A Brief History of VI-HPS

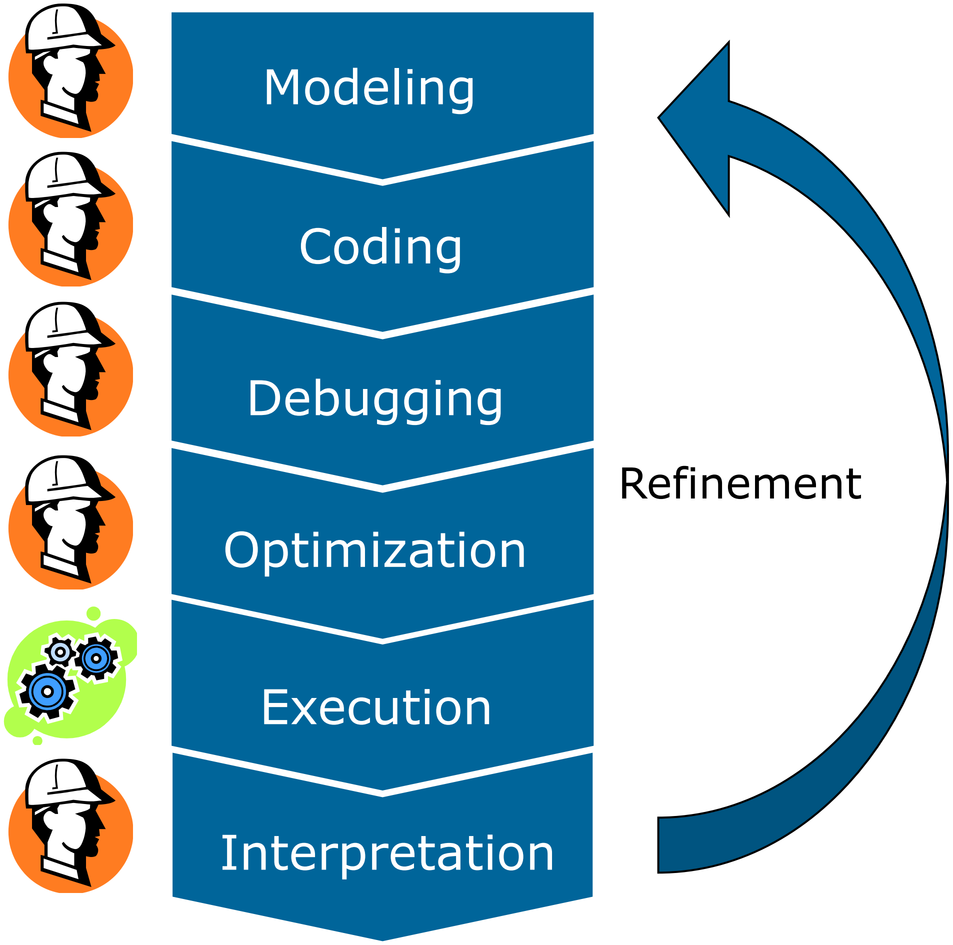
Felix Wolf
TU Darmstadt

About 12 years ago…

- Major investments in supercomputing hardware
  - Commodity clusters and massively parallel processors
  - Unprecedented levels of “peak” performance

- Little investment in software
  - Software taken for granted
  - Reliance on growing HPC open-source community
- Parallelism accessible via low-level interfaces
  - Hard to learn and use
  - Performance behavior hard to understand and predict

Increasing awareness that hardware is not everything…



- Focus shifts to **productivity**
  - Metric: time to solution
- Beyond hardware
  - People
  - Software environment

Examples of productivity-related initiatives

Since 1990s	DoD HPC Modernization Program
1998 – 2004	EU Working Group APART
2002 – 2010	DARPA High Productivity Computing Systems Program

- More to come such as
  - BMBF HPC program
  - DOE CScADS
  - EU FETHPC
  - PASC (CH)
  - JST CREST
  - DOE IDEAS

2006: Call for Helmholtz Virtual Institutes

- Funding instrument of the President of the Helmholtz Association of German Research Centers
  - Supplemented with up to €300K per year for three years
  - Main goal: expanding networks with universities
- Unite national and international competencies and resources
- Have their own management structure
- Qualify young researchers

HELMHOLTZ | ASSOCIATION

Founders

Jülich
Supercomputing Centre

Felix
Wolf

Bernd
Mohr

Thomas
Lippert

RWTH
Aachen

Christian
Bischof

Dieter
an Mey

TU
Dresden

Wolfgang
Nagel

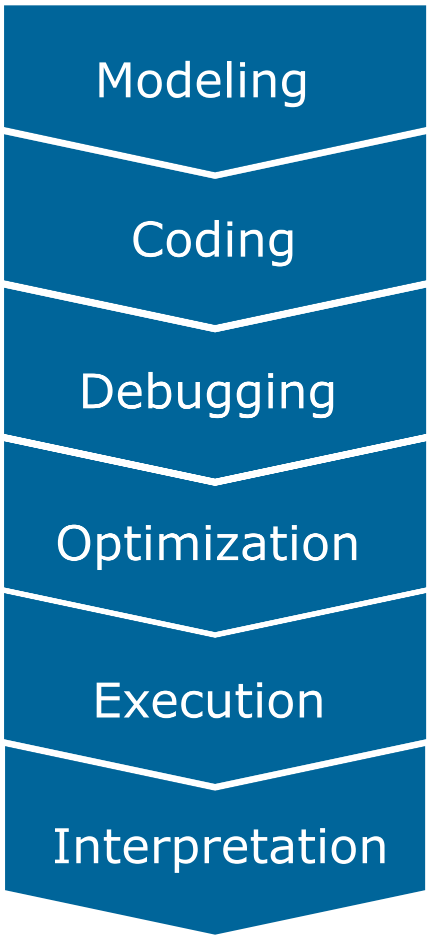
Matthias
Müller

University of
Tennessee

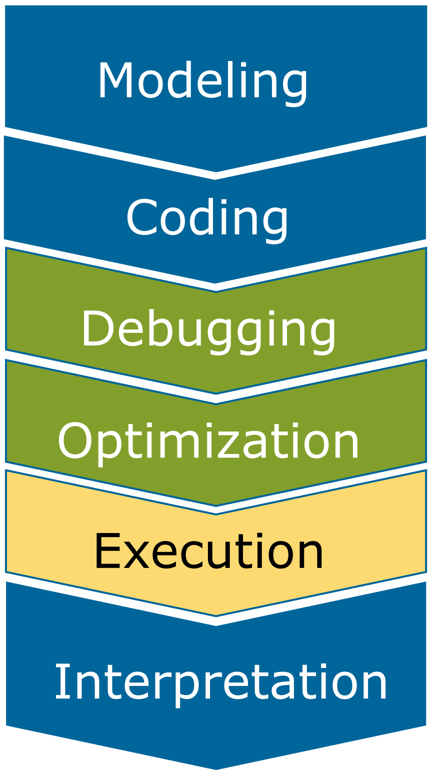
Jack
Dongarra

Shirley
Moore

Objectives

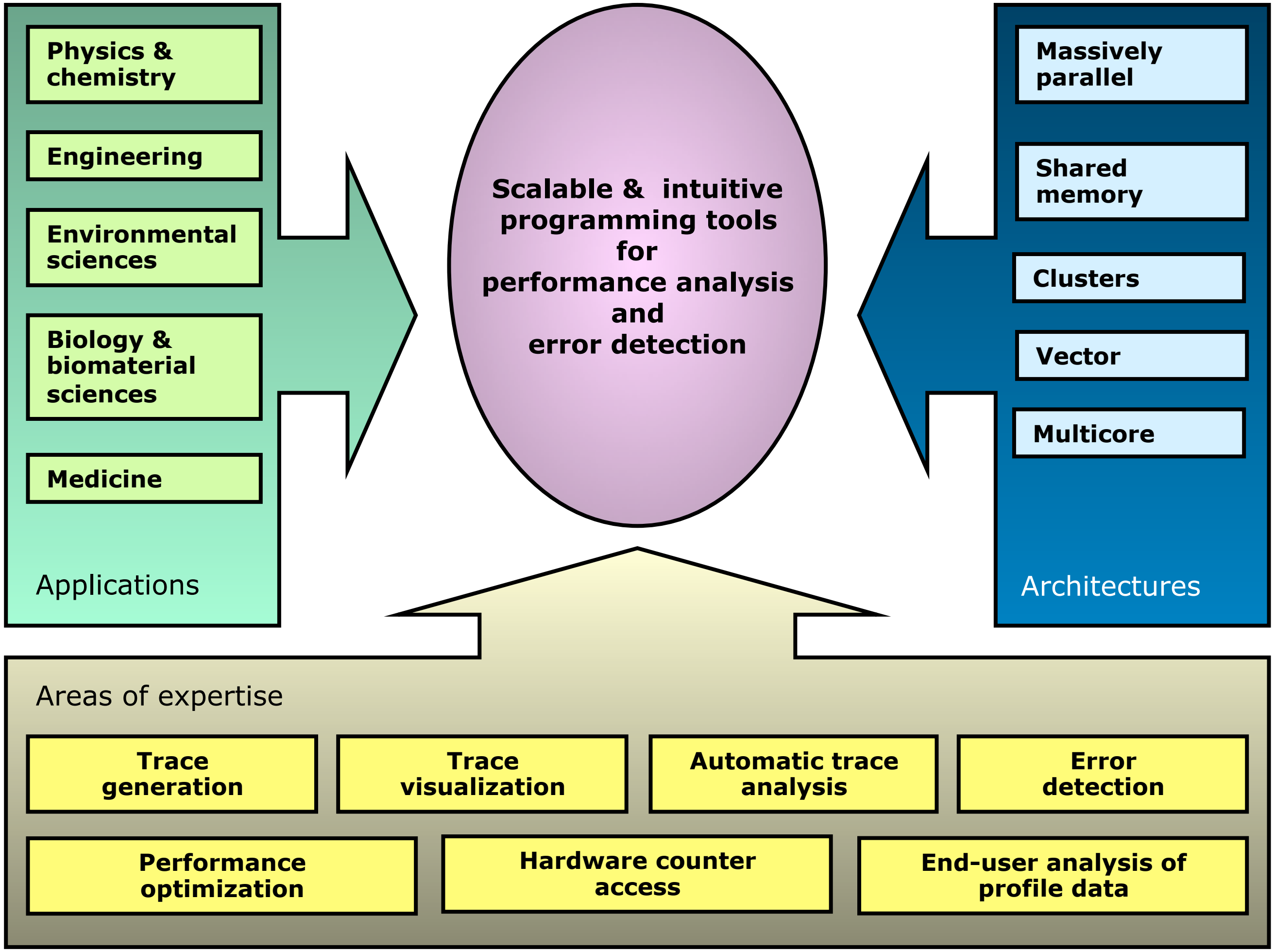

Modeling
Coding
Debugging
Optimization
Execution
Interpretation

Objectives



- Develop advanced HPC programming tools to improve the code quality
  - Remove errors
  - Increase performance
- Accelerate the development process
  - Make error detection and performance optimization not only better but also faster
- Tools must be easy to use
- Offer training & support

Proposal accepted end of 2006

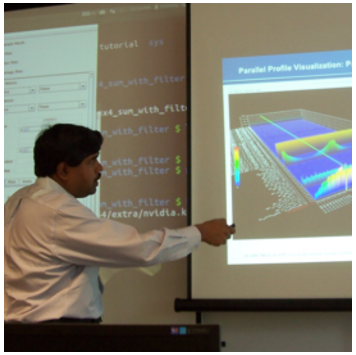


Inauguration – Jülich, July 4th 2007

Keynote **Doug Post**
The Opportunities and Challenges of Computational Science and Engineering
Chief Scientist, United States Department of Defense High Performance Computing Modernization Program

Mission components

Development of HPC programming tools



Training

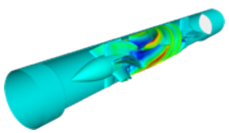
Integration of theses tools

Academic workshops

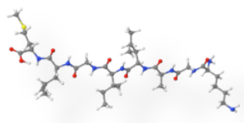
Tools

Single Node Performance	Parallel Performance		Debugging & Correctness
Callgrind	Allinea MAP	Paraver	Allinea DDT
	Allinea Performance Reports	Periscope	Archer
		Scalasca	MUST
		TAU	STAT
	Dimemas	Vampir	
	MAQAO		
	Open|SpeedShop		
Modeling	**Measurement**		**Visualization**
Extra-P	PAPI		Cube
	Score-P		

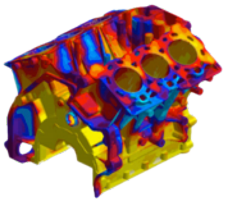
Performance optimization examples



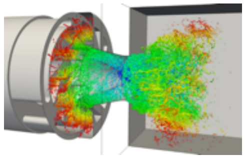
XNS CFD
RWTH Aachen
4-5x faster



QMC=CHEM Quantum chemistry
IRSAMC
> 3x faster

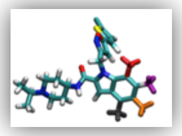


MAGMAfill CFD
MAGMASOFT® GmbH
25% faster

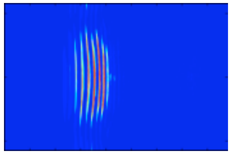


Yales2 CFD
CORIA
up to 2,8x faster

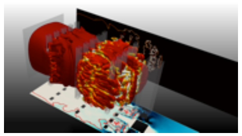
INDEED FEM
GNS mbH
30-40% faster



Polaris - molecular dynamics
CEA
50-70% faster



Illumination - particle simulation
Queen's University
2x faster



AVBP CFD
CERFACS
~ 10% faster

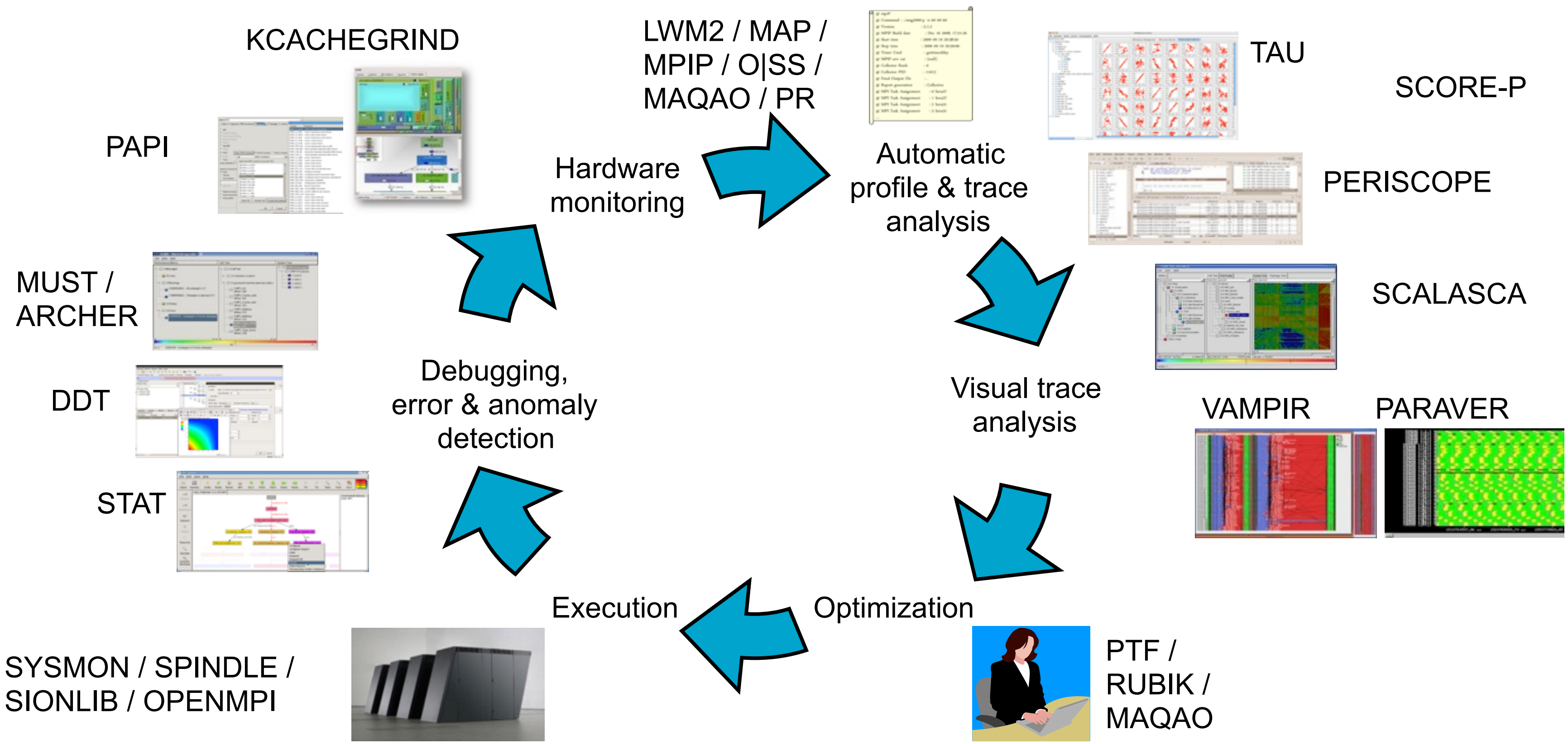
R&D projects

… with at least 2 partners from VI-HPS

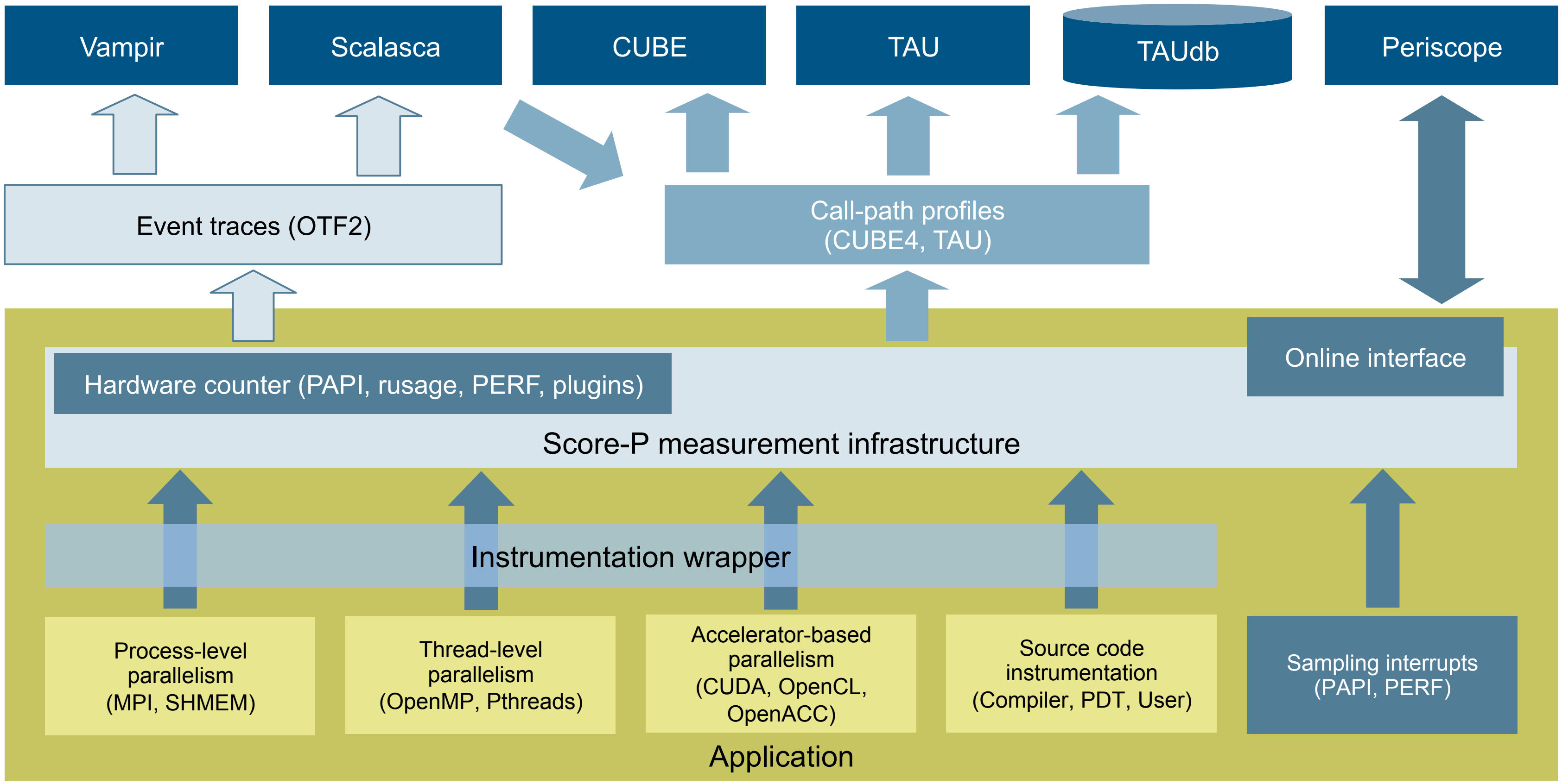
Project	Agency	Time
SILC	BMBF	2009 - 2011
PRIMA	DOE	2009 - 2013
TEXT	EU	2010 - 2012
H4H	ITEA	2010 - 2013
HOPSA	EU	2011 - 2013
ECS	G8	2011 - 2014
LMAC	BMBF	2011 - 2014

Project	Agency	Time
ELP	BMBF	2013 – 2016
Catwalk	DFG	2013 - 2017
Score-E	BMBF	2013 - 2016
PRIMA-X	DOE	2014 - 2017
POP	EU	2015 - 2018
ProPE	DFG	2017 - 2019

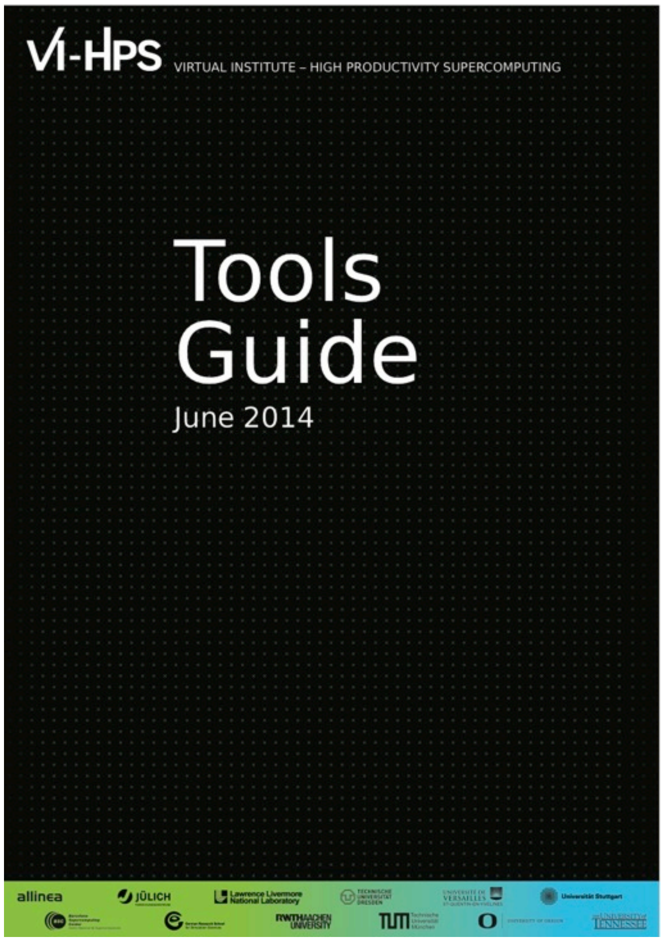
Technologies and their integration

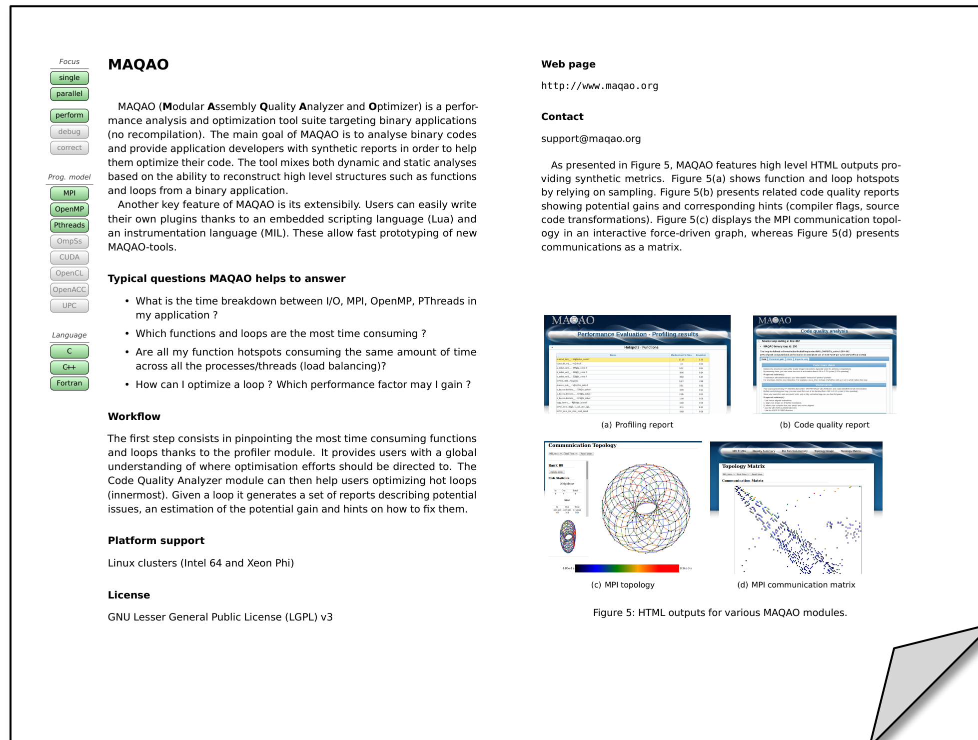


Score-P


Vampir
Scalasca
CUBE
TAU
TAUdb
Periscope
Event traces (OTF2)
Call-path profiles (CUBE4, TAU)
Hardware counter (PAPI, rusage, PERF, plugins)
Online interface
Score-P measurement infrastructure
Instrumentation wrapper
Process-level parallelism (MPI, SHMEM)
Thread-level parallelism (OpenMP, Pthreads)
Accelerator-based parallelism (CUDA, OpenCL, OpenACC)
Source code instrumentation (Compiler, PDT, User)
Sampling interrupts (PAPI, PERF)
Application

Tools guide – first edition in June 2014





Focus: single, parallel, perform, debug, correct

Prog. model: MPI, OpenMP, Pthreads, OmpSs, CUDA, OpenCL, OpenACC, UPC

Language: C, C++, Fortran

MAQAO

MAQAO (**M**odular **A**ssembly **Q**uality **A**nalyzer and **O**ptimizer) is a performance analysis and optimization tool suite targeting binary applications (no recompilation). The main goal of MAQAO is to analyse binary codes and provide application developers with synthetic reports in order to help them optimize their code. The tool mixes both dynamic and static analyses based on the ability to reconstruct high level structures such as functions and loops from a binary application.

Another key feature of MAQAO is its extensibily. Users can easily write their own plugins thanks to an embedded scripting language (Lua) and an instrumentation language (MIL). These allow fast prototyping of new MAQAO-tools.

Typical questions MAQAO helps to answer

- What is the time breakdown between I/O, MPI, OpenMP, PThreads in my application ?
- Which functions and loops are the most time consuming ?
- Are all my function hotspots consuming the same amount of time across all the processes/threads (load balancing)?
- How can I optimize a loop ? Which performance factor may I gain ?

Workflow

The first step consists in pinpointing the most time consuming functions and loops thanks to the profiler module. It provides users with a global understanding of where optimisation efforts should be directed to. The Code Quality Analyzer module can then help users optimizing hot loops (innermost). Given a loop it generates a set of reports describing potential issues, an estimation of the potential gain and hints on how to fix them.

Platform support

Linux clusters (Intel 64 and Xeon Phi)

License

GNU Lesser General Public License (LGPL) v3

Web page

http://www.maqao.org

Contact

support@maqao.org

As presented in Figure 5, MAQAO features high level HTML outputs providing synthetic metrics. Figure 5(a) shows function and loop hotspots by relying on sampling. Figure 5(b) presents related code quality reports showing potential gains and corresponding hints (compiler flags, source code transformations). Figure 5(c) displays the MPI communication topology in an interactive force-driven graph, whereas Figure 5(d) presents communications as a matrix.

(a) Profiling report
(b) Code quality report
(c) MPI topology
(d) MPI communication matrix

Figure 5: HTML outputs for various MAQAO modules.

Membership

Year	Member
2007	FZ Jülich RWTH Aachen TU Dresden University of Tennessee
2008	University of Stuttgart
2009	TU München
2010	GRS Aachen University of Oregon
2011	University of Versailles Saint-Quentin
2012	Barcelona Supercomputing Center Lawrence Livermore National Laboratory
2014	Allinea (today part of ARM)
2015	TU Darmstadt (replaces GRS Aachen)

4

12

Bylaws – adopted in 2013

Virtual Institute – High Productivity Supercomputing

Bylaws

October 17th, 2013

1. Goals and Activities

1.1 The Virtual Institute – High Productivity Supercomputing (VI-HPS) is an international association of programming-tool builders for high-performance computing (HPC). Its goal is to improve the quality and accelerate the development process of complex simulation codes in science and engineering that are being designed to run on highly parallel computer systems.

1.2. To fulfill its goal, VI-HPS pursues the following activities:

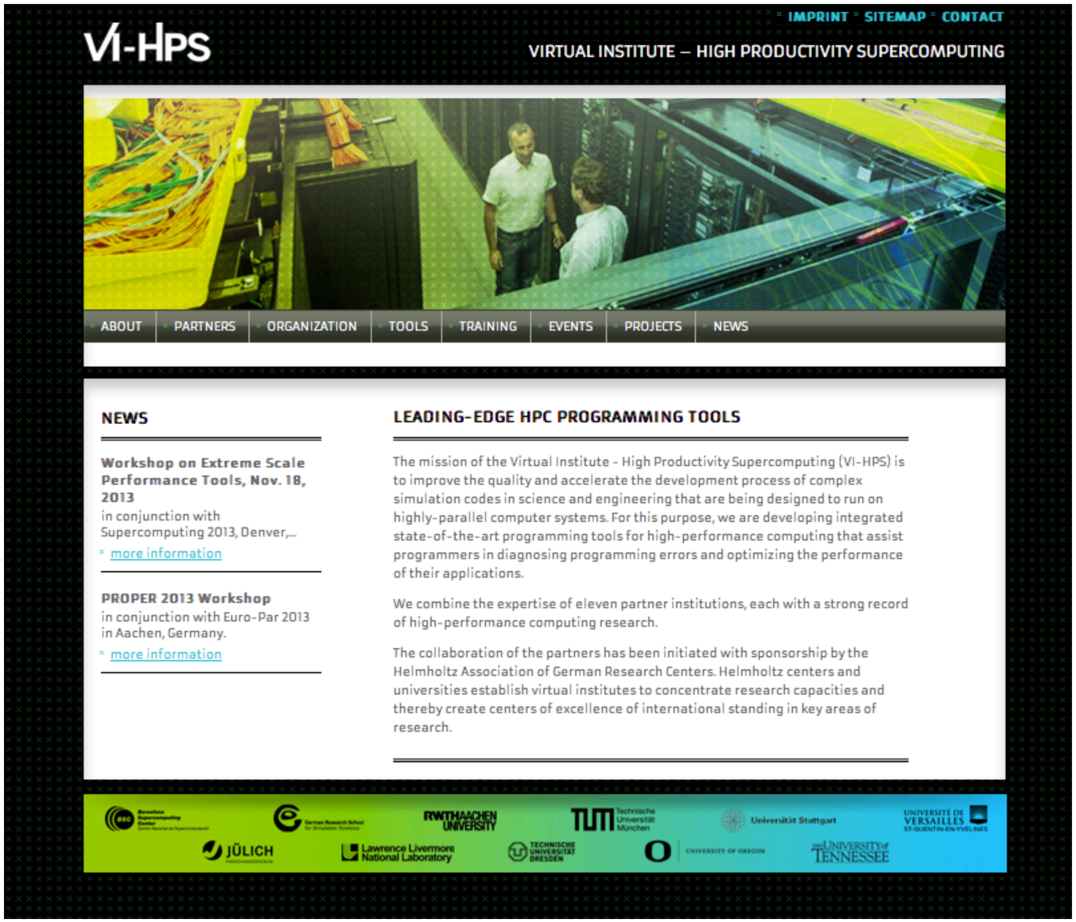
1.2.1. Development of portable programming tools that assist programmers in diagnosing programming errors and optimizing the performance of their applications.

1.2.2. Integration of these tools.

1.2.3. Organization of training events designed to teach the application of these tools.

1.2.4. Organization of academic workshops to facilitate the exchange of ideas on tool development and to promote young scientists.

New website online in 2013



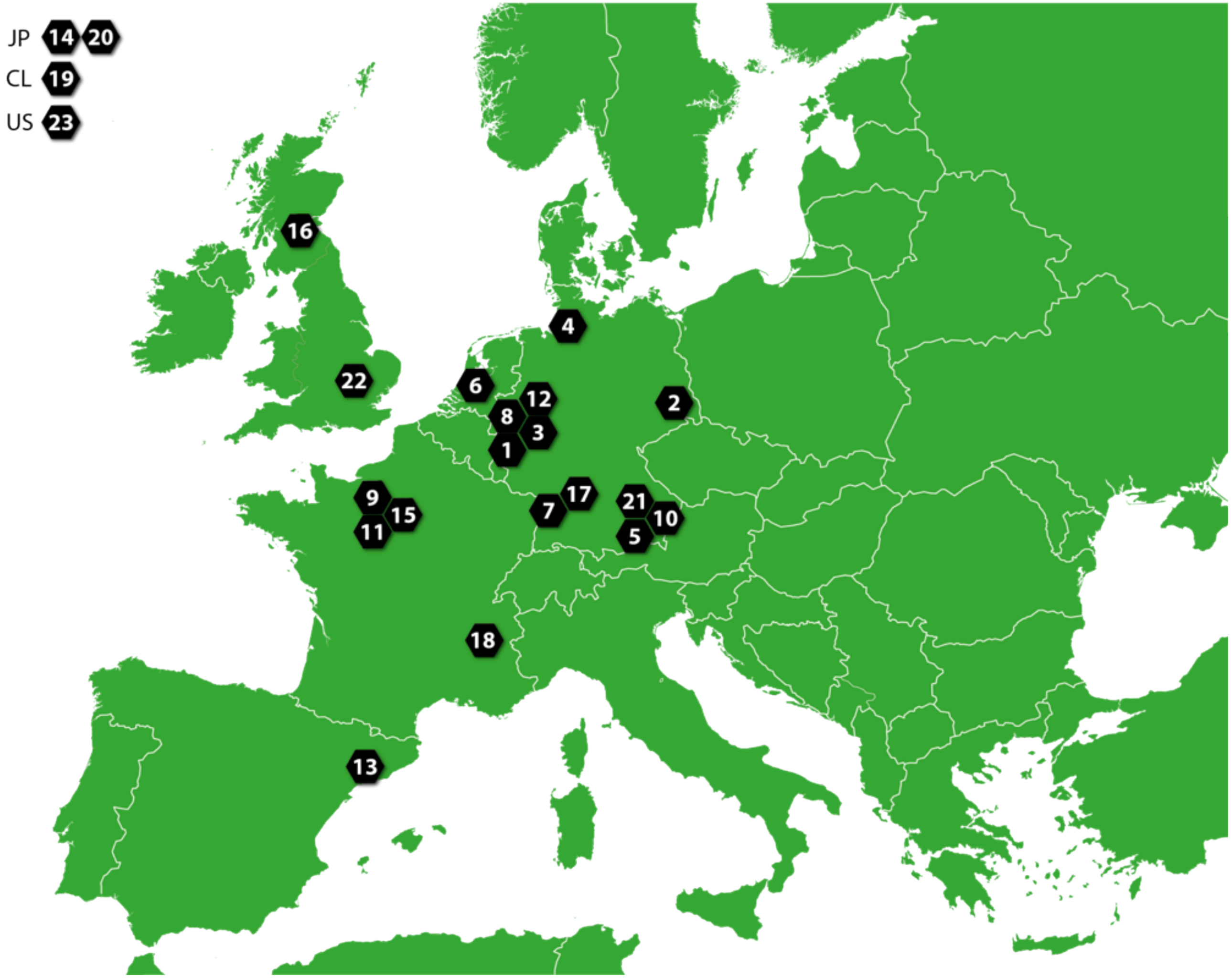
www.vi-hps.org

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 & analyze/tune your own code(s)!

- VI-HPS Tuning Workshop series
  - Multi-day format, (local) HPC system(s), exercises + participant applications
- VI-HPS Tutorials
  - Single-day, notebook computer + Linux ISO/OVA, exercises only
  - Half-day, presentations with live demonstrations only

VI-HPS Tuning Workshop series



- 25 workshops
- 11 as part of PRACE
- 19 hosts in 9 countries
- Over 512 participants
- Contributions from all partners

Further training events

- 18 dedicated external training events using local HPC systems
  - CSCS, DiRAC, DKRZ, DLR, ECP, ENSIMAG, ETHZ, IT4I, KAUST, MSU, NCAR, NERSC
  - Over 243 participants, generally 2 to 3 days with hands-on work with own application code
- Numerous conf. tutorials and contributions to seasonal schools
  - Often with hands-on exercises using Live-ISO/OVA or remote HPC
  - ISC-HPC, SC, XSEDE, Cluster, EuroMPI, EuroPar, ICCS, IHPCSS, PRACE

Academic workshops

Target audience includes many young researchers (e.g., PhD students)

PROPER

- @Euro-Par (2008 - 2014)
- Focus on productivity and performance

ESPT

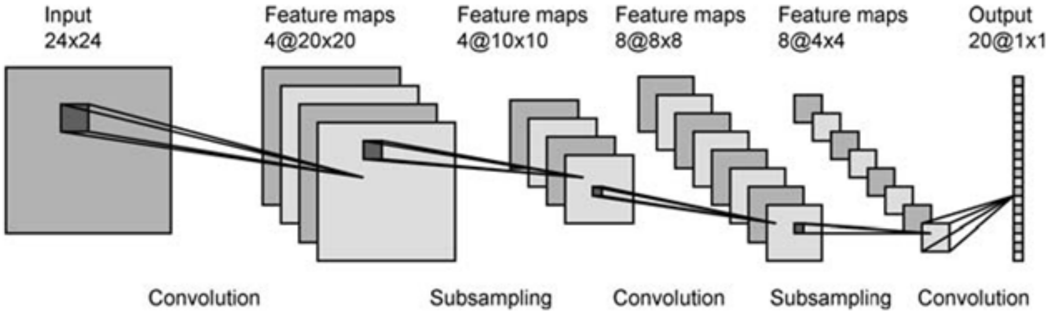
- @SC (since 2012)
- Focus on extreme-scale programming tools

Sponsor of Euro-Par 2013 conference

Outlook

Challenges

- Growing scale and heterogeneity
- New classes of applications
  - Big Data
  - Deep Learning



Thank You