Parallel Programming
Assignment 1

Krunal Shah (2015EE10476)

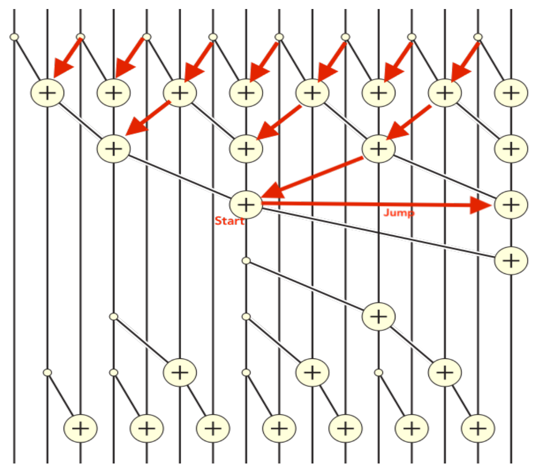
1 Parallel prefix sum

1.1 Implementation

The work efficient parallel algorithm to compute prefix sum as described in [1], is implemented. The prefix sum computation is performed **in place** and the algorithm proceeds as follows

1. Compute the sum of the adjacent elements of the array x and store them in the latter element. For clarity it can be thought of as the new elements being stored in a virtual array, say z_1.
2. Now compute the sum of the adjacent updated elements and store them in the latter updated element as before i.e the adjacent elements of the virtal array z_1 are added and stored in another virtual array, say z_2.
3. Note that the arrays mentioned are only for clarity and they are not actually allocated in the implementation and the updates are performed in place by the use of the variable $jump$ which doubles on every iteration giving the same effect as we described. We perform the above computations iteratively till we can create no more virtual arrays.
4. Next we go up the ladder we came down and now we update the elements which were not updated because they were the left in the updates in the previous steps.
5. We do the above updation by revisiting the above steps and updating the elements starting from $start + jump/2$ in intervals of jump (i.e $start + jump/2 * k * jump$) by adding to them the most updated values just before them which will be $start$ in intervals of jump (i.e $start + k * jump$).

Figure 1: Image taken from Wikipedia and edited.



1.2 Speedup and efficiency plots

Figure 2: Speedup plot

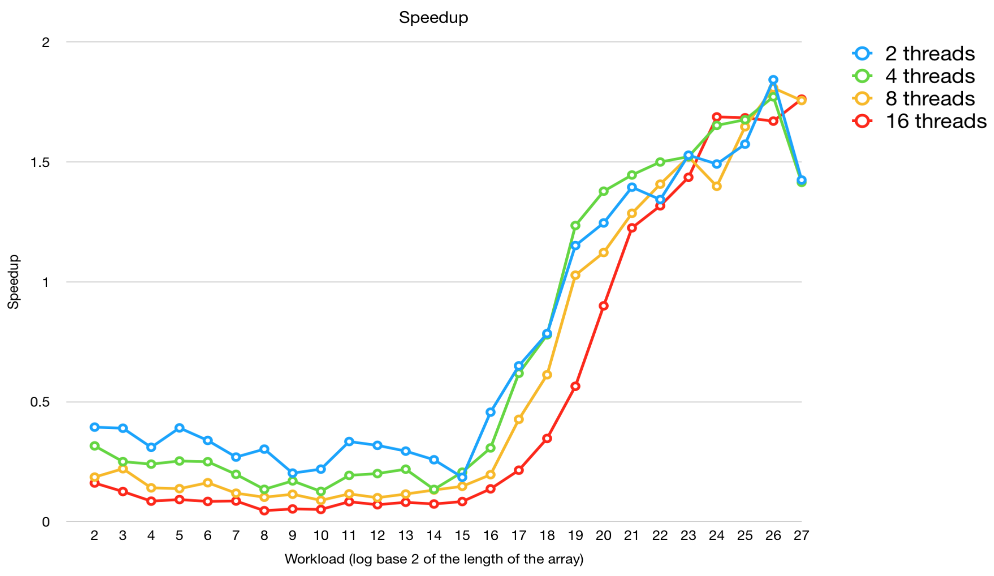
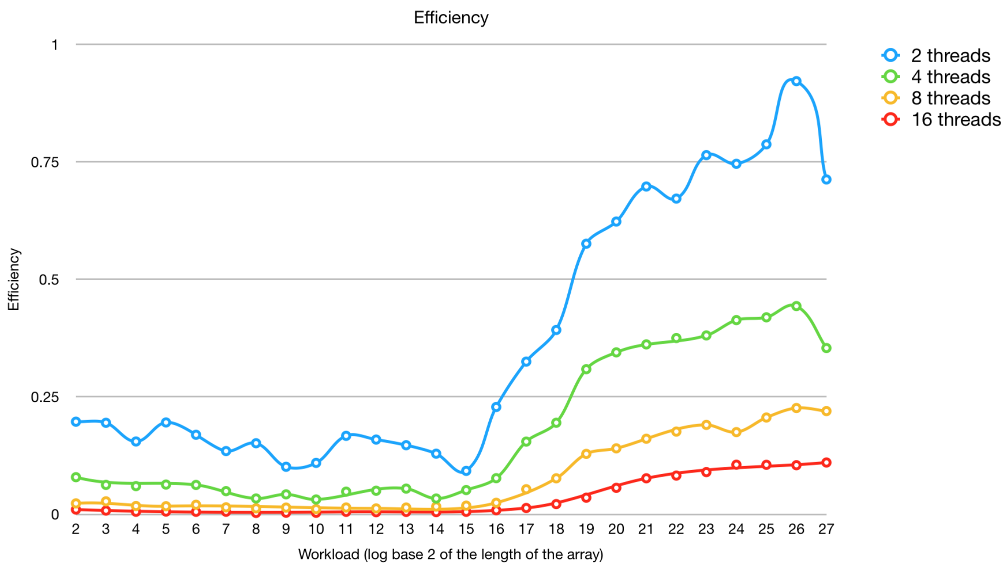


Figure 3: Efficiency plot



References

[1] RICHARD E. LADNER, MICHAEL J. FISCHER *Parallel Prefix Computation*. Journal of the Assoaatton for Computing Machinery,Vol 27.No 4,October198