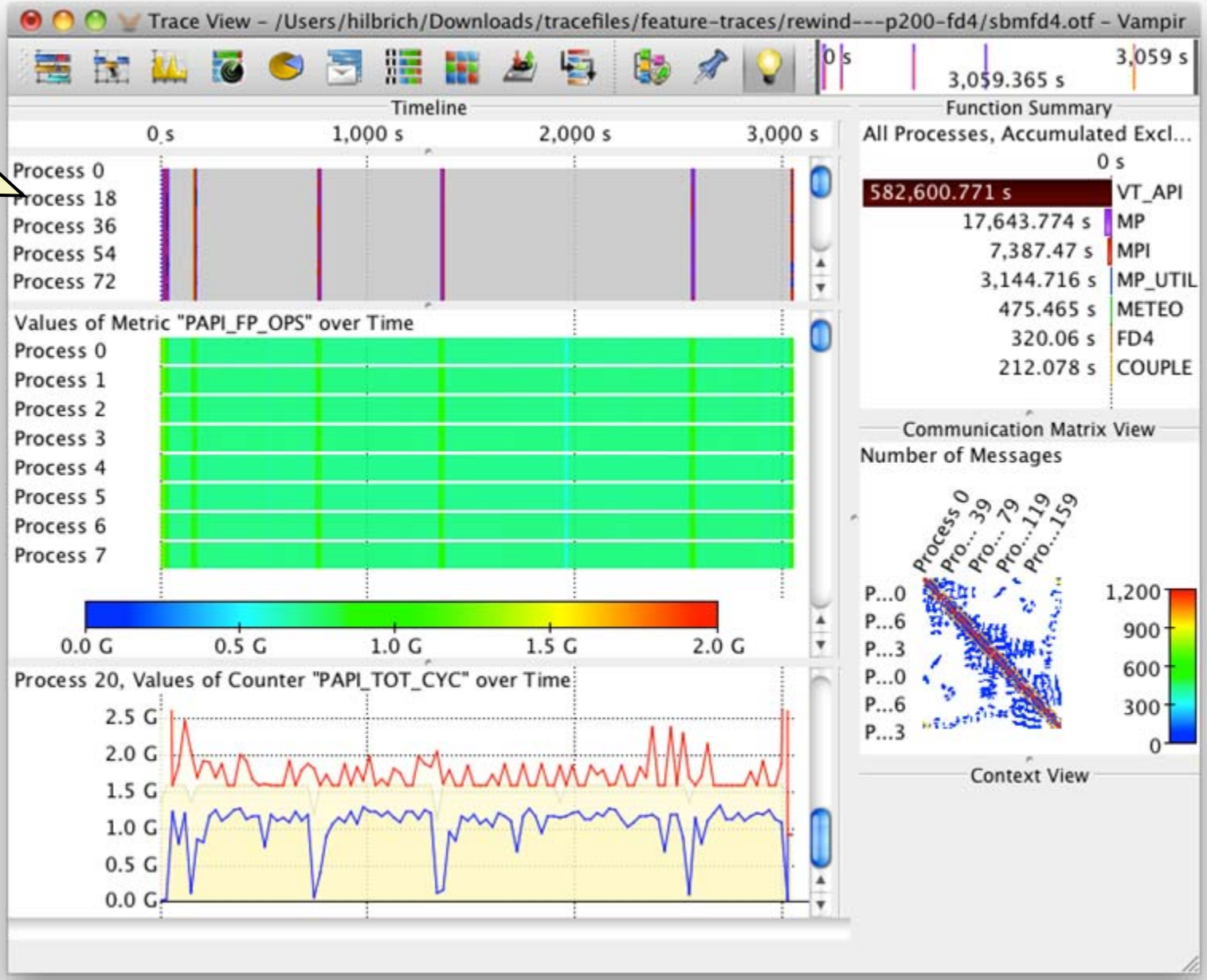
- HPC systems evolve; tools need adapt:
  - Scalability
  - Paradigms
  - Heterogeneity
  - Fault tolerance
  - Usability
- Several extensions address these challanges

Meeting the Challenges – Long Tracing Runs (1)

- Tracing long running applications is hard
- Case:
  - Long running application (~50min)
  - Non-reproducible bad performance in iterations
  - Coarse-grained traces gave no insight
  - Fine-grained traces would not fit into memory
- Solution:
  - VampirTrace feature “Rewind”
  - Allows to discard parts of the trace buffer
  - Controlled via VampirTrace instrumentation API

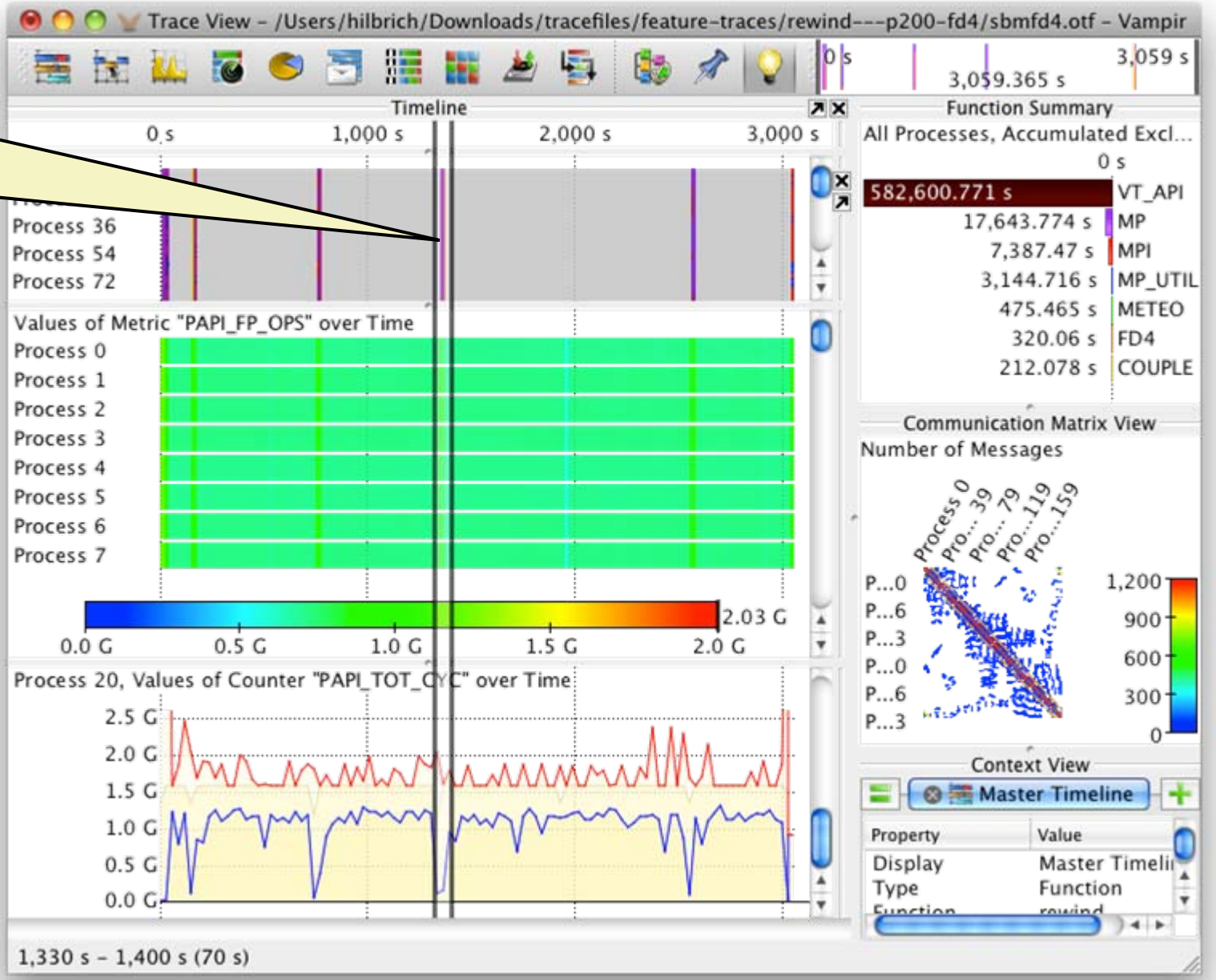
Meeting the Challenges – Long Tracing Runs (2)

First and last iterations
+ 4 “slow” iterations



Meeting the Challenges – Long Tracing Runs (3)

Zoom into a slow iteration

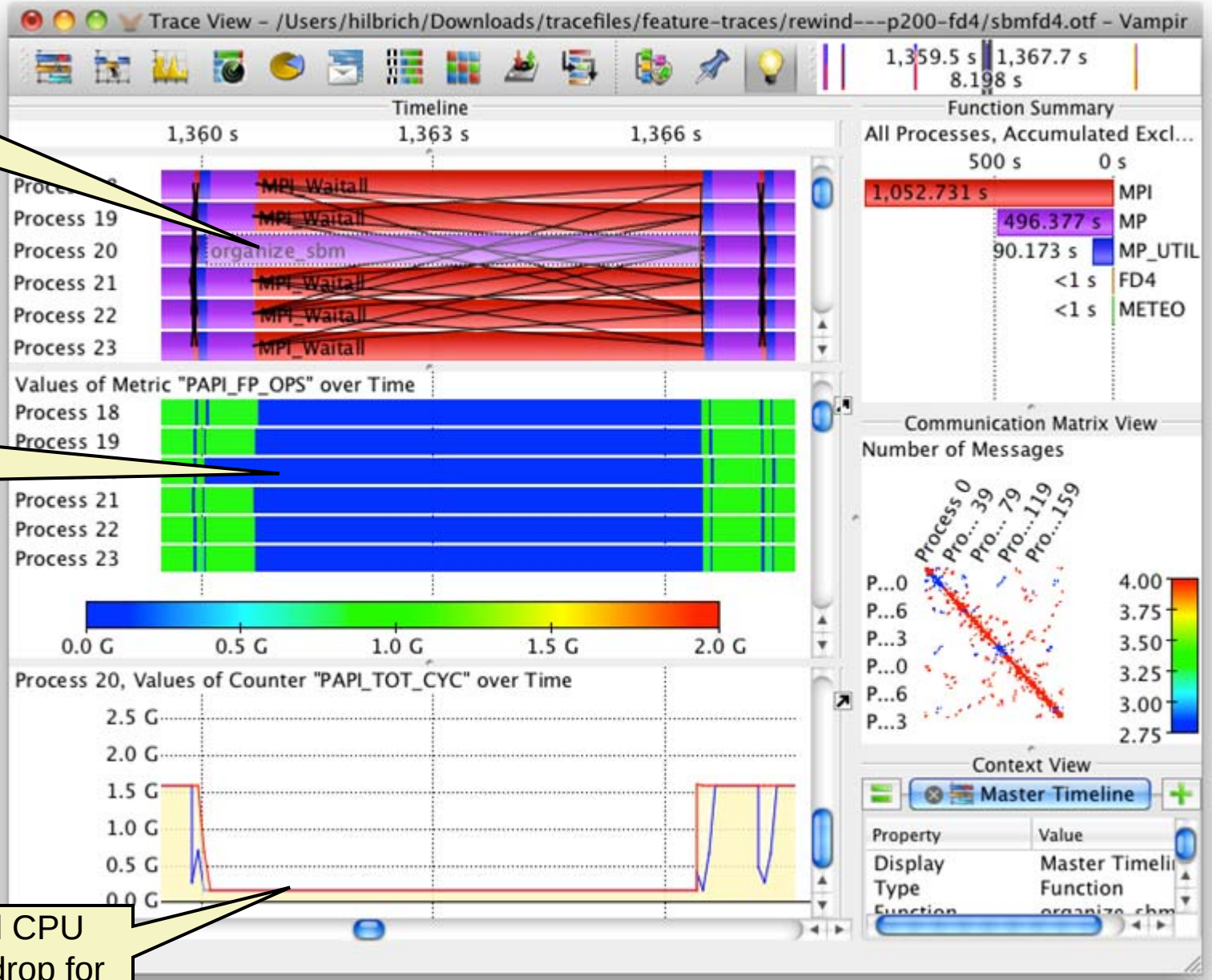


Meeting the Challenges – Long Tracing Runs (4)

All except rank 20 need to wait

Can also be spotted in Radar

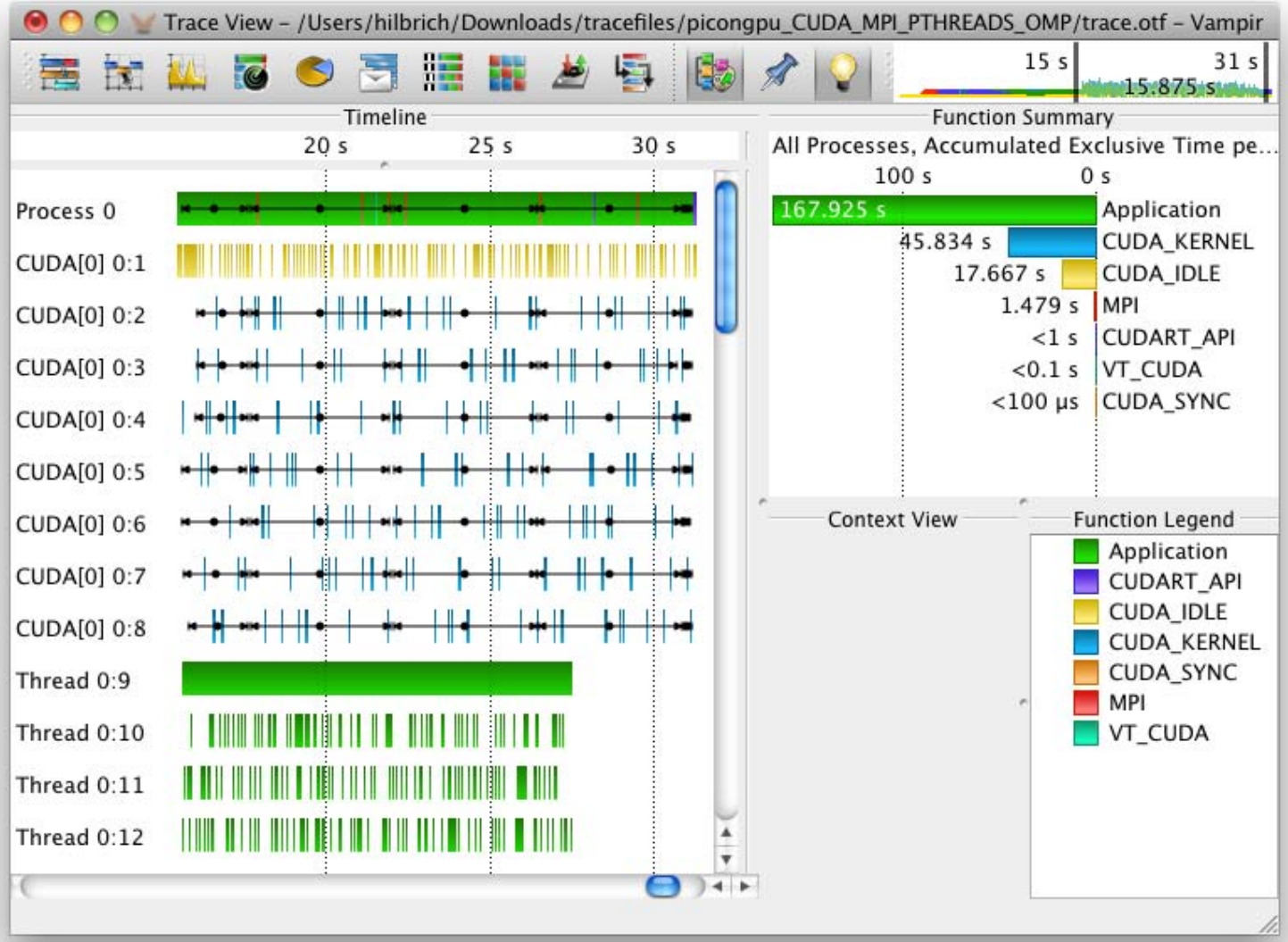
Process 20 total CPU cycles suddenly drop for > 6 seconds



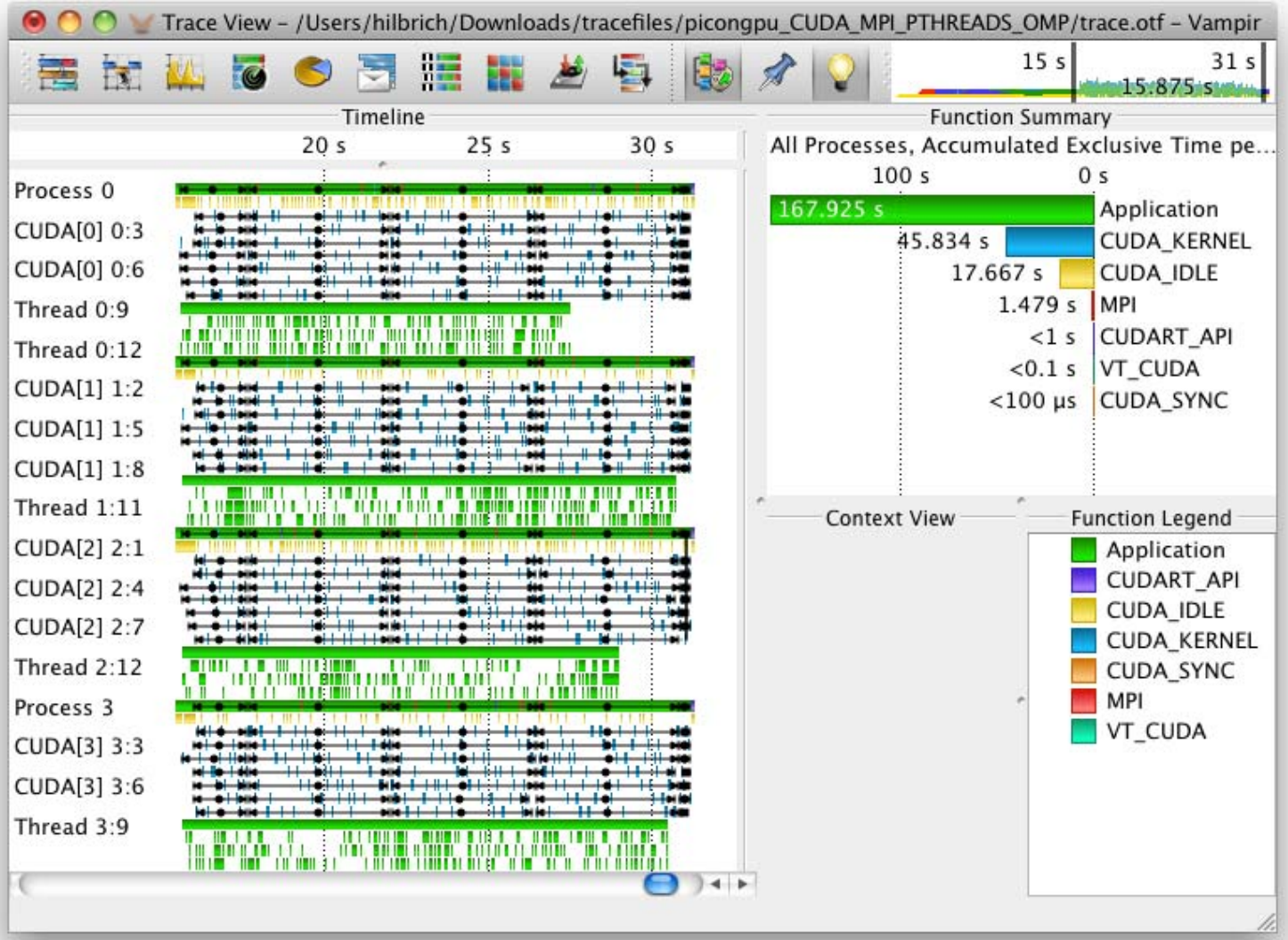
Meeting the Challenges – CUDA, PicOnGPU

- Case:
  - Particle in cell code on NVIDIA GPGPUs
  - Multi-level parallelism: MPI + CUDA + pthreads
  - Performance is a black box without tools
- Solution:
  - VampirTrace support for CUDA
  - Close collaboration with NVIDIA

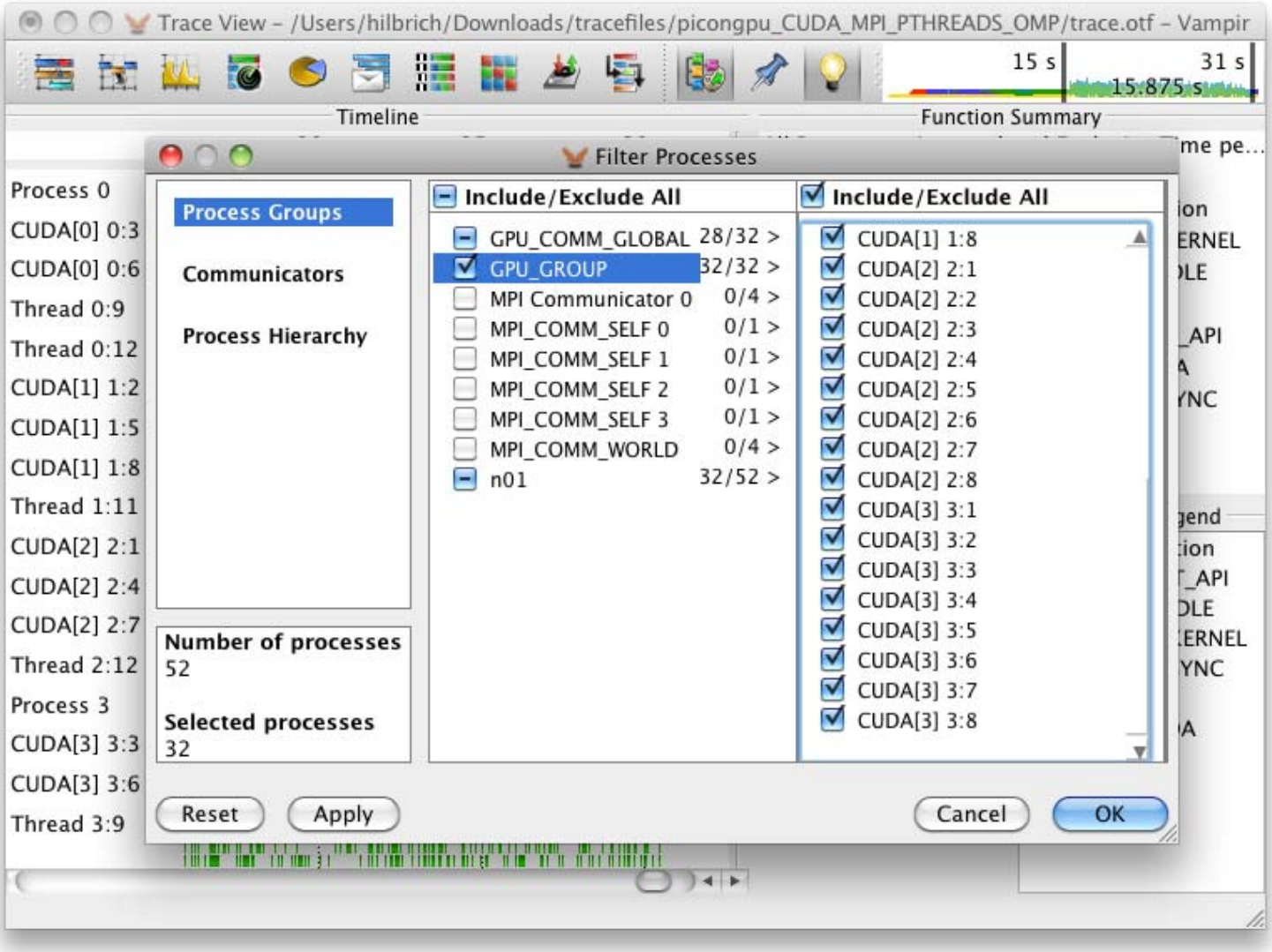
Meeting the Challenges – CUDA, PicOnGPU



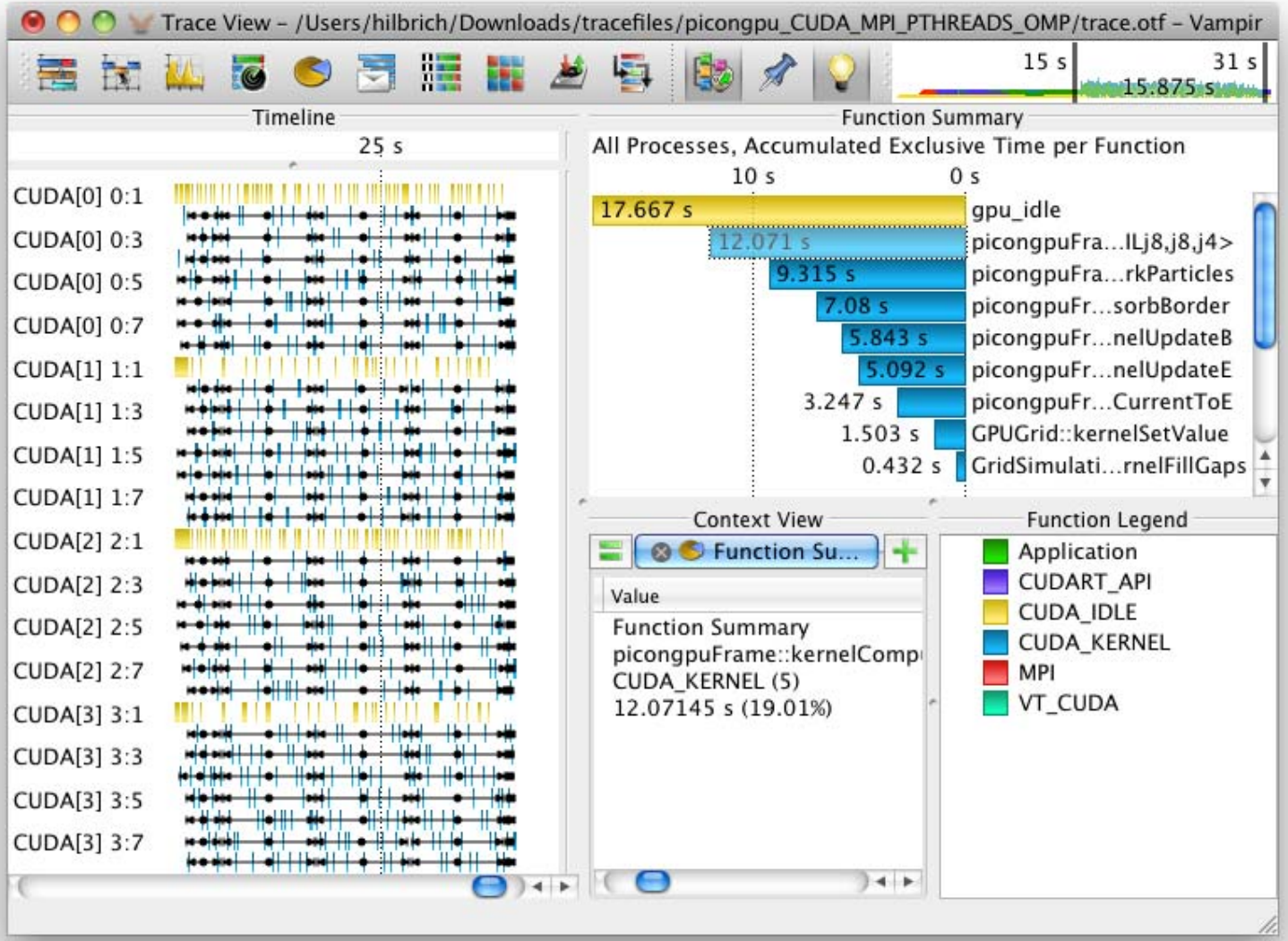
Meeting the Challenges – CUDA, PicOnGPU



Meeting the Challenges – CUDA, PicOnGPU



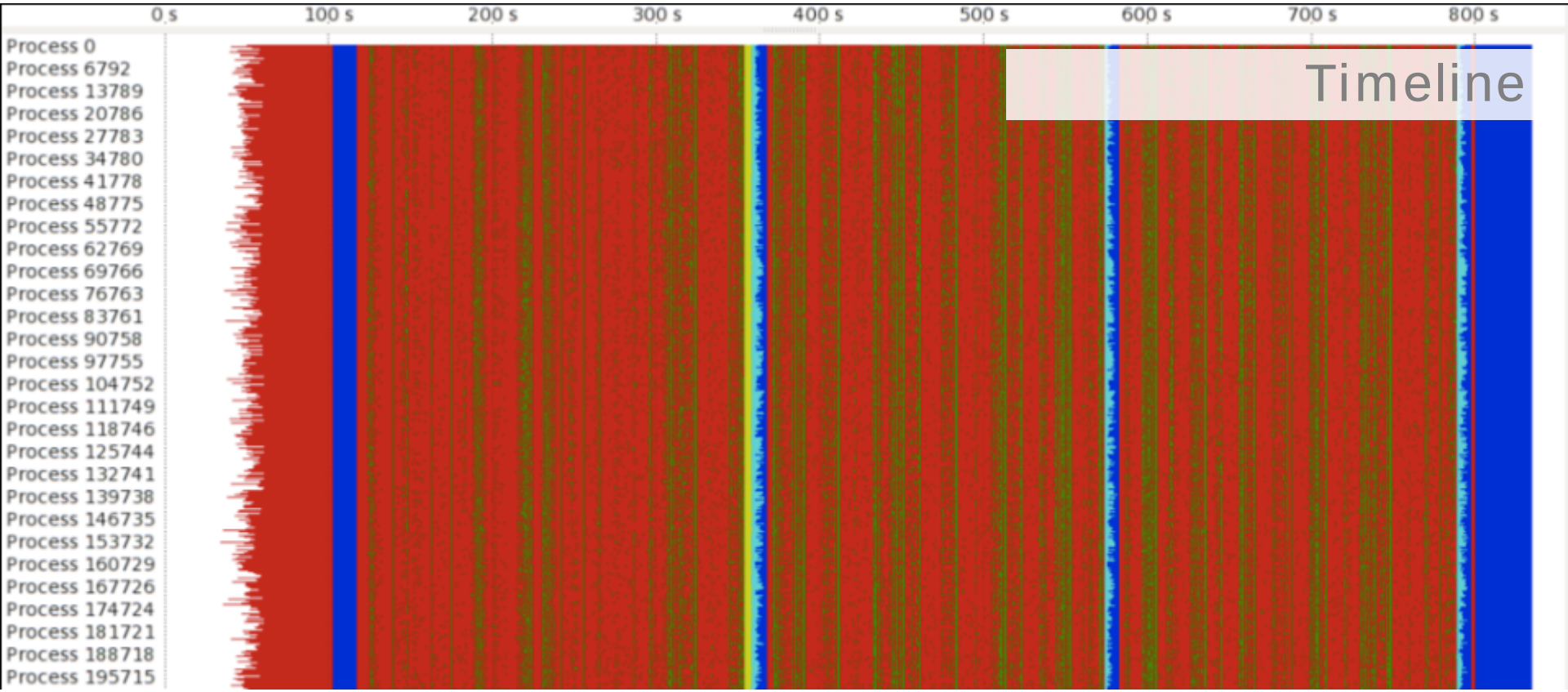
Meeting the Challenges – CUDA, PicOnGPU



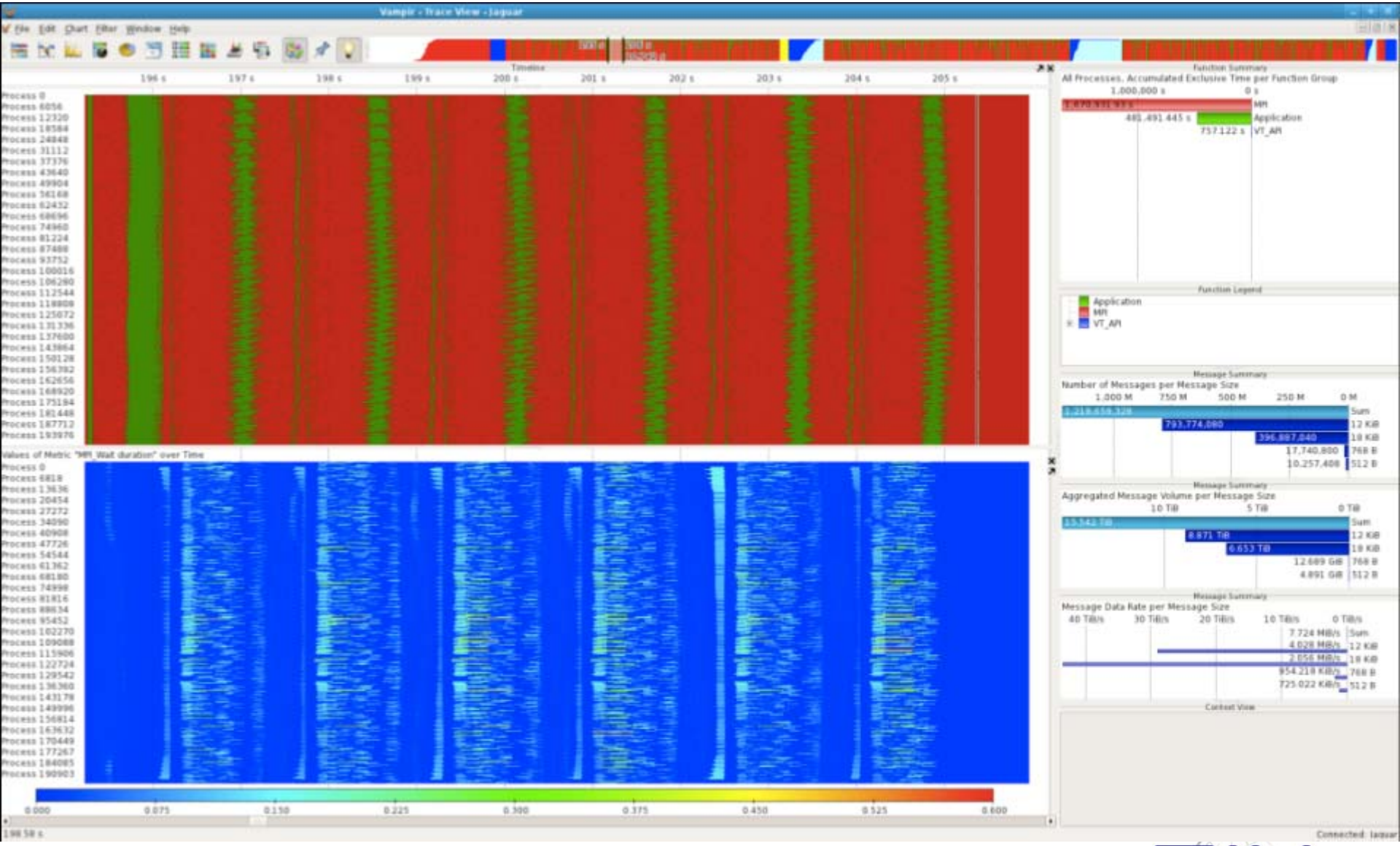
Meeting the Challenges – Collaboration with ORNL

- 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

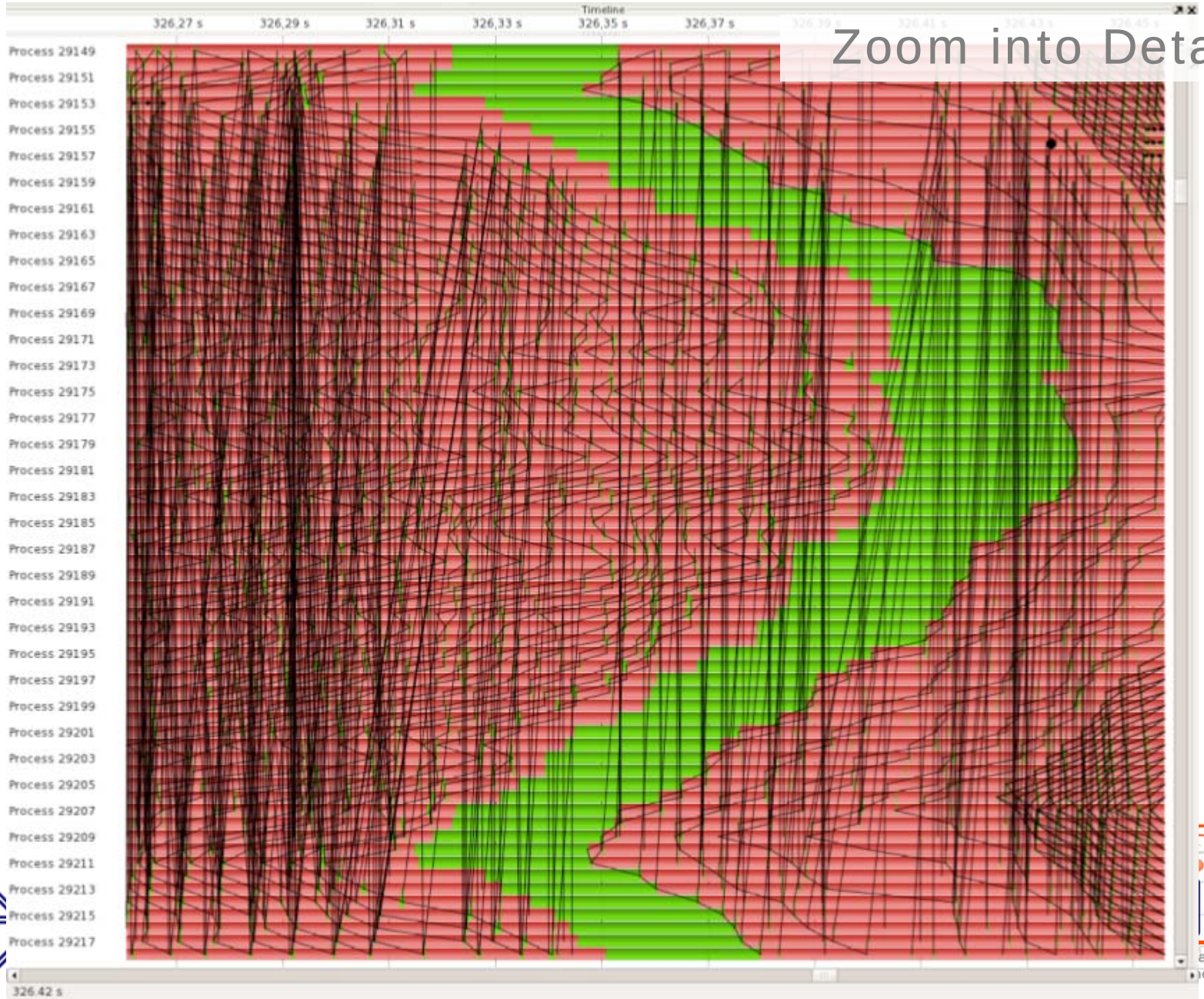
Meeting the Challenges – Collaboration with ORNL



Meeting the Challenges – Collaboration with ORNL



Meeting the Challenges – Collaboration with ORNL

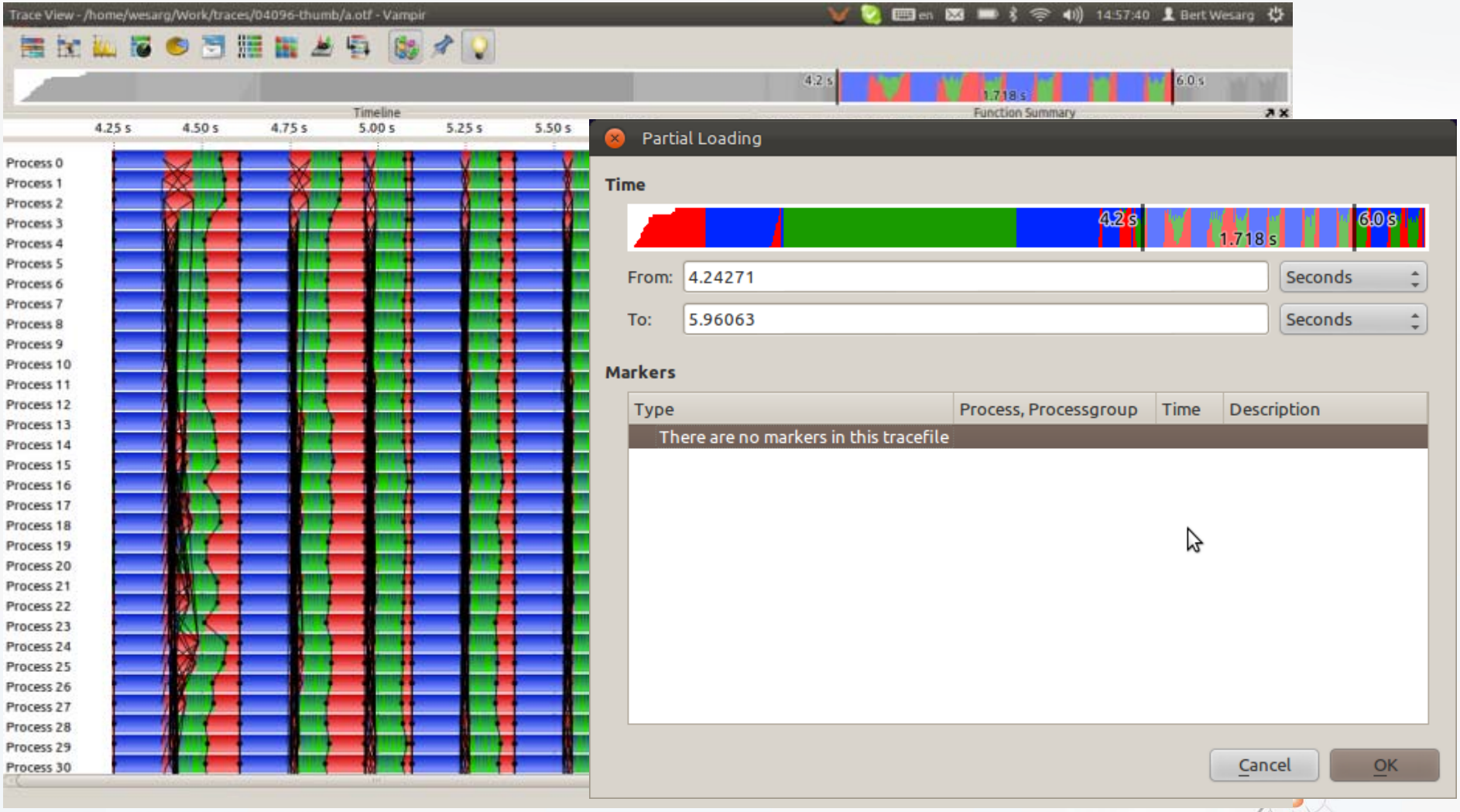


Work in Progress

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de

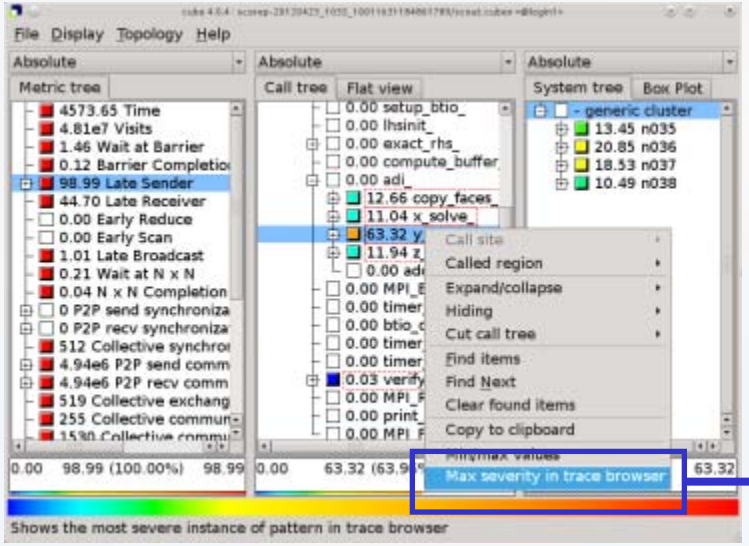
Vampir Partial Loading

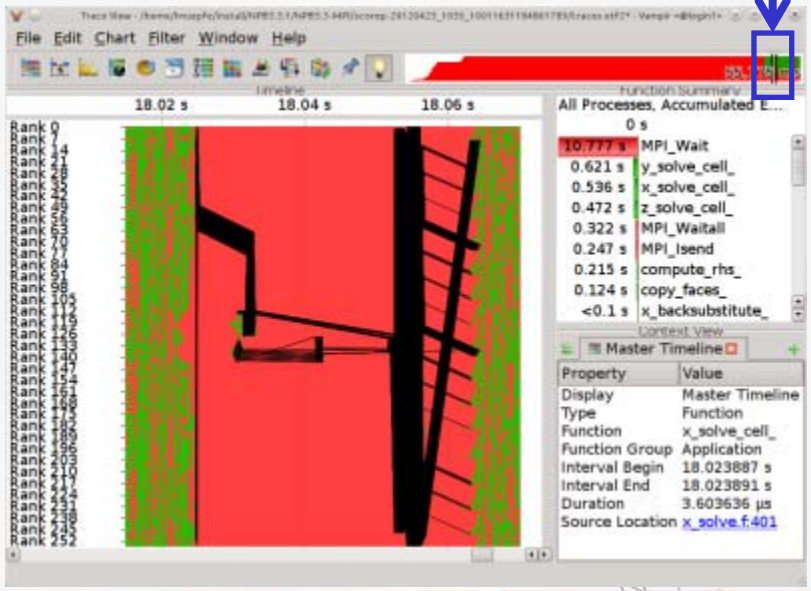
- Selectable time interval to load, save analysis resources



Scalasca-Vampir Interoperation

- Scalasca's CUBE GUI calls Vampir GUI
- Open worst instance of selected performance problem
  - Start Vampir or use existing instance and open trace
  - Zoom to time interval of interest
  - Control via DBus, allow multiple successive operations
  - Requires '-i' option for Scalasca (**scout –i**)





Scalasca-Vampir Interoperation

cube 4.0.4 : scorep-20120423_1035_10011631184861789/scout.cubex <@login1>

File Display Topology Help

Absolute | Absolute | Absolute

Metric tree | Call tree | Flat view | System tree | Box Plot

- 4573.65 Time
- 4.81e7 Visits
- 1.46 Wait at Barrier
- 0.12 Barrier Completio
- 98.99 Late Sender
- 44.70 Late Receiver
- 0.00 Early Reduce
- 0.00 Early Scan
- 1.01 Late Broadcast
- 0.21 Wait at N x N
- 0.04 N x N Completion
- 0 P2P send synchroniza
- 0 P2P recv synchroniza
- 512 Collective synchro
- 4.94e6 P2P send comm
- 4.94e6 P2P recv comm
- 519 Collective exchang
- 255 Collective commun
- 1530 Collective commu

0.00 98.99 (100.00%) 98.99

- 0.00 setup_btio_
- 0.00 lhsinit_
- 0.00 exact_rhs_
- 0.00 compute_buffer
- 0.00 adi_
  - 12.66 copy_faces_
  - 11.04 x_solve_
  - 63.32 y_
  - 11.94 z_
  - 0.00 ad
- 0.00 MPI_B
- 0.00 timer
- 0.00 btio_c
- 0.00 timer
- 0.00 timer
- 0.03 verify
- 0.00 MPI_R
- 0.00 print_
- 0.00 MPI_F

0.00 63.32 (63.96%

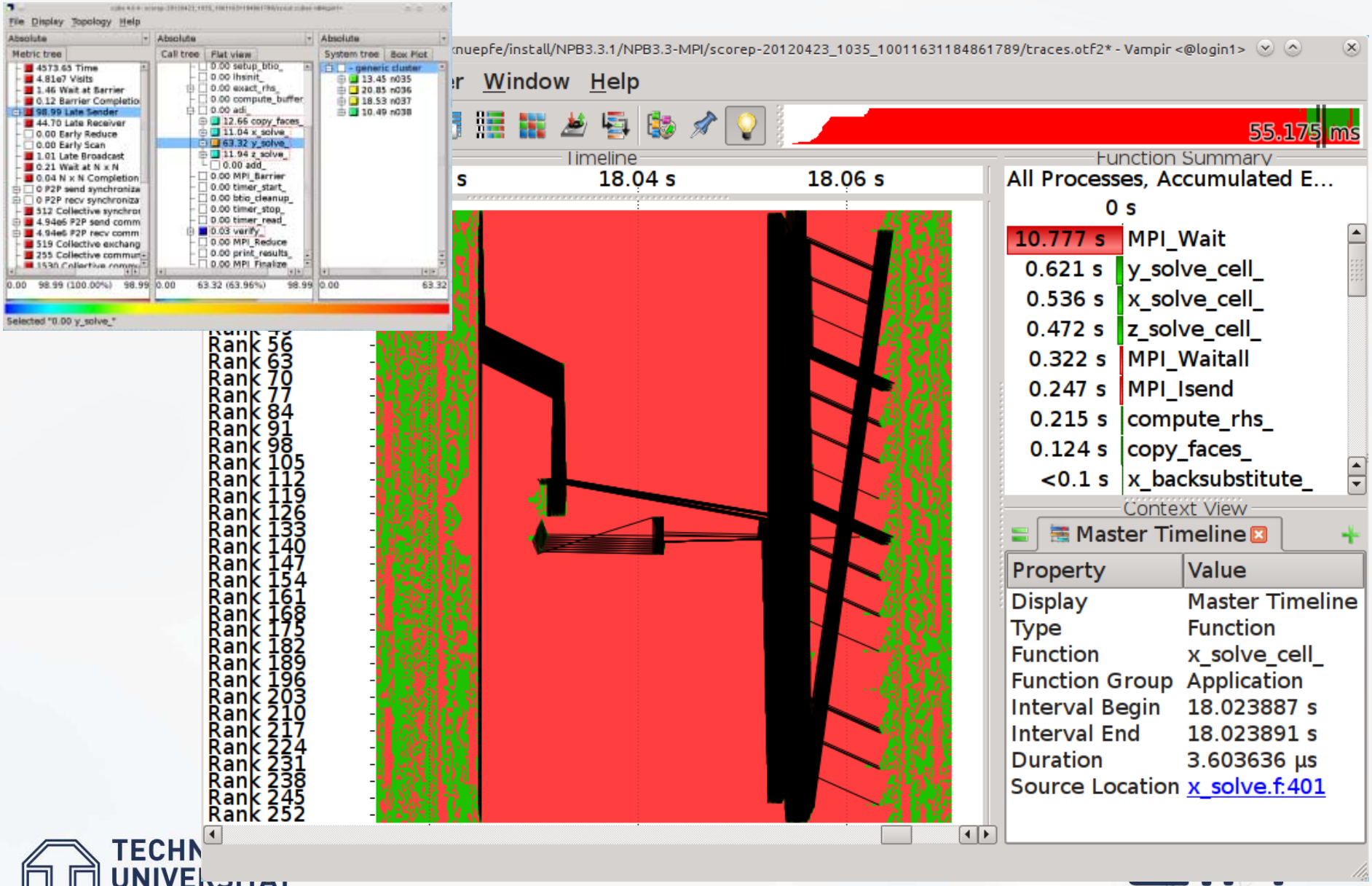
- Call site
- Called region
- Expand/collapse
- Hiding
- Cut call tree
- Find items
- Find Next
- Clear found items
- Copy to clipboard
- Min/max values
- Max severity in trace browser

- generic cluster
  - 13.45 n035
  - 20.85 n036
  - 18.53 n037
  - 10.49 n038

63.32

Shows the most severe instance of pattern in trace browser

Scalasca-Vampir Interoperation



Roadmap

June 2012

- Thinned out trace files (removal of similar/redundant tasks)
- Temporal pattern highlighting during tracing
- OpenMP 3.0 task support
- VampirServer: improved remote connections
- ScoreP becomes official measurement system for Vampir
- CUDA 4.1 support with native CUPTI based tracing

November 2012

- Further MPI performance metrics in performance radar
- Support for HMPP Instrumentation API
- OTF2 extensions
- Temporal pattern based trace compression
- Score-P: Scalability, rewind, selective tracing

Longer term

- In memory performance event data compression
- Alignment based trace comparison
- Improved correlation of event types.
- Scalability improvements
- Score-P: measurement of system background counters (I/O, energy, …)
- Vampir: Background counter view
- PGAS/SHMEM support