**Title:** Simplifying Reproducibility
**Topic:** Compiling; Operating systems; Software Engineering
**City and country:** Menlo Park, California, Unites States of America
**Team or project in lab:** Object Culling and Concretization for Assurance Maximization (OCCAM)
**Name, contact of advisor:** Ashish Gehani (ashish.gehani@sri.com), `http://www.csl.sri.com/~gehani`
**Name, contact of head of lab:** Patrick Lincoln, `http://www.csl.sri.com/~lincoln`

**General presentation of topic:**

Hypothesis testing and reproducibility have been cornerstones of the scientific method for hundreds of years. Modern scientists depend heavily on software-based analysis of their data when drawing conclusions about experiments. In order to allow others to explore their data and to facilitate validation of empirical results, scientists often share their software with other members of the community.

Increasingly, an entire virtual machine is published to ensure that the recipient does not have to replicate the compute environment, retrieve data and code dependencies, or invest effort into configuring the system. However, this approach scales poorly with the growth in size of the included data sets, the extraneous functionality in applications that utilize versatile software libraries, and the irrelevant code in stock operating system distributions. This research investigates principled approaches for software and data distribution, utilization, and debloating.

**Objective of the internship:**

Computational aspects of scientific experiments have been growing steadily. This brings a need to be able to reproduce such results [3]. Science is also increasingly performed by exploring diverse sets of data. Unsurprisingly, there is a demand for being able to easily repeat the numerous transformations performed. By simplifying the process of releasing usable code and data, the tools developed will induce more scientists to share their software.

The project will focus on the design, development, and evaluation of components in a toolchain that allows scientists to transform their software into specialized applications with all the necessary environmental conditions and portions of required data sets built directly into the code. It will explore the use of techniques such as *partial evaluation* of code with respect to static deployment configurations [2] and *application containerization* to create easily distributed software appliances [1].

Software packaged with tools from this project will allow scientists to publish their programs in a form that can be utilized by others with minimal effort. By eliminating many of the challenges of building, configuring, and running software, it will allow members of the scientific community to more easily reproduce each others' computational results.

**Bibliographic references:**

[1] Carl Boettiger, **An introduction to Docker for reproducible research**, **ACM Operating Systems Review**, Vol. 49(1), 2015.
[2] Gregory Malecha, Ashish Gehani, and Natarajan Shankar, **Automated Software Winnowing**, *30th ACM Symposium on Applied Computing*, 2015.
[3] Victoria Stodden and Sheila Miguez, **Best Practices for Computational Science**, *Journal of Open Research Software*, Vol. 2(1), 2014.

**Expected ability of the student:** Since the project is significant in scope, there are multiple internship opportunities. In particular, students with a background or interest in compilers, operating systems, or software engineering will find rewarding problems to collaborate on.