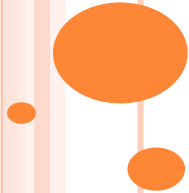


Blackbox Fuzzing of Distributed Systems with Multi-Dimensional Inputs and Symmetry-Based Feedback Pruning

Yonghao Zou^{1 2}, Jia-Ju Bai¹, Zu-Ming Jiang³, Ming Zhao⁴, Diyu Zhou²

¹Beihang University, ²Peking University

³ETH Zurich, ⁴Arizona State University



ETH zürich



Background

- Distributed systems

- Distributed systems are critical to modern infrastructures
- Subtle bugs can cause significant economic losses

- Different applications

- Distributed databases
 - ClickHouse, RethinkDB
- Distributed coordination systems
 - ZooKeeper, etcd

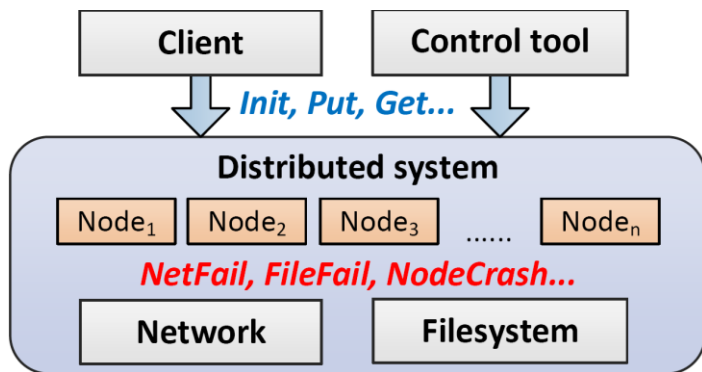


Background

- Distributed systems common features

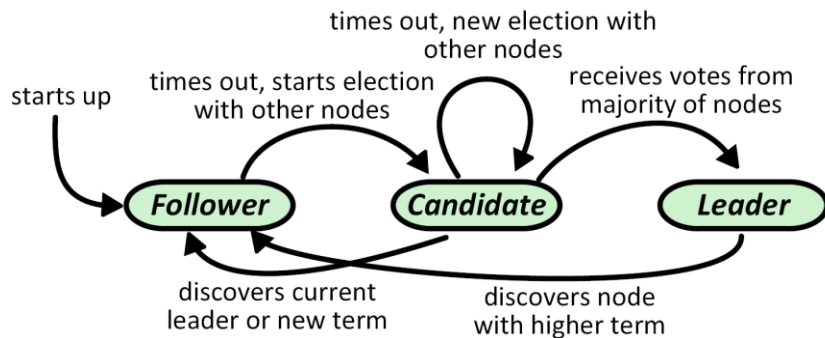
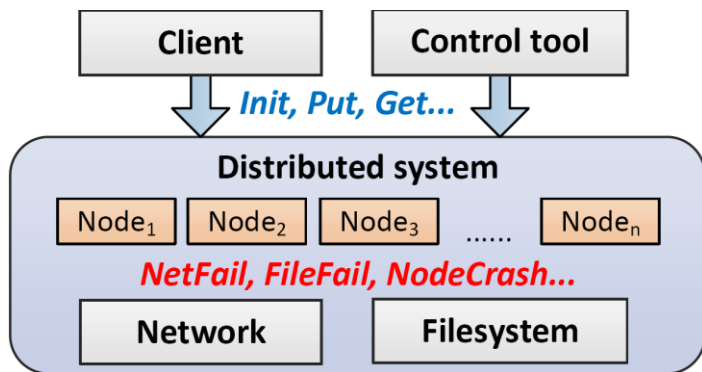
Background

- Distributed systems common features
 - Various kinds of input events: regular and fault events



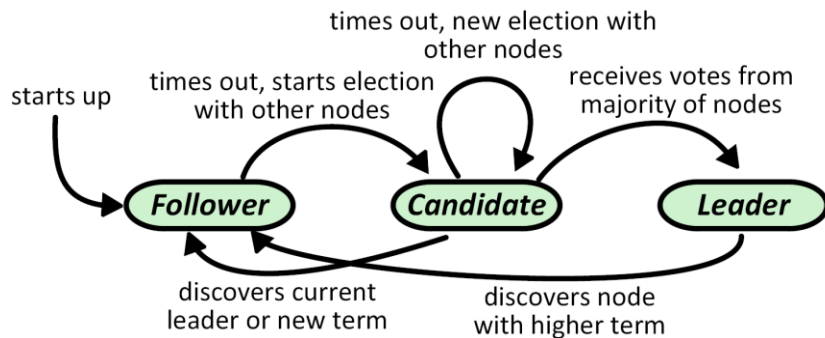
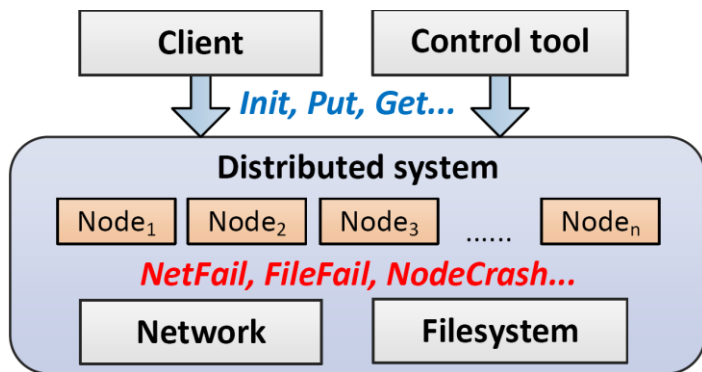
Background

- Distributed systems common features
 - Various kinds of input events: regular and fault events
 - Network messages capture important state changes



Background

- Distributed systems common features
 - Various kinds of input events: regular and fault events
 - Network messages capture important state changes
 - Timing-dependence between events



Background

- Existing fuzzing tools
 - Limited feedback metrics
 - Inefficient mutation space

Tools	Mutation space	Feedback
Jepsen	Fault	None
CrashFuzz	Fault	Code coverage
Mallory	Fault	Annotations + Messages (Similarity pruning)

Background

- Existing fuzzing tools
 - Limited feedback metrics
 - Inefficient mutation space

Tools	Mutation space	Feedback
Jepsen	Fault	None
CrashFuzz	Fault	Code coverage
Mallory	Fault	Annotations + Messages (Similarity pruning)
DistFuzz	Regular, fault and timing	Messages (Symmetry pruning)

Key Contributions

- **C1: Mutation space extension**

- Extending mutation space with **regular events and timing intervals**

- **C2: Effective fuzzing feedback**

- Proposing **network message sequences with symmetry-based pruning** as fuzzing feedback

- **C3: Practical realization**

- Building **DistFuzz**, a novel fuzzing framework that finds **52 real bugs** across **10 systems**, with 28 confirmed and **4 CVEs**

Mutation Space Extension

- Regular events
 - Client requests: *Get*, *Put*, *CreateDB* operations ...
 - Management commands: *NodeStart*, *NodeStop*, *StatCheck* ...
- Timing intervals
 - Relative timing intervals between events

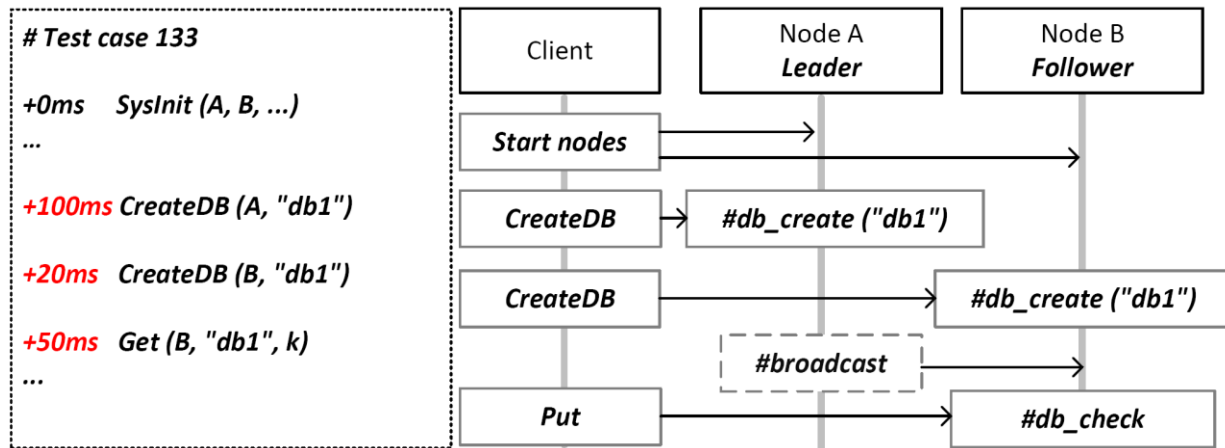
Mutation Space Extension

Regular events

- Client requests: *Get*, *Put*, *CreateDB* operations ...
- Management commands: *NodeStart*, *NodeStop*, *StatCheck* ...

Timing intervals

- Relative timing intervals between events

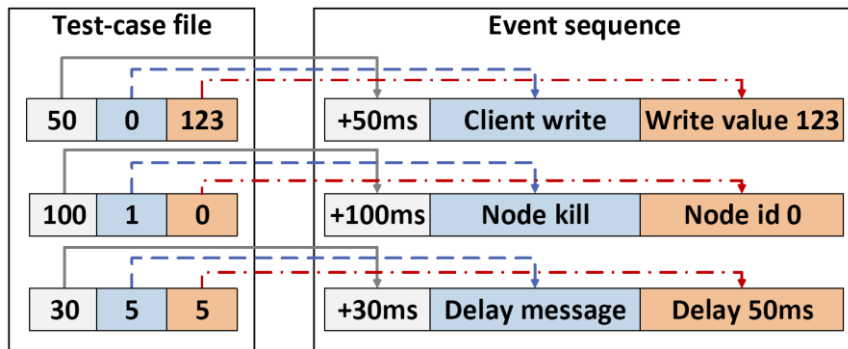


Mutation Space Extension

- DistFuzz systematically co-mutates
 - Regular events
 - Fault events
 - Timing intervals
- Representation
 - event as 3-tuple $\langle \textit{timing_interval}, \textit{event_type}, \textit{parameters} \rangle$

Mutation Space Extension

- DistFuzz systematically co-mutates
 - Regular events
 - Fault events
 - Timing intervals
- Representation
 - event as 3-tuple $\langle \textit{timing_interval}, \textit{event_type}, \textit{parameters} \rangle$



Effective fuzzing feedback

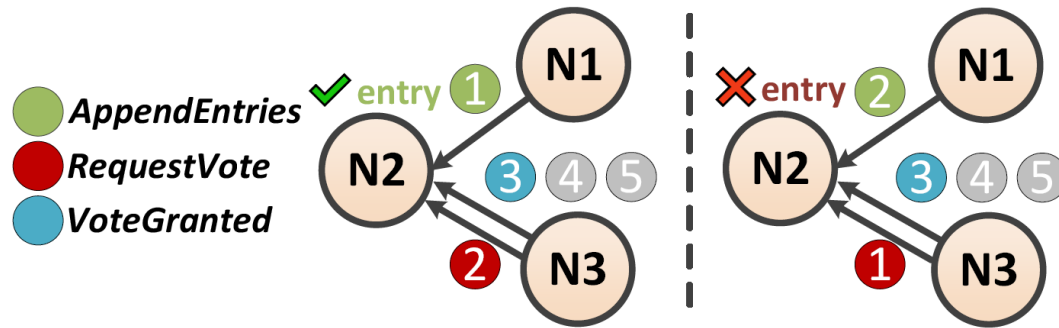
- Network messages as feedback
 - Sufficient: no need for user annotations
 - Network messages capture important state changes
 - Redundant: pruning is required
 - Similarity-based pruning is ad-hoc and ill-suited

Effective fuzzing feedback

- Network messages as feedback
 - Similarity-based pruning
 - Misses interesting states

Effective fuzzing feedback

- Network messages as feedback
 - Similarity-based pruning
 - Misses interesting states



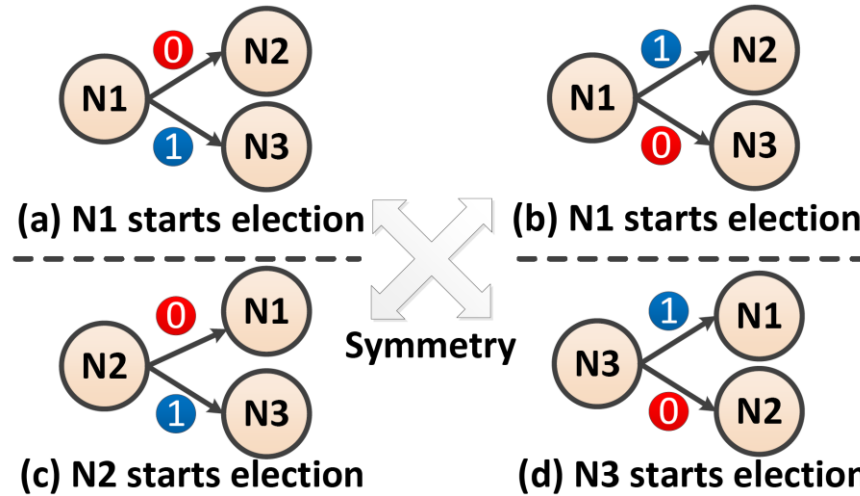
Different states output similar network messages

Effective fuzzing feedback

- Network messages as feedback
 - Similarity-based pruning
 - Explores redundant states due to symmetry

Effective fuzzing feedback

- Network messages as feedback
 - Similarity-based pruning
 - Explores redundant states due to symmetry



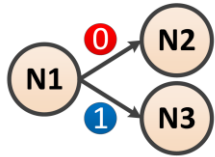
Symmetry of the states in a distributed system

Effective fuzzing feedback

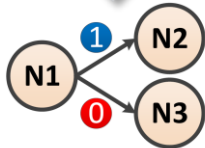
- Network messages as feedback
 - Symmetry-based pruning
 - Order symmetry

Effective fuzzing feedback

- Network messages as feedback
 - Symmetry-based pruning
 - Order symmetry



(a) N1 starts election

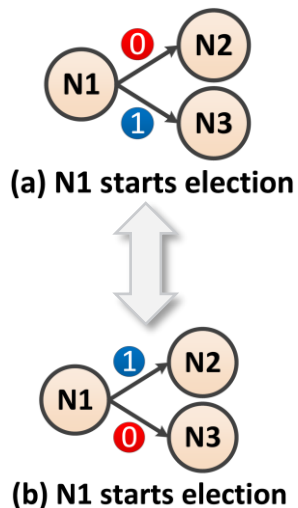


(b) N1 starts election

Effective fuzzing feedback

- Network messages as feedback

- Symmetry-based pruning
 - Order symmetry



Global	Type	SID	RID	Content
0	Send	1	2	Elect
1	Send	1	3	Elect
2	Recv	1	2	Elect
3	Recv	1	3	Elect
4	Send	2	1	Vote
5	Send	3	1	Vote
6	Recv	2	1	Vote
7	Recv	3	1	Vote

Optimize:
Global to
Local

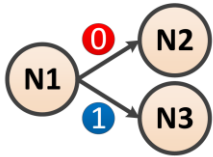
NID	Local	Type	SID	RID	Content
1	0	Send	1	2	Elect
1	1	Send	1	3	Elect
1	2	Recv	2	1	Vote
1	3	Recv	3	1	Vote
2	0	Recv	1	2	Elect
2	1	Send	2	1	Vote
3	0	Recv	1	3	Elect
3	1	Send	3	1	Vote

Effective fuzzing feedback

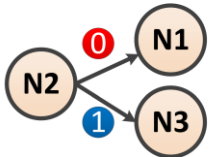
- Network messages as feedback
 - Symmetry-based pruning
 - Role symmetry

Effective fuzzing feedback

- Network messages as feedback
 - Symmetry-based pruning
 - Role symmetry



(a) N1 starts election

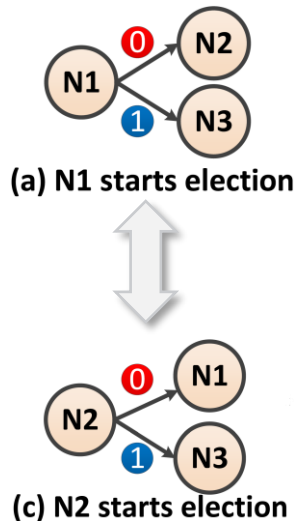


(c) N2 starts election

Effective fuzzing feedback

- Network messages as feedback

- Symmetry-based pruning
 - Role symmetry



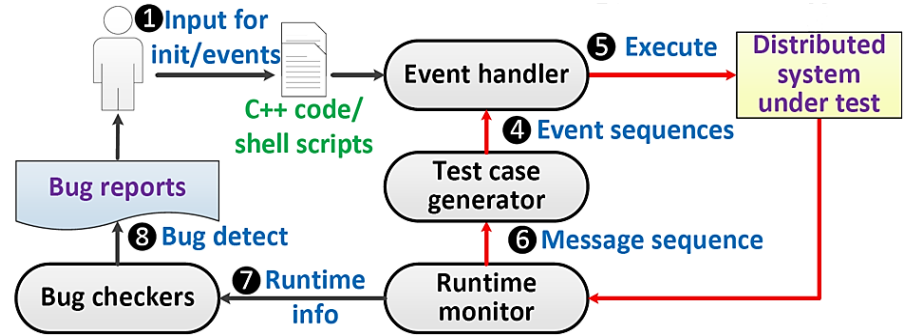
NID	Local	Type	SID	RID	Content
1	0	Send	1	2	Elect
1	1	Send	1	3	Elect
1	2	Recv	2	1	Vote
1	3	Recv	3	1	Vote
2	0	Recv	1	2	Elect
2	1	Send	2	1	Vote
3	0	Recv	1	3	Elect
3	1	Send	3	1	Vote

Optimize:
Remove ID,
Content to
Length

Local	Type	Length
0	Send	10
1	Send	10
2	Recv	4
3	Recv	4
0	Recv	10
1	Send	4
0	Recv	10
1	Send	4

Framework

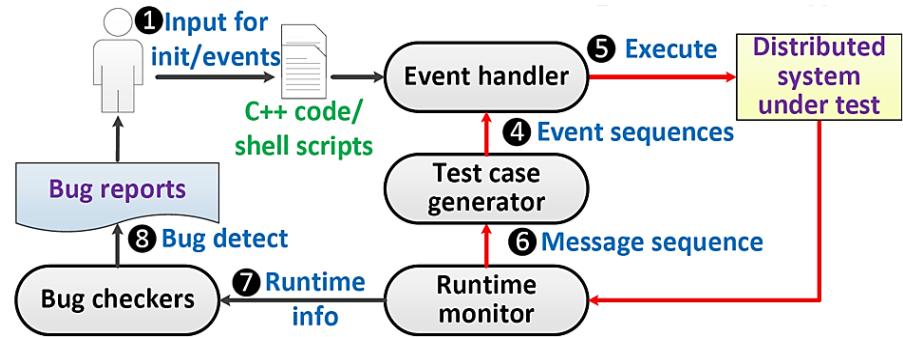
○ Main Workflow



Framework

○ Main Workflow

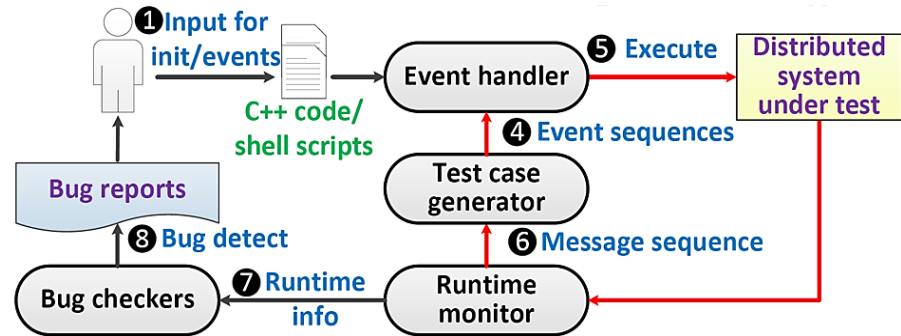
- User configuration
 - System initialization
 - Specific events



Framework

○ Main Workflow

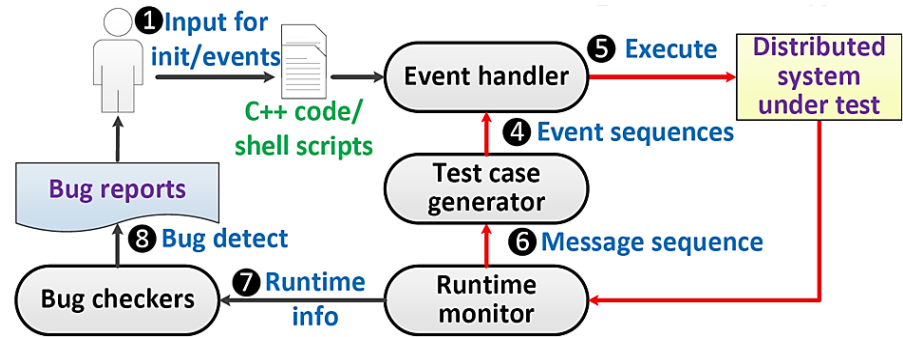
- User configuration
 - System initialization
 - Specific events
- Input Generation
 - Event sequences
 - Execute



Framework

○ Main Workflow

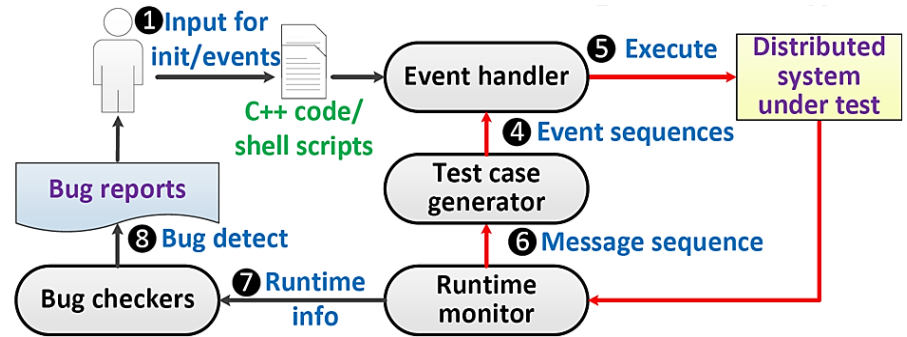
- User configuration
 - System initialization
 - Specific events
- Input Generation
 - Event sequences
 - Execute
- Feedback collection
 - Network messages
 - Runtime info



Framework

○ Main Workflow

- User configuration
 - System initialization
 - Specific events
- Input Generation
 - Event sequences
 - Execute
- Feedback collection
 - Network messages
 - Runtime info
- Bug detection



Framework

- Implementation – Events

Framework

- Implementation – Events

- Regular events: *SysInit*, *Get*, *Put*, ...
 - Need user input

Framework

○ Implementation – Events

- Regular events: *SysInit, Get, Put, ...*
 - Need user input
- Fault events: *NetFail, NetDelay, NodeRestart, ...*
 - Universal
 - Based on system call interception (strace)

Framework

- Implementation – Checkers

Framework

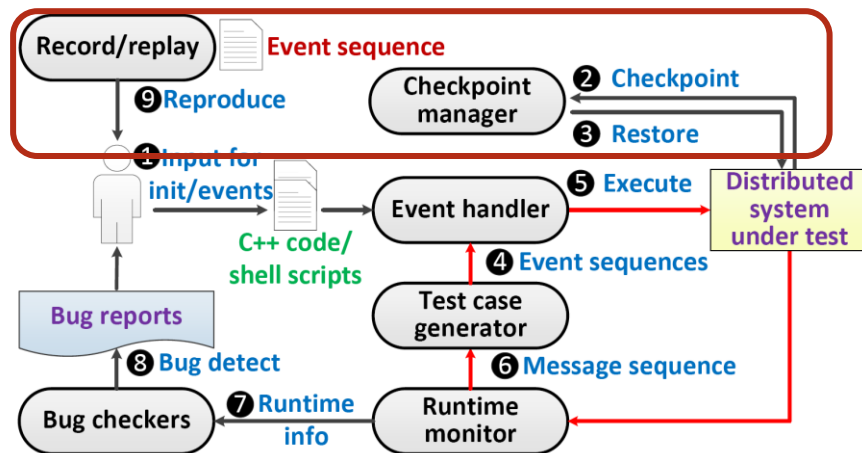
○ Implementation – Checkers

- Memory checker for low-level languages
- Linearizability checker
- Node crash checker
- Availability checker
- Log checker

Framework

Implementation

- Checkpoint
 - Accelerating the boot process
 - CRIU
- Reproduction
 - Offline bug reproduction
 - RR (record replay debugger)



Evaluation

Experimental setup

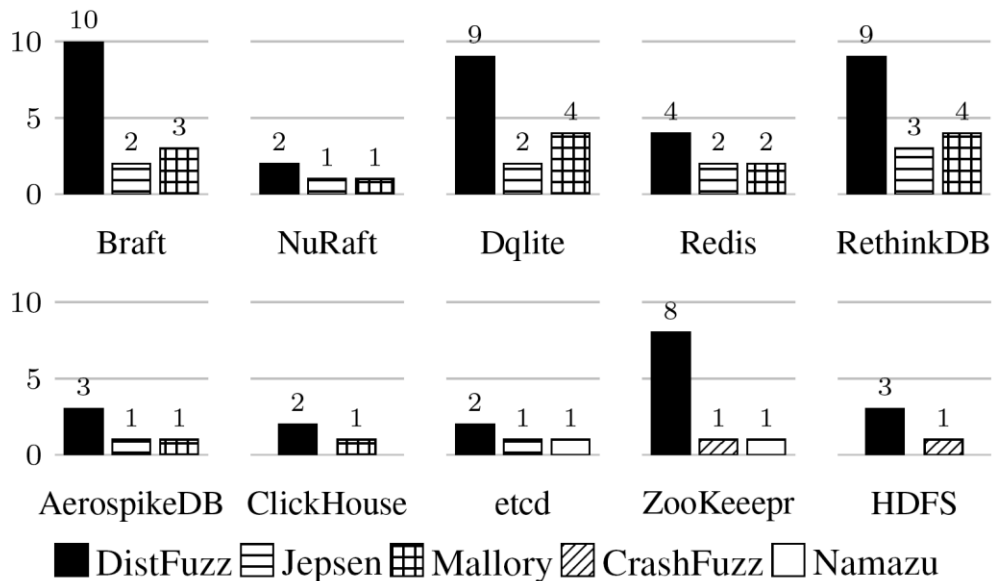
- 10 open-source and popular distributed systems
- C/C++, Go, Java

System	Description	Lang	Version
Braft	Raft implementation by Baidu	C++	commit 0c5a59
NuRaft	Raft implementation by eBay	C++	commit 5a7a40
Dqlite	Embeddable distributed DBMS	C	commit 37af7c
Redis	Distributed key-value Store	C	commit e18c38
RethinkDB	Distributed NoSQL DBMS	C++	v2.4.1
AerospikeDB	Distributed NoSQL DBMS	C	v5.6.0.4
ClickHouse	Distributed DBMS	C++	v21.9.2.17
etcd	Distributed key-value store	Go	v2.2.0
ZooKeeper	Distributed coordination system	Java	v3.5.1
HDFS	Distributed file system	Java	v3.2.4

Evaluation

Found bugs

- 52 bugs in total
- 28 confirmed
- 4 CVEs



Thanks

Open source: github.com/zouyonghao/DistFuzz

E-mail: zouyonghao@live.cn

