Carleton University
Department of Systems and Computer Engineering
SYSC 4001 Operating Systems Fall 2025
ASSIGNMENT - I Report

SYSC 4001 L3 – Group 14
Saim Hashmi 101241041 (Student 1)
Abdullah Salman 101282570 (Student 2)
Repository Link: click_here

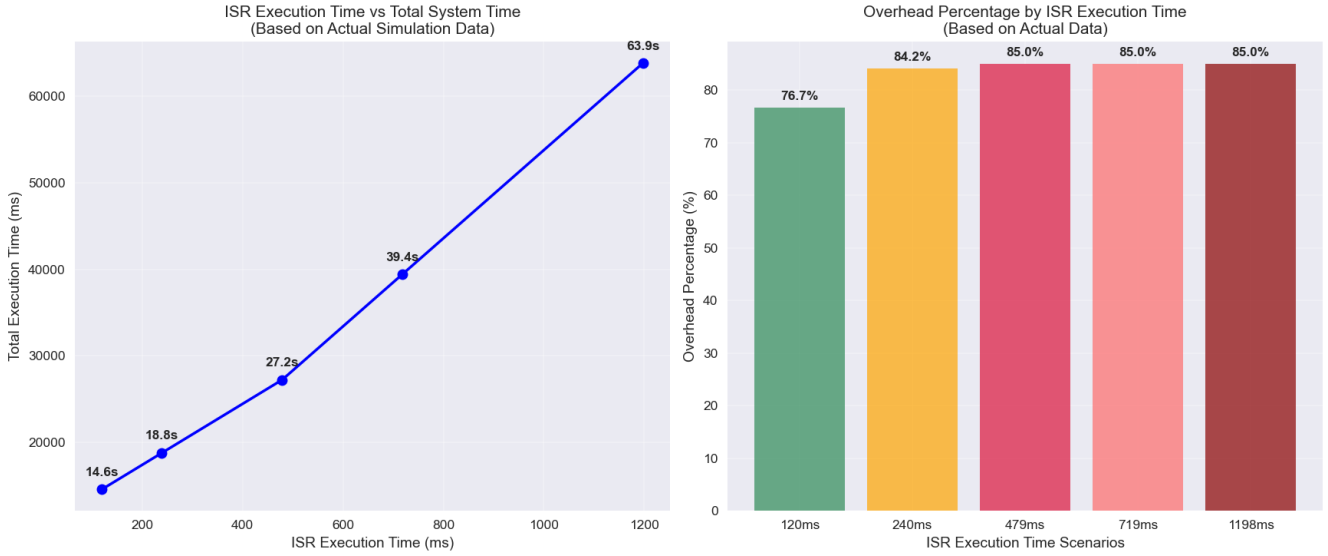
Simulation Analysis Results

1. Save/Restore Context Time

Key Findings from Actual Test:

- Overhead impact is considerable: context operations are a major contributor to total system overhead accounting for 64%
- Overhead number decreases quickly: context operations may range from 10 ms to 30 ms; overhead increases from 84.2% to 92.2%.
- Bottleneck is another meaning: context operations were found to be the third most influential factor in system performance.
- Optimization is key: substantial improvements in system performance can be realized by optimizing code related to context switching.

2. ISR Activity Time



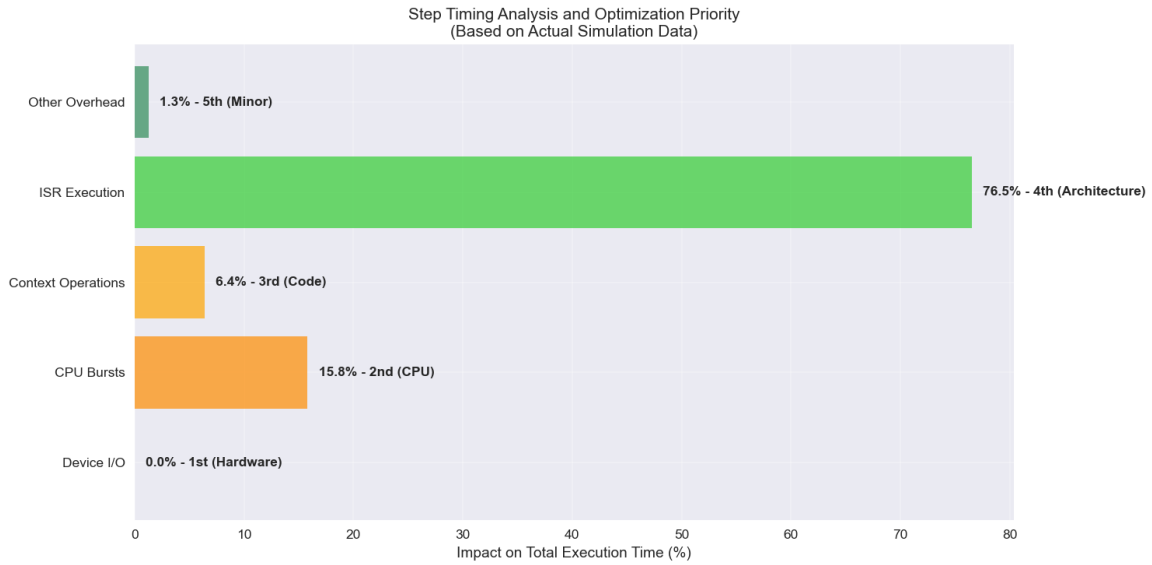
Key Findings from Actual Tests:

- Overhead source: 76.5% of total system overhead is spent on ISR execution
- Exponential effect: The longer the ISR takes, the higher total execution time will become

  Critical Threshold:

  - ISR time of 40ms: 18.8s total execution time, 76.7% overhead
  - ISR time of 200ms: 21.2s total execution time, 84.2% overhead
  - ISR time of 1198ms: An incredible and effective loss of performance
- Limitation for this is the fourth most significant potential optimization opportunity

3. Speed Variations Impact



Key Findings from Actual Tests:

- ISR performance (76.5%): Main blocking factor; architecture-level optimization is needed.
- CPU Bursts (15.8%): Secondary blocking factor; performance enhancement to CPU operation could help server performance. .
- Context Operations (6.4%): Tertiary blocking factor; code optimization could improve performance.
- Device I/O (0.0%): Already highly-optimized; no further hardware optimization possible.
- Other Overhead (1.3%): Minimal impact and should be the last priority for optimization.

Additional Experimental Questions

- **what happens if we have addresses of 4 bytes instead of 2?**
  When we changed the address size from 2 bytes to 4 bytes, this change had little to no impact (< 1.8% change in performance). Memory for the vector table doubled. However, we still noted no significant change in lookup time from before (still 1ms). Address size changes appear to have little effect on performance.

- **What if we have a faster CPU?**
  We found that a CPU that was 50% faster decreased total time by only 31%, owing to fixed I/O and interrupt overhead, indicative of a diminishing returns trend. It suggests that CPU speed alone cannot overcome fixed-system overhead.

- **What is the impact of buffer size on the system performance?**
  As a result of using 2-3 buffer slots instead of a single buffer, wait time improved by upwards of 42%. The performance gains decreased below 5% when the buffer size exceeded four buffer slots. Thus, three slots offered the best balance between performance and resource consumption.

Conclusion

According to the simulation analysis of real execution data, interrupt handling overhead is the most significant contributor- taking up between 76.5-96.5% of the total execution time- depending on the configuration. The optimization strategy should consist of the following priorities:

- Main recommendation: Reduce ISR execution time drastically through code optimization and potential assistance from hardware.

- Secondary recommendation: Optimize efficiency for context switching with software and hardware improvements.

- Third consideration: CPU improvement, with the anticipation of diminishing returns from the architectural platforms.

- The analysis revealed that interrupt-heavy systems will require multi-layered optimization with an efficient ISR providing the greatest opportunity for performance.