KeyViz: 睁开双眼看业务

Ed Huang

关于我

- 黄东旭
- Co-founder & CTO, PingCAP
- 分布式系统工程师 / 开源信仰者
- Golang / Rust
- Ti{DB, KV} / Codis
- MSRA -> Netease -> WandouLabs -> PingCAP
- Beijing ⇔ SF
- h@pingcap.com

插播一个广告

5 月末 TiDB 4.0 发布

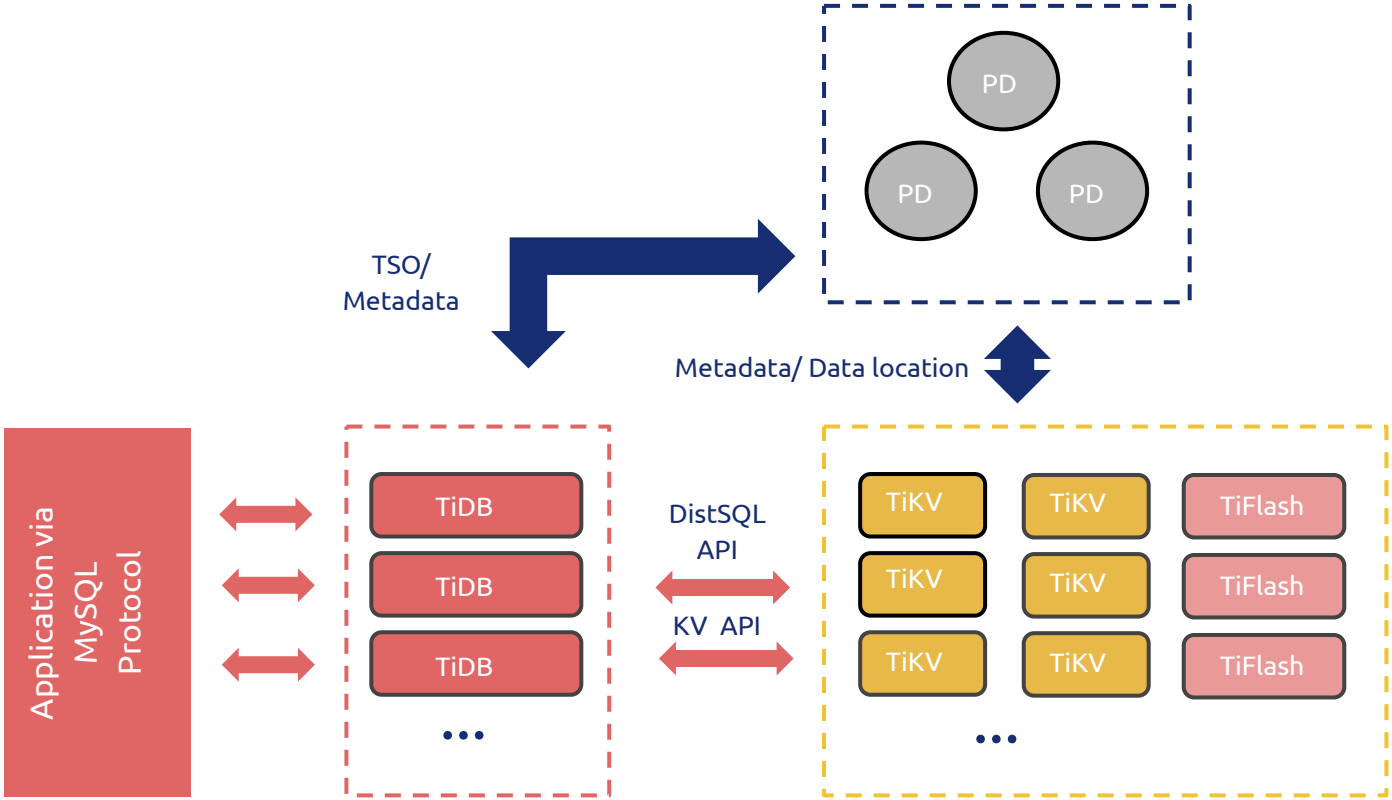
（Hopefully）

TiDB 101

- TiDB 是一个分布式数据库（废话）
- TiDB 支持 SQL（以 MySQL 的协议）
  - ACID 事务，二级索引，Online DDL
  - 弹性伸缩，不需要分库分表！
  - 强一致高可用
- TiDB 是一个 **HTAP** 数据库
  - TiFlash：列式存储 (支持**实时更新以及事务隔离级别**的列式存储！)
  - TiKV：行式存储
  - TiDB SQL：计算 - 存储分离的分布式SQL层

TiDB 101

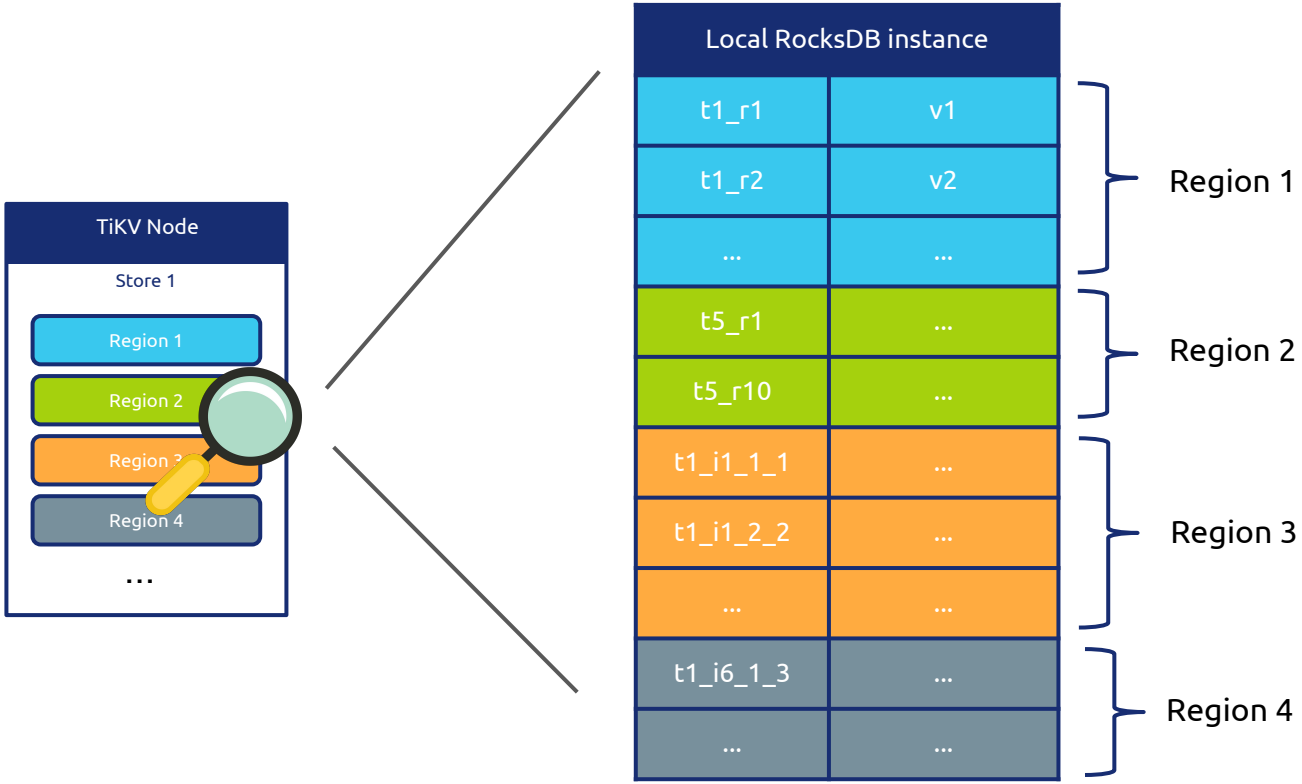
- TiDB
- TiKV
- PD
- TiFlash



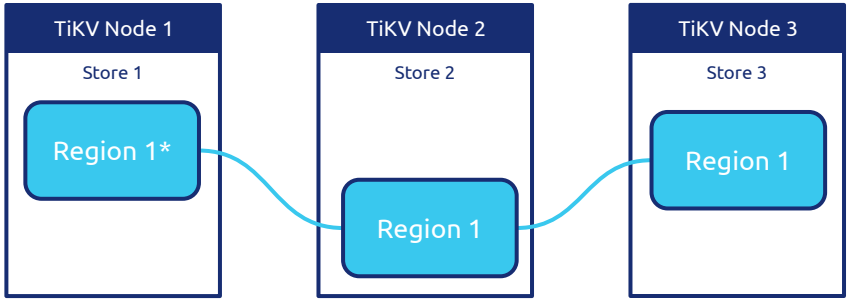
TiDB 101

- 计算存储高度分离
  - SQL 层是无状态的
  - 数据其实存储在一个名为 TiKV 的分布式 K-V 数据库上（也是我们做的）
  - 这些 KV 的分布信息是存储在一个叫 PD 的元信息模块中

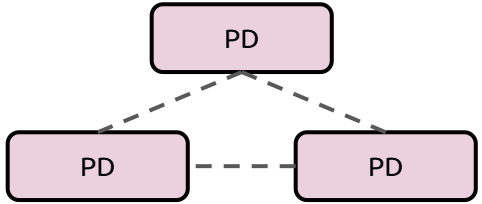
TiDB 101


TiKV Node
Store 1
Region 1
Region 2
Region 3
Region 4
...
Local RocksDB instance
t1_r1
v1
t1_r2
v2
...
...
t5_r1
...
t5_r10
...
t1_i1_1_1
...
t1_i1_2_2
...
...
...
t1_i6_1_3
...
...
...
Region 1
Region 2
Region 3
Region 4

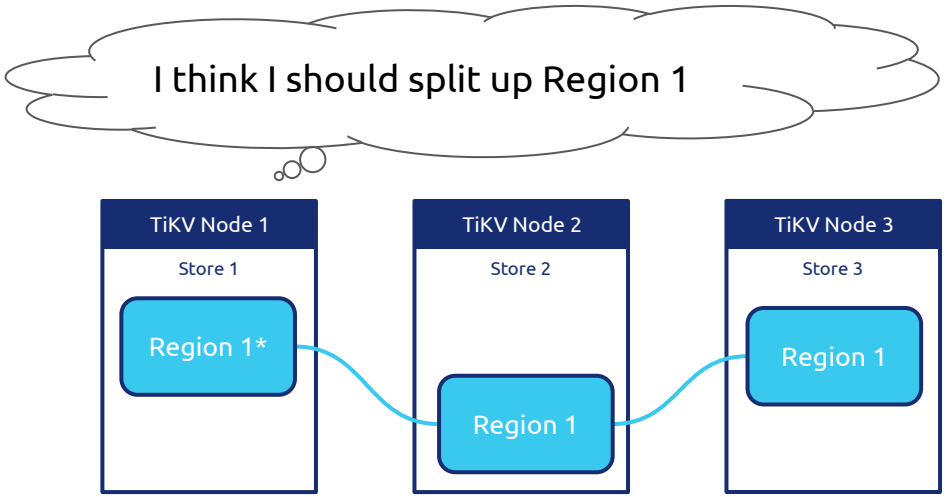
How scale-out works inside TiDB



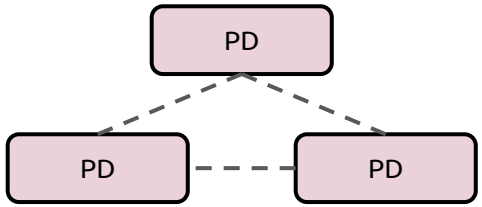
Let's say, the amount of data within Region 1 exceeds the threshold (default: 96MB)



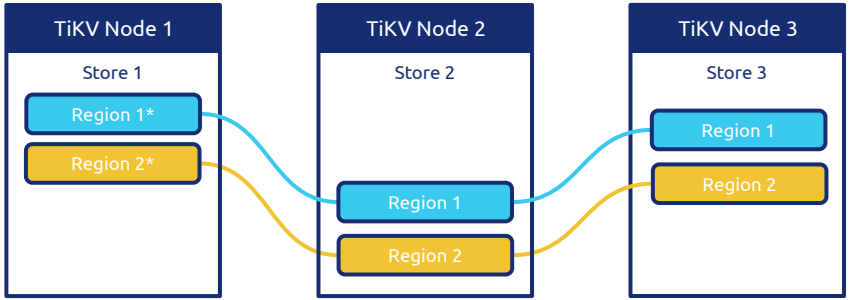
How scale-out works inside TiDB



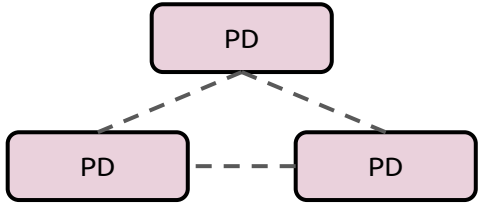
Let's say, the amount of data within Region 1 exceeds the threshold (default: 96MB)



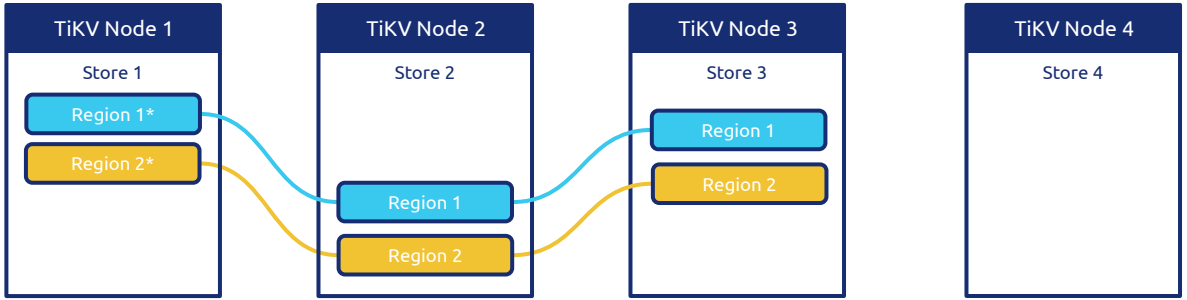
How scale-out works inside TiDB



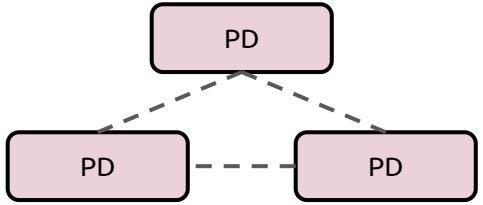
Region 1 will be split into two smaller regions.
(the leader of Region 1 sends a Split command as a special log to its replicas via the Raft protocol.
Once the Split command is successfully committed by Raft, that means the region has been successfully split.)



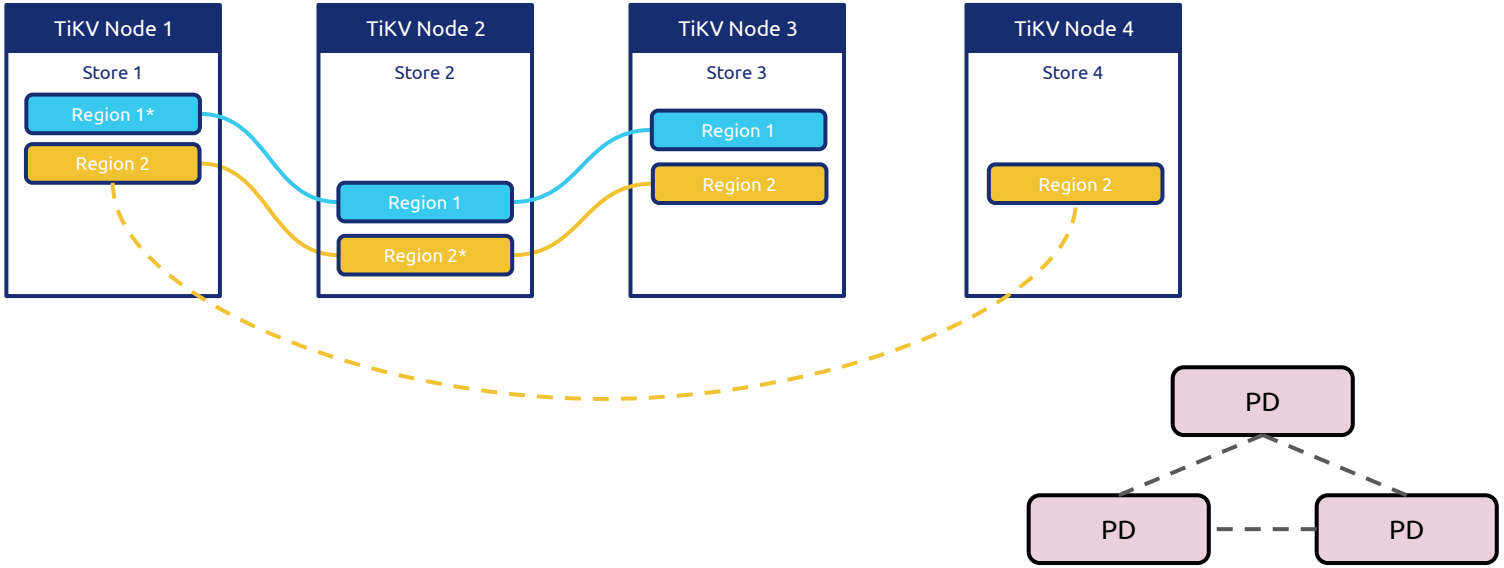
How scale-out works inside TiDB



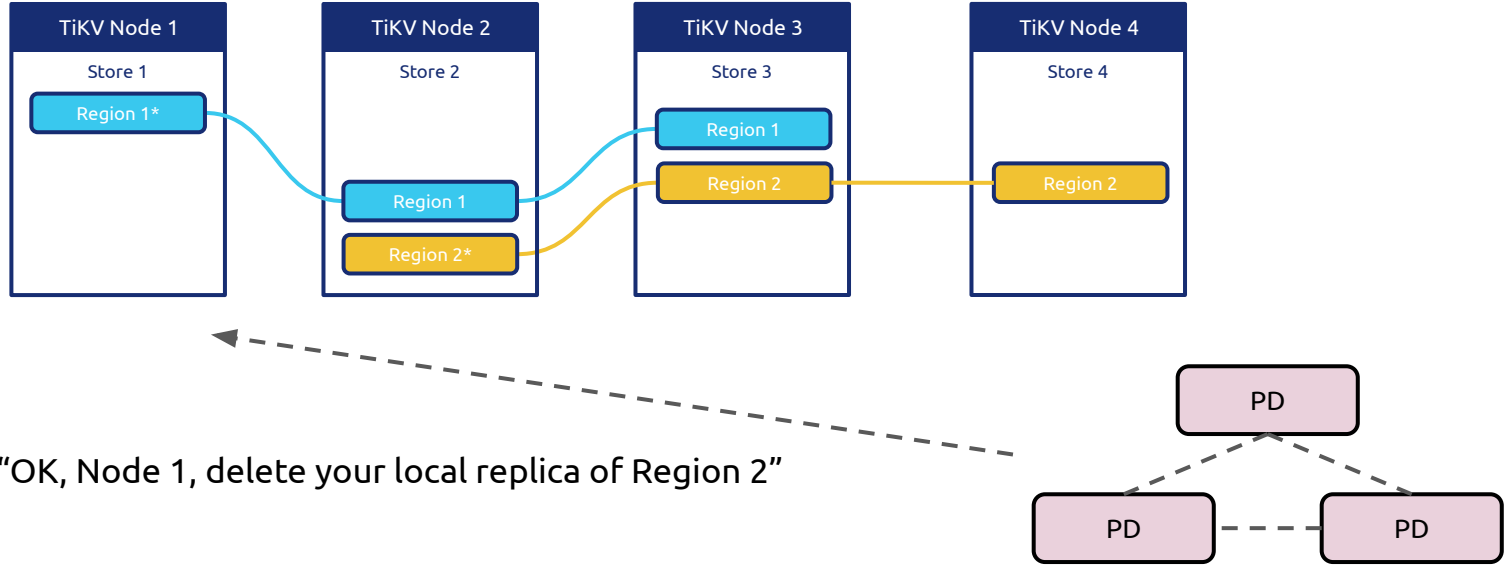
PD: “Hey, Node1, create a new replica of Region 2 in Node 4, and transfer your leadership of Region 2 to Node 2”



How scale-out works inside TiDB



How scale-out works inside TiDB



PD: “OK, Node 1, delete your local replica of Region 2”

为什么分布式数据库更需要观测？

为什么分布式数据库更需要观测？

- 状态更加分散
- 拓扑更加复杂
- 数据量和流量的分布不确定
- QPS 和 TPS 其实并不一定能反映问题

回想一下过去我们怎么查问题

tidb	142	10.6%
dm	120	30%
tikv	112	27.7%
热点	87	49.5%
binlog	74	28.4%

来，我们试试。。。

监控 vs 观测

Understanding

Unknown Knowns

- Things we understand but are not aware of
- "We implemented an orchestrator to ensure the system is always running"

Known Knowns

- Things we are aware of and understand
- "The system has a 1GB memory limit"

Unknown Unknowns

- Things we are neither aware of nor understand
- "Instances churn because the orchestrator restarts the process when it approaches its memory limit, causing sporadic failures and slowdowns"

Known Unknowns

- Things we are aware of but don't understand
- "The system exceeded its memory limit and crashed, causing an outage"

Data Available

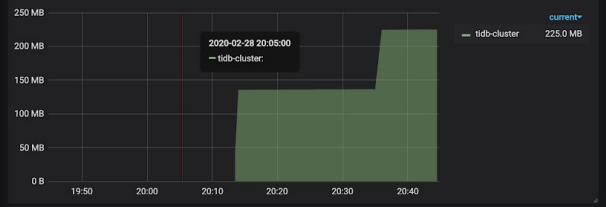
tidb18 > Test-Cluster-TiKV

Last 1 hour | 1m | Report

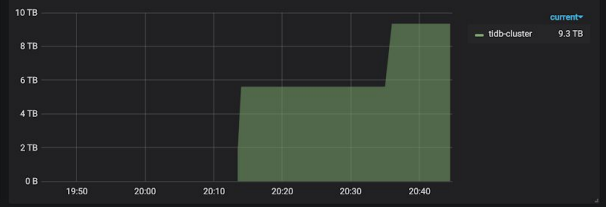
db All | command All

Cluster

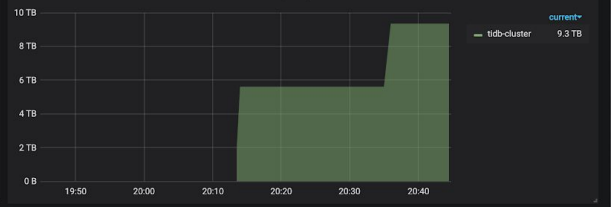
store size



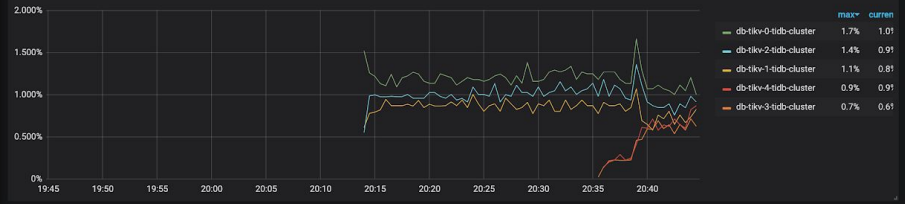
available size



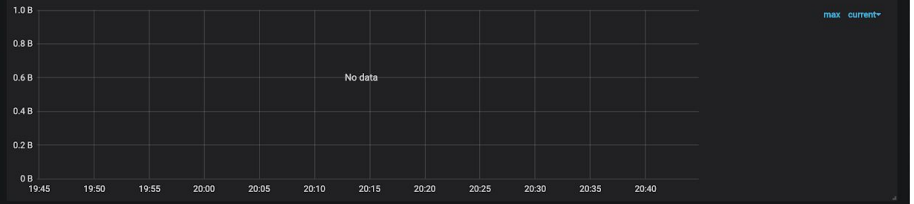
capacity size



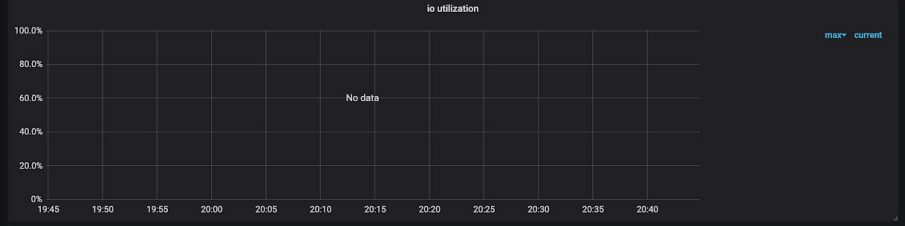
cpu



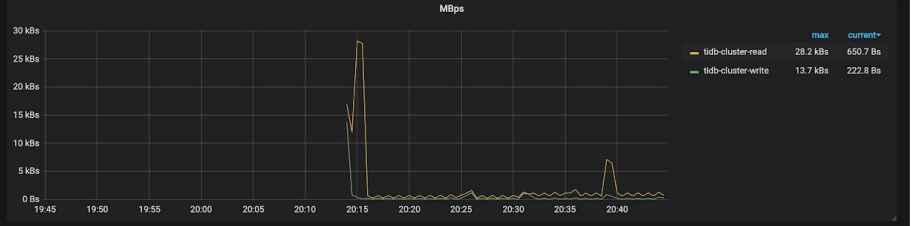
memory



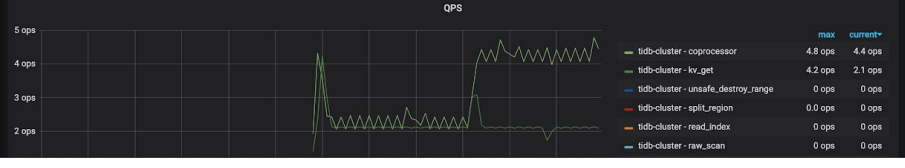
io utilization



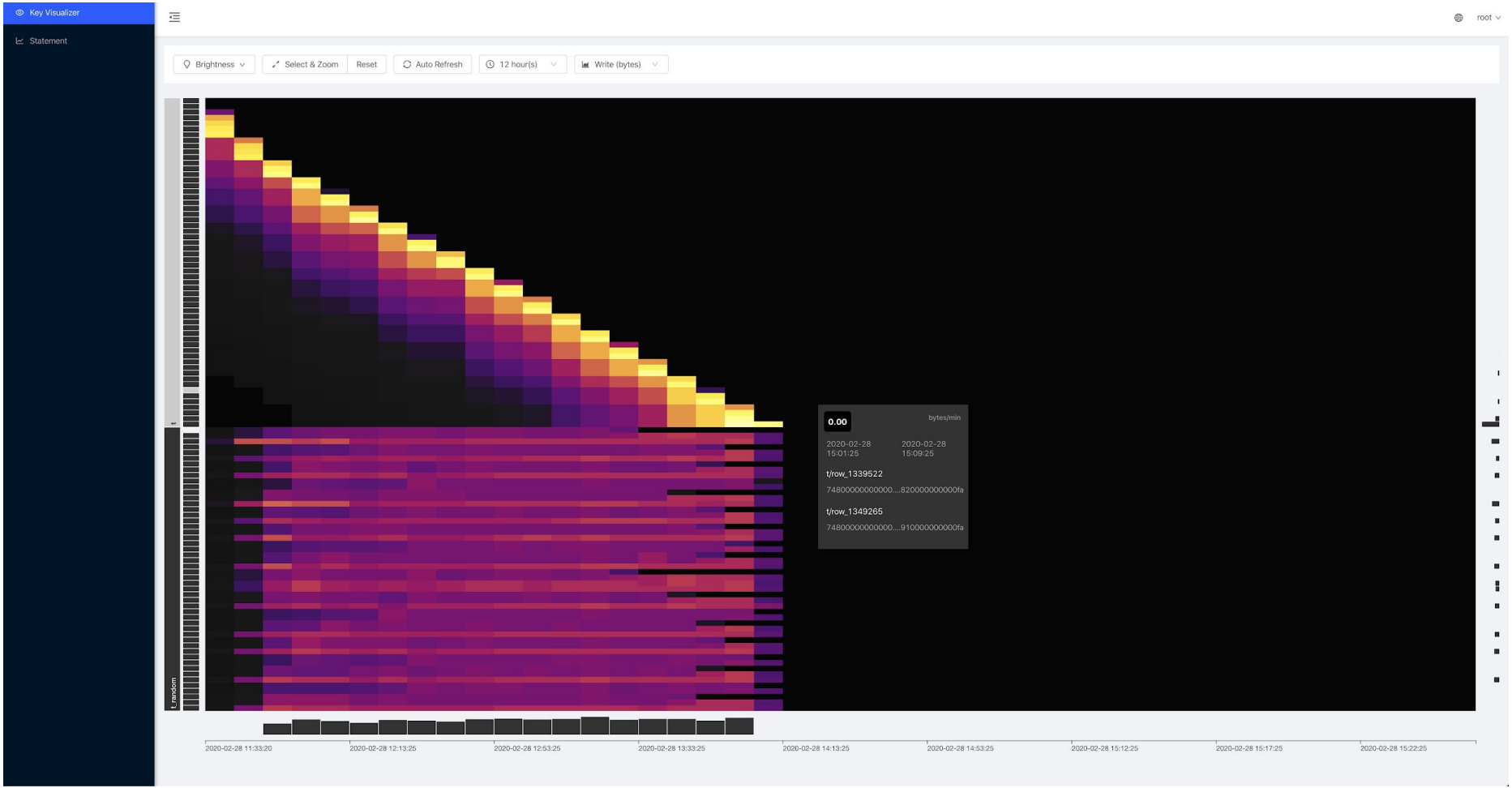
MBps



QPS



Errps


Key Visualizer
Statement
root
Brightness
Select & Zoom
Reset
Auto Refresh
12 hour(s)
Write (bytes)
t
t_random
0.00
bytes/min
2020-02-28 15:01:25
2020-02-28 15:09:25
t/row_1339522
74800000000000...820000000000fa
t/row_1349265
74800000000000...910000000000fa
2020-02-28 11:33:20
2020-02-28 12:13:25
2020-02-28 12:53:25
2020-02-28 13:33:25
2020-02-28 14:13:25
2020-02-28 14:53:25
2020-02-28 15:12:25
2020-02-28 15:17:25
2020-02-28 15:22:25

KeyViz 的原理？

- X 轴：时间轴
- Y 轴：Key轴
- 颜色：热度

回想一下业务和 DBA 沟通的一个典型场景

自增主键 or Random ID？
读写比例多少？
操作数据有没有特征？

看到了热点。。。然后呢?

如何有效避免热点？

- 重新设计你的主键
- TiDB Partition
- SHARD_ROW_ID_BITS

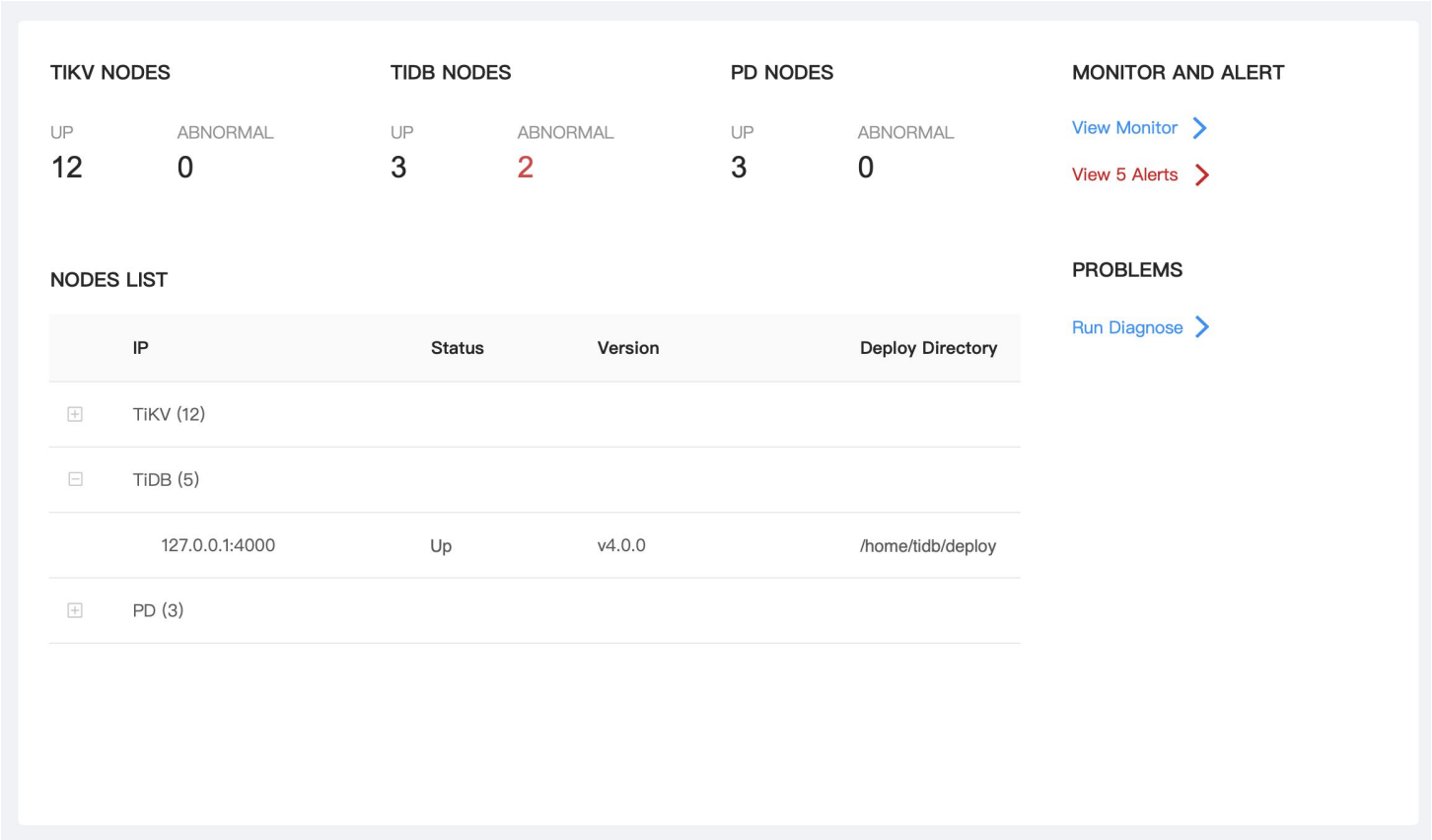
In 4.0:

```
CREATE TABLE operation_log (
    id serial PRIMARY KEY auto_random,
    account_id INTEGER NOT NULL,
    type ENUM('INSERT', 'UPDATE', 'SELECT', 'DELETE') NOT NULL,
    detail LONGTEXT
);
```

可观测性在 TiDB 4.0 还有哪些体现？

TiDB Dashboard

Overview


TIKV NODES
UP
12
ABNORMAL
0
TIDB NODES
UP
3
ABNORMAL
2
PD NODES
UP
3
ABNORMAL
0
MONITOR AND ALERT
View Monitor
View 5 Alerts
NODES LIST
IP
Status
Version
Deploy Directory
TiKV (12)
TiDB (5)
127.0.0.1:4000
Up
v4.0.0
/home/tidb/deploy
PD (3)
PROBLEMS
Run Diagnose

Statements

Statements Overview

2020-02-28 23:00:00 ~ 2020-02-28 23:30:00 | Select schemas

Schema	SQL Category	Sum Latency	Exec Count	Avg Affected Rows	Avg Latency	Avg Cost Memory
	select high_priority * from mysql . global_variables where variable_name in (...)	18.18 ms	3	0	6.06 ms	0 B
	insert high_priority into mysql . tidb values (...) on duplicate key update variable_value = ? , comment = ?	15.99 ms	5	2	3.20 ms	406.00 B
	select original_sql , bind_sql , default_db , status , create_time , update_time , charset , collation from mysql . bind_info where update_time > ? order by update_time	11.33 ms	7	0	1.62 ms	30.68 KiB

Statements

root

Statements Overview / Statement Detail

Schema	
Time Range	2020-02-25 00:30:00 ~ 2020-02-25 01:00:00
SQL Category	select high_priority (variable_value) from mysql . tidb where variable_name = ? for update
Last SQL Statement	SELECT HIGH_PRIORITY (variable_value) FROM mysql.tidb WHERE variable_name='tikv_gc_leader_uuid' FOR UPDATE
Last Seen	2020-02-25 00:56:10

Sum Latency: 321.20 ms

Exec Count: 110

Avg Affected Rows: 0

Avg Scan Rows: 1

Node	Sum Latency	Exec Count	Avg Latency	Max Latency	Avg Cost Memory	Sum Backoff Times
127.0.0.1:10080	321.20 ms	110	2.92 ms	5.29 ms	10.51 KiB	0

Diagnosis Report

TiDB Dashboard | Diagnose Report

127.0.0.1:12333/dashboard/api/diagnose/reports/7

Incognito

TiDB SQL Diagnosis System Report

Expand All | Fold All

header

Report Time Range

START_TIME	END_TIME
2020-02-27 21:43:41	2020-02-28 21:43:41

cluster

The hardwareInfo of each node

HOST		PU_CORES	MEMORY (GB)	DISK (GB)	UPTIME (DAY)
localhost		0/0	0.000000		
127.0.0.1	tikv*1 pd*1	4/8	0.015625	/dev/disk1s4: 233.469135 /dev/disk1s1: 233.469135 /dev/disk1s5: 233.469135	
0.0.0.0	tidb*1	4/8	16.000000	/dev/disk1s4: 233.469135 /dev/disk1s5: 233.469135 /dev/disk1s1: 233.469135	

cluster info

TYPE	INSTANCE	STATUS_ADDRESS	VERSION	GIT_HASH	START_TIME	UPTIME
tidb	0.0.0.0:4000	0.0.0.0:10080	5.7.25-TiDB-ea04375a7	ea04375a76ba9f08c40e7e66a200e10b409b2cb9	2020-02-28T21:42:10+08:00	1m38.61108s
pd	127.0.0.1:2379	127.0.0.1:2379	4.1.0-alpha	6556145ad21b4ca68754bbf6296e8bd57261544b	2020-02-19T20:50:08+08:00	216h53m40.611085s
tikv	127.0.0.1:20160	127.0.0.1:20180	4.1.0-alpha	ce20c70c3c37c9b8bb76d0066334d44f5c0d8e21	2020-02-19T20:54:06+08:00	216h49m42.611086s

Diagnosis Report

127.0.0.1:12333/dashboard/api/diagnose/reports/7

Incognito

overview

Time Consume

The table contain the event time consume in TiDB/TiKV/PD. METRIC_NAME is the event name; LABEL is the event label, such as instance, event type ...; TIME_RATIO is the TOTAL_TIME of this event devide by the TOTAL_TIME of upper event which TIME_RATIO is 1; TOTAL_TIME is the total time cost of this event; TOTAL_COUNT is the total count of this event; P999 is the max time of 0.999 quantile; P99 is the max time of 0.99 quantile; P90 is the max time of 0.90 quantile; P80 is the max time of 0.80 quantile;

METRIC_NAME	LABEL	TIME_RATIO	TOTAL_TIME	TOTAL_COUNT	P999	P99	P90	P80
tidb_query fold		1	689.38	19902	3.87	2.04	1.95	1.84
|-- tidb_query	Select	0.83	569.74	5527	3.87	2.04	1.95	1.84
|-- tidb_query	internal	0.17	118.97	14118	2.02	1.8	0.06	0.05
|-- tidb_query	general	0.0005	0.37	193	0.03	0.03	0.02	0.01
|-- tidb_query	Show	0.0004	0.29	43	0.06	0.06	0.06	0.05
|-- tidb_query	Set	0.000007	0.005	21	0.004	0.004	0.004	0.003
tidb_get_token expand		0.000002	0.001	5787	0.0004	0.00003	0.000001	0.000001
tidb_parse expand		0.003	2.33	29477	0.008	0.003	0.002	0.002
tidb_compile expand		0.01	6.99	29477	0.02	0.006	0.005	0.004
tidb_execute expand		0.02	10.58	29477	0.2	0.13	0.006	0.005
tidb_distsql_execution expand		0.04	25.6	10518	2.03	1.87	0.05	0.03

Log Searching

Log Searching / Detail

Time Range: Start date ~ End date

Log Level: Warn

Components: 192.168.1.8:4000 × 192.168.1.8:20160 × 192.168.1.8:2379 ×

Keywords: Optional, separated by spaces Search

The preview shows only the first 500 logs

Time	Level	Component	Log
2020-01-07T11:34:58+08:00	Warn	PD	[store.go:1317] ["simple token is not cryptographically signed"]
2020-01-07T11:35:00+08:00	Warn	PD	[history_buffer.go:138] ["load history index failed"] [error="leveldb: not found"]
2020-01-07T11:50:00+08:00	Error	PD	[heartbeat_streams.go:121] ["send keepalive message fail"] [target-store-id=1] [error=EOF]
2020-01-07T14:22:43+08:00	Warn	PD	[grpclog.go:60] ["transport: http2Server.HandleStreams failed to read frame: read tcp 127.0.0.1:2379->...
2020-01-20T13:55:57+08:00	Warn	PD	[store.go:1317] ["simple token is not cryptographically signed"]
2020-01-20T13:55:59+08:00	Warn	PD	[history_buffer.go:138] ["load history index failed"] [error="leveldb: not found"]
2020-01-20T14:43:19+08:00	Warn	PD	[grpclog.go:60] ["transport: http2Server.HandleStreams failed to read frame: read tcp 127.0.0.1:2379->...
2020-01-20T14:43:19+08:00	Warn	PD	[grpclog.go:60] ["grpc: addrConn.createTransport failed to connect to {127.0.0.1:2379 0 <nil>}. Err :con...
2020-02-25T16:59:52+08:00	Warn	PD	[store.go:1317] ["simple token is not cryptographically signed"]

Searching progress

3 completed

Download selected Cancel

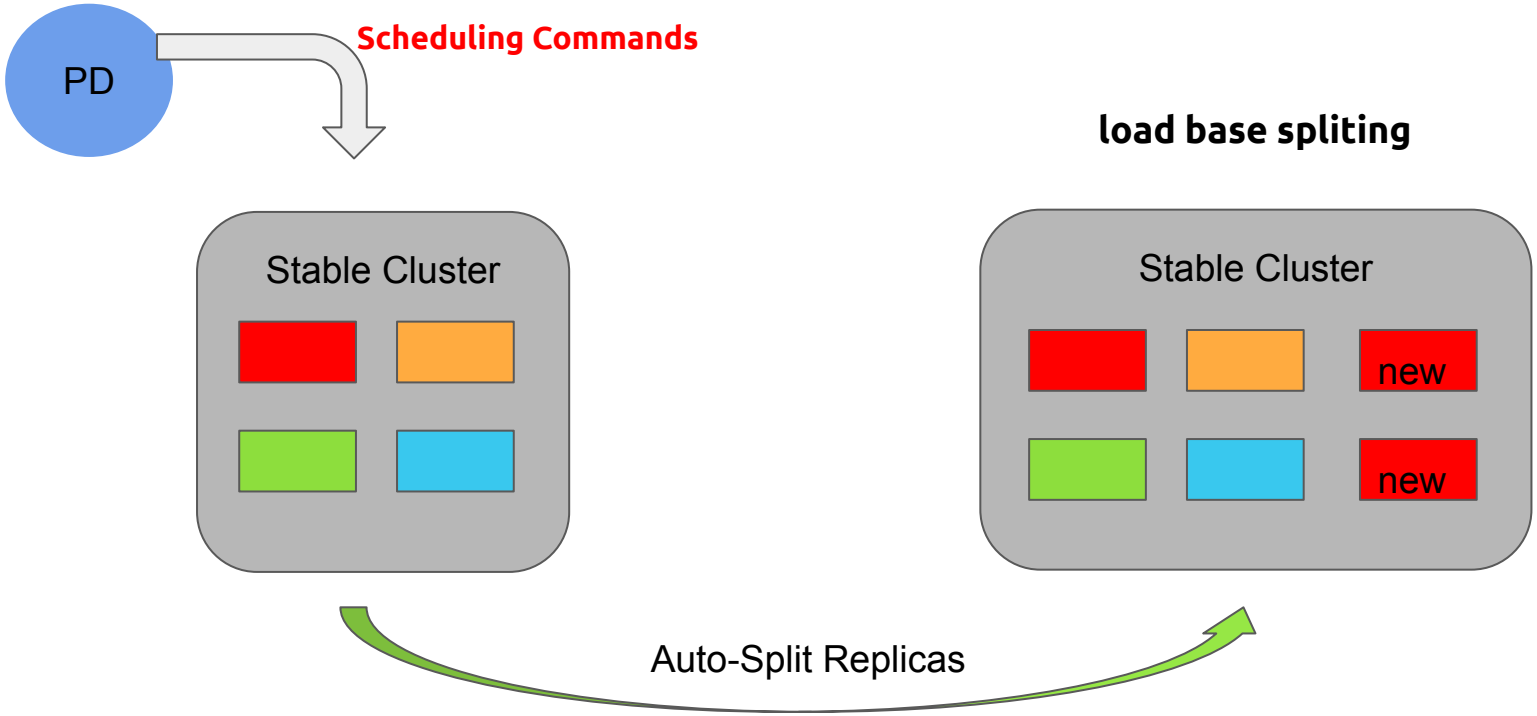
Retry

- TiDB 1 completed
  - 192.168.1.8:10080
- TiKV 1 completed
  - 192.168.1.8:20160
- PD 1 completed
  - 192.168.1.8:2379

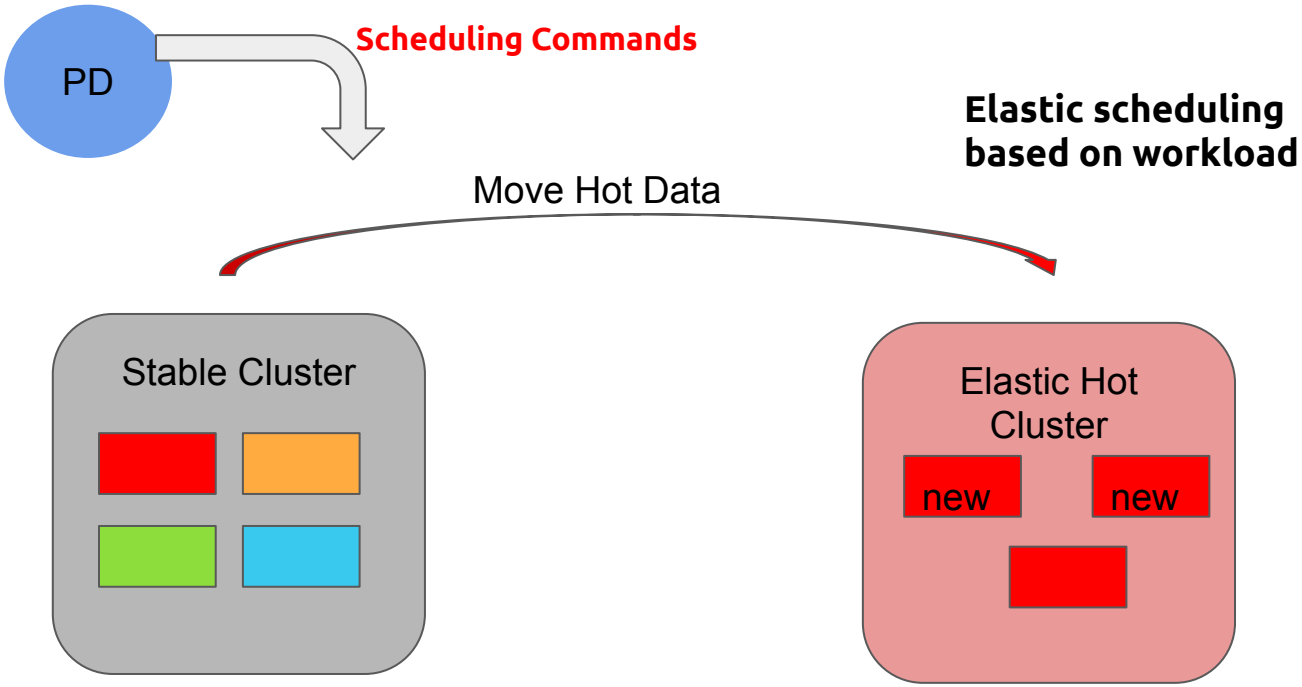
What's the future

我们可能做出了下一代数据库（？）的雏形？

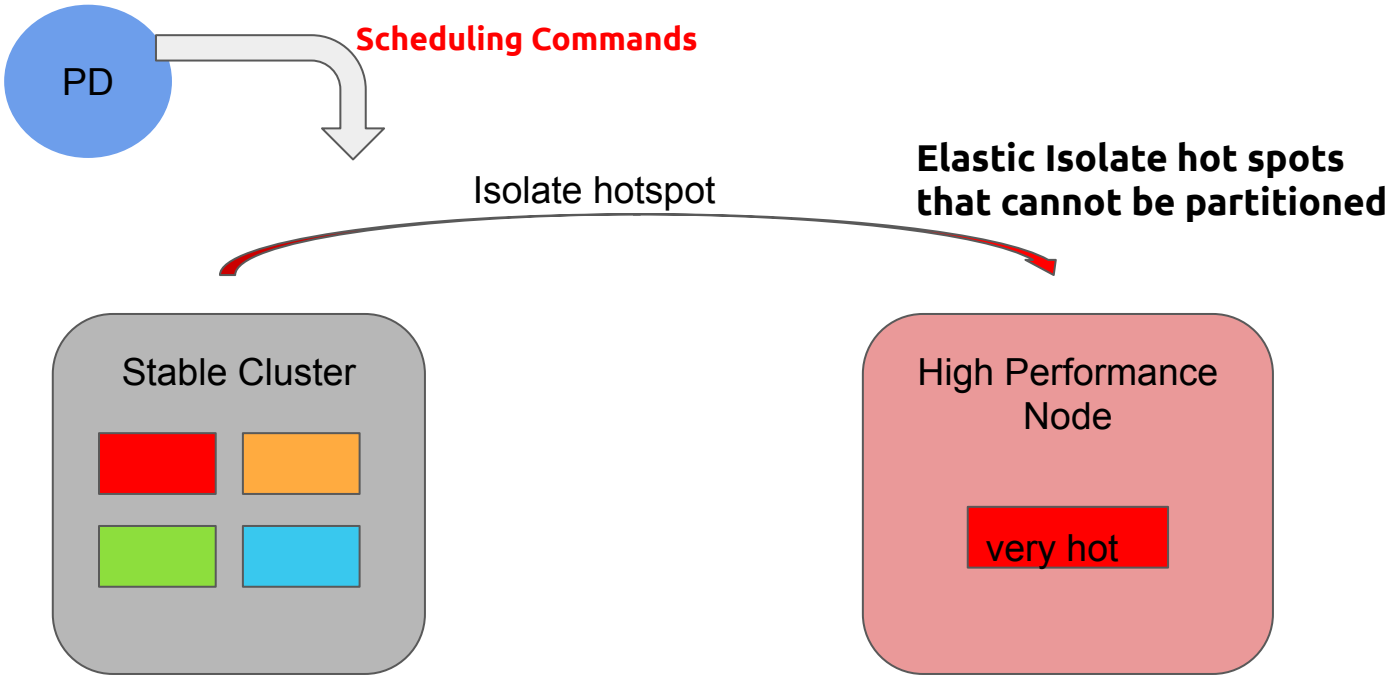
弹性调度 - 基于负载的分裂均衡及调整副本



弹性调度 - 自动节点扩充


PD
Scheduling Commands
Elastic scheduling
based on workload
Move Hot Data
Stable Cluster
Elastic Hot
Cluster
new
new

弹性调度 - 自主热点隔离


PD
Scheduling Commands
Isolate hotspot
Elastic Isolate hot spots
that cannot be partitioned
Stable Cluster
High Performance
Node
very hot

在更长远的未来...

- 数据库是否能够
  - 真正的根据 Workload 自动的决定存储介质？
  - 实时的跟踪热点并弹性的伸缩？
  - 跨数据中心，跨地域做到真正的高可用和根据 业务的数据智能分布？
  - ...
- 这一切背后的基础是：
  - 调度能力
- TiDB 在 4.0 中第一次拥有了这个能力的雏形，让我们期待 TiDB 展翅翱翔那一天

One more thing...

预告

Features	相
Key Visualization	
Cascading Placement Rules	
SQL Plan Managemen t	
Statements	
TiDB Dashboard	
SQL 诊断	

弹性调度
Unified thread pool
Follower read
Join中间数据实时写入本地临时盘
写入路径上的内存追踪
完善的 SQL 引擎 Hint
新热点调度器
Index Merge
Explain 格式优化
Chunk IPC
Index Join 优化
Enhanced Hint Set
Full Vectorized

Expression Evaluation
Expression Index
TiDB CI Collations
新的 row format
Coprocessor Cache 功能
优化 GC 性能，确保 GC 过程中不影响业务
悲观锁功能
大事务

支持添加/删除主键功能
支持 AutoRandom Key
BR
支持 Sequence 功能
支持快速恢复通过 Truncate Table 功能删除的数据
动态修改配置，配置移到内核
TPCC 性能提升 50%
4.0 重要特性概览介绍
TiDB/TiKV/PD 支持动态更新 TLS 证书
TiKV 支持磁盘数据加密（encryption at rest）
CDC

TiOps
DM

其他

产品
TiFlash
DBaaS

DBaaS...终于要见人了....

One more thing…

Or things? :)

第一期挑战赛 - **性能**挑战赛(2019.11.04 - 2020.02.04)

第二期挑战赛 - **易用性**挑战赛(2020.03.02 - 2020.06.02)

Talent Plan 2.0

「我们希望为中国大学的计算机教育做一些事情」

- 真的从零到一，教你写一个数据 库
  - 连编程语言都教！
  - 由浅入深地逐步了解分布式系 统和数据库的基础知识
  - 以 TiDB 为例子，深入了解数据 库的内部设计原理和源码解析

https://university.pingcap.com/talent-plan/

TiDB 4.0 是一个具有里程碑意义的产品
我们还能做点什么？

Book Rush!

一起做一件以前不敢想的事情

找个周末，我们 48 小时写本书怎么样

《TiDB in Action: 4.0+》

征集社区志愿者（30名），活动细则下周发布

署名社区，还会有特殊的纪念品