

Project Details

Name of project: Improve conda-forge CI infrastructure and packages auto-updating dependencies

Project lead and contact details: Rich Signell (rsignell@usgs.gov)

Project partners and contact details: Filipe Fernandes (ocefpaf@gmail.com), Christopher J. Wright (cjwright4242@gmail.com), Anthony Scopatz (scopatz@gmail.com), Marius van Niekerk (marius.v.niekerk@gmail.com), Michael Sarahan (msarahan@anaconda.com)

Proposed start and end date: 2019-02-1 to 2019-8-1

Budget Requested: \$5,000

Budget Summary:

Continuous Integration services to build the packages (AppVeyor, CircleCI, and Travis-CI) - \$2000.

Promote a code sprint with conda-forge core members to implement the changes in the proposal - \$3000.

Project Outline

Project description:

Packaging fills an important gap (sometimes gulf) between writing software and using software by making it easy for users to install pre-compiled binaries that are complex and costly to build. Building packages across platforms, languages, and userbases is a challenging problem. Conda-forge addresses these issues to create a community resource which is simultaneously: community-driven with high standards, a binary distribution with all the binaries built in the open, welcoming to beginners with support from packaging veterans. Two key motivations for conda-forge are ecosystem self-consistency and shipping the latest software. The former makes certain that disparate tools have interoperability, enabling the sharing of techniques from different software packages, whereas the latter enables researchers to take advantage of cutting edge advancements. The conda-forge ecosystem is powered by combining a simple GitHub workflow with powerful CI tools, distributed package reviews, and heavy automation to provide ease of use and accelerate delivery of the most up to date code. Much of our ecosystem relies on package maintainers to make certain that packages correctly list their dependencies and are consistent with the rest of the ecosystem. This poses challenges as many times maintainers are not aware of updates to their packages or their dependencies. In particular, package Application Binary Interface (ABI) changes can be subtle and especially frustrating to keeping package ecosystems working. This proposal aims to (a) solve these issues by extending our current automation to correctly express the package dependencies in the automated version bump Pull Requests (PR) and (b) create a framework for

automatic rebuilds when a package ABI is changed. This project will reduce human errors, reviewer burden, and streamline package updates. This would impact scientific computing environments and cloud deployment that rely on conda and conda-forge to install their packages by providing a quicker and more dependable package publication.

Thrusts:

Automating package dependency updates

While the automatic version bumping bot is currently able to issue PRs when a new version comes out, it currently does not update the package dependencies. This can cause problems as new releases of software can have new dependencies. We plan on extending the bot to inspect package dependencies by using functionality in two ecosystem tools: “conda skeleton” and “depfinder”. Once any new dependencies are discovered using those tools the bot’s PRs can be modified to include the new dependencies.

Automating ABI rebuilds

Due to a constantly shifting landscape of ABI incompatibilities conda-forge periodically needs to change the pinnings on critical packages to guarantee interoperability of the entire stack. When this happens portions of the stack must be rebuilt so that they use the new binaries. These rebuild efforts have been named “migrations.” Currently setting up these migrations is a manual process. Therefore, as part of this proposed work, the process will be automated so that when pinnings change, the stack can seamlessly be rebuilt to take advantage of the newer binaries. This will require inspection of the dependency graph and extraction of the portions touched by the newly pinned package, and automatically determining if a new pinning has been issued.

Description of key project steps and timeline:

2019-2-1 to 2019-6-30: initial coding of the auto-updating bot

2019-7-1 to 2019-7-31: remote sprint and live sprint at SciPy

2019-8-1 to 2019-8-30: project wrap up and test the new updating bot in production

Outreach

What groups/audiences will be engaged in the project?

The group/audience targeted are conda users using conda-forge as their main provider of packages for their scientific computing environment. Anaconda, Inc., the maintainers of the conda and conda-build open source projects, support this effort and will be involved in development of tools for this effort.

How will you judge that project has had impact?

The expected impacts are faster package availability when a new releases is issued and reduction on errors due to bad dependency specifications.

How will you share the knowledge generated by the project?

We'll author blog posts describing key insights into this effort, including descriptions of the ABI problem in its many faces, as well as any insights we discover or repurpose from efficient navigation of our graph of package dependencies.

Project Partners (as applicable)

Description of project partners (individuals and/or organizations) and their involvement:

How will this project engage members of the ESIP community:

Rich Signell from USGS is an ESIP Partner who will ensure that the advancements of this project will be spread widely throughout the community. Conda-forge not only benefits any ESIP user who uses Python, but anyone who uses Python through the geocommunity. Even beyond Python, conda-forge allows geoscientists to easily install troublesome packages such as gdal and nco, reducing the barriers to effective data access and analysis.

The ESIP JupyterHub instance and other JupyterHub instances, such as those of the Pangeo project, will also benefit, as reliance on the conda-forge channel when constructing Docker containers minimizes the likelihood of package conflicts.

Conda-forge is a community-driven, cross-platform effort to package the libraries we all use, helping to accelerate research by shipping the latest software in a self-consistent system. As the largest conda package ecosystem it relies on a federated system of maintainers to keep all the packages in the ecosystem up to date and consistent with the rest of the ecosystem. This poses challenges as many times maintainers

are not aware of updates to their packages or their dependencies. The two most common issues faced at conda-forge are shifting software dependencies and compatibility breakages caused by updating core libraries that are low in the stack.

Anaconda, Inc., the maintainers of the conda and conda-build open source projects, participate in the conda-forge community. Anaconda will dedicate staff time to developing tooling towards this proposal, as it is of general benefit to all users of the conda and conda-build tools and ecosystem.

