**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).
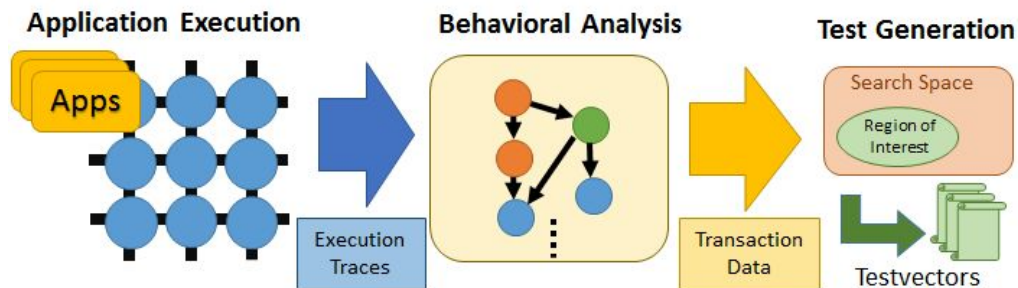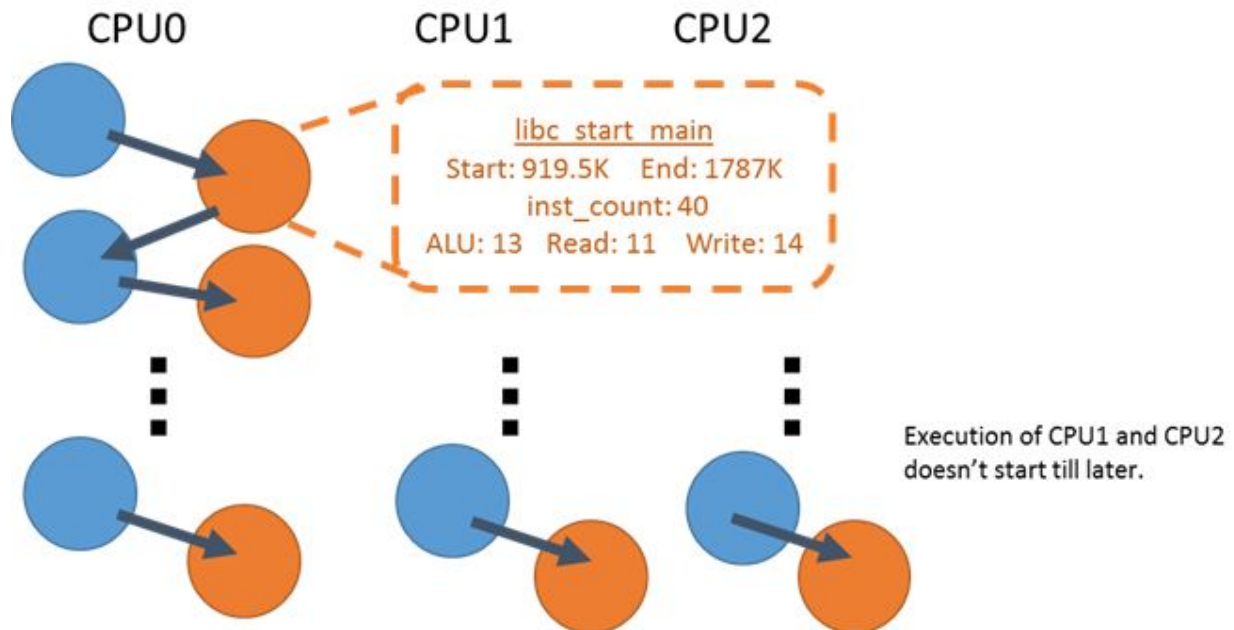

Progress so Far

So far we accomplished:

- Configured a ARM system with directory-based MOESI cache coherency on a 4x4 Mesh network on Gem5 simulator.
- Generated instruction and network traces of the system running Parsec Benchmark in system emulation mode.
- Extracted instruction flow of each CPU to model the execution behavior. We identified the functional execution blocks based on the instruction stream. We created data vectors based on: start and end cycle, label of the execution segment, number of instructions executed, number of read, write and alu operations, and the cpu ID.
- Sample data:

- Sample Behavior Extraction:

CPU0   CPU1   CPU2

libc_start_main
Start: 919.5K   End: 1787K
inst_count: 40
ALU: 13   Read: 11   Write: 14

Execution of CPU1 and CPU2
doesn't start till later.

Issues/Showstoppers
Main challenges we are currently facing:
- It is difficult to extract network packet traces from the debug outputs we get from gem5. We are working on tweaking the gem5 source code to extract the information directly.
- The size of the debug and trace files are extremely large and slowing down our analysis. We are working with a highly condensed dataset, but even then the file size is extremely large.
- Some of the parsec benchmarks are having trouble running on gem5 due to problems in cross-compiling with ARM ISA.