Understanding Real-World Timeout Problems in Cloud Server Systems

Ting Dai, Jingzhu He, Xiaohui (Helen) Gu, Shan Lu*

*NC State University   *University of Chicago
Real-world timeout problems

Amazon DynamoDB service was down for 5 hours.
https://aws.amazon.com/cn/message/5467D2/

Storage servers

Metadata server

Timeout
Timeout
Timeout
Timeout

Send req 1
Send req 2
Send req 3
...

Bug

No proper limit of retry.

Overloaded
A Motivating Example

HDFS-6166

Send job req 1
Send job req 2
Send job req 3

Move data block

Thread quota exceeded error

Balancer

DataNode 1

DataNode 2

Timeout

1 min

Misconfigured timeout value

NC STATE UNIVERSITY Computer Science
A Motivating Example

HDFS-6166

Balancer

DataNode 1

DataNode 2

Send job req 1

Send job response 1

Move data block

patch
What are timeout bugs?

Timeout bugs happen when the server applications lack proper configuration and handling of the timeout events.
Why are timeout bugs are prevalent?

• Cloud server systems have become increasingly complex.
• Timeout is one of the commonly used mechanisms to handle unexpected failures in distributed computing environments.
Methodology

• We searched timeout bugs in 11 popular cloud server applications from Apache JIRA.
• We extensively studied 156 bugs.

<table>
<thead>
<tr>
<th>System</th>
<th># of bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cassandra</td>
<td>17</td>
</tr>
<tr>
<td>Flume</td>
<td>13</td>
</tr>
<tr>
<td>Hadoop Common</td>
<td>15</td>
</tr>
<tr>
<td>Hadoop Mapreduce</td>
<td>15</td>
</tr>
<tr>
<td>Hadoop Yarn</td>
<td>4</td>
</tr>
<tr>
<td>HDFS</td>
<td>26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th># of bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBase</td>
<td>28</td>
</tr>
<tr>
<td>Phoenix</td>
<td>6</td>
</tr>
<tr>
<td>Qpid</td>
<td>20</td>
</tr>
<tr>
<td>Spark</td>
<td>4</td>
</tr>
<tr>
<td>Zookeeper</td>
<td>8</td>
</tr>
</tbody>
</table>

Total 156
Methodology

We classified the 156 timeout bugs in regard to three characteristics:

- root causes
- impact to systems or applications
- diagnosability
Misused timeout value & Missing timeout checking **dominate.**
Root Cause

**Misused timeout value** *(65 bugs)*
- Misconfigured timeout value *(38 bugs)*
- Ignored timeout value *(10 bugs)*
- Incorrectly reused timeout value *(8 bugs)*
- Inconsistent timeout value *(4 bugs)*
- Stale timeout value *(3 bugs)*
- Improper timeout scope *(2 bugs)*
An Ignored Timeout Value Example

HBase-8581

// HTable class
operationTimeout = isMetaTable(tableName) ?
HConstants.DEFAULT_HBASE_CLIENT_OPERATION_TIMEOUT :
config.getInt(HConstants.HBASE_CLIENT_OPERATION_TIMEOUT...);

The configured timeout value is ignored
Observation

Misused timeout value bugs often occur when:
- lack extensive testing on timeout configurations;
- do not understand the system’s timeout mechanisms.

Setting proper timeout value is challenging.
Root Cause

Missing timeout checking (42 bugs)
- Missing timeout for network communication (26 bugs)
- Missing timeout for synchronization (16 bugs)
A Missing Timeout Example

Zookeeper-2224

//FourLetterWordMain class
74 sock = new Socket(host, port);

//Socket class
425 connect(address);

• Missing timeout for network communication
Another Missing Timeout Example

HBase-13971

seqNumAssignedLatch\texttt{.wait()};

logSeqNum = sequence;
seqNumAssignedLatch\texttt{.countDown()};

Missing timeout for synchronization

RegionServer

\texttt{getSequenceId}

\texttt{logSeqNum}

WALKey

\texttt{setLogSeqNum}

FSHLog
Missing timeout bugs often occur when developers do not consider the system’s failover mechanisms.
Root Cause

**Improper timeout handling (16 bugs)**

- Insufficient/missing retries (8 bugs)
- Excessive retries (3 bugs)
- Incorrect retry (2 bugs)
- Incomplete abort (2 bugs)
- Incorrect abort (1 bug)
Insufficient/missing retries cause job failure

Hadoop-3831

Job failure

DFSClient

Read files req

Timeout

DataNode 1

Try other DataNodes

File contents

Insufficient/missing retries cause job failure

Hadoop-3831

Job failure

DFSClient

Read files req

Timeout

DataNode 1

Try other DataNodes

File contents

Insufficient/missing retries cause job failure

Hadoop-3831

Job failure

DFSClient

Read files req

Timeout

DataNode 1

Try other DataNodes

File contents
Observation

It is **challenging** to implement proper timeout handling mechanisms, which requires developers to understand:

- the **tradeoffs** between handling schemes (e.g., aborting v.s. retry);
- each handling scheme’s **impact** to the systems and applications.
Root Cause

Unnecessary timeout protection (7 bugs)

Those bugs occur when developers mistakenly use timeout retry mechanisms over operations which requires continuous or at-most-once-execution semantics.
Root Cause

Clock drifting (7 bugs)

Those bugs occur when the clocks are out-of-synchronization, the elapsed time is miscalculated, which generates a wrong timer value.
Impact

- **System unavailability**: 26%
- **Job failure**: 33%
- **Performance degradation**: 2%
- **Data loss**: 40%
Unavailability caused by missing timeout

HDFS-4858

Power outage

NameNode

DataNodes

Secondary NameNode

TCP RST

Reboot

DataNodes miss timeout. HDFS becomes unavailable.
Diagnosability

- No error message: 60%
- Correct error message: 29%
- Wrong error message: 12%

Only 29% timeout bugs report the correct error messages.
A Wrong Error Message Example

Cassandra-3651

```
try {
    ...
} catch (TimeoutException e) {
    throw new UnavailableException();
}
```
Future Work

Enhanced timeout detection tool

- Feature extraction
- Semi-supervised machine learning scheme
State of the Art

General bug studies [Gunawi et al. SoCC’14, Huang et al. SoCC’15, etc]
  ❖ They found timeout bugs widely exist in distributed systems.

Specific bug studies [Yin et al. SOSP’11, Wang et al. IC2E’15, etc]
  ❖ Misconfigurations; Data Corruption; Performance; Concurrency.

Performance bug diagnosis [Dean et al. SoCC’14, etc]
  ❖ Existing tools cannot detect/diagnose performance anomalies caused by timeout bugs [ICAC’15].

Concurrency bug detection/fix [Jin et al. OSDI’12, PLDI’12, etc]
  ❖ Our study reveals under-studied types of root causes for concurrency bugs: missing, misused, and unnecessary timeout.
Conclusion

• We perform a characteristic study of 156 real-world timeout bugs in 11 popular open source cloud server systems.

• 81% timeout bugs are caused by either misused timeout values or missing timeout checking.

• Timeout problems have serious impact to both cloud server systems and applications.

• Existing timeout issues are difficult to diagnose with 71% bugs producing no error message or misleading error messages.

Thank you!