

TENTAMEN / EXAM

TDDC78/TANA77

Programmering av paralleldatorer /

Programming of parallel computers

2009-01-12, 08:00–12:00

Christoph Kessler och Lars Eldén
Institutionen för datavetenskap, Matematiska institutionen
Tekniska Högskolan i Linköping

Hjälpmedel / Allowed aids: Miniräknare, Engelsk ordbok /
pocket calculator, dictionary from English to your native language

Examinator: Christoph Kessler (TDDC78/TDDB78), Lars Eldén (TANA77)

Jourhavande lärare:

Christoph Kessler (IDA) 013-28-2406; 0703-666687

Lars Eldén (MAI) 013-28-2183; 0733-428842

Maximalt antal poäng / Max. #points: 40

Betyg / Grading (prel.):	MatNat		C, D, Y, DI	ECTS-graded students ^a
< 20	U	< 20	U	FX
20 – 30	G	20 – 27	3	C
31 – 40	VG	28 – 33	4	B
		34 – 40	5	A

^aSwedish grades will be automatically translated to the ECTS marks for exchange and international master program students as given above, according to a decision by the LiU rector in 2008.

General instructions

- Please use a new sheet for each assignment. Number all your sheets, and mark each sheet on top with your exam ID number and the course code.
- You may answer in either English or Swedish.
- Write clearly. Unreadable text will be ignored.
- Be precise in your statements. Unprecise formulations may lead to a reduction of points.
- Motivate clearly all statements and reasoning.
- Explain calculations and solution procedures.
- The assignments are *not* ordered according to difficulty.

1. (7.5 p.) Performance tools and analysis

- (a) (1p) Modern processors provide several built-in hardware counters that allow to collect performance data. Give two typical examples for performance-related information that can be obtained from such counters.
- (b) (3.5p)
 - (i) Explain the *general* approach of performance data collection using counters. (1p)
 - (ii) give at least two examples of performance-related information that can be derived from these. (1p)
 - (iii) Explain briefly the main difference from using hardware performance counters (see question 1a). (0.5p)
 - (iv) Describe the advantages and disadvantages of the general counter-based approach compared to *tracing*. (1p)
- (c) (3 p.) Derive Gustafsson's law and give its interpretation. Explain how it differs from Amdahl's law, and for what kind of parallel computations it is more appropriate.

2. (4 p.) Foster's design methodology consists of four stages. Name and explain them. Give details about their goals. What are the important properties of the result of each stage? Be thorough!

3. (3 p.) Can both the following do-loops be parallelized? Explain carefully why or why not.

```
do i=2,n-1
  y(i)=y(i+1)
enddo
```

```
do i=2,n-1
  y(i)=y(i+1)-y(i-1)
enddo
```

4. (6 p.) The following is an OpenMP code for matrix-vector multiplication, where the matrix is $m \times n$. It is assumed that n is a multiple of the number of threads.

```
y(1:m)=0.
!$omp parallel private
  nthr=omp_get_num_threads()
  q=n/nthr                ! Number of vectors
                          ! in each chunk
  myid=omp_get_thread_num()
  first=myid*q+1
  last=(myid+1)*q
  yp(1:m)=0.              ! Partial sum
  do j=first,last
    yp(1:m)=yp(1:m)+a(1:m,j)*x(j)
  enddo
  y(1:m)=y(1:m)+yp(1:m)
!$omp end parallel
```

- (a) Illustrate in a figure what each processor computes!
- (b) Which variable must be made private? Explain carefully!

- (c) Explain in detail why the code does not give the correct answer!
 - (d) Modify the code so that it gives the right answer (if you do not remember the syntax, explain in detail what the modification is supposed to do)!
5. (6p) We want to solve a linear system on a parallel computer with message passing. The following code is for Gaussian elimination (without pivoting).

```

do k=1,n-1
  a(k+1:n,k)=(1/a(k,k))*a(k+1:n,k)
  do j=k+1, n
    a(k+1:n,j)=a(k+1:n,j)-a(k,j)*a(k+1:n,k)
  enddo
enddo

```

- (a) Assume that the logical communication network is a ring, and that we distribute the matrix by block columns. Can we get good load-balancing? Motivate your answer (draw a figure of the matrix in a typical step).
 - (b) For the communication network in (a), how can we distribute the matrix so that we get good load-balancing? Motivate your answer.
 - (c) With the same assumptions as in (b), draw a figure illustrating the matrix in a typical step. What is communicated?
6. (6.5 p.) **MPI**

- (a) (2.5 p.) MPI supports *nonblocking* (also called *incomplete*) point-to-point communication. What does that mean? Give an example scenario demonstrating how using a nonblocking send (MPI_ISEND) or receive (MPI_IRECV) routine instead of their blocking counterpart could speed up program execution. What kind of routine is necessary with nonblocking communication if we need to make sure that data dependences are preserved?
- (b) (4 p.) Assume we have p processors connected by some network, and each processor holds a local array of N integer elements.

Write a MPI program for p processors that computes the *global maximum* of all pN array elements in parallel and that uses only point-to-point communication (i.e., send and receive operations, not collective communication operations). To obtain full points, your program should be *time-optimal* and scale well for large p .

Draw a figure to show the flow of communication between processors, and draw a processor-time diagram to show the communication over time.

Then, use the LogP model to analyze the execution time, work and cost of your program (as formulas of N , p and possibly some other model parameters as appropriate).

7. (7 p.) **Parallel computer architecture**

- (a) (1.5 p.) What is *grid computing*? What sort of applications (with what kind of parallelism) can use computational grids effectively?
- (b) (2.5 p.) What is 'false sharing'? In what kind of parallel computers and in what situations does it occur, and how does it affect performance? Suggest one possible way how the problem could be reduced.

- (c) (1.5 p.)
- (i) Give an example of an interconnection network topology (name and sketch) where the node degree is bounded by a constant.
 - (ii) What is the advantage of a constant node degree?
- (d) (1.5p) What is a *fat-tree* interconnection network? How does it improve communication bandwidth over ordinary tree networks?