

## SAXPY Problem

### Solution Overview

Below are two proposed parallel solutions to the SAXPY problem, the stride method and chunking method. Both solutions are multi-threaded.

Stride Method:

The stride method consists of worker threads iterating over the array in  $k$  element increments where  $k$  is the number of threads. For example, in a two-thread solution, thread 1 would handle elements 0, 2, 4, 6, etc. while thread 2 handles elements 1, 3, 5, 7, etc.

Chunk Method:

The chunk method consists of worker threads taking sequential sections of the parent array. Back to the two-thread solution, in chunking thread 1 would handle elements 0-50 while thread 2 would handle elements 51-99.

### Performance Results

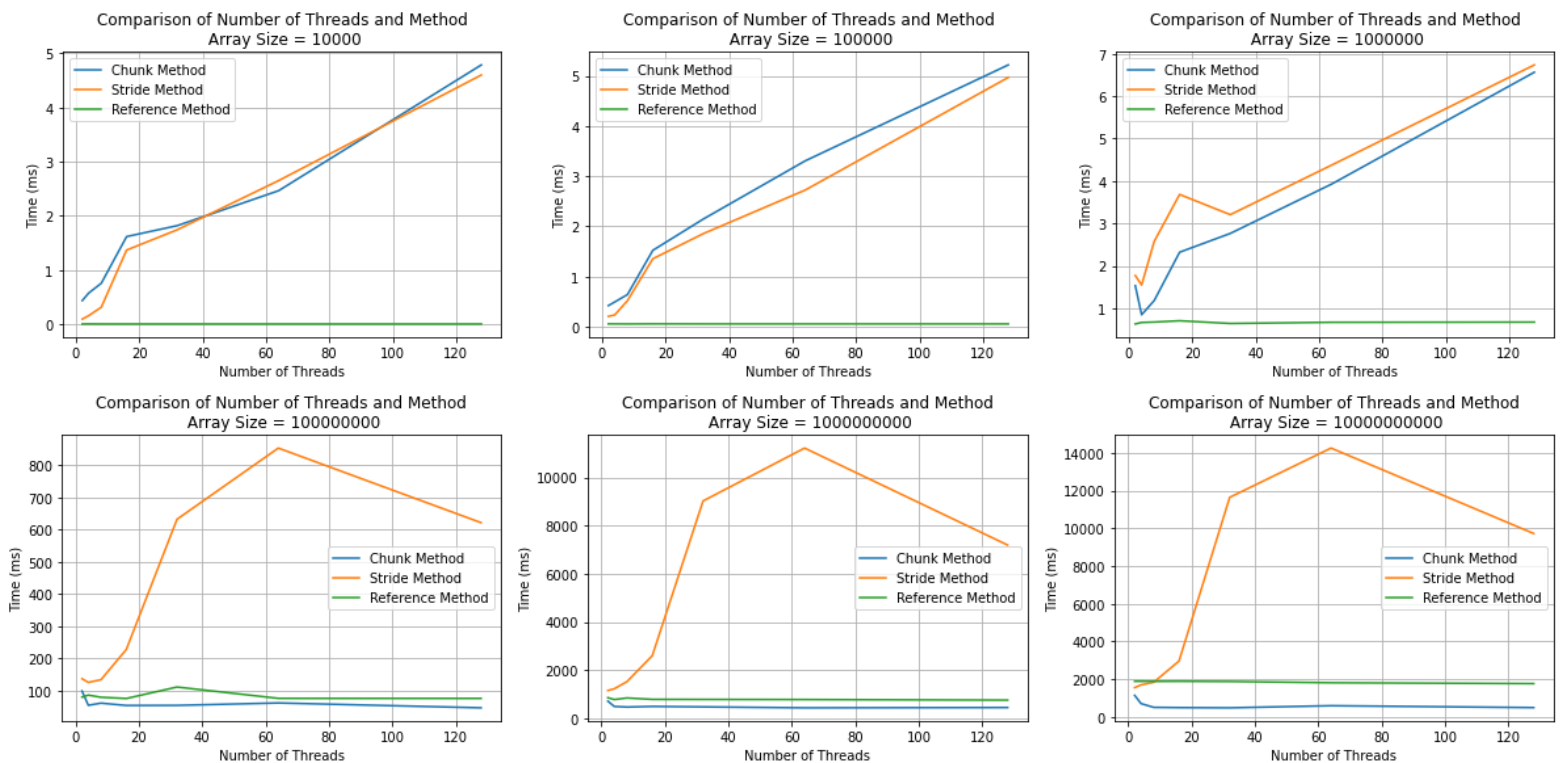


Figure 1: Plots comparing different solutions across various array sizes.

### Reflection

Across the various trials as seen in Figure 1, neither parallelization implementation outperforms a serial implementation until the array sizes exceed close to 100,000,000 elements. The parallel chunking solution is the only viable parallel solution compared to stride method for arrays that exceed the 100,000,000-element size. The stride method performs poorly due to having an increased rate in cache misses and false sharing especially as the quantity of threads increases. The chunking method in comparison avoids most of these cache misses by exploiting the locality of the elements stored in the array. It is important to note that the reference method performs so well despite the “embarrassingly” parallel problem due to how efficient modern-day compilers are at extracting inherent parallelism.