

TENTAMEN / EXAM

TDDC78/TANA77

Programmering av paralleldatorer /
Programming of parallel computers

2008-05-26, 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^a, D, Y, M	Exchange students^b
	20 – 30 G	20 – 27 3	ECTS-scale
	31 – 40 VG	28 – 33 4	
		34 – 40 5	

^a*OBS C:are antagna före 2001:* Om du vill ha ditt betyg i det gamla betygssystemet (U, G, VG) skriv detta tydligt på omslaget av tentan. Annars kommer vi att använda det nya systemet (U, 3, 4, 5).

^bECTS marks will be applied for students with a 'P' in the personnummer.

General instructions

- Please use a new sheet for each assignment. Number all your sheets, and mark each sheet on top with your name, personnummer, 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. (4 p.) Performance tools and analysis
 - (a) (2 p.) Describe performance analysis through counters. What are the advantages and disadvantages of the approach?
 - (b) (2 p.) Derive Amdahl's law and give its interpretation.
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. (5p) Consider the following two OpenMP variants of Gaussian elimination:

```

do k=1,n-1
  a(k+1:n,k)=(1/a(k,k))*a(k+1:n,k)
  !$omp parallel do
  do j=k+1, n
    a(k+1:n,j)=a(k+1:n,j)-a(k,j)*a(k+1:n,k)
  enddo
  !$omp end parallel do
enddo

```

and

```

!$omp parallel
do k=1,n-1
  a(k+1:n,k)=(1/a(k,k))*a(k+1:n,k)
  !$omp do
  do j=k+1, n
    a(k+1:n,j)=a(k+1:n,j)-a(k,j)*a(k+1:n,k)
  enddo
  !$omp end do
enddo
!$omp end parallel

```

The first gives correct results. The second gives wrong answers, but it executes approximately 20% faster.

- (a) Explain why the first is slower.
 - (b) In what way is the second incorrect? Modify the code so that it gives correct results (without making it slower). Explain carefully the modifications. (If you have forgotten the syntax of OpenMP, you may invent your own provided that you explain it well).
4. (5 p.)
 - (a) There are three important machine-specific parameters in the timing analysis of computations on a message passing system using the Delay model as cost model. Explain what these parameters are; what are typical values on modern parallel systems? (Only very rough estimates are required, it is the relative magnitudes that are interesting).
 - (b) In a 2-D parallel particle simulation of n^2 particles we have a $d \times d$ mesh of processors, each handling $k^2 = (n/d)^2$ particles. In each time step assume that the number of floating point operations in each processor is $C_1 k^2$ and the temperature is so low that the number

of particles that cross a border to a neighbouring processor is C_2k . For simplicity, put $C_1 = C_2$. Assume the usual communication model for message passing parallel systems. What is the (relative) speedup of the algorithm? What is the (relative) efficiency?

5. (5 points) Consider the code

```

do k=1,n-1
  a(k+1:n,k)=a(k+1:n,k)/a(k,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) Describe using figures how the matrix elements are accessed after some steps of the algorithm, $k \approx n/2$, say.
- (b) Assume that we want to implement the algorithm on a message passing system, organized as a ring of processors. How should we distribute the matrix elements for good load balancing? Motivate in detail.
- (c) If the system is a 2-D grid of processors, how should we distribute the matrix for good load balancing? Motivate in detail.

6. (6 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) (1 p.) Give two good reasons for using collective communication operations instead of equivalent combinations of point-to-point communication (MPI_send and MPI_receive) operations.
- (c) (2.5 p.) Explain the principle of one-sided communication in MPI-2. (1.5p)
 Why is one-sided communication considered being “closer” to a shared-memory programming model than ordinary two-sided message passing? (0.5p)
 And how does one-sided communication still differ from the shared-memory programming model? (0.5p)

7. (2 p.) **OpenMP**

- (a) (2 p.) What is the purpose of the `flush` directive in OpenMP? Give a short example to illustrate how it is used. Name at least one technical cause that makes the explicit use of `flush` in the program necessary to guarantee a correct program execution.

8. (9 p.) **Parallel computer architecture**

- (a) (1.5 p.) What does ‘simultaneous multithreading’ mean? What kind of programs can take the most advantage of such an architecture?
- (b) (1.5 p.) What is *grid computing*? What sort of applications (with what kind of parallelism) can use computational grids effectively?

- (c) (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.
- (d) (1 p.) Give an example of an interconnection network topology (name and sketch) where the node degree is bounded by a constant. What is the advantage of a constant node degree?
- (e) (2.5 p.) Describe the d -dimensional Hypercube network architecture (1p).
How does the maximum latency scale with the number p of processors? (give a formula in p , 0.5p)
Give the idea of a routing algorithm that computes a shortest path for communication from a processor i to a processor j (1p).