Geophysical Research Abstracts
Vol. 17, EGU2015-8340, 2015
EGU General Assembly 2015

The Five ‘R’s’ for Developing Trusted Software Frameworks to increase confidence in, and maximise reuse of, Open Source Software.

Ryan Fraser (1), Lutz Gross (2), Lesley Wyborn (3), Ben Evans (3), and Jens Klump (1)

(1) ARRC, CSIRO, Kensington, Australia , (2) School of Earth Sciences, The University of Queensland, Brisbane, Australia, (3) National Computational Infrastructure, Australian National University, Canberra, Australia

Recent investments in HPC, cloud and Petascale data stores, have dramatically increased the scale and resolution that earth science challenges can now be tackled. These new infrastructures are highly parallelised and to fully utilise them and access the large volumes of earth science data now available, a new approach to software stack engineering needs to be developed. The size, complexity and cost of the new infrastructures mean any software deployed has to be reliable, trusted and reusable.

Increasingly software is available via open source repositories, but these usually only enable code to be discovered and downloaded. As a user it is hard for a scientist to judge the suitability and quality of individual codes: rarely is there information on how and where codes can be run, what the critical dependencies are, and in particular, on the version requirements and licensing of the underlying software stack.

A trusted software framework is proposed to enable reliable software to be discovered, accessed and then deployed on multiple hardware environments. More specifically, this framework will enable those who generate the software, and those who fund the development of software, to gain credit for the effort, IP, time and dollars spent, and facilitate quantification of the impact of individual codes. For scientific users, the framework delivers reviewed and benchmarked scientific software with mechanisms to reproduce results.

The trusted framework will have five separate, but connected components: Register, Review, Reference, Run, and Repeat.

1) The Register component will facilitate discovery of relevant software from multiple open source code repositories. The registration process of the code should include information about licensing, hardware environments it can be run on, define appropriate validation (testing) procedures and list the critical dependencies.

2) The Review component is targeting on the verification of the software typically against a set of benchmark cases. This will be achieved by linking the code in the software framework to peer review forums such as Mozilla Science or appropriate Journals (e.g. Geoscientific Model Development Journal) to assist users to know which codes to trust.

3) Referencing will be accomplished by linking the Software Framework to groups such as Figshare or ImpactStory that help disseminate and measure the impact of scientific research, including program code.

4) The Run component will draw on information supplied in the registration process, benchmark cases described in the review and relevant information to instantiate the scientific code on the selected environment.

5) The Repeat component will tap into existing Provenance Workflow engines that will automatically capture information that relate to a particular run of that software, including identification of all input and output artefacts, and all elements and transactions within that workflow.

The proposed trusted software framework will enable users to rapidly discover and access reliable code, reduce the time to deploy it and greatly facilitate sharing, reuse and reinstallation of code. Properly designed it could enable an ability to scale out to massively parallel systems and be accessed nationally/ internationally for multiple use cases, including Supercomputer centres, cloud facilities, and local computers.