The Hessian competence center for high performance computing (www.hpc-hessen.de)

Christian Bischof
Spokesperson of the Directorate of HPC-Hessen

University Computing Center (HRZ)
Institute for Scientific Computing
Technische Universität Darmstadt

Rationale for a HPC Competence Center (1)

Software is essential

- for competitiveness in science
- for efficient use of computing resources, i.e.
  - The code performs well (% of peak performance)
  - The system is energy-efficient
  - The code is energy-efficient (vgl. Talk by Armin Jäger at ENA-HPC Workshop at 16:30 in room Megabyte: Impact of compilers on energy efficiency of HPCG benchmark)

"In the context of engineering sciences, many TOP500 systems achieve only a fraction of their peak computation capability."
Harlacher et.al. DOI 10.1007/978-3-642-23869-7 22

Rationale for a HPC Competence Center (2)

Software development for scientific simulations

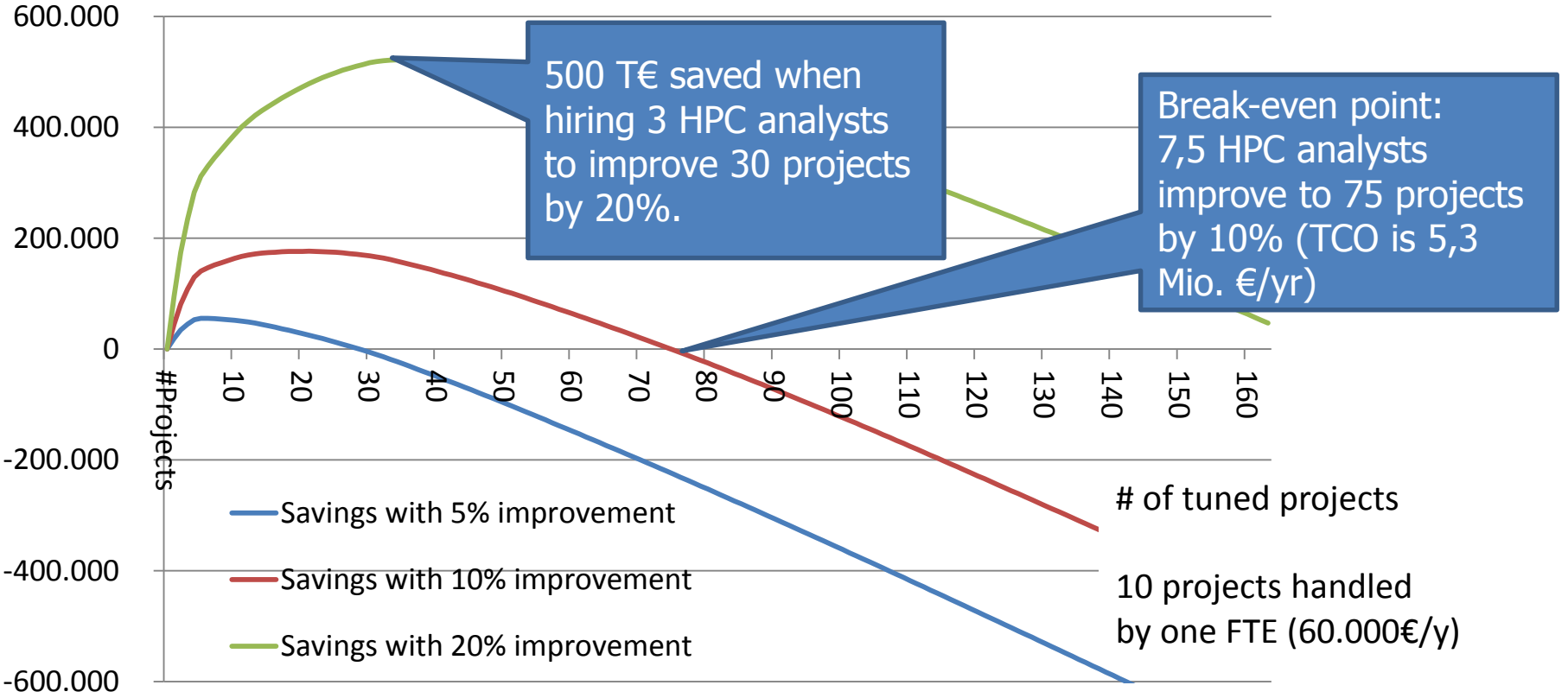
- Lead developers are usually domain experts, not HPC experts.
- In a research context often frequent personnel turnover.

Different kinds of users

- Black box user
- Source code augmentor (e.g. user-defined subroutine)
- Source code developer

Performance is usually of a secondary concern, in particular HPC is usually „free" in academia w.r.t. invest and operations.

Brainware for HPC pays off!!



Brainware for Green HPC, C. Bischof, D. an Mey, C. Iwainsky, Int. Conf. on Energy-Aware High-Performance Computing (ENA-HPC 2011), Computer Science – Research and Development, 27(4):227-233, 2012.

Hessian competence center for high performance computing (HKHLR):

brain ware for science

- A distributed competence center with staff at each site, but a common place to turn to.
- Educational offers to teach competences needed for HPC
- Advice on efficient use of our systems and efficient programming
- Outreach activities

Funding

- HKHLR has been funded by the Hessen State Ministry for Higher Education, Research, and the Arts.
  - Review by German Science Foundation (DFG) in June 2013
  - First round funding 7/13-3/16
  - Second review by DFG in July 2015
  - Second round funding 7/16-12/20, ~ 600 T€/yr.
  - Evaluation scheduled for end of 2018.
  - Universities committed to 25% and 50% co-funding in 2019 and 2020.

- 7 FTE scientific staff (2 each in DA/F, 1 each in GI/MR/KS), and 1 FTE for managing directorate.

Engagement with users

- Engage with users early on: Each month intro sessions in DA and F, on demand at other sites.
- Earn their trust by providing a reliable place to turn to.

Period 3/14-8/16 Location	User Queries	Consulting Protocols
Darmstadt	1132	319
Frankfurt	1096	221
Gießen	148	91
Kassel	90	80
Marburg	316	58
Total	**2782**	**769**

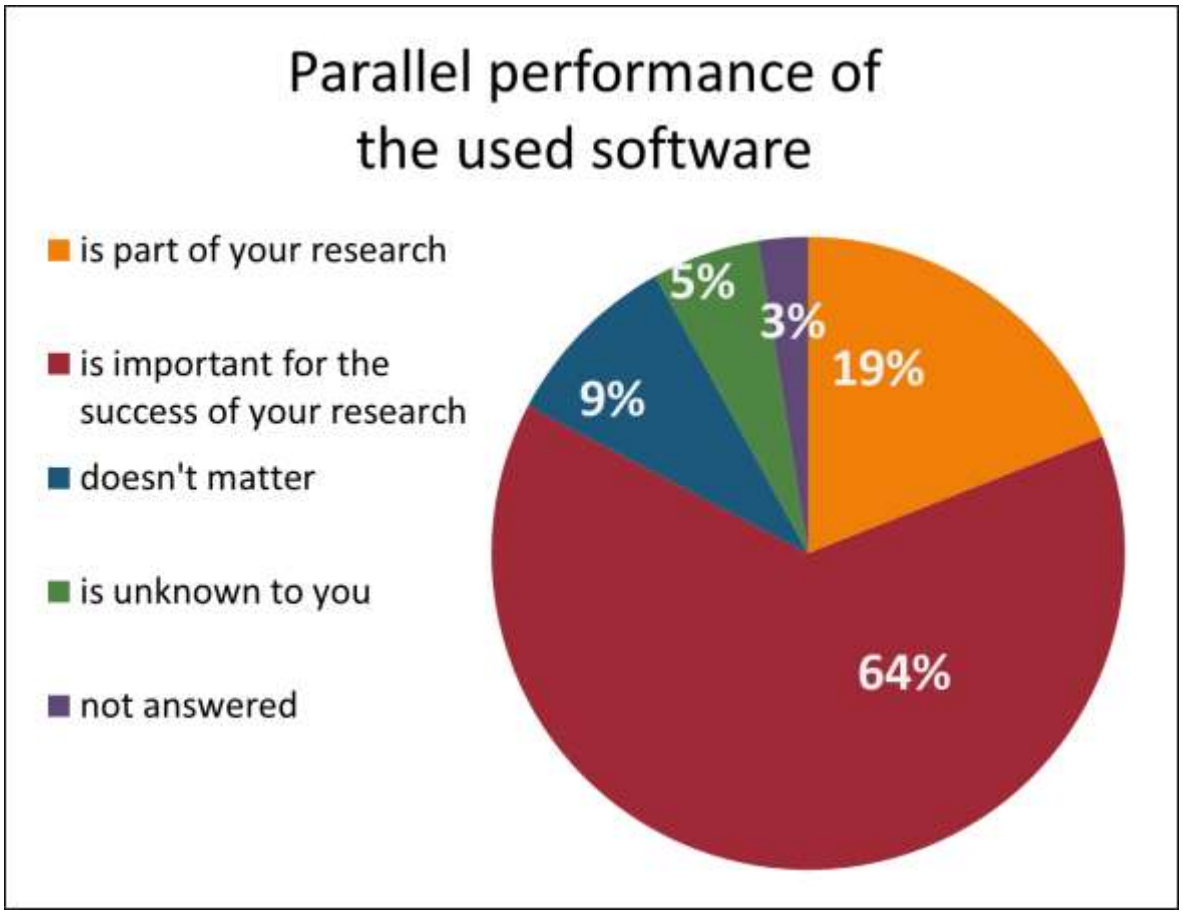
Empowering the HPC Users: The High-Performance Computing in Hesse (HiPerCH) Workshop Series

- For the users, it is very convenient if HKHLR staff writes their scripts, parallelizes and optimizes their code.
- This approach does not scale, there is not enough staff.
- This approach does not have a lasting impact, the changes by HKHLR staff are much less likely to be reflected in the main development branch of the codes.
- HiPerCH provides training on:
  - Linux and Software development tools
  - Vectorization, CUDA, OpenCL, OpenMP, MPI
  - Debugging and profiling tools (Funding for Totalview)

HiPerCH over the years

	HiPerCH 2 Darmstadt Sept 14	HiPerCH 3 Gießen March 15	HiPerCH 4 Darmstadt Sept 15	HiPerCH 5 Kassel Feb 16	HiPerCH 6 Darmstadt Sept 16	HiPerCH 7 Frankfurt March 17
Total	**49**	**25**	**49**	**36**	**56**	**61**
HKHLR	6	4	9	5	9	6
TU Darmstadt	19	4	18	7	23	11
Goethe-Uni Frankfurt	7	3	7	7	7	31
JLU Gießen	3	7	4	1	3	1
Uni Kassel	2	1		14	1	2
Philipps Uni Marburg	2	3		1	4	
FRA-UAS	1	1				
JGU Mainz			3			4
Other extern	3		4		2	4
Speakers and Aides	6	2	4	1	7	2

Yearly online survey (2016)

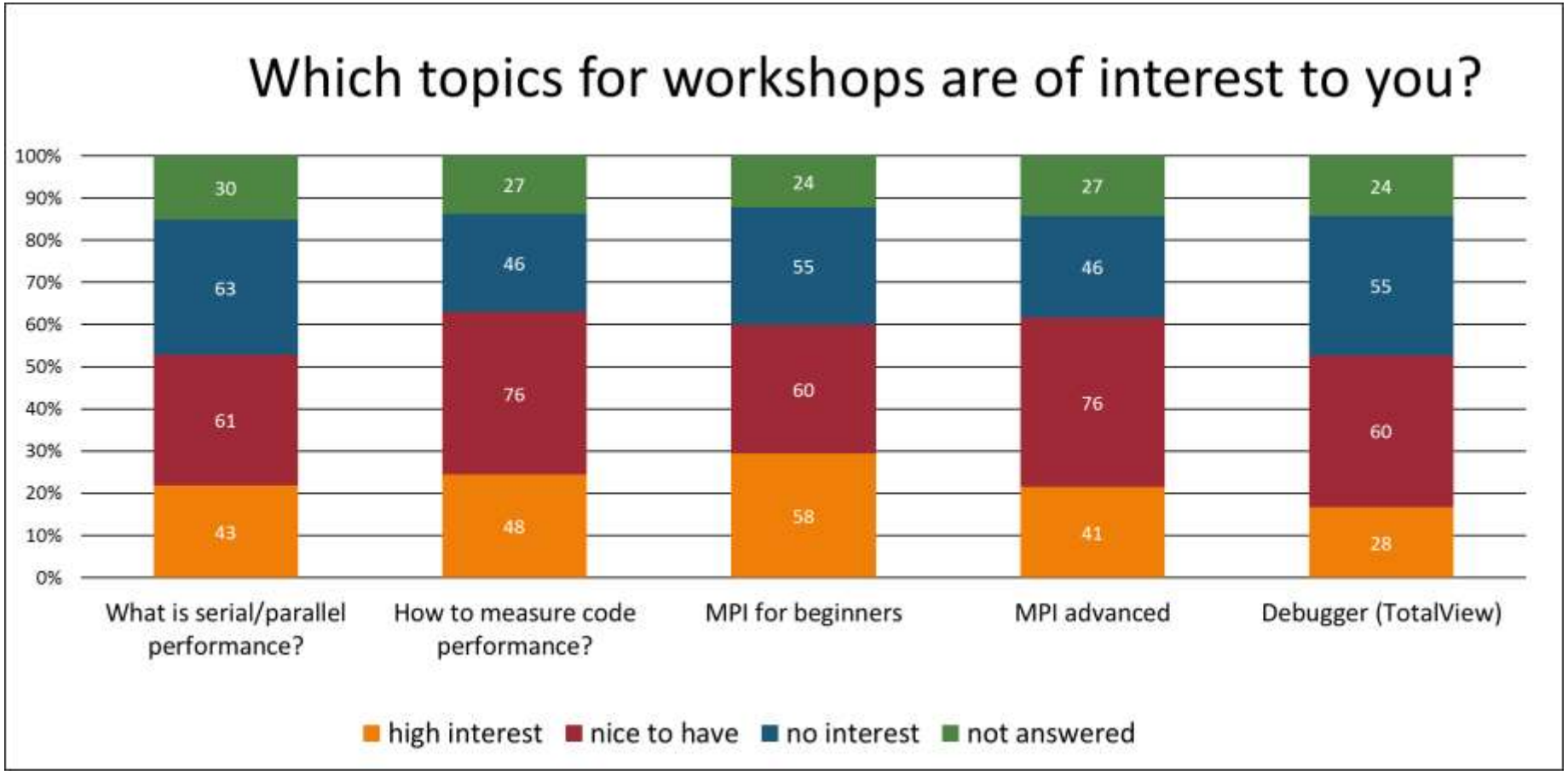


This is the third survey:

Year	Participants
2014	210
2015	230
2016	240

Little change in topical emphasis over the years

Online Survey 2016



HiPerCH6

Agenda

HiPerCH 6 includes four independent modules. Special attention should be paid to module 4, the „introduction to the TotalView debugger": HKHLR organized a Hessian-wide license (funded by HMWK), which can be used free of charge from all Hessian universities.

Tuesday, Sept. 27

10:30 - 18:15
S4|10 ROOM 1 (DOLIVOSTRASSE 15)

Modul 1: Brainware for Science

(Talks, discussions, networking)

- 10:00 - 10:30 REGISTRATION & COFFEE
- 10:30 - 12:00
  - The story continues: HKHLR - Hessian Competence Center for High Performance Computing (C. HEILIGER)
  - The Hessian Hardware infrastructure for HPC viewed from a German perspective (D. STERNEL)
  - Application procedure for computing time (A. WOLF, D. STERNEL)
- 12:00 - 13:00 LUNCH
  SNACKS AND BEVERAGE SERVED
- 13:00 - 15:00
  - Presentation of HPC projects on Hessian clusters
- 15:00 - 15:30 COFFEE BREAK
- 15:30 - 17:00
  - Recent trends in scientific HPC - methods and hardware (C. IWAINSKY, A. WOLF)
- 17:00 - 18:00 GET-TOGETHER
  SNACKS AND BEVERAGE SERVED

ATTENDANCE FEE: 15 EURO

Wednesday, Sept. 28

08:30 - 18:00
S4|22 ROOM 5 (LANDWEHRSTRASSE 48A)

Thursday, Sept. 29

08:30 - 18:00

Module 2: Software Tools for Unix/Linux Systems

PROF. C. HASSE (TU BERGAKADEMIE FREIBERG) AND STAFF

This two-day course consists of lectures supplemented with practical exercises.

CONTENT:

- Introduction to Unix
- Shell, Bash Basics
- Regular Expressions
- Editor VIM
- Shell Scripting
- Sed, awk
- Compiler
- Make, debugging, profiling
- Documentation, visualization
- GIT

PREREQUISITES: For the tutorials all participants are kindly asked to use their own laptop (Linux or Max OS X). (Eduroam WLAN access is available).

ATTENDANCE FEE: 30 EURO, INCL. EVENING EVENT

Wednesday, Sept. 28

08:30 - 18:00
S4|22 ROOM 6 (LANDWEHRSTRASSE 48A)

Thursday, Sept. 29

08:30 - 18:00

Module 3: OpenACC

NVIDIA TRAINERS

This two-day course consists of lectures supplemented with practical exercises.

CONTENT:
The course gives an introduction to OpenACC. You can learn how to programm OpenACC and how to accelerate your code with OpenACC.

PREREQUISITES: Participants need basic programming skills on Linux/Unix systems with C++ and some experience in programming. For the excercises all participants are kindly asked to use their own laptops (Eduroam WLAN access is available).

ATTENDANCE FEE: 30 EURO, INCL. EVENING EVENT

Evening event of module 2 & 3

19:00 (WEDNESDAY)

Friday, Sept. 30

09:30 - 13:30
S4|22 ROOM 5 (LANDWEHRSTRASSE 48A)

A coffee break is included in the program and will be scheduled by the trainers. No lunch!

Module 4: Introduction to the TotalView Debugger

HKHLR STAFF

This course consists of lectures supplemented with practical exercises.

PREREQUISITES: For the tutorials all participants are kindly asked to use their own laptop (Eduroam WLAN access is available).

NO ATTENDANCE FEE

All modules can be booked spearately. Please be aware that module 3 should be only booked by persons with programming experience.

September 22, 2016

Hessisches Kompetenzzentrum für Hochleistungsrechnen

HESSEN

HiPerCH7

Previous event

This introductory course is especially recommended for new LOEWE-CSC and FUCHS cluster users.

Thursday, March 09

10:00 - 14:00

Module 0:
Cluster Computing Workshop

HKHLR STAFF

ROOM N 310 02.114
BUILDING N 310 (PHYSICS)
MAX-VON-LAUE-STRASSE 1, 60438 FRANKFURT

Introduction to LOEWE-CSC & FUCHS cluster

- Hardware resources
- File system
- Environments modules
- Partitions on the cluster
- Architecture of the partitions
- Batch system

NO ATTENDANCE FEE

Agenda

Monday - Wednesday, March 13-15

08:30 - 18:00

Module 1:
Parallelization with MPI and OpenMP

DR. R. RABENSEIFNER (HLRS) AND STAFF

ROOM N 100 0.12 A+B BCC BEILSTEIN COMPUTER CENTER
BUILDING N 100 (BIOZENTRUM)
MAX-VON-LAUE-STRASSE 9, 60438 FRANKFURT

The focus of this three-day workshop is on the programming models MPI and OpenMP. Hands-on sessions (in C and Fortran) will allow users to immediately test and understand the basic constructs of the Message Passing Interface (MPI) and the shared memory directives of OpenMP. Content Level: 70% for beginners, 30% advanced.

- Hardware architectures and parallel programming models
- Parallel programming with Message Passing Interface (MPI-3) and OpenMP
- Tools for performance optimization and parallel debugging
- Short introduction to Portable, Extensible Tool kit for Scientific Computation (PETSc)

Prerequisites: Unix / C or Fortran

ATTENDANCE FEE: 50,00 €
INCL. EVENING EVENT (MONDAY)

Evening event

Monday, March 13

18:30

Guided tour

Monday, March 13

20:00

Dinner

HiPerCH 7

Thursday, March 16

09:00 - 15:00

Module 2:
Introduction to the TotalView Debugger

HKHLR STAFF

ROOM N 100 1.14
BUILDING N 100 (BIOZENTRUM)
MAX-VON-LAUE-STRASSE 9, 60438 FRANKFURT

This course consists of lectures supplemented with practical exercises.

PREREQUISITES: For the tutorials all participants are kindly asked to use their own laptop (Eduroam WLAN access is available).

NO ATTENDANCE FEE

All modules can be booked separately.

Hessisches Kompetenzzentrum für Hochleistungsrechnen

HESSEN

Improving codes and scripts

- HKHLR staff provides tuning support for codes and scripts
  - To demonstrate the need and potential of performance tuning (→ „earning trust“)
  - To improve efficiency of use of our systems by focussing on the „power users“ and their codes.

- Following „best-practices“ to maximize return on time investment. (e.g. Enhancing Brainware Productivity through a Performance Tuning Workflow, C. Iwainsky, R. Altenfeld, D. an Mey, C. Bischof, in Euro-Par 2011: Parallel Processing Workshops, M. Alexander et al., Eds, LNCS 7156, pp. 1987-207, Springer Verlag, 2012.)

Tuning a Physics User-Code

- Premise: User inquires about unexpected performance behavior on Lichtenberg system in Darmstadt.
- Planned runtime of code approximately 2.8 mio. core-h
- Profile revealed substantial use of libmath function cpow

Time	Cumulative Time	Function Name
36,97 %	88,82 s	cpow
10,15 %	113,20 s	sindl
9,73 %	136,59 s	cosdl
7,34 %	154,22 s	get_qNN(qNN*,double*)
5,67 %	167,84 s	powl
5,16 %	180,24 s	atan2dl
2,62 %	186,53 s	f_NN_4th_5(double*,unsigned long, void*)

Code analysis (Part I):

- Widespread use of cpow.
- E.g. 82946 x cpow in a single file:

```
... *
q[2*4+1].qi[2]* ... *q[1*4+3].qi[2]*
(
  cpow(mpi,2)+
  cpow(q[0*4+2].qi[0]+q[1*4+3].qi[0],2)+
  cpow(q[0*4+2].qi[1]+q[1*4+3].qi[1],2)+
  cpow(q[0*4+2].qi[2]+q[1*4+3].qi[2],2)
)*...
```

- Solution: Text-replace cpow with self-written csquare

```
inline complex
csquare(complex val, int exp){
  if (exp == 2)
    return val*val;
  else
    return cpow(val, exp);
}
```

Code analysis (Part II): Bycatch

Analyzing the code revealed additional opportunity

```
for (mt1 = ...)
  for (mt2 = ...)
    ...
      for (mt8 = ...){
...
for (ms1 = ...)
  for (ms2 = ...)
    ...
      for (ms8 = ...) {
  int c1 =abs(ms1-1)*pow(2,3)+abs(mt1-1)*pow(2,2)+abs(ms6-1)*2+abs(mt6-1)*1;
  int cp1=abs(ms4-1)*pow(2,3)+abs(mt4-1)*pow(2,2)+abs(ms5-1)*2+abs(mt5-1)*1;
  ...
  int c4 =abs(ms4-1)*pow(2,3)+abs(mt4-1)*pow(2,2)+abs(ms5-1)*2+abs(mt5-1)*1;
  int cp4=abs(ms3-1)*pow(2,3)+abs(mt3-1)*pow(2,2)+abs(ms8-1)*2+abs(mt8-1)*1;
  sum+= NNContr(..., c1,cp1, ...) + NNContr( ..., c2, cp2, ...)
}}
```

Background:

- **pow returns a floating-point number**
- **floating-point casting required**
- **replacing**
  - **pow(2,3) with 8 and**
  - **pow(2,2) with 4**

  improved performance considerably

Tuning Impact

- Tuning results on small test data set:

Variant	Runtime/Speedup	
Original code	253.5 s	
Loop-body optimization	124.9 s	2x
Use of csquared	63.8 s	4x

- User applied the same mechanics to reach speedups up to 8x on production data sets.
- HKHLR-supported parallelization of the used numerical integration library (libcuba) is on the way.

Good scheduling of a Biophysics code

- Code RELION on Loewe-CSC in Frankfurt

```
SBATCH --partition=long
SBATCH --nodes=6
SBATCH --ntasks=128
SBATCH --cpus-per-task=1
SBATCH --mem-per-cpu=5800
SBATCH --ntasks-per-node=22
SBATCH --constraint=intel20
...
export OMP_NUM_THREADS=1
```

- User was unfamiliar with the SLURM batch environment.
- Careful choice of scheduling parameters at least halved the turnaround time of the jobs.

Code optimization of a plasma code

- Plasmapic code is highly parallel (up to 1000 cores) and dominant code on skylla cluster in Gießen.

- Profiling with Vampir and Intel vTune.
- Function inlining, memory layout changes and optimization of inner loops improved runtime by 49%.

- Better load balancing can improve performance by another 50%, code adaption in progress.

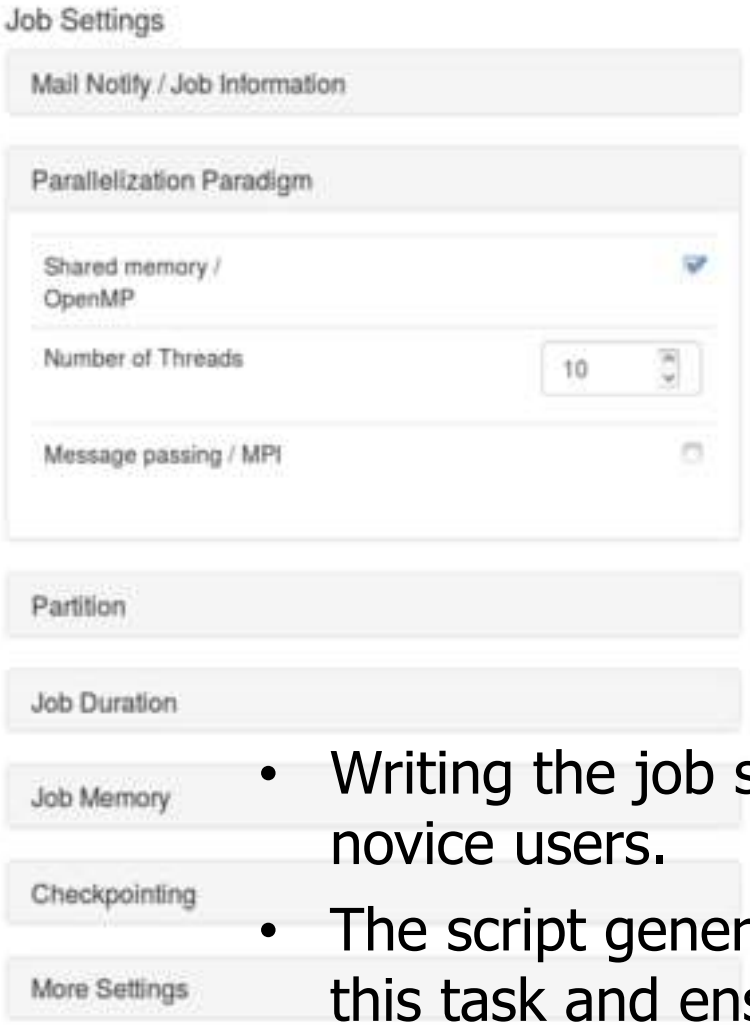
Improving scaling behavior

- Hybrid (OpenMP+MPI)-parallelized code does not scale on MaRC2 Cluster in Marburg.

- As it turned out, dynamic setting of thread number (omp_set_dynamic) did not work with the scheduler, so the code only ran on one thread on each node.

- Explicit setting of threads resulted in factor 2x speedup on 4 threads.

Script Generator



Submit-Script

```
#!/bin/bash
#########################################
##  Submit-Script Generator LC Kassel  ##
#########################################

####### Mail Notify / Job Name / Comment #######
#SBATCH --mail-user=user@uni-kassel.de
#SBATCH --mail-type=BEGIN,END
#SBATCH --job-name=myscript

####### Partition #######
#SBATCH --partition=exec

####### Ressources #######
#SBATCH --time=0-10:00:00
#SBATCH --mem-per-cpu=200

####### Parallelism #######
export OMP_NUM_THREADS=10
#SBATCH --cpus-per-task=10
```

- Writing the job scheduler script is a major hurdle for novice users.
- The script generator developed at Kassel structures this task and ensures a syntactically correct script.

Lessons learned

- There are many ways how users‘ codes cannot perform well.
- Users by and large are not performance-aware, and not familiar with HPC environments and tools.

HKHLR tries to provide structures to streamline support:

- Common set of base knowledge at all sites.
- Common knowledge base.
- Development of specialist knowledge.
- „Early intervention“ with our users.
- On-site personnel presence is essential.

Outreach

- HPC is a growing field, with growing demands.

- But why is HPC important for science and industry?
- And why does one need a competence center focussing on software?

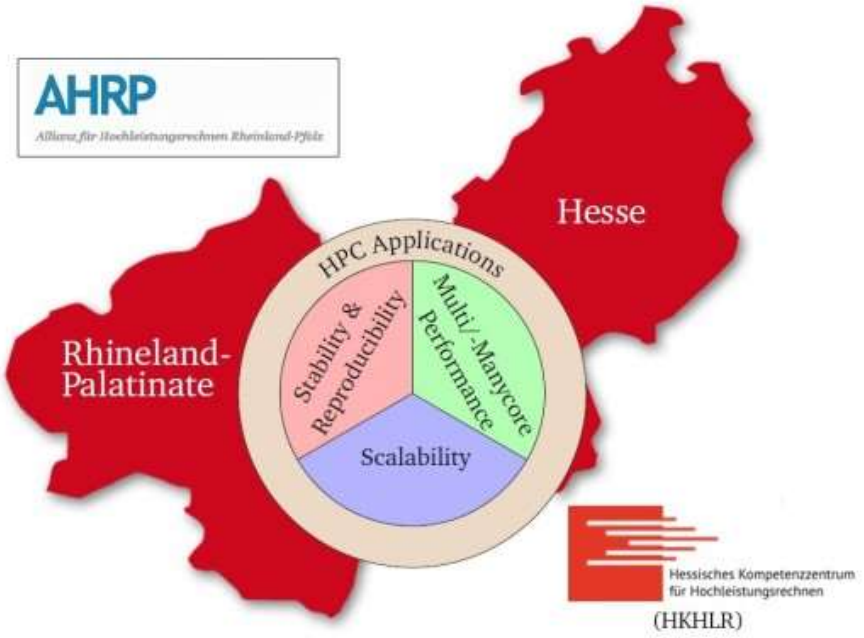
To answer these questions:

- Short video clips with researchers highlighting the importance of HPC for their work.
- HPC-Hessen at ISC


Competence Center for High Performance Computing in Hessen
HPC Hessen (www.hpc-hessen.de)
brain ware for science
Kompetenzzentrum
ISC High Performance
The HPC Event.
JUNE 18 - 22, 2017
FRANKFURT, GERMANY

Reaching out beyond Hesse: Enabling Performance Engineering in Hesse and Rhineland-Palatine

- Expand and deepen HPC support in areas where existing scientific expertise coincides with critical user needs.
- Bundle the distributed expertise for HPC support in Hesse and Rhineland-Palatine
- Funded by German Science Foundation



- Darmstadt: Bischof, Sternel, Wolf
- Frankfurt: Lindenstruth
- Kaiserslautern: Gauger
- Mainz: Brinkmann

www.hpc-hessen.de

apply

educate

brain ware for science

investigate

support