Systematic Testing of Distributed and Multi-Threaded Systems at Scale

Jiri Simsa (student), Randy Bryant (advisor), Garth Gibson (advisor)

Project Description

The broad goal of the Systematic Testing of Distributed and Multi-Threaded Systems at Scale project is to qualitatively change the process and experience of developing long-running, stateful, highly concurrent information systems for the largest scales. With the advent of the multi-core era exploding the concurrency in large systems, developers of todays large-scale systems lose confidence in the traditional stress testing techniques primarily used and resort to slower, smaller changes in code. A transformation in large-scale systems development is possible if concurrent software testing technologies can automate discovery of complex, deeply interacting multi-actor bugs. This project directly tackles the application of systematic testing technologies, using prototype tools, to real, large-scale, concurrent code such as high-performance computing storage systems and cloud key-value storage systems. Using:

1. virtual machine technology to assist in the directed dynamic execution of large numbers of stateful threads,
2. large scale computing and state space reduction techniques to manage the combinatorial explosion in sequenced interactions, and
3. automated instrumentation of interactions achievable by interposition in key communications and scheduling libraries,

this project will evaluate the potential to transform practical large-scale systems development by finding bugs in real codes in development and use.

Project Context

eScience Studying evolution of galaxies, modelling protein folding, simulating impacts of earthquakes and nuclear explosions, mapping the surface of oceans, discovering cures for terminal diseases, are all examples of areas in which research has been greatly advanced by the ability to generate and process ever increasing amount of experimental data. An ongoing project we are actively involved in aims to equip researchers from other sciences with large data sets and tools to analyze the data using the OpenCloud iniciative and Hadoop platform. This proposal complements this effort by building infrastructure for correct and rapid development of tools to support eScience and applying the systematic techniques to new code going into the Apache HDFS project.

Computational thinking for all Systematic evaluation of complex concurrent systems is a prime example of computational thinking. It simultaneously attacks a hard problem with the brute force of parallel processing, with the rigor of formal methods, and with the elegance of systems techniques. It weights the cost of using virtual machines against the control their provide. It trades off the space needed to store explored states against the time needed to reconstruct these states when needed. It bends time and space continuum by simulating time outs and failures. Ultimately, systematic evaluation of complex concurrent system serves not only as a technique for finding and

diagnosing complex bugs, but also as an educational tool (15-213) for students to better understand concurrency.

Collaboration with MSR

Ongoing Collaborations In past we have estabilished a working relationship with the programming languages group at MSR Cambridge in England. In particular, we have collaborated with Byron Cook and Satnam Singh on synthesizing hardware from C programs.

Prospective Collaborations The following research projects and groups represent prospective MSR collaborations:

- Dryad and DryadLINQ – The project is investigating programming models for writing parallel and distributed programs to scale from a small cluster to a large data-center. We recently interacted Yuan Yu, DryadLINQ project lead, and discussed our shared interest in designing a large-scale model checker.
- Software Reliability Research – A group lead by Thomas Ball that studies how program analysis, program verification and software measurement techniques can be used to improve the quality of software. We have had research conversation with several individuals from this group, including Patrice Godefroid, Shaz Qadeer, and Shuvendu Lahiri.
- Rigorous Software Engineering – A group lead by Sriram Rajamani that works on tools, languages and methodologies to dramatically increase the productivity of software development.

Budget Information

This proposal requests support for 1 student for 2 years. No faculty support is requested. Equipment support for the project has already been acquired from other sources. The student for which the support is requested is not currently on E&GO. The student had been funded by a DOE grant, which recently ended.

Note: We have submitted an NSF proposal which, if funded, could reduce the budgetary needs of this proposal. Nevertheless, we would still like this proposal to be an opportunity for us to spur new collaborations with MSR.