Difference of Distributed Systems Course at University of Tartu and Other European Universities

Asmar Hasanova

asmar@ut.ee

Student number: B56795

Supervised by Artjom Lind

Abstract

Distributed Systems Course is one of the core courses at most of the universities. Taking this course is compulsory for students from Computer Science and other related faculties. There could be students with difficult backgrounds, whom this course would be difficult. The purpose of this research is improve course's structure by adding useful and interesting topics, which have not covered during the course, and give suggestions to mitigating some very difficult topics.

1. Introduction

This report written based on research of the Distributed Systems course at the following universities.

- University of Tartu
- The University of Edinburgh
- University of Helsinki
- University of Waterloo

These universities' teaching method and structure including lectures, assignments and exercises of the Distributed Systems course compared with the University of Tartu's teaching method and structure of this course.

2. Distributed Systems course at The University of Edinburgh

The lecture topics

- Introduction
- Basic Algorithms: Broadcast and Aggregation
- Time and clocks
- Models: Thinking about communication and computation
- Clocks, Ordering and Global snapshots
- Leader Election| About course and assignment
- Mutual Exclusion
- Agreement and Multicast
- Operation Systems
- Termination detection
- Coloring and Maximal independent sets
- Distributed object based systems
- Computing in mobile and sensor networks
- Clouds, and Internet of things
- Networking issues in mobile systems
- Content Distribution Networks

- Final thoughts on Mobile, Modern distributed systems and the course

2.1 Topics, which had not covered at University of Tartu during the course

Basic Algorithms: Broadcast and Aggregation

In this lecture have more deeply covered topics such Network as a graph, computing sums in a tree, Communication complexity and Reducing Communication complexity, Global Message broadcast, Flooding dot Broadcast, Time complexity, Computing Tree from a network, Tree based broadcast, Observation on complexity, Bit complexity of communication.

In addition, there are in details touched to some mathematical topics, such as Unit disk graphs, Directed graphs, Big oh – upper bounds, Trees (BFS, spanning), Big Ω – lower bounds, Big θ – tight bounds etc.

At University of Tartu, there is a core course Discrete Mathematics, where we have learnt these topics thoroughly. Beside it, there are themes in this lecture, which have taught in other lectures at University of Tartu, for example, "Flooding for Broadcast" theme have explained in-depth at "Indirect communication" lecture. I consider giving this topic as separate lecture not so essential, however, can be added information about Communication complexity.

Termination detection

In this lecture-slides, described Termination detection – weight throwing, Dijkstra-Scholten etc. Given explanation to the questions like, "How do know when a distributed computation has ended?", tracking nodes if are in state "idle" Vs "Active" and so on.

It is quite interesting topic for including more information about Termination detection (weight throwing, Dijkstra-Scholten) to lecture, also this topic have given more detailed in "Distributed Systems Concepts and Design "book.

Coloring and Maximal independent sets

At this lecture have given themes such as, Coloring a graph, Application of color, the Independent set (IS), Maximum independent set (maxIS), Maximal IS (MIS), MIS algorithm, the vFast-MIS etc.

From point of my research, many Universities, also Distributed Systems books have not covered this topic. Personally speaking, I do not consider this an interesting topic for the Distributed Systems subject.

Computing in mobile and sensor networks. Mobile networking

Despite that, at University of Tartu, during the Distributed Systems course topic "Networking and internetworking" have covered, but had not given a description about Mobile and sensor networking and Mobile networking topics.

At Mobile and sensor networking lecture deep description about Mobile and Ubiquitous computing, Context aware computing, given example of: Indoor vs Outdoor, Context detection, networking in mobile systems, routing in ad hoc wireless networks, Dynamic Source Routing (DSR),

Routing Discovery: RREQ, Routing Discovery in DSR, As hoc On-Demand Distance Vector Routing (AODV), Sequence numbers in AODV, Sensor network, Data centric routing, Distributed Database, Distributed hash tables, Geographic Hash Tables, GHT, Rumor Routing.

At Mobile networking lecture-slides have described "How do mobile phones work?", information about Channel assignments, Spatial Reuse, also about the real mobility issues (Call, Handoff, Internet issues), Mobile IP, Mobile computing and so on.

I consider, Computing in mobile and sensor networks and Mobile networking are contemporary the most required topics and worth to add at the Distributed Systems course.

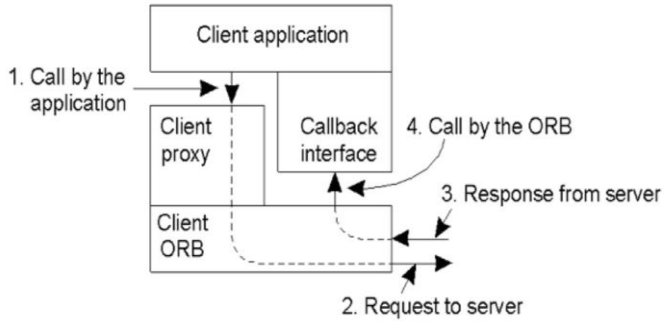
Clouds and Internet of things

In this lecture, they are more deeply described about Cloud Computing (information and examples about SAAS, PAAS, and IAAS), Internet of Things (Scalability, Security, and Privacy of Internet if Things) and so on.

At University of Tartu there are courses "Basics of Cloud computing", "Mobile and Cloud computing", where teaching these topics in-detail. So, I consider, adding this topic to lectures separately as a part of the Distributed Systems course not so necessary.

2.2 Overview of lectures of Distributed Systems course at The University of Edinburgh

At lectures have comprehensively covered Packets, Transport management (TCP and more slightly UDP), multicast, multicast IP, Distributed Operating System, Network Operating System, Remote procedure call (RPC), Common Object Request Broker Architecture (CORBA) topics.



[http://www.inf.ed.ac.uk/teaching/courses/ds/slides1516/Distributed-objects.pdf]

Explained COM and .NET and their relation, just mentioned about Remote method invocation (RMI), Peer-to-Peer systems, SOAP (Simple Object Access Protocol) and REST (REpresentational State Transfer) but not covered in detail like at University of Tartu. There are not any information about MPI (The Message Passing Interface), API, IP protocols.

2.3 Assignment during the course

In the assignment, students have to create a simulation of a network and implement the ring based Chang and Roberts algorithm for leader election among the nodes of the network. Also, given and input file, which contains the network; i.e., the nodes with their neighbors.

The assignment is divided in two parts.

Part A

- Reading the network specification from the input file.
- Construct the ring topology.
- Implement the Chang and Roberts algorithm for leader election among the nodes and initiate and execute elections according to the ELECT lines.

Part B

- This part handles node failures. When node X fails, we assume that the network knows this, and informs all neighbors of X.

Given example of Input files.

```
Node_id Neighbours
15 1 16
1 15 18 12 3
3 1 20 6 12
20 3 12 16
12 20 6 3 1 18
6 12 7 3 16
7 6 18 16
18 7 16 12 1
16 18 15 7 6 20
ELECT 5 18 7
ELECT 8 3
ELECT 15 12 1
ELECT 50 18
FAIL 20
FAIL 6
FAIL 3
```

```
Node_id Neighbours
1 3 6 8
3 1 2
2 3 5
5 2 4
4 5 7
7 4 6
6 7 8 1
8 6 1
ELECT 1 1 3 2 5 4 7 6 8
```

3. University of Helsinki

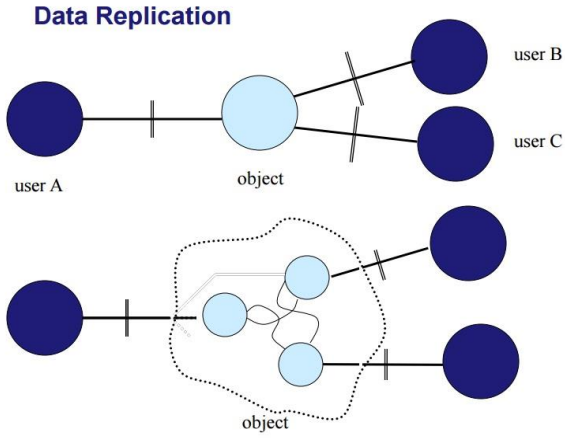
The lecture topics

- What is a distributed system?
- Interprocess communication
- Synchronization
- Replication and Consistency
- Fault Tolerance
- The Web
- Warehouse-Scale Computing

3.1 Topics below, had not covered at University of Tartu during the course

Replication and Consistency

At this lecture have given information about Data Replication (Reasons for Data Replication)



[https://www.cs.helsinki.fi/webfm_send/952]

Data-Centric Consistency Models, Consistency models (Strict, Linearizability, Sequential, Causal, FIFO, Entry Consistency, Eventual and Client-Centric Consistency), Push vs. Pull propagation, Information Dissemination Methods (Gossiping) Distribution Protocols (Replication placement, Update propagation, and Epidemic protocols), and Consistency protocols etc.

At University of Tartu during the Distributed Systems course also given information some of these themes such as, Gossiping, DNS server replication, Epidemic protocols. I consider the information which, given at University of Tartu is enough for this topic. Have given essential parts without looking aside.

Warehouse-Scale Computing

At this lecture have covered Warehouse-scale computing (WSC), WSC Data Centers, Architectural Overview (Storage, Networking, Storage hierarchy, Latency, bandwidth, capacity, Power usage, Handling failures), Workload and Software infrastructure (MapReduce), Failures and repairs etc. themes.

In my point of research, this topic not so deeply related to Distributed Systems, at giving it as a separate lecture not so essential.

3.2 Overview about lectures of Distributed Systems course at University of Helsinki

In lectures, comprehensively covered Multicomputer Operating Systems, Distributed Shared Memory Systems, Network Operating System, Middleware, Remote Procedure Calls (RPC), Multicasting, also, given information about Client-Server Architectures. However, not any information about Remote Method Invocation (RMI), Peer-to-Peer systems, SOAP (Simple Object Access Protocol) and REST (REpresentational State Transfer), MPI (The Message Passing Interface), API, IP protocols (IPv6 and IPv4), TCP and UDP transport protocols.

3.3 Assignments and exercises during the course

First Assignment: Communication and Synchronization

- Consult for a client who wishes to distribute a local application. Among other things, some database-accessing procedures will be relocated to their own node, which is responsible for handling mutual exclusion and serving requests in the order that they were sent.

Second Assignment: Chandy-Lamport distributed snapshot algorithm

- Implement a simple coordinated checkpoint algorithm. If you use Chandy-Lamport, assume the nodes are organized in a chain: o-o-o-o.

Third Assignment: The Domain Name System

- Students have to write an educational summary for a fellow student about DNS is and how it works as a distributed system, focusing on the point of the learning goals of the course.
- And cover at least the following topics:
  - ✓ Horizontal and vertical distribution.
  - ✓ Management of replicas
  - ✓ Fault tolerance

Fourth Assignment: Standardization

- Implementing an election algorithm from the book/slides on top of student's check pointing script from assignment 2, so that 'number' is the assigned number of the node, and 'result' stores the current coordinator. In other words, first run the coordinator election, then save state when the coordinator election is done.

There have given six weekly exercises during the course. Questions related to lectures.

4. University of Waterloo

The lecture topics

- Introduction
- Architectures and Models

- Computer Networks
- Distributed Objects and Remote Invocation
- Distributed Naming
- Distributed Filesystems
- Synchronization
- Replication
- Fault Tolerance
- Security

4.1 Topics below, had not covered at University of Tartu during the course

Replication

At University of Tartu, only one part "DNS server replication" of this topic has covered. At lecture-slides of the University of Waterloo very deeply given information about Replication (Object Replication, Chain Replication, Active Replication), Consistency (Client-Centric Consistency), Linearizability, Pull versus Push Protocols, Epidemic Protocols, Primary-Backup, Problems and solution to Replication Invocations etc.

As I mentioned above, about this topic given a lot of information at lecture-slides of the University of Helsinki, also in "Distributed Systems Concepts and Design" book and At University of Tartu during the course also given information some of these themes such as, Gossiping, DNS server replication, Epidemic protocols. Like motioned above given at University of Tartu is enough for this topic from my point of view.

Fault Tolerance

In this lecture-slides have given comprehensive information about Fault Types, Classification, Failures, Fault Tolerance Measures, Improvement of Dependability, Transactional Dependability, Hardware Redundancy, Process Redundancy, Fault Tolerance of Process Groups, Distributed Reliability Protocols, Two-Phase Commit (2PC), Three-Phase Commit (3PC), Transactions and Network Partitions etc.

From my point of view, this topic not deeply related to Distributed Systems. Can added information about Distributed Reliability Protocols, Transactions and Network Partitions to lecture would be interesting, but giving as a separate topic, I think not necessary.

4.2 Overview about lectures of Distributed Systems course at University of Waterloo

At University of Waterloo, during the Distributed Systems course in-detail have covered Client-Server Communication, Peer-to-Peer Systems, Protocols, Remote Procedure Call (RPC), Remote Method Invocation (RMI), CORBA, and API. Moreover, have given information about TCP and UDP, but not touched MPI (The Message Passing Interface) topic.

4.3 Assignments and exercises during the course

First Assignment:

- Given questions related to client's request and server's reply, TCP, RPC and RMI to support exceptions, Latency.

Second Assignment: (Group work)

- Students have to implement a Distributed Hash Table (DHT), a system that is both peer-to-peer (nodes act as

both clients and servers) and decentralized (system is self-regulating).

- A DHT, as one can surmise from its name, provides a very simple and familiar hash table interface: a key/value pair can be stored using a put operation, and the value for an associated key can but retrieved using a get operation.
- In addition, much like a standard hash table, each key is hashed to determine where it and its associated value should be stored.
- However, in place of determining hash buckets, the key hashes determine which machines are responsible for storing each key/value pair.

Third Assignment:

- Questions based on Replication, Quorum-based replication example, Ordering – Timestamp ordering, Multiversion timestamp ordering
- In addition, capability missing in your DHT from assignment 2 is replication. Students, have to describe, what modifications, would need to make DHT to incorporate replication schemes.

5. Comparison table

In the table 1 (see below) you can see structure of the Distributed Systems course at the different universities.

6. Conclusion

In a report I have researched Distributed Systems course's teaching method and structure at the four European universities. I thoroughly examined lecture-slides, assignments, exercises of this course at each university, then have compared them with each other. I have suggested which topics would be interesting for adding to the course.

The structure of this course completely different at The University of Edinburgh from other universities. The structure of the lectures' of the course at University of Waterloo almost the same with structure at University of Tartu. At both universities comprehensively have covered very important topics for Distributed Systems such as Client-Server Communication, RPC and RMI, TCP and UDP and so on. At University of Helsinki, half of the given assignment related to programming, but half of them related to theory. It would be good to add theoretical assignment for students who suffering from lack of programming background. In addition, workload of the weekly exercises is very high at the University of Tartu; it would better to reduce to half load of labs, like at University of Helsinki labs for each second week.

Table 1 – Course structure comparison			
University of Tartu	The University of Edinburgh	University of Helsinki	University of Waterloo
3 assignments 13 practical tasks	1 assignment	4 assignments weekly exercise session	3 assignments
6 ETCS	10 ETCS	4 ETCS	
Active participation: 10% Seminar classes exercises: 10% Assignments: 40% Exam: 40% (mid-term 15%; final exam 25%)	Final Exam: 75%. Assignments: 25%	Every second week exerciese:12 % Assignments: 12 % Final Exam: 42 % Max: 68%	Assignments: 30% Exam: 70% (mid-term 30%; final exam 70%)

7. References

1. "The University of Edinburgh School of Informatics" http://www.inf.ed.ac.uk/teaching/courses/ds/
2. "DSAssignment.pdf" http://www.inf.ed.ac.uk/teaching/courses/ds/assignment1516/DSAssignment.pdf
3. "University of Helsinki Faculty of Science" https://www.cs.helsinki.fi/en/courses/582417/2012/S/K/
4. "University of Waterloo" https://cs.uwaterloo.ca/~kdaudjee/courses/cs454/
5. "University of Tartu Institute of Computer Science" https://courses.cs.ut.ee/2015/ds/fall