

**01 TiDB**集群部署与基本使⽤

1. TiDB集群⼯具部署
   1. TiUP⾃动化部署集群
   2. 企业级TiDB集群部署
   3. 集群性能压测
   4. TiUP其他重要命令
2. TiDB连接管理,配置
   1. Mysql协议与连接
   2. TiDB集群配置与作⽤域
   3. TiDB数据库⽂件
3. TiDB集群管理
   1. 在线扩容
   2. 在线缩容
   3. 集群销毁
   4. 集群配置修改
4. TiDB监控体系

4.1.Prometheus+Grafana

* + 1. 基本介绍
    2. 监控指标**Overview**
    3. 监控指标TiDB

4.1.4.PD

4.1.5.TiKV-Details 4.2.Dashboard

1. **TiDB**集群⼯具部署
   1. **TiUP**⾃动化部署集群

tiup简介

1.4.0版本引⼊,集群管理与运维⼯具

2.可以进⾏部署,启动,关闭,销毁,扩缩容,集群配置变更,升级等重要操作

使⽤TiDB-playgroud组件部署本地集群(不⽤复数节点搭建)

1. 环境确认
   * 防⽕墙
   * ⽹络连接
   * linux centos7.5以上

2.下载tiup

curl --proto '=https' --tlsv1.2 -sSf https://tiup-mirrors.pingcap.com/install.sh | sh

1. 声明全局环境变量

source .bash\_profile

1. 安装tiup cluster组件 or 更新

tiup cluster or tiup update --self && tiup update cluster

1. 部署tidb集群，tidb-tikv,pd-tiflash各⼀个,成功部署后不要动 tiup playground

复数版本

tiup playground v5.0.0 --db 2 --pd 3 --kv 3 --monitor

1. 连接,另起⼀个窗⼝

mysql -uroot -p -P4000 -h 127.0.0.1

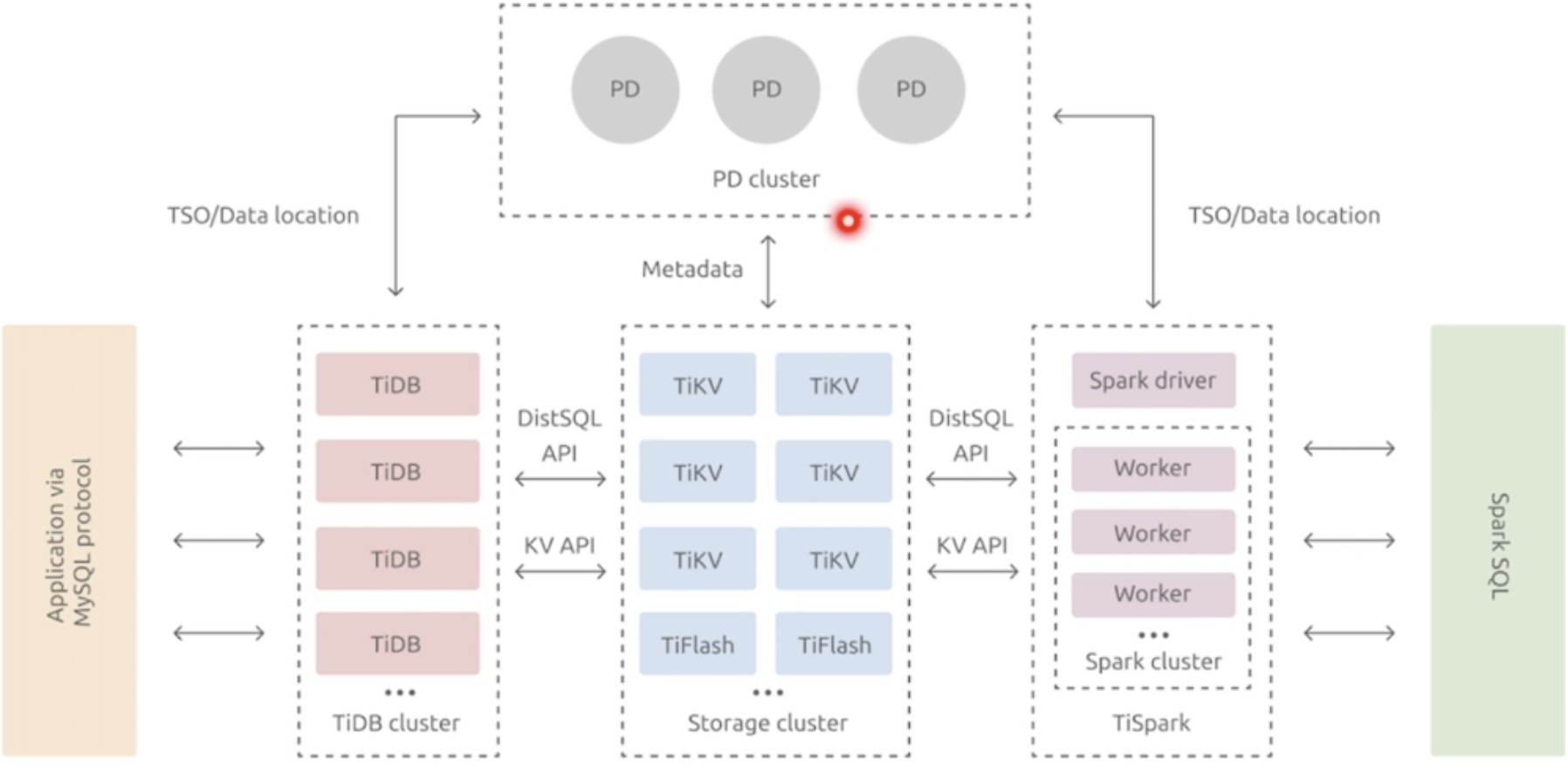
1. 观察集群与dashboard是否可⽤ 8.清理集群

tiup clean --all

* 1. 企业级**TiDB**集群部署

DBA要求

1.tispark⼀般留给⼤数据部门同学使⽤



1. 部署前置硬件部署



1. 实例最低要求 2tidb 3pd 3tikv,基础oltp集群
2. 轻量htap集群 1-2tiflash节点
3. 实时数仓集群功能: 1-2 ticdc 集群

环境要求

1. 关闭SWAP
2. 关闭防⽕墙
3. 安装NTP
4. 操作系统优化 5.ssh互信 6.numactl安装
5. 部署拓扑与配置⽂件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IP | hostname | role | 备注 |  |
| 172.21.64.7 | tidb-1 | tidb,pd,promethus,grafana |  |  |
| 172.21.64.8 | tidb-2 | tidb,pd |  |  |
| 172.21.64.9 | tidb-3 | tidb,pd |  |  |
| 172.21.64.10 | tikv-1 | tikv |  |  |
| 172.21.64.24 | tikv-2 | tikv |  |  |
| 172.21.64.25 | tikv-3 | tikv |  |  |
| 172.21.64.26 | tiflash-1 | tiflash |  |  |
| 172.21.64.23 | tiflash-2 | tiflash |  |  |

配置⽂件

1.

mkdir -p tidb-deploy/tidb\_test

tiup cluster template > topology.yaml

vim tidb\_test.yaml global:

user: "tidb" ssh\_port: 22

deploy\_dir: "/chj/app/tidb/deploy"

data\_dir: "/chj/app/tidb/data" arch: "amd64"

monitored: node\_exporter\_port: 9100

blackbox\_exporter\_port: 9115

deploy\_dir: "/chj/app/tidb/monitored/monitored-9100" data\_dir: "/chj/app/tidb/monitored/monitored-9100/data" log\_dir: "/chj/app/tidb/monitored/monitored-9100/log"

server\_configs: tidb:

split-table: true

mem-quota-query: 2147483648 oom-use-tmp-storage: true tmp-storage-quota: 2147483648 oom-action: "log"

max-server-connections: 2000

max-index-length: 6144

table-column-count-limit: 4096

index-limit: 64 log.level: "info" log.format: "text" log.enable-slow-log: true log.slow-threshold: 3000

log.record-plan-in-slow-log: 1

log.expensive-threshold: 1000000

log.query-log-max-len: 40960000

log.file.max-days: 30 binlog.enable: false binlog.ignore-error: false performance.max-procs: 32

performance.server-memory-quota: 128849018880

performance.memory-usage-alarm-ratio: 0.8

performance.txn-entry-size-limit: 6291456

performance.txn-total-size-limit: 209715200 performance.cross-join: false performance.pseudo-estimate-ratio: 0.5 status.record-db-qps: true

stmt-summary.max-stmt-count: 200000

stmt-summary.max-sql-length: 409600

pessimistic-txn.max-retry-count: 64

pessimistic-txn.deadlock-history-capacity: 10000 experimental.allow-expression-index: true

tikv:

raftdb.defaultcf.force-consistency-checks: false raftstore.apply-max-batch-size: 256

raftstore.apply-pool-size: 10 raftstore.hibernate-regions: true raftstore.messages-per-tick: 1024

raftstore.perf-level: 5

raftstore.raft-max-inflight-msgs: 2048

raftstore.store-max-batch-size: 256

raftstore.store-pool-size: 8 raftstore.sync-log: false readpool.coprocessor.use-unified-pool: true readpool.storage.use-unified-pool: true readpool.unified.max-thread-count: 12

rocksdb.defaultcf.force-consistency-checks: false rocksdb.lockcf.force-consistency-checks: false rocksdb.raftcf.force-consistency-checks: false rocksdb.writecf.force-consistency-checks: false server.grpc-concurrency: 8

storage.block-cache.capacity: 32G storage.scheduler-worker-pool-size: 8

pd:

schedule.leader-schedule-limit: 4

schedule.region-schedule-limit: 1024

schedule.replica-schedule-limit: 8 tiflash:

path\_realtime\_mode: false logger.level: "info"

tiflash-learner: log-level: "info"

raftstore.apply-pool-size: 4

raftstore.store-pool-size: 4

pd\_servers:

- host: 172.21.64.7

ssh\_port: 22 name: "pd-55" client\_port: 2379

peer\_port: 2380

deploy\_dir: "/chj/app/tidb/deploy/pd-2379" data\_dir: "/chj/app/tidb/data/pd-2379" log\_dir: "/chj/app/tidb/deploy/pd-2379/log"

- host: 172.21.64.8

ssh\_port: 22 name: "pd-56" client\_port: 2379

peer\_port: 2380

deploy\_dir: "/chj/app/tidb/deploy/pd-2379" data\_dir: "/chj/app/tidb/data/pd-2379" log\_dir: "/chj/app/tidb/deploy/pd-2379/log"

- host: 172.21.64.9

ssh\_port: 22 name: "pd-57" client\_port: 2379

peer\_port: 2380

deploy\_dir: "/chj/app/tidb/deploy/pd-2379" data\_dir: "/chj/app/tidb/data/pd-2379" log\_dir: "/chj/app/tidb/deploy/pd-2379/log"

tidb\_servers:

- host: 172.21.64.7

ssh\_port: 22

port: 4000

status\_port: 10080

deploy\_dir: "/chj/app/tidb/deploy/tidb-4000" log\_dir: "/chj/app/tidb/deploy/tidb-4000/log" config:

log.level: info

log.slow-query-file: tidb\_slow\_query.log

- host: 172.21.64.8

ssh\_port: 22

port: 4000

status\_port: 10080

deploy\_dir: "/chj/app/tidb/deploy/tidb-4000" log\_dir: "/chj/app/tidb/deploy/tidb-4000/log" config:

log.level: info

log.slow-query-file: tidb\_slow\_query.log

- host: 172.21.64.9

ssh\_port: 22

port: 4000

status\_port: 10080

deploy\_dir: "/chj/app/tidb/deploy/tidb-4000" log\_dir: "/chj/app/tidb/deploy/tidb-4000/log" config:

log.level: info

log.slow-query-file: tidb\_slow\_query.log

tikv\_servers:

- host: 172.21.64.10

ssh\_port: 22

port: 20160

status\_port: 20180

deploy\_dir: "/chj/app/tidb/deploy/tikv-20160" data\_dir: "/chj/app/tidb/data/tikv-20160" log\_dir: "/chj/app/tidb/deploy/tikv-20160/log"

- host: 172.21.64.24

ssh\_port: 22

port: 20160

status\_port: 20180

deploy\_dir: "/chj/app/tidb/deploy/tikv-20160" data\_dir: "/chj/app/tidb/data/tikv-20160" log\_dir: "/chj/app/tidb/deploy/tikv-20160/log"

- host: 172.21.64.25

ssh\_port: 22

port: 20160

status\_port: 20180

deploy\_dir: "/chj/app/tidb/deploy/tikv-20160" data\_dir: "/chj/app/tidb/data/tikv-20160" log\_dir: "/chj/app/tidb/deploy/tikv-20160/log"

tiflash\_servers:

- host: 172.21.64.23

ssh\_port: 22

tcp\_port: 9000

http\_port: 8123

flash\_service\_port: 3930

flash\_proxy\_port: 20170

flash\_proxy\_status\_port: 20292

metrics\_port: 8234

deploy\_dir: "/chj/app/tidb/deploy/tiflash-9000" data\_dir: "/chj/app/data/tiflash-9000"

log\_dir: "/chj/app/tidb/deploy/tiflash-9000/log"

- host: 172.21.64.26

ssh\_port: 22

tcp\_port: 9000

http\_port: 8123

flash\_service\_port: 3930

flash\_proxy\_port: 20170

flash\_proxy\_status\_port: 20292

metrics\_port: 8234

deploy\_dir: "/chj/app/tidb/deploy/tiflash-9000" data\_dir: "/chj/app/data/tiflash-9000"

log\_dir: "/chj/app/tidb/deploy/tiflash-9000/log"

monitoring\_servers:

- host: 172.21.64.7

ssh\_port: 22

port: 9090

deploy\_dir: "/chj/app/tidb/deploy/prometheus-8249" data\_dir: "/chj/app/tidb/data/prometheus-8249" log\_dir: "/chj/app/tidb/deploy/prometheus-8249/log"

grafana\_servers:

- host: 172.21.64.7

port: 3000

deploy\_dir: /chj/app/tidb/deploy/grafana-3000

alertmanager\_servers:

- host: 172.21.64.7

ssh\_port: 22

web\_port: 9093

cluster\_port: 9094

deploy\_dir: "/chj/app/tidb/deploy/alertmanager-9093" data\_dir: "/chj/app/tidb/data/alertmanager-9093" log\_dir: "/chj/app/tidb/deploy/alertmanager-9093/log"

1. TiUP集群部署
2. 检查集群配置

tiup cluster check ./tidb\_online\_10100.yaml --user tidb

1. 修复⻛险

tiup cluster check ./tidb\_online\_10100.yaml --user tidb --apply

1. 部署集群

tiup cluster deploy tidb\_online\_10100 v5.4.0 ./tidb\_online\_10100.yaml --user tidb

1. 检查所有集群

tiup cluster list

1. 检查制定集群

tiup cluster display tidb\_online\_10100

1. 启动集群

tiup cluster start tidb\_online\_10100 --init 1)启动顺序:pd-tikv-tidb-tiflash

2)提⽰初始密码 如果没有--int则为空

Started cluster `tidb\_online\_10100` successfully

The root password of TiDB database has been changed. The new password is: '9wm2-E7@P8Fa\_X^R16'.

Copy and record it to somewhere safe, it is only displayed once, and will not be stored.

The generated password can NOT be get and shown again.

7.检查dashboard和grafana状态

8.停⽌集群

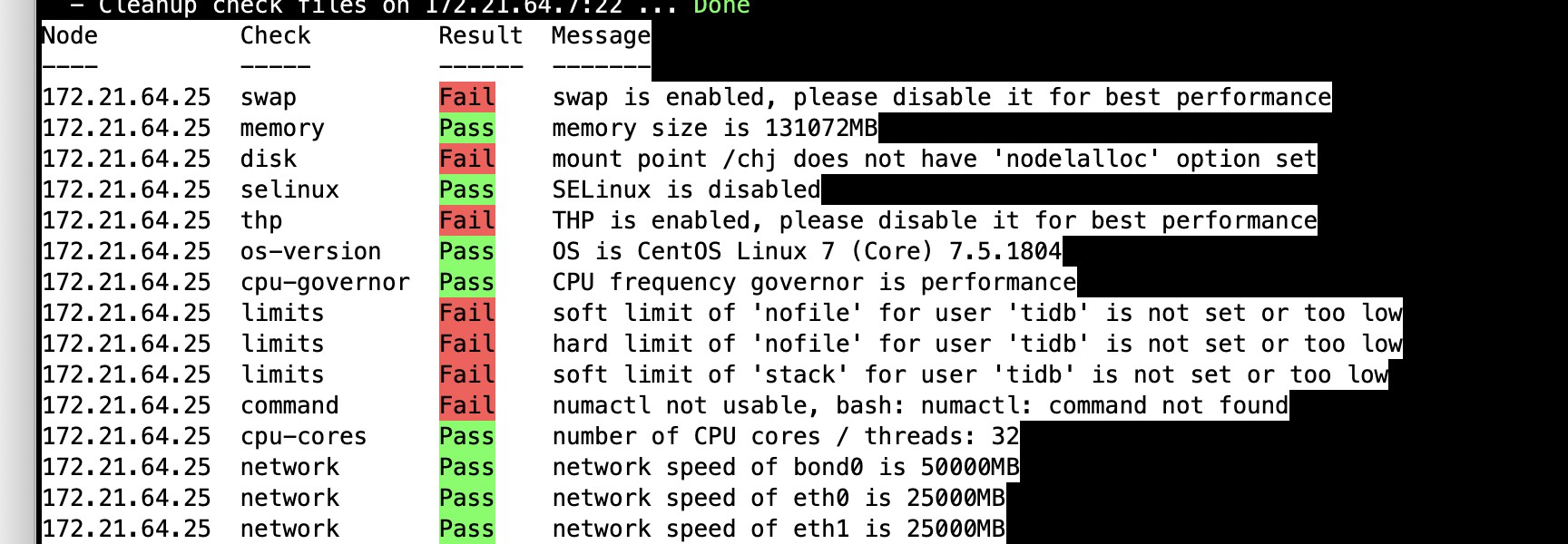
tiup cluster stop tidb\_online\_10100

1)停⽌顺序：tiflash-->tidb-->tikv-->od

9.销毁集群

tiup cluster destory tidb\_online\_10100

fail代表监测有问题 需要修复



# 集群性能压测

* 1. **TiUP**其他重要命令

tiup [flags] <command> [args...]

#核⼼命令

1. 指定节点or⾓⾊开启

tiup cluster start tidb\_online\_10100 --node 172.21.64.7:2379 tiup cluster start tidb\_online\_10100 --role tidb

1. 指定节点or⾓⾊停⽌

tiup cluster stop tidb\_online\_10100 --node 172.21.64.7:2379

tiup cluster stop tidb\_online\_10100 --role tidb

1. 指定节点or⾓⾊重启

tiup cluster restart tidb\_online\_10100 --node 172.21.64.7:2379 tiup cluster restart tidb\_online\_10100 --role tidb

1. 集群配置改变

tiup cluster edit-config tidb\_online\_10100

1. 集群reload平滑变更

tiup cluster reload tidb\_online\_10100 -R

6.集群rename

tiup cluster rename tidb\_online\_10100 tidb\_online\_10100\_new

1. **TiDB**连接管理**,**配置
   1. **Mysql**协议与连接
2. 使⽤准备

#1.账号创建

create user 'qianlong'@'172.21.%' IDENTIFIED BY '123456';

GRANT all privileges ON \*.\* TO 'qianlong'@'172.21.%' with grant option ;

#2.blb节点绑定(nginx即可 不做演⽰)

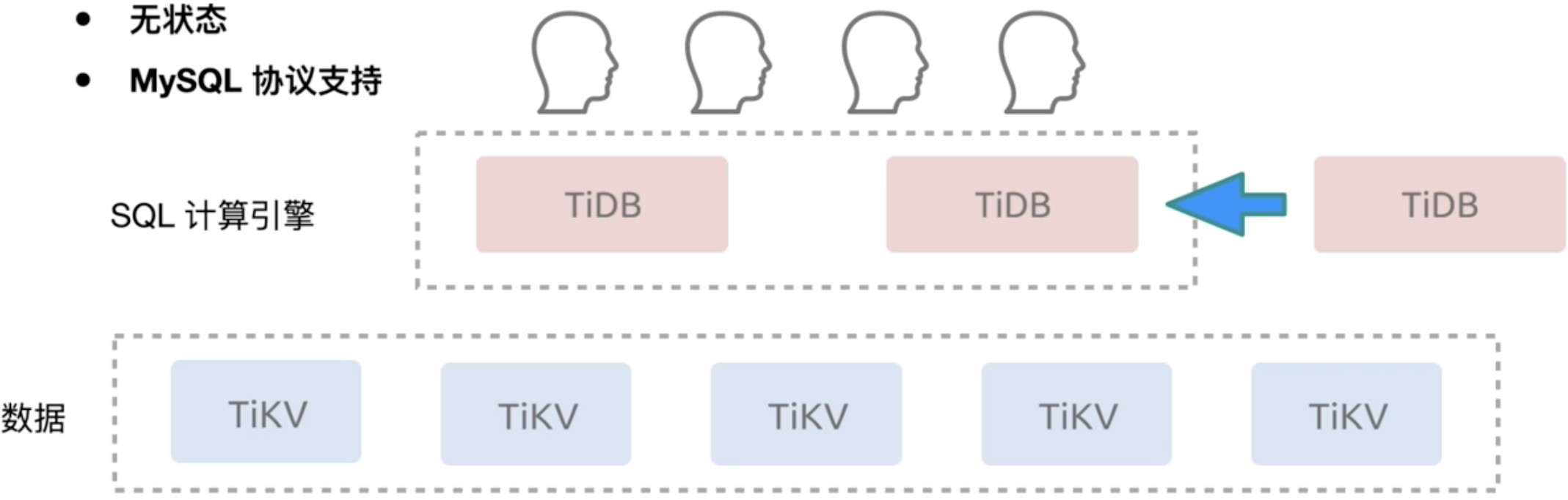
实验blb:172.21.77.51

1. TiDB Server与链接
2. tidb server⽆状态+⽀持mysql协议(5.7)
3. ⽤户增加并发可以增加tidb节点扩缩容(每个节点⼤约能⽀撑tp业务-2000~3000 qps)
4. ⽀持⼤部分Msyql5.7语法,不⽀持
   * 外键
   * 存储过程
   * 触发器
   * 多个ddl组合在⼀起
   * 更多语法兼容性 https://docs.pingcap.com/zh/tidb/stable/sql-statement-add-column
5. API⽀持

Mysql⽀持的都⽀持,⽐如

1. 使⽤限制

https://docs.pingcap.com/zh/tidb/stable/tidb-limitations



1. 基本使⽤
2. 查看数据库版本

select tidb\_version(); or \s

1. 建库建表

CREATE TABLE `tidb\_test` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT COMMENT '主键',

`indicate\_array\_config\_id` int(10) NOT NULL COMMENT 'test',

`metric\_result` varchar(32) NOT NULL COMMENT 'test',

`create\_time` datetime NOT NULL ON UPDATE CURRENT\_TIMESTAMP COMMENT '创建时间',

PRIMARY KEY (`id`) /\*T![clustered\_index] CLUSTERED \*/

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_bin AUTO\_INCREMENT=199041 COMMENT='tidb

\_test';

1. 插⼊⼀条数据

insert into tidb\_test(indicate\_array\_config\_id,metric\_result,create\_time) values(1,'test',now

());

* 1. **TiDB**集群配置与作⽤域

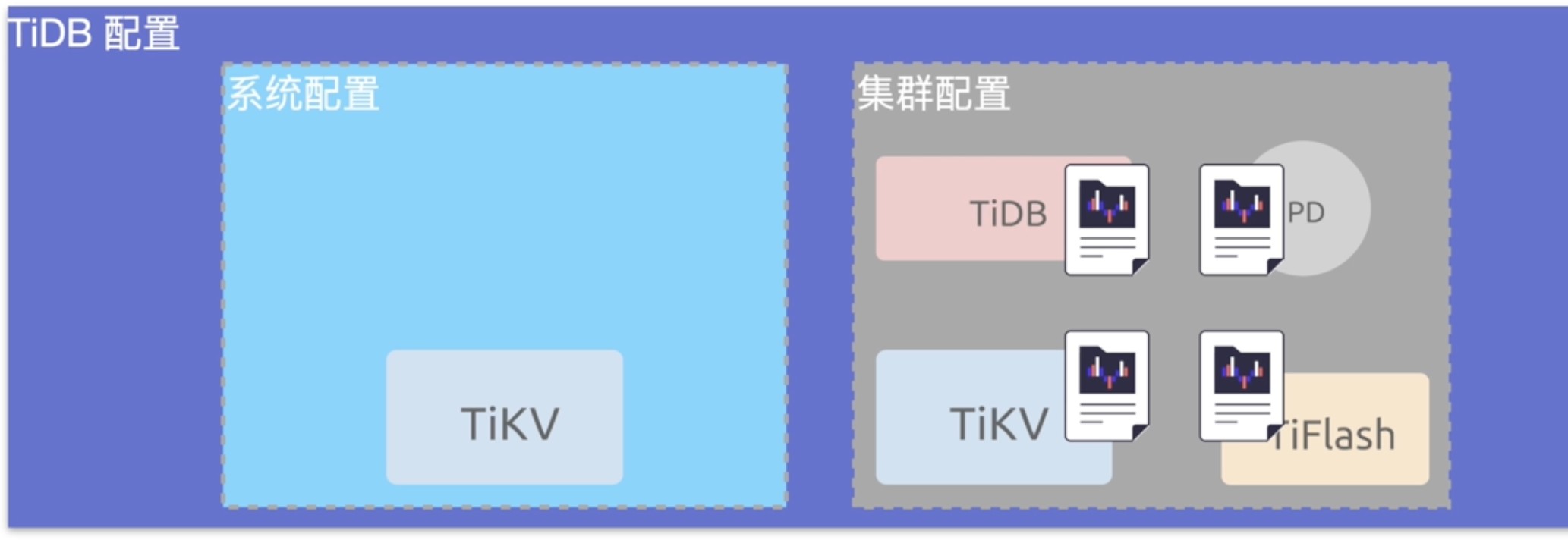
系统配置+集群配置

1.系统配置:持久化在tikv中

- ⼀部分在tikv中

* 有些专指tidb-server参数
* 不需要重启可持久化(set global xx = '';)
* 有作⽤域: global session instance(实例级别)
  + set @@global.tidb\_distsql\_scan\_concurrency = 10
  + set global tidb\_distsql\_scan\_concurrency = 10

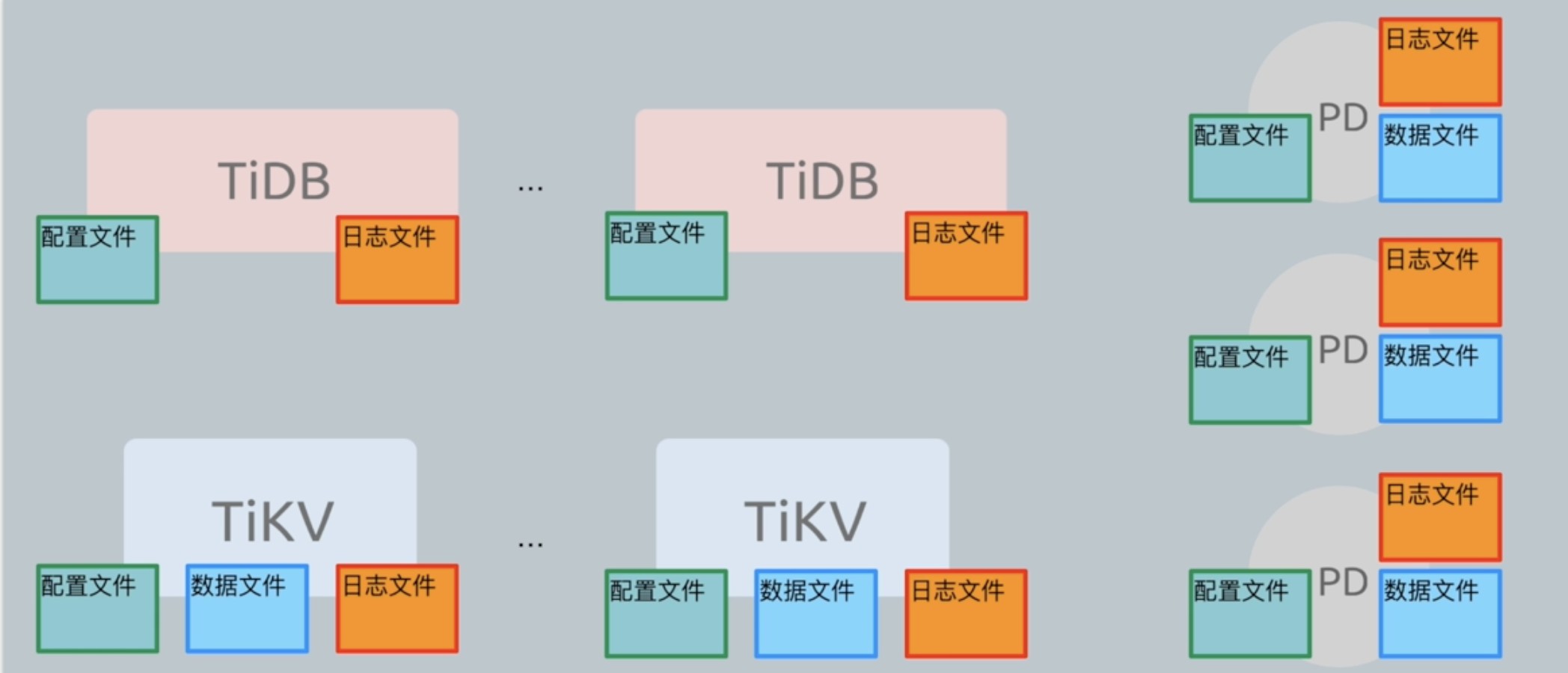
1. 集群配置:修改后需要reload 重启节点
   * 都在配置⽂件中
   * 修改后必须重启节点 tiup edit-config and tiup reload



* 1. **TiDB**数据库⽂件

数据⽂件

* 配置⽂件
* ⽇志⽂件
* 脚本命令⽂件
* 数据⽂件



1. **TiDB**集群管理

# 在线扩容

1. 扩容tidb节点

1.配置⽂件

vi scale-out-tidb.yaml

1. 执⾏扩容命令与检查 tiup cluster check

tiup cluster scale-out tidb\_online\_10100 ./scale-out-tidb.yaml

1. 检查

tiup cluster display tidb\_online\_10100 tidb属于⽆状态节点,扩容⽴⻢⽣效

1. 扩容tikv,tiflash节点

1.配置⽂件

vi scale-out-tikv.yaml

1. 执⾏扩容命令与检查 tiup cluster check

tiup cluster scale-out tidb\_online\_10100 ./scale-out-tikv.yaml

1. 检查

tiup cluster display tidb\_online\_10100

tikv有状态节点,扩容后均衡集群数据,通过pd调度⽣效

# 在线缩容

1. tidb节点缩容
2. LB中摘除节点
3. 确定已有业务流量降为0
4. 执⾏缩容

tiup cluster scale-in tidb\_online\_10100 --node xx

1. pd节点缩容

1.确认不是leader 2.执⾏缩容

tiup cluster scale-in tidb\_online\_10100 --node xx

4.tikv,tiflash缩容

1. 调整副本数为0

ALTER TABLE aisp\_community.t\_product\_browse SET TIFLASH REPLICA 2;

1. 确认删除

select \* from INFORMATION\_SCHEMA.TIFLASH\_REPLICA where TABLE\_NAME='t\_product\_browse' ;

1. 缩容节点

tiup cluster scale-in tidb\_online\_10100 --node xx

1. 查看集群状态

tiup cluster display tidb\_online\_10100

等待节点为Tome状态

# 集群销毁

tiup cluster destory cluster-name

* 1. 集群配置修改

1.查看集群配置

show config;

show config where type='tidb'

show config where instance in (...) show config where name like '%log%'

show config where type='tikv' and name='log.level'

1. 直接修改,节点重启失效

set config tikv `split.qps-threshold`=1000 #修改tikv配置

1. 永久修改

tiup cluster edit-config xxx

tiup cluster reload xxx

1. **TiDB**监控体系
   1. **Prometheus+Grafana**
      1. 基本介绍

利⽤Promethus原⽣监控,多个监控⾯板组成,各个⾯板明命名

1. 重要⾯板

Overview：重要组件监控概览。 Node\_exporter:监控主机。 TiDB：TiDB server 组件详细监控项。

TiKV-Summary：TiKV server 监控项概览。

TiKV-Trouble-Shooting：TiKV 错误诊断相关监控项。 TiKV-Details：TiKV server 组件详细监控项。 PD：PD server 组件相关监控项。

Disk-Performance：磁盘性能相关监控项。

1. 次级

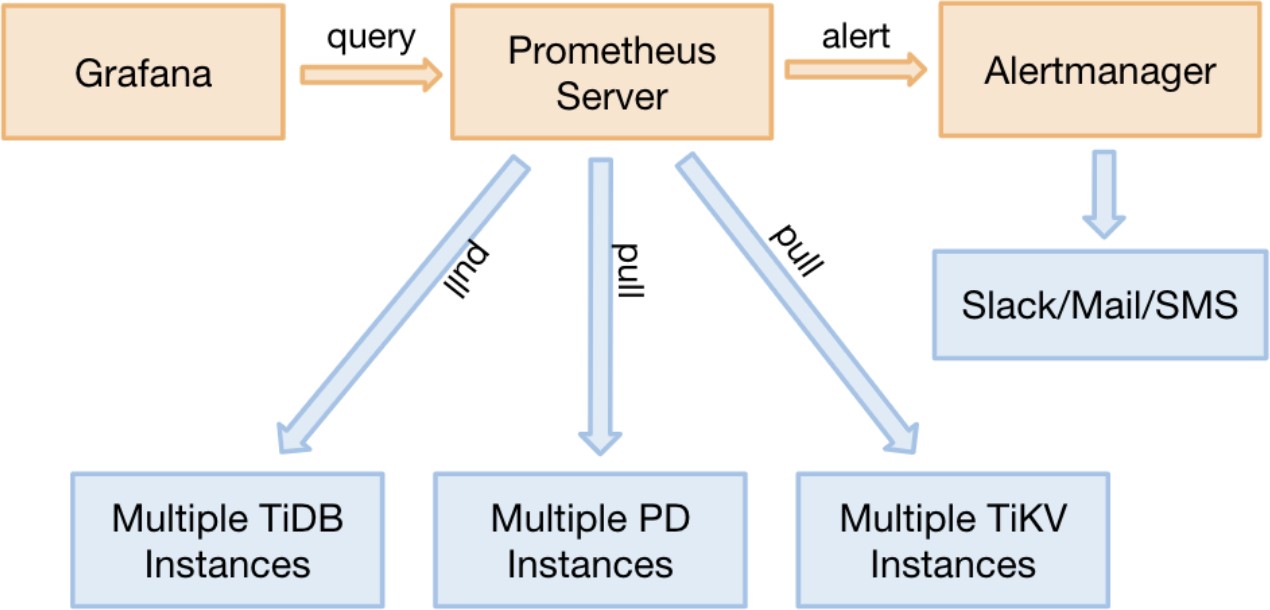
TiDB-Summary：TiDB server 相关监控项概览。 Performance-Read：读性能相关监控项。 Performance-Write：写性能相关监控项。

TiFlash-Summary：TiFlash server 相关监控项概览。

TiCDC 组件详细监控项。

1. 功能性

Backup-Restore：备份恢复相关的监控项。 Binlog：TiDB Binlog 相关的监控项。 Blackbox\_exporter：⽹络探活相关监控项。 Kafka-Overview：Kafka 相关监控项。 Lightning：TiDB Lightning 组件相关监控项。



* + 1. 监控指标**Overview**

<https://docs.pingcap.com/zh/tidb/stable/grafana-overview-dashboard>

#重要指标-PD

Region heartbeat report：TiKV 向 PD 发送的⼼跳个数

99% Region heartbeat latency：99% 的情况下，⼼跳的延迟

Hot write Region's leader distribution：每个 TiKV 实例上是写⼊热点的 leader 的数量

Hot read Region's leader distribution：每个 TiKV 实例上是读取热点的 leader 的数量

##重要指标-TiDB

Statement OPS：不同类型 SQL 语句每秒执⾏的数量。按 SELECT、INSERT、UPDATE 等来统计

Duration：执⾏的时间

CPS By Instance：每个 TiDB 实例上的命令统计。按照命令和执⾏结果成功或失败来统计

Failed Query OPM：每个 TiDB 实例上，每秒钟执⾏ SQL 语句发⽣错误按照错误类型的统计（例如语法错误、主键冲突等）。包含了错误所属的模块和错误码

Lock Resolve OPS：TiDB 清理锁操作的数量。当 TiDB 的读写请求遇到锁时，会尝试进⾏锁清理

TiClient Region Error OPS：TiKV 返回 Region 相关错误信息的数量

Memory Usage：每个 TiDB 实例的内存使⽤统计，分为进程占⽤内存和 Golang 在堆上申请的内存

#TiKV

leader：各个 TiKV 节点上 Leader 的数量分布

region：各个 TiKV 节点上 Region 的数量分布

CPU：各个 TiKV 节点的 CPU 使⽤率

raft store CPU：raftstore 线程的 CPU 使⽤率，线程数量默认为 2

#System Info

IO Util：磁盘使⽤率，最⾼为 100%，⼀般到 80% - 90% 就需要考虑加节点

CPU Usage：CPU 使⽤率，最⼤为 100%

Memory：内存总⼤⼩

* + 1. 监控指标**TiDB**

<https://docs.pingcap.com/zh/tidb/stable/grafana-tidb-dashboard>

TiDB为例

#Query Summary

Command Per Second：TiDB 按照执⾏结果成功或失败来统计每秒处理的命令数。

Slow query：慢查询的处理时间（整个慢查询耗时、Coprocessor 耗时、Coprocessor 调度等待时间），慢查询分为 inte rnal 和 general SQL 语句。

999/99/95/80 Duration：不同类型的 SQL 语句执⾏耗时（不同百分位）。

#Query Detail

Duration 80/95/99/999 By Instance：每个 TiDB 实例执⾏ SQL 语句的耗时（不同百分位）。

Failed Query OPM Detail：每个 TiDB 实例执⾏ SQL 语句发⽣的错误按照错误类型统计（例如语法错误、主键冲突等）

#Server

Disconnection Count：每个 TiDB 实例断开连接的数量

Event OPM：每个 TiDB 实例关键事件，例如 start，close，graceful-shutdown，kill，hang 等。

Panic And Critical Error：TiDB 中出现的 Panic、Critical Error 数量。

Client Data Traffic：TiDB 和客户端的数据流量。

#Transaction

Transaction Retry Num：事务重试次数

Session Retry Error OPS：事务重试时每秒遇到的错误数量，分为重试失败和超过最⼤重试次数两种类型

Commit Token Wait Duration：事务提交时的流控队列等待时间。当出现较⻓等待时，代表提交事务过⼤，正在限流。如果系统还有资源可以使⽤，可以通过增⼤ TiDB 配置⽂件中 committer-concurrency 值来加速提交

#Executor

Expensive Executor OPS：每秒消耗系统资源⽐较多的算⼦。包括 Merge Join、Hash Join、Index Look Up Join、Has h Agg、Stream Agg、Sort、TopN 等。

#Distsql

#KV Errors

KV Backoff Duration：KV 每个请求重试的总时间。TiDB 向 TiKV 的请求都有重试机制，这⾥统计的是向 TiKV 发送请求时遇到错误重试的总时间

TiClient Region Error OPS：TiKV 返回 Region 相关错误信息的数量

KV Backoff OPS：TiKV 返回错误信息的数量

Lock Resolve OPS：TiDB 清理锁操作的数量。当 TiDB 的读写请求遇到锁时，会尝试进⾏锁清理

Other Errors OPS：其他类型的错误数量，包括清锁和更新 SafePoint

#PD Client

PD Client CMD Duration：PD Client 执⾏命令耗时

PD Client CMD Fail OPS：PD Client 每秒执⾏命令失败的数量

#Schema Load

Schema Lease Error OPM：Schema Lease 出错统计，包括 change 和 outdate 两种，change 代表 schema 发⽣了变化，outdate 代表⽆法更新 schema，属于较严重错误，出现 outdate 错误时会报警

#DDL

DDL Waiting Jobs Count：等待的 DDL 任务数量

DDL add index progress in percentage：添加索引的进度展⽰

#Statistics

Stats Inaccuracy Rate：统计信息不准确度

#Meta

Meta Operations Duration 99：元数据操作延迟

#GC

#Batch Client

Pending Request Count by TiKV：TiKV 批量消息处理的等待数量 Batch Client Unavailable Duration 95：批处理客户端的不可⽤时⻓。 No Available Connection Counter：批处理客户端不可⽤的连接数。

* + 1. **PD**

#基础

Region health：集群所有 Region 的状态。通常情况下，pending 或 down 的 peer 应该少于 100，miss 的 peer

不能⼀直⼤于 0，empty Region 过多需及时打开 Region Merge

Abnormal stores：处于异常状态的节点数⽬，正常情况应当为 0

#Cluster

PD scheduler config：PD 调度配置列表

#Operator

Schedule operator timeout：已超时的 operator 的数量

Schedule operator replaced or canceled：已取消或者被替换的 operator 的数量

#Statistics - hot write

Hot Region's leader distribution：每个 TiKV 实例上成为写⼊热点的 leader 的数量

Total written bytes on hot leader Regions：每个 TiKV 实例上所有成为写⼊热点的 leader 的总的写⼊流量⼤⼩

#Scheduler

Balance leader movement：leader 移动的详细情况

Balance Region movement：Region 移动的详细情况

#Heartbeat

Region heartbeat report error：TiKV 向 PD 发送的异常的⼼跳个数

99% Region heartbeat latency：99% 的情况下，⼼跳的延迟

* + 1. **TiKV-Details**

#Errors

Critical error：严重错误的数量

Server is busy：各种会导致 TiKV 实例暂时不可⽤的事件个数，如 write stall，channel full 等，正常情况下应当为 0

Server report failures：server 报错的消息个数，正常情况下应当为 0 Raftstore error：每个 TiKV 实例上 raftstore 发⽣错误的个数 Scheduler error：每个 TiKV 实例上 scheduler 发⽣错误的个数 Coprocessor error：每个 TiKV 实例上 coprocessor 发⽣错误的个数 gRPC message error：每个 TiKV 实例上 gRPC 消息发⽣错误的个数 Leader drop：每个 TiKV 实例上 drop leader 的个数

Leader missing：每个 TiKV 实例上 missing leader 的个数

#Thread CPU

Raft store CPU：raftstore 线程的 CPU 使⽤率，通常应低于 80%

Coprocessor CPU：coprocessor 线程的 CPU 使⽤率

#Scheduler

Scheduler pending commands：每个 TiKV 实例上 pending 命令的 ops

#Scheduler - prewrite

Scheduler stage total：prewrite 中每个命令所处不同阶段的 ops，正常情况下，不会在短时间内出现⼤量的错误

#Scheduler - rollback

Scheduler stage total：rollback 中每个命令所处不同阶段的 ops，正常情况下，不会在短时间内出现⼤量的错误

#Task

Worker pending tasks：当前 worker 中，每秒钟 pending 和 running 的任务的数量，正常情况下，应该⼩于 1000

#Coprocessor Overview

Total Request Errors：Coprocessor 每秒请求错误的数量，正常情况下，短时间内不应该有⼤量的错误

Total KV Cursor Operations：各种类型的 KV cursor 操作的总数量的 ops，例如 select、index、analyze\_table、 analyze\_index、checksum\_table、checksum\_index 等

#Lock manager

Waiter lifetime duration：事务等待锁释放的时间

Wait table：wait table 的状态信息，包括锁的数量和等锁事务的数量

Deadlock detect duration：处理死锁检测请求的耗时 Detect error：死锁检测遇到的错误数量，包含死锁的数量

Deadlock detector leader：死锁检测器 leader 所在节点的信息

* 1. **Dashboard**

