Motivation

- Quick prototyping, testing and evaluation of variability aware software.
- Use of variability models from expedition experiments and literature to generate variations in a virtual execution environment.
- Lesser cost for development, compared to hardware testbed.
- Faster Prototyping.
- Allows development of many test cases and tests that are more robust and with better coverage.

Variability Model

- **General Architecture**: The state of the virtual machine is used as input to variability models, which return variations that should be pushed into the virtual execution environment.
- **Power Model**: Determines expected power of the various components in the VM based on instance, temperature, state, and other parameters.
- **Frequency Model**: Adjusts the real-time clock in the VM to account for variations in clock frequencies.
- **Error Model**: Encapsulates errors like bit flipping in registers and memory errors.

VMM Development Plan

- QEMU as the virtual machine monitor: Open Source, Several architectures supported.
- VM state is used as input to the variability model.
- Virtual variability sensors provide variability information to guest OS and change state.

Guest OS Development Plan

- Drivers probe variability information from virtual variability sensors.
- Guest OS provides variability information to applications, or uses information to drive adaptation strategies.