分布式系统
01-概述

Weixiong Rao 饶卫雄
Tongji University 同济大学软件学院
2023 秋季
wxrao@tongji.edu.cn

同济大学软件学院
School of Software Engineering, Tongji University

本课程内容结构

- **基础篇 (5周)**
  - 分布式系统的特点及模型
  - 网络及互联网、进程间通信、间接通信
- **原理篇 (6周)**
  - 容错机制
  - 副本管理
  - 一致性模型
  - 可扩展性设计
- **案例篇 (6周)**
  - 分布式文件系统
  - 分布式存储系统
  - 分布式编程框架

同济大学软件学院
School of Software Engineering, Tongji University
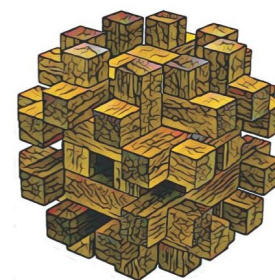
参考书籍

- **分布式系统:概念与设计(原书第5版)**
  - 作者:George Coulouris等
  - 译者: 金蓓弘 马应龙
  - 机械工业出版社
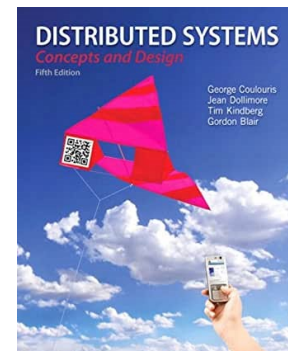
- **Distributed Systems: Principles and Paradigms (Third edition).**
  - Published by Maarten van Steen, 2017. Van Steen, Maarten , Tanenbaum, Andrew.
  - Free download from https://www.distributed-systems.net
  - 分布式系统原理与范型(第2版) 清华大学

同济大学软件学院
School of Software Engineering, Tongji University

课程评分

课后作业	**40%** (4次)

编程作业	**40%**
1. Linux脚本编程	8% (第3周交作业)
2. Socket网络编程	10% (第6周交作业)
3. 实现一个分布式数据查询系统	12% (第10周交作业)
4. 实现一个分布式节点成员服务模块	15% (第15周提交)
课程项目	**20%**
分布式编程 编程实践	(第18周提交并演示)

所有的编程作业和项目均需要在**Linux环境**完成和验证

预备知识: 操作系统、计算机网络、编程语言如Java/Python/Scala

同济大学软件学院
School of Software Engineering, Tongji University

1分钟讨论

同济大学软件学院
School of Software Engineering, Tongji University

本课程内容结构

- **基础篇 (5周)**
  - ◆ 分布式系统的特点及模型 ✔
  - ◆ 网络及互联网、进程间通信、间接通信

- **原理篇 (6周)**
  - ◆ 容错机制
  - ◆ 副本管理
  - ◆ 一致性模型
  - ◆ 可扩展性设计

- **案例篇 (6周)**
  - ◆ 分布式文件系统
  - ◆ 分布式存储系统
  - ◆ 分布式编程框架

本次课程内容

1. 基本概念
2. 常见示例
3. 发展趋势
4. 资源共享
5. 分布式系统的设计难点
6. 小结

同济大学软件学院
School of Software Engineering, Tongji University

1. 基本概念

- **What is a Distributed System什么是分布式系统?**
  - A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages

- **Characteristics of distributed systems分布式系统的特点:**
  - concurrency of components,
  - lack of a global clock,
  - and independent failures of components.
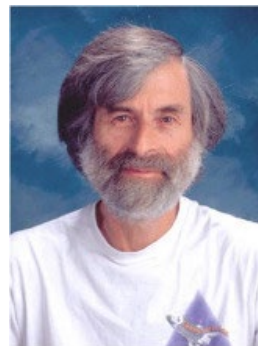

因为在分布式计算方面的杰出贡献，获得ACM颁发的2013年度图灵奖

- **Leslie Lamport :-)**
  - *You know you have a distributed system when the crash of a computer you've never heard of stops you from getting any work done!*

- **Prime Motivation: to share resources**

2．常见的分布式系统

- Transactional applications - Banking systems
- Manufacturing and process control
- Inventory systems
- General purpose (university, office automation)
- Communication – email, IM, VoIP, social networks
- Distributed information systems
  - WWW
  - Cloud Computing Infrastructures
  - Federated and Distributed Databases

同济大学软件学院
School of Software Engineering, Tongji University
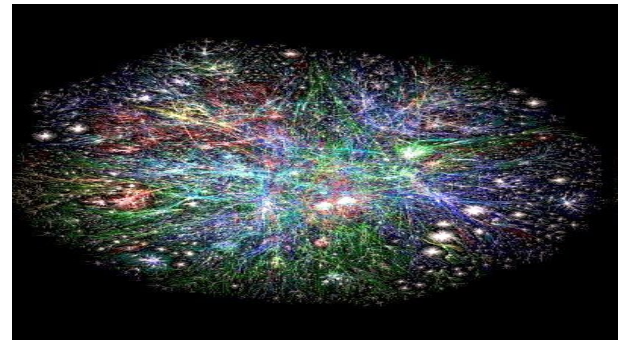
Web Search 搜索引擎

- **The global number of searches has risen to over 10 billion per calendar month**

- **The task of a web search engine**
  - ① index the entire contents of the World Wide Web
  - ② analyze the entire web content
  - ③ carry out sophisticated processing on this enormous database

---
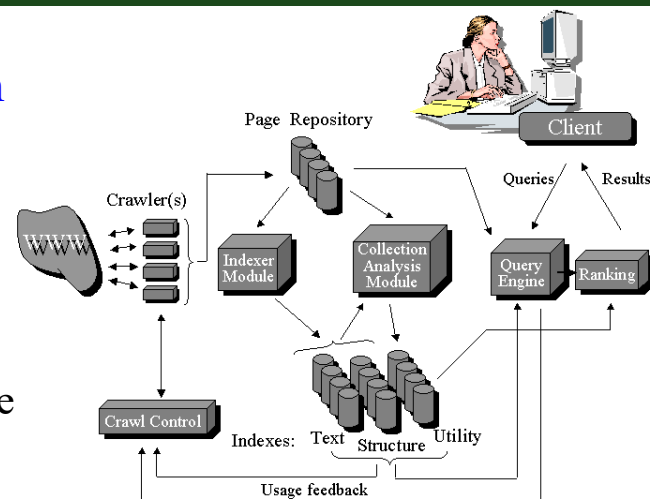
Google: a sophisticated distributed system infrastructure to support Web search

- One of the largest and most complex **distributed systems** installations in the history of computing and hence demands close examination.

Highlights of Google infrastructure include

- an underlying physical **infrastructure** consisting of **very large numbers of networked computers** located at **data centers all around the world**;
- a **distributed file system** designed to support **very large files** and heavily optimized for the style of usage required by search and other Google applications (reading from files at high and sustained rates);
- an associated structured **distributed storage system** that offers fast access to **very large datasets**;
- a **lock service** that offers distributed system functions such as **distributed locking and agreement**;
- a **programming model** that supports the management of **very large parallel and distributed computations** across the underlying physical infrastructure.

Google PageRank算法

- PageRank网页**排名**算法, 是一种由根据网页之间相互的超链接, 计算网页排名, 以评价网页的**相关性**和**重要性**，在搜索引擎优化操作中是经常被用来评估网页优化的成效因素之一。

- Google公司创办人拉里·佩奇Larry Page之姓来命名, Google的创始人拉里·佩奇和谢尔盖·布林Sergey Brin于1998年在斯坦福大学发明了这项技术，PageRank是谷歌的商标, 但该技术专利权属于斯坦福大学, 而非谷歌公司。

Yahoo通过人工维护页面目录

- Larry Page于1973/3/26出生在美国密歇根州东兰辛市的一个犹太家庭，1995密歇根大学本科, 1998 斯坦福大学硕士

- 斯坦福大学博士学习期间: 分析万维网World Wide Web的**数学**特性, 利用通过页面的超链接构建**超大图结构**, 导师Terry Winograd鼓励Larry Page探索该方向的研究
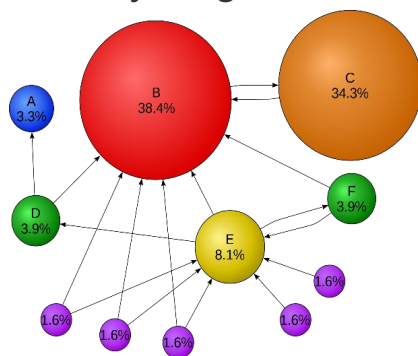
PageRank原理
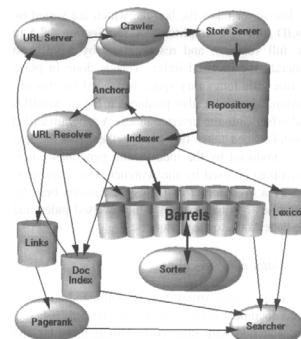
Google基本构架

- 谢尔盖·布随后加入Larry Page的研究项目"BackRub"并发表了有关PageRank和Google结构的论文"**The Anatomy of a Large-Scale Hypertextual Web Search Engine**", 该论文是互联网年代下载量最高的学术论文之一

The anatomy of a large-scale hypertextual web search engine
S Brin, L Page - Computer networks and ISDN systems, 1998 - Elsevier

同济大学**软件学院**
School of Software Engineering, Tongji University

大型多人在线游戏
Massively multiplayer online games (MMOGs)

- **EVE online: the largest online game, a *client-server* architecture**
    - a single copy of the state of the world is maintained on a **centralized server**
    - and accessed by client programs running on players' consoles or other devices
    - To support large numbers of clients, the server consists of **a cluster architecture featuring hundreds of computer nodes**
    - Why a client-server architecture
        - helps the management of the virtual world and the single copy also eases consistency concerns.
        - to ensure fast response through optimizing network protocols and a rapid response to incoming events.
        - the load is partitioned by allocating individual 'star systems' to particular computers within the cluster, with highly loaded star systems having their own dedicated computer and others sharing a computer.
        - Incoming events are directed to the right computers within the cluster by keeping track of movement of players between star systems.

- **EverQuest: more distributed architecture**
    - The universe is **partitioned** across a (potentially very large) number of servers that may also be geographically distributed.
    - Users are dynamically allocated a particular server based on current usage patterns and also the network delays to the server

- **Research on completely decentralized approaches based on peer-to-peer (P2P) technology**

My PhD Thesis: Efficient Content Dissemination in Large-Scale Distributed Systems 2009

同济大学软件学院
School of Software Engineering, Tongji University

3.发展趋势

■ **近年来分布式系统呈现了明显的发展和变化**

- ◆ Pervasive networking and the modern Internet → 网络无处不在, 4G/5G
- ◆ Mobile and ubiquitous computing → 智能手机、穿戴设备…
- ◆ Distributed multimedia systems → 视频播放, 爱奇艺、Youtube等
- ◆ Distributed computing as a utility → 云计算

同济大学软件学院
School of Software Engineering, Tongji University

Pervasive networking and the modern Internet

- Networking has become a pervasive resource and devices can be connected (if desired) at any time and in any place

intranet

ISP

backbone

satellite link

desktop computer:
server:
network link:
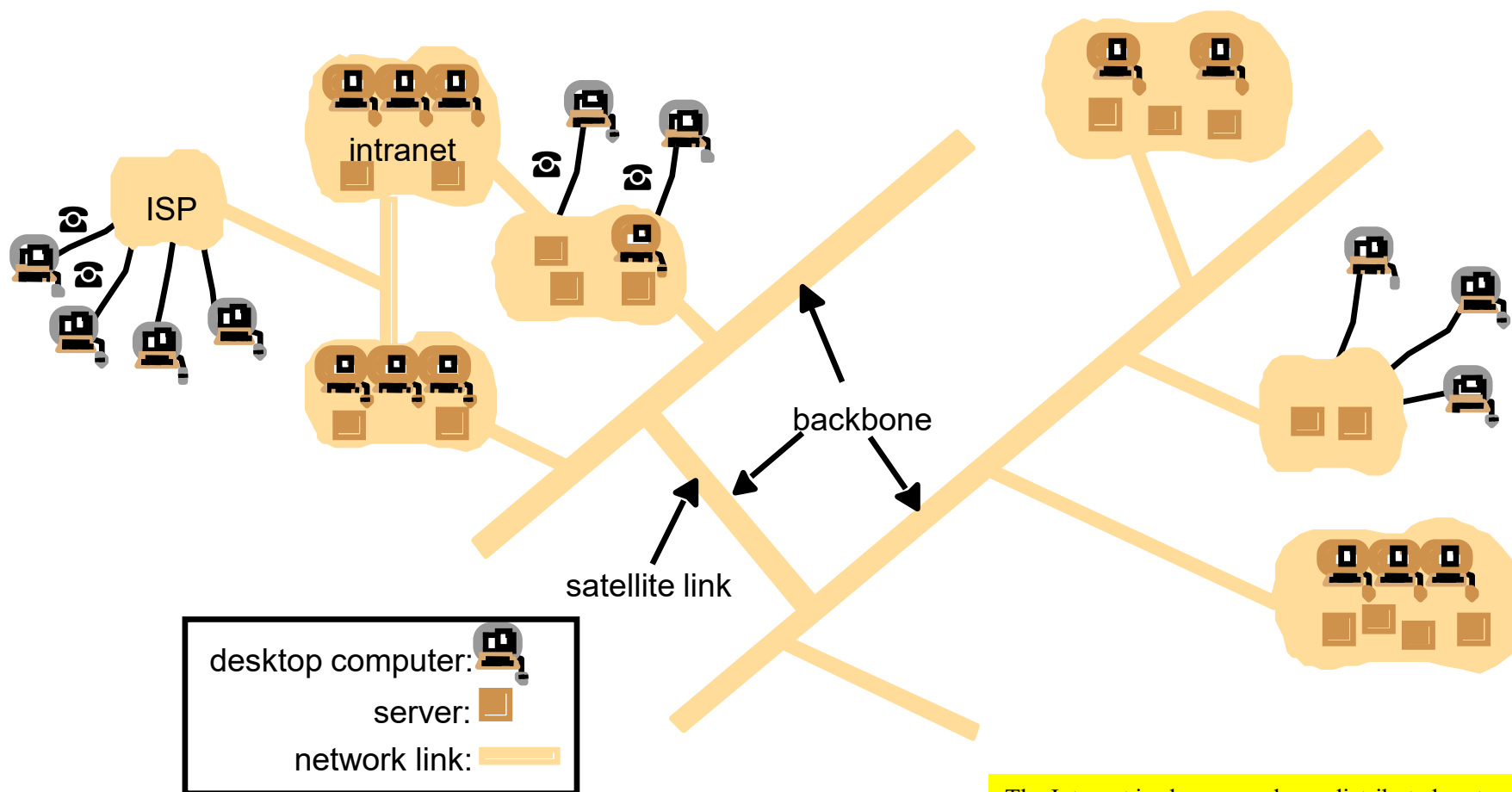
The Internet is also a very large distributed system

同济大学软件学院
School of Software Engineering, Tongji University

Mobile and ubiquitous computing

- Laptop computers

- Handheld devices (mobile phones, smart phones, tablets, GPS-enabled devices, PDAs, video and digital cameras)

- Wearable devices (smart watches, glasses, etc.)

- Devices embedded in appliances (washing machines, refrigerators, cars, etc.)
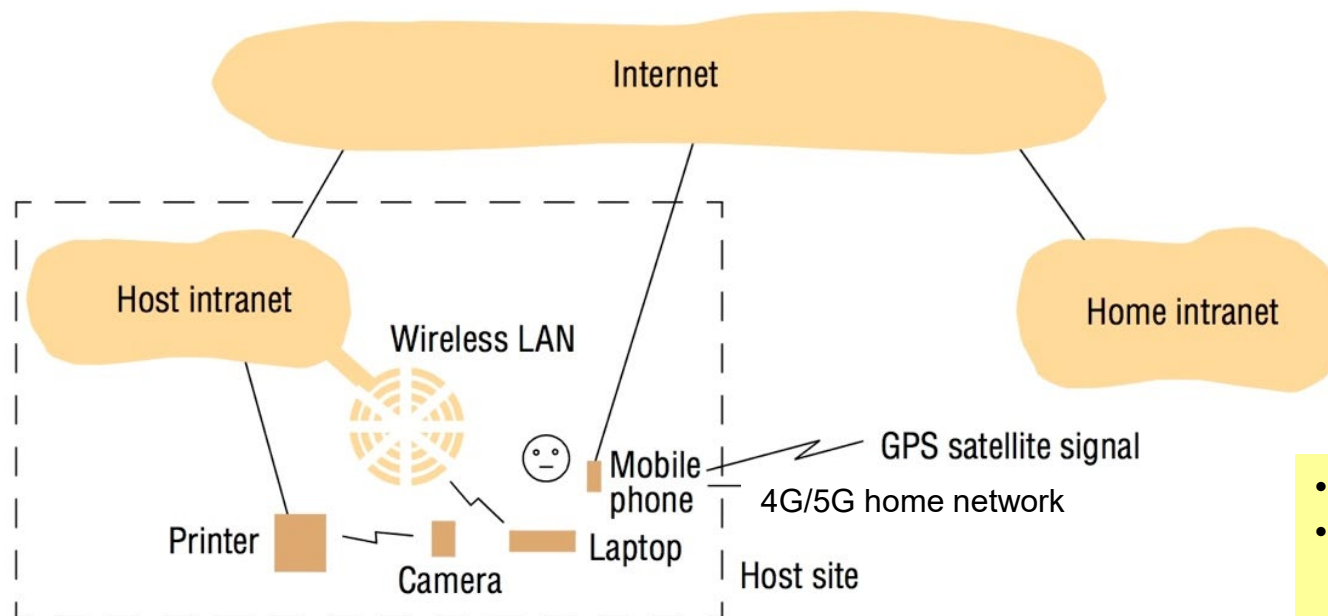

- *Mobile computing*
- *Location-aware* or *context-aware computing*
- *Ubiquitous computing*
- *Spontaneous Interoperation*
- *Service discovery*

同济大学软件学院
School of Software Engineering, Tongji University

Distributed multimedia systems

- live or pre-ordered television broadcasts → B站
- video-on-demand → 爱奇艺
- music libraries → qq音乐
- audio and video conferencing → zoom、腾讯视频、喜马拉雅FM

同济大学软件学院
School of Software Engineering, Tongji University

Distributed computing as a utility

- Cluster computing
- Grid computing
- Utility computing
- Cloud computing
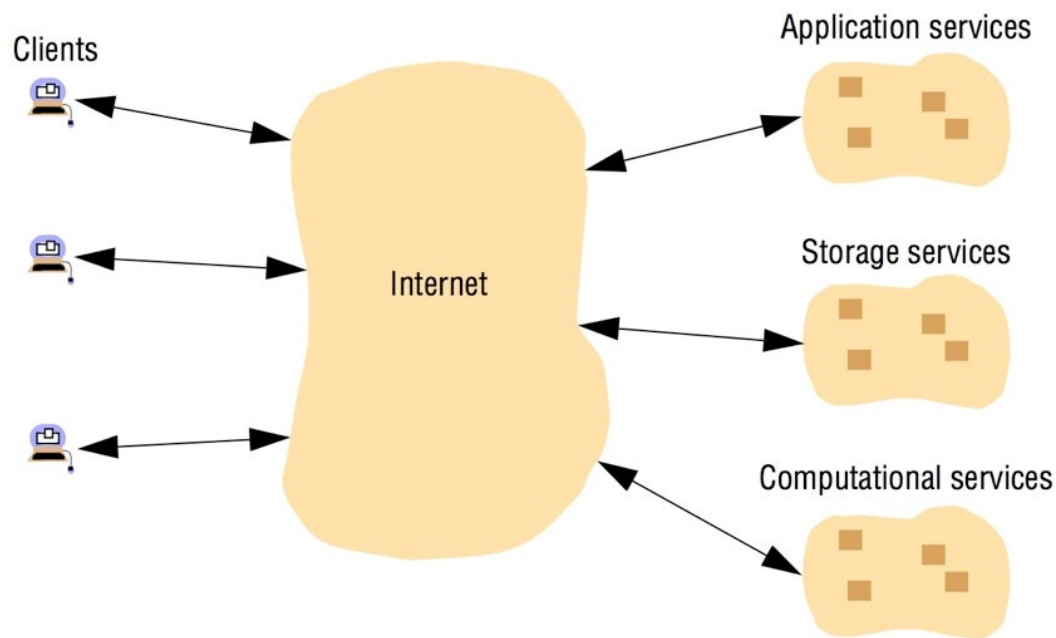
4.Resource Sharing 资源共享

- ## What are the resources?

  - From the point of view of hardware provision: we share equipment such as printers and disks to reduce costs.

  - Of far greater significance to users is the sharing of the higher-level resources:
    - Users are concerned with sharing data e.g., a shared database or a set of web pages, not the *disks* and *processors* on which they are implemented.
    - users think in terms of shared resources such as a search engine or a currency converter, without regard for the *server* or *servers* that provide these.

- ## Service

 - Service is a distinct part of a computer system that manages a collection of related resources and presents their functionality to users and applications.
    - access shared files through a file service:
    - send documents to printers through a printing service:
    - buy goods through an electronic payment service:
  - The only access to the service is via the set of operations that it exports.
    - For example, a file service provides read, write and delete operations on files.

- **Different resources are handled in different ways, there are however some generic requirements**
  - Namespace for identification
  - Name translation to network address
  - Synchronization of multiple access

同济大学软件学院
School of Software Engineering, Tongji University

5. Challenges 分布式系统的设计难点

- Heterogeneity
- Openness
- Security
- Scalability
- Failure handling
- Concurrency
- Transparency
- Quality of service
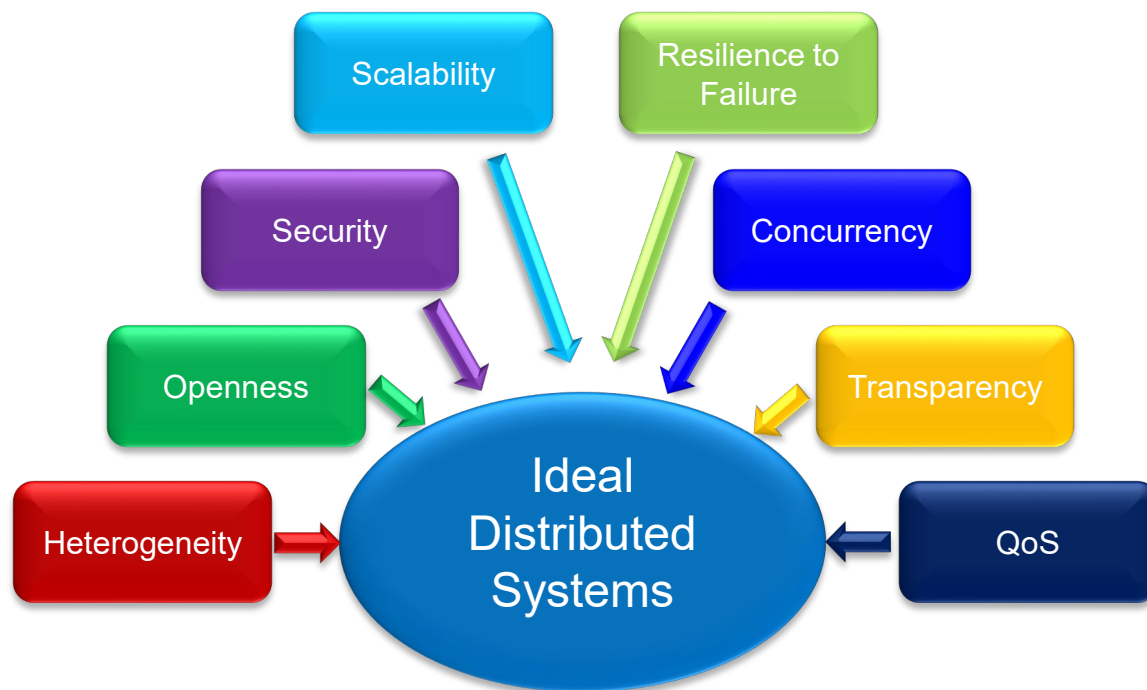
Heterogeneity 异构性

- Heterogeneity (that is, variety and difference) applies to all of the following:

  - networks;
  - computer hardware;
  - operating systems;
  - programming languages;
  - implementations by different developers

Each different sort of network (e.g., Ethernet) will need an implementation of the Internet protocols for that network.	For example, the calls for exchanging messages in UNIX are different from the calls in Windows.

  These differences in representation must be dealt with if messages are to be exchanged between programs running on different hardware.

  > The Internet consists of many different sorts of network, their differences are masked by the fact that all of the computers attached to them use the Internet protocols to communicate with one another.

  > Data types such as integers may be represented in different ways on different sorts of hardware – for example, there are two alternatives for the byte ordering of integers.

- Middleware: software layer providing

  - programming abstraction
  - masking heterogeneity of:
    - underlying networks, hardware, operating systems

  尺寸规格不一致, SRV
  Mem: 1GB VS 64 GB

- Heterogeneity and mobile code

  - Mobile code – programming code that can be transferred from one computer to another and run at the destination (Example: think Java applets, Javascript)
  - Virtual machine approach – way of making code executable on a variety of host computers – the compiler for a particular language (e.g., Java VM) generates code for a virtual machine instead of a particular hardware order code

同济大学软件学院
School of Software Engineering, Tongji University

Openness 开放性

- The openness of a computer system is determined
  - whether the system can be extended and reimplemented in various ways.
- The openness of distributed systems is determined primarily by
  - the degree to which new resource-sharing services can be added and be made available for use by a variety of client programs.

注意分布式系统
开放性的定义

- Open Distributed System
  - Systems that are designed to support resource are extensible.

- Open systems are characterized by the fact that their key interfaces are published.
- Open distributed systems are based on the provision of a uniform communication mechanism and published interfaces for access to shared resources.
- Open distributed systems can be constructed from heterogeneous hardware and software, possibly from different vendors
  - At the hardware level by the addition of computers to the network.
  - At the software level by the introduction of new services and the reimplementation of old ones, enabling application programs to share resources.
 - their independence from individual vendors

同济大学软件学院
School of Software Engineering, Tongji University

Security 安全性

- **Confidentiality** – protection against disclosure to unauthorized individuals

- **Integrity** – protection against alteration or corruption

- **Availability** – protection against interference with the means to access the resources

- **Security challenges not yet fully met:**
  - denial of service attacks
  - security of mobile code

同济大学软件学院
School of Software Engineering, Tongji University

Scalability 可扩展性

- A system is described as *scalable* if it will remain effective when there is a significant increase in the number of resources and the number of users.
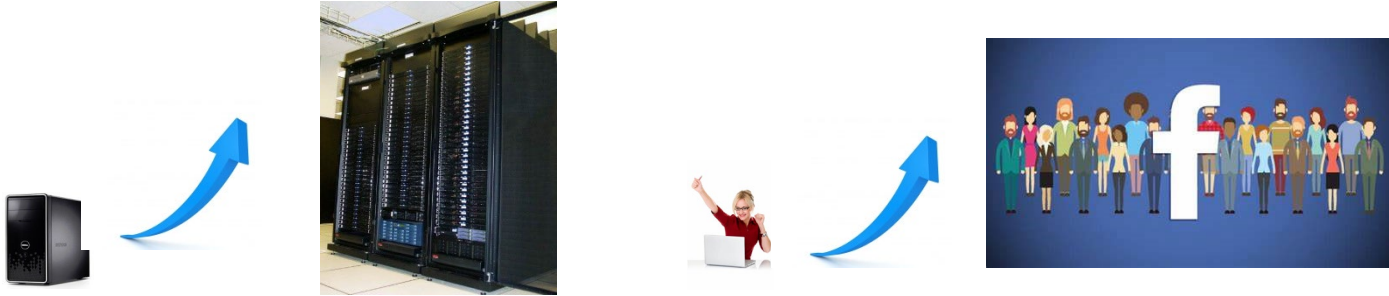
- The design of scalable distributed systems presents the following challenges:
  - *Controlling the cost of physical resources*
  - *Controlling the performance loss*
  - *Preventing software resources running out*
  - *Avoiding performance bottlenecks*
    - Example: some web-pages accessed very frequently – remedy: Caching and Replication

同济大学软件学院
School of Software Engineering, Tongji University

现代分布式系统的数量规模(用户和对象)

- Facebook: 10亿活跃用户

- Google: 每天处理12亿次请求, 访问270亿内容

- YouTube: 每天观看的视频数 >20亿次

- Flickr: >60亿照片

- 截止2016/08/18, 微信和**WeChat**合并月活跃用户数达**8.06**亿，同比增长34％

数据量有多大呢？

- **现代分布式系统使用海量的数据:**
  - 'Avatar'电影渲染数据 >1 PB的存储空间
  - eBay 包括 >6.5PB的用户数据
  - Google早在2008年每天就是产生20PB的数据
  - Google 现在已经着手设计 1 EB的存储系统
  - NSA Utah数据中心据说有5ZB的存储系统 (!)

  National Security Agency
  美国国家安全局

- **1ZB的数据到底有多少？**
  - 1,000,000,000,000,000,000,000 bytes (21个零)
  - 如果用1TB的硬盘累计起来有25,400 km 的高度
    - 上海到纽约的距离是多少？ **14500 km**

25,400 km

计算能力的情况是怎样？

- **一台机器不可能处理那么多的数据**
  - 那么就利用**多个**计算机!

- **现代分布式系统需要多少台计算机？**
  - Facebook：> 60,000 服务器
  - Akamai 在71个国家有95,000 服务器
  - Intel 在97个数据中心有 ~100,000 台服务器
  - Microsoft 在2008年有至少 200,000 服务器
  - Google 据说有 >1百万台服务器, 目前规划1千万台服务器
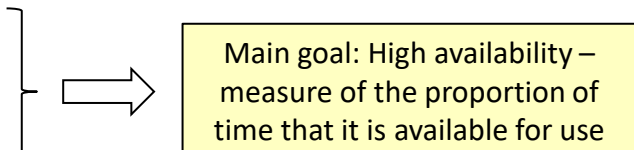
你来构建下一代的*Google*和微信系统

- **你该怎么构建下一代的Google呢？**
  - … 如何下载和存储10亿次Web页面和图片？
  - … 如何快速找到包括输入关键字(例如: 上海 同济大学)的Web页面？
  - … 如何找到一个给定查询最相关的页面？
  - … 如何<u>每天</u>响应12亿次的查询？

- **你该怎么构建下一代的WebChat微信平台呢？**
  - … 如何存储5亿个用户的profile (画像)数据？
  - … 如何确保所有的profile数据都没有丢失呢？
  - … 如何找到你潜在的朋友呢？

  2022、08、30

同济大学软件学院
School of Software Engineering, Tongji University
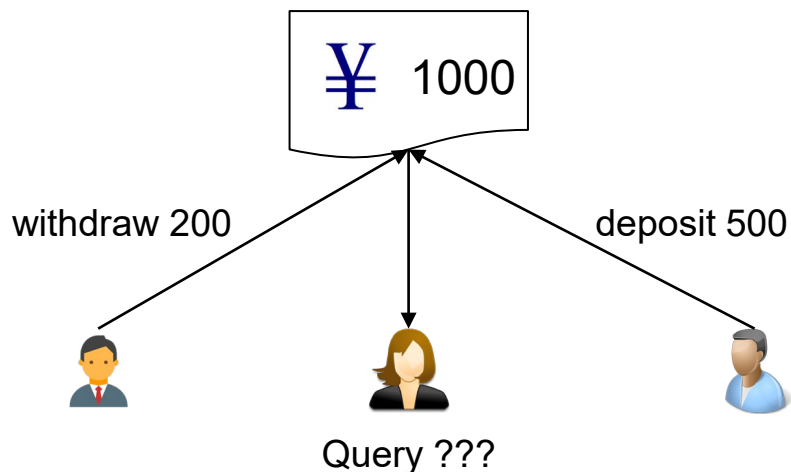
Failure handling 故障处理

- Computer systems sometimes fail. When faults occur in hardware or software, programs may produce incorrect results or may stop before they have completed the intended computation.

- Failures in a distributed system are partial – that is, some components fail while others continue to function. Therefore the handling of failures is particularly difficult.

- Techniques for dealing with failures
  - ◆ Detecting failures
  - ◆ Masking failures
    - □ messages can be retransmitted
    - □ disks can be replicated in a synchronous action
  - ◆ Tolerating failures
  - ◆ Recovery from failures
  - ◆ Redundancy: redundant components
    - □ at least two different routes
    - □ like in DNS every name table replicated in at least two different servers
    - □ database can be replicated in several servers

Main goal: High availability – measure of the proportion of time that it is available for use

同济大学软件学院
School of Software Engineering, Tongji University

Concurrency并发性

- Example: Several clients trying to access shared resource at the same time
- Any object with shared resources in a DS must be responsible that it operates correctly in a concurrent environment

¥ 1000

withdraw 200

deposit 500

Query ???

同济大学软件学院
School of Software Engineering, Tongji University

Transparency透明性

- ■ -- the concealment from the user and the application programmer of the separation of components in a distributed system
  - ◆ so that the system is perceived as a whole rather than as a collection of independent components.
① Access transparency enables local/remote resources to be accessed using identical operations.
② Location transparency enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).
③ Concurrency transparency enables several processes to operate concurrently using shared resources without interference between them.
④ 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.
⑤ Failure transparency enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.
⑥ Mobility transparency allows the movement of resources and clients within a system without affecting the operation of users or programs.
⑦ Performance transparency allows the system to be reconfigured to improve performance as loads vary.
⑧ Scaling transparency allows the system and applications to expand in scale without change to the system structure or the application algorithms.

同济大学软件学院
School of Software Engineering, Tongji University

Quality of service服务质量

- The main nonfunctional properties of systems that affect the quality of the service experienced by clients and users are *reliability*, *security* and *performance*.

- *Adaptability* to meet changing system configurations and resource availability has been recognized as a further important aspect of service quality.

- *time-critical data* (multimedia applications) – streams of data that are required to be processed or transferred from one process to another at a fixed rate.

同济大学软件学院
School of Software Engineering, Tongji University

小结

- **Resource sharing is the main motivating factor for constructing distributed systems.**
    - Resources such as printers, files, web pages or database records are managed by servers of the appropriate type. Resources are accessed by clients.
- **The construction of distributed systems produces many challenges**

同济大学软件学院
School of Software Engineering, Tongji University