UNIVERSITY OF GOTHENBURG

COMPUTER SCIENCE AND ENGINEERING

DIT261 Parallel Functional Programming, 7.5 credits

Parallell Funktionell Programmering, 7,5 högskolepoäng

Second Cycle

Confirmation

This course syllabus was confirmed by The IT Faculty Board on 2012-02-22 and was last revised on 2017-06-16 by Department of Computer Science and Engineering to be valid from 2017-08-20, autumn semester of 2017.

Field of education: Science 100%
Department: Computer Science and Engineering

Position in the educational system

The course is a part of the Computer Science Master's Programme and a single subject course at the University of Gothenburg

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS) and 2) Applied Data Science Master's Programme (N2ADS)

Main field of studies	*Specialization*
Computer Science-Algorithms and Logic	A1F, Second cycle, has second-cycle course/s as entry requirements

Entry requirements

The requirement for the course is to have successfully completed two years within the subject Computer Science or equivalent. The course participants must have completed an introductory course in Functional Programming, preferably using Haskell or Erlang (Examples at GU are DIT440 Introduction to Functional Programming and DIT142 Functional Programming).

Applicants must prove knowledge of English: English 6/English B or the equivalent level of an internationally recognized test, for example TOEFL, IELTS.

Learning outcomes
After completion of the course the student is expected to be albe to:

1. Knowledge and understanding

- Distinguish between concurrency and parallelism.

- Give an overview of approaches to parallelism in functional programming languages in the scientific literature.

2. Skills and abilities

- Write, modify and test parallel functional programs, to run on a variety of architectures such as shared memory multiprocessors, networks of commodity servers, and GPUs.

- Interpret parallelism profiles and address bottlenecks.

3. Judgement and approach

- Identify when using a functional language may be appropriate for solving a parallel programming problem.

- Select an appropriate form of parallel functional programming for a given problem, and explain the choice.

Course content
The course introduce the principles and practice of parallel programming in a functional programming language. In this course, the term paralle programming means using multiple hardware cores of processors in order to gain speed. The course covers approaches to parallel fuctional programming in both Haskell and Erlang. It covers curret research on these topics, and relies heavily on scientific papers as its source materials.

- Advantages of functional approaches to parallelism: immutability, absence of data races, determinism.
- Profiling parallel functional programs: granularity, bottlenecks, locality, data-dependencies.
- Parallel functional algorithms: divide-and-conquer.

- Approaches to expressing parallelism in Haskell: the Eval monad, the Par monad, parallel strategies, skeletons, data parallelism.
- Functional approaches to GPU programming
- Parallelisation and distribution for Erlang. Scalability. Handling errors in a massively parallel system.
- Case studies of industrial parallel functional programming, such as map-reduce and scalable no-SQL databases.

Form of teaching

Language of instruction: English

Assessment

An individual written examination taken in an examination hall and two compulsory laboratory exercises. The laboratory exercises are normally done in groups of 2.

A student who has failed the examination twice has the right to request of the department a change of examiner. The request is to be in writing and submitted as soon as possible. The department is to grant such a request without delay.

In cases where a course has been discontinued or major changes have been made a student should be guaranteed at least three examination occasions (including the ordinary examination occasion) during a time of at least one year from the last time the course was given.

Grades

The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).
In order to be awarded a Pass (G) for a full course, the laboatory assignments must be approved and a passing mark must be obtained in the written exam. To be awarded Pass with Distinction (VG), the student must receive a VG in the written exam and the laboratory assignments must be approved.

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.