

# Chaos Practice in TiDB

TangLiu @ PingCAP

## About Me

- Chief Engineer of PingCAP
- Open sources: go-mysql, go-mysql-elasticsearch, LedisDB, etc.

# Agenda

- Why Chaos
- How
- Schrodinger
- Miscellaneous

ACMUG & CRUG

Why?



## Error can happen anywhere, at anytime!!!

- Unit test is not enough
- Integration test is not enough
- Performance test is not enough

ACMUG&CRUG



How

# Chaos Engineering

1. Study the **stable status** of the system, usually through Monitor and Statistics
2. Inject a fault to disturb the system
3. Study the **unstable status** of the system and analyze the results
4. Repeat step 1 to 3
5. Automate

ACMUG&CRUG



IT大咖说  
知识共享平台

# Fault Injection



# Common

- kill
- kill -9
- SIGSTOP, SIGCONT
- renice
- Maxing out CPU

# Network

- Drop
- Reject
- Delay
- Reorder
- Full

# Disk

- libfuse
- libfiu
- rm -rf
- echo 0 > file
- NoSpace

# Kernel Fault Injection

Need rebuild kernel with fault injection support

```
echo 1 > /sys/block/vdb/vdb1/make-it-fail
mount debugfs /debug -t debugfs
cd /debug/fail_make_request
echo 10 > interval # interval
echo 100 > probability # 100% probability
echo -1 > times # how many times: -1 means no limit
```

> Buffer I/O error on device vdb1, logical block 32538624

> lost page write due to I/O error on vdb1

# Kernel Fault Injection

```
echo 1 > cache-filter
echo 1 > /sys/kernel/slab/ext4_inode_cache/failslab
echo N > ignore-gfp-wait
echo -1 > times
echo 100 > probability

> cp linux-3.10.1.tar.xz linux-3.10.1.tar.xz.6
> cp: cannot create regular file 'linux-3.10.1.tar.xz.6':
Cannot allocate memory
```

# Systemtap

```
probe vfs.read.return {  
    if (target() != pid()) next  
    udelay(300)  
}
```

```
probe vfs.write.return {  
    if (target() != pid()) next  
    udelay(300)  
}
```

# Systemtap

```
probe vfs.read.return {  
    if (target() != pid()) next  
    // Interrupted by a signal  
    $return = -4  
}
```

```
probe vfs.write.return {  
    if (target() != pid()) next  
    // No space  
    $return = -28  
}
```

# Fail

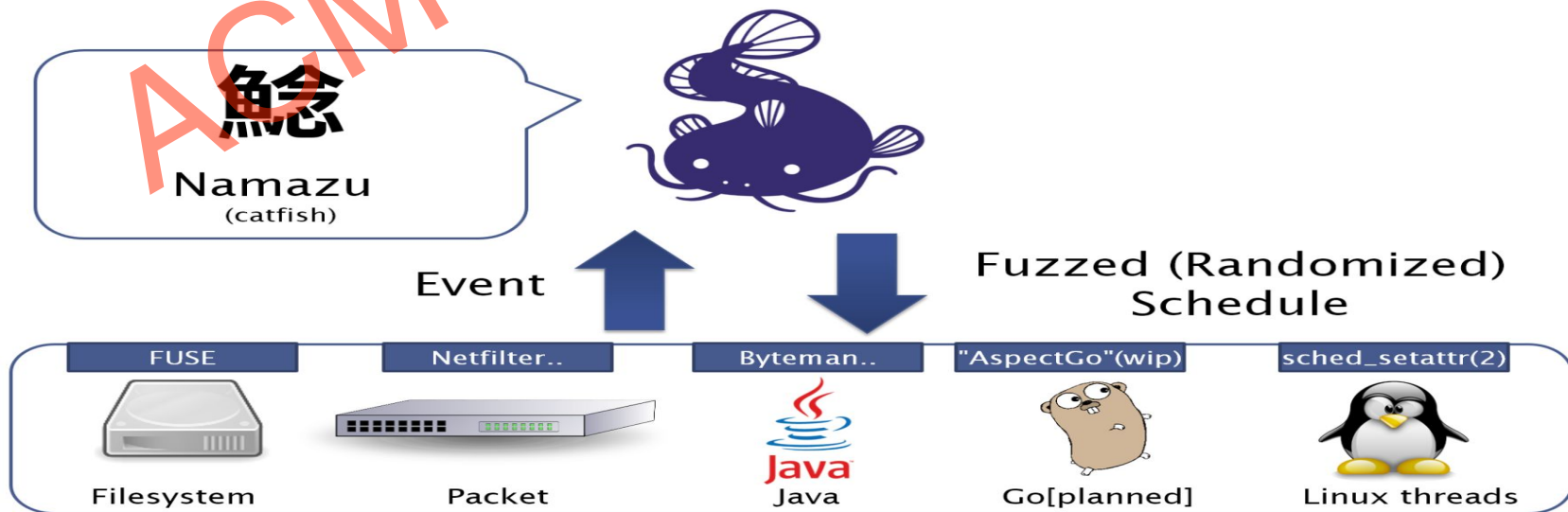
- Go fail - <https://github.com/coreos/gofail>
- Rust fail - <https://github.com/pingcap/fail-rs>

```
// Ingest a failure
fn save_snapshot() {
    save_data();
    fail_point!("snapshot");
    save_meta();
}
// Run and Trigger the failure panic
FAILPOINTS=snapshot=panic(msg) cargo run
```



# Namazu

- Filesystem Inspector - Use Fuse to delay or inject faults
- Ethernet Inspector - Use iptables to drop or delay packages
- Process Inspector - Change scheduler attributes



# Jepsen

- Run Nemesis to disturb a cluster
- Record the request and response
- Verify the linearizability of history

Focus on network and linearizability testing

ACMUG&CRUG

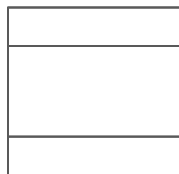


# Practice

ACMUG&CRUG



In Ancient Time...



Build Binary

Watch Results

Prepare Machines

Inject Faults

Deploy Cluster

Run Test Cases



ACMUG&CRUG

We need automation

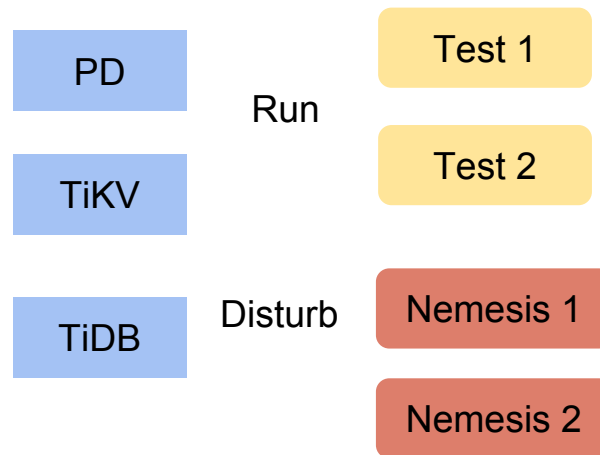
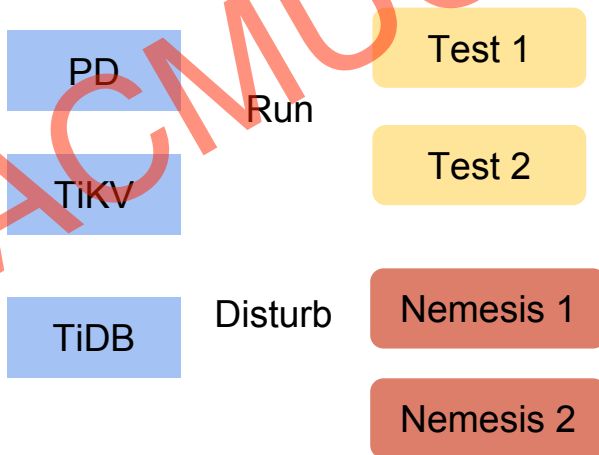
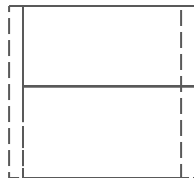
ACMUG&CRUG



Schrodinger



# kubernetes





# Schrodinger

TiDB Cluster Stability Test

Daily View

Mission

Scenes

Case Template

Cluster Template

History

Release

Count: 9

daily

weekly

monthly

yearly

Mission ID	Mission Name	Status	Scenes	Update Time
152048022511	raft-merge-2	ERROR	merge-with-nemesis	2018-03-08 15:23:25
15204908606	learn-test	RUNNING	learn-test	2018-03-08 15:18:42
15204865773	learn-test	ERROR	learn-test	2018-03-08 14:34:20

## Create Cluster Template



\* Name:

Chaos-Test

\* Creator:

tangliu

\* PD Size:

3

\* TiDB  
Size:

3

\* TiKV  
Size:

5

\* ConfigMap:

tidb-cluster-config-ten

Description:

For Chaos Test

Cancel

Reset

OK

Daily View

Mission

Scenes

Case Template

Cluster Template

History

Release

Count: 4

New

Name	Creator	PD Size	TiDB Size	TiKV Size	ConfigMap
Chaos-Test	tangliu	3	3	5	tidb-cluster-config-template tidb-cluster-learn-to

## Create Case Template



\* Name:

Chaos-Test

\* Type:

test case

\* Creator:

tangliu

\* Description:

For Chaos Test

### Source:

\* Binary Name:

bank

\* Source Type:

bin

Git Repo:

Git Value:

Binary URL:

http://pingcap-dev.hk.t

Args:

Cancel

Reset

OK

## Create New Scenes



Name: Chaos-Test

Creator: tangliu

Desc: For Chaos Test

**Boxes:** [Add Box](#)

Box0 [Remove](#)

Name: Chaos Test

Cluster  
Template: Chaos-Test ▼

Cases  
Template:

Chaos-Test X ▼

Lables:

Cancel

Reset

OK

## Create New Mission



\* Name:

Chaos Test

\* Scenes:

Chaos-Test

\* PD  
Version:

branch:master

\* TiDB  
Version:

branch:master

\* TiKV  
Version:

branch:master

\* Slack  
Channel:

alerts-for-test

\* Timeout:

168h

IgnoreErrors:

Cancel

Reset

OK

Daily View

Mission

Scenes

Case Template

Cluster Template

History

Release

Count: 10

daily

weekly

monthly

yearly

Mission ID	Mission Name	Status	Scenes	Update Time
152049681312	Chaos Test	NEW	Chaos-Test	2018-03-08 16:13:33

ACMUG&CRUG



## Miscellaneous



TLA+

<https://github.com/pingcap/tla-plus>

## Dr. TLA+ Series - learn an algorithm and protocol, study a specification

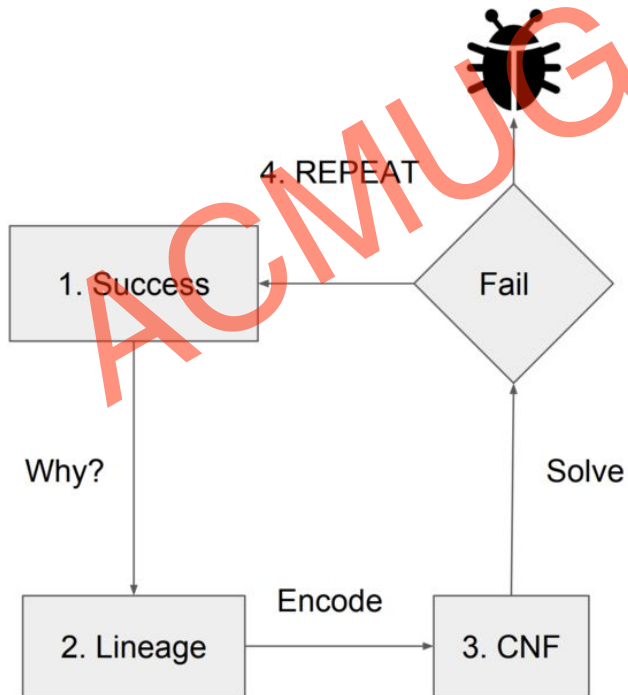
Lecture schedule (Time: 10-11:30am PDT)

	Date	Speaker	Topic	Media
	06.22.2016	Andrew Helwer	Paxos	video, slides
	07.21.2016	Jin Li	Raft	video, slides
	08.29.2016	Cheng Huang	Fast Paxos	video, slides
	09.23.2016	Rustan Leino	Global Snapshots	video, slides
	11.09.2016	Heidi Howard	Flexible Paxos	video, slides
	01.20.2017	Shuai Mu	Byzantine Paxos	video, slides
⇒	03.01.2018	Ed Huang	Verifying Distributed Transaction with TLA+ in TiDB	
	TRD	Stephan Merz	Termination Detection	

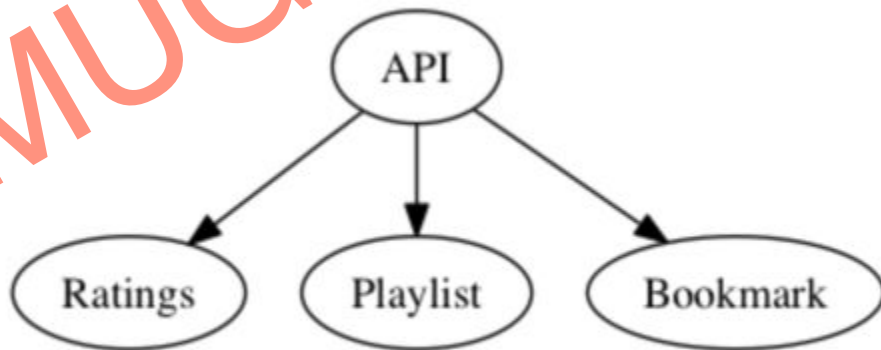
# Automating Failure Test

For Fault-Tolerance System  
Fault-Tolerance is Redundancy

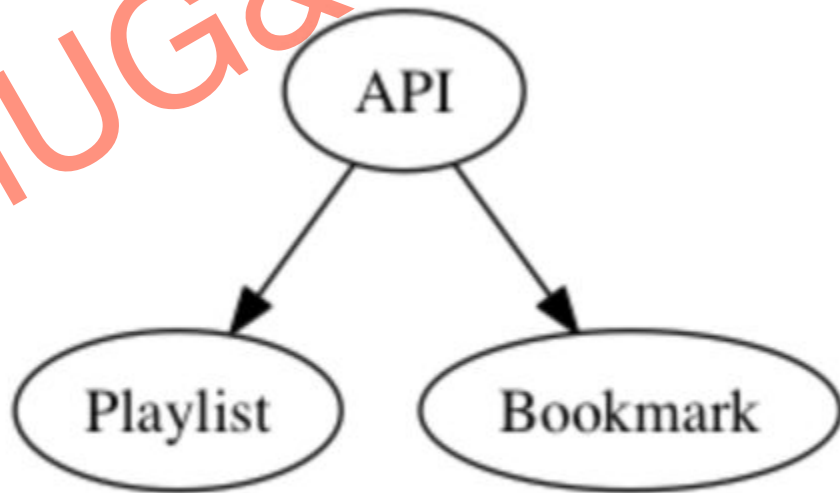
# Lineage



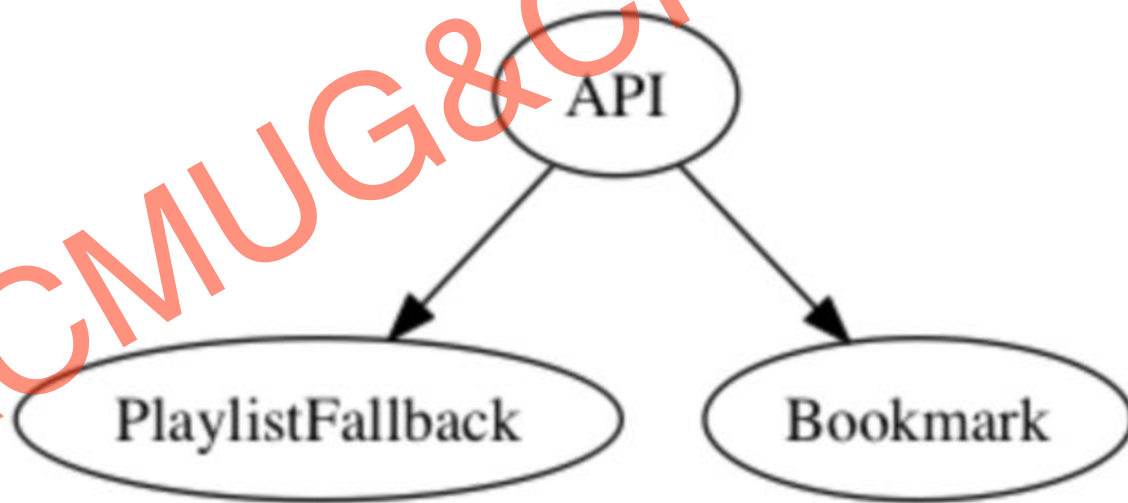
1. Starts with a successful outcome. Work backwards.
2. Ask why it happened: Lineage
3. Convert lineage to a boolean formula and solve
4. Lather, rinse and repeat



(A or R or P or B)



(A or P or B) and (A or P or B or R)



(A or PF or B) and (A or P or B) and (A or P or B or R)

Fail point: [{A}, {PF}, {B}, {P, PF}, {R, A}, {R, B}...]



1. How to define Success?
2. Idempotence
3. Narrow Scope

ACMUG&CRUG



# Oasis

ACMUG&CRUG

Something wrong!!!



# Oasis



Datasource

Prometheus

Log

...

Model

iForest

K-NN

SVM

Rules

...

Job

Prometheus

iForest

Prometheus

Rules

# Oasis

How can we find the anomaly automatically?

- iForest Model
  1. Learn the history metrics, extract a features vector
  2. Use iForest model to train the history and get a predict
  3. Check current metric feature match predict, if not, sometime maybe wrong.

ACMUG&CRUG



Thank you!

We are hiring!