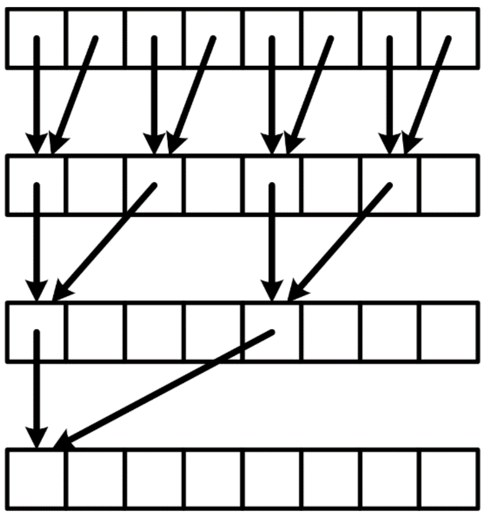
Example Exam Questions ‘Practical Parallel Programming’

1. (a) Point-to-point communication consists of a simple send and receive. Data is copied from the memory of one computer into the memory of another. In what ways can a protocol optimize the operation in order to speedup the execution of the parallel program in which it is used? What should be taken into account when choosing among the possibilities? What are the advantages and disadvantages?

   (b) Analyse the overheads in the following data-parallel application: the master partitions the data in N packages and sends the packages one by one to the N slaves which after receiving, process the received package. Assume that the data can be partitioned in any number of packages and that each package can be processed independently. The amount of data is considerable so that the overheads have a big impact on the performance. What are the overheads coming from the communication? *Hint:* draw the execution profile.

   (c) How could the algorithm be optimized in order to reduce the communication overhead?

2. Given are two reduction operations, one that should be executed in the order of a binary tree (see figure below), the other can be executed in any order (because the operation is associative).

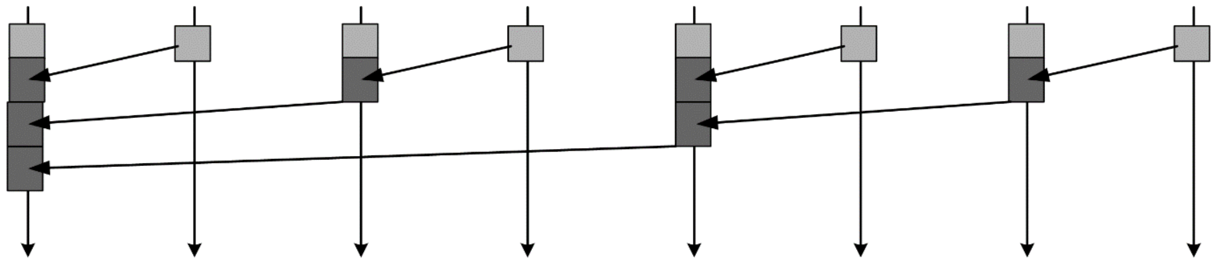


Array reduction in the order of a binary tree.

a. Write (pseudocode) a parallel multithreaded algorithm for both. You may assume that N (array size) is a power of 2. Take N/2 threads, do not create additional threads. In the balanced case, the optimal parallel algorithm is a binary tree reduction. On the other hand, when the reduction of 2 elements into 1 does not take the same amount of time, the reduction gets imbalanced: some threads will have to wait for slow threads. If the reduction operation is the associative, it allows other orderings and fast threads can reduce with each other, they do not have to wait for the slow threads.

Next, consider the distributed-memory case in which the data is evenly distributed among all nodes. Develop (pseudocode) two message-passing solutions based on the MPI point-to-point communication functions (do not use the MPI collective communications).

b. the first should execute the reductions in order, as shown in the diagram:


Parallel binary tree reduction. The Light gray box shows the local reductions. The arrows represent the messages being sent.

c. the second one can do the reductions in any order. Develop an algorithm that improves in the case of load imbalances: some processes finish earlier with their local reduction than others. It has been proven that it is optimal to merge results from finished processes.

3. Implement in pseudocode an ***n*-shift collective communication** based on MPI point-to-point messages that only make use of the links of a star topology which has node 0 in the center. Thus: only messages with the root are allowed.

The definition of a shift is:
public void MPI_Shift(void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Comm comm, int shiftAmount);
every process sends the contents of its send buffer to the process with **rank = (rank + shiftAmount + nbrProcesses) % nbrProcesses**
Note that we add **nbrProcesses** to ensure that the destination rank is positive. This is necessary because **shiftAmount** can be negative.
This function is called by each process.

4. To enable shared-memory multi-threaded programs on a multicore, the hardware has to solve several problems and provide several functionalities. Explain them and how they are implemented nowadays.
   Consider the question as follows: we put several CPUs together and connect them to the same RAM memory. Now: what should be provided additionally on the hardware level to be able to run shared-memory multi-threaded programs efficiently?

   b. What is the role of the Operating System?

   c. Considering the 3 programming primitives for writing multi-threaded programs, explain the role of hardware and operating system in all three of them.