D1.1 Annual Report

Kai Himstedt   Julian Kunkel   Nathanael Hübbe

Work Package: WP1
Responsible Institution: DKRZ, RRZ
Date of Submission: June 2018

Abstract

This first annual report contains information about the project progress, status and next steps and is published on the project webpage. For the sake of completeness, the report gives a brief overview of the major project goals, the partners and scientific institutions involved in the project. The relevant information about the general organization of the project (e.g. about a delayed project start due to the difficulties in hiring of qualified scientific employees) is also included.

The results achieved in the six work packages are listed in more detail in the technical section of the report. The emphasis is on modeling HPC usage costs to calculate costs and statistics for SLURM jobs and on developing an HPC certification program to improve the education of HPC users. We have presented the results at various conferences and workshops. The related documents (handouts, presentation slides, project posters, deliverables, ...) are available for download on the project webpage.

Contents

Chapter 1

Introduction

The objectives of the Performance Conscious HPC (PeCoH) project are, firstly, to raise awareness and knowledge of users for performance engineering, i.e., to assist in identification and quantification of potential efficiency improvements in scientific codes and code usage. Secondly, the goal is to increase the coordination of performance engineering activities in Hamburg and to establish novel services to foster performance engineering that are then evaluated and implemented on participating data centers.

The PeCoH project started on March 1, 2017 and has a duration of three years. It was originally planned for one PhD candidate and one postdoctoral full time position. This first annual report contains information about the project's state of progress, status and next steps. It covers the period from March 1, 2017 to February 28, 2018 (i.e., M1 – M12) and is published on the project's webpage [PeC18].

1.1 Partners

The project partners consist of three research groups from the Universität Hamburg and three local computing centers are affiliated.

1.1.1 Computing Centers

German Climate Computing Center / Deutsches Klimarechenzentrum (DKRZ)

The German Climate Computing Center (DKRZ) is a national service facility and a major partner for climate research. With the hosted high-performance computers Mistral, long-term data storage and services, it provides the central research infrastructure for simulation-based climate science in Germany. One department of the DKRZ is dedicated to user consultancy. They support the users of the DKRZ in the effective use of its systems by providing general personal advice on the use of the systems, helping users to port applications to the HPC systems, and by offering conceptual guidance on parallelization and optimization strategies for user code, specifically with respect to the provided HPC system.

Regional Computing Center / Regionales Rechenzentrum der Universität Hamburg (RRZ)

The HPC team at RRZ operates a 396 node Linux cluster and more than 2 PByte of disk storage. The team is part of the consulting network of the North German Supercomputing Alliance (Höchstleistungsrechenzentrum Nord (HLRN) in German). HPC activities (locally and for HLRN) include user support, user education (in parallel programming and single-processor optimization) and benchmarking. The RRZ maintains and further develops BQCD

(Berlin quantum chromodynamics program) [ABS08] as one of the HLRN application benchmark codes [ABS18].

Computer Center of Hamburg University of Technology / RZ der Technischen Universität Hamburg (TUHH RZ)

The TUHH RZ has HPC consultants to support local users as well as users of the North German Supercomputing Alliance (HLRN). The TUHH RZ operates a 244 nodes Linux cluster. TUHH RZ and RRZ cooperate in sharing specialized parts of their HPC hardware.

1.1.2 Research Groups

Scientific Computing / Wissenschaftliches Rechnen at Universität Hamburg

The Scientific Computing group of Prof. Thomas Ludwig has a long history in parallel file system research but also investigates energy efficiency and cost-efficiency aspects and has developed tools for performance analysis. Prof. Ludwig is the director of German Climate Computing Center (DKRZ). The group is embedded into the German Climate Computing Center (DKRZ) since 2009 and, thus, also addresses important aspects for earth system scientists. With regards to teaching, the group offers interdisciplinary seminars and other courses about software engineering in science since 2012.

Scientific Visualization and Parallel Processing (SVPP) at Universität Hamburg

The Scientific Visualization group of Prof. Stephan Olbrich focuses on the development of methods for parallel data extraction and efficient rendering for volume and flow visualization of high-resolution, unsteady phenomena. Prof. Olbrich is the director of Regional Computing Center (RRZ). The group integrated their software into HPC applications, e.g., for climate research. Their activities are part of the cluster of excellence Integrated Climate System Analysis and Prediction (CliSAP) that focuses on post-processing data sets as well as on parallel data extraction at the runtime as part of the simulation.

Software Engineering and Construction Methods / Softwareentwicklungs- und -kontruktionsmethoden at Universität Hamburg

The Software Construction Methods group of Prof. Matthias Riebisch and his prior group at the Ilmenau University of Technology have experience in the adaptation and optimization of software architectures and software development processes. This includes, for example, measuring software quality properties, the forecast of properties after changes using impact analysis techniques, and methods for partitioning software applications for optimized execution on parallel computing platforms.

1.2 Starting the Project

The kickoff meeting for the PeCoH project took place on August 22, 2016. The agenda included the presentation of the individual work packages for Performance Engineering, Performance Awareness, HPC Certification, Tuning and Dissemination and the plan for 2017 with a subsequent discussion.

Originally, the project start was scheduled for January 1, 2017. We applied for funding of two full time positions for the duration of three years: A PhD candidate is responsible to conduct the mentioned research and helps to transfer them into production via a corresponding service infrastructure. A postdoctoral researcher is responsible to establish performance engineering and to implement the corresponding services.

The good labor market situation for computer scientists and IT experts made it difficult to fill the vacancies as planned. The postdoc position could finally be filled on March 1, 2017, and the project started at this date. The PhD candidate position has been divided among two persons: The first candidate started on July 1, 2017 (part time position – 24 months), and the second candidate started on January 1, 2018 (full time position – 12 months).

Chapter 2

Technical Report

The PeCoH project plan contains six work packages (WPs) and associated tasks in each WP. This section reports on the activities and the progress made in each WP.

2.1 WP1 Management

During the reporting period, the project's progress was regularly verified and compared to the expected status. In regular face-to-face meetings, conference calls and video conferences, the project status was determined and discussed with the partners in order to avoid arising risks like delay of deliverables and the overall project. Furthermore a series of working-level meetings were held for processing the work packages. Based on the situation caused by the late hiring and the split of positions, we made several minor adjustments to the project plan during the first year.

2.1.1 Project Management (Task 1.1)

We set up a Git repository at Universität Hamburg to host all artifacts created and related to the project. This collaborative workspace was provided in M2 and allows secure communication of results, deliverables, source code, meeting notes, etc.

2.1.2 Coordination between Data-Centers (Task 1.2)

The coordination is supported by the novel Hamburg regional HPC Competence Center (HHCC) that we bootstrapped as part of PeCoH. HHCC acts as a virtual organization and is represented by a corresponding website [HHC18a]. A first version of the website was established in M1 and is available online since M2. The idea behind HHCC is to bundle the know-how and to combine the strengths of the three compute centers in order to give their users a better support, particularly by education and by giving feedback. HHCC is also the basis to disseminate existing and new performance engineering concepts. We attended various meetings in the data centers that covered aspects of performance engineering and provided a basic level of coordination. In addition to these achievements we see further opportunities that will need to be realized in the future.

2.2 WP2 Performance Engineering Concepts

2.2.1 Identification of Suitable Concepts (Task 2.1)

A skeleton of the (future) deliverable has been created, where various ideas are documented. A first concept and classification matrix that aims to assess the different concepts has been created. Additionally, we identified essential performance engineering concepts like performance patterns and anti-patterns, and efficient algorithms and data structures already during the classification of HPC skills for establishing the HPC certification program (see also Section 2.4). These performance engineering concepts are also relevant for this task and will be deepened in the further progress of the project, i.e. in the second annual reporting period.

2.2.2 Benefit of Data Analytics (Task 2.2)

There were no activities planned and conducted in the reporting period.

2.2.3 Benefit of in-situ Visualization (Task 2.3)

There were no activities planned and conducted in the reporting period.

2.2.4 Compiler-assisted Development (Task 2.4)

Due to the delay in hiring qualified scientific staff, we have postponed this task to the second annual reporting period.

2.2.5 Code Co-Development (Task 2.5)

We started a first activity in this task to speed up an R program for analyzing satellite night images (see also Section 2.5.3, Use Case C) by exploiting the parallelization potential of sequential loops without data dependencies. The performance engineering know-how was successfully transferred to end users.

2.3 WP3 Performance Awareness, HPC Cost Model

Most work of this WP could be achieved during the reporting period. We developed cost models, we implemented them into the SLURM workload manager, and we explored their use in production environments. For a detailed description of the outcome refer to the published deliverables 3.1 & 3.3 [Hü18b]. Deliverable D3.2 consists of the source code [Hü18a].

2.3.1 Modeling Costs of Running Scientific Applications (Task 3.1)

Based on cost relevant inputs (run time of a job, cold data amount usage, cluster node type, power consumption, infrastructure costs, ...), we developed four distinct cost models, taking the Mistral cluster (the current supercomputer of the DKRZ) as an example.

To summarize results briefly: For the Simple Model, the initial costs of the Mistral cluster (40 million EUR) are simply divided by the total amount of node hours provided (3336 cluster nodes operating for about 5 years) resulting in 2 398 EUR per node per year. The remaining three models are more complex: The Extra Model takes additionally into account that some cluster nodes have extra hardware like GPU cards or additional main memory. The Full Model can be used to include the operating costs into the cost estimations and the Partitioned Model allows for a better separation of the CPU time and storage usage impacts.

In order to allow for automatic analysis of jobs according to the four different cost models above, we developed a simple model description format: Essentially, the text file with the description of a concrete cost model may contain several sets of node names, specified in the node list format of SLURM, and for each node set a list of cost rates can be defined (also for energy based rates on top of time based ones). In this way, the runtime costs of jobs can be estimated by analyzing their node lists and mapping the used nodes to the cost model description.

2.3.2 Reporting Costs of User Jobs (Task 3.2)

We have developed two tools to apply the four cost models to SLURM jobs. The first tool is implemented as a shell script (`job-cost-meter.sh`) supposed to be run from a job-epilogue script. The `job-cost-meter.sh` script reads in a cost model description and applies the related cost rates appropriately to the nodes effectively used for a job to estimate its costs.

The second tool is also implemented as a shell script (`batch-cost-analyzer.sh`) and supposed to be run from the command line for post mortem analyses of the accounting records SLURM has written to its database. The `batch-cost-analyzer.sh` takes job-selection parameters (e.g. a date and time range) on the command line and calculates the different cost estimates that are associated with the jobs selected, using the same cost model description format as the `job-cost-meter.sh` script. The data collected can be used to give detailed information for all jobs and to produce statistics for each cost rate defined in the model.

For a detailed description of modeling the cost refer to deliverable 3.1 & 3.3 [Hü18b].

2.3.3 Deploying Feedback Tools for User Jobs (Task 3.3)

There were no activities planned in the reporting period. Nevertheless, we explored the production use on Mistral. The tools are capable to be run in production, however, it requires the agreement of User Group and management. These discussions were started but lead to complex questions and we have not yet produced a decision on production roll-out.

2.3.4 Analyzing Data and Giving Feedback to Users (Task 3.4)

There were no activities planned in the reporting period.

2.4 WP4 HPC Certification Program

2.4.1 Classification of Competences (Task 4.1)

For the classification of the HPC competences, we implemented a meaningful tree of HPC skills. Skills are modeled as dependencies and may build upon one another. The tree acts as a database for the HPC certification program. The implementation is based on the Extensible Markup Language (XML) [W318a], and a corresponding XML Schema Definition (XSD) [W318b] assures consistency.

For a detailed description of the classification of HPC competences, refer to deliverable 4.1 [HHK+18] and our corresponding concept paper [HHKS18].

2.4.2 Development of the Certification Program (Task 4.2)

For the development of the certification program, we separated the certificate definition from the providing of content, similar to the concept of a high school graduation exam (Zentralabitur in German). A certificate is defined by bundling an appropriate set of skills, using the skill tree as a database. The learning material can be provided by different content providers, e.g. by different scientific institutions. Since the start of the project, we welcomed the collaboration with other scientific institutions to receive comments on the analysis and classification of HPC competences and their value for scientists. Additionally, we are hosting a mailing list for the HPC Certification Program [HHC18b].

For a detailed description of the classification of HPC competences refer to deliverable 4.1 [HHK+18] and our corresponding concept paper [HHKS18].

Since we saw an opportunity for sustaining the activities, we started to establish the HPC Cerfification Forum[1] – a new entity that curates and manages the skill tree and welcomes

[1] `https://hpc-certification.org`

international collaborators. It has the role of an institution that establishes accepted certificate definitions and corresponding exams. We submitted a proposal for a Birds-of-a-Feather session to the ISC-HPC conference to start the discussion with contributors. The proposal was not accepted, yet we will incorporate the comments of the referees, and we see a good chance that it will be accepted next year. PeCoH contributed the state of the certification tree to the Forum for an initial state.

2.4.3 Workshop Material (Task 4.3)

The creation of training material is a bit delayed. A reason is that we are in discussion with other performance engineering teams and international stakeholders to coordinate the creation of material as part of the HPC certification program. Our goal is to augment the existing material and ongoing efforts to produce valuable teaching material that is complementary and hence most beneficial to users.

As part of the certification program, we decided to separate the role of certification provider from the role of content provider for basic HPC skills. Meanwhile, we have produced initial workshop material for several basic HPC topics in the field of performance modeling (e.g. Moore's Law, Amdahl's Law, Speedup, Efficiency, and Scalability, ...), using the Markdown language as a common denominator and starting point in the content production workflow.

Markdown is an easy to use lightweight markup language, which is widely used for documentation purposes (e.g. on GitHub [Git18]). A plain text editor is already sufficient to create markdown files. Advanced editors like Atom [Ato18] can be used to get real-time representations of the rendered Markdown content. Markdown supports the required features to adequately teach HPC skills with regard to the presentation of the content (e.g. rendering of formulas, syntax highlighting of source code, tables, hyperlink support, including of images, etc.).

The content of each single skill is represented by a set of corresponding Markdown-files. The well-known Pandoc-Tool [Pan18] is used to convert these Markdown-files to various target formats like HTML, PDF, and TeX.

Programs based on the Extensible Stylesheet Language Transformations (XSLT) [W318c] are used in the content production workflow to generate Makefiles [Fel79] with corresponding Pandoc calls using the information from the skill tree. In this way we have fully automated the transformation process.

2.4.4 Online Tutorial (Task 4.4)

Using the results from tasks 4.1 and 4.2, we derived a workflow to produce online tutorials.

2.4.5 Online Examinations (Task 4.5)

There were no activities planned in the reporting period.

2.5 WP5 Tuning of Software Configurations

2.5.1 Help Desk (Task 5.1)

The help desk was set up as a ticketing system (see also below on Support / Help Desk in Section 2.6.1). It received only a few emails during the reporting period, hence we have to continue the marketing and help the data centers to work closer together (Task 1.2).

2.5.2 Determination of Tuning Possibilities (Task 5.2)

We approached the users and prioritized the activities to focus on the statistics package R. So far, three use cases for R-programs have been studied in order to describe the specific tuning information relevant for the R software package. Tuning ideas were determined by expert knowledge and the software documentation and implemented. The performance gains have also been documented (see Task 5.3) and users should be given feedback.

2.5.3 Setup of Realistic Use Cases (Task 5.3)

In tuning of standard software, it is less a matter of implementing better parallel algorithms, but rather to analyze the potential of high level tuning of an application from “outside”. High level tuning can be done by selecting particularly efficient libraries and by using a suitable combination of compiler and OpenMP/MPI environment, for example.

Use Case A

Experiments were performed based on the R Benchmark 2.5 test suite [Urb18]. The test suite consists mainly of a mix of matrix operations (e.g. cross product, eigenvalues, ...) and algorithmic parts (e.g. recursion, loops, ...). First results of the experiments showed that the selection of an efficient library like OpenBLAS or MKL lead to good speedups with respect to the use of the standard library. The speedup results for MKL were usually better than those achieved with OpenBLAS by a small margin. Using one core on a single cluster node, we achieved a speedup factor of ca. 4 for the configuration with the MKL library. There were additional experiments performed using the environment variable OMP_NUM_THREADS in order to exploit parallelism using several threads. But, using up to 16 cores on a single cluster node, hardly any additional speedup could be achieved (ca. 15% compared to a single core).

Use Case B

Experiments were performed taking the rlassoEffects-function [SCH18] in the context of regression analysis as a real-world example. For this use case, the parallelization was implemented by replacing sequential loops with parallel loops using appropriate R packages like doMPI, foreach, iterators, and Rmpi [Wes17]. On 4 cluster nodes, each using 16 cores, we achieved a speedup factor of 30.

Use Case C

For the experiments, we parallelized - in collaboration with the user - a program for analyzing satellite night images using a foreach() paradigm. In a sense, this can be regarded as co-development (see also Section 2.2.5). On 32 cluster nodes, each using 4 cores, we achieved a speedup factor of 126.

2.5.4 Benchmarking (Task 5.4)

In this task, first real benchmark experiments were performed for the three uses cases described above in order to verify that good tuning settings were found. These real measurements also allow us to report the effect of the tuning in success stories.

2.5.5 Documentation (Task 5.5)

There were no activities planned in the reporting period.

2.6 WP6 Dissemination

2.6.1 Web Presence of the Hamburg HPC Competence Center (Task 6.1)

In this task, the web presence of the HHCC was created. The implementation is based on the Content Management System (CMS) Fiona [Fio18] as it is used at the Universität Hamburg for the implementation of interactive and dynamic websites [UHH18].

The main topics covered by the website are listed below:

PeCoH Project

This item contains background information about the PeCoH project.

Perfomance Engineering

This item focuses on using systematic approaches (e.g. software engineering concepts, benchmarking and tuning, cost models) to meet performance requirements (e.g. high throughput, low latency, low memory usage, low power consumption) in a cost-effective way. The primary objective in the context of HPC is to reduce the runtimes of parallel programs and to use the resources of the HPC system appropriately for that purpose.

In the further progress of the project, topics like measuring system performance, benchmarking, tuning, cost awareness, optimization cycle, etc. will be addressed.

HPC Certification Program

In the course of the project, basic and advanced HPC knowledge and HPC competences will be provided for this item.

Success Stories

As part of the PeCoH project, standard software, as well as individual software, will be tuned. Tuning of individual software will also be based on co-development with users. Representative examples are used for the success stories to demonstrate the benefit of performance engineering methods.

We have successfully started our examination with performance improvements for examples using the language R, along with OpenMP and MPI, in order to compare promising combinations of various compilers and libraries.

Support / Help Desk

The HHCC support and help desk provide a joint user support service of DKRZ, RRZ, and TUHH RZ. If users need assistance with tuning of standard application packages, as well as their self-developed software, they can fill in a corresponding form or write an email to `helpdesk.hhcc@uni-hamburg.de` (which is connected to an open-source enterprise grade ticketing system [RT18]). The idea is to answer the question directly or delegate it to appropriate consultants of the HHCC.

Subtopics are a list of answers to the most Frequently Asked Questions (FAQ) (e.g. how to get a user account, how to login, ...), a list of Abbreviations and Acronyms, a list of Recommended Readings (e.g. Viewings, Tutorials, Papers, Books, and Useful Links), and an area for Downloads (e.g. for handouts, presentation slides, project posters, ...).

2.6.2 Collecting Success Stories (Task 6.2)

This task started in M9 and is currently based on collecting first results with performance improvements for examples using the language R, along with OpenMP and MPI (see also Section 2.5 and above on Success Stories in Section 2.6.1).

2.6.3 Knowledge Base (Task 6.3)

There were no activities planned in the reporting period.

2.7 Deliverables

In this reporting period, the following deliverables were completed according to the plan:

D1.1: "Annual Report" [HK18] state: completed (with a slight delay)

D3.1 "Costs for Running Applications" [Hü18b] state: completed

D3.2 "Integration of Cost-Efficiency in SLURM" (source code) [Hü18a] state: completed

D4.1 "HPC Competences and Certification Program" [HHK+18] state: completed (with a slight delay)

D6.1 "Web presence of the Hamburg HPC competence Center" [HHC18a] state: completed

The following deliverables are in preparation
D4.2 "Workshop Material":
This deliverable will be available when the material has been completed. We assume that the slight delay is not a problem for the project.

2.8 Dissemination Activities

The first version of the HHCC Website is online since April 2017 [HHC18a]. We will continuously extend the content for the duration of the PeCoH project.

In June 2017, we presented our project poster at the ISC 2017 [KKL+17]. Additionally, we distributed a handout [Tea17a] to the work in progress of our HPC Certification Program, which is one of the major topics of the poster. We received positive feedback in several meetings and discussions.

In July 2017, we presented the PeCoH project [Stü17] at the FEPA workshop [FEP17]. We received positive feedback in particular for our HPC Certification Program proposal.

At the SC17 [Tea17b], we distributed a handout [Tea17b] to the work in progress of our HPC Certification Program and received particularly positive feedback for the handout.

In December 2017, we presented the PeCoH project [Hü17] at the HPC-Status-Conference of the Gauß-Allianz [Gau17]. The main topics of the presentation were modeling of HPC costs to raise the cost awareness of HPC users and our HPC Certification Program.

Over the whole reporting period, there was a lively exchange with other HPC projects, in particular with the ProfiT-HPC (Profiling Toolkit for HPC Applications) project [Pro18]. For the sharing of experiences between these projects, we have met five times at the DKRZ. We also had some discussions with the teams from ProPe.

Chapter 3

Next Steps

The next steps which are planned for processing the various WPs in the near future are listed below. Tasks for which no activities are planned are omitted. The same applies to completed tasks.

3.1 WP1 Management

We will continue to verify and compare the project's progress to the expected status and hold regular face-to-face meetings, conference calls and video conferences.

We will also continue to hold working-level meetings for processing the work packages.

3.2 WP2 Performance Engineering Concepts

3.2.1 Identification of Suitable Concepts (Task 2.1)

We will complete the identification of suitable performance engineering concepts.

3.2.2 Benefit of Data Analytics (Task 2.2)

No activities are planned until M19 (start of the task).

3.2.3 Benefit of in-situ Visualization (Task 2.3)

No activities are planned until M19 (start of the task).

3.2.4 Compiler-assisted Development (Task 2.4)

(This task was postponed due to the delay in hiring qualified scientific staff.) The investigation of tools for compiler assisted development will be in the main focus of a PhD position.

3.2.5 Code Co-Development (Task 2.5)

We will continue to support users to speedup their applications based on co-development. In this way performance engineering know-how is transferred.

3.3 WP3 Performance Awareness, HPC Cost Model

As part of the risk management, we have to deal with the risk that data centers refuse to implement the cost-modeling in production; we are engaging the DKRZ user group and management to identify possibilities.

3.3.1 Deploying Feedback Tools for User Jobs (Task 3.3)

We will encourage other computing centers to use our software that reads in a cost model description and generates job cost estimates based on the SLURM accounting records in order to give feedback to users.

3.3.2 Analyzing Data and Giving Feedback to Users (Task 3.4)

No activities are planned until M19 (start of the task).

3.4 WP4 HPC Certification Program

3.4.1 Workshop Material (Task 4.3)

We will finish the coordination with other stakeholders that may provide content and then work on the completion of the complementary workshop material.

3.4.2 Online Tutorial (Task 4.4)

We will work on the completion of the online tutorial.

3.4.3 Online Examinations (Task 4.5)

We will develop of a multiple choice questionnaire to validate the basic HPC skills. It is also planned to bundle teaching material required to master the various certification levels and organize standardized online examinations for participants to acquire the certificates.

3.5 WP5 Tuning of Software Configurations

3.5.1 Determination of Tuning Possibilities (Task 5.2)

We will examine the following tuning possibilities:

- tuning without the need to (re-)build a parallel program, e.g. by tuning a parallel program from the outside via runtime options,
- tuning without modifying the source code, e.g. by using optimized libraries and setting appropriate compiler/linker options.

For this task, tuning via reprogramming will be rather an exception and limited to using higher abstraction levels like replacing sequential for loops by parallel foreach loops.

3.5.2 Setup of Realistic Use Cases (Task 5.3)

We will look — in close cooperation with the three computing centers involved in PeCoH — for the most promising and representative use cases.

3.5.3 Benchmarking (Task 5.4)

For the selected use cases, as before for benchmarking R programs, we will perform real benchmark experiments in order to verify that good speedups are achieved.

3.5.4 Documentation (Task 5.5)

We will document the results for the selected uses cases.

3.6 WP6 Dissemination

3.6.1 Web Presence of the Hamburg HPC Competence Center (Task 6.1)

We will update our HHCC website [HHC18a] regularly.

3.6.2 Collecting Success Stories (Task 6.2)

Further success stories are planned to demonstrate the benefit of performance engineering and to enable us to derive best practices and strategies supporting the scientists in their daily work.

3.6.3 Knowledge Base (Task 6.3)

We will collect ideas on how to increase the searchability of existing performance engineering material.

Chapter 4

Summary and Conclusions

The start of the PeCoH project, which involves three computing centers and three scientific organization as project partners, was originally scheduled for January 1, 2017 with a duration of three years for two fulltime positions (one PhD candidate and a postdoc position). Due to the good labor market situation for computer scientists, the postdoc position could not be filled before March 2017. The PhD candidate position has been divided among two persons (starting on July, 2017 and January 2018). These deviations from the planning should not be a major issue for the project but caused various minor adjustments to the project plan.

The PeCoH project contains six work packages. In WP1 (Management), a collaborative workspace (Website, Git repository, ...) was established at Universität Hamburg which fosters the cooperation between the project members and the coordination between the computing centers.

In WP2 (Performance Engineering Concepts), it became clear that the classification of HPC skills we performed in WP4 represents a good starting point for deepening the identification of performance engineering concepts. In the code co-development task of WP2, we speed up an R program for analyzing satellite night images by parallelizing appropriate loops with the foreach() paradigm, achieving a speedup factor of 126 on 32 cluster nodes, each node using 4 cores. The performance engineering know-how acquired in this task was successfully transferred. Due to the delay in hiring qualified scientific staff, we have postponed some activities in WP2 to the second annual reporting period.

In WP3 (Performance Awareness, HPC Cost Model), four distinct cost models (from simple to more complex) were developed. A text file contains the cost model description with one or more sets of node names and an associated list of cost rates to each node set. Two tools were developed to apply the four cost models to SLURM jobs: The first tool is supposed to be run from the job-epilogue script and reports the cost of a single job. The second tool is supposed to be run from the command line and calculates costs and statistics for a set of selected jobs based on the accounting records of SLURM.

In WP4 (HPC Certification Program), we implemented an HPC skill tree (based on XML) for the classification of HPC competences. For the development of the HPC certification program, we separated the certificate definition from content providing. This way, the learning material can be provided by different scientific institutions. We kick-started a (virtual) entity, the HPC certification forum, that will be responsible to establish accepted certificate definitions and corresponding exams. In several meetings and discussions, we received positive feedback for our new approach. For producing workshop material, we implemented a content production workflow which is based on a lightweight and easy to use markup language Markdown in combination with the Pandoc tool to convert the Markdown files to target formats like HTML or PDF. We developed programs based on XSLT to automate the process of creating the affected output target files (target format) after changes of a Markdown source file.

In WP5 (Tuning of Software Configurations), we initially set up a help desk via a ticketing systems in order to assist users with tuning of standard application packages as well as their self-developed software. In this context, we performed experiments and benchmarks for three R-programs. The results of the three use cases showed that a) the use of an efficient math library (e.g. OpenBLAS or MKL) is mandatory and b) sequential loops without data dependencies can be easily replaced by (MPI-capable) parallel loops using appropriate R packages.

In WP6 (Dissemination) the web presence of the Hamburg HPC Competences Center was created. This site gives information about the PeCoH project, provides a help desk to support users in tuning their applications and will e.g. host the HPC certification program, success stories, and a knowledge base for performance engineering material. The web site also provides recommended readings, a list of (HPC) abbreviations and acronyms, and an area for downloads.

To foster the collaboration with other scientific institutions, we presented the progress in the PeCoH project at several scientific events. Despite the delay in hiring qualified scientific staff, we are reasonably well on schedule, and the essential next steps were planned for the second annual reporting period accordingly.

Acknowledgement

The PeCoH project has received funding from the German Research Foundation (DFG) under grants LU 1353/12-1, OL 241/2-1, and RI 1068/7-1.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bibliography

[ABS08] M. Allalen, M. Brehm, and H. Stüben. Performance of quantum chromodynamics (qcd) simulations on the SGI Altix. *Computational Methods in Science and Technology*, 14(2):69–75, 2008.

[ABS18] M. Allalen, M. Brehm, and H. Stüben. Berlin quantum chromodynamics program (BQCD) — source code download. `https://www.rrz.uni-hamburg.de/services/hpc/bqcd`, 2018.

[Ato18] Atom. Atom editor — home page. `https://atom.io/`, 2018.

[Fel79] Stuart I. Feldman. Make — a program for maintaining computer programs. *Software: Practice and Experience*, 9(4):255–265, 1979.

[FEP17] FEPA. Flexible framework for energy and performance analysis in HPC centers — workshop (July 20 — 21, 2017 in Erlangen, Germany. `https://blogs.fau.de/prope/fepa-workshop-2017/`, 2017.

[Fio18] Fiona. Fiona CMS — home page. `https://infopark.com/de/plattform/cms-fiona`, 2018.

[Gau17] Gauss. 7^{th} HPC-status-conference of the Gauß Allianz (December 4 – 5, 2017 in Stuttgart, Germany). `https://gauss-allianz.de/en/hpc-status-konferenz-2017?name=value`, 2017.

[Git18] GitHub. GitHub development platform — home page. `https://github.com/`, 2018.

[Hü17] Nathanael Hübbe. PeCoH raising awareness for costs and performance [presentation of the work in progress at the HPC-status-conference of the Gauß-Allianz [Gau17]]. Slides — online available via our HHCC website `https://www.hhcc.uni-hamburg.de/en/support/downloads.html`, 2017.

[Hü18a] Nathanael Hübbe. Integration of cost-efficiency in slurm — PeCoH deliverable D3.2 (source code). Deliverable — online available via the PeCoH project webpage `https://wr.informatik.uni-hamburg.de/research/projects/pecoh/start`, 2018.

[Hü18b] Nathanael Hübbe. Modelling HPC usage costs — PeCoH deliverable D3.1 & D3.3. Deliverable — online available via the PeCoH project webpage `https://wr.informatik.uni-hamburg.de/research/projects/pecoh/start`, 2018.

[HHC18a] HHCC. Hamburg HPC Competence Center — home page. `https://www.hhcc.uni-hamburg.de`, 2018.

[HHC18b] HHCC. Hamburg HPC Competence Center – mailing list of the HPC certification program. `certification.hhcc@lists.uni-hamburg.de`, 2018.

[HHK+18] Kai Himstedt, Nathanael Hübbe, Julian Kunkel, Sandra Schröder, and Hinnerk Stüben. D4.1 HPC competences and certification program. Deliverable — online available via the PeCoH project webpage `https://wr.informatik.uni-hamburg.de/research/projects/pecoh/start`, 2018.

[HHKS18] Kai Himstedt, Nathanael Hübbe, Julian Kunkel, and Hinnerk Stüben. An HPC certification program proposal meeting HPC users' varied backgrounds. Concept paper, DKRZ and RRZ, 2018.

[HK18] Kai Himstedt and Julian Kunkel. D1.1 annual report. Deliverable — online available via the PeCoH project webpage `https://wr.informatik.uni-hamburg.de/research/projects/pecoh/start`, 2018.

[KKL+17] Julian Kunkel, Michael Kuhn, Thomas Ludwig, Matthias Riebisch, Stephan Olbrich, Hinnerk Stüben, Kai Himstedt, Hendryk Bockelmann, and Markus Stammberger. Performance Conscious HPC (PeCoH) — project poster. In *ISC High Performance 2017 (June 18 — 22, 2017). Frankfurt, Germany*, 2017.

[KYTS17] Julian Kunkel, Rio Yokota, Michela Taufer, and John Shalf, editors. *ISC High Performance 2017 (June 18 — 22, 2017). Frankfurt, Germany*, 2017.

[Pan18] Pandoc. Pandoc: a universal document converter — home page. `https://pandoc.org/`, 2018.

[PeC18] PeCoH. Scientific computing at Universität Hamburg: PeCoH project – home page. `https://wr.informatik.uni-hamburg.de/research/projects/pecoh/start`, 2018.

[Pro18] ProfiT. ProfiT-HPC: Profiling Toolkit for HPC Applications — home page. `https://profit-hpc.de/`, 2018.

[RT18] RT. Request tracker: The open-source enterprise grade ticketing system — home page. `https://bestpractical.com/`, 2018.

[SC.17] *SC17: International Conference for High Performance Computing, Networking, Storage and Analysis (November 12 — 17, 2017 Denver, Colorado)*, 2017.

[SCH18] Martin Spindler, Victor Chernozhukov, and Christian Hansen. hdm: High-dimensional metrics. `https://cran.r-project.org/web/packages/hdm/index.html`, 2018.

[Stü17] Hinnerk Stüben. PeCoH — Performance Conscious HPC [presentation of the work in progress at the FEPA workshop [FEP17]]. Slides — online available via our HHCC website `https://www.hhcc.uni-hamburg.de/en/support/downloads.html`, 2017.

[Tea17a] HHCC Team. Hamburg HPC Competence Center – handout to the work in progress of the HPC certification program at the ISC 17 [KYTS17]. Handout — online available via our HHCC website `https://www.hhcc.uni-hamburg.de/en/support/downloads.html`, 2017.

[Tea17b] HHCC Team. Hamburg HPC Competence Center – handout to the work in progress of the HPC certification program at the SC 17 [SC.17]. Handout — online available via our HHCC website `https://www.hhcc.uni-hamburg.de/en/support/downloads.html`, 2017.

[UHH18] UHH. Fiona: WCMS at Hamburg Universität. https://www.fiona.uni-hamburg.de/fiona.html, 2018.

[Urb18] Simon Urbanek. R Benchmark 2.5. http://r.research.att.com/benchmarks/R-benchmark-25.R, 2018.

[W318a] W3. World Wide Web consortium – extensible markup language (XML). https://www.w3.org/XML/, 2018.

[W318b] W3. World Wide Web consortium – XML schema. https://www.w3.org/2001/XMLSchema, 2018.

[W318c] W3. World Wide Web consortium – XSL transformations (XSLT). https://www.w3.org/TR/xslt/, 2018.

[Wes17] Steve Weston. Introduction to doMPI. https://cran.r-project.org/web/packages/doMPI/vignettes/doMPI.pdf, 2017.