Coordination in distributed systems

- Coordination is needed by distributed systems but hard to achieve:
  - Events happen concurrently
  - Communication links are not reliable
  - Computers can crash
  - New nodes can join the systems
  - Asynchronous environments

⇒ expect an efficient way to coordinate a group of processes

A Distributed System in WAR: Synchronous Example

A Distributed System in WAR: Reality

Group communication

- What is a group?
  - A number of processes which cooperate to provide a service.
  - An abstract identity to name a collection of processes.

- Group Communication
  - For coordination among processes of a group.
Who Needs Group Communication?

- Highly available servers (client-server)
- Database Replication
- Multimedia Conferencing
- Online Game
- Cluster management
- ...

Distributed Web Server

- High availability

Online Game

- Fault-tolerance, Order

Different Comm. Methods

- Unicast
  - Point-to-Point Communication
  - Multiple copies are sent.
- Broadcast
  - One-to-All Communication
  - Abuse of Network Bandwidth
- Multicast
  - One-to-multiple Communication

Today’s schedule

- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service

Group Comm. Properties

- Name Abstraction
- Efficiency
- Delivery Guarantee
  - Ordering
  - Failure behavior
  - Reliability
  - ...
- Dynamic Membership
  - Group membership service
Properties of Communication

- Ordering
  - Total ordering, causal ordering
- Failure behavior
- Reliability
  - Validity, integrity, agreement

Properties of Group

- Name of group
- Addresses of group members
- Dynamic group membership
- Options:
  - Peer group or client-server group
  - Closed or Open Group

Peer Group

- All the members are equal.
- All the members send messages to the group.
- All the members receive all the messages.

Client-Server Group

- Replicated servers.
- Clients do not care which server answers.

Desired Group Communication

- Name Abstraction
- Efficiency ⇒ Multicast
- Delivery Guarantees ⇒ Reliability, Ordering
- Dynamic Membership ⇒ Group membership service

Today’s schedule

- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service
Multicast communication
- Use network hardware support for broadcast or multicast when it is available.
- Send message over a distribution tree.
- Minimize the time and bandwidth utilization

Reliability
Correct processes: those that never fail.
- Integrity
  A correct process delivers a message at most once.
- Validity
  A message from a correct process will be delivered by the process eventually.
- Agreement
  A message delivered by a correct process will be delivered by all other correct processes in the group.
⇒ Validity + Agreement = Liveness

Ordering
Assumptions: a process belongs to at most one group.
- FIFO
  - if \( m_0 \rightarrow m_1 \), all correct processes that deliver \( m_1 \) will deliver \( m_0 \) before \( m_1 \).
- Causal
  - if \( m_0 \rightarrow m_1 \), all correct processes that deliver \( m_1 \) will deliver \( m_0 \) before \( m_1 \).
- Total
  - if a correct process delivers \( m_0 \) before \( m_1 \), all other correct processes that deliver \( m_1 \) will deliver \( m_0 \) before \( m_1 \).

Examples
- Assumption:
  – Reliable one-to-one send operation (e.g. TCP)
- Basic multicast
  – Requirement:
    - All correct processes will eventually deliver the message from a correct sender.
  – Implementation:
    - \( \text{Received} := \emptyset \)
    - \( \text{R-multicast}(g, m) \) at process \( p \): \( \text{B-multicast}(g, m) \);
    - On \text{B-deliver}(m) at process \( q \)
      \( \text{if} (m \in \text{Received}) \)
      \( \text{Received} := \text{Received} \cup \{m\}; \)
      \( \text{if} (q \neq p) \text{B-multicast}(g, m); \)
    - \( \text{R-deliver}(m) \);
⇒ Properties: integrity, validity

Basic multicast: Agreement?
- Reliable multicast
  – Requirements: integrity, validity, agreement
  – Implementation:
    - \( \text{Received} := \emptyset \)
    - \( \text{R-multicast}(g, m) \) at process \( p \): \( \text{B-multicast}(g, m) \);
    - On \text{B-deliver}(m) at process \( q \)
      \( \text{if} (m \in \text{Received}) \)
    - \( \text{Received} := \text{Received} \cup \{m\}; \)
    - \( \text{if} (q \neq p) \text{B-multicast}(g, m); \)
    - \( \text{R-deliver}(m) \);
⇒ Inefficient: each message is sent \( |g| \) times to each process
  – Encourage to implement in more efficient ways (e.g. IP-multicast)
Examples (cont.)

- FIFO-ordered multicast:
  - Assumption:
    - A process belongs to at most one group.
  - Implementation:
    - Local variables at \( p \): \( S_0, R_0 \)
      - \( S_0 \) multicast at \( p \):
      - \( S_0 = 1 \)
      - \( S_0 \) multicast at \( q \):
      - \( S_0 = 0 \)
    - On \( B-deliver \) at \( p \):
      - \( S = R_0 + 1 \)
      - \( S_0 = 0 \)
      - \( S_0 = S \)
      - \( S_0 \) in the queue until \( S = R_0 + 1 \)
      - \( S_0 \) deliver
    - \( R_0 \) deliver

- Encourage to implement causally ordered, totally ordered multicasts.

Today's schedule

- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service

Group membership service

- Four tasks:
  - Interface for group membership changes
  - Failure detector
  - Membership change notification
  - Group address expansion

- Group partition:
  - Primary-partition
  - Partitionable

Group views

- Group views:
  - Lists of the current ordered group members
  - A new one is generated when processes join or leave fail.
- View delivery
  - When the membership changes & a member is notified of it.
- Requirements
  - Order
    - \( p \) delivers \( v \) if \( v \) is sent before \( p \).
  - Integrity
    - \( p \) delivers \( v \) if \( p \) is in the view.
  - Non-triviality
    - If \( q \) joins a group and becomes indefinitely reachable from \( p \), eventually \( q \) is always in the view \( p \) delivers.

View-synchronous group comm.

- Extend the reliable multicast semantics with group views.
  - Agreement
    - Correct processes deliver the same set of messages in any given view.
  - Validity (closed group)
    - Correct processes always deliver the messages they send.
  - \( p \) delivers \( m \) in \( v \) if \( p \) delivers \( m \) in \( v \).
  - Integrity
    - \( p \) delivers \( v \) if \( p \) is in the view.

Examples

- Ensemble: reliable group communication toolkit
  - Previous talk
IP-multicast

- IP: 224.0.0.1 - 224.0.0.255 (permanent)
- Multicast:
  - Yes: efficiency
  - No: Reliability, Ordering
- Group membership service:
  - Yes: Interface for group membership change, Group address expansion
  - No: Failure detector, Membership change notification

References

  - Section 4.5 Group Communication
  - Section 11.4 Multicast Communication
  - Section 14.2.2 Group Communication
  - ...