Extreme-scale Scientific Software Stack (E4S)
(the slides are available under “Presentation Materials” in the above URL)
Date: January 13, 2021
Presented by: Sameer Shende (University of Oregon) and David Honegger Rogers (Los Alamos National Laboratory)

Q. It’s interesting to see aarch64 being supported. Are there any plans for ARM64-based supercomputers in the US getting unveiled soon?

A. Sandia has one; RIKEN (Fugaku) is not in the US but #1 on Top500. Also ‘Ookami’ a64fx system at SUNY Stony Brook. We plan to support as many platforms as possible.

Q. Which module infrastructure does Spack use? Lmod or environment modules (or both)?

A. Both

Q. I don’t see in the list of packages things like BLAS and LAPACK. Do I miss them or are there any plans to include those as they are typically used at the bottom level of the software stack?

A. There are PLASMA for CPU (OpenMP) and MAGMA for GPU (CUDA, HIP, KNL). See next slide.

Q. For which x86_64 CPU microarchitectures are the binaries provided by the Spack build cache built? Same question for the container images. It seems like these are generic installations, so there must be a significant impact on the performance of these software installations?

A. Custom images are next, for instance Cooper Lake

Q. Follow-up question: so to clarify, currently only generic container images and binaries (via build cache) are available? Is there a list of planned target CPU architectures?

A. We plan to support target CPU and GPU architectures that target the early access systems and planned exascale systems.

Q. Are there any licensing issues with including the CUDA runtime libraries in the E4S container images (and the build cache)? Is anyone allowed to redistribute CUDA, or do you have some agreement with NVIDIA on this?

A. NVIDIA Software License Agreement (SLA) for the HPC SDK section 2A) (iii) states that anything in the REDIST folder, which includes CUDA runtime libraries, can be freely redistributed. See related post on NVIDIA developer forum:
https://forums.developer.nvidia.com/t/deb-package-for-files-in-redist-folder/165690
We also have support for ROCm for AMD GPUs.

Q. Can Spack place the downloaded container content in a shared, environment module cache - or must each user of the given module replicate the downloaded container content in their own environment?

A. Use Shifter or Singularity to achieve this. You can have a single location that is shared with all users.

Q. Are there plans to support Intel GPUs and any parts of the oneAPI software stack?

A. Yes, we're currently testing it. It's in the December 2020 image on Docker Hub.

Q. How is Spack and other containers different from VirtualBox? Can I install OS on Spack?

A. Spack is a package manager, like apt or dnf/yum. So, no. You can use VirtualBox to virtualize a Linux system and use Spack within it.

Q. Plan for builds with more recent compilers? Ie, latest gcc@10.2.0, latest Intel@19.1(?), latest LLVM

A. Yes, with each release of a new version we'll support more compilers (e.g. upcoming v1.3 will support LLVM 11). We plan to support Intel OneAPI 2021.1 compilers in an upcoming release. ROCm 3.9.0 is currently available with HIP compilers that use Clang 11 in E4S v1.2. Also, CUDA 10.2 and 11 compilers are supported in E4S v1.2.

Commentary: The spack upgrade to 0.16 will uncover all sorts of interesting issues. Be aware, if you aren't already. :-O

A. Yes, E4S v1.2 supports our build cache (https://oaciss.uoregon.edu/e4s/inventory.html) that already supports Spack v0.16.0. We are working closely with the Spack team as we transition to the next Spack v0.16.1.

Q. Any plans to support NVIDIA HPC SDK (formerly PGI compilers) too? (PS: they're already available on Cori-GPU @ NERSC)

A. We are evaluating these NVHPC compilers. We have these installed on the Frank system @ U. Oregon and are looking at how many products support these compilers. It may take some time for the configure/build systems of all 67 products to support NVHPC compilers, but this is of interest to us, as is support for DOE LLVM fork.

Q. @David (slide 4?) Can you elaborate on “isolate builds”? In what sense is a given, fixed (tagged) container image subject to variation of dependencies?

A. Our scope is currently to have these workflows run on Summit. Since we do not create a container, we are limited in isolating our builds to what we include in our setup scripts. For this scope, we effectively are limited by the dependencies that are locked in by the spack scripts and git commits/versions we specify in our workflows. In practice, this means that the workflows depend on some aspects of Summit’s software stack, because we are not including everything that a container would. In practice, this is still very useful, because they workflows are being updated through revisions in the ECP products (Ascent, Cinema, Nyx, etc.) *much* more rapidly than the Summit environment is moving.

Q. Is Pantheon considered a CI/CD suite? How easy is it to use, compared to, say, GitHub Actions or Travis CI? Can Pantheon be run in a development workstation too, or only on Summit?

A. No, but it can be used in integration with CI/CD suites. You can try Pantheon on your workstation, but some choices (like shell type) are biased towards Summit.

Q. The list of libs & packages… is there any effort to categorize them by type or offer a kind of reference architecture which covers the aspects of a well-formed ECP-ready apps?

A. Yes, we have an SDK effort that categorizes these libraries by area. You can also find this categorization in the E4S DocPortal [https://e4s-project.github.io/DocPortal.html]. These SDKs include math libraries (xSDK), data and vis, tools, programming models and runtime (PMR), etc.

Please join us for office hours:
E4S testing task force meeting: https://exascaleproject.zoomgov.com/j/1618055149
Friday, Jan 15th, 11am - noon PT. Repeats every two weeks!