Project Title: saveCHIMP (save CHIp Multi Processor)

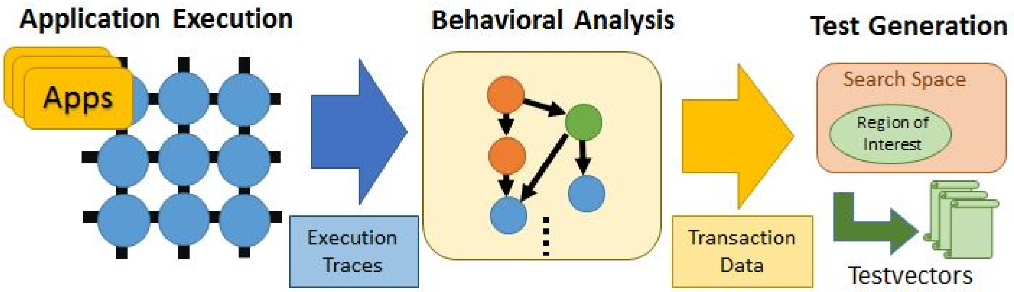
Team Name: DAT

Team Members: Arjun Khurana, Dong-hyeon Park, Timothy Wong

Idea/solution to be investigated by the project

We want a framework to do the following:

1. Given a program/software expected to run on the system, identify the characteristic transactions and behaviors that are exercised by the program.
2. Represent the transactions/behaviors in a model that can help us identify a search space.
3. Generate several series of targeted tests for the particular search space we are interested in. We will focus our effort in this project on applying our idea on a homogeneous system (e.g. a network of CPUs).



Details on Behavioral Analysis:

1. Given the trace, use a hash/mapping function to quantify the instruction or network state to a single value/index. The function needs to be able to capture the information we are interested in (type of memory operation, address, destination address) while not losing locality.
2. By using this function, we breakdown the trace files to a sequence of states.
3. Create a transition table of the the states for the each segment, to represent the behavior.

Progress so Far

Since last checkpoint we have:

- We completed a prototype workflow for test case generation. First we obtained the characteristic patterns from the execution of benchmarks, then we represented the patterns using memory states by hashing, and finally we generate random test cases based on the memory states in different CPUs.
- We injected a preliminary bug model into our 4x4 system to test our bug detection. In our early result, the original benchmark detects the bug in 32 seconds, while our testframe detects the bug in mere 6 seconds, which is a 5x reduction in bug detection time.

Issues/Showstoppers

Main challenges we are currently facing:

- Some of the benchmarks we are running are extremely long, which delays our data collection.
- Our bug model may be a bit too premature.