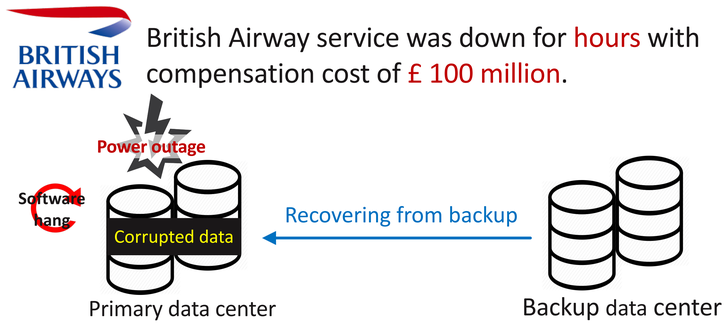
DScope: Detecting Real-World Data Corruption Hang Bugs in Cloud Server Systems

NC STATE UNIVERSITY

Ting Dai , Jingzhu He , Xiaohui Gu , Shan Lu*, Peipei Wang

NC State University *University of Chicago

Motivation



Contributions

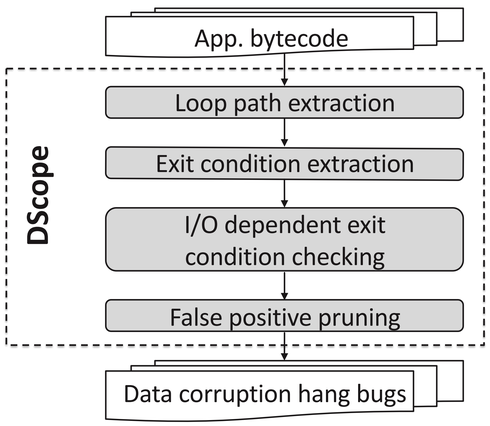
- **DScope:** a static analysis tool to detect data-corruption related hang bugs.
  - Detecting potential infinite loops caused by data corruptions;
  - Pruning always-exit loops through loop stride and bound analysis;
  - Categorizing real-world data-corruption hang bugs into four common types.

Overview

Example: Hadoop-8614

```
183 public static void skipFully(InputStream in, long len) throws IOException {
184   while (len > 0) {
185     long ret = in.skip(len);
        ...
189     len -= ret;
    } }
```

The loop stride (ret) is always 0 when in is corrupted.



False Positive Filtering

```
307 public static long readVLong(DataInput stream) throws IOException {
308   byte firstByte = stream.readByte();
309   int len = decodeVIntSize(firstByte);
      ...
314   for (int idx = 0; idx < len-1; idx++) {
        ...
318   }
319 }
```

len is I/O dependent

It's a FP b/c the loop stride is always 1 and the upper bound (len-1) is fixed.

- **False positive conditions:**
  - The loop stride is always **positive** when the loop index has a fixed **upper** bound;
  - The loop stride is always **negative** when the loop index has a fixed **lower** bound.

Data Corruption Hang Bug Types

1 **Error codes** returned by I/O operations **directly** affect loop strides.

e.g., Hadoop-8614

2 Corrupted **data content indirectly** affects loop strides.

Example: HDFS-13514

```
194 BUFFER_SIZE = conf.getInt();        Corrupted configuration file

78  private void readLocalFile(Path path, ...) throws IOException {
      ...                                   0
84    byte[] data = new byte[BUFFER_SIZE];
85    long size = 0;
86    while (size >= 0) {
87      size = in.read(data);          empty array
    } }
```

The loop stride (size) is always 0 when conducting read op on an empty array.

3 **Improper exception handling directly** affects loop strides.

Example: Yarn-6991

```
64  try {
73    //create pidFile
        ...
79  } catch (IOException ioe) {
80    LOG.info("...failed...");
81  }
```

throw exception

silent failure

```
88  File f = new File(pidFile);
89  while( !f.exists() ) {
        ...
91    Thread.sleep(500);
        ...
99  }
```

Thread #1

The silent failure makes the loop exit condition always be false.

Thread #2

4 **Improper exception handling indirectly** affects loop strides.

Example: Cassandra-9881

```
120 while (!dataFile.isEOF()) {
      ...
129   try {
130     decorateKey(ByteBufferUtil.readWithShortLength(dataFile));
        ...
134     dataSize = dataFile.readLong();
139   } catch (Throwable th) {
140     //ignore exception
    } }
```

throw exception

The improper exception handling makes the loop stride always be 0 (readLong API is skipped).

Detection Results

#	Bug name	System version	Bug type	Known or new	Detected: DScope	Findbugs	Infer
1	Cassandra-7330	2.0.8	#1	known	✓	×	×
2	Cassandra-9881	2.0.8	#3	known	✓	×	✓
3	Compress-87	1.0	#1	known	✓	×	×
4	Compress-451	1.0	#2	new	✓	×	×
5	Hadoop-8614	0.23	#1	known	✓	×	×
6	Hadoop-15088	2.5	#1	new	✓	×	×
7	Hadoop-15415	0.23	#2	new	✓	×	×
8		2.5	#2	new	✓	×	×
9	Hadoop-15417	0.23	#2	new	✓	×	×
10		2.5	#2	new	✓	×	×
11	Hadoop-15424	2.5	#1	new	✓	×	×
12	Hadoop-15425	2.5	#1	new	✓	×	×
13	Hadoop-15429	0.23	#2	new	✓	×	×
14		2.5	#2	new	✓	×	×
15	HDFS-4882	0.23	#3	known	✓	×	×
16	HDFS-5892	2.5	#2	known	✓	✓	×
17	HDFS-13513	2.5	#2	new	✓	×	×
18	HDFS-13514	2.5	#2	new	✓	×	×
19	Mapreduce-2185	0.23	#2	known	✓	×	×
20	Mapreduce-2862	0.23	#2	known	✓	×	×
21	Mapreduce-6990	0.23	#1	new	✓	×	×
22	Mapreduce-7088	2.5	#1	new	✓	×	×
23	Mapreduce-7089	2.5	#1	new	✓	×	×
24	Yarn-163	0.23	#1	known	✓	✓	×
25	Yarn-2905	2.5	#1	known	✓	×	×
26	Yarn-6991	0.23	#4	new	✓	×	×
27		2.5	#4	new	✓	×	×
28	Hive-5235	1.0	#1	known	✓	×	×
29	Hive-13397	1.0	#2	known	✓	×	×
30	Hive-18142	1.0	#2	new	✓	×	×
31	Hive-18216	2.3.2	#1	new	✓	×	×
32	Hive-18217	2.3.2	#1	new	✓	×	×
33	Hive-18219	1.0	#2	new	✓	×	×
34		2.3.2	#2	new	✓	×	×
35	Hive-19391	1.0	#2	new	✓	×	×
36	Hive-19392	1.0	#2	new	✓	×	×
37		2.3.2	#2	new	✓	×	×
38	Hive-19395	1.0	#1	new	✓	×	×
39	Hive-19406	2.3.2	#2	new	✓	×	×
40	Kafka-6271	0.10	#1	new	✓	×	×
41	Lucene-772	2.1	#2	new	✓	×	×
42	Lucene-8294	2.1	#2	known	✓	×	×

- DScope identifies **42** true data corruption hang bugs including **29** new bugs.
- Findbugs identifies **2** true data corruption hang bugs.
- Infer identifies **1** true data corruption hang bug.

Conclusion

- DScope is a new data corruption hang bug detection tool for cloud server systems.
  - Combines candidate bug discovery and false positive pattern filtering.
  - Evaluated over 9 cloud server systems and detects **42** true corruption hang bugs including **29** new bugs.

Supported in part by NSF CNS1513942 grant and NSF CNS1149445 grant.