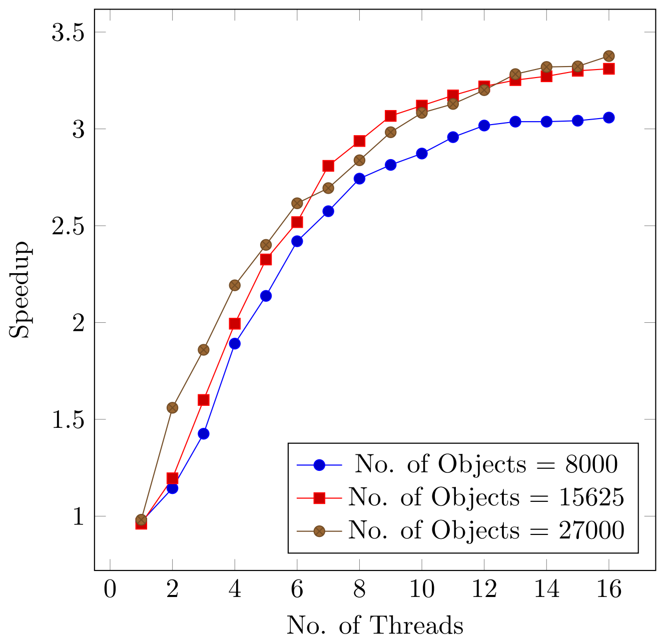
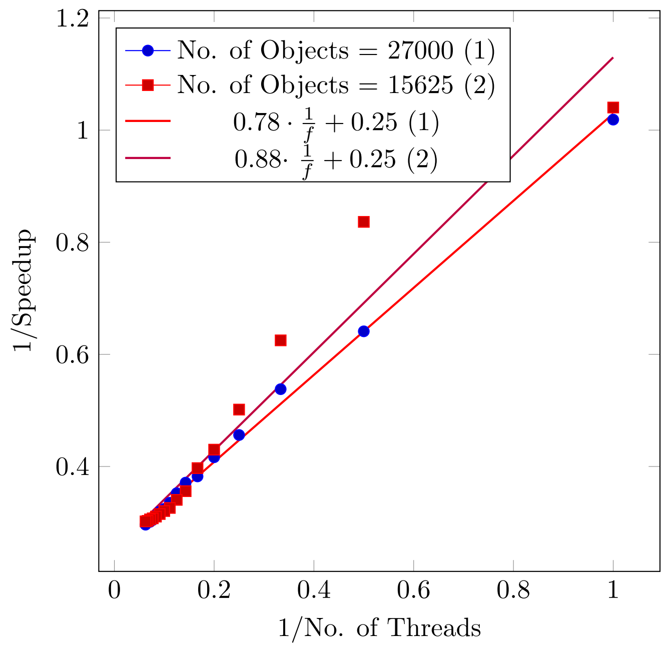
Solution Step 5 - **wtxd25** - Durham University

Experimentation Machine: Home PC - Intel® Core^{TM} i7-9800X X-Series Processor - 16 Threads

1 Results





2 How do my results compare to strong and weak scaling model?

The results above clearly show that the method used for parallelization fits the strong scaling law(Amdahl's Law). The speedup is defined as $S(p) = \frac{t(1)}{t(p)}$ and Amdahl's Law is $t(p) = f \cdot t(1) + (1-f) \cdot \frac{t(p)}{p}$. If we combine these two equations we can rewrite the speedup law as $S(p) = \frac{1}{f+\frac{(1-f)}{p}}$. This fits our curve better than the equivalent speedup law using the weak scaling law; if we follow weak scaling we should get a linear relationship between the number of threads and speedup.

3 What are the constants?

In the second plot we have graphed the inverse of the number of threads(p) against the inverse of the speedup. If we use the speedup definition from above which makes use of the strong scaling law and rearrange this equation we can get $\frac{1}{S(p)} = \frac{1-f}{p} + f$. $\therefore$ having plotted the inverses against one another, the gradient of the regression line should be $(1-f)$ and the intercept should be f. As such we get $0.22 \leq f \leq 0.25$. This is the constant which defines the amount of the code which is parallelized, and is given within a range due to potential inaccuracies in the result (machine precision,etc.). I have also included the same plot for 15625 objects, as can be sent the value of f is the same for the intercept but the gradient is incorrect. The ranges of the hypothesised f values overlap; this error could be caused by imprecision in measurement or some other factors.