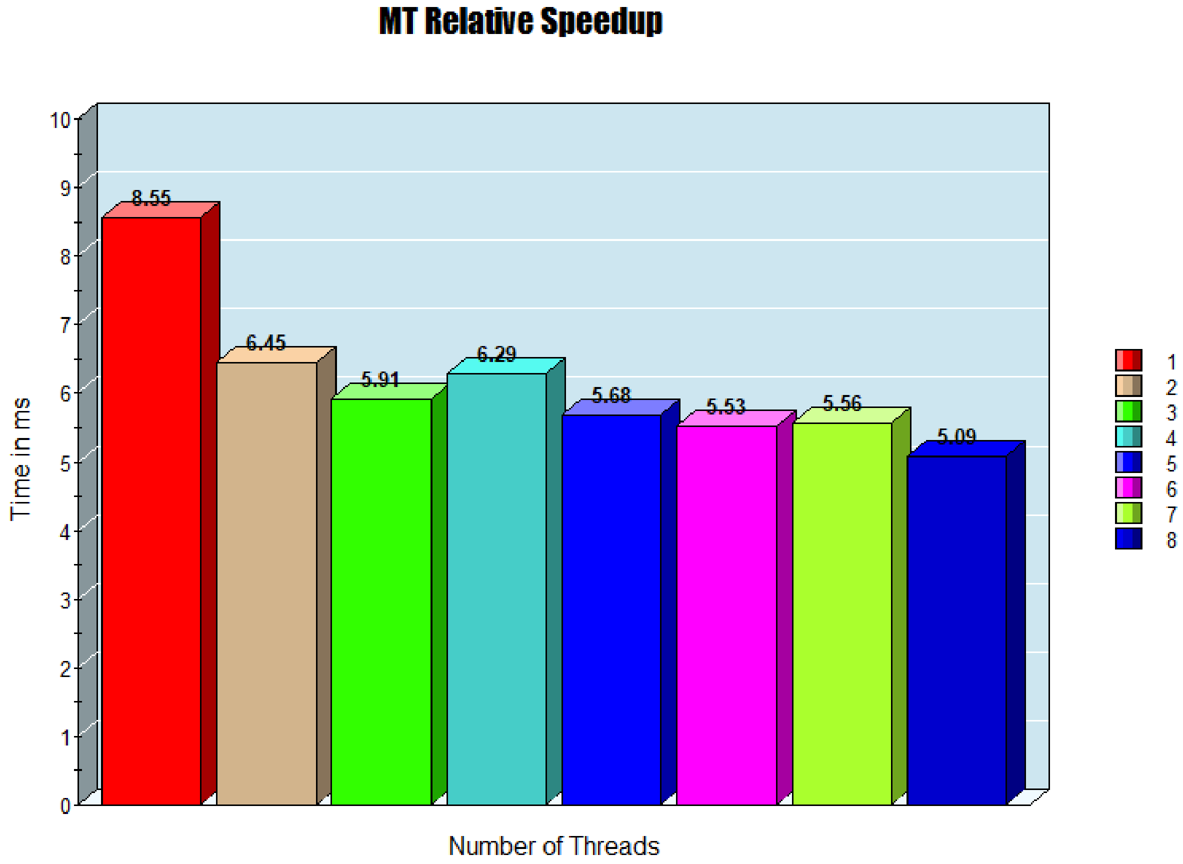

The results I have recorded show a clear reduction in execution time as the thread count increases. Utilising 2 threads results in around a 25% reduction in execution time relative to using just 1 thread. As you increase the number of threads used you get diminishing returns as thread overheads slow things down, and there is still quite a bit of code that is not parallelisable. The machine I am running my tests on is a 4 core, 8 thread intel CPU, so I only tested in the range of 1-8 threads inclusive. With 8 threads in action, we observe an execution time reduction of just over 40%, which is quite impressive.

My synchronisation strategy for this task was to synchronise access to a list of snowmen, where each thread could check the list for overlaps before adding their new snowman to the list if there were no overlaps. These checks are optimised by only checking for overlaps in detail if the two snowmen are relatively close to each other. This means list access is fairly efficient, and the threads do not need the lock to draw the actual snowmen. All access to the BufferedImage is a free for all because I have guaranteed there will be no data races as the drawings will never overlap. This strategy minimises synchronised regions while still preventing data races and concurrency bugs.