TiDB | COMMUNITY

TiDB资源管控特性解读及应用探索

李文杰

TiDB 社区版主
2019-2023 社区MVA/MOA

TiDB | COMMUNITY

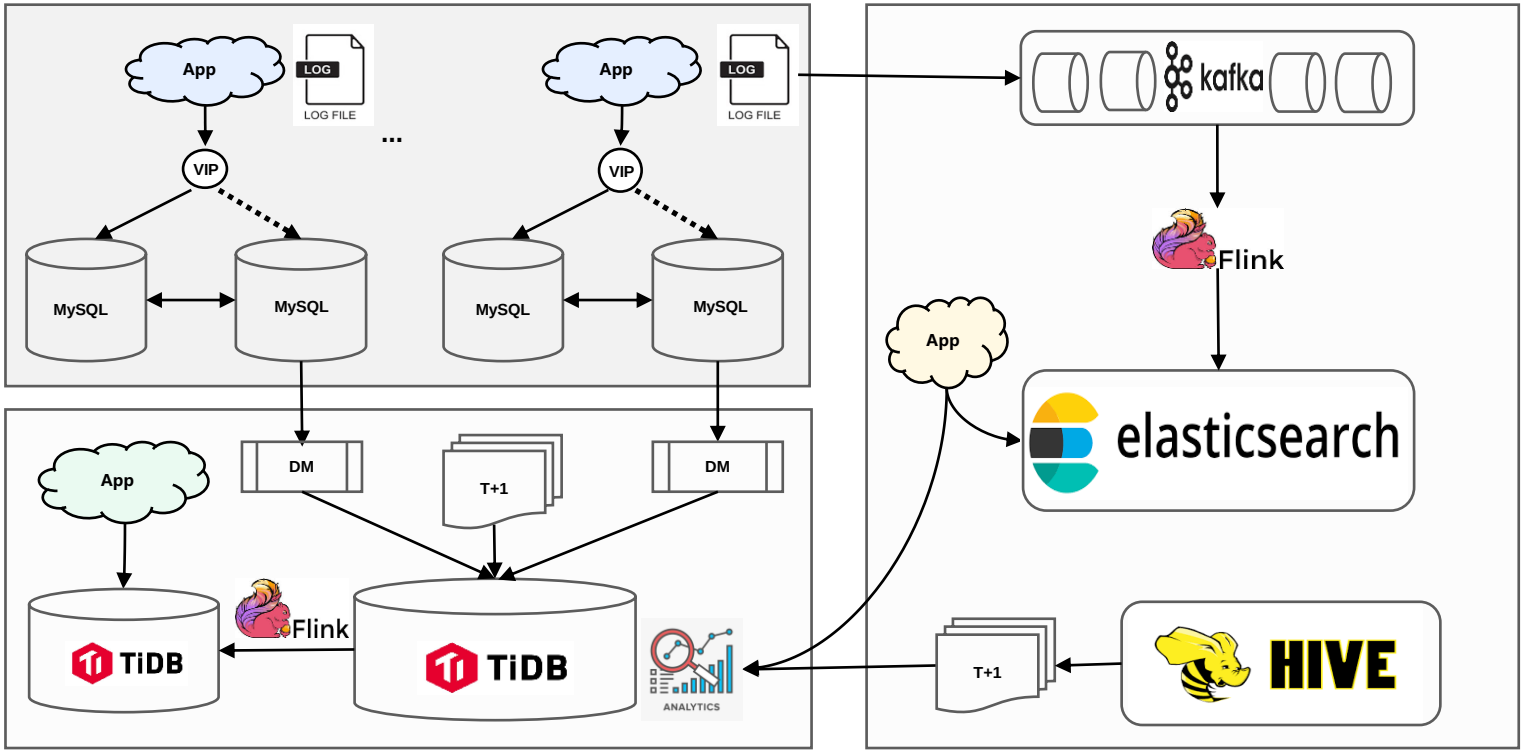
TiDB资源管控特性解读及应用探索

- HTAP多业务融合架构
- Resource Control
- 总结与展望

1 HTAP多业务融合架构

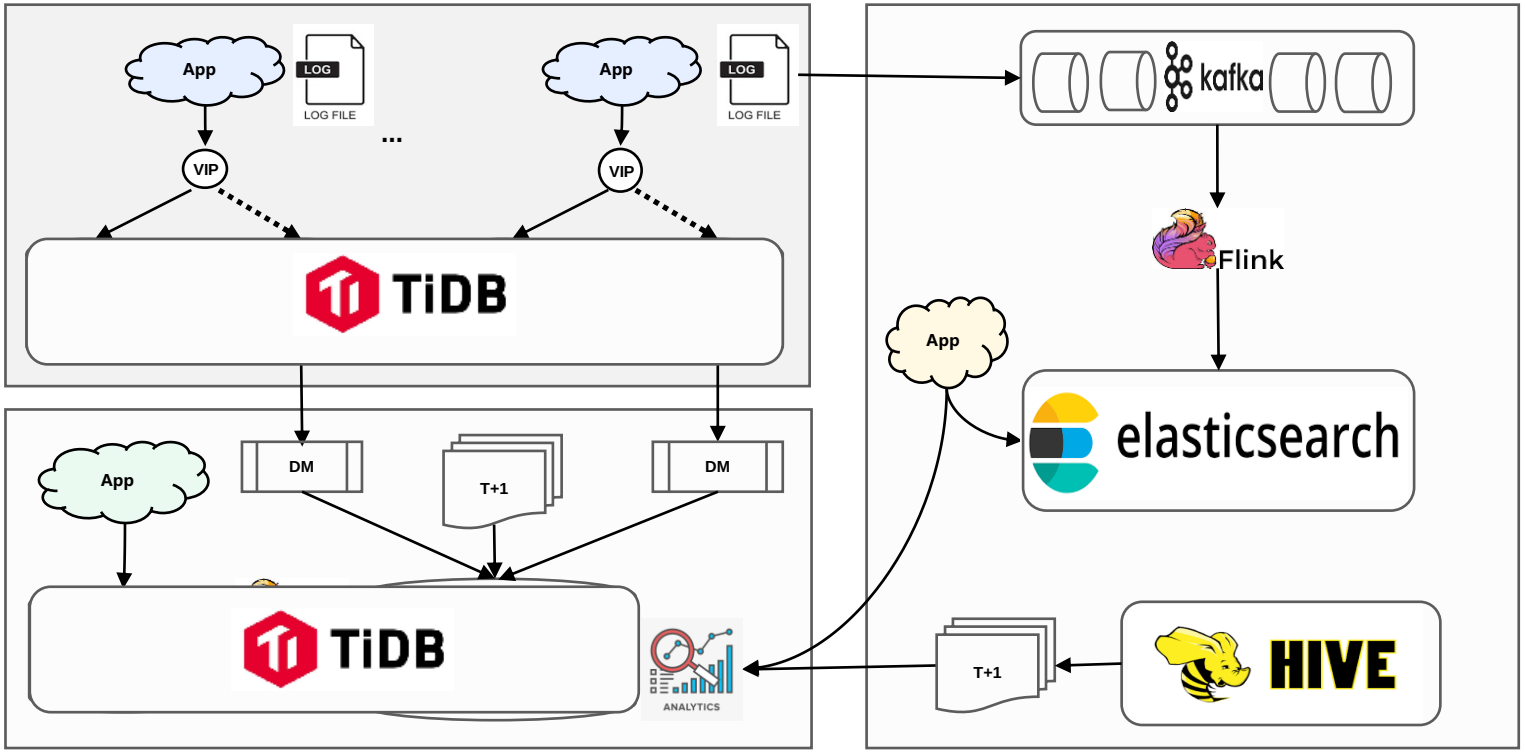
业务系统

- 业务独立
- 硬件成本
- 管理成本



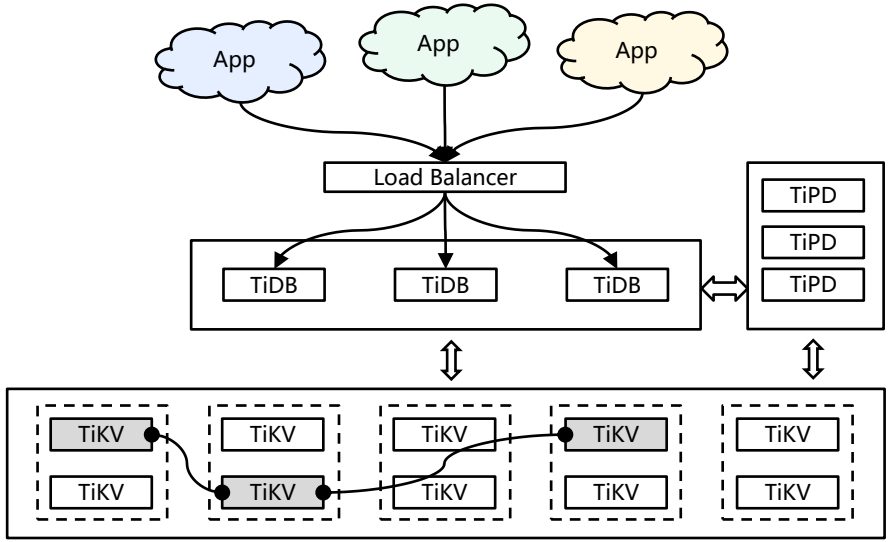
多业务融合系统

- 架构简化
- HTAP
- DB数下降
- 成本缩减
- 相互干扰



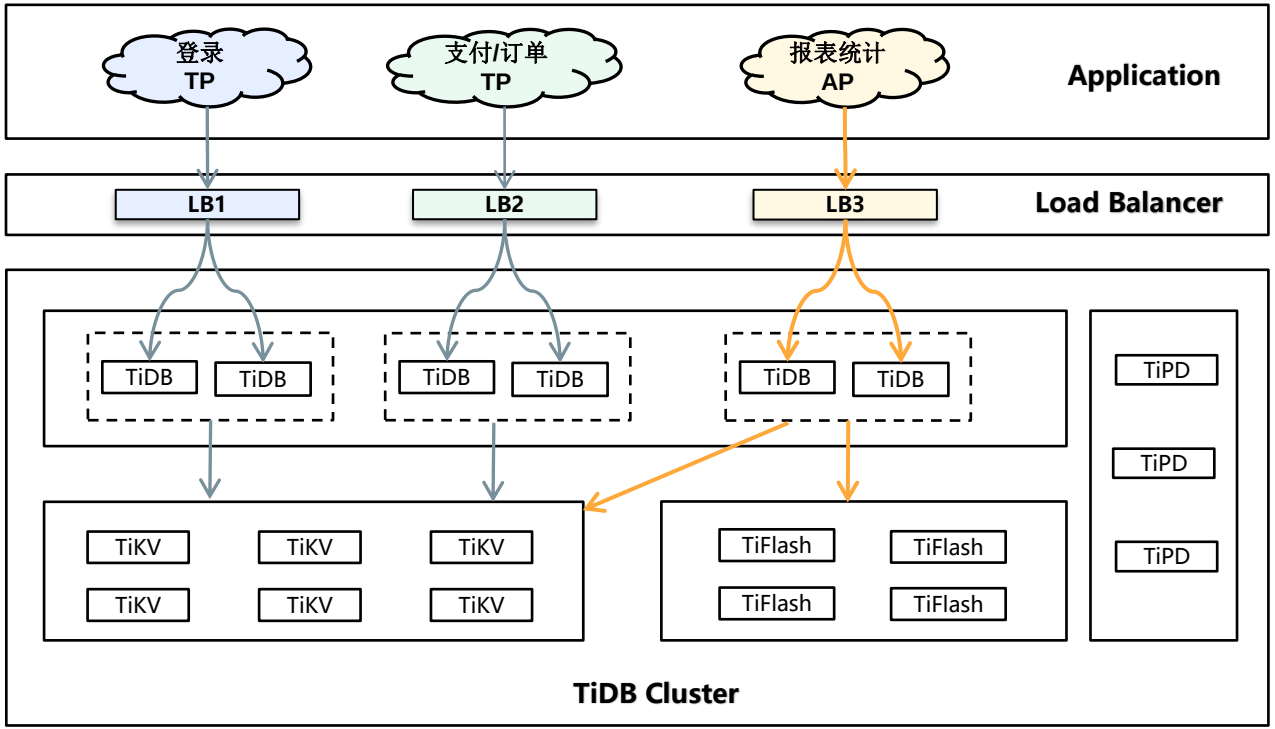
HTAP业务架构

- **单一入口**
- **跨系统负载相互干扰**
  - OLTP
  - OLAP
- **资源混用挤兑**



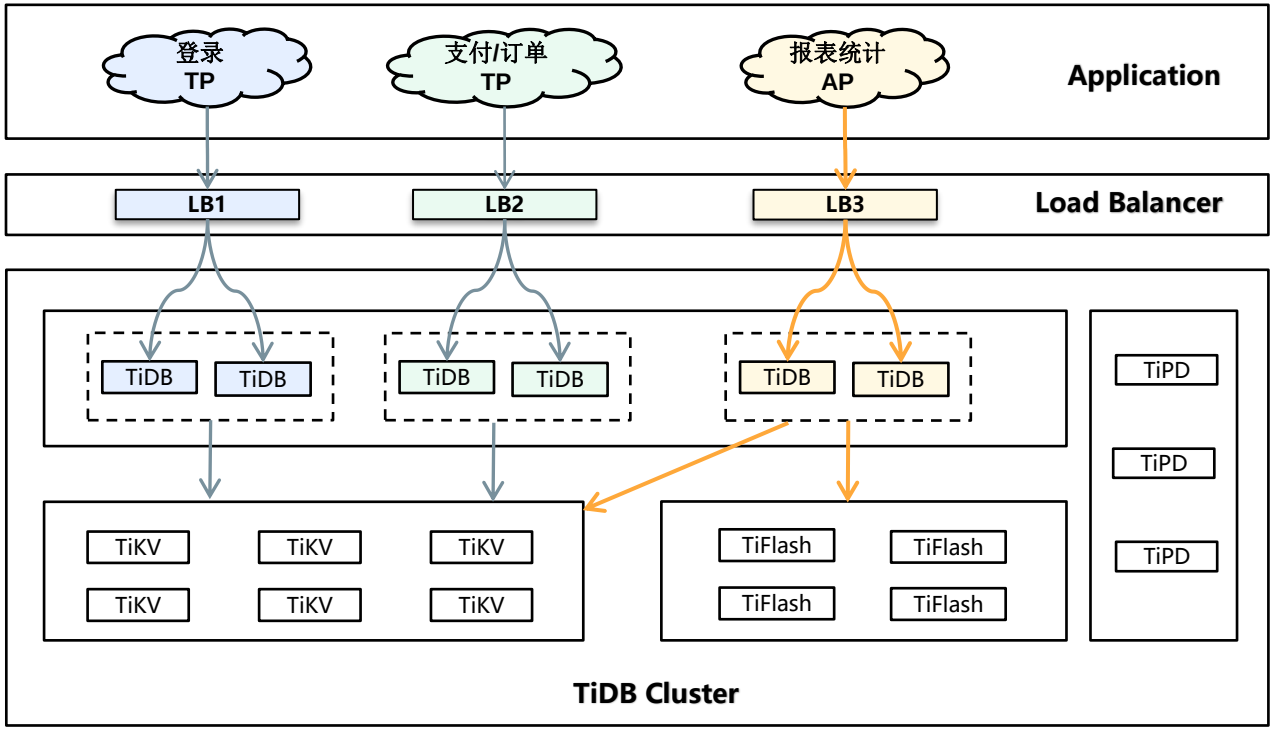
HTAP业务架构

- ~~单一入口~~
  - ✓ 流量隔离
  - ✓ 域名独享
- 跨系统负载相互干扰
  - OLTP
  - OLAP
- 资源混用挤兑



HTAP业务架构

- ~~单一入口~~
  - ✓ 流量隔离
  - ✓ 域名独享
- 跨系统负载相互干扰
  - ✓ OLTP独享计算
  - ✓ OLAP独享计算
  - ✓ 计算跨机房容灾
- 资源混用挤兑



HTAP业务架构

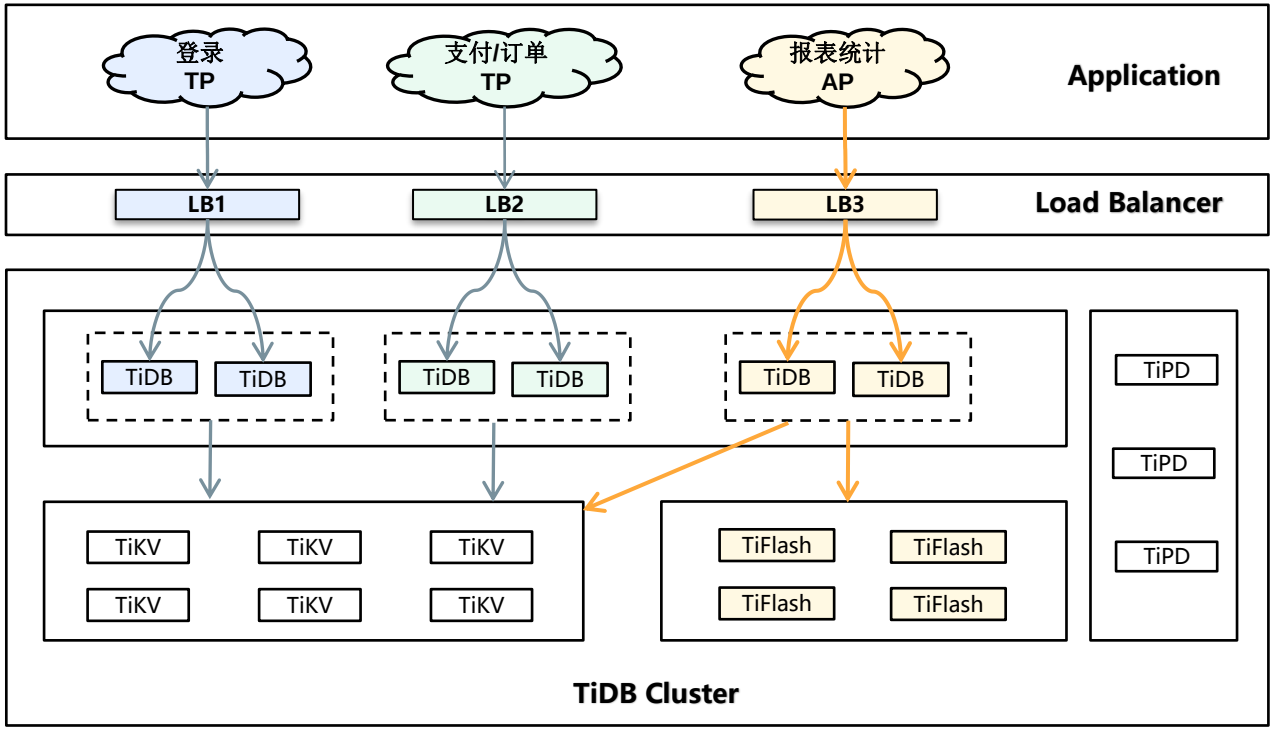
- ~~单一入口~~
  - ✓ 流量隔离
  - ✓ 域名独享

物理隔离

- 跨系统负载相互干扰
  - ✓ OLTP独享计算
  - ✓ OLAP独享计算
  - ✓ 计算跨机房容灾

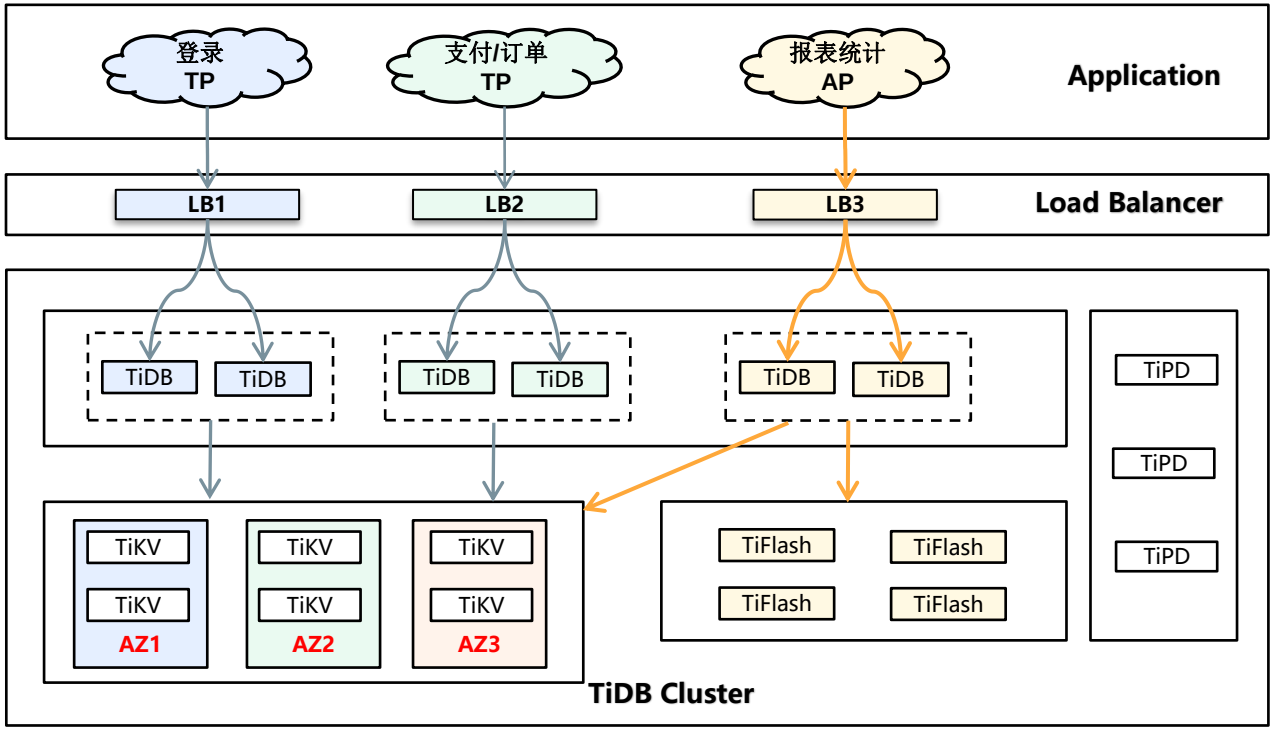
极大改善

- 资源混用挤兑
  - ✓ 行、列存隔离
  - ✓ AP独享列存



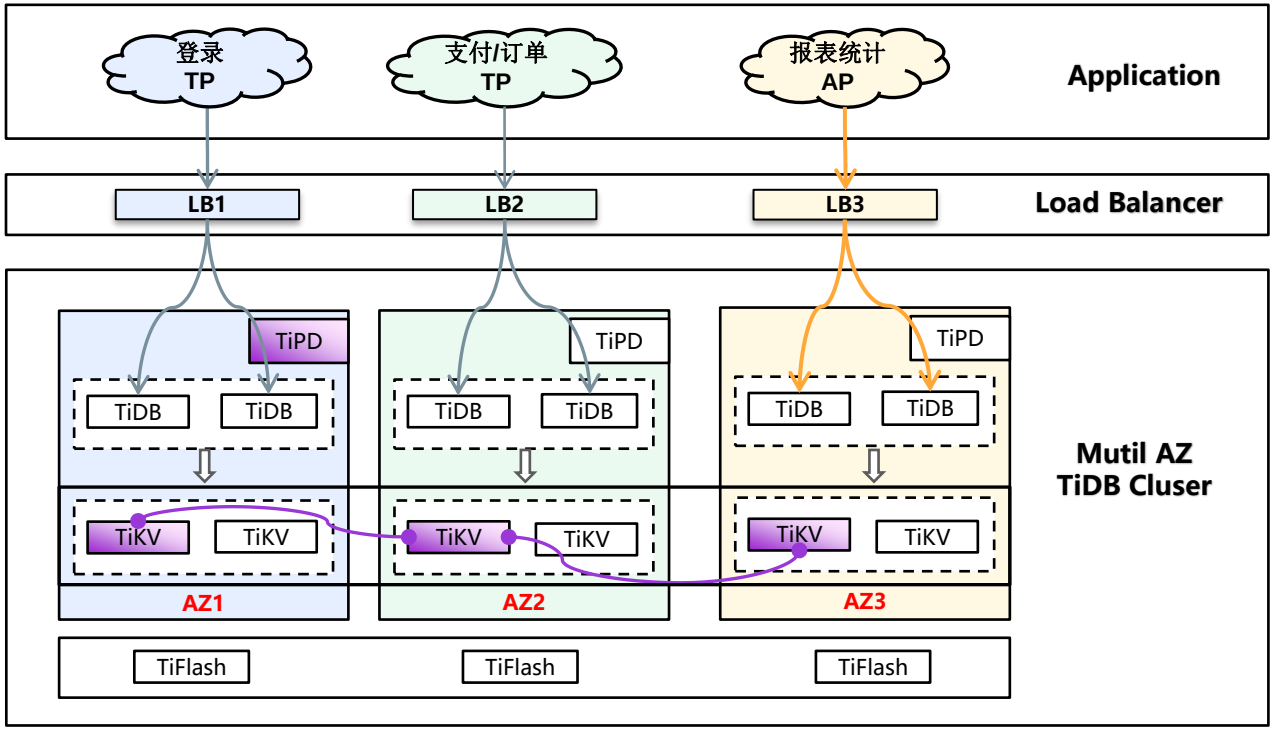
HTAP业务架构

- 单一入口
  - ✓ 流量隔离
  - ✓ 域名独享
- 跨系统负载相互干扰（物理隔离）
  - ✓ OLTP独享计算
  - ✓ OLAP独享计算
  - ✓ 计算跨机房容灾
- 资源混用挤兑（极大改善）
  - ✓ 行、列存隔离
  - ✓ AP独享列存
  - ✓ 存储跨机房容灾



HTAP业务架构

- 单一入口
  - ✓ 流量隔离
  - ✓ 域名独享
- 跨系统负载相互干扰（物理隔离）
  - ✓ OLTP独享计算
  - ✓ OLAP独享计算
  - ✓ 计算跨机房容灾
- 资源混用挤（极大改善）
  - ✓ 行、列存隔离
  - ✓ AP独享列存
  - ✓ 存储跨机房容灾

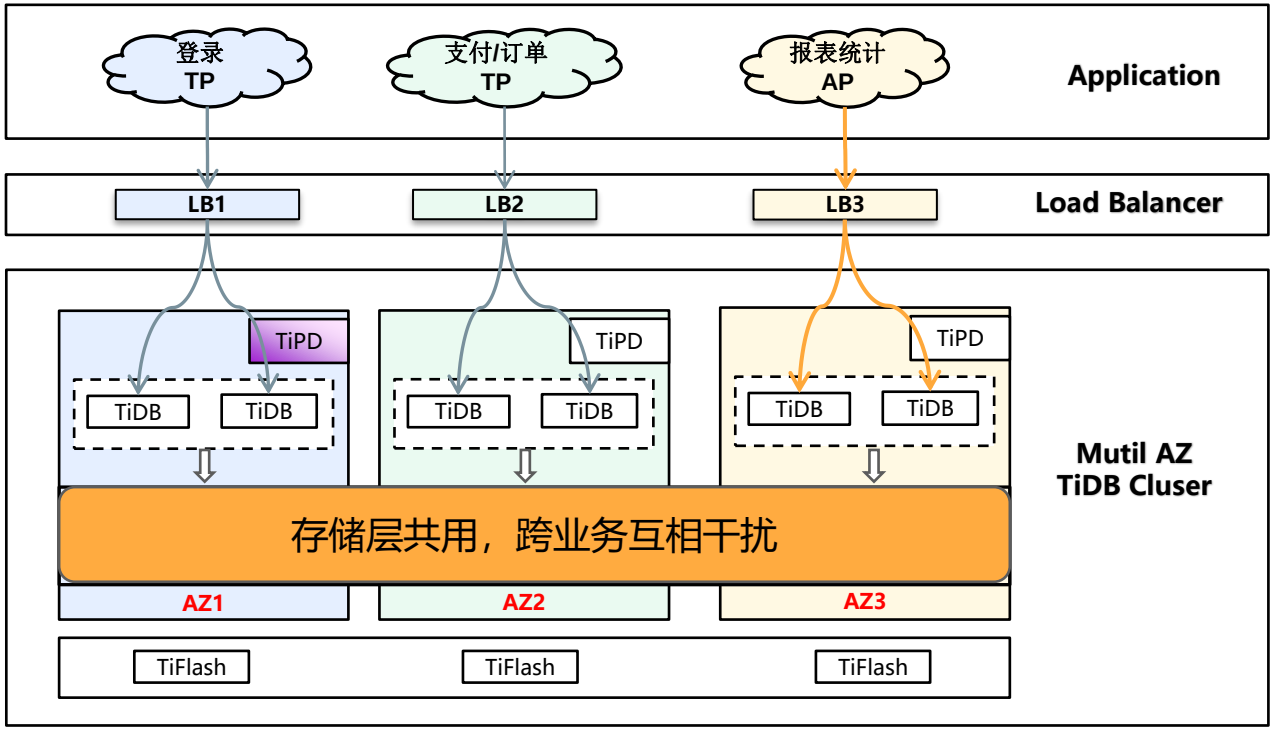


HTAP业务架构

- 单一入口
  - ✓ 流量隔离
  - ✓ 域名独享
- 跨系统负载相互干扰
  - ✓ OLTP独享计算
  - ✓ OLAP独享计算
  - ✓ 计算跨机房容灾
- 资源混用挤占
  - ✓ 行、列存隔离
  - ✓ AP独享列存
  - ✓ 存储跨机房容灾

物理隔离

极大改善



优势：

计算/行列存储物理隔离，多AZ容灾

不足：

存储层跨系统共用，无法隔离

HTAP业务架构

多业务共用一个集群，在我们尽可能将 TP 业务和 AP 业务分离部署的前提下，通常还是会遇到下面的痛点问题。

高峰挤兑

- 当一个业务处于高峰期时，会过多占用集群资源，影响别的业务

✓ 希望能保护不同业务的资源持有情况，保证业务能分配到基本的运行资源而不被挤兑。

低谷过剩

- 当集群中的重要业务处于低谷值时，有较多的剩余资源

✓ 希望引入错峰运行的业务，充分使用资源，实现降本增效。同时要求业务能得到控制，其他时候不会占用过多资源。

异常放大

- 当集群遇到临时的问题 SQL 引发的性能问题时，影响整个集群，只能停掉对应业务 。

✓ 希望不是干掉它的执行，而是临时限制它的资源消耗，允许它缓慢运行，但又不会影响集群其他业务。

2 Resource Control

资源管控-原理

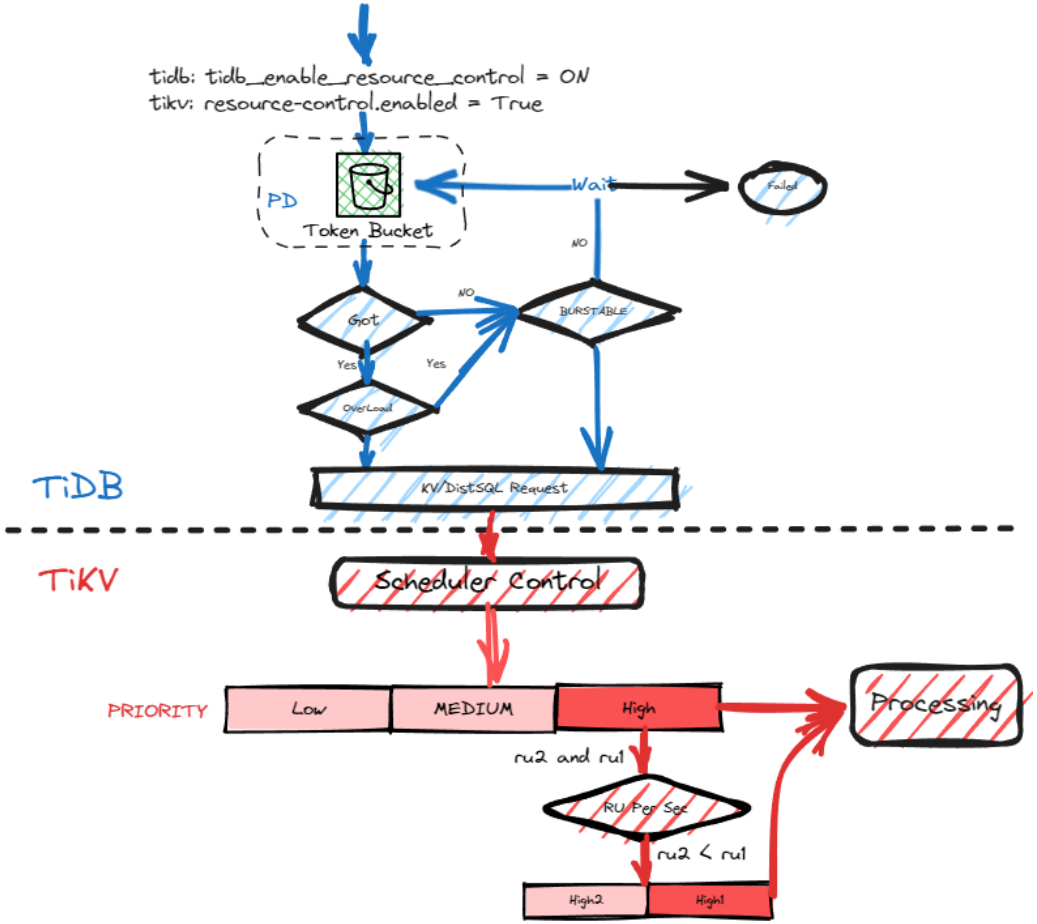
- Request Unit (RU)
  - 系统资源的统一抽象计量单位
  - 包含CPU、磁盘IO 和网络 IO

- **用户**绑定**资源组**（RG），通过RG实现管控

管控有 2 层实现：

- TiDB 流控
  - 根据配额对读写做**流控**
  - 令牌桶算法

- TiKV 优先级调度
  - 根据配额映射的**优先级**来做调度



资源管控-计算

- Request Unit (RU)
  - 系统资源的统一抽象计量单位
  - 包含CPU、磁盘IO 和网络 IO

- 资源组处理 SQL 时：
  - TiKV 处理的时长是 c 毫秒
  - r1 次请求读取了 r2 KB 数据
  - w1 次写请求写入了 w2 KB 数据
  - 复制的副本数是 n

消耗资源示意计算公式（非精确）

RU = c * 1/3 + (r1 * 0.25 + r2 * 1/64)
+ (w1 * 1.5 + w2 * 1 * n)

资源	RU 权重
消耗 CPU	1 ms = 1/3 RU
读数据 IO	1 KB = 1/64 RU
写数据 IO	1 KB = 1 RU
1 次读请求 RPC 开销	0.25 RU
1 次写请求 RPC 开销	1.5 RU

资源管控-多租户实践

在资源管控技术基础上，为三类业务负载（用户/租户）分别创建资源组。

- 为租户 app_oltp 分配一个较高的用量，app_olap 和 app_other 因业务重要程度相对较低，则分配较低的配额。

✓ 在系统资源紧张时，最优先保证租户 app_oltp 的服务质量。

- 租户 app_oltp 和 app_olap 的资源组设置为 burstable

✓ 租户 app_oltp 发生超预期的负载，仍旧可能会保证质量；

✓ 而当整个集群负载有空余时， 租户 app_olap 可以利用空闲资源加速其工作。

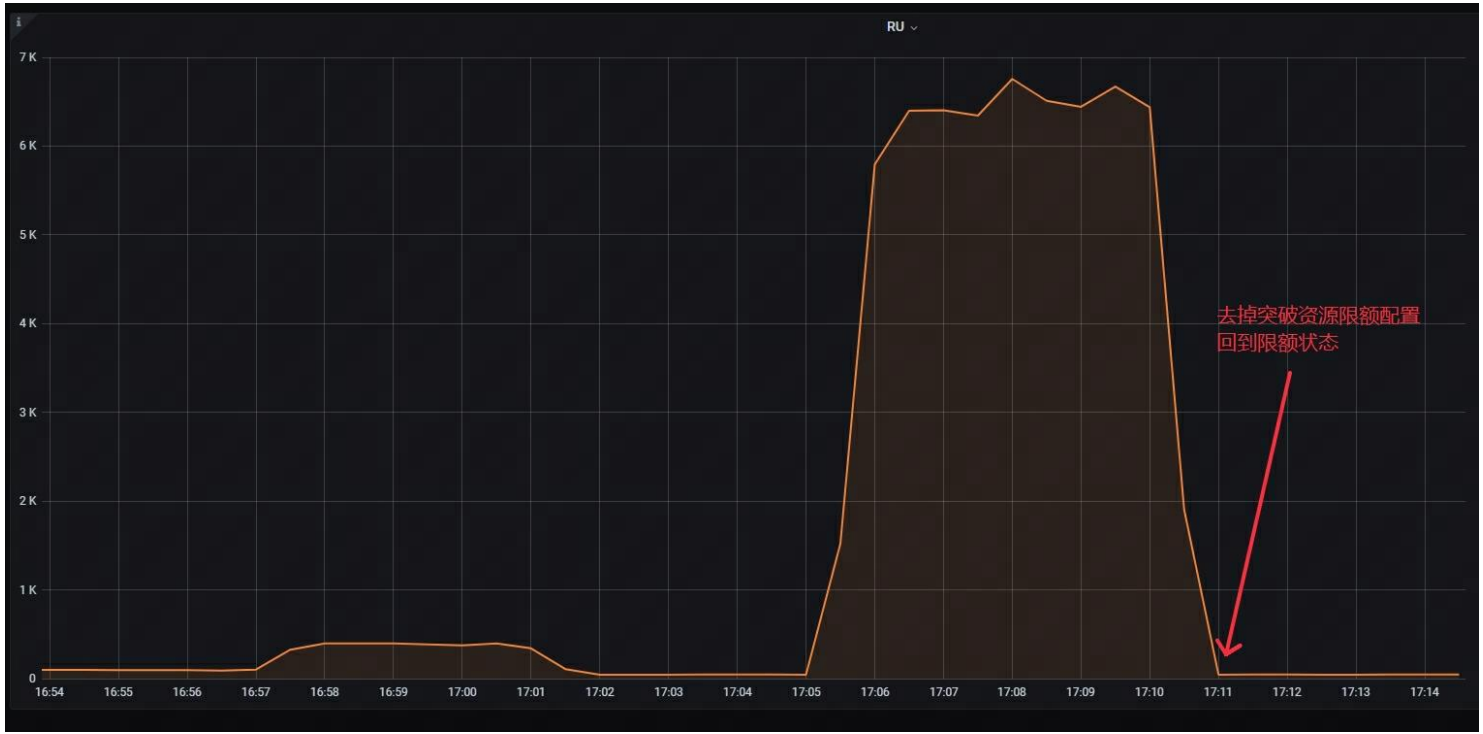
租户	重要程度	业务说明	资源组	RU 用量 RU_PER_SEC	是否超额分配 BURSTABLE	优先级 PRIORITY
app_oltp	高	运行在线交易事务。表示在线实时流量，有明显的高峰和波谷，需要 24h 保证稳定	rg_oltp	1000	是	HIGH
app_olap	中	运行分析事务。表示高消耗资源的流量，高吞吐，重要但不紧急，任务尽可能快完成	rg_olap	400	是	MEDIUM
app_other	低	集群中普通的租户，资源消耗低，优先级不高，优先保证不影响其他租户运行	rg_other	100	否	MEDIUM

资源管控-多租户实践

- 场景1：　实时在线**增加/减少**业务租户可用资源。根据市场营销活动随时增加可用资源，活动结束就减少配额，非常灵活。

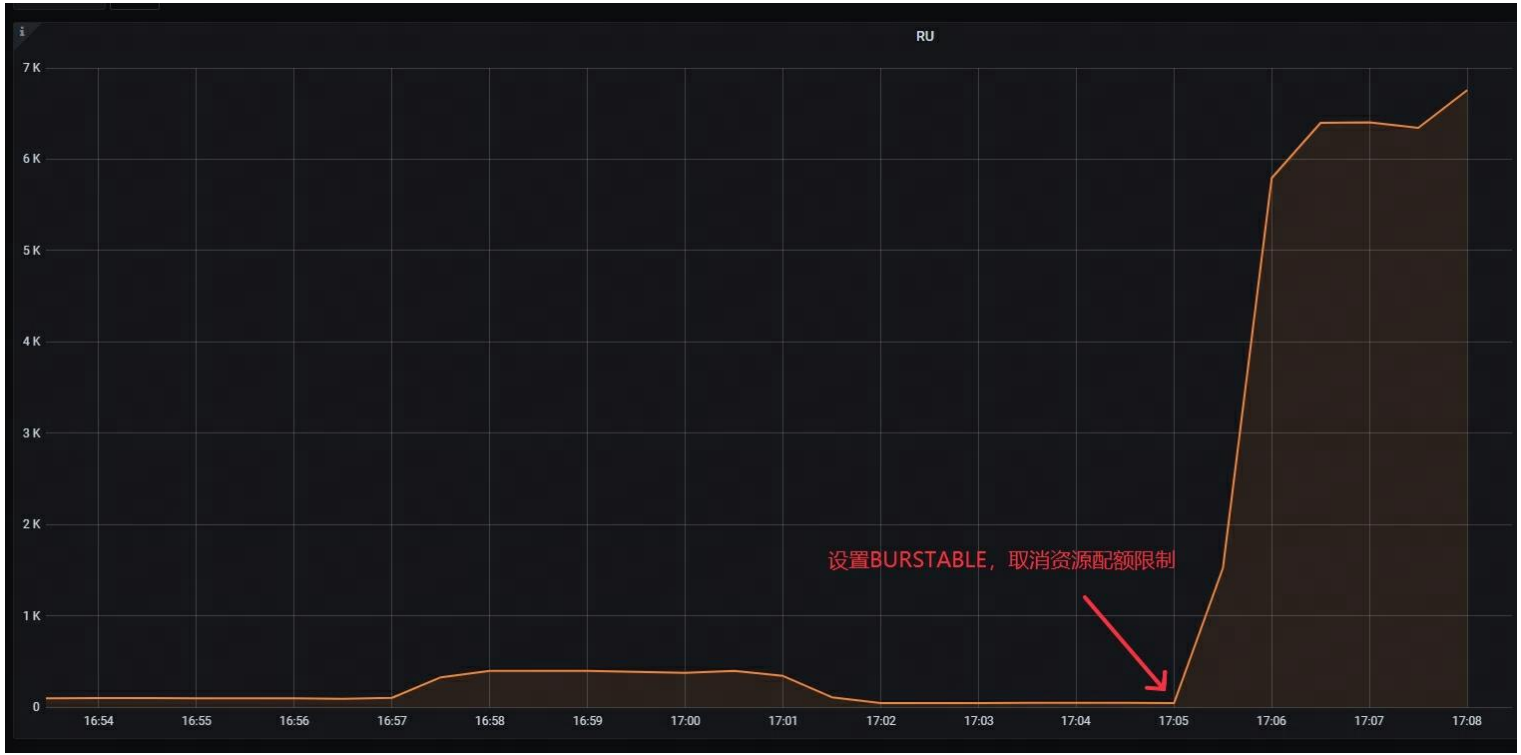
资源管控-多租户实践

- 场景2： 实时在线**限制**租户业务可用配额。对突发故障，对性能异常的业务加以限制，避免干扰其他业务。



资源管控-多租户实践

- 场景3： 实时在线**取消**租户业务配额**限制**。突发故障问题得以解决，取消临时限制，业务恢复正常运行。



资源管控-多租户实践

- **场景4： 多业务共用一个集群，实现资源错峰使用，提高资源使用率，降低集群数量，缩减硬件成本、管理成本。**

- 根据业务运行动态，实时扩大、缩小、取消限额、加限制等资源使用，极大提高集群资源使用率。

- 支持对用户、会话、SQL语句（Hint）级别的管控
  - 管控粒度过大
  - 改业务代码

资源管控-Runaway Queries

v7.2.0 起引入 Runaway Queries，是指执行时间或消耗资源超出预期的查询，在运行时间和资源消耗上有显著特征。

功能作用

- 系统**自动识别和管理资源消耗超出预期的查询。**
- 降低突发 SQL 性能问题带来的负面影响
- 保护复杂工作负载下TiDB系统的稳定性，提高集群的可靠性。

识别执行

识别：
运行时间或SQL特征。

操作：
- DRYRUN ：仅识别记录不处理。可用于检测规则。
- COOLDOWN ：降到资源组的最低优先级。
- KILL ：终止查询，防止其进一步影响数据库性能。

资源管控-Runaway Queries

基于执行时间的管控

- 阶段1：业务正常运行
- 阶段2：突发异常SQL出现
- 阶段3：自动识别异常SQL
- 阶段4：持续管控异常SQL
- 阶段5：业务正常运行

效果：
- 自动识别负面SQL并处置
- 保障系统稳定性、安全性

阶段3限制操作：QUERY_LIMIT=(EXEC_ELAPSED='2s', ACTION=KILL)

阶段4限制操作：
QUERY_LIMIT=(EXEC_ELAPSED='2s', ACTION=KILL, WATCH=SIMILAR DURATION='5m')

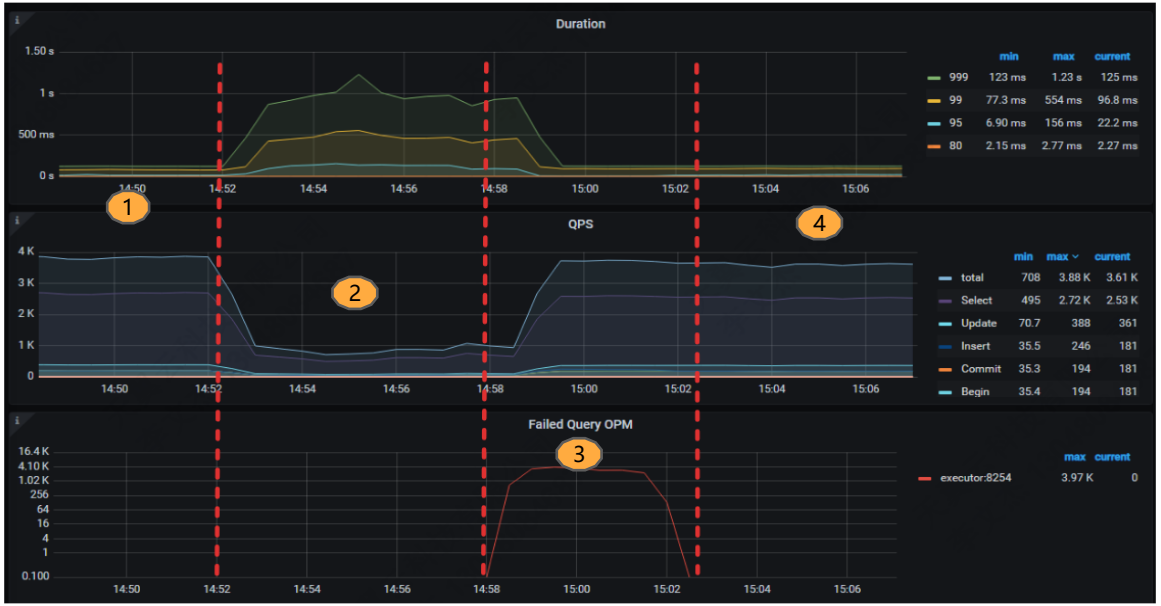
阶段	访问流量	负面SQL限制	QPS	P999	效果
1	业务	无	3.8k	124ms	正常
2	业务+持续注入大SQL	无	737 (-81%)↓	1.01s (+714%)↑	**剧烈干扰**
3	业务+持续注入大SQL	执行超阈值，kill	1.2k (-68%)↓	712ms (+474%)↑	**有所优化**
4	业务+持续注入大SQL	执行超阈值，kill 持续5min	3.7k (-3%)↓	128ms (+3%)↑	**基本正常**
5	业务	无	3.8k	126ms	正常

资源管控-Runaway Queries

基于异常SQL的管控（SQL黑名单）

v7.3.0引入了手动管理异常SQL的功能，通过指定SQL或Digest，Query Watch快速识别加黑语句，实现异常业务隔离，保障重要在线业务的稳定性。

- 阶段1：业务正常运行
- 阶段2：突发异常SQL出现
- 阶段3：添加SQL黑名单识别
- 阶段4：业务正常运行



阶段	访问流量	负面SQL限制	QPS	P999	效果
1	业务	无	3.8k	124ms	正常
2	业务+持续注入高消耗资源的SQL	无	825 (-78%)↓	975ms (+686%)↑	剧烈干扰
3	业务+持续注入高消耗资源的SQL	加黑SQL	2.8k (-26%)↓	381ms (+207%)↑	逐渐恢复至正常
4	业务+持续注入高消耗资源的SQL	黑名单持续生效	3.8k	124ms	正常

阶段3：
QUERY WATCH ADD RESOURCE GROUP `default` **SQL TEXT EXACT|SIMILAR** TO 'select * from Order.order_line limit 100000';

资源管控-后台任务

对于非 SQL 类型的系统任务管控，v7.4.0 引入后台任务功能实现。后台任务会被TiKV限制资源的使用，尽量避免对其他前台任务的性能影响。

支持管控的后台任务类型：

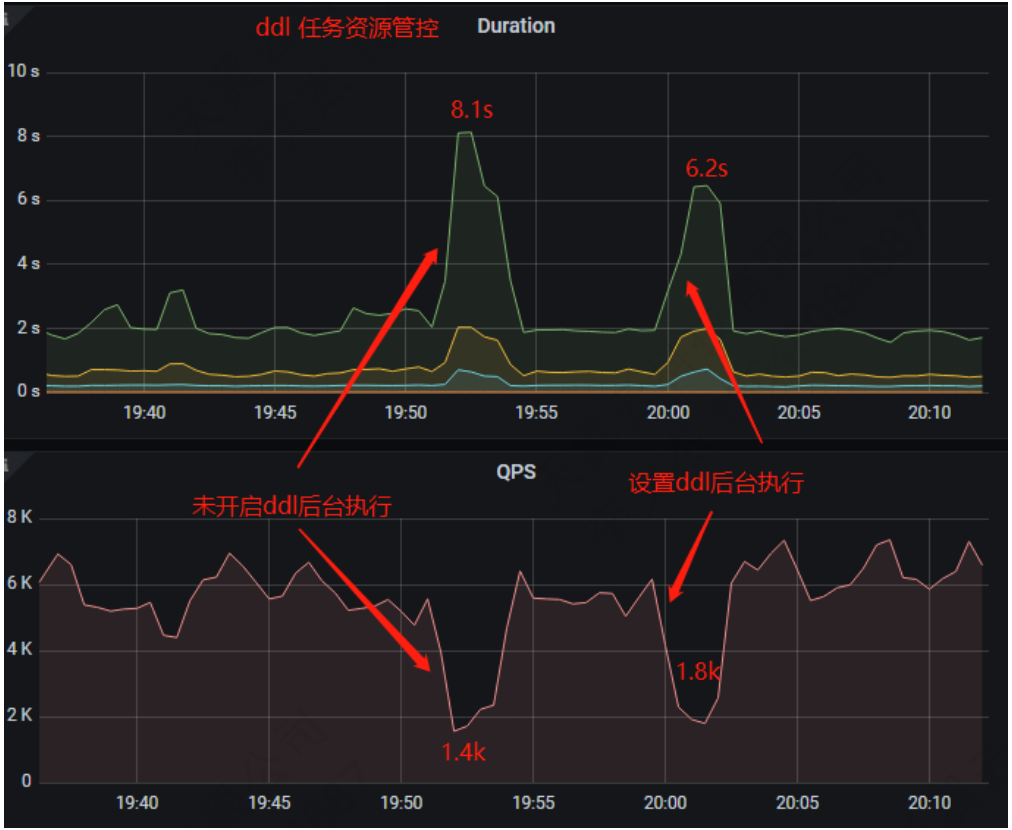
- ddl：对于 Reorg DDL，控制批量数据回写阶段的资源使用。
- stats：对应手动执行或系统自动触发的收集统计信息任务。
- lightning：使用Lightning 执行导入任务，支持物理和逻辑导入模式。
- br：使用 BR 执行数据备份和恢复。目前不支持 PITR。
- background：指定当前会话的任务类型为 background。

资源管控-后台任务

大表 DDL 是一个非常消耗资源的操作，后台任务可以控制批量数据回写阶段的资源使用

```
ALTER RESOURCE GROUP `default`
BACKGROUND=(TASK_TYPES='ddl');
```

- 调低优先级，限制大表DDL资源消耗
- 大表加索引场景，缓解对线上业务的负面影响

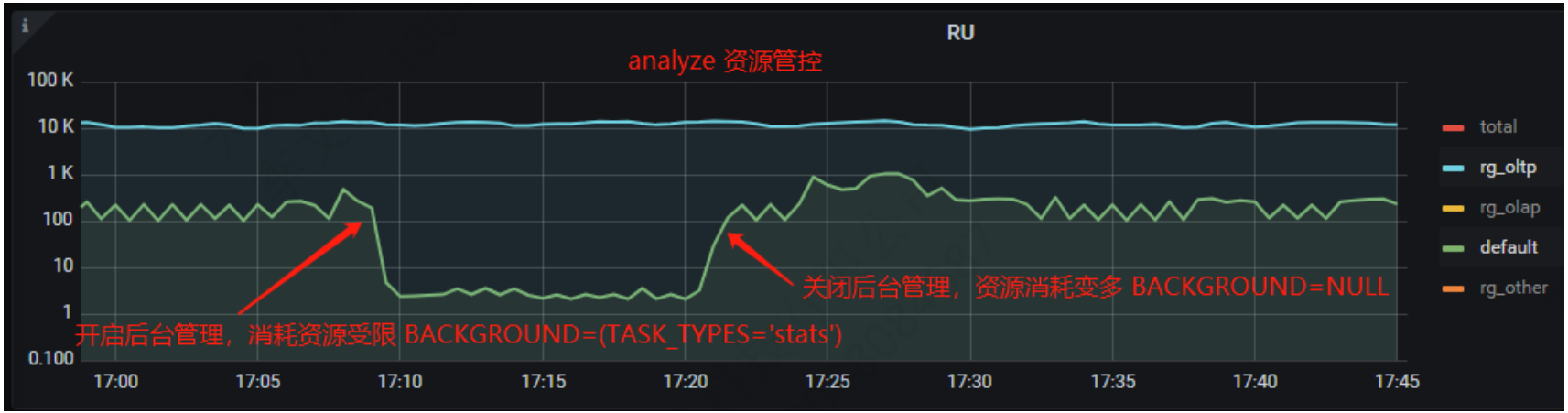


资源管控-后台任务

限制 Analyze 资源消耗情况，适用于手动或系统自动收集统计信息任务。

```
ALTER RESOURCE GROUP `default` BACKGROUND=(TASK_TYPES='stats');
```

- 调低优先级，限制统计信息收集时的资源使用
- 减少大表统计信息收集时对集群资源的占用，为更重要的业务省出资源

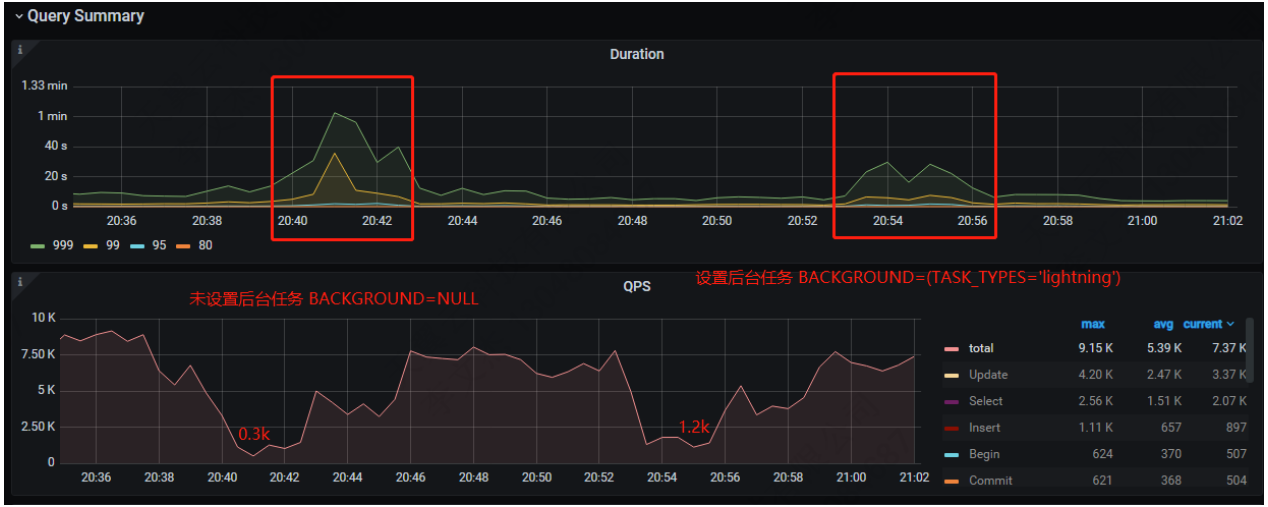


资源管控-后台任务

Lightning大量数据导入，在执行的时候会消耗大量资源，从而影响在线的高优先级任务的性能

Lightning在导入方式backend="local"

设置后台任务
BACKGROUND=(TASK_TYPES='lightning')



Lightning 物理导入方式 backend = "local"

设置后台任务
BACKGROUND=(TASK_TYPES='lightning')

不设置后台任务：
BACKGROUND=NULL

阶段	Lightning 导入模式	Lightning 是否设置为后台任务	导入时长	业务最低QPS	业务平均QPS	平均延迟（95%）
1	无任务	否	-	5.6k	7.3k	230ms
2	物理导入	否	7m36s	0.3k (-95%)↓	2.3k (-68%)↓	990ms (+330%)↑
3	物理导入	是	8m10s	1.2k (-78%)↓	4.1k (-43%)↓	780ms (+239%)↑

资源管控-小结

通过Runaway Queries和后台任务特性，实现：

- 可以限制异常SQL对集群资源的使用，尽量避免对其他任务的性能影响。
- 对于大表添加索引DDL、Lightning数据导入、Analyze更新统计信息等系统任务，动态识别和限制资源使用，优先保证集群业务，大大提升集群的稳定性和可靠性。

应用场景：

- 自动识别并处理异常 SQL 性能问题，保障重要系统的服务质量。
- 对突发的 SQL 性能问题，在没有立即有效的修复手段时，可以对其限流，减少负面影响。
- 当已知个别 SQL 有安全或性能问题，可以加入黑名单进行限流。
- 通过设置后台任务，可以在任何时段在执行大表添加索引、Lightning导入数据、Analyze等系统任务，不再仅限于业务低估时期执行，大大提高运维管理的便捷性。

3 总结与展望

HTAP业务架构

多业务共用一个集群，在我们尽可能将 TP 业务和 AP 业务分离部署的前提下，通常还是会遇到下面的痛点问题。

依靠资源管控的功能优化，提高集群的资源使用效率，真正在实现降本增效的同时，大大提升集群的稳定性、可靠性。

高峰挤兑

- 当一个业务处于高峰期时，会过多占用集群资源，可能影响其他共存业务

✓ **希望能保护不同业务的资源持有情况，保证业务能分配到基本的运行资源而不被挤兑。**

低谷过剩

- 当集群中的重要业务处于低谷值时，有较多的剩余资源未被充分使用。

✓ **希望引入错峰运行的业务，充分使用资源，实现降本增效。同时要求业务能得到控制，其他时候不会占用过多资源。**

异常放大

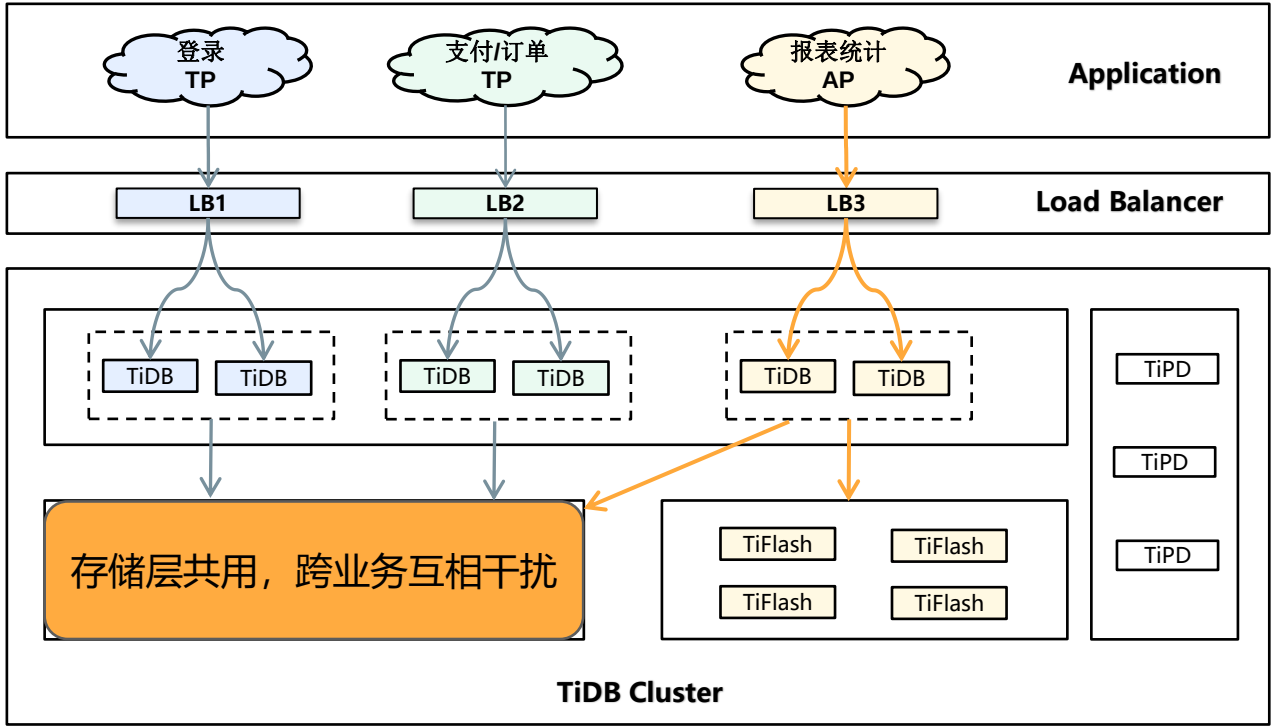
- 当集群有突发的异常 SQL 导致性能问题时，影响整个集群，只能停掉对应业务 。

✓ **希望不是干掉它的执行，而是临时限制它的资源消耗，允许它缓慢运行，但又不会影响集群其他业务。**

TiDB 多租户架构

架构说明

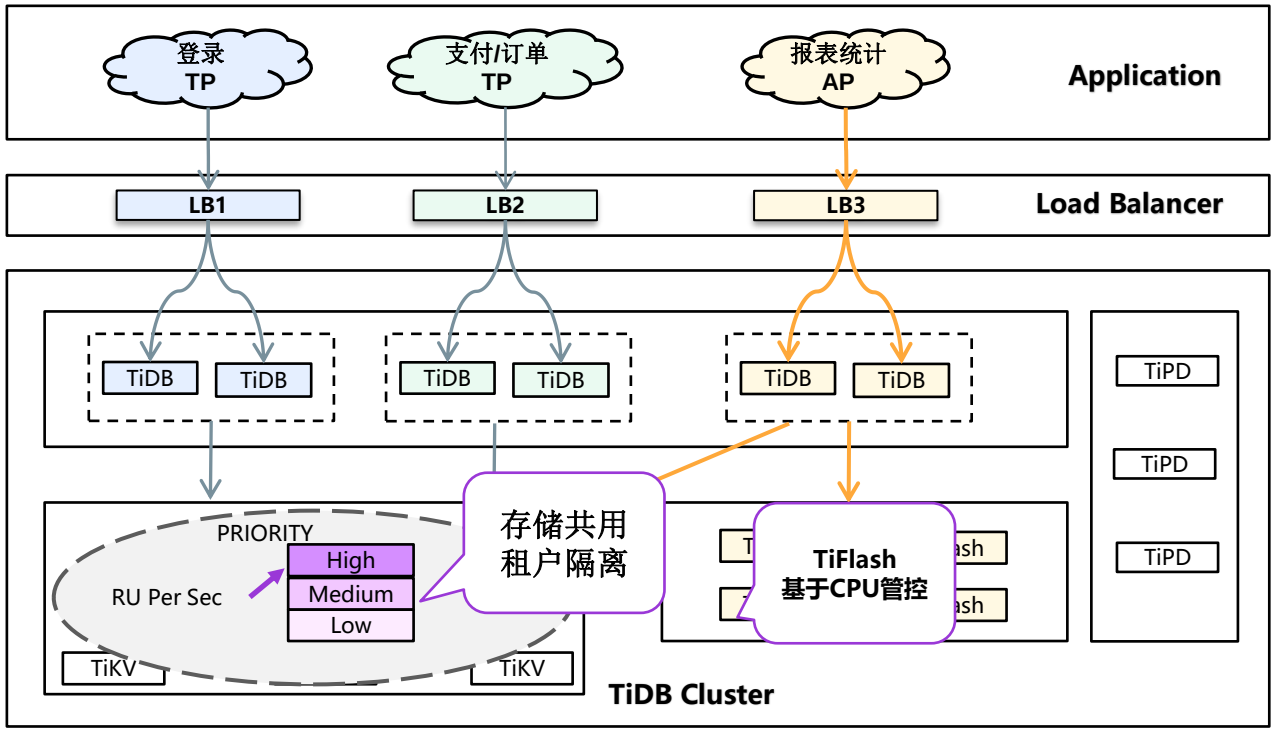
- 为不同业务分配不同租户资源组管控
- 为重要租户允许超额使用和高优先级，优先保证关键业务，兼顾其他业务



TiDB 多租户架构

架构说明

- 为不同业务分配不同租户资源组管控
- 为重要租户允许超额使用和高优先级，优先保证关键业务，兼顾其他业务



TiDB 多租户架构

架构优势

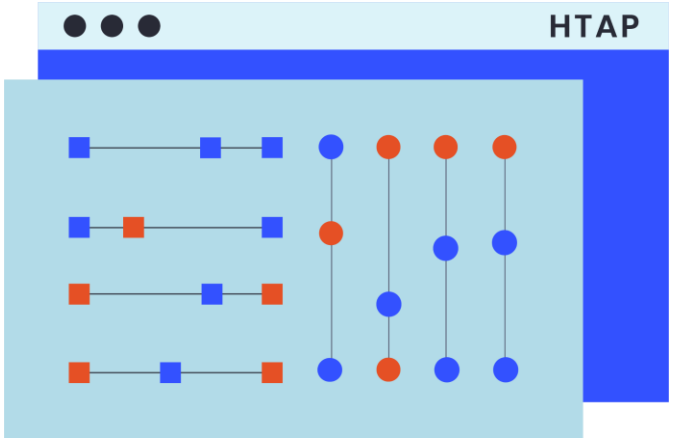
- **节约硬件成本**
  - 跨业务混合共用集群，不会相互影响，减少集群数量，降低部署成本
  - 错峰使用资源，提升整体利用率
- **高可扩展**
  - 低负载时，繁忙应用**可超越限额**使用资源， 提高系统的可扩展性
- **灵活管控资源**
  - 按需在线实时调整业务资源使用，充分利用资源
  - **在线限制负面SQL**，保障在线业务

架构劣势

- **复杂度高**
  - 系统复杂，技术实现较难
  - 集群管理门槛高
- **总资源计算不准**
  - 资源总量先硬件估计，运行后负载校准再人工调整，易出错
- **资源精准分配难**
  - 人工管理集群资源池，分配策略不好定
- **可视化观测不足**

未来展望

- **增加内存管控**
  - 将内存资源纳入 RU，进一步优化集群内存使用
- **支持更多样的方式**
  - 目前仅支持RU用量管控，需根据不同场景提供百分比、权重等多种资源限额定义方式
- **更智能的管控**
  - 通过分析历史运行数据，系统产出资源管理的建议报告
  - 智能生成管控规则自动调控，实现基于 AI 的智能自动化运维
- **提升易用性**
  - 引入**图形化管理**的方式进一步提升用户体验，全面提升相关功能的可观测性和易用性



TiDB | COMMUNITY

THANKS