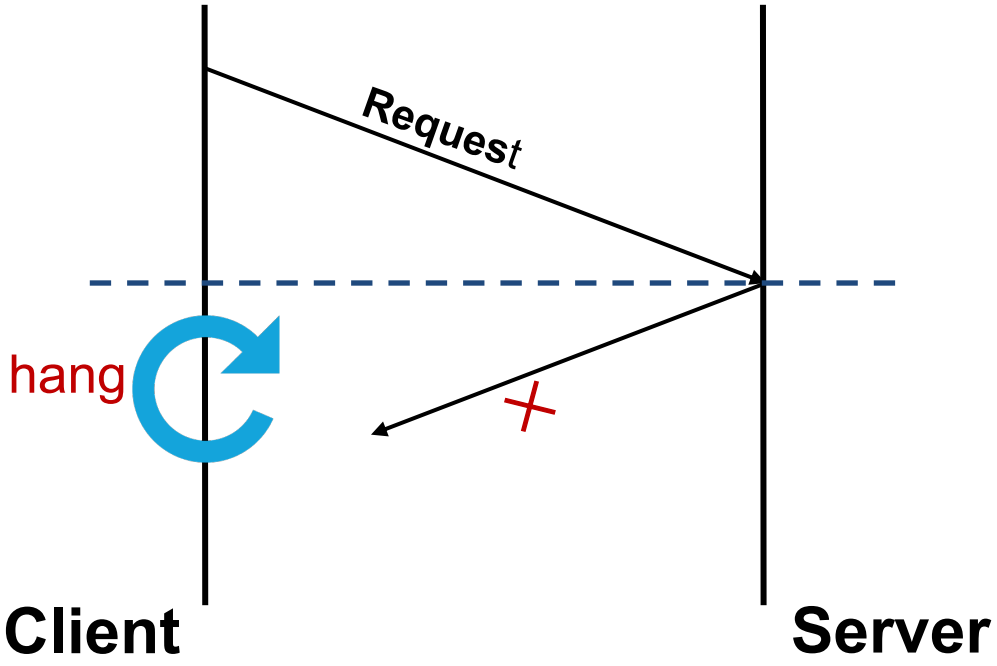
TScope: Automatic Timeout Bug Identification for Server Systems

Jingzhu He, Ting Dai, Xiaohui (Helen) Gu
NC State University

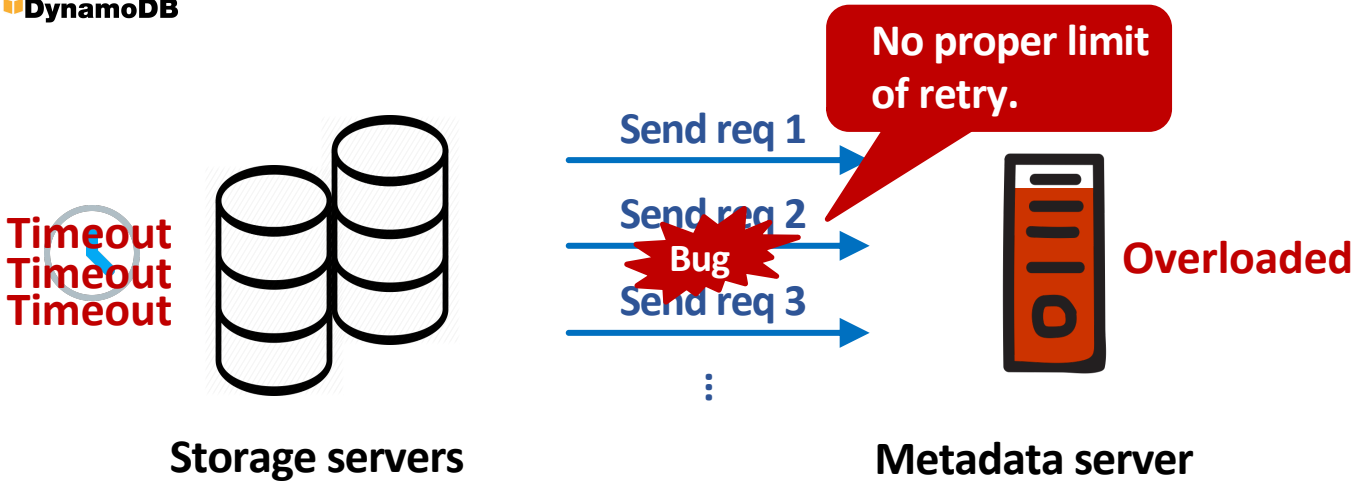
What Is Timeout?



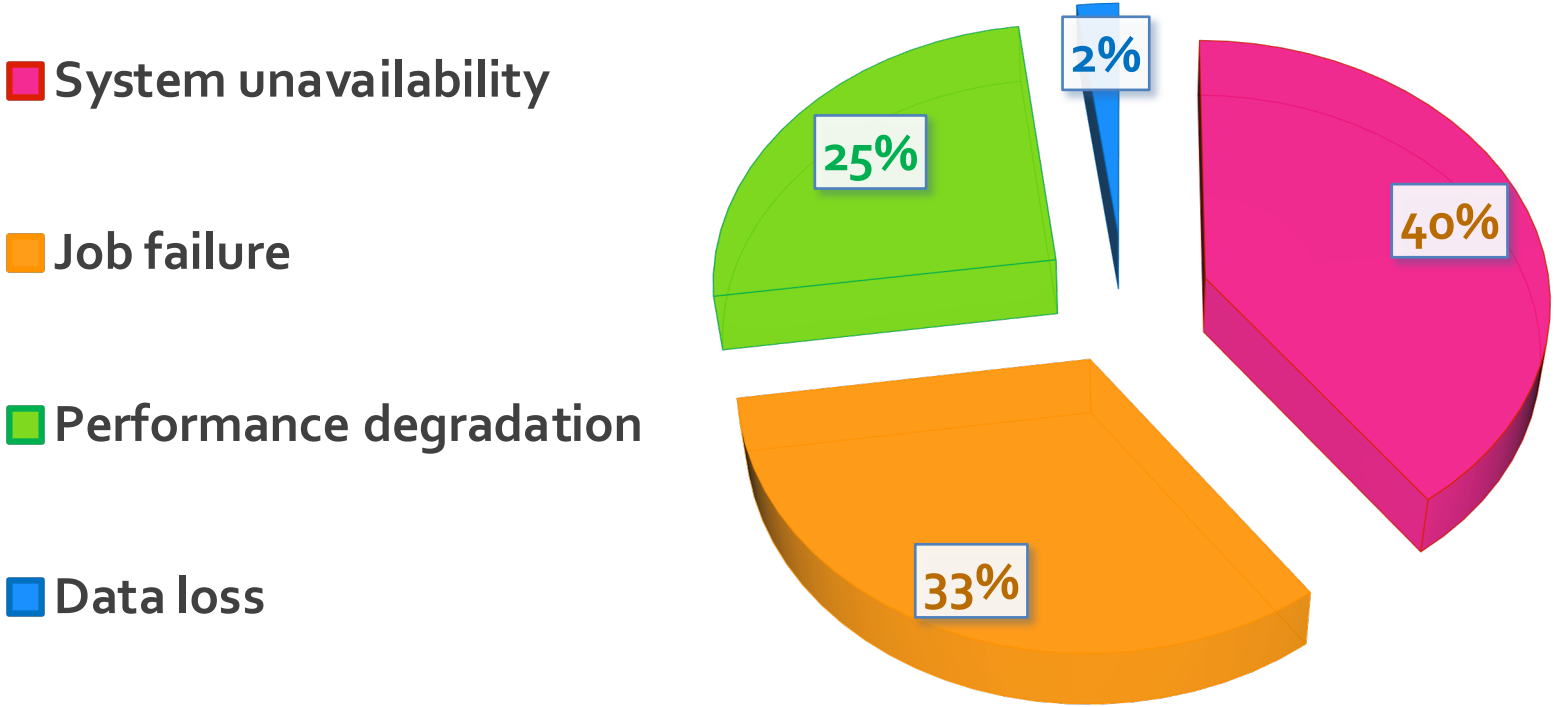
Real-world Timeout Problems

Amazon DynamoDB service was down for 5 hours.

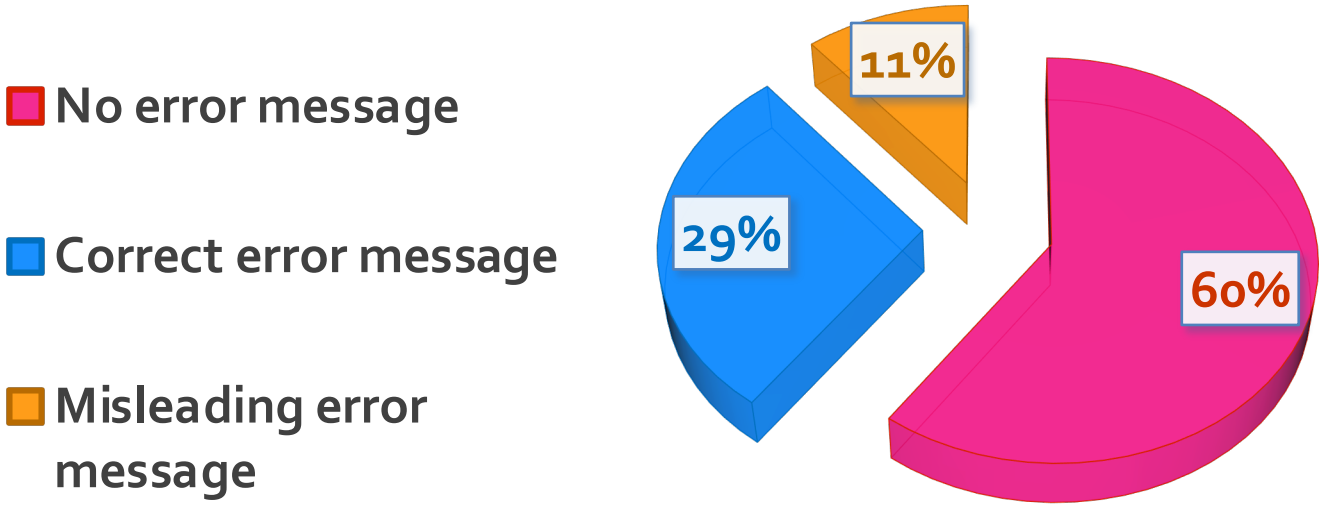
https://aws.amazon.com/cn/message/5467D2/



Impacts of Timeout Bugs



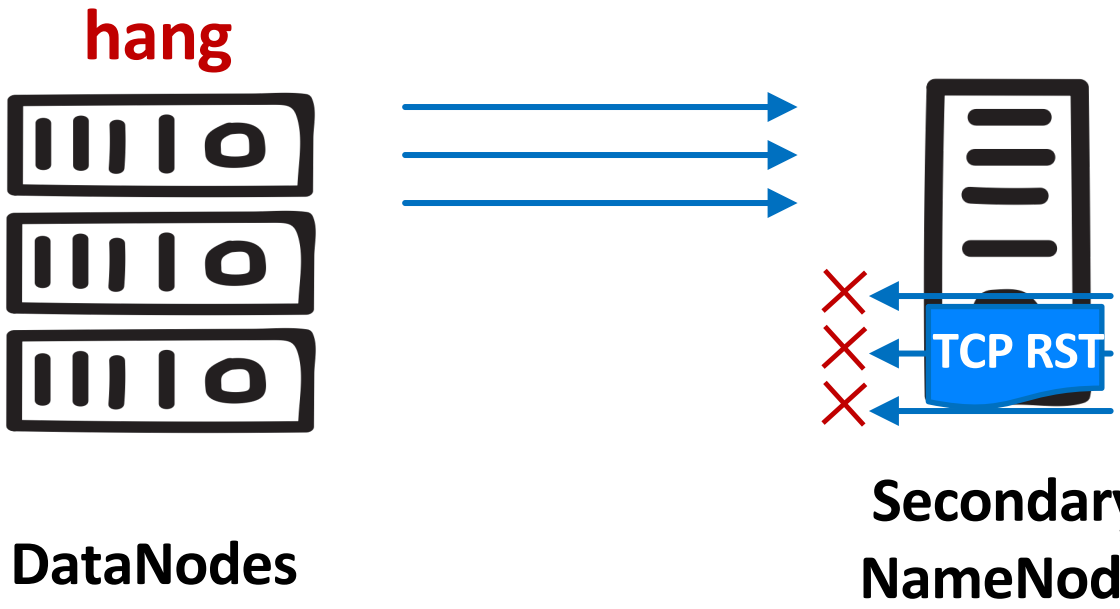
Diagnosability of Timeout Bugs



Only 29% timeout bugs report the correct error messages.

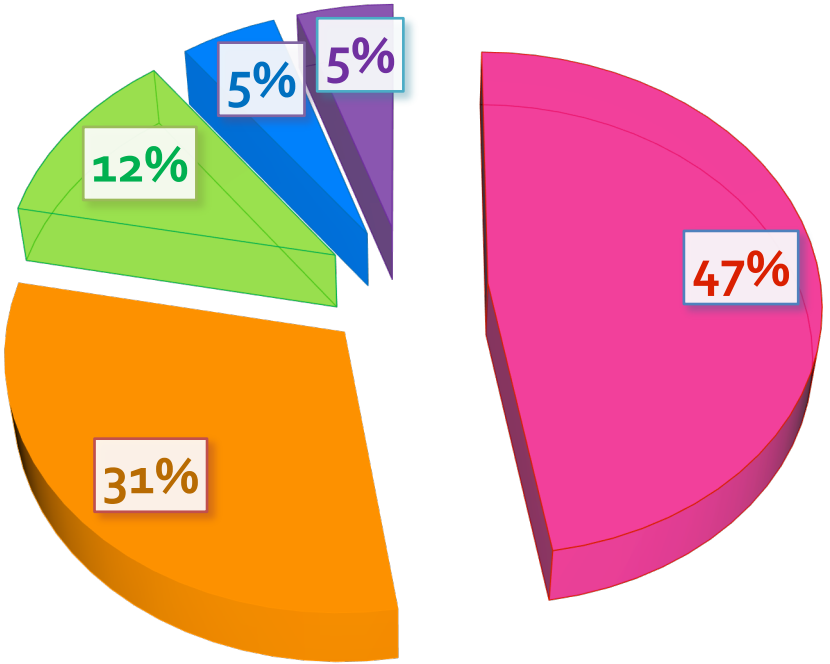
Motivating Example (Hadoop-11252)

Root cause: missing RPC timeout between DataNodes and the Secondary NameNode.

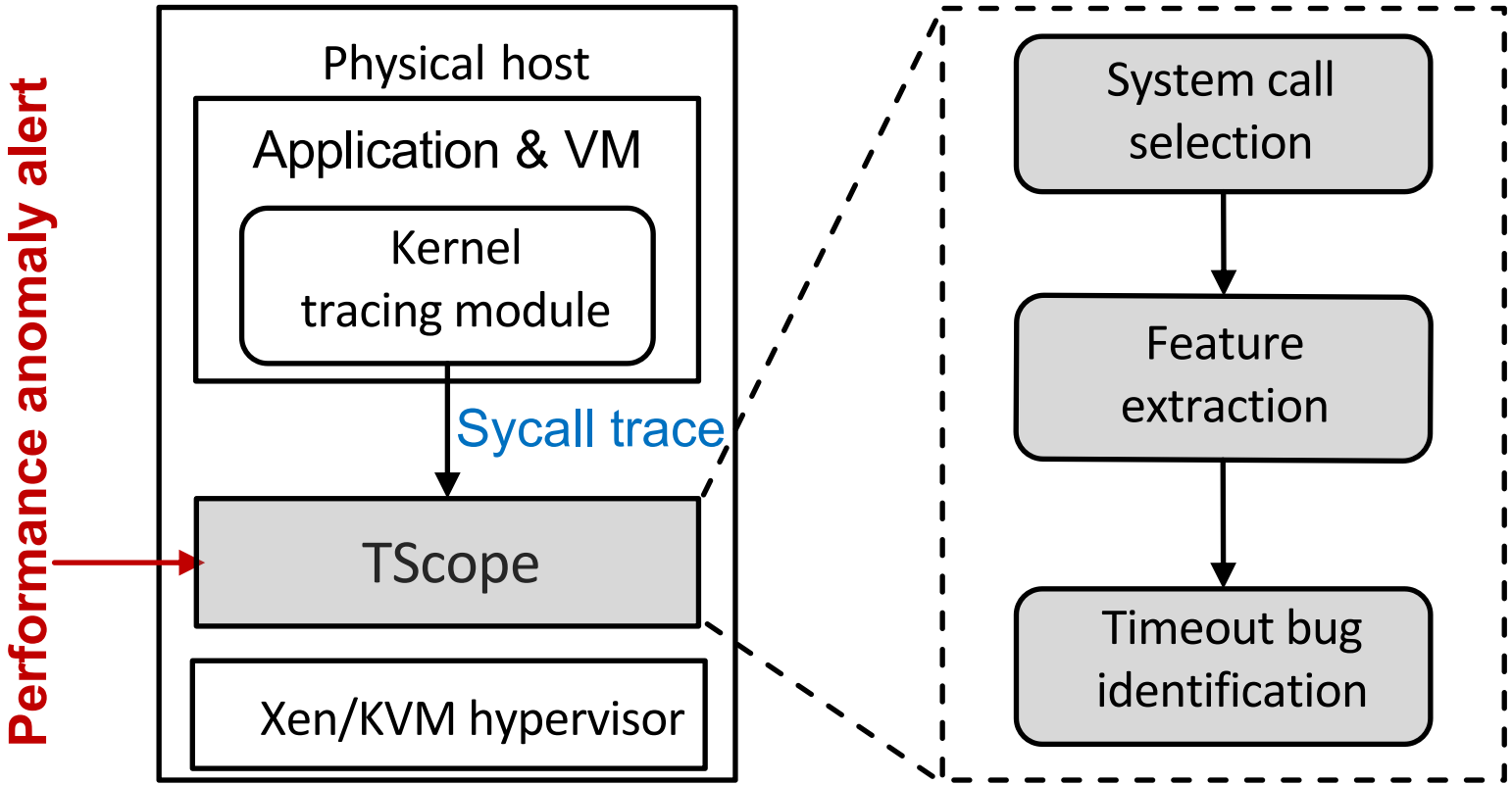


Root Causes of Timeout Bugs

- Misused timeout value
- Missing timeout checking
- Improper handling
- Unnecessary timeout
- Clock drifting



TScope's Overall Architecture



System Call Tracing

LTTng: incur negligible overhead to the server system compared to other tracing tools.

[14:24:43.520759222] syscall_entry_read: {cpu_id=...}, {..., pid=5004, ..., tid=5038}, {fd=3, ...}
[14:24:43.520759222] syscall_exit_read: {cpu_id=...}, {..., pid=5004, ..., tid=5038}, {ret = 30, ...}
[14:24:43.520760005] syscall_entry_write: {cpu_id=...}, {..., pid=5004, ..., tid=5038}, {fd=5, ...}
[14:24:43.520760218] syscall_exit_write: {cpu_id=...}, {..., pid=5004, ..., tid=5038}, {ret=1, ...}
[14:24:43.520943737] syscall_entry_poll {cpu_id=...}, {..., pid=5004, ..., tid=5038}, {..., timeout_msecs=60000}
[14:24:43.520943940] syscall_exit_poll: {cpu_id=...}, {..., pid=5004, ..., tid=5038}, {ret = -516, ...}

System Call Selection

- System calls with timeout related parameters
- System calls related to network and synchronization
- System calls used by timeout configuration functions

Selecting System Calls with Timeout Related Parameters

- Manually examine all the Linux system calls and discover those system calls that contain timeout related parameters.

- Example:
  syscall_select: has a timeout to determine how long a program should wait for files to become ready for I/O operations.
  syscall_futex: has a timeout to determine how long a synchronization operation should be blocked.

Selecting System Calls Related to Network and Synchronization

- Manually extract all the system calls which are used by network communication or synchronization.

- Example:
  syscall_connect: connects a socket to a specified address.
  syscall_fsync: synchronizes a file's state with storage devices.

Selecting System Calls Used by Timeout Configuration Functions

- Check library functions which provide timeout configurations in standard C or Java libraries.
  For example, wait() of java.lang.Object, sleep() and join() of java.lang.Thread.

- Write simple programs to run those functions and collect the system calls produced by those functions.

Timeout Bug Identification

- **Anomaly detection**: Use SOM (Self-Organizing Map) model to detect system calls with abnormal execution time.
- **Classification**: Identify timeout bugs by examining whether abnormal system calls contain timeout related parameters.
- **Example**:
  Cassandra-5064 (non-timeout bug): *sys_sche_yield* ✗
  MapReduce-5066 (timeout bug): *sys_epoll_wait* ✓

Benchmark

- **10 Server systems**: built by Java and C, 6 systems are set up in distributed modes.
- **19 bugs**: 12 timeout bugs and 7 non-timeout bugs.
- **Workloads**: run simple workloads on each system.
- **Diagnosability**:
  1) 17 out of 19 bugs produce no error messages or misleading error messages.
  2) All 12 timeout bugs produce no error messages or misleading error messages.

Timeout Bug Benchmark

Bug ID	Root cause	Impact
Hadoop-11252(v2.5.0)	Missing timeout	Hang
Hadoop-11252(v2.6.4)	Misused timeout	Hang
HDFS-10223	Misused timeout	Several hours slowdown
Phoenix-2496	Missing timeout	10 secs slowdown
MapReduce-5066	Missing timeout	Hang
Cassandra-7886	Wrong timeout handling	Hang
Flume-1842	Misused timeout	Several hours slowdown
Zookeeper-1366	Clock drifting	Crash
Tomcat-56684	Misused timeout	Hang
Flume-1819	Missing timeout	Slowdown
Flume-1316	Misused timeout	Slowdown
MapReduce-5724	Missing timeout	Hang

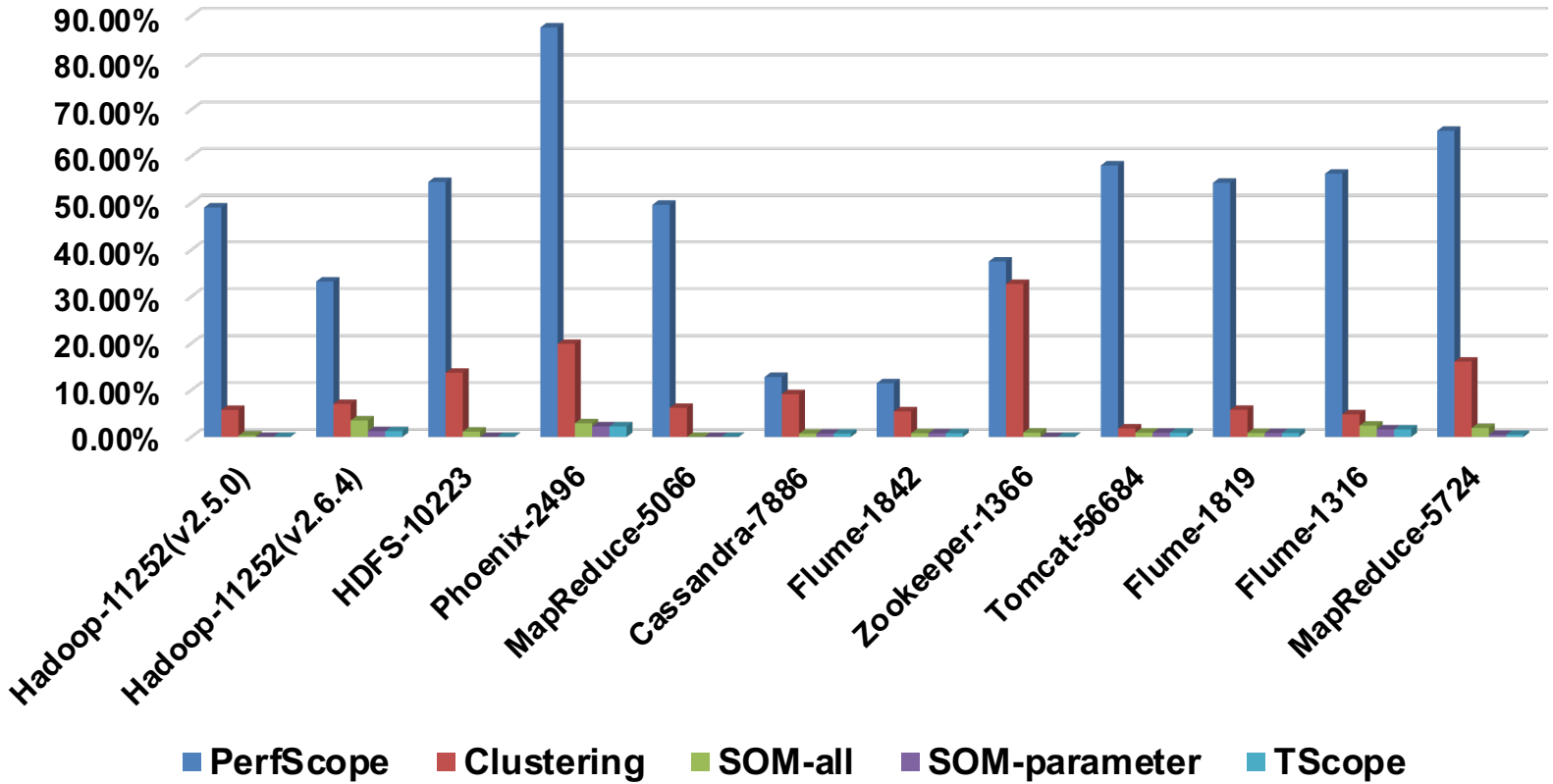
Non-timeout Bug Benchmark

Bug ID	Root cause	Impact
Cassandra-5064	Incorrect return value handling	Hang
Apache-37680	Incorrect flag	Hang
Tomcat-48827	Error in validating empty tag	Failure
Tomcat-53450	Upgrade a read lock to a write lock wrongly	Hang
MapReduce-3738	Hang on waiting for setting an atomic variable	Hang
MySQL-65615	Incorrect truncating tables	Slowdown
MySQL-54332	Two threads are deadlocked	Hang

Alternative Approaches

- PerfScope (*SOCC'14*)
- Clustering: DBScan algorithm
- SOM-all: do not perform system call selection
- SOM-parameter: only select system calls with timeout related parameters

False Positive Rates of Anomaly Detection



Explanation of False Positive Rates

- **Clustering based method (PerfScope and clustering)**:
  curse of dimensionality
  The time vectors formulate a 125-dimensional sparse matrix.

- **SOM based model (SOM-all and SOM-parameter)**:

1) Detected anomalies should be reduced when we consider less system calls in the selection set.
2) The selection set is more correlated to timeout when we narrow the selection set.

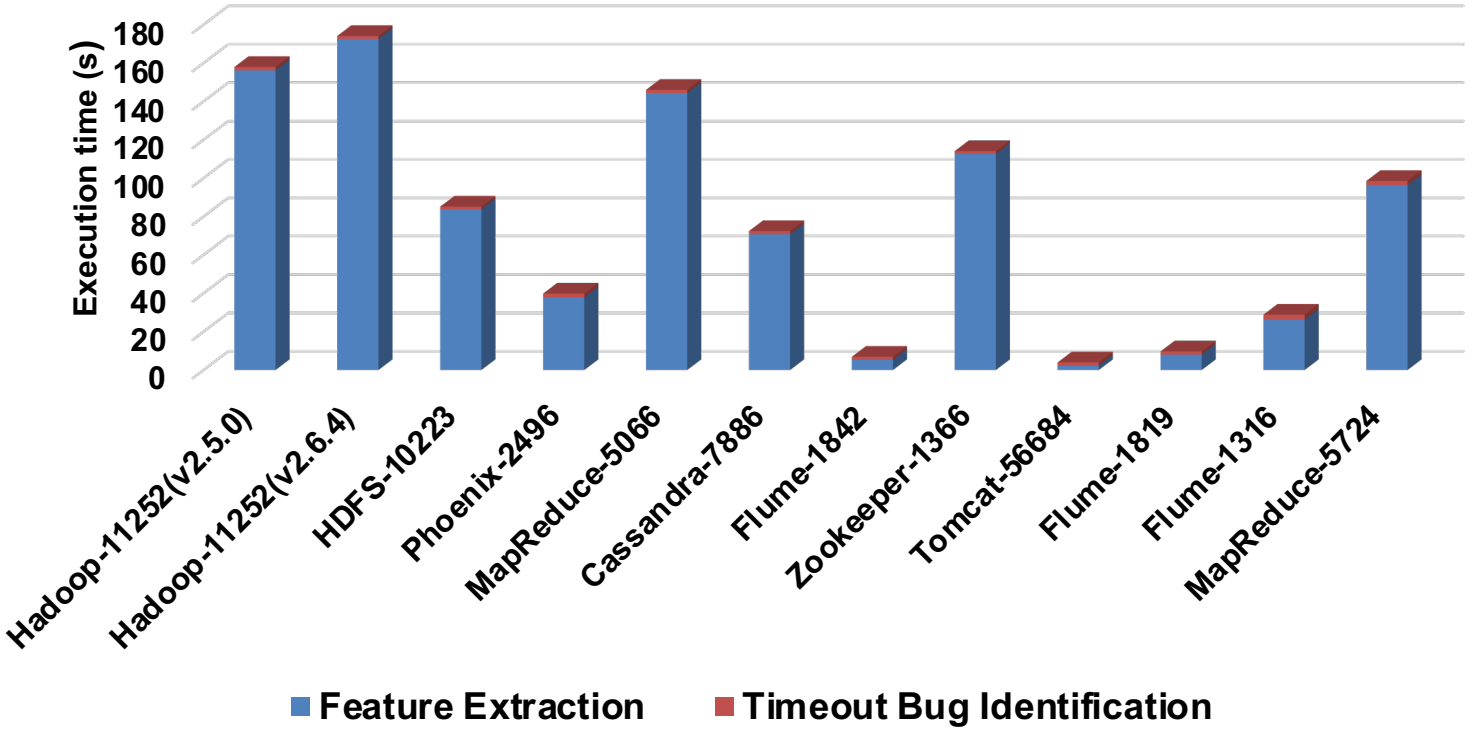
Classification Results of Timeout Bugs

Bug ID	PerfScope	Clustering	SOM-all	SOM-parameter	TScope
Hadoop-11252(v2.5.0)	✓	✓	✓	x	✓
Hadoop-11252(v2.6.4)	✓	x	x	x	✓
HDFS-10223	✓	x	✓	✓	✓
Phoenix-2496	x	✓	x	✓	x
MapReduce-5066	✓	x	✓	x	✓
Cassandra-7886	✓	✓	✓	x	✓
Flume-1842	✓	✓	✓	✓	✓
Zookeeper-1366	✓	✓	✓	✓	✓
Tomcat-56684	✓	✓	✓	✓	✓
Flume-1819	✓	x	✓	x	✓
Flume-1316	✓	x	✓	✓	✓
MapReduce-5724	✓	✓	✓	✓	✓

Classification Results of Non-timeout Bugs

Bug ID	PerfScope	Clustering	SOM-all	SOM-parameter	TScope
Cassandra-5064	x	x	✓	x	✓
Apache-37680	x	x	✓	x	✓
Tomcat-48827	x	x	x	x	✓
Tomcat-53450	x	✓	✓	✓	✓
MapReduce-3738	x	x	x	x	✓
MySQL-65615	x	x	x	✓	✓
MySQL-54332	x	x	x	✓	✓

TScope's Overhead



Related Work

- **Performance bug detection and diagnosis:** X-ray(OSDI'12), PerfCompass(TPDS'16), Fournier et al.(SIGOPS'10), PerfScope(SOCC'14)
  TScope identifies timeout bugs that cause performance problems.
- **Machine learning based performance debugging:** EntomoModel(MASCOTS'10), UBL(ICAC'12), Lee et al.(ICAC'16), Fchain(ICDCS'13)
  TScope performs unique feature selection to achieve high detection precision.
- **Static bug detection tools:** Jin et al.(PLDI'12), Toddler(ICSE'13), Dcatch(ASPLOS '17), Xiao et al.(ISSTA'13), Chen et al.(ICSE'14)
  TScope identifies timeout anomalies by performing feature selection statically and anomaly detection dynamically.

Conclusion

- TScope combines timeout related feature selection and runtime anomaly detection to achieve higher bug identification precision.
- TScope does not require any application instrumentation for bug detection.
- We implemented a prototype of TScope and conducted extensive experiments using 19 real world bugs.
- TScope is light-weight and efficient, which imposes less than 1% runtime overhead and produces identification results within minutes.

Acknowledgements

- Thanks for the comments from anonymous reviewers.
- TScope is supported in part of NSF CNS1513942 grant and NSF CNS1149445 grant.
- Thanks for the audience.