# HPC tools and usage:
# a view from the community

**DDN**®

AI | BIG DATA | HPC

Jean-Thomas Acquaviva
Jtacquaviva@ddn.com

# Goal: Refinement of the Skill tree
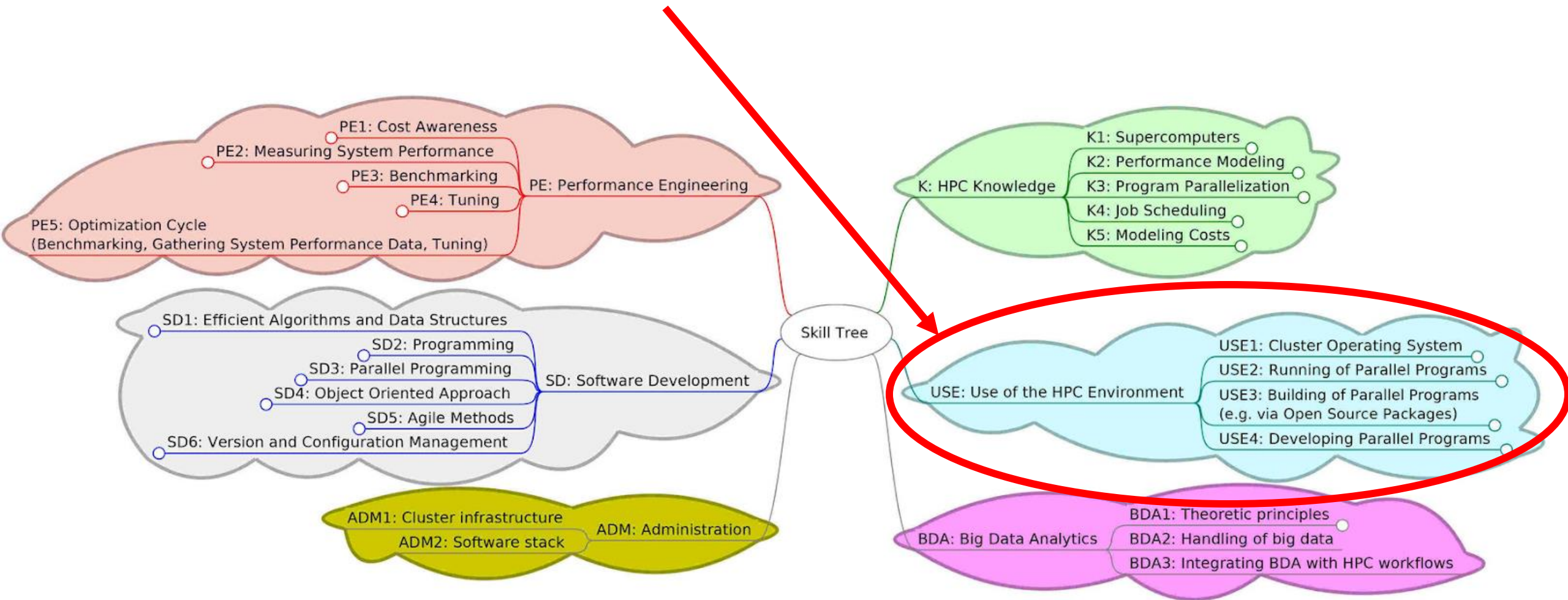


Capture the relevant skills to require for certification

# On-line survey

▶ EPCC Blog

- http://www.epcc.ed.ac.uk/blog/2019/hpc-certification-programme-hpc-environment-survey

▶ HPC mailing lists

- Actually it works quite well

▶ LinkedIn

- Good for click not so much for contribution

# HPC is a community

¾ in HPC since 5-10 or more than 10 years

⅓ from the industry

⅓ computer science engineer

⅙ from production in public organization

# Question 1: What is HPC Environment?

▶ **It's really about production!**

- Interacting (50%), developing (50%)
- Building code (60%)
- Running code (80%)

▶ **Interaction is about ssh and Shell scripting (75%)**

▶ **In the case where shell is not enough Python is #1 language (70%)**

- Perl and R are lagging behind (1 over 4)

# Running code: The importance of Virtual Machine and Containers

▶ 45% are daily or frequent users of VMs

▶ 30% are daily or frequency users of Containers

- an extra 30% is planning to use them

- Interesting item to follow up for next year

# Writing code: The non-importance of Integrated Dev. Environment

► ⅔ do not care about IDE

► Finally VIM (65%) has won over Emacs (20%)

► Interestingly Emacs ~ Eclipse ~ Jupyter

- Jupyter is already leading the pack

- Notebook trend to track

► Make and CMake dominate, autotool still in the loop

- Marginal presence of other building tool such scons
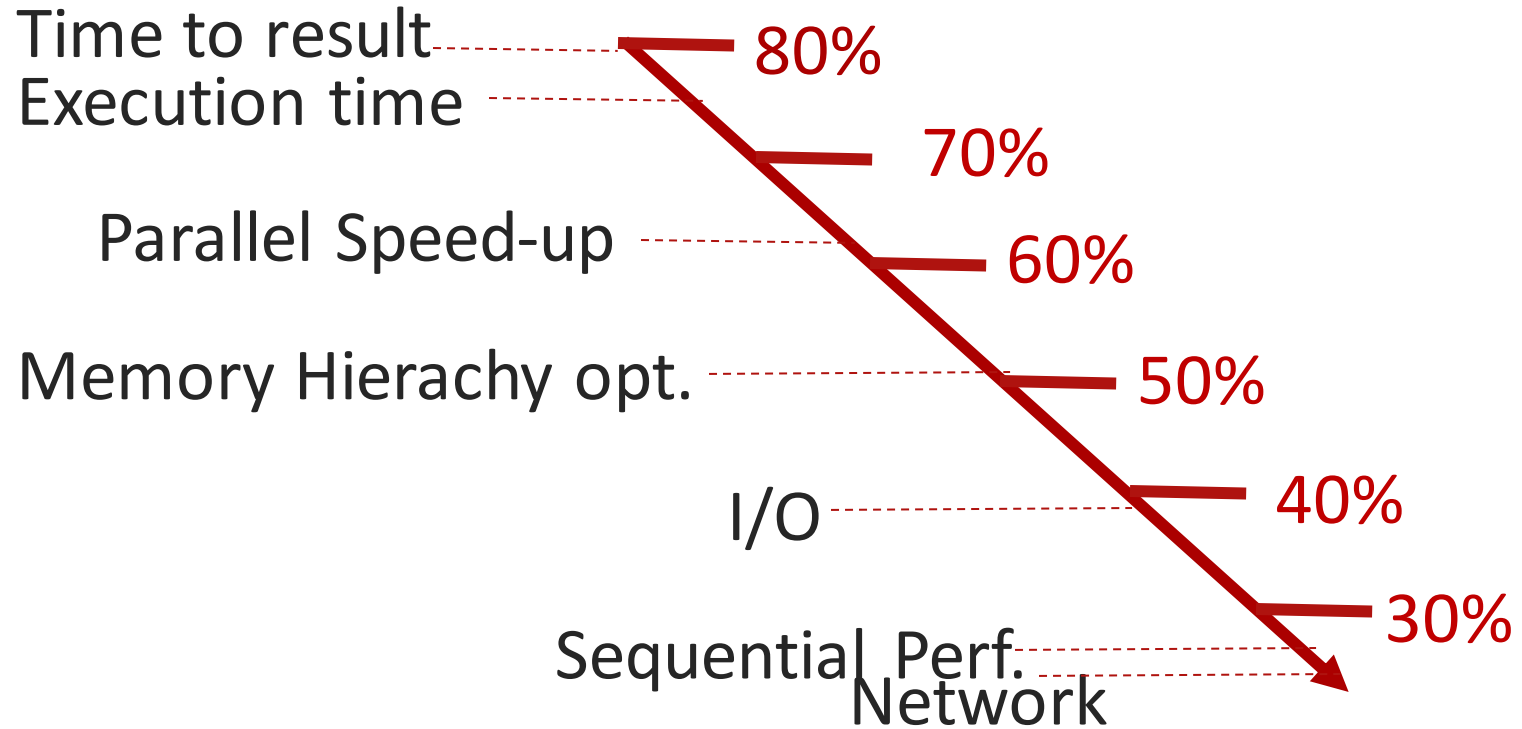
► 80% use Git!

# Compilers & Debuggers

▶ Strange intermixed of open source and proprietary software

- GCC (80%) the prefered compiler with slight advantage over ICC (75%)
- LLVM (40%) more used than NVCC (30%)

▶ GDB for debugging with a good presence of Allinea DDT

- Valgrind and PMPI used by 1/3

▶ Integration and testing done with Gitlab and a bit of Jenkins / Travis

# Expressing Parallelism

► OpenMP is matching MPI in term of popularity

- OpenMPI and IntelMPI are the most popular MPI distro

► GDB for debugging with a good presence of Allinea DDT (1/3)

► LD_PRELOAD used by ¼

► Integration and testing done with Gitlab and a bit of Jenkins  / Travis

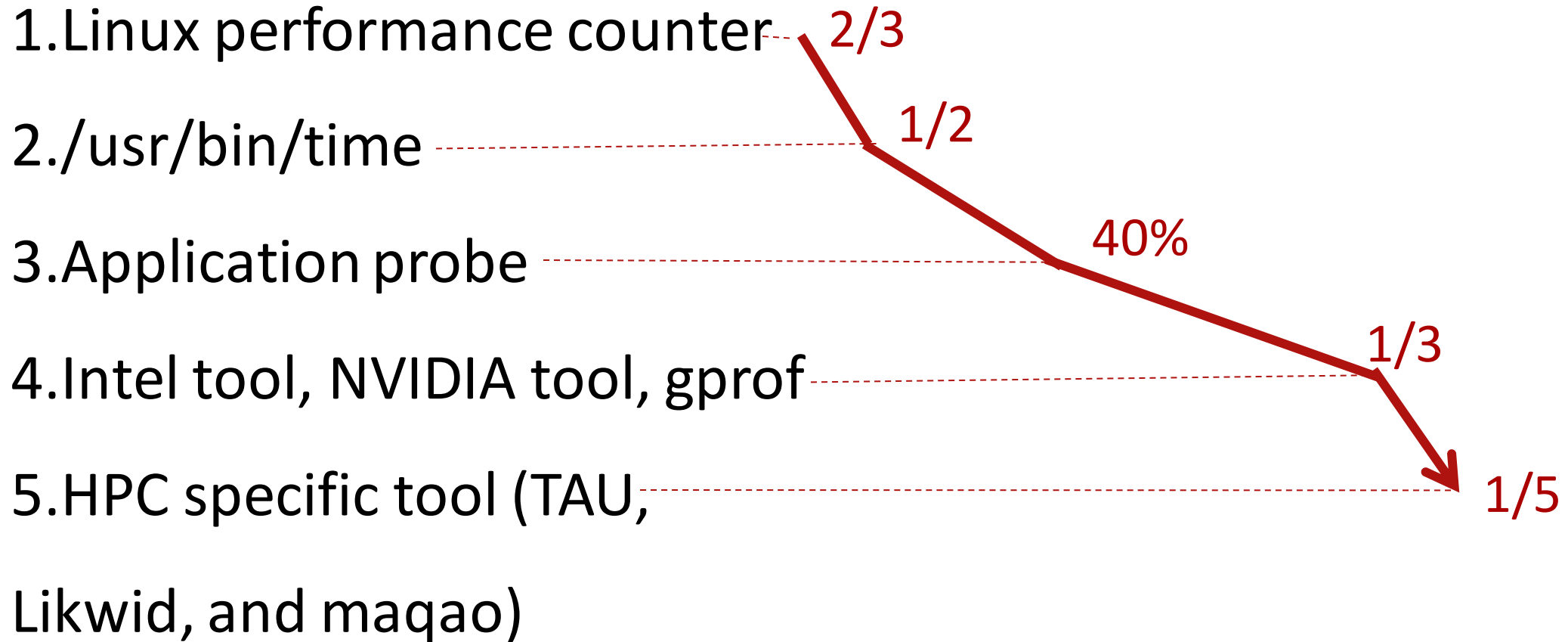# Performance Criteria

From most to least importance

Time to result — 80%
Execution time — 70%

Parallel Speed-up — 60%

Memory Hierachy opt. — 50%

I/O — 40%

Sequential Perf. — 30%
Network

USer defined metrics lead, μbenchmarking lags

# Perfomance tools

1. Linux performance counter     2/3

2. /usr/bin/time                       1/2

3. Application probe                          40%

4. Intel tool, NVIDIA tool, gprof                     1/3

5. HPC specific tool (TAU,                                      1/5

Likwid, and maqao)

# Job Schedulers

▶ Slurm 3/4

▶ PBS 1/3

▶ Singularity 1/8

# People are consistent!

⅔ of voters consider a certification program as... good or very good :-)