SC19 BOF: QUALITY ASSURANCE AND CODING STANDARDS FOR PARALLEL SOFTWARE

Dirk Pleiter | SC'19, Denver | 21.11.2019

JÜLICH
Forschungszentrum

Why is Quality Assurance Increasingly Important?

Increasing complexity of application increases risk of introducing errors

- Quality assurance procedures help avoiding introducing (and keeping undetected) errors

Code correctness and standard compliance becomes more important when code has to run on a variety of architectures

- Errors may show-up when moving to other architectures

Accomplish code refactoring for exascale systems efficiently

- Need to keep "technical debt" low

Write code that allows to leverage different levels of parallelism

- Example: Robust frameworks facilitating data layout transformations

Digression: Technical Debt

exa2pro

Technical debt as a metaphor: Financial debt vs. future costs associated to fixing immature software

- Economic concept

Aim for developers being aware of the technical debt that they incur during application development

- Aim for continues assessment of technical debt, e.g. application of coding quality rules

Technical debt management tools

- Based on static code analysis
- Technical debt assessment
  - Systematic approach to prioritise code updates
  - Approach to jointly defining and prioritising code standards

How to Promote Quality Assurance?

Develop/enhance/promote uptake of relevant tools

- Need to establish benefits for code developers

Involve software engineers in development of scientific applications

- Very hard to achieve due to lacking attractive career paths

Improve education and training

- "Software Development in Science" is often not part of curricula at universities or HPC training programs