Contribution ID: **90** Type: **Group assignment**

Group assignment for Track 2: Parallel and Optimised Scientific Software

Wednesday, 16 June 2021 14:00 (1h 30m)

Topic for group 1: Large software systems are more and more difficult to maintain over years. In addition, programming languages evolve and the relevant expertise is lost (e.g. about programming in Fortran). When is the right moment to restart from scratch a large software project? Is it possible at all?

Topic for group 2: How to ensure long term data preservation? Today, we can read the writings of Newton, and redo his computations. However, in 300 years, will someone be able to rerun todays' software? How to make it happen? Is it feasible at all?

Topic for group 3: How to ensure good test coverage of a large code base? How to test software that will run on thousands of machines concurrently?

Topic for group 4: A lot of bad quality code and bugs are introduced in physics software due to lack of knowledge of computing languages by non expert software developers. How can we spread better the computer science knowledge and best practices in large scientific collaborations?

Topic for group 5: Every now and then, new hardware or software appears, with often very promising prospects. However, the risk is that they disappear within a few years (think of object oriented databases, Google glasses etc.). How to take benefit of latest technologies without jeopardising a multi-decade project?

Topic for group 6: What's the impact of hardware evolution and choices on software and programming languages? Is is realistic to have hardware agnostic programming languages?

Topic for group 7: According to Donald Knuth, "Premature optimization is the root of all evil". Have you ever had similar experiences? How to decide when a good moment to do optimisation, and what to optimise?

Summary

Primary author: PONCE, Sebastien (CERN)

Track Classification: Track 2: Parallel and Optimised Scientific Software