Virtual Institute – High Productivity Supercomputing

9-10 August 2012

Brian Wylie
Jülich Supercomputing Centre
b.wylie@fz-juelich.de

Outline

Thursday 9 August

- 09:00 (start)
- Performance analysis and tools overview (Geimer, JSC)
- Accessing & using CSCS resources
- 10:30-10:45 (break)
- Profile & automatic trace analysis with Scalasca (Wylie/Geimer, JSC)
- 12:30-13:30 (lunch)
- Trace analysis with Vampir (Knüpfer, TU Dresden)
- 15:00-15:15 (break)
- Assisted analysis & tuning of participants' codes
- 17:30 (adjourn)

Participant survey

We'd like to know a little about you, your application(s), and your expectations and desires from this tutorial

- What programming paradigms do you use in your app(s)?
  - only MPI, only OpenMP, mixed-mode/hybrid OpenMP/MPI, ...
  - Fortran, C, C++, multi-language, ...
- What platforms/systems *must* your app(s) run well on?
  - Cray XT/XE/XK, IBM BlueGene, SGI Altix, Linux cluster™, ...
- Who's already familiar with *serial* performance analysis?
  - Which tools have you used?
    - time, print/printf, prof/gprof, ...
- Who's already familiar with *parallel* performance analysis?
  - Which tools have you used?
    - time, print/printf, prof/gprof, Periscope, Scalasca, TAU, Vampir, ...

Prepare to analyse your own application codes

- Ensure your application codes build and run to completion with appropriate datasets
  - initial configuration should ideally run in less than 15 minutes with 1-4 compute nodes (up to 64 processes/threads)
    - to facilitate rapid turnaround and quick experimentation
  - larger/longer scalability configurations are also interesting
    - turnaround may be limited due to busyness of batch queues
- Compare your application performance on other systems
  - VI-HPS tools already installed on a number of HPC systems
    - if not, ask your system administrator to install them
      (or install a personal copy yourself)

Goal: Improve the quality and accelerate the development process of complex simulation codes running on highly-parallel computer systems

- Start-up funding (2006-2011)
  by Helmholtz Association
  of German Research Centres

HELMHOLTZ
ASSOCIATION

- Activities
  - Development and integration of HPC programming tools
    - Correctness checking & performance analysis
  - Training workshops
  - Service
    - Support email lists
    - Application engagement
  - Academic workshops

www.vi-hps.org

VI-HPS partners (founders)

Forschungszentrum Jülich

- Jülich Supercomputing Centre

RWTH Aachen University

- Centre for Computing & Communication

Technical University of Dresden

- Centre for Information Services & HPC

University of Tennessee (Knoxville)

- Innovative Computing Laboratory

VI-HPS partners (cont.)

Barcelona Supercomputing Center

- Centro Nacional de Supercomputación

German Research School

- Laboratory of Parallel Programming

Lawrence Livermore National Lab.

- Centre for Applied Scientific Computing

Technical University of Munich

- Chair for Computer Architecture

University of Oregon

- Performance Research Laboratory

University of Stuttgart

- HPC Centre

University of Versailles St-Quentin

- LRC ITACA

VI-HPS productivity tools

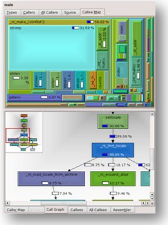
- Marmot/MUST
  - MPI correctness checking
- PAPI
  - Interfacing to hardware performance counters
- Periscope
  - Automatic analysis driven by on-line distributed search
- Scalasca
  - Large-scale parallel performance analysis
- TAU
  - Integrated parallel performance system
- Vampir/VampirTrace
  - Event tracing and graphical trace visualization & analysis
- Score-P
  - Common instrumentation & measurement infrastructure

VI-HPS productivity tools (cont.)

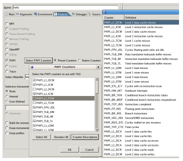
- KCachegrind
  - Callgraph-based cache analysis [x86 only]
- MAQAO
  - Assembly instrumentation & optimization [x86 only]
- mpiP/mpiPview
  - MPI profiling tool and analysis viewer
- ompP
  - OpenMP profiling tool
- OpenMPI
  - Memory checking
- Open|SpeedShop
  - Integrated parallel performance analysis environment
- Paraver/Extrae
  - Event tracing and graphical trace visualization & analysis

Technologies and their integration

KCACHEGRIND

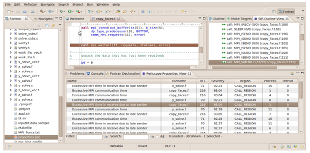


PAPI

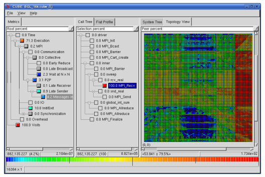


Hardware monitoring

PERISCOPE

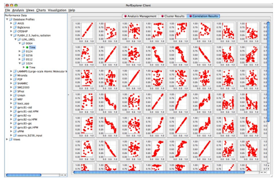


SCALASCA

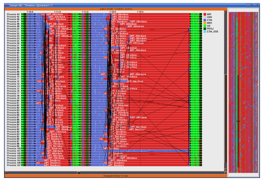


Automatic profile & trace analysis

TAU



VAMPIR / PARAVER



Visual trace analysis

Optimization

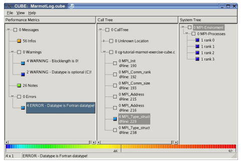
MAQAO

Execution

SIONLIB / OPENMPI

Error detection

MARMOT / MUST



Disclaimer

Tools will ***not*** automatically make you, your applications or computer systems more *productive*.

However, they can help you understand ***how*** your parallel code executes and ***when / where*** it's necessary to work on *correctness* and *performance* issues.

VI-HPS Training & Tuning Workshops

- Goals
  - Give an overview of the programming tools suite
  - Explain the functionality of individual tools
  - Teach how to use the tools effectively
  - Offer hands-on experience and expert assistance using tools
  - Receive feedback from users to guide future development
- For best results, bring & analyse/tune your own code(s)!

- VI-HPS Tutorial series
  - SC'08, ICCS'09, SC'09, Cluster'10, SC'10, SC'11, **EuroMPI'12**
- VI-HPS Tuning Workshop series
  - 2008 (Aachen & Dresden), 2009 (Jülich & Bremen),
    2010 (Garching & Amsterdam), 2011 (Stuttgart & Aachen)
  - 2012/04/23-27 (St-Quentin), **2012/10/15-19 (Garching)**

Upcoming VI-HPS training events

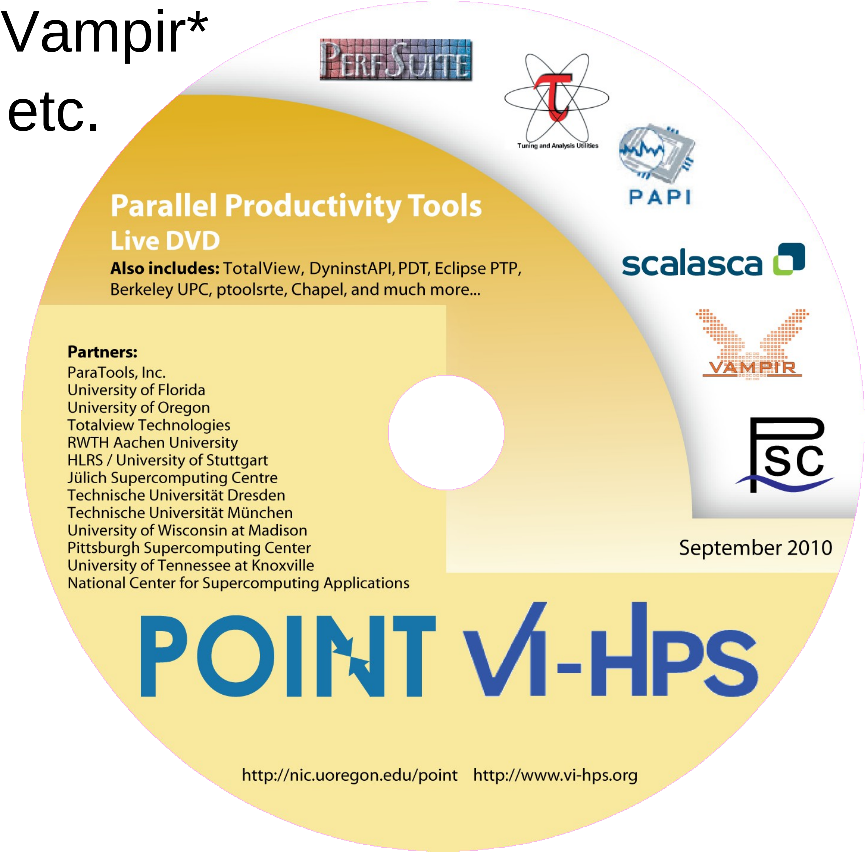
- EuroMPI hands-on tutorial (23 Sep 2012, Vienna, Austria)
  - Periscope, Scalasca, TAU, Vampir (using Score-P)
- 10th VI-HPS Tuning Workshop (15-19 Oct 2012)
  - hosted by LRZ, Garching-bei-München, Germany
  - using PRACE Tier-0 *SuperMUC* iDataPlex system
  - Scalasca, Vampir, TAU, Periscope, KCachegrind, MAQAO, ...
- Further events to be determined
  - (one-day) tutorials
    - with guided exercises usually using Live DVD
  - (multi-day) training workshops
    - with your own applications on real HPC systems

Check www.vi-hps.org/training for announced events

- Contact us if you might be interested in hosting an event

POINT/VI-HPS Live-ISO/DVD

- Bootable Linux installation ISO (on DVD or USB drive)
- Includes everything needed to try out our parallel tools on an x86-architecture notebook computer
  - VI-HPS tools: KCachegrind, Marmot, PAPI, Periscope, Scalasca, TAU, VT/Vampir*
  - Also: Eclipse/PTP, TotalView*, etc.
    - * time/capability-limited evaluation licences provided for commercial products
  - GCC (w/ OpenMP), OpenMPI
  - Manuals/User Guides
  - Tutorial exercises & examples
- Produced by U. Oregon PRL
  - Sameer Shende



Parallel Productivity Tools
Live DVD

Also includes: TotalView, DyninstAPI, PDT, Eclipse PTP, Berkeley UPC, ptoolsrte, Chapel, and much more...

Partners:
ParaTools, Inc.
University of Florida
University of Oregon
Totalview Technologies
RWTH Aachen University
HLRS / University of Stuttgart
Jülich Supercomputing Centre
Technische Universität Dresden
Technische Universität München
University of Wisconsin at Madison
Pittsburgh Supercomputing Center
University of Tennessee at Knoxville
National Center for Supercomputing Applications

September 2010

http://nic.uoregon.edu/point http://www.vi-hps.org

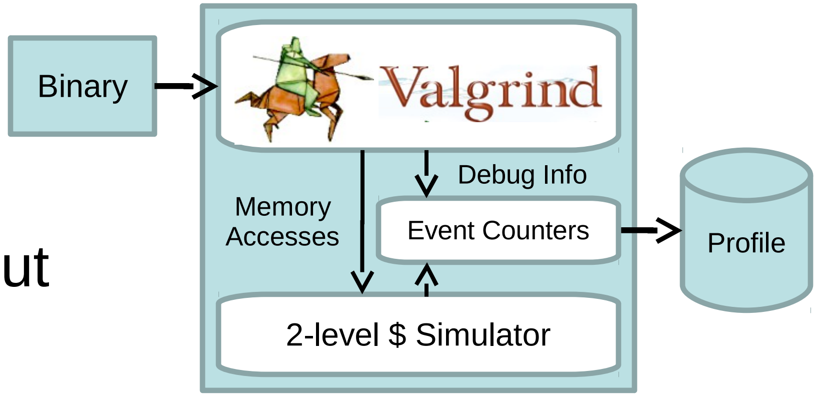
POINT/VI-HPS Live-ISO/DVD

- ISO image approximately 4GB
  - distributed on DVD or USB drive
  - or download from website
- Boot directly from disk
  - enables hardware counter access and offers best performance
- Boot within virtual machine
  - faster boot time and can save/resume state, but no hardware counter access
- Boots into Linux environment
  - supports building and running provided MPI and/or OpenMP parallel application codes
  - and experimentation with VI-HPS (and other) tools

KCachegrind

Cachegrind: cache analysis by simple cache simulation

- Captures dynamic callgraph
- Based on valgrind dynamic binary instrumentation
- Runs on x86/PowerPC/ARM unmodified binaries
  - No root access required
- ASCII reports produced



Binary

Valgrind

Debug Info

Memory Accesses

Event Counters

Profile

2-level $ Simulator

[KQ]Cachegrind GUI

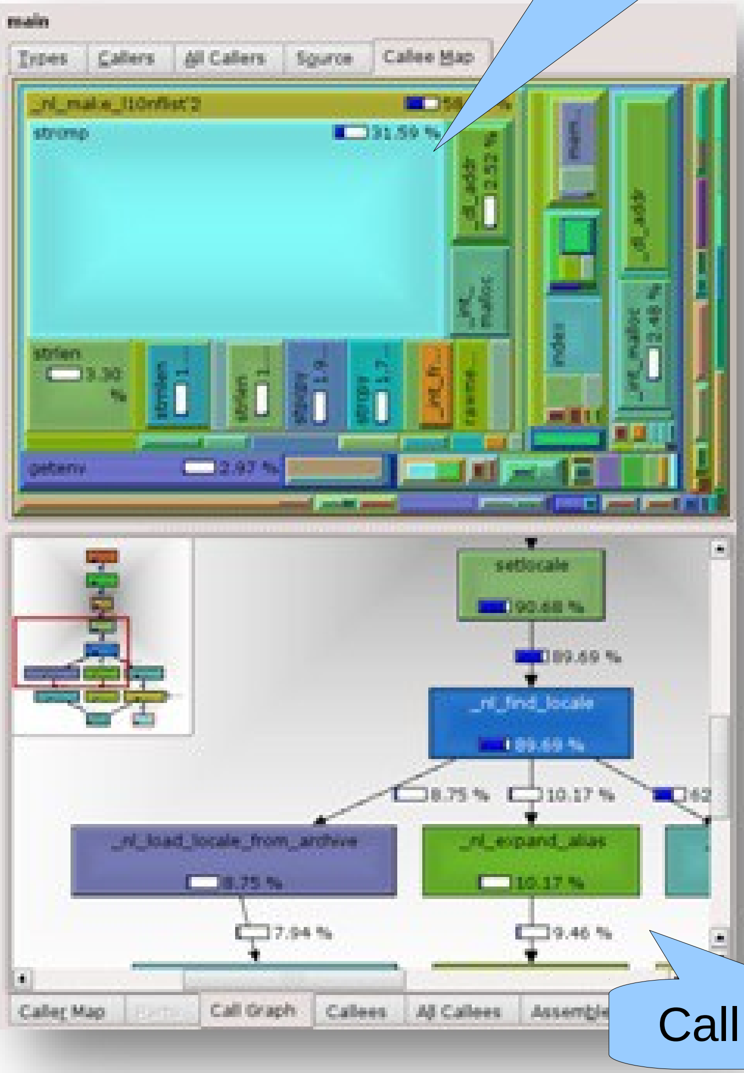
- Visualization of cachegrind output

Developed by TU Munich

- Released as GPL open-source
- http://kcachegrind.sf.net/

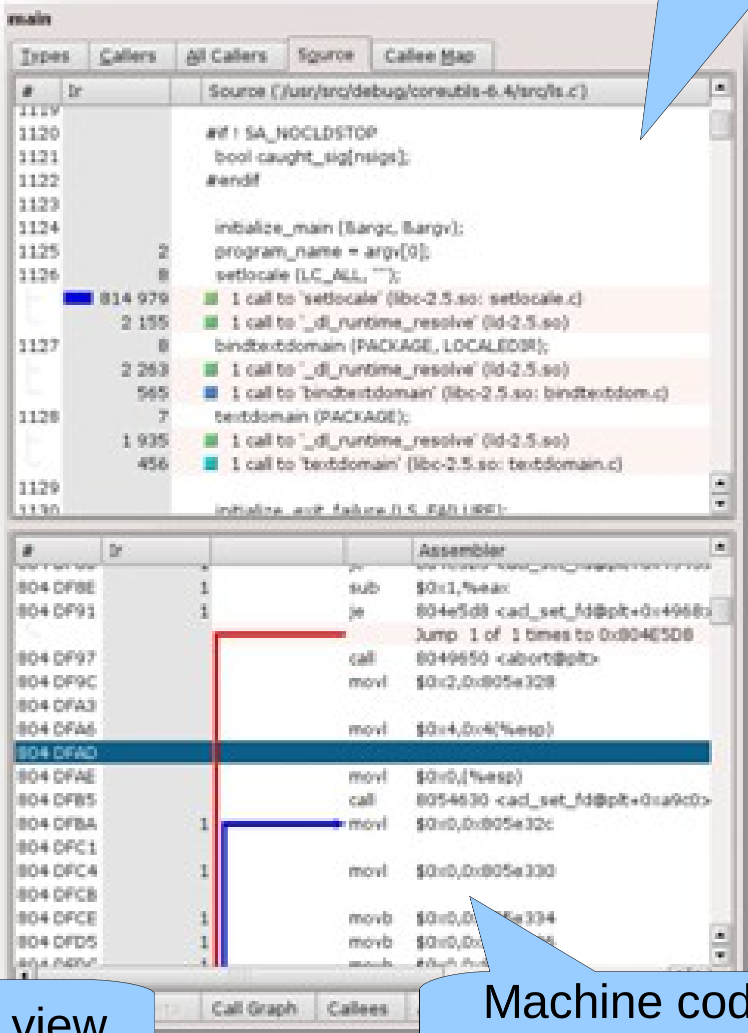
KCachegrind GUI

Event cost tree map



Call graph view

Source code view



```
#     Ir          Source ('/usr/src/debug/coreutils-6.4/src/ls.c')
1120              #if ! SA_NOCLDSTOP
1121                bool caught_sig[nsigs];
1122              #endif
1123
1124                initialize_main (&argc, &argv);
1125            2   program_name = argv[0];
1126            8   setlocale (LC_ALL, "");
          814 979   1 call to 'setlocale' (libc-2.5.so: setlocale.c)
            2 155   1 call to '_dl_runtime_resolve' (ld-2.5.so)
1127            8   bindtextdomain (PACKAGE, LOCALEDIR);
            2 263   1 call to '_dl_runtime_resolve' (ld-2.5.so)
              565   1 call to 'bindtextdomain' (libc-2.5.so: bindtextdom.c)
1128            7   textdomain (PACKAGE);
            1 935   1 call to '_dl_runtime_resolve' (ld-2.5.so)
              456   1 call to 'textdomain' (libc-2.5.so: textdomain.c)
1129
1130                initialize_exit_failure (LS_FAILURE);
```

```
#          Ir          Assembler
804 DF8E    1   sub    $0x1,%eax
804 DF91    1   je     804e5d8 <acl_set_fd@plt+0x4968>
                       Jump 1 of 1 times to 0x804E5D8
804 DF97        call   8049650 <abort@plt>
804 DF9C        movl   $0x2,0x805e328
804 DFA3
804 DFA6        movl   $0x4,0x4(%esp)
804 DFAD
804 DFAE        movl   $0x0,(%esp)
804 DFB5        call   8054630 <acl_set_fd@plt+0xa9c0>
804 DFBA    1   movl   $0x0,0x805e32c
804 DFC1
804 DFC4    1   movl   $0x0,0x805e330
804 DFCB
804 DFCE    1   movb   $0x0,0x805e334
804 DFD5    1   movb   $0x0,0x
```

Machine code annotation

Marmot

Tool to check for correct MPI usage at runtime

- Checks conformance to MPI standard
  - Supports Fortran & C bindings of MPI-1.2
- Checks parameters passed to MPI
- Monitors MPI resource usage

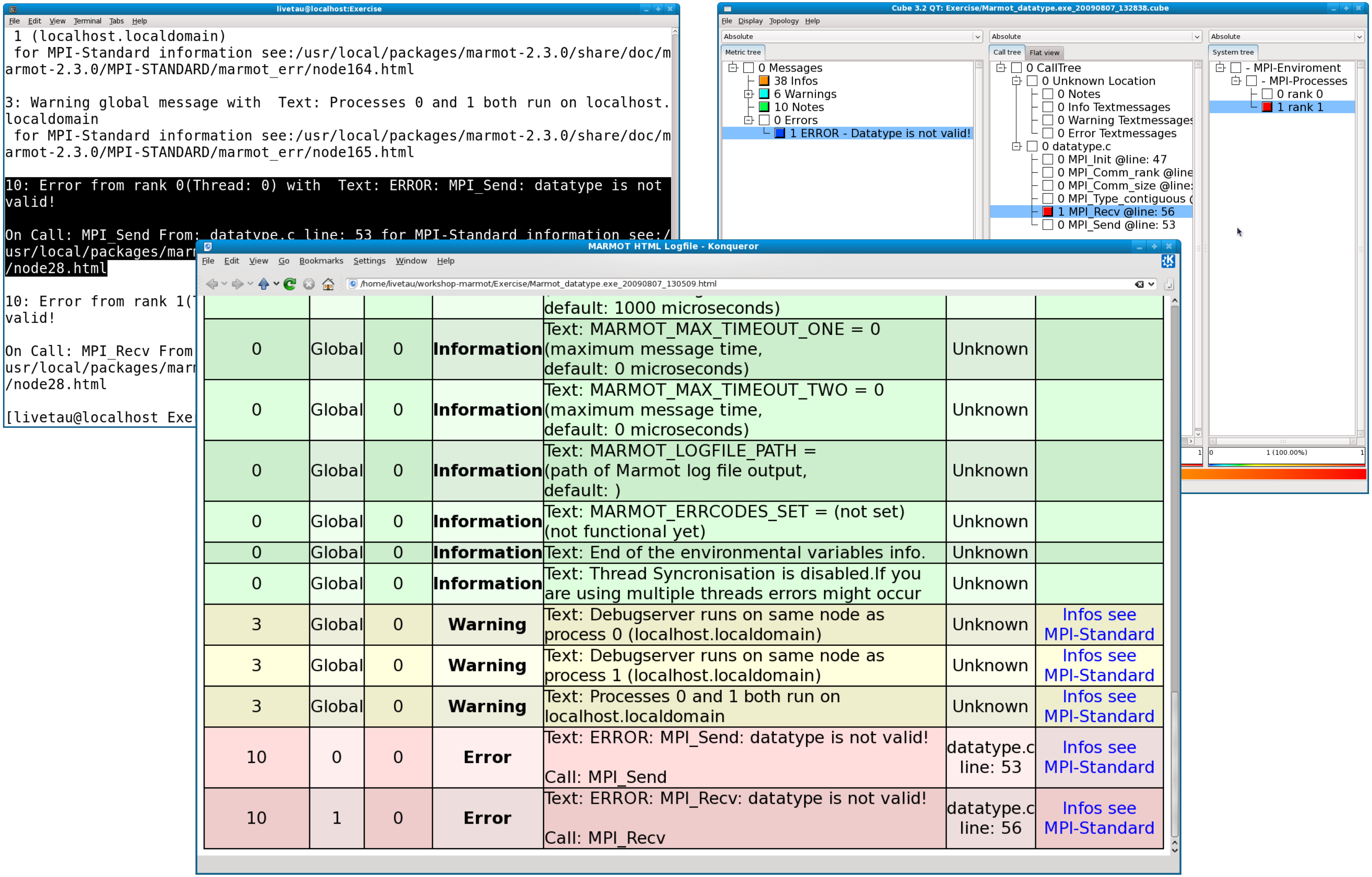
Implementation

- C++ library gets linked to the application
- Does not require source code modifications
- Additional process used as DebugServer
- Results written in a log file (ASCII/HTML/CUBE)

Developed by HLRS & TU Dresden

- Released as open-source
- http://www.hlrs.de/organization/av/amt/projects/marmot

Marmot logfiles


livetau@localhost:Exercise
File Edit View Terminal Tabs Help
Cube 3.2 QT: Exercise/Marmot_datatype.exe_20090807_132838.cube
File Display Topology Help
Absolute
Absolute
Absolute
Metric tree
Call tree
Flat view
System tree
0
1 (100.00%)
1
MARMOT HTML Logfile - Konqueror
File Edit View Go Bookmarks Settings Window Help
/home/livetau/workshop-marmot/Exercise/Marmot_datatype.exe_20090807_130509.html

```
 1 (localhost.localdomain)
 for MPI-Standard information see:/usr/local/packages/marmot-2.3.0/share/doc/m
armot-2.3.0/MPI-STANDARD/marmot_err/node164.html

3: Warning global message with  Text: Processes 0 and 1 both run on localhost.
localdomain
 for MPI-Standard information see:/usr/local/packages/marmot-2.3.0/share/doc/m
armot-2.3.0/MPI-STANDARD/marmot_err/node165.html

10: Error from rank 0(Thread: 0) with  Text: ERROR: MPI_Send: datatype is not
valid!

On Call: MPI_Send From: datatype.c line: 53 for MPI-Standard information see:/
usr/local/packages/marm
/node28.html

10: Error from rank 1(
valid!

On Call: MPI_Recv From
usr/local/packages/marm
/node28.html

[livetau@localhost Exe
```

```
□ 0 Messages
    🟧 38 Infos
    🟦 6 Warnings
    🟩 10 Notes
    □ 0 Errors
        🟦 1 ERROR - Datatype is not valid!
```

```
□ 0 CallTree
    □ 0 Unknown Location
        □ 0 Notes
        □ 0 Info Textmessages
        □ 0 Warning Textmessages
        □ 0 Error Textmessages
    □ 0 datatype.c
        □ 0 MPI_Init @line: 47
        □ 0 MPI_Comm_rank @line
        □ 0 MPI_Comm_size @line:
        □ 0 MPI_Type_contiguous (
        🟥 1 MPI_Recv @line: 56
        □ 0 MPI_Send @line: 53
```

```
□ - MPI-Enviroment
    □ - MPI-Processes
        □ 0 rank 0
        🟥 1 rank 1
```

				default: 1000 microseconds)		
0	Global	0	**Information**	Text: MARMOT_MAX_TIMEOUT_ONE = 0 (maximum message time, default: 0 microseconds)	Unknown	
0	Global	0	**Information**	Text: MARMOT_MAX_TIMEOUT_TWO = 0 (maximum message time, default: 0 microseconds)	Unknown	
0	Global	0	**Information**	Text: MARMOT_LOGFILE_PATH = (path of Marmot log file output, default:)	Unknown	
0	Global	0	**Information**	Text: MARMOT_ERRCODES_SET = (not set) (not functional yet)	Unknown	
0	Global	0	**Information**	Text: End of the environmental variables info.	Unknown	
0	Global	0	**Information**	Text: Thread Syncronisation is disabled.If you are using multiple threads errors might occur	Unknown	
3	Global	0	**Warning**	Text: Debugserver runs on same node as process 0 (localhost.localdomain)	Unknown	Infos see MPI-Standard
3	Global	0	**Warning**	Text: Debugserver runs on same node as process 1 (localhost.localdomain)	Unknown	Infos see MPI-Standard
3	Global	0	**Warning**	Text: Processes 0 and 1 both run on localhost.localdomain	Unknown	Infos see MPI-Standard
10	0	0	**Error**	Text: ERROR: MPI_Send: datatype is not valid! Call: MPI_Send	datatype.c line: 53	Infos see MPI-Standard
10	1	0	**Error**	Text: ERROR: MPI_Recv: datatype is not valid! Call: MPI_Recv	datatype.c line: 56	Infos see MPI-Standard

MUST

Next generation MPI runtime error detection tool

- Successor of the Marmot and Umpire tools
- Initial merge of Marmot's many local checks with Umpire's non-local checks
- Improved scalability expected in future
- Exploits CMake, GTI & PnMPI infrastructure

Developed by TU Dresden, LLNL & LANL

- BSD license open-source initial release in November 2011
- http://tu-dresden.de/zih/must/

PAPI

- Portable performance counter library & utilities
  - Configures and accesses hardware/system counters
  - Predefined events derived from available native counters
  - Core component for CPU/processor counters
    - instructions, floating point operations, branches predicted/taken, cache accesses/misses, TLB misses, cycles, stall cycles, …
    - performs transparent multiplexing when required
  - Extensible components for off-processor counters
    - InfiniBand network, Lustre filesystem, system hardware health, …
  - Used by multi-platform performance measurement tools
    - Periscope, Scalasca, TAU, VampirTrace, ...
- Developed by UTK-ICL
  - Available as open-source for most modern processors
    http://icl.cs.utk.edu/papi/

PAPI preset counters (and their definitions)

```
juropa$ papi_avail
Available events and hardware information.
-----------------------------------------------
PAPI Version              : 4.1.0.0
Vendor string and code    : GenuineIntel (1)
Model string and code     : Intel(R) Xeon(R) CPU
                            X5570 @ 2.93GHz (26)
CPU Revision              : 5.000000
CPUID Info                : Family: 6  Model: 26
                            Stepping: 5
CPU Megahertz             : 1600.000000
CPU Clock Megahertz       : 1600
Hdw Threads per core      : 2
Cores per Socket          : 4
NUMA Nodes                : 2
CPU's per Node            : 8
Total CPU's               : 16
Number Hardware Counters  : 16
Max Multiplex Counters    : 512
-----------------------------------------------
    Name         Code     Avail Deriv Description
PAPI_L1_DCM  0x80000000  Yes   No
                      Level 1 data cache misses
PAPI_L1_ICM  0x80000001  Yes   No
               Level 1 instruction cache misses
...
-----------------------------------------------
Of 107 possible events, 35 are available, of
which 9 are derived.
```

```
juropa$ papi_avail -d
...
Symbol       Event Code  Count   |Short Descr.|
 |Long Description|
 |Developer's Notes|
 |Derived|
 |PostFix|
 Native Code[n]: <hex> |name|
PAPI_L1_DCM    0x80000000  1 |L1D cache misses|
 |Level 1 data cache misses|
 ||
 |NOT_DERIVED|
 ||
 Native Code[0]: 0x40002028 |L1D:REPL|
PAPI_L1_ICM    0x80000001  1 |L1I cache misses|
 |Level 1 instruction cache misses|
 ||
 |NOT_DERIVED|
 ||
 Native Code[0]: 0x40001031 |L1I:MISSES|
PAPI_L2_DCM    0x80000002  2 |L2D cache misses|
 |Level 2 data cache misses|
 ||
 |DERIVED_SUB|
 ||
 Native Code[0]: 0x40000437 |L2_RQSTS:MISS|
 Native Code[1]: 0x40002037 |
L2_RQSTS:IFETCH_MISS|
...
```

PAPI native counters (and qualifiers)

```
juropa$ papi_native_avail
Available native events and hardware information.
...
Event Code   Symbol  | Long Description |
--------------------------------------------------------------------------------
0x40000000   UNHALTED_CORE_CYCLES  | count core clock cycles whenever the cloc |
           | k signal on the specific core is running (not halted). Alias to e |
           | vent CPU_CLK_UNHALTED:THREAD                                      |
--------------------------------------------------------------------------------
0x40000001   INSTRUCTION_RETIRED  | count the number of instructions at retire |
           | ment. Alias to event INST_RETIRED:ANY_P                           |
--------------------------------------------------------------------------------
...
--------------------------------------------------------------------------------
0x40000086   UNC_SNP_RESP_TO_REMOTE_HOME  | Remote home snoop response - LLC d |
           | oes not have cache line                                           |
  40000486   :I_STATE  | Remote home snoop response - LLC does not have cache  |
           | line                                                              |
  40000886   :S_STATE  | Remote home snoop response - LLC has  cache line in S |
           |  state                                                            |
  40001086   :FWD_S_STATE  | Remote home snoop response - LLC forwarding cache |
           |  line in S state.                                                 |
  40002086   :FWD_I_STATE  | Remote home snoop response - LLC has forwarded a  |
           | modified cache line                                               |
  40004086   :CONFLICT  | Remote home conflict snoop response                  |
  40008086   :WB  | Remote home snoop response - LLC has cache line in the M s |
           | tate                                                              |
  40010086   :HITM  | Remote home snoop response - LLC HITM                    |
--------------------------------------------------------------------------------
Total events reported: 135
```

Periscope

Automated profile-based performance analysis

- Iterative on-line performance analysis
  - Multiple distributed hierarchical agents
- Automatic search for bottlenecks based on properties formalizing expert knowledge
  - MPI wait states, OpenMP overheads and imbalances
  - Processor utilization hardware counters
- Clustering of processes/threads with similar properties
- Eclipse-based integrated environment

Supports

- SGI Altix Itanium2, IBM Power and x86-based architectures

Developed by TU Munich

- Released as open-source
- http://www.lrr.in.tum.de/periscope

Periscope properties & strategies (examples)

MPI

- Excessive MPI communication time
- Excessive MPI time due to many small messages
- Excessive MPI time in receive due to late sender
- …

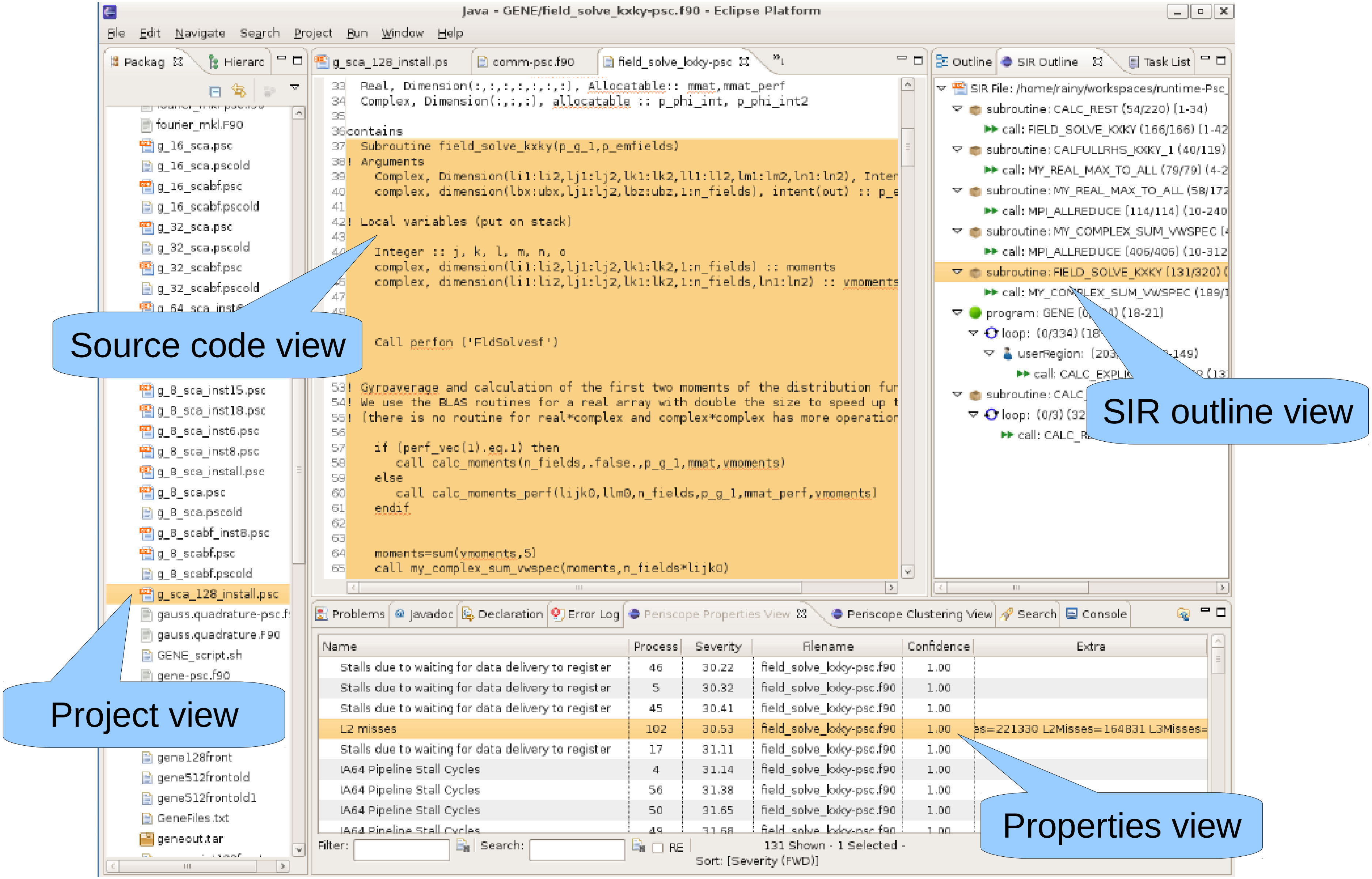
OpenMP

- Load imbalance in parallel region/section
- Sequential computation in master/single/ordered region
- ...

Hardware performance counters (platform-specific)

- Cycles lost due to cache misses
  - High L1/L2/L3 demand load miss rate
- Cycles lost due to no instruction to dispatch
- ...

Periscope plug-in to Eclipse environment



```
  Real, Dimension(:,:,:,:,:,:,:), Allocatable:: mmat,mmat_perf
  Complex, Dimension(:,:,:), allocatable :: p_phi_int, p_phi_int2

contains
  Subroutine field_solve_kxky(p_g_1,p_emfields)
! Arguments
    Complex, Dimension(li1:li2,lj1:lj2,lk1:lk2,ll1:ll2,lm1:lm2,ln1:ln2), Inter
    complex, dimension(lbx:ubx,lj1:lj2,lbz:ubz,1:n_fields), intent(out) :: p_e

! Local variables (put on stack)

    Integer :: j, k, l, m, n, o
    complex, dimension(li1:li2,lj1:lj2,lk1:lk2,1:n_fields) :: moments
    complex, dimension(li1:li2,lj1:lj2,lk1:lk2,1:n_fields,ln1:ln2) :: vmoments



    Call perfon ('FldSolvesf')


! Gyroaverage and calculation of the first two moments of the distribution fur
! We use the BLAS routines for a real array with double the size to speed up t
! (there is no routine for real*complex and complex*complex has more operatior

    if (perf_vec(1).eq.1) then
       call calc_moments(n_fields,.false.,p_g_1,mmat,vmoments)
    else
       call calc_moments_perf(lijk0,llm0,n_fields,p_g_1,mmat_perf,vmoments)
    endif


    moments=sum(vmoments,5)
    call my_complex_sum_vwspec(moments,n_fields*lijk0)
```

```
SIR File: /home/rainy/workspaces/runtime-Psc_
  subroutine: CALC_REST (54/220) (1-34)
      call: FIELD_SOLVE_KXKY (166/166) (1-42
  subroutine: CALFULLRHS_KXKY_1 (40/119)
      call: MY_REAL_MAX_TO_ALL (79/79) (4-2
  subroutine: MY_REAL_MAX_TO_ALL (58/172
      call: MPI_ALLREDUCE (114/114) (10-240
  subroutine: MY_COMPLEX_SUM_VWSPEC (4
      call: MPI_ALLREDUCE (406/406) (10-312
  subroutine: FIELD_SOLVE_KXKY (131/320) (
      call: MY_COMPLEX_SUM_VWSPEC (189/1
  program: GENE (0/     ) (18-21)
    loop: (0/334) (18
      userRegion: (203/         -149)
          call: CALC_EXPLI           (13
  subroutine: CALC_
    loop: (0/3) (32
        call: CALC_R
```

Name	Process	Severity	Filename	Confidence	Extra
Stalls due to waiting for data delivery to register	46	30.22	field_solve_kxky-psc.f90	1.00	
Stalls due to waiting for data delivery to register	5	30.32	field_solve_kxky-psc.f90	1.00	
Stalls due to waiting for data delivery to register	45	30.41	field_solve_kxky-psc.f90	1.00	
L2 misses	102	30.53	field_solve_kxky-psc.f90	1.00	es=221330 L2Misses=164831 L3Misses=
Stalls due to waiting for data delivery to register	17	31.11	field_solve_kxky-psc.f90	1.00	
IA64 Pipeline Stall Cycles	4	31.14	field_solve_kxky-psc.f90	1.00	
IA64 Pipeline Stall Cycles	56	31.38	field_solve_kxky-psc.f90	1.00	
IA64 Pipeline Stall Cycles	50	31.65	field_solve_kxky-psc.f90	1.00	
IA64 Pipeline Stall Cycles	49	31.68	field_solve_kxky-psc.f90	1.00	

Filter: Search: □ RE 131 Shown - 1 Selected - Sort: [Severity (FWD)]

Source code view

SIR outline view

Project view

Properties view

Scalasca

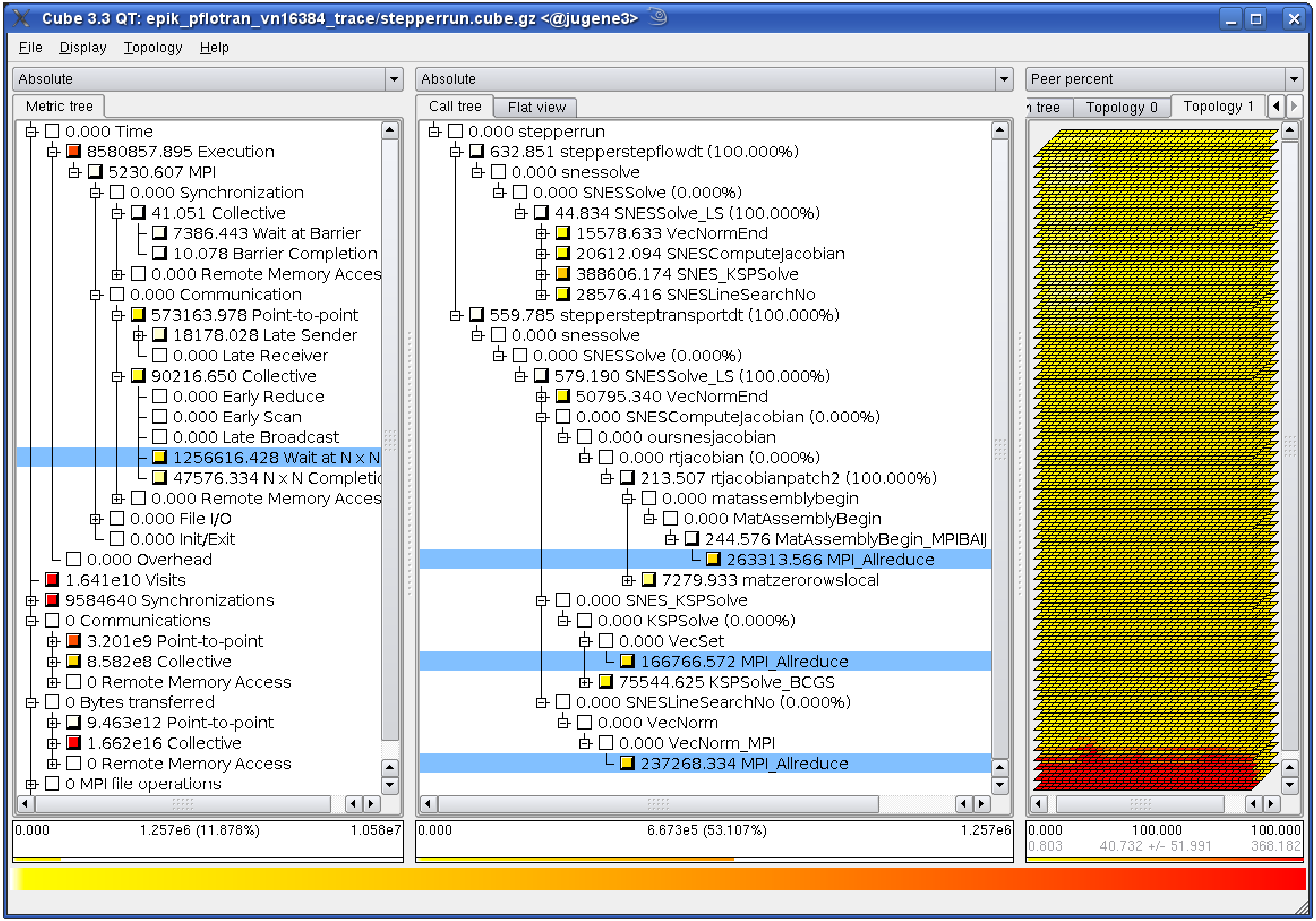
Automatic performance analysis toolset

- Scalable performance analysis of large-scale applications
  - particularly focused on MPI & OpenMP paradigms
  - analysis of communication & synchronization overheads
- Automatic and manual instrumentation capabilities
- Runtime summarization and/or event trace analyses
- Automatic search of event traces for patterns of inefficiency
  - Scalable trace analysis based on parallel replay
- Interactive exploration GUI and algebra utilities for XML callpath profile analysis reports

Developed by JSC & GRS

- Released as open-source
- http://www.scalasca.org/

Scalasca automatic trace analysis report



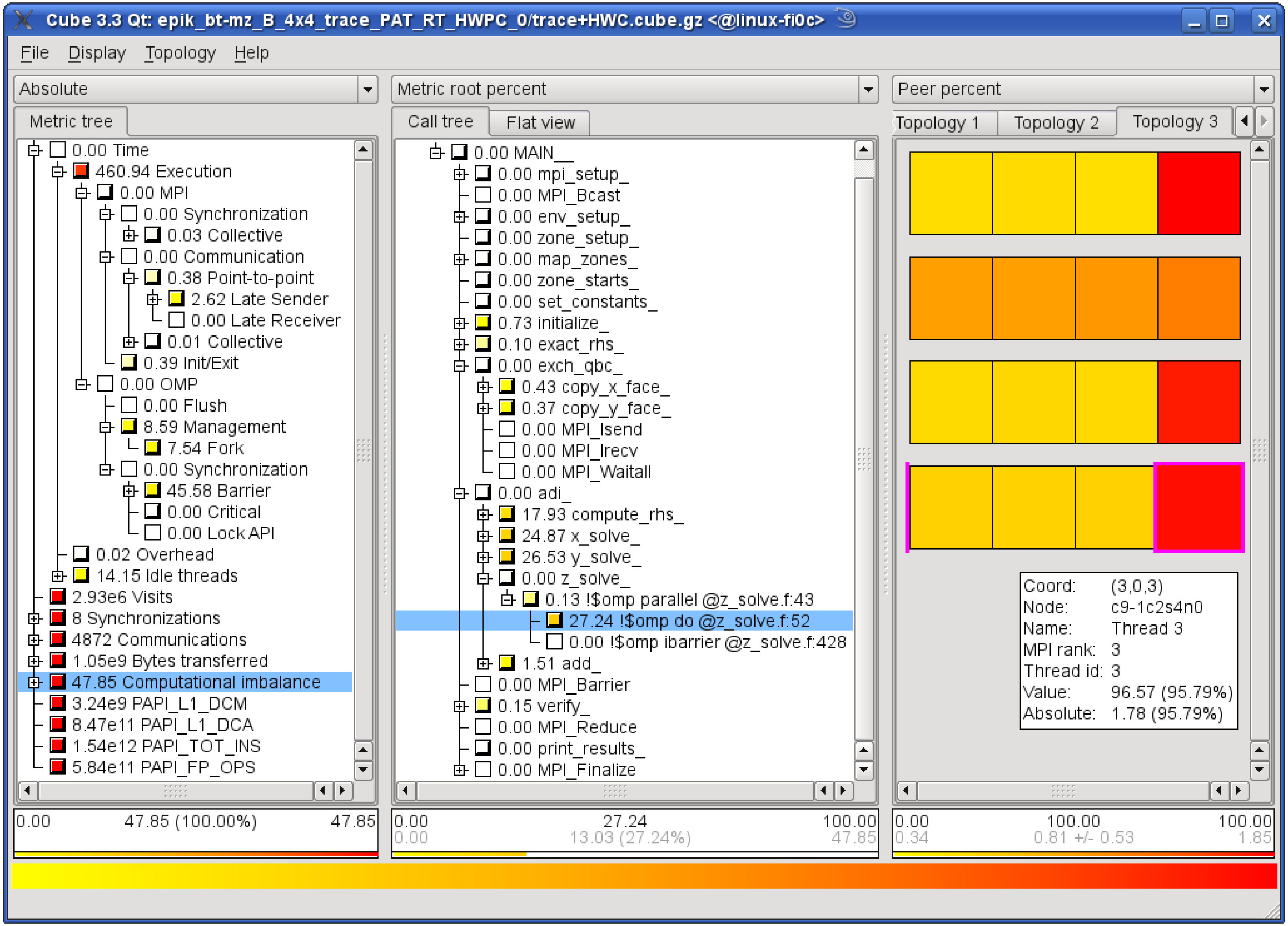
Cube 3.3 QT: epik_pflotran_vn16384_trace/stepperrun.cube.gz <@jugene3>

```
□ 0.000 Time
  🟧 8580857.895 Execution
    □ 5230.607 MPI
      □ 0.000 Synchronization
        □ 41.051 Collective
          □ 7386.443 Wait at Barrier
          □ 10.078 Barrier Completion
        □ 0.000 Remote Memory Acces
      □ 0.000 Communication
        🟨 573163.978 Point-to-point
          🟨 18178.028 Late Sender
          □ 0.000 Late Receiver
        🟨 90216.650 Collective
          □ 0.000 Early Reduce
          □ 0.000 Early Scan
          □ 0.000 Late Broadcast
          🟨 1256616.428 Wait at N x N
          🟨 47576.334 N x N Completi
        □ 0.000 Remote Memory Acces
      □ 0.000 File I/O
      □ 0.000 Init/Exit
  □ 0.000 Overhead
🟥 1.641e10 Visits
🟥 9584640 Synchronizations
□ 0 Communications
  🟧 3.201e9 Point-to-point
  🟨 8.582e8 Collective
  □ 0 Remote Memory Access
□ 0 Bytes transferred
  □ 9.463e12 Point-to-point
  🟥 1.662e16 Collective
  □ 0 Remote Memory Access
□ 0 MPI file operations
```

```
□ 0.000 stepperrun
  □ 632.851 stepperstepflowdt (100.000%)
    □ 0.000 snessolve
      □ 0.000 SNESSolve (0.000%)
        □ 44.834 SNESSolve_LS (100.000%)
          🟨 15578.633 VecNormEnd
          🟨 20612.094 SNESComputeJacobian
          🟨 388606.174 SNES_KSPSolve
          🟨 28576.416 SNESLineSearchNo
  □ 559.785 stepperstepstransportdt (100.000%)
    □ 0.000 snessolve
      □ 0.000 SNESSolve (0.000%)
        □ 579.190 SNESSolve_LS (100.000%)
          🟨 50795.340 VecNormEnd
          □ 0.000 SNESComputeJacobian (0.000%)
            □ 0.000 oursnesjacobian
              □ 0.000 rtjacobian (0.000%)
                □ 213.507 rtjacobianpatch2 (100.000%)
                  □ 0.000 matassemblybegin
                    □ 0.000 MatAssemblyBegin
                      □ 244.576 MatAssemblyBegin_MPIBAIJ
                        🟨 263313.566 MPI_Allreduce
                  🟨 7279.933 matzerorowslocal
          □ 0.000 SNES_KSPSolve
            □ 0.000 KSPSolve (0.000%)
              □ 0.000 VecSet
                🟨 166766.572 MPI_Allreduce
              🟨 75544.625 KSPSolve_BCGS
          □ 0.000 SNESLineSearchNo (0.000%)
            □ 0.000 VecNorm
              □ 0.000 VecNorm_MPI
                🟨 237268.334 MPI_Allreduce
```

```
0.000     1.257e6 (11.878%)     1.058e7
0.000     6.673e5 (53.107%)     1.257e6
0.000     100.000               100.000
0.803     40.732 +/- 51.991     368.182
```

Scalasca hybrid analysis report



Cube 3.3 Qt: epik_bt-mz_B_4x4_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz <@linux-fi0c>

Metric tree:

```
0.00 Time
  460.94 Execution
    0.00 MPI
      0.00 Synchronization
        0.03 Collective
      0.00 Communication
        0.38 Point-to-point
          2.62 Late Sender
          0.00 Late Receiver
        0.01 Collective
      0.39 Init/Exit
    0.00 OMP
      0.00 Flush
      8.59 Management
        7.54 Fork
      0.00 Synchronization
        45.58 Barrier
        0.00 Critical
        0.00 Lock API
  0.02 Overhead
  14.15 Idle threads
2.93e6 Visits
8 Synchronizations
4872 Communications
1.05e9 Bytes transferred
47.85 Computational imbalance
3.24e9 PAPI_L1_DCM
8.47e11 PAPI_L1_DCA
1.54e12 PAPI_TOT_INS
5.84e11 PAPI_FP_OPS
```

Call tree:

```
0.00 MAIN__
  0.00 mpi_setup_
  0.00 MPI_Bcast
  0.00 env_setup_
  0.00 zone_setup_
  0.00 map_zones_
  0.00 zone_starts_
  0.00 set_constants_
  0.73 initialize_
  0.10 exact_rhs_
  0.00 exch_qbc_
    0.43 copy_x_face_
    0.37 copy_y_face_
    0.00 MPI_Isend
    0.00 MPI_Irecv
    0.00 MPI_Waitall
  0.00 adi_
    17.93 compute_rhs_
    24.87 x_solve_
    26.53 y_solve_
    0.00 z_solve_
      0.13 !$omp parallel @z_solve.f:43
        27.24 !$omp do @z_solve.f:52
        0.00 !$omp ibarrier @z_solve.f:428
    1.51 add_
  0.00 MPI_Barrier
  0.15 verify_
  0.00 MPI_Reduce
  0.00 print_results_
  0.00 MPI_Finalize
```

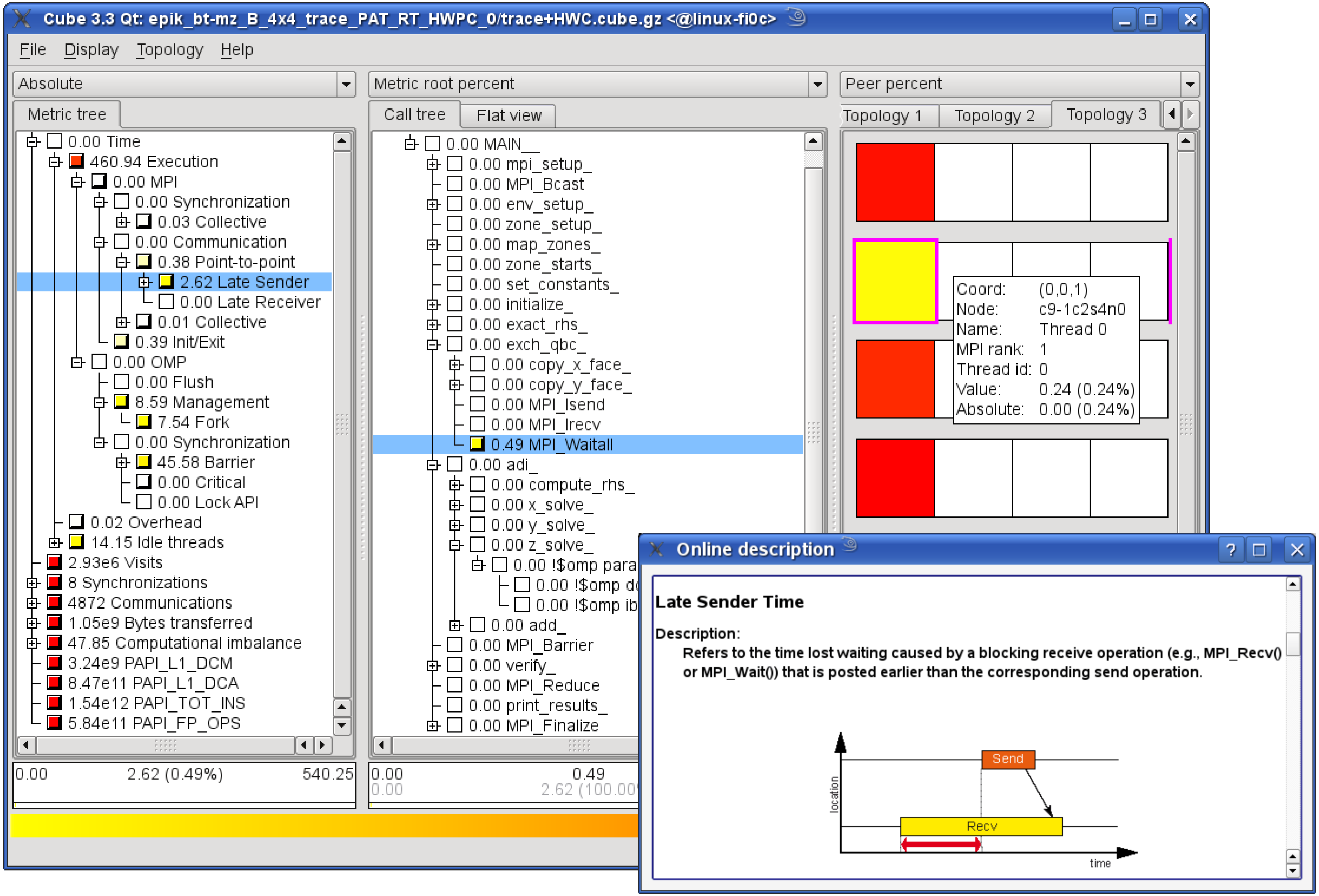
Coord:	(3,0,3)
Node:	c9-1c2s4n0
Name:	Thread 3
MPI rank:	3
Thread id:	3
Value:	96.57 (95.79%)
Absolute:	1.78 (95.79%)

```
0.00        47.85 (100.00%)        47.85
```

```
0.00              27.24               100.00
0.00         13.03 (27.24%)            47.85
```

```
0.00              100.00              100.00
0.34           0.81 +/- 0.53            1.85
```

Scalasca automatic trace analysis report



Cube 3.3 Qt: epik_bt-mz_B_4x4_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz <@linux-fi0c>

Metric tree:

- ☐ 0.00 Time
  - 🟥 460.94 Execution
    - ☐ 0.00 MPI
      - ☐ 0.00 Synchronization
        - ☐ 0.03 Collective
      - ☐ 0.00 Communication
        - 🟨 0.38 Point-to-point
          - 🟨 2.62 Late Sender
          - ☐ 0.00 Late Receiver
        - ☐ 0.01 Collective
      - 🟨 0.39 Init/Exit
    - ☐ 0.00 OMP
      - ☐ 0.00 Flush
      - 🟨 8.59 Management
        - 🟨 7.54 Fork
      - ☐ 0.00 Synchronization
        - 🟨 45.58 Barrier
        - ☐ 0.00 Critical
        - ☐ 0.00 Lock API
    - ☐ 0.02 Overhead
    - 🟨 14.15 Idle threads
- 🟥 2.93e6 Visits
- 🟥 8 Synchronizations
- 🟥 4872 Communications
- 🟥 1.05e9 Bytes transferred
- 🟥 47.85 Computational imbalance
- 🟥 3.24e9 PAPI_L1_DCM
- 🟥 8.47e11 PAPI_L1_DCA
- 🟥 1.54e12 PAPI_TOT_INS
- 🟥 5.84e11 PAPI_FP_OPS

0.00 2.62 (0.49%) 540.25

Call tree:

- ☐ 0.00 MAIN__
  - ☐ 0.00 mpi_setup_
  - ☐ 0.00 MPI_Bcast
  - ☐ 0.00 env_setup_
  - ☐ 0.00 zone_setup_
  - ☐ 0.00 map_zones_
  - ☐ 0.00 zone_starts_
  - ☐ 0.00 set_constants_
  - ☐ 0.00 initialize_
  - ☐ 0.00 exact_rhs_
  - ☐ 0.00 exch_qbc_
    - ☐ 0.00 copy_x_face_
    - ☐ 0.00 copy_y_face_
    - ☐ 0.00 MPI_Isend
    - ☐ 0.00 MPI_Irecv
    - 🟨 0.49 MPI_Waitall
  - ☐ 0.00 adi_
    - ☐ 0.00 compute_rhs_
    - ☐ 0.00 x_solve_
    - ☐ 0.00 y_solve_
    - ☐ 0.00 z_solve_
      - ☐ 0.00 !$omp para
        - ☐ 0.00 !$omp d
        - ☐ 0.00 !$omp ib
    - ☐ 0.00 add_
  - ☐ 0.00 MPI_Barrier
  - ☐ 0.00 verify_
  - ☐ 0.00 MPI_Reduce
  - ☐ 0.00 print_results_
  - ☐ 0.00 MPI_Finalize

0.00 0.49
0.00 2.62 (100.00

Coord:	(0,0,1)
Node:	c9-1c2s4n0
Name:	Thread 0
MPI rank:	1
Thread id:	0
Value:	0.24 (0.24%)
Absolute:	0.00 (0.24%)

Online description

Late Sender Time

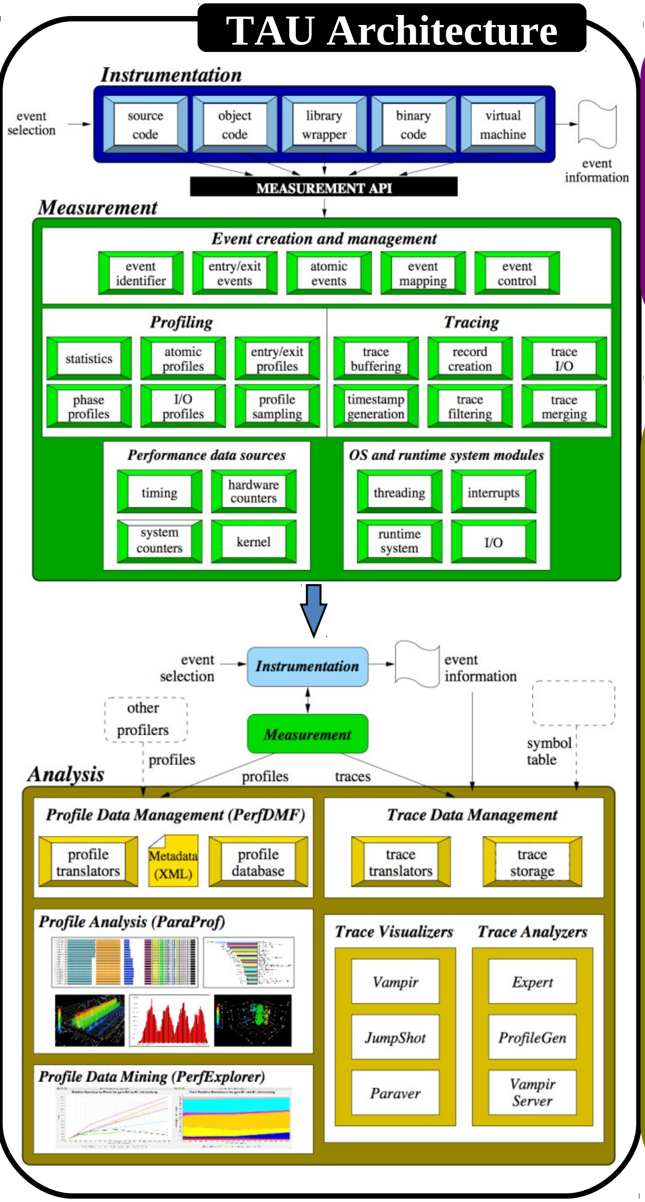
Description:
Refers to the time lost waiting caused by a blocking receive operation (e.g., MPI_Recv() or MPI_Wait()) that is posted earlier than the corresponding send operation.

TAU Performance System

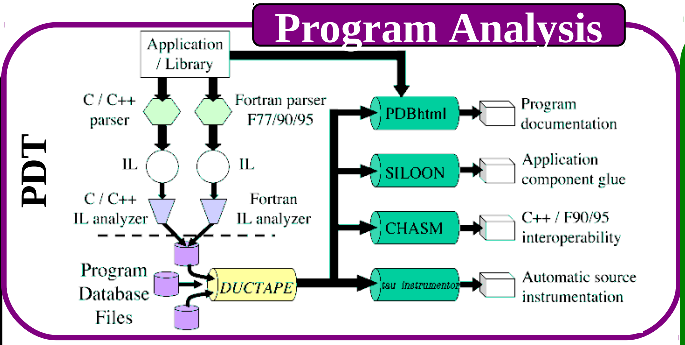
- Integrated performance toolkit
  - Instrumentation, measurement, analysis & visualization
    - Highly customizable installation, API, envvars & GUI
    - Supports multiple profiling & tracing capabilities
  - Performance data management & data mining
  - Targets all parallel programming/execution paradigms
    - Ported to a wide range of computer systems
  - Performance problem solving framework for HPC
  - Extensive bridges to/from other performance tools
    - PerfSuite, Scalasca, Vampir, ...
- Developed by U. Oregon/PRL
  - Broadly deployed open-source software
  - http://tau.uoregon.edu/

TAU Performance System components

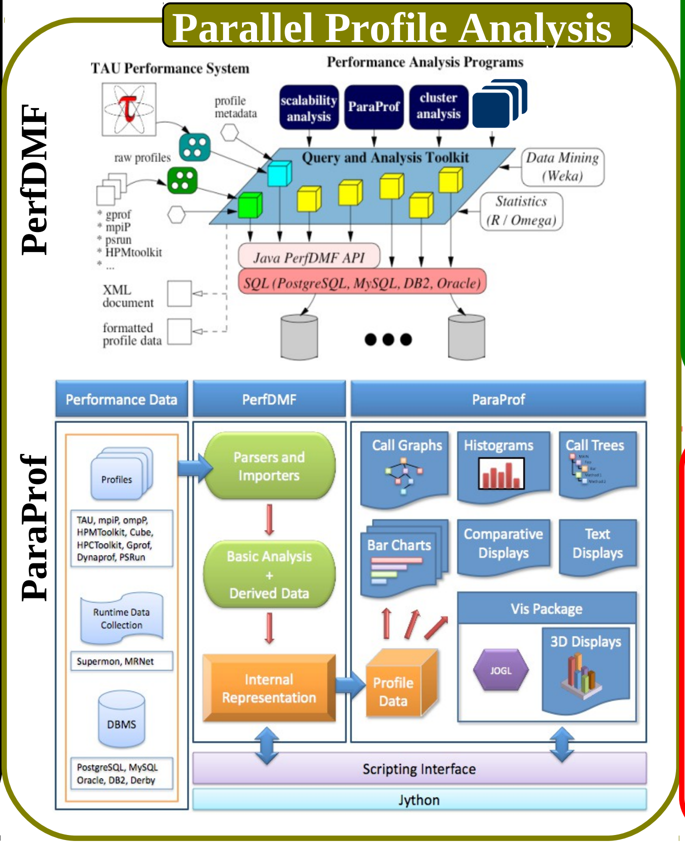
TAU Architecture



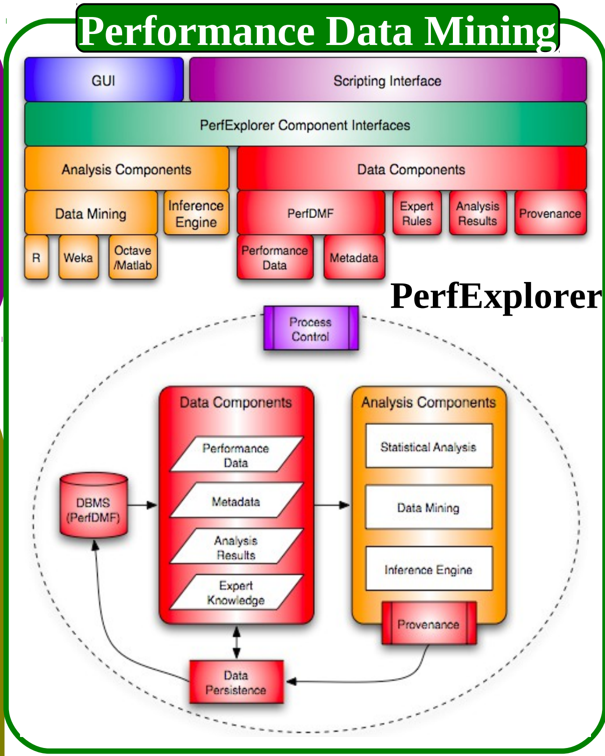
Program Analysis



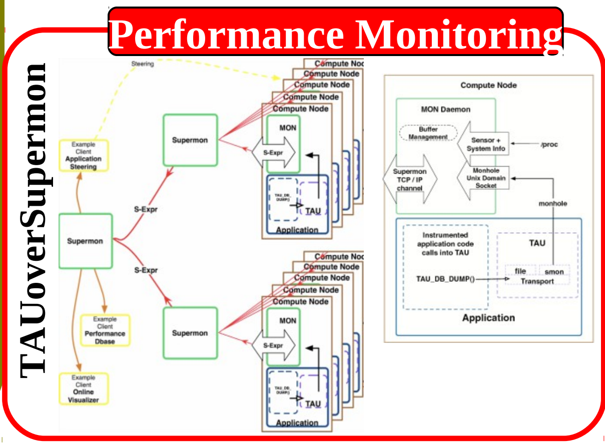
Parallel Profile Analysis



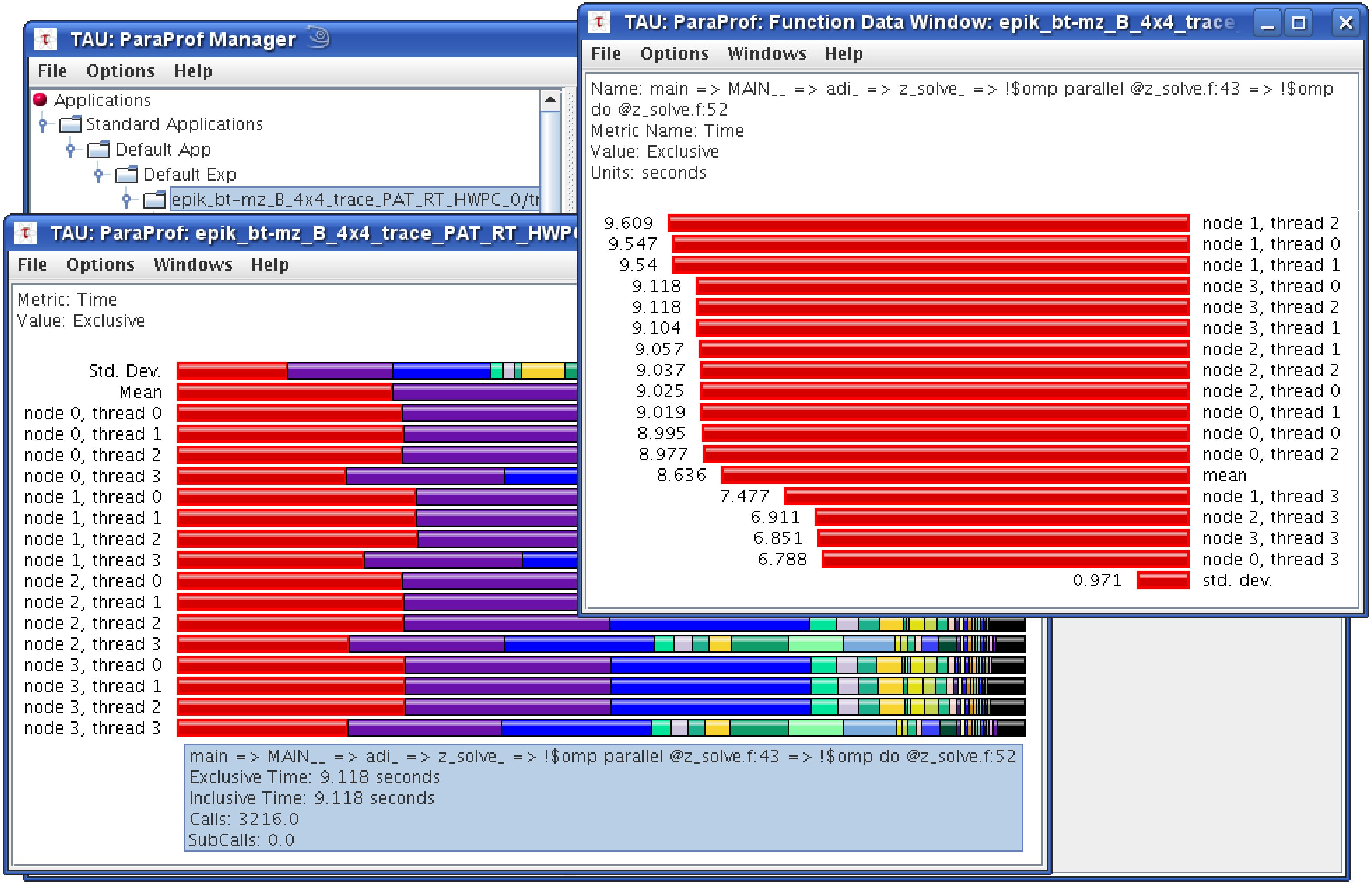
Performance Data Mining



Performance Monitoring



TAU ParaProf GUI displays (selected)


TAU: ParaProf Manager
File Options Help
Applications
Standard Applications
Default App
Default Exp
epik_bt-mz_B_4x4_trace_PAT_RT_HWPC_0/tr
TAU: ParaProf: epik_bt-mz_B_4x4_trace_PAT_RT_HWP
File Options Windows Help
TAU: ParaProf: Function Data Window: epik_bt-mz_B_4x4_trace
File Options Windows Help

Metric: Time
Value: Exclusive

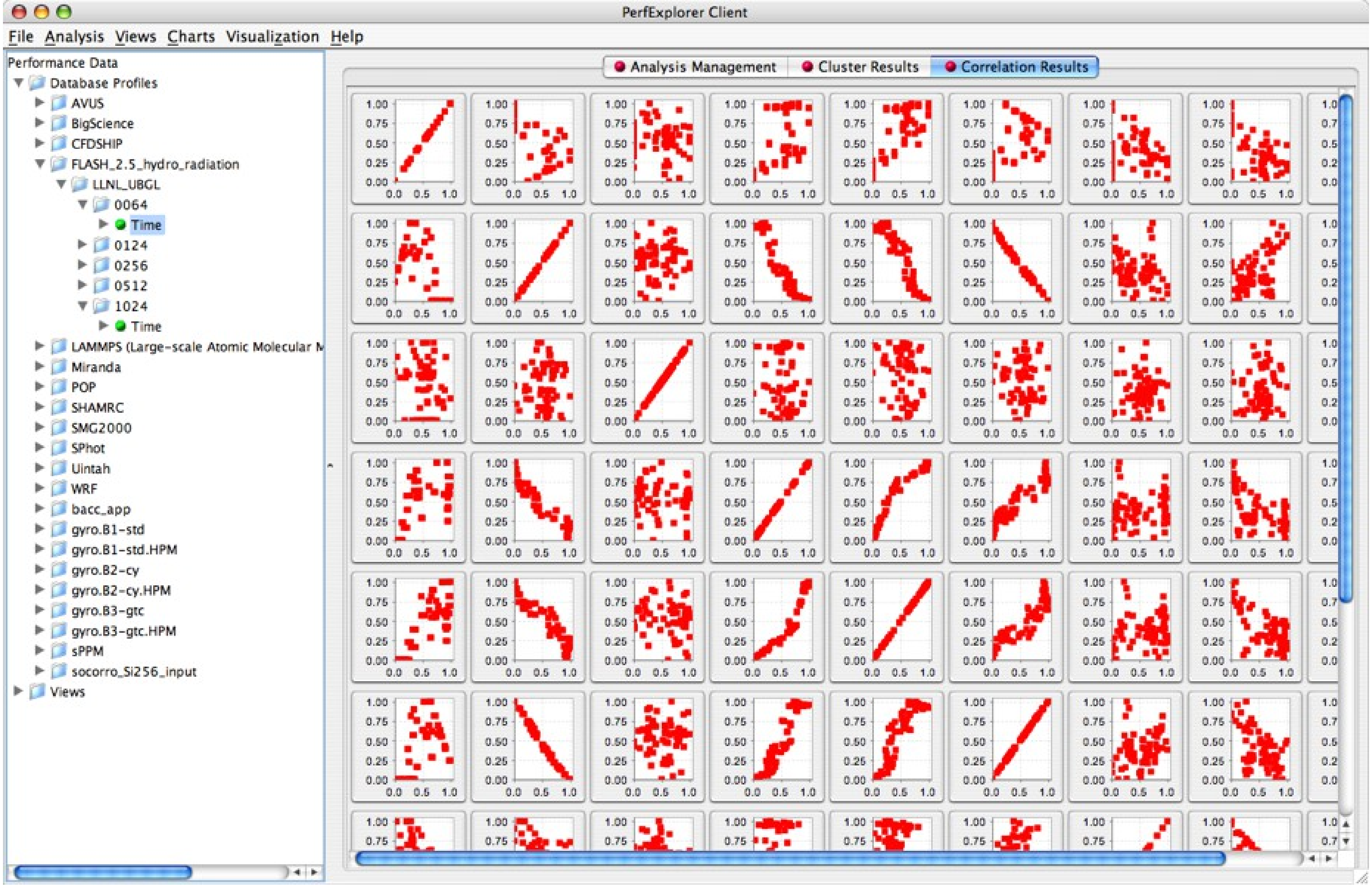
Std. Dev.
Mean
node 0, thread 0
node 0, thread 1
node 0, thread 2
node 0, thread 3
node 1, thread 0
node 1, thread 1
node 1, thread 2
node 1, thread 3
node 2, thread 0
node 2, thread 1
node 2, thread 2
node 2, thread 3
node 3, thread 0
node 3, thread 1
node 3, thread 2
node 3, thread 3

main => MAIN__ => adi_ => z_solve_ => !$omp parallel @z_solve.f:43 => !$omp do @z_solve.f:52
Exclusive Time: 9.118 seconds
Inclusive Time: 9.118 seconds
Calls: 3216.0
SubCalls: 0.0

Name: main => MAIN__ => adi_ => z_solve_ => !$omp parallel @z_solve.f:43 => !$omp do @z_solve.f:52
Metric Name: Time
Value: Exclusive
Units: seconds

```
9.609  node 1, thread 2
9.547  node 1, thread 0
9.54   node 1, thread 1
9.118  node 3, thread 0
9.118  node 3, thread 2
9.104  node 3, thread 1
9.057  node 2, thread 1
9.037  node 2, thread 2
9.025  node 2, thread 0
9.019  node 0, thread 1
8.995  node 0, thread 0
8.977  node 0, thread 2
8.636  mean
7.477  node 1, thread 3
6.911  node 2, thread 3
6.851  node 3, thread 3
6.788  node 0, thread 3
0.971  std. dev.
```

TAU PerfExplorer data mining



Vampir & VampirTrace

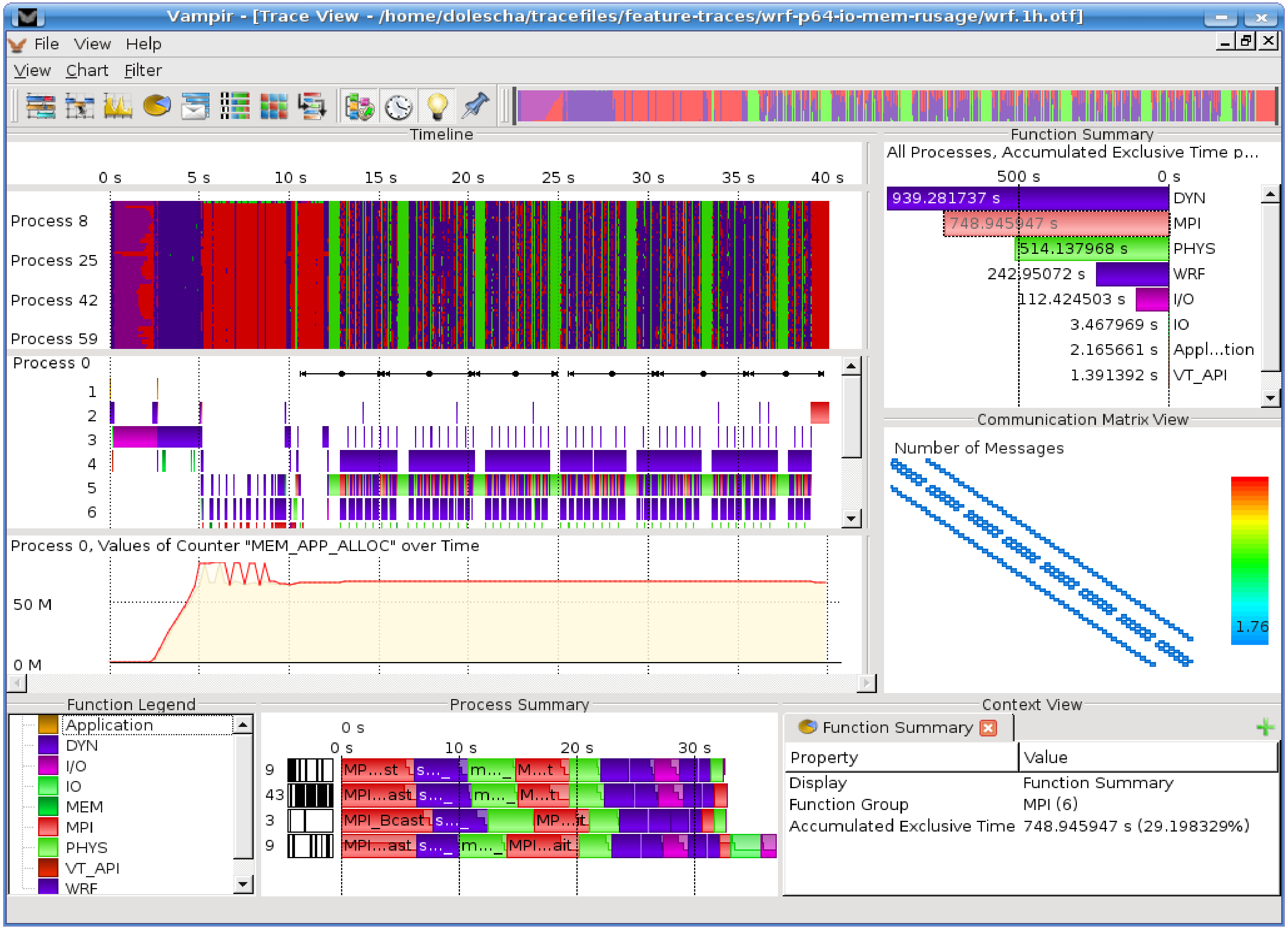
Interactive event trace analysis

- Alternative & supplement to automatic trace analysis
- Visual presentation of dynamic runtime behaviour
  - event timeline chart for states & interactions of processes/threads
  - communication statistics, summaries & more
- Interactive browsing, zooming, selecting
  - linked displays & statistics adapt to selected time interval (zoom)
  - scalable server runs in parallel to handle larger traces

Developed by TU Dresden ZIH

- Open-source VampirTrace library bundled with OpenMPI 1.3
- http://www.tu-dresden.de/zih/vampirtrace/
- Vampir Server & GUI have a commercial license
- http://www.vampir.eu/

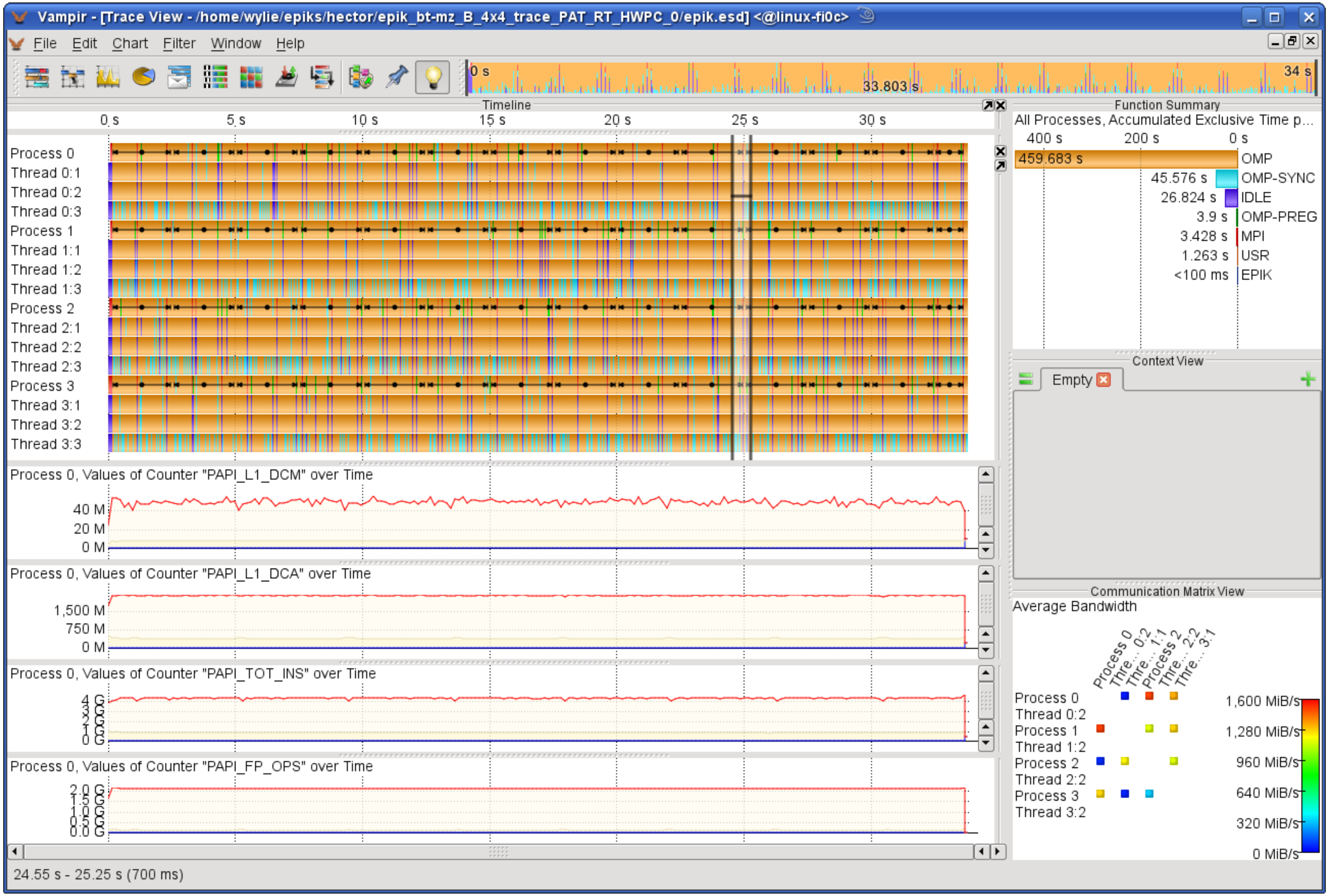
Vampir interactive trace analysis GUI



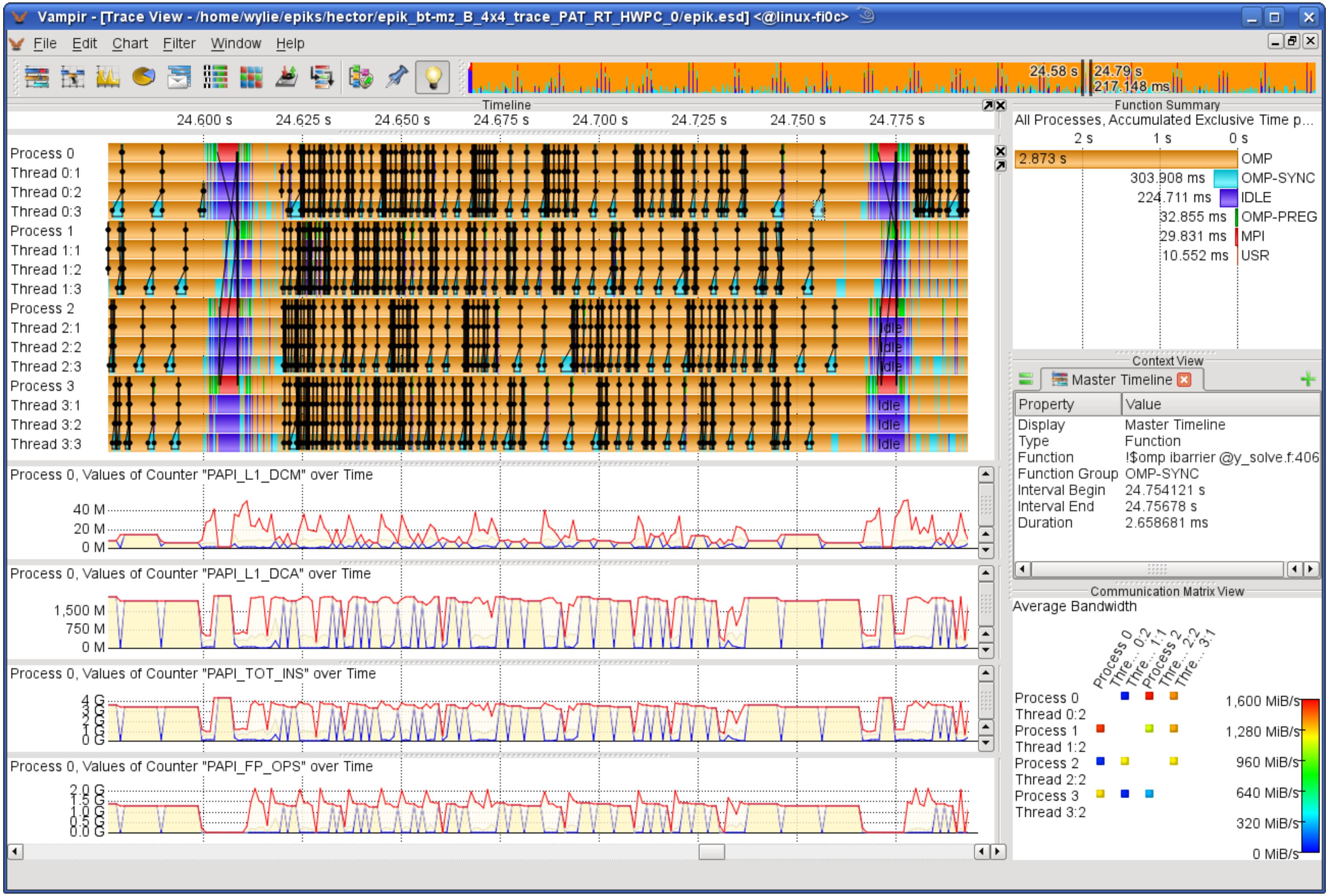
Vampir - [Trace View - /home/dolescha/tracefiles/feature-traces/wrf-p64-io-mem-rusage/wrf.1h.otf]

Property	Value
Display	Function Summary
Function Group	MPI (6)
Accumulated Exclusive Time	748.945947 s (29.198329%)

Vampir interactive trace analysis GUI



Vampir interactive trace analysis GUI (zoom)



Vampir - [Trace View - /home/wylie/epiks/hector/epik_bt-mz_B_4x4_trace_PAT_RT_HWPC_0/epik.esd] <@linux-fi0c>

Property	Value
Display	Master Timeline
Type	Function
Function	!$omp ibarrier @y_solve.f:406
Function Group	OMP-SYNC
Interval Begin	24.754121 s
Interval End	24.75678 s
Duration	2.658681 ms

Function Summary

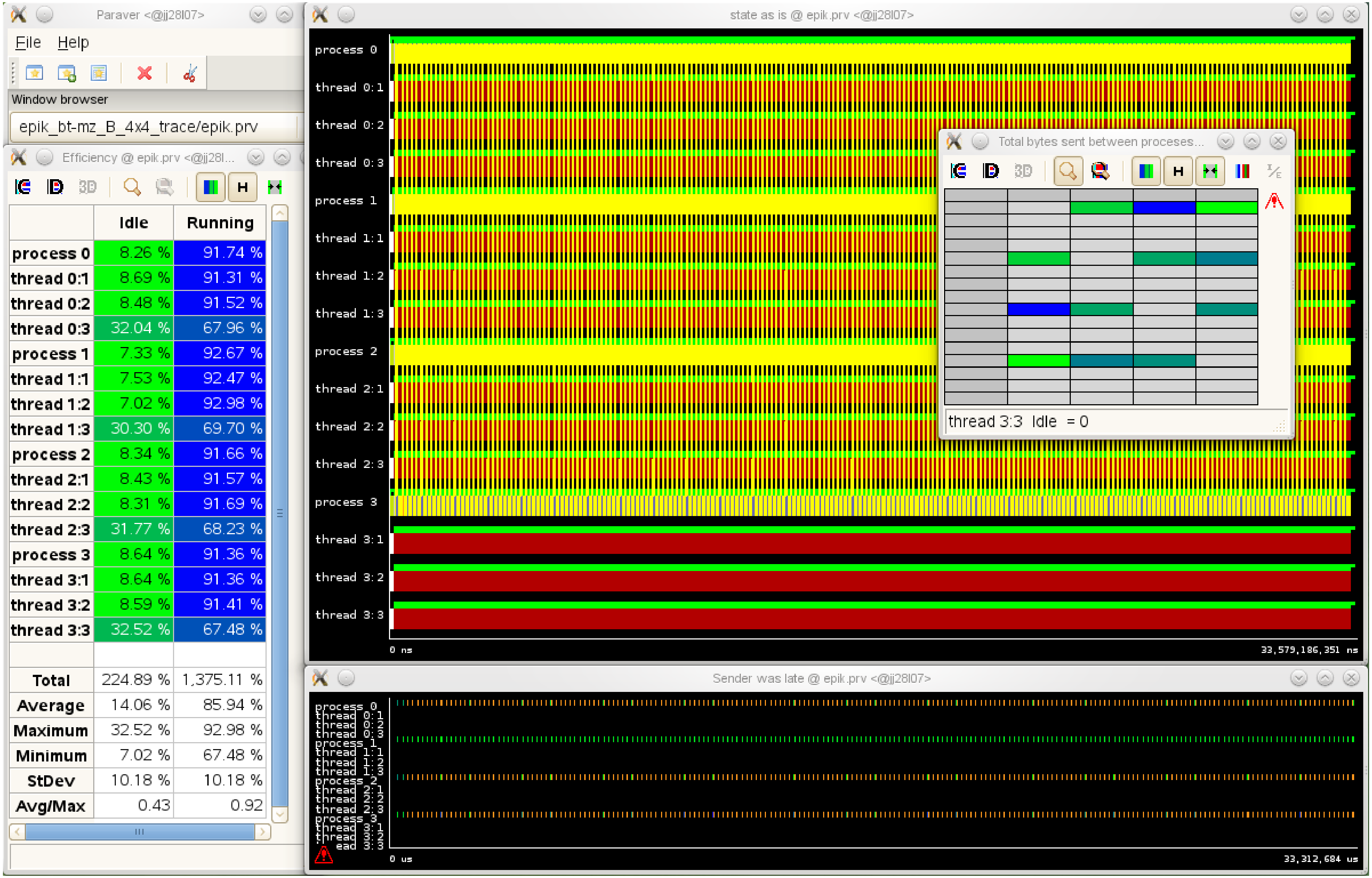
All Processes, Accumulated Exclusive Time p...

Time	Function Group
2.873 s	OMP
303.908 ms	OMP-SYNC
224.711 ms	IDLE
32.855 ms	OMP-PREG
29.831 ms	MPI
10.552 ms	USR

Paraver & Extrae

- Interactive event trace analysis
  - Visual presentation of dynamic runtime behaviour
    - event timeline chart for states & interactions of processes
    - Interactive browsing, zooming, selecting
  - Large variety of highly configurable analyses & displays
- Developed by Barcelona Supercomputing Center
  - Paraver trace analyser and Extrae measurement library
  - Open source available from http://www.bsc.es/paraver/

Paraver interactive trace analysis GUI



	Idle	Running
process 0	8.26 %	91.74 %
thread 0:1	8.69 %	91.31 %
thread 0:2	8.48 %	91.52 %
thread 0:3	32.04 %	67.96 %
process 1	7.33 %	92.67 %
thread 1:1	7.53 %	92.47 %
thread 1:2	7.02 %	92.98 %
thread 1:3	30.30 %	69.70 %
process 2	8.34 %	91.66 %
thread 2:1	8.43 %	91.57 %
thread 2:2	8.31 %	91.69 %
thread 2:3	31.77 %	68.23 %
process 3	8.64 %	91.36 %
thread 3:1	8.64 %	91.36 %
thread 3:2	8.59 %	91.41 %
thread 3:3	32.52 %	67.48 %
Total	224.89 %	1,375.11 %
Average	14.06 %	85.94 %
Maximum	32.52 %	92.98 %
Minimum	7.02 %	67.48 %
StDev	10.18 %	10.18 %
Avg/Max	0.43	0.92

thread 3:3 Idle = 0

MAQAO

- Modular Assembler Quality Analyzer & Optimizer
  - Framework for binary manipulation
    - using plugins and scripting language
  - Tool exploiting framework to produce reports
    - fast prototyping and batch interface
  - STAN static performance model
  - MIL instrumentation language for dynamic analysis
    - building custom performance evaluation tools using HWCs
    - instrumentation of functions, loops, blocks & instructions
- Developed by UVSQ Exascale Computing Research lab
  - Supports Intel x86_64 microarchitecture
  - Available from www.maqao.org

VI-HPS component technologies

Key tool components also provided as open-source

- Program development environment
  - Eclipse PTP ETFw, UNITE
- Program/library instrumentation
  - COBI, OPARI, PDToolkit
- Runtime measurement systems
  - P^{n}MPI, Score-P, UniMCI
- Scalable I/O
  - SIONlib
- Libraries & tools for handling (and converting) traces
  - EPILOG, OTF, PEARL
- Analysis algebra & hierarchical/topological presentation
  - CUBE

Score-P

Scalable performance measurement infrastructure

- Supports instrumentation, profiling & trace collection, as well as online analysis of HPC parallel applications
- Works with Periscope, Scalasca, TAU & Vampir prototypes
- Based on updated tool components
  - CUBE4 profile data utilities & GUI
  - OA online access interface to performance measurements
  - OPARI2 OpenMP & pragma instrumenter
  - OTF2 open trace format

Created by German BMBF SILC & US DOE PRIMA projects

- JSC, RWTH, TUD, TUM, GNS, GRS, GWT & UO PRL
- Available as BSD open-source from http://www.score-p.org/

SIONlib

Portable native parallel I/O library & utilities

- Scalable massively-parallel I/O to task-local files
- Manages single or multiple physical files on disk
  - optimizes bandwidth available from I/O servers by matching blocksizes/alignment, reduces metadata-server contention
- POSIX-I/O-compatible sequential & parallel API
  - adoption requires minimal source-code changes
- Tuned for common parallel filesystems
  - GPFS (BlueGene), Lustre (Cray), ...
- Convenient for application I/O, checkpointing,
  - Used by Scalasca tracing (when configured)

Developed by JSC

- Available as open-source from http://www.fz-juelich.de/jsc/sionlib/

UNITE

Uniform integrated tool environment

- Manages installation & access to program development tools
  - based on software environment management “modules”
  - commonly used on most cluster and HPC systems
  - configurable for multiple MPI libraries & compiler suites
- Specifies how & where tools packages get installed
  - including integrating tools where possible
- Defines standard module names and different versions
- Supplies pre-defined module files
- Configurable to co-exist with local installations & policies

Developed by JSC, RWTH & TUD

- Available as open-source from http://www.vi-hps.org/projects/unite/