Distributed Systems
分布式系统

Instructor: Hongwei Du

（堵宏伟）

Class Description

- Instructor
  - Hongwei Du（堵宏伟）
    Email: hwdu@hit.edu.cn
- Tutor
  - Qiang He(贺强）
    Email:heqiang96@126.com
    Mobile:13728997185
  - Huizhen Wang(王慧珍)
    Email:20S151167@stu.hit.edu.cn
    Mobile:13670075590
- QQ Group: 616344207

Schedule Time:
(T5503)

November 3– December 24	Wednesday 8:30-10:15 Friday 14:00-15:45

【2021秋】本科生分...

群号：616344207

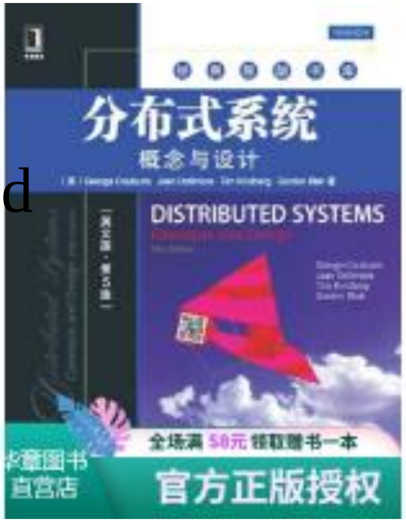
扫一扫二维码，加入群聊。

Textbook

- Distributed Systems: Concepts and Design

  George Coulouris, Jean Dollimore, Tim Kindberg and Gordan Blair(5th edition)

- Distributed Systems: Principles and Paradigms, Andrew Tanenbaum and Maarten Steen (2nd edition)
- Lecture notes



Pre-requirement:

- Principles of networks
- Operating systems
- Java/c programming

Topics

- Introduction
- Distributed Systems models
- Distributed Time and Clock Synchronization
- Socket Communication
- Remote Method Invocation(RMI)
- Group Communication
- Mutual exclusion & election algorithms
- Replication

…………..

Grading

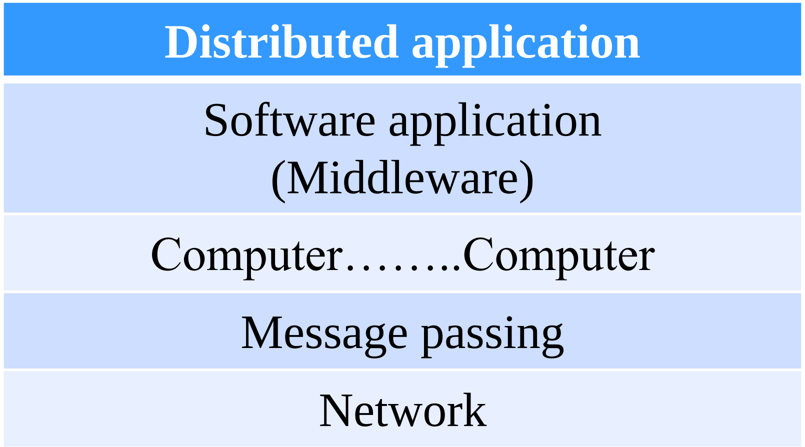
- Assignments 作业 (10%)
- Course projects 课程设计(20 %)
- Final exam 期末考试(70 %)

Motivation

- Resource sharing
  - Computers connected by the network and share resources.
  - Hardware sharing, software sharing, data sharing, service sharing. media stream sharing.
- Collaborative computing
  - Parallel computing, distributed computing

Definition

What is
a distributed system?



Distributed application

Software application
(Middleware)

Computer……..Computer

Message passing

Network

A distributed system is defined as one in which components at networked computers communicate and coordinate their actions only by passing messages.

- Distributed Application

- Middleware (Distributed core layer)
  - RMI, CORBA, DCOM….
- Network
  - Mobile phone networks, corporate networks, factory networks, campus networks, home networks, in-car networks, wireless sensor networks, etc.

Distributed Applications

Finance and commerce	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
The information society	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and MySpace.
Creative industries and entertainment	online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
Healthcare	health informatics, on online patient records, monitoring patients
Education	e-learning, virtual learning environments; distance learning
Transport and logistics	GPS in route finding systems, map services: Google Maps, Google Earth
Science	The Grid as an enabling technology for collaboration between scientists
Environmental management	sensor technology to monitor earthquakes, floods or tsunamis

Features

The distributed system features.

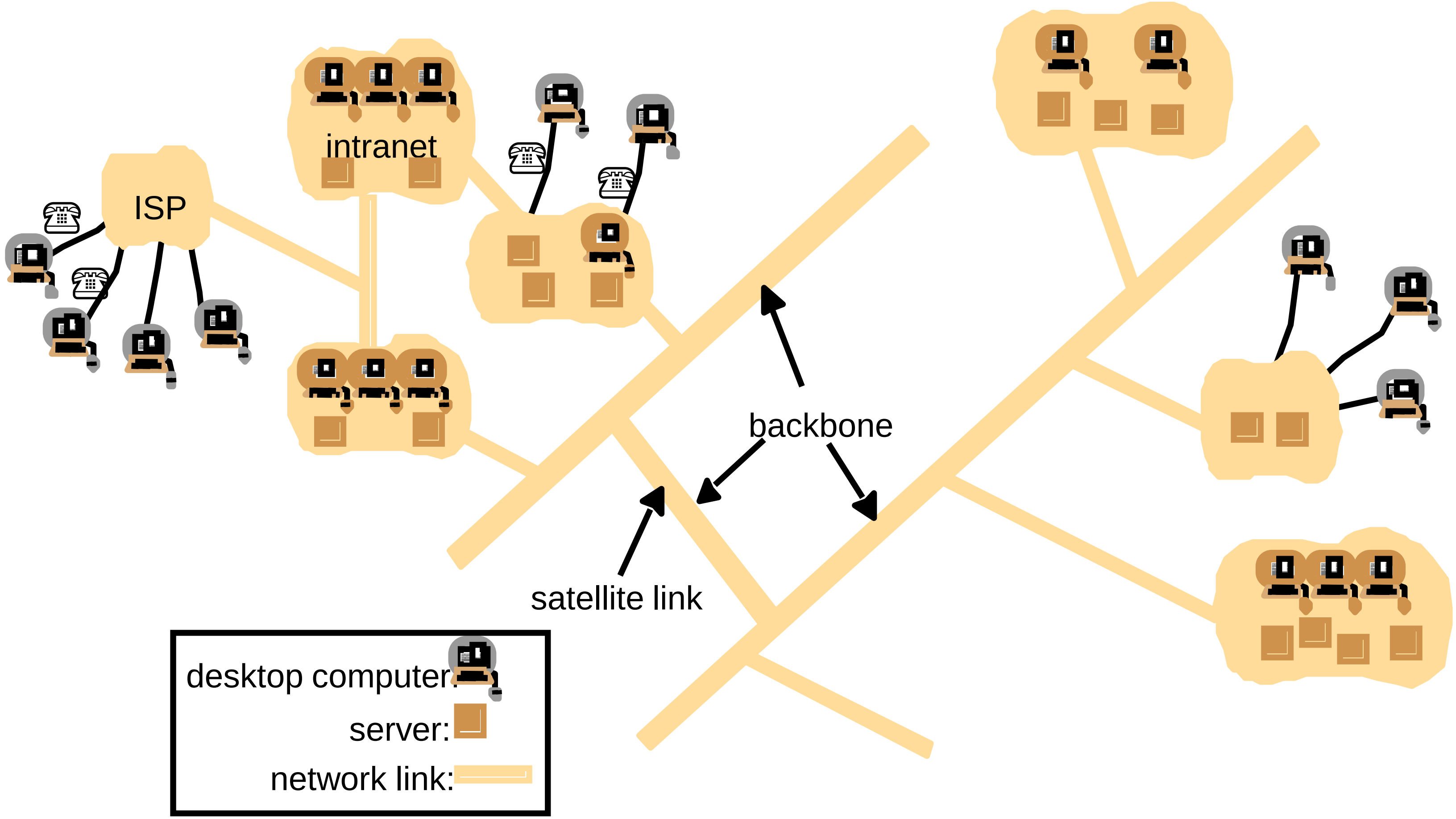
- Concurrency
  - Multi-process and multi-threads concurrently execute, share resources.
- No global clock
  - Program coordination depend on message passing.
- Independent failure
  - Some processes failure, can not be known by other processes.

Examples of Distributed Systems

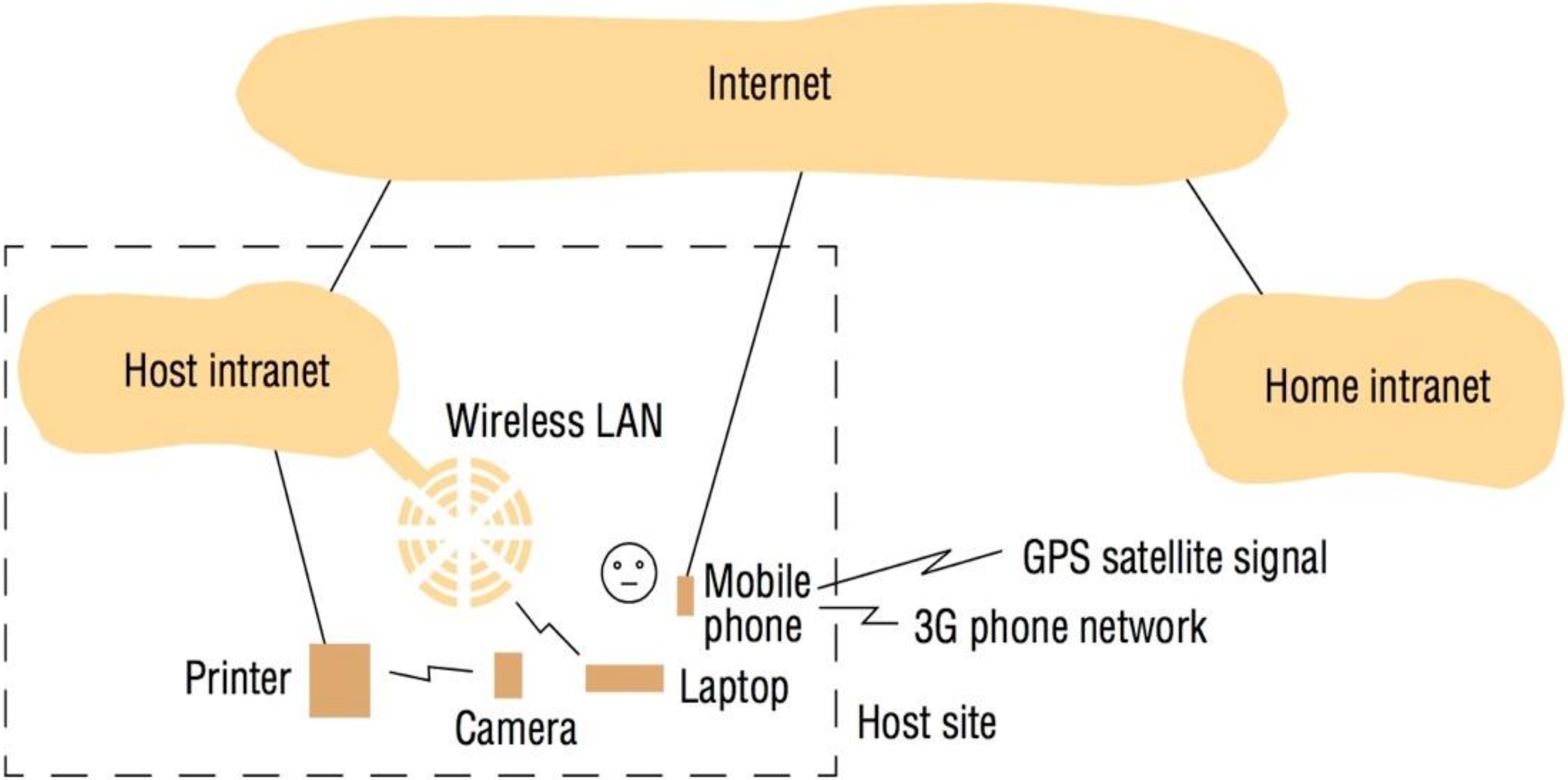
- The Large-scale Distributed System

  The Internet & Intranet
- Typical Distributed System
  - DNS service
  - Distributed file system
  - Global position system (GPS)
- New trend Distributed System
  - Mobile computing
  - P2P (BT, Emule)
  - Cloud computing

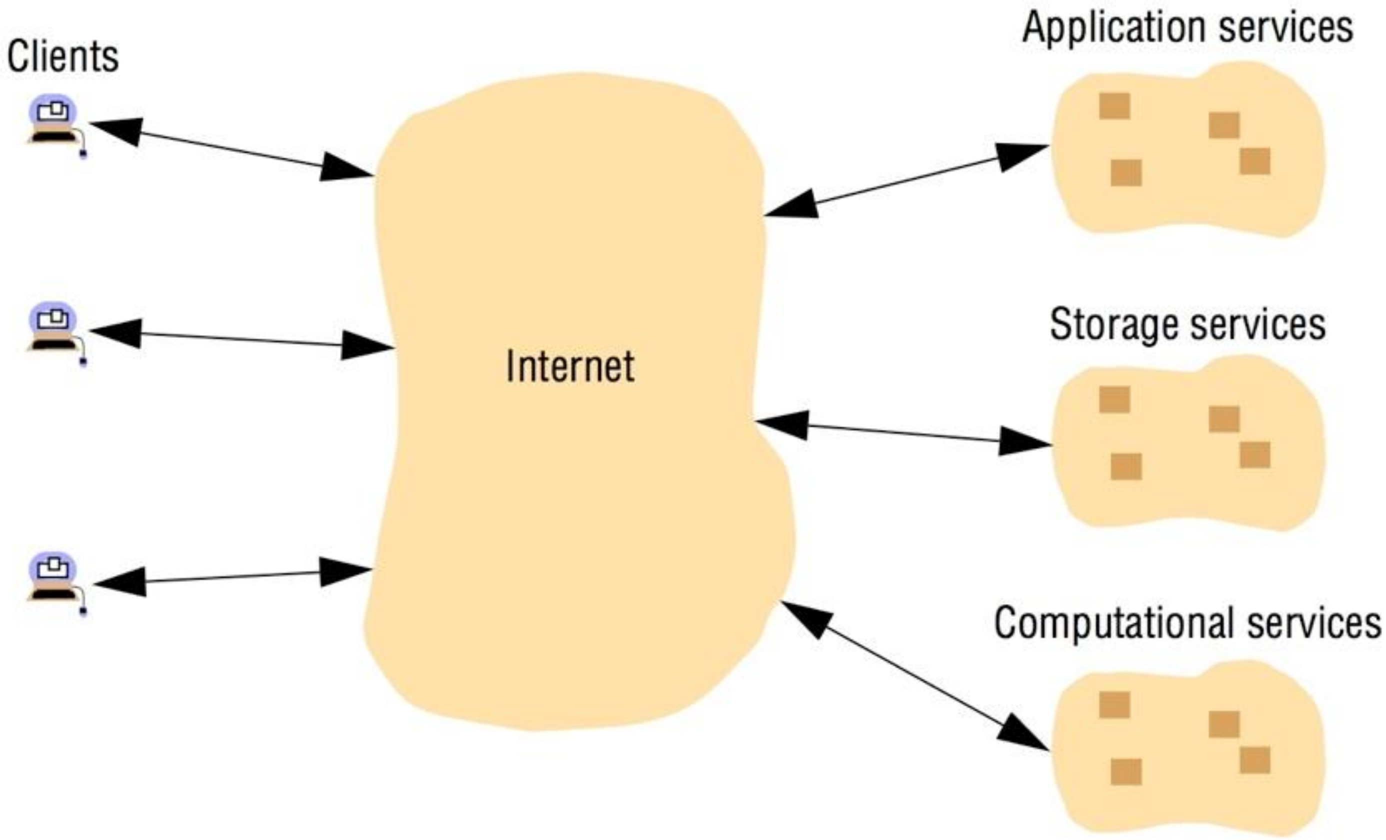
Examples of Distributed Systems (Internet)



Examples of Distributed Systems (Mobile Computing)



Examples of Distributed Systems (Cloud computing)



Challenges-Heterogeneity

- Middleware
  - Apply to a software layer that provides a program abstraction as well as masking the heterogeneity of the underlying layers (networks, hardware, operating systems and programming languages).

  Example: Java RMI

- Mobile code
  - Program code that can transferred from one computer to another and run at the destination. Example: Java Applet.
  - The Java virtual machine (JVM) provide a way of making code executable on a variety of host computers.

Challenges-Openness

- Computer System Openness
  - Determines whether the system can be extended and reimplemented in various ways. For example: UNIX.
- Distributed System Openness
  - The degree to which new resource-sharing services can be added and be made available for use by a variety of client programs.
  - RFC (‘Request For Comments’)

Challenges-Security

- Confidentiality(机密性）
  - Protection against disclosure to unauthorized individuals.
- Integrity （完整性）
  - Protection against alteration or corruption.
  - e.g. Checksum (校验和）
- Availability（可用性）
  - Protection against interference with the means to access the resources.

Challenges-Scalability

- Controlling the cost of physical resources
  - As the demand for a resource grows, it should be possible to extend the system, at reasonable cost, to meet it.
- Controlling the performance loss
  - Consider the management of a set of data whose size is proportional to the number of users or resources in the system.
- Preventing software resources running out
  - IPv4, IPv6…..
- Avoiding performance bottlenecks
  - In general, algorithms should be decentralized to avoid having performance bottlenecks.

Challenges-Scalability

Date	*Computers*	*Web servers*	*Percentage*
1993, July	1,776,000	130	0.008
1995, July	6,642,000	23,500	0.4
1997, July	19,540,000	1,203,096	6
1999, July	56,218,000	6,598,697	12
2001, July	125,888,197	31,299,592	25
2003, July	~200,000,000	42,298,371	21
2005, July	353,284,187	67,571,581	19

Challenges-Failure handling

- Detecting failures
  - Some failure can be detected.
- Masking failures
  - Some failure that have been detected can be hidden or made less severe.
- Tolerating failures
  - Most of the services in the Internet do exhibit failures.
- Recovering from failures
  - Recovery involves the design of software so that the state of permanent data can be recovered or ‘roll back’ after a server has crashed.
- Redundancy
  - Services can be made to tolerate failures by the use of redundant components.

Challenges-Concurrency

- Consistent
  - Multi-thread concurrent access the sharing resource.
- Performance

Challenges-Transparency

- *Access transparency*: enables local and remote resources to be accessed using identical operations.
- *Location transparency*: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).
- *Concurrency transparency*: enables several processes to operate concurrently using shared resources without interference between them.
- *Replication transparency*: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Challenges-Transparency

- *Failure transparency*: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.
- *Mobility transparency*: allows the movement of resources and clients within a system without affecting the operation of users or programs.
- *Performance transparency*: allows the system to be reconfigured to improve performance as loads vary.
- *Scaling transparency*: allows the system and applications to expand in scale without change to the system structure or the application algorithms.

Challenges-Quality of Service

- Reliability
- Security
- Performance
- Adaptability

Conclusion

- Distributed system is everywhere.
- The motivation of constructing a distributed system is resource sharing and collaborative computing
- Distributed system features.
  - Concurrency
  - No global clock
  - Independent failure
- Distributed system challenges.
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Failure handling
  - Concurrency
  - Transparency
  - Quality of Service