ETP4HPC “HPC Software” Working Group

Michèle Weiland and Mark Parsons
EPCC, The University of Edinburgh

WG Remit

- Key motivation: increase the ***impact of software*** that is developed in the EC, e.g. as part of FP7 and H2020 projects
- In the context of the WG, software not limited to scientific applications, but includes system level software and libraries
- Separate from and in addition to the focussed topics that are addressed in the SRA

ETP4HPC "HPC Software" Working Group

Prof Mark Parsons & Dr Michèle Weiland, EPCC

10th March 2016

Motivation

ETP4HPC recognises the importance of software on the context of HPC and it would like to establish a Working Group (WG) that focusses on issues that are specific to HPC software. This is separate from and in addition to the focussed topics that are addressed in the SRA, and will take a higher level view on assessment, integration and sustainability of HPC software as we head towards the Exascale era.

The key motivation for the HPC Software WG is to increase the long-term impact of software that is developed in the European Union, e.g. as part of FP7 and H2020 projects. In this context, software is not limited to scientific applications, but includes system level software and libraries (often called 'systemware').

Proposed topics

In order to increase its impact, HPC software should be highly scalable, well designed and engineered, and sustainable. The initial list of proposed topics for this WG to address is therefore:

1. **Scalability:** the aim is to collect information on which applications scale well and which do not, and analyse the reasons behind this. Although this focusses primarily on scientific and industrial modelling and simulation software, it should not be limited to it and should also include system software and tools.
2. **Software design and software engineering:** when it comes to extreme scale parallelism, these are poorly understood topics. For instance, which software design patterns work particularly well for extreme scale parallelism? How can software be designed and engineered in such a way that it is highly tuned for parallelism and performance, yet remains usable by domain scientists who want to contribute to the software? The outcome of this topic would be a best practice guide for HPC software development, based on an analysis of existing, well-designed and engineered applications.
3. **Sustainability:** a large percentage of the software that is developed by research projects in Europe is probably never used outside or beyond the lifetimes of those projects. There are two main obstacles to the wider adoption of such software: firstly, the software is not easily "discoverable"; and secondly, the quality of software developed as part of research projects is often substandard. The goal of this topic would be to investigate different mechanisms for improving the sustainability of software, such as a dedicated repository that collates contributions and/or a reviewing process that would give software a quality seal.

The topics outlined here are a preliminary list, which will be expanded over time as the working group matures.

Status update

- Kick-off call on 10th June
- Organised at short notice, but well attended (13 participants)

- Agenda of call
  - Discussion of WG purpose and remit
  - Expected outcomes

- Discussion of topics based on 1-page outline

Topic 1: Scalability

- Scalability topic should include efficiency
  - Need to define efficiency, e.g. to be “efficient use of resources”
- Must include data analytics
- Performance modelling?

- Proposal: make “Efficient use of resources” the topic, with “Scalability” being a subtopic?

Topic 2: Software design & engineering

- Collection of success stories of properly designed & engineered code
  - Highlighting the advantages (e.g. efficiency)
  - Pilot examples
- It is important to bring the world of CS and HPC software closer together
- Need a metric for “good” software practice

- Question: who is our target audience?
  - Who needs to/should know about HPC software design & engineering?

Topic 3: Sustainability

- Need an uptake metric for software
- How can we make software discoverable?
  - Software “market”?
  - Public software repository, plus documentation?

- It is already possible to get a DOI for software
  - Software journals, e.g. Journal of Open Research Software
  - SSI

Next steps

- Define outputs
  - Case studies for website?
  - Technical report?
  - Software repository?
  - Input into Work Programme, e.g. around Software Engineering for Extreme Scale?
  - …
- Define a timeline for the WG
- Circulate minutes & arrange next call