A cast of thousands: How the IDEAS Productivity project has advanced software productivity and sustainability Era: New Code Capabilities, and Challenges

Date: November 8, 2023

Presented by: David Bernholdt (Oak Ridge National Laboratory)

(The slides are available under “Materials from the Webinar” in the above link.)

Q. Communicators may consume hardware resources. I wonder if #3 (slide 11) should be "must allow a user-provided communicator" so communicators can be reused.

A. A closer look at the latest version of the xSDK community policies (https://doi.org/10.6084/m9.figshare.13087196.v1) does indeed say that libraries must accept a user provided communicator, and must not directly use MPI_COMM_WORLD. So the xSDK community is thinking the same way you are.

Q. Is DOE committed to funding ECP in the foreseeable future?

A. The Exascale Computing Project is a kind of project that, by DOE rules, has a defined period of performance. Technical work on ECP-supported projects will end by 31 December 2023. There will be a few additional months for the ECP leadership team to prepare for the final project reviews and for the ECP “project office” to finalize payments, gather documentation, and other activities to officially close out the project.

At present, there is not a specific follow-on to the full scope of ECP. As I mentioned toward the end of my presentation, the Next-Generation Scientific Software Technologies (NGSST) initiative from DOE/ASCR is expected to address stewardship issues for some of the ECP software products. But the anticipated budget for NGSST is a small fraction of what ECP’s budget has been, and it is far more narrowly focused than ECP. Generally, the individual ECP subproject teams are in the midst of figuring out how to continue the applications, software products, and lines of research opened by ECP.

There are a lot of discussions around AI, nationally and internationally. Not surprisingly, the DOE is very active in that area already, and there has been talk for several years about a major R&D initiative in the area of AI for science, which some think could be similar in scale to ECP (https://ww2.aip.org/fyi/doe-pitching-major-ai-r-d-initiative-to-congress). But presently, nothing is finalized.

Q. Could you say a little more about the IDEAS community? How many, who, and any trends you’ve captured?

A. The IDEAS-ECP team comprises just a handful of FTEs (full-time equivalents of effort) spread over ~30 people. The ECP project overall has around 1,000 people involved. Beyond that, the DOE scientific software community probably has a few tens of thousands all told. And of course the general scientific software community is still larger.

By being creative about what we do, I think the IDEAS-ECP project has done a pretty good job of reaching out to the broader community, not just ECP or even DOE. A few examples, just looking at some numbers: The HPC-BP webinar series has received more than 12,000 registrations to our 80 events, and the mailing list on which we announce IDEAS-organized events has more than 4,000 subscribers. More than 600 folks have signed up to receive monthly highlights from the BSSw.io resource portal.

As to trends: Over time, we've definitely seen a growing awareness of the importance of software as a tool for scientific research, and increasing recognition of software-focused contributions and of the people behind them. While reproducibility has always been an issue in science in general, it is only relatively recently that *computationally* based science has started to be held to similar standards to experimental and observational sciences, and in my opinion that's also a very important trend in terms of recognizing the importance of software – "Science through computing is, at best, only as credible as the software that produces it." But at the same time, we've also seen enormous growth in the complexity of the models (e.g,. multiscale and multiphysics coupled simulations, often coupled with data from experiments and observations), and the complexity of computer hardware (the rise of GPU accelerators from multiple vendors and the emergence of other types of processors that may become important in the scientific community). So the challenges facing developers of scientific software are also expanding. Hopefully, these challenges will provide additional motivation for applications and software technology teams to pay more attention to their software development practices, but I feel like a lot of people working at the bleeding edge are still bleeding more than they "have to" to get the science out.