Parallel computing Exercise 6

Submit each solution separately to Moodle by 16.2. 09:00 (1 hour before exercises).

- 26. Implement the parallel prefixsum in MPI. Each process has initially M elements. The prefix-sum will go over all $P \times M$ elements. There will be a skeleton program in www-page.
- 27. Write a short MPI program to measure point-to-point messaging latency (µs) and bandwidth (GB/s) of our workstation. Use double MPI_Wtime(void) to measure time.
- 28. Analyse the blocking prefix-sum algorithm of 26 using LogGP parameters. How these values reflect to the performance of the actual implementation of parallel prefix (tasks 26 and 27)? Estimate the running time if P=16, Latency = 60 µs, Bandwidth = 90 MB/s, Overhead = 5 µs, (1Gb/s Ethernet, TCP).
- 29. Write a PRAM algorithm that finds the location of the longest common substring of two strings (character arrays). E.g., in strings "subsequence" and "timequestion" the longest common substring is "eque", and the locations are 5 and 4 (or 4 and 3 if you use 0-indexed arrays). You may not use CRCW PRAM. Try to make a sublinear time complexity and $O(N^2)$ work algorithm. Hint: you can use the segmented prefix sum and maximum finding algorithms.
- 30. Write a parallel C implementation of the longest common substring -task using OpenMP. Take the sequential implementation from www-page, parallelize it using OpenMP, test it using a multiprocessor/core system, and record the speedup you got. Now it is enough to parallelize only one loop as OpenMP is only for modest parallelization.