TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Performance Optimization on the Next Generation of Supercomputers

How to meet the Challenges?

Zellescher Weg 12

Tel. +49 351 - 463 - 35450

Jülich, July 4th, 2007

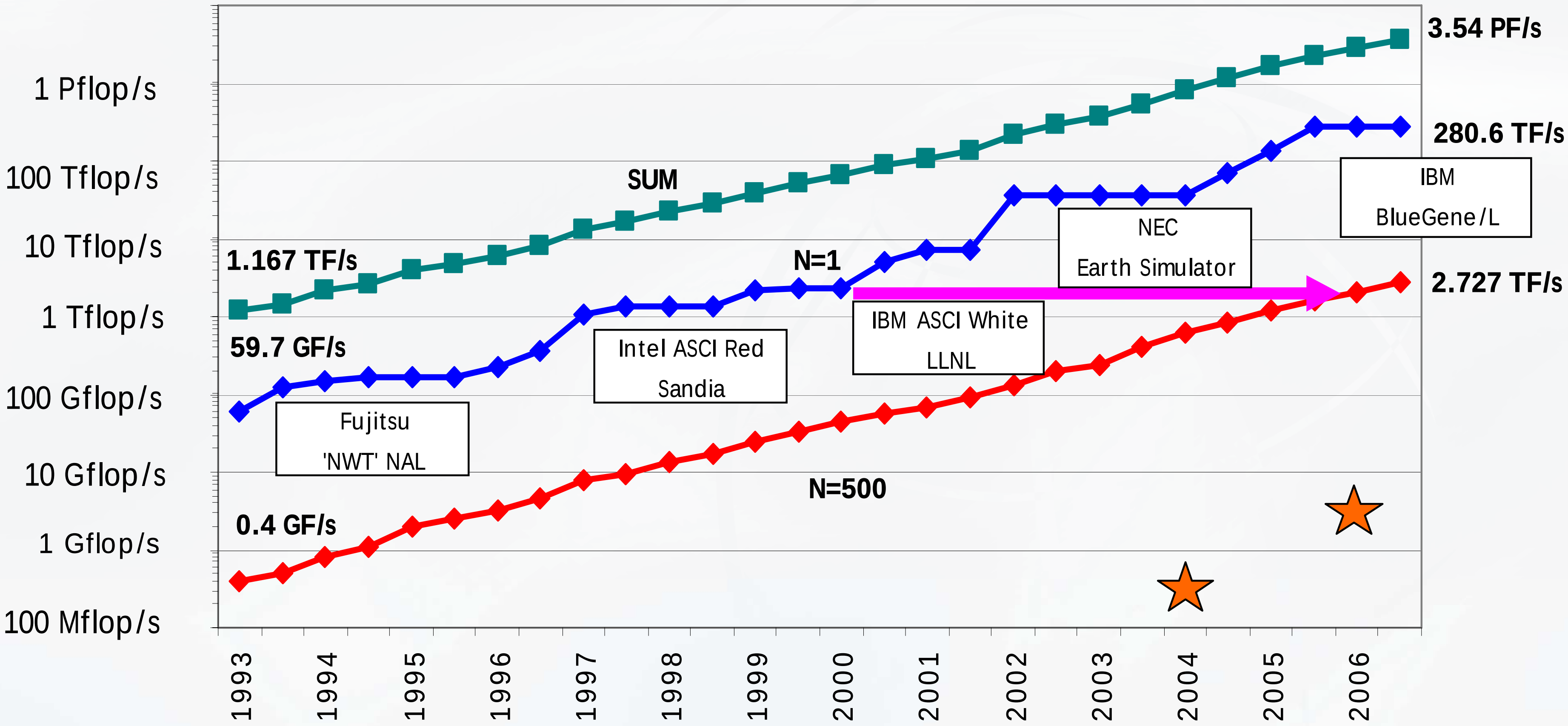
Wolfgang E. Nagel (wolfgang.nagel@tu-dresden.de)

Center for Information Services & High Performance Computing

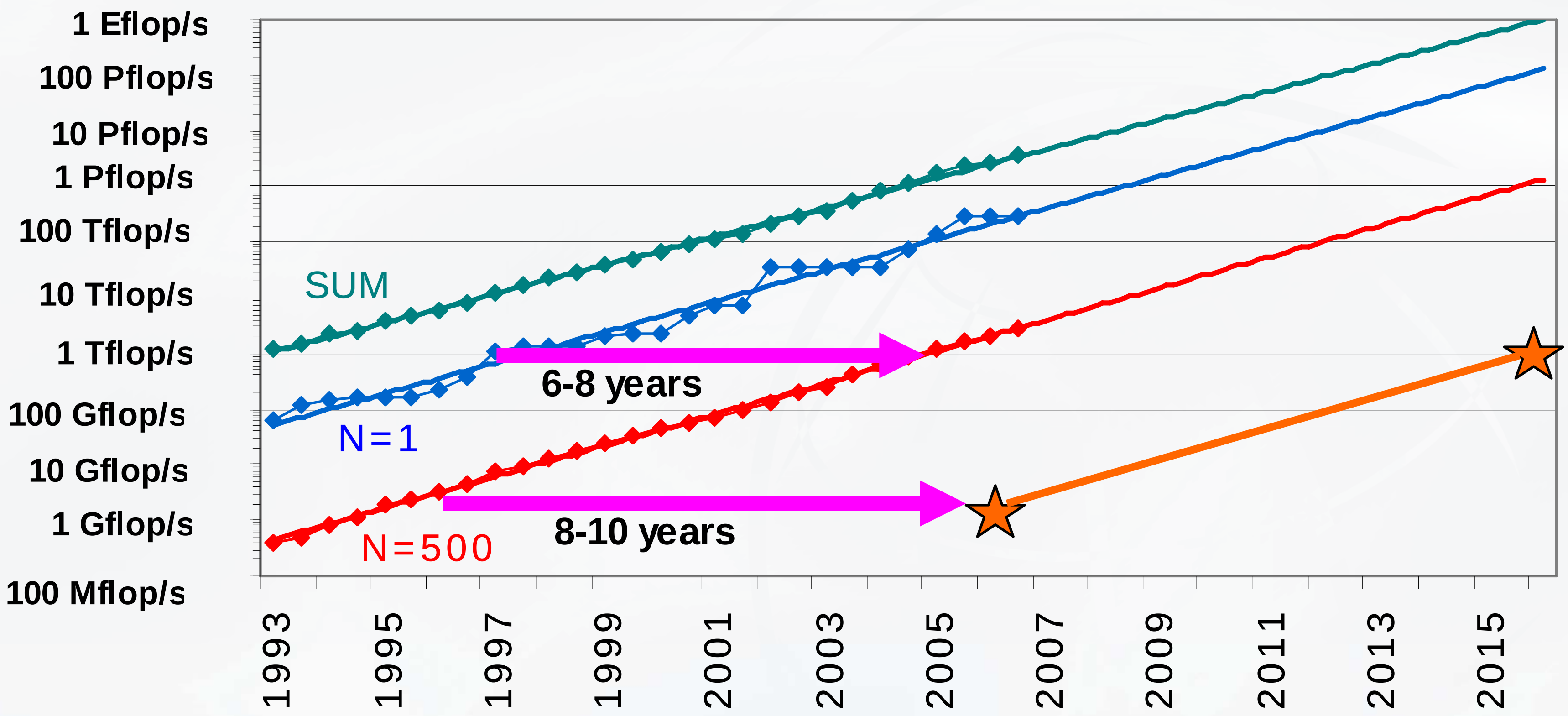
Contents

- Petaflops – the future is about there!
- Where are we today? A view from Dresden
  - Measurements
  - BenchIT
  - Vampir
- Some concluding remarks

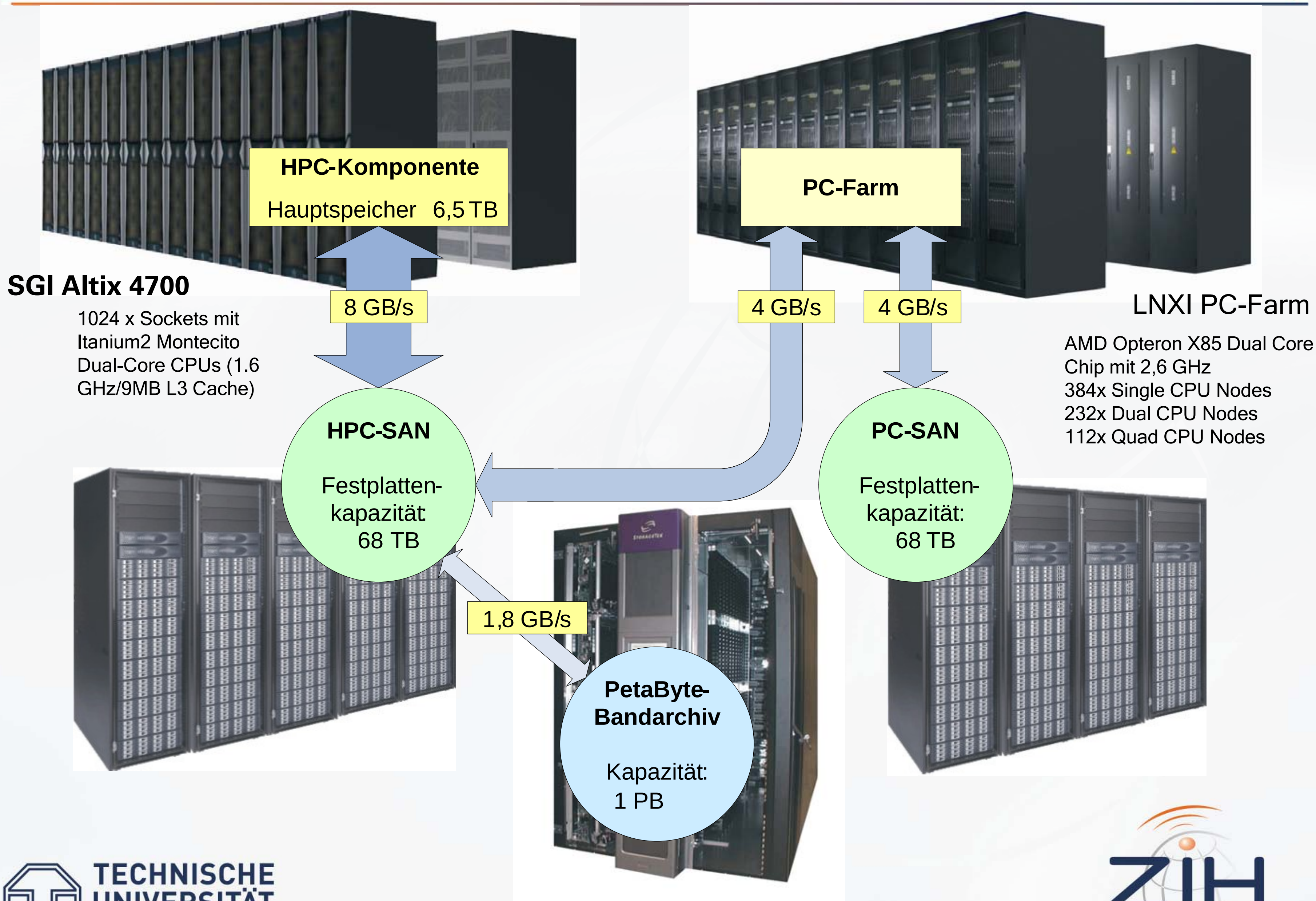
Performance Development



Performance Projection



HPC Situation at TU Dresden (HRSK)



TECHNISCHE UNIVERSITÄT DRESDEN

ZIH
Center for Information Services & High Performance Computing

Challenges which need PetaFlops (Scientific Case)

- Weather, Climatology and Earth Sciences
  - Climate change
  - Oceanography and Marine Forecasting
  - Meteorology, Hydrology and Air Quality
  - Earth Sciences
- Astrophysics, HEP and Plasma Physics
  - Astrophysics
  - Elementary Particle Physics
  - Plasma physics
- Materials Science, Chemistry and Nanoscience
  - Understanding Complex Materials
  - Understanding Complex Chemistry
  - Nanoscience

Challenges which need PetaFlops (Scientific Case)

- Life Sciences
  - Systems Biology
  - Chromatine Dynamics
  - Large Scale Protein Dynamics
  - Protein association and aggregation
  - Supramolecular Systems
  - Medicine
- Engineering
  - Complete Helicopter
  - Simulation
  - Biomedical Flows
  - Gas Turbines & Internal Combustion Engines
  - Forest Fires
  - Green Aircraft
  - Virtual Power Plant

What are the major challenges … in our area

- Getting Petaflops machines
- Getting Petaflops machines also in Europe!

- Performance problem isolation debugging when problems occur
- Metrics that capture new issues such as space, power, cooling (TCO)
- Metrics that capture hardware and software reliability and consistency
- How to measure performance on a diverse set of architectures, including heterogeneous, large-scale, etc.?
- How do you set performance expectations? Aggregate measures or performance modeling? How do you know you can trust the models?
- Users that view the machines as “utilities”
  - Usability and performance issues

What about a real Petascale system?

(from **Patricia Kovatch, SDSC)**

- Probably 1 million cores with 1 GB of memory/core -> 1 PB of total memory
- We generally allocate 2 GB of memory/CPU now so what will apps do? Idle cores due to memory size and BW
  - 1 PB probably too expensive, assume 0.5 PB
- Performance changes with Petascale?
  - Expect parallel file system performance ~1 TB/s
  - ~10 minutes to write to disk at this speed, no significant changes
- Memory will be more important than cores!

How much will it cost to save Petascale systems memory to tape?

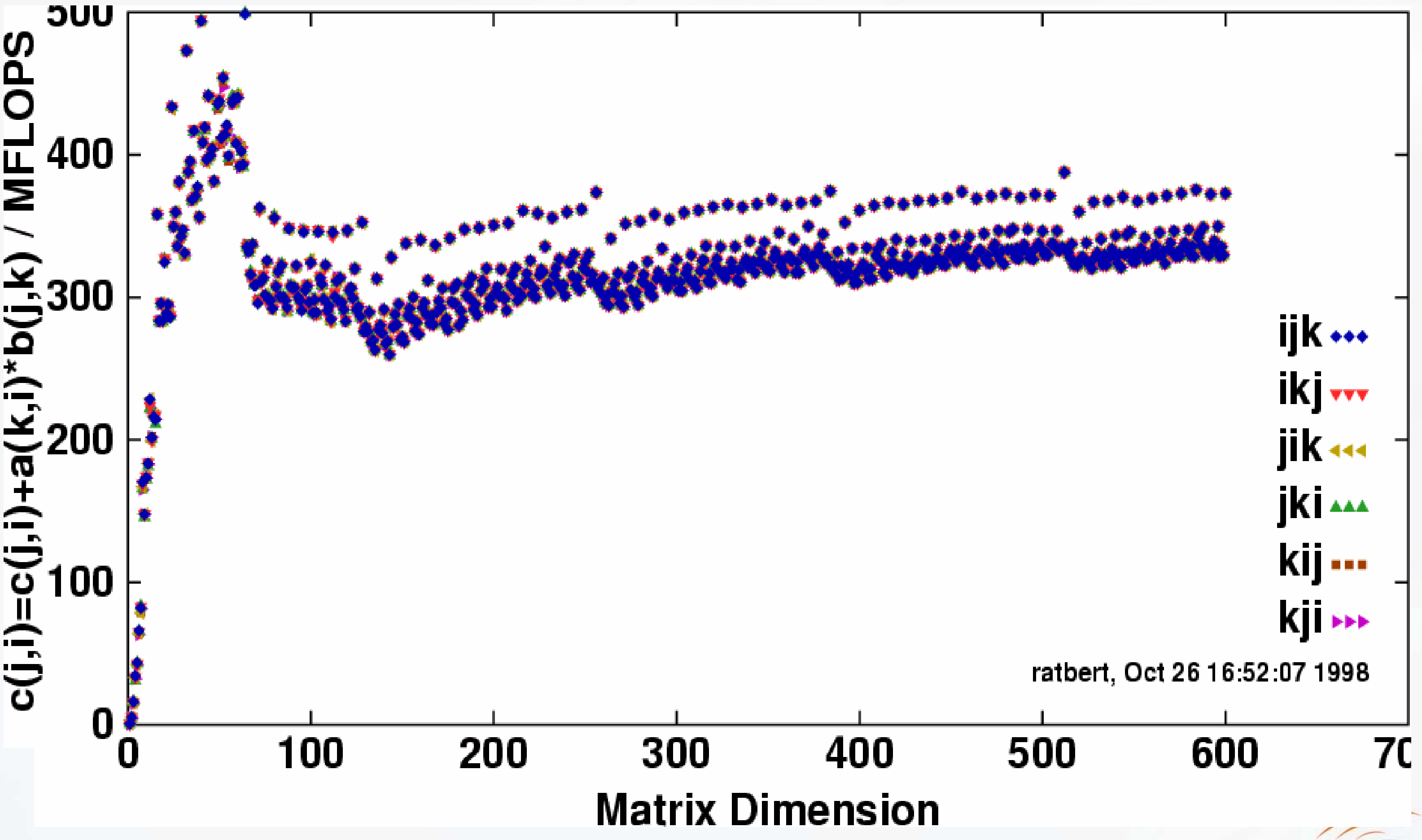
- 1 TB cartridge costs ~$100 -> 0.5 PB costs ~$50K!
  - Assuming 0.5 PB total system memory
  - Write memory 4X/week -> $10M/year for tapes?!!
- Or consider actual archival usage at SDSC
  - DataStar: ~5 TB/memory -> $1M/year in tapes
  - Petascale: 100X memory -> $100M/year in tapes?!!!
  - **Much more expensive than power costs!**
- Storage size?
  - Data parked on parallel file system for 3 months at a time -> 24 PB file system, 50 PB more likely
  - Assuming full system memory written 4X/wk * 0.5 PB * 12 weeks
- Storage **more expensive than flops**!

How will Petascale systems change what we have been doing?

- Memory-driven computing
  - Allocations and queuing based on memory, not cores
  - Memory is the scare resource, cores sit unused
- Disk-driven storage
  - Allocations based on storage
  - Tape costs are prohibitive
  - RAID 6 and other schemes essential for highly reliable file systems
  - Integrated global parallel file system and archival storage
  - Users perform real-time, concurrent analysis and visualization
  - Faster/cheaper to rerun job than to restage data from archive
- It is all about **data …**

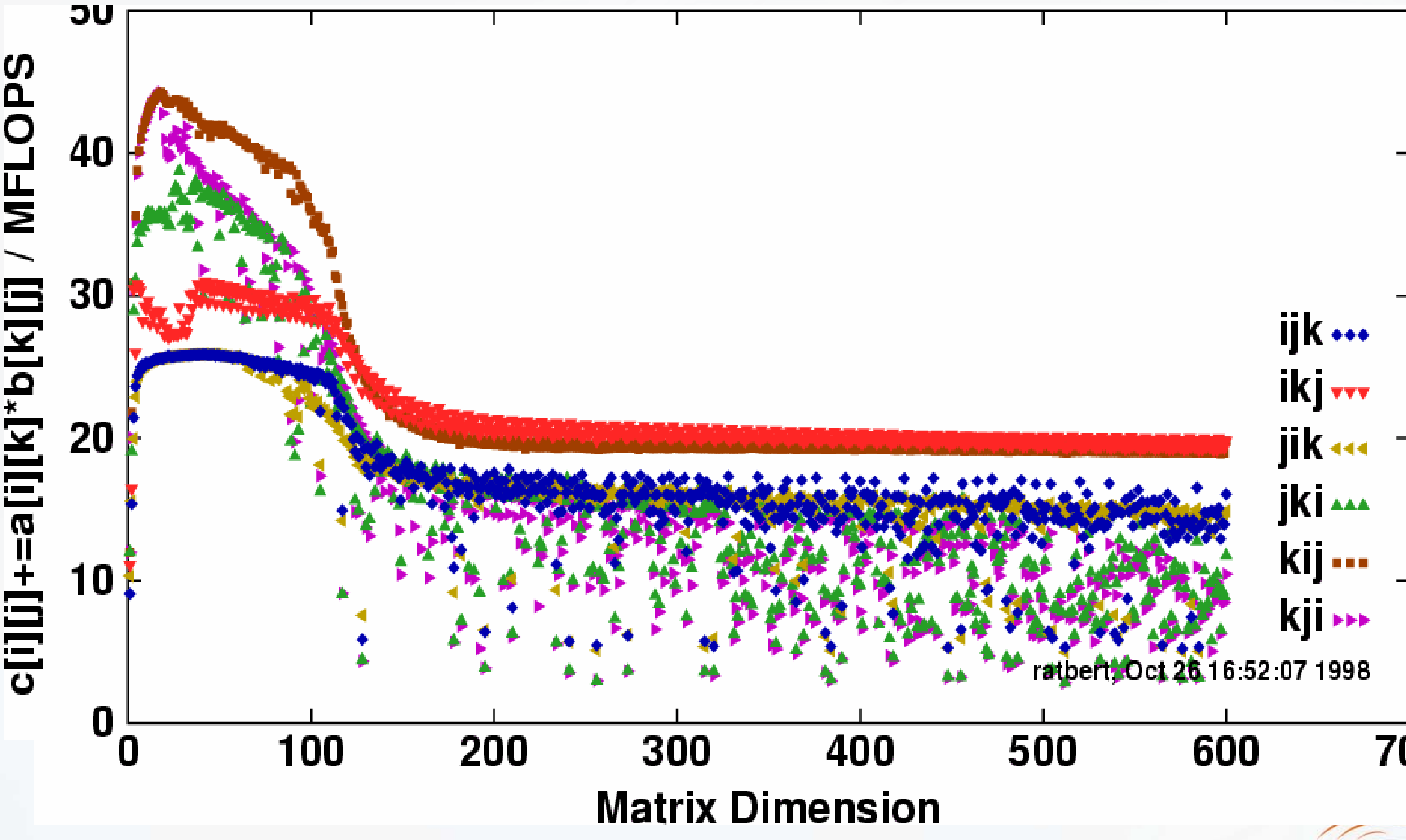
Matrix Multiplication

Cray T3E - Fortran Version

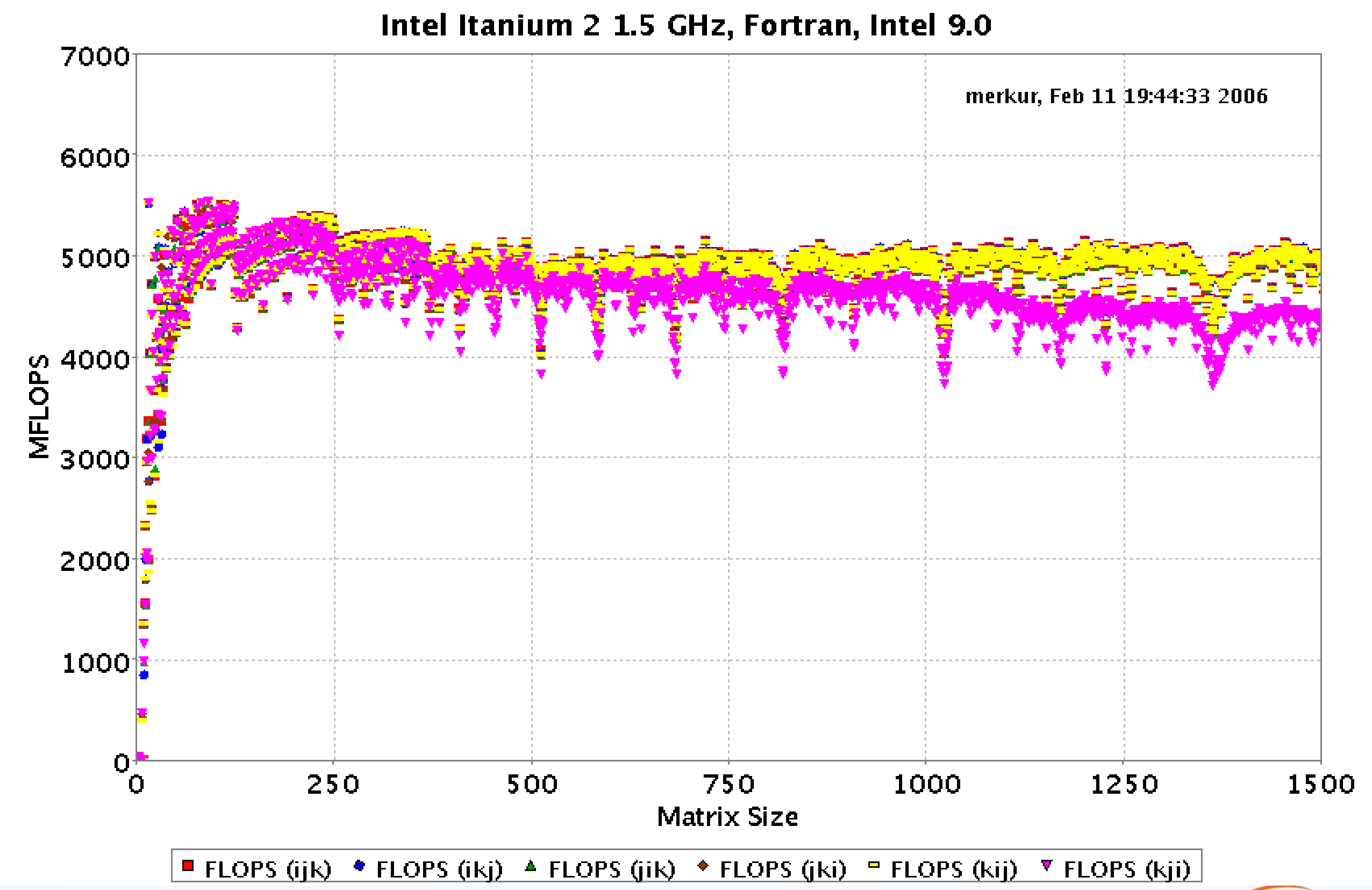


Matrix Multiplication

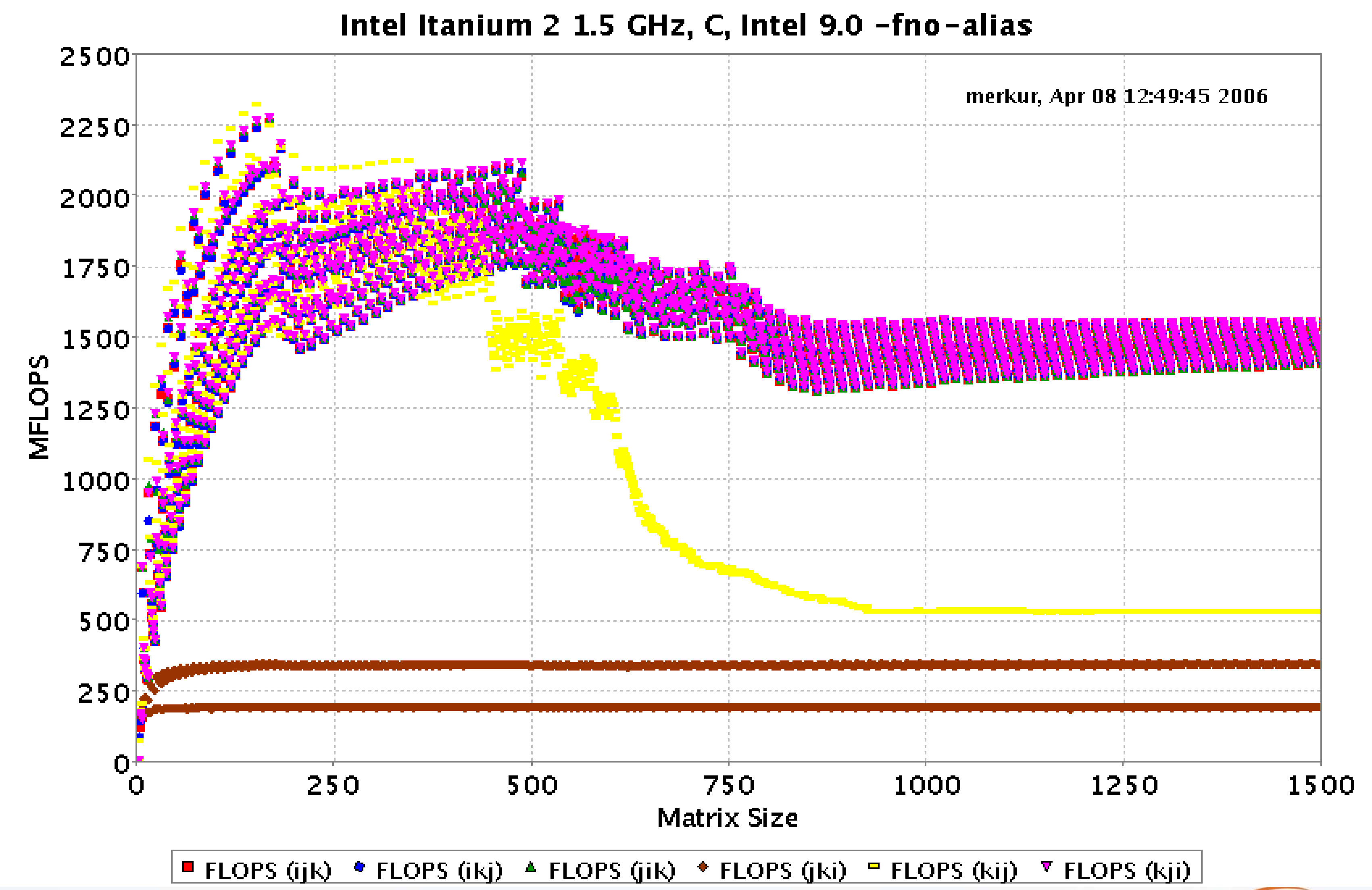
Cray T3E - C Version



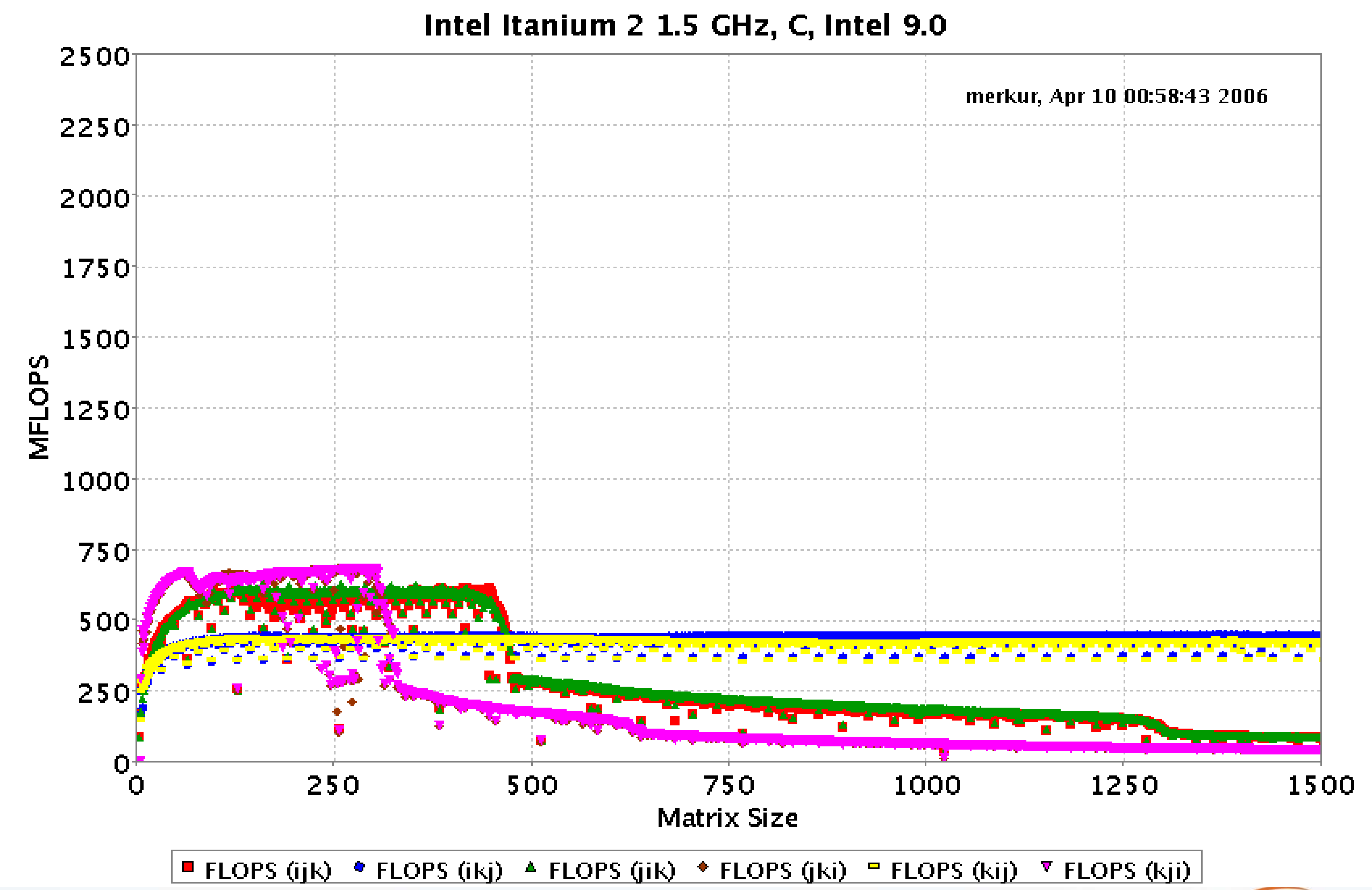
Matrix Multiplication



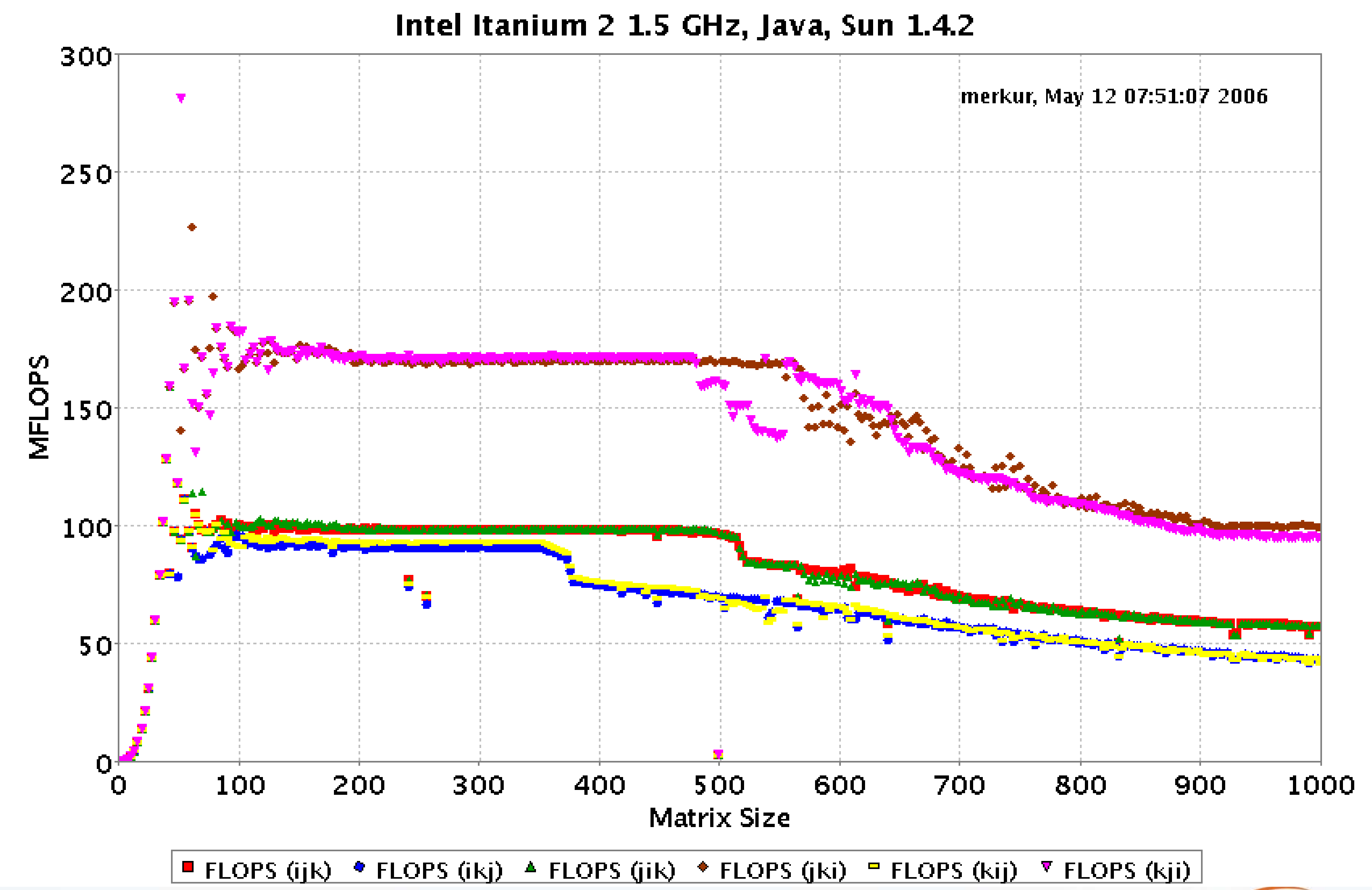
Matrix Multiplication



Matrix Multiplication



Matrix Multiplication



Performance benefit ?

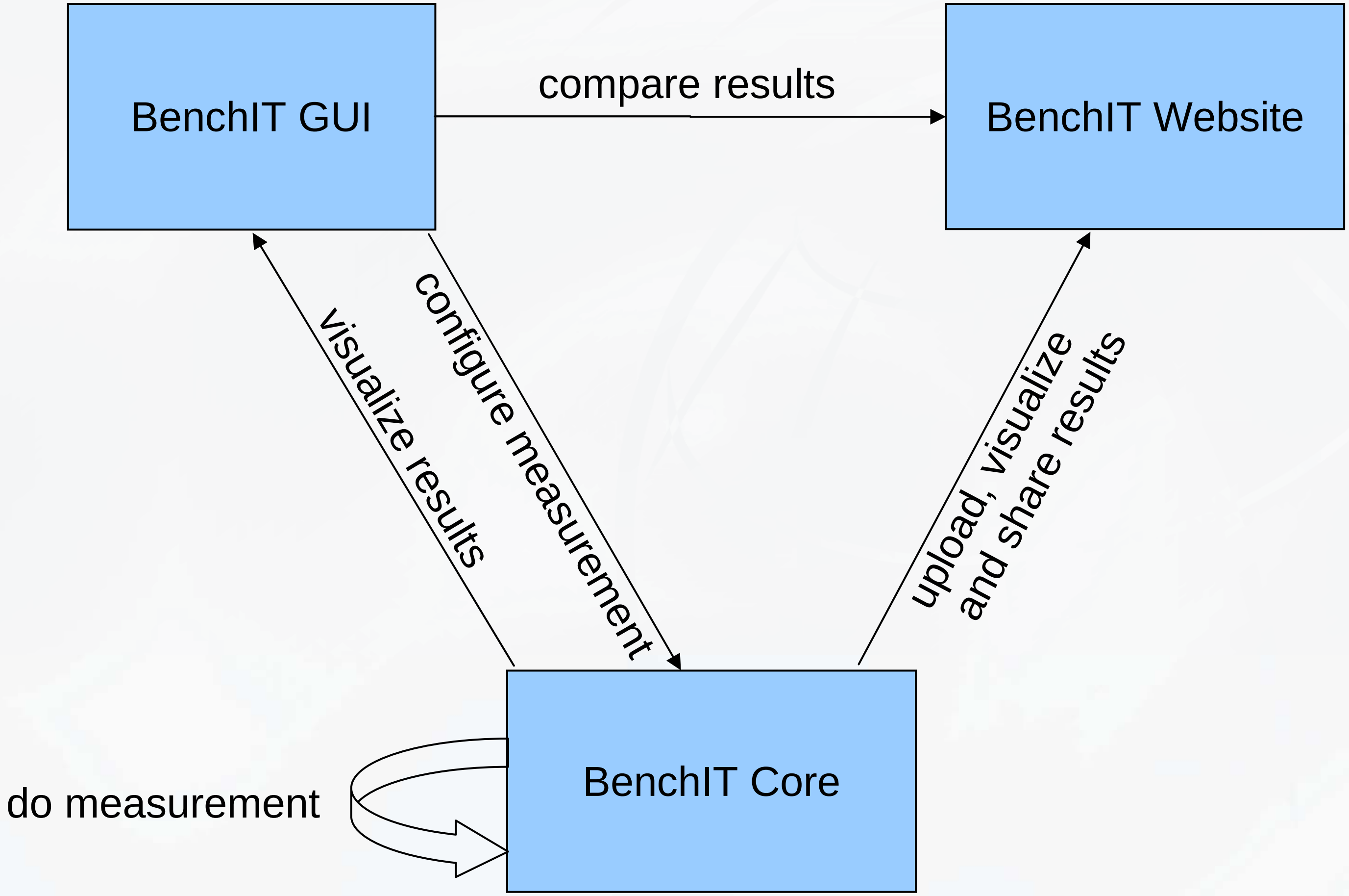
Current hardware architectures allow to achieve reasonable performance, but most times only with hand-tuned code

Very difficult to predict, which hardware architecture and which software algorithm is the best for a certain user application!

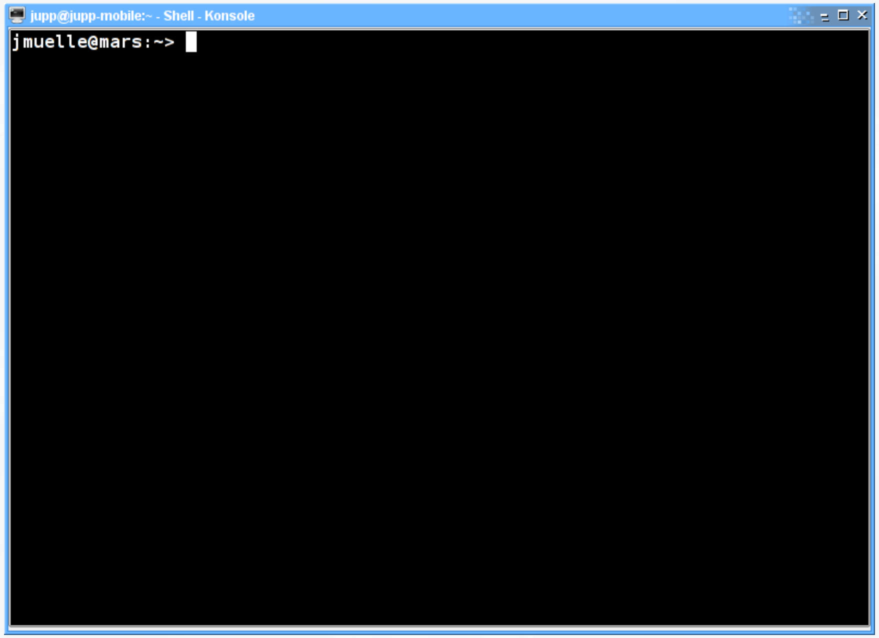
Users have to be supported to run parallel systems efficiently!

Feature Overview

BenchIT provides tools to …



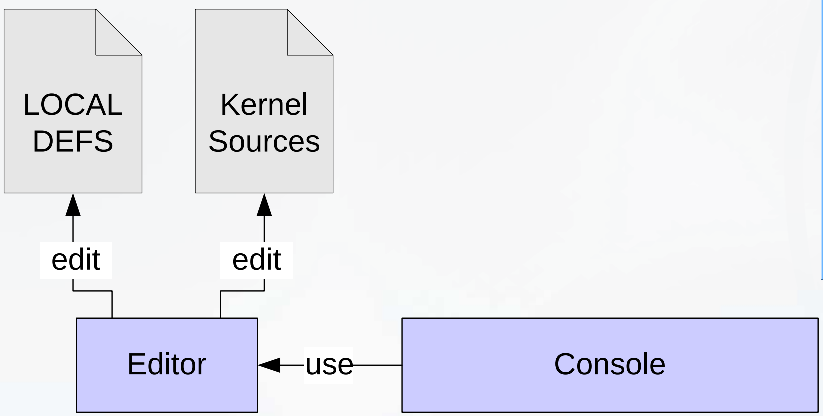
BenchIT – Step by Step

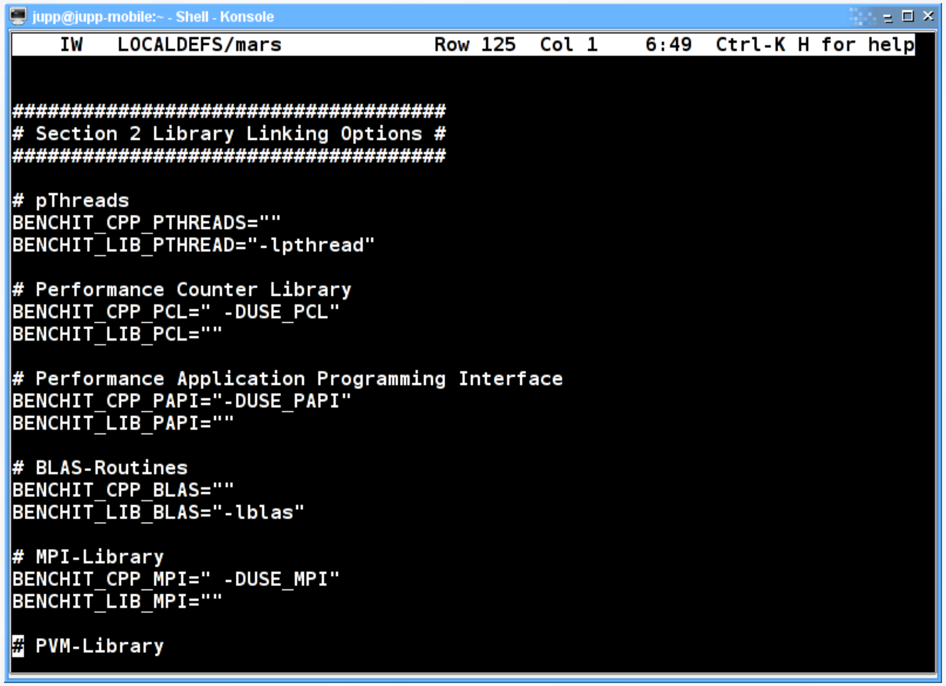


Editor

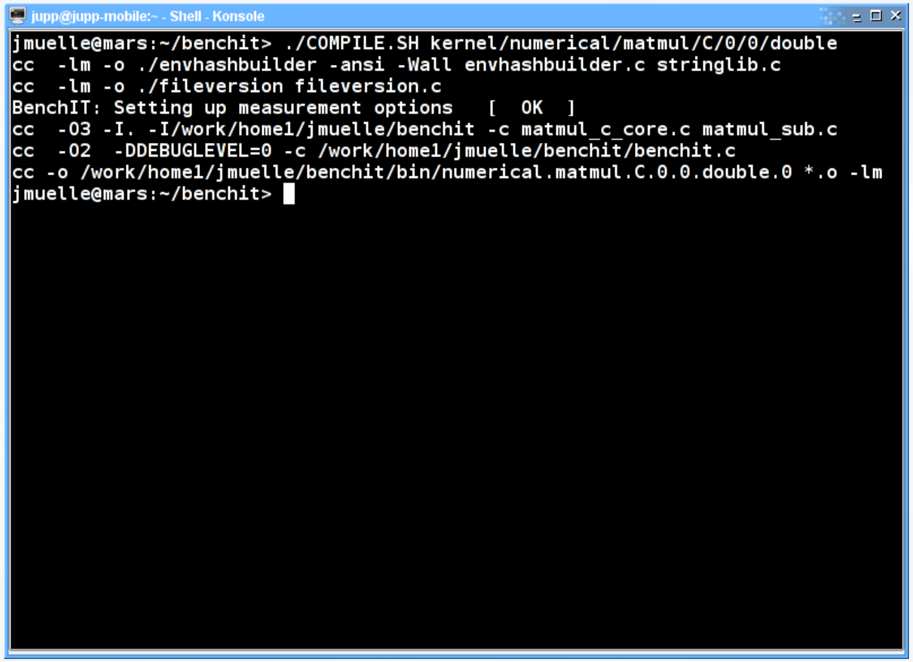
Console

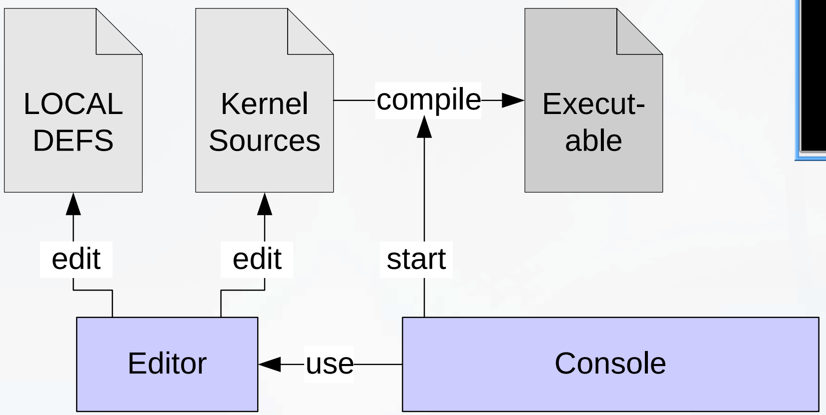
BenchIT – Step by Step



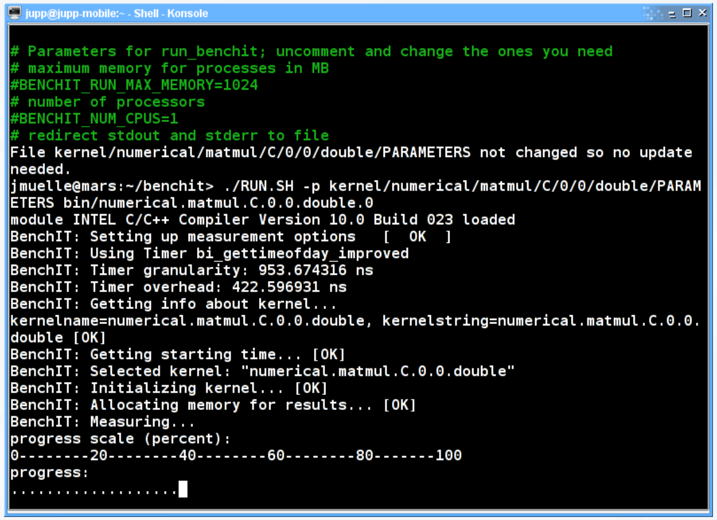


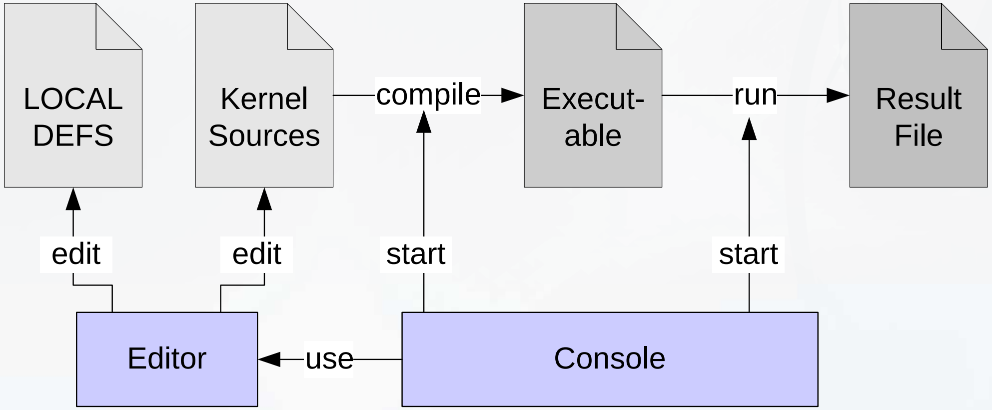
BenchIT – Step by Step



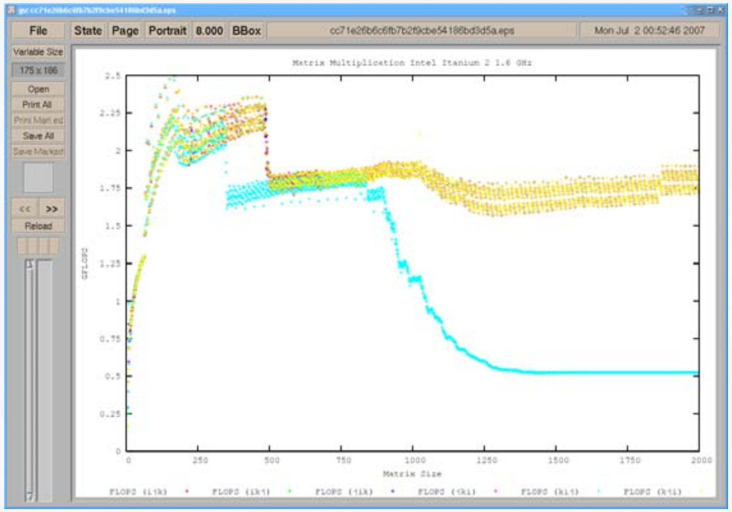


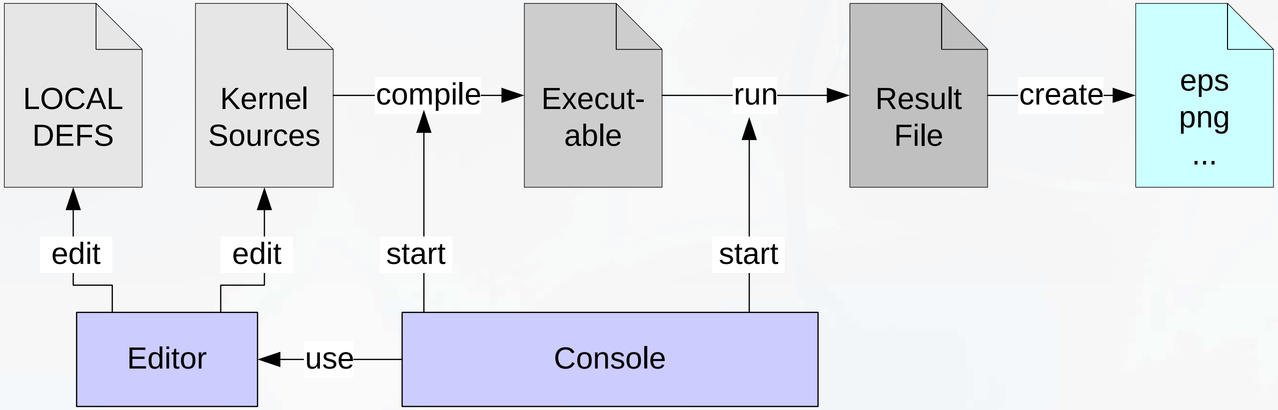
BenchIT – Step by Step



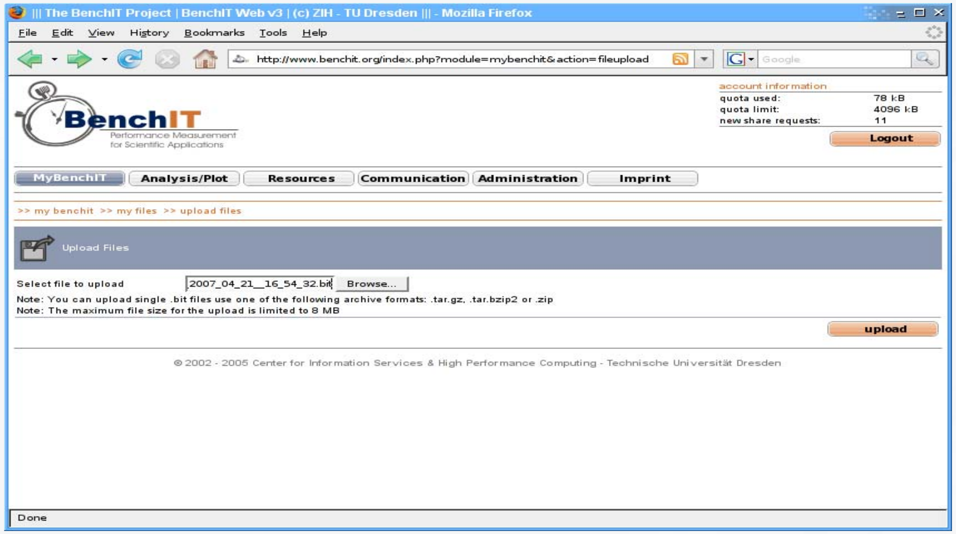


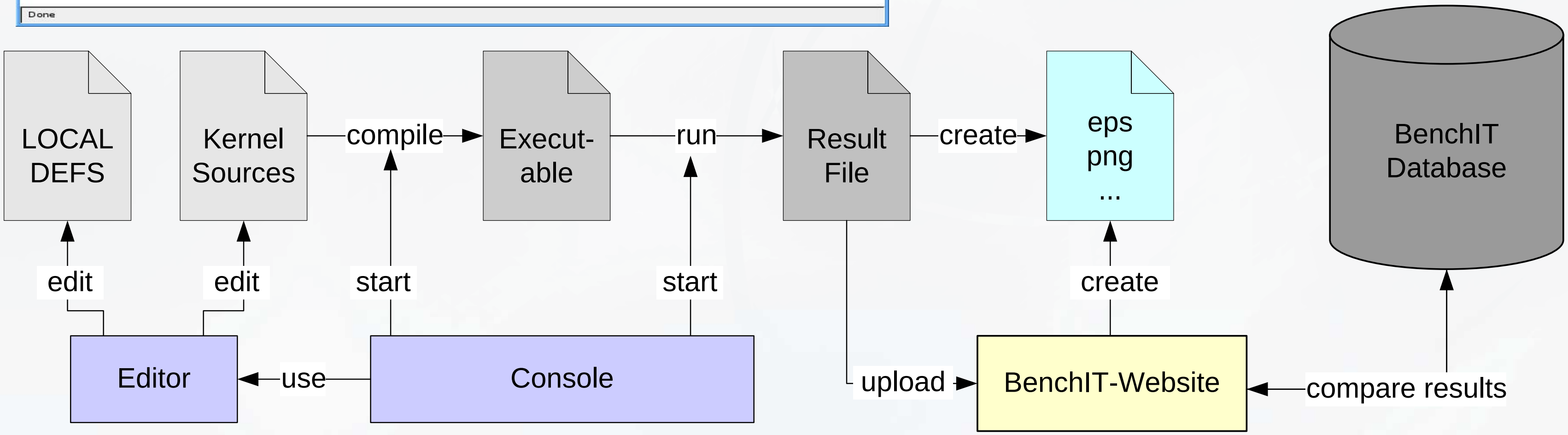
BenchIT – Step by Step



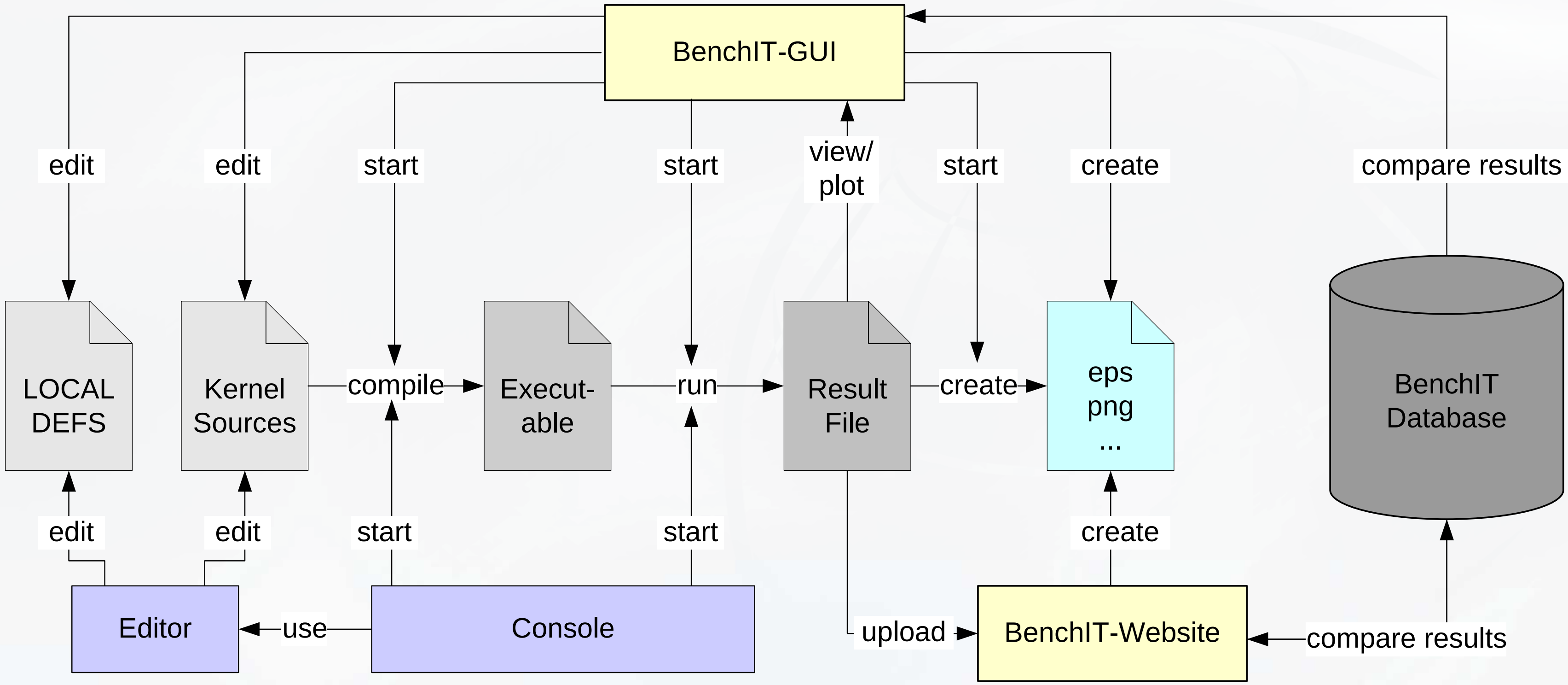


BenchIT – Step by Step





BenchIT – Step by Step



Getting the Sources

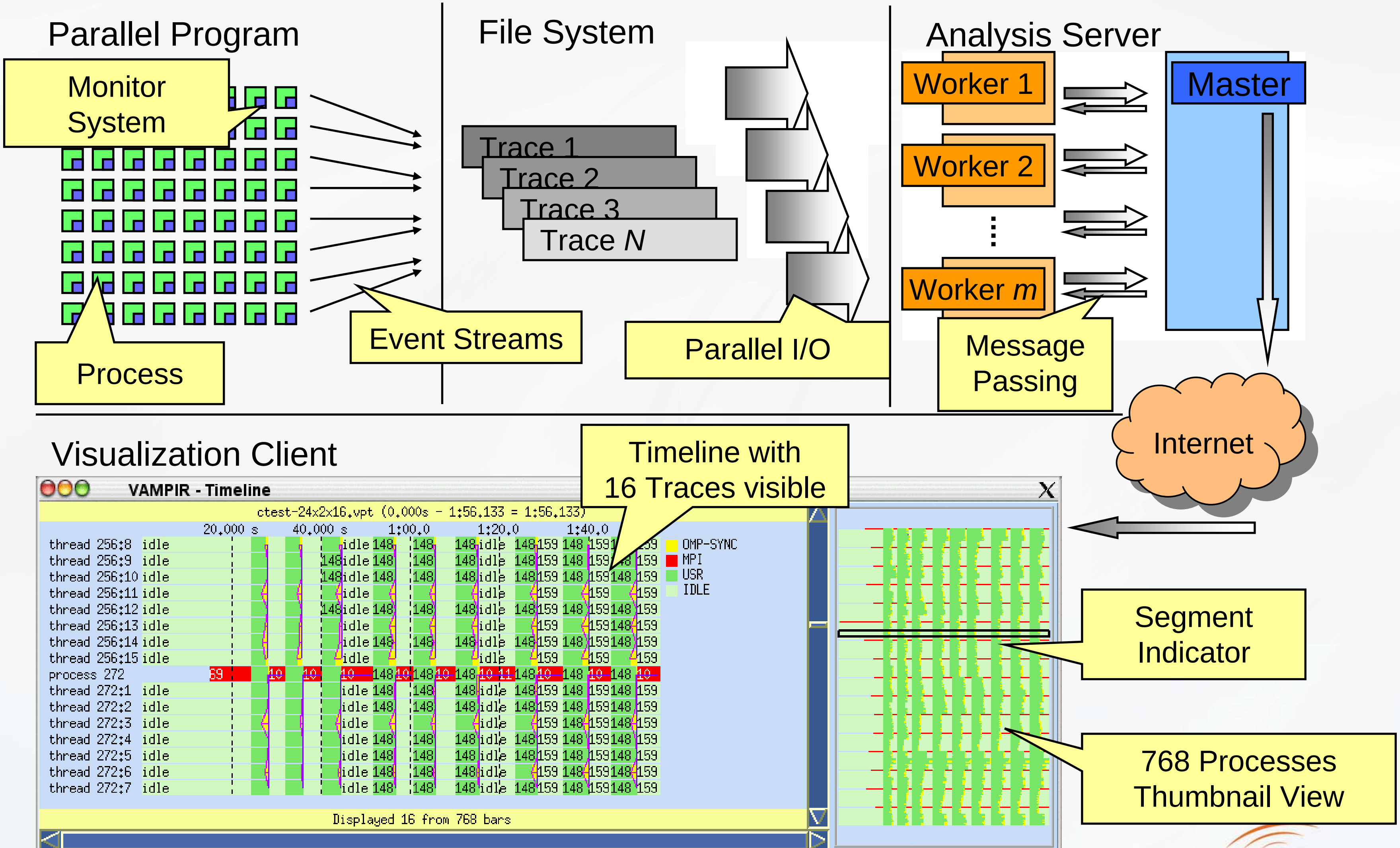
www.benchit.org

Performance Analysis

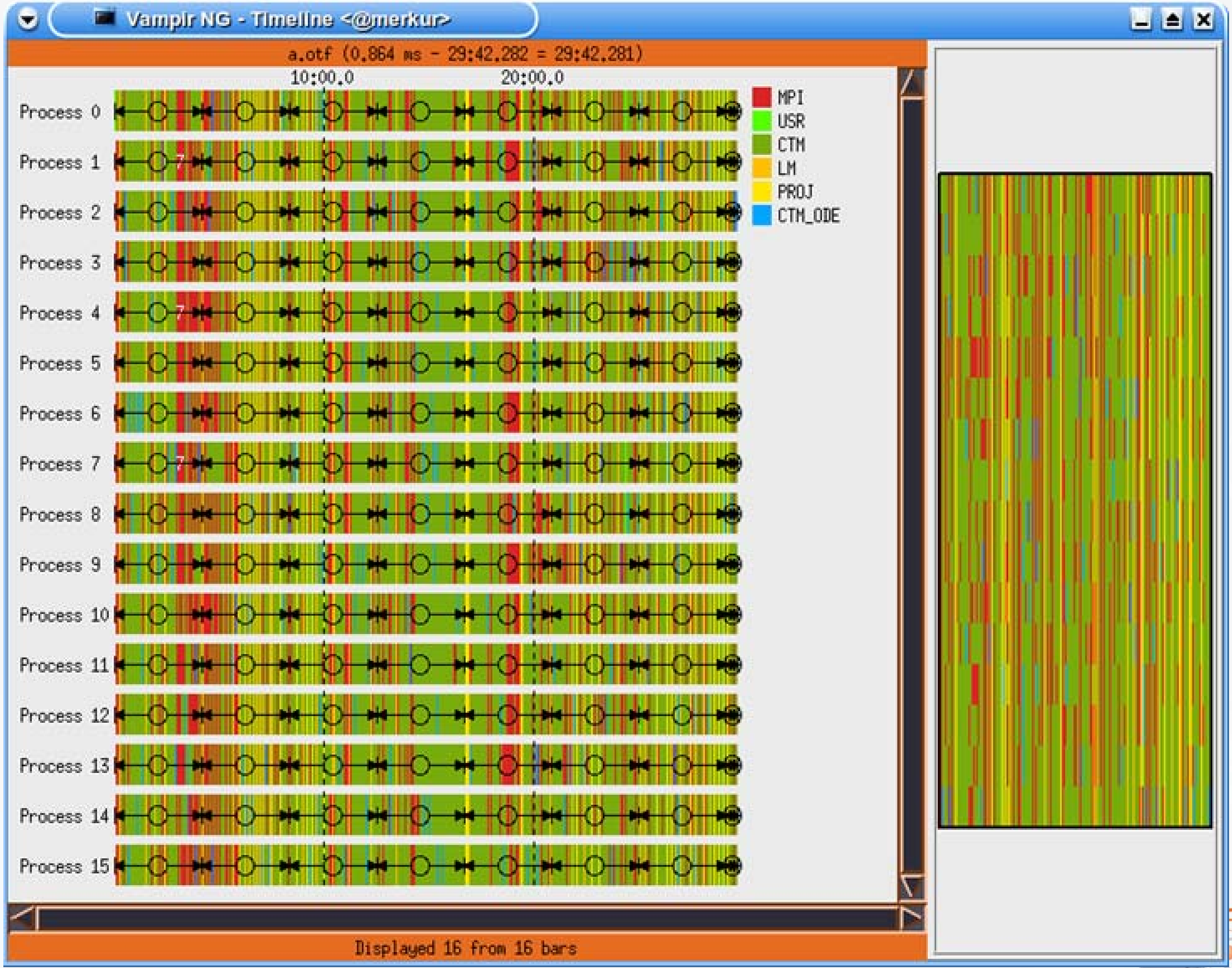
⇒**Software-Tool: „Vampir“**

- Performance visualization and analysis tool
- Enables detailed understanding of dynamic process changes on massively parallel systems
- X Window based system
  (implemented in C, based on OSF/Motif)
- Development started more than 15 years ago at Research Centre Jülich, ZAM
- Since 1997, Vampir is developed at TU Dresden
  (first: collaboration with Pallas GmbH,
  from 2003-2005: Intel Software & Solutions Group,
  since January 2006: TU Dresden / ZIH)
- Vampir pretty much accepted in the field

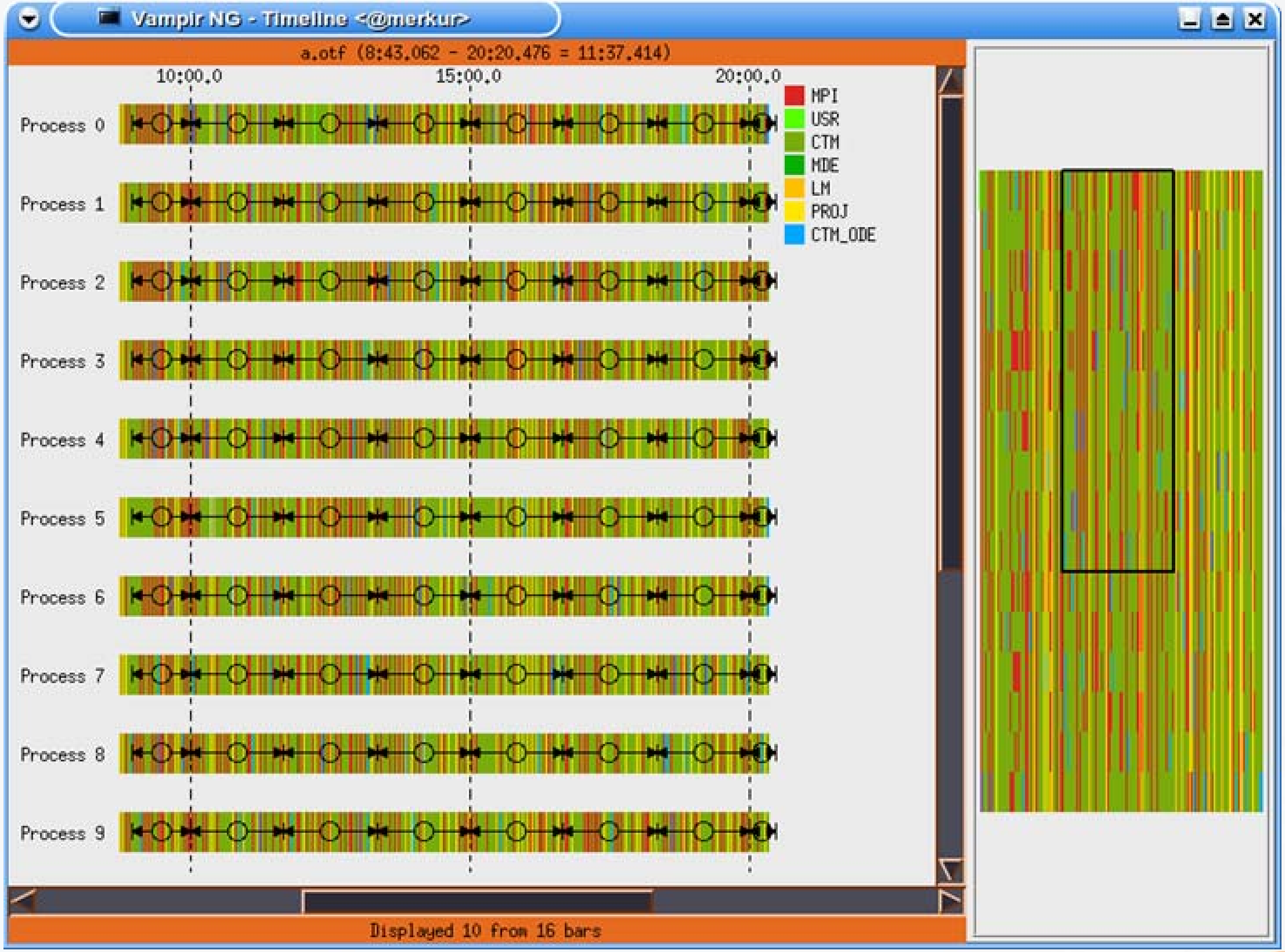
Vampir Server Architecture



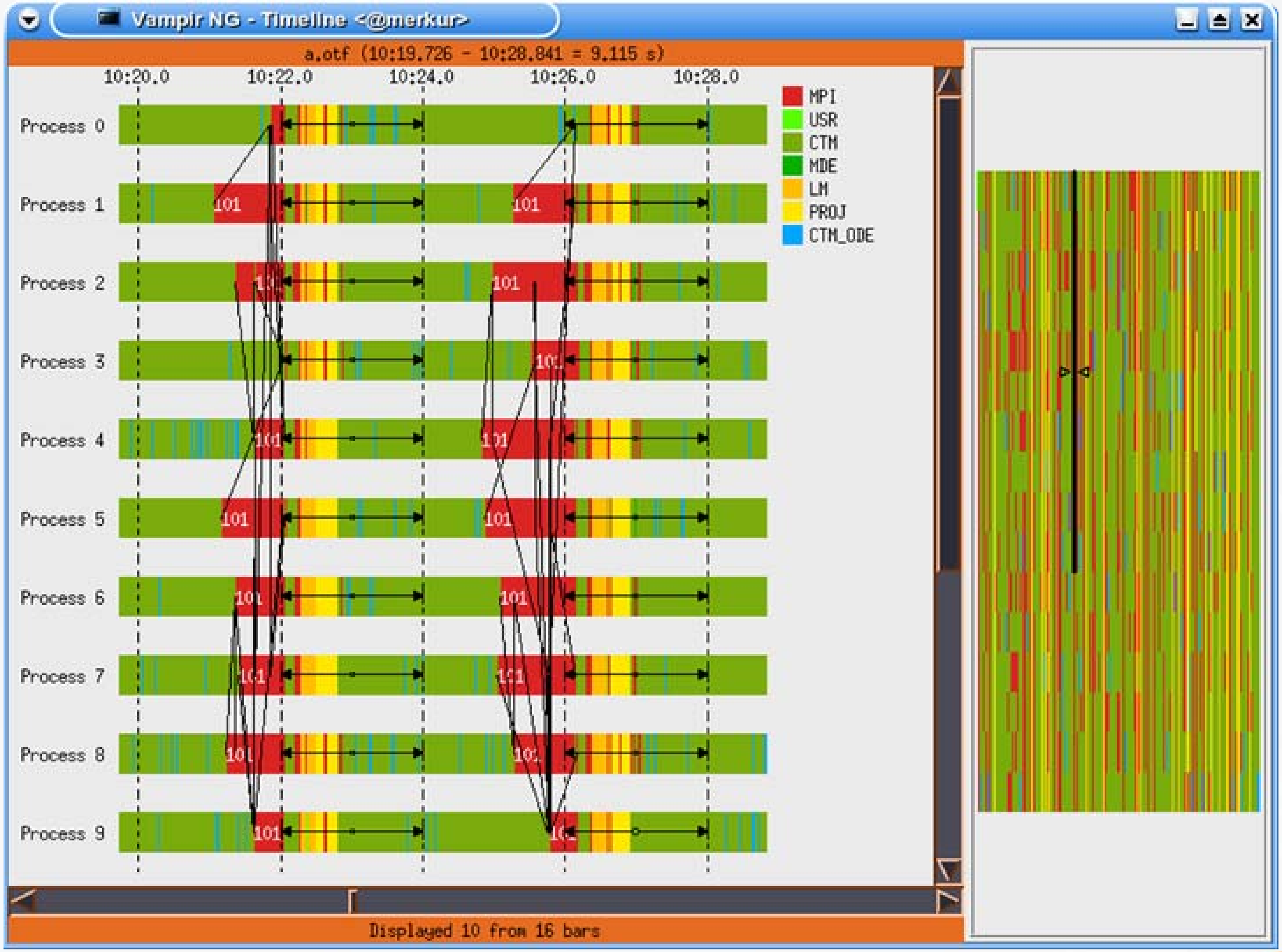
Vampir: Global Timeline



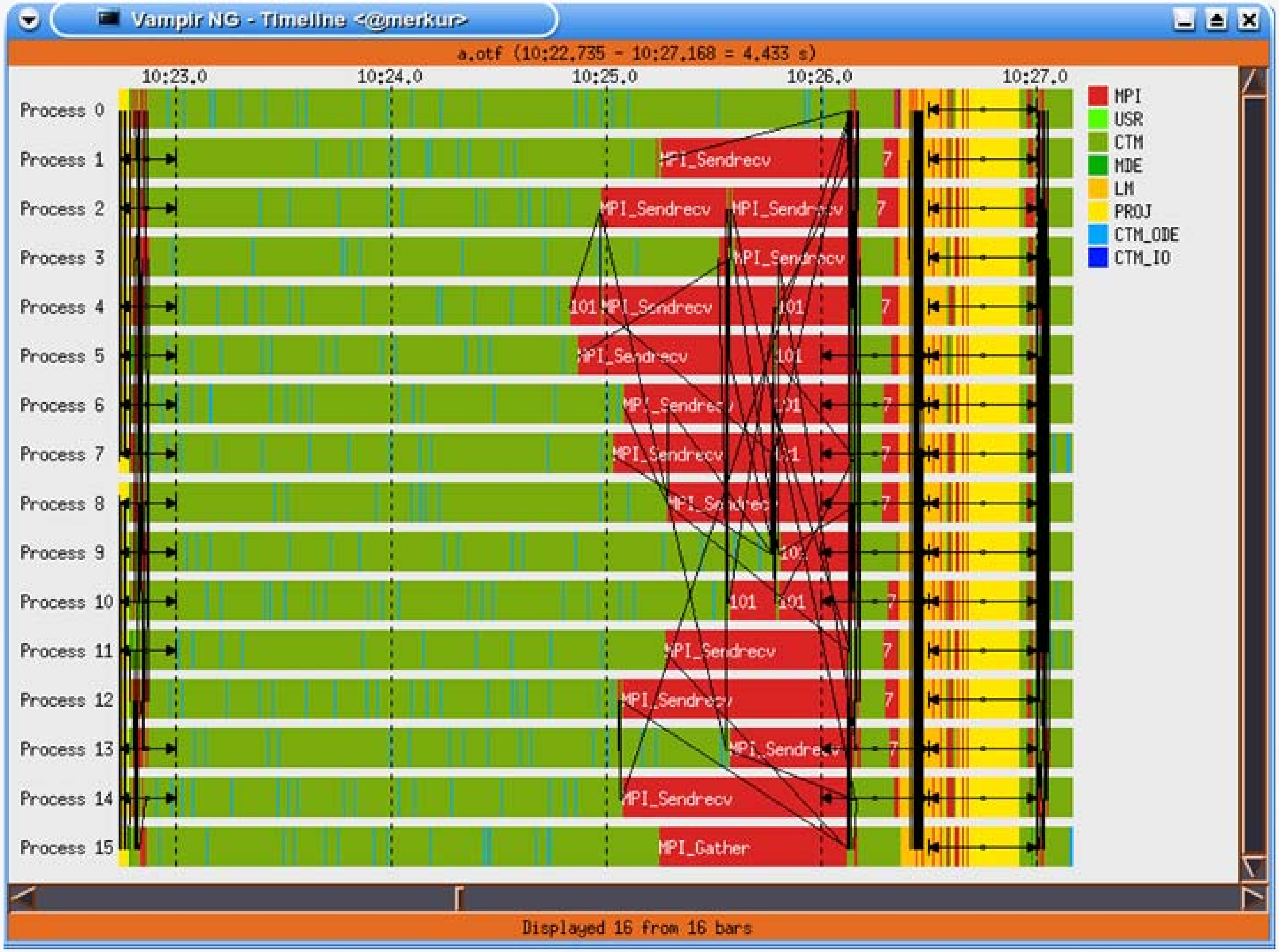
Vampir: Global Timeline



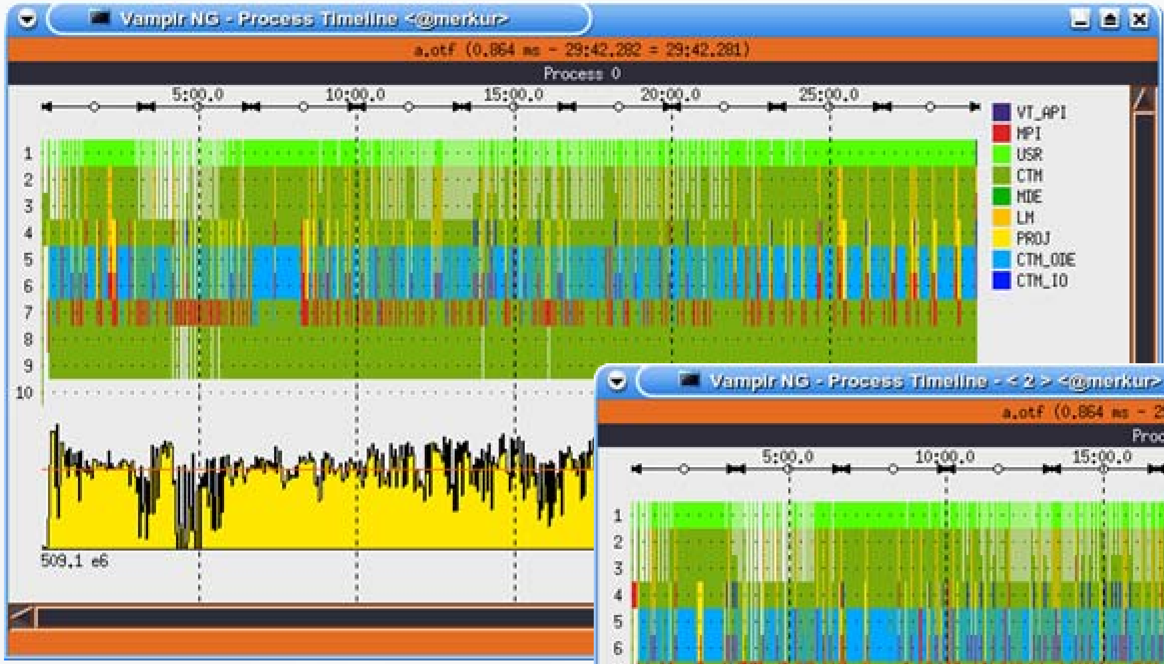
Vampir: Global Timeline

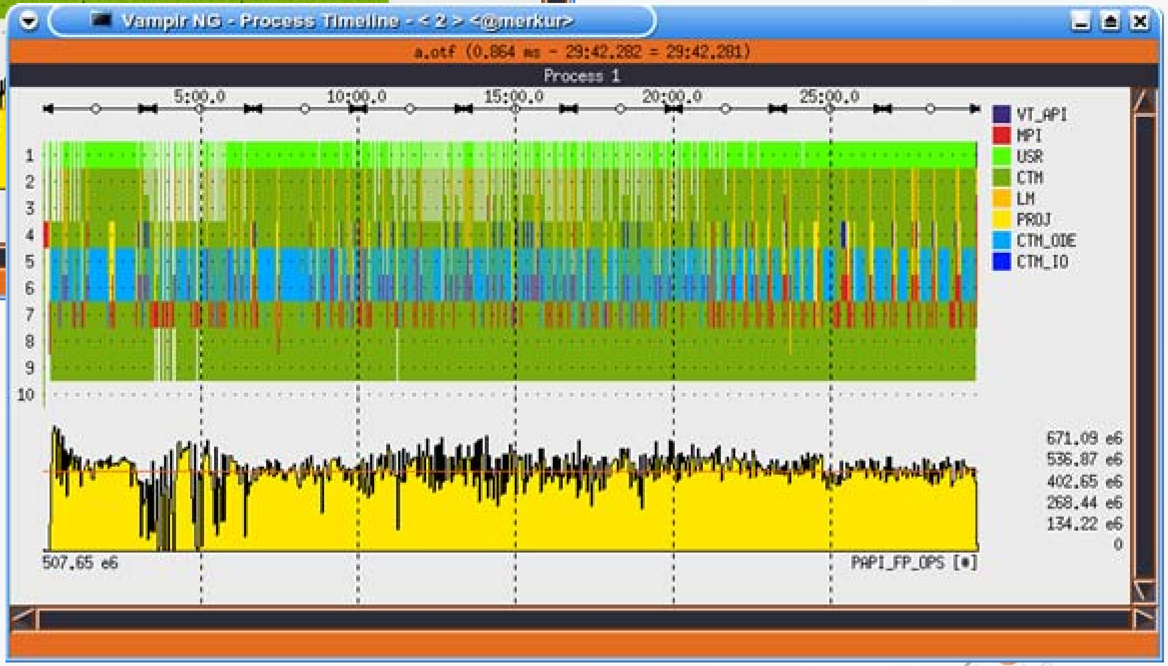


Vampir: Global Timeline

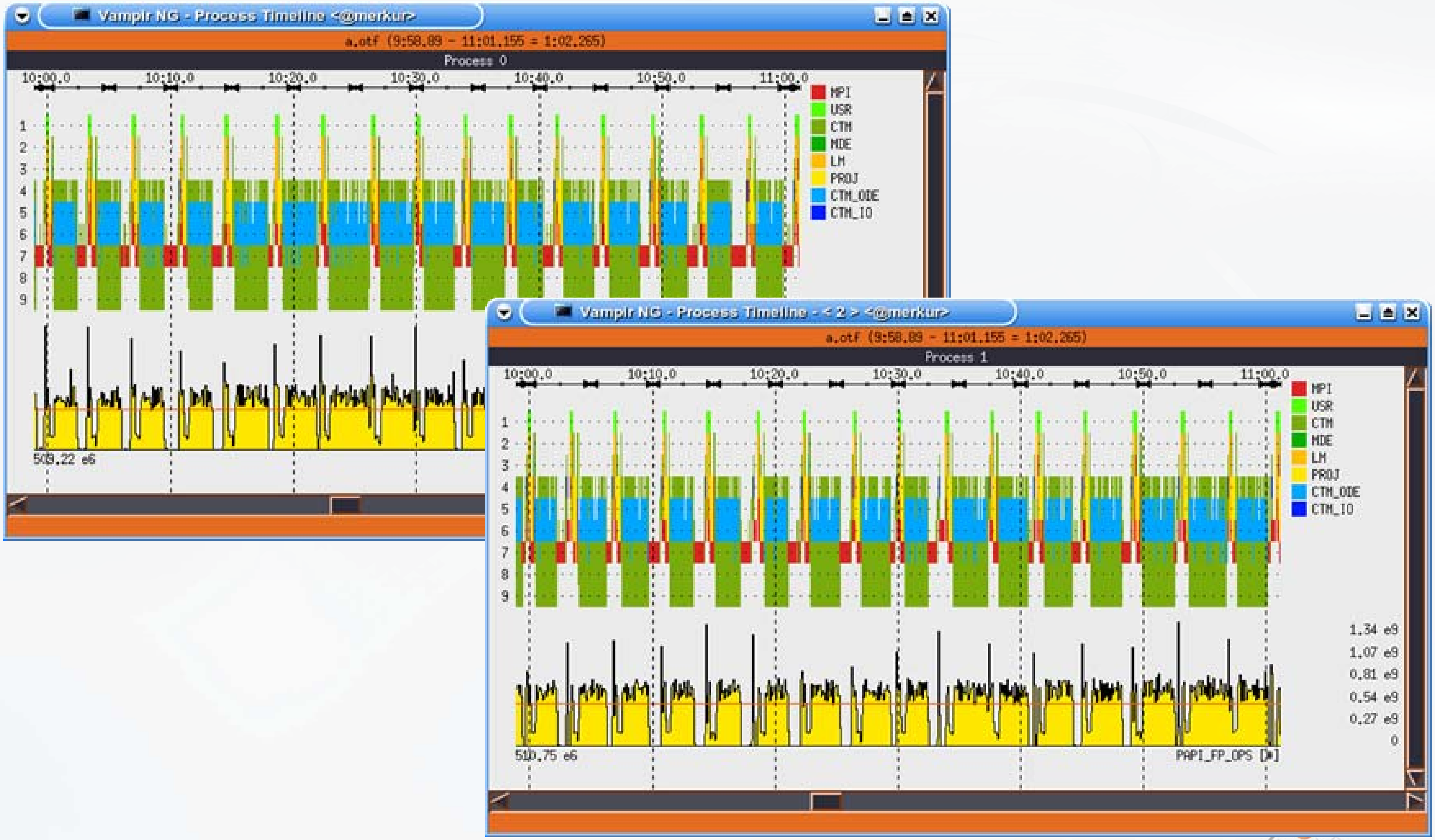


Vampir: Process Timeline



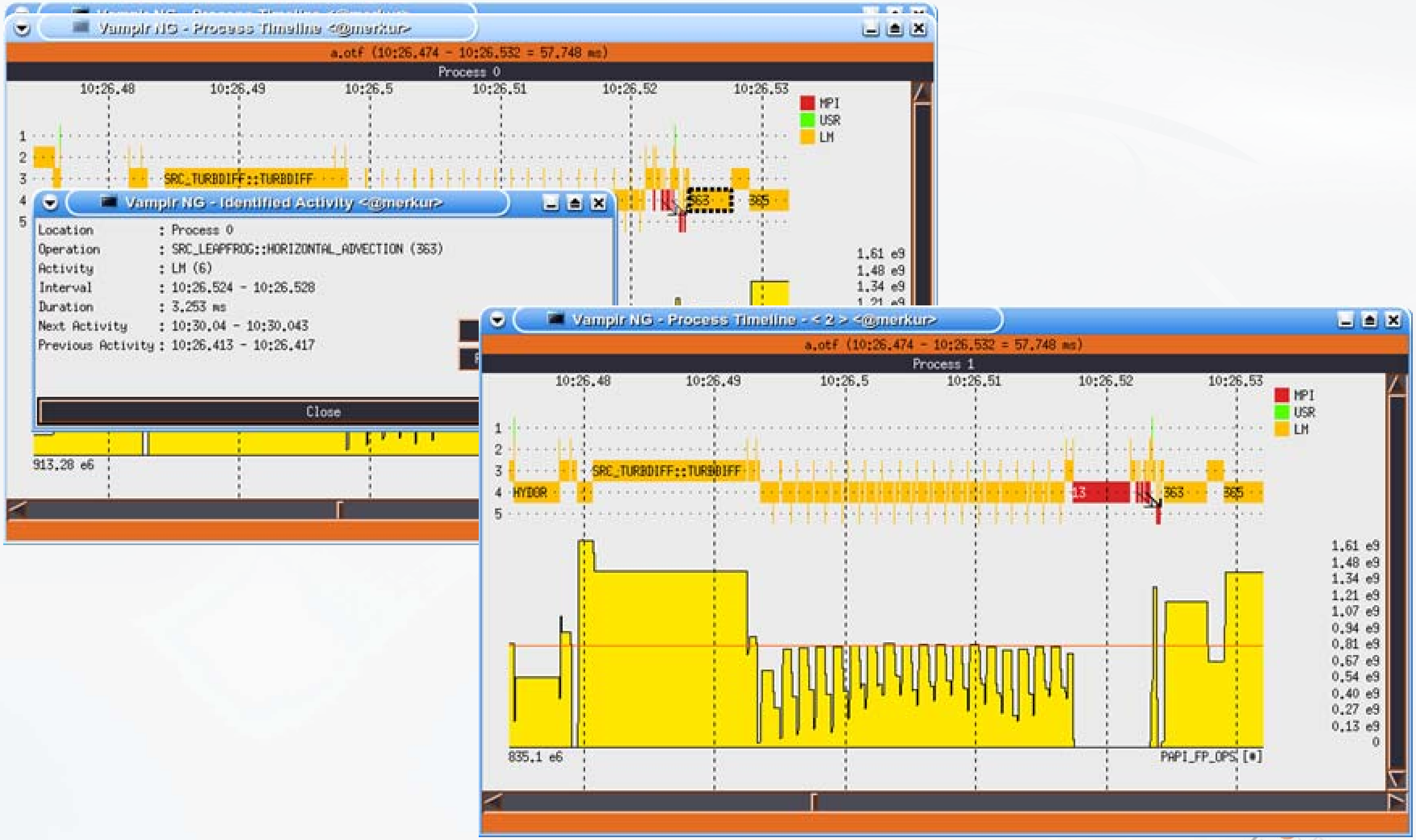


Vampir: Process Timeline

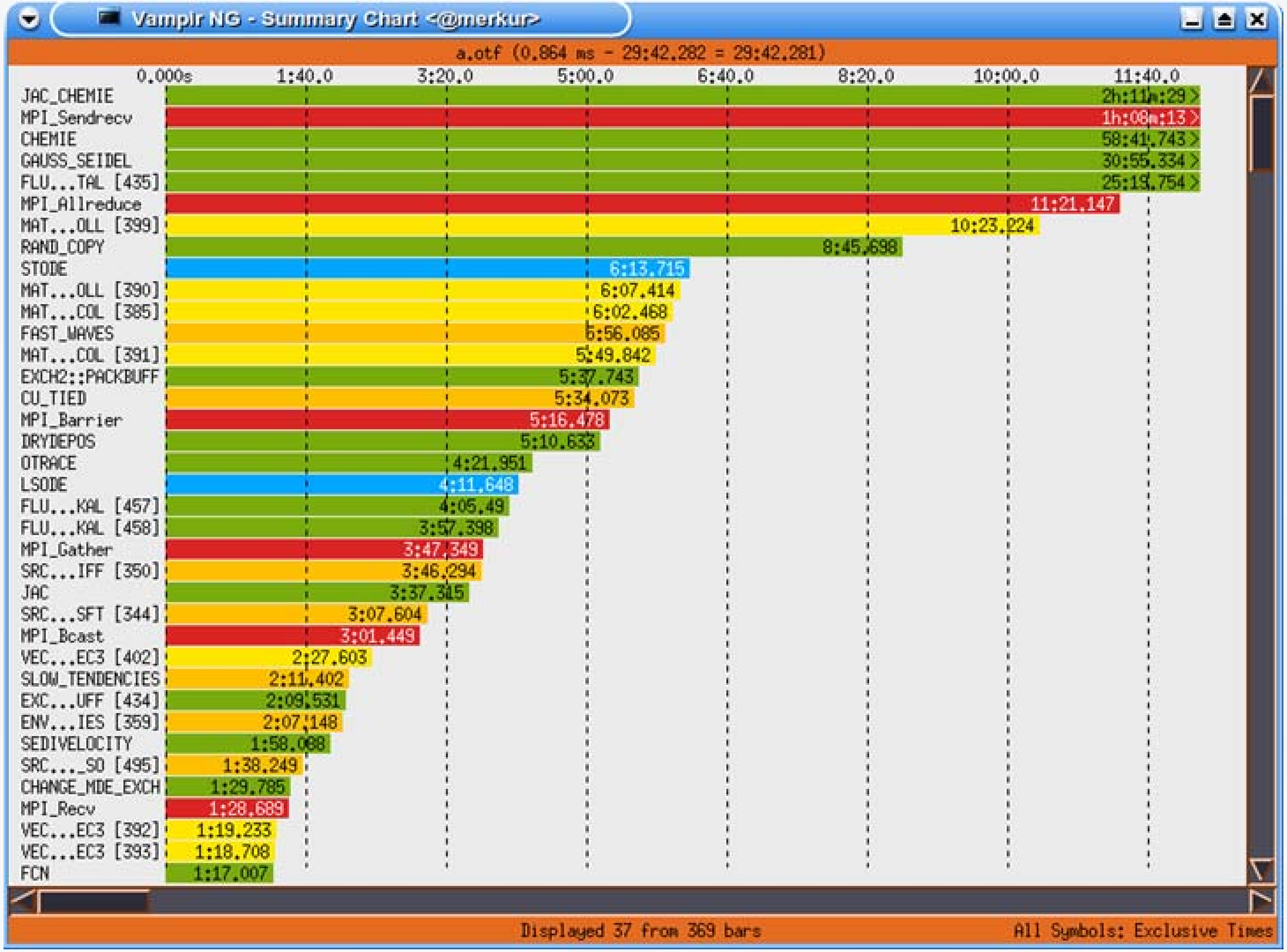


Vampir: Process Timeline

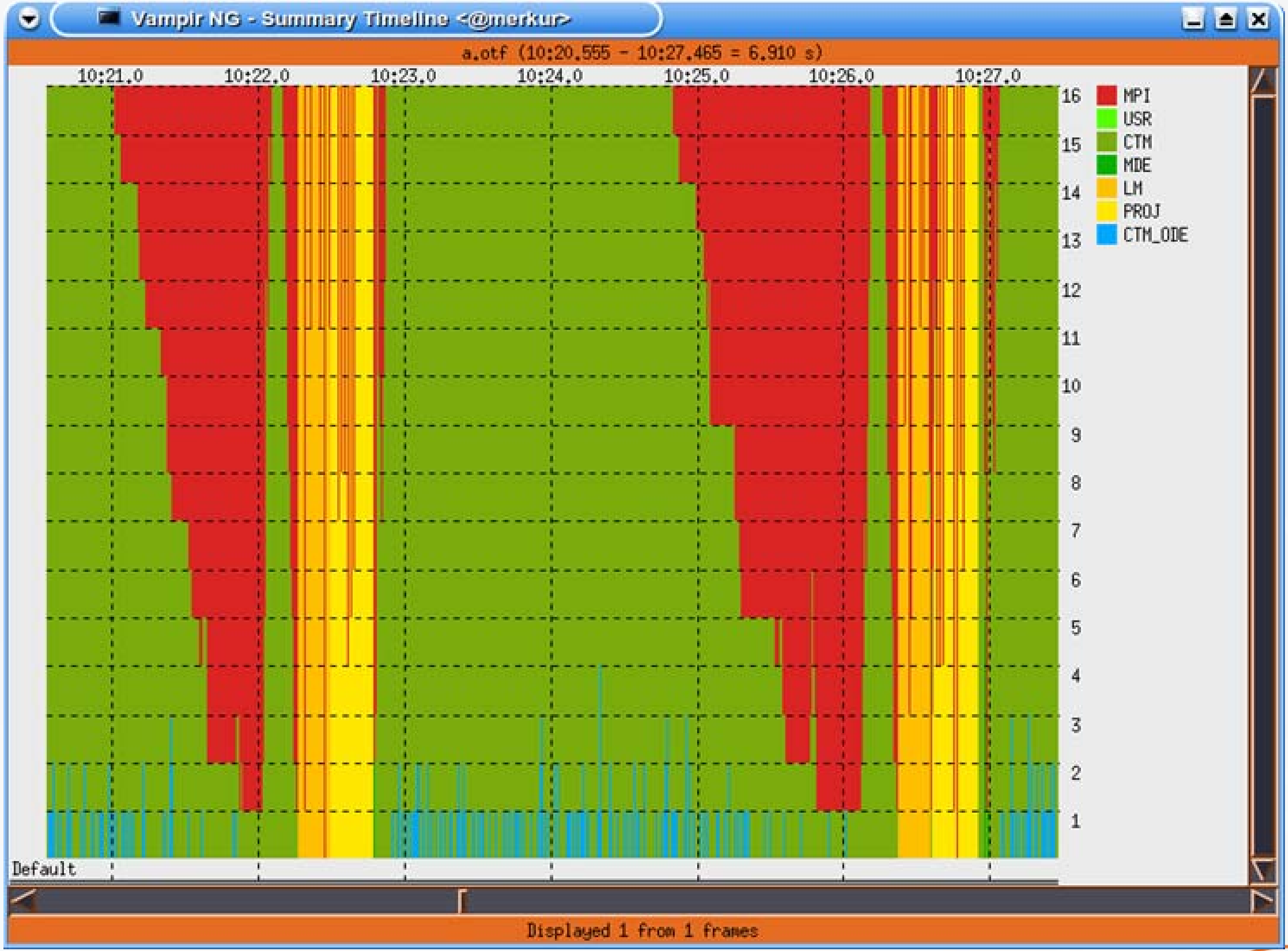
Vampir: Process Timeline



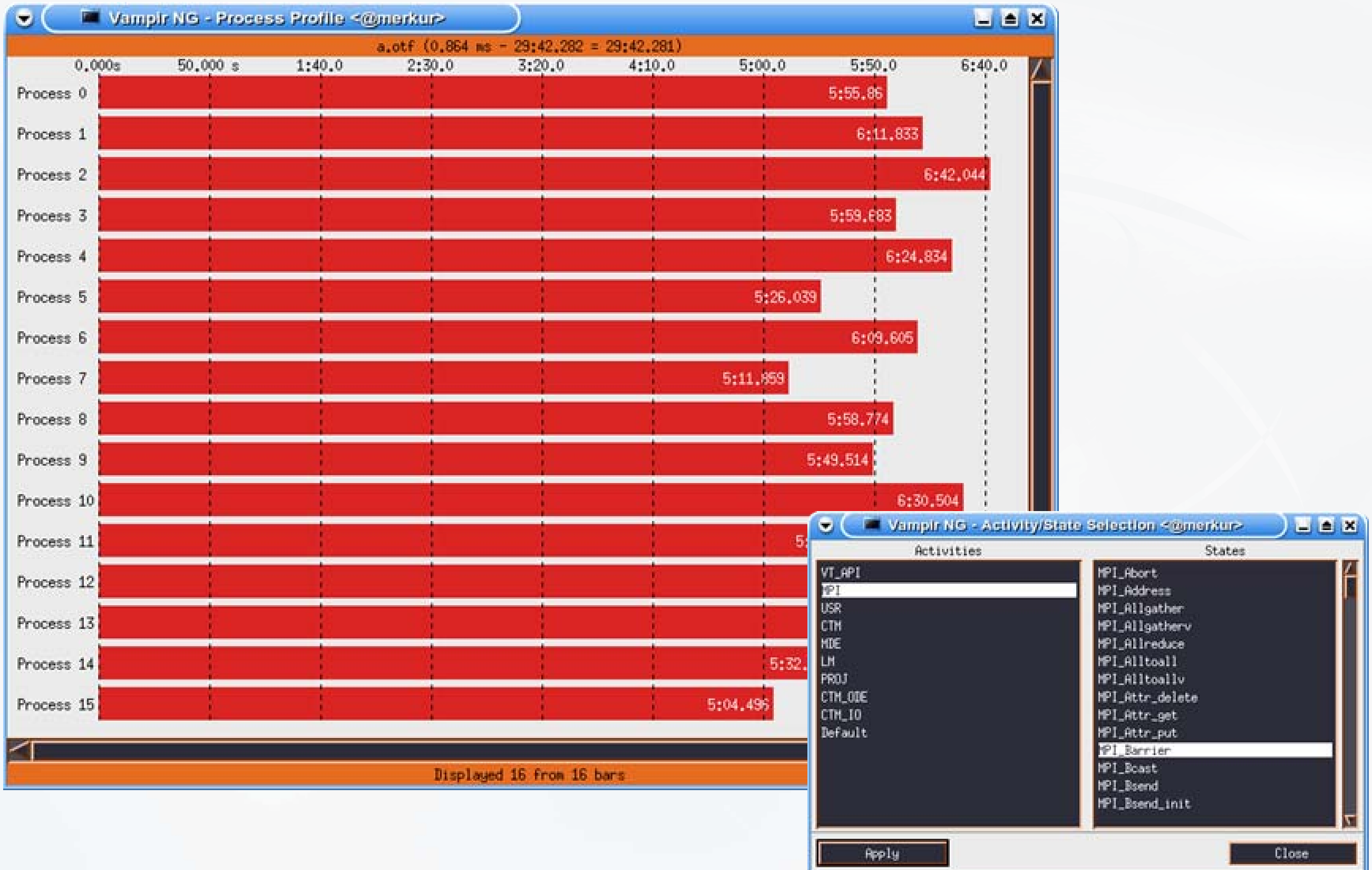
Vampir: Summary Chart



Vampir: Summary Timeline



Vampir: Process Profile

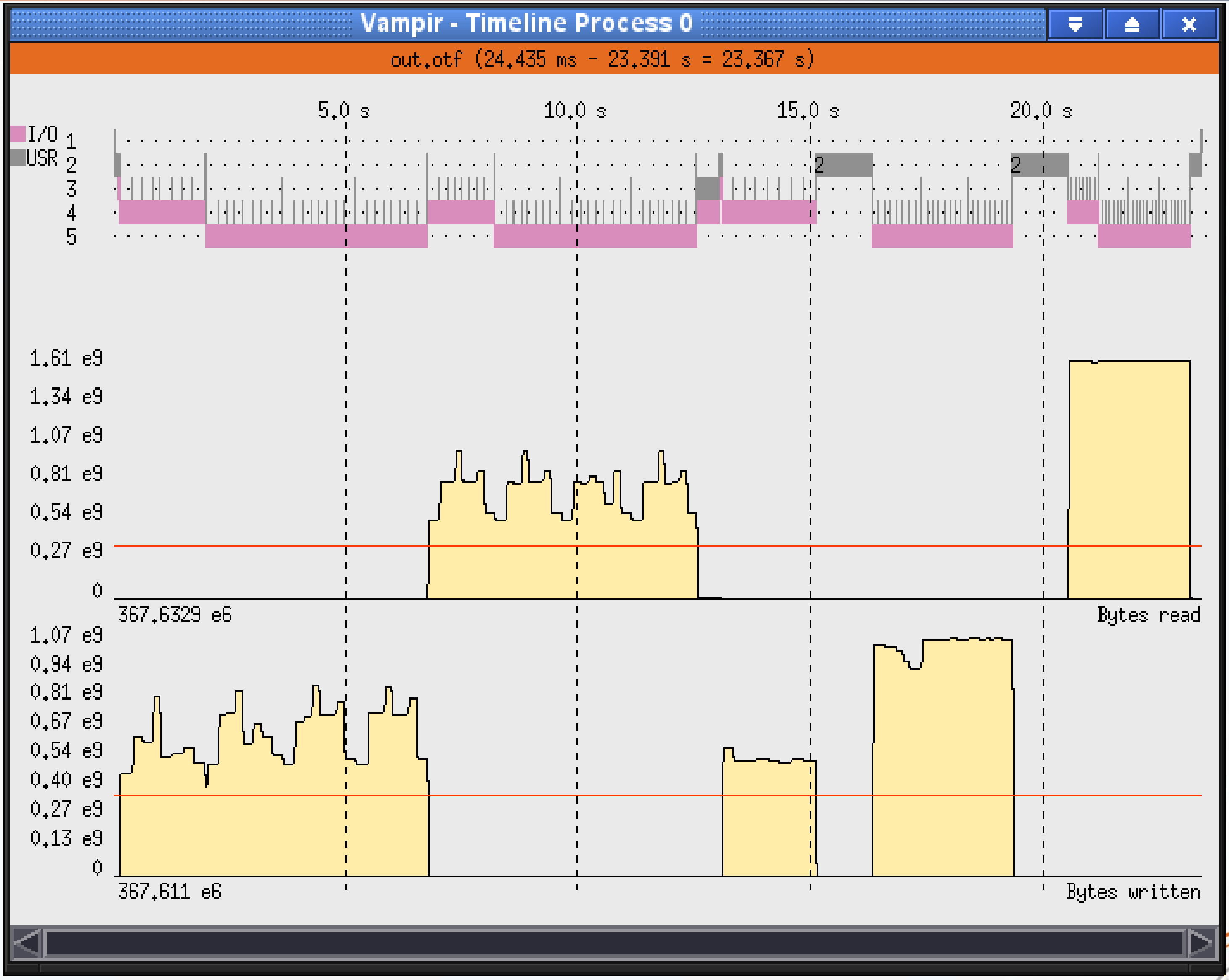


Focus I/O: Getting Data from the Application

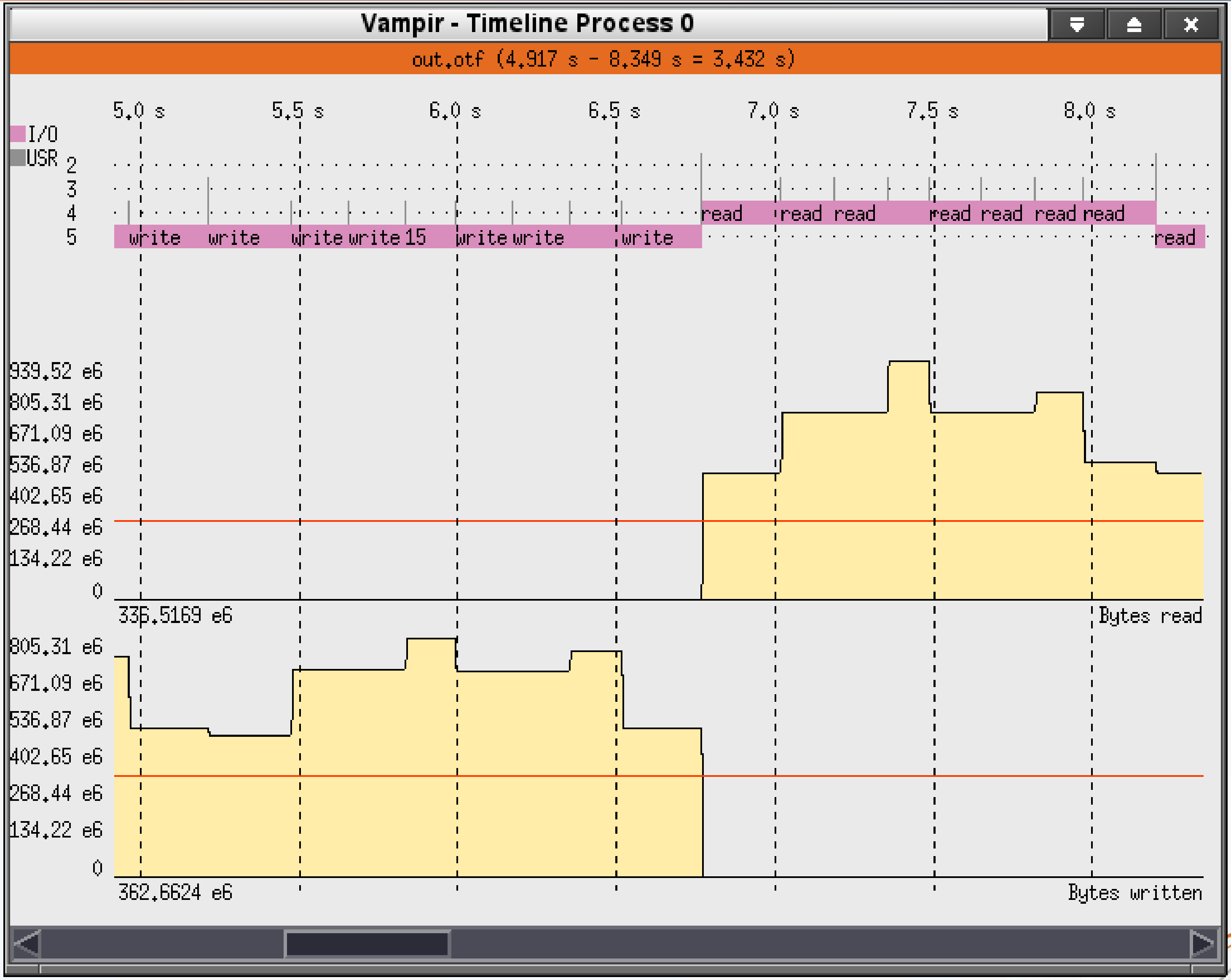
- Catching all I/O calls from the application
  - Adding them to the trace as performance counter data
  - Include filenames, offsets, and request sizes as OTF comments
  - Currently done with LD_PRELOAD library
  - Data needs to be merged with application OTF trace
  - Captured data: open (filename, filedescriptor), read/write (filedescriptor, size), close/dup (filedescriptor), seek (filedescriptor, position)

- Tracing I/O requests within the kernel
  - To follow the path of the request to the devices

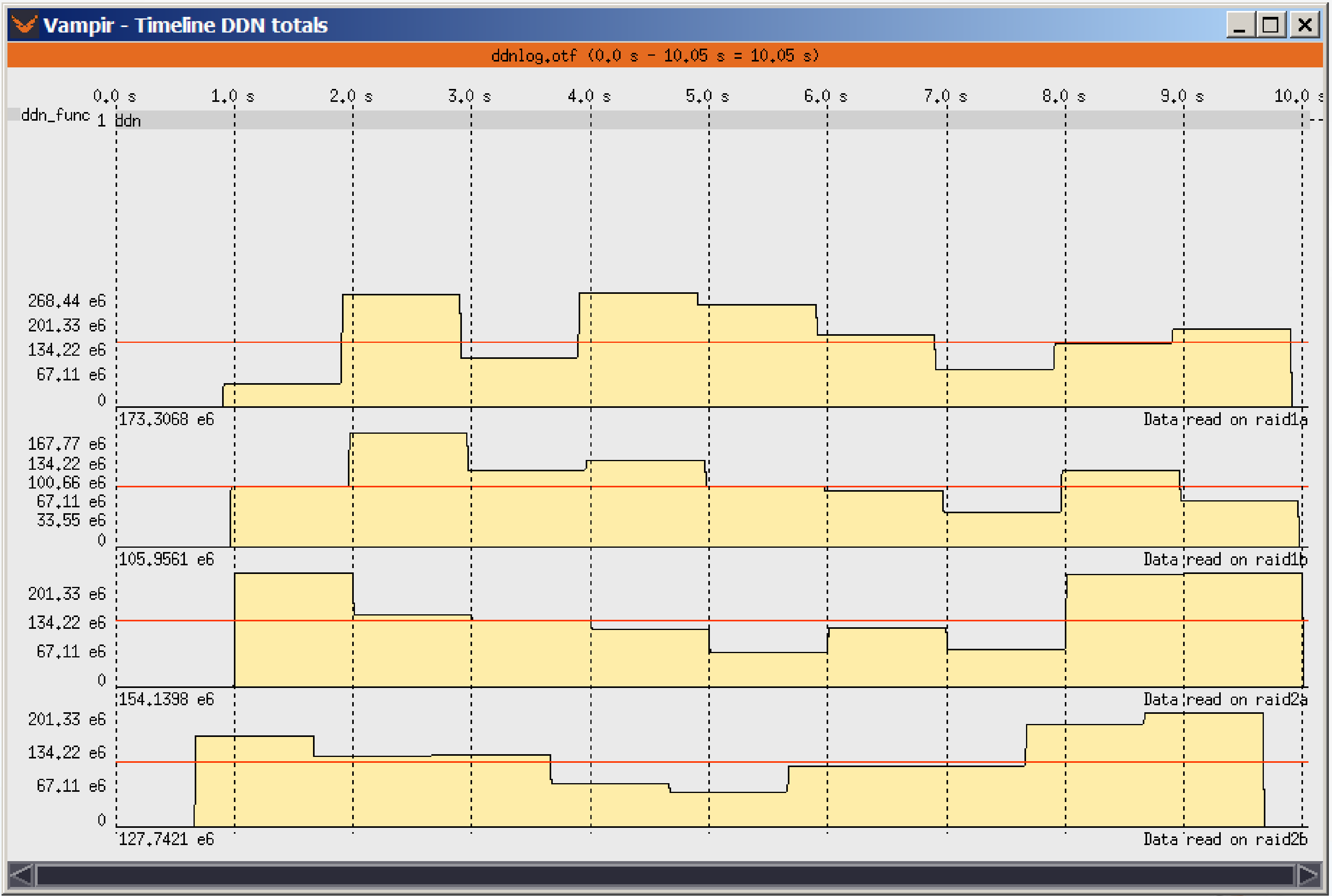
Vampir I/O Stats per Process (work in progress)



Vampir I/O Stats per Process (work in progress)



Including DDN Statistics (work in progress)



Perspective for Vampir and Vampir NG

- Parallel systems will become larger and cheaper!
- Software tools will become even more important!
- Difference between peak performance and sustained performance is huge
- Necessity for performance optimization increases with peak performance
- Unfortunately, complexity of the tools increases, too!
- Tool development today has to focus on computer architecture from tomorrow
- We will keep Vampir and Vampir NG as portable tools in the market

Activities in the VI-HPS:

we will strongly focus on an Integrated Tool Ecosystem

Thanks and Contacts

- Vampir/OTF
  - Holger Brunst, Heike Jagode, Hartmut Mix , Reinhard Neumann, …
  - Andreas Knüpfer, Matthias Jurenz, …
- I/O Tracing Facilities
  - Guido Juckeland, Michael Kluge
  - Holger Mickler
- Overall responsibility
  - Dr. Matthias Müller
- Many many more

Visit us at `www.tu-dresden.de/zih` and `www.vampir.eu`