OSC源创会 年终盛典 2016

TiDB Theory and Practice

liuqi@pingcap.com

Who am I

- Qi Liu (刘奇)
- Co-founder & CEO of PingCAP
- JD/Wandoulabs
- Infrastructure software engineer / Open source hacker
- Codis / TiDB / TiKV

What’ s TiDB

- NewSQL database inspired by Google Spanner / F1
- Open source, of course

https://github.com/pingcap/tidb

TiDB
A Distributed SQL Database

What’ s new at the end of 2016

- TiDB
  - Open source for 1+ years
  - 5300+ stars
  - 58+ people
  - 4000+ commits
  - 31 meetups
  - Alpha → Beta → RC1

What surprises me?

- Game companies need new technology
- Internet companies
- Other traditional companies

Why TiDB?

First, I want to ask one question:

How to scale your MySQL database?

Why TiDB?

- **No more:**
  - splitting DB/Table
  - choosing sharding keys
  - workarounds for cross-shard transaction support
  - inconsistent data
  - waking up at midnight to do DDL or re-shard :)
  - slow queries that can’t scale

Why TiDB?

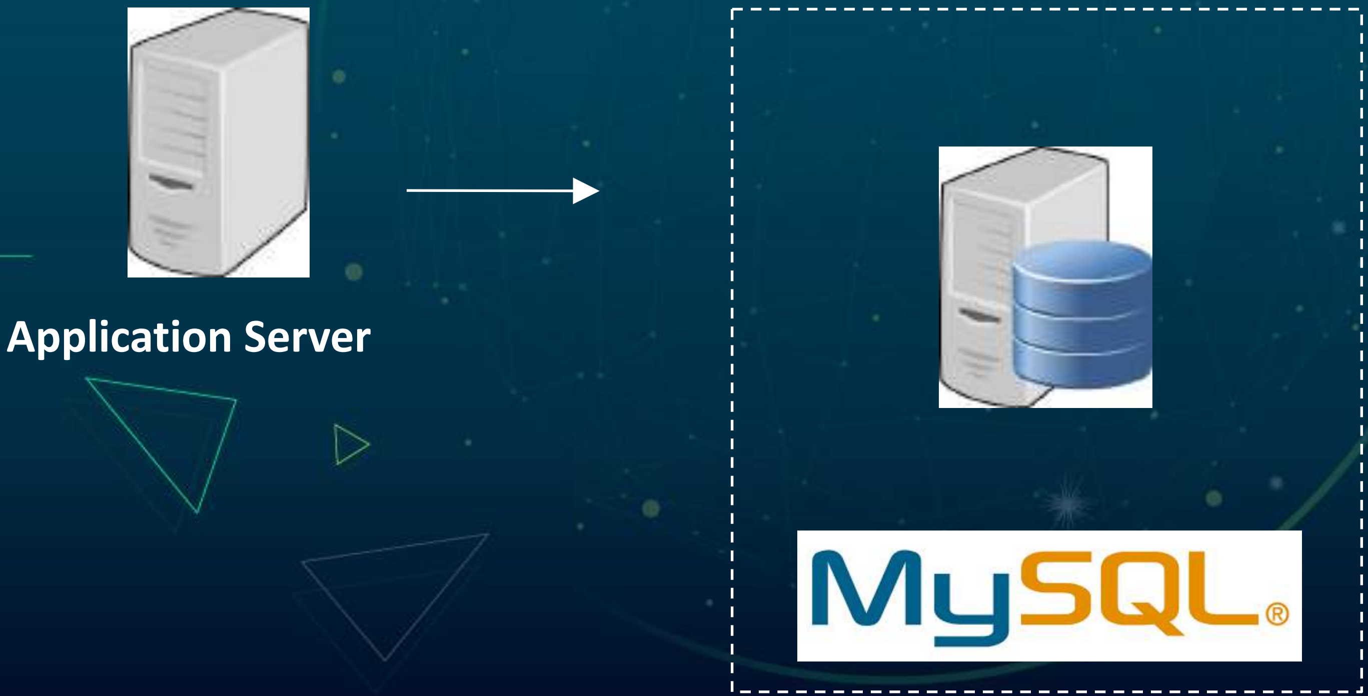
- MySQL grammar and protocol compatibility ✓
- Complex query support: Join / Subquery / Group By / … ✓
- ACID Transaction ✓
- Elastic scaling ✓
- Auto-failover ✓

Patterns.

All come from real user cases.

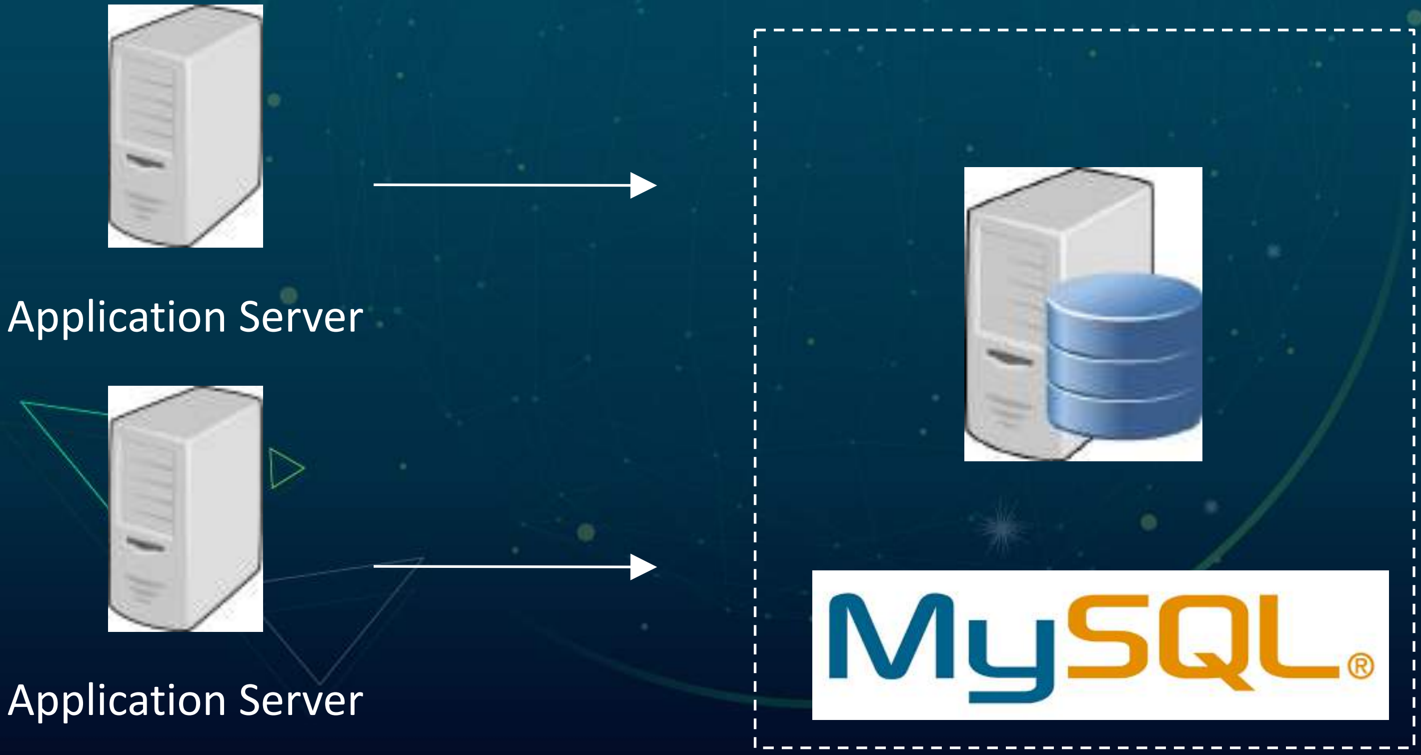
Pattern 1: 100x MySQL

- At first, you got a MySQL and one application server.



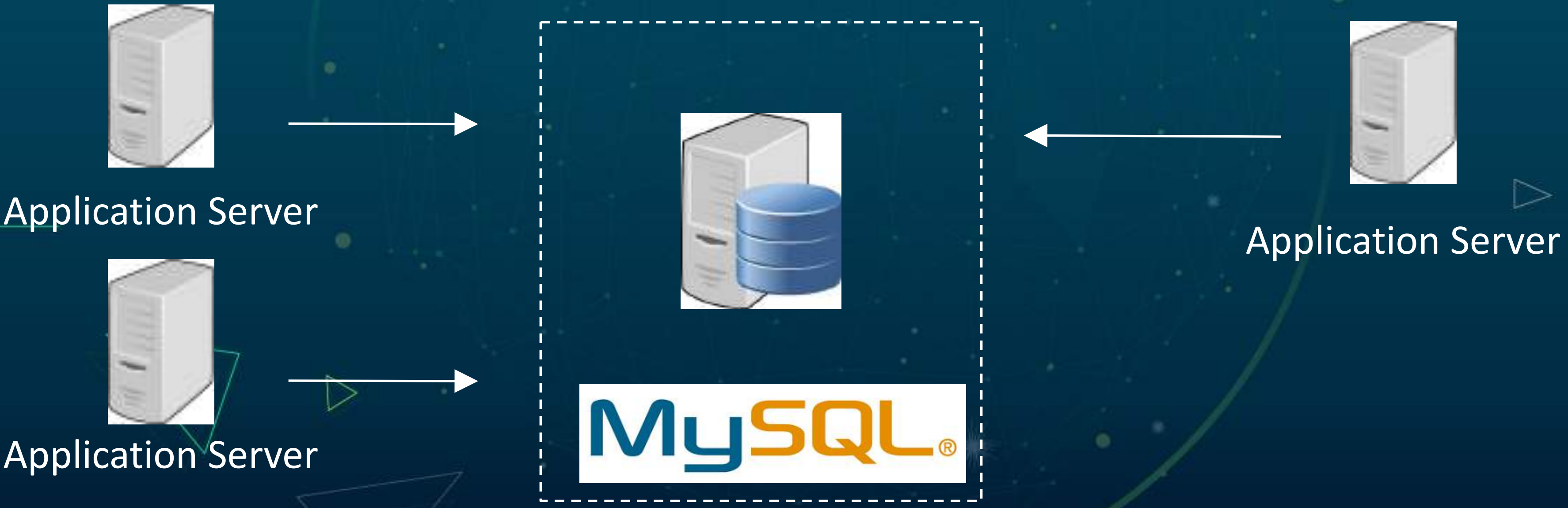
Pattern 1: 100x MySQL

- And then, workload continuously increases.



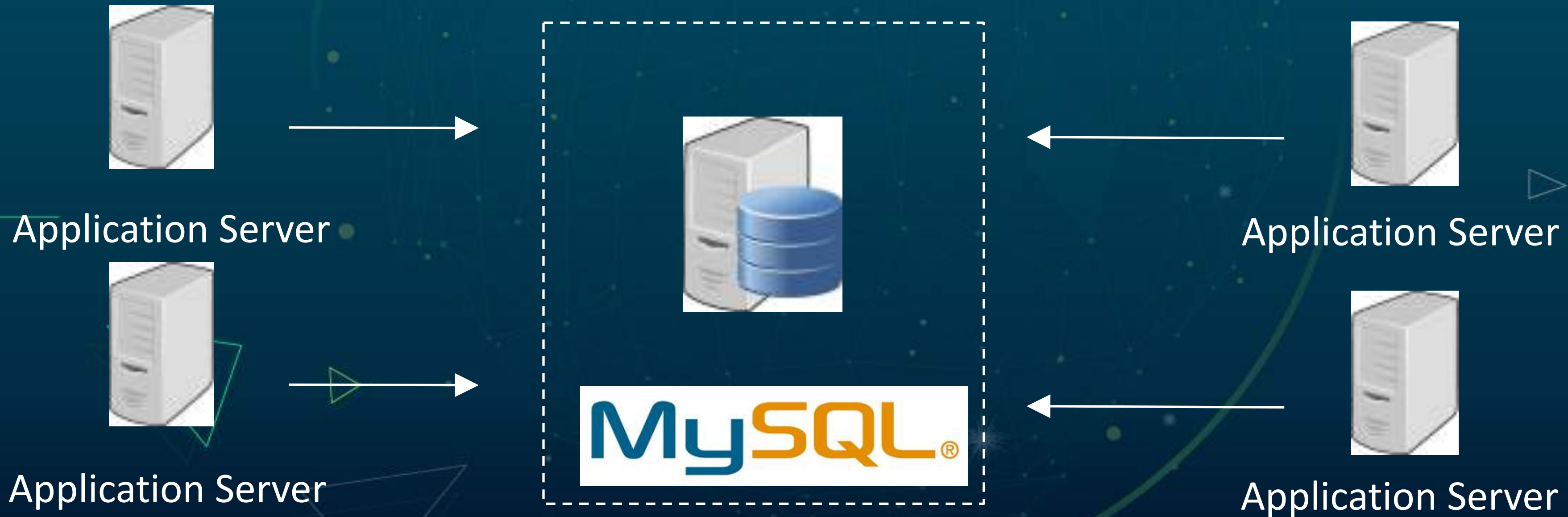
Pattern 1: 100x MySQL

- And then, workload continuously increases.



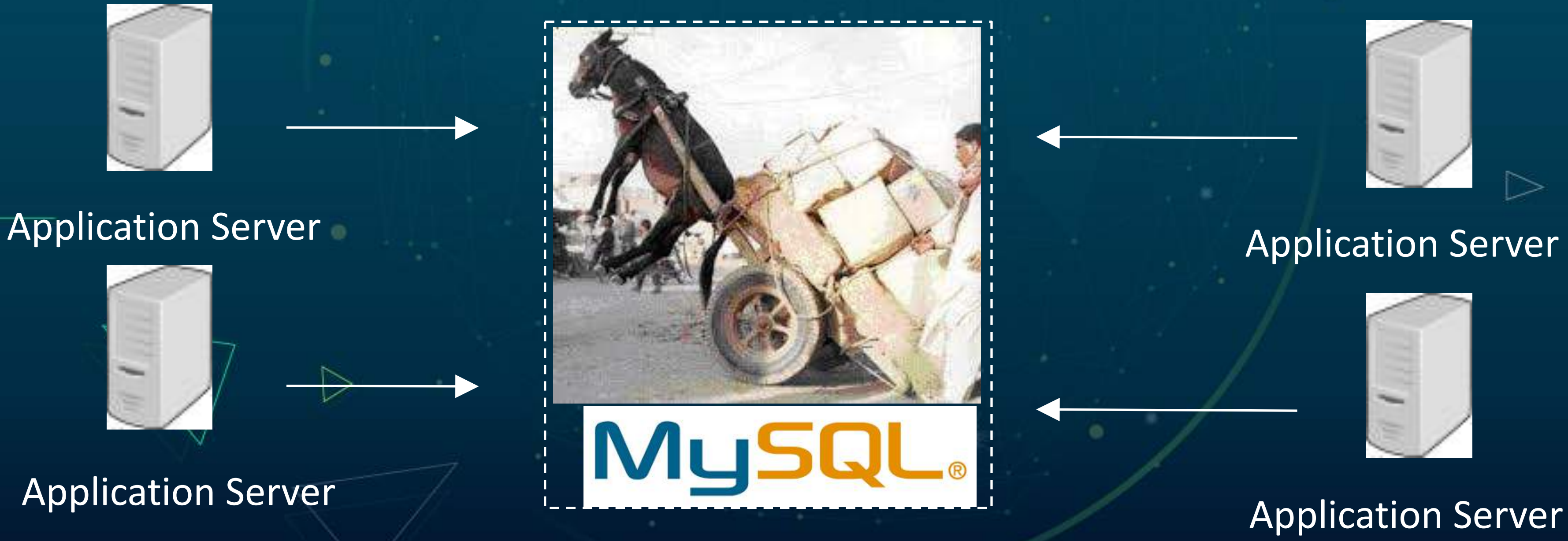
Pattern 1: 100x MySQL

- To cope with the continuously increasing workload, you add more and more application servers.



Pattern 1: 100x MySQL

- One day, shit happens.



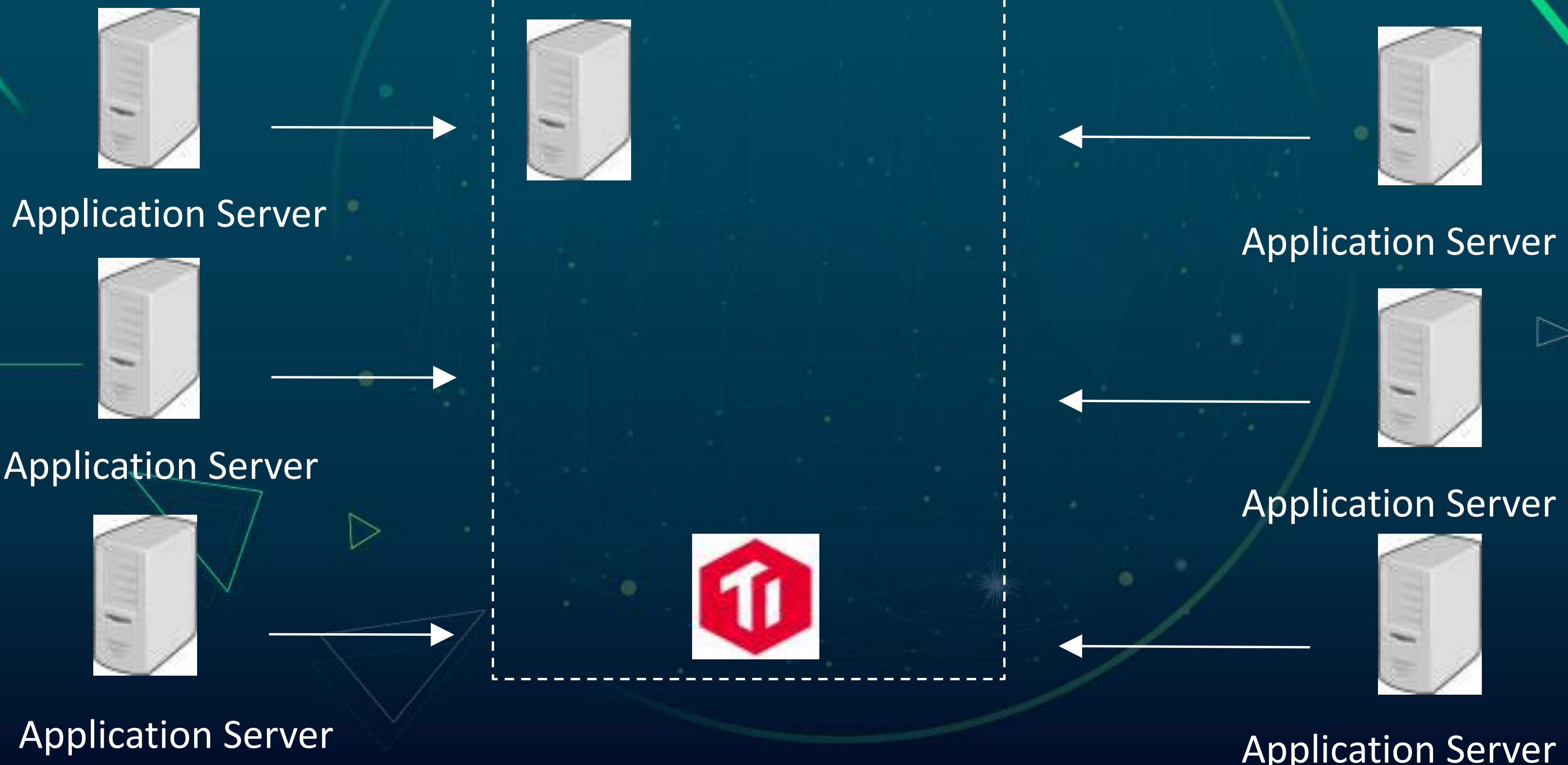
Application Server

Application Server

Application Server

Application Server

Pattern 1: 100x MySQL



Application Server

Application Server

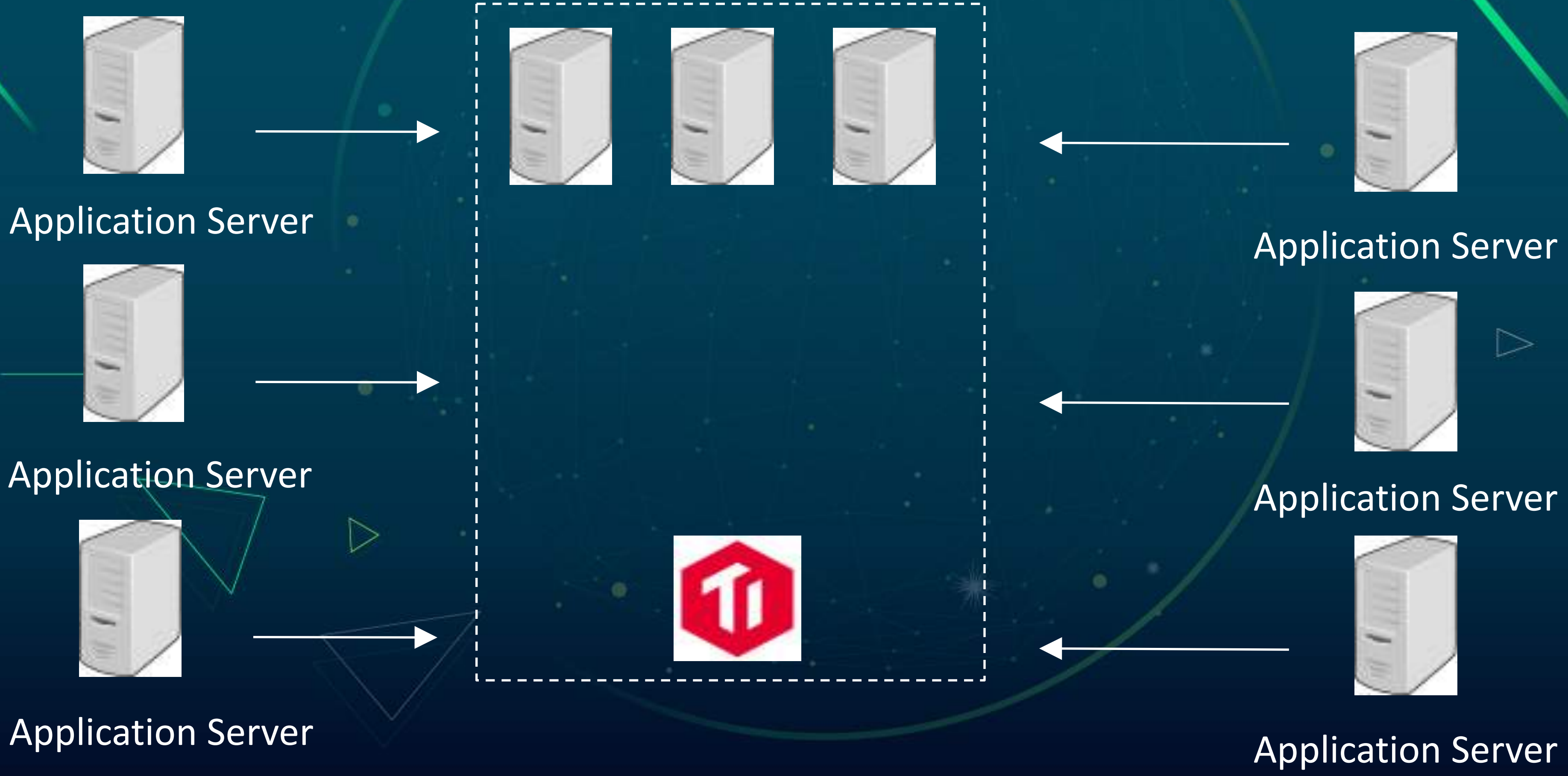
Application Server

Application Server

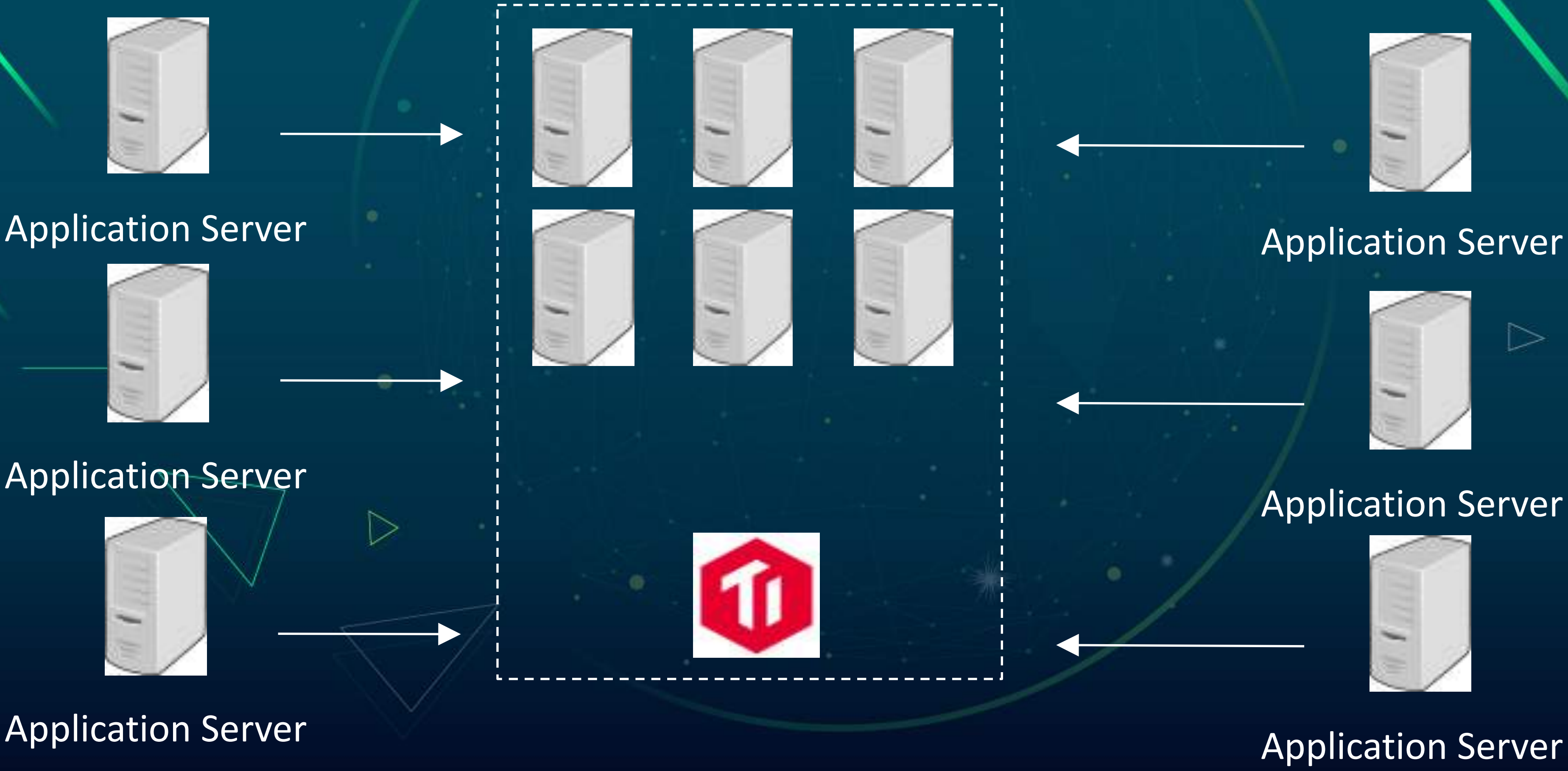
Application Server

Application Server

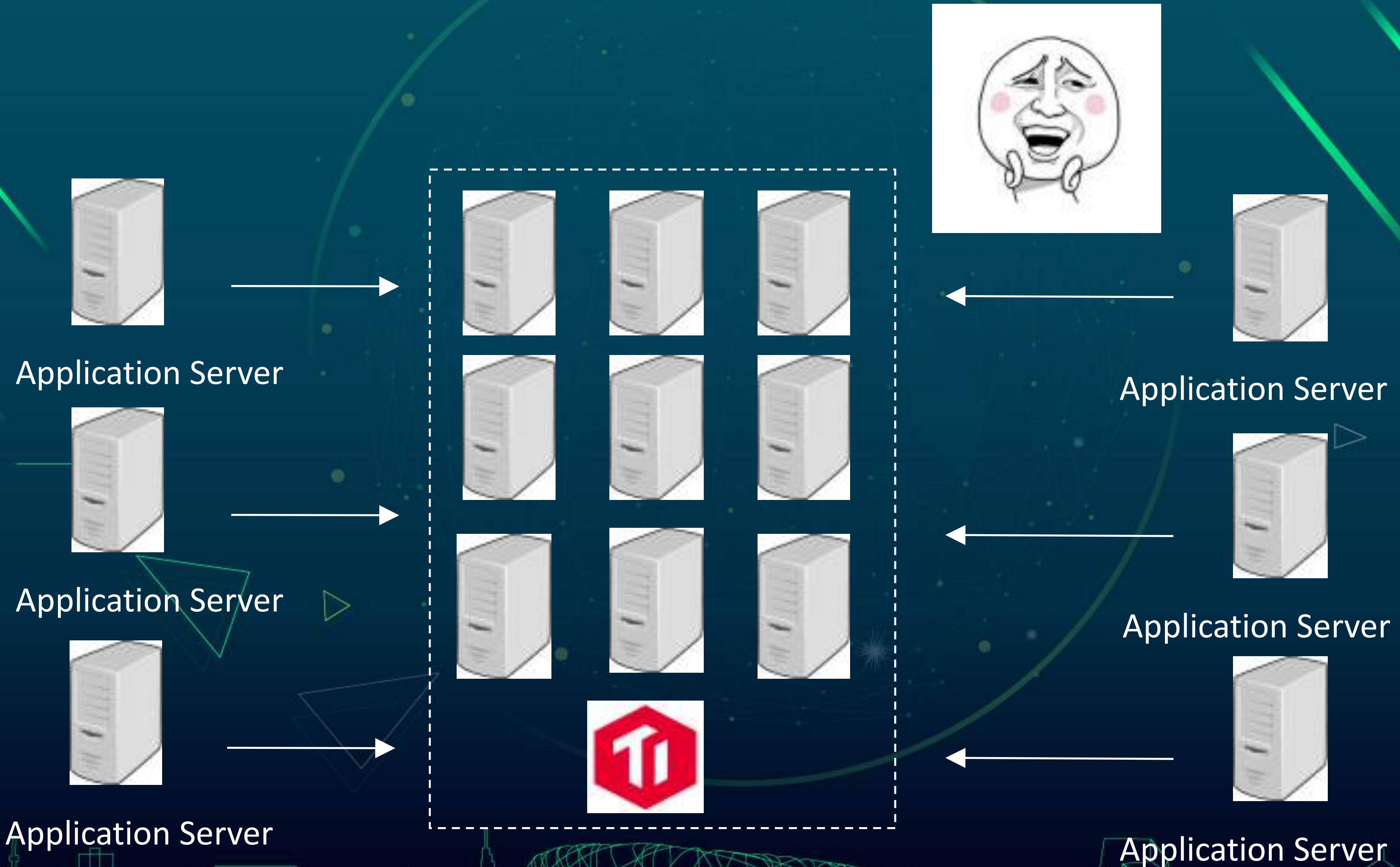
Pattern 1: 100x MySQL



Pattern 1: 100x MySQL



Pattern 1: 100x MySQL



Pattern 1: 100x MySQL

- TiDB supports elastic scaling.
- Adding more machines, TiDB will rebalance the load and data.
- Thanks to the Raft consensus algorithm.

Manual sharding

Remember that, as your business grows rapidly, you don’t want to waste time on refactoring your code...

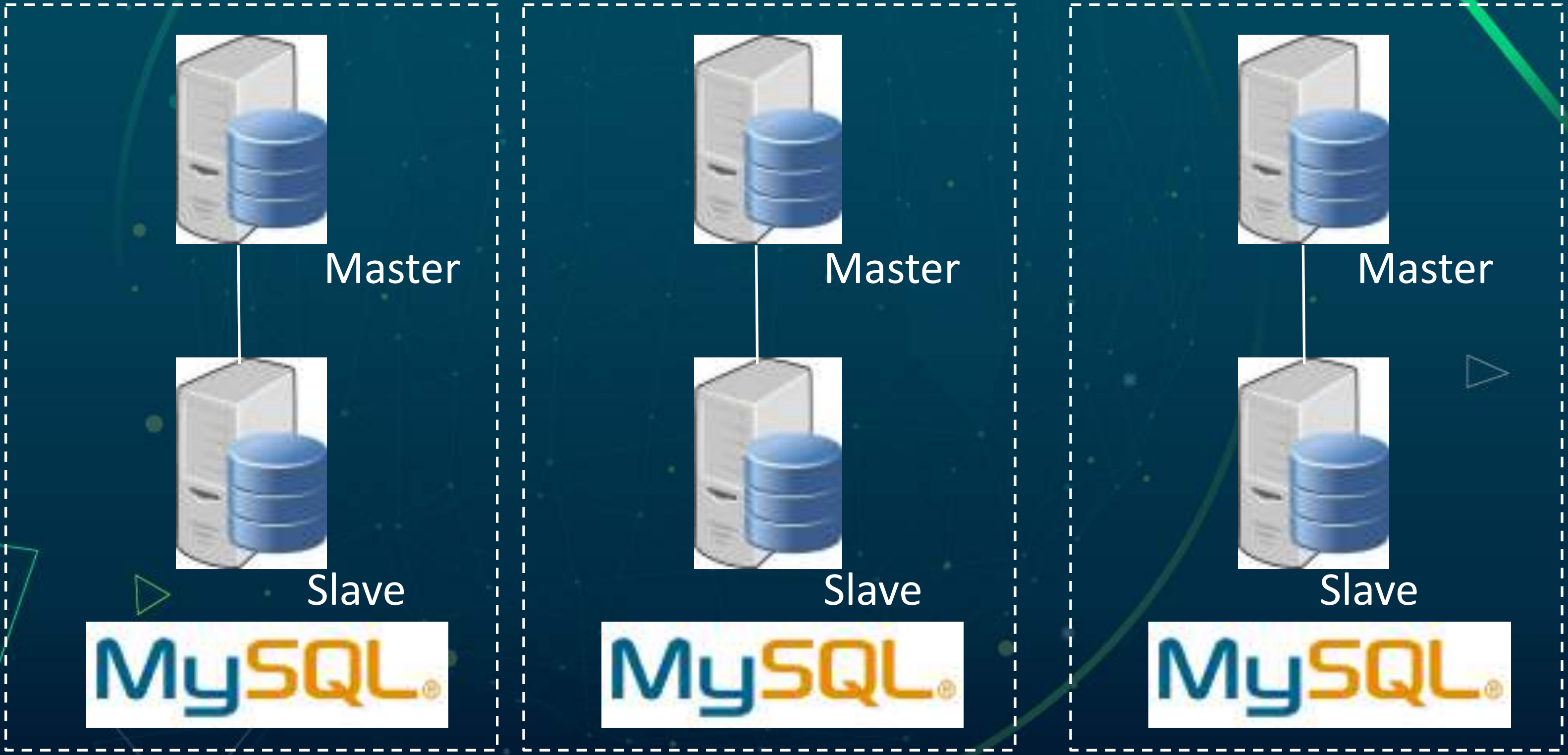
Pattern 1: 100x MySQL

Scale without changing a single line of code.

Pattern 2: Real-time backup

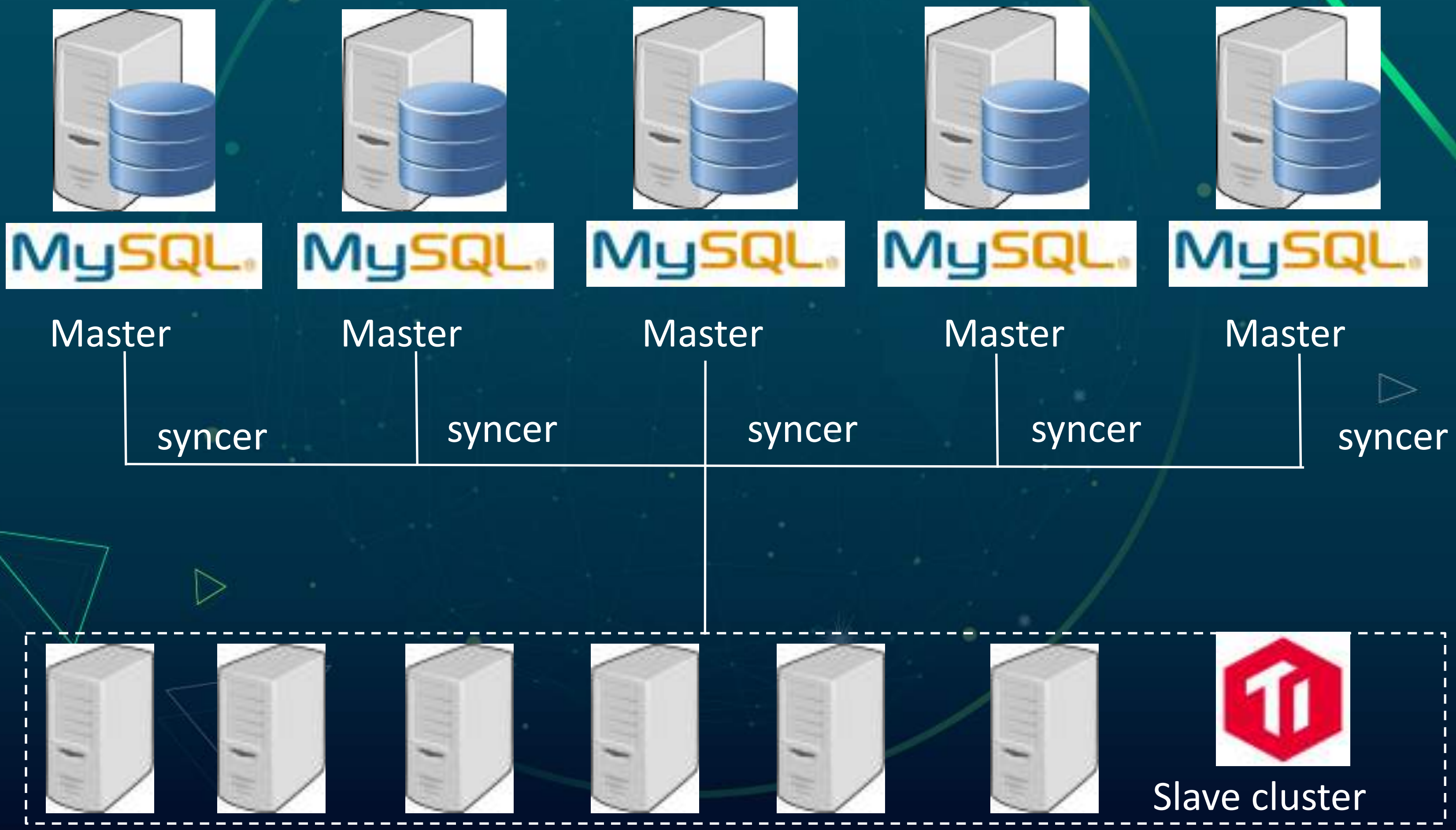
OSC源创会 年终盛典 2016

Old days:



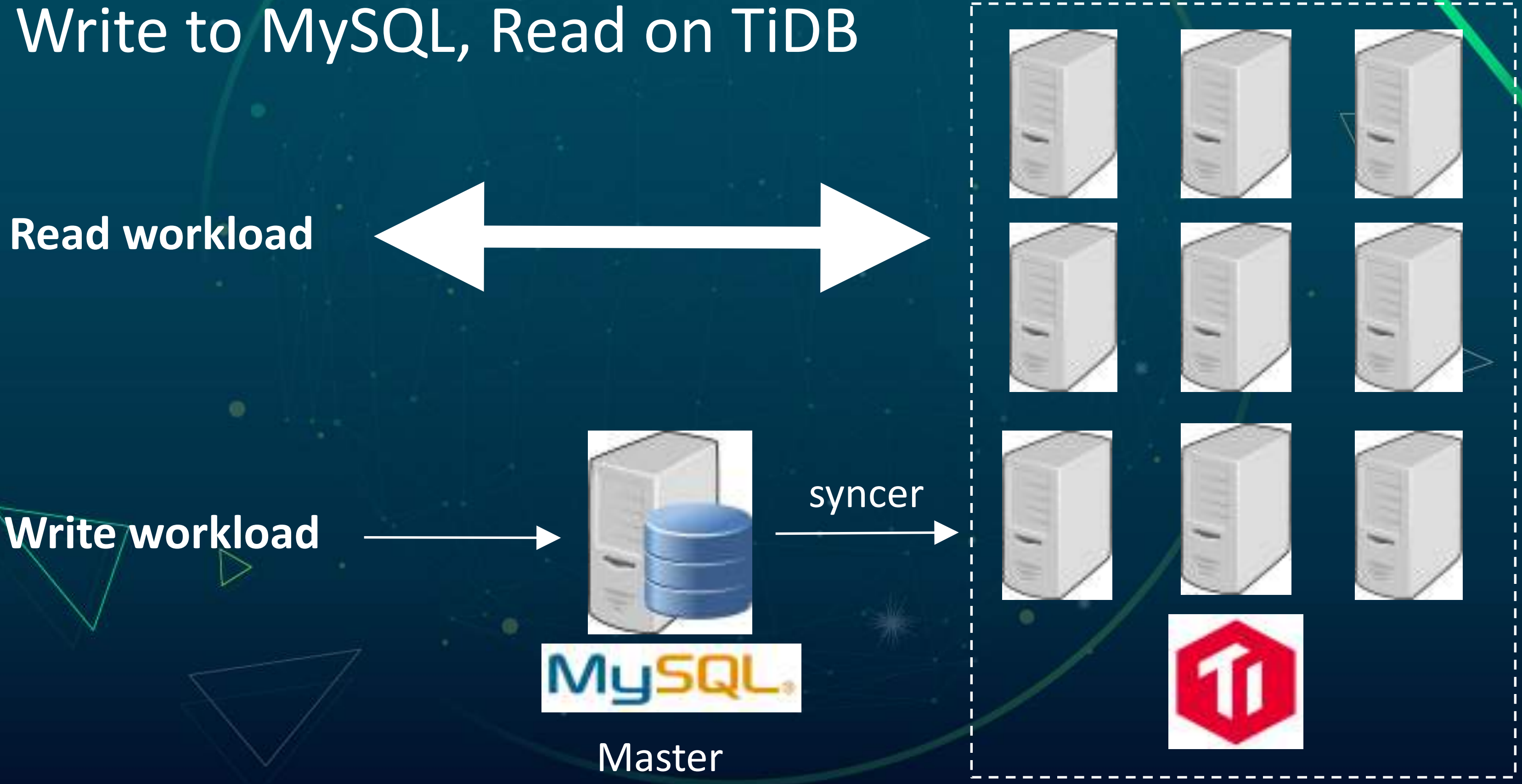
Pattern 2: Real-time backup

Now:



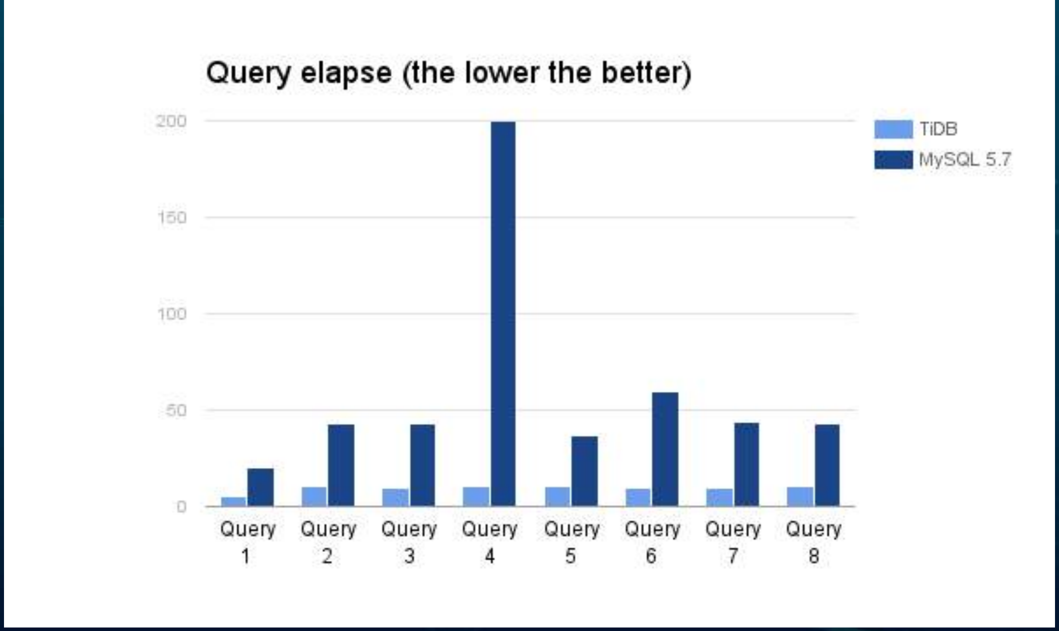
Pattern 3: Read/Write splitting

- Write to MySQL, Read on TiDB



Pattern 4: Ad-Hoc OLAP

- Why MySQL?
- Why MySQL sucks?



TiDB Elapse	MySQL Elapse
5.07699437s	19.93s
10.524703077s	43.23s
10.077812714s	43.33s
10.285957629s	>20 mins
10.462306097s	36.81s
9.968078965s	1 min 0.27 sec
9.998030375s	44.05s
10.866549284s	43.18s

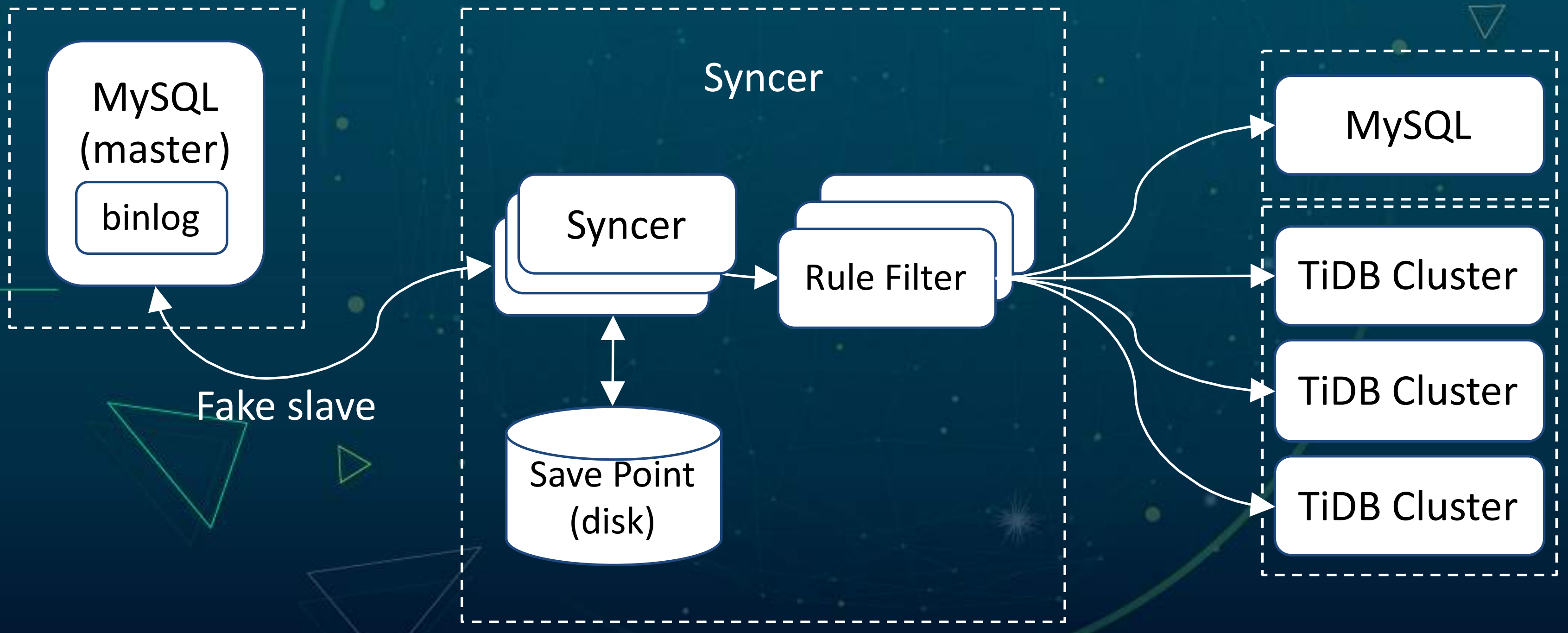
Tools matter.

Make miracles happen

syncer

- MySQL row-based binlog parser and real-time data synchronization to any point which is compatible with MySQL protocol, like MySQL, TiDB.
- Auto reconnection, high concurrent and savepoint support.
- For more information, see syncer.

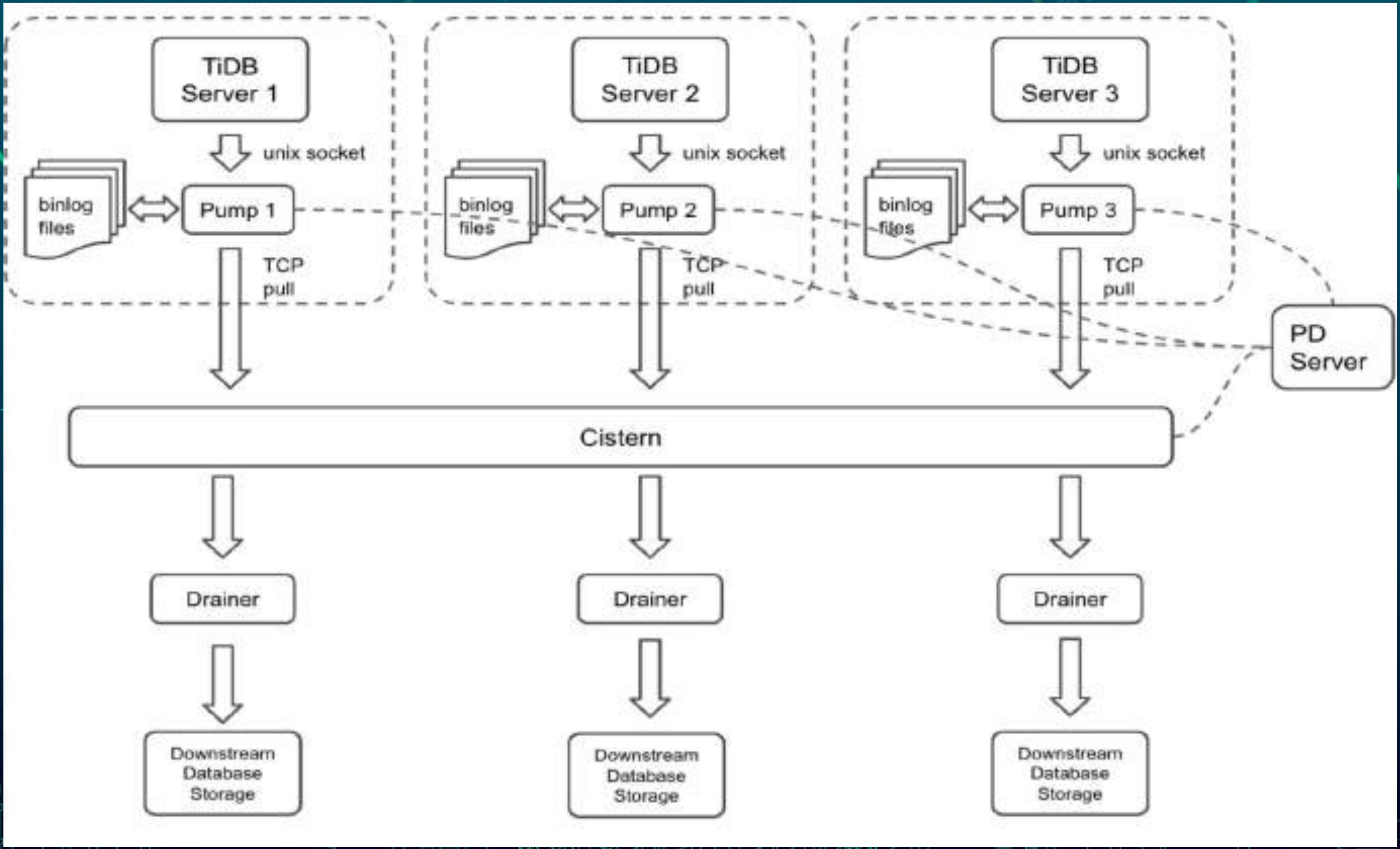
syncer


MySQL
(master)
binlog
Fake slave
Syncer
Syncer
Rule Filter
Save Point
(disk)
MySQL
TiDB Cluster
TiDB Cluster
TiDB Cluster

TiDB binlog

- Collect TiDB's binlogs for quasi real-time data backup and synchronization. Of course, it’s distributed.
- Self-description, support syncing from any point
- Awesome tool for production with mydumper/myloader.

TiDB binlog


TiDB Server 1
TiDB Server 2
TiDB Server 3
unix socket
unix socket
unix socket
binlog files
binlog files
binlog files
Pump 1
Pump 2
Pump 3
TCP pull
TCP pull
TCP pull
PD Server
Cistern
Drainer
Drainer
Drainer
Downstream Database Storage
Downstream Database Storage
Downstream Database Storage

TiDB binlog

- **pump**

Pump is a daemon that receives real-time binlog from tidb-server and writes in sequential disk files synchronously.

- **cistern**

Cistern collects binlog from each pump in cluster, and stores them on disk in order of commitTS.

- **drainer**

Drainer transforms binlog to various dialects of SQL, and applies to downstream database or filesystem. (Not only MySQL :))

mydumper / myloader

- Pros:
  - Multithread/Fast
  - Not LSM engine friendly
- Cons:
  - Lacks of retry logic.

The reason we rewrite myloader with go

Reliable

More friendly to LSM engine

Community matter

Work with Spark.

More raw KV interfaces：get/set/cas

More and more documents

Thanks

Project Repo:

https://github.com/pingcap/tidb

https://github.com/pingcap/tikv

Documents:

https://github.com/pingcap/docs English

https://github.com/pingcap/docs-cn 简体中文

TiDB 交流群