1. Introduction

1.1. Problem Statement

Avionics systems are responsible for controlling many safety-critical aspects of modern airplanes, such as engine management modules and flight envelope protections. As these systems become increasingly computerized, the amount of input data they must consume and act upon becomes progressively larger, and the need for more advanced hardware to base the systems on has become apparent.

One method of solving this issue is to use a multicore processor, which, in essence, allows for parallel processing of data (or multiple distinct running programs) on one computer system. However, when multiple processes are being executed on a multi-core processor, it is possible that contention for shared device resources (such as memory or I/O (Input/Output) bandwidth) between the processes can introduce an undesirable amount of performance degradation in one or several of the programs running on the system.

With this consideration, our client Boeing has recognized the need for a method of quantitatively identifying a worst-case execution time (WCET) in a scenario in which resource contention is extreme. This is where our project comes into play. Described as a MOAT or Multicore Operational Analysis Tool, our software allows the user to input system stressors onto a certain aera of the platform; I/O, cache, main memory, etc. Once the user knows what effects stress plays into the system, mitigation methods to prevent catastrophic events can be applied to the contention points. In addition to the performance benefits this tool brings, a multicore stress platform is essential for avionics certification; it provides a definitive way to prove that performance degradation is within an acceptable bound, or that mitigations can be applied to move the execution time inside of that bound.

Overall, the desired outcome of this project is a stress-testing platform that will allow us to pinpoint areas of resource contention, apply stress to those points, and apply relevant performance mitigations to the system to arrive at a WCET for our reference hardware platform.

1.2. Intended Users

When designing our empathy maps, we were able to identify several key stakeholders/users directly affected by our design project. This includes the Boeing avionics development team working hands on with the multicore stress testing platform, the Boeing team managers in charge of a technical avionics testing team and ensuring regulatory compliance, as well as Boeing customers purchasing a designed/tested product using this multicore stress test tool.

Looking at the user personas for each group, we can see that the avionics development team can be described as a group that manages Linux avionics development and validation for Boeing aircraft. They would need a system stress testing tool developed on an ARM-based platform with resource testing deliverables like execution time, system temperatures, etc. easily accessible. As this user group is directly involved in the testing of in-development avionics, a stress test tool is of the utmost value to the team.

Looking at another user group, the Boeing managers, although not needing a stress testing tool for the success of their role as a manager, they would need a stress tool easily accessible to their project engineers such that they can ensure their systems can be flight-certified under civilian or military authority. Again, as stated by the development team, this tool ensures the success of their in-development product, making the value extremely high for the company. When considering this user group from an economic standpoint, the Boeing manager role consists of client-to-team communication (e.g. economic considerations). Although less important than the safety considerations a product such as this brings up, the economic factors surrounding a stress tool are still important.

Looking at the last user group described, the product customer, although less involved with the testing suite than the avionics development team, wants to purchase the safest product available. This tool ensures independence from core-to-core interference when running multi-core applications, making the product safer as a result. A customer in this context can be closely described as an airline company wishing to purchase a new Boeing airliner to utilize on commercial flights. Their operation relies heavily on the FAA (Federal Aviation Administration) approval of the aircraft. Therefore, this analysis tool is crucial to the success of their purchase.