UNIVERSITY OF GOTHENBURG

The Board of the IT Faculty

DIT360, Parallel Computer Organization and Design, 7.5 higher education credits

Second Cycle/AIF

---

1 Confirmation

The Board of the IT Faculty established the course plan at 2006-11-17. This course plan is valid from autumn 2007. The syllabus has been revised by the Board of IT Faculty 2009-09-15 to be valid from the autumn 2010.

Field of education: Sciences

Responsible department: Computer Science and Engineering.

Main field: Computer Science

2. Position in the educational system

The course is a part of the Computer Science Master's programme and an elective course at the University of Gothenburg.

Second cycle, the course has course/courses at first cycle level as entry requirements. (A1F)

3. Entry requirements

The requirement for the course is to have successfully completed a bachelor degree within Computer Science or equivalent. Specifically, the course DIT051, Computer Architecture is required.

4. Course content

The course covers programming models and architectural techniques to implement them for parallel computers. The content is divided into the following parts:

1. The first part provides a taxonomy and an overview of different programming models and architectural paradigms for parallel computers.
2. The second part deals with two prevailing programming models: shared memory and message passing. The course covers the key primitives used by these models and how they are integrated in mainstream programming languages such as C.
3. The third part deals with the different issues involved in designing highly efficient parallel software. Important concepts are decomposition of a sequential program into parallel threads, balancing the load, techniques to reduce communication, and techniques to synchronize threads. Practical experiences will be gained through an assignment and exercises.
4. The fourth part deals with the design principles for small-scale parallel computers under shared memory, e.g., design principles for multi-core microprocessors. Important concepts covered are cache coherence and consistency and techniques considered are snoopy-cache protocols, the inclusion property, multi-phase protocols.
5. The fifth part deals with scalability of parallel computers, i.e., architectural techniques for scaling the number of processors to a high count. This part deals with message passing protocols and scalable cache coherence protocols.
6. The sixth part deals with interconnection networks, an important component for scalable parallel computer architectures. Important concepts covered are routing, switching and topologies for scalable interconnection networks.
7. The fifth and final part deals with latency tolerance techniques, i.e., techniques that can hide the latency of communication evens such as sending a message from one node to another. Techniques that are treated include multithreading, relaxed memory consistency models, and prefetching.

5. Outcomes

After the course, the student shall be able to:

- master the terminology and key concepts in the parallel architecture field in order to follow the research advances in this field;
- understand the principles behind a parallel computer; especially principles for the design of the communication substrate to support shared-memory as well as message passing programming models;
- understand the principles behind programming models for parallel computer systems; and
- basic skills in the design of software for parallel computers (e.g. multi-core) and especially the issues involved in designing highly efficient parallel software.

6. Required reading

See separate literature list

7. Assessment

Written exam(6.0 ECTS Credits) and approved assignment(1.5 ECTS Credits)

A student who has failed a test twice has the right to change examiners, if it is possible. A written application should be sent to the Department.

8. Grading scale

The course is graded with the following marks: Fail, Pass, Pass with Distinction. The course can also, at the students' request, be marked according to ECTS standards.

9. Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire can be used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

10. Additional information

The course held in English ang organized into lectures, exercises, and two assignments that focus on the design of parallel software and assessment of the impact of performance for architectural techniques.