Software support for ARC infrastructure

Ryan Taylor
Senior Research Computing Specialist, University of Victoria
rptaylor@uvic.ca

In the current model, research software funding is granted to research groups to support the development of bespoke software applications to address specific research goals. I suggest that funding for research software should also be provided to support the software components that form the *underlying shared infrastructure and platforms of ARC systems*, on which all ARC user applications run.

Many core infrastructure components of the Compute Canada Federation ARC sites use open source software, for example SLURM for batch job scheduling, Lustre as a shared cluster filesystem, and CVMFS for software distribution across sites[1]. Reliable and performant operation of these components is critical for the CCF ARC sites to fulfill their mission.

By funding one or more software development positions dedicated to the open source software projects that are of strategic importance to the infrastructure of Canada's national ARC sites, NDRIO will benefit in several ways.

1. **Improved infrastructure support.** In some cases the effort to resolve a problem on a production system may reach the point of reading and debugging code, in which case progress can be difficult due to complexity of the code base, and the challenge of writing, compiling and testing specially patched code and applying it to running production systems. Instead, by cultivating in-house software support and development expertise on these open source software projects, the combined skill sets of system administrators and software developers can work together to identify a problem on a production system and resolve it in a future upstream version of the software, packaged and distributed as a standard release. Similar considerations apply in terms of performance and efficiency, which are essential characteristics of ARC systems. Identifying real-world bottlenecks and performance issues may only be possible in the production environment where real workloads are running, while streamlining code to improve performance requires knowledge of the code base and software engineering skills, so optimizing code for production on ARC systems often requires overlapping proficiencies.

[1] The open source software projects and components used in CCF infrastructure are too numerous to comprehensively identify here. Identifying which ones would most benefit from software development support is out of scope of this short paper. CVMFS, SLURM and Lustre are listed only as some prominent examples and are not an exclusive list. My personal perspective is mostly in the context of CVMFS.

2. **Giving back, not just taking.** Openness, collaboration, community, and contribution are fundamental to the success of open source software projects. Open source software is in turn fundamental to ARC - indeed, all of the Top 500 supercomputers in the world run Linux. Contributing back to the FOSS (free and open source software) projects that we rely on for ARC infrastructure is a virtuous cycle that will make those projects better and healthier, benefiting their users and the research ecosystem, including us.
3. **Improved relationships and planning.** By funding software developer(s) who make code contributions back to the software projects that CCF ARC infrastructure relies on, we will foster productive working relationships with those open source communities, earn a say in the future development and implementation of features that may be of particular importance to us, allowing us to advocate for our priorities, and enabling us to conduct long-term planning of national ARC infrastructure deployments in conjunction with software development roadmaps.

The developer positions should be integrated within the same structure of the current Compute Canada Federation and future NDRIO organization to facilitate support and interaction within the context of national sites and teams.