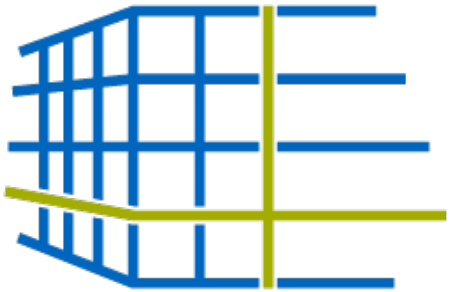
VI-HPS

MAC / VI-HPS Workshop
Profiling & Performance Analysis of Parallel Applications
KAUST, Saudi Arabia

October/November 2010



Munich Centre for Advanced Computing

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Presenters/Guides
  - Hans-Joachim Bungartz, Michael Gerndt, Josef Weidendorfer, Tobias Weinzierl (TU Munich)
  - Matthias Weber (TU Dresden ZIH)
  - Felix Wolf (GRS-Sim)
  - Brian Wylie (JSC)
- Thanks
  - VI-HPS/POINT partners & associates
    - U Oregon PRL for preparing the LiveDVD
  - Local arrangements & facilities
    - KSL, KAUST

Starting 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++, mixed-language, ...
- What platforms/systems *must* your app(s) run well on?
  - Cray XT, IBM BlueGene/P, 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, mpiP/ompP, IBM HPC Toolkit, ...

Virtual Institute – High Productivity Supercomputing

VI-HPS

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

- Funded by Helmholtz Association of German Research Centres

- 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 & associates

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

German Research School
- Laboratory of Parallel Programming

Technical University of Munich
- Chair for Computer Architecture

University of Stuttgart
- HPC Centre

Productivity tools

- Marmot
  - Free MPI correctness checking tool
- PAPI
  - Free library interfacing to hardware performance counters
- Periscope
  - Prototype automatic analysis tool using an on-line distributed search for performance inefficiencies
- Scalasca
  - Open-source toolset for analysing the performance behaviour of parallel applications to automatically identify inefficiencies
- Vampir/VampirTrace
  - Commercial tool for graphical trace visualization & analysis, and open-source event tracing library

[Productivity Tools Live-DVD contains latest tools releases]

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

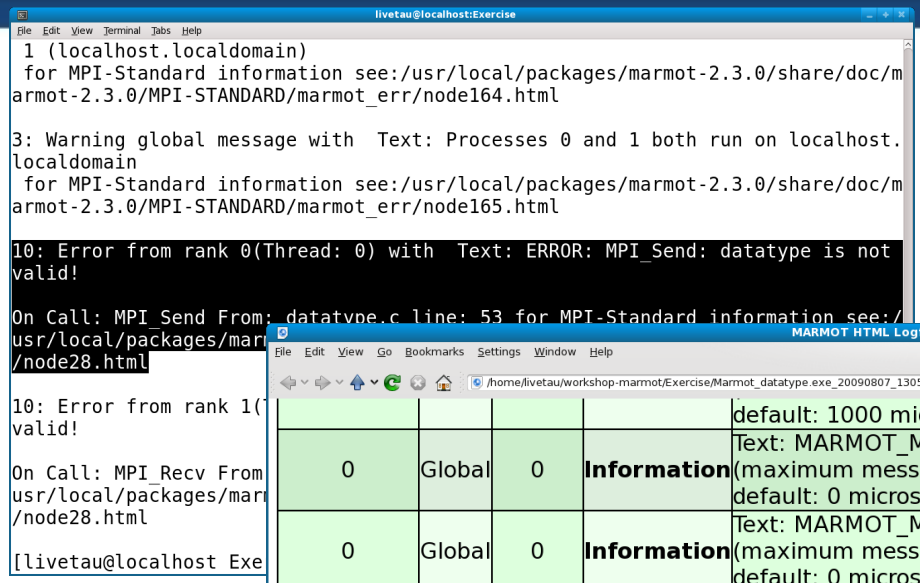
Implementation

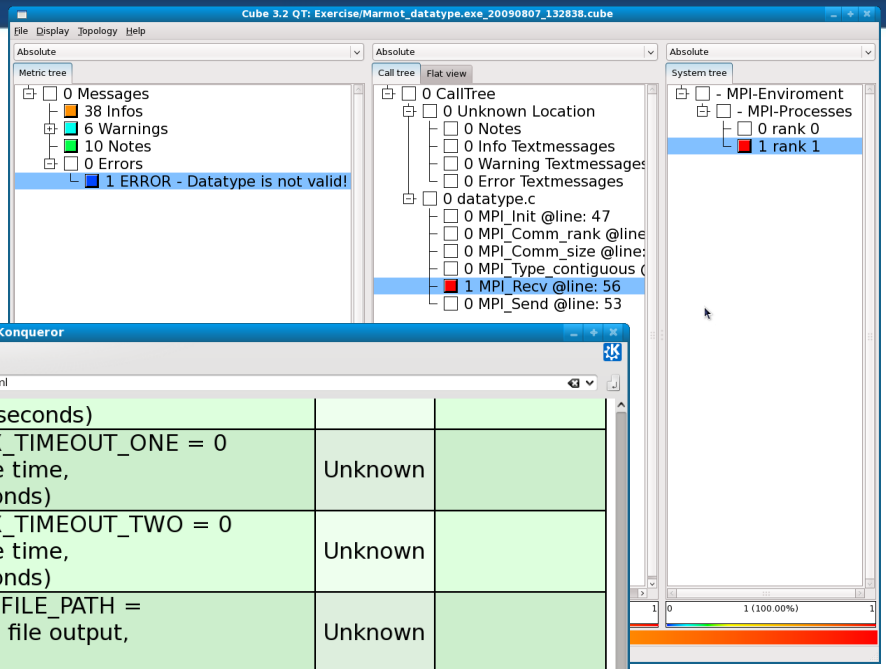
- C++ library gets linked to the application
- Does not require source code modifications
- Additional process used as DebugServer
- Results written in a log file (ASCII/HTML/CUBE)

Developed by HLRS & TU Dresden

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

Marmot logfiles





MARMOT HTML Logfile - Konqueror

File Edit View Go Bookmarks Settings Window Help

/home/livetau/workshop-marmot/Exercise/Marmot_datatype.exe_20090807_130509.html

				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

PAPI

Portable performance counter library

- 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
  - PerfSuite, Periscope, Scalasca, TAU, VampirTrace, ...

Developed by UTK-ICL

- Available as open-source for most modern processors
  http://icl.cs.utk.edu/papi/

Periscope

Automated profile-based performance analysis

- Iterative on-line performance analysis
- Automatic search for bottlenecks based on properties formalizing expert knowledge
  - MPI wait states
  - Processor utilization hardware counters
- Multiple distributed hierarchical agents
- Eclipse-based integrated environment

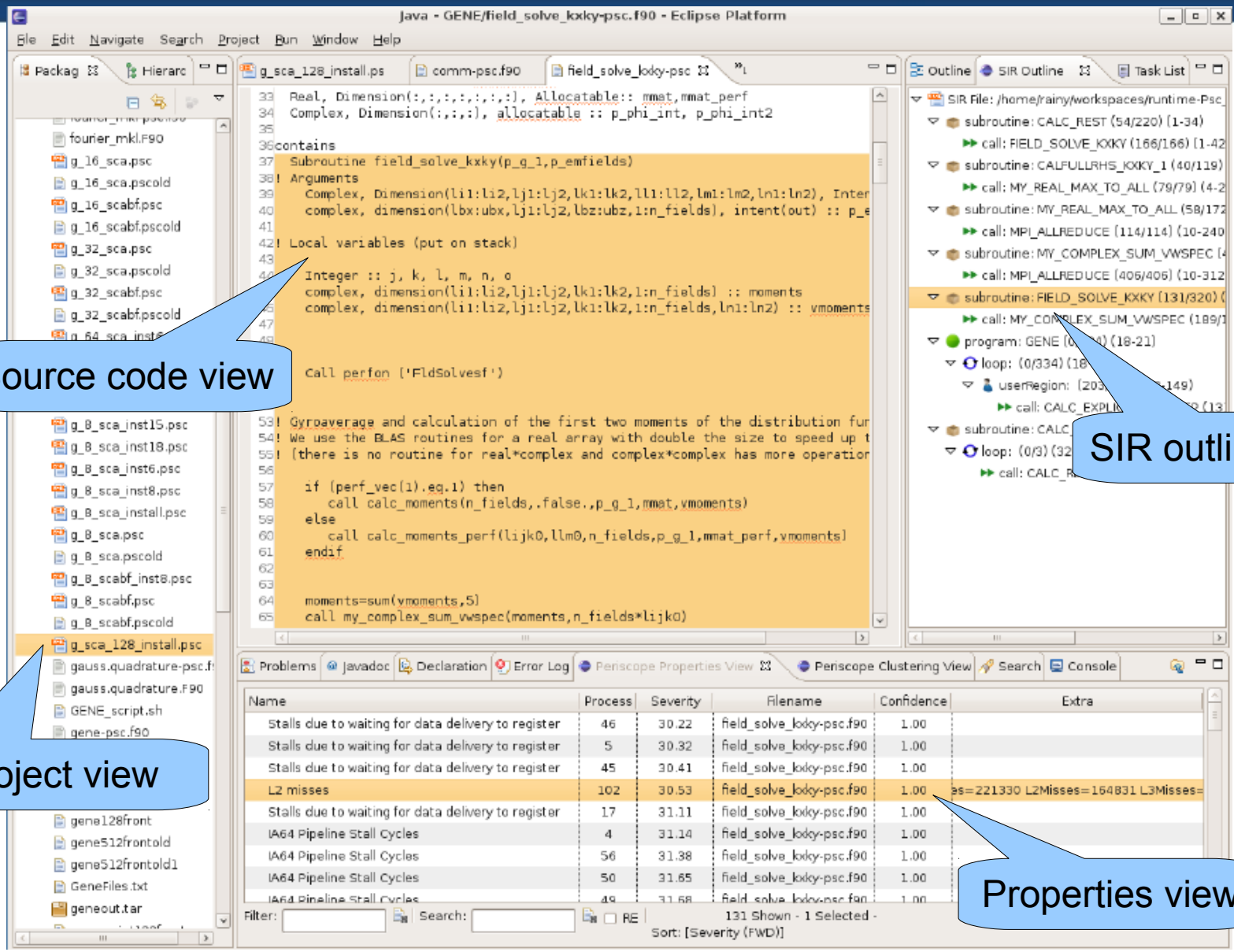
Supports

- 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 plug-in to Eclipse



Scalasca

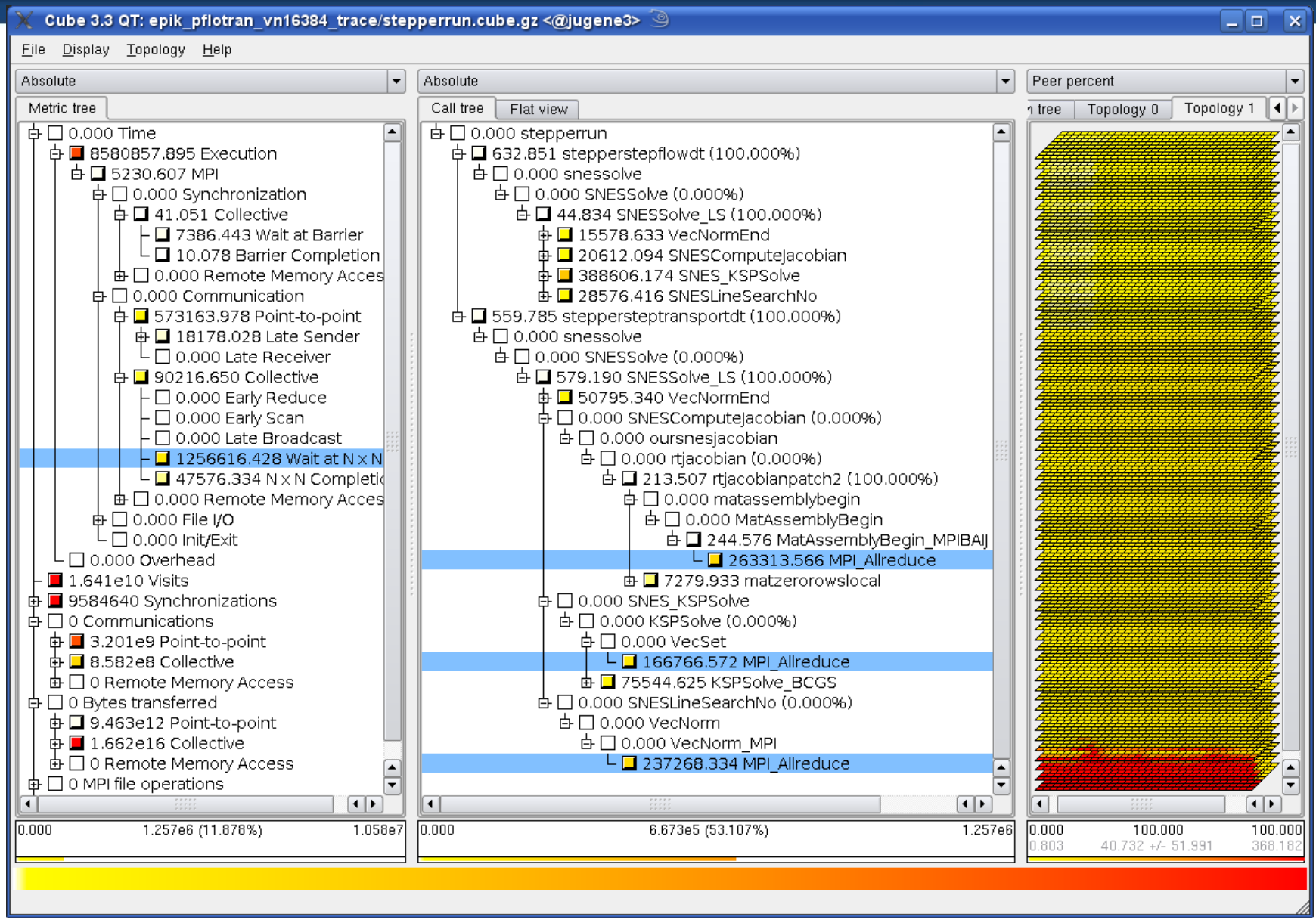
Automatic performance analysis toolset

- 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



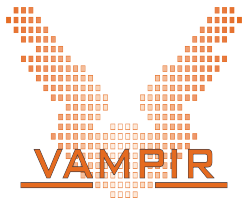
Vampir & VampirTrace

Interactive event trace analysis

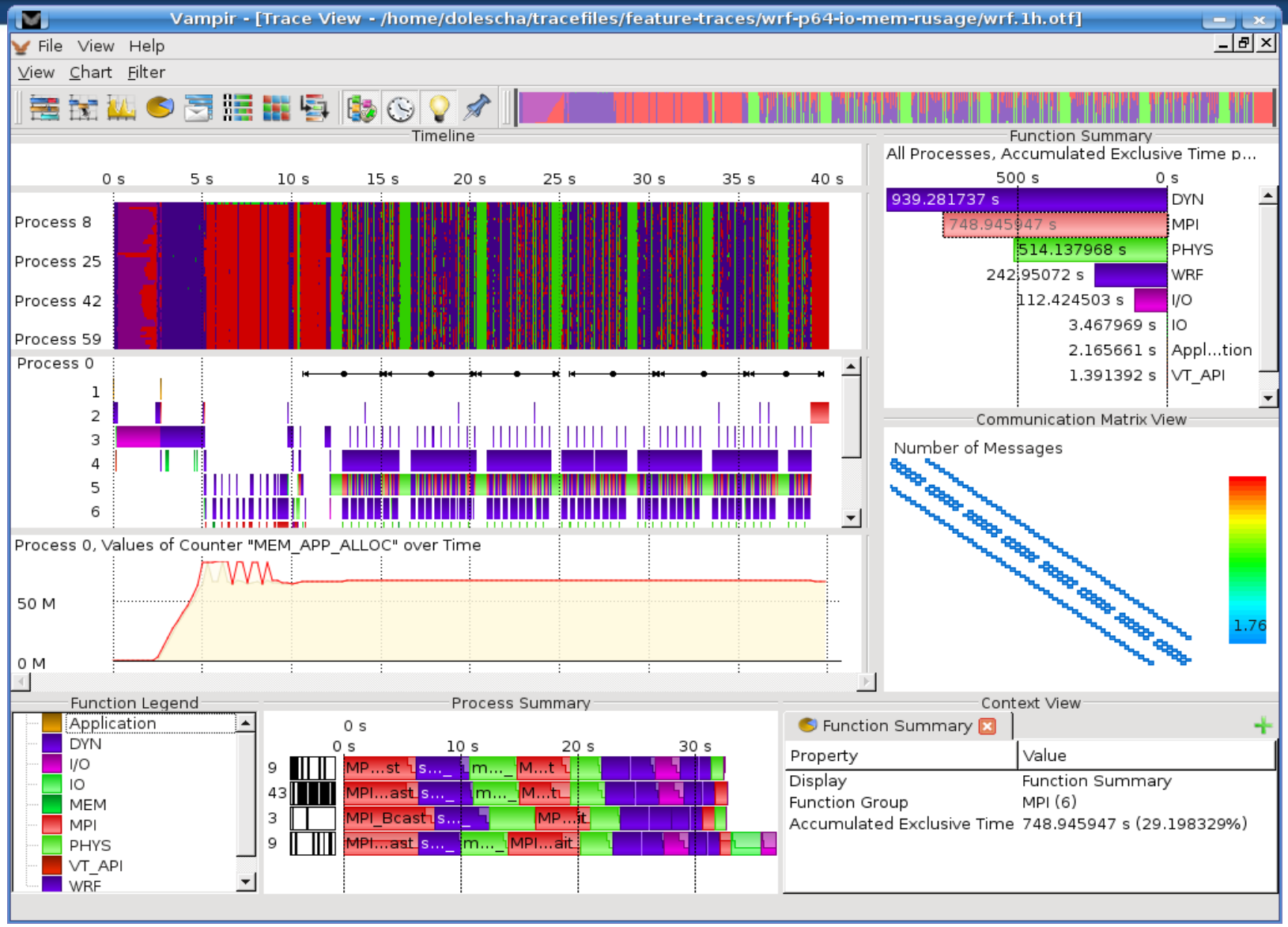
- Alternative & supplement to automatic trace analysis
- Visual presentation of dynamic runtime behaviour
  - event timeline chart for states & interactions of processes/threads
  - communication statistics, summaries & more
- Interactive browsing, zooming, selecting
  - linked displays & statistics 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 offered with a commercial license
- http://www.vampir.eu/



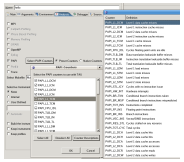
Vampir interactive trace analysis GUI



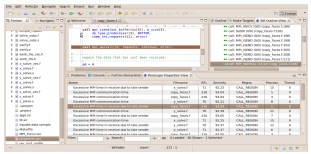
Technologies and their integration

VI-HPS

PAPI



THE UNIVERSITY of TENNESSEE

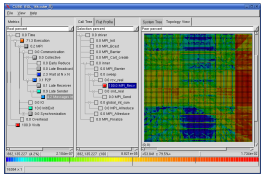


PERISCOPE

TECHNISCHE UNIVERSITÄT MÜNCHEN

Hardware monitoring

Automatic profile & trace analysis

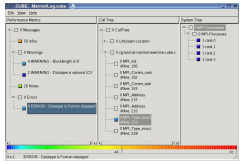


SCALASCA

JÜLICH FORSCHUNGSZENTRUM

German Research School for Simulation Sciences

MARMOT



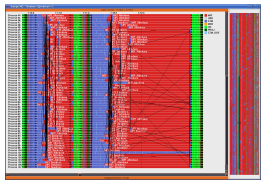
TECHNISCHE UNIVERSITÄT DRESDEN

Universität Stuttgart

Error detection

Visual trace analysis

VAMPIR



TECHNISCHE UNIVERSITÄT DRESDEN

Execution

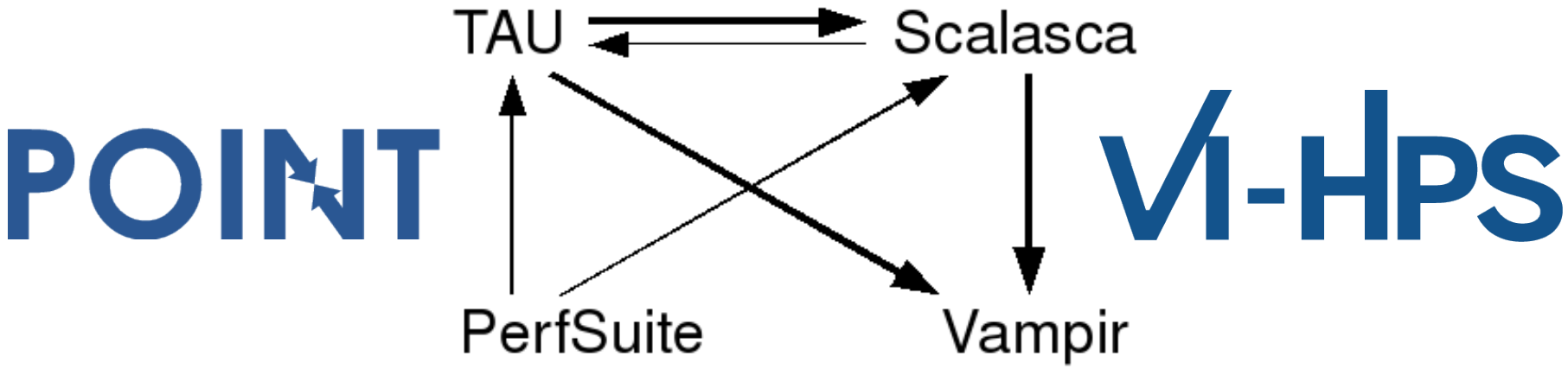
Optimization

VI-HPS component technologies

Key tool components also provided as open-source

- Program/library instrumentation
  - OPARI, POMP, PDToolkit
- MPI library/tool integration
  - UniMCI
- Scalable I/O
  - SIONlib
- Libraries & tools for handling (and converting) traces
  - EPILOG, EARL, PEARL, OTF
- Analysis algebra & hierarchical/topological presentation
  - CUBE

POINT/VI-HPS collaboration



VI-HPS collaborates with the POINT project in the USA

- Petascale Productivity from Open, Integrated Tools
- Funded by US NSF SDCI, Software Improvement & Support
- University of Oregon, University of Tennessee, UIUC NCSA, and Pittsburgh Supercomputing Center
- www.nic.uoregon.edu/point

PerfSuite

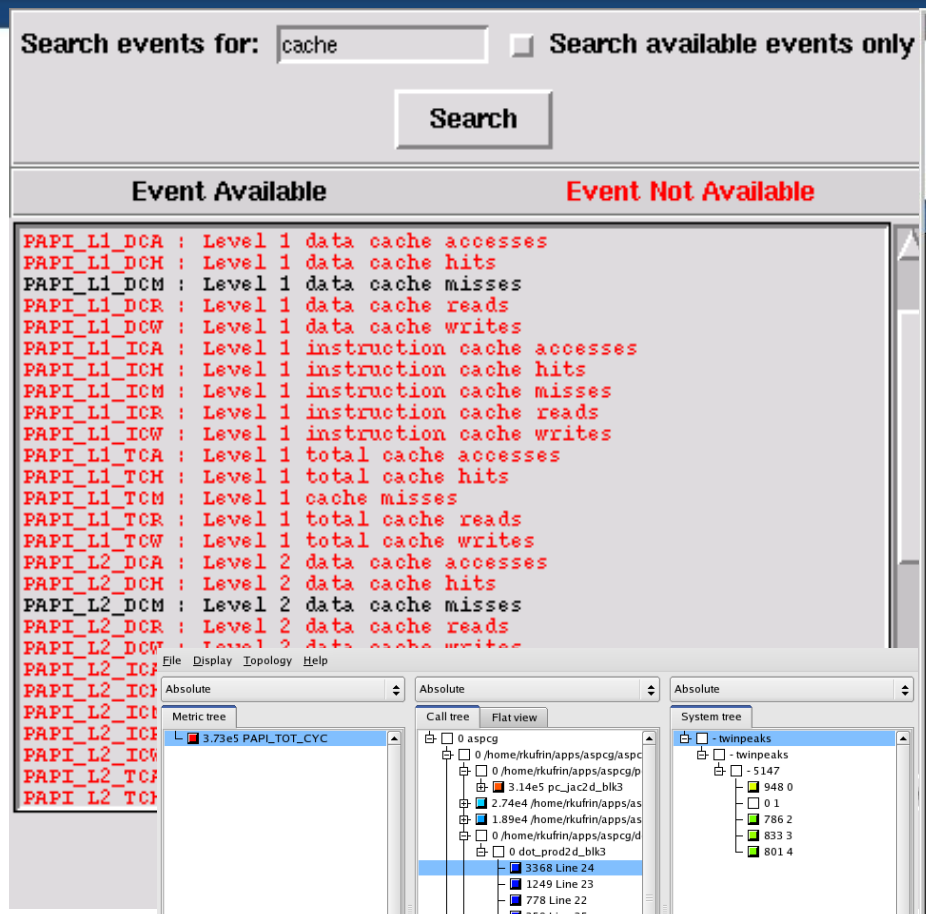
Entry-level (routine) profiling tools

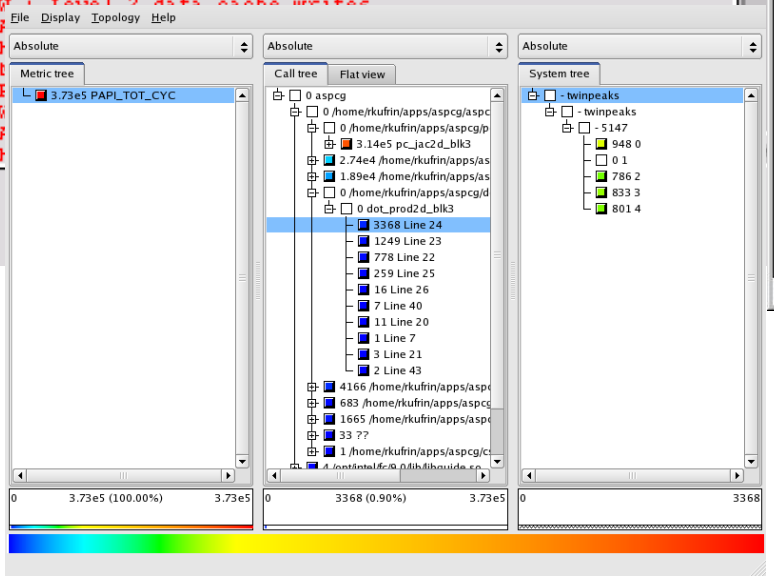
- Intended to be simple to use, low measurement overhead
  - works with unmodified, dynamically-linked executables
- Statistical sampling profiles based on time or HWC events
- XML-based reports with configurable processing utilities
  - can be viewed in Web browser, with ParaProf or Cube3
- Processor inventory utility captures measurement metadata
- Performance event measurement configuration utility

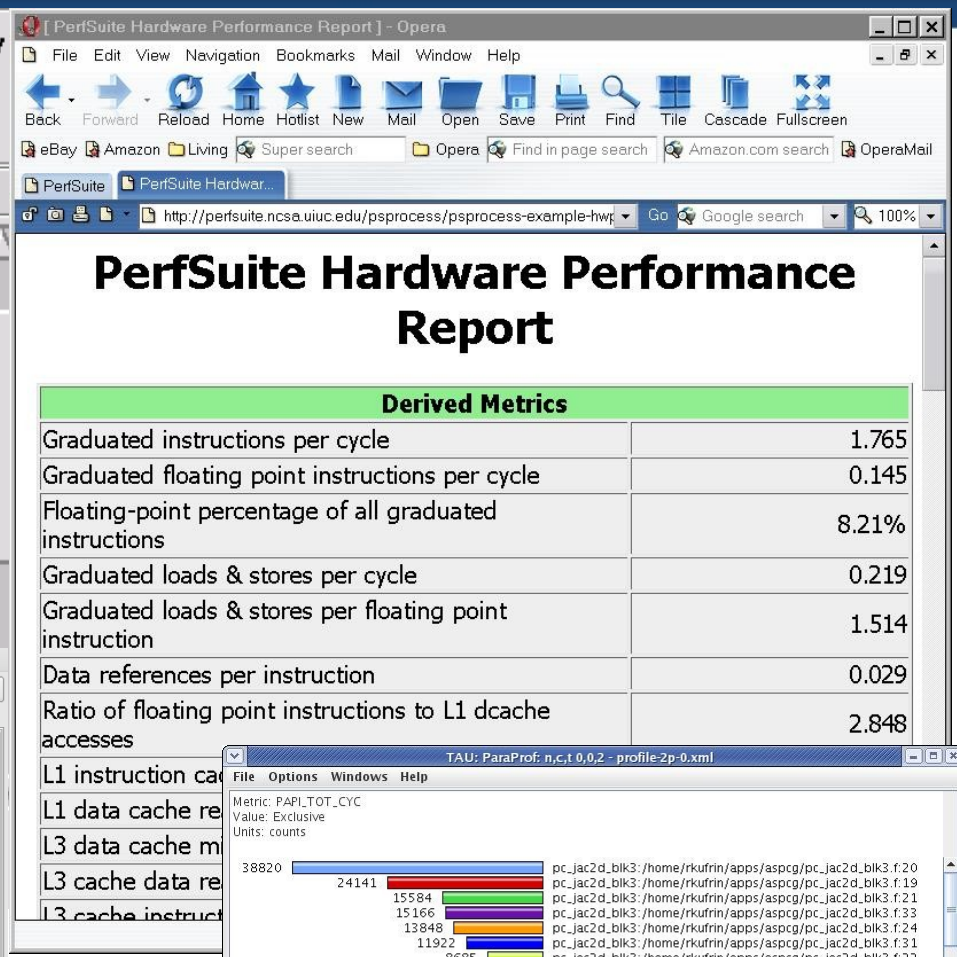
Developed by UIUC/NCSA

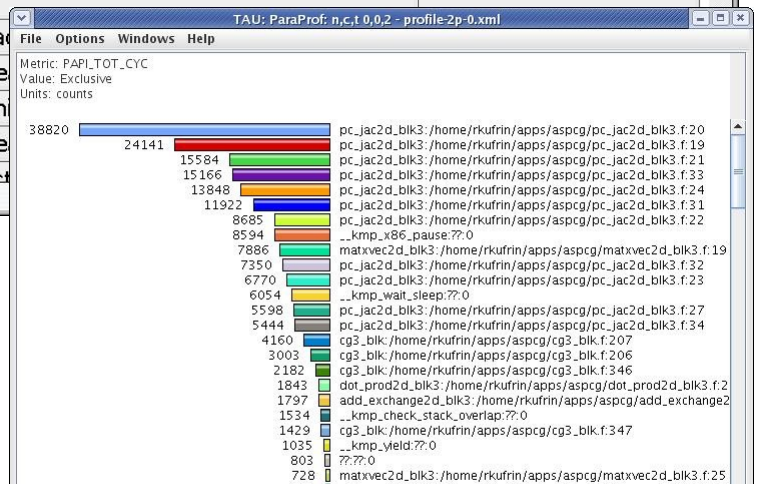
- Available as open-source for x86, x86-64 & ia64 Linux
- Can be used with MPI, OpenMP & pthreads
- http://perfsuite.ncsa.uiuc.edu/

PerfSuite psconfig & performance reports









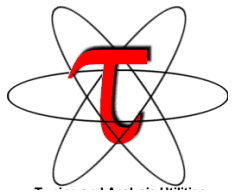
TAU Performance System

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

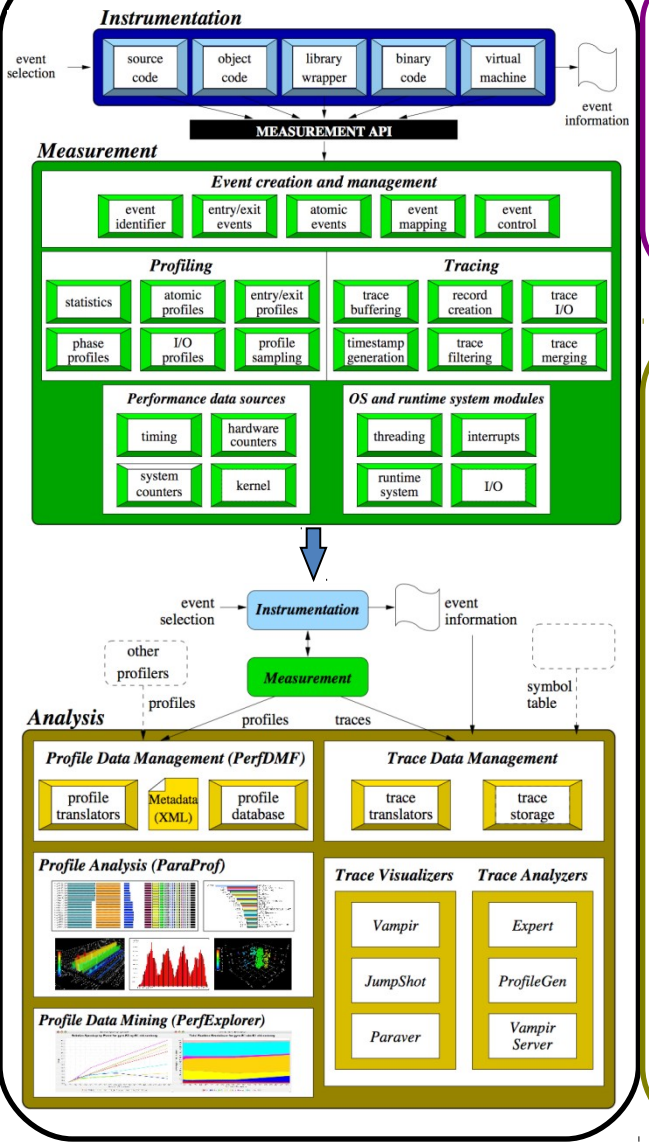
- Broadly deployed open-source software
- http://tau.uoregon.edu/



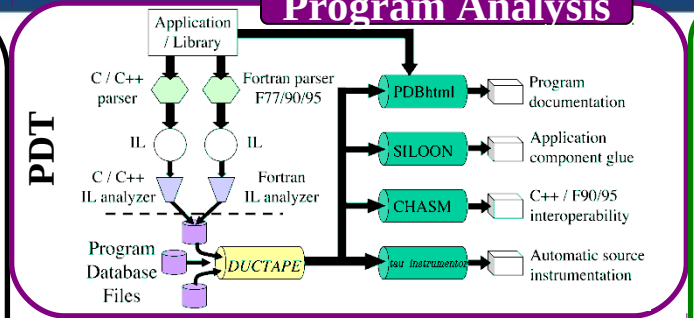
TAU Performance System components

VI-HPS

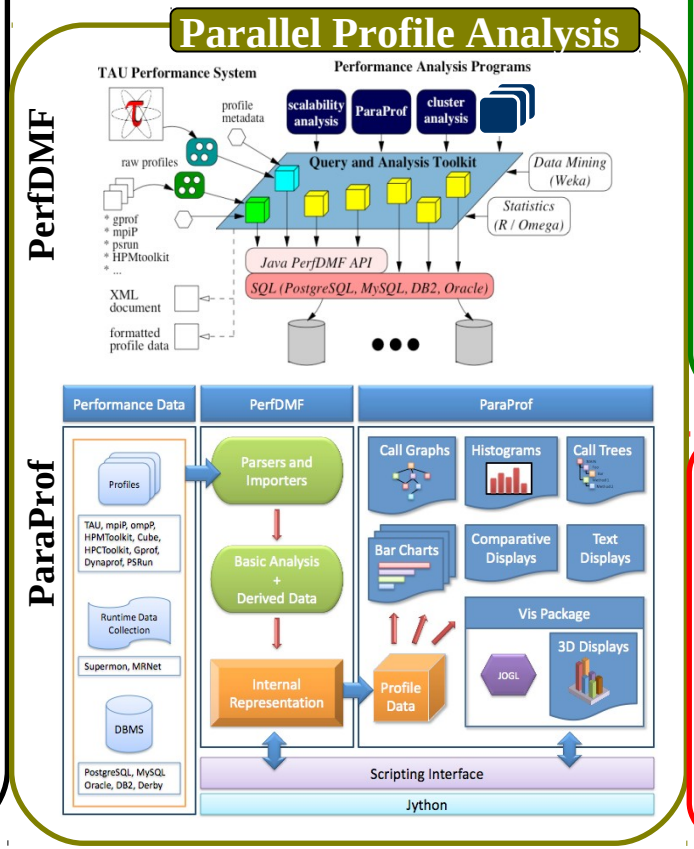
TAU Architecture



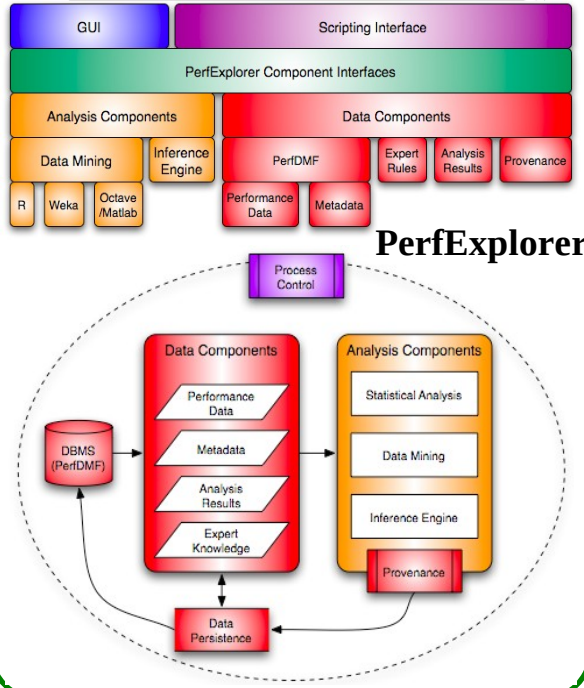
Program Analysis



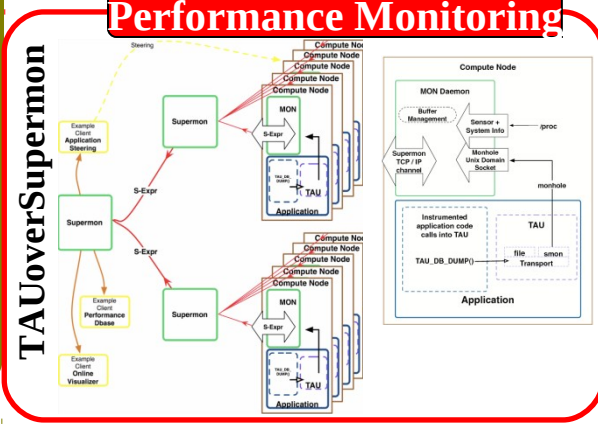
Parallel Profile Analysis



Performance Data Mining



Performance Monitoring



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 Tuning Workshop series
  - Aachen (3/08), Dresden (10/08), Jülich (2/09), Bremen (9/09), Garching (3/10), Amsterdam (05/10)
- Joint POINT/VI-HPS Tutorial series
  - SC (11/08), ICCS (5/09), SC (11/09), SC (11/10)
- Training with individual tools & platforms (e.g., BlueGene)

Outline (day 1)

Sunday 31st October

- 08:30 (registration & notebook computer set-up)
- 09:00 Welcome [Bungartz, TUM]
- 09:15 Introduction [Wylie, JSC]
  - Virtual Institute – High Productivity Supercomputing
  - Building and running the tutorial exercise NPB3.3-MPI/BT
- 09:45 Intro. to parallel performance analysis [Wolf, GRS]
- 10:30 ***Cachegrind*** [Weidendorfer, TUM]
- 12:00 (lunch)
- 13:30 ***Periscope*** [Gerndt, TUM]
- 15:00 ***Scalasca*** [Wylie, JSC]
- 16:30 Review of day
- 17:00 (adjourn)

Outline (day 2)

Monday 1st November

- 09:00 ***Vampir*** [Weber, TUD-ZIH]
- 10:30 IBM HPC Toolkit [Allsopp, KSL]
- 11:00 Wrap-up discussion [Bungartz/Weinzierl, TUM]
- 11:30 (lunch)
- 13:30 KSL User Assistance [all]
  - Analysis & tuning of participants' application codes
    - Ensure your code builds and runs to completion in a reasonable time (say 15 minutes) with an appropriate dataset for initial analyses
    - Also prepare some larger/longer scalability “production” configurations (perhaps with only a few iterations/steps) to analyse if time permits
    - Consider preparing different versions (algorithms/optimizations) or systems to compare
- 17:00 (adjourn)

POINT/VI-HPS Live-DVD

- Bootable Linux installation on DVD (or USB memory stick)
- Includes everything needed to try out our parallel tools on an x86-architecture notebook computer
  - GCC compiler suite (with OpenMP support), OpenMPI library
  - POINT tools: PAPI, PerfSuite, TAU
  - VI-HPS tools: Marmot, Periscope, Scalasca, VT/Vampir*
  - Other tools: BUPC, dyninst, Eclipse/PTP, PPW, TotalView*
    - * time/capability-limited evaluation licences provided for commercial products
  - Manuals/User Guides
  - Tutorial exercises and examples
- Prepared by U. Oregon Performance Research Laboratory
  - Sameer Shende