

Tools to Analyze the Performance of Open MPI Programs

Jens Doleschal (jens.doleschal@tu-dresden.de)

Tel. +49 351 463 35876

Center for Information Services and HPC

Technische Universität Dresden

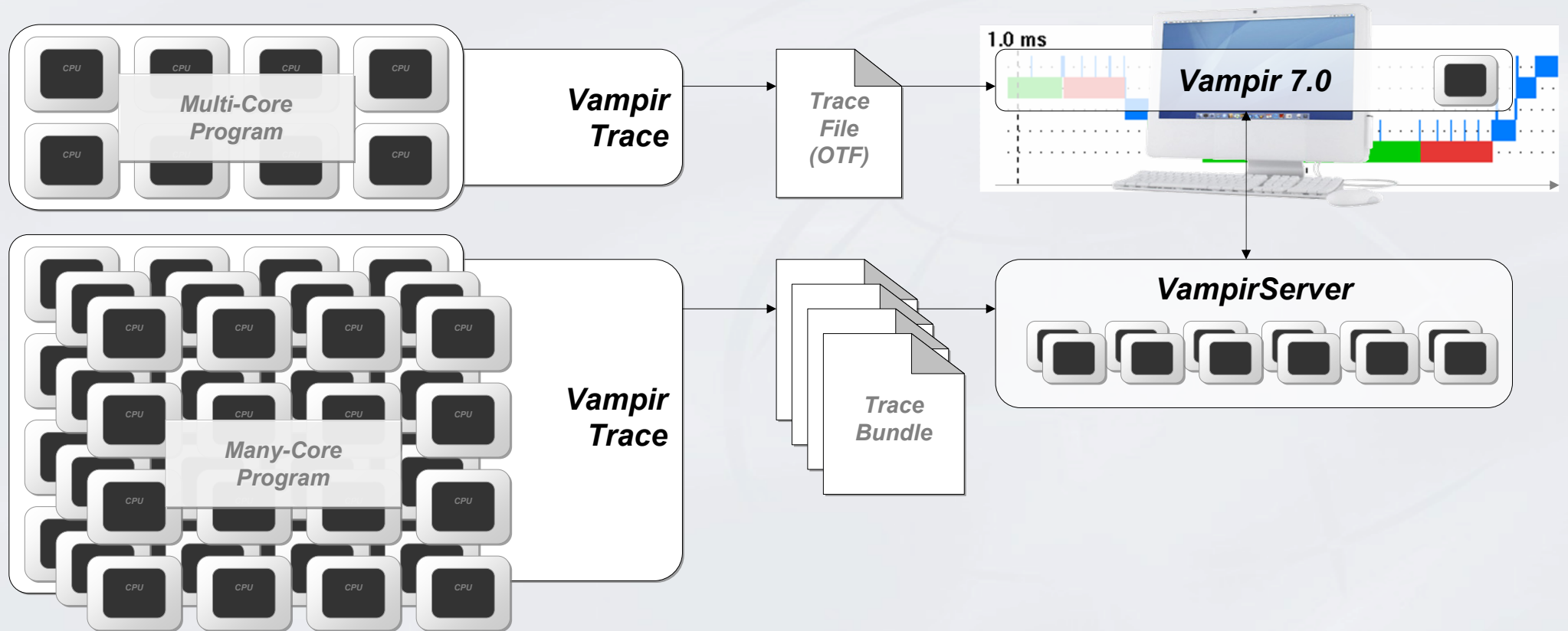
Germany

Motivation

- Parallel programming is about performance!
- Scaling to thousands of cores is required
- You need a decent MPI implementation, e.g. Open MPI
- You also need a ready-to-use performance monitoring and analysis tool
- Involves two components:
 - Data collector: measurement environment, e.g. VampirTrace
 - Data browser: performance visualizer, e.g. Vampir

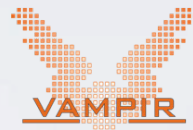


Performance Measurement and Analysis



Performance Analyzer and Tools using the Open Trace Format

- Non-commercial
 - VampirTrace (Integrated in Open MPI v1.3 or later)
 - OTF profiler
 - Scalasca
 - TAU
 - Open|Speedshop
- Commercial
 - Vampir
 - Sun Performance Analyzer
 - Microsoft HPC Pack 2008 SDK



Performance Analysis with OTF Profiler

Performance Analysis with OTF profiler

- Profiler that provides
 - flat profile,
 - message statistics,
 - summaries combined with metrics of your application run
- Scalable up to thousands of processes

Performance Analysis with OTF profiler

file:///home/dolescha/Desktop/bt_altix_5_papi.otf_result.ps - KGhostView

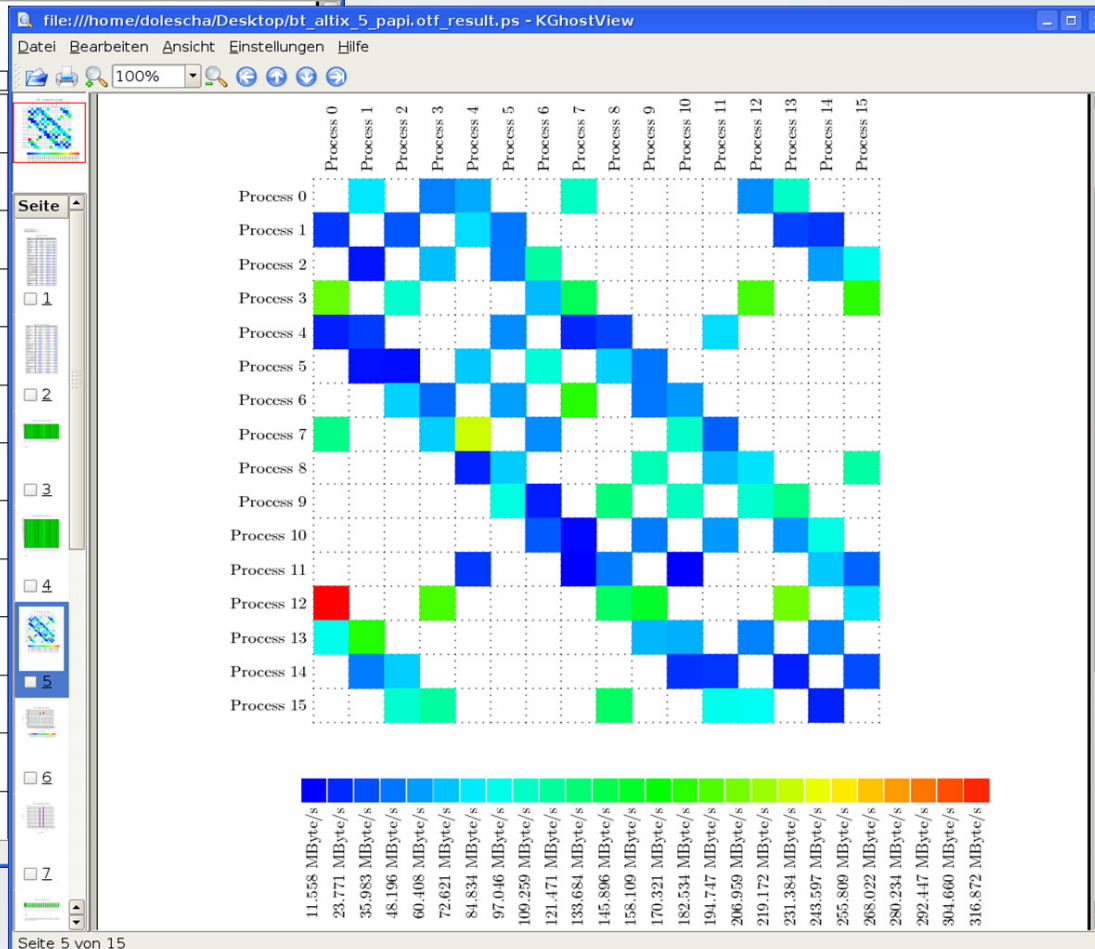
Datei Bearbeiten Ansicht Einstellungen Hilfe

100%

Top 50 of 67 Functions

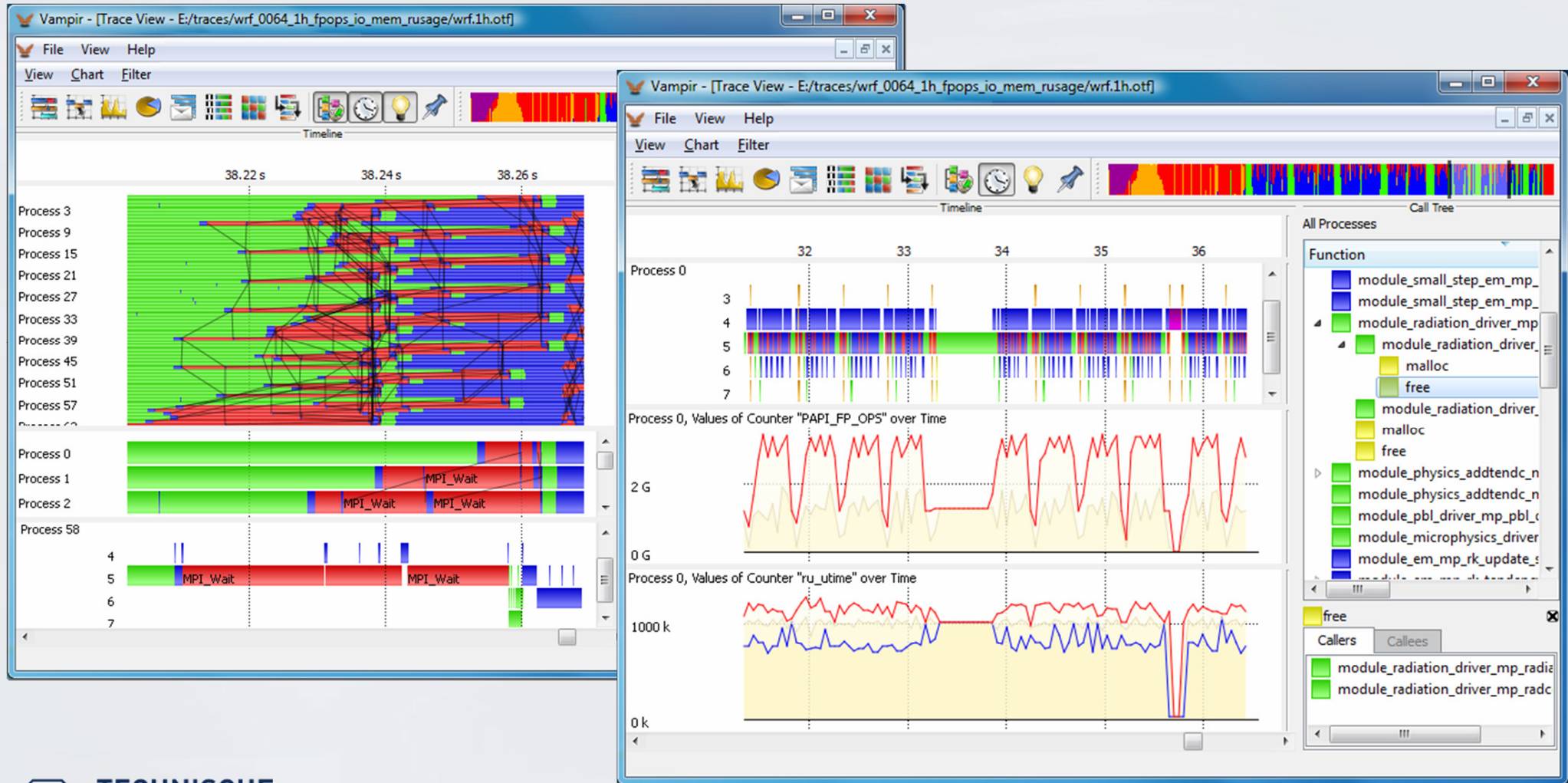
Function	invocations[#]	excl. time[sec]
MPI_Finalize	16	16.040951
sync time	32	11.194839
y_solve_cell_	12864	7.932583
z_solve_cell_	12864	7.924111
x_solve_cell_	12864	7.883323
MPI_Wait	115776	3.696342
binvrhs_	291852	0.853374
copy_faces_	3232	0.678426
exact_solution_	238896	0.673068
compute_rhs_	3232	0.591711
MPI_Irecv	77280	0.529846
MPI_Isend	77280	0.528854
MPI_Waitall	3232	0.490394
initialize_	32	0.476545
y_solve_	3216	0.379208
x_solve_	3216	0.378921
z_solve_	3216	0.378064
z_send_solve_info_	9648	0.317867
y_send_solve_info_	9648	0.290590
x_send_solve_info_	9648	0.180819
x_backsubstitute_	12864	0.161153
exact_rhs_	16	0.159253
y_backsubstitute_	12864	0.123173
z_backsubstitute_	12864	0.121998
lhsabinit_	38592	0.118691
MPI_Init	16	0.080417
z_send_backsub_info_	9648	0.075871
y_send_backsub_info_	9648	0.075121
x_send_backsub_info_	9648	0.067860
adi_	3216	0.057856
y_receive_solve_info	9648	0.056335
x_receive_solve_info	9648	0.055817
x_receive_backsub_in	9648	0.055490
z_receive_solve_info	9648	0.055213
y_receive_backsub_in	9648	0.055167
z_receive_backsub_in	9648	0.054876
x_unpack_solve_info_	9648	0.050514
y_unpack_solve_info_	9648	0.049544
z_unpack_solve_info_	9648	0.045698

Seite 1 von 15

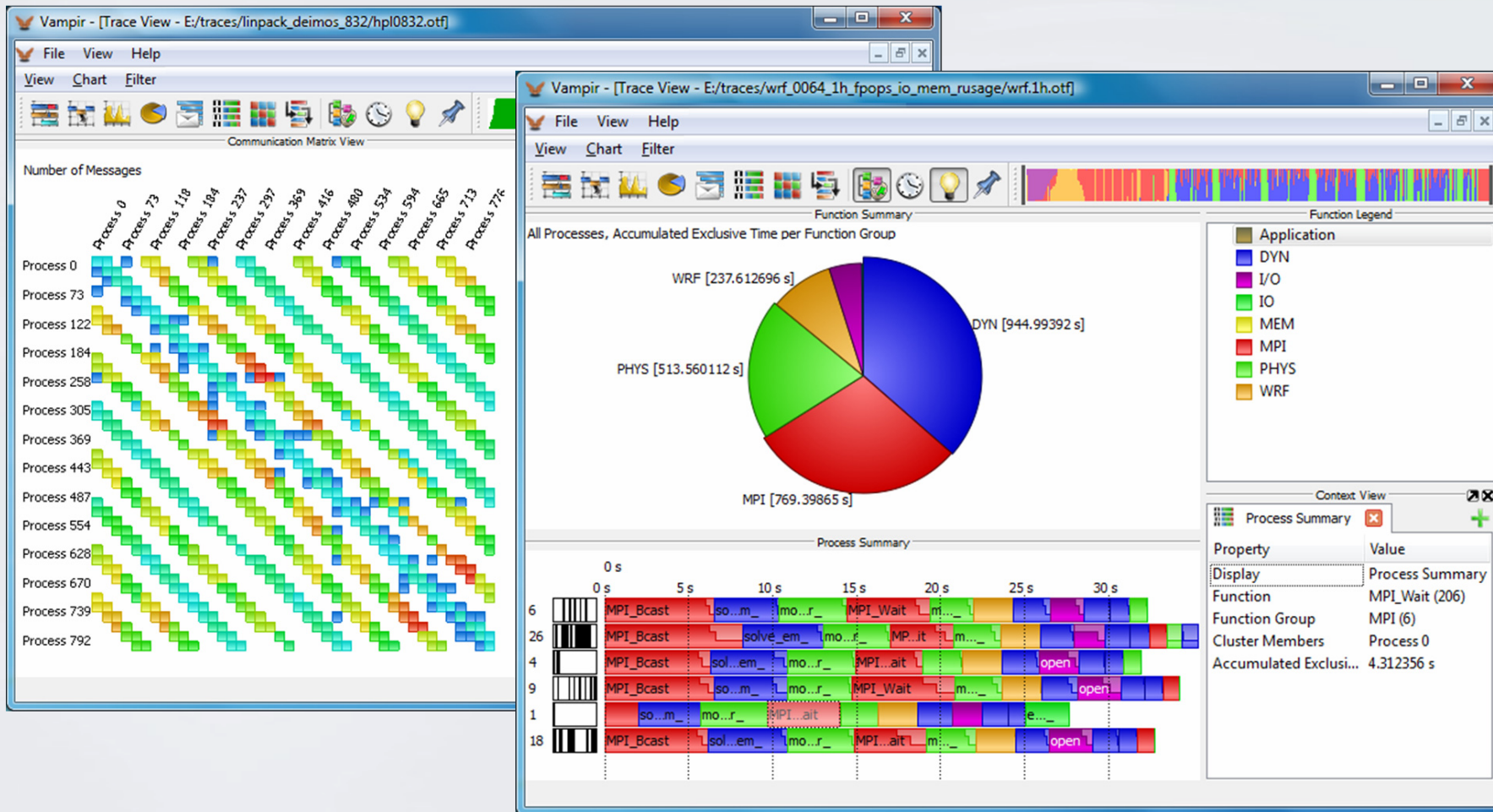


Performance Analysis with Vampir 7

Performance Analysis with Vampir 7 - Timelines



Performance Analysis with Vampir 7 - Summaries

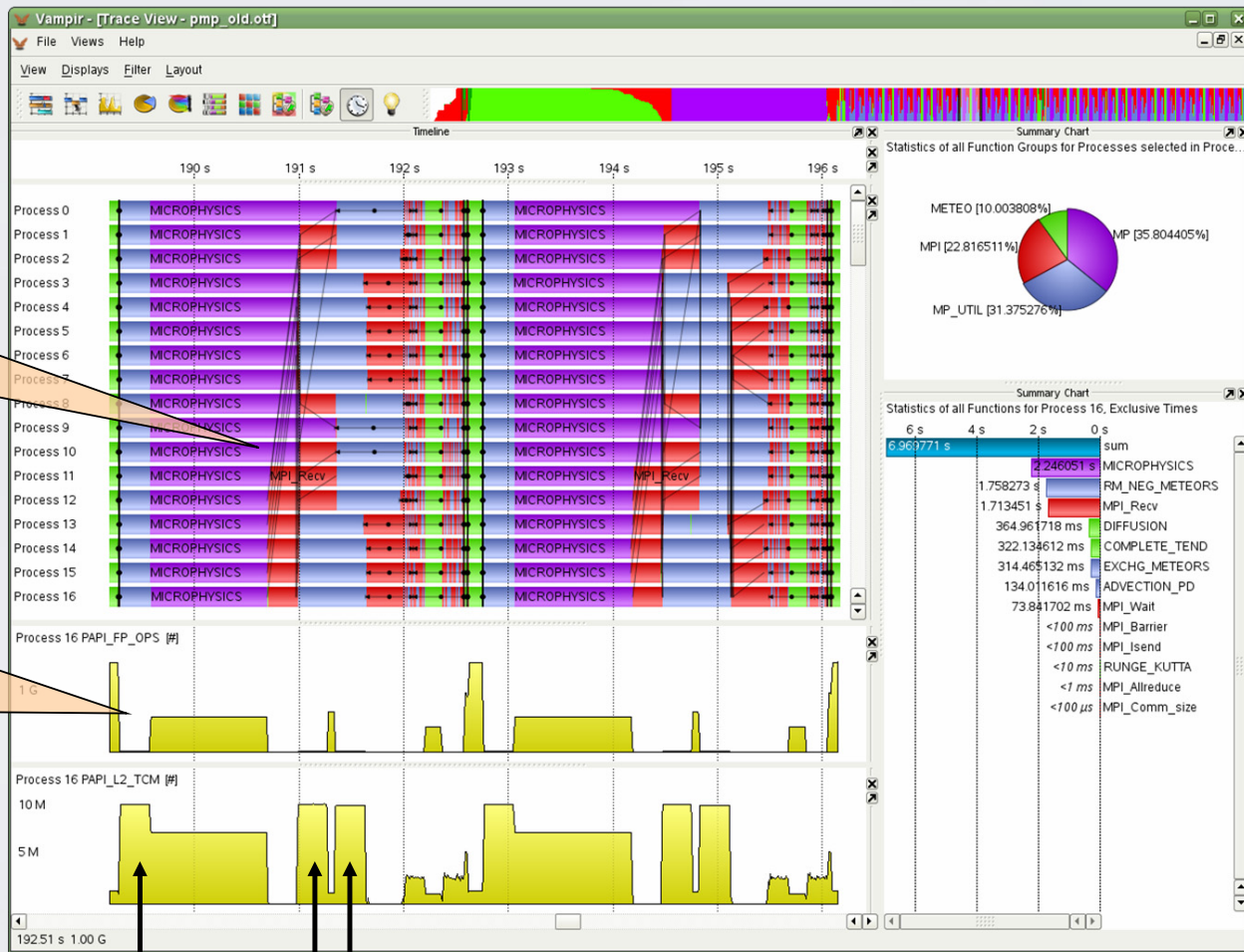


Performance Analysis of COSMO-SPECS with Vampir 7.0

Looking at two time steps of the original COSMO-SPECS (100 cores)

Load imbalance in most expensive routine (SPECS)

Routine without computations, but many cache misses ...



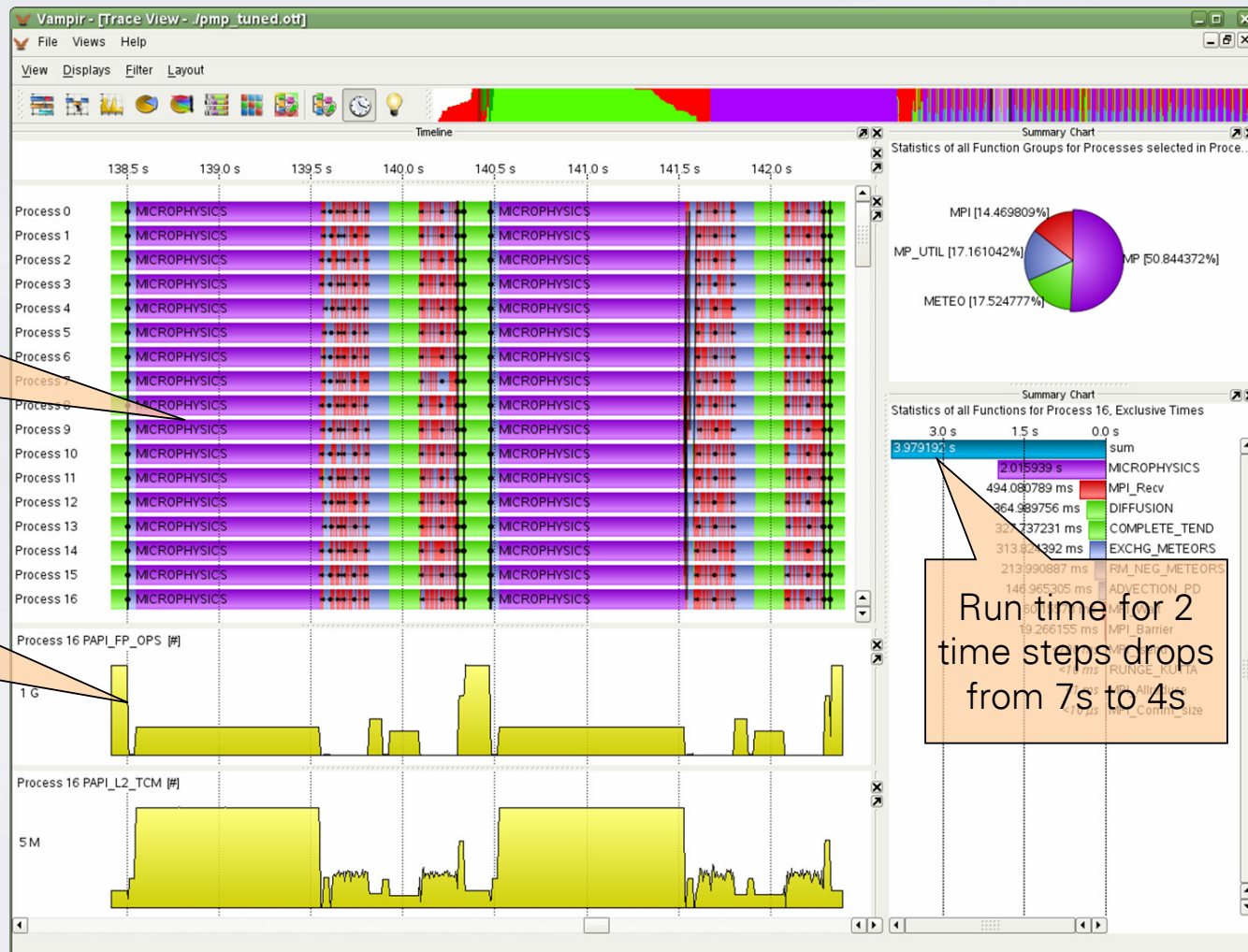
... is called 3 times per time step

Performance Analysis of COSMO-SPECS with Vampir 7.0

After tuning with the findings from the Vampir analysis

Perfect balance, some serial optimizations in SPECS

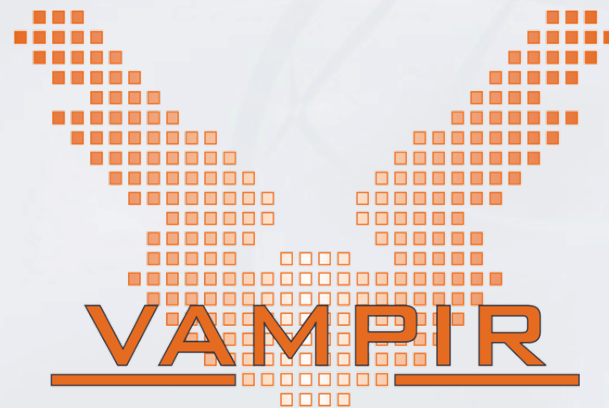
Routine much faster after cache tuning



Run time for 2 time steps drops from 7s to 4s

Thank You

Interested in Vampir or in a Vampir live demo?



Visit us at booth 2085 or www.vampir.eu