Course Analysis DD2443

Course round: Per 1, 2021

Course responsible: Mads Dam

Note: This course analysis is based on own notes, results from the LEQ questionnaire, and reflections and comments from student discussions in class.

Course structure

14 lectures, twice weekly, 7 exercise sessions, once a week, 1 seminar session, 3 labs

Oral exam, A-F scale

Grading based on:
- Exam
- Exercises
- Seminar attendance (obligatory)
- Seminar presentations (voluntary, gave optional bonus point)
- Peer review
- Labs

General Reflections

At this time the course is in a stable state and has been for a couple of years. The course is generally appreciated by students, and fills a need for more in-depth studies in parallel and distributed computing than what is covered in routine courses.

There are two distinct parts to the course, parallel (shared memory) computing, and distributed (message passing) computing, partly due to practicalities (the main textbook focuses on the former), and partly due to inherent differences in the subject matter. It would be an option to split the course into separate parallel and distributed parts. I have chosen not to do this, since most students will take only one course on parallel/distributed computing. Also there is some pedagogic value in teaching both parts in one course, to emphasize the commonalities and distinctions in these two closely related branches.

The HT20 and HT21 versions of the course were strongly influenced by the Covid situation. Lectures were given online only, and exercises in a mixed format, which worked adequately. The main issues were ones experienced by most teachers, namely much reduced contact with students and lack of direct student feedback at lectures.

The seminars are probably the most questionable course elements. Seminar attendance is obligatory, and seminar presentations are obligatory for PhD students, but voluntary for first and second cycle students. In the latter case one bonus grade is given for presentations of adequate quality. Typically, only a few non-PhD

students volunteer presentations, however. I am still of the opinion that it is worthwhile to keep this course element, mainly to cater for PhD students.

Student feedback

No formal course committee was appointed. I take regular feedback throughout the course (although this has been much more difficult with online lectures), and the final 30 minutes of the final lecture with all "surviving" students present was devoted to feedback.

Here are the main feedback issues (from the LEQ):

Exercises:
- More practice, less theory

Mads's reflections: Proofs and theory are hard for students, yes, but high pedagogical value to show students that you can actually reason formally and rigorously about phenomena that have direct practical implications. Indeed this interplay between theory and practice is one major attraction of the subject.
- More creative work than exercises

Mads's reflections: This is good point though not clear how to achieve this within the time and hp constraints of the course. Really what the course needs is a supplementary project element.
- Better introduction to link course subjects

Mads's reflections: Good comment, maybe to partially revamp the intro lecture.
- Reference solutions too brief and hard to understand, or in some cases simply missing

Mads's reflections: Noted.
- Introduce lab hours

Mads's reflections: Good suggestion, but needs resources.
- Explore collaboration with interaction programming course e.g. to animate difficult concepts and algorithms

Mads's reflections: Not a bad idea, actually
- Heavy workload

Mads's reflections: Judging from the LEQ responses this does not seem to be borne out. Student appear to spend in average some 20-25 hrs/week, which would well reflect the 7,5 hp assigned to the course.

Planned Changes

- In HT22 more lectures will be given by Karl Palmskog, in order to do a smooth switchover of course responsibility in HT23.
- Lab 3 is in need of revision. Lab goals are not sufficiently clear, and it is probably a bit too long.

Mads Dam, Professor
Theoretical Computer Science

KTH Royal Institute of Technology