CPSC 426/526
Failures in Distributed Systems

Ennan Zhai
Computer Science Department
Yale University
Recall: Lec-14

• In lec-14, we learned:
  - Difference between privacy and anonymity
  - Anonymous communications
  - Case study: Dissent [OSDI’12]
Lecture Roadmap

• Failures in Distributed Systems
• Root Causes
• How to handle failures
• Case Study: Catastrophic Failures
Failures in Distributed Systems

• What is the meaning of failure:
  - Your system does not function correctly
  - Your system implementation does not meet your spec
  - What is your intuition when you say some code has bugs?
Data Center Outages Generate Big Losses

Downtime in a data center can cost an average of $740,357 per incident -- cloud outage cost report in 2016.

Cost of Data Center Outages

Ponemon Institute and Vertiv are pleased to present the results of the latest Cost of Data Center Outages study. Previously published in 2010 and 2013, the purpose of this third study is to continue to analyze the cost behavior of unplanned data center outages. According to our new study, the average cost of a data center outage has steadily increased from $505,502 in 2010 to $740,357 today (or a 38 percent net change). Our benchmark analysis focuses on representative samples of organizations in different industry sectors that experienced at least one complete or partial unplanned data center outage during the past 12 months. Utilizing activity-based costing methods, this year’s analysis is derived from 63 data centers located in the United States. Following are the functional leaders within each organization who participated in the benchmarking process:
Failures in Distributed Systems

- Different aspects of system failures:
  - Reliability: Operation & job failures, data loss & corruption
  - Performance: Traffic overloaded, connection delay
  - Availability: Node and cluster downtime
  - Consistency: Permanent inconsistent replicas
  - Scalability: Function incorrectly in large-scale clusters
  - ... ...

Failures in Distributed Systems

• Different aspects of system failures:
  - Reliability: Operation & job failures, data loss & corruption
  - Performance: Traffic overloaded, connection delay
  - Availability: Node and cluster downtime
  - Consistency: Permanent inconsistent replicas
  - Scalability: Function incorrectly in large-scale clusters
  - ... ...
We’d like to share more about the service event that occurred on Monday, October 22nd in the US-East Region. We have now completed the analysis of the events that affected AWS customers, and we want to describe what happened, our understanding of how customers were affected, and what we are doing to prevent a similar issue from occurring in the future.

The Primary Event and the Impact to Amazon Elastic Block Store (EBS) and Amazon Elastic Compute Cloud (EC2)

Correlated failures resulting from EBS due to bugs in one EBS server
Reliability & Availability

Elastic Compute Cloud (EC2)

Elastic Block Store (EBS)
Reliability & Availability

Elastic Block Store (EBS)
Global Outages

Final Root Cause Analysis and Improvement Areas: Nov 18 Azure Storage Service Interruption

Posted on December 17, 2014

Jason Zander, CVP, Microsoft Azure Team

On November 18, 2014, many of our Microsoft Azure customers experienced a service interruption that impacted Azure Storage and several other services, including Virtual Machines. Following the incident, we posted a [blog](#) that outlined a preliminary Root Cause Analysis (RCA), to ensure customers understood how we were working to address the issue. Since that time, our highest priority has been actively investigating and mitigating this incident. Today, we’re sharing our final RCA, which includes a comprehensive outline of steps we’ve taken to mitigate against this situation happening again, as well as steps we’re taking to improve our communications and support response. We sincerely apologize and recognize the significant impact this service interruption may have had on your applications and services. We appreciate the trust our customers place in Microsoft Azure, and I want to personally thank everyone for the feedback which will help our business continually improve.

Root Cause Analysis

On November 18th [PST] (November 19th [UTC]) Microsoft Azure experienced a service interruption that resulted in intermittent connectivity issues with the Azure Storage service in multiple regions. Dependent services, primarily
A more tricky example
Lightning strikes Amazon's European cloud

**Summary:** The lightning strike damaged a power company's transformer, causing disruption to Amazon Web Services's European cloud, and may have affected Microsoft's BPOS as well.

The outage, which Amazon Web Services (AWS) acknowledged on Sunday evening, affected its Dublin-based Elastic Compute Cloud (EC2) and Relational Database Service (RDS) cloud services, among others. The damage to the electricity infrastructure may have affected Microsoft's Business Productivity Online Services (BPOS) cloud as well, Microsoft said in a separate statement.
Video App

Cloud Provider A

Cloud Provider B
Cloud Provider A

Video App

Cloud Provider B

Third-party infrastructure components
Cloud Provider A \( \rightarrow \) Video App \( \rightarrow \) Cloud Provider B

Third-party infrastructure components

ISP Router A \( \rightarrow \) ISP Router B \( \rightarrow \) ISP Router C
Video App

Cloud Provider A

Cloud Provider B

Third-party infrastructure components

ISP Router A

ISP Router B

ISP Router C

Power Source
Cloud providers do not usually share information about their dependencies.
Failures in Distributed Systems

• Different aspects of system failures:
  - Reliability: Operation & job failures, data loss & corruption
  - Performance: Traffic overloaded, connection delay
  - Availability: Node and cluster downtime
  - Consistency: Permanent inconsistent replicas
  - Scalability: Function incorrectly in large-scale clusters
  - ... ...
Failures in Distributed Systems

• Different aspects of system failures:
  - Reliability: Operation & job failures, data loss & corruption
  - Performance: Traffic overloaded, connection delay
  - Availability: Node and cluster downtime
  - Consistency: Permanent inconsistent replicas
  - Scalability: Function incorrectly in large-scale clusters
  - ... ...
**Consistency Failures**

**ZooKeeper** (synchronization service)  
*Issue #335.*

1. Nodes A, B, C start (w/ latex txid: 10)  
2. B becomes leader  
3. B crashes  
4. C becomes leader  
5. C commits new txid-value pair (11, X)  
6. A crashes, before committing the new txid 11  
7. C loses quorum and C crashes  
8. A and B are back online after C crashes  
9. A becomes leader  
10. A's commits new txid-value pair (11, Y)  
11. C is back online after A's new tx commit  
12. C announce to B (11, X)  
13. B replies diff starting with tx 12  
14. Inconsistency: A has (11, Y), C has (11, X)

**PERMANENT INCONSISTENT REPLICA**
Failures in Distributed Systems

• Different aspects of system failures:
  - Reliability: Operation & job failures, data loss & corruption
  - Performance: Traffic overloaded, connection delay
  - Availability: Node and cluster downtime
  - Consistency: Permanent inconsistent replicas
  - Scalability: Function incorrectly in large-scale clusters
  - ... ...
Scalability Failures

Large data

In HBase

Tens of minutes

Insufficient lookup operation

R1
R2
R3
R100K
R...

Insufficient lookup operation
Lecture Roadmap

• Failures in Distributed Systems
• Root Causes
• How to handle failures
• Case Study: Catastrophic Failures
Root Causes

- Misconfiguration: 31%
- Bugs: 20%
- Hardware: 15%
- Customer Environment: 25%
- 9%

5 Storage Systems, 2011
Root Causes

- Customer Environment: 25%
- Misconfiguration: 31%
- Hardware: 15%
- Bugs: 20%
- Other: 9%

More Cloud Systems

5 Storage Systems, 2011
Root-cause types: Logic

• Logic (29%)
  – Many domain-specific issues
Root-cause types: Error handling

- Logic
- Error handling (18%)
Example

- Developers think the errors *will never happen*
  - Code evolution may enable the errors
  - The judgment can be wrong
    ```java
    } catch (IOException e) {
        // will never happen
    }
    ```

- Error handling is difficult
  - Errors can be returned by 3rd party libraries
    ```java
    } catch (NoTransitionException e) {
        /* Why this can happen? Ask God not me. */
    }
    ```
Root-cause types: Optimization

- Logic
- Error handling
- **Optimization (15%)**
Root-cause types: Configuration

- Logic
- Error handling
- Optimization
- **Configuration (14%)**
Example

**Configuration File**

Specify parameters

MySQL

```plaintext
datadir = /var/lib/mysql
tmpdir = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 127.0.0.1

# * Fine Tuning

key_buffer = 16M
max_allowed_packet = 16M
thread_stack = 192K
thread_cache_size = 8

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover = BACKUP
max_connections = 100

# Query Cache Configuration

query_cache_limit = 1M
query_cache_size = 16M

# * Logging and Replication
```
Example from MySQL

general_log = /var/log/mysql/mysql.log
Example from MySQL

general_log = /var/log/mysql/mysql.log

Problem Type: Value type error

Description: The parameter “general_log” should be an integer, rather than path (string). In MySQL, there is another parameter “general_log_file” used to point the log path.

Impact: Log cannot be correctly written.
Root-cause types: Race

- **Race (12%)**
  - < 50% local concurrency bugs
    - Buggy thread interleaving
    - Tons of work
  - > 50% distributed concurrency bugs
    - Reordering of messages, crashes, timeouts
    - More work is needed
      - SAMC [OSDI ’14]
Root-cause types: Hang

- **Hang (4%)**
  - Classical deadlock
  - **Un-served jobs, stalled operations, ...**
    - Root causes?
    - How to detect them?
Root-cause types: Space

- Space (4%)
  - Big data + leak = Big leak
  - Clean-up operations must be flawless.
Root-case types: Load

- **Load (4%)**
  - Happen when systems face high request load
  - Relates to QoS and admission control
Lecture Roadmap

- Failures in Distributed Systems
- Root Causes
- **How to handle failures**
- Case Study: Catastrophic Failures
Failure Handling in Distributed Systems
Failure Handling in Distributed Systems

- Service initialization
- Changing network paths
- Upgrading software components
- Outage

Service Runtime
Failure Handling in Distributed Systems

- **Failure Prevention**
  - Service initialization

- **Service Runtime**
  - Changing network paths
  - Upgrading software components

- **Post-Failure Troubleshooting**
  1. Diagnosis tools
  2. Accountability
  3. Provenance
  4. ... ...

- **Outage**
Failure Handling in Distributed Systems

Failure Prevention

- Service initialization

Failure Tolerance

- Changing network paths
- Upgrading software components

Outage

Post-Failure Troubleshooting

1. Diagnosis tools
2. Accountability
3. Provenance
4. ...
Lecture Roadmap

• Failures in Distributed Systems
• Root Causes
• How to handle failures
• Case Study: Catastrophic Failures
Next Lecture

• In the lec-16, I will learn:
  - Misconfigurations in distributed systems
  - How important they are
  - How to handle misconfiguration