2. Introduction

2.1 Requirements & Constraints

Physical / Resource Requirements

- Our project needs to be developed on an ARMv8 processor subsystem to closely resemble what is found on avionics platforms
- To accommodate the primary users available design space (eg. a Boeing avionics engineer) a single-board computer form factor is required (Raspberry Pi, Pine64 family, etc.)
- Toolset must employ a type 1 hypervisor (Xen) to partition underlying system resources

User Experiential Requirements

- Our project requires users to possess a moderate level of Linux environment knowledge to correctly run the tool suite from the software command line
- A knowledge of worst-case execution time and its influencing factors is imperative to know what our eventual interference generating tools reports
- A familiarity with multi-core computer architectures is heavily encouraged (this is somewhat assumed however as our primary user will be an avionics engineer specializing in multi-core computer systems)
- Our team will provide enough documentation with a sufficient level of detail to allow the user to learn any of the above at a high level

Functional/technical requirements

- Toolset must thoroughly and methodically stress the system in a reproducible way
- Toolset must focus on major points of resource contention (processor time, memory usage, IO bus usage, etc.)
- Accurately produce potential worst-case scenarios (i.e., a rogue process uses too much CPU time/memory/IO bandwidth)
- Toolset must collect and analyze performance data to demonstrate an upper bound on worst-case execution time for our platform

UI Requirements

- Develop a well-documented command line tool to interface with our design
- Ensure a proper command-line utility for automated testing
- Provide a user-friendly GUI for managing and interpreting test results by less technical users
- Ensure the same functionality exists in the GUI and command line tools

2.2 Engineering Standards

- FAA: AC 20-193
    - This standard is defined by the U.S. Department of Transportation. It is concerned with the use of multi-core processors in avionics systems. Our design is directly applicable to this area, hence its inclusion.
- IEEE Code of Ethics
    - While this standard applies to any engineering effort, our design must ensure the public's safety. Our design provides critical information to systems whose failure could lead to severe injury or death.
- CAST-32A
    - This document outlines the aspects of multi-core systems of concern to the safety and performance of avionics systems. This outline directly relates to the scope of our current project: multi-core interference mitigation to improve system safety & performance.

- SAE Aerospace Standards
  - These standards define the safety and reliability of various aspects of avionics systems. Our design will stress test multi-core systems that will support avionics systems like controls and communications.
- RCTA/DO-178C
  - This standard is concerned with the quality of software used in avionics systems. It defines a safety assessment process that categorizes software into five tiers of criticality. Our design must adequately characterize a hardware platform to assess the criticality of a software fault or failure.
- ASTM (American Society for Testing and Standards)
  - ASTM publishes technical standards that are directly applicable to many engineering efforts, including avionics. As our project requires us to develop a compliant platform utilizing both FAA standards, as well as any other surrounding engineering & aviation standards, its imperative that we reference ASTM for the applicable data.
- POSIX (Portable Operating System Interface)
  - POSIX defines a set of standards that ensure combability between operating systems. Our design will be a part of a larger set of software tools and systems. It should therefore be able to interface with these tools in a standardized way.
- ARINC 653
  - This standard is concerned with the space and time partitioning of safety critical avionics systems. Our design must be able to isolate and test distinct aspects of a multi-core system, like CPU usage, memory, and bus traffic to inform the decisions made by this standard.
- FACE (Future Airborne Capability Environment)
  - This standard defines an avionics environment for military airborne platforms. It is concerned with making real-time safety-critical computing applications more robust, portable, and secure. This is the end goal of our design.