



7.5 LTS 解读 | Runaway Queries 管理、高性能数据批处理方案、DDL 后停特性

TiDB Club

🕒 2024-03-11

👁 133

💬

本文将带您深入解读 TiDB 7.5 LTS 的关键特性，包括 Runaway Queries 管理、高性能数据批处理方案，以及 DDL 后停特性。帮助您更好地理解 TiDB 的功能，并在实际应用中实现更高效的数据处理。

01

Runaway Queries 管理：

提升 TiDB 稳定性的智能引擎

什么是 Runaway Queries

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。

Runaway Queries 管理旨在提供一种高效、可控、自动化的资源识别和管控机制，以降低突发 SQL 性能问题带来的负面影响，保护复杂工作负载下系统的稳定性，让 TiDB 更加可靠。

Runaway Queries 管理适用哪些场景

- 为了保障重要系统的服务质量，需要能够自动识别并处理异常 SQL 性能问题。
- 当遇到突发 SQL 性能问题，但又没有立即有效的修复手段时，希望临时缓解其影响。
- 当已知个别 SQL 有安全或性能问题，希望加入黑名单或对其进行限流。

Runaway Queries 管理能做什么

1 查询的识别

- 动态识别 - 根据运行时规则识别。
- 静态识别 - 根据 SQL 特征识别。

2 查询的处置

处置，指被识别到的 Runaway Queries 要如何处理。目前支持以下几个处理方式。

热门文章

- 解决方案 | 从 Oracle 到 TiDB 的迁移方案与实践

2023-05-281403浏览
- PingCAP 2023 年度报告

2024-01-021025浏览
- 白皮书 | 《实时数据服务平台 — 金融行业实时 HTAP 场景实践》

2022-05-201007浏览
- TiDB 6.5 新特性：TiFlash、在线 DDL、PITR 性能优化与原理解析

2023-03-02747浏览
- TiDB 7.0 发版

2023-04-10661浏览

最新文章

- TiDB 2024 年度报告：增长的故事

2025-02-0630浏览
- 案例合集 | TiDB 分布式数据库在银行业的创新实践

2025-01-1355浏览
- TiDB 8.5 LTS 发版——支持无限扩展，开启 AI 就绪新时代

2025-01-0636浏览
- 【TiDB 上海地区交流回顾】TiDB 在小红书、爱奇艺、咪咕、华安基金、数禾科...

2025-01-0247浏览
- TiDB 优化器系列：执行计划、统计信息与资源管控深度解析

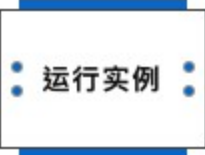
2024-12-2360浏览

- **DRYRUN**：仅识别不做处理，在日志和对应视图中显示。初期配置的时候，可以利用 DRYRUN试运行一段时间，检测是否有误判的风险。
- **COOLDOWN**：将查询置于资源组的最低优先级，限制其处理速度。
- **KILL**：终止被识别的查询，防止其进一步影响数据库性能。

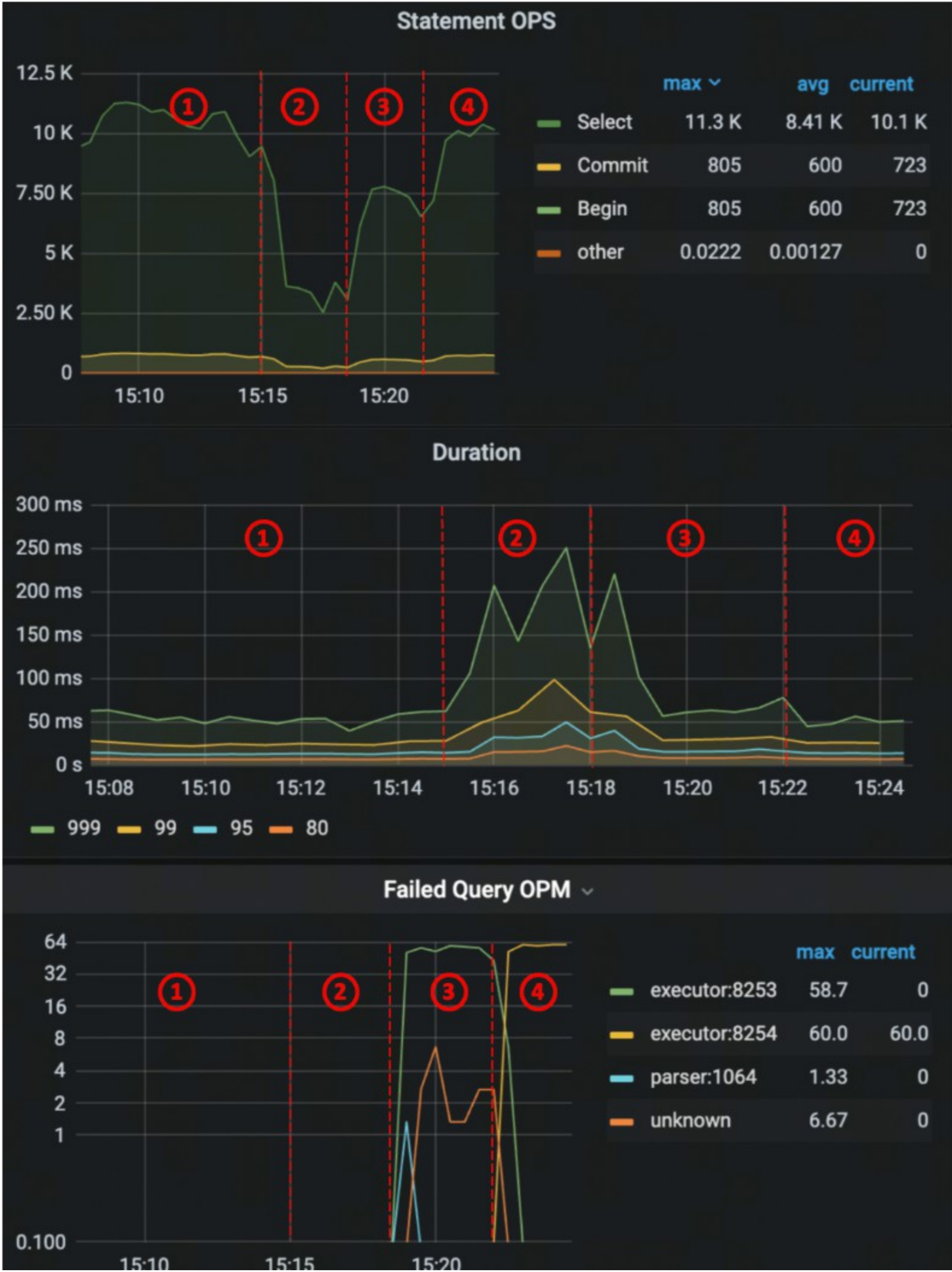
3 历史记录及观测性

以上所有的设置，及识别和处置的历史记录，TiDB 提供了一组系统表用于查询：

- **INFORMATION_SCHEMA.RESOURCE_GROUPS**：资源组定义，包括对 Runaway Queries 识别规则和处置设置。
- **INFORMATION_SCHEMA.RUNAWAY_WATCHES**：监控队列中的规则。
- **MYSQL.TIDB_RUNAWAY_QUERIES**：记录被识别和处置的 Runaway Queries 历史记录。



阶段	负载	资源组配置	QPS	P99
1	正常	无	11K	50ms
2	正常 + 持续注入1条高消耗查询	无	3K	200ms
3	正常 + 持续注入1条高消耗查询	QUERY_LIMIT=(EXEC_ELAPSED='1s', ACTION=KILL)	7.5K	70ms ~ 80ms
4	正常 + 持续注入1条高消耗查询	QUERY_LIMIT=(EXEC_ELAPSED='1s', ACTION=KILL, WATCH=EXACT DURATION='5m')	11K	50ms





1. 正常负载下，整体 QPS 接近 11k, P999 在 50ms 上下。
2. 出现一个异常查询，每秒提交一次，运行时间在 3~8 秒，QPS 从 11K 急剧下降至 3K 左右，P999 由 60ms增加到 200ms。
3. 这时我们尝试向 default 资源组加入一条规则，自动杀掉运行时间超过 1 秒的查询。QPS 回升至 7.5k，P999 下降。
4. 修改资源组规则，把符合 runaway 规则的查询的文本，加入到监控列表中，时长为 5 分钟。

在上述示例中，即使没有设置资源组对查询的自动识别，在出现 SQL 性能问题时，我们仍可以通过“慢日志”或者系统表找出问题查询的“特征”，用 QUERY WATCH 手工将查询加入监视列表，达到设置黑名单的效果。

[点击此处 | 查看原文](#)

02

TiDB 7.5.0 LTS 高性能数据批处理方案

• TiDB 上之前有哪些批处理方案 •

1. INSERT INTO ... SELECT 完成查询和写入
2. 针对 INSERT INTO/INSERT INTO ... ON DUPLICATE .../REPLACE INTO 这些 SQL 使用批量接口执行，降低应用与数据库之间的交互次数，提升批量写入时的性能
3. 通过 ETL 和调度平台提供的数据读取和写入能力实现大批量数据的处理
4. 针对上传过来的 csv 文件的数据，使用 LOAD DATA 来完成批量数据的写入，提升批量写入时的性能

针对以上几种批处理方案，以及最新推出的 IMPORT INTO 功能，我们开展了一次测试，探索哪种批处理方案效率最高，消耗资源更低，以及使用上更加简单。

• TiDB 中不同批处理方案的测试 •

1 测试环境

1. TiDB 资源：3 台 16VC/64GB 虚拟机 + 500GB SSD 云盘（3500 IOPS + 250MB/S 读写带宽）

a. TiDB 版本：TiDB V7.5.0 LTS

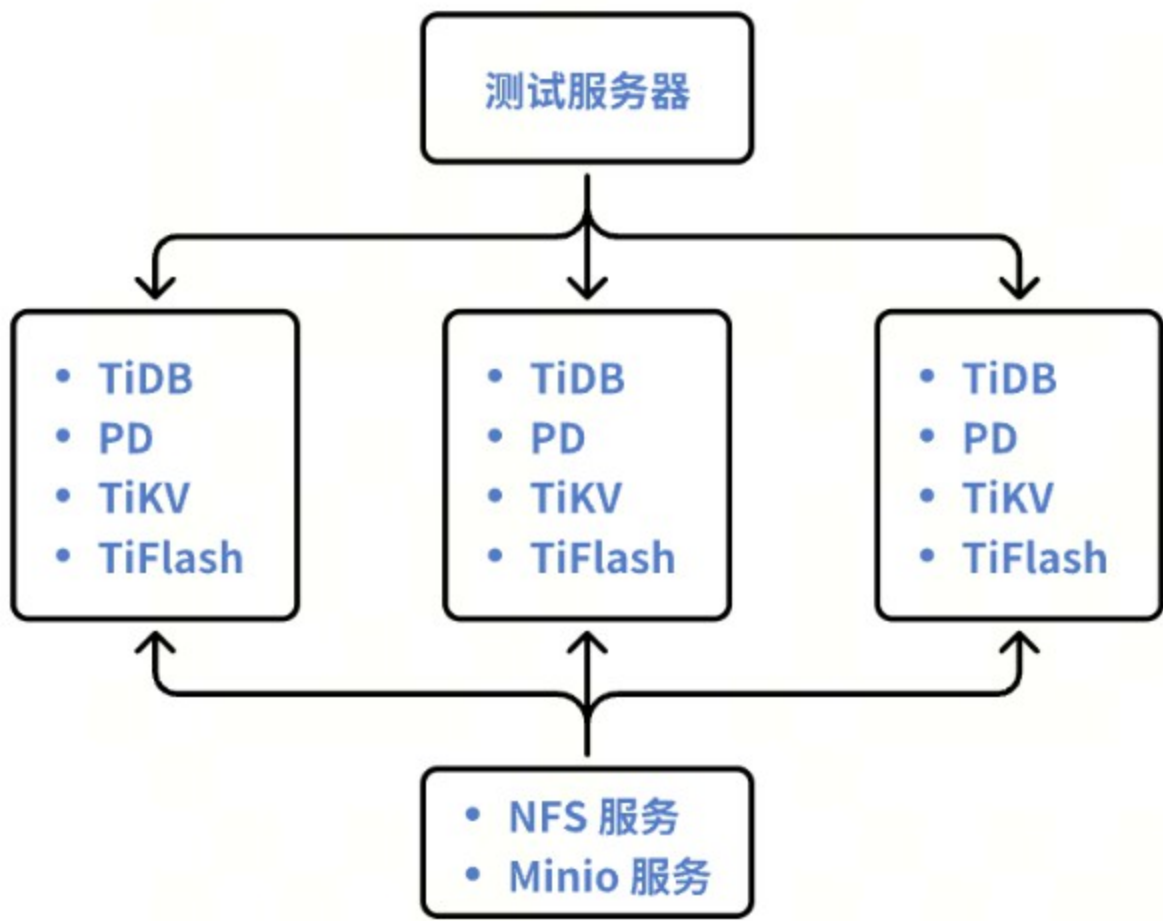
b. TiDB 组件：TiDB/PD/TiKV/TiFlash（混合部署）

2. 存储资源：8C/64GB 虚拟机 + 500GB SSD 云盘（3500 IOPS + 250MB/S 读写带宽）

● 存储服务：NFS 服务、Minio 对象存储

3. 测试资源：8C/64GB 虚拟机 + 500GB SSD 云盘（3500 IOPS + 250MB/S 读写带宽）

● datax + Dolphin 调度/java 程序/dumpling、tidb-lightning 工具以及 MySQL 客户端



2 测试场景

将大批量查询结果快速写入到目标表，既考验查询性能，同时也考验批量写入的性能。

2.1 查询部分：多表关联+聚合

2.2 写入：29 列，1 个主键+2 个索引

3 测试结果

场景	SQL/工具	耗时 秒	TiDB 节点 最大内存	TiDB 节点 最大 CPU	说明
一条 SQL： 查询+写入	INSERT INTO ... SELECT ...	586	23GB	1C	-
Java 程序： 查询+写入	Java 1 读 + 1 写 使用 streaming Resultset	940	15GB	1C	-
	Java 1 读 + 1 写 使用 limit 分批	24184	25GB	1C	错误示例
	Java 1 读 + 16 写	214	15GB	7C	16 个 queue
	Java 1 读 + 16 写（模拟写放大）	224			查询时 order by 非主键字段
	Java 1 读 + 16 写（模拟写入热点）	242			c_phone 字段使用固定值
ETL： 查询+写入	datax mysqlreader+mysqlwriter	1011	15.5GB	1C	多表 query 时，无法并发
导出 CSV	SQL: SELECT ... INTO OUTFILE	145	13.5GB	1C	导出到 NFS，单文件 3.2GB
	工具: dumpling	133	13.5GB	1C	导出到 NFS，12 个文件

导出 CSV	SQL: SELECT ... INTO OUTFILE	145	13.5GB	1C	导出到 NFS, 单文件 3.2GB
	工具: dumpling	133	13.5GB	1C	导出到 NFS, 12 个文件
导入 CSV	工具: tidb-lightning 单个文件	184	-	-	消耗的是应用节点资源
	工具: tidb-lightning 多个文件	181	-	-	(其中 analyze 18s)
	SQL: LOAD DATA 单个文件	509	11.6GB	1C	单个 3.2GB CSV
	SQL: LOAD DATA 多个文件+8并发	183	-	3C	拆分成 2 万行一个, 手动并行
	SQL: IMPORT INTO 单个文件	99	-	7C	具备自动切分文件功能
	SQL: IMPORT INTO 多个文件	97	-	7C	可导入dumpling生成的多个文件
	datax: txtfilereader 单个文件	1011	-	1C	单个 csv, 1 channel
	datax: txtfilereader 多个文件	231	-	6C	12 个 csv, 8 channel 并行

4 测试分析

4.1 JAVA 程序使用 SQL 进行批处理

4.2 LOAD DATA 方式

4.3 ETL+调度平台方式

4.4 SELECT ... INTO OUTFILE 导出查询结果（当前仅支持导出到文件系统）

4.5 IMPORT INTO 导入 CSV（当前支持 S3 协议对象存储以及文件系统）

5 测试小结

批处理方案	性能	内存消耗	使用复杂度
INSERT INTO ... SELECT	大数据量：性能一般	高	最简单（注意当需要使用 TiFlash 时，请将 sql_mode 指定为非严格模式）
JAVA 程序方式	高	中	较复杂，需要注意编码、以及规避热点
ETL 方式	性能一般	中	比较简单，但需要配合 ETL + 调度平台使用
导出： SELECT ... INTO OUTFILE	高	低	简单，SQL 可编码到程序 需挂载 NFS/对象存储到 TiDB 文件系统
导出： dumpling	高	低	复杂，难以编码到程序中，参数比较多
导入：LOAD DATA	高	取决于单个 CSV 大小	应用需要对 CSV 大小进行处理，同时应用需要多线程写入，以及规避热点问题
导入： IMPORT INTO	高	极低	简单，SQL 可编码到程序，不消耗应用节点 CPU 需挂载 NFS，或使用 s3 api 使用对象存储
导入： tidb-lightning	高	极低	复杂，难以编码到程序中，消耗应用节点 CPU 较高

[点击此处 | 查看原文](#)

03

TiDB v7.5.0 DDL 后停特性分析

• TiDB DDL 基本原理 •

TiDB 采用在线异步变更的方式执行 DDL 语句，从而实现 DDL 语句的执行不会阻塞其他会话中的 DML 语句。

1.1 逻辑及物理 DDL

按照是否需要操作表的数据来划分，DDL 语句可以划分为：

- 逻辑 DDL 语句
- 物理 DDL 语句

执行物理 DDL 对业务有影响主要在于两个方面，一方面需要从 TiKV 中读出和回填全表数据，可能会消耗较多的 CPU 及 I/O。另一方面，DDL Owner 所在的 TiDB 节点需要进行相应的计算，因此会消耗更多的 CPU 资源。所以，根据上面的特点，我们需要重点关注的是物理 DDL 造成的影响。

1.2 DDL Owner

TiDB 通过 etcd 选举出一个节点来担任 Owner，Owner 节点会主动续约自己的任期。当 Owner 节点宕机后，其他节点可以通过 etcd 感知到并重新选举出新的 Owner，在集群中继续担任 DDL 任务执行者的角色。我们可以通过 [ADMIN SHOW DDL](#) 语句查看当前 DDL owner 节点所在的 tidb-server。

TiDB 在线变更 DDL 通过多个小版本演进的方式，确保多个 TiDB 节点能够正确同步 DDL 变更前后的元数据，并保证期间执行用户事务更改数据的正确性与一致性。如下图是对于 add index 或者 add column DDL 的状态变更，对于 drop index 或者 drop column 则是完全相反的过程。



1.3 执行框架

在 v6.2.0 及之后，引入了并行 DDL 处理流程：

- DDL Owner 能够并行执行 DDL 任务。
- 改善了 DDL Job 队列先入先出的问题。DDL Owner 不再是 FIFO，而是选择当前可以执行的 DDL Job。
- 因为 TiDB 中的 DDL 都是在线变更，通过 Owner 即可对新的 DDL Job 进行相关性判断，并根据相关性结果进行 DDL 任务的调度，从而使分布式数据库实现了和传统数据库中 DDL 并发相同的效果。

1.4 DDL 暂停与恢复

在实际的生成环境中，添加索引可能会消耗大量资源并影响在线流量，如果是非常敏感的金融场景就更需要注意了。

即使在资源组中进行了限制，或对标记的节点进行了隔离，仍然可能需要在紧急情况下暂停这些任务。

所以，从 v7.2.0 开始支持同时暂停任意数量的后台任务，释放所需的资源，无需取消或重后任务，时候再继续以“断点续传”的方式继续 DDL 操作。这个功能在 v7.5.0 版本正式 GA，可以说是非常期待的。



2.1 操作命令

2.2 验证测试

2.3 小结

关于暂停 DDL 任务，总结几点注意内容：

- 手动 pause 操作可以暂停 DDL 作业。
- 只能暂停正在运行 running 或正在等待 queueing 的任务，否则会报错。
- 集群自动暂定的场景，只有版本升级时才会自动触发，其他情况不会自动暂停 DDL 作业。
- 在版本升级时，正在运行的 DDL 会被暂停，在升级过程中发起的 DDL 也将被暂停。升级结束后，所有已暂停的 DDL 作业将自动恢复执行。
- 正式执行中或排队中的 DDL 任务，可以直接暂停，任务暂停后不消耗集群资源。
- DDL pause 命令支持同时批量暂停多个 DDL 任务。



3.1 操作命令

3.2 验证测试

3.3 小结

- 处于非暂停状态中的作业无法被恢复，操作将失败。
- 只能恢复已被暂停的 DDL 作业。如果重复恢复同一个 DDL 作业，会报错。
- 可以同时恢复多个 DDL 作业。
- 版本升级时正在运行的 DDL 作业将被自动暂停，同时在升级过程中发起的 DDL 作业也将被自动暂停。升级结束后，所有已暂停的 DDL 作业将自动恢复执行。



欢迎来到金融专区！

与 PingCAP 一同打造创新的金融数据基础平台，

加速数字化和智能化转型！



 [tidb](#) [ddl](#) [lts](#) [批处理](#)

文章转载自TIDB Club，如果涉嫌侵权，请发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

评论

分享你的看法，一起交流吧~

相关阅读

- 【大盘点】2024年国产数据库行业有哪些大事发生？

墨天轮编辑部714次阅读2025-01-20 12:30:33
- 大年初一值班记：当重庆DBA在客户现场“捞”数据库的底料配方

李先生558次阅读2025-01-29 17:48:24
- TiDB x DeepSeek 打造更好用的国产知识库问答系统解决方案

严少安253次阅读2025-02-07 00:51:17
- 重磅！近4800万国产数据库大单落锤！达梦、海量数据库、金仓、虚谷、TiDB 五大厂商中标！网思科技双包中标~

天下观查177次阅读2025-02-05 07:00:45
- 新版本发布 | 全面兼容SQLServer，让迁移更丝滑

金仓数据库89次阅读2025-01-15 10:08:39
- OB Cloud 两项重大发布！跨云容灾再升级、AP 场景首个 LTS 版本即将上线

OceanBase81次阅读2025-01-21 09:56:11
- 一行代码不用写，用 Autoflow + Gitee AI 搭建本地知识库问答机器人

PingCAP50次阅读2025-01-24 10:03:56
- “TikTok 难民”涌入小红书，流量激增迎来用户暴增，而平台并未因流量过载崩溃且保持稳定运行背后的秘密武器是？

青年数据库学习互助会44次阅读2025-01-15 10:08:27
- 黄东旭：2025 数据库技术展望

数据库应用创新实验室42次阅读2025-02-06 10:07:08
- 百亿大表的实时分析：华安基金 HTAP 数据库的选型历程与 TiDB 使用体验

PingCAP41次阅读2025-01-20 09:47:53