University of Notre Dame
Research Experience for Undergraduates
Summer 2014 REU
Project Summary

Name: Natalie Sanders
School: University of Notre Dame
Major: Computer Science
Current Classification: Junior
Mentor: Alexander Vyushkov

Project title: Windows Azure as a Platform for Malaria Modeling

- **New skills acquired during summer research**

  - While at the CRC, I’ve had the opportunity to learn many new concepts. Among these was learning how to manipulate Windows Azure resources. Not only did this entail familiarizing myself with the Azure service management API for Python, but I’ve also learned basic networking concepts, such as certificate authentication. This led to more learning about cloud computing in general and how it differs from other technologies like grid computing. I’ve also been able to strengthen my skills in writing Python scripts. I’m now much more aware of different Python libraries and Python syntax is now second nature. Before this summer I had only worked in Linux environments. Working with Windows OS for this project, I’m now more comfortable with Windows style commands. In addition, this project has given me the opportunity to discover what research in computer science is and how it is conducted. Never having conducted research before, I was unaware of the nature of computer science research. After finishing my REU this summer, I can say that I have thoroughly enjoyed my experience.

- **Project Summary**

  My project this summer addressed the task of running malaria transmission simulations in the cloud using the Windows Azure cloud platform. A simulation, depending on its size, may take several hours to complete. If the models are run on a personal computer, any updates or unexpected restart could cause the user to lose all of their results. By running the models in the cloud, the user is guaranteed that their model will run uninterrupted and can retrieve their results when convenient, without keeping their own computer running continuously for hours.

  To bring the simulation into the cloud, I have written two Python scripts—one that the user interacts with to send jobs to the cloud and receive results and the other which runs on the virtual machine side that
executes the actual simulation. The first script, also called the host script, offers the user command line access or a simple user interface. A user will create an account or login if one has already been created. Each account provides the user with their own uniquely named cloud service on the Azure PaaS where VMs will be generated for simulations. In addition, each account is granted a storage container in the Azure cloud to which the user will upload inputs. After connecting to their account, the user is able to deploy one of three models—EMOD, Open Malaria, or a mock model (which simply returns the inputs as a test). When choosing to run a simulation, the user will assign the job with a project name and give the host script the path to their inputs. The inputs will then be zipped and uploaded to the user’s storage account on the Azure platform. In order to run the simulation, the host script will then generate an appropriate VM for the specified model type. The user may check if their model has finished by specifying a project name. If the results are ready, they will be downloaded from the cloud and saved to a project folder on the user’s computer. Finally, if the user wishes to delete their account, the script will delete their storage account and their cloud service.

The second script, referred to as the VM script, runs on the virtual side. There are three VM script variations, each corresponding to a different model, and thus, three operating system images that are exclusively used to generate VMs. The VM script is configured as a startup task on the virtual machines such that the script will automatically run after a VM is spun up. The VM script first downloads the archived input specific to the simulation project and extracts all files. Using these inputs, the script then runs the transmission model. After zipping all the results and a text file containing the standard output, the zipped results are uploaded to the user’s storage container. Finally, the script deletes the VM on which it is running to conserve cores.

- **Publications (papers/posters/presentations):**
  - Poster presentation at the Summer Undergraduate Research Symposium (SURS) at the University of Notre Dame on August 1, 2014