Understanding and Detecting Fail-Slow Hardware Failure Bugs in Cloud Systems

Gen Dong, Yu Hua, Yongle Zhang*, Zhangyu Chen, Menglei Chen
Huazhong University of Science and Technology
*Purdue Univeristy

USENIX ATC 2025

Hardware Failures in the Wild

HEALTHY

Everything is OK

FAIL-SLOW

Fine-grained and Still functional but with lower-than-expected performance

FAIL-STOP

Coarse-grained and Non-functional

Fail-Slow Hardware is a Real-World Problem

FAIL-SLOW

Severe

A 1Gb NIC card on a machine that suddenly only transmits at 1 kbps[1]

Fail-slow NVMe SSDs can degrade to SATA SSD or HDD-level performance[2]

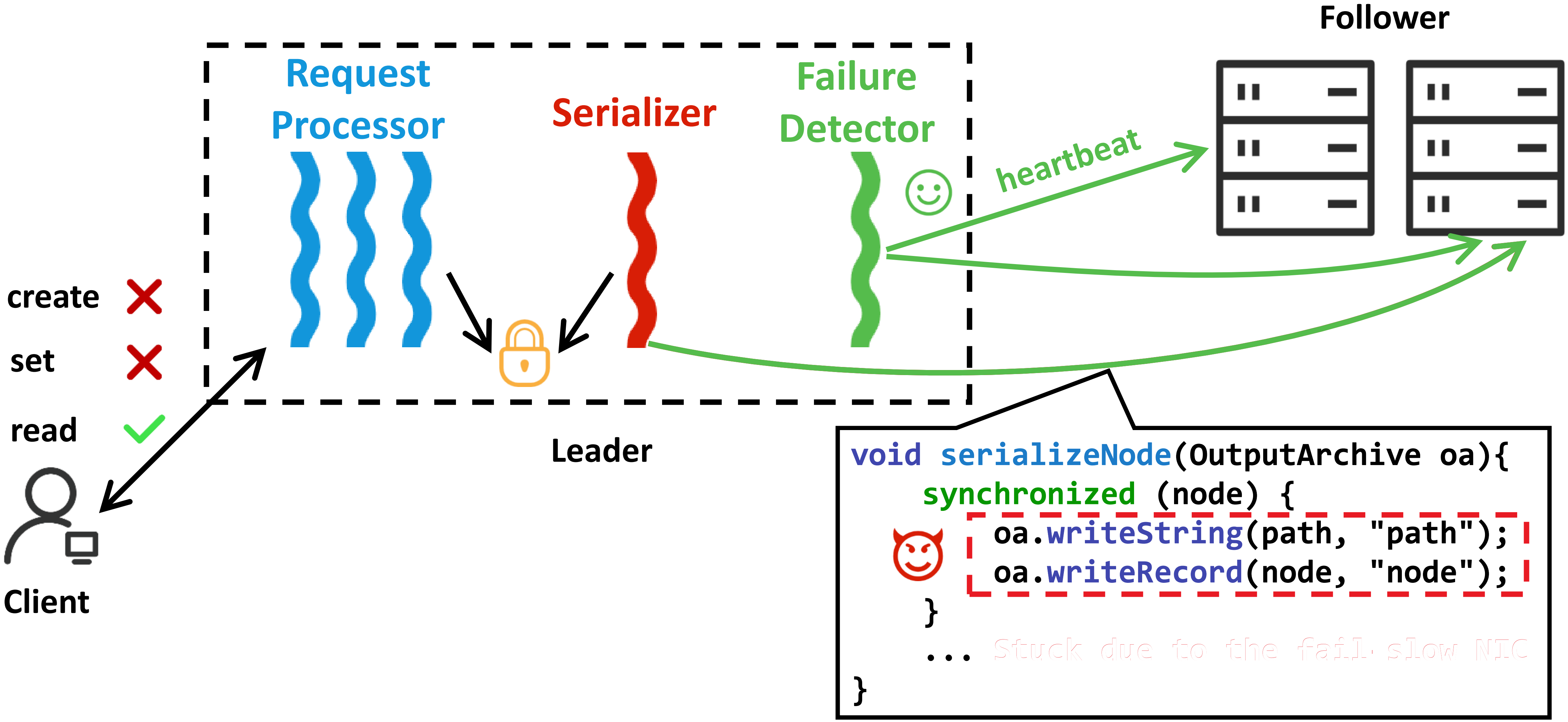
Common

As frequent as fail-stop incidents[2]

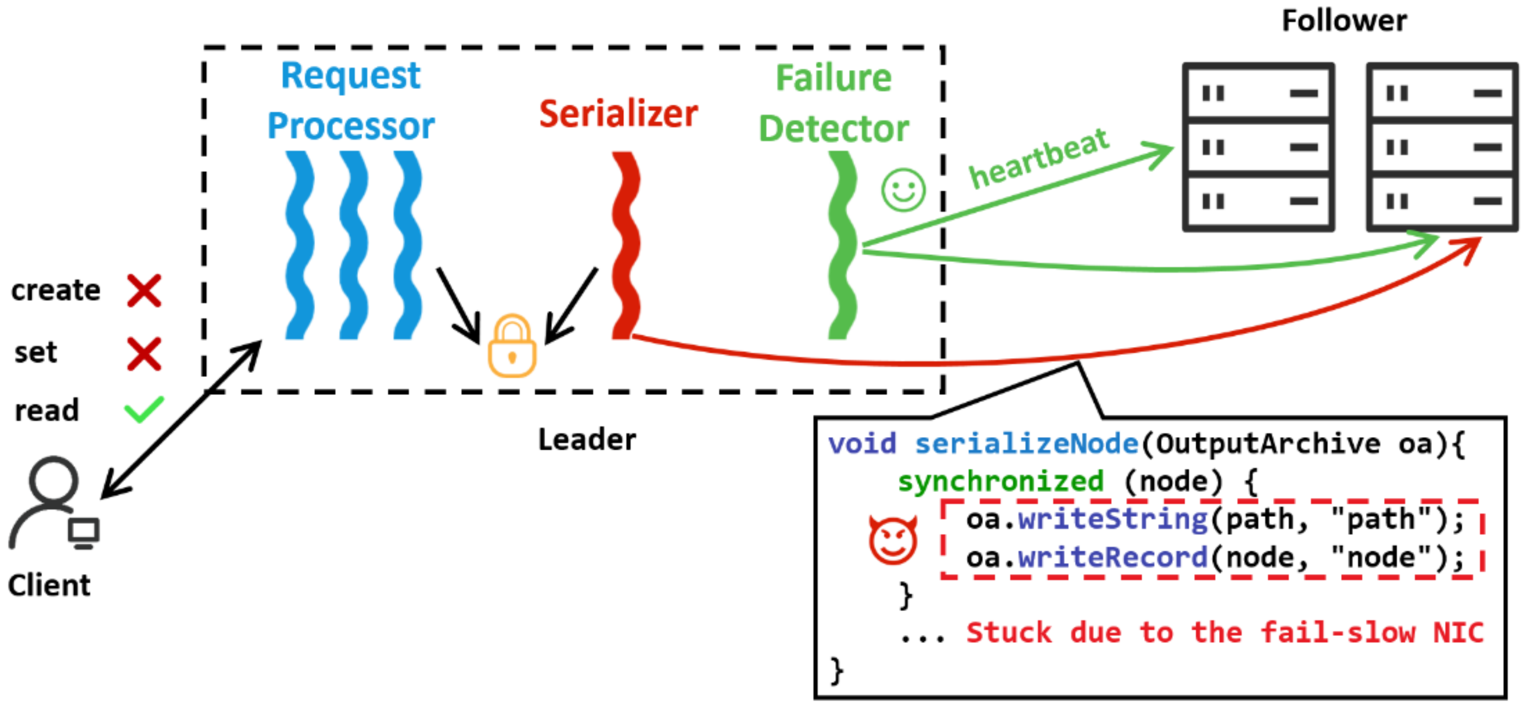
Annual fail-slow failure rate is 1-2%[3]

[1] Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems, Guanwai et al.
[2] NVMe SSD Failures in the Field: the Fail-Stop and the Fail-Slow, Lu et al.
[3] IASO: A Fail-Slow Detection and Mitigation Framework for Distributed Storage Services, Panda et al.

A Real Bug in ZooKeeper

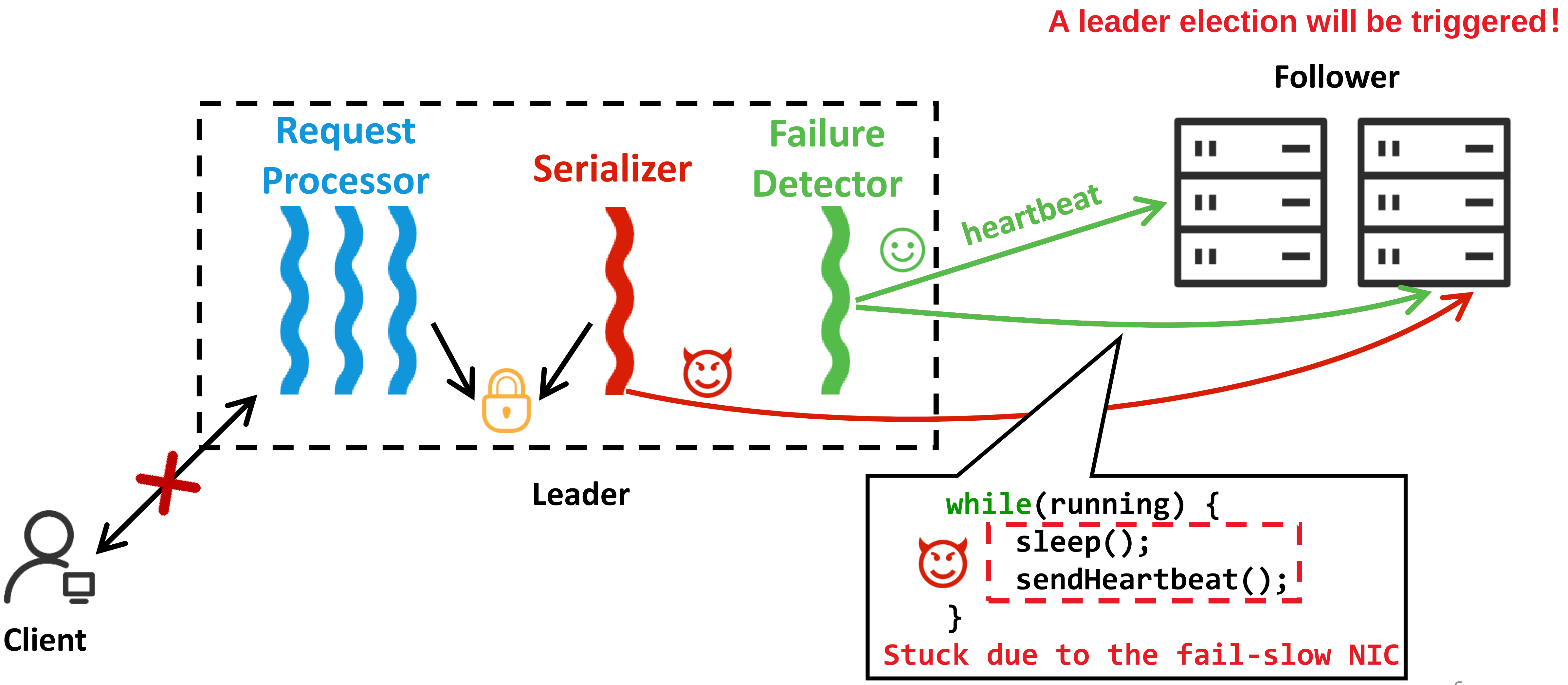


A Real Bug in ZooKeeper

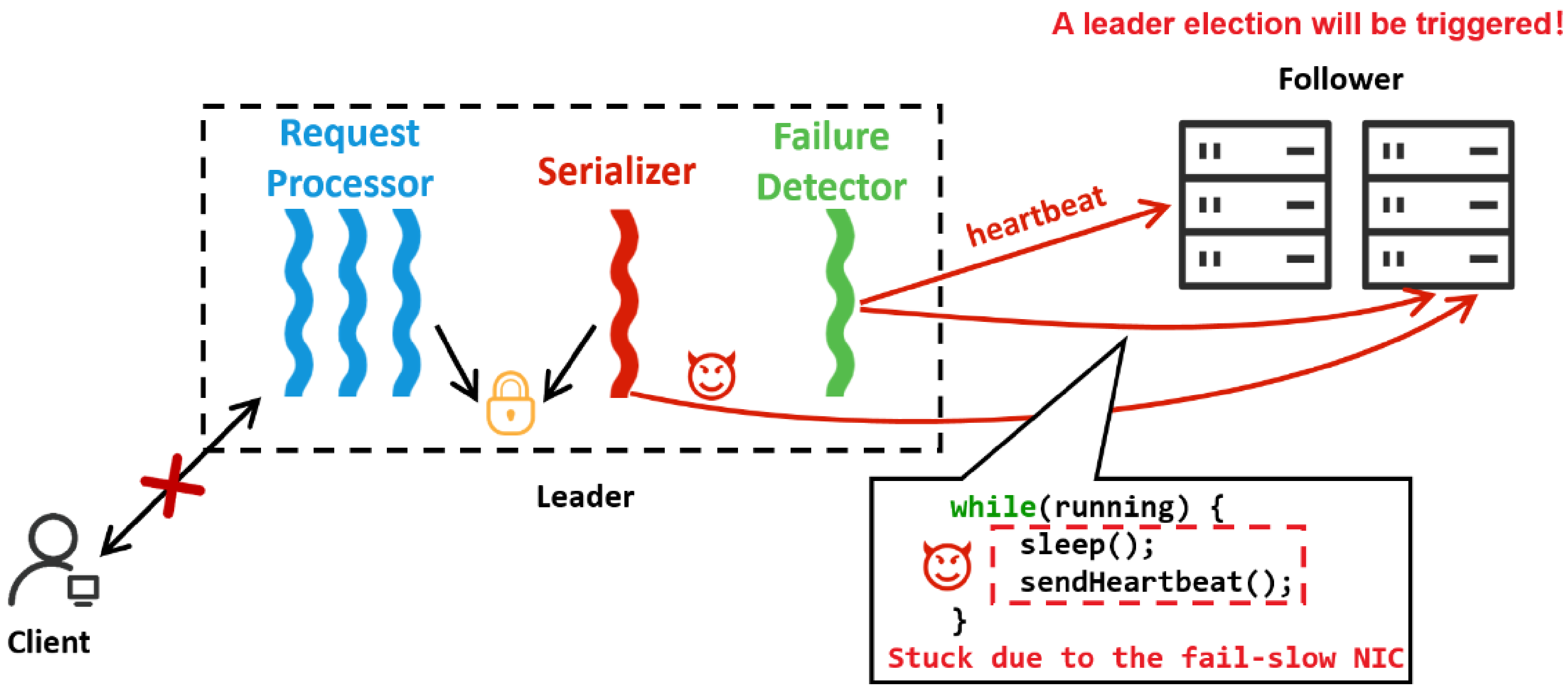


We define **fail-slow hardware failures (FSH failure)** as software-level failures caused by fail-slow hardware.

A Real Bug in ZooKeeper



A Real Bug in ZooKeeper



The fine granularity of fail-slow hardware is necessary to trigger FSH failures (a subset of I/O operations)

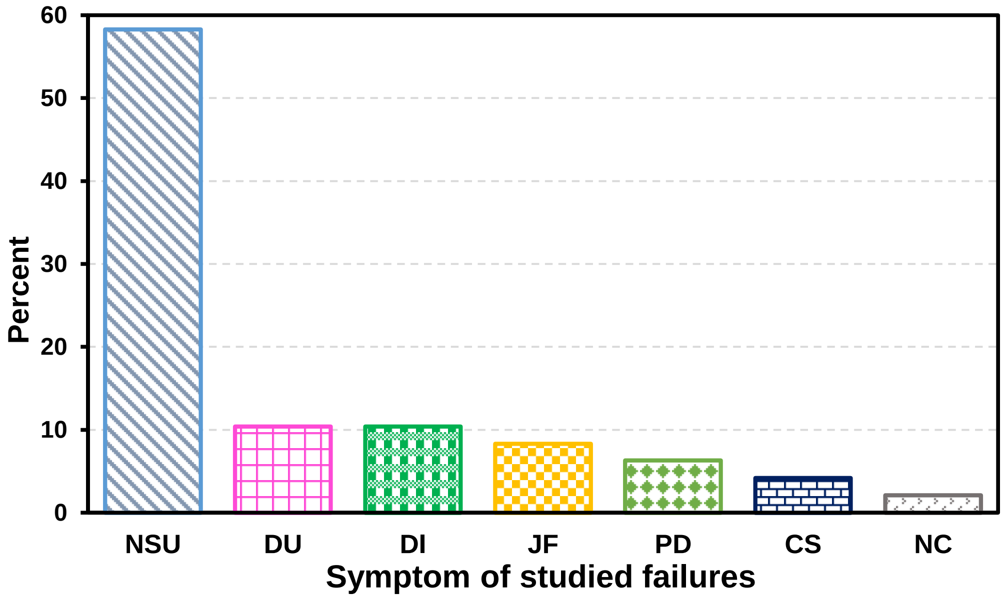
Study Methodology

- We study 48 FSH failure cases from five large, widely-used cloud systems.
  - Diverse services
    - Coordination service, file system, data-analytic framework, and database

Systems	Cases	Versions	Date
ZooKeeper	11	16	2009/05/27-2023/10/13
HDFS	18	25	2012/07/02-2022/09/07
HBase	10	18	2014/03/24-2023/12/16
MapReduce	4	3	2010/05/20-2022/05/22
Cassandra	5	7	2010/08/26-2020/12/09

Understanding FSH failures

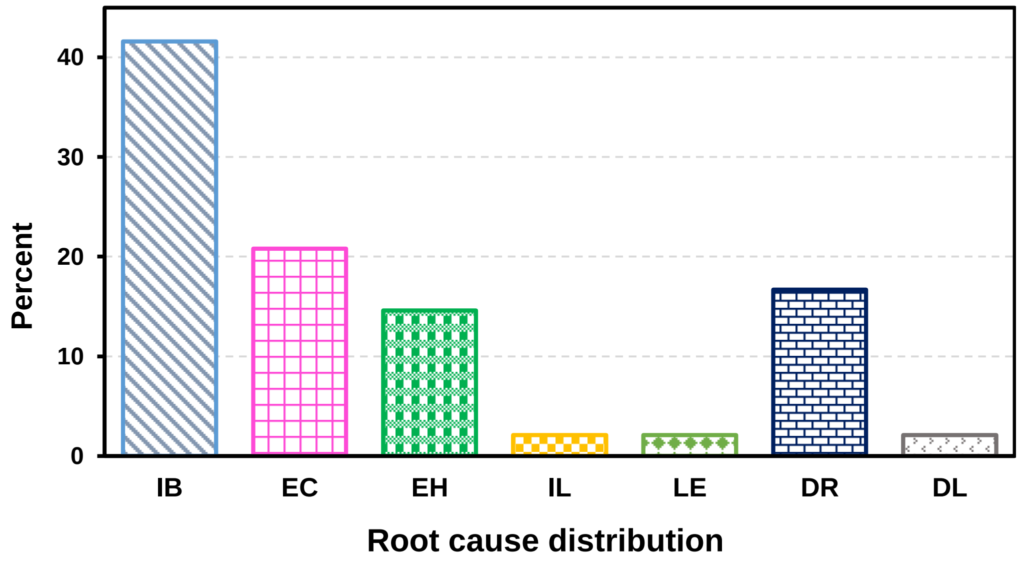
- Finding 1: over half (58.3%) of FSH failures cause node service to be unavailable.
- Finding 2: 20.8% of FSH failures are silent (including data unavailability and inconsistency).



NSU: node service unavailable; DU: data unavailability; DI: data inconsistency; JF: job failure
PD: performance degradation; CS: client stuck; NC: node crash

Understanding FSH failures

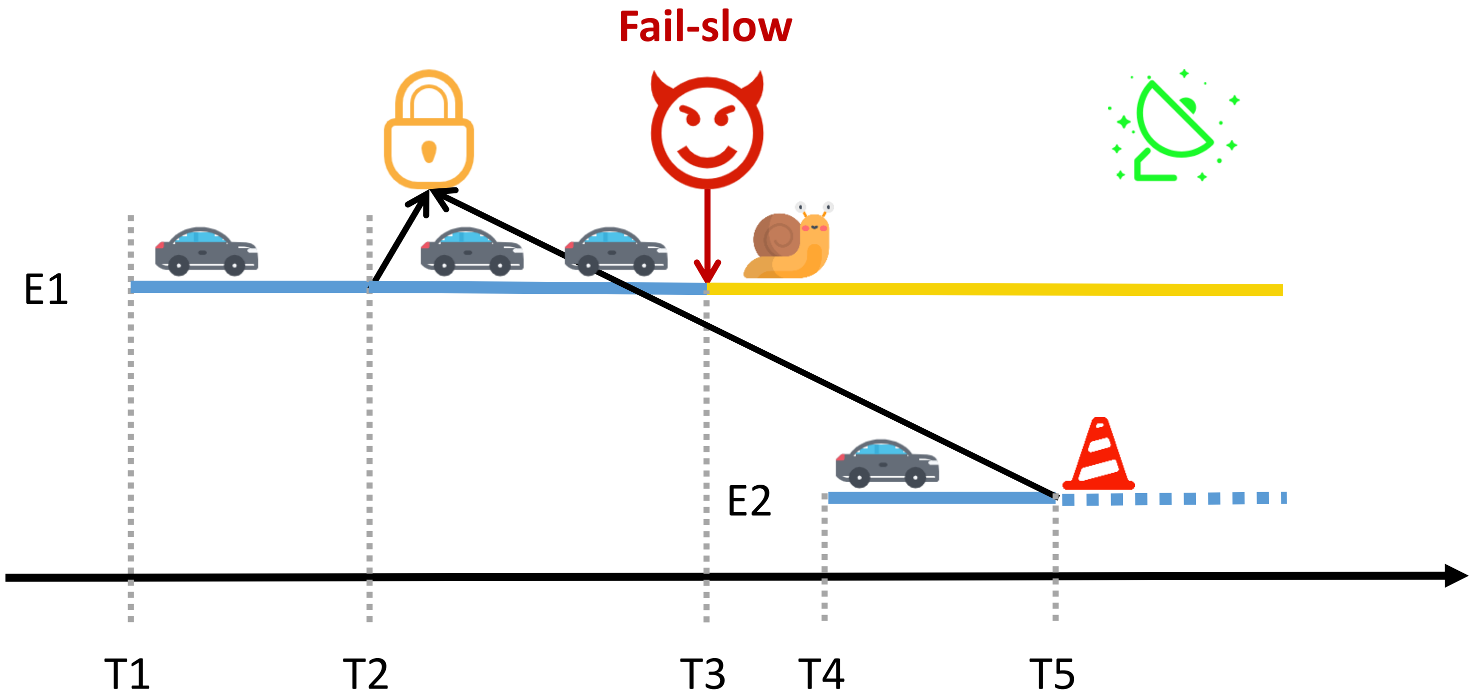
- Finding 3
  - Root causes are diverse.
  - The top three (total 93.7%) root causes are indefinite blocking, buggy internal checker, and data race.



Synchronized and timeout mechanisms are vulnerable.

Understanding FSH failures

- Synchronized mechanisms are vulnerable
- Fine granularity of fail-slow hardware is necessary



How to Deal with FSH failures

- Existing in-production detectors
  - Panorama[OSDI'18]
  - IASO[ATC'19]
  - OmegaGen[NSDI'20]
  - PERSEUS[FAST'23]

FSH failures already cause damages!

- Existing fault injection tools
  - FATE[NSDI'11]
  - CrashTuner[SOSP'19]
  - Legolas[NSDI'24]
  - Chronos[S&P'24]

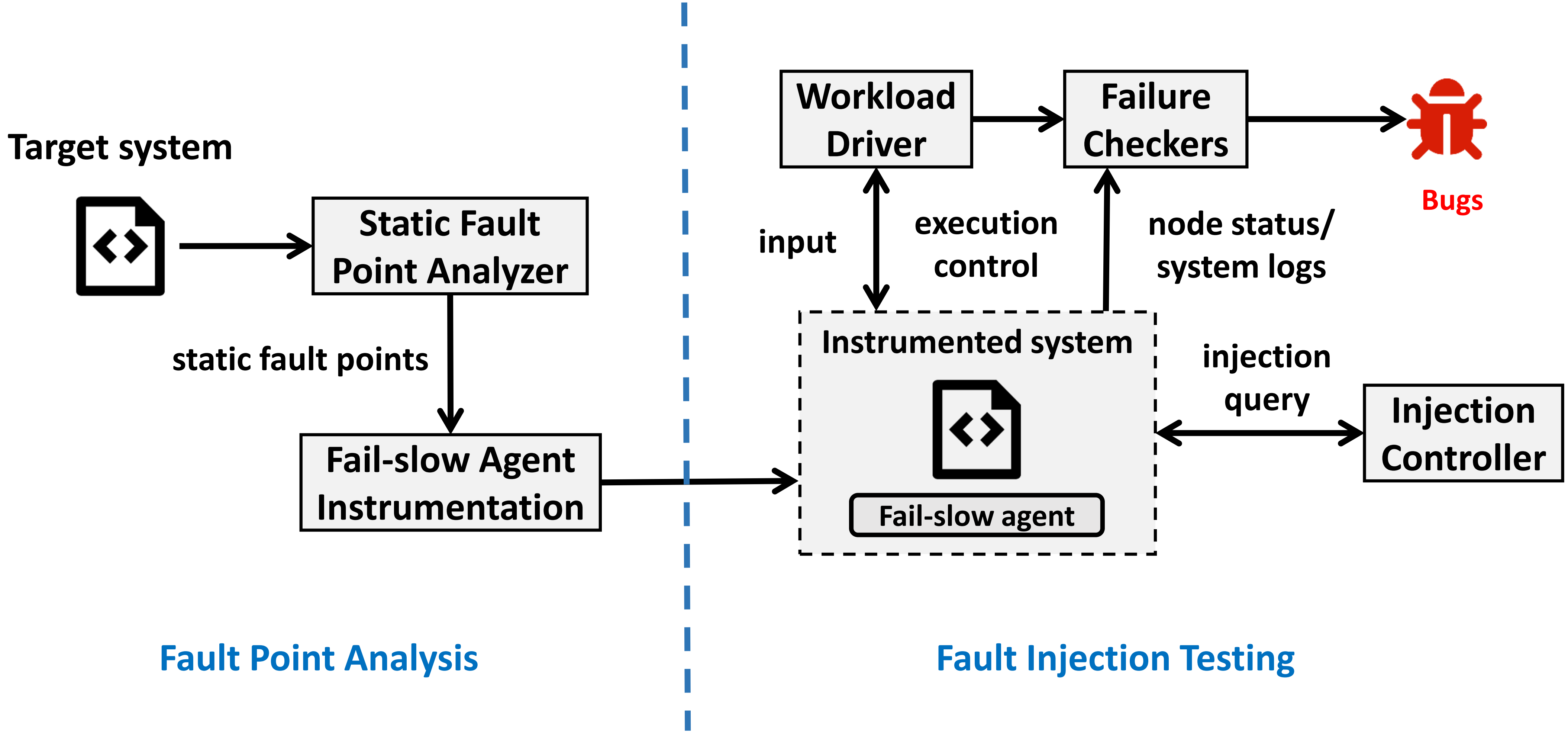
Overlooking characteristics of FSH failures!

Our Solution: Sieve

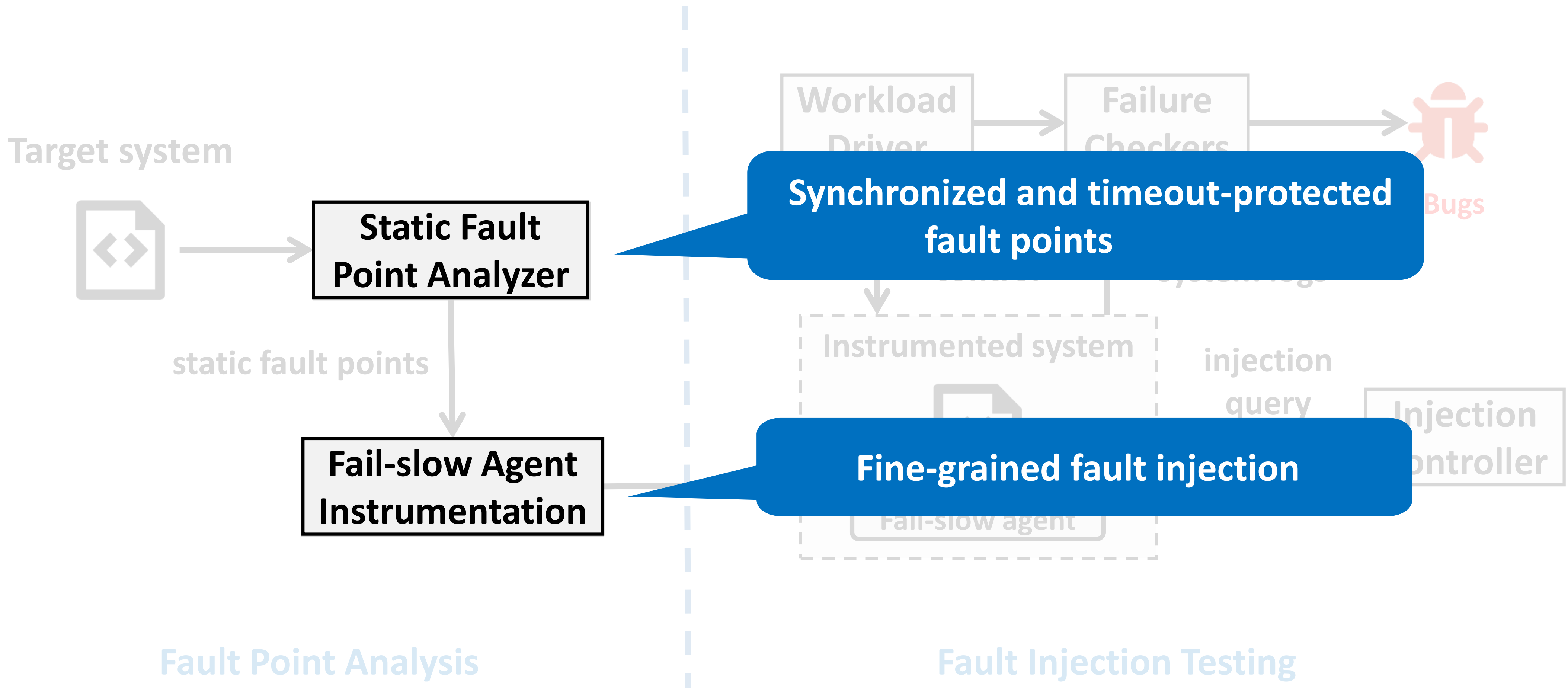
A fault injection testing framework for cloud systems to detect FSH failures

- Efficiently explore the large fault injection space
  - Statically analyze synchronized and timeout-protected fault points
- Enable fine-grained fault injection
  - Automatically instrument hooks to precisely simulate fail-slow hardware within a system

Sieve Workflow



Sieve Workflow

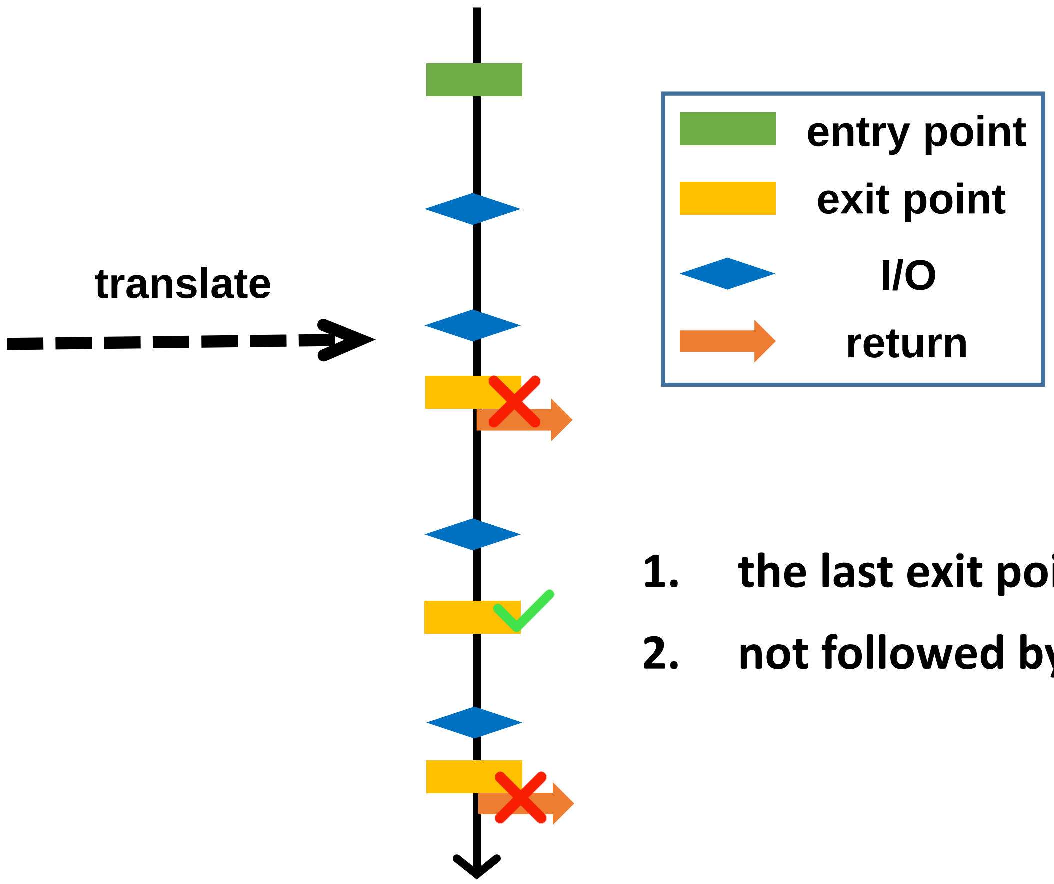


Static Fault Point Analyzer

- Identify synchronized fault points

```
if(cond1){
    synchronized(…){
        if(cond2){
            I/O1;
        }else{
            I/O2;
            return …;
        }
        I/O3;
    }
}else{…;}
I/O4;
return …;
```

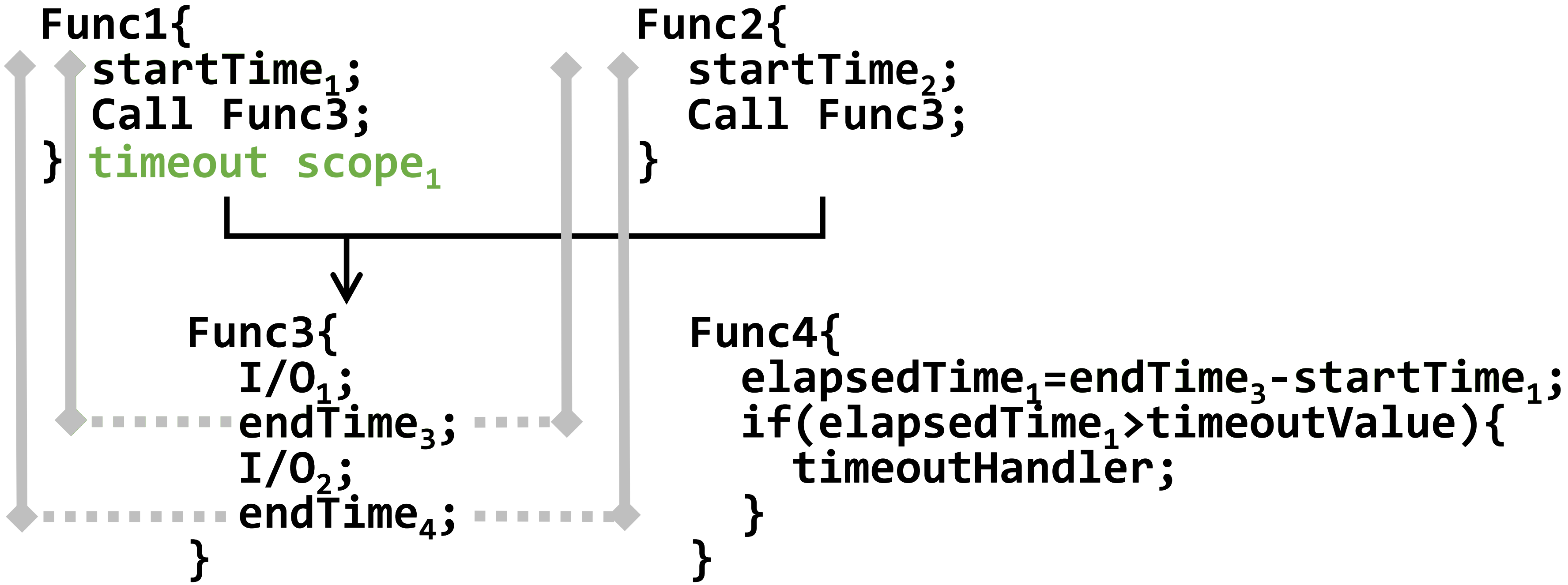
Critical region



1. the last exit point
2. not followed by return

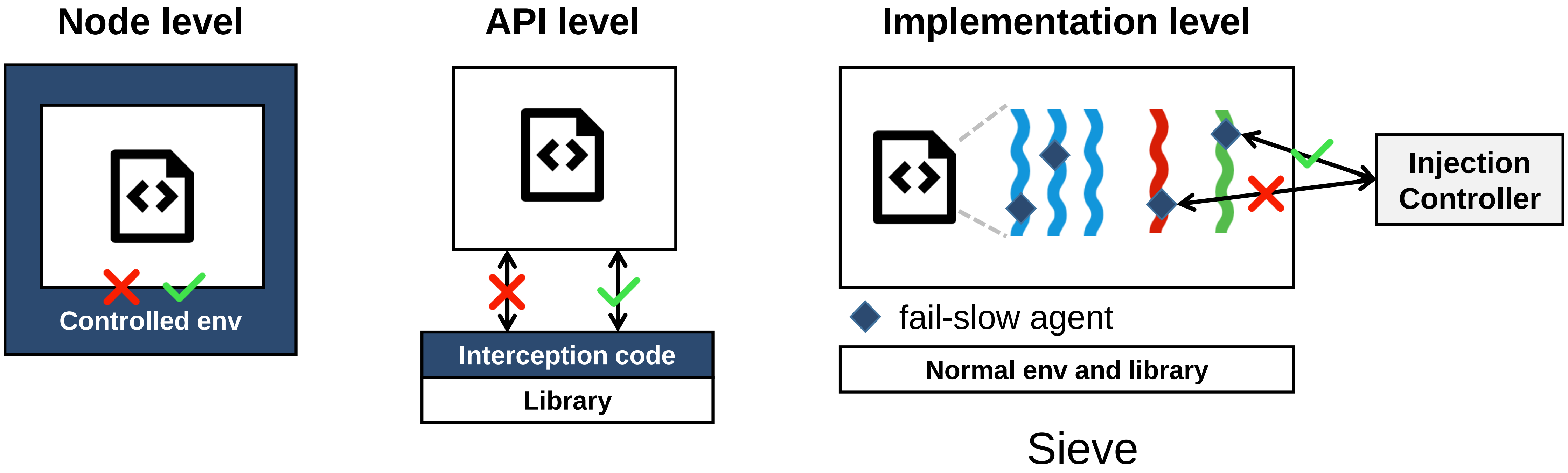
Static Fault Point Analyzer

- Identify timeout-protected fault points



Fail-Slow Agent Instrumentation

- Coarse-grained vs. Fine-grained



Evaluation

- Applied Sieve to three cloud systems
  - ZooKeeper, HDFS, Kafka
- Can Sieve effectively find new bugs?
  - Detected six new bugs, two of which are confirmed

Bug ID	Failure Symptoms	Status
ZK-4816	A follower cannot follow the leader for a long time	Pending
ZK-4817	CancelledKeyException cannot catch the client disconnection exception	Pending
ZK-4844	Fail-slow disk while executing writeLongToFile causes the follower to hang	Pending
ZK-4836	Inconsistent ACL index leads to MarshallingError	Confirmed
KA-16401	One request consumes all request handler threads	Pending
KA-16412	An uncreated topic is considered as a created one	Confirmed

More Details

- More bug study details
- Fault injection strategies
- Bug explanation
- ……

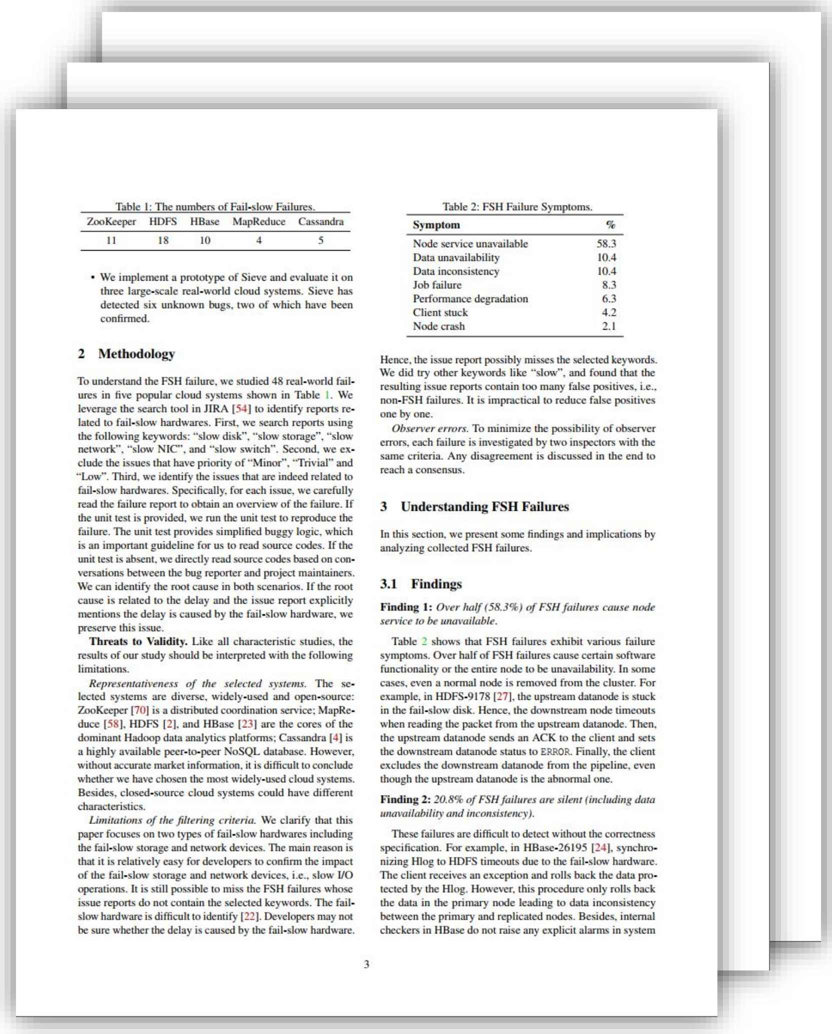


Table 1: The numbers of Fail-slow Failures.

ZooKeeper	HDFS	HBase	MapReduce	Cassandra
11	18	10	4	5

- We implement a prototype of Sieve and evaluate it on three large-scale real-world cloud systems. Sieve has detected six unknown bugs, two of which have been confirmed.

2 Methodology

To understand the FSH failure, we studied 48 real-world failures in five popular cloud systems shown in Table 1. We leverage the search tool in JIRA [54] to identify reports related to fail-slow hardwares. First, we search reports using the following keywords: "slow disk", "slow storage", "slow network", "slow NIC", and "slow switch". Second, we exclude the issues that have priority of "Minor", "Trivial" and "Low". Third, we identify the issues that are indeed related to fail-slow hardwares. Specifically, for each issue, we carefully read the failure report to obtain an overview of the failure. If the unit test is provided, we run the unit test to reproduce the failure. The unit test provides simplified buggy logic, which is an important guideline for us to read source codes. If the unit test is absent, we directly read source codes based on conversations between the bug reporter and project maintainers. We can identify the root cause in both scenarios. If the root cause is related to the delay and the issue report explicitly mentions the delay is caused by the fail-slow hardware, we preserve this issue.

Threats to Validity. Like all characteristic studies, the results of our study should be interpreted with the following limitations.

Representativeness of the selected systems. The selected systems are diverse, widely-used and open-source: ZooKeeper [70] is a distributed coordination service; MapReduce [58], HDFS [2], and HBase [23] are the cores of the dominant Hadoop data analytics platforms; Cassandra [4] is a highly available peer-to-peer NoSQL database. However, without accurate market information, it is difficult to conclude whether we have chosen the most widely-used cloud systems. Besides, closed-source cloud systems could have different characteristics.

Limitations of the filtering criteria. We clarify that this paper focuses on two types of fail-slow hardwares including the fail-slow storage and network devices. The main reason is that it is relatively easy for developers to confirm the impact of the fail-slow storage and network devices, i.e., slow I/O operations. It is still possible to miss the FSH failures whose issue reports do not contain the selected keywords. The fail-slow hardware is difficult to identify [22]. Developers may not be sure whether the delay is caused by the fail-slow hardware.

Table 2: FSH Failure Symptoms.

Symptom	%
Node service unavailable	58.3
Data unavailability	10.4
Data inconsistency	10.4
Job failure	8.3
Performance degradation	6.3
Client stuck	4.2
Node crash	2.1

Hence, the issue report possibly misses the selected keywords. We did try other keywords like "slow", and found that the resulting issue reports contain too many false positives, i.e., non-FSH failures. It is impractical to reduce false positives one by one.

Observer errors. To minimize the possibility of observer errors, each failure is investigated by two inspectors with the same criteria. Any disagreement is discussed in the end to reach a consensus.

3 Understanding FSH Failures

In this section, we present some findings and implications by analyzing collected FSH failures.

3.1 Findings

Finding 1: *Over half (58.3%) of FSH failures cause node service to be unavailable.*

Table 2 shows that FSH failures exhibit various failure symptoms. Over half of FSH failures cause certain software functionality or the entire node to be unavailability. In some cases, even a normal node is removed from the cluster. For example, in HDFS-9178 [27], the upstream datanode is stuck in the fail-slow disk. Hence, the downstream node timeouts when reading the packet from the upstream datanode. Then, the upstream datanode sends an ACK to the client and sets the downstream datanode status to `ERROR`. Finally, the client excludes the downstream datanode from the pipeline, even though the upstream datanode is the abnormal one.

Finding 2: *20.8% of FSH failures are silent (including data unavailability and inconsistency).*

These failures are difficult to detect without the correctness specification. For example, in HBase-26195 [24], synchronizing Hlog to HDFS timeouts due to the fail-slow hardware. The client receives an exception and rolls back the data protected by the Hlog. However, this procedure only rolls back the data in the primary node leading to data inconsistency between the primary and replicated nodes. Besides, internal checkers in HBase do not raise any explicit alarms in system

3

Conclusion

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

- Fail-slow hardware causes severe damages in cloud systems
  - Existing fault injection testing is inefficient
- We conduct a study on 48 FSH failure cases
- Sieve: a fault injection testing framework to detect FSH failure bugs
  - Identify synchronized and timeout-protected fault points
  - Enable fine-grained fault injection
- Found six bugs, two of which are confirmed
- Open Sourced at https://github.com/RabbitDong-on

Thank you! Q&A