Summary and Examinable Material

INF5040/9040 Autumn 2017
Eli Gjørven and Roman Vitenberg

November 21, 2016
Learning goals

- Provide a basic understanding of
  - fundamental principles, concepts and state-of-the-art
  - key technologies for realising distributed interactive systems of the future

- Gain practical experience of crafting a distributed application with a state-of-the-art platform

- Provide knowledge about today’s challenges in distributed processing technology, including
  - Communication paradigms
  - Mobility
  - Fault-tolerance
Overview of examinable material

- All lectures and lecture slides
- Selected chapters from the textbooks
- Completed programming assignments
From Coulouris et al (1/2)

- Chap 1: All
- Chap 2: All except 2.4.3
- Chap 4: 4.3
- Chap 5: All except 5.3.3
- Chap 6: 6.1, 6.2, 6.3 and 6.4
- Chap 8: All
- Chap 9: 9.1-9.4, 9.6
- Chap 10: All
Chap 14: All except 14.3 and 14.6
Chap 15: All
Chap 16: 16.1 – 16.4
Chap 17: 17.1 – 17.3: (17.5, 17.6 cursory knowledge)
Chap 18: 18.1 – 18.3
Chap 19: 19.1 – 19.3, 19.4.1, 19.4.2 and 19.6
Chap 20: 20.1 – 20.5
Chapter 1: All
Chapter 4: 4.1.2, 4.2.1, 4.2.2, 4.3.1, 4.3.2, 4.4, 4.5
Chapter 6: All except 6.1 and 6.4
Chapter 7: All except 7.3
Chapter 8: 8.4 (except 8.4.2), 8.5
Chapter 10: 10.1.1, 10.1.2, 10.2.1, 10.3, 10.4.1
Chapter 12: 12.1-12.4, 12.6
Chapter 13: 13.4.1

From Tanenbaum and van Steen (TvS) 2nd ed.
From Tanenbaum and van Steen 3rd edition

- pages 438 to 449 about Paxos
From “Internet of Things: A Hands-On Approach”

- Chapter 1:
  - 1.1
  - 1.1.1
  - 1.2.1
  - 1.2.2
  - 1.3 (Introduction)
  - 1.3.1
  - 1.3.2
  - 1.5 (Introduction)
  - 1.5.1-1.5.6

- Google Books link to the textbook:
  - https://goo.gl/NdIHZv
Additional examinable material for PhD students

- P. Eugster et. al.: Epidemic Information Dissemination in Distributed Systems
- Qi Zhang, Lu Cheng, and Raouf Boutaba: Cloud computing: state-of-the-art and research challenges
Mandatory programming assignments

- Knowledge related to the mandatory programming assignments *can* be tested during the exam
The exam will be conducted in the following way:

- Each student will be assigned a time and place (room) to meet. It is important that you show up at the precise time and place.
- You will be given a set of written questions that you have 30 minutes to prepare answers to (e.g., by making notes, diagrams, keywords or similar). You may not use any kind of aid (written or otherwise).
- After 30 minutes you will be brought to the examination room, where you will be allowed 20 minutes to present your answers to the written questions, followed by an additional 10 minutes of questioning from other parts of the examinable material.
- Bring picture id!
- Ifiadm will prepare a schedule for the exam (time and place for each student).
  - If you have particular wishes wrt your scheduling, contact IFI studadm
Introduction and design

- What is a distributed system
- Goals of distributed systems:
  - resource sharing, scalability, fault tolerance, ...
- Implications of distributed systems:
  - Concurrency, partial failure, unreliable and unsecure comm, ...
- Pitfalls when developing distribute systems
- Types of distributed systems
  - Distributed systems organized as middleware
System models

- Three types of system models
- Physical model: capture the hardware composition of a system
- Architecture models: defines the components of the system, the way they interact, and the way they are deployed in a network of computers
  - roles and responsibilities
  - placement strategies: client-server models (many variants), P2P, ...
  - architectural patterns
- Fundamental models: formal description of the properties that are common to all architecture models
  - interaction models (synchronous vs asynchronous, event ordering)
  - failure models
  - security models (not covered in this course, but see e.g., INF3190)
Communication paradigms

- Communication properties
  - Persistence, synchrony, time and space decoupling
- MPI
  - Motivation, main ideas, and uses
- MOM
  - Architecture, routing, applications
- Publish-subscribe
  - Architecture and properties
  - Subscription semantics and content-based routing
- Overview of overlay-based multicast
  - Desired properties
  - Small-world phenomenon
- Epidemic-based dissemination
  - Concepts and ideas
  - Push-based and pull-based mechanisms
  - Applications
Object-based distributed systems (1)

- Local procedure call
  - Parameter passing

- Remote procedure call
  - RPC steps
  - Parameter passing (by value or reference)
  - Client/server stubs

- Characteristics of distributed objects: Distributed objects execute in different processes.
  - Remote interfaces
  - Remote Object References

- Remote method invocation
  - Implementation of RMI
  - Elements of the RMI software
Object-based distributed systems (2)

- Object servers: support distributed objects
- Role of object adaptor.

**Principles of CORBA**
- Clients may invoke methods of remote objects without worrying about: object location, programming language, operating system platform, communication protocols, or hardware.

**Principles of Java RMI**
- Similar to CORBA but limited to a Java environment

**Complicating factors with remote objects**
Logical time
- Implementations: logical clocks & vector clocks

The snapshot problem
- Consistent global states
- [Chandy,Lamport] protocol

Distributed consensus
- Known impossibility results

Distributed mutual exclusion
- Requirements
- 3 Algorithms: centralised, ring-based, and [Ricart,Agrawala]

Distributed leader election
- Bully and ring-based algorithms
Paxos and Distributed transactions

- Paxos
  - Motivation and use examples
  - Roles of participants
  - Basic Paxos – full algorithm
  - Multi-Paxos and other variations

- Distributed transactions
  - Atomic commitment problem
  - 2PC protocol
    - State diagram
    - Recovery
Software components and distributed systems

- Rationale for components
- Basic design concepts for components
  - Component model with contractually specified required and provided interfaces
  - Connection model
  - Composition
  - Deployment model
- Application servers
  - Based on basic design concept for components
  - Implements the container pattern
- Distributed components technologies: EJB and Fractal
- Limitations of components
- Objects vs components with regard to
  - Inversion of control pattern and separation of concerns
  - Explicit vs. implicit middleware
  - Container, lifecycle and deployment
Motivation and objectives
 Placement of replicas
 Propagation of updates among the replicas
   Consistency guarantees
 Replication schemes
   Active and passive replication
 Group communication
   Group membership, view synchrony, ordered message delivery
- P2P systems distribute processing load and network traffic between all nodes that participate in a distributed information system.
- P2P systems are self-organising and independent of any central entity.
- The efficiency critically depends on algorithms for distributing the data among a high number of nodes and subsequently locating the data.
- P2P middleware is an application-independent software layer that implements a "routing overlay".
- Case study and evaluation: Pastry.
Mobile and ubiquitous computing

- Mobile and ubiquitous systems are volatile
- Applications in mobile and ubiquitous environments are
  - Integrated with the physical world through sensing and context awareness
  - Adaptive to changes in the physical circumstances by changing behavior
- Common system model for mobile and ubiquitous computing (and their subfields). Elements of the model
  - smart spaces
  - device model
  - volatile connectivity
  - spontaneous interoperation (role of discovery services, interoperability)
- RFID, Wireless Sensor Networks, Internet of Things (components and design models)
Multimedia applications require mechanisms that enable them to handle large amounts of time dependent data

- Streaming vs. RMI

- Most important mechanisms: media synchronization and QoS/resource management

- QoS driven resource management
  - admission control (by mapping QoS requirements to resource needs)
  - scheduling function (making resources available (CPU, memory, bw, ..) when needed)
  - QoS models for streaming

- Streaming over the Internet
  - Best effort, no guarantees.
  - Compensating for quality degradation
    - Jitter-compensation (different buffering strategies)
    - Compression
    - Traffic shaping (leaky bucket, token bucket)
    - Media scaling (stream adaptation)
Web-based and cloud systems

- Communication architecture
  - Web proxies
  - Proxy-based caching (hierarchical and cooperative)
  - Server-side backend
    - Content-distribution networks

- Web Services
  - Applications
  - Architecture
    - URIs, WSDL, SOAP
  - Comparison with distributed objects and CORBA
  - RESTful services

- Cloud systems
  - Principles and characteristics
  - IaaS, PaaS, and SaaS
A template for many (but not all) exam questions

- For a given topic A
  - Present the motivation
  - Explain the main problem and challenge
  - Present the solution studied in the course
  - Reason about the solution, compare it with other solutions, name advantages and shortcomings
  - Explain limitations and give an example when the solution does not work