SYSC 4001 Assignment 1

Group Submission

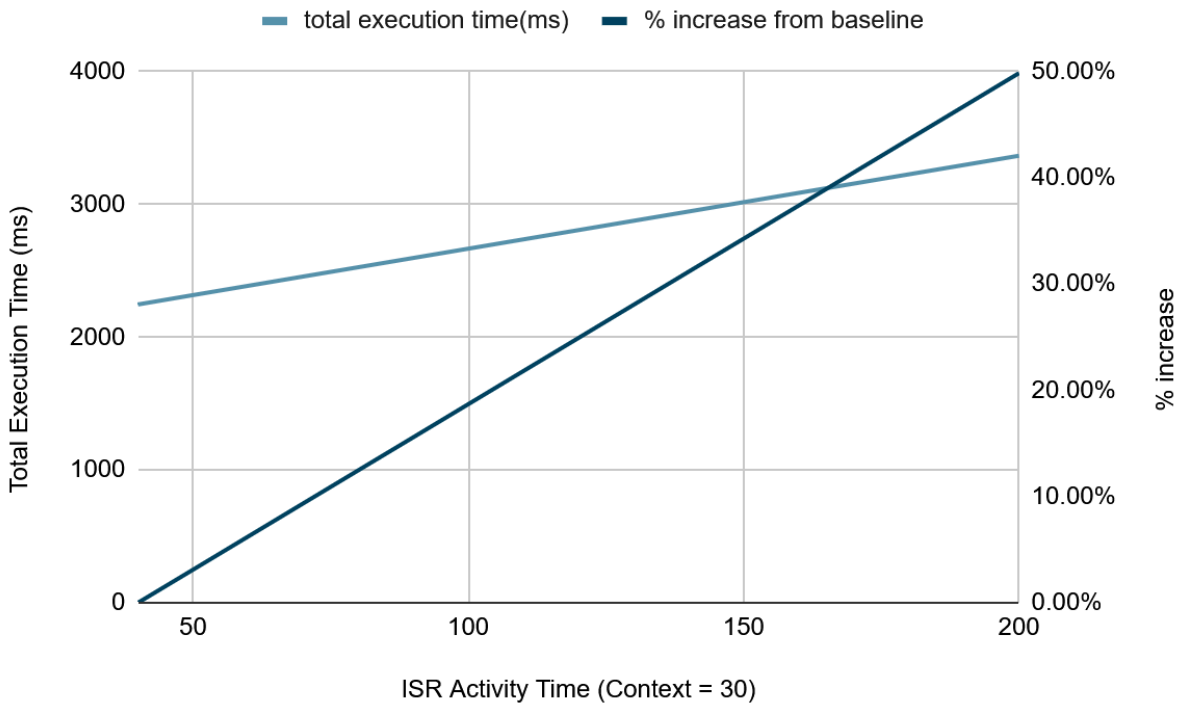
Jason Huang 101265573 (Student 1)
Lawrence Chen 101303243 (Student 2)
Github Repo: https://github.com/Dashx2/SYSC4001_A1

Part 2:

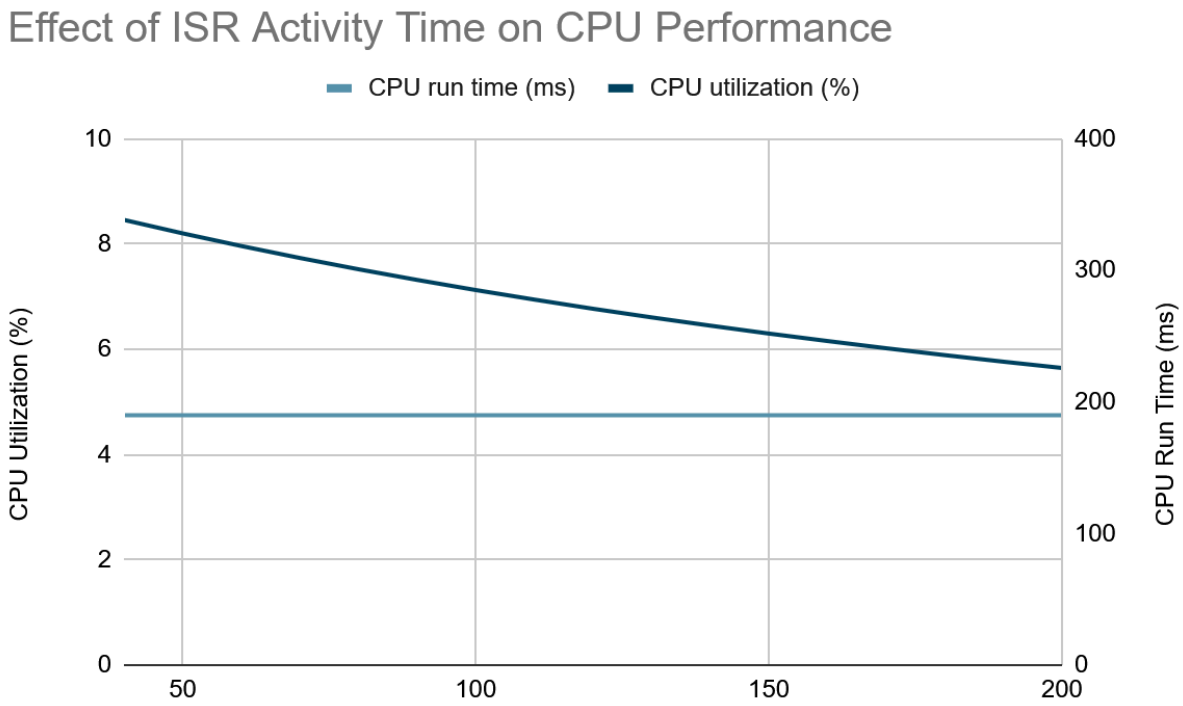
Github Repo: https://github.com/Dashx2/SYSC4001_A1

1. **Change the value of the save/restore context time from 10, to 20, to 30ms. What do you observe?**
   When the context time is increased by 10ms to 20ms to 30ms, the total time of the execution also increases from 2166 to 2206 to 2246. This lines up with the trace.txt where everytime context is saved, it would take 10ms more for each execution. Where 30ms is 3.56% slower, and 20ms is 1.81% slower

2. **Vary the ISR activity time from between 40 and 200, what happens when the ISR execution takes too long?**
   When the ISR execution is increased, the entire execution time increases at a steady rate due to the ISR being run for longer, thus taking up more of the CPU's time while waiting on the ISR to finish. Originally starting at 2246ms with an ISR of 40ms and then increasing to 3366ms when the ISR gets increased to 200ms. This is a 49.87% increase in processing time.



The result that occurs to when the ISR takes longer to complete its process is that the CPU's total utilization goes down, which mean it spends more time doing no calculations and slowly degrading without recouping the operating costs, as the Utilization goes from 8% usage of the total time to only ~5% usage when the ISR increases



3. **How does the difference in speed of these steps affect the overall execution time of the process? You can process this data using a Python script, Excel spreadsheet or any other tools for separating the overhead from the actual work of your program (i.e., CPU use for processing and actual I/O needed by the program).**

4. **Ask yourselves other interesting questions and try to answer them through simulations. For instance: what happens if we have addresses of 4 bytes instead of 2? What if we have a faster CPU**
   If the Addresses were 4 bytes instead of 2 then this error:

   “terminate called after throwing an instance of 'std::out_of_range'
   what(): vector::_M_range_check: __n (which is 1401) >= this->size() (which is 26)
   [1] 65734 IOT instruction (core dumped) ./bin/interrupts ./inputs/trace.txt vector_table.txt device_table.txt”

   is printed which means that an address that does not exist on the vector table was attempted to be accessed and would need more vector values added in order to accommodate the larger address size

If the CPU is faster, then the time it takes to execute does marginally go down linearly, but the bulk of the time is dedicated to processing the interrupt and waiting on the slower I/O to finish its operation first. Although, slowing down the CPU still slows down the entire process as the CPU is now working slower.