CSCI 5105

Instructor: Abhishek Chandra

Today

- Fault Tolerance in Distributed Systems
  - Overview and Basics
  - 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?

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

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?

5

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?

6

Fault Tolerance Metrics

- Availability
  - A(t) = fraction of time in [0,t) that system is up
- Reliability
  - R(t) = Prob(system is up during [0,t) | system is up at T=0)
- Mean time to failure (MTTF)
- Mean time to repair (MTTR)
- Mean time between failures (MTBF)
  - MTBF = MTTF + MTTR

7

Failure Models

- Distributed System: Set of communicating servers
- Crash Failure: Server working correctly until crash
- Omission failure: Server fails to respond to incoming messages
- Commission failure: Server does something that it should not have done
- 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

8

Failure Detection

- How can a process detect that another process has failed?
- Depends on the system model
- Asynchronous: No bounds on process execution speeds or message delivery times
- Synchronous: Process execution speeds or message delivery times are bounded
- Partially synchronous: Mostly behaves as a synchronous system, but no bound on asynchronous behavior

9

Failure Detection Modes

- Fail-stop: Server stops and others can detect this failure
  - Assumes reliable links and bounds on delays
- Fail-noisy: Failure will be eventually detected
  - Some period of time when state may be unreliable
- Fail-silent: No responses from server
  - Reliable links, but crash and omission failure cannot be distinguished
- Fail-safe: Server has arbitrary failure, but is benign
- Fail-arbitrary (Byzantine) failures: Server has arbitrary failure, cannot be detected, could be harmful

10

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
- Disseminating fault information through a network:
  - Gossiping-based
  - FUSE: Nodes arranged in a tree, faults cascasde up the tree

11

Fault Tolerance Techniques

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

12

Redundancy

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

13

Recovery

- Checkpointing
- Message logging
- Rebooting

14