Individual Programming

Give an array A[0..n-1], prefix sum on array A is given by array PS[0..n-1] such that PS[i]=$\sum_{j=0}^{i}$ A[j]. Thus, PS[0] = A[0], PS[1] = A[0]+A[1], PS[2] = A[0]+A[1]+A[2], etc.

One can find the prefix sum in parallel as follows using p processors: (i) partition the array into p subarrays, assigning processor P_i to the ith subarray, for $0 \leq i \leq p-1$, (ii) find local sums of individual subarrays, s_i, for $0 \leq i \leq p-1$, (iii) find prefix sums on the local sums, (iv) P_i uses the prefix sum value for the previous partition (i.e., $\sum_{j=0}^{i-1} s_j$) to calculate the prefix sum of its own subarray. For Step(iii), one processor can do this, or all processors can replicate this process.

Implement a shared-memory program on Hydra to find Prefix Sum of n integers. Keep the parallel overheads to a minimum. (One barrier synchronization, m_sync(), may be enough.) Vary n as 2^{10}, 2^{15}, and 2^{20} (or more if desired), and p, the number of processes, from 1 to 6 (or, how much is available up to 6), and obtain the speedup relative to the sequential timings without any overheads. Produce a speedup plot with p on x-axis and S_p on y-axis. Each n will result in a separate speedup curve. Use same plot for all the curves to see how speedup varies as problem size increases.

Submit the speedup plot, and the source code file, adequately documented. Due Thursday.