Application: Adding a WebAssembly Toolchain to conda-forge

Wolf Vollprecht - wolf.vollprecht@quantstack.net
EOSS5: Essential Open Source Software for Science (Cycle 5)

Summary

ID: EOSS5-0000000293
Last submitted: Apr 19 2022 07:45 PM (CEST)

1. Applicant Details

Completed - Apr 17 2022

1. Applicant Details

Complete the following information for the Applicant (required)

The information entered should be for the individual submitting the application who will act as the main person responsible for the application and as its point of contact. **To edit your name or email**, navigate to Account Information by clicking your name in the upper right corner.

Name: Wolf Vollprecht

Email: wolf.vollprecht@quantstack.net

Add your home institution, company, or organization. This does not need to be the organization to which a grant would ultimately be awarded, if selected for funding.

Institution/Affiliation	NumFOCUS / QuantStack

2. Proposal Details

Completed - Apr 19 2022

2. Proposal Details

a. Proposal Title: Adding a WebAssembly Toolchain to conda-forge

To edit your proposal title, navigate to the main page; click on the three dots to the right of the application title; and select Rename from the dropdown menu. Proposal title is limited to 60 characters including spaces.

---

b. Amount Requested

Enter requested budget in USD, including indirect costs. This number should be between $100k and $400k over a two year period. Enter whole numbers only (no dollar signs, commas, or cents).

150000

---

c. Proposal Summary/Scope of Work

Provide a short summary of the work being proposed (maximum of 500 words)

WebAssembly is playing an increasingly important role for accessible data science and education. Using tools such as Jupyterlite (the in-browser distribution of JupyterLab), it is very simple to get started with a computational environment running in the web browser sandbox, without having to install anything on the local system. The benefits over more traditional deployments are a fast startup, decent speed, and excellent sandboxing of untrusted code.

A WebAssembly and JupyterLite-based setup already powers interactive consoles on the NumPy.org and Sympy.org websites, serving a ready-to-use computing environment to the millions of monthly visitors of these websites, without having to provide a complex scalable server architecture to provide such environments in the backend. A similar setup underlies the "Capytale" project which is used for teaching Python to hundreds of thousands of high-schoolers in France.

The Python distribution for WebAssembly that currently underlies the Python environment for JupyterLite is Pyodide, which provides a monolithic distribution of several packages of the Python data science ecosystem. As an alternative, we have developed an alternative way to deliver binary packages compiled to WebAssembly in the form of conda packages. These packages are derived from conda-forge recipes

and are fully interoperable with each other. With this grant application, we propose to integrate this conda-based distribution into the conda-forge project.

Our vision for the project is to make it easy for an educator or a researcher to prepare an environment that contains a given set of packages (such as Python, NumPy, or Matplotlib) locally, and then serve that content statically on any file hosting service from which it can be used by students or peers. On the local machine, the dependency resolution will be done using the mamba package manager.

Since the environment is "statically" bundled and hosted on a static file hosting service it needs almost no ongoing maintenance, which enables long-term reproducibility. Researchers will be able to run these bundled environments in the future, even if the underlying host operating systems change drastically, since WebAssembly is designed to be portable, and can run on different operating systems.

We want to make these WebAssembly use cases more accessible to other open-source projects which is possible with the packaging infrastructure provided by the conda-forge project.To turn this into reality, we need to clear the two following roadblocks:

- First, this grant will enable us to build a proper Emscripten package on conda-forge. We will then use Emscripten and the conda-forge build systems to package the most important low-level libraries for WebAssembly, including the Python and R programming languages.
- Second, while Emscripten makes it possible to compile C and C++ code, it does not support Fortran yet, which is used extensively in the open-source scientific stack in packages like Scipy. We will build a robust Fortran toolchain for WASM that will enable a clean build of SciPy and other libraries relying on Fortran.

d. Value to Biomedical Users

Described the expected value the proposed work to the biomedical research community (maximum of 250 words)

Currently, researchers that want to build WebAssembly code have to figure out a viable toolchain themselves – mainly involving Emscripten. When researchers need multiple dependencies (Python, R, or other lower-level libraries) it can quickly become tricky to build them properly, especially for WebAssembly.

We want to give researchers the same simple access to the CI-based build tooling for WebAssembly that they already enjoy with conda-forge for Windows, macOS, and Linux.

Conda-forge closely collaborates with the Bioconda project, which offers over 8000 biomedical and bioinformatics related Conda packages and relies on conda-forge packages as dependencies. Bioconda is the backbone of reproducible package distribution and deployment in bioinformatics with over 140 million downloads in total (http://bioconda.github.io/#stats). Together, we will ensure that the WebAssembly support will be available in Bioconda from the start and thereby enable reproducible biomedical research environments in the browser.

This will be highly useful for teaching, as well as the interactive exploration of biomedical results, for example as a complement to journal publications.

---

e. Open Source Software Projects

Number of software projects are involved in your proposal (maximum of five):

1

Complete the table with the following information for each software project. If there is no homepage URL, re-enter the main code repository URL.

	Software project name	Main code repository URL	Homepage URL
1	conda-forge	https://github.com/conda-forge	https://conda-forge.org

f. Landscape Analysis

Briefly describe the other software tools (either proprietary or open source) that the audience for this proposal primarily uses. How do the software project(s) in this proposal compare to these other tools in terms of user base size, usage, and maturity? How do existing tools and the project(s) in this proposal interact? (maximum of 250 words)

conda-forge and bioconda are the dominant packaging tools in the bioinformatics community.

There is a specialized WebAssembly package management solution called "wapm.io" which is far less featureful compared to conda-forge / bioconda and does not have many of the low-level libraries available as packages. With our solution we would cover many more low-level packages as individual building blocks as well as making many more biology and science specific packages available.

The Emscripten WebAssembly tooling also comes bundled with a couple of "recipes" for lower level libraries that can be used as building blocks for a WebAssembly build of an end-user application. However, these need to be built from source and therefore are not quickly accessible for the general researcher or educator.

There is an exchange of ideas happening between the Pyodide distribution of scientific Python WebAssembly packages and conda-forge. Parts of the Pyodide build system are inspired by conda / conda-forge and the Pyodide maintainers are very happy to see support for WebAssembly making its way back into conda-forge. Pyodide doesn't cover the same amount of packages and does not have the same scale of a community as conda-forge and bioconda.

g. Category

Choose the two categories that best describe the software project(s) audience.

	Category
Category 1	Data management and workflows
Category 2	Bioinformatics

h. Previous CZI Funding

Did you previously apply for funding for this or a related proposal under the CZI EOSS program?	No
Have you previously received funding for this proposal under the CZI EOSS program?	No

3. Equal Opportunity & Diversity

Completed - Apr 17 2022

Equal Opportunity & Diversity

CZI Science supports the science and technology that will make it possible to cure, prevent, or manage all diseases by the end of this century. Everyone is affected by disease, yet different communities are affected by or experience disease in different ways. Moreover, due to systemic barriers, the scientific enterprise itself is not a place where all voices and talents thrive. We believe the strongest scientific teams — encompassing ourselves, our grantees, and our partners — incorporate a wide range of backgrounds, lived experiences, and perspectives that guide them to the most important unsolved problems. To enable our work, we incorporate diverse perspectives into our strategy and processes, and we also seek to empower community partners to engage in science.

We request demographic information associated with applications submitted to CZI in response to our open calls. This information helps us learn from the RFA process, as well as improve our strategies to help ensure members of underrepresented or marginalized groups in science are aware of and able to apply to CZI opportunities. **Please note that answering all questions below is voluntary, and demographic information will not be used to make final grant funding decisions.** All responses will be shared