CSCI 5105

Instructor: Abhishek Chandra

Today

- Fault Tolerance in Distributed Systems
  - Types of Faults
  - Fault Tolerance Techniques

Faults

- What is a fault?
  - Cause of an error or a failure
- Examples of faults?
  - Machines crash, disks fail, bugs occur, packets lost
- How is the effect of faults different in single-machines vs. distributed systems?

Fault Tolerance

- Fault Tolerance
  - Ability of a system to continue functioning normally in the presence of faults
- Questions:
  - How can we detect faults?
  - How can we hide the effects of faults?
  - How can we recover from failures?
Fault Tolerance Properties

- **Availability**: What percentage of time is a system available for use?
- **Reliability**: How long can a system stay up continuously?
- **Safety**: Small failures should not have catastrophic effects
- **Maintainability**: How easy is it to repair faults?

Types of Faults

- **Transient faults**: Happen once and disappear
  - E.g.: Temporary network outage
- **Intermittent faults**: Happen occasionally but unpredictably
  - E.g.: System deadlocks, race conditions
- **Permanent faults**: Faulty component must be repaired/replaced
  - E.g.: Disk crash, software bug

Failure Models

- Distributed System: Set of communicating servers
- **Crash Failure**: Server working correctly until crash
- **Omission failure**: Server fails to respond to incoming messages
- **Timing failure**: Server’s response is too slow or too fast
- **Response failure**: Incorrect response from server
- **Arbitrary (Byzantine) failure**: Incorrect but undetectable, could be malicious

Failure Detection

- **Fail-stop**: Server stops and others can detect this failure
- **Fail-silent**: No responses from server
- **Fail-safe**: Server is working incorrectly, but has a benign failure
- **Byzantine failures**: Server is working incorrectly, but cannot be detected
Fault Detection Techniques

- Timeout-based
  - Heartbeat messages: Ping periodically
  - Regular communication: Getting steady stream of messages
- Distinguishing between node and network faults
  - Use multiple points of probing
- Byzantine faults
  - Possible under limited scenarios

Fault Tolerance Techniques

- Redundancy and agreement
  - Hiding effect of faults
- Recovery and rollback
  - Bringing system to a consistent state

Redundancy

- Information redundancy
  - Add extra bits of information
- Time redundancy
  - Repeat failed operations
- Physical redundancy
  - Replicate system components

Replication

- Use multiple replicas/copies of a single server
- Client-server:
  - Redundant servers
- Group of servers:
  - Agreement among servers
**Amount of Redundancy**
- Depends on:
  - How many faults can a system handle?
  - What kind of faults can happen?
  - Is it a client-server system or group of servers?
- k-fault tolerant system:
  - Can handle k faulty servers

**Client-Server Environment**
- Client needs only one response
  - Each server has ability to respond
- How many total servers do we need for a k-fault tolerant system if failures are:
  - Fail-stop/fail-silent?
  - Byzantine?

**Voting**
- Replicate component across servers
- Vote on the result
  - Majority voting
  - Other values for quorum