University of Freiburg, Germany
Department of Computer Science

Distributed Systems

Chapter 1 Introduction, Motivation, & Organization

Christian Schindelhauer

23. April 2012

1.1: Organization

Prof. Dr. Georg Lausen

- Databases and Information Systems
- lausen@informatik.uni-freiburg.de

Prof. Dr. Christian Schindelhauer

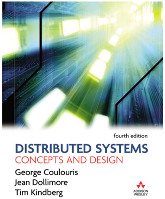
- Computer Networks and Telematics
- schindel@informatik.uni-freiburg.de

Organization

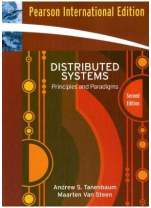
- Web-page
  `http://cone.informatik.uni-freiburg.de/cone_teach/cone_teach_current/distributed-systems-ss2012`
  - with slides, exercise, literature
- Forum
  `http://archive.cone.informatik.uni-freiburg.de/forum3/viewforum.php?f=10`
- Lecture
  - Monday 10-12, room 101-00-010/014
  - Friday 14-17, room 101-01-016 (in rotation with exercise)

Literature

- *Distributed Systems: Concepts and Design*. G. Coulouris, J. Dollimore, T. Kindberg. Addison Wesley, fourth edition 2005.



- *Distributed Systems*. A.S. Tanenbaum, M. Van Steen. Pearson Int. Edition, 2007.



- Further literature during the lecture

Lectures & Exercise 1st half

- 23.04.2012 Lecture: Introduction, Motivation & Organization
- 27.06.2012 Lecture: System Models
- 30.04.2012 Lecture: Time & Global States I
- 04.05.2012 Exercise
- 07.05.2012 Lecture: Time & Global States II
- 11.05.2012 Lecture: Time & Global States III
- 14.05.2012 Lecture: Coordination & Agreement I
- 18.05.2012 Exercise
- 21.05.2012 Lecture: Coordination & Agreement II
- 25.05.2012 Lecture: Coordination & Agreement II
- 04.06.2012 Lecture: Coordination & Agreement IV
- 08.06.2012 Exercise

Lectures & Exercise 2nd half

- 11.06.2012 Lecture: Transactions and Concurrency Control I
- 15.06.2012 Lecture: Transactions and Concurrency Control II
- 18.06.2012 Lecture: Transactions and Concurrency Control III
- 22.06.2012 Exercise
- 25.06.2012 Lecture: Distributed Transactions I
- 29.06.2012 Lecture: Distributed Transactions II
- 02.07.2012 Lecture: Distributed Transactions III
- 06.07.2012 Exercise
- 09.07.2012 Lecture: Replication I
- 13.07.2012 Lecture: Replication II
- 16.07.2012 Lecture: Applications Peer-to-Peer-Networks
- 20.07.2012 Exercise

Exercises & Exam

Exercises

- Voluntary exercises
- Every two weeks, two hours
- 04.05.2012, 18.05.2012, 22.06.2012, 06.07.2012

Exam

- Master students: written closed exam book (90 minutes)
- Bachelor students: oral exam for
- Register online (in-time)
- Dates to be announced

Related Lectures

- This semester
  - Network Algorithms (Kuhn)
  - Computer Networks / Rechnernetze I (Systeme II — Schindelhauer)
- Required knowledge:
  - Operation Systems/Betriebssysteme (Systeme I — Scholl)
- Continuing
  - Data Bases and Information Systems/Datenbanken und Informationssysteme (Lausen)
  - Peer-to-Peer Networks (Schindelhauer)

Network Algorithms

- Lecture by Prof. Fabian Kuhn
- a world-renown expert in this field
- Schedule
  - Tuesdays 8-10, room 101-01-018
  - Wednesdays 12-14, room 101-01-018
- Topics
  - distributed algorithms in networks
  - how to design algorithms for networks and distributed systems.
  - fundamental principles and techniques for distributed algorithms
- no specific prerequisites necessary for this lecture

1.2: Motivation

Distributed Systems are everywhere!!

- The Internet
- WWW
- Local Area Networks
- Multi-core processors
- Smart phones
- Massive Multiplayer Games
- Peer-to-Peer Networks
- Data centers
- . . .

Special Problems

Distributed Systems have special problems:

- How to organize a distributed system
- There is no global time
- Agreement with lazy, faulty and malicious partners
- Coordination of heterogeneous partners

1.3: Introduction

Definition: Distributed System (DS)

In a distributed system hardware or software components located at networked computers communicate and coordinate their actions **only by passing messages.**

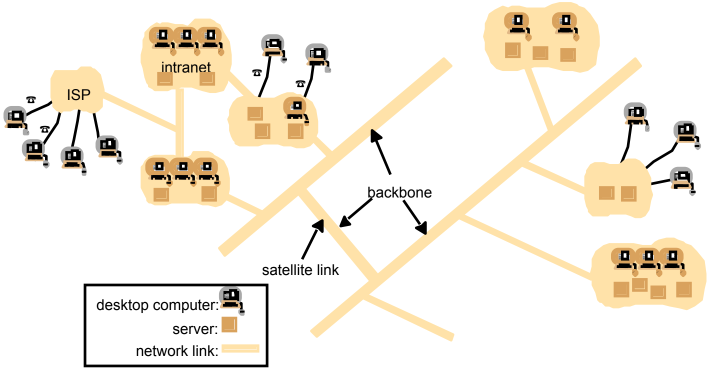
Consequences

- Concurrency
- No global clock
- Independent failures

Examples of DS

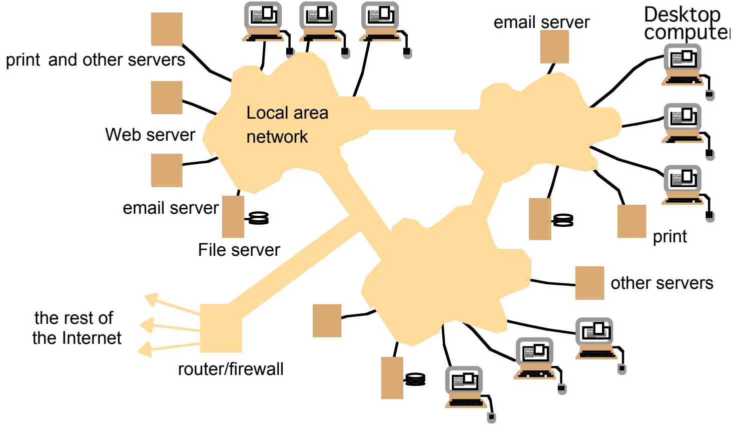
- The Internet
- Intranets
- Mobile and ubiquitous computing

The Internet



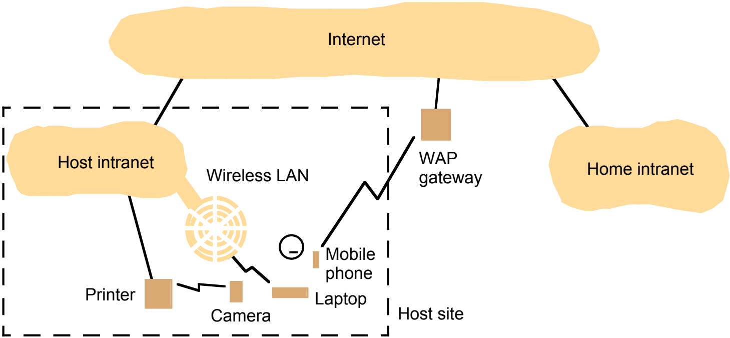
from *Distributed Systems – Concepts and Design*, Coulouris, Dollimore, Kindberg

A Typical Intranet



from *Distributed Systems – Concepts and Design*, Coulouris, Dollimore, Kindberg

Portable and Handheld Devices in a Distributed System



from *Distributed Systems – Concepts and Design*, Coulouris, Dollimore, Kindberg

Challenges of DS: Heterogeneity

- networks
- computer hardware
- operating systems
- programming languages
- implementations

Definition: Middleware

is a software layer that provides a programming abstraction wich masks the heterogeneity of the underlying networks, hardware, operation systems and programming languages.

Definition: Mobile Code

refers to code that can be sent from one computer to another and run at the destination.

Challenges of DS: Openness

By definition: the key interfaces of **open systems** are published

- Open distributed systems provide uniform communication mechanism
- Open DS publish interfaces for access to shared resources
- Open DS can be constructed from heterogeneous hardware and software
- Open DS must be carefully tested and verified.

Challenges of DS: Security

Components

- confidentiality
- integrity
- availability

Typical cases

- A doctor requesting access to hospital data.
- Electronic commerce and banking

Unsolved recurity challenges

- Denial of service attacks
- Security of mobile code

Challenges of DS: Scalability

A system is described as scalable

if it remains effective when there is a significant increase in the number of resources and the number of users.

- Controlling the cost of physical resources
- Controlling the performance loss
- Preventing software resources running out
- Avoiding performance bottlenecks

Challenges of DS: Failure Handling

Failures in a DS are partial. Some components fail, while other continue to function.

- Detecting failures
- Marking failures
- Tolerating failures
- Recovery from failures
- Redundancy

Challenges of DS: Concurrency

- Services and applications provide resources that can be shared
- Resources can be accessed at the same time
- A shared resource in a DS must ensure correct operation in a concurrent environmnent
- Operation must be synchronized such that the data of a shared object remains consistent

Challenges of DS: Transparancy

1. Access transparency

   enable local and remote resource to be accessed with identical operations
2. Location transparency

   access without knowledge of their physical location
3. Concurrency transparency

   concurrently operate severely processes using shared resources
4. Replication transparency

   enables multiple instances of recources to be used to increase reliability and performance without the users knowing
5. Failure transparency

   concealment of faults, allowing users to complete their tasks despite failures
6. Mobility transparency

   allows the movement of resources and clients without affection the operation
7. Performance transparency

   allows the system to be reconfigured to improve the performance as loads vary
8. Scaling transparency

   allows the system and application to expand in scale without a change to the system

End of Section 1