**SciPaaS: a Python-based execution platform middleware for**

**running scientific applications in the cloud**

Wesley Brewer, Ph.D.

**Abstract**

SciPaaS is an execution platform middleware designed to make it easy for scientists to rapidly deploy their scientific applications (apps) to the cloud. It provides all the necessary infrastructure for running typical IXP (Input-eXecute-Plot) style apps, including: web interface, post-processing and plotting capabilities, job scheduling, real-time monitoring of running jobs, and even a file/case manager. In this paper, first the system architecture is described, then three example apps are demonstrated: (1) Mendel’s Accountant, a forward-time population genetics simulation model, (2) FluTE, a stochastic influenza epidemic simulation model, and (3) Cav2dbl, a computational fluid dynamics code for predicting sheet cavitation on a hydrofoil. Finally, implementation issues with running on Amazon EC2, OpenShift, and Google Compute Engine (GCE) are discussed. Future work involves building the parallel execution substructure, possible integration with Opal2, and also allowing users to define their own workflows.

**Introduction**

With the promise of on-demand computing access, cloud computing has become an invaluable resource for computational scientists. The only problem is that most computational scientists that I know have little knowledge about the cloud, and much less about how to possible get their app running in the cloud. The concept of SciPaaS is that a scientist could easily create a zip archive of their code, upload it to the cloud, and SciPaaS would manage all the infrastructure for them, including the input interface, job scheduling, plotting, etc. , leaving them the main responsibility of just writing good code that takes some numerical inputs and produces some numerical outputs.

There has been a number of software packages written over the past few years to address the need of being able to run scientific applications in the cloud. Wu et al. (2010) developed a scientific application framework based on OpenSocial gadgets. Unfortunately, the code is not open source, so cannot be freely downloaded and used. Krishnan et al. (2010) developed Opal2, a toolkit basically which can be used to wrap scientific applications and expose them as web services. Opal2 also provides plugin integration with EC2 and Hadoop. Hadoop provides much of the backend infrastructure for running applications, but relies on other software such as Kepler for pre-processing, and other codes for post-processing. Essentially, there was no package I could find that I could simply upload my app and start to run on the cloud. Furthermore, many of the codes available have become quite large and sophisticated, and have rather steep learning curves.

**System Architecture**

**References**

1. Wu, W., Uram, T., Wilde, M., Herald, M., and Papka, M. “A Web 2.0-Based Scientific Application Framework”, IEEE 2010.
2. Krishnan, S., Clementi, L., Ren, J., Papadopoulos, P., and Li, W. Design and Evalution of Opal2: A Tookit for Scientific Software as a Service. 2010