Exam Parallel Programming 8 January 2009
Department of Computer Science, Faculty of Sciences

1. (a) Explain what a hypercube topology is. What in general are the diameter and bisection width of a hypercube with dimension N?

   (b) What is the main disadvantage of the hypercube topology?

2. What is Flynn's taxonomy? Which classes of computer systems does Flynn define?

3. Explain why the usage of asynchronous message passing may lead to the problem of buffer overflows, while synchronous message passing does not have this problem.

4. MPI has many forms of message passing:

   (a) It provides both blocking and nonblocking sends (not to be confused with synchronous and asynchronous sends); describe the differences and advantages/disadvantages of these two forms of transmission.

   (b) It provides four different modes for sending: standard, buffered, synchronous and ready mode. Again describe their differences and advantages/disadvantages.

5. Consider the following Fortran program

```
integer ind(1000000)
real X(1000000), Y(1000000)
do i = 1,1000000
  ind(i) = rand(1,1000000)  ! a random integer number between 1 and 1000000
  Y(i) = random()       ! random floating point number
  X(i) = 0.0
enddo

do j = 1, 1000
   do i = 2, 1000000
     X(i) = (Y(ind(i)) + X(ind(i-1))) / 2
   enddo
enddo
```

   Someone wants to parallelize the program using HPF (High Performance Fortran) by partitioning the large arrays X, Y, and ind among the different processors. Explain why HPF is unsuitable for this type of application. Why is it difficult to design efficient HPF data-distribution directives for the program? Also explain what communication primitives would be generated by the HPF compiler for this program.

6. Both branch-and-bound (as used, for example, for the Traveling Salesman Problem) and Barnes-Hut (used for N-body problems) use techniques to cut-off (prune) part of the computations. Parallel branch-and-bound algorithms can sometimes obtain superlinear speedups, because the parallel version may (for certain input problems) perform less work than a sequential algorithm. Can the parallel Barnes-Hut algorithm also obtain superlinear speedups in this way? Explain your answer.

7. Explain why divide-and-conquer parallelism is a good model for programming computational grids. Why do divide-and-conquer systems like Satin obtain nearly the same speedup on a 64-node local cluster as on a wide-area grid with 4 clusters of 16 nodes (64 nodes in total), despite the fact that the wide-area network between the clusters is about 1000 times slower than the local network (e.g., Myrinet) used within a cluster?

8. In the field of Multimedia Content Analysis (MMCA) a common approach to providing a parallelization tool for non-experts in high-performance computing is to build a software library containing a set of pre-parallelized routines. In case the library's application programming interface (API) hides all details of parallel execution, such a library is said to be 'user transparent' (e.g. Parallel-Horus).

   (a) Why is the efficiency of parallel execution of a full application implemented with a user transparent library partially still in the hands of the application programmer (i.e. the library user)?

   (b) What is "lazy parallelization" and how does it work in Parallel-Horus?

Points

1a	1b	2	3	4a	4b	5	6	7	8a	8b
6	6	10	10	7	7	10	10	10	7	7

Total: 90 (+ 10 = 100)