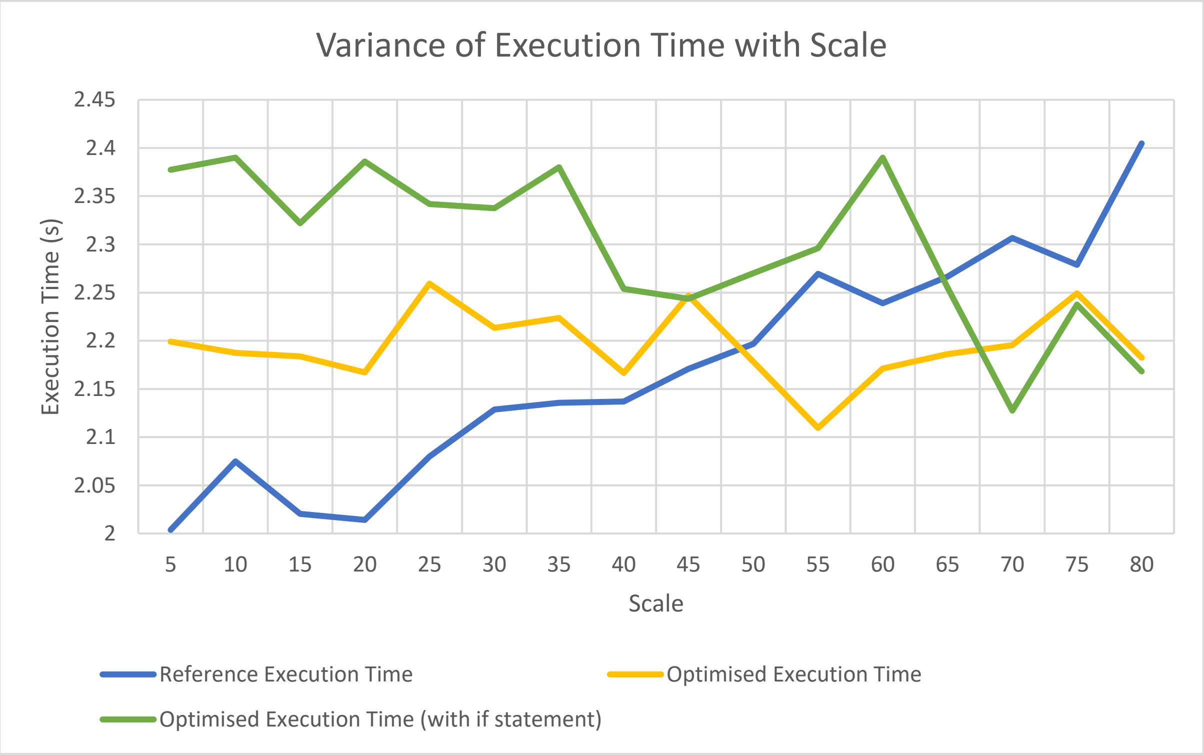
Is it better to use the reference implementation at low scales due to the overheads of TBB and OpenCL?

For the integral puzzle, we found the reference implementation performed better in terms of execution time, compared to our optimised solution, for low scale values, and therefore considered whether it would be better to use the reference implementation for small values and switch to the optimal solution at higher values. This would require determining a breakeven point, where reference and optimised have similar execution times, and whether a conditional implementation would improve performance or add an additional overhead.

When testing, the same inputs were used for all three implementations, and the results were taken multiple times and averaged to minimise variation.

Through testing* we discovered this point to be around 50 (as shown by the crossover on the below graph between the blue and gold lines). We then discovered the relationship between scale and resolution to be an addition of 20, and therefore added a conditional statement which chose the reference implementation if resolution was less than 70, or our optimal solution if resolution was 70 or above.

We tested this, and the results are shown in the green line below. This was slower than the original optimal solution, due to the additional overhead of the if statement. This is to be expected as the processor has extra logic to process which will take time, which was a maximum of 0.2 seconds extra. The breakeven point was also pushed back to around 65 due to this additional overhead.



*The times are only for the execute function, which includes loading the input.