



SHAREing

Skills Hub For Accelerated Research Environments
Inspiring the Next Generation

Thomas Flynn, Eva Fernandez Amez and Tobias Weinzierl



We focus on Research Technical Professionals (RTPs) such as software and platform engineers, researchers, etc. **Training** is split into: technical skills in accelerated computing and professional skills such as communication, project delivery, and leadership skills.



Performance Assessments rigorously analyse software not only to detect performance issues but, paired with machine assessments, can help advise the viability of software to accelerator porting and if there are specific architectures that are best suited.



SHAREing is coordinating a performance focused **Ecosystem and Community** consisting of an integrated testbed landscape to grant RTPs access to new and novel hardware. The community delivers through events, workshops and conferences to build a firm knowledge base.

Overview

A **flowchart** of our performance assessment is given below beginning with an incoming request, through the **assessment methodology** and resulting in a finalised performance **report**:



Preassessment

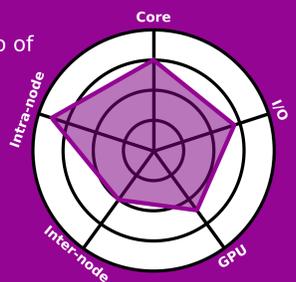
We need certain information to begin running and assessing code:

- Code repository with an archive tag and characteristic **benchmark**
- Software **dependencies** and libraries
- Compilers
- Parallel libraries/models
- **Build & run** instructions
- Example runtimes and parallel resource configurations

High-level assessment

We break down performance into 5 topics: **core**, **intra-node**, **inter-node**, **GPU**, and **I/O** for which we define metrics:

1. **Core** - ratio of peak FLOPS
2. **Intra-node** - using a strong scaling, what ratio of total node core count does the parallel efficiency maintain at least 60%
3. **Inter-node** - similar method to intra-node but using a weak scaling
4. **GPU** - the total occupancy
5. **I/O** - proportion of runtime in reads and writes



Metrics are scored by **threshold** criteria into: good, potential flaw and serious flaw. These metrics then direct our subsequent **lower-level analysis**.

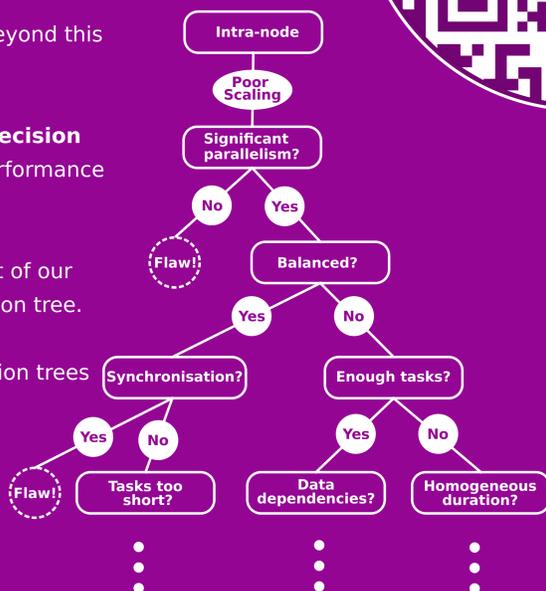
Lower-level assessment

• How to make progress beyond this high-level analysis?

• We propose a series of **decision trees** to isolate specific performance issues.

• Here we show an excerpt of our example **intra-node** decision tree.

• We plan that these decision trees will be hosted online as **interactive** resources with support given to answer each question on the decision tree.



Tools & Training

• We are designing a performance methodology which is not based on a set tool suite. We want a performance methodology that '**reaches for the right tool for the job**'.

• Alongside DiRAC we have built an '**HPC Computing Concepts**' knowledge graph that is interactive and contains short videos.

• We are building the stepping stones from the fundamentals to implementing **good performance practice**.

