Virtual Institute – High Productivity Supercomputing

# 4th VI-HPS Tuning Workshop
in conjunction with POINT
hosted by HLRN in Bremen

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

9-11 September 2009

POINT

HLRN

# Introduction

- Presenters/Guides
  - Markus Geimer & Brian Wylie (Jülich Supercomputing Centre)
  - Tobias Hilbrich & Jens Doleschal (TU Dresden ZIH)
  - Allen Malony (University of Oregon)

- Thanks
  - Host: Universität Bremen
    - Thorsten Coordes (ZARM), ...
  - Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen
    - Hinnerk Stüben & Wolfgang Baumann (ZIB), Lars Nerger (AWI), ...
  - VI-HPS/POINT partners
    - Sameer Shende & Alan Morris (UO), Agnes Mendes (RWTH), ...
  - You
    - *Your Name Here*

# Participant survey

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

- 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?
  - SGI Altix ICE, SGI Altix, IBM Power/AIX, Linux cluster™, ...
- Who's already familiar with *serial* performance analysis?
  - Which tools have you used?
    - time, print/printf, prof/gprof, SpeedShop, VTune, ...
- Who's already familiar with *parallel* performance analysis?
  - Which tools have you used?
    - time, print/printf, prof/gprof, mpiP/ompP, MPInside, ...

# Warning

The workshop concentrates on ***hands-on*** use of correctness and performance analysis tools with your own application(s):

- Who has prepared their app(s) to be analysed and tuned?
  - A small yet *representative* test case should build
  - ... and run (correctly) to completion within a few minutes
  - ... on a relatively small number of processors/cores
- Who has prepared to analyse and improve scalability?
  - One or more larger test cases should build
  - ... and run (correctly) to completion within less than an hour
  - ... on larger numbers of processors/cores (in batch mode)
    - *Strong* scaling = fixed total problem size
      - speed-up expected with increasing numbers of processors
    - *Weak* scaling = constant problem size per process/thread
      - time not expected to change for increasing numbers or processors

# It happens

- You may experience problems with the HPC systems
  - They're continually being pushed to the limit (and beyond)
    - It's probably not your fault when they break

- You may discover bugs in your application(s)
  - That's presumably why you're here
    - ... and we're here to help
    - Think of identified bugs as opportunities for improvement, to make your application fitter and more competitive, as HPC systems continue to get larger and more complex

- You may encounter defects/deficiencies with the tools
  - We want to learn from you what should be improved
    - It's also nice to hear when they work well and are helpful

# 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

HELMHOLTZ ASSOCIATION

www.vi-hps.org

# Partners

## 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

## Technical University of Munich

- Chair for Computer Architecture

## University of Stuttgart

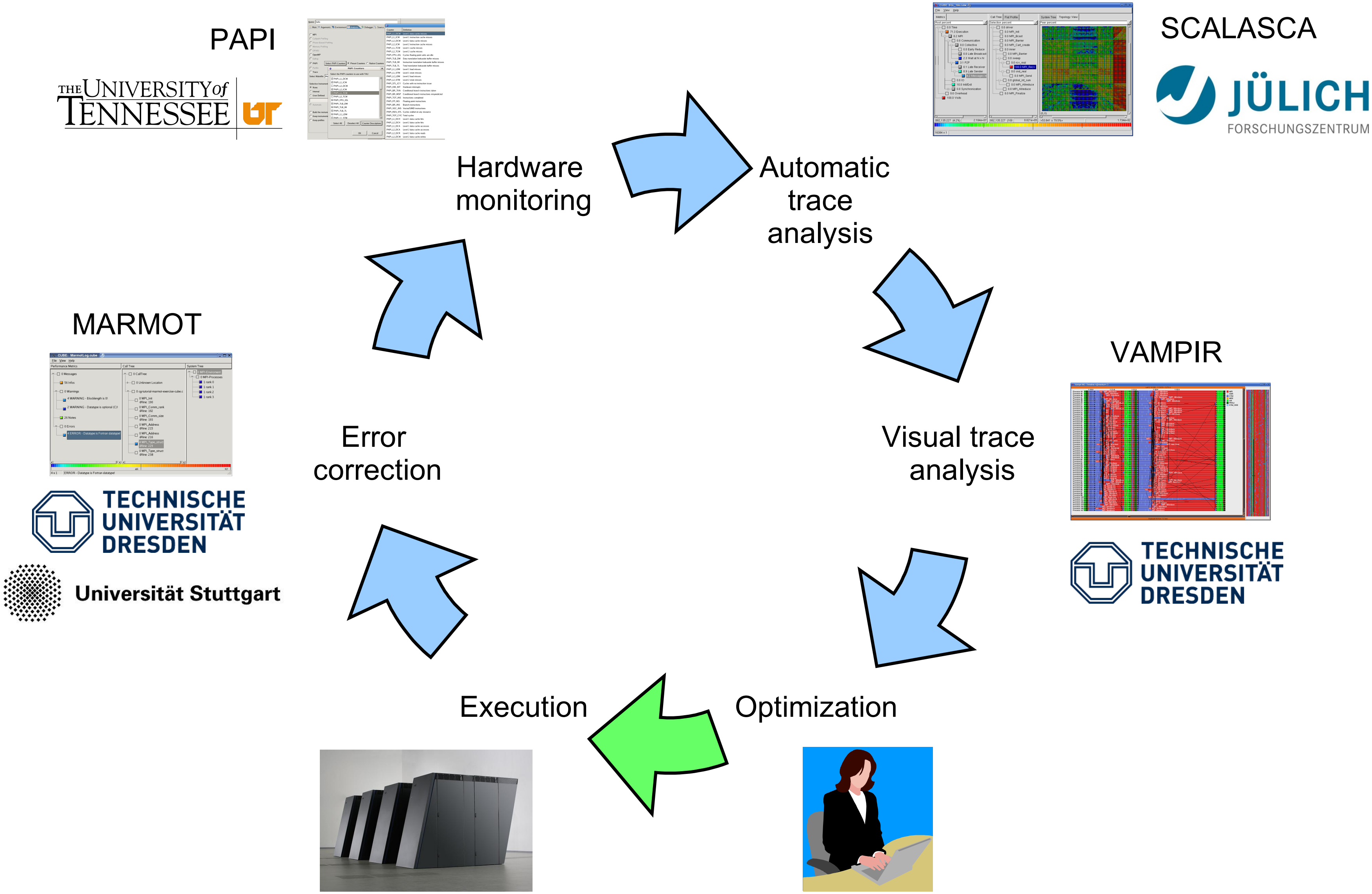
- High Performance Computing Centre

# Productivity tools

- Marmot
  - Free MPI correctness checking tool
- PAPI
  - Free library interfacing to hardware performance counters
- Scalasca
  - Open-source toolset for analysing the performance behaviour of parallel applications to automatically identify inefficiencies
- Vampir
  - Commercial framework and graphical analysis tool to display and analyse event traces
- VampirTrace
  - Open-source tool generating event traces for analysis and visualization by Vampir

[Tuning Workshop Live-DVD contains latest tools releases]

# Technologies and their integration



PAPI

THE UNIVERSITY of TENNESSEE UT

Hardware monitoring

Automatic trace analysis

SCALASCA

JÜLICH FORSCHUNGSZENTRUM

VAMPIR

Visual trace analysis

TECHNISCHE UNIVERSITÄT DRESDEN

Optimization

Execution

Error correction

MARMOT

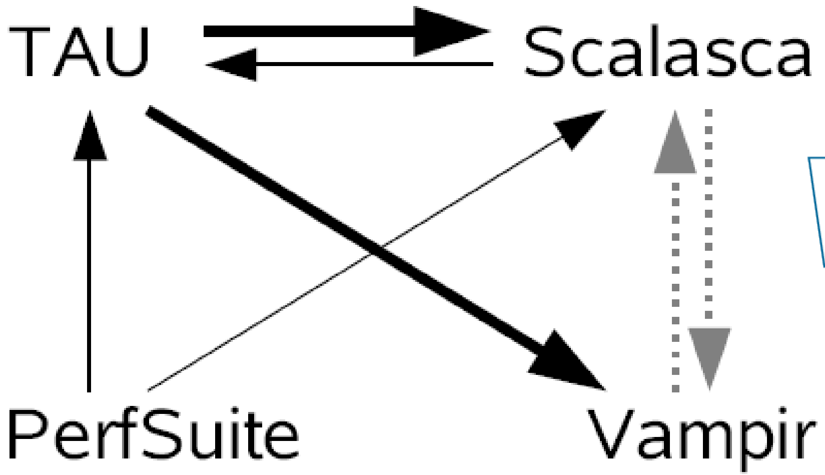
TECHNISCHE UNIVERSITÄT DRESDEN

Universität Stuttgart

# VI-HPS component technologies

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

# POINT/VI-HPS tool interoperability



- PerfSuite can generate reports in CUBE format
- TAU can use Scalasca & VampirTrace measurement libs and can present reports in PerfSuite & CUBE formats
- TAU & Vampir use OPARI to instrument OpenMP sources, and Scalasca can use TAU source instrumenter
- Scalasca & Vampir traces can be inter-converted

# 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)
- Joint POINT/VI-HPS Tutorial series
  - SC (11/08), ICCS (5/09), SC (11/09)
- Training with individual tools & platforms (e.g., BlueGene)

# Outline (tutorial, am)

- Monday 9th September
  - 08:30 (registration & notebook set-up)
  - 09:00 Welcome & Introduction to VI-HPS [Wylie, JSC]
    - Virtual Institute – High Productivity Supercomputing
    - Building and running the tutorial exercise NPB3.3-MPI/BT
  - 09:30 Parallel performance engineering [Malony, U.Oregon]
    - Introduction to performance analysis techniques and tools
  - 10:15 Marmot correctness checking tool [Hilbrich, TUD-ZIH]
    - Hands-on tutorial exercise with Marmot
  - 10:45 (break)
  - 11:15 TAU performance analysis toolset [Malony, U.Oregon]
    - Hands-on tutorial exercise with TAU
    - Case studies using TAU
  - 12:30 (lunch)

# Outline (tutorial, pm)

- Wednesday 9th September
  - 13:30 Scalasca performance analysis toolset [Geimer, JSC]
    - Scalable performance analysis of large-scale parallel applications
    - Hands-on tutorial exercises with Scalasca
    - Case studies using Scalasca
  - 14:45 (break)
  - 15:15 Introduction to the Vampir toolset [Doleschal, TUD-ZIH]
    - Vampir, VampirServer & VampirTrace
    - Hands-on tutorial exercises with Vampir
    - Case studies using Vampir
  - 16:30 Review and preparation for rest of workshop
    - Prepare participants' own codes for analysis on HLRN systems
    - Further exercises with tools
  - 17:30 (adjourn)

# Outline (workshop)

- Thursday 10th September
  - 09:00 Scalasca hands-on [Geimer & Wylie, JSC]
    - Advanced use of the Scalasca toolset
  - 12:30 (lunch)
  - 13:30 Vampir/Marmot hands-on [Doleschal & Hilbrich, ZIH]
    - Advanced used of Vampir & Marmot
  - 17:00 Review of day and schedule for tomorrow
  - 17:30 (adjourn)
- Friday 11th September
  - 09:00 Tools coaching & possible additional presentations
  - 12:00 (lunch)
  - 14:30 Review of workshop and participants' experiences
  - 15:00 (adjourn or continue to work to 17:30)

# POINT/VI-HPS Live-DVD

- Bootable Linux installation on DVD
- Includes everything needed to try out our tools on an x86-architecture notebook computer
  - GCC compiler suite (with OpenMP support), OpenMPI library
  - POINT tools: PAPI, PerfSuite, TAU
  - VI-HPS tools: MARMOT, Scalasca, VT/Vampir*
  - Other tools: Eclipse, 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 & Alan Morris