

TiDB VS MySQL 场景选择

li_zhenhuan 发表于 2024-04-02

原创

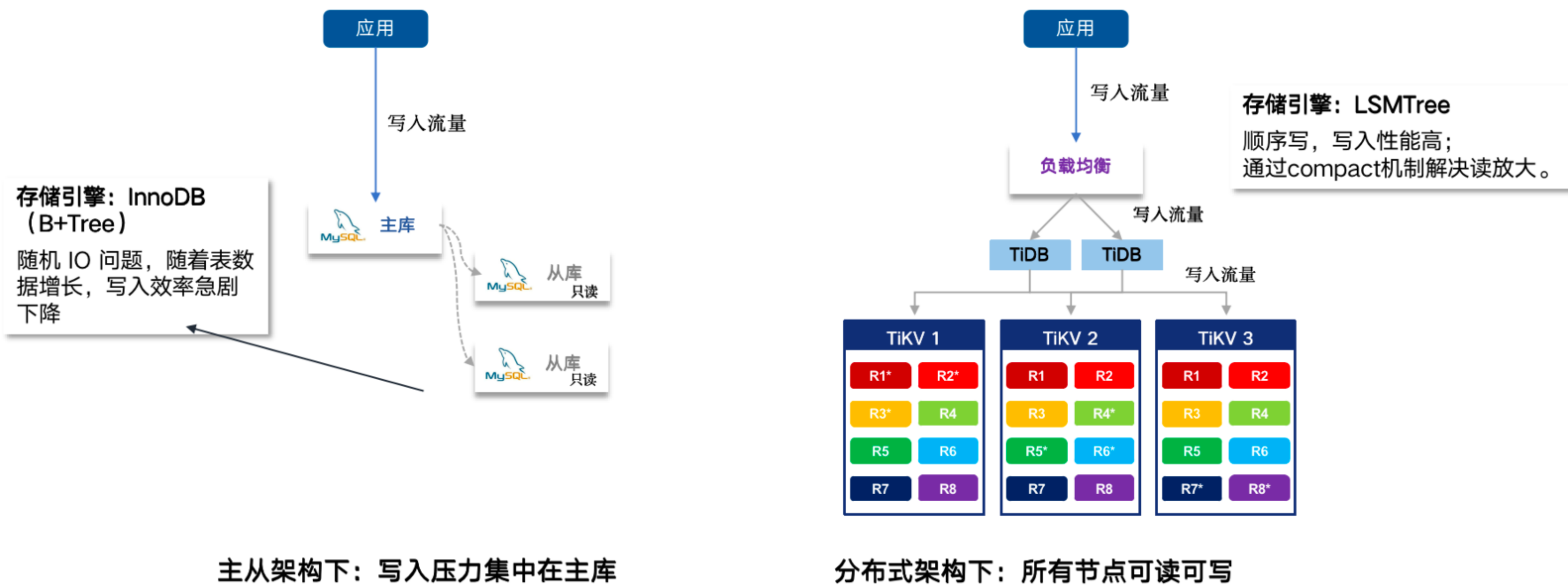
7.x 实践

TiDB VS MySQL 写入性能

数据量在百万级以下 MySQL 写入性能优于 TiDB，在数据量达到千万级以上 MySQL 写入性能更差，因为 MySQL 数据量大时无法打散均匀分布到其它服务器，导致 B+ 树高度过高进而影响写入性能。TiDB 数据量大时会均匀将数据分散在多台 TiKV 服务器，写入性能不会随着数据量增长而下降。读取性能类似

写入性能 vs MySQL

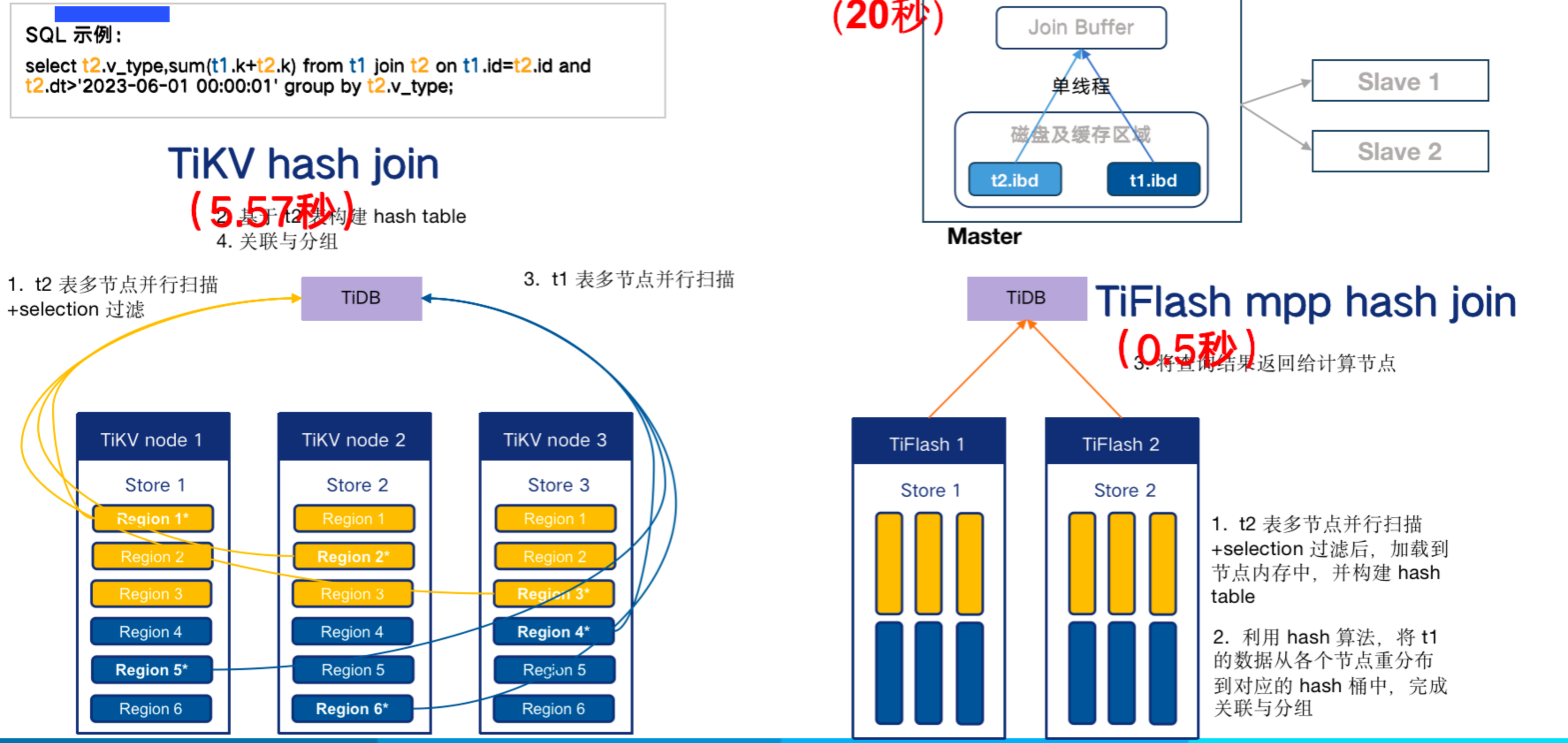
性能优势：一般在单表千万较为明显，单行数据越长越明显



TiDB VS MySQL 复杂 SQL 处理能力

TiDB 复杂 SQL 处理能力显著优于 MySQL，如下图所示两张千万级别表关联 MySQL 耗时 20 秒，使用 TiDB + TiKV 处理耗时 5.57 秒，大部分过滤计算可以下推到 TiKV 充分利用分布式能力。使用 TiDB + TiFlash 处理耗时 0.5 秒，TiFlash MPP Join 进一步提升数据和计算亲和性，减少数据网络传输并利用列存优势只读取需要的列，减少 IO 和 CPU 消耗

大表 join 性能 vs MySQL



TiDB VS MySQL 写入性能

TiDB VS MySQL 复杂 SQL 处理能力

TiDB VS MySQL 扩容能力

TiDB VS MySQL 高可用能力

TiDB VS MySQL 在线 DDL 能力

TiDB VS MySQL 资源管控能力

TiDB VS MySQL 总结和场景说明

附录：查询时间对比

MySQL [test]> select version(),t2.v_type,sum(t1.k+t2.k),count(*) from t1 join t2 on t1.id=t2.id and t2.dt>'2023-06-01 00:00:01' group by t2.v_type;

version() | v_type | sum(t1.k+t2.k) | count(*) |
-----+-----+-----+-----+
5.7.26-log | 00000 | 146047803184 | 1460270 |
5.7.26-log | 00001 | 146100096276 | 1460271 |
5.7.26-log | 00002 | 145963074614 | 1460271 |
5.7.26-log | 00003 | 146058369416 | 1460271 |
5.7.26-log | 00004 | 146029897954 | 1460270 |
5 rows in set (20.10 sec)

MySQL [test]> select version(),t2.v_type,sum(t1.k+t2.k),count(*) from t1 join t2 on t1.id=t2.id and t2.dt>'2023-06-01 00:00:01' group by t2.v_type;

version() | v_type | sum(t1.k+t2.k) | count(*) |
-----+-----+-----+-----+
8.0.32 | 00001 | 145040639002 | 1460271 |
8.0.32 | 00002 | 146030634272 | 1460271 |
8.0.32 | 00003 | 145967388358 | 1460271 |
8.0.32 | 00004 | 146017702894 | 1460270 |
8.0.32 | 00000 | 146031060334 | 1460270 |
5 rows in set (17.13 sec)

MySQL [test]> select substr(tidb_version(),1,23),t2.v_type,sum(t1.k+t2.k),count(*) from t1 join t2 on t1.id=t2.id and t2.dt>'2023-06-01 00:00:01' group by t2.v_type;

substr(tidb_version(),1,23) | v_type | sum(t1.k+t2.k) | count(*) |
-----+-----+-----+-----+
Release Version: v6.5.3 | 00000 | 135594803422 | 1355443 |
Release Version: v6.5.3 | 00002 | 135501509002 | 1355443 |
Release Version: v6.5.3 | 00003 | 135077305012 | 1355443 |
Release Version: v6.5.3 | 00001 | 135545196874 | 1355444 |
Release Version: v6.5.3 | 00004 | 135505504000 | 1355443 |
5 rows in set (5.57 sec)

MySQL [test]> select substr(tidb_version(),1,23),t2.v_type,sum(t1.k+t2.k),count(*) from t1 join t2 on t1.id=t2.id and t2.dt>'2023-06-01 00:00:01' group by t2.v_type;

substr(tidb_version(),1,23) | v_type | sum(t1.k+t2.k) | count(*) |
-----+-----+-----+-----+
Release Version: v6.5.3 | 00001 | 135545196874 | 1355444 |
Release Version: v6.5.3 | 00002 | 135501509002 | 1355443 |
Release Version: v6.5.3 | 00004 | 135505504000 | 1355443 |
Release Version: v6.5.3 | 00000 | 135594803422 | 1355443 |
Release Version: v6.5.3 | 00003 | 135077305012 | 1355443 |
5 rows in set (0.49 sec)

MySQL 5.7.26

MySQL 8.0.32

TiDB V6.5.3 + TiKV

V6.5.3 + TiFlash

MySQL [test]> show create table t2\G

***** 1. row *****
Table: t2
Create Table: CREATE TABLE 't2' (
 'id' int(11) NOT NULL AUTO_INCREMENT,
 'k' bigint(16) DEFAULT NULL,
 'c' varchar(50) NOT NULL DEFAULT 'dddd-dddd',
 'pad' varchar(50) DEFAULT 'cccc-cccc',
 'dt' datetime(3) DEFAULT CURRENT_TIMESTAMP(3),
 'v_type' varchar(5) DEFAULT '00000',
 'ut' datetime(3) DEFAULT CURRENT_TIMESTAMP(3) ON UPDATE CURRENT_TIMESTAMP(3),
 'v1' varchar(64) DEFAULT 'aaa',
 PRIMARY KEY ('id') /*!!(clustered_index) CLUSTERED */,
 KEY 'idx_k' ('k'),
 KEY 'idx_dt' ('dt')
)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin AUTO_INCREMENT=16818720
1 row in set (0.00 sec)

MySQL [test]> select count(*) from t1;

count(*) |
-----+
16777216 |
1 row in set (0.03 sec)

MySQL [test]> select count(*) from t2;

count(*) |
-----+
16777216 |
1 row in set (0.02 sec)

MySQL [test]> select count(*) from t2 where dt>'2023-06-01 00:00:01';

count(*) |
-----+
6777216 |
1 row in set (0.02 sec)

查询数据范围约占总数据量的 40%

某客户业务适配 TiDB 过程中验证 SQL 耗时，统计结果显示 MySQL 超过1秒的 SQL 在 TiDB 运行速度显著提升。

某客户 MySQL VS TiDB

性能数据统计

时间分布	复杂业务	
	MySQL SQL 数	TiDB SQL 数
<100us	11854	0 (变少)
(100us,1000us)	155753	5026 (变少)
(1ms,10ms)	175240	440318 (变多)
(10ms,100ms)	148834	426487 (变多)
(100ms,1s)	545721	712551 (变多)
(1s,10s)	452981	55361 (变少)
>10s	170494	21039 (变少)

原本在 MySQL 中大于10秒的 SQL:
1.95% 的 SQL 均缩短到 1秒内
2.其中有65%的 SQL 缩短到10ms 到100ms
3.还有10%左右缩短到1ms 到 10ms 之间

原本在 MySQL 中大于1秒小于10秒的 SQL:
1.88% 的 SQL 缩短到1秒内

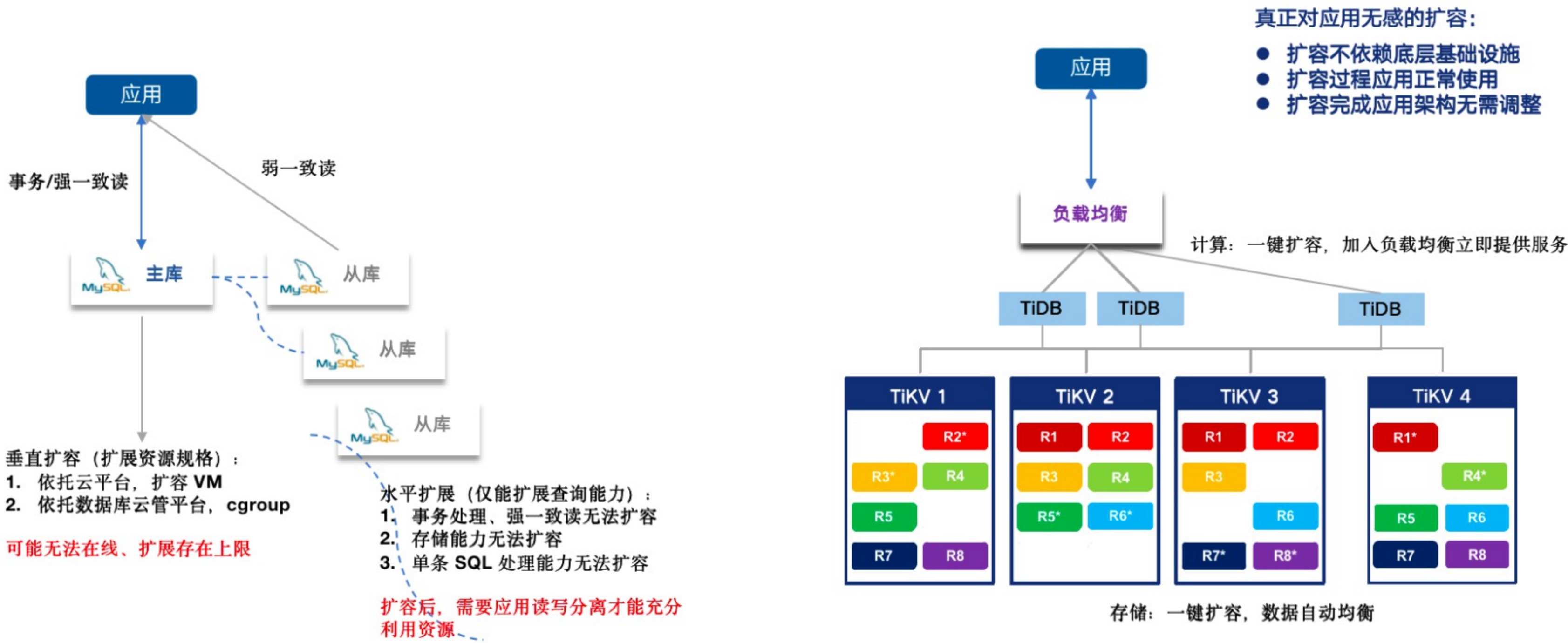
PingCAP

TiDB

TiDB VS MySQL 扩容能力

MySQL 智能一主多从，主节点无法横向扩展（除非接受分库分表），从节点扩容需要应用支持读写分离，并且单条 SQL 处理能力无法扩展。TiDB 的存储和计算节点都可以扩容，数据规模支持 PB 级别，复杂 SQL 可以利用分布式计算能力加速。

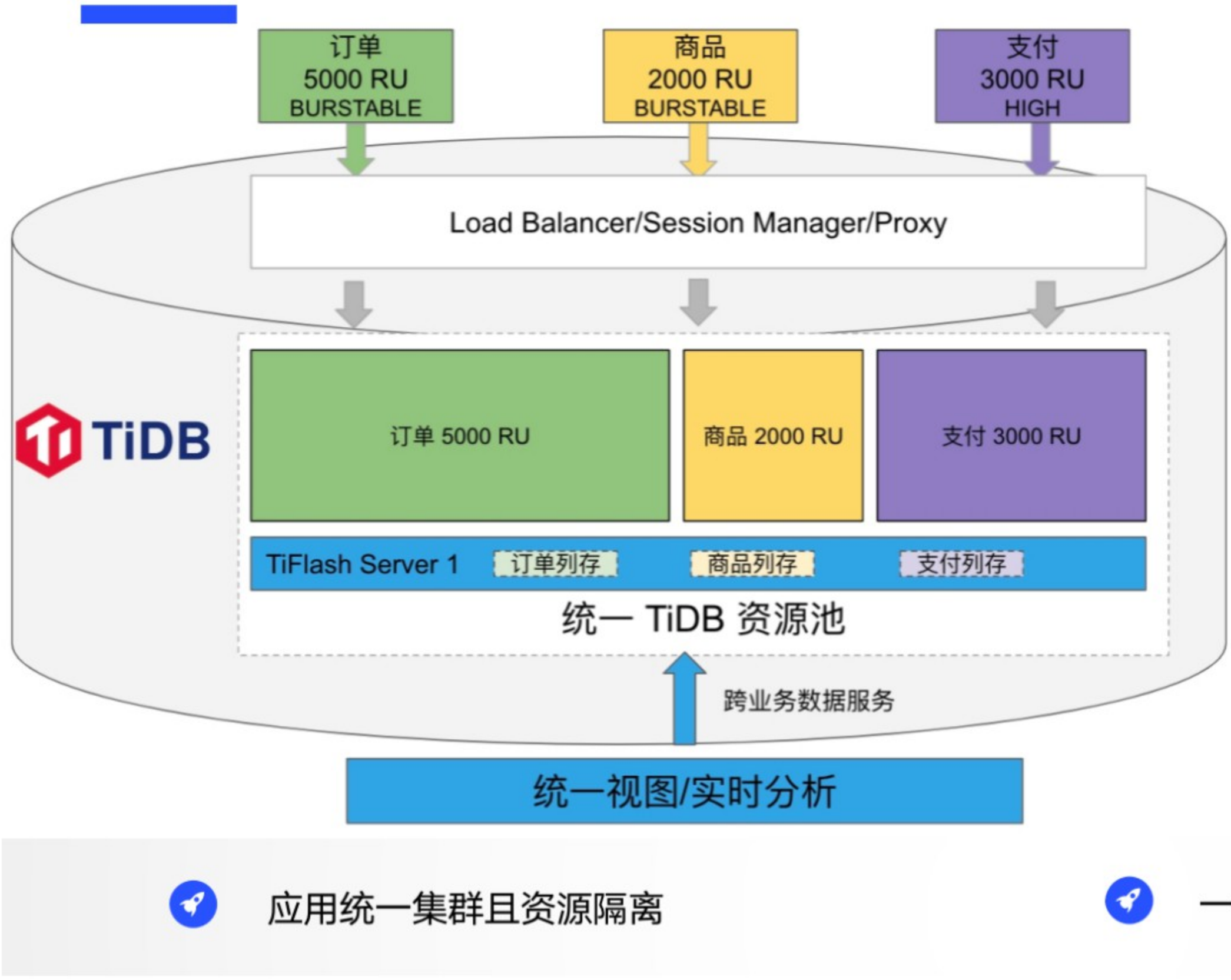
扩容能力 vs MySQL



TiDB VS MySQL 高可用能力

MySQL 使用增强半同步和 MGR 方案复制效率低，主节点故障影响业务处理，大事务影响从库数据时效性。TiDB 将数据打散分布，并通过 Raft 实现数据一致性和复制，单机故障对集群影响小并可以保证一致性

高可用/数据一致性 vs MySQL



解决方案

- 基于 TiDB 资源管控技术，设置单个业务只使用部分 RU (虚拟资源)
- 集群繁忙时控制单个资源总量
- 资源空闲时单个业务资源超用
- 设置业务优先级，保证高优先级业务稳定

方案优势

- 节约硬件成本、错峰资源借用、弹性扩缩容
- 统一集群降低开发和运维成本
- 解决数据孤岛问题：支持跨业务强一致性数据查询、实时分析和统一视图

TiDB VS MySQL 总结和场景说明

总结优势如下：

1. 数据量大的场景点查点写才会有优势
2. MySQL 中耗时1秒以上的复杂 SQL TiDB 分布式 HTAP 优势明显
3. TiDB 高可用能力更强
4. TiDB 扩展能力更强
5. TiDB 在线 DDL 能力更强
6. TiDB 多业务融合能力更强
7. 数据量少的点点点写的简单场景 TiDB 完全可以支持，但是 MySQL 性价比更高

总结 TiDB 合适场景：

1. 数据量大
2. 复杂 SQL 较多
3. TP 和 AP 都包含
4. 多业务融合场景
5. TiDB 在简单点查点写且数据量小的场景由于没有利用分布式优势但是却包含了分布式性能开销因此性价比不高

版权声明：本文为 TiDB 社区用户原创文章，遵循 **CC BY-NC-SA 4.0** 版权协议，转载请附上原文出处链接和本声明。

评论

T

添加评论

评论

暂无评论

< 1 >

互助与交流

活动

问答论坛

TiKV 社区

Chaos Mesh 社区

学习与应用

文档

专栏

视频课程

考试认证

典型案例

开发者指南

发现社区

TiDB User Group

问答之星

社区准则

联系我们

电子书

