TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Developing Scalable Applications with Vampir

Matthias Müller, Holger Brunst, Matthias Jurenz, Andreas Knüpfer, Matthias Lieber, Hartmut Mix, Wolfgang E. Nagel

Zellescher Weg 12

Willers-Bau A113

Tel. +49 351 - 463 - 39835

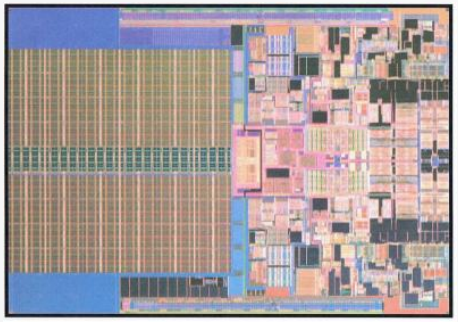
Matthias S. Mueller (matthias.mueller@tu-dresden.de)

ZIH
Center for Information Services & High Performance Computing

Outline

- Vampir, VampirServer and VampirTrace
- Applications used for evaluation
- Results

Motivation



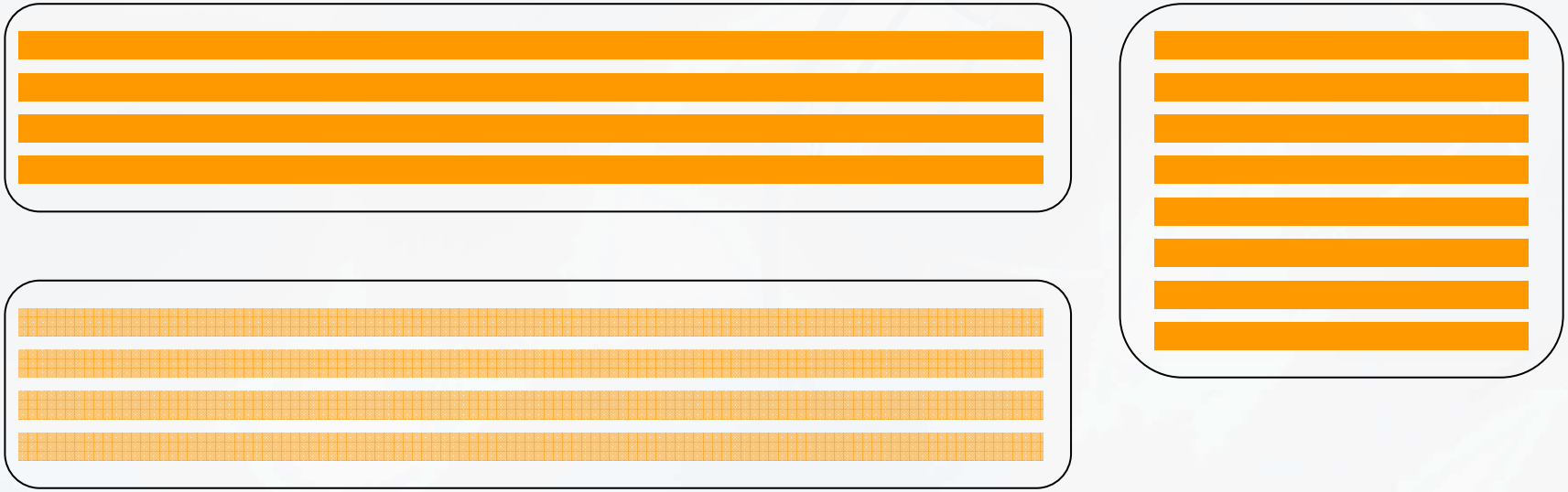
2 cores

131072 cores

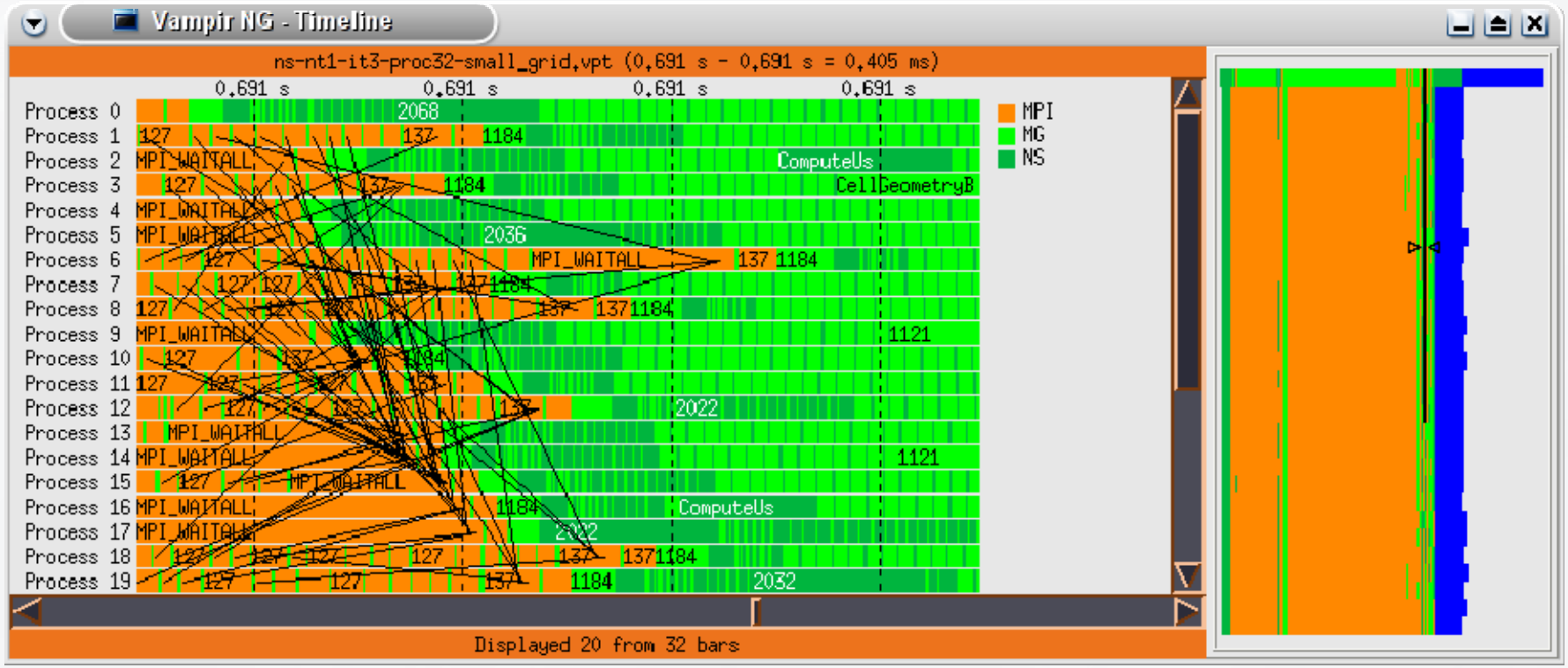
Scalability is not just about number of cores

Challenge for Vampir depends on

- Number of cores
- Run time
- “density” of performance events to capture

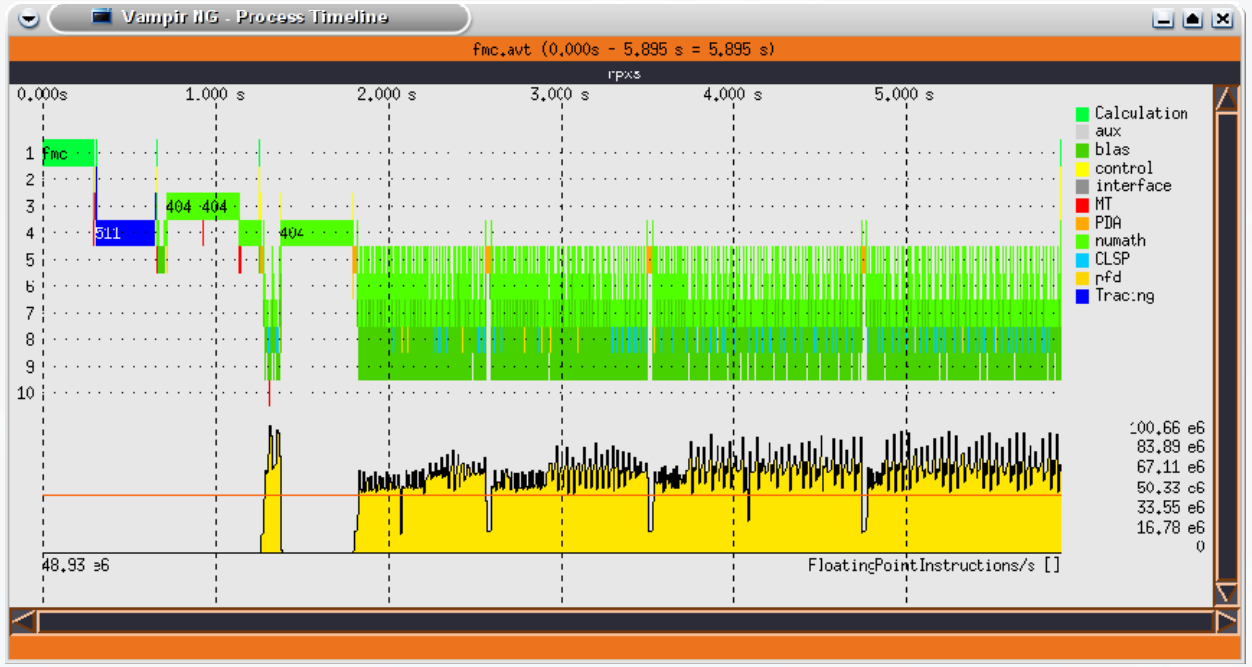


Vampir Displays - Global Timeline with Thumbnail

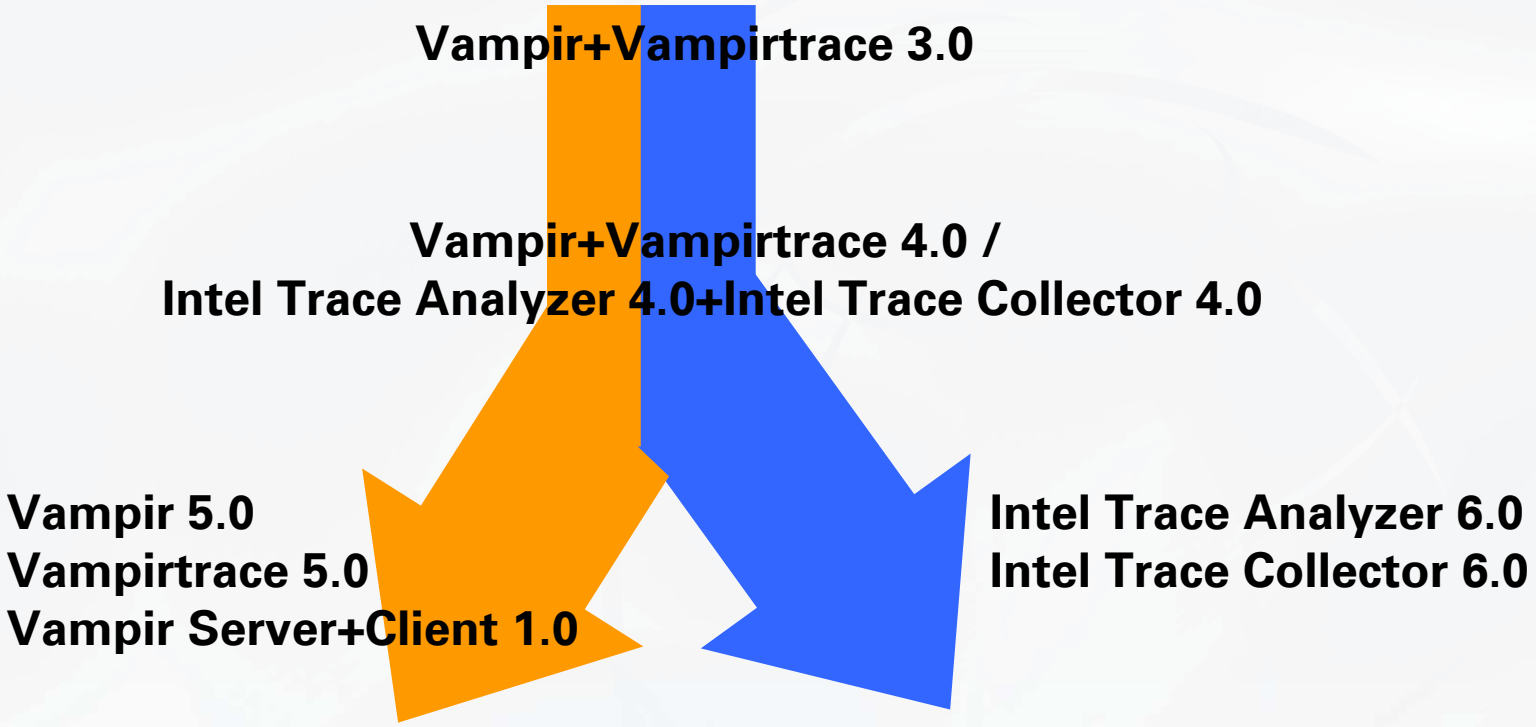


Use and abuse of performance analysis tools

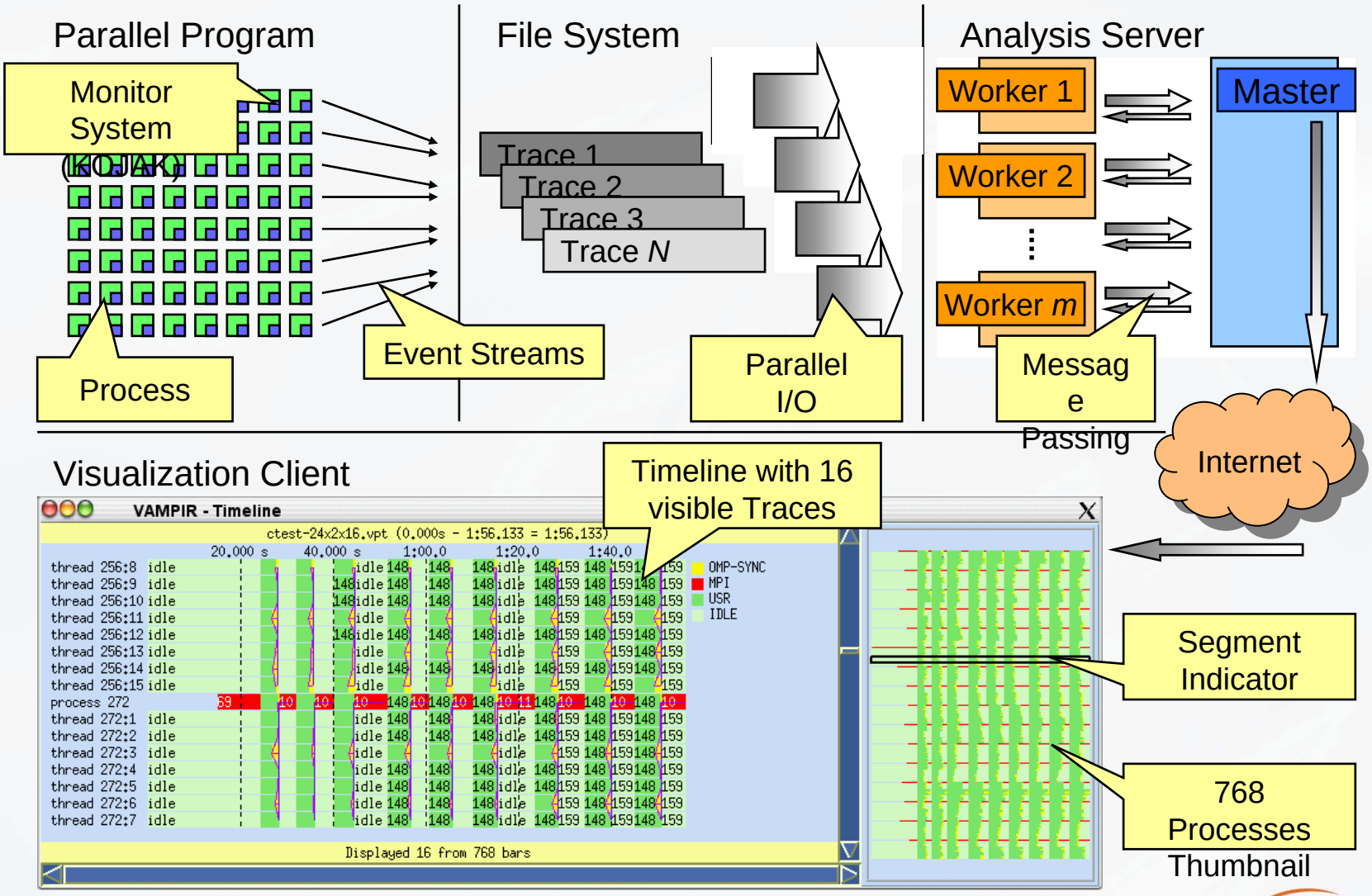
- We also use performance analysis tools for other things:
  - Finding Bugs
  - “Reverse Engineering”
  - Checking to see that what the code is doing is what the application developer think the code is doing



Vampir History



Vampir Server workflow



VampirTrace

- VampirTrace 5.x,y supports VTF and OTF
- Library implemented in collaboration with FZ Juelich and Univ. Oregon
  - OTF (developed with the TAU team at Univ. Oregon)
  - Epilog MPI wrappers (FZ Juelich)
  - Hardware performance counters (PAPI)
- Available as open source under BSD license:
  http://www.tu-dresden.de/zih/vampirtrace

Design of Open Trace Format

- Design of OTF is directed at 3 objectives:

- **Openness**

  ⇨ open format defines record types and file structure so that OTF traces can be generated and read correctly

  ⇨ external wishes will be considered .. just talk to us!

- **Flexibility**

  ⇨ efficiently selective access is supported

- **Performance**

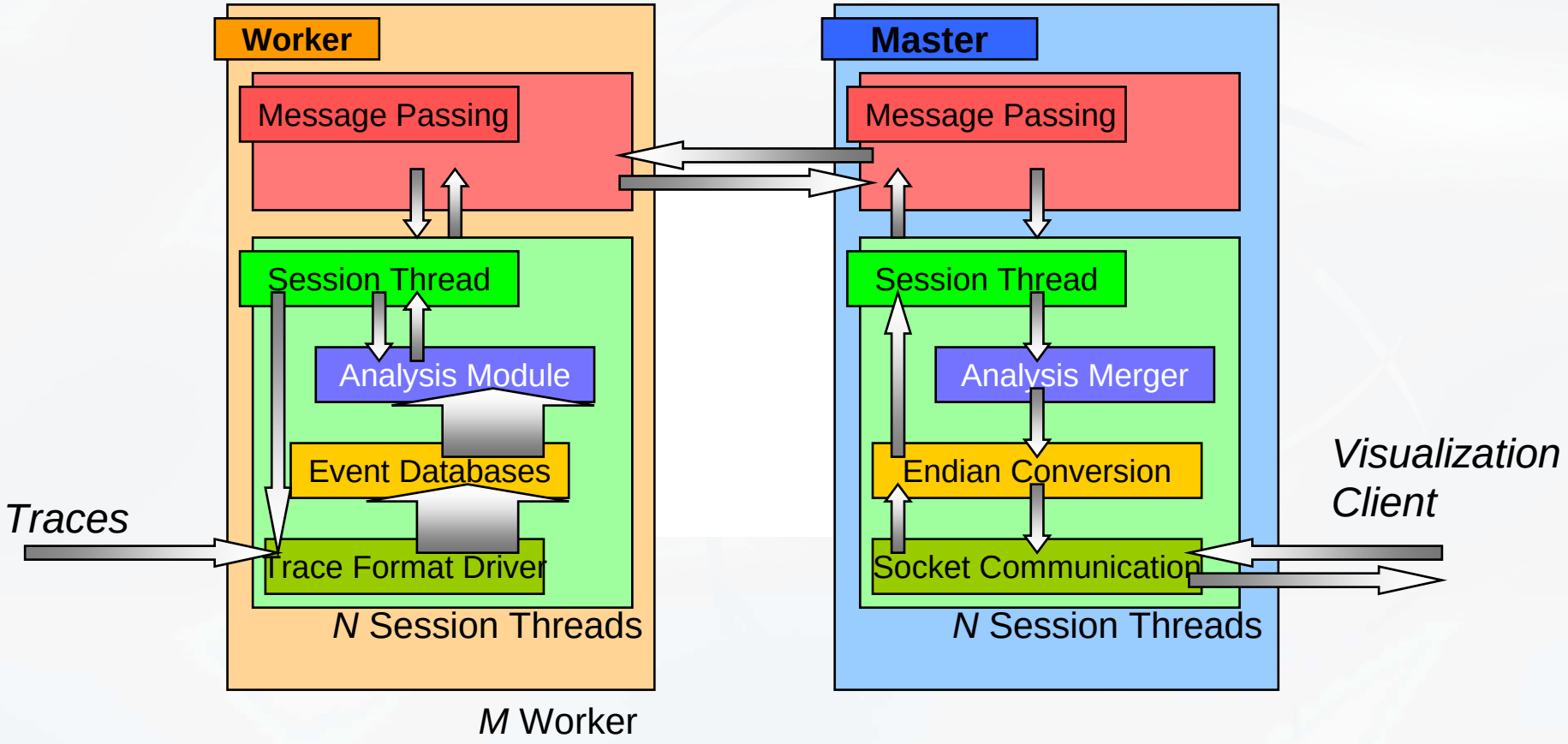
  ⇨ is determined by how efficient & fast OTF trace query and manipulation can be done

  ⇨ parallel I/O

OTF Features

- Supports fast and selective access to large amount of performance trace data
- Based on a stream model ⇨ single separate units represent segments of the overall data
- OTF streams may contain multiple independent processes whereas on process belongs to a single stream exclusively
- **Encourages parallel I/O**
- Strictly sequential reading of parallel traces still supported
- **Allows transparent ZLib compression**

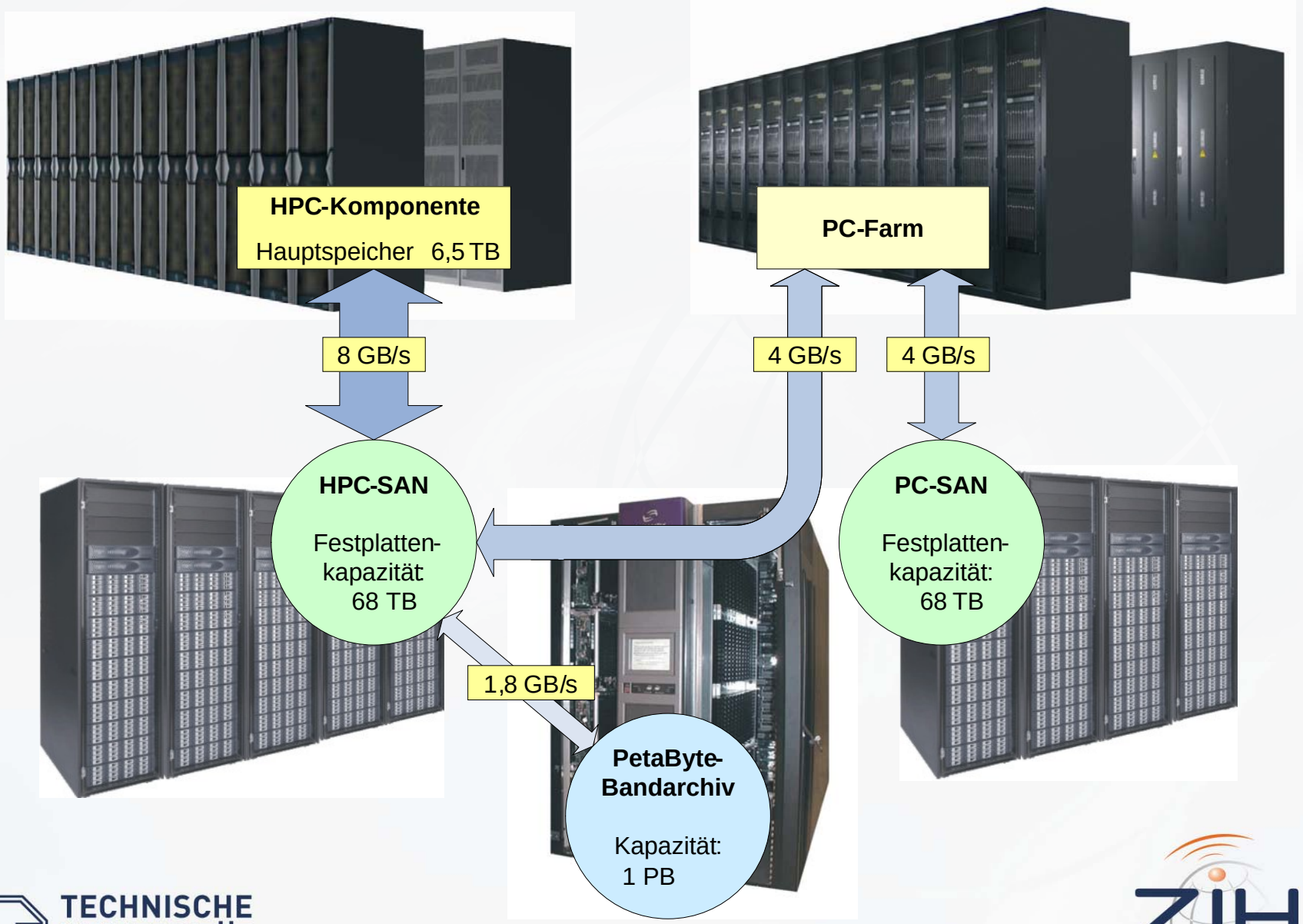
Parallel Analysis: Architecture of Vampir Server



Tools and Methods applied

1. Automatically instrumenting the source code by using compiler instrumentation (each function entry/exit)
2. Running a full instrumented run with a small dataset on 32 processors (with two hardware performance counters)
3. Analyzing with Vampir
4. Running a “production” run with only the MPI calls instrumented on 256 processors (with two hardware performance counters)
5. Analyzing with Vampir

Supercomputers at ZIH: measurement environment



Overview of the SPEC MPI2007 applications

Code	LOC	Language	MPI	MPI	Area
			call sites	calls	
104.milc	17987	C	51	18	Lattice QCD
107.leslie3d	10503	F77,F90	43	13	Combustion
113.GemsFDTD	21858	F90	237	16	Electrodynamic simulation
115.fds4	44524	F90,C	239	15	CFD
121.pop2	69203	F90	158	17	Geophysical fluid dynamics
122.tachyon	15512	C	17	16	Ray tracing
126.lammps	6796	C++	625	25	Molecular dynamics
127.wrf2	163462	F90,C	132	23	Weather forecast
128.GAPgeofem	30935	F77,C	58	18	Geophysical FEM
129.tera_tf	6468	F90	42	13	Eulerian hydrodynamics
130.socorro	91585	F90	155	20	density-functional theory
132.zeusmp2	44441	C,F90	639	21	Astrophysical CFD
137.lu	5671	F90	72	13	SSOR

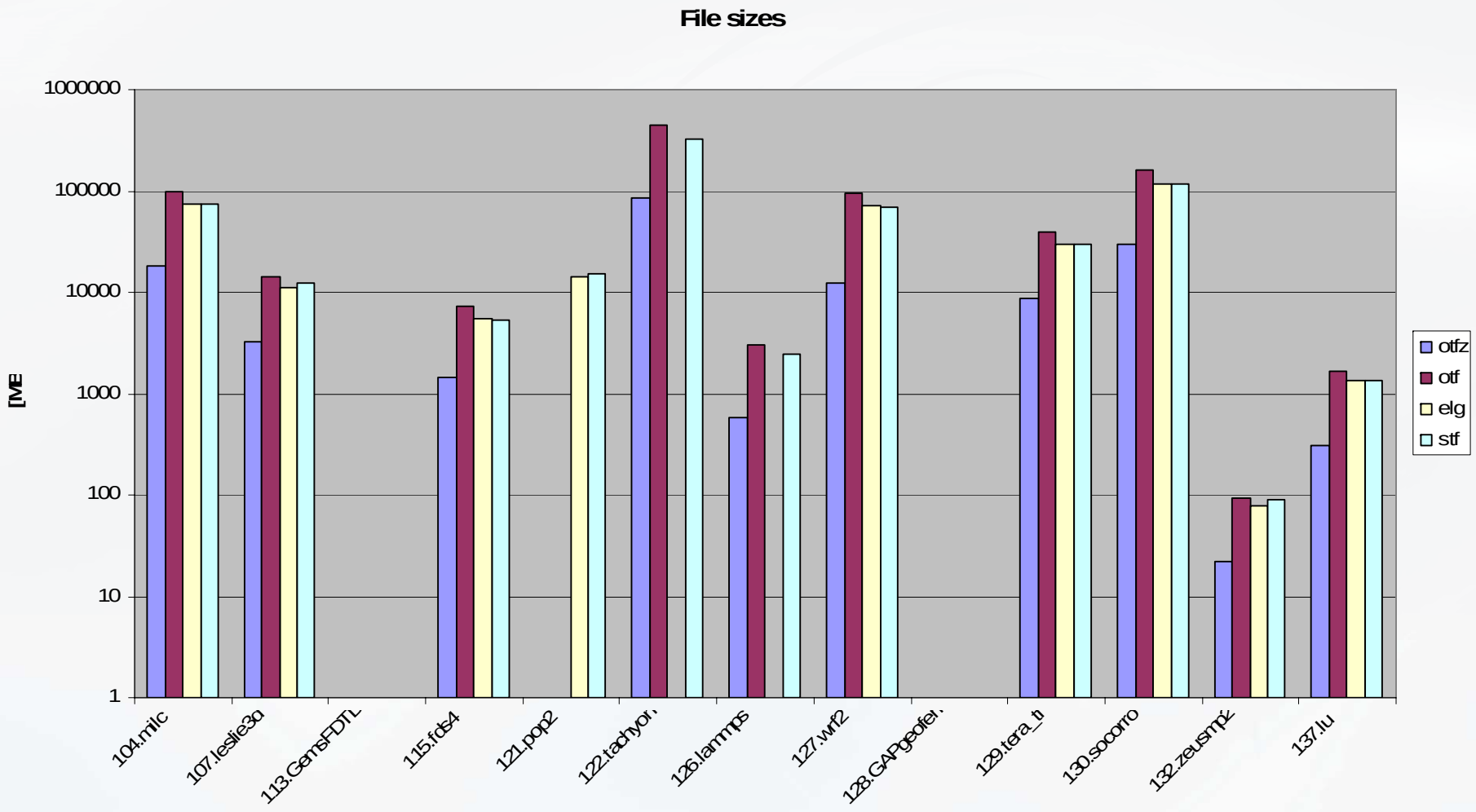
Datasets of SPEC MPI2007

- Each application comes with three data sets
  - Test
  - Train
  - Ref
- Ref dataset runs on up to 512 processors
- We used the train dataset on 32 processors for the full instrumented run
- We used the ref dataet on 256 processors for the MPI only run
- Vampir Server was running on 33 (32+1) processors

Examined Metrics

- Trace File Size
- Event rates
- Tracing Overhead
- Loading time
- Response time of Vampir Server

File sizes


File sizes
1000000
100000
10000
1000
100
10
1
otfz
otf
elg
stf
104.milc
107.leslie3d
113.GemsFDTL
115.fds4
121.pop2
122.tachyon
126.lammps
127.wrf2
128.GAPgeofem
129.tera_tf
130.socorro
132.zeusmp2
137.lu

Summary of trace file properties

- Uncompressed trace file sizes between 7 Megabyte and 767 Gigabyte
- Between 1 million and more than 11 billion events
- OTF files with transparent zlib compression take between 13% and 26% of file space (compression ration between 4 and 8)
- Each performance events consumes between 17 and 36 bytes in the uncompressed trace file (rule of thumb 20 Bytes)

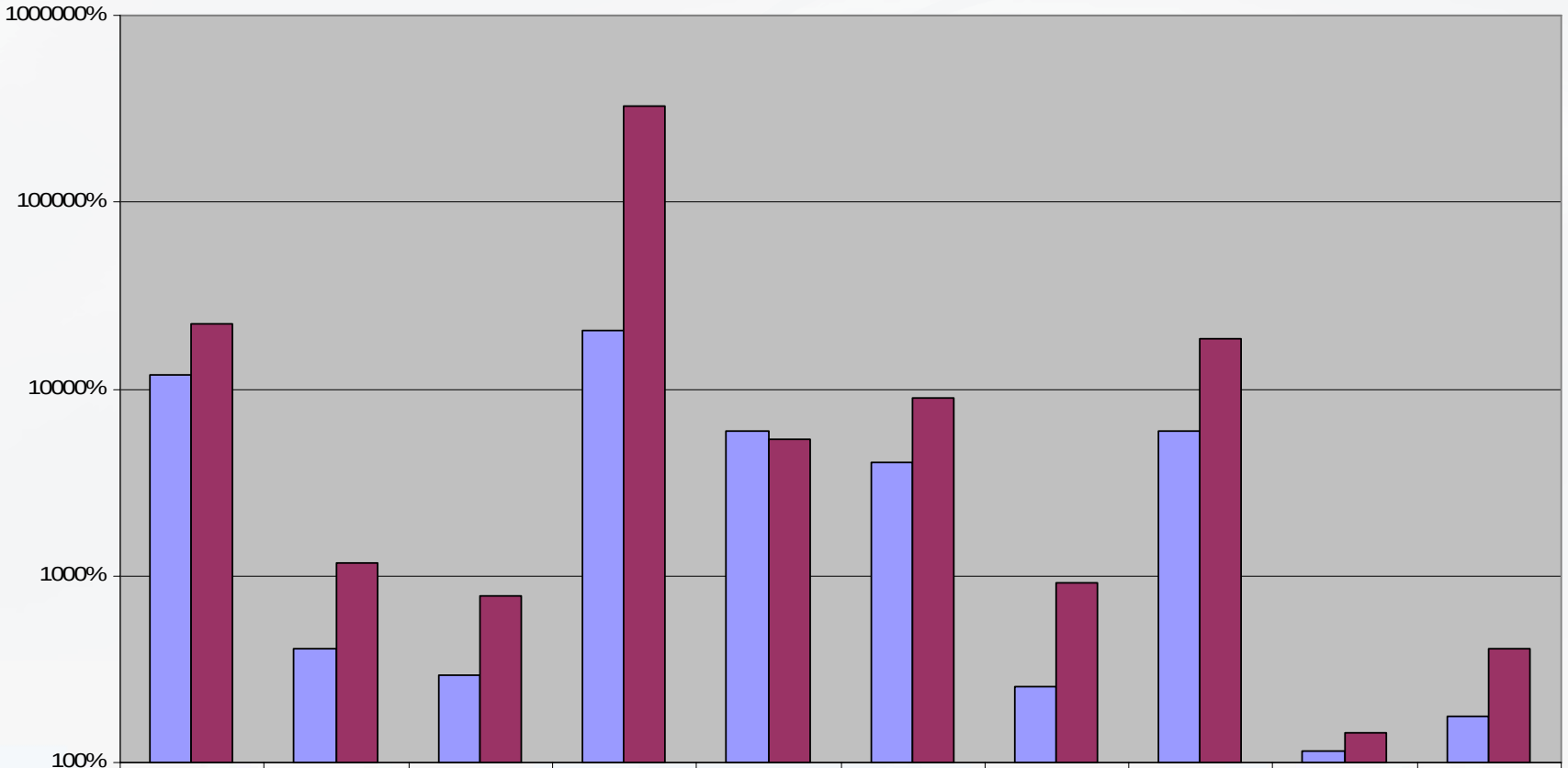
Full traces

	File size (MB)	Megaevents	Runtime	Megaevents/s
104.milc	98304	7214	9,1	792,76
107.leslie3d	14336	1022	24,5	41,71
115.fds4	7168	543	18,7	29,06
126.lammps	3073	235	36,1	6,51
127.wrf2	95232	6998	24,3	287,97
129.tera_tf	39936	2866	89,1	32,17
130.socorro	160768	11470	29,3	391,48
132.zeusmp2	94	7	25,7	0,26
137.lu	1638	118	12,4	9,50

MPI traces on 256 processes

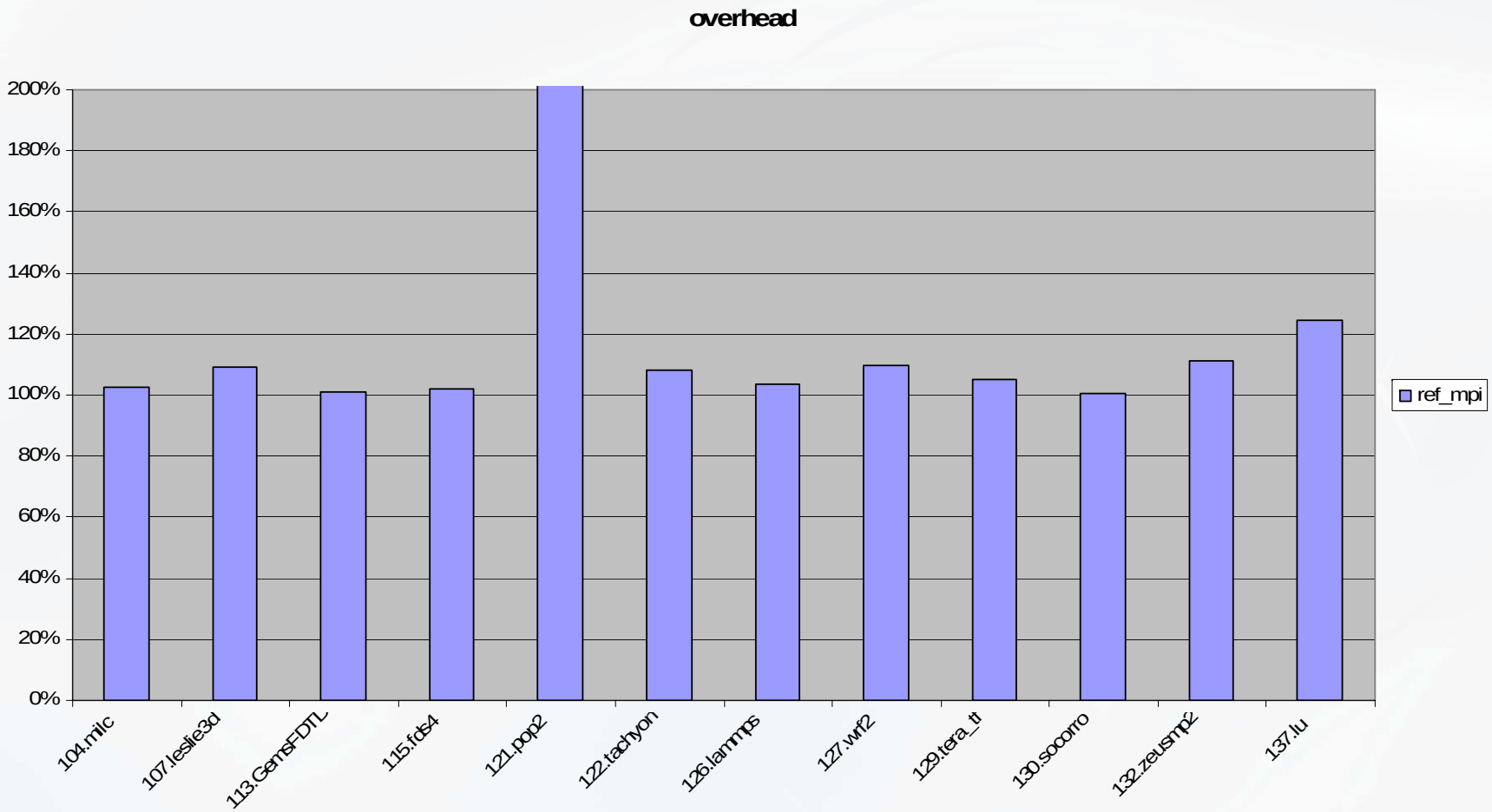
	File size (MB)	Megaevents	Runtime	Megaevents/s
104.milc	786	53	267	0,20
107.leslie3d	9318	572	192	2,98
113.GemsFDTD	1843	115	1281	0,09
115.fds4	263	16	605	0,03
121.pop2	766976	30	444	0,07
122.tachyon	19	1	264	0,00
126.lammps	284	18	493	0,04
127.wrf2	19456	1251	331	3,78
128.GAPgeofem	0	0	106	0,00
129.tera_tf	4090	213	290	0,74
130.socorro	8294	569	195	2,92
132.zeusmp2	2150	131	160	0,82
137.lu	15360	986	92,4	10,67

Overhead of fully instrumented run

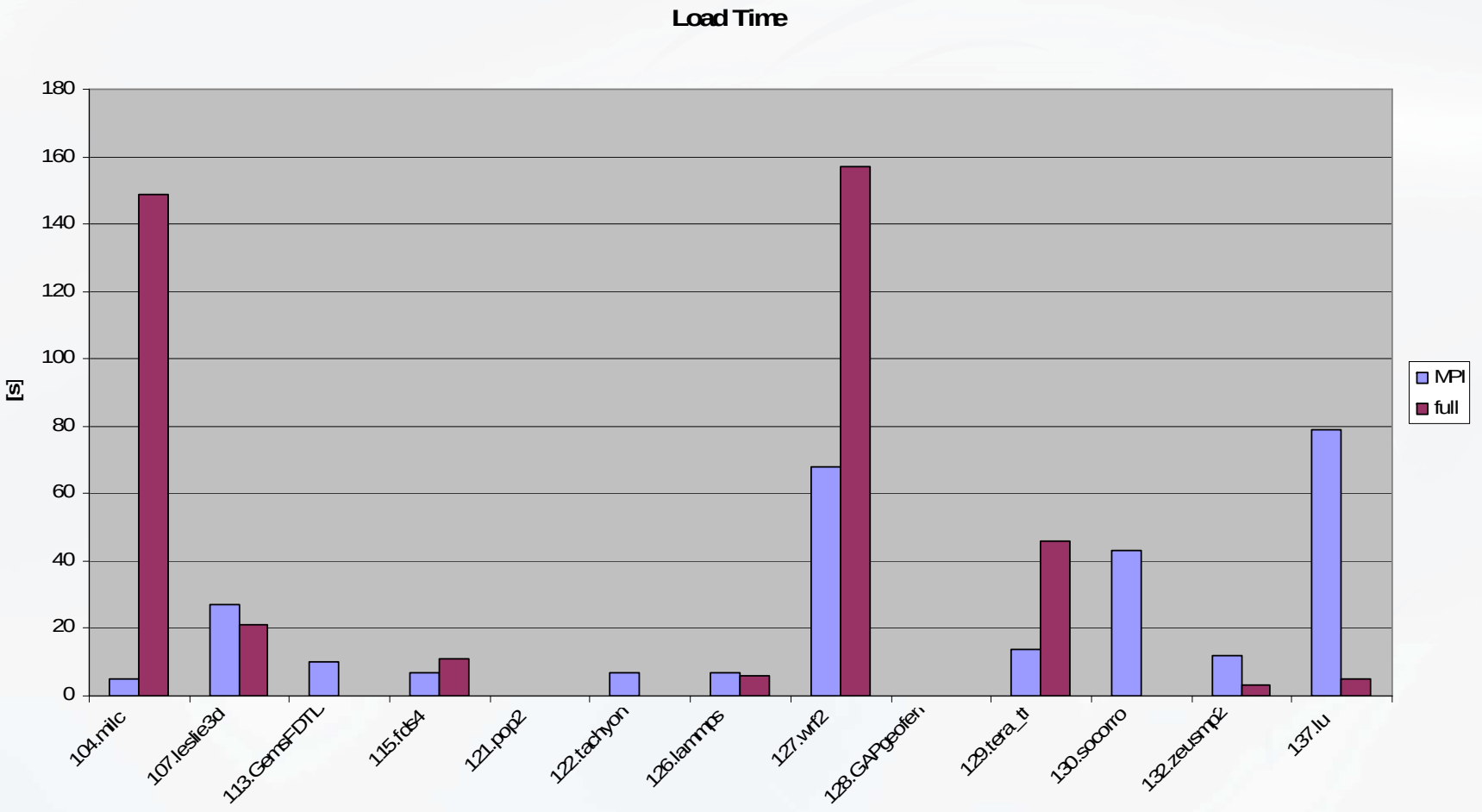


	104.milc	107.leslie3d	115.fds4	122.tachyon	126.lammps	127.wrf2	129.tera_tf	130.socorro	132.zeusmp2	137.lu
vt	11879%	408%	296%	20433%	5967%	4082%	254%	5990%	116%	178%
itc	22153,85%	1175,51%	786,10%	328181,10%	5426,59%	8979,42%	910,21%	18470,99%	143,19%	403,23%

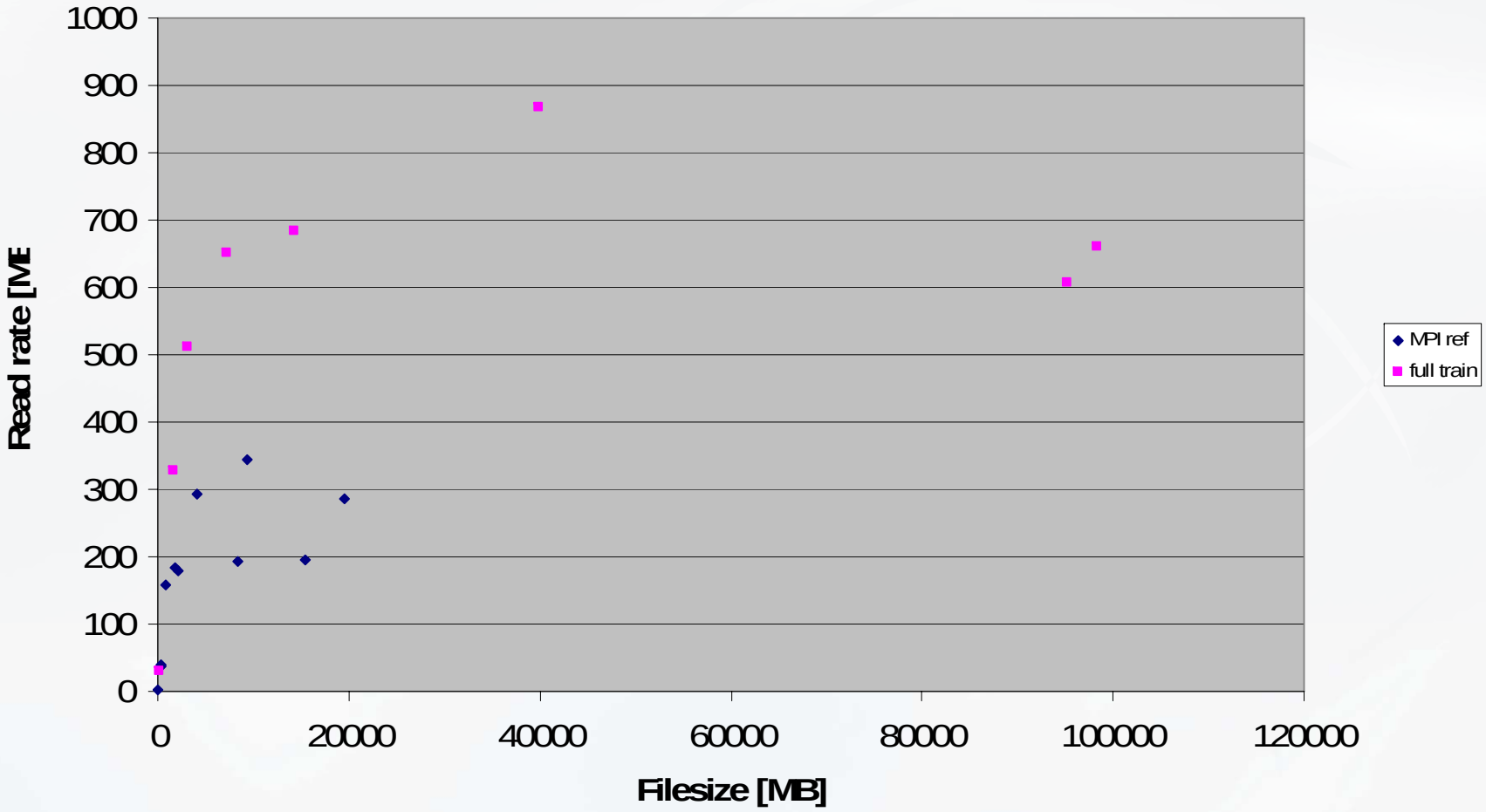
Overhead of MPI tracing



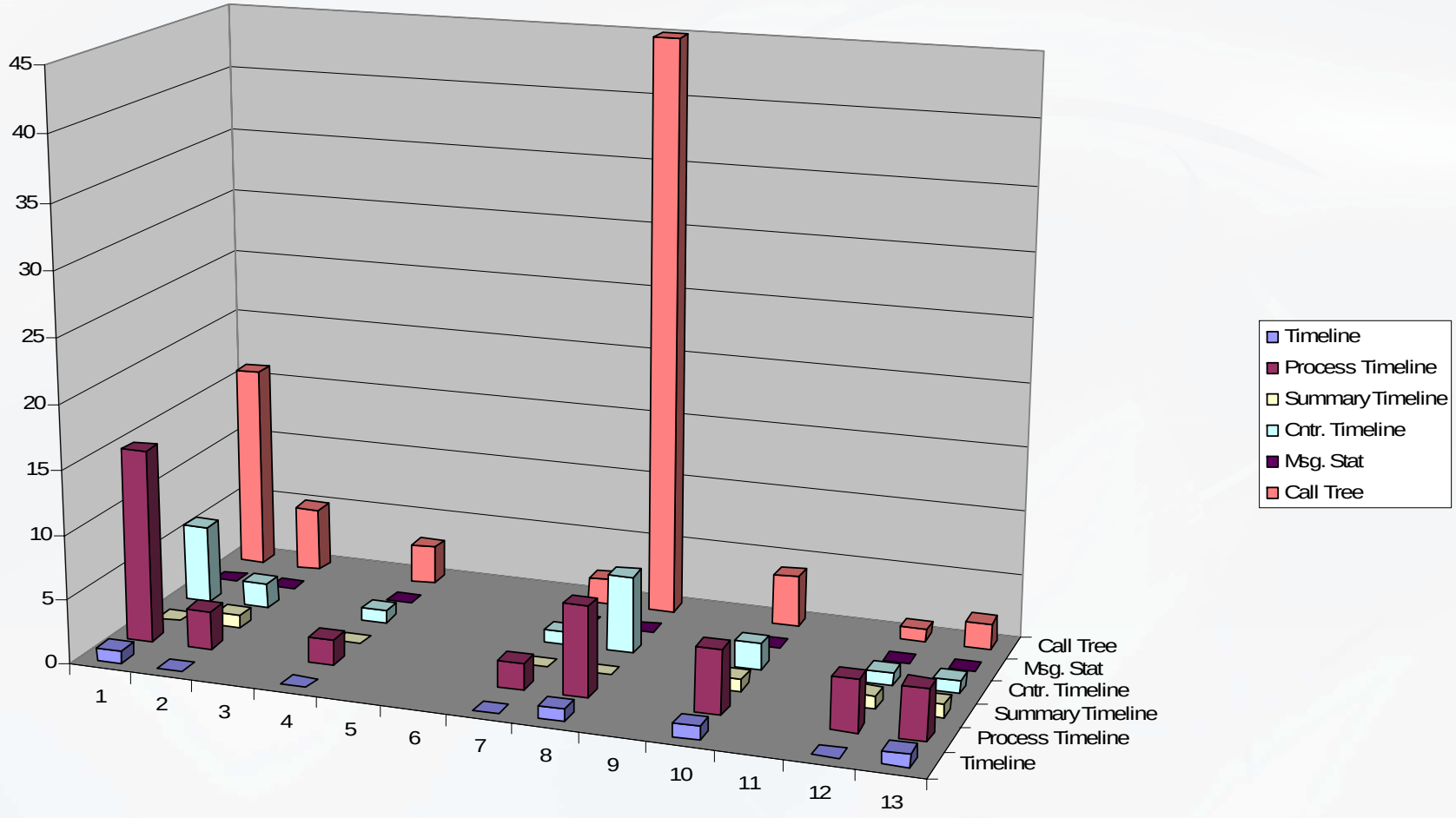
Load Time


Load Time
[s]
180
160
140
120
100
80
60
40
20
0
104.milc
107.leslie3d
113.GemsFDTL
115.fds4
121.pop2
122.tachyon
126.lammps
127.wrf2
128.GAPgeofem
129.tera_tf
130.socorro
132.zeusmp2
137.lu
MPI
full

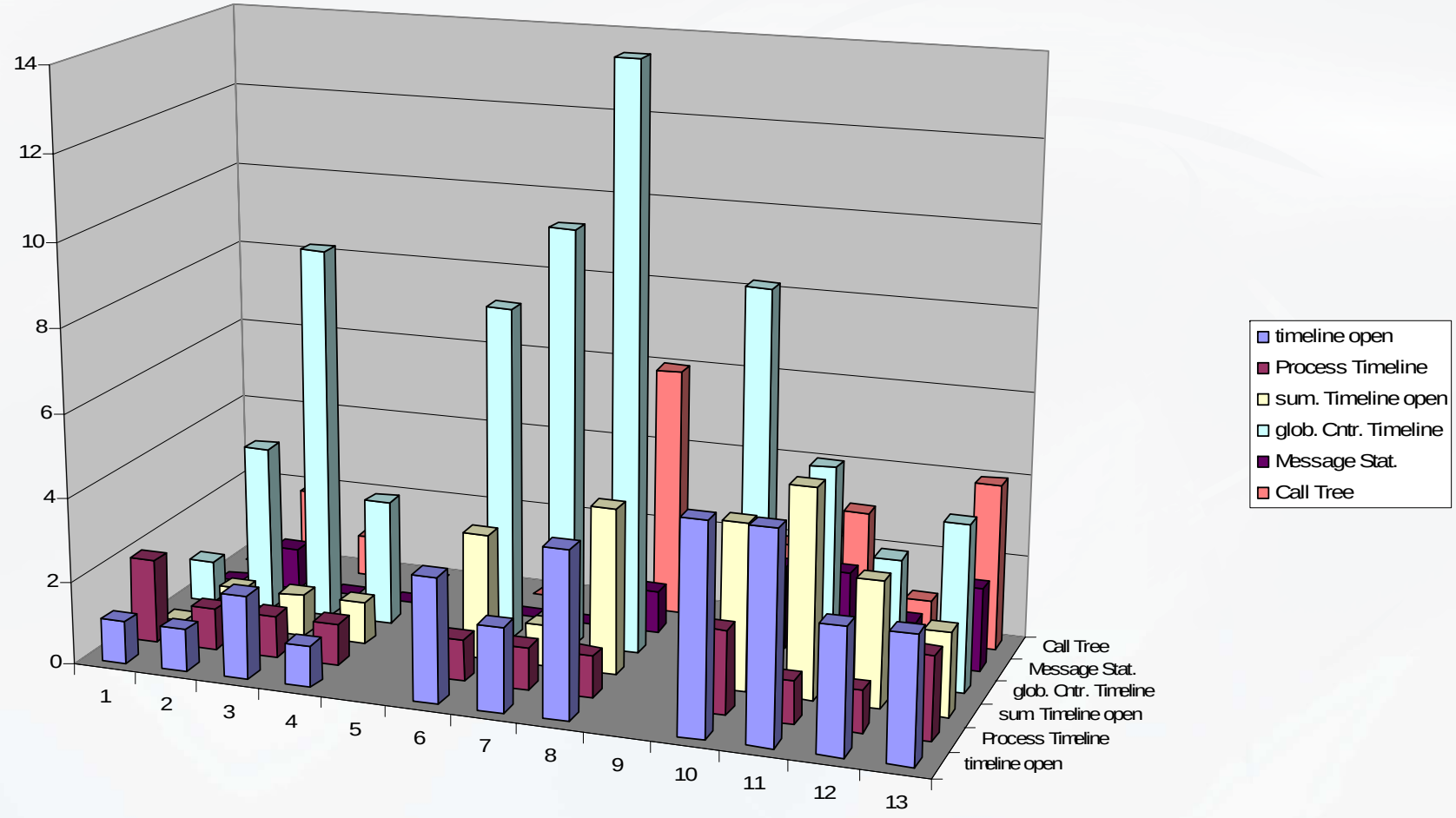
Read performance of Vampir Server



Responsiveness of Vampir Server (fully instrumented)



Responsiveness of Vampir Server (MPI instrumented)



TECHNISCHE UNIVERSITÄT DRESDEN

Future Direction

Zellescher Weg 12

Willers-Bau A113

Tel. +49 351 - 463 - 39835

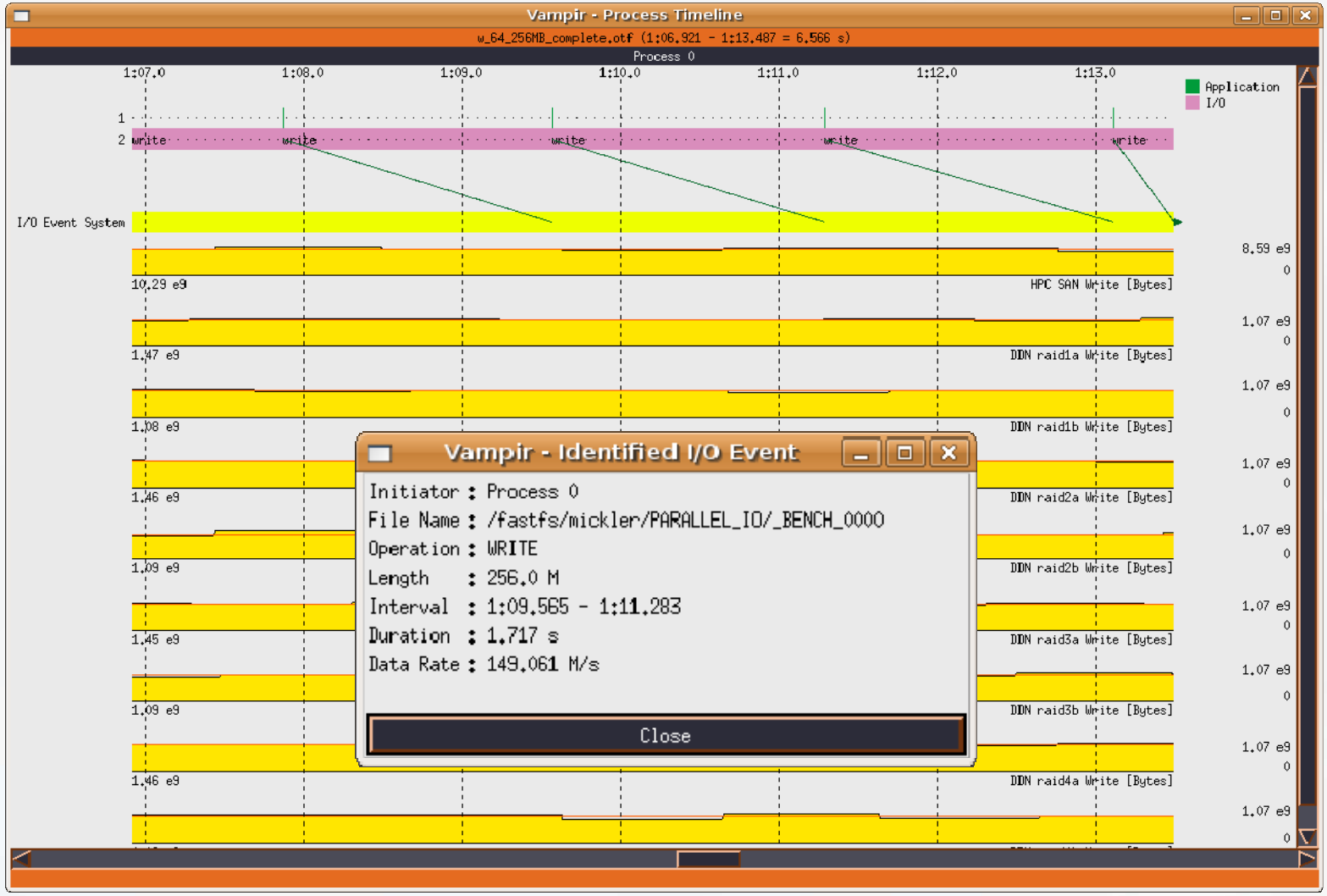
Matthias S. Mueller (matthias.mueller@tu-dresden.de)

ZIH
Center for Information Services & High Performance Computing

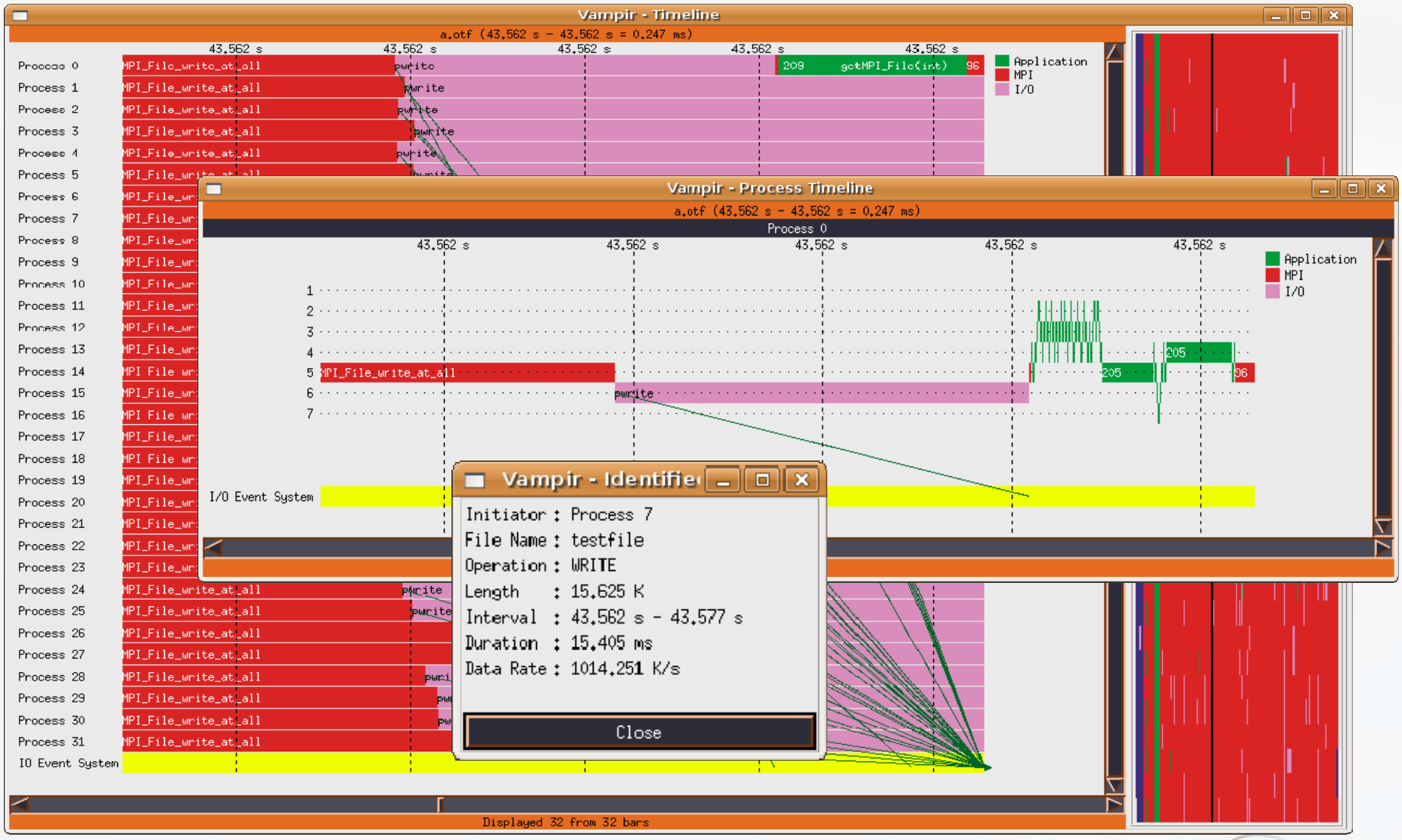
Future Steps

- Increased system monitoring
- I/O performance analysis (see next slides)
- Improvements for handling large traces (selective reads)
- Performance analysis for hybrid machines with accelerators
- Vampir on MS Windows
- Vampir integration into Eclipse ?
- …

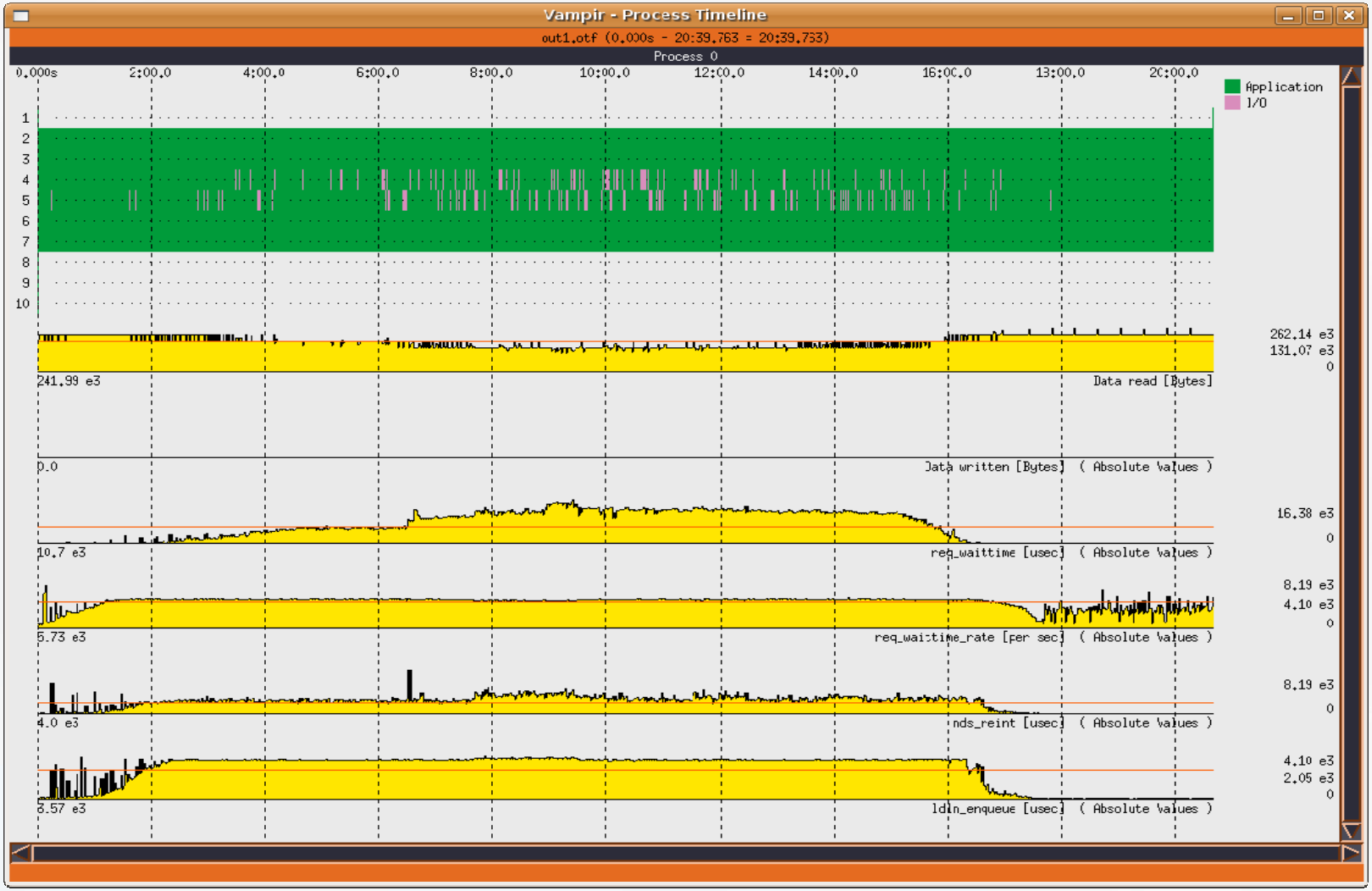
Example – I/O calls and SAN counters



Example – I/O calls inside MPI-I/O



Example – Lustre counters



Summary and conclusion

- VampirTrace and Vampir/VampirServer offer a reliable environment for manual performance analysis, covering many aspect relevant for performance
- Demonstrated scalability to analyze applications running on many processors and long runs with many performance events (up to 10 Billion events)
- Will it work for you?
  - #cores * runtime * event density * 20 Bytes/event < memory ??