Distributed Systems

Topics

Final Exam

- When and where?
  - 11-May-2019 (Tuesday), 13:30-15:30 (G707)
- What to bring?
  - Close Book exam!
- What are the questions?
  - Total marks are 100 (2 hours)
  - Answer all 8 questions
- How to prepare?
  - Understanding lecture notes and tutorial questions.

Introduction

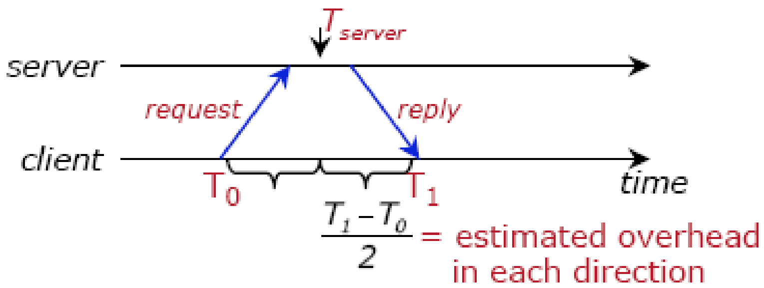
- Distributed system is everywhere.
- The motivation of constructing a distributed system is resource sharing and collaborative computing
- Distributed system features.
  - Concurrency
  - No global clock
  - Independent failure
- Distributed system challenges.
  - Heterogeneity
  - Openness
  - Security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

System Model

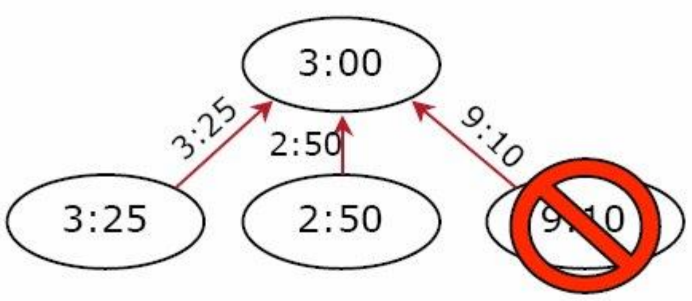
- Difficulties for and threats to distributed systems
- **Physical Model**
  - Three generations of distributed systems and the emergence of ultra-large-scale (ULS) distributed systems.
- **Architectural Model**
  - Entities that are communicating in the distributed system
  - Types of communication paradigms:
    - Interprocess communication
    - Remote invocation
    - Indirect communication
  - Architectural styles: client-server and peer-to-peer
  - Vertical distribution (Multi-Tier) and horizontal distribution of c/s systems
- **Fundamental Model**
  - Characteristics of synchronous distributed systems and asynchronous distributed system(Interaction model)

Physical Clock Synchronization

- Cristian's algorithm:
  - synchronize clocks with a UTC server

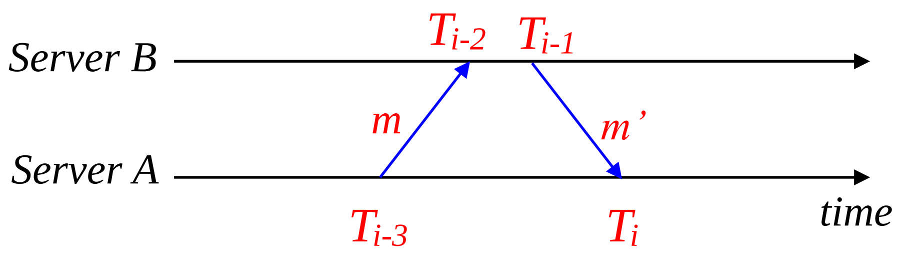


- Berkeley Algorithm:
  - synchronize a set of clocks as close as possible



NTP Symmetric mode

- There can be a non-negligible delay between the arrival of one message and the dispatch of the next



- Delay = total transmission time of the two messages

$$d_i = (T_i - T_{i-3}) - (T_{i-1} - T_{i-2})$$

- Offset of clock *A* relative to clock *B*:
  - Offset of clock *A*: $o_i = ((T_{i-2} - T_{i-3}) + (T_{i-1} - T_i))/2$
  - Accuracy bound: $d_i /2$

Logical Clock Synchronization

- Event ordering: happened before ($\rightarrow$), concurrent ($\|$)
- Lamport's algorithm: single timestamp clock synchronization & features
  - If $a \rightarrow b$ then $V(a) < V(b)$
  - The reverse is not always true
- Timestamp vector synchronization and features
  - $a \rightarrow b$ **iff** $V(a) < V(b)$
  - $a \| b$ **iff** neither $V(a) \leq V(b)$ nor $V(b) \leq V(a)$
- Application of timestamp vectors: causal ordered multicast

Interprocess Communication (IPC)

Socket operations and Socket types

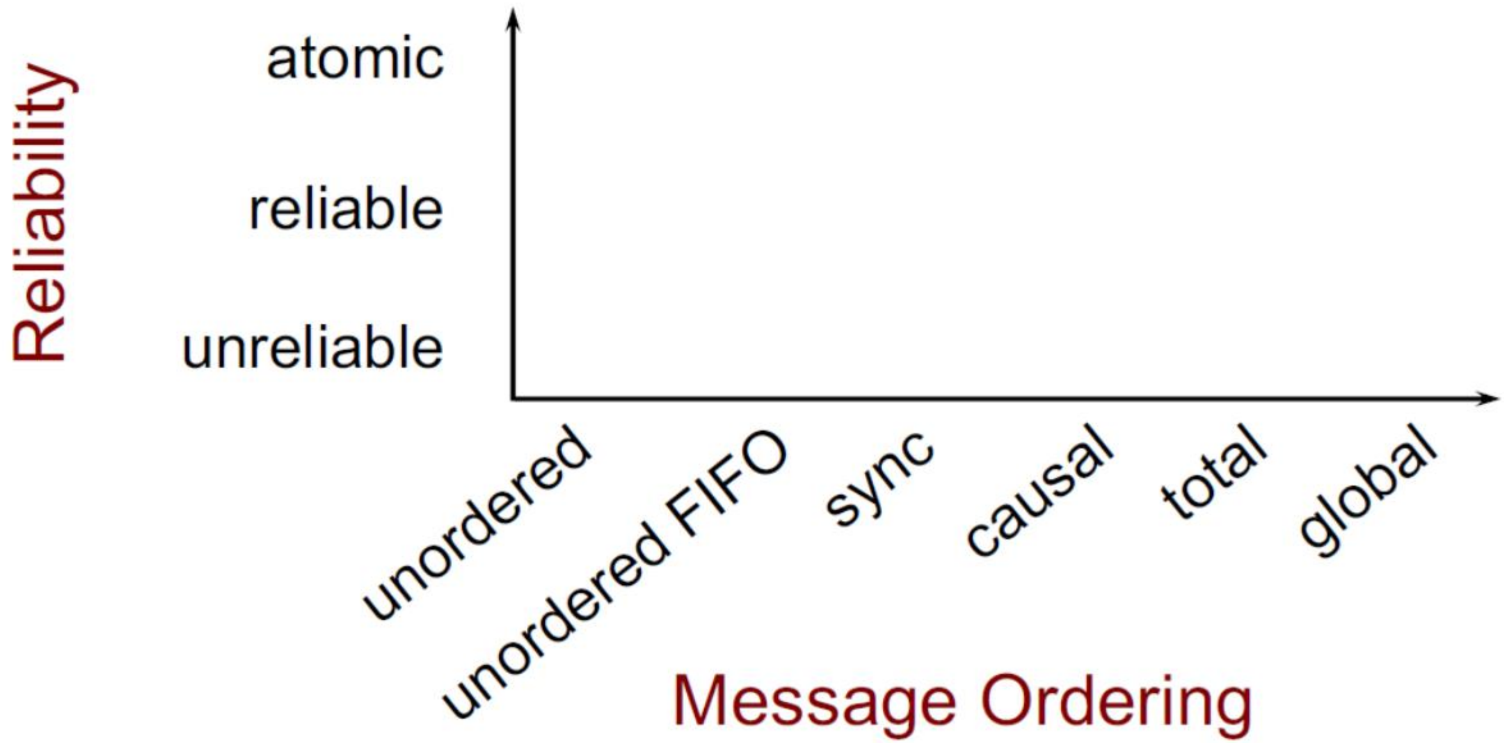
a) Socket/bind
b) listen, accept, connect
c) Read/write functionalities
d) close/shutdown

Programming by using sockets

Remote Invocation (RPC)

- Interface Definition of a client-server system
- Client program development
- Server program development
- Client-server interactions in RPC

Group Communication



Mutual Exclusion & Election Algorithm

Mutual Exclusion

- Centralized Algorithm
- Token Ring Algorithm
- Ricart & Agrawala algorithm
- Distributed Mutual Exclusion

Election

- Bully Algorithm
- Ring election Algorithm
- Robert & Chang Algorithm

Data Centric Consistency Models

Consistency	Description
Strict	Absolute time ordering of all shared accesses matters.
Linearizability	All processes must see all shared accesses in the same order. Accesses are furthermore ordered according to a (nonunique) global timestamp
Sequential	All processes see all shared accesses in the same order. Accesses are not ordered in time
Causal	All processes see causally-related shared accesses in the same order.
FIFO	All processes see writes from each other in the order they were used. Writes from different processes may not always be seen in that order

(a)

Consistency	Description
Weak	Shared data can be counted on to be consistent only after a synchronization is done
Release	Shared data are made consistent when a critical region is exited
Entry	Shared data pertaining to a critical region are made consistent when a critical region is entered.

(b)

a) Consistency models not using synchronization operations.

b) Models with synchronization operations.

Client Centric Consistency

1. 4 rules of client centric consistency.
2. What is to be propagated?
3. How is it to be propagated?
4. Consistency Protocols
   – Primary-based protocols
   – Replicated write protocols
   – Cache-coherence protocols

Internetworking

- CIDR - Classless InterDomain Routing
- NAT - Network Address Translation
- DHCP - Dynamic Host Configuration Protocol
- ARP - Address Resolution Problem
- OSPF - Interior Gateway Routing Protocol
- BGP - Exterior Gateway Routing Protocol

Name Services

- DNS Name Servers
- Recursive query & Iterative query
- DNS Records
- IP-address to Name: Reverse Mapping

Unix File System

- Internal File Structure : file attributes and data index
- Unix Directory File Directory structure and name resolution
- Superblock
- Internal Structure for File Accesses
- Mount and Unmount

Distributed File System

- Distributed File System Components
- Differences between DFS and stand-alone FS
- Google FS and NFS

Transaction Processing System

- Definitions of Atomic Transactions
- RM-ODP properties
- How to achieve atomicity of transactions?
  - **Failure Recovery** (guarantee nothing-or-all)
    - Intention list approach
    - Shadow version approach
  - **Concurrency Control** (guarantee serializability)
    - 2-Phase Locking
    - Timestamp Ordering
    - Optimistic Method

Web Searching

- Search Engines Architecture
  - Spider
  - Indexer
  - Database
  - Search Engine
- Web Data Mining: Digraph
- FAN - WebPage Ranking
- Inverted File: indexing for search

GOOD LUCK