High Performance Computing (HPC), which has become the fundamental part of scientific investigation and discovery, faces the affliction of hard and soft faults that causing errors in computer systems. Soft faults, in particular, are not systematically reproducible and therefore propose a big challenge to application’s fault tolerance capabilities. Common soft faults, like a bit-flip caused by a charged particle, can happen to a large scale HPC system on a daily basis. [1] Consequently, Fault Injectors are designed to simulate manifestation of soft faults and help determine their impacts on HPC applications. Flipit is a fault injector structured as an LLVM compiler pass. It instruments code allowing the result of an instruction to be incorrect with a certain probability [2].

Flipit, despite its improvement over existing fault injectors, can bring considerable overhead in the compiling and running the application, thus making these already time-consuming procedures even worse. Therefore, our main focus is optimizing the fault injection framework Flipit, improving its compile-time and run-time performance.

We analyzed the implementations of various synchronization mechanisms, and integrated POSIX file lock into FLIPIT, therefore enabling it to compile multiple source code simultaneously, and further improved its compile-time performances. (Graph-1 below)

Profiled Flipit running-time performance to identify the bottlenecks of existing C/C++ functions. By restructuring fault injection and activation mechanisms, shrinking number of arguments for some C functions, trimming unnecessary methods and branches, we were able to reduce roughly 35% of original code, in the meantime improve Flipit’s run-time performance. (Graph-2 below)

Redefined variables definitions, employed more standardized POSIX & C library utilities, which make Flipit more portable on various platforms and easier to maintain in the future.

There is more to do: We are planning to include support for multi-thread applications in Flipit, and undergo more thorough test on a wider range of platforms in order to gain more insights about Flipit’s stability and performances.

A Big Thank You To:
I would like to thank my mentor, Jon Calhoun, for giving me this chance to work on this fascinating project; without his support I wouldn’t have been able to finish our goals. Some information displayed was adopted from:

