TRL's Response to DOE ASCR's Stewardship of Software for Scientific and High-Performance Computing RFI

Robert E. Bartolo, Ph.D.
CEO and Founder
Transformational Liaisons (TRL) LLC

In brief, what having a robust "Stewardship for Scientific and High-Performance Computing" comes down to, especially for maintaining opensource software, is sustained funding for staff positions for dedicated "software stewards". Furthermore, a dedicated reward-system for developing such scientific software and responding to often mundane user requests and bug fixes from a diverse open-source user ecosystem is also needed. Such software usually must piggyback on a specific DOE (or other funding agency) program mission with a specific deliverable in a given focus area, even though the open-source code benefits a diverse collection of researchers from many different disciplines. Maintaining such open-source software for the benefit of a diverse user ecosystem is often a side job for evenings and weekends with little reward or acknowledgement on the part of "management" whether it is at a university, national lab, or a funding agency. All parties ("management") want "new science" and publications in prestigious journals. More mundane software development and user support is often a thankless task for which there is little reward other than personal satisfaction. There needs to be a dedicated cross government agency open-source cyberinfrastructure funding category that supports salaried staff funded and dedicated mundane bug fixes and software improvements in response to requests from a diverse ecosystem of users. Such tasks are not currently rewarded in terms of recognition or promotion. There needs to be a separate budget category for open-source scientific software stewardship, analogous to a dedicated user facility, like a synchrotron light with dedicated support staff (or software stewards). Such a budget category, and software support, would be relatively inexpensive to fund (compared to a synchrotron light source for example) yet would pay huge dividends in terms of supporting a very diverse scientific ecosystem.

Infrastructure: Providing infrastructure for software packaging, hosting, testing, and other common capabilities. What infrastructure requirements do you have in order to productively develop state-of-the-art software for scientific and high-performance computing?
Response: From the above, dedicated staff with long term support for developing and maintaining such state-of-the-art software.

Figure 1 below encapsulates some possible issues for further study regarding DOE funded open-source codes and issues facing the theme of this RFI. It can be seen that the top 2 most highly used material science DFT codes, VASP a commercial code from Vienna (in grey) and Quantum Espresso an open source code from Italy (green), were developed outside the US [1]. Note that NWChem development is being funded as part of the Exascale Computing Program (ECP), yet as can be seen from Figure 1, has relatively small user requests through NERSC. The notable exception is Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) run out of Sandia National Laboratory [2], also part of the ECP program (the EXAALT program), that has a sizable user request on DOE NERSC systems. The core LAMMPS developers are to be commended for doing such a great job fostering a user ecosystem around this open-source code.

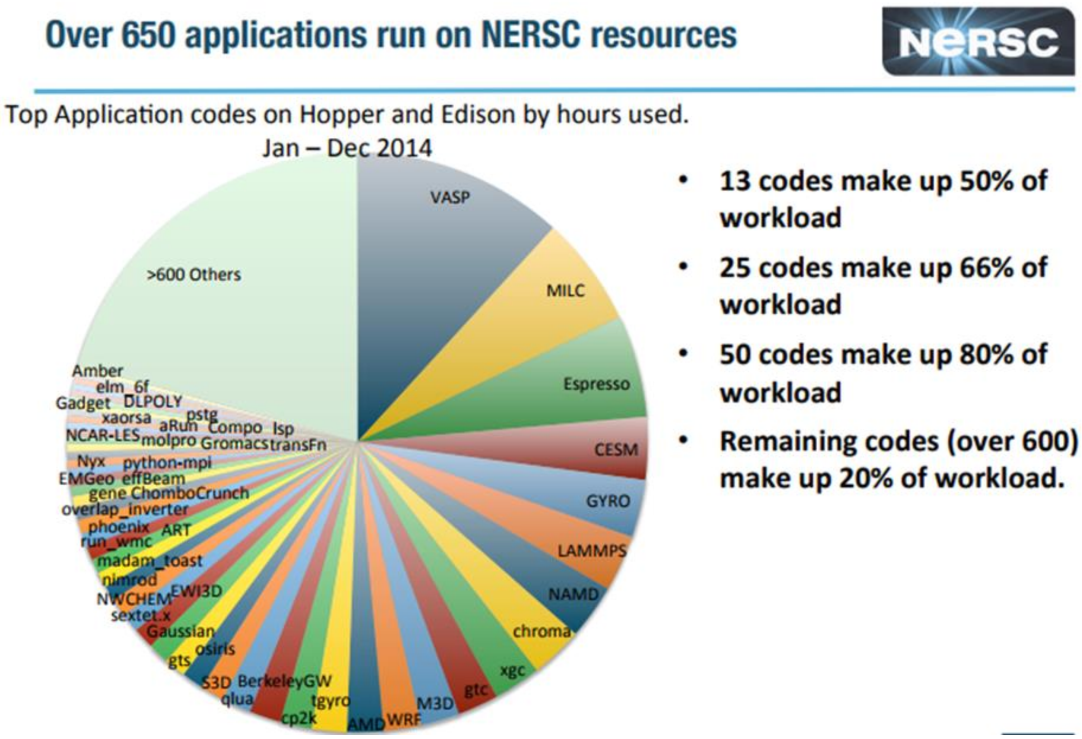

Figure 1: Showing the top 30 applications being run on NERSC resources. Taken from [1]. See specifically the 2014 study page 9 at: https://portal.nersc.gov/project/mpccc/baustin/NERSC_2014_Workload_Analysis_v1.1.pdf

Related question from the ASCR RFI*: What are key obstacles, impediments, or bottlenecks to progress by, and success of, future development of software for scientific and high-performance computing?*

This is the key question to which there are likely many different answers. I will just enumerate several based on my observations. This may sound completely obvious, but first and foremost scientists need software that actually works, doesn't crash when you run it, and doesn't have bugs in them. Scientific users don't have time to fix software bugs in open-source code and will therefore naturally migrate to code that actually gets the job done and is the most reliable. Time is money, so a user license is often not an issue in academia where there are often steep discounts.

The view from the code developers brings up a different set of issues, "I don't get paid nor promoted based on fixing all the different bugs that might arise" from any user ecosystem that has developed around it. I do the best I can while working on many different sponsored programs, which are my main responsibility. Open-source code development is usually just a side job for evening and weekends, especially for open-source code that depends on an army of volunteers.

In conclusion, I want to thank DOE and ASCR for taking the time to publish this RFI on this extremely important topic that has until recently been overlooked. There are many other points I would like to make, but in the interest of time I will conclude here. I hope that ASCR continues to take comments from the scientific community going forward.

References:

[1]"Benchmarking & Workload Characterization," NERSC. https://www.nersc.gov/research-and-development/benchmarking-and-workload-characterization/ (accessed Dec. 13, 2021).
[2] A. P. Thompson et al., "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," Computer Physics Communications, vol. 271, p. 108171, Feb. 2022, doi: 10.1016/j.cpc.2021.108171.