TiDB Release Notes：从 v7.2.0 到 v7.5.0 的变更

PingCAP

20231201

Abstract

本文档包含 TiDB v7.2.0-DMR、v7.3.0-DMR、v7.4.0-DMR 和 v7.5.0-LTS 的 release notes。当你从 v7.1.x 升级到 v7.5.0 时，你可以参考本文档，以全面了解这些版本的新功能、兼容性变更、改进和错误修复。

关于 TiDB v7.5.0 的详细信息和使用文档，请参考 TiDB v7.5 文档。

目录

1 TiDB 7.5.0 Release Notes 4

1.1 功能详情 6

1.1.1 可扩展性 6

1.1.2 性能 7

1.1.3 数据库管理 8

1.1.4 可观测性 8

1.1.5 数据迁移 9

1.2 兼容性变更 10

1.2.1 系统变量 10

1.2.2 配置文件参数 11

1.3 离线包变更 12

1.4 废弃功能 13

1.5 改进提升 13

1.6 错误修复 14

1.7 性能测试 17

1.8 贡献者 17

2 TiDB 7.4.0 Release Notes 17

2.1 功能详情 21

2.1.1 可扩展性 21

2.1.2 性能 22

2.1.3 稳定性 24

2.1.4 SQL 功能 28

2.1.5 数据库管理 28

2.1.6 可观测性 29

2.1.7 数据迁移 29

2.2 兼容性变更 31

2.2.1 行为变更 31

2.2.2 系统变量 31

2.2.3 配置文件参数 33

2.3 废弃功能 34

2.4 改进提升 34

2.5 错误修复 36

2.6 贡献者 41

3 TiDB 7.3.0 Release Notes 42

3.1 功能详情 43

3.1.1 性能 43

3.1.2 稳定性 44

3.1.3 SQL 功能 45

3.1.4 可观测性 46

3.1.5 数据迁移 46

3.2 兼容性变更 48

3.2.1 行为变更 48

3.2.2 系统变量 48

3.2.3 配置文件参数 49

3.2.4 系统表 53

3.3 废弃功能 53

3.4 改进提升 53

3.5 错误修复 54
3.6 贡献者 57
4 TiDB 7.2.0 Release Notes 58
4.1 功能详情 59
4.1.1 性能 59
4.1.2 稳定性 61
4.1.3 SQL 功能 62
4.1.4 数据库管理 63
4.1.5 数据迁移 63
4.2 兼容性变更 64
4.2.1 系统变量 64
4.2.2 配置文件参数 65
4.3 改进提升 67
4.4 错误修复 69
4.5 贡献者 72

1 TiDB 7.5.0 Release Notes

发版日期：2023 年 12 月 1 日

TiDB 版本：7.5.0

试用链接：快速体验 | 生产部署 | 下载离线包

TiDB 7.5.0 为长期支持版本 (Long-Term Support Release, LTS)。

相比于前一个 LTS（即 7.1.0 版本），7.5.0 版本包含 7.2.0-DMR、7.3.0-DMR 和 7.4.0-DMR 中已发布的新功能、提升改进和错误修复。当你从 7.1.x 升级到 7.5.0 时，可以下载 TiDB Release Notes PDF 查看两个 LTS 版本之间的所有 release notes。下表列出了从 7.2.0 到 7.5.0 的一些关键特性：

分类	功能	描述
可扩展性与性能	支持并行运行多个 ADD INDEX 语句	通过该功能，为同一个表添加多个索引的任务可以变为并发运行。以前同时运行 2 个添加索引语句 X 和 Y 需要花费 X 的时间 + Y 的时间，现在在一个 SQL 语句中同时添加索引 X 和 Y，并发运行后，添加索引总耗时显著减少了。尤其是在宽表的场景，内部测试数据显示同时添加多个索引的性能最高可提升 94%。
稳定性与高可用	优化全局排序（实验特性，从 v7.4.0 开始引入）	TiDB v7.2.0 中引入了后端任务分布式并行执行框架。在 v7.4.0 中，TiDB 以该框架为基础，引入全局排序，消除了数据 reorg 任务期间临时无序数据导致的不必要的 I/O、CPU 和内存峰值。全局排序利用外部对象存储（目前为 Amazon S3）来存储系统作业期间的中间文件，提高灵活性并降低成本。ADD INDEX 和 IMPORT INTO 等操作将更快速灵活、稳定可靠，且运行成本较低。

分类	功能	描述
	资源管控支持自动管理后台任务（实验特性，从 v7.4.0 开始引入）	从 v7.1.0 开始，资源管控成为正式功能，该特性有助于缓解不同工作负载间的资源与存储访问干扰。TiDB v7.4.0 将此资源控制应用于后台任务的优先级。资源管控可以识别和管理后台任务执行的优先级，例如自动收集统计信息、备份和恢复、TiDB Lightning 批量数据导入以及在线 DDL。未来，所有后台任务都将纳入资源管控。
	资源管控支持管理资源消耗超出预期的查询（实验特性，从 v7.2.0 开始引入）	资源管控是一个通过资源组 (Resource Group) 对工作负载进行资源隔离的框架，但它并不对每个资源组内的查询产生影响。TiDB v7.2.0 引入了运行超出预期的查询 (Runaway Queries) 时的资源控制功能，你可以控制 TiDB 如何识别和处理每个资源组的查询。根据需要，长时间运行的查询可能会被终止或节流，你可以通过准确的 SQL 文本、SQL Digest 或 Plan Digest 来识别查询。在 TiDB v7.3.0，你可以主动监视已知的不良查询，类似于数据库级别的 SQL Blocklist。
SQL	MySQL 8.0 兼容性（从 v7.4.0 开始引入）	MySQL 8.0 的默认字符集为 utf8mb4，其默认排序规则是 utf8mb4_0900_ai_ci。TiDB v7.4.0 增强了与 MySQL 8.0 的兼容性。现在你可以更轻松地将在 MySQL 8.0 中使用默认排序规则创建的数据库迁移或复制到 TiDB。
数据库管理与	IMPORT INTO 语句集成 TiDB Lightning 物理导入模式的能力 (GA)	在 v7.2.0 之前，如需基于文件系统进行数据导入，你需要安装 TiDB Lightning 并使用其物理导入模式。目前，该功能已集成到 IMPORT INTO 语句中，你可以使用此语句快速导入数据，而无需安

分类	功能	描述
可观测性		装任何额外的工具。该语句还支持分布式执行框架，可分布式执行导入任务，提升了大规模数据导入时的效率。
	选择适用的 TiDB 节点分布式执行 ADD INDEX 或 IMPORT INTO SQL 语句 (GA)	你可以灵活选择在现有 TiDB 节点或新增 TiDB 节点执行 ADD INDEX 和 IMPORT INTO SQL 语句。该方法可以实现与其他 TiDB 节点的资源隔离，确保在执行上述语句时的最佳性能，并避免对已有业务造成性能影响。在 v7.5.0 中，该功能正式 GA。
	DDL 任务支持暂停和恢复操作 (GA)	添加索引可能会消耗大量资源并影响在线流量。即使在资源组中进行了限制，或对标记的节点进行了隔离，你仍然可能需要在紧急情况下暂停这些任务。从 v7.2.0 开始，TiDB 原生支持同时暂停任意数量的后台任务，释放所需的资源，无需取消或重启任务。
	TiDB Dashboard 性能分析支持 TiKV 堆内存分析	在之前版本中调查 TiKV OOM 或内存使用高的问题时，往往需要在实例环境下手动运行 jeprof 生成 Heap Profile。从 v7.5.0 开始，TiKV 支持远程处理 Heap Profile，你可以通过 TiDB Dashboard 直接获取 Heap Profile 的火焰图和调用图。该功能提供了与 Go 堆内存分析同等的简单易用体验。

1.1 功能详情

1.1.1 可扩展性

- 支持设置 TiDB 节点的服务范围，用于选择适用的 TiDB 节点分布式执行 ADD INDEX 或 IMPORT INTO 任务 (GA) #46258 @ywqzzy

在资源密集型集群中，并行执行 ADD INDEX 或 IMPORT INTO 任务可能占用大量 TiDB 节点的资源，从而导致集群性能下降。为了避免对已有业务造成性能影响，v7.4.0 以实验特性引入了变量 tidb_service_scope，用于控制 TiDB 后端任务分布式框架下各 TiDB 节点的服务范围。你可以从现有 TiDB 节点中选择几个节点，或者对新增 TiDB 节点设置服务范围，所有分布式执行的 ADD INDEX 和 IMPORT INTO 的任务只会运行在这些节点。在 v7.5.0 中，该功能正式 GA。

更多信息，请参考用户文档。

1.1.2 性能

- TiDB 后端任务分布式并行执行框架成为正式功能 (GA)，提升并行执行的 ADD INDEX 或 IMPORT INTO 任务的性能和稳定性 #45719 @wjhuang2016

  在 v6.6.0 中引入的 TiDB 后端任务分布式并行执行框架成为正式功能 (GA)。TiDB v7.1.0 之前的版本中，在同一时间只有一个 TiDB 节点能够执行 DDL 任务。从 v7.1.0 开始，在分布式并行执行框架下，多个 TiDB 节点可以并行执行同一项 DDL 任务。从 v7.2.0 开始，分布式并行执行框架支持多个 TiDB 节点并行执行同一个 IMPORT INTO 任务，从而更好地利用 TiDB 集群的资源，大幅提升 DDL 和 IMPORT INTO 任务的性能。此外，你还可以通过增加 TiDB 节点来线性提升 DDL 和 IMPORT INTO 任务的性能。

  如果要使用分布式并行执行框架，只需将 tidb_enable_dist_task 的值设置为 ON：

  ```
  SET GLOBAL tidb_enable_dist_task = ON;
  ```

  更多信息，请参考用户文档。

- 提升了在一个 SQL 语句中同时添加多个索引的性能 #41602 @tangenta

  在 v7.5.0 之前，用户在一个 SQL 语句里添加多个索引 (ADD INDEX) 时，其性能与使用多个独立的 SQL 语句添加多个索引的性能接近。从 v7.5.0 起，

在一个 SQL 语句中添加多个索引的性能有了显著的变化，尤其是在宽表的场景，内部测试数据显示最高性能可提升 94%。

1.1.3 数据库管理

- DDL 任务支持暂停和恢复操作成为正式功能 (GA) #18015 @godouxm

  在 v7.2.0 中引入的 DDL 任务的暂停和恢复功能成为正式功能 (GA)。该功能允许临时暂停资源密集型的 DDL 操作（如创建索引），以节省资源并最小化对在线流量的影响。当资源允许时，你可以无缝恢复 DDL 任务，而无需取消和重新开始。DDL 任务的暂停和恢复功能提高了资源利用率，改善了用户体验，并简化了 schema 变更过程。

  你可以通过如下 ADMIN PAUSE DDL JOBS 或 ADMIN RESUME DDL JOBS 语句暂停或者恢复多个 DDL 任务：

  ```
  ADMIN PAUSE DDL JOBS 1,2;
  ADMIN RESUME DDL JOBS 1,2;
  ```

  更多信息，请参考用户文档。

- BR 支持备份和恢复统计信息 #48008 @Leavrth

  从 TiDB v7.5.0 开始，BR 备份工具支持备份和恢复数据库统计信息，在备份命令中引入了参数 --ignore-stats。当指定该参数值为 false 时，BR 备份工具支持备份和恢复数据库的列、索引、和表级别的统计信息。因此，从备份中恢复的 TiDB 数据库不再需要手动运行统计信息收集任务，也无需等待自动收集任务的完成，从而简化了数据库维护工作，并提升了查询性能。

  更多信息，请参考用户文档。

1.1.4 可观测性

- TiDB Dashboard 性能分析支持 TiKV 堆内存分析 #15927 @Connor1996

在之前版本中调查 TiKV OOM 或内存使用高的问题时，往往需要在实例环境下手动运行 jeprof 生成 Heap Profile。从 v7.5.0 开始，TiKV 支持远程处理 Heap Profile，你可以通过 TiDB Dashboard 直接获取 Heap Profile 的火焰图和调用图。该功能提供了与 Go 堆内存分析同等的简单易用体验。

更多信息，请参考用户文档。

1.1.5 数据迁移

- IMPORT INTO SQL 语句成为正式功能 (GA) #46704 @D3Hunter

  在 v7.5.0 中，IMPORT INTO SQL 语句正式 GA。该语句集成了 TiDB Lightning 物理导入模式的能力，可以将 CSV、SQL 和 PARQUET 等格式的数据快速导入到 TiDB 的一张空表中。这种导入方式无需单独部署和管理 TiDB Lightning，在降低了数据导入难度的同时，大幅提升了数据导入效率。

  更多信息，请参考用户文档。

- Data Migration (DM) 支持拦截不兼容（破坏数据一致性）的 DDL 变更 #9692 @GMHDBJD

  在 v7.5.0 之前，使用 DM 的 Binlog Filter 功能只能迁移或过滤指定的 Event，且颗粒度比较粗，例如只能过滤 ALTER 这种大颗粒度的 DDL Event。这种方式在某些业务场景会受限，如业务允许 ADD COLUMN，但是不允许 DROP COLUMN，但之前的 DM 版本都会被 ALTER Event 过滤。

  因此，v7.5.0 细化了 DDL Event 的处理粒度，如支持过滤 MODIFY COLUMN（修改列数据类型）、DROP COLUMN 等会导致数据丢失、数据被截断、精度损失等问题的细粒度 DDL Event。你可以按需配置。同时还支持拦截不兼容的 DDL 变更，并报错提示，你可以及时介入手工处理，避免对下游的业务数据产生影响。

  更多信息，请参考用户文档。

- 支持实时更新增量数据校验的 checkpoint #8463 @lichunzhu

  在 v7.5.0 之前，你可以使用增量数据校验功能来判断 DM 同步到下游的数据是否与上游一致，并以此作为业务流量从上游数据库割接到 TiDB 的依据。然而，由于增量校验 checkpoint 受到较多限制，如同步延迟、不一致的数据等待重新校验等因素，需要每隔几分钟刷新一次校验后的 checkpoint。对于某些只有几十秒割接时间的业务场景来说，这是无法接受的。

  v7.5.0 引入实时更新增量数据校验的 checkpoint 后，你可以传入上游数据库指定的 binlog 位置。一旦增量校验程序在内存里校验到该 binlog 位置，会立即刷新 checkpoint，而不是每隔几分钟刷新 checkpoint。因此，你可以根据该立即返回的 checkpoint 快速进行割接操作。

  更多信息，请参考用户文档。

1.2 兼容性变更

注意：

以下为从 v7.4.0 升级至当前版本 (v7.5.0) 所需兼容性变更信息。如果从 v7.3.0 或之前版本升级到当前版本，可能也需要考虑和查看中间版本 Release Notes 中提到的兼容性变更信息。

1.2.1 系统变量

变量名	修改类型	描述
tidb_enable_fast_analyze	废弃	用于控制是否启用统计信息快速分析功能。自 v7.5.0 起，统计信息快速分析功能被废弃。
tidb_analyze_partition_concurrency	修改	经进一步的测试后，默认值由 1 改为 2。
tidb_build_stats_concurrency	修改	经进一步的测试后，默认值由 4 改为 2。

变量名	修改类型	描述
tidb_merge_partition_stats_concurrency	修改	该变量从 v7.5.0 开始生效，用于设置 TiDB analyze 分区表时，对分区表统计信息进行合并时的并发度。
tidb_build_sampling_stats_concurrency	新增	设置 ANALYZE 过程中的采样并发度。
tidb_enable_async_merge_global_stats	新增	设置 TiDB 使用异步方式合并统计信息，以避免 OOM 问题。
tidb_gogc_tuner_max_value	新增	控制 GOGC Tuner 可调节 GOGC 的最大值。
tidb_gogc_tuner_min_value	新增	控制 GOGC Tuner 可调节 GOGC 的最小值。

1.2.2 配置文件参数

配置文件	配置项	修改类型	描述
TiKV	raftstore.region-compact-min-redundant-rows	修改	触发 RocksDB compaction 需要的冗余的 MVCC 数据行数。从 v7.5.0 开始，该配置项对 "raft-kv" 存储引擎生效。
TiKV	raftstore.region-compact-redundant-rows-percent	修改	触发 RocksDB compaction 需要的冗余的 MVCC 数据行所占比例。从 v7.5.0 开始，该配置项对 "raft-kv" 存储引擎生效。
TiKV	raftstore.evict-cache-on-memory-ratio	新增	当 TiKV 的内存使用超过系统可用内存的 90%，并且 Raft 缓存条目占用的内存超过已使用内存的 evict-cache-on-memory-ratio 比例时，TiKV 会逐出 Raft 缓存条目。
TiKV	memory.enable-heap-profiling	新增	控制是否开启 TiKV 堆内存分析功能，以跟踪 TiKV 的内存使用情况。
TiKV	memory.profiling-sample-per-bytes	新增	设置 TiKV 堆内存分析每次采样的数据量，以 2 的指数次幂向上取整。

配置文件	配置项	修改类型	描述
BR	--ignore-stats	新增	用于备份和恢复数据库统计信息。当指定该参数值为 false 时，BR 备份工具支持备份和恢复数据库的列、索引、和表级别的统计信息。
TiCDC	case-sensitive	修改	经进一步的测试后，默认值由 true 改为 false，即默认情况下 TiCDC 配置文件中涉及的表名、库名大小写不敏感。
TiCDC	sink.dispatchers.partition	修改	控制增量数据的 Kafka Partition 分发策略，可选值新增 columns 选项，即使用明确指定的列值计算 partition 编号。
TiCDC	sink.column-selectors	新增	控制 TiCDC 将增量数据分发到 Kafka 时，只发送指定的列的数据变更事件。
TiCDC	sql-mode	新增	设置 TiCDC 解析 DDL 时使用的 SQL 模式，默认值和 TiDB 的默认 SQL 模式一致。
TiDB Lightning	--importer	删除	该配置项用于指定 TiKV-importer 的地址。从 v7.5.0 起，TiKV-importer 组件被废弃。

1.3 离线包变更

从 v7.5.0 开始，TiDB-community-toolkit 二进制软件包中移除了以下内容：

- tikv-importer-{version}-linux-{arch}.tar.gz
- mydumper
- spark-{version}-any-any.tar.gz
- tispark-{version}-any-any.tar.gz

1.4 废弃功能

- Mydumper 在 v7.5.0 中废弃，其绝大部分功能已经被 Dumpling 取代，强烈建议切换到 Dumpling。
- TiKV-importer 组件在 v7.5.0 中废弃，建议使用 TiDB Lightning 物理导入模式作为替代方案。
- 从 v7.5.0 开始，不再提供 TiDB Binlog 数据同步功能的技术支持，强烈建议使用 TiCDC 实现高效稳定的数据同步。尽管 TiDB Binlog 在 v7.5.0 仍支持 Point-in-Time Recovery (PITR) 场景，但是该组件在未来 LTS 版本中将被完全废弃，推荐使用 PITR 替代。
- 统计信息的快速分析（实验特性）在 v7.5.0 中废弃。
- 统计信息的增量收集（实验特性）在 v7.5.0 中废弃。

1.5 改进提升

- TiDB
  - 优化合并 GlobalStats 的并发模型：引入 tidb_enable_async_merge_global_stats 实现同时加载统计信息并进行合并，从而加速分区表场景下 GlobalStats 的生成。同时优化合并 GlobalStats 的内存使用，以避免 OOM 并减少内存分配 #47219 @hawkingrei
  - 优化 ANALYZE 流程：引入 tidb_build_sampling_stats_concurrency 精细化控制 ANALYZE 并发度，减少资源消耗。同时优化 ANALYZE 的内存使用，通过复用部分中间结果，减少内存分配，避免频繁 GC #47275 @hawkingrei
  - 改进 Placement Policy 的使用：增加对全局范围的策略配置，完善常用场景的语法支持 #45384 @nolouch

- 提升启用索引加速功能 tidb_ddl_enable_fast_reorg 后添加索引的性能，在内部测试中 v7.5.0 相比 v6.5.0 性能最高提升 62.5% #47757 @tangenta

- TiKV
  - 避免写 Titan manifest 文件时持有锁导致影响其他线程 #15351 @Connor1996
- PD
  - 提升 evict-slow-trend 调度的稳定性和易用性 #7156 @LykxSassinato
- Tools
  - Backup & Restore (BR)
    - 快照备份新增表间备份参数 table-concurrency，用于控制统计信息备份、数据校验等元信息的表间并发度 #48571 @3pointer
    - 快照备份恢复在遇到某些网络错误时会进行重试 #48528 @Leavrth

1.6 错误修复

- TiDB
  - 禁止非整型聚簇索引进行 split table 操作 #47350 @tangenta
  - 修复使用错误的时区信息对时间字段进行编码的问题 #46033 @tangenta
  - 修复 Sort 算子在落盘过程中可能导致 TiDB 崩溃的问题 #47538 @windtalker
  - 修复查询使用 GROUP_CONCAT 时报错 Can't find column 的问题 #41957 @AilinKid
  - 修复 client-go 中 batch-client panic 的问题 #47691 @crazycs520
  - 修复 INDEX_LOOKUP_HASH_JOIN 内存使用量估算错误的问题 #47788 @SeaRise

  - 修复长时间下线的 TiFlash 节点重新加入集群后造成的负载不均衡的问题 #35418 @windtalker
  - 修复 HashJoin 算子 Probe 时无法复用 chunk 的问题 #48082 @wshwsh12
  - 修复 COALESCE() 函数对于 DATE 类型参数返回结果类型不正确的问题 #46475 @xzhangxian1008
  - 修复带子查询的 UPDATE 语句被错误地转成 PointGet 的问题 #48171 @hi-rustin
  - 修复当被缓存的执行计划包含日期类型和 unix_timestamp 的比较时，结果出现错误的问题 #48165 @qw4990
  - 修复默认内联且带聚合函数或窗口函数的公共表表达式 (CTE) 被递归的 CTE 引用时会报错的问题 #47881 @elsa0520
  - 修复优化器为减少窗口函数引入的 sort 而错误地选择了 IndexFullScan 的问题 #46177 @qw4990
  - 修复当 CTE 被多次引用时，条件下推 CTE 导致结果错误的问题 #47881 @winoros
  - 修复了 MySQL 压缩协议无法处理超大负载数据 (>= 16M) 的问题 #47152 #47157 #47161 @dveeden
  - 修复 TiDB 在通过 systemd 启动时无法读取 cgroup 资源限制的问题 #47442 @hawkingrei
- TiKV
  - 修复悲观事务中 prewrite 请求重试在极少数情况下影响数据一致性的风险 #11187 @MyonKeminta
- PD
  - 修复 evict-leader-scheduler 丢失配置的问题 #6897 @HuSharp
  - 修复 store 下线后对应的统计数据监控指标未删除的问题 #7180 @rleungx

  - 修复采用自适应同步部署模式 (DR Auto-Sync) 的集群在 Placement Rule 的配置较复杂时，canSync 和 hasMajority 可能计算错误的问题 #7201 @disksing
  - 修复 rule checker 未按照设定的 Placement Rule 添加 Learner 的问题 #7185 @nolouch
  - 修复 TiDB Dashboard 不能正常读取 PD trace 数据的问题 #7253 @nolouch
  - 修复 PD 内部获取的 Region 可能为空导致 PD Panic 的问题 #7261 @lhy1024
  - 修复采用自适应同步部署模式 (DR Auto-Sync) 的集群 available_stores 计算错误的问题 #7221 @disksing
  - 修复当 TiKV 节点不可用时 PD 可能删除正常 Peers 的问题 #7249 @lhy1024
  - 修复在大集群中添加多个 TiKV 节点可能导致 TiKV 心跳上报变慢或卡住的问题 #7248 @rleungx
- TiFlash
  - 修复 UPPER() 和 LOWER() 函数在 TiDB 和 TiFlash 中计算结果不一致的问题 #7695 @windtalker
  - 修复在空分区上执行查询报错的问题 #8220 @JaySon-Huang
  - 修复同步 TiFlash 副本时可能创建表失败导致 panic 的问题 #8217 @hongyunyan
- Tools
  - Backup & Restore (BR)
    - 修复 PITR 可能跳过恢复 CREATE INDEX DDL 的问题 #47482 @Leavrth
    - 修复大宽表场景下，日志备份在某些场景中可能卡住的问题 #15714 @YuJuncen
  - TiCDC

- 修复同步数据到对象存储时访问 NFS 目录导致的性能问题 #10041 @CharlesCheung96
- 修复开启 claim-check 导致存储路径拼写错误的问题 #10036 @3AceShowHand
- 修复 TiCDC 在某些情况下调度不均衡的问题 #9845 @3AceShowHand
- 修复同步数据到 Kafka 时可能卡住的问题 #9855 @hicqu
- 修复某些场景下 TiCDC Processor 可能 panic 的问题 #9849 #9915 @hicqu @3AceShowHand
- 修复开启 kv-client.enable-multiplexing 导致同步任务卡住的问题 #9673 @fubinzh
- 修复开启 Redo log 时，NFS 出现故障导致 Owner 节点卡住的问题 #9886 @3AceShowHand

1.7 性能测试

如需了解 TiDB v7.5.0 的性能表现，你可以参考 TiDB Dedicated 集群的 TPC-C 性能测试报告和 Sysbench 性能测试报告（英文版）。

1.8 贡献者

感谢来自 TiDB 社区的贡献者们：

- jgrande（首次贡献者）
- shawn0915

2 TiDB 7.4.0 Release Notes

发版日期：2023 年 10 月 12 日

TiDB 版本：7.4.0

试用链接：快速体验 | 下载离线包

在 7.4.0 版本中，你可以获得以下关键特性：

分类	功能	描述
稳定性与高可用	引入全局排序能力，提升 IMPORT INTO 和 ADD INDEX 任务的性能和稳定性（实验特性）	在 v7.4.0 以前，使用分布式并行执行框架执行 ADD INDEX 或 IMPORT INTO 等任务时，只能对部分数据进行局部排序。这导致 TiKV 需要采取额外操作，并且在将数据导入到 TiKV 之前，TiDB 节点还需要为其分配本地磁盘空间以进行排序。 随着 v7.4.0 引入全局排序特性，可以将数据暂时存储在外部存储（如 S3）中进行全局排序后再导入到 TiKV 中。这一改进降低了 TiKV 对资源的额外消耗，并显著提高了 ADD INDEX 和 IMPORT INTO 等操作的性能和稳定性。
	资源管控支持自动管理后台任务（实验特性）	从 v7.1.0 开始，资源管控成为正式功能，该特性有助于缓解不同工作负载间的资源与存储访问干扰。TiDB v7.4.0 将此资源控制应用于后台任务。资源管控可以识别和管理

分类	功能	描述
		后台任务，例如自动收集统计信息、备份和恢复、TiDB Lightning 批量数据导入以及在线 DDL。未来，所有后台任务都将纳入资源管控。
	TiFlash 支持存储计算资源分离和 S3 共享存储 (GA)	TiFlash 存算分离架构和 S3 共享存储成为正式功能： • 支持分离 TiFlash 的存储和计算资源，提升 HTAP 资源的弹性能力。 • 支持基于 S3 的存储引擎，以更低的成本提供共享存储。
SQL	TiDB 支持完整的分区类型管理功能	在 v7.4.0 之前，Range/List 分区表支持分区管理操作包括 TRUNCATE、EXCHANGE、ADD、DROP、REORGANIZE 等，Hash/Key 分区表支持分区管理操作包括 ADD 和 COALESCE 等。

分类	功能	描述
		现在 TiDB 新增支持了以下分区类型管理操作： • 将分区表转换为非分区表 • 对现有的非分区表进行分区 • 修改现有分区表的分区类型
	MySQL 8.0 兼容性：支持排序规则 utf8mb4_0900_ai_ci	MySQL 8.0 的一个显著变化是默认字符集更改为 utf8mb4，其默认排序规则是 utf8mb4_0900_ai_ci。TiDB v7.4.0 增强了与 MySQL 8.0 的兼容性。现在你可以更轻松地将在 MySQL 8.0 中使用默认排序规则创建的数据库迁移或复制到 TiDB。
数据库管理与可观测性	选择适用的 TiDB 节点来并行执行 ADD INDEX 或 IMPORT INTO SQL 语句（实验特性）	你可以选择在现有 TiDB 节点、或者新增 TiDB 节点执行 ADD INDEX 和 IMPORT INTO SQL 语句。该方法可以实现与其他 TiDB 节点的资源隔离，确保在执行上述语句

分类	功能	描述
		时的最佳性能，并避免对已有业务造成性能影响。

2.1 功能详情

2.1.1 可扩展性

- 支持设置 TiDB 节点的服务范围，用于选择适用的 TiDB 节点来执行并行的 ADD INDEX 或 IMPORT INTO 任务（实验特性）#46453 @ywqzzy

  在资源密集型集群中，并行执行 ADD INDEX 或 IMPORT INTO 任务可能占用大量 TiDB 节点的资源，从而导致集群性能下降。从 v7.4.0 起，你可以通过变量 tidb_service_scope 控制 TiDB 后端任务分布式框架下各 TiDB 节点的服务范围。你可以从现有 TiDB 节点中选择几个节点，或者对新增 TiDB 节点设置服务范围。所有并行执行的 ADD INDEX 和 IMPORT INTO 的任务只会运行在这些节点，避免对已有业务造成性能影响。

  更多信息，请参考用户文档。

- 增强 Partitioned Raft KV 存储引擎（实验特性）#11515 #12842 @busyjay @tonyxuqqi @tabokie @bufferflies @5kbpers @SpadeA-Tang @nolouch

  TiDB v6.6.0 引入了 Partitioned Raft KV 存储引擎作为实验特性，该引擎使用多个 RocksDB 实例存储 TiKV 的 Region 数据，每个 Region 的数据都独立存储在单独的 RocksDB 实例中。

  在 TiDB v7.4.0 中，Partitioned Raft KV 引擎在兼容性和稳定性方面得到了进一步提升。通过大规模数据测试，确保了 Partitioned Raft KV 引擎与 DM、Dumpling、TiDB Lightning、TiCDC、BR、PITR 等关键生态组件或功能的兼容性。同时，在读写混合工作负载下，Partitioned Raft KV 引擎提供了更稳定的性能，特别适合写多读少的场景。此外，每个 TiKV 节点支持 8 core CPU，并可搭配 8 TB 的数据存储和 64 GB 的内存。

更多信息，请参考用户文档。

- TiFlash 存算分离架构成为正式功能 (GA) #6882 @JaySon-Huang @JinheLin @breezewish @lidezhu @CalvinNeo @Lloyd-Pottiger

  在 v7.0.0 中，TiFlash 以实验特性引入了存算分离架构。经过一系列的改进，从 v7.4.0 起，TiFlash 正式支持存算分离架构。

  在存算分离架构下，TiFlash 节点分为 Compute Node（计算节点）和 Write Node（写入节点）两种类型，并使用兼容 S3 API 的对象存储。这两种节点都可以单独扩缩容，独立调整计算或数据存储能力。在存算分离架构下，TiFlash 的使用方式与存算一体架构一致，例如创建 TiFlash 副本、查询、指定优化器 Hint 等。

  需要注意的是，TiFlash 的存算分离架构和存算一体架构不能混合使用、相互转换，需要在部署 TiFlash 时进行相应的配置指定使用其中的一种架构。

  更多信息，请参考用户文档。

2.1.2 性能

- 支持下推 JSON 运算符 MEMBER OF 到 TiKV #46307 @wshwsh12

  - value MEMBER OF(json_array)

  更多信息，请参考用户文档。

- 支持下推包含任意帧定义类型的窗口函数到 TiFlash #7376 @xzhangxian1008

  在 v7.4.0 之前的版本中，TiFlash 不支持包含 PRECEDING 或 FOLLOWING 的窗口函数，所有包含此类帧定义的窗口函数都无法下推至 TiFlash。从 v7.4.0 开始，TiFlash 支持了所有的窗口函数的帧定义。该功能自动启用，满足要求时，包含帧定义的窗口函数会自动下推至 TiFlash 执行。

- 引入基于云存储的全局排序能力，提升并行执行的 ADD INDEX 或 IMPORT INTO 任务的性能和稳定性（实验特性）#45719 @wjhuang2016

  在 v7.4.0 以前，当用户执行分布式并行执行框架的 ADD INDEX 或 IMPORT INTO 任务时，TiDB 节点需要准备一块较大的本地磁盘，对编码后的索引 KV pairs 和表数据 KV pairs 进行排序。由于无法从全局角度进行排序，各个 TiDB 节点间以及节点内部导入的数据可能存在重叠情况。这会导致在将这些 KV pairs 导入到 TiKV 时，TiKV 需要频繁进行数据整理 (compaction)，降低了 ADD INDEX 或 IMPORT INTO 的性能和稳定性。

  v7.4.0 引入全局排序特性后，编码后的数据不再写入本地进行排序，而是写入云存储，并在云存储中进行全局排序。然后，TiDB 将经过全局排序的索引数据和表数据并行导入到 TiKV 中，从而提升了性能和稳定性。

  更多信息，请参考用户文档。

- 支持缓存非 Prepare 语句的执行计划 (GA) #36598 @qw4990

  TiDB v7.0.0 引入了非 Prepare 语句的执行计划缓存作为实验特性，以提升在线交易场景的并发处理能力。在 v7.4.0 中，该功能正式 GA。执行计划缓存技术将会被应用于更广泛的场景，从而提升 TiDB 的并发处理能力。

  开启非 Prepare 语句执行计划缓存可能会带来额外的内存和 CPU 开销，并不一定适用于所有场景。从 v7.4.0 开始，非 Prepare 语句的执行计划缓存默认关闭。你可以通过系统变量 tidb_enable_non_prepared_plan_cache 控制是否开启该功能并通过 tidb_session_plan_cache_size 设置缓存大小。

  此外，该功能默认不支持 DML 语句，对 SQL 的模式也有一定的限制，具体参见使用限制。

  更多信息，请参考用户文档。

2.1.3 稳定性

- TiFlash 支持查询级别的数据落盘 #7738 @windtalker

  从 v7.0.0 起，TiFlash 支持控制 GROUP BY、ORDER BY、JOIN 这三种算子的数据落盘功能，避免数据量超过内存总大小时，导致查询终止甚至系统崩溃的问题。然而，单独控制每个算子的落盘较为麻烦，也无法有效进行整体资源控制。

  在 v7.4.0 中，TiFlash 引入了查询级别数的据落盘功能。通过设置单个查询在单个 TiFlash 节点使用内存的上限 tiflash_mem_quota_query_per_node 及触发数据落盘的内存阈值 tiflash_query_spill_ratio，你可以方便地控制单个查询的内存使用，更好地管控 TiFlash 内存资源。

  更多信息，请参考用户文档。

- 支持自定义 TiKV 读取超时时间 #45380 @crazycs520

  在通常情况下，TiKV 处理请求非常快，只需几毫秒。但是，当某个 TiKV 节点遇到磁盘 I/O 抖动或网络延迟时，请求处理时间可能会大幅增加。在 v7.4.0 以前的版本中，TiKV 请求的超时限制是固定的，不能调整。因此，当 TiKV 节点出现问题时，TiDB 必须等待固定时长的超时响应，这导致了抖动期间应用程序的查询性能受到明显影响。

  TiDB 在 v7.4.0 中引入了一个新系统变量 tikv_client_read_timeout，你可以自定义查询语句中 TiDB 发送给 TiKV 的 RPC 读请求的超时时间。这意味着，当某个 TiKV 节点因磁盘或网络问题导致请求延迟时，TiDB 可以更快地超时并将请求重新发送给其他 TiKV 节点，从而降低查询延迟。如果所有 TiKV 节点的请求都超时，TiDB 将使用默认的超时时间进行重试。此外，你也可以在查询语句中使用 Optimizer Hint /*+ SET_VAR(TIKV_CLIENT_READ_TIMEOUT=N) */ 来设置 TiDB 发送 TiKV RPC 读

请求的超时时间。这一改进将使 TiDB 在面对不稳定的网络或存储环境时，更灵活地适应各种情况，提高查询性能，提升用户体验。

更多的信息，请参考用户文档。

- 支持通过优化器提示临时修改部分系统变量的值 #45892 @winoros

  TiDB v7.4.0 新增支持与 MySQL 8.0 相似的优化器提示 SET_VAR()。通过在 SQL 语句中添加 Hint SET_VAR()，可以在语句运行过程中临时修改部分系统变量，以针对不同语句设置环境。例如，可以主动提升高消耗 SQL 的并行度，或者通过变量修改优化器行为。

  支持使用 Hint SET_VAR() 修改的系统变量请参考系统变量。强烈建议不要利用此 Hint 修改没有明确支持的变量，这可能会引发不可预知的行为。

  更多信息，请参考用户文档。

- TiFlash 支持资源管控特性 #7660 @guo-shaoge

  在 TiDB v7.1.0 中，资源管控成为正式功能，提供对 TiDB 和 TiKV 的资源管理能力。在 v7.4.0 中，TiFlash 支持资源管控特性，完善了 TiDB 整体的资源管控能力。TiFlash 的资源管控与已有的 TiDB 资源管控特性完全兼容，现有的资源组将同时管控 TiDB、TiKV 和 TiFlash 中的资源。

  通过配置 TiFlash 参数 enable_resource_control，你可以控制是否开启 TiFlash 资源管控特性。开启后，TiFlash 将根据 TiDB 的资源组配置进行资源调度管理，确保整体资源的合理分配和使用。

  更多信息，请参考用户文档。

- TiFlash 支持 Pipeline 执行模型 (GA) #6518 @SeaRise

  在 v7.2.0 中，TiFlash 引入了 Pipeline 执行模型作为实验特性。该模型对所有线程资源进行统一管理，并对所有任务的执行进行统一调度，充分利用

线程资源，同时避免资源超用。从 v7.4.0 开始，TiFlash 完善了线程资源使用量的统计，Pipeline 执行模型成为正式功能 (GA) 并默认开启。由于该功能与 TiFlash 资源管控特性相互依赖，TiDB v7.4.0 移除了之前版本中用于控制是否启用 Pipeline 执行模型的变量 tidb_enable_tiflash_pipeline_model。现在你可以通过 TiFlash 参数 tidb_enable_resource_control 同时开启或关闭 Pipeline 执行模型和 TiFlash 资源管控特性。

更多信息，请参考用户文档。

- 新增优化器模式选择 #46080 @time-and-fate

  TiDB 在 v7.4.0 引入了一个新的系统变量 tidb_opt_objective，用于控制优化器的估算方式。默认值 moderate 维持之前版本的优化器行为，即优化器会利用运行时统计到的数据修改来校正估算。如果设置为 determinate，则优化器不考虑运行时校正，只根据统计信息来生成执行计划。

  对于长期稳定的 OLTP 业务，或者用户对已有的执行计划非常有把握的情况，推荐在测试后切换到 determinate 模式，减少执行计划跳变的可能。

  更多信息，请参考用户文档。

- 资源管控支持自动管理后台任务（实验特性）#44517 @glorv

  后台任务是指那些优先级不高但是需要消耗大量资源的任务，如数据备份和自动统计信息收集等。这些任务通常定期或不定期触发，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。在 TiDB v7.4.0 中，资源管控引入了对后台任务的自动管理。该功能有助于降低低优先级任务对在线业务的性能影响，实现资源的合理分配，大幅提升集群的稳定性。

  目前 TiDB 支持如下几种后台任务的类型：

  - lightning：使用 TiDB Lightning 或 IMPORT INTO 执行导入任务。

  - br：使用 BR 执行数据备份和恢复。目前不支持 PITR。
  - 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。

  默认情况下，被标记为后台任务的任务类型为空，此时后台任务的管理功能处于关闭状态，其行为与 TiDB v7.4.0 之前版本保持一致。你需要手动修改 default 资源组的后台任务类型以开启后台任务管理。

  更多信息，请参考用户文档。

- 增强锁定统计信息的能力 #46351 @hi-rustin

  在 v7.4.0 中，TiDB 增强了锁定统计信息的能力。现在，锁定和解锁统计信息需要与收集统计信息 (ANALYZE TABLE) 相同的权限，以确保操作的安全性。此外，还新增了对特定分区的统计信息进行锁定和解锁操作的支持，提高了功能灵活性。当用户对数据库中的查询和执行计划有把握，并且不希望发生变化时，可以使用锁定统计信息来提升统计信息的稳定性。

  更多信息，请参考用户文档。

- 引入系统变量控制是否选择表的哈希连接 #46695 @coderplay

  表的哈希连接是 MySQL 8.0 引入的新特性，主要用于连接两个相对较大的表和结果集。但对于交易类负载，或者一部分在 MySQL 5.7 稳定运行的业务来说，选择表的哈希连接可能会对性能产生风险。MySQL 通过优化器开关 optimizer_switch 能够在全局或者会话级控制哈希连接的选择。

  从 v7.4.0 开始，TiDB 引入系统变量 tidb_opt_enable_hash_join 对表的哈希连接进行控制。默认开启 (ON)。如果你非常确定执行计划中不需要选择表之间的哈希连接，则可以修改变量为 OFF，降低执行计划回退的可能性，提升系统稳定性。

  更多信息，请参考用户文档。

2.1.4 SQL 功能

- TiDB 支持完整的分区类型管理功能 #42728 @mjonss

  在 v7.4.0 之前，TiDB 中的分区表不能调整分区类型。从 v7.4.0 开始，TiDB 支持将分区表修改为非分区表、将非分区表修改为分区表、修改分区类型功能。你可以根据需要灵活调整表的分区类型、数量。例如，通过 ALTER TABLE t PARTITION BY ... 语句修改分区类型。

  更多信息，请参考用户文档。

- TiDB 支持 WITH ROLLUP 修饰符和 GROUPING 函数 #44487 @AilinKid

  WITH ROLLUP 修饰符和 GROUPING 函数是数据分析中常用的功能，用于对数据进行多维度的汇总。从 v7.4.0 开始，TiDB 支持在 GROUP BY 子句中使用 WITH ROLLUP 修饰符和 GROUPING 函数。例如，你可以通过 SELECT ... FROM ... GROUP BY ... WITH ROLLUP 语法使用 WITH ROLLUP 修饰符。

  更多信息，请参考用户文档。

2.1.5 数据库管理

- 新增排序规则 utf8mb4_0900_ai_ci 和 utf8mb4_0900_bin #37566 @YangKeao @zimulala @bb7133

  TiDB v7.4.0 增强了从 MySQL 8.0 迁移数据的支持。新增两个排序规则 (Collation) utf8mb4_0900_ai_ci 和 utf8mb4_0900_bin。其中 utf8mb4_0900_ai_ci 为 MySQL 8.0 的默认排序规则。

  同时新增支持 MySQL 8.0 兼容的系统变量 default_collation_for_utf8mb4，允许用户为 utf8mb4 字符集指定默认的排序方式，以兼容从 MySQL 5.7 或之前版本迁移或数据复制的场景。

  更多信息，请参考用户文档。

2.1.6 可观测性

- 支持向日志中添加会话标识和会话别名 #46071 @lcwangchao

  在对 SQL 执行问题做故障定位的时候，经常需要把 TiDB 各组件日志中的内容进行关联，由此找到问题的根本原因。从 v7.4.0 开始，TiDB 将会话标识 (CONNECTION_ID) 写入与会话相关的日志内容中，包括 TiDB 日志、慢查询日志、以及 TiKV 上 coprocessor 的慢日志记录。你可以根据会话标识，将几个日志中的内容关联起来，提升故障定位和诊断的效率。

  除此之外，通过设置会话级变量 tidb_session_alias，你可以向上述日志中添加自定义的标识。借助这个能力，把业务识别信息注入日志，可以将日志中的内容与业务关联，打通了业务到日志的链路，降低了诊断工作的难度。

- TiDB Dashboard 提供表格视图的执行计划 #1589 @baurine

  在 v7.4.0 中，TiDB Dashboard 的 **Slow Query** 页面和 **SQL Statement** 页面提供表格视图的执行计划，以提升用户的诊断体验。

  更多信息，请参考用户文档。

2.1.7 数据迁移

- 增强 IMPORT INTO 功能 #46704 @D3Hunter

  从 v7.4.0 起，你可以通过在 IMPORT INTO 的 CLOUD_STORAGE_URI 选项中指定编码后数据的云存储地址，开启全局排序功能（实验特性），提升性能和稳定性。

  此外，在 v7.4.0 中，IMPORT INTO 还引入了以下功能：

  - 支持配置 Split_File 选项，可将单个大 CSV 文件切分成多个 256 MiB 的小 CSV 文件进行并行处理，提升导入性能。
  - 支持导入压缩后的 CSV 和 SQL 文件，支持的压缩格式包括 .gzip、.gz、.zstd、.zst 和 .snappy。

更多信息，请参考用户文档。

- Dumpling 在将数据导出为 CSV 文件时支持用户自定义换行符 #46982 @GMHDBJD

  在 v7.4.0 之前，Dumpling 导出数据为 CSV 文件时，换行符为 "\r\n"，导致一些只能解析 "\n" 换行符的下游系统无法解析该 CSV 文件，或者要通过第三方工具转换后才能解析。

  从 v7.4.0 起，Dumpling 引入了新的参数 --csv-line-terminator。当你将数据导出为 CSV 文件时，可以通过该参数传入所需的换行符。该参数支持 "\r\n\" 和 "\n"，默认值为 "\r\n"，即和历史版本保持一致。

  更多信息，请参考用户文档。

- TiCDC 支持同步数据至 Pulsar #9413 @yumchina @asddongmen

  Pulsar 是一款云原生的分布式消息流平台，它能够显著提升你的实时数据流体验。从 v7.4.0 起，TiCDC 支持以 canal-json 格式同步变更数据至 Pulsar，实现与 Pulsar 的无缝集成。通过该功能，TiCDC 可以让你轻松捕获和同步 TiDB 变更数据到 Pulsar，为数据处理和分析功能提供新的可能性。你可以开发自己的消费应用程序，从 Pulsar 中读取并处理新生成的变更数据，以满足特定的业务需求。

  更多信息，请参考用户文档。

- TiCDC 支持 Claim-Check 功能，改进对大型消息的处理 #9153 @3AceShowHand

  在 v7.4.0 之前，TiCDC 无法向下游发送超过 Kafka 最大消息大小 (max.message.bytes) 的大型消息。从 v7.4.0 开始，在配置下游为 Kafka 的 Changefeed 的时候，你可以指定一个外部存储位置，用于存储超过 Kafka 限制的大型消息。TiCDC 会向 Kafka 发送一条引用消息，其中记录了该大

型消息在外部存储中的地址。当消费者收到该引用消息后，可以根据其中记录的外部存储地址信息，获取对应的消息内容。

更多信息，请参考用户文档。

2.2 兼容性变更

注意：

以下为从 v7.3.0 升级至当前版本 (v7.4.0) 所需兼容性变更信息。如果从 v7.2.0 或之前版本升级到当前版本，可能也需要考虑和查看中间版本 release notes 中提到的兼容性变更信息。

2.2.1 行为变更

- 自 v7.4.0 起，TiDB 已经兼容 MySQL 8.0 的主要功能，version() 将返回以 8.0.11 为前缀的版本信息。
- 升级到 TiFlash v7.4.0 后，不支持原地降级到之前的版本。这是因为，从 v7.4.0 开始，为了减少数据整理时产生的读、写放大，TiFlash 对 PageStorage V3 数据整理时的逻辑进行了优化，导致底层部分存储文件名发生了改动。详情请参考 TiFlash 升级帮助。
- 新增函数 TIDB_PARSE_TSO_LOGICAL()，用于从 TiDB TSO 时间戳中提取逻辑时间戳。
- 新增表 information_schema.CHECK_CONSTRAINTS，提高与 MySQL 8.0 的兼容性。

2.2.2 系统变量

变量名	修改类型	描述
tidb_enable_tiflash_pipeline_model	删除	这个变量用来控制是否启用 TiFlash Pipeline Model。从 v7.4.0 开启，开启 TiFlash 资源管控功能时，Pipeline Model 模型将自动启用。

变量名	修改类型	描述
tidb_enable_non_prepared_plan_cache	修改	经进一步的测试后，该变量默认值从 ON 修改为 OFF，即默认关闭非 Prepare 语句执行计划缓存。
default_collation_for_utf8mb4	新增	该变量用于设置 utf8mb4 字符集的默认排序规则，默认值为 utf8mb4_bin。
tidb_cloud_storage_uri	新增	该变量用于指定全局排序中使用的云存储的 URI。
tidb_opt_enable_hash_join	新增	控制优化器是否会选择表的哈希连接。默认打开 (ON)。设置为 OFF 时，除非没有计划可用，否则优化器会避免选择表的哈希连接。
tidb_opt_objective	新增	该变量用于设置优化器优化目标。moderate 维持旧版本的默认行为，优化器会利用更多信息尝试生成更优的计划；determinate 则倾向于保守，保持执行计划稳定。
tidb_schema_version_cache_limit	新增	该变量用于限制 TiDB 实例可以缓存多少个历史版本的表结构信息。默认值为 16，即默认缓存 16 个历史版本的表结构信息。
tidb_service_scope	新增	该变量是一个实例级别的变量，用于控制 TiDB 后端任务分布式框架下各 TiDB 节点的服务范围。当设置 TiDB 节点的 tidb_service_scope 为 background 时，后端任务分布式框架将调度该节点执行后端任务（如 ADD INDEX 和 IMPORT INTO）。
tidb_session_alias	新增	用来自定义当前会话相关日志中 session_alias 列的值。
tiflash_mem_quota_query_per_node	新增	用于设置单个查询在单个 TiFlash 节点上的内存使用上限，超过该限制时 TiFlash 会报错并终止该查询。默认值为 0，表示无限制。
tiflash_query_spill_ratio	新增	用于控制 TiFlash 查询级别的落盘机制的阈值。默认值为 0.7。
tikv_client_read_timeout	新增	该变量用于设置查询语句中 TiDB 发送 TiKV RPC 读请求的超时时间。默认值 0，表示使用默认的超时时间（通常是 40 秒）。

2.2.3 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	enable-stats-cache-mem-quota	修改	默认值由 false 改为 true，即默认开启 TiDB 统计信息缓存的内存上限。
TiKV	rocksdb.[defaultcf|writecf|lockcf].periodic-compaction-seconds	修改	默认值从 "30d" 修改为 "0s"，默认关闭 RocksDB 的周期性 compaction 以避免升级后集中触发大量 compaction 影响前台读写性能。
TiKV	rocksdb.[defaultcf|writecf|lockcf].ttl	修改	默认值从 "30d" 修改为 "0s"，SST 文件默认不会由于 TTL 触发 compaction，避免影响前台读写性能。
TiFlash	flash.compact_log_min_gap	新增	在当前 Raft 状态机推进的 applied_index 和上次落盘时的 applied_index 的差值高于 compact_log_min_gap 时，TiFlash 将执行来自 TiKV 的 CompactLog 命令，并进行数据落盘。
TiFlash	profiles.default.enable_resource_control	新增	控制是否开启 TiFlash 资源管控功能。
TiFlash	storage.format_version	修改	默认值从 4 修改为 5，该格式可以合并小文件从而减少了物理文件数量。
Dumpling	--csv-line-terminator	新增	控制导出数据为 CSV 文件的换行符，支持 "\r\n" 和 "\n"，默认值为 "\r\n"，即和历史版本保持一致。
TiCDC	claim-check-storage-uri	新增	当指定 large-message-handle-option 为 claim-check 时，claim-check-storage-uri 必须设置为一个有效的外部存储服务地址，否则创建 Changefeed 将会报错。

配置文件	配置项	修改类型	描述
TiCDC	large-message-handle-compression	新增	控制是否开启编码时的压缩功能，默认为空，即不开启。
TiCDC	large-message-handle-option	修改	该配置项新增一个可选值 claim-check。当设置为 claim-check 时，TiCDC Kafka sink 支持在消息大小超过限制时将该条消息发送到外部存储服务，同时向 Kafka 发送一条含有该大消息在外部存储服务中的地址的消息。

2.3 废弃功能

- Mydumper 计划在 v7.5.0 中废弃，其绝大部分功能已经被 Dumpling 取代，强烈建议切换到 Dumpling。
- TiKV-importer 组件计划在 v7.5.0 中废弃，建议使用 TiDB Lightning 物理导入模式作为替代方案。

2.4 改进提升

- TiDB
  - 优化 ANALYZE 分区表的内存使用和性能 #47071 #47104 #46804 @hawkingrei
  - 优化统计信息垃圾回收的内存使用和性能 #31778 @winoros
  - 优化索引合并进行交集操作时的 limit 下推，提高查询性能 #46863 @AilinKid
  - 改进代价模型 (Cost Model) 以尽量避免在 IndexLookup 回表任务多时错误地选择全表扫描 #45132 @qw4990
  - 优化 join 消除规则，提高 join on unique keys 时的查询性能 #46248 @fixdb
  - 将多值索引列的排序规则变更为 binary，避免执行失败 #46717 @YangKeao
- TiKV

  - 改进 Resolver 的内存使用，防止 OOM #15458 @overvenus
  - 消除 Router 对象中的 LRUCache，降低内存占用，防止 OOM #15430 @Connor1996
  - 降低 TiCDC Resolver 的内存占用 #15412 @overvenus
  - 降低 RocksDB compaction 带来的内存抖动 #15324 @overvenus
  - 降低 Partitioned Raft KV 中流控模块的内存占用 #15269 @overvenus
  - 新增 PD Client 连接重试过程中的 backoff 机制。异常错误重试期间，逐步增加重试时间间隔，减小 PD 压力 #15428 @nolouch
  - 支持动态调整 RocksDB 的 background_compaction #15424 @glorv
- PD
  - 优化 TSO 的追踪信息，方便调查 TSO 相关问题 #6856 @tiancaiamao
  - 支持复用 HTTP Client 连接，降低内存占用 #6913 @nolouch
  - 优化无法连接到备份集群时 PD 自动更新集群状态的速度 #6883 @disksing
  - 改进 resource control client 的配置获取方式，使其可以动态获取最新配置 #7043 @nolouch
- TiFlash
  - 改进 TiFlash 写入过程的落盘策略，提升随机写入负载下的写性能 #7564 @CalvinNeo
  - 为 TiFlash 处理 Raft 同步过程添加更多观测指标 #8068 @CalvinNeo
  - 改进 TiFlash 文件格式，减少小文件数量以避免造成文件系统 inode 耗尽的问题 #7595 @hongyunyan
- Tools
  - Backup & Restore (BR)

    - 缓解了 Region leadership 迁移导致 PITR 日志备份进度延迟变高的问题 #13638 @YuJuncen
    - 通过设置 HTTP 客户端 MaxIdleConns 和 MaxIdleConnsPerHost 参数，增强日志备份以及 PITR 恢复任务对连接复用的支持 #46011 @Leavrth
    - 增强 BR 在连接 PD 或外部 S3 存储出错时的容错能力 #42909 @Leavrth
    - 新增 restore 参数 WaitTiflashReady。当打开这个参数时，restore 操作将会等待 TiFlash 副本复制成功后才结束 #43828 #46302 @3pointer
    - 减少日志备份 resolve lock 的 CPU 开销 #40759 @3pointer
  - TiCDC
    - 优化同步 ADD INDEX DDL 的执行逻辑，从而不阻塞后续的 DML 语句 #9644 @sdojjy
  - TiDB Lightning
    - 优化 TiDB Lightning 在 Region scatter 阶段的重试逻辑 #46203 @mittalrishabh
    - 优化 TiDB Lightning 在导入数据阶段对 no leader 错误的重试逻辑 #46253 @lance6716

2.5 错误修复

- TiDB
  - 修复 BatchPointGet 算子在非 Hash 分区表下执行结果错误的问题 #45889 @Defined2014
  - 修复 BatchPointGet 算子在 Hash 分区表下执行结果错误的问题 #46779 @jiyfhust
  - 修复 TiDB parser 状态残留导致解析失败的问题 #45898 @qw4990
  - 修复 EXCHANGE PARTITION 没有检查约束的问题 #45922 @mjonss

- 修复 tidb_enforce_mpp 系统变量不能被正确还原的问题 #46214 @djshow832
- 修复 LIKE 语句中 _ 没有被正确处理的问题 #46287 #46618 @Defined2014
- 修复当获取 schema 失败时会导致 schemaTs 被设置为 0 的问题 #46325 @hihihuhu
- 修复 AUTO_ID_CACHE=1 时可能导致 Duplicate entry 的问题 #46444 @tiancaiamao
- 修复 AUTO_ID_CACHE=1 时 TiDB panic 后恢复过慢的问题 #46454 @tiancaiamao
- 修复 AUTO_ID_CACHE=1 时 SHOW CREATE TABLE 中 next_row_id 错误的问题 #46545 @tiancaiamao
- 修复在子查询中使用 CTE 时，解析出现 panic 的问题 #45838 @djshow832
- 修复在交换分区失败或被取消时，分区表的限制残留在原表上的问题 #45920 #45791 @mjonss
- 修复 List 分区的定义中不允许同时使用 NULL 和空字符串的问题 #45694 @mjonss
- 修复交换分区时，无法检测出不符合分区定义的数据的问题 #46492 @mjonss
- 修复 tmp-storage-quota 配置无法生效的问题 #45161 #26806 @wshwsh12
- 修复函数 WEIGHT_STRING() 不匹配排序规则的问题 #45725 @dveeden
- 修复 Index Join 出错可能导致查询卡住的问题 #45716 @wshwsh12
- 修复 DATETIME 或 TIMESTAMP 列与数字值比较时，行为与 MySQL 不一致的问题 #38361 @yibin87
- 修复无符号类型与 Duration 类型常量比较时产生的结果错误 #45410 @wshwsh12

  - 修复 access path 的启发式规则会忽略 READ_FROM_STORAGE(TIFLASH[...]) Hint 导致 Can't find a proper physical plan 的问题 #40146 @AilinKid
  - 修复 group_concat 无法解析 ORDER BY 列的问题 #41986 @AilinKid
  - 修复深嵌套的表达式的 HashCode 重复计算导致的高内存占用和 OOM 问题 #42788 @AilinKid
  - 修复 cast(col)=range 条件在 CAST 无精度损失的情况下会导致 FullScan 的问题 #45199 @AilinKid
  - 修复 MPP 执行计划中通过 Union 下推 Aggregation 导致的结果错误 #45850 @AilinKid
  - 修复带 in (?) 条件的 binding 无法匹配 in (?,...?) 的问题 #44298 @qw4990
  - 修复 non-prep plan cache 复用执行计划时未考虑 connection collation 导致的错误 #47008 @qw4990
  - 修复执行计划没有命中 plan cache 但不报 warning 的问题 #46159 @qw4990
  - 修复 plan replayer dump explain 会报错的问题 #46197 @time-and-fate
  - 修复执行带 CTE 的 DML 会导致 panic 的问题 #46083 @winoros
  - 修复当 JOIN 两个子查询时执行 TIDB_INLJ Hint 不生效的问题 #46160 @qw4990
  - 修复 MERGE_JOIN 的结果错误的问题 #46580 @qw4990
- TiKV
  - 修复开启 Titan 后，TiKV 遇到 Blob file deleted twice 报错无法正常启动的问题 #15454 @Connor1996
  - 修复 Thread Voluntary 和 Thread Nonvoluntary 监控面板没有数据的问题 #15413 @SpadeA-Tang
  - 修复 raftstore-applys 不断增长的数据错误 #15371 @Connor1996

  - 修复由于 Region 的元数据不正确造成 TiKV panic 的问题 #13311 @zyguan
  - 修复切换 sync_recovery 到 sync 后 QPS 降至 0 的问题 #15366 @nolouch
  - 修复 Online Unsafe Recovery 超时未中止的问题 #15346 @Connor1996
  - 修复 CpuRecord 可能导致的内存泄漏 #15304 @overvenus
  - 修复当备用集群关闭后，查询主集群出现 "Error 9002: TiKV server timeout" 的问题 #12914 @Connor1996
  - 修复当主集群恢复后 TiKV 再次启动时，备用 TiKV 会卡住的问题 #12320 @disksing
- PD
  - 修复在 Flashback 时不更新保存 Region 信息的问题 #6912 @overvenus
  - 修复因为同步 store config 慢而导致 PD Leader 切换慢的问题 #6918 @bufferflies
  - 修复 Scatter Peer 时未考虑 Group 的问题 #6962 @bufferflies
  - 修复 RU 消耗小于 0 导致 PD 崩溃的问题 #6973 @CabinfeverB
  - 修复修改隔离等级时未同步到默认放置规则中的问题 #7121 @rleungx
  - 修复在集群规模大时 client-go 周期性更新 min-resolved-ts 可能造成 PD OOM 的问题 #46664 @HuSharp
- TiFlash
  - 修复 Grafana 面板的 max_snapshot_lifetime 监控指标显示有误的问题 #7713 @JaySon-Huang
  - 修复 TiFlash 部分监控页面最长耗时指标显示错误的问题 #8076 @CalvinNeo
  - 修复 TiDB 误报 MPP 任务失败的问题 #7177 @yibin87
- Tools

- Backup & Restore (BR)
    - 修复备份失败时 BR 误报 resolve lock timeout 掩盖了实际错误的问题 #43236 @YuJuncen
    - 修复 PITR 恢复隐式主键可能冲突的问题 #46520 @3pointer
    - 修复 PITR 恢复数据元信息 (meta-kv) 出错的问题 #46578 @Leavrth
    - 修复 BR 集成测试用例出错的问题 #45561 @purelind
- TiCDC
    - 修复 PD 做扩缩容场景下 TiCDC 访问无效旧地址的问题 #9584 @fubinzh @asddongmen
    - 修复某些特殊场景下 changefeed 失败的问题 #9309 #9450 #9542 #9685 @hicqu @CharlesCheung96
    - 修复在上游同一个事务中修改多行唯一键场景下，TiCDC 可能导致同步写冲突的问题 #9430 @sdojjy
    - 修复在上游同一条 DDL 中重命名多个表的场景下同步出错的问题 #9476 #9488 @CharlesCheung96 @asddongmen
    - 修复 CSV 格式下没有校验中文分隔符的问题 #9609 @CharlesCheung96
    - 修复所有 changefeed 被移除时会阻塞上游 TiDB GC 的问题 #9633 @sdojjy
    - 修复开启 scale-out 时流量在节点间分配不均匀问题 #9665 @sdojjy
    - 修复日志中记录了用户敏感信息的问题 #9690 @sdojjy
- TiDB Data Migration (DM)
    - 修复 DM 在大小写不敏感的 collation 下无法正确处理冲突的问题 #9489 @hihihuhu
    - 修复 DM validator 死锁问题并增强重试 #9257 @D3Hunter

    - 修复 DM 在跳过失败 DDL 并且后续无 DDL 执行时显示延迟持续增长的问题 #9605 @D3Hunter
    - 修复 DM 在跳过 Online DDL 时无法正确追踪上游表结构的问题 #9587 @GMHDBJD
    - 修复 DM 在乐观模式恢复任务时跳过所有 DML 的问题 #9588 @GMHDBJD
    - 修复 DM 在乐观模式中跳过 Partition DDL 的问题 #9788 @GMHDBJD
  - TiDB Lightning
    - 修复 TiDB Lightning 导入 NONCLUSTERED auto_increment 和 AUTO_ID_CACHE=1 表后，插入数据报错的问题 #46100 @tiancaiamao
    - 修复 checksum = "optional" 时 Checksum 阶段仍然报错的问题 #45382 @lyzx2001
    - 修复当 PD 集群地址变更时数据导入失败的问题 #43436 @lichunzhu

2.6 贡献者

感谢来自 TiDB 社区的贡献者们：

- aidendou
- coderplay
- fatelei
- highpon
- hihihuhu （首次贡献者）
- isabella0428
- jiyfhust
- JK1Zhang
- joker53-1（首次贡献者）
- L-maple
- mittalrishabh

- paveyry
- shawn0915
- tedyu
- yumchina
- ZhuohaoHe

3 TiDB 7.3.0 Release Notes

发版日期：2023 年 8 月 14 日

TiDB 版本：7.3.0

试用链接：快速体验 | 下载离线包

v7.3.0 引入了以下主要功能。功能详情中列出的部分功能旨在增强 TiDB 和 TiFlash 的查询稳定性，不直接面向用户，因此未包含在下表中。

分类	功能	描述
可扩展性与性能	TiDB Lightning 支持 Partitioned Raft KV（实验特性）	TiDB Lightning 的数据导入服务支持新的 Partitioned Raft KV 架构，为 Partitioned Raft KV 在 TiDB 后续版本中 GA 做好准备。
稳定性与高可用	TiDB Lightning 引入冲突数据的自动检测和处理机制	TiDB Lightning 物理导入模式支持新版本的冲突检测机制，支持在遇到数据冲突时替换 (replace) 或忽略 (ignore) 冲突数据的语义。TiDB Lightning 会自动处理冲突数据，同时提高了冲突处理的性能。
	手动标记资源使用超出预期的查询（实验特性）	查询耗费的时间有时会超出预期。通过资源组新增的 Runaway Queries 监控列表，你可以设置降低 Runaway Queries 的优先级或终止查询，从而更有效地管理查询。该功能允许算子在资源组级别通过匹配 SQL 文本、SQL digest 或执行计划标记查询，并

分类	功能	描述
		对这些查询进行处理，从而更好地控制非预期的大型查询可能对集群产生的影响。
SQL	添加更多优化器提示，加强对算子的控制，提升查询稳定性	新增优化器提示：NO_INDEX_JOIN()、NO_MERGE_JOIN()、NO_INDEX_MERGE_JOIN()、NO_HASH_JOIN()、NO_INDEX_HASH_JOIN()
数据库管理与可观测性	显示统计信息收集的进度	支持使用 SHOW ANALYZE STATUS 语句或通过 mysql.analyze_jobs 系统表查看 ANALYZE 任务的进度。

3.1 功能详情

3.1.1 性能

- TiFlash 支持副本选择策略 #44106 @XuHuaiyu

  在 v7.3.0 之前，当 TiFlash 进行数据扫描和 MPP 计算时，会尽可能使用其所有节点的副本，以提供最强大的性能。从 v7.3.0 起，TiFlash 引入副本选择策略，该策略由系统变量 tiflash_replica_read 控制，可以根据节点的区域属性选择特定的副本，调度部分节点进行数据扫描及 MPP 计算。

  当集群部署在多个机房且每个机房都拥有完整的 TiFlash 数据副本时，你可以设置该策略只选择使用当前机房的 TiFlash 副本，即只在当前机房的 TiFlash 节点中进行数据扫描和 MPP 计算，从而避免大量跨机房的网络数据传输。

  更多信息，请参考用户文档。

- TiFlash 支持节点内的 Runtime Filter #40220 @elsa0520

Runtime Filter 是在查询规划阶段生成的一种**动态取值谓词**。在表连接的过程中，这些动态谓词能够有效过滤掉不满足连接条件的行，减少扫描时间和网络开销，提升表连接的效率。自 v7.3.0 起，TiFlash 支持节点内的 Runtime Filter，提升了数据分析类查询的整体性能，在部分 TPC-DS 数据集的查询中可达到 10% ~ 50% 的性能提升。

该功能在 v7.3.0 默认关闭。要启用此功能，需将变量 tidb_runtime_filter_mode 设置为 LOCAL。

更多信息，请参考用户文档。

- TiFlash 支持执行公共表表达式 (CTE)（实验特性）#43333 @winoros

  在 v7.3.0 版本之前，TiFlash 的 MPP 引擎默认无法执行包含 CTE 的查询，你需要通过系统变量 tidb_opt_force_inline_cte 强制 inline CTE，达到让查询尽可能在 MPP 框架下执行的效果。在 v7.3.0 中，TiFlash MPP 引擎支持执行包含 CTE 的查询，无需强制 inline CTE 也可以尽可能地在 MPP 框架中执行查询。在 TPC-DS 基准测试中，与强制 inline 的执行方式相比，该功能可以将包含 CTE 的查询的总执行速度提升 20%。

  该功能为实验特性，默认关闭，由变量 tidb_opt_enable_mpp_shared_cte_execution 控制。

3.1.2 稳定性

- 新增部分优化器提示 #45520 @qw4990

  TiDB 在 v7.3.0 新增了几个优化器提示，用来控制表之间的连接方式，包括：

  - NO_MERGE_JOIN() 选择除 Merge Join 以外的连接方式。
  - NO_INDEX_JOIN() 选择除 Index Nested Loop Join 以外的连接方式。

  - NO_INDEX_MERGE_JOIN() 选择除 Index Nested Loop Merge Join 以外的连接方式。
  - NO_HASH_JOIN() 选择哈希连接以外的连接方式。
  - NO_INDEX_HASH_JOIN() 选择除 Index Nested Loop Hash Join 以外的连接方式。

  更多信息，请参考用户文档。

- 手动标记资源使用超出预期的查询（实验特性）#43691 @Connor1996 @CabinfeverB

  在 v7.2.0 中，TiDB 自动管理资源使用超出预期的查询 (Runaway Query)，即自动降级或取消运行时间超出预期的查询。在实际运行时，只依靠规则无法覆盖所有情况。因此，TiDB v7.3.0 新增手动标记查询的功能。利用新增的命令 QUERY WATCH，你可以根据 SQL 的文本、SQL Digest 或执行计划标记查询，命中的查询可以被降级或取消。

  手动标记 Runaway Query 的功能为数据库中突发的性能问题提供了有效的干预手段。针对由查询引发的性能问题，在定位根本原因之前，该功能可以快速缓解其对整体性能的影响，从而提升系统服务质量。

  更多信息，请参考用户文档。

3.1.3 SQL 功能

- List 和 List COLUMNS 分区表支持默认分区 #20679 @mjonss @bb7133

  在 v7.3.0 以前，当使用 INSERT 语句向 List 或 List COLUMNS 分区表插入数据时，这些数据需要满足分区表指定的分区条件。如果要插入的数据不匹配任何分区条件，该语句将执行失败或忽略不符合分区条件的数据。

  在 v7.3.0 中，List 和 List COLUMNS 分区表支持默认分区功能。在创建默认分区后，如果要插入的数据不匹配任何分区条件，则数据将被写入默认分区。默认分区功能可以提升 List 分区和 List COLUMNS 分区的使用便捷

性，避免不符合分区条件的数据导致 INSERT 语句执行失败或者数据被忽略。

需要注意的是，该功能是 TiDB 对 MySQL 语法的扩展。创建默认分区后，该分区表的数据无法直接同步到 MySQL 中。

更多信息，请参考用户文档。

3.1.4 可观测性

- 显示统计信息收集的进度 #44033 @hawkingrei

  收集大表的统计信息经常会持续较长时间。在之前的版本中，无法了解统计信息收集的进度，进而无法预测完成时间。TiDB v7.3.0 新增显示统计信息收集进度的功能。你可以通过系统表 mysql.analyze_jobs 或者 SHOW ANALYZE STATUS 查看各个子任务的总体工作量、当前进度以及预计的完成时间。在大规模数据导入、SQL 性能优化等场景下，该功能有助于了解整体任务进度，提升用户体验。

  更多信息，请参考用户文档。

- Plan Replayer 支持导出历史统计信息 #45038 @time-and-fate

  自 v7.3.0 起，通过新增的 dump with stats as of timestamp 子句，你可以使用 Plan Replayer 导出指定 SQL 相关对象在指定时间点的统计信息。在执行计划问题的诊断过程中，通过对历史统计信息的准确抓取，能够更精确地分析出执行计划在问题发生的时间点是如何生成的，从而找到问题的根本原因，大大提升执行计划问题的诊断效率。

  更多信息，请参考用户文档。

3.1.5 数据迁移

- TiDB Lightning 引入新版冲突数据检测和处理机制 #41629 @lance6716

在之前的版本中，TiDB Lightning 的逻辑导入模式和物理导入模式各自使用独立的冲突检测和处理方式，其配置较为复杂且不易理解。另外，在物理导入模式下，无法通过替换 (replace) 或忽略 (ignore) 策略处理冲突的数据。从 v7.3.0 开始，TiDB Lightning 引入新版冲突检测和处理机制，逻辑导入模式和物理导入模式都使用相同的冲突检测和处理方式，即可以选择在遇到冲突数据时报错 (error)、替换 (replace) 或忽略 (ignore)。同时还支持设置冲突记录的上限，例如在处理指定数量冲突记录后任务中断退出，也可以记录哪些数据发生了冲突，以便后续排查。

当导入数据存在大量冲突时，推荐使用新版冲突检测和处理机制，以获得更好的性能。在实验环境下，相比旧版，新版机制最高可将冲突检测和处理的性能提升 3 倍。该性能数据仅供参考，实际性能会受到环境配置、表结构、冲突数据的占比等因素影响。注意新版和旧版冲突处理机制不能同时使用。未来将废弃旧版冲突检测和处理机制。

更多信息，请参考用户文档。

- TiDB Lightning 支持 Partitioned Raft KV（实验特性）#14916 @GMHDBJD

  TiDB Lightning 支持 Partitioned Raft KV，该功能可以提升 TiDB Lightning 导入数据的性能。

- TiDB Lightning 引入新的参数 enable-diagnose-log 用于打印更多的诊断日志，方便定位问题 #45497 @D3Hunter

  默认情况下，该功能未启用，即只打印包含 lightning/main 的日志。开启该功能后，将打印所有包（包括 client-go 和 tidb）的日志，以帮助诊断与 client-go 和 tidb 相关的问题。

  更多信息，请参考用户文档。

3.2 兼容性变更

注意：

以下为从 v7.2.0 升级至当前版本 (v7.3.0) 所需兼容性变更信息。如果从 v7.1.0 或之前版本升级到当前版本，可能也需要考虑和查看中间版本 release notes 中提到的兼容性变更信息。

3.2.1 行为变更

- Backup & Restore (BR)
  - 全量恢复前增加了空集群检查，默认不支持恢复到非空集群。如果强制恢复，可以使用 --filter 指定对应表名。
- TiDB Lightning
  - 废弃 tikv-importer.on-duplicate，由 conflict.strategy 替代。
  - TiDB Lightning 停止迁移任务之前能容忍的最大非严重 (non-fatal errors) 错误数的配置项 max-error 不再包含导入数据冲突记录的上限，由 conflict.threshold 控制可容忍的最大冲突的记录数。
- TiCDC
  - 当 Kafka sink 使用 Avro 协议时，如果开启了 force-replicate 参数，创建 changefeed 会报错。
  - 由于 delete-only-output-handle-key-columns 和 force-replicate 参数不兼容，同时开启两个参数时，创建 changefeed 会报错。
  - 当使用 Open Protocol 作为输出协议时，UPDATE 类型的事件将仅输出变更的列。

3.2.2 系统变量

变量名	修改类型	描述
tidb_opt_enable_mpp_shared_cte_execution	修改	该变量从 v7.3.0 开始生效，用于控制非递归的公共表表达式 (CTE) 是否可以在 TiFlash MPP 执行。

变量名	修改类型	描述
tidb_lock_unchanged_keys	新增	用于控制部分场景下，对于事务中涉及但并未修改值的 key 是否进行上锁。
tidb_opt_enable_non_eval_scalar_subquery	新增	这个变量用于控制 EXPLAIN 语句是否禁止提前执行可以在优化阶段展开的常量子查询。
tidb_skip_missing_partition_stats	新增	这个变量用于控制当分区统计信息缺失时生成 GlobalStats 的行为。
tiflash_replica_read	新增	这个变量用于设置当查询需要使用 TiFlash 引擎时，TiFlash 副本的选择策略。

3.2.3 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	enable-32bits-connection-id	新增	这个变量用于控制是否开启生成 32 位 connection ID 的功能。
TiDB	in-mem-slow-query-recent-num	新增	这个变量用于控制缓存在内存中的最近使用的 slow query 个数。
TiDB	in-mem-slow-query-topn-num	新增	这个变量用于控制缓存在内存中的最慢的 slow query 个数。
TiKV	coprocessor.region-bucket-size	修改	为降低客户端超时的可能性，默认值从 96MiB 修改为 50MiB。
TiKV	raft-engine.format-version	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，会引入 Ribbon filter，因此将默认值从 2 修改为 5。
TiKV	raftdb.max-total-wal-size	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，TiKV 会跳过写 WAL，因此默认值从 "4GB" 修改为 1，即禁用 WAL。
TiKV	rocksdb.[defaultcf|writecf|lockcf].compaction-guard-min-output-file-size	修改	为解决大数据写入情况下 compaction 速度跟不上写入速度的

配置文件	配置项	修改类型	描述
			问题，默认值从 "1MB" 修改为 "8MB"。
TiKV	rocksdb.[defaultcf|writecf|lockcf].format-version	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，会引入 Ribbon filter，因此将默认值从 2 修改为 5。
TiKV	rocksdb.lockcf.write-buffer-size	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，为加快 lockcf 上 compaction 的速度，默认值从 "32MB" 修改为 "4MB"。
TiKV	rocksdb.max-total-wal-size	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，TiKV 会跳过写 WAL，因此默认值从 "4GB" 修改为 1，即禁用 WAL。
TiKV	rocksdb.stats-dump-period	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，为关闭冗余日志的打印，默认值从 "10m" 修改为 "0"。
TiKV	rocksdb.write-buffer-limit	修改	为减小 memtable 的内存开销，当 storage.engine="raft-kv" 时，默认值从本机内存的 25% 修改为 0，即不限制。当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，默认值从本机内存的 25% 修改为本机内存的 20%。
TiKV	storage.block-cache.capacity	修改	当使用 Partitioned Raft KV (storage.engine="partitioned-raft-kv") 时，为弥补 memtable 的内存开销，将默认值从系统总内存

配置文件	配置项	修改类型	描述
			大小的 45% 修改为系统总内存大小的 30%。
TiFlash	storage.format_version	修改	引入新的 DTFile 储存文件格式 format_version = 5，该格式可以合并小文件从而减少物理文件数量。注意该格式目前为实验特性，默认未启用。
TiDB Lightning	tikv-importer.incremental-import	删除	TiDB Lightning 并行导入参数。因为该参数名容易被误认为是增量导入的参数，因此更名为 tikv-importer.parallel-import。如果用户传入旧的参数名，会被自动转成新的参数名。
TiDB Lightning	tikv-importer.on-duplicate	废弃	TiDB Lightning 逻辑导入模式插入冲突数据时执行的操作。从 v7.3.0 起，该参数由 conflict.strategy 取代。
TiDB Lightning	conflict.max-record-rows	新增	TiDB Lightning 新版冲突检测与处理策略，用于记录在数据导入过程中遇到的冲突记录，并允许设置最大上限，默认值为 100。
TiDB Lightning	conflict.strategy	新增	TiDB Lightning 新版冲突检测与处理的策略，包含 ""（不做冲突检测），"error"（遇到冲突数据即报错并停止导入），"replace"（遇到冲突记录替换已有的冲突记录），"ignore"（遇到冲突记录忽略需要插入的该条冲突记录）四种策略。默认值为 ""，即不做冲突检测。
TiDB Lightning	conflict.threshold	新增	TiDB Lightning 新版冲突检测与处理策略允许的冲突上限，

配置文件	配置项	修改类型	描述
			conflict.strategy="error" 时默认值为 0，当 conflict.strategy="replace" 或 conflict.strategy="ignore" 时默认值为 maxint。
TiDB Lightning	enable-diagnose-logs	新增	是否开启诊断日志。默认为 false，即只输出和导入有关的日志，不会输出依赖的其他组件的日志。设置为 true 后，既输出和导入相关的日志，也输出依赖的其他组件的日志，并开启 GRPC debug，可用于问题诊断。
TiDB Lightning	tikv-importer.parallel-import	新增	TiDB Lightning 并行导入参数。用于替代原有的 tikv-importer.incremental-import 参数，因为原有参数会被误认为是增量导入的参数而误用。
BR	azblob.encryption-key	新增	BR 为外部存储 Azure Blob Storage 提供加密密钥支持
BR	azblob.encryption-scope	新增	BR 为外部存储 Azure Blob Storage 提供加密范围支持
TiCDC	large-message-handle-option	新增	默认为空，即消息大小超过 Kafka Topic 的限制后，同步任务失败。设置为 “handle-key-only” 时，如果消息超过大小，只发送 handle key 以减少消息的大小；如果依旧超过大小，则同步任务失败。
TiCDC	sink.csv.binary-encoding-method	新增	CSV 协议中二进制类型数据的编码方式，可选 'base64' 与 'hex'。默认值为 'base64'。

3.2.4 系统表

- 新增系统表 mysql.tidb_timers 用来存储系统内部定时器的元信息。

3.3 废弃功能

- TiDB
  - 统计信息的快速分析(实验特性)计划在 v7.5.0 中废弃。
  - 统计信息的增量收集(实验特性)计划在 v7.5.0 中废弃。

3.4 改进提升

- TiDB
  - 新增 tidb_opt_enable_non_eval_scalar_subquery 系统变量用于控制 EXPLAIN 语句是否在优化阶段提前执行子查询 #22076 @winoros
  - 在启用 Global Kill 的情况下，可以通过 Control+C 终止当前会话 #8854 @pingyu
  - 支持锁函数 IS_FREE_LOCK() 和 IS_USED_LOCK() #44493 @dveeden
  - 优化与落盘相关的 chunk 读取的性能 #45125 @YangKeao
  - 以 Optimizer Fix Controls 的方式改进了 Index Join 内表的高估问题 #44855 @time-and-fate
- TiKV
  - 添加 Max gap of safe-ts 和 Min safe ts region 监控项以及 tikv-ctl get-region-read-progress 命令，用于更好地观测和诊断 resolved-ts 和 safe-ts 的状态 #15082 @ekexium
- PD
  - 未开启 Swagger server 时，PD 默认屏蔽 Swagger API #6786 @bufferflies
  - 提升 etcd 的高可用性 #6554 #6442 @lhy1024
  - 减少 GetRegions 请求的内存占用 #6835 @lhy1024
- TiFlash

  - 支持新的 DTFile 格式版本 storage.format_version = 5，可以合并小文件从而减少物理文件数量（实验特性） #7595 @hongyunyan

- Tools
  - Backup & Restore (BR)
    - 使用 BR 备份数据到 Azure Blob Storage 时，支持使用加密范围或加密密钥对数据进行服务端加密 #45025 @Leavrth
  - TiCDC
    - 优化了 Open Protocol 输出的消息大小，在发送 UPDATE 类型事件时仅输出被更新的列值 #9336 @3AceShowHand
    - Storage Sink 支持对 HEX 格式的数据进行十六进制编码输出，使其兼容 AWS DMS 的格式规范 #9373 @CharlesCheung96
    - Kafka Sink 支持在消息过大时只发送 Handle Key 数据，减少数据大小 #9382 @3AceShowHand

3.5 错误修复

- TiDB
  - 修复当使用 MySQL 的 Cursor Fetch 协议时，结果集占用的内存超过 tidb_mem_quota_query 的限制导致 TiDB OOM 的问题。修复后，TiDB 会自动将结果集写入磁盘以释放内存资源 #43233 @YangKeao
  - 修复数据争用导致 TiDB panic 的问题 #45561 @genliqi
  - 修复带 indexMerge 的查询被 kill 时可能会卡住的问题 #45279 @xzhangxian1008
  - 修复当开启 tidb_enable_parallel_apply 时，MPP 模式下的查询结果出错的问题 #45299 @windtalker
  - 修复 resolve lock 在 PD 时间跳变的情况下可能卡住的问题 #44822 @zyguan

- 修复 GC resolve lock 可能错过一些悲观锁的问题 #45134 @MyonKeminta
- 修复动态裁剪模式下使用了排序的查询返回结果错误的问题 #45007 @Defined2014
- 修复 AUTO_INCREMENT 与列的默认值 DEFAULT 可以指定在同一列上的问题 #45136 @Defined2014
- 修复某些情况下查询系统表 INFORMATION_SCHEMA.TIKV_REGION_STATUS 返回结果错误的问题 #45531 @Defined2014
- 修复某些情况下分区表分区裁剪不正确的问题 #42273 @jiyfhust
- 修复 TRUNCATE 分区表的某个分区时，全局索引无法清除的问题 #42435 @L-maple
- 修复在 TiDB 节点故障后其它 TiDB 节点没有接管 TTL 任务的问题 #45022 @lcwangchao
- 修复 TTL 运行过程中内存泄漏的问题 #45510 @lcwangchao
- 修复向分区表插入数据时某些报错信息不准确的问题 #44966 @lilinghai
- 修复 INFORMATION_SCHEMA.TIFLASH_REPLICA 表的读取权限有误的问题 #7795 @Lloyd-Pottiger
- 修复使用错误分区表名时报错的问题 #44967 @River2000i
- 修复某些情况下启用 tidb_enable_dist_task 时，创建索引卡住的问题 #44440 @tangenta
- 修复通过 BR 恢复 AUTO_ID_CACHE=1 的表时，会遇到 duplicate entry 报错的问题 #44716 @tiancaiamao
- 修复执行 TRUNCATE TABLE 消耗的时间和 ADMIN SHOW DDL JOBS 显示的任务执行时间不一致的问题 #44785 @tangenta
- 修复读取元数据时间超过一个 DDL lease 导致升级 TiDB 卡住的问题 #45176 @zimulala

  - 修复 SELECT CAST(n AS CHAR) 语句中的 n 为负数时，查询结果出错的问题 #44786 @xhebox
  - 修复开启 tidb_opt_agg_push_down 时查询可能返回错误结果的问题 #44795 @AilinKid
  - 修复带有 current_date() 的查询使用 Plan Cache 导致结果错误的问题 #45086 @qw4990
- TiKV
  - 修复在一些罕见的情况下，在 GC 的同时读取数据可能导致 TiKV panic 的问题 #15109 @MyonKeminta
- PD
  - 修复重启 PD 可能导致 default 资源组被重新初始化的问题 #6787 @glorv
  - 修复当 etcd 已经启动，但 client 尚未连接上 etcd 时，调用 client 会导致 PD panic 的问题 #6860 @HuSharp
  - 修复 Region 的 health-check 输出可能与通过 ID 所查到的 Region 信息不一致的问题 #6560 @JmPotato
  - 修复 unsafe recovery 中失败的 learner peer 在 auto-detect 模式中被忽略的问题 #6690 @v01dstar
  - 修复 Placement Rules 选择了不满足规则的 TiFlash learner 的问题 #6662 @rleungx
  - 修复在 rule checker 选定 peer 时，unhealthy peer 无法被移除的问题 #6559 @nolouch
- TiFlash
  - 修复由于死锁导致 TiFlash 无法成功同步分区表的问题 #7758 @hongyunyan
  - 修复系统表 INFORMATION_SCHEMA.TIFLASH_REPLICA 包含用户没有访问权限的表的问题 #7795 @Lloyd-Pottiger

  - 修复当同一个 MPP Task 内有多个 HashAgg 算子时，可能导致 MPP Task 编译时间过长而严重影响查询性能的问题 #7810 @SeaRise

- Tools
  - TiCDC
    - 修复由于 PD 短暂不可用而导致同步任务报错的问题 #9294 @asddongmen
    - 修复 TiCDC 部分节点发生网络隔离时可能引发的数据不一致问题 #9344 @CharlesCheung96
    - 修复当 Kafka Sink 遇到错误时可能会无限阻塞同步任务推进的问题 #9309 @hicqu
    - 修复在 TiCDC 节点状态发生改变时可能引发的 panic 问题 #9354 @sdojjy
    - 修复对默认 ENUM 值编码错误的问题 #9259 @3AceShowHand
  - TiDB Lightning
    - 修复 TiDB Lightning 导入完成后执行 checksum 可能遇到 SSL 错误的问题 #45462 @D3Hunter
    - 修复逻辑导入模式下，导入期间下游删除表可能导致 TiDB Lightning 元信息未及时更新的问题 #44614 @dsdashun

3.6 贡献者

感谢来自 TiDB 社区的贡献者们：

- charleszheng44
- dhysum
- haiyux
- Jiang-Hua
- Jille
- jiyfhust
- krishnaduttPanchagnula

- L-maple
- pingandb
- testwill
- tisonkun
- xuyifangreeneyes
- yumchina

4 TiDB 7.2.0 Release Notes

发版日期：2023 年 6 月 29 日

TiDB 版本：7.2.0

试用链接：快速体验 | 下载离线包

在 7.2.0 版本中，你可以获得以下关键特性：

分类	功能	描述
可扩展性与性能	资源组支持管理资源消耗超出预期的查询（实验特性）	通过此功能，你可以更细粒度地管理执行时间超时的查询，根据查询的不同类型实现不同的行为。符合指定阈值的查询将按照你的设置被降低优先级或者终止执行。
	TiFlash 支持 Pipeline 执行模型（实验特性）	TiFlash 支持 Pipeline 执行模型，优化对线程资源的控制。
SQL	支持新的 SQL 语句 IMPORT INTO，可以通过 TiDB 进行数据导入（实验特性）	TiDB 引入了一个新的 SQL 语句 IMPORT INTO。该语句集成了 TiDB Lightning 的物理导入模式的能力，使你无需单独部署和管理 TiDB Lightning 即可导入数据文件到 TiDB 中。例如，通过该语句，你可以直接从 Amazon S3 或 Google Cloud Storage (GCS) 远程导入数据到 TiDB 中。

分类	功能	描述
数据库管理与可观测性	DDL 任务支持暂停和恢复操作（实验特性）	该功能允许临时暂停资源密集型的 DDL 操作，例如索引创建，以节省资源并最小化对在线流量的影响。当资源许可时，你可以无缝恢复 DDL 任务，而无需取消和重新开始。该功能提高了资源利用率，改善了用户体验，并简化了 schema 更改过程。

4.1 功能详情

4.1.1 性能

- 新增支持下推以下两个窗口函数到 TiFlash #7427 @xzhangxian1008
  - FIRST_VALUE
  - LAST_VALUE
- TiFlash 支持 Pipeline 执行模型（实验特性）#6518 @SeaRise

  在 v7.2.0 版本之前，TiFlash 引擎中各个任务在执行时，需要自行申请线程资源。TiFlash 引擎通过控制任务数的方式限制线程资源使用，以避免线程资源超用，但并不能完全避免此问题。因此，在 v7.2.0 中，TiFlash 引入 Pipeline 执行模型，对所有线程资源进行统一管理，并对所有任务的执行进行统一调度，充分利用线程资源，同时避免资源超用。新增系统变量 tidb_enable_tiflash_pipeline_model 用于设置是否启用 Pipeline 执行模型。

  更多信息，请参考用户文档。

- 降低 TiFlash 等待 schema 同步的时延 #7630 @hongyunyan

  当表的 schema 发生变化时，TiFlash 需要及时从 TiKV 同步新的表结构信息。在 v7.2.0 之前，当 TiFlash 访问表数据时，只要检测到数据库中某张表的 schema 发生了变化，TiFlash 就会重新同步该数据库中所有表的 schema 信息。即使一张表没有 TiFlash 副本，TiFlash 也会同步该表的

schema 信息。当数据库中有大量表时，通过 TiFlash 只读取一张表的数据也可能因为需要等待所有表的 schema 信息同步完成而造成较高的时延。

在 v7.2.0 中，TiFlash 优化了 schema 的同步机制，只同步拥有 TiFlash 副本的表的 schema 信息。当检测到某张有 TiFlash 副本的表的 schema 有变化时，TiFlash 只同步该表的 schema 信息，从而降低了 TiFlash 同步 schema 的时延，同时减少了 DDL 操作对于 TiFlash 同步数据的影响。该优化自动生效，无须任何设置。

- 提升统计信息收集的性能 #44725 @xuyifangreeneyes

  TiDB v7.2.0 优化了统计信息的收集策略，会选择跳过一部分重复的信息，以及对优化器价值不高的信息，从而将统计信息收集的整体速度提升了 30%。这一改进有利于 TiDB 更及时地更新数据库对象的统计信息，生成更准确的执行计划，从而提升数据库整体性能。

  默认设置下，统计信息收集会跳过类型为 JSON、BLOB、MEDIUMBLOB、LONGBLOB 的列。你可以通过设置系统变量 tidb_analyze_skip_column_types 来修改默认行为。当前支持设置跳过 JSON、BLOB、TEXT 这几个类型及其子类型。

  更多信息，请参考用户文档。

- 提升数据和索引一致性检查的性能 #43693 @wjhuang2016

  ADMIN CHECK [TABLE|INDEX] 语句用于校验表中数据和对应索引的一致性。在 v7.2.0 中，TiDB 优化了数据一致性的校验方式，大幅提升了 ADMIN CHECK [TABLE|INDEX] 语句的执行效率，在大数据量场景中性能能够提升百倍。

  该优化默认开启（tidb_enable_fast_table_check 默认为 ON），可以大幅减少大型表数据一致性检查的时间，提升运维体验。

更多信息，请参考用户文档。

4.1.2 稳定性

- 自动管理资源超出预期的查询（实验特性）#43691 @Connor1996 @CabinfeverB @glorv @HuSharp @nolouch

  突发的 SQL 性能问题引发数据库整体性能下降，是数据库稳定性最常见的挑战。造成 SQL 性能问题的原因有很多，例如未经充分测试的新 SQL、数据量剧烈变化、执行计划突变等，这些问题很难从源头上完全规避。TiDB v7.2.0 增加了对资源超出预期的查询的管理能力，在上述问题发生时，能够快速降低影响范围。

  你可以针对某个资源组 (Resource Group) 设置查询的最长执行时间。当查询的执行时间超过设置值时，自动降低查询的优先级或者取消查询。你还可以设置在一段时间内通过文本或者执行计划立即匹配已经识别出的查询，从而避免问题查询的并发度太高时，在识别阶段就造成大量资源消耗的情况。

  对资源超出预期查询的自动管理能力，为你提供了有效的手段，快速应对突发的查询性能问题，降低对数据库整体性能的影响，从而提升数据库的稳定性。

  更多信息，请参考用户文档。

- 增强根据历史执行计划创建绑定的能力 #39199 @qw4990

  TiDB v7.2.0 进一步增强根据历史执行计划创建绑定的能力，加强对复杂语句的解析和绑定，使绑定更稳固，并新增支持对以下 Hint 的绑定：

  - AGG_TO_COP()
  - LIMIT_TO_COP()
  - ORDER_INDEX
  - NO_ORDER_INDEX()

  更多信息，请参考用户文档。

- 提供 Optimizer Fix Controls 机制对优化器行为进行细粒度控制 #43169 @time-and-fate

  为了生成更合理的执行计划，TiDB 优化器的行为会随产品迭代而不断演进。但在某些特定场景下，这些变化可能引发性能回退。因此 TiDB 引入了 Optimizer Fix Controls 来控制优化器的一部分细粒度行为，你可以对一些新的变化进行回滚或控制。

  每一个可控的行为，都有一个与 Fix 号码对应的 GitHub Issue 进行说明。所有可控的行为列举在文档 Optimizer Fix Controls 中。通过设置系统变量 tidb_opt_fix_control 可以为一个或多个行为设置目标值，进而达到行为控制的目的。

  Optimizer Fix Controls 机制加强了你对 TiDB 优化器的细粒度管控能力，为升级过程引发的性能问题提供了新的修复手段，提升 TiDB 的稳定性。

  更多信息，请参考用户文档。

- 轻量统计信息初始化 GA #42160 @xuyifangreeneyes

  轻量统计信息初始化自 v7.2.0 成为正式功能。轻量级的统计信息初始化可以大幅减少启动时必须加载的统计信息的数量，从而提升启动过程中统计信息的加载速度。该功能提升了 TiDB 在复杂运行环境下的稳定性，并降低了部分 TiDB 节点重启对整体服务的影响。

  从 v7.2.0 起，新建的集群在启动阶段将默认加载轻量级统计信息，并在加载完成后再对外提供服务。对于从旧版本升级上来的集群，可通过修改 TiDB 配置项 lite-init-stats 和 force-init-stats 为 true 开启此功能。

  更多信息，请参考用户文档。

4.1.3 SQL 功能

- 支持 CHECK 约束 #41711 @fzzf678

从 v7.2.0 开始，你可以通过 CHECK 约束限制表中的一个或者多个字段值必须满足特定的条件。当为表添加 CHECK 约束后，在插入或者更新表的数据时，TiDB 会先检查约束条件是否满足，只允许满足约束的数据写入。

该功能默认关闭，你可以通过将变量 tidb_enable_check_constraint 设置为 ON 开启该功能。

更多信息，请参考用户文档。

4.1.4 数据库管理

- DDL 任务支持暂停和恢复操作（实验特性）#18015 @godouxm

  TiDB v7.2.0 之前的版本中，当 DDL 任务在执行期间遇到业务高峰时，为了减少对业务的影响，只能手动取消 DDL 任务。TiDB v7.2.0 引入了 DDL 任务的暂停和恢复功能，你可以在高峰时间点暂停 DDL 任务，等到业务高峰时间结束后再恢复 DDL 任务，从而避免了 DDL 操作对业务负载的影响。

  例如，可以通过如下 ADMIN PAUSE DDL JOBS 或 ADMIN RESUME DDL JOBS 语句暂停或者恢复多个 DDL 任务：

  ```
  ADMIN PAUSE DDL JOBS 1,2;
  ADMIN RESUME DDL JOBS 1,2;
  ```

  更多信息，请参考用户文档。

4.1.5 数据迁移

- 引入新的 SQL 语句 IMPORT INTO，大幅提升数据导入效率（实验特性）#42930 @D3Hunter

  IMPORT INTO 集成了 TiDB Lightning 物理导入模式的能力。通过该语句，你可以将 CSV、SQL 和 PARQUET 等格式的数据快速导入到 TiDB 的一张空表中。这种导入方式无需单独部署和管理 TiDB Lightning，在降低了数据导入难度的同时，大幅提升了数据导入效率。

对于存储在 Amazon S3 或 GCS 的数据文件，在开启了后端任务分布式框架后，IMPORT INTO 还支持将数据导入任务拆分成多个子任务，并将子任务调度到多个 TiDB 节点并行导入，进一步提升导入性能。

更多信息，请参考用户文档。

- TiDB Lightning 支持将字符集为 latin1 的源文件导入到 TiDB 中 #44434 @lance6716

  通过此功能，你可以使用 TiDB Lightning 将字符集为 latin1 的源文件直接导入到 TiDB 中。在 v7.2.0 之前，导入这样的文件需要额外的预处理或转换。从 v7.2.0 起，你只需在配置 TiDB Lightning 导入任务时指定 character-set = "latin1"，TiDB Lightning 就会在导入过程中自动处理字符集的转换，以确保数据的完整性和准确性。

  更多信息，请参考用户文档。

4.2 兼容性变更

注意：

以下为从 v7.1.0 升级至当前版本 (v7.2.0) 所需兼容性变更信息。如果从 v7.0.0 或之前版本升级到当前版本，可能也需要考虑和查看中间版本 release notes 中提到的兼容性变更信息。

4.2.1 系统变量

变量名	修改类型	描述
last_insert_id	修改	该变量的最大值从 9223372036854775807 修改为 18446744073709551615，和 MySQL 保持一致。
tidb_enable_non_prepared_plan_cache	修改	经进一步的测试后，该变量默认值从 OFF 修改为 ON，即默认开启非 Prepare 语句执行计划缓存。

变量名	修改类型	描述
tidb_remove_orderby_in_subquery	修改	经进一步的测试后，该变量默认值从 OFF 修改为 ON，即优化器改写会移除子查询中的 ORDER BY 子句。
tidb_analyze_skip_column_types	新增	这个变量表示在执行 ANALYZE 命令收集统计信息时，跳过哪些类型的列的统计信息收集。该变量仅适用于 tidb_analyze_version = 2 的情况。使用 ANALYZE TABLE t COLUMNS c1, ..., cn 语法时，如果指定的列的类型在 tidb_analyze_skip_column_types 中，则不会收集该列的统计信息。
tidb_enable_check_constraint	新增	这个变量用于控制 CHECK 约束功能是否开启。默认值 OFF 表示该功能默认关闭。
tidb_enable_fast_table_check	新增	这个变量用于控制是否使用基于校验和的方式来快速检查表中数据和索引的一致性。默认值 ON 表示该功能默认开启。
tidb_enable_tiflash_pipeline_model	新增	这个变量用来控制是否启用 TiFlash 新的执行模型 Pipeline Model，默认值为 OFF，即关闭 Pipeline Model。
tidb_expensive_txn_time_threshold	新增	控制打印 expensive transaction 日志的阈值时间，默认值是 600 秒。expensive transaction 日志会将尚未 COMMIT 或 ROLLBACK 且持续时间超过该阈值的事务的相关信息打印出来。

4.2.2 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	lite-init-stats	修改	经进一步的测试后，默认值从 false 修改为 true，表示 TiDB 启动时默认采用轻量级的统计信息初始化，以提高启动时统计信息初始化的效率。
TiDB	force-init-stats	修改	配合 lite-init-stats，默认值从 false 修改为 true，表示 TiDB 启动时会等到统计信息初始化完成后再对外提供服务。
TiKV	rocksdb.[defaultcf|writecf|lockcf].compaction-	修改	为减小 RocksDB 中 compaction 任务的数据量，该变量默认值从 "8MB" 修改为 "1MB"。

配置文件	配置项	修改类型	描述
	guard-min-output-file-size		
TiKV	rocksdb.[defaultcf|writecf|lockcf].optimize-filters-for-memory	新增	控制是否生成能够最小化内存碎片的 Bloom/Ribbon filter。
TiKV	rocksdb.[defaultcf|writecf|lockcf].periodic-compaction-seconds	新增	控制周期性 compaction 的时间，修改时间超过此值的 SST 文件将被选中进行 compaction，并被重新写入这些 SST 文件所在的层级。
TiKV	rocksdb.[defaultcf|writecf|lockcf].ribbon-filter-above-level	新增	控制是否对于大于等于该值的 level 使用 Ribbon filter，对于小于该值的 level，使用非 block-based bloom filter。
TiKV	rocksdb.[defaultcf|writecf|lockcf].ttl	新增	设置 SST 文件被自动选中执行 compaction 的 TTL 时间。更新时间超过 TTL 的 SST 文件将被选中并进行 compaction。
TiDB Lightning	send-kv-pairs	废弃	从 v7.2.0 版本开始，send-kv-pairs 不再生效。你可以使用新参数 send-kv-size 来指定物理导入模式下向 TiKV 发送数据时一次请求的最大大小。
TiDB Lightning	character-set	修改	扩展支持导入的字符集，新增 latin1 选项，用于导入字符集为 latin1 的源文件。
TiDB Lightning	send-kv-size	新增	用于设置单次发送到 TiKV 的键值对的大小。当键值对的大小达到设定的阈值时，它们将被 TiDB Lightning 立即发送到 TiKV，避免在导入大宽表的时候由于 TiDB Lightning 节点内存积累键值对过多导致 OOM 的问题。通过调整该参数，你可以在内存使用和导入速度之间找到平衡，提高导入过程的稳定性和效率。
Data Migration	strict-optimistic-shard-mode	新增	用于兼容历史版本 TiDB Data Migration v2.0 的分库分表同步 DDL 的行为。当用户选择乐观模式时，可以启用该参数，开启后，乐观模式下，同步任务遇到二类 DDL 时，整个任务会中断。在多

配置文件	配置项	修改类型	描述
			个表的 DDL 变更有依赖关系的场景，可以及时中断同步，在用户手动处理完各表的 DDL 后，再继续同步数据，保障上下游数据的一致性。
TiCDC	sink.protocol	修改	扩展下游类型是 Kafka 时的可选值范围：增加 "open-protocol"。用于指定编码消息时使用的格式协议。
TiCDC	sink.delete-only-output-handle-key-columns	新增	指定 DELETE 事件的输出内容，只对 "canal-json" 和 "open-protocol" 协议有效。默认值为 false，即输出所有列的内容。当设置为 true 时，只输出主键列，或唯一索引列的内容。

4.3 改进提升

- TiDB
  - 优化构造索引扫描范围的逻辑，支持将一些复杂条件转化为索引扫描范围 #41572 #44389 @xuyifangreeneyes
  - 新增 Stale Read OPS、Stale Read Traffic 监控指标 #43325 @you06
  - 当 Stale Read 的 retry leader 遇到 lock 时，resolve lock 之后强制重试 leader，避免无谓开销 #43659 @you06
  - 使用估计时间计算 Stale Read ts，减少 Stale Read 的开销 #44215 @you06
  - 添加 long-running 事务日志和系统变量 #41471 @crazycs520
  - 支持通过压缩的 MySQL 协议连接 TiDB，提升数据密集型查询在低网络质量下的性能，并节省带宽成本。该功能支持基于 zlib 和 zstd 的压缩算法 #22605 @dveeden
  - 支持将 utf8 和 utf8mb3 识别为旧的三字节 UTF-8 字符集编码，有助于将具有旧字符集编码的表从 MySQL 8.0 迁移到 TiDB #26226 @dveeden
  - 支持在 UPDATE 语句中使用 := 进行赋值操作 #44751 @CbcWestwolf

- TiKV
  - 支持通过 pd.retry-interval 配置在连接请求失败等场景下 PD 连接的重试间隔 #14964 @rleungx
  - 优化资源管控调度算法，将全局的资源使用量作为调度因素 #14604 @Connor1996
  - 使用 gzip 压缩 check_leader 请求以减少流量 #14553 @you06
  - 为 check_leader 请求增加相关监控项 #14658 @you06
  - 详细记录 TiKV 处理写入命令过程中的时间信息 #12362 @cfzjywxk
- PD
  - PD Leader 选举使用单独的 gRPC 连接，防止受到其他请求的影响 #6403 @rleungx
  - 默认开启 bucket split 以改善多 Region 的热点问题 #6433 @bufferflies
- Tools
  - Backup & Restore (BR)
    - 为外部存储 Azure Blob Storage 提供共享访问签名 (SAS) 的访问方式 #44199 @Leavrth
  - TiCDC
    - 优化同步到对象存储场景下发生 DDL 时存放数据文件目录的结构 #8891 @CharlesCheung96
    - 在同步到 Kafka 场景下，支持 OAUTHBEARER 认证方式 #8865 @hi-rustin
    - 在同步到 Kafka 场景下，对于 DELETE 操作，支持选择只输出 Handle Key #9143 @3AceShowHand
  - TiDB Data Migration (DM)
    - 支持读取 MySQL 8.0 中的压缩 binlog 作为增量同步的数据源 #6381 @dveeden
  - TiDB Lightning

- 优化导入数据过程中的重试机制，避免因 leader 切换而导致的错误 #44478 @lance6716
- 数据导入完成后使用 SQL 方式校验 checksum，提升数据校验的稳定性 #41941 @GMHDBJD
- 优化导入宽表时 TiDB Lightning 发生 OOM 的问题 #43853 @D3Hunter

4.4 错误修复

- TiDB
  - 修复使用 CTE 的查询导致 TiDB 卡住的问题 #43749 #36896 @guo-shaoge
  - 修复 min, max 查询结果出错的问题 #43805 @wshwsh12
  - 修复 SHOW PROCESSLIST 语句无法显示子查询时间较长语句的事务的 TxnStart 的问题 #40851 @crazycs520
  - 修复由于 Coprocessor task 中 TxnScope 缺失导致 Stale Read 全局优化不生效的问题 #43365 @you06
  - 修复 follower read 未处理 flashback 错误而进行重试，导致查询报错的问题 #43673 @you06
  - 修复 ON UPDATE 语句没有正确更新主键导致数据索引不一致的问题 #44565 @zyguan
  - 修改 UNIX_TIMESTAMP() 函数的上限为 3001-01-19 03:14:07.999999 UTC，与 MySQL 8.0.28+ 保持一致 #43987 @YangKeao
  - 修复在 ingest 模式下创建索引失败的问题 #44137 @tangenta
  - 修复取消处于 rollback 状态的 DDL 任务导致相关元数据出错的问题 #44143 @wjhuang2016
  - 修复 memTracker 配合 cursor fetch 使用导致内存泄漏的问题 #44254 @YangKeao
  - 修复删除数据库导致 GC 推进慢的问题 #33069 @tiancaiamao

  - 修复分区表在 Index Join 的 probe 阶段找不到对应行而报错的问题 #43686 @AilinKid @mjonss
  - 修复在创建分区表时使用 SUBPARTITION 没有警告提醒的问题 #41198 #41200 @mjonss
  - 修复执行时间超过 MAX_EXECUTION_TIME 的 query 被 kill 时的报错信息和 MySQL 不一致的问题 #43031 @dveeden
  - 修复 LEADING hint 不支持查询块别名 (query block alias) 的问题 #44645 @qw4990
  - 修复 LAST_INSERT_ID() 函数的返回类型，从 VARCHAR 变更为 LONGLONG，与 MySQL 一致 #44574 @Defined2014
  - 修复在带有非关联子查询的语句中使用公共表表达式 (CTE) 可能导致结果错误的问题 #44051 @winoros
  - 修复 Join Reorder 可能会造成 Outer Join 结果错误的问题 #44314 @AilinKid
  - 修复 PREPARE stmt FROM "ANALYZE TABLE xxx" 会被 tidb_mem_quota_query kill 掉的问题 #44320 @chrysan
- TiKV
  - 修复处理 stale 悲观锁冲突时事务返回值不正确的问题 #13298 @cfzjywxk
  - 修复内存悲观锁可能导致 Flashback 失败和数据不一致的问题 #13303 @JmPotato
  - 修复处理过期请求时 fair lock 的正确性问题 #13298 @cfzjywxk
  - 修复 autocommit 和 point get replica read 可能破坏线性一致性的问题 #14715 @cfzjywxk
- PD
  - 修复在特殊情况下冗余副本无法自动修复的问题 #6573 @nolouch
- TiFlash

- 修复查询在 Join build 侧数据非常大，且包含许多小型字符串类型列时，消耗的内存可能会超过实际需要的问题 #7416 @yibin87

- Tools
  - Backup & Restore (BR)
    - 修复某些情况下误报 checksum mismatch 的问题 #44472 @Leavrth
    - 修复某些情况下误报 resolved lock timeout 的问题 #43236 @YuJuncen
    - 修复在恢复统计信息的时候可能会 panic 的问题 #44490 @tangenta
  - TiCDC
    - 修复在某些特殊情况下 Resolved TS 不能正常推进的问题 #8963 @CharlesCheung96
    - 修复使用 Avro 或 CSV 协议场景下 UPDATE 操作不能输出旧值的问题 #9086 @3AceShowHand
    - 修复同步到 Kafka 场景下，读取下游 Metadata 太频繁导致下游压力过大的问题 #8959 @hi-rustin
    - 修复同步到 TiDB 或 MySQL 场景下，频繁设置下游双向复制相关变量导致下游日志过多的问题 #9180 @asddongmen
    - 修复 PD 节点宕机导致 TiCDC 节点重启的问题 #8868 @asddongmen
    - 修复 TiCDC 同步到 Kafka-on-Pulsar 时不能正确建立连接的问题 #8892 @hi-rustin
  - TiDB Lightning
    - 修复开启 experimental.allow-expression-index 且默认值是 UUID 时导致 TiDB Lightning panic 的问题 #44497 @lichunzhu

- 修复划分数据文件时任务退出导致 TiDB Lightning panic 的问题 #43195 @lance6716

4.5 贡献者

感谢来自 TiDB 社区的贡献者们：

- asjdf
- blacktear23
- Cavan-xu
- darraes
- demoManito
- dhysum
- HappyUncle
- jiyfhust
- L-maple
- nyurik
- SeigeC
- tangjingyu97