Slide 1

DISTRIBUTED SYSTEMS (COMP9243)

Lecture 12: Review

① Review
② What to Study
③ Example Exam Questions

Slide 2

LECTURE 1 - INTRODUCTION

Distributed System:

A distributed system is a collection of independent computers that are used jointly to perform a single task or to provide a single service.

- ➜ Examples of distributed systems
- ➜ Advantages and disadvantages of distributed systems
- ➜ Hardware architecture: Multicomputer
- ➜ Software architecture: Distributed OS and Middleware
- ➜ Principles: key principles/topics underlying all distributed systems
- ➜ Paradigms: model of how communication takes place and how data and control are distributed.

Slide 3

Problems and Challenges in Distributed Systems:

Transparency : Concealment of the separation of components

Scalability : Can handle the addition of users and resources without suffering a noticeable loss of performance

- Hide communication latencies
- Distribution/partitioning
- Replication

Dependability : Availability (requires consistency, security, fault tolerance)

Performance : Strive for maximum performance

- Conflicts with other challenges

Flexibility : Extensibility, openness, interoperability

Slide 4

LECTURE 2 - SYSTEM ARCHITECTURE & COMMUNICATION

Architecture of Distributed Systems:

- ➜ Client-server (multi-tier)
  - vertical, horizontal
- ➜ Peer-to-Peer, Hybrid
- ➜ Server design
- ➜ Code mobility

Approaches to Communication in Distributed Systems:

- ➜ Shared Memory
- ➜ Message Passing

Communication Modes:

- ➜ Data oriented vs control oriented
- ➜ Synchronous vs asynchronous
- ➜ transient vs persistent

Slide 5

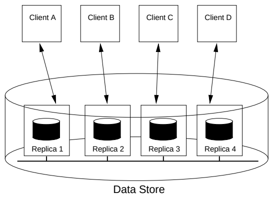
Communication Abstractions:

- ➜ Remote Procedure Call (RPC)
  - Sun RPC, SOAP, XML-RPC
  - Danger of RPC/RMI transparency
- ➜ Remote Method Invocation (RMI)
- ➜ Message Oriented
- ➜ Event-based
- ➜ Group communication
- ➜ Streams

Slide 6

Lecture 3a - Replication and Consistency

Distributed Data Store:



Consistency Models:

- ➜ Data-centric vs client-centric consistency models
- ➜ Strong ordering vs weak ordering
- ➜ Sequential consistency
- ➜ Causal consistency
- ➜ FIFO (PRAM) consistency

Slide 7

Weak Consistency:

Consistency of 'critical sections'

- ➜ Release consistency
- ➜ Lazy release consistency
- ➜ Entry consistency

Client-Centric Consistency:

Ordering of operations for a single client

- ➜ Monotonic reads
- ➜ Monotonic writes
- ➜ Read your writes
- ➜ Write follows read

Slide 8

Consistency Protocols:

Implementing Replication and Consistency Models

- ➜ Remote/local Write
  - Single server
  - Primary backup
  - Migration
  - Migrating primary
- ➜ Replicated write
  - Active replication
  - Quorum based

Distribution Policy:

- ➜ Update propagation (Push vs Pull and Leases)
- ➜ Replica placement (Static vs Dynamic)

LECTURE 5 & 6 - SYNCHRONISATION & COORDINATION

Slide 9

Time and Clocks:

- ➜ Global time
- ➜ Synchronising physical clocks
- ➜ Logical clocks
- ➜ Vector clocks
- ➜ Causal message ordering

Global State:

- ➜ Global properties
- ➜ Consistent cut
- ➜ Chandy and Lamport snapshots

Mutual Exclusion:

- ➜ Central server approach
- ➜ Token ring approach
- ➜ Ricart & Agrawala: multicast and logical clocks

Slide 10

Multicast:

- ➜ Ordering guarantees
- ➜ FIFO, Causal, Totally ordered

Elections:

- ➜ Bully algorithm
- ➜ Ring algorithm

Transactions:

- ➜ ACID properties
- ➜ Private workspace vs writeahead log
- ➜ Conflicts and serialisability
- ➜ Concurrency Control: pessimistic vs optimistic

Distributed Transactions:

- ➜ Distributed locking
- ➜ Distributed commit: 2PC
- ➜ 2PC with timeouts

LECTURE 7 - FAULT TOLERANCE

Slide 11

A fault tolerant system can provide its services even in the presence of faults

Dependability, Failure and Faults:

- ➜ Faults cause errors
- ➜ Errors cause failures

Types of Failure:

- ➜ Total failure
- ➜ Partial failure
- ➜ Failure models
  - Crash failure
  - Omission failure
  - Timing failure
  - Response failure
  - Arbitrary (byzantine) failure

Slide 12

Failure Masking:

Try to hide occurrence of failure from other processes

- ➜ Redundancy
- ➜ Process resilience: replicate processes
  - k fault tolerance
  - Agreement: 2-army problem, byzantine generals

Reliable Communication:

- ➜ Mask crash and omission failures
- ➜ TCP/IP
- ➜ Reliable multicast

Slide 13

Failure Recovery:

Restoring an erroneous state to an error free state

- ➜ Forward error recovery
- ➜ Backward error recovery
  - Log-based (undo operations)
  - State-based (checkpointing and rolling back)
  - Checkpointing and domino effect
  - Consistent checkpointing
  - Synchronous vs Asynchronous checkpointing

Slide 14

Lecture 8 - Security

Availability, Confidentiality and Integrity

Threats:

- ➜ Interception
- ➜ Interruption
- ➜ Modification
- ➜ Fabrication

Security Mechanisms:

- ➜ Encryption (cryptography)
- ➜ Authentication
- ➜ Authorisation
- ➜ Auditing

Slide 15

Cryptography:

- ➜ Symmetric ciphers (DES), Asymmetric ciphers (RSA)
- ➜ Block ciphers, Stream ciphers
- ➜ Digests, Digital signatures

Cryptographic Protocols:

- ➜ Key Distribution (Needham Schroeder)
- ➜ Public Key Distribution (Certificates)

Secure Communication:

- ➜ Secure Socket Layer (SSL/TLS)

Slide 16

Authentication:

- ➜ *Verify the claimed identity of an entity*
- ➜ Shared secret key
- ➜ Key distribution centre (Kerberos)
- ➜ Public Key
- ➜ Hybrid (SSH)

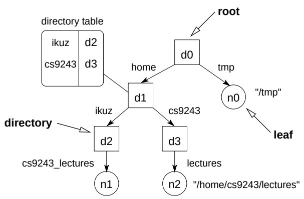
Authorisation (Protection):

- ➜ Access control matrix
- ➜ Access control list (ACL)
- ➜ Capabilities

Slide 17

Lecture 9 - Naming

Names and Name Spaces:



Name Resolution:

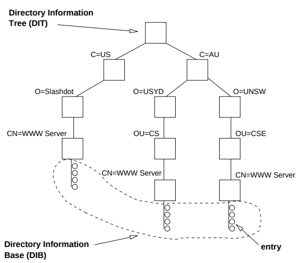
- ➜ Iterative
- ➜ Recursive

Naming Services:

- ➜ Client & Resolver
- ➜ Server
- ➜ Replication & Partitioning
- ➜ Caching
- ➜ DNS

Slide 18

Attribute-Based Naming:

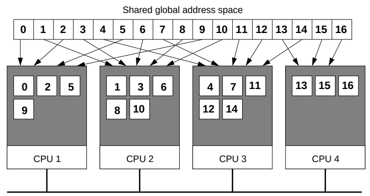


Directory Services:

- ➜ Server
  - Partitioned
  - Replicated
- ➜ Search performance
- ➜ X.500 and LDAP

Slide 19

Lecture 3b - Distributed Shared Memory



DSM Implementations:

- ➜ Hardware
- ➜ Operating System
- ➜ Middleware

Slide 20

DSM Models:

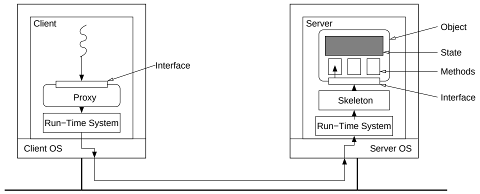
- ➜ Shared page
- ➜ Shared region
- ➜ Shared variable
- ➜ Shared structure

Case Studies:

- ➜ Design and implementation details
- ➜ TreadMarks – user-level page-based DSM

Lecture 4 - Middleware: Distributed Objects & PubSub

Slide 21

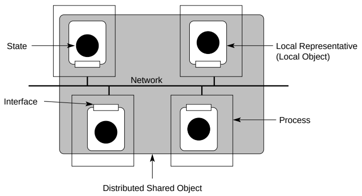


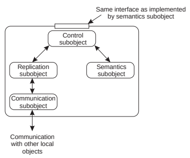
Remote Objects:

- Interfaces, Remote Method Invocation
- Invocation semantics
- CORBA

Slide 22

Distributed Shared Objects:



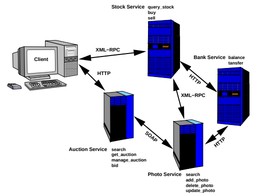


Publish/Subscribe Middleware:

- Loose coupling
- Matching subscribers and publishers
- TIB/Rendezvous, JMS, Scribe

Slide 23

SOA: Service Oriented Architectures:



- SOAP
- REST
- Web services & Web 2.0
- Mashups

Slide 24

Lecture 10 - Distributed File Systems

File Access Semantics:

- Unix semantics (read returns latest write)
- Session semantics
- Immutable files

Implementation:

- Stateless vs Stateful servers
- Caching and Replication
- NFS
- AFS & Coda
- Google file system

Slide 25

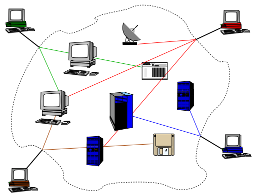
Lecture 10b - Parallel Computing, Grid, & Cloud

Parallel Computing:

- ➜ Performance and Overhead
  - Speedup
  - What factors affect speedup
  - Amdahl's law
- ➜ Architectures and parallel programming models
- ➜ Suitable problems
- ➜ Designing parallel programs

Slide 26

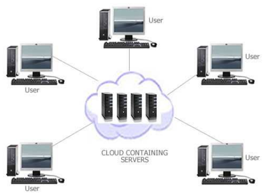
Grid Computing:



- ➜ Grid architecture
- ➜ Globus
  - Security
  - Metadata
  - Resource Access

Slide 27

Cloud Computing:



- ➜ Platform as a Service
- ➜ Infrastructure as a Service
- ➜ Mega Giga PetaFLOPS of information!

Slide 28

Lecture 11 - Distributed Systems at Google

MapReduce:

- ➜ `map`
- ➜ shuffle
- ➜ `reduce`
- ➜ Specific for particular class of parallel applications

BigTable:

- ➜ one BIG table
- ➜ rows and columns
- ➜ tablets for distribution

Chubby:

- ➜ simple file store
- ➜ coarse-grained locking
- ➜ also used as name sever

Assignment 1 - Distributed Shared Memory

Slide 29

- ➜ Implementing a DSM
- ➜ Implementing communication in a distributed system
- ➜ Implementing synchronisation primitives (e.g., barrier)
- ➜ C programming

Assignment 2 - Router Network

Slide 30

- ➜ Implementing 2PC
- ➜ Implementing a snapshot protocol
- ➜ Erlang programming

Survey

Slide 31

CATEI survey:

- ➜ Run by UNSW
- ➜ Generic
- ➜ Important for CSE and rankings
- ➜ At myUNSW until end of June

PHPSurvey:

- ➜ Run by CSE
- ➜ Anonymous
- ➜ Course specific
- ➜ Used to improve cs9243 course based on your feedback
- ➜ 2 bonus marks for filling it in
- ➜ at `http://mahler.cse.unsw.edu.au/phpSurvey2` until June 19

The Exam

Slide 32

- ➜ Friday June 12, 2009
- ➜ 8:45-12:00 (10 minutes reading time)
- ➜ OMB 151 (Old Main Building - K15)
- ➜ open book
- ➜ timetable: `http://www.cse.unsw.edu.au/export/sites/cse/people/downloads/CSE_INTERNAL_EXAM_TIMETABLE.pdf`

Slide 33

What to Study

Material on Lecture Notes Web Page:

- All slides
- All notes (note the notes linked to from the *Lecture Notes* Web page cover the same material as the slides but may provide more detail)
- Papers linked from *Lecture Notes* Web page (except optional ones)

Assignments:

- Assignment 1
- Assignment 2

Slide 34

Example Exam Questions

Question 1:

In an interview about designing distributed systems Ken Arnold (the original lead architect of JavaSpaces) said *"State is hell. You need to design systems under the assumption that state is hell. Everything that can be stateless should be stateless"*. Why does state pose a problem in distributed systems?

moral: be able to understand the issues of distributed systems, even in other peoples words. Be able to determine which issues the people are referring to.

Slide 35

Answer to Question 1:

Slide 36

Question 2:

SOAP and XML-RPC are two examples of RPC protocols that use XML as their data format and HTTP as their transport. Discuss the benefits and drawbacks of using XML as a data format. Also discuss the benefits and drawbacks of using HTTP as a transport.

moral: Be able to discuss the benefits and drawbacks of aproaches taken in technologies that we may not have discussed in class.

Answer to Question 2:

Slide 37

Question 3:

Servers hosting popular Web sites often receive more requests than they can handle, causing their performance to suffer. A typical way of overcoming this problem is to replicate the contents on other servers. Describe some of the problems that this may introduce. Discuss possible solutions to the problems you mention.

Slide 38

moral: A topic we didn't explicitly discuss in class, though it was often used as an example. Be able to apply the concepts (and knowledge of problems and their solutions) to new scenarios

Answer to Question 3:

Slide 39

Question 4: Read your writes is a client-centric consistency model.

① Describe this model.

② A naive implementation of this model requires that each replica server maintain a write set of the write operations that it has seen. Likewise, each client maintains a write set of the write operations that it has performed. For this implementation we assume that each write operation is identified by a unique identifier (note that because we are dealing with replication a write operation will be executed at each replica server. Each replicated execution of the same write operation will have the same identifier). This identifier is generated by the server that accepts the operation for the first time (i.e., the server where the write is initiated). Complete (and describe) the design of this naive implementation of read your writes. Provide an example of how your design works.

③ A problem with the naive implementation introduced above is that the write sets may become very large, leading to poor performance. A more efficient solution is possible using vector timestamps. Sketch the design of a read your writes implementation that avoids the overhead of large write sets by using vector timestamps. Provide an example of how your design works.

Slide 40

Slide 41 moral: when studying the models, think of how they might be implemented. Also be able to combine knowledge presented in different lectures.

Slide 42

Answer to Question 4:

Part 1:

Slide 43

Part 2:

Slide 44

Part 3: