The HPC Skill Tree – A Brief Overview

Kai Himstedt – On Behalf of the HPC-CF Board

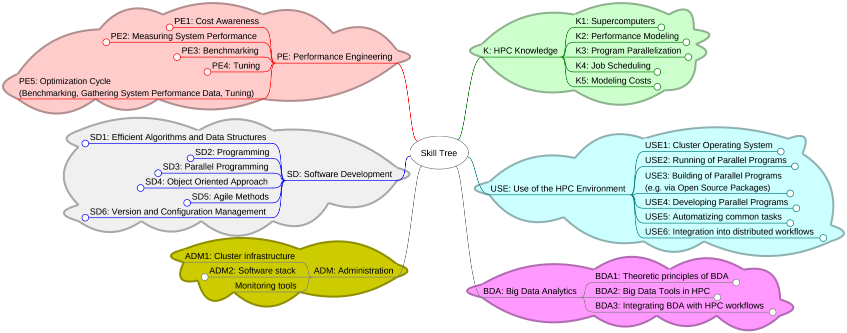
BoF: International HPC Certification Program

ISC 19, Frankfurt, Germany

June 18, 2019

The HPC Certification Forum: https://www.hpc-certification.org/

Representing HPC Competences by Skills



Skill Tree

PE: Performance Engineering
- PE1: Cost Awareness
- PE2: Measuring System Performance
- PE3: Benchmarking
- PE4: Tuning
- PE5: Optimization Cycle (Benchmarking, Gathering System Performance Data, Tuning)

SD: Software Development
- SD1: Efficient Algorithms and Data Structures
- SD2: Programming
- SD3: Parallel Programming
- SD4: Object Oriented Approach
- SD5: Agile Methods
- SD6: Version and Configuration Management

ADM: Administration
- ADM1: Cluster infrastructure
- ADM2: Software stack
- Monitoring tools

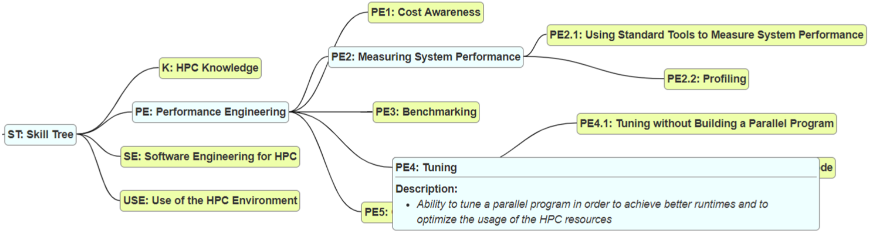
K: HPC Knowledge
- K1: Supercomputers
- K2: Performance Modeling
- K3: Program Parallelization
- K4: Job Scheduling
- K5: Modeling Costs

USE: Use of the HPC Environment
- USE1: Cluster Operating System
- USE2: Running of Parallel Programs
- USE3: Building of Parallel Programs (e.g. via Open Source Packages)
- USE4: Developing Parallel Programs
- USE5: Automatizing common tasks
- USE6: Integration into distributed workflows

BDA: Big Data Analytics
- BDA1: Theoretic principles of BDA
- BDA2: Big Data Tools in HPC
- BDA3: Integrating BDA with HPC workflows

First Two Levels of the Current Skill Tree

Classification of HPC Competences

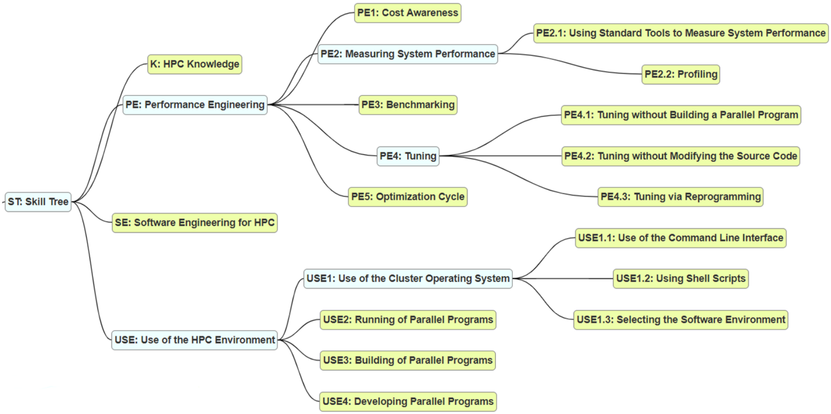

ST: Skill Tree
K: HPC Knowledge
PE: Performance Engineering
SE: Software Engineering for HPC
USE: Use of the HPC Environment
PE1: Cost Awareness
PE2: Measuring System Performance
PE2.1: Using Standard Tools to Measure System Performance
PE2.2: Profiling
PE3: Benchmarking
PE4.1: Tuning without Building a Parallel Program
de
PE5:
PE4: Tuning

Description:

- *Ability to tune a parallel program in order to achieve better runtimes and to optimize the usage of the HPC resources*

- Skills close to the root: Generic
- Skills at leaf level: Specific
  - Granularity: 1.5 to 4h of learning material per leaf

Why Do We Use a Tree?


ST: Skill Tree
K: HPC Knowledge
PE: Performance Engineering
PE1: Cost Awareness
PE2: Measuring System Performance
PE2.1: Using Standard Tools to Measure System Performance
PE2.2: Profiling
PE3: Benchmarking
PE4: Tuning
PE4.1: Tuning without Building a Parallel Program
PE4.2: Tuning without Modifying the Source Code
PE4.3: Tuning via Reprogramming
PE5: Optimization Cycle
SE: Software Engineering for HPC
USE: Use of the HPC Environment
USE1: Use of the Cluster Operating System
USE1.1: Use of the Command Line Interface
USE1.2: Using Shell Scripts
USE1.3: Selecting the Software Environment
USE2: Running of Parallel Programs
USE3: Building of Parallel Programs
USE4: Developing Parallel Programs

- Skills are generally built upon one another
- Skills depend on sub-skills

Current Skill Tree Statistics

There are 6 major branches at level 1

- HPC Knowledge (K)
- Performance Engineering (PE)
- Software Engineering / Software Development (SE / SD)
- Use of the HPC Environment (USE)
- Big Data Analytics (BDA) (recently added)
- Administration (ADM) (recently added)

Skills at level 2: $\approx$ 31; at level 3: $\approx$ 50; at level 4: $\approx$ 5

Skills at the leaf level: $\approx$ 66

Definition of a Skill (1)

Each skill consists of

- Unique name / ID

  e.g. *Benchmarking / PE3*

- Background information
  - Motivation

    Benchmarking example:
    Benchmarking is essential in the HPC environment to determine speedup and efficiencies of a parallel program
  - Main focus

    Benchmarking example:
    Benchmarking emphasizes on carrying out controlled experiments to measure the runtimes of parallel programs

- Educational level: *Basic*, *Intermediate*, *Expert*

...

Definition of a Skill (2)

...

- Aim ("What is covered by the skill")

  Benchmarking example:
  comprehending and describing the basic approach of benchmarking to assess speedups and efficiencies of a parallel program

- Learning outcomes ("What are the students learning")

  Benchmarking example (extract):
  measuring runtimes (e.g. /usr/bin/time)
  performing experiments using 1, 2, 4, 8, 16, ... nodes
  generating a typical speedup plot
  ...

- List of dependencies from sub-skills

  Analogy: targets and dependencies in a *Makefile*

Using the Skill Tree in the PeCoH Project

K. Himstedt, N. Hübbe, S. Schröder, M. Kuhn, J. Kunkel,
T. Ludwig, S. Olbrich, M. Riebisch, H. Stüben

BoF: International HPC Certification Program

ISC 19, Frankfurt, Germany

June 18, 2019

PeCoH is supported by Deutsche Forschungsgemeinschaft (DFG) under grants LU 1335/12-1, OL 241/2-1, RI 1068/7-1

Performance Conscious HPC (PeCoH)

Three Hamburg compute centers involved

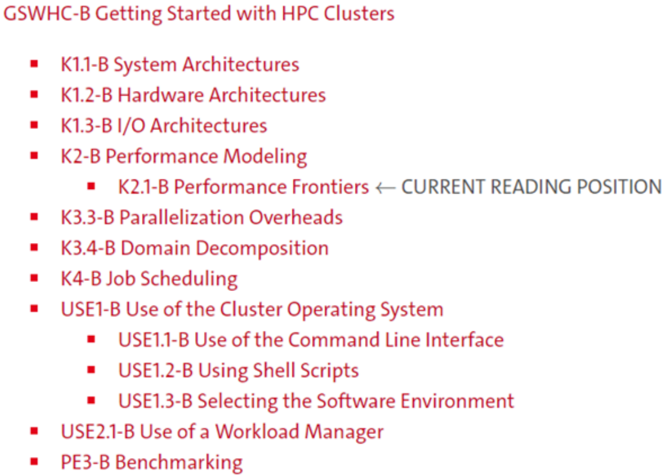
- German Climate Computing Center / Deutsches Klimarechenzentrum (DKRZ)
- Regional Computing Center / Regionales Rechenzentrum der Universität Hamburg (RRZ)
- Computer Center of Hamburg University of Technology / RZ der Technischen Universität Hamburg (TUHH RZ)

Major project goals

- Efficient usage of HPC resources by well-trained users
- Reduced efforts for user support

We have the role as content provider for *Basic* HPC skills

Sets of Skills Can Easily Be Bundled



GSWHC-B Getting Started with HPC Clusters

- K1.1-B System Architectures
- K1.2-B Hardware Architectures
- K1.3-B I/O Architectures
- K2-B Performance Modeling
  - K2.1-B Performance Frontiers ← CURRENT READING POSITION
- K3.3-B Parallelization Overheads
- K3.4-B Domain Decomposition
- K4-B Job Scheduling
- USE1-B Use of the Cluster Operating System
  - USE1.1-B Use of the Command Line Interface
  - USE1.2-B Using Shell Scripts
  - USE1.3-B Selecting the Software Environment
- USE2.1-B Use of a Workload Manager
- PE3-B Benchmarking

Available soon via Hamburg HPC Competence Center (HHCC): https://www.hhcc.uni-hamburg.de/

Mapping 110 Slides to the Skill Tree (1)

How to produce content for *Basic* level skills?

- Idea: mapping of existing material to the skill tree
- Selected: course in Parallel Programming (110 slides)
- Assign each slide to the appropriate skill in the tree

Observations

- All slides could be uniquely assigned
- Often blocks of consecutive slides were assigned to the same skill
- Consecutive blocks are "scattered" in the tree
- Slides do not fully cover all learning objectives for every skill in "Getting started with HPC Clusters"

Mapping 110 Slides to the Skill Tree (2)

Findings

- Mapping requires a certain amount of time
  - Overhead for creating and managing a "mapping table"
  - Scanning of skill tree for each slide to find the match
  - Time needed on average: a few minutes per slide
- Original course structure gets lost during mapping
  - Not a problem: focus was on the content of the course
- Further mapping of existing material is necessary to complete "Getting started with HPC Clusters"