HIS Seminar: Distributed Algorithms

Jörg Schäfer
Fachhochschule Frankfurt am Main
University of Applied Sciences

Fachbereich 2 Informatik und Ingenieurwissenschaften
Computer Science and Engineering

Course Description
Wintersemester 2011/2012

Abstract

The Modul *14.2. "Elective Subjects II Module: Current Topics in High Integrity Systems"* covers current (research) topics relevant for High Integrity Systems. Upon completion of this course, the student is able to assess new developments in Computer Science and incorporate new methods into the software development process. Students learn to search for, read, summarize and cite scientific literature on a large scale, to write a report as a scientific paper, and to give a scientific talk.
The topic for this semester's seminar is *Distributed Algorithms* – a topic that is getting increasingly more relevant as today's computer systems are designed ever more as distributed systems. Furthermore, the topic is highly relevant for High Integrity Systems and many active research problems have emerged from practical problems designing High Integrity Systems such as e.g. flight control software or other mission critical software.

1 Content

This course is intended for master students who are interested in understanding the theoretical background of distributed algorithms and who want to learn practically relevant distributed algorithms. This seminar addresses algorithms, which run on distributed processes or different computers connected via a network. Examples of such algorithms include leader election, general resource allocation, consensus strategies, synchronization and data access algorithms and many more. With the ubiquity of distributed computers, multi core systems and networked devices a basic knowledge of distributed algorithms and of the underlying fundamental design- and correctness principles becomes more relevant even for the non-specialist. Despite the heterogeneity found in distributed networks, the theoretical study of algorithms in this area has revealed a core set of fundamental principles both for the design and modelling as well as for proving correctness. Examples for these design principles include

- consensus (agreement among independent processes)
- electing a leader (i.e. selecting a distinguished process and breaking possibly symmetry)
- produce synchronicity
- select a consistent global view (global snapshot) etc.

In this seminar you will not only learn concrete algorithms but also these underlying unifying principles.

Although the algorithms are often compact, elegant and not very long, reasoning about them is difficult because of the uncertainty introduced by concurrency and asynchronicity. To deal with these challenges, researchers and practitioners have classified systems into

1. Synchronous network model,
2. Asynchronous shared memory model, and the
3. Asynchronous network model

We will cover them all and provide you with tools for reasoning about such systems. The prime, unifying model we will use to model distributed systems are *input/output automatons* (state machines). Examples for proof strategies covered include

- invariant assertions, and
- simulations.

2 Literature

The seminar is primarily based on the following books:

1. Nancy Lynch, *Distributed Algorithms*, Morgan Kaufman 1996
2. Gerard Tel, *Introduction to Distributed Algorithms*, Cambridge University Press 2000

For selected topics or talks there could be original research papers in addition. Both books will be available in our library[1] and will constitute a "Semesterapparat", i.e. one copy is always available for study or making photocopies in the library for the duration of this seminar.

3 Prerequisites

The course has the following prerequisites

- Ability to read English literature (primarily books, for some topics also research papers are useful)
- Basic command of discrete mathematics, in particular for proofs is necessary
- Basic understanding of graphs is helpful
- Knowledge of "normal", i.e. serialized non-distributed algorithms such as covered in the Bachelor module "Algorithmen und Datenstrukturen" helpful but not mandatory

4 Logistics

1. Weekly meetings, date and time to be announced (see schedule `http://www.fh-frankfurt.de/de/fachbereiche/fb2/studiengaenge/high_integrity_systems/timescale.html`)
2. There will be two to three introductory lectures about problem space and formalism by the lecturer, i.e. myself.
3. There is an e-Learning course in Moodle "Schäfer: HIS Distributed Algorithms - WS2011/12" which we use for disseminating information as well as uploading presentations: `https://elearning.fh-frankfurt.de/course/view.php?id=2576`, Password: HISDistAlg201112, please enroll yourself!

[1] at the latest end of September presumably

4. You have to apply for a topic (first-come, first-serve basis) by answering activity "2 Available Topics: Your Choice!" in the e-Learning course in Moodle, see section "Presentation Topics" for content! In case the number of students exceeds available slots, I might chose to allow working in pairs or to reject (to be announced). You can also suggest a topic related to the subject but not included in this list, see "talk 12", section "Misc" below!
5. You will prepare your talk (45-60mins) and get back to me with questions *prior* to presenting. We will use the "free weeks" in October to answer questions for the first presenters and the regular seminars for subsequent presenters.
6. You will submit your presentation via Moodle!
7. You will give your talk!
8. After the talk you will upload the final written, extended version (see below)!

Guidelines for giving a good presentation, you find here: `http://www.st.cs.uni-saarland.de/zeller/vortrag.pdf` (German only) or `http://www.mnlab.cs.depaul.edu/seminar/resources/tech-pres.pdf`.

5 Presentation Format

Your presentation must be in English and could be in either slide format, or a handout and you will use a black- or whiteboard for your talk. If you chose slides, the written, extended version (see next section) must contain additional explanatory notes to make them self-contained. You can use any of the following tools: LaTeX (report.sty, beamer.sty or powerdot.sty), MS Word, or OpenOffice.

For the report you find inspiration with regards to the format in the guidelines of the IEEE, see `http://standards.ieee.org/guides/style/2009_Style_Manual.pdf`, in particular have a look at the word templates `http://standards.ieee.org/resources/development/writing/writinginfo.html` or the LaTeX Templates `http://www.ieee.org/publications_standards/publications/authors_journals.html#sect2`!

6 Credit

You get credit for presenting a lecture on the subject chosen to the audience. The lecture has not only to be presented but also a written, extended version has to be submitted via the e-Learning platform in time (will be announced). *Regular* attendance is *mandatory*. An official written note of absence is required if you are unable to attend.

7 Timetable

1. Week 40: Preliminary discussion
2. Week 41-43: Two to three introductory talks and QA sessions for preparation
3. Week 44 onwards: Presentations

8 Presentation Topics

The following topics are available for talks (the chapter and section numbers resp. refer to Lynch's book):

Model	Talk	Date	Topics	Chapter
Synchronous network model	1	tbd	Leader Election – selecting a leader in a distributed manner	2, 3.1-3.3, 4.1
	2	tbd	Selected Distributed algorithms such as BFS or Minimum Spanning Trees	4.2,4.3, and 4.4
	3	tbd	Distributed consensus with link failures: Coordinated Attack Problem	5.1 and 5.2
	4	tbd	Commit problem in distributed databases	7.3
Asynchronous shared memory model	5	tbd	Mutual exclusion – Given access to critical sections	10.3, 10.6, and 10.7
	6	tbd	Resource Allocation – the famous Dining Philosopher problem	11.1, 11.2, and 11.3
	7	tbd	Consensus – getting processes to agree, even in case of failures	12 selected
Asynchronous network model	8	tbd	Leader Election – again, but now in the asynchronous case	15.1 and 15.2
	9	tbd	Synchronizers – achieving synchronicity in an asynchronous world	16 selected
	10	tbd	Logical Time – what is time anyway in a distributed world?	18.1, 18.2, and 18.3
	11	tbd	Global Snapshots – get consistent view even in absence of global time	19 selected
Misc	12	tbd	Free topic: You will suggest a topic, I will (most likely) approve!	NA