Contributing HPC Skills to the HPC Certification Forum

Julian Kunkel
University of Reading
Reading, United Kingdom
j.m.kunkel@reading.ac.uk

Kai Himstedt
Universität Hamburg
Hamburg, Germany

Weronika Filinger
EPCC, The University of Edinburgh
Edinburgh, United Kingdom

Jean-Thomas Acquaviva
DDN
Paris, France

Anja Gerbes
Goethe-Universität
Frankfurt am Main, Germany

Lev Lafayette
University of Melbourne
Melburne, Australia

ABSTRACT

The International HPC Certification Program has been officially launched over a year ago at ISC'18 and since then made significant progress in categorising and defining the skills required to proficiently use a variety of HPC systems. The program reached the stage when the support and input from the HPC community is essential. For the certification to be recognised widely, it needs to capture skills required by majority of HPC users, regardless of their level. This cannot be achieved without contributions from the community. This extended abstract briefly presents the current state of the developed Skill Tree and explains how contributors can extend it. In the talk, we focus on the contribution aspects.

1 INTRODUCTION

Training was always important for the HPC community. However, creating and providing training for practitioners with diverse backgrounds and different levels of computer literacy is challenging. The continuous growth of the HPC community make the traditionally accepted training solutions insufficient. The multitude of paths leading into HPC means the training providers can hardly assume any previous knowledge or programming experience. There is no common base knowledge possessed by all new users. This makes the development and delivery of any training complicated. The main goal of the *International HPC Certification Forum (HPCCF)* is to ease the provision and uptake of training by clearly categorising, defining and eventually testing the skills required to efficiently use HPC resources. For this effort to be successful, the community needs to support and contribute to the process of defining the HPC Skill Tree. Input from members of different HPC branches is crucial. This extended abstract aims at presenting the current high-view state of the Skill Tree and describe the process of contributing.

2 SKILL TREE

The skills are organised in a tree structure from a coarse-grained to a fine-grained representation, allowing users to browse the skill based on the semantics. A skill is defined as a set of **learning outcomes** and relevant metadata. Within a single skill, there can also be multiple levels (basic, intermediate and expert level) building upon each other and further distinguishing the expertise. We expect the practitioners to acquire the lower levels before progressing to more complex levels. Currently, the Skill Tree[1] contains six branches: HPC Knowledge (**K**), Use of HPC Environment (**USE**), Performance Engineering (**PE**), Software Development (**SD**), System Administration(**ADM**), and Big Data Analytics (**BDA**). These are briefly described in the subsequent sections.

2.1 HPC Knowledge

This branch contains the basic information necessary to understand what supercomputers are, how they work and how to make use of them. It should give enough background to allow new HPC practitioners to understand different aspects of HPC environments and how to make use them. Its basic level sub-branches are: **K1-B** *Supercomputers*, **K2-B** *Performance modelling*, **K3-B** *Program Parallelisation*, **K4-B** *Job Scheduling*, and **K5-B** *Cost Modelling*.

Learning outcomes: A practitioner familiar with this branch should: understand different aspects of HPC hardware, software and operation of HPC systems; know how to use simple performance models for systems and applications; understand scaling and parallel efficiency; know different parallelisation paradigms; be familiar with using HPC systems, and understand job scheduling principles.

2.2 Use of HPC Environments

HPC environments are different from local systems and cloud environments, and different HPC systems may utilise specific solutions to setup and execute parallel applications. Although, knowing how to use a specific system is important, most users will eventually need to use more than one system. Therefore, understanding the underlying principles is equally important. This sub-tree covers skills for different user roles: users, testers and developers, allowing to efficiently develop, build, run and monitor parallel applications and automate repetitive tasks. The current basic level sub-branches include: **USE1-B** *Cluster Operating System*, **USE2-B** *Running of Parallel Programs*, **USE3-B** *Building of Parallel Programs*, **USE4-B** *Developing Parallel Programs*, **USE5-B** *Automating Common Tasks*, and **USE6-B** *Workflow Integration*.

A practitioner familiar with this branch should be able to: apply tools provided by the operating system to navigate and manage files and executables; select the software environment to effectively build and develop existing and novel applications; use a workload

https://doi.org/10.22369/issn.2153-4136/11/1/17

[1] https://www.hpc-certification.org/skills/

manager to allocate HPC resources; construct workflows that utilise remote (distributed) environments to execute parallel workflows; and design and deploy scripts that automate repetitive tasks.

2.3 Performance Engineering

Time to solution is one of the basic metrics in HPC - it's vital to obtaining the results in a timely manner and using the most optimal resources. Performance engineering gives a systematic approach to measuring and analyzing performance of systems and applications. This sub-tree should cover the performance of applications and systems, optimising of the runtime settings and applications and strategies for efficient use of HPC resources. It has five basic level sub-branches focusing on: **PE1-B** *Cost awareness*, **PE2-B** *Measuring System Performance*, **PE3-B** *Benchmarking*, **PE4-B** *Tuning*, and **PE5-B** *Optimisation Cycle*. A practitioner familiar with this branch should be able to: describe the optimisation cycle; estimate the cost a job on an HPC system; understand the typical performance pitfalls; know how to perform benchmarks and use their results as baseline; use profiling tools to analyse the performance and identify bottlenecks; understand how various system and application settings influence the performance; and finally be aware of optimised libraries and how to use them.

2.4 Software Development

Software engineering is often neglected in computational science. However, it can increase productivity by providing scaffolding for the collaborative programming, reducing the coding errors and increasing the manageability of software. This branch covers concepts, practices and methods from software engineering that should be applied in HPC environments. The current basic level sub-branches are: **SD1-B** *Programming Concepts for HPC*, **SD2-B** *Programming Best Practices*, **SD3-B** *Software Configuration Management*, **SD4-B** *Agile Methods*, **SD5-B** *Software Quality*, **SD6-B** *Software Design and Architecture*, and **SD7-B** *Software Documentation*. A practitioner familiar with this branch should be able to: apply software engineering methods and best practices when developing parallel applications; write a modular and reusable code by using software design principles; apply HPC design patterns; know how to configure and use integrated development environments (IDEs) to seamlessly perform a typical development cycle; use sophisticated debuggers for parallel programs; define and establish coding standards and conventions in a project; apply version and configuration management to establish and maintain consistency of a program or software system throughout its life; configure an environment for continuous integration with basic processing steps like compiling and automated testing; apply unit testing in a specific programming language using appropriate unit testing frameworks; and appropriately document the entire software ecosystem.

2.5 Big Data Analytics

The analysis of large volumes of data was traditionally performed in the cloud environment, utilising cheaper but less-reliable hardware. However, it's becoming more integral part of HPC workflows, utilising tools and methodology from Data Science (DS) and Artificial Intelligence (AI) to process data in order to obtain results quickly. AI can be used inside simulations or to steer workflows, while data science can be used to find interesting patterns inside the data. This branch should cover concepts and tools required for effective data analysis on HPC systems. The current basic level sub-branches include: **BDA1-B** *Theoretical principles of Big Data Analytics*, **BDA2-B** *Big Data Tools in HPC*, and **BDA3-B** *Integrating BDA with HPC workflows*. A practitioner familiar with this branch should be able to: describe and apply the concepts of artificial intelligence and data science; differentiate the various tools that could be used in an HPC environment effectively; and design a workflow consisting of HPC and BD tools to analyze the data.

2.6 System Administration

The administration of HPC systems requires integrating the state-of-the-art hardware and software at various stages of their life-cycle, to provide optimal environments for the users while managing them efficiently. This branch should cover the concepts and tools enabling efficient and cost-effective administration of HPC systems. The current basic level sub-branches include: **ADM1-B** *Cluster Infrastructure*, **ADM2-B** *Software Stack*, **ADM3-B** *Monitoring Tools*. The practitioners familiar with this branch should be able to: understand the differences between different hardware options; apply best practices for managing software and users; monitor the system and software usage; manage and maintain the optimal environment for users; and know how to establish the support structures.

3 HOW TO CONTRIBUTE

Ultimately, the chairs of the respective skill sub-trees will be in charge of curating suggestions and change requests. However, direct change requests are adopted for the phase of building the first release version of the Tree. The MindMap and Skill definitions are available in Markdown format[2] and a wiki is available to render them directly online. The skills are structured in directories according to the hierarchy in the skill tree. The MindMap structures are synchronised with the tree directory to test more invasive changes.

Contributions to the skill definitions can be made by 1) discussing them on Slack (it is a good idea to talk through non-trivial changes), 2) adjusting the skill-tree in the MindMap (editable via *FreePlane*), 3) editing the skill definitions on the Wiki, or 4) directly preparing a pull request that changes the Markdown files. As GitHub allows for commenting of individual lines, this provides means for rapid feedback as well.

4 CONCLUSIONS

The *HPC Certification Forum* is an effort to structure the HPC-related skills and to offer certification to users. The high-level descriptions of the sub-trees provide an overview and indicate our goal of creating a comprehensive tree. The tree itself will be released in stable versions (and version controlled) and updated periodically. We work towards the first release but need the input of the community to refine and complete, particularly the leaf-levels of the skill-tree.

ACKNOWLEDGMENTS

We thank the contributors to the HPC Certification Forum.

[2] See our GitHub https://github.com/HPC-certification-forum