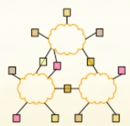
15-446 Distributed Systems Spring 2009



L-1 Introduction

1

Today's Lecture

- Administrivia
- Example topics

2

Instructors

- Instructor
  - Srini Seshan
  - srini@cmu.edu, Wean Hall 8113
  - Office hours: Thursday 1-2pm
- Teaching assistant
  - Dongsu Han
  - dongsuh@cs.cmu.edu, Wean 8218
  - Office hours: Mon 1:30-2:30

3

Course Goals

- Become familiar with the principles and practice of distributed systems
  - algorithms, APIs, …
- Learn how to write distributed applications that use the network
  - How does a distributed file system work?
- Mobile/Ubiquitous computing focus
  - Projects and programming done using Android platform

4

Android

- Dramatically updated class focus
  - Distributed systems + mobile/ubiquitous
- New projects
  - Project 1
    - Part A → discovery protocol for mobile phones
    - Part B → distributed file updates with eventual consistency
  - Project 2
    - Self-defined
    - 3-person groups
    - We'll provide some ideas, but you can make up your own

5

Course Format

- ~30 lectures
  - Cover the "principles and practice"
  - Complete readings before lecture
- 4 paper/lab homework assignments
  - Loosely tied to lecture materials
  - Teach networking concepts/tools
- 2.5 programming projects
  - How to use and build networks / networked applications
  - Application-layer programming
  - Larger, open-ended projects. *Start early!*
- Midterm and final
  - Covers each of the above 3 parts of class

6

Lecture Schedule

- Syllabus online
- Lectures Tue/Thu 10:30-11:50
- Recitations Wed 3:30-4:20
  - Used for:
    - Project/programming related lectures
    - Exam reviews
    - Makeup lectures
  - Not taught every week – keep an eye on syllabus

7

Reading

- Recommended textbooks
  - Distributed Systems: Concepts and Design, 4rd ed by Coulouris, G, Dollimore, J., and Kindberg, T., Addison-Wesley, 2006. ISBN: 0321263545
  - Distributed Systems: Principles and Paradigms, 2nd ed by Tanenbaum, A. and van Steen, M., Prentice Hall, 2007. ISBN: 0132392275
  - Either is fine, will post readings on Web page for both
    - Both will be placed on reserve in library
- Paper readings
  - Classic academic papers on different topics

8

Sounds Great! How Do I Get In?

- Currently 17 people are enrolled, and 13 people are on the waiting list.
  - Limited to about 20 ☹
  - If you do not plan to take the course, please drop it ASAP so somebody else can take your place!
- Preference will not be based on online waitlist order
  - We give preference to students attending class.
    - Sign in sheet
  - Preference given to undergraduates

9

Administrative Stuff

- Watch the course web page
  - http://www.cs.cmu.edu/~srini/15-446/S09/
  - Handouts, readings, ..
- Read bboards
  - **cyrus.academic.cs.15-441.announce** for official announcements
  - **cyrus.academic.cs.15-441** for questions/ answers
- Course secretary
  - Barbara Grandillo, Wean Hall 8018

10

Grading

- Roughly equal weight in projects and testing
  - 20% for Project I
  - 20% for Project II
  - 15% for Midterm exam
  - 25% for Final exam
  - 20% for Homework
- You **MUST** demonstrate competence in both projects and tests to pass the course
  - Fail either and you fail the class!

11

Policy on Collaboration

- Working together is important
  - Discuss course material in general terms
  - Work together on program debugging, ..
- Final submission must be your own work
  - Homeworks, midterm, final
- Project 1
  - Done independently
- Project 2
  - Teams of three
  - Collaboration, group project skills
  - All students should understand the entire project

12

Late Work and Regrading

- Late work will receive a 15% penalty/day
  - No assignment can be more than 2 days late
  - Only exception is documented illness and family emergencies
- Requests for regrading must be submitted in writing to course secretary within 2 weeks.
  - Regrading will be done by original grader
- No assignments with a "short fuse"
  - Homeworks: ~1-2 weeks
  - Projects: ~5 weeks
  - Start on time!
  - Every year some students discover that a 5 week project cannot be completed in a week

13

Today's Lecture

- Administrivia
- Examples topics

14

Topics

Traditional

- Networking
- Naming
- Time synchronization
- Replication/ Consistency
- Security
- Transactions
- Distributed file systems
- Group communication
- Cluster computing

Ubicomp

- Wireless networks
- Localization
- P2P
- Sensor networks
- Adaptive applications
- Social networks
- P2P applications
- Android APIs

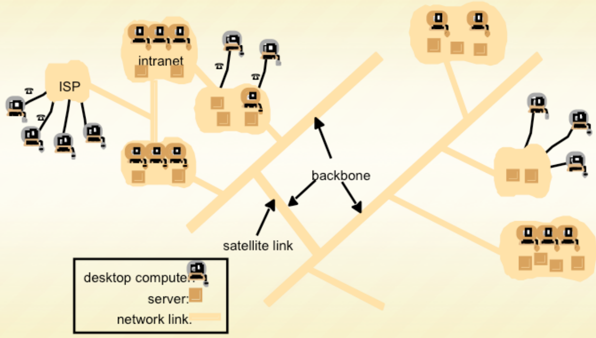
15

Definition of a Distributed System

- A distributed system is:
- A collection of independent computers that appears to its users as a single coherent system
- "A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable." – Leslie Lamport

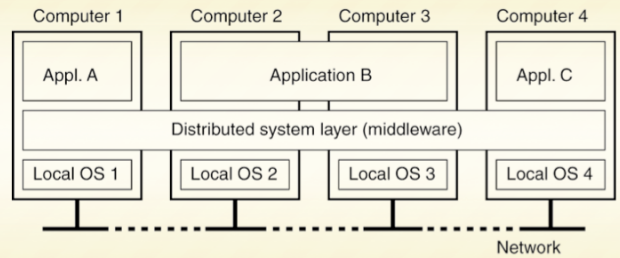
16

Figure 1.1
A typical portion of the Internet



17

Middleware



- The middleware layer extends over multiple machines, and offers each application the same interface.

18

Transparency in a Distributed System

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource is replicated
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

19

Pitfalls when Developing Distributed Systems

False assumptions made by first time developer:

- The network is reliable.
- The network is secure.
- The network is homogeneous.
- The topology does not change.
- Latency is zero.
- Bandwidth is infinite.
- Transport cost is zero.
- There is one administrator.

20

Scalability Problems

Characteristics of decentralized algorithms:

- No machine has complete information about the system state.
- Machines make decisions based only on local information.
- Failure of one machine does not ruin the algorithm.
- There is no implicit assumption that a global clock exists.

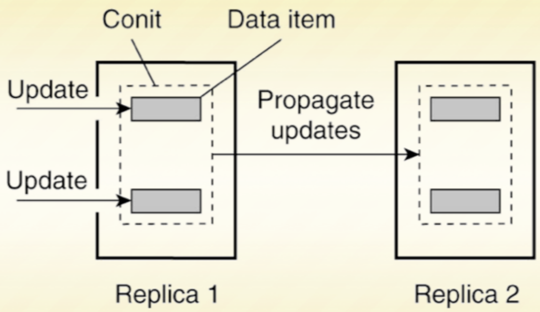
21

Reasons for Replication

- Data are replicated to increase the reliability of a system.
- Replication for performance
  - Scaling in numbers
  - Scaling in geographical area
- Caveat
  - Gain in performance
  - Cost of increased bandwidth for maintaining replication

22

Continuous Consistency



23

Transaction Processing Systems

Primitive	Description
BEGIN_TRANSACTION	Mark the start of a transaction
END_TRANSACTION	Terminate the transaction and try to commit
ABORT_TRANSACTION	Kill the transaction and restore the old values
READ	Read data from a file, a table, or otherwise
WRITE	Write data to a file, a table, or otherwise

24

Transaction Processing Systems

ACID properties of transactions:

- **Atomic**: To the outside world, the transaction happens indivisibly.
- **Consistent**: The transaction does not violate system invariants.
- **Isolated**: Concurrent transactions do not interfere with each other.
- **Durable**: Once a transaction commits, the changes are permanent.

25

Failure Models

Type of failure	Description
Crash failure	A server halts, but is working correctly until it halts
Omission failure *Receive omission* *Send omission*	A server fails to respond to incoming requests A server fails to receive incoming messages A server fails to send messages
Timing failure	A server's response lies outside the specified time interval
Response failure *Value failure* *State transition failure*	A server's response is incorrect The value of the response is wrong The server deviates from the correct flow of control
Arbitrary failure	A server may produce arbitrary responses at arbitrary times

26

Next Lecture

- Networking intro
- Read "End-to-End Arguments" paper

27