Försättsblad till skriftlig tentamen vid Linköpings Universitet

(fylls i av ansvarig)

Datum för tentamen	2011 – 05 – 28
Sal	TER2
Tid	14:00 – 18:00
Kurskod	TDDC78
Provkod	TEN1
Kursnamn/benämning	Programmering av parallelldatorer – metoder och verktyg
Institution	IDA
Antal uppgifter som ingår i tentamen	7
Antal sidor på tentamen (inkl. försättsbladet)	5
Jour/Kursansvarig	Christoph Kessler, Henrik Branden
Telefon under skrivtid	Se täckbladet (sida 1) av tentan
Besöker salen ca kl.	Se täckbladet (sida 1) av tentan
Kursadministratör (namn + tfnnr + mailadress)	Gunilla Mellheden, IDA, 2297, gunme @ ida.liu.se
Tillåtna hjälpmedel	Engelsk ordbok
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	Se täckbladet (sida 1) av tentan

TENTAMEN / EXAM

TDDC78/TANA77

Programmering av parallelldatorer /
Programming of parallel computers

2011-05-28, 14:00–18:00, TER2

Christoph Kessler and Henrik Brandén
Dept. of Computer Science (IDA), Dept. of Mathematics (MAI)
Linköping University

Hjälpmedel / Allowed aids: Engelsk ordbok /
dictionary from English to your native language

Examinator: Christoph Kessler (TDDC78), Henrik Brandén (TANA77)

Jourhavande lärare:
Christoph Kessler (IDA) 013-28-2406; 0703-666687; visiting at ca. 16:00
Henrik Brandén (MAI) 013-28-5759; 0706-011969; visiting at ca. 16:00

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

[a] Swedish 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.

Note: In the case of questions regarding the assignments 3 to 4, please contact Henrik Brandén in the first hand; for questions regarding the other assignments, contact Christoph Kessler in the first hand.

1. (3.5 p.) **Performance tools and analysis**

   (a) (0.5p) Give one typical example for performance-related data about *message passing* programs that can be collected using *software counters*.

   (b) (1 p.) Which performance data collection method is required in order to be able to draw a processor-time diagram (also known as a *Gantt chart*)? Justify your answer (technical reasons).

   (c) (2 p.) (*i*) Derive Amdahl's law and (*ii*) give its interpretation.
   (Note: a picture is nice but is not a proof; a calculation is expected for the derivation of Amdahl's Law.)

2. (4 p.) **Parallel program design methodology**

   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. (6 p.) The FORTRAN 77 code

```
do i=1,n-1
  y(i) = a*y(i+1) + x(i)
end do
y(n) = c
```

   can be written

```
y(1:n-1) = a*y(2:n) + x(1:n-1)
y(n) = c
```

   using FORTRAN 90.

   (a) What is the benefit of the FORTRAN 90 notation? (1p)

   Consider data parallel programming on a distributed memory machine with p processors organized as a ring. Assume a vector length $n = rp$ for some integer $r > 0$, a network latency α, a network transfer speed β^{-1}, and that the time for one arithmetic operation is γ.

   (b) How is the given FORTRAN 90 code executed in terms of communication, synchronization, and arithmetics? (2p)

   (c) What is the efficiency E_p? (2p)

   (d) How would you rewrite the FORTRAN 77 code

```
y(1) = c
do i=2,n
  y(i) = a*y(i-1) + x(i)
end do
```

   in FORTRAN 90 notation? (1p)

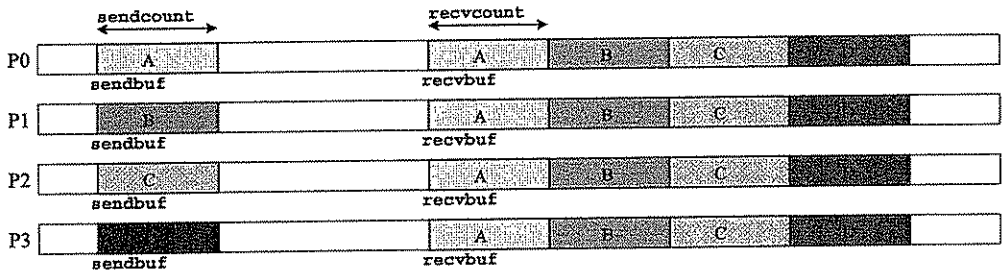

Figure 1: Effect of the `MPI_Allgather` operation on the processes' local memories, here shown for $p = 4$ processes.

4. (9 p.) Consider matrix-vector multiplication $y = Ax$ on a

   (a) distributed memory machine using data parallel programming, (3p)

   (b) distributed memory machine using message passing, and on a (3p)

   (c) shared memory machine using shared memory programming. (3p)

   For each case, explain one algorithm of your own choice, including the partitioning and distribution of data.

5. (7 p.) **MPI**

   (a) (5.5 p.) The `MPI_Allgather` operation is a collective communication operation with the following parameters:

```
int MPI_Allgather ( void *sendbuf, int sendcount, MPI_Datatype sendtype,
                    void *recvbuf, int recvcount, MPI_Datatype recvtype,
                    MPI_Comm comm );
```

   `MPI_Allgather` is a generalization of `MPI_Gather`: If executed by a group of p MPI processes each contributing a local array in `sendbuf` with `sendcount` elements of type `sendtype`, each of the p processes will afterwards hold a copy of the entire gathered array in its `recvbuf`, which is composed of the $p \times$`recvcount` elements coming from each send buffer in the order of the processor ranks. See Figure 1.

   Usually, the arguments for `sendcount` and `recvcount` and for `sendtype` and `recvtype`, respectively, are equal in calls to `MPI_Allgather`. Also, all participating processes must pass equal argument values for each of these parameters. A typical call could look as follows:

```
MPI_Allgather ( Arr1, n, MPI_FLOAT, Arr2, n, MPI_FLOAT, MPI_COMM_WORLD );
```

   i. What is the difference in behavior to `MPI_Gather`? (0.5p)

   ii. Write an implementation of `MPI_Allgather` using only `MPI_Send` and `MPI_Recv` operations (i.e., no other communication operations). Use C, Fortran, or equivalent pseudocode (explain your language constructs if you are not sure about the right syntax). Explain your code. (2.5p)

   iii. Show how your code behaves by drawing an annotated processor-time diagram for $p = 4$ showing *when* messages are sent from *which* source to *which* destination, and *what* they contain. (1p)

iv. Let n denote the value of `sendcount` and `recvcount`, and assume that the `send/recvtype` denotes normal floatingpoint numbers. Analyze asymptotically the (worst-case) *parallel execution time* of your implementation (for arbitrary values of p and n) as a function in p and n. You may use the delay model, the BSP model or the LogP / LogGP model for this purpose (state which model you use). If you need to make further assumptions, state them clearly. (1.5p)

(b) (1.5 p.) Explain the principle of one-sided communication in MPI-2.

6. (4.5 p.) **OpenMP**

(a) (3 p.) What scheduling methods (3) for parallel loops are defined in the OpenMP standard? Explain each of them briefly. When should they be used? How does the user setting for the chunk size affect the (expected) performance of the dynamic methods?

(b) What is the purpose of the `reduction` clause in OpenMP parallel loops? Be thorough! (1.5p)

7. (6 p.) **Parallel computer architecture**

(a) What kind of parallelism can be exploited efficiently in computational grids? (0.5p)

(b) Given an application with a fixed performance requirement (in GFlops). Why can, provided that the application can be parallelized (work-)efficiently, the transition from a single-core to a multicore execution platform be advantageous from a power efficiency point of view? (1p)

(c) Explain the *write-invalidate protocol* used to achieve sequential consistency in a cache-based shared-memory system with a bus as interconnection network. (1.5p)

(d) (2 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.

(e) (1p) Give an example of an interconnection network topology (name and sketch) where the maximum distance between any two nodes grows logarithmically in the number of nodes.

Gunilla Mellheden

From: Petru Ion Eles [petru.eles@liu.se]
Sent: den 25 maj 2011 11:02
To: gunme@ida.liu.se
Subject: RE: Booking for presentation

Dear Gunilla,

Can you register grade 5 to
- Raúl Martínez López 860227P135
- Begoña Montánchez Crespo 840705P161

for their project work

Best regards,

Petru

- Petru Eles, Professor Linkoping University, Dept. of Computer and Information Science
S-581 83 Linkoping, SWEDEN
email: petru.eles@liu.se
phone: +46-13-281396, fax: +46-13-284499
URL: http://www.ida.liu.se/~petel

From: peter.dalenius@gmail.com [peter.dalenius@gmail.com] On Behalf Of Peter Dalenius [peter.dalenius@liu.se]
Sent: Thursday, April 28, 2011 12:29 PM
To: Petru Ion Eles
Cc: Gunilla Mellheden
Subject: Re: Booking for presentation

Jo, det stämmer. De har gjort ett Erasmus-projekt och är registrerade på projektkursen TGZD20. Den enda examinationen är en skriftlig eller muntlig projektrapport. Exakt hur det redovisas är upp till examinatorn. Se kursplan:
http://kdb-5.liu.se/liu/lith/studiehandboken/svkursplan.lasso?&k_kurskod=TGZD20&k_budget_year=2011

/Peter

2011/4/28 Petru Ion Eles <petru.eles@liu.se>:
> Dear Gunilla,
>
> They are not in a master program. They are Erasmus students.
> They, in fact, don't get any "exam" from Linköping.
> They only want to present their thesis and the presentation to be
> registered in LADOK. Can you check what the requirement is in this case?
>
> I have also Cc Peter.
>
> This might be a "grey zone" case. Since they don't get any exam from
> us, just simply presenting the thesis, without auscultation and
> opposition might be reasonable.
>
>
> Best regards,
>
>
> /Petru
>

> - Petru Eles,
> Professor Linkoping University, Dept. of Computer and Information
> Science
> S-581 83 Linkoping, SWEDEN
> e-mail: petru.eles@liu.se
> phone: +46-13-281396, fax: +46-13-284499
> URL: http://www.ida.liu.se/~petel
> _____
> From: Gunilla Mellheden
> Sent: Wednesday, April 27, 2011 7:35 PM
> To: Raúl Martínez López
> Cc: Petru Ion Eles; Gunilla Mellheden
> Subject: RE: Booking for presentation
>
> Hi
>
> It seems that you have not been reading the instructions for the work regarding the
> asters program.
>
> 1. None of you have done any auscultations! You need three!
> 2. None of you are registered to any exjob code or the code TEXMAS!

> You are not allowed to present your thesis as long as you do not fulfill these
requirements!
>
> Please read the instructions that I sent you earlier and let me know when you are ready!
>
> Best regards,
> Gunilla
>
>
> -----Original Message-----
> From: Raúl Martínez López [mailto:rmlopez01@gmail.com]
> Sent: den 27 april 2011 17:49
> To: Gunilla Mellheden
> Subject: Re: Booking for presentation
>

> Here you have the information needed:

> - Names and SSN:
> - Raúl Martínez López 860227P135
> - Begoña Montánchez Crespo 840705P161

> - Thesis title:
> - Design and Implementation of an Android-device Based Remote
> Control System
>
> - Abstract:
> The new technologies present on smartphones allow the development of
> new applications and it is becoming more and more interesting to
> connect them to other types of consumer electronics appliances, such
> as a set-top box (STB), making it possible to improve solutions
> already existent, in this case, a remote control system, that can take
> advantage of these new technologies, and perform task in a better way
> while adding new functionalities.
> Our main goal is providing the user of an Android mobile device the
> capability of controlling the IPTV Set-Top-Box. The solution relies on
> a mobile device application emulating the functioning of a normal IR
> remote control and a server application that resides in the middleware
> platform which should handle the received commands.