TiDB Release Notes：从 v7.6.0 到 v8.1.0 的变更

PingCAP

20240524

Abstract

本文档包含 TiDB v7.6.0-DMR、v8.0.0-DMR 和 v8.1.0-LTS 的 release notes。当你从 v7.5.x 升级到 v8.1.0 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 v8.1.0 的详细信息和使用文档，请参考 TiDB v8.1 文档。

目录

1 TiDB 8.1.0 Release Notes

发版日期：2024 年 5 月 24 日

TiDB 版本：8.1.0

试用链接：快速体验 | 生产部署 | 下载离线包

TiDB 8.1.0 为长期支持版本 (Long-Term Support Release, LTS)。

相比于前一个 LTS（即 7.5.0 版本），8.1.0 版本包含 7.6.0-DMR 和 8.0.0-DMR 中已发布的新功能、提升改进和错误修复。当你从 7.5.x 升级到 8.1.0 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 7.6.0 到 8.1.0 的一些关键特性：

分类	功能	描述
可扩展性与性能	提升 BR 快照恢复速度（从 v8.0.0 开始 GA）	通过该功能，BR 可以充分利用集群的规模优势，使 TiKV 集群中的所有节点都能参与到数据恢复的准备阶段，从而显著提升大规模集群中大数据集的恢复速度。实际测试表明，该功能可将下载带宽打满，下载速度可提升 8 到 10 倍，端到端恢复速度大约提升 1.5 到 3 倍。
	建表性能提升 10 倍（实验特性，从 v7.6.0 开始引入）	在 v7.6.0 中引入了新的 DDL 架构，批量建表的性能提高了 10 倍。这一重大改进极大地缩短了创建大量表所需的时间。特别是在 SaaS 场景中，快速创建大量表（从数万到数十万不等）是一个常见的挑战，使用该特性能显著提升 SaaS 场景的建表速度。

分类	功能	描述
	通过 Active PD Follower 提升 PD Region 信息查询服务的扩展能力（实验特性，从 v7.6.0 开始引入）	TiDB v7.6.0 实验性地引入了 Active PD Follower 特性，允许 PD follower 提供 Region 信息查询服务。在 TiDB 节点数量较多和 Region 数量较多的集群中，该特性可以提升 PD 集群处理 GetRegion、ScanRegions 请求的能力，减轻 PD leader 的 CPU 压力。
	用于处理更大事务的批量 DML 执行方式（实验特性，从 v8.0.0 开始引入）	大批量的 DML 任务，例如大规模的清理任务、连接或聚合，可能会消耗大量内存，并且在非常大的规模上受到限制。批量 DML (tidb_dml_type = "bulk") 是一种新的 DML 类型，用于更高效地处理大批量 DML 任务，同时提供事务保证并减轻 OOM 问题。该功能与用于数据加载的导入、加载和恢复操作不同。
	增强在有大量表时缓存 schema 信息的稳定性（实验特性，从 v8.0.0 开始引入）	对于使用 TiDB 作为多租户应用程序记录系统的 SaaS 公司，经常需要存储大量的表。在以前的版本中，尽管支持处理百万级或更大数量的表，但可能会影响用户体验。TiDB v8.0.0 通过以下增强功能改善了这一问题： • 引入新的 schema 缓存系统，为表元数据

分类	功能	描述
		提供了基于 LRU (Least Recently Used) 算法的缓存策略，优先将最近访问频率较高的表元数据存储在缓存中，从而减少表数量较多场景下的内存占用。 • 支持在 auto analyze 中配置优先队列，使流程更加流畅，并在大量表的情况下提高稳定性。
稳定性与高可用	全局排序成为正式功能（从 v8.0.0 开始 GA）	全局排序功能旨在提高 IMPORT INTO 和 CREATE INDEX 的稳定性与效率。通过对需要处理的数据进行全局排序，可以提高数据写入 TiKV 的稳定性、可控性和可扩展性，从而提升数据导入与索引添加的用户体验和服务质量。启用全局排序后，单条 IMPORT INTO 或 CREATE INDEX 语句目前已经支持对高达 40 TiB 的数据进行导入或者添加索引。
	跨数据库绑定执行计划（从 v7.6.0 开始引入）	在处理上百个 schema 相同的数据库时，针对其中一个数据库的 SQL binding 通常也适用于其它的数据库。例如，在 SaaS 或 PaaS 数据平台中，每个用户通常各自维护单独的数

分类	功能	描述
		据库，这些数据库具有相同的 schema 并运行着类似的 SQL。在这种情况下，逐一为每个数据库做 SQL 绑定是不切实际的。TiDB v7.6.0 引入跨数据库绑定执行计划，支持在所有 schema 相同的数据库之间匹配绑定计划。
	支持 TiProxy（从 v8.0.0 开始 GA）	全面支持 TiProxy，可通过部署工具轻松部署。TiProxy 可以管理和维护客户端与 TiDB 的连接，在滚动重启、升级以及扩缩容过程中保持连接。
	Data Migration (DM) 正式支持迁移 MySQL 8.0（从 v7.6.0 开始 GA）	在 v7.6.0 之前，DM 迁移 MySQL 8.0 仅为实验特性，不能用于生产环境。TiDB v7.6.0 增强了该功能的稳定性、兼容性，可在生产环境帮助你平滑、快速地将数据从 MySQL 8.0 迁移到 TiDB。在 v7.6.0 中，该功能正式 GA。
	资源管控支持管理资源消耗超出预期的查询（从 v8.1.0 开始 GA）	通过资源组的规则，TiDB 能够自动识别出运行超出预期的查询，并对该查询进行限流或取消处理。即使没有被规则识别，你仍然可以手动添加查询特征以及采取对应的措施，从而降低突发的查询性能问题对整个数据库的影响。
数据库管理与可观测性	支持观测索引使用情况（从 v8.0.0 开始引入）	正确的索引设计是提升数据库性能的重要前提。TiDB v8.0.0 引入内存表

分类	功能	描述
		INFORMATION_SCHEMA.TIDB_INDEX_USAGE 和视图 sys.schema_unused_indexes，用于记录索引的使用情况。该功能有助于用户评估数据库中索引的效率并优化索引设计。
数据迁移	TiCDC 支持 Simple 协议（从 v8.0.0 开始引入）	TiCDC 支持了新的 Simple 消息协议，该协议通过在 DDL 和 BOOTSTRAP 事件中嵌入表的 schema 信息，实现了对 schema 信息的动态追踪 (in-band schema tracking)。
	TiCDC 支持 Debezium 协议（从 v8.0.0 开始引入）	TiCDC 支持了新的 Debezium 协议，TiCDC 可以使用该协议生成 Debezium 格式的数据变更事件并发送给 Kafka sink。
	TiCDC 支持客户端鉴权（从 v8.1.0 开始引入）	TiCDC 支持使用 mTLS（双向传输层安全性协议）或 TiDB 用户名密码进行客户端鉴权。该功能允许命令行工具或 OpenAPI 客户端验证与 TiCDC 的连接。

1.1 功能详情

1.1.1 稳定性

- 管理资源消耗超出预期的查询成为正式功能 (GA) #43691 @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 引入了对资源超出预期的查询的管理能力，以快速减小 SQL 性能造成的影响范围。该功能在 v8.1.0 成为正式功能。

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。

TiDB 同时支持手动标记查询的功能。利用命令 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，达到添加 SQL 黑名单的目的。

对资源消耗超出预期的查询的自动管理能力为用户提供了有效的手段，在根本原因被定位之前，该功能可以快速缓解查询问题对整体性能的影响，从而提升数据库的稳定性。

更多信息，请参考用户文档。

1.1.2 SQL 功能

- 支持在 TiDB 建表时使用更多的表达式设置列的默认值成为正式功能 (GA) #50936 @zimulala

  在 v8.0.0 之前，建表时指定列的默认值仅限于固定的字符串、数字、日期和个别表达式。从 v8.0.0 开始，TiDB 支持使用更多表达式作为列的默认值，例如将列的默认值设置为 DATE_FORMAT，从而满足多样化的业务需求。在 v8.1.0 中，该特性成为正式功能。

  从 v8.1.0 开始，支持在使用 ADD COLUMN 添加列时使用表达式作为默认值。

  更多信息，请参考用户文档。

1.1.3 数据库管理

- 默认开启 TiDB 分布式执行框架，提升并行执行 ADD INDEX 或 IMPORT INTO 任务的性能和稳定性 #52441 @D3Hunter

  TiDB 分布式执行框架在 v7.5.0 中成为正式功能 (GA)，但默认关闭，即一个 ADD INDEX 或 IMPORT INTO 任务默认只能由一个 TiDB 节点执行。

  从 v8.1.0 起，该功能默认开启（tidb_enable_dist_task 默认为 ON）。开启后，分布式执行框架可以调度多个 TiDB 节点并行执行同一个 ADD INDEX 或 IMPORT INTO 任务，从而充分利用 TiDB 集群的资源，大幅提升这些任务的性能。此外，你还可以通过增加 TiDB 节点并为新增的节点配置 tidb_service_scope 来线性提升 ADD INDEX 和 IMPORT INTO 任务的性能。

  更多信息，请参考用户文档。

1.1.4 安全

- 增强 TiDB 日志脱敏成为正式功能 (GA) #52364 @xhebox

  TiDB 日志脱敏增强是通过对日志文件中的 SQL 文本信息进行标记，支持在查看日志时删除敏感数据。你可以控制是否对日志信息进行标记，以实现在不同场景下安全使用 TiDB 日志，提升了使用日志脱敏能力的安全性和灵活性。要使用此功能，可以将系统变量 tidb_redact_log 的值设置为 MARKER，此时 TiDB 运行日志中的 SQL 文本会被标记。还可以通过 TiDB server 的 collect-log 子命令将日志中标记的敏感数据删除，在数据安全的情况下展示日志；或移除所有标记，获取正常日志。该功能在 v8.1.0 成为正式功能。

  更多信息，请参考用户文档。

1.1.5 数据迁移

- IMPORT INTO ... FROM SELECT 语法成为正式功能 (GA) #49883 @D3Hunter

在 v8.0.0 之前的版本中，将查询结果导入目标表只能通过 INSERT INTO ... SELECT 语句，但该语句在一些大数据量的场景中导入效率较低。在 v8.0.0 中，TiDB 以实验特性新增支持通过 IMPORT INTO ... FROM SELECT 将 SELECT 的查询结果导入到一张空的 TiDB 目标表中，其性能最高可达 INSERT INTO ... SELECT 的 8 倍，可以大幅缩短导入所需的时间。此外，你还可以通过 IMPORT INTO ... FROM SELECT 导入使用 AS OF TIMESTAMP 查询的历史数据。

在 v8.1.0 中，IMPORT INTO ... FROM SELECT 语法成为正式功能 (GA)，丰富了 IMPORT INTO 语句的功能场景。

更多信息，请参考用户文档。

- TiDB Lightning 简化冲突处理策略，同时支持以 replace 方式处理冲突数据的功能成为正式功能 (GA) #51036 @lyzx2001

  在 v8.0.0 之前的版本中，TiDB Lightning 逻辑导入模式有一套数据冲突处理策略，而物理导入模式有两套数据冲突处理策略，不易理解和配置。

  在 v8.0.0 中，TiDB Lightning 废弃了物理导入模式下的旧版冲突检测策略，并以实验特性支持通过 conflict.strategy 参数统一控制逻辑导入和物理导入模式的冲突检测策略，并简化了该参数的配置。此外，在物理导入模式下，当导入遇到主键或唯一键冲突的数据时，replace 策略支持保留最新的数据、覆盖旧的数据。在 v8.1.0 中，以 replace 方式处理冲突数据的功能成为正式功能 (GA)。

  更多信息，请参考用户文档。

- TiCDC 支持客户端鉴权 #10636 @CharlesCheung96

  在 v8.1.0 中，当使用 TiCDC CLI 或 OpenAPI 时，TiCDC 支持客户端鉴权。你可以配置 TiCDC 要求客户端使用证书进行鉴权，以实现 mTLS（双

向传输层安全性协议）。此外，你还可以使用 TiDB 用户名密码进行客户端鉴权。

更多信息，请参考用户文档。

1.2 兼容性变更

注意：

以下为从 v8.0.0 升级至当前版本 (v8.1.0) 所需兼容性变更信息。如果从 v7.6.0 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 行为变更

- 在之前的版本中，TiDB Lightning 的配置项 tidb.tls 在取值为 "false" 和 "" 时的行为是相同的，在取值为 "skip-verify" 和 "preferred" 时的行为也是相同的。从 v8.1.0 开始，TiDB Lightning 对 tidb.tls 取值为 "false"、""、"skip-verify" 和 "preferred" 时的行为进行了区分。更多信息，请参考 TiDB Lightning 配置参数。
- 对于设置了 AUTO_ID_CACHE=1 的表，TiDB 支持中心化分配自增 ID 服务。在之前的版本中，该服务的“主”TiDB 节点在进程退出（如该 TiDB 节点重启）时会自动执行 forceRebase 操作，以确保自动分配的 ID 尽可能连续。然而，当设置过 AUTO_ID_CACHE=1 的表过多时，执行 forceRebase 会非常耗时，导致 TiDB 无法及时重启，甚至阻塞数据写入，影响系统可用性。因此，从 v8.1.0 起，TiDB 取消了 forceRebase 操作，解决了上述问题，但会造成主备切换期间部分自动分配的 ID 出现不连续。

1.2.2 系统变量

变量名	修改类型	描述
tidb_auto_analyze_ratio	修改	取值范围从 [0, 18446744073709

变量名	修改类型	描述
		551615] 修改为 (0, 1]。
tidb_enable_dist_task	修改	默认值从 OFF 修改为 ON，代表默认开启分布式执行框架，从而充分利用 TiDB 集群的资源，大幅提升 ADD INDEX 和 IMPORT INTO 任务的性能。如果要从低版本的集群升级到 v8.1.0 或更高版本，且该集群已开启分布式执行框架，为了避免升级期间 ADD INDEX 操

变量名	修改类型	描述
		作可能导致数据索引不一致的问题，请在升级前关闭分布式执行框架（即将 tidb_enable_dist_task 设置为 OFF），升级后再手动开启。
tidb_service_scope	修改	该变量的可选值从 "" 或 background 修改为长度小于或等于 64 的字符串，可用合法字符包括数字 0-9、字母 a-zA-Z、下划线 _ 和连字符 -，从而更

变量名	修改类型	描述
		灵活地控制各 TiDB 节点的服务范围。分布式执行框架会根据该变量的值决定将分布式任务调度到哪些 TiDB 节点上执行，具体规则请参考任务调度。

1.2.3 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	concurrently-init-stats	新增	用于控制 TiDB 启动时是否并发初始化统计信息。默认值为 false。
TiDB Lightning	conflict.max-record-rows	修改	从 v8.1.0 开始，TiDB Lightning 会自动将 conflict.max-record-rows 的值设置为 conflict.threshold 的值，并忽略用户输入，因此无需再单独配置 conflict.max-record-rows。conflict.max-record-rows 将在未来版本中废弃。
TiDB Lightning	conflict.threshold	修改	默认值从 9223372036854775807 修改为 10000，从而迅速中断异常任务，以便用户尽快进行相应调整。这避免了在导入完成后，才发现是因为数据源异常或表结

配置文件	配置项	修改类型	描述
			构定义错误导致导入了大量冲突数据，从而节省时间和计算资源。
TiCDC	security.client-allowed-user	新增	指定可用于客户端鉴权的用户名，列表中不存在的用户的鉴权请求将被直接拒绝。默认值为 null。
TiCDC	security.client-user-required	新增	控制是否使用 TiDB 的用户名和密码进行客户端鉴权，默认值为 false。
TiCDC	security.mtls	新增	控制是否开启 TLS 客户端鉴权，默认值为 false。
TiCDC	sink.debezium.output-old-value	新增	控制是否输出行数据更改前的值。默认值为 true。关闭后，UPDATE 事件不会输出 “before” 字段的数据。
TiCDC	sink.open.output-old-value	新增	控制是否输出行数据更改前的值。默认值为 true。关闭后，UPDATE 事件不会输出 “p” 字段的数据。

1.3 废弃功能

- 计划在后续版本重新设计执行计划绑定的自动演进，相关的变量和行为会发生变化。
- TiDB Lightning 参数 conflict.max-record-rows 计划在未来版本中废弃，并在后续版本中删除。该参数将由 conflict.threshold 替代，即记录的冲突记录数和单个导入任务允许出现的冲突记录数的上限数保持一致。
- 从 v8.0.0 开始，TiDB Lightning 废弃了物理导入模式下的旧版冲突检测策略，支持通过 conflict.strategy 参数统一控制逻辑导入和物理导入模式的冲突检测策略。旧版冲突检测的参数 duplicate-resolution 将在未来版本中被移除。

1.4 改进提升

- TiDB

  - 优化外键在 SHOW CREATE TABLE 结果中的 MySQL 兼容性 #51837 @negachov
  - 优化表达式默认值在 SHOW CREATE TABLE 结果中的 MySQL 兼容性 #52939 @CbcWestwolf
  - 允许使用 ingest 模式并发添加多个索引 #52596 @lance6716
  - 允许将系统变量 tidb_service_scope 设置为不同的值，以更好地利用分布式框架功能 #52441 @ywqzzy
  - 增强 TiDB 对始终为 false 的 DNF 项的处理能力，直接忽略这种过滤条件，以避免不必要的全表扫描 #40997 @hi-rustin
  - 当查询可以选择除全表扫描以外的单索引扫描方式时，支持以 Optimizer Fix Controls 的方式解除优化器不会自动选择索引合并的限制 #52869 @time-and-fate
  - 在 Coprocessor 算子的实际执行信息 execution info 中添加 total_kv_read_wall_time 指标 #28937 @cfzjywxk
  - 在 Resource Control 面板中添加 RU (max) 监控指标 #49318 @nolouch
  - 为 LDAP 身份认证添加超时机制，避免资源锁 (RLock) 无法及时释放的问题 #51883 @YangKeao
- TiKV
  - 在 Raftstore 线程中避免进行快照文件的 IO 操作，提高 TiKV 稳定性 #16564 @Connor1996
  - 加快 TiKV 停机的速度 #16680 @LykxSassinator
  - 增加每个线程内存使用量的监控指标 #15927 @Connor1996
- PD
  - 优化 OperatorController 的逻辑，减少竞争锁的开销 #7897 @nolouch
- TiFlash

- 降低 TiFlash 在开启 TLS 后因更新证书而导致 panic 的概率 #8535 @windtalker

- Tools
  - Backup & Restore (BR)
    - 增加 PITR 集成测试用例，覆盖对日志备份与添加索引加速功能的兼容性测试 #51987 @Leavrth
    - 移除日志备份启动时检查是否存在活动 DDL job 的无效检查 #52733 @Leavrth
    - 增加测试用例，用于测试 PITR 和添加索引加速之间的兼容性 #51988 @Leavrth
    - BR 在恢复数据过程中，会清理空的 SST 文件 #16005 @Leavrth
  - TiCDC
    - 提升使用 redo log 恢复数据过程中的内存稳定性，减少 OOM 的概率 #10900 @CharlesCheung96
    - 显著提升事务冲突场景中的数据同步的稳定性，性能最高提升可达 10 倍 #10896 @CharlesCheung96

1.5 错误修复

- TiDB
  - 修复当 SQL 语句涉及包含多值索引的表时，执行可能报错 Can't find a proper physical plan for this query 的问题 #49438 @qw4990
  - 修复自动统计信息收集在 OOM 后卡住的问题 #51993 @hi-rustin
  - 修复使用 BR 恢复一张表后，即使该表没有统计信息，统计信息健康度仍然显示为 100% 的问题 #29769 @winoros
  - 修复 TiDB 在升级过程中会为系统表创建统计信息的问题 #52040 @hi-rustin
  - 修复 TiDB 在统计信息初始化完成前就开始自动收集的问题 #52346 @hi-rustin

- 修复启用 tidb_mem_quota_analyze 时，更新统计信息使用的内存超过限制可能导致 TiDB crash 的问题 #52601 @hawkingrei
- 修复 TiDB 统计信息同步加载机制无限重试加载空统计信息并打印 fail to get stats version for this histogram 日志的问题 #52657 @hawkingrei
- 修复关闭新排序规则框架时，涉及不同排序规则的表达式可能导致查询 panic 的问题 #52772 @wjhuang2016
- 修复 CPS by type 监控项显示错误的问题 #52605 @nolouch
- 修复查询 INFORMATION_SCHEMA.TIKV_REGION_STATUS 出现空指针的问题 #52013 @JmPotato
- 修复指定非法列默认值时的错误提示信息 #51592 @danqixu
- 修复 ingest 模式添加索引时，在某些特殊情况下导致数据索引不一致的问题 #51954 @lance6716
- 修复在恢复含有外键的表时 DDL 卡住的问题 #51838 @YangKeao
- 修复加索引期间 TiDB 网络隔离导致加索引失败的问题 #51846 @ywqzzy
- 修复重命名索引后再添加同名索引时报错的问题 #51431 @lance6716
- 修复添加索引期间升级集群导致数据索引不一致的问题 #52411 @tangenta
- 修复开启分布式执行框架后为大数据量的表添加索引不成功的问题 #52640 @tangenta
- 修复并发添加索引时报 no such file or directory 错误的问题 #52475 @tangenta
- 修复添加索引失败后无法清理临时数据的问题 #52639 @lance6716
- 修复元数据锁在计划缓存场景下未能阻止 DDL 推进的问题 #51407 @wjhuang2016
- 修复执行 IMPORT INTO 大数据量任务时卡住的问题 #52884 @lance6716

  - 修复打印 gRPC 错误日志时 TiDB 意外重启的问题 #51301 @guo-shaoge
  - 修复 IndexHashJoin 在执行 Anti Left Outer Semi Join 计算时输出冗余数据的问题 #52923 @yibin87
  - 修复关联子查询中 TopN 算子结果不正确的问题 #52777 @yibin87
  - 修复 HashJoin probe 执行时间统计不精确的问题 #52222 @windtalker
  - 修复在分区裁剪模式为 static 的情况下 (tidb_partition_prune_mode='static')，使用 TABLESAMPLE 返回错误结果的问题 #52282 @tangenta
  - 修复 TTL 在夏令时情况下出现 1 小时偏差的错误 #51675 @lcwangchao
  - 修复 TiDB Dashboard 监控页面中连接数 (Connection Count) 的计算和显示错误 #51889 @YangKeao
  - 修复回滚改写分区 DDL 任务时，状态卡住的问题 #51090 @jiyfhust
  - 修复 EXPLAIN ANALYZE 执行结果中 max_remote_stream 的值不正确的问题 #52646 @JaySon-Huang
  - 修复查询 TIDB_HOT_REGIONS 表时结果返回内存表 INFORMATION_SCHEMA 的问题 #50810 @Defined2014
  - 修复当某些列的统计信息没有完全加载时，EXPLAIN 语句的结果中可能会显示错误的列 ID 的问题 #52207 @time-and-fate
  - 修复 IFNULL 函数返回的类型和 MySQL 不一致的问题 #51765 @YangKeao
  - 修复添加唯一索引可能导致 TiDB panic 的问题 #52312 @wjhuang2016
- TiKV
  - 修复由于过时的 Region peer 忽略 GC 消息导致 resolve-ts 被阻塞的问题 #16504 @crazycs520

  - 修复 RocksDB 中非活跃的 WAL (Write Ahead Log) 可能损毁数据的问题 #16705 @Connor1996

- PD
  - 修复切换 PD 微服务模式时 TSO 可能卡住的问题 #7849 @JmPotato
  - 修复 DR Auto-Sync 的 State 监控指标未显示数据的问题 #7974 @lhy1024
  - 修复检查 binary 版本时可能导致 PD panic 的问题 #7978 @JmPotato
  - 修复解析 TTL 参数时发生的类型转换错误 #7980 @HuSharp
  - 修复两数据中心部署切换时 Leader 无法迁移的问题 #7992 @TonsnakeLin
  - 修复 pd-ctl 中 PrintErrln 无法将错误信息输出到 stderr（标准错误）的问题 #8022 @HuSharp
  - 修复 PD 在生成 Merge 调度时可能出现的 panic 问题 #8049 @nolouch
  - 修复 GetAdditionalInfo 导致的 panic 问题 #8079 @HuSharp
  - 修复 PD 的 Filter target 监控指标未提供 scatter range 信息的问题 #8125 @HuSharp
  - 修复 SHOW CONFIG 的查询结果包含已废弃的 trace-region-flow 配置项的问题 #7917 @rleungx
  - 修复扩缩容进度显示不准确的问题 #7726 @CabinfeverB
- TiFlash
  - 修复在非严格 sql_mode 下插入数据到带有异常默认值的列可能导致 TiFlash panic 的问题 #8803 @Lloyd-Pottiger
  - 修复 TiFlash 在高并发读的情况下，可能返回瞬时不正确结果的问题 #8845 @JinheLin

- 修复存算分离架构下，修改 TiFlash 计算节点 storage.remote.cache.capacity 配置项的值后，Grafana 中硬盘使用量监控指标 used_size 显示不正确的问题 #8920 @JinheLin
- 修复从低于 v6.5.0 的集群升级到 v6.5.0 及以上版本后，可能出现 TiFlash 元数据损坏以及进程 panic 的问题 #9039 @JaySon-Huang
- 修复存算分离架构下，TiFlash 计算节点进程停止时可能出现 panic 的问题 #8860 @guo-shaoge
- 修复 TiFlash 在执行带有虚拟生成列的查询时可能报错的问题 #8787 @guo-shaoge

- Tools
  - Backup & Restore (BR)
    - 修复在包含 AUTO_RANDOM 列的联合聚簇索引中，BR 无法备份 AUTO_RANDOM ID 分配进度的问题 #52255 @Leavrth
    - 修复在日志备份任务被暂停后，移除任务无法立即恢复 GC safepoint 的问题 #52082 @3pointer
    - 修复在小概率情况下，由于特殊的事件时序导致日志备份数据丢失的问题 #16739 @YuJuncen
    - 修复因 TiKV 重启，日志备份的 global checkpoint 推进提前于实际备份文件写入点，可能导致少量备份数据丢失的问题 #16809 @YuJuncen
    - 修复全量备份时日志中出现无效的 --concurrency 相关信息的问题 #50837 @BornChanger
    - 修复在 BR 恢复数据或 TiDB Lightning 物理导入模式下导入数据时，从 PD 获取到的 Region 没有 Leader 的问题 #51124 #50501 @Leavrth
    - 修复日志备份在暂停、停止、再重建任务操作后，虽然任务状态显示正常，但 Checkpoint 不推进的问题 #53047 @RidRisR

- 修复不稳定测试用例 TestClearCache #51671 @zxc111
- 修复不稳定测试用例 TestGetMergeRegionSizeAndCount #52095 @3pointer
- 修复不稳定集成测试 br_tikv_outage #52673 @Leavrth
- 修复测试用例 TestGetTSWithRetry 执行时间过长的问题 #52547 @Leavrth
- 修复恢复暂停的日志备份任务时，如果与 PD 的网络连接不稳定可能导致 TiKV panic 的问题 #17020 @YuJuncen

– TiCDC

- 修复调用驱逐 TiCDC owner 节点的 API (/api/v2/owner/resign) 导致 TiCDC 任务意外重启的问题 #10781 @sdojjy
- 修复当下游 Pulsar 下线后，移除 changefeed 会导致 TiCDC 正常流程卡住，从而引起其他 changefeed 进度卡住的问题 #10629 @asddongmen
- 修复 Grafana 监控中的 **Ownership history** 面板显示不稳定的问题 #10796 @hongyunyan
- 修复重启 PD 可能导致 TiCDC 节点报错重启的问题 #10799 @3AceShowHand
- 修复 PD 磁盘 IO 延迟较大导致 TiCDC 同步大幅延迟的问题 #9054 @asddongmen
- 修复 TIMEZONE 类型的值没有按照正确的时区设置默认值的问题 #10931 @3AceShowHand
- 修复没有正确同步 DROP PRIMARY KEY 和 DROP UNIQUE KEY 的问题 #10890 @asddongmen
- 修复上游写入 Exchange Partition ... With Validation DDL 后，TiCDC 向下游执行该 DDL 时失败，导致 changefeed 卡住的问题 #10859 @hongyunyan

– TiDB Lightning

- 修复 TiDB Lightning 导入数据时，因源文件存在不兼容的 SQL 语句而报 no database selected 的问题 #51800 @lance6716
- 修复 TiDB Lightning 在服务器模式下可能会将敏感信息打印到日志中的问题 #36374 @kennytm
- 修复 TiDB Lightning 导入数据时，kill PD Leader 会导致 invalid store ID 0 报错的问题 #50501 @Leavrth
- 修复 TiDB Lightning 使用 replace 方式处理冲突数据时报 Unknown column in where clause 错误的问题 #52886 @lyzx2001
- 修复导入 Parquet 格式的空表时，TiDB Lightning panic 的问题 #52518 @kennytm

1.6 性能测试

如需了解 TiDB v8.1.0 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）。

1.7 贡献者

感谢来自 TiDB 社区的贡献者们：

- arturmelanchyk（首次贡献者）
- CabinfeverB
- danqixu（首次贡献者）
- imalasong（首次贡献者）
- jiyfhust
- negachov（首次贡献者）
- testwill
- yzhan1（首次贡献者）
- zxc111（首次贡献者）

2 TiDB 8.0.0 Release Notes

发版日期：2024 年 3 月 29 日

TiDB 版本：8.0.0

试用链接：快速体验 | 下载离线包

在 8.0.0 版本中，你可以获得以下关键特性：

分类	功能/增强	描述
可扩展性与性能	支持拆分 PD 功能为微服务，提高可扩展性（实验特性）	Placement Driver (PD) 包含了多个确保 TiDB 集群能正常运行的关键模块。当集群的工作负载增加时，PD 中各模块的资源消耗也会随之增加，造成这些模块间功能的相互干扰，进而影响整个集群的服务质量。为了解决该问题，从 v8.0.0 起，TiDB 支持将 PD 的 TSO 和调度模块拆分成可独立部署的微服务，可以显著降低当集群规模扩大时模块间的互相影响。通过这种架构，TiDB 能够支持更大规模、更高负载的集群。
	用于处理更大事务的批量 DML 执行方式（实验特性）	大批量的 DML 任务，例如大规模的清理任务、连接或聚合，可能会消耗大量内存，并且在非常大的规模上受到限制。批量 DML (tidb_dml_type = "bulk") 是一种新的 DML 类型，用于更高效地处理大批量 DML 任务，同时提供事务保证并减轻

分类	功能/增强	描述
		OOM 问题。该功能与用于数据加载的导入、加载和恢复操作不同。
	提升 BR 快照恢复速度 (GA)	通过该功能，BR 可以充分利用集群的规模优势，使 TiKV 集群中的所有节点都能参与到数据恢复的准备阶段，从而显著提升大规模集群中大数据集的恢复速度。实际测试表明，该功能可将下载带宽打满，下载速度可提升 8 到 10 倍，端到端恢复速度大约提升 1.5 到 3 倍。
	增强在有大量表时缓存 schema 信息的稳定性	对于使用 TiDB 作为多租户应用程序记录系统的 SaaS 公司，经常需要存储大量的表。在以前的版本中，尽管支持处理百万级或更大数量的表，但可能会影响用户体验。TiDB v8.0.0 通过以下增强功能改善了这一问题： • 引入新的 schema 缓存系统，为表元数据提供了懒加载的 LRU (Least Recently Used) 缓存，并更有效地管理 schema 版本变更。 • 支持在 auto analyze 中配置优先队列，使流程更加流

分类	功能/增强	描述
		畅，并在大量表的情况下提高稳定性。
数据库管理与可观测性	支持观测索引使用情况	正确的索引设计是提升数据库性能的重要前提。TiDB v8.0.0 引入内存表 INFORMATION_SCHEMA.TIDB_INDEX_USAGE 和视图 sys.schema_unused_indexes ，用于记录索引的使用情况。该功能有助于用户评估数据库中索引的效率并优化索引设计。
数据迁移	TiCDC 支持 Simple 协议	TiCDC 支持了新的 Simple 消息协议，该协议通过在 DDL 和 BOOTSTRAP 事件中嵌入表的 schema 信息，实现了对 schema 信息的动态追踪 (in-band schema tracking)。
	TiCDC 支持 Debezium 协议	TiCDC 支持了新的 Debezium 协议，TiCDC 可以使用该协议生成 Debezium 格式的数据变更事件并发送给 Kafka sink。

2.1 功能详情

2.1.1 可扩展性

- PD 支持微服务模式（实验特性）#5766 @binshi-bing

  从 v8.0.0 开始，PD 支持微服务模式。该模式可将 PD 的时间戳分配和集群调度功能拆分为以下微服务单独部署，从而实现 PD 的性能扩展，解决大规模集群下 PD 的性能瓶颈问题。

  - tso 微服务：为整个集群提供单调递增的时间戳分配。
  - scheduling 微服务：为整个集群提供调度功能，包括但不限于负载均衡、热点处理、副本修复、副本放置等。

  每个微服务都以独立进程的方式部署。当设置某个微服务的副本数大于 1 时，该微服务会自动实现主备的容灾模式，以确保服务的高可用性和可靠性。

  目前 PD 微服务仅支持通过 TiDB Operator 进行部署。当 PD 出现明显的性能瓶颈且无法升级配置的情况下，建议考虑使用该模式。

  更多信息，请参考用户文档。

- 增强 Titan 引擎的易用性 #16245 @Connor1996

  - 默认启用 Titan Blob 文件和 RocksDB Block 文件的共享缓存（shared-blob-cache 默认为 true），无需再单独配置 blob-cache-size。
  - 支持动态修改 min-blob-size、blob-file-compression、discardable-ratio，以提升使用 Titan 引擎时的性能和灵活性。

  更多信息，请参考用户文档。

2.1.2 性能

- BR 快照恢复速度提升 GA #50701 @3pointer @Leavrth

  从 TiDB v8.0.0 版本起，BR 快照恢复提速功能正式发布并默认启用。通过采用粗粒度打散 Region 算法、批量创建库表、降低 SST 文件下载和 Ingest 操作之间的相互影响、加速表统计信息恢复等改进措施，快照恢复的速度有大幅提升。在实际案例中，单个 TiKV 节点的数据恢复速度稳定在 1.2 GiB/s，并且能够在 1 小时内完成对 100 TiB 数据的恢复。

  这意味着即使在高负载环境下，BR 工具也能够充分利用每个 TiKV 节点的资源，显著减少数据库恢复时间，增强数据库的可用性和可靠性，减少因数

据丢失或系统故障引起的停机时间和业务损失。需要注意的是，恢复速度的提升是因为使用了大量的 goroutine 来并行工作，会有比较大的内存消耗，特别是在表或者 Region 数很多的时候，推荐使用内存规格较高的机器来运行 BR 的客户端。如果机器的内存规格较小，建议改用细粒度的 Region 分裂打散策略。此外，因为粗粒度打散 Region 算法会占用大量的外部存储带宽，请避免因为外部带宽不足导致的对其他业务的影响。

更多信息，请参考用户文档。

- 新增支持下推以下函数到 TiFlash #50975 #50485 @yibin87 @windtalker
  - CAST(DECIMAL AS DOUBLE)
  - POWER()

  更多信息，请参考用户文档。

- TiDB 的并发 HashAgg 算法支持数据落盘（实验特性）#35637 @xzhangxian1008

  在之前的 TiDB 版本中，HashAgg 算子的并发算法不支持数据落盘。当 SQL 语句的执行计划包含并发的 HashAgg 算子时，该 SQL 语句的所有数据都只能在内存中进行处理。这导致内存需要处理大量数据，当超过内存限制时，TiDB 只能选择非并发 HashAgg 算法，无法通过并发提升性能。

  在 v8.0.0 中，TiDB 的并发 HashAgg 算法支持数据落盘。在任意并发条件下，HashAgg 算子都可以根据内存使用情况自动触发数据落盘，从而兼顾性能和数据处理量。目前，该功能作为实验特性，引入变量 tidb_enable_concurrent_hashagg_spill 控制是否启用支持落盘的并发 HashAgg 算法。当该变量为 ON 时，代表启用。该变量将在功能正式发布后废弃。

  更多信息，请参考用户文档。

- 自动统计信息收集引入优先级队列 #50132 @hi-rustin

维持优化器统计信息的时效性是稳定数据库性能的关键，绝大多数用户依赖 TiDB 提供的自动统计信息收集来保持统计信息的更新。自动统计信息收集轮询所有对象的统计信息状态，并把健康度不足的对象加入队列，逐个收集并更新。在之前的版本中，这些对象的收集顺序是随机的，可能导致更需要更新的对象等待时间过长，从而引发潜在的数据库性能回退。

从 v8.0.0 开始，自动统计信息收集引入了优先级队列，根据多种条件动态地为对象分配优先级，确保更有收集价值的对象优先被处理，比如新创建的索引、发生分区变更的分区表等。同时，TiDB 也会优先处理那些健康度较低的表，将它们安排在队列的前端。这一改进优化了收集顺序的合理性，能减少一部分统计信息过旧引发的性能问题，进而提升了数据库稳定性。

更多信息，请参考用户文档。

- 解除执行计划缓存的部分限制 #49161 @mjonss @qw4990

  TiDB 支持执行计划缓存，能够有效降低交易类业务系统的处理时延，是提升性能的重要手段。在 v8.0.0 中，TiDB 解除了执行计划缓存的几个限制，含有以下内容的执行计划均能够被缓存：

  - 分区表
  - 生成列，包含依赖生成列的对象（比如多值索引）

  该增强扩展了执行计划缓存的使用场景，提升了复杂场景下数据库的整体性能。

  更多信息，请参考用户文档。

- 优化器增强对多值索引的支持 #47759 #46539 @Arenatlx @time-and-fate

  TiDB 自 v6.6.0 开始引入多值索引，提升对 JSON 数据类型的检索性能。在 v8.0.0 中，优化器增强了对多值索引的支持能力，使其在复杂场景下能够正确识别和利用多值索引来优化查询。

  - 优化器能够收集多值索引的统计信息，并利用这些信息来估算查询。当一条 SQL 可能选择到数个多值索引时，优化器可以识别开销更小的索引。
  - 在查询条件中使用 OR 连接多个 member of 条件时，优化器能够为每个 DNF Item（member of 条件）匹配一个有效的 Index Partial Path 路径，并通过 Union 操作将这些路径集合起来，形成一个 Index Merge，以实现更高效的条件过滤和数据读取。

  更多信息，请参考用户文档。

- 支持设置低精度 TSO 的更新间隔 #51081 @Tema

  TiDB 的低精度 TSO 功能使用定期更新的 TSO 作为事务时间戳。在可以容忍读到旧数据的情况下，该功能通过牺牲一定的实时性，降低小的只读事务获取 TSO 的开销，从而提升高并发读的能力。

  在 v8.0.0 之前，低精度 TSO 功能的 TSO 更新周期固定，无法根据实际业务需要进行调整。在 v8.0.0 中，TiDB 引入变量 tidb_low_resolution_tso_update_interval 来控制低精度 TSO 功能更新 TSO 的周期。该功能仅在低精度 TSO 功能启用时有效。

  更多信息，请参考用户文档。

2.1.3 稳定性

- 支持根据 LRU 算法缓存所需的 schema 信息，以减少 TiDB server 的内存消耗（实验特性）#50959 @gmhdbjd

  在 v8.0.0 之前，每个 TiDB 节点都会缓存所有表的 schema 信息。在表数量较多的情况下，例如高达几十万张表，仅缓存这些表的 schema 信息就会占用大量内存。

  从 v8.0.0 开始，TiDB 引入 tidb_schema_cache_size 系统变量，允许设置缓存 schema 信息所能使用的内存上限，从而避免占用过多的内存。开启该

功能后，TiDB 将使用 Least Recently Used (LRU) 算法缓存所需的表，有效降低 schema 信息占用的内存。

更多信息，请参考用户文档。

2.1.4 高可用

- 代理组件 TiProxy 成为正式功能 (GA) #413 @djshow832 @xhebox

  TiDB v7.6.0 引入了代理组件 TiProxy 作为实验特性。TiProxy 是 TiDB 的官方代理组件，位于客户端和 TiDB server 之间，为 TiDB 提供负载均衡、连接保持功能，让 TiDB 集群的负载更加均衡，并在维护操作期间不影响用户对数据库的连接访问。

  在 v8.0.0 中，TiProxy 成为正式功能，完善了签名证书自动生成、监控等功能。

  TiProxy 的应用场景如下：

  - 在 TiDB 集群进行滚动重启、滚动升级、缩容等维护操作时，TiDB server 会发生变动，导致客户端与发生变化的 TiDB server 的连接中断。通过使用 TiProxy，可以在这些维护操作过程中平滑地将连接迁移至其他 TiDB server，从而让客户端不受影响。
  - 所有客户端对 TiDB server 的连接都无法动态迁移至其他 TiDB server。当多个 TiDB server 的负载不均衡时，可能出现整体集群资源充足，但某些 TiDB server 资源耗尽导致延迟大幅度增加的情况。为解决此问题，TiProxy 提供连接动态迁移功能，在客户端无感的前提下，将连接从一个 TiDB server 迁移至其他 TiDB server，从而实现 TiDB 集群的负载均衡。

  TiProxy 已集成至 TiUP、TiDB Operator、TiDB Dashboard 等 TiDB 基本组件中，可以方便地进行配置、部署和运维。

  更多信息，请参考用户文档。

2.1.5 SQL 功能

- 支持处理大量数据的 DML 类型（实验特性）#50215 @ekexium

  在 TiDB v8.0.0 之前，所有事务数据在提交之前均存储在内存中。当处理大量数据时，事务所需的内存成为限制 TiDB 处理事务大小的瓶颈。虽然 TiDB 非事务 DML 功能通过拆分 SQL 语句的方式尝试解决事务大小限制，但该功能存在多种限制，在实际应用中的体验并不理想。

  从 v8.0.0 开始，TiDB 支持处理大量数据的 DML 类型。该 DML 类型在执行过程中将数据及时写入 TiKV，避免将所有事务数据持续存储在内存中，从而支持处理超过内存限制的大量数据。这种 DML 类型在保证事务完整性的同时，采用与标准 DML 相同的语法。INSERT、UPDATE、REPLACE 和 DELETE 语句均可使用这种新的 DML 类型来执行大数据量的 DML 操作。

  支持处理大量数据的 DML 类型依赖于 Pipelined DML 特性，仅支持在自动提交的事务中使用。你可以通过 tidb_dml_type 系统变量控制是否启用该 DML 类型。

  更多信息，请参考用户文档。

- 支持在 TiDB 建表时使用更多的表达式设置列的默认值（实验特性）#50936 @zimulala

  在 v8.0.0 之前，建表时指定列的默认值仅限于固定的字符串、数字和日期。从 v8.0.0 开始，TiDB 支持使用部分表达式作为列的默认值，例如将列的默认值设置为 UUID()，从而满足多样化的业务需求。

  更多信息，请参考用户文档。

- 支持系统变量 div_precision_increment #51501 @yibin87

  MySQL 8.0 增加了变量 div_precision_increment，用于指定除法 / 运算结果增加的小数位数。在 v8.0.0 之前，TiDB 不支持该变量，而是按照 4 位小数

进行除法计算。从 v8.0.0 开始，TiDB 支持该变量，你可以根据需要指定除法运算结果增加的小数位数。

更多信息，请参考用户文档。

2.1.6 数据库管理

- PITR 支持 Amazon S3 对象锁定 #51184 @RidRisR

  Amazon S3 对象锁定功能支持用户通过设置数据留存期，有效防止备份数据在指定时间内被意外或故意删除，提升了数据的安全性和完整性。从 v6.3.0 起，BR 为快照备份引入了对 Amazon S3 对象锁定功能的支持，为全量备份增加了额外的安全性保障。从 v8.0.0 起，PITR 也引入了对 Amazon S3 对象锁定功能的支持，无论是全量备份还是日志数据备份，都可以通过对象锁定功能提供更可靠的数据保护，进一步加强了数据备份和恢复的安全性，并满足了监管方面的需求。

  更多信息，请参考用户文档。

- 支持在会话级将不可见索引 (Invisible Indexes) 调整为可见 #50653 @hawkingrei

  在优化器选择索引时，默认情况下不会选择不可见索引。这一机制通常用于在评估是否删除某个索引之前。如果担心删除索引可能导致性能下降，可以先将索引设置为不可见，以便在必要时快速将其恢复为可见。

  从 v8.0.0 开始，你可以将会话级系统变量 tidb_opt_use_invisible_indexes 设置为 ON，让当前会话识别并使用不可见索引。利用这个功能，在添加新索引并希望测试其效果时，可以先将索引创建为不可见索引，然后通过修改该系统变量在当前会话中测试新索引的性能，而不影响其他会话。这一改进提高了性能调优的安全性，并有助于增强生产数据库的稳定性。

  更多信息，请参考用户文档。

- 支持将 general log 写入独立文件 #51248 @Defined2014

  general log 是与 MySQL 兼容的功能，开启后能够记录数据库执行的所有 SQL 语句，为问题诊断提供依据。TiDB 也支持此功能，你可以通过设置变量 tidb_general_log 开启该功能。在之前的版本中，general log 的内容只能和其他信息一起写入实例日志中，这对于需要长期保存日志的用户来说并不方便。

  从 v8.0.0 开始，你可以通过配置项 log.general-log-file 指定一个文件名，将 general log 单独写入该文件。和实例日志一样，general log 也遵循日志的轮询和保存策略。

  另外，为了减少历史日志文件所占用的磁盘空间，TiDB 在 v8.0.0 支持了原生的日志压缩选项。你可以将配置项 log.file.compression 设置为 gzip，使得轮询出的历史日志自动以 gzip 格式压缩。

  更多信息，请参考用户文档。

2.1.7 可观测性

- 支持观测索引使用情况 #49830 @YangKeao

  正确的索引设计是提升数据库性能的重要前提。TiDB v8.0.0 新增内存表 INFORMATION_SCHEMA.TIDB_INDEX_USAGE，用于记录当前 TiDB 节点中所有索引的访问统计信息，包括：

  - 扫描该索引的语句的累计执行次数
  - 访问该索引时扫描的总行数
  - 扫描索引时的选择率分布
  - 最近一次访问该索引的时间

  通过这些信息，你可以识别未被优化器使用的索引以及过滤效果不佳的索引，从而优化索引设计，提升数据库性能。

此外，TiDB v8.0.0 新增与 MySQL 兼容的视图 sys.schema_unused_indexes，用于记录自 TiDB 上次启动以来未被使用的索引信息。对于从 v8.0.0 之前版本升级的集群，sys 中的内容不会自动创建。你可以参考 sys.schema_unused_indexes 手动创建。

更多信息，请参考用户文档。

2.1.8 安全

- TiKV 静态加密支持 Google Key Management Service (Cloud KMS)（实验特性）#8906 @glorv

  TiKV 通过静态加密功能对存储的数据进行加密，以确保数据的安全性。静态加密的安全核心点在于密钥管理。从 v8.0.0 起，你可以通过 Google Cloud KMS 管理 TiKV 的主密钥，构建基于 Cloud KMS 的静态加密能力，从而提高用户数据的安全性。

  要启用基于 Google Cloud KMS 的静态加密，你需要在 Google Cloud 上创建一个密钥，然后在 TiKV 配置文件中添加 [security.encryption.master-key] 部分的配置。

  更多信息，请参考用户文档。

- 增强 TiDB 日志脱敏 #51306 @xhebox

  TiDB 日志脱敏增强是通过对日志文件中的 SQL 文本信息进行标记，支持在查看时安全展示敏感数据。你可以控制是否对日志信息进行脱敏，以实现在不同场景下安全使用 TiDB 日志，提升了使用日志脱敏能力的安全性和灵活性。要使用此功能，可以将系统变量 tidb_redact_log 的值设置为 MARKER，此时 TiDB 的运行日志中的 SQL 文本会被标记，查看时将基于标记进行数据的安全展示，从而保护日志信息。

  更多信息，请参考用户文档。

2.1.9 数据迁移

- TiCDC 支持 Simple 协议 #9898 @3AceShowHand

  TiCDC 支持了新的 Simple 消息协议，该协议通过在 DDL 和 BOOTSTRAP 事件中嵌入表的 schema 信息，实现了对 schema 信息的动态追踪 (in-band schema tracking)。

  更多信息，请参考用户文档。

- TiCDC 支持 Debezium 协议 #1799 @breezewish

  通过 Debezium 协议，TiCDC 可以生成 Debezium 格式的数据变更事件，并将这些事件发送到 Kafka sink。这有助于为当前使用 Debezium 从 MySQL 拉取数据进行下游处理的用户简化从 MySQL 迁移到 TiDB 的过程。

  更多信息，请参考用户文档。

- DM 支持使用用户提供的密钥对源数据库和目标数据库的密码进行加密和解密 #9492 @D3Hunter

  在之前的版本中，DM 使用了一个内置的固定秘钥，安全性相对较低。从 v8.0.0 开始，你可以上传并指定一个密钥文件，用于对上下游数据库的密码进行加密和解密操作。此外，你还可以按需替换秘钥文件，以提升数据的安全性。

  更多信息，请参考用户文档。

- 支持 IMPORT INTO ... FROM SELECT 语法（实验特性），增强 IMPORT INTO 功能 #49883 @D3Hunter

  在之前的 TiDB 版本中，将查询结果导入目标表只能通过 INSERT INTO ... SELECT 语句，但该语句在一些大数据量的场景中的导入效率较低。从

v8.0.0 开始，TiDB 新增支持通过 IMPORT INTO ... FROM SELECT 将 SELECT 的查询结果导入到一张空的 TiDB 目标表中，其性能最高可达 INSERT INTO ... SELECT 的 8 倍，可以大幅缩短导入所需的时间。

此外，你还可以通过 IMPORT INTO ... FROM SELECT 导入使用 AS OF TIMESTAMP 查询的历史数据。

更多信息，请参考用户文档。

- TiDB Lightning 简化冲突处理策略，同时支持以 replace 方式处理冲突数据（实验特性）#51036 @lyzx2001

  在之前的版本中，TiDB Lightning 逻辑导入模式有一套数据冲突处理策略，而物理导入模式有两套数据冲突处理策略，不易理解和配置。

  从 v8.0.0 开始，TiDB Lightning 废弃了物理导入模式下的旧版冲突检测策略，支持通过 conflict.strategy 参数统一控制逻辑导入和物理导入模式的冲突检测策略，并简化了该参数的配置。此外，在物理导入模式下，当导入遇到主键或唯一键冲突的数据时，replace 策略支持保留最新的数据、覆盖旧的数据。

  更多信息，请参考用户文档。

- 全局排序成为正式功能 (GA)，可显著提升 IMPORT INTO 任务的导入性能和稳定性 #45719 @lance6716

  在 v7.4.0 以前，当使用分布式执行框架执行 IMPORT INTO 任务时，由于本地存储空间有限，TiDB 只能对部分数据进行局部排序后再导入到 TiKV。这导致了导入到 TiKV 的数据存在较多的重叠，需要 TiKV 在导入过程中执行额外的 compaction 操作，影响了 TiKV 的性能和稳定性。

  随着 v7.4.0 引入全局排序实验特性，TiDB 支持将需要导入的数据暂时存储在外部存储（如 Amazon S3）中进行全局排序后再导入到 TiKV 中，使

TiKV 无需在导入过程中执行 compaction 操作。全局排序在 v8.0.0 成为正式功能 (GA)，可以降低 TiKV 对资源的额外消耗，显著提升 IMPORT INTO 的性能和稳定性。启用全局排序后，单个 IMPORT INTO 任务支持导入 40 TiB 以内的数据。

更多信息，请参考用户文档。

2.2 兼容性变更

注意：

以下为从 v7.6.0 升级至当前版本 (v8.0.0) 所需兼容性变更信息。如果从 v7.5.0 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。

- 移除未 GA 但默认启用的 witness 相关调度器 #7765 @rleungx

2.2.1 行为变更

- 在安全增强模式 (SEM) 下禁止设置 require_secure_transport 为 ON，避免用户无法连接的问题 #47665 @tiancaiamao
- DM 移除了固定的加解密 key，并支持设置自定义加解密 key。如果升级前数据源配置和迁移任务配置里使用了加密密码，需参考 DM 自定义加解密 key 中的升级步骤进行额外操作。#9492 @D3Hunter
- 在之前版本中，启用添加索引加速功能 (tidb_ddl_enable_fast_reorg = ON) 后，编码后的索引键值 ingest 到 TiKV 的过程使用了固定的并发数 (16)，并未根据下游 TiKV 的处理能力进行动态调整。从 v8.0.0 开始，支持使用 tidb_ddl_reorg_worker_cnt 设置并发数。该变量默认值为 4，相比之前的默认值 16，在 ingest 索引键值对时性能可能会有所下降。你可以根据集群的负载按需调整该参数。

2.2.2 MySQL 兼容性

- KEY 分区类型支持分区字段列表为空的语句，具体行为和 MySQL 保持一致。

2.2.3 系统变量

变量名	修改类型	描述
tidb_disable_txn_auto_retry	废弃	从 v8.0.0 开始，该系统变量被废弃，TiDB 不再支持乐观事务的自动重试。推荐使用悲观事务模式。如果使用乐观事务模式发生冲突，请在应用里捕获错误并重试。
tidb_ddl_version	更名	用于控制是否开启 TiDB DDL V2。为了使变量名称更直观，从 v8.0.0 起，该变量更名为 tidb_enable_fast_create_table。

变量名	修改类型	描述
tidb_enable_collect_execution_info	修改	增加控制是否维护访问索引有关的统计信息，默认值为 ON。
tidb_redact_log	修改	控制在记录 TiDB 日志和慢日志时如何处理 SAL 文本中的用户信息，可选值为 OFF（对用户输入的信息不做任何处理）和 ON（屏蔽日志中的用户信息）。为了提供更丰富的处理日志中用户信息的方式，v8.0.0 中增加了 MARKER

变量名	修改类型	描述
		选项，支持标记日志信息。
div_precision_increment	新增	用于指定使用运算符 / 执行除法操作时，结果增加的小数位数。该功能与 MySQL 保持一致。
tidb_dml_type	新增	设置 DML 语句的执行方式，可选值为 "standard" 和 "bulk"。
tidb_enable_auto_analyze_priority_queue	新增	控制是否启用优先队列来调度自动收集统计信息的任务。开启该变量后，TiDB 会优先收集那些最需要收集

变量名	修改类型	描述
		统计信息的表的统计信息。
tidb_enable_concurrent_hashagg_spill	新增	控制 TiDB 是否支持并发 HashAgg 进行落盘。当该变量设置为 ON 时，并发 HashAgg 将支持落盘。该变量将在功能正式发布时废弃。
tidb_enable_fast_create_table	新增	用于控制是否开启 TiDB 加速建表。将该变量的值设置为 ON 可以开启该功能，设置为 OFF 关闭该功能。默认值为 OFF。开

变量名	修改类型	描述
		启后，将使用 CREATE TABLE 加速建表。
tidb_load_binding_timeout	新增	控制加载绑定的超时时间。当加载绑定的执行时间超过该值时，会停止加载。
tidb_low_resolution_tso_update_interval	新增	设置 TiDB 缓存 timestamp 的更新时间间隔。
tidb_opt_ordering_index_selectivity_ratio	新增	当一个索引满足 SQL 语句中的 ORDER BY 和 LIMIT 子句，但有部分过滤条件未被该索引覆盖时，该系统变量

变量名	修改类型	描述
		用于控制该索引的估算行数。默认值为 -1，表示禁用此系统变量。
tidb_opt_use_invisible_indexes	新增	控制当前会话中是否允许优化器选择不可见索引。当修改变量为 ON 时，对该会话中的查询，优化器可以选择不可见索引进行查询优化。
tidb_schema_cache_size	新增	设置缓存 schema 信息可以使用的内存上限，避免占用过多的内存。开启该功能

变量名	修改类型	描述
		后，将使用 LRU 算法来缓存所需的表，有效减少 schema 信息占用的内存。

2.2.4 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	instance.tidb_enable_collect_execution_info	修改	增加控制是否维护访问索引有关的统计信息，默认值为 true。
TiDB	tls-version	修改	该参数不再支持 "TLSv1.0" 和 "TLSv1.1"，只支持 "TLSv1.2" 和 "TLSv1.3"。
TiDB	log.file.compression	新增	指定轮询日志的压缩格式。默认为空，即不压缩轮询日志。
TiDB	log.general-log-file	新增	指定 general log 的保存文件。默认为空，general log 将会写入实例文件。
TiDB	tikv-client.enable-replica-selector-v2	新增	控制给 TiKV 发送 RPC 请求时，是否使用新版本的 Region 副本选择器。默认值为 true。

配置文件	配置项	修改类型	描述
TiKV	log-backup.initial-scan-rate-limit	修改	增加了最小值为 1MiB 的限制。
TiKV	raftstore.store-io-pool-size	修改	为了提升 TiKV 性能，该参数默认值从 0 修改为 1，表示 StoreWriter 线程池的大小默认为 1。
TiKV	rocksdb.defaultcf.titan.blob-cache-size	修改	从 v8.0.0 开始，TiKV 引入了 shared-blob-cache 配置项并默认开启，因此无需再单独设置 blob-cache-size。只有当 shared-blob-cache 设置为 false 时，blob-cache-size 的设置才生效。
TiKV	security.encryption.master-key.vendor	修改	主密钥可选的服务商类型新增 gcp。
TiKV	rocksdb.defaultcf.titan.shared-blob-cache	新增	控制是否启用 Titan Blob 文件和 RocksDB Block 文件的共享缓存。默认值为 true。
TiKV	security.encryption.master-key.gcp.credential-file-path	新增	在 security.encryption.master-key.vendor 为 gcp 时，用于指定 Google Cloud 认证凭证文件的路径。

配置文件	配置项	修改类型	描述
TiDB Lightning	tikv-importer.duplicate-resolution	废弃	用于在物理导入模式下设置是否检测和解决唯一键冲突的记录。从 v8.0.0 开始被参数 conflict.strategy 替代。
TiDB Lightning	conflict.precheck-conflict-before-import	新增	控制是否开启前置冲突检测，即导入数据到 TiDB 前，先检查所需导入的数据是否存在冲突。该参数默认值为 false，表示仅开启后置冲突检测。仅当导入模式为物理导入模式 (tikv-importer.backend = "local") 时可以使用该参数。
TiDB Lightning	logical-import-batch-rows	新增	在逻辑导入模式下，用于限制每个事务中可插入的最大行数，默认值为 65536。
TiDB Lightning	logical-import-batch-size	新增	在逻辑导入模式下，用于设置下游 TiDB 服务器上执行的每条 SQL 语句的最大值。默认值为 "96KiB"，单位可以为 KB、KiB、MB、MiB 等存储单位。

配置文件	配置项	修改类型	描述
Data Migration	secret-key-path	新增	用于指定加解密上下游密码的密钥文件所在的路径。该文件内容必须是长度为 64 个字符的十六进制的 AES-256 密钥。
TiCDC	tls-certificate-file	新增	用于指定 Pulsar 启用 TLS 加密传输时，客户端的加密证书文件路径。
TiCDC	tls-key-file-path	新增	用于指定 Pulsar 启用 TLS 加密传输时，客户端的加密私钥路径。

2.2.5 系统表

- 新增系统表 INFORMATION_SCHEMA.TIDB_INDEX_USAGE 和 INFORMATION_SCHEMA.CLUSTER_TIDB_INDEX_USAGE 用于记录 TiDB 节点中索引的访问统计信息。
- 新增系统数据库 sys 和 sys.schema_unused_indexes 视图，用于记录自 TiDB 上次启动以来未被使用的索引信息。

2.3 废弃功能

- 从 v8.0.0 开始，tidb_disable_txn_auto_retry 变量被废弃。废弃后，TiDB 不再支持乐观事务的自动重试。作为替代，当使用乐观事务模式发生冲突时，请在应用里捕获错误并重试，或改用悲观事务模式。
- 从 v8.0.0 开始，TiDB 不再支持 TLSv1.0 和 TLSv1.1 协议。请升级 TLS 至 TLSv1.2 或 TLSv1.3。
- 从 v8.0.0 开始，TiDB Lightning 废弃了物理导入模式下的旧版冲突检测策略，支持通过 conflict.strategy 参数统一控制逻辑导入和物理导入模式的冲

突检测策略。旧版冲突检测的参数 duplicate-resolution 将在未来版本中被移除。

- 计划在后续版本重新设计执行计划绑定的自动演进，相关的变量和行为会发生变化。

2.4 改进提升

- TiDB
  - DDL 创建表语句 CREATE TABLE 执行性能加速 10 倍，并且可线性扩展 #50052 @GMHDBJD
  - 支持同时提交 16 个 IMPORT INTO ... FROM FILE 任务，方便批量导入数据到目标表，极大地提升了数据文件导入的效率和性能 #49008 @D3Hunter
  - 提升 Sort 算子的数据落盘性能 #47733 @xzhangxian1008
  - 优化数据落盘功能的退出机制，支持在数据落盘过程中取消查询 #50511 @wshwsh12
  - 在处理包含多个等值条件的表连接查询时，支持使用匹配部分条件的索引构造 Index Join #47233 @winoros
  - 增强 Index Merge 能力，使其能识别查询中的排序需求，并能选中满足排序要求的索引 #48359 @AilinKid
  - 当 Apply 算子没有并发执行时，支持通过执行 SHOW WARNINGS 查看阻碍并发的算子名 #50256 @hawkingrei
  - 优化点查的索引选择，在所有的索引都支持点查时选择其中性能最优的一个用于查询 #50184 @elsa0520
  - 将统计信息同步加载任务的优先级暂时调整为 High，避免在 TiKV 高负载时同步加载任务大面积超时，从而导致统计信息无法加载 #50332 @winoros
  - 在 PREPARE 语句无法命中执行计划缓存时，支持通过执行 SHOW WARNINGS 查看原因 #50407 @hawkingrei

- 提升当多次更新同一行的数据时查询估算信息的准确性 #47523 @terry1purcell
- Index Merge 支持在 AND 谓词中内嵌多值索引和 OR 操作符 #51778 @time-and-fate
- 当设置 force-init-stats 为 true 时，即 TiDB 启动时等待统计信息初始化完成后再对外提供服务，这一设置不再影响 HTTP server 提供服务，用户仍可查看监控 #50854 @hawkingrei
- 支持 MemoryTracker 追踪 IndexLookup 算子的内存使用情况 #45901 @solotzg
- 支持 MemoryTracker 追踪 MemTableReaderExec 算子的内存使用情况 #51456 @wshwsh12
- 支持从 PD 批量加载 Region，加快在对大表进行查询时，从 KV Range 到 Regions 的转换过程 #51326 @SeaRise
- 优化系统表 INFORMATION_SCHEMA.TABLES、INFORMATION_SCHEMA.STATISTICS、INFORMATION_SCHEMA.KEY_COLUMN_USAGE、INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS 的查询性能。相比之前版本，性能提升最高可达 100 倍 #50305 @ywqzzy

- TiKV
  - 增强 TSO 校验检测，提升配置或操作不当时集群 TSO 的鲁棒性 #16545 @cfzjywxk
  - 优化清理悲观锁的逻辑，提高未提交事务的处理性能 #16158 @cfzjywxk
  - 增加 TiKV 统一健康控制，降低单个 TiKV 节点异常对集群访问性能的影响。可通过 tikv-client.enable-replica-selector-v2 禁用该优化 #16297 #1104 #1167 @MyonKeminta @zyguan @crazycs520
  - PD client 使用元数据存储接口代替原有的全局配置接口 #14484 @HuSharp

- 通过 write cf stats 决定数据加载行为，以提升扫描性能 #16245 @Connor1996
- 在 Raft conf change 过程中，增加了检查删除节点和 Voter 降级的最近一次心跳，确保此行为不会导致该 Region 不可访问 #15799 @tonyxuqqi
- 为 Pipelined DML 增加 Flush 和 BufferBatchGet 接口 #16291 @ekexium
- 增加 cgroup CPU 和内存限制的监控 #16392 @pingandb
- 增加 Region worker 和快照生成 worker 的 CPU 监控 #16562 @Connor1996
- 增加 peer 和 store 消息的 slow log #16600 @Connor1996

- PD
  - 增强 PD 客户端的服务发现能力，提升其高可用性和负载平衡 #7576 @CabinfeverB
  - 增强 PD 客户端的重试机制 #7673 @JmPotato
  - 增加 cgroup CPU 和内存的监控和告警 #7716 #7918 @pingandb @rleungx
  - 提升使用 etcd watch 的性能和高可用性 #7738 #7724 #7689 @lhy1024
  - 增加更多心跳监控，以便更好地分析性能瓶颈 #7868 @nolouch
  - 减少 etcd leader 对 PD leader 的影响 #7499 @JmPotato @HuSharp
  - 增强对不健康的 etcd 节点的检测机制 #7730 @JmPotato @HuSharp
  - 优化 pd-ctl 中 GC safepoint 的相关显示 #7767 @nolouch
  - 支持动态修改热点调度器中的历史窗口配置 #7877 @lhy1024
  - 减少创建 operator 中的锁争用问题 #7837 @Leavrth
  - 调整 GRPC 配置以提升可用性 #7821 @rleungx
- TiFlash

- 支持 JSON_EXTRACT() 函数中的 json_path 参数为非常量 #8510 @SeaRise
- 支持 JSON_LENGTH(json, path) 函数 #8711 @SeaRise

- Tools
  - Backup & Restore (BR)
    - 支持通过 br 命令行工具新增的恢复参数 --load-stats 控制是否恢复统计信息 #50568 @Leavrth
    - 支持通过 br 命令行工具新增的恢复参数 --tikv-max-restore-concurrency 控制每个 TiKV 节点的最大 download 和 ingest 文件数量，并通过控制作业队列的最大长度，进而控制 BR 节点的内存消耗 #51621 @3pointer
    - 粗粒度打散 Region 算法支持自适应获取并发参数，提升恢复性能 #50701 @3pointer
    - 在 br 的命令行帮助信息中显示 log 命令 #50927 @RidRisR
    - 支持在恢复过程中提前分配好 Table ID，从而最大限度地复用 Table ID，提升恢复性能 #51736 @Leavrth
    - 使用 BR 时，禁用 TiDB 内部的 GC memory limit tuner 功能，避免 OOM 问题 #51078 @Leavrth
    - 使用更优的算法，提升数据恢复过程中 SST 文件合并的速度 #50613 @Leavrth
    - 支持在数据恢复过程中批量创建数据库 #50767 @Leavrth
    - 在日志备份过程中，增加了在日志和监控指标中打印影响 global checkpoint 推进的最慢的 Region 的信息 #51046 @YuJuncen
    - 提升了 RESTORE 语句在大数据量表场景下的建表性能 #48301 @Leavrth
  - TiCDC

    - 优化 RowChangedEvent 的内存占用，降低 TiCDC 同步数据时的内存消耗 #10386 @lidezhu
    - 增加在创建和恢复 changefeed 任务时验证 start-ts 参数是否合法 #10499 @3AceShowHand
  - TiDB Data Migration (DM)
    - 在 MariaDB 主从复制的场景中，即 MariaDB 主实例 -> MariaDB 从实例 -> DM -> TiDB 的迁移场景，当 gtid_strict_mode = off 且 MariaDB 从实例的 GTID 不严格递增时（例如，有业务数据写入 MariaDB 从实例），此时 DM 任务会报错 less than global checkpoint position。从 v8.0.0 开始，TiDB 兼容该场景，数据可以正常迁移到下游。#10741 @okJiang
  - TiDB Lightning
    - 在逻辑导入模式下，支持使用 logical-import-batch-rows 配置批处理的最大行数 #46607 @kennytm
    - 当 TiFlash 的导入空间不足时，TiDB Lightning 会报错 #50324 @okJiang

2.5 错误修复

- TiDB
  - 修复在无数据变更的情况下，auto analyze 被多次触发的问题 #51775 @hi-rustin
  - 修复 auto analyze 并发设置错误的问题 #51749 @hawkingrei
  - 修复使用单个 SQL 语句添加多个索引导致的索引不一致问题 #51746 @tangenta
  - 修复查询使用 NATURAL JOIN 时可能报错 Column ... in from clause is ambiguous 的问题 #32044 @AilinKid
  - 修复 TiDB 错误地消除 group by 中的常量值导致查询结果出错的问题 #38756 @hi-rustin

- 修复 LEADING hint 在 UNION ALL 语句中无法生效的问题 #50067 @hawkingrei
- 修复 BIT 类型的列在参与一些函数计算时，可能会因为 decode 失败导致查询出错的问题 #49566 #50850 #50855 @jiyfhust
- 修复通过 tiup cluster upgrade/start 方式进行滚动升级时，与 PD 交互出现问题可能导致 TiDB panic 的问题 #50152 @zimulala
- 修复执行包含 ORDER BY 的 UNIQUE 索引点查时可能报错的问题 #49920 @jackysp
- 修复常量传播在处理 ENUM 或 SET 类型时结果出错的问题 #49440 @winoros
- 修复包含 Apply 操作的查询在报错 fatal error: concurrent map writes 后导致 TiDB 崩溃的问题 #50347 @SeaRise
- 修复使用 SET_VAR 控制字符串类型的变量可能会失效的问题 #50507 @qw4990
- 修复当 tidb_sysdate_is_now 设置为 1 时，SYSDATE() 函数错误地使用了计划缓存中的时间的问题 #49299 @hawkingrei
- 修复执行 CREATE GLOBAL BINDING 语句时，如果数据库名为大写，则绑定不生效的问题 #50646 @qw4990
- 修复 Index Path 选中重复索引的问题 #50496 @AilinKid
- 修复当 CREATE GLOBAL BINDING 语句中包含 IN() 时，PLAN REPLAYER 无法加载绑定的问题 #43192 @King-Dylan
- 修复当多个 analyze 任务失败时，没有正确记录失败原因的问题 #50481 @hi-rustin
- 修复系统变量 tidb_stats_load_sync_wait 设置不生效的问题 #50872 @jiyfhust
- 修复 max_execute_time 多层设置相互影响的问题 #50914 @jiyfhust
- 修复并发更新统计信息导致的线程安全问题 #50835 @hi-rustin
- 修复对分区表执行 auto analyze 可能导致 TiDB panic 的问题 #51187 @hi-rustin

- 修复 SQL 语句中 IN() 谓词包含的值的个数不同时，可能导致 SQL 绑定不生效的问题 #51222 @hawkingrei
- 修复 TiDB 无法正确转换表达式中系统变量类型的问题 #43527 @hi-rustin
- 修复在配置 force-init-stats 的情况下，TiDB 没有监听对应端口的问题 #51473 @hawkingrei
- 修复在 determinate 模式下 (tidb_opt_objective='determinate')，如果查询不包含谓词，可能无法加载统计信息的问题 #48257 @time-and-fate
- 修复 init-stats 流程可能导致 TiDB panic 以及 load stats 流程直接退出的问题 #51581 @hawkingrei
- 修复 IN() 谓词中包含 NULL 时，查询结果错误的问题 #51560 @winoros
- 修复当 DDL 任务中包含多张表时，MDL View 中不显示 blocked DDL 的问题 #47743 @wjhuang2016
- 修复表的 ANALYZE 任务统计的 processed_rows 可能超过表的总行数的问题 #50632 @hawkingrei
- 修复当 HashJoin 算子落盘失败时 goroutine 可能泄露的问题 #50841 @wshwsh12
- 修复 CTE 查询使用的内存超限时可能会导致 goroutine 泄露的问题 #50337 @guo-shaoge
- 修复使用聚合函数分组计算时可能报错 Can't find column ... 的问题 #50926 @qw4990
- 修复当 CREATE TABLE 语句中包含特定分区或约束的表达式时，表名变更等 DDL 操作会卡住的问题 #50972 @lcwangchao
- 修复 Grafana 监控指标 tidb_statistics_auto_analyze_total 没有显示为整数的问题 #51051 @hawkingrei

- 修复修改变量 tidb_server_memory_limit 后，tidb_gogc_tuner_threshold 未进行相应调整的问题 #48180 @hawkingrei
- 修复当查询语句涉及 JOIN 操作时可能出现 index out of range 报错的问题 #42588 @AilinKid
- 修复当列的默认值被删除时，获取该列的默认值会报错的问题 #50043 #51324 @crazycs520
- 修复 TiFlash 延迟物化在处理关联列时结果可能出错的问题 #49241 #51204 @Lloyd-Pottiger
- 修复 LIKE() 函数在处理 binary collation 的输入时可能结果错误的问题 #50393 @yibin87
- 修复 JSON_LENGTH() 函数在第二个参数为 NULL 时结果错误的问题 #50931 @SeaRise
- 修复 CAST(AS DATETIME) 在特定情况下可能会丢失时间精度的问题 #49555 @SeaRise
- 修复并行 Apply 在表为聚簇索引时可能导致结果错误的问题 #51372 @guo-shaoge
- 修复主键类型是 VARCHAR 时，执行 ALTER TABLE ... COMPACT TIFLASH REPLICA 可能会错误地提前结束的问题 #51810 @breezewish
- 修复 EXCHANGE PARTITION 语句交换分区表时，对 DEFAULT NULL 属性产生的 NULL 值检查有错误的问题 #47167 @jiyfhust
- 修复使用非 UTF8 字符集时，分区表定义可能导致错误行为的问题 #49251 @YangKeao
- 修复一些系统变量的默认值在 INFORMATION_SCHEMA.VARIABLES_INFO 表中显示错误的问题 #49461 @jiyfhust
- 修复一些情况下使用了空字符串作为数据库名但没有报错的问题 #45873 @yoshikipom

  - 修复 SPLIT TABLE ... INDEX 语句可能会导致 TiDB panic 的问题 #50177 @Defined2014
  - 修复查询 KeyPartition 类型的分区表可能会报错的问题 #50206 #51313 #51196 @time-and-fate @jiyfhust @mjonss
  - 修复查询 Hash 分区类型的分区表时，结果可能不正确的问题 #50427 @Defined2014
  - 修复 opentracing 不能正常工作的问题 #50508 @Defined2014
  - 修复 ALTER INSTANCE RELOAD TLS 报错时，错误信息不完整的问题 #50699 @dveeden
  - 修复 AUTO_INCREMENT 属性在分配自增 ID 时，由于不必要的事务冲突导致 ID 不连续的问题 #50819 @tiancaiamao
  - 修复 TiDB 日志中某些报错的栈信息不完整的问题 #50849 @tiancaiamao
  - 修复当 LIMIT 子句中的数字过大时，一些查询的内存使用过大的问题 #51188 @Defined2014
  - 修复 TTL 功能在某些情况下因为没有正确切分数据范围而造成数据热点的问题 #51527 @lcwangchao
  - 修复当 SET 语句出现在显式事务的第一行时不生效的问题 #51387 @YangKeao
  - 修复一些情况下查询 BINARY 类型的 JSON 可能会报错的问题 #51547 @YangKeao
  - 修复 TTL 在计算过期时间时，不能正确处理夏令时跳变的问题 #51675 @lcwangchao
  - 修复某些情况下 SHOW CREATE PLACEMENT POLICY 语句不显示 SURVIVAL_PREFERENCES 属性的问题 #51699 @lcwangchao
  - 修复当配置文件中出现不合规的配置项时，配置文件不生效的问题 #51399 @Defined2014
- TiKV
  - 修复开启 tidb_enable_row_level_checksum 可能导致 TiKV panic 的问题 #16371 @cfzjywxk

- 修复休眠的 Region 在异常情况下未被及时唤醒的问题 #16368 @LykxSassinator
- 通过在执行下线节点操作前检查该 Region 所有副本的上一次心跳时间，修复下线一个副本导致整个 Region 不可用的问题 #16465 @tonyxuqqi
- 修复 JSON 整型数值在大于 INT64 最大值但小于 UINT64 最大值时会被 TiKV 解析成 FLOAT64 导致结果和 TiDB 不一致的问题 #16512 @YangKeao
- 修复监控指标 tikv_unified_read_pool_thread_count 有时没有数据的问题 #16629 @YuJuncen

- PD
  - 修复调用 MergeLabels 函数时存在数据竞争的问题 #7535 @lhy1024
  - 修复调用 evict-leader-scheduler 接口时没有输出结果的问题 #7672 @CabinfeverB
  - 修复 PD 监控项 learner-peer-count 在发生 Leader 切换后未同步旧监控值的问题 #7728 @CabinfeverB
  - 修复 watch etcd 没有正确关闭导致内存泄露的问题 #7807 @rleungx
  - 修复 TSO 部分日志没有打印报错原因的问题 #7496 @CabinfeverB
  - 修复重启后 PD 部分监控出现非预期负数的问题 #4489 @lhy1024
  - 修复 Leader 租约的过期时间晚于日志时间的问题 #7700 @CabinfeverB
  - 修复在 TiDB（PD 客户端）和 PD 之间的 TLS 开关不一致时，TiDB panic 的问题 #7900 #7902 #7916 @CabinfeverB
  - 修复 Goroutine 由于没有正确关闭而泄露的问题 #7782 @HuSharp
  - 修复 pd-ctl 无法移除包含特殊字符的调度器的问题 #7798 @JmPotato
  - 修复 PD 客户端获取 TSO 可能被阻塞的问题 #7864 @CabinfeverB

- TiFlash
  - 修复副本迁移时，因 TiFlash 与 PD 之间网络连接不稳定可能引发的 TiFlash panic 的问题 #8323 @JaySon-Huang
  - 修复慢查询导致内存使用显著增加的问题 #8564 @JinheLin
  - 修复移除 TiFlash 副本后重新添加可能导致 TiFlash 数据损坏的问题 #8695 @JaySon-Huang
  - 修复在执行 PITR 恢复任务或 FLASHBACK CLUSTER TO 后，TiFlash 副本数据可能被意外删除，导致数据异常的问题 #8777 @JaySon-Huang
  - 修复在执行 ALTER TABLE ... MODIFY COLUMN ... NOT NULL 时，将原本可为空的列修改为不可为空之后，导致 TiFlash panic 的问题 #8419 @JaySon-Huang
  - 修复存算分离架构下，出现网络隔离后查询可能会被永久阻塞的问题 #8806 @JinheLin
  - 修复存算分离架构下，TiFlash 关闭过程中可能 panic 的问题 #8837 @JaySon-Huang
  - 修复 TiFlash 在发生远程读时可能会因为数据竞争导致 crash 的问题 #8685 @solotzg
  - 修复 CAST(AS JSON) 函数中没有对 JSON object key 去重的问题 #8712 @SeaRise
  - 修复 ENUM 列在 chunk encode 时可能会导致 TiFlash crash 的问题 #8674 @yibin87
- Tools
  - Backup & Restore (BR)
    - 修复在 Region 成为 Leader 后立刻分裂或合并，导致日志备份 Checkpoint 不推进的问题 #16469 @YuJuncen
    - 修复在某些极端情况下，全量备份因找不到 peer 导致 TiKV panic 的问题 #16394 @Leavrth

- 修复在同一节点上更改 TiKV IP 地址导致日志备份卡住的问题 #50445 @3pointer
- 修复从 S3 读文件内容时出错后无法重试的问题 #49942 @Leavrth
- 修复数据恢复失败后，使用断点重启报错 the target cluster is not fresh 的问题 #50232 @Leavrth
- 修复停止日志备份任务导致 TiDB crash 的问题 #50839 @YuJuncen
- 修复由于某个 TiKV 节点缺少 Leader 导致数据恢复变慢的问题 #50566 @Leavrth
- 修复全量恢复指定 --filter 选项后，仍然要求目标集群为空的问题 #51009 @3pointer

– TiCDC

- 修复使用 storage sink 时，在存储服务生成的文件序号可能出现回退的问题 #10352 @CharlesCheung96
- 修复并发创建多个 changefeed 时 TiCDC 返回 ErrChangeFeedAlreadyExists 错误的问题 #10430 @CharlesCheung96
- 修复在 ignore-event 中设置了过滤掉 add table partition 事件后，TiCDC 未将相关分区的其它类型 DML 变更事件同步到下游的问题 #10524 @CharlesCheung96
- 修复上游表执行了 TRUNCATE PARTITION 后 changefeed 报错的问题 #10522 @sdojjy
- 修复恢复 changefeed 时 changefeed 的 checkpoint-ts 小于 TiDB 的 GC safepoint，没有及时报错 snapshot lost caused by GC 的问题 #10463 @sdojjy
- 修复 TiCDC 在开启单行数据正确性校验后由于时区不匹配导致 TIMESTAMP 类型 checksum 验证失败的问题 #10573 @3AceShowHand

- 修复 Syncpoint 表可能被错误同步的问题 #10576 @asddongmen
- 修复当使用 Apache Pulsar 作为下游时，无法正常启用 OAuth2.0、TLS 和 mTLS 的问题 #10602 @asddongmen
- 修复当 TiKV 升级、重启或驱逐 Leader 时，changefeed 可能卡住的问题 #10584 @asddongmen
- 修复在频繁执行 DDL 的场景中，由于错误的 BarrierTS 导致数据被写入到错误的 CSV 文件的问题 #10668 @lidezhu
- 修复 KV Client 数据争用导致 TiCDC panic 的问题 #10718 @asddongmen
- 修复在调度表的同步任务时 TiCDC panic 的问题 #10613 @CharlesCheung96

– TiDB Data Migration (DM)

- 修复上游为 binary 类型主键时丢失数据的问题 #10672 @GMHDBJD

– TiDB Lightning

- 修复检查 TiKV 空间导致的性能回退的问题 #43636 @lance6716
- 修复在扫描数据文件时，遇到不合法符号链接文件而报错的问题 #49423 @lance6716
- 修复当 sql_mode 中不包含 NO_ZERO_IN_DATE 时，TiDB Lightning 无法正确解析包含 0 的日期值的问题 #50757 @GMHDBJD

2.6 贡献者

感谢来自 TiDB 社区的贡献者们：

- Aoang
- bufferflies
- daemon365
- eltociear

- lichunzhu
- jiyfhust
- pingandb
- shenqidebaozi
- Smityz
- songzhibin97
- tangjingyu97
- Tema
- ub-3
- yoshikipom

3 TiDB 7.6.0 Release Notes

发版日期：2024 年 1 月 25 日

TiDB 版本：7.6.0

试用链接：快速体验 | 下载离线包

在 7.6.0 版本中，你可以获得以下关键特性：

分类	功能/增强	描述
可扩展性与性能	跨数据库绑定执行计划	在处理上百个 schema 相同的数据库时，针对其中一个数据库的 SQL binding 通常也适用于其它的数据库。例如，在 SaaS 或 PaaS 数据平台中，每个用户通常各自维护单独的数据库，这些数据库具有相同的 schema 并运行着类似的 SQL。在这种情况下，逐一为每个数据库做 SQL 绑定是不切实际的。TiDB v7.6.0 引入跨数据库绑定执行计划，支持在所有 schema 相同的数据库之间匹配绑定计划。
	BR 快照恢复速度最高提升 10 倍（实验特性）	BR v7.6.0 实验性地引入了粗粒度打散 Region 算法，用于提升集群的快照恢复速度。在 TiKV 节点较多的集群中，该算法可显著提高集群资源利用率，更均匀地分配负载，同时更好地利用每个节点的网络带宽。在一些实际案例中，该特性可将恢复速度最高提升约 10 倍。

分类	功能/增强	描述
	建表性能提升 10 倍（实验特性）	在 v7.6.0 中引入了新的 DDL 架构，批量建表的性能提高了 10 倍。这一重大改进极大地缩短了创建大量表所需的时间。特别是在 SaaS 场景中，快速创建大量表（从数万到数十万不等）是一个常见的挑战，使用该特性能显著提升 SaaS 场景的建表速度。
	通过 Active PD Follower 提升 PD Region 信息查询服务的扩展能力（实验特性）	TiDB v7.6.0 实验性地引入了 Active PD Follower 特性，允许 PD follower 提供 Region 信息查询服务。在 TiDB 节点数量较多和 Region 数量较多的集群中，该特性可以提升 PD 集群处理 GetRegion、ScanRegions 请求的能力，减轻 PD leader 的 CPU 压力。
稳定性与高可用	支持 TiProxy（实验特性）	全面支持 TiProxy，可通过部署工具轻松部署。TiProxy 可以管理和维护客户端与 TiDB 的连接，在滚动重启、升级以及扩缩容过程中保持连接。
	Data Migration (DM) 正式支持迁移 MySQL 8.0 (GA)	在 v7.6.0 之前，DM 迁移 MySQL 8.0 仅为实验特性，不能用于生产环境。TiDB v7.6.0 增强了该功能的稳定性、兼容性，可在生产环境帮助你平滑、快速地将数据从 MySQL 8.0 迁移到 TiDB。在 v7.6.0 中，该功能正式 GA。

3.1 功能详情

3.1.1 可扩展性

- 通过 Active PD Follower 提升 PD 上 Region 信息查询服务的扩展能力（实验特性）#7431 @CabinfeverB

  当集群的 Region 数量较多时，PD leader 处理心跳和调度任务的开销也较大，可能导致 CPU 资源紧张。如果同时集群中的 TiDB 实例数量较多，查询 Region 信息请求并发量较大，PD leader CPU 压力将变得更大，可能会造成 PD 服务不可用。

  为确保服务的高可用性，TiDB v7.6.0 引入了 Active PD Follower 特性提升 PD 上 Region 信息查询服务的扩展能力。你可以通过设置系统变量 pd_enable_follower_handle_region 开启 Active PD Follower 特性。启用该

特性后，TiDB 在获取 Region 信息时会将请求均匀地发送到所有 PD 节点上，使 PD follower 也可以处理 Region 请求，从而减轻 PD leader 的 CPU 压力。

更多信息，请参考用户文档。

3.1.2 性能

- BR 快照恢复速度最高提升 10 倍（实验特性）#33937 #49886 @3pointer

  随着 TiDB 集群规模的不断扩大，故障时快速恢复集群以减少业务中断时间显得尤为重要。在 v7.6.0 之前的版本中，Region 打散算法是性能恢复的主要瓶颈。在 v7.6.0 中，BR 优化了 Region 打散算法，可以迅速将恢复任务拆分为大量小任务，并批量分散到所有 TiKV 节点上。新的并行恢复算法充分利用每个 TiKV 节点的所有资源，实现了并行快速恢复。在实际案例中，大规模 Region 场景下，集群快照恢复速度最高提升约 10 倍。

  目前，新的粗粒度 Region 打散算法为实验特性，你可以配置 br 新增的命令行参数 --granularity="coarse-grained" 启用新算法。例如：

  ```
  br restore full \
  --pd "${PDIP}:2379" \
  --storage "s3://${Bucket}/${Folder}" \
  --s3.region "${region}" \
  --granularity "coarse-grained" \
  --send-credentials-to-tikv=true \
  --log-file restorefull.log
  ```

  更多信息，请参考用户文档。

- 默认开启 Titan 引擎 #16245 @Connor1996 @v01dstar @tonyxuqqi

  为了更好地支持 TiDB 宽表写入场景，特别是在支持 JSON 之后，从 TiDB v7.6.0 开始，默认开启 Titan 引擎，自动将超过 32 KB 的大 Value 从 RocksDB 的 LSM Tree 中分离出来，单独存储在 Titan 中，以提升对大 Value 的处理性能。Titan 引擎与 TiKV 所使用的 RocksDB 特性完全兼容。

这一变更不仅降低了写入放大效应，在处理大 Value 的写入、更新和点查场景时也表现得更加出色。同时，在 Range Scan 场景下，通过对 Titan 引擎的优化，默认配置下 Titan 引擎的性能测试结果和 RocksDB 基本持平。

该配置的变更对历史版本兼容，已有的 TiDB 集群在升级到 TiDB v7.6.0 或之后版本时，仍会默认保持关闭 Titan 引擎。你可以根据实际的需求手动开启或者关闭 Titan 引擎。

更多信息，请参考用户文档。

- 支持下推以下字符串函数到 TiKV #48170 @gengliqi
  - LOWER()
  - UPPER()

  更多信息，请参考用户文档。
- 新增支持下推以下 JSON 函数到 TiFlash #48350 #48986 #48994 #49345 #49392 @SeaRise @yibin87
  - JSON_UNQUOTE()
  - JSON_ARRAY()
  - JSON_DEPTH()
  - JSON_VALID()
  - JSON_KEYS()
  - JSON_CONTAINS_PATH()

  更多信息，请参考用户文档。
- 建表性能提升 10 倍（实验特性）#49752 @gmhdbjd

  在之前的版本里，将上游数据库上万张表迁移到 TiDB 时，TiDB 创建这些表耗时长，效率低。从 v7.6.0 开始，引入了新的 TiDB DDL V2 架构，你可以通过设置系统变量 tidb_ddl_version 开启。相比之前的版本，新版本的 DDL 批量建表性能提升了高达 10 倍，从而大幅减少了建表时间。

  更多信息，请参考用户文档。

- 支持周期性全量数据整理（实验特性）#12729 afeinberg

  从 v7.6.0 开始，TiDB 支持 TiKV 周期性全量数据整理。该功能可以作为垃圾回收 (GC) 的增强，用以消除冗余的数据版本。在业务活动呈现明显的高峰和低谷的场景中，利用该功能可在系统空闲时段进行数据整理，以提升高峰期间业务处理的性能。

  你可以通过配置 TiKV 配置项 periodic-full-compact-start-times 指定启动周期性全量数据整理的时间，并通过 periodic-full-compact-start-max-cpu 控制 TiKV 执行周期性全量数据整理时的 CPU 使用率阈值。periodic-full-compact-start-max-cpu 默认是 10%，即为了减少对业务流量的影响，只有当 TiKV 的 CPU 利用率低于 10% 时，才会触发周期性全量数据整理。

  更多信息，请参考用户文档。

3.1.3 稳定性

- 跨数据库绑定执行计划 #48875 @qw4990

  在 TiDB 上运行 SaaS 服务时，为了方便数据维护和管理，通常会将每个租户的数据独立存储于不同数据库中，并执行相同的业务逻辑。这导致数百个数据库中存在相同的表和索引定义，执行类似的 SQL 语句。在这种场景下，对一条 SQL 语句的执行计划进行绑定 (SQL Binding) 时，这条绑定通常也适用于其他数据库中的 SQL 语句。

  针对这种应用场景，TiDB v7.6.0 引入跨数据库绑定，可以为模式相同的 SQL 语句绑定相同的执行计划，即使这些 SQL 运行在不同的数据库上。创建跨数据库绑定时，需要将数据库名用通配符 * 表示，如下所示。此时，无论 t1 和 t2 表位于哪个数据库，TiDB 都将尝试使用此绑定来生成执行计划，无需为每个数据库中的 SQL 单独创建绑定。

```
CREATE GLOBAL BINDING FOR
USING
  SELECT /*+ merge_join(t1, t2) */ t1.id, t2.amount
```

```
FROM *.t1, *.t2
WHERE t1.id = t2.id;
```

此外，跨数据库绑定能有效缓解由于用户数据和负载的不均衡及其快速变化所引发的 SQL 性能问题。通过跨数据库绑定，SaaS 服务商可以固定由拥有大量数据的用户已验证的执行计划，从而固定所有用户的执行计划。对于 SaaS 服务商，该功能提供了显著的便利性和体验提升。

由于跨数据库绑定会带来系统开销（小于 1%），TiDB 默认将其关闭。要使用跨数据库绑定，首先需要开启 tidb_opt_enable_fuzzy_binding 系统变量。

更多信息，请参考用户文档。

3.1.4 高可用

- 支持代理组件 TiProxy（实验特性）#413 @djshow832 @xhebox

  TiProxy 是 TiDB 的官方代理组件，位于客户端和 TiDB server 之间，为 TiDB 提供负载均衡、连接保持功能，让 TiDB 集群的负载更加均衡，并在维护操作期间不影响用户对数据库的连接访问。其应用场景如下：

  - 在 TiDB 集群进行滚动重启、滚动升级、缩容等维护操作时，TiDB server 会发生变动，导致客户端与发生变化的 TiDB server 的连接中断。通过使用 TiProxy，可以在这些维护操作过程中平滑地将连接迁移至其他 TiDB server，从而让客户端不受影响。
  - 所有客户端对 TiDB server 的连接都无法动态迁移至其他 TiDB server。当多个 TiDB server 的负载不均衡时，可能出现整体集群资源充足，但某些 TiDB server 资源耗尽导致延迟大幅度增加的情况。为解决此问题，TiProxy 提供连接动态迁移功能，在客户端无感的前提下，将连接从一个 TiDB server 迁移至其他 TiDB server，从而实现 TiDB 集群的负载均衡。

TiProxy 已集成至 TiUP、TiDB Operator、TiDB Dashboard 等 TiDB 基本组件中，可以方便地进行配置、部署和运维。

更多信息，请参考用户文档。

3.1.5 SQL 功能

- LOAD DATA 支持显式事务和回滚 #49079 @ekexium

  与 MySQL 相比，v7.6.0 之前的 LOAD DATA 语句在不同 TiDB 版本中的事务行为存在差异，导致使用该语句时可能需要额外进行调整。具体来说：在 v4.0.0 之前，每导入 20000 行数据就会进行一次提交。从 v4.0.0 到 v6.6.0，TiDB 默认在一个事务中提交所有行，但也支持通过设置 tidb_dml_batch_size 系统变量实现每固定的行数进行一次提交。自 v7.0.0 起，tidb_dml_batch_size 对 LOAD DATA 语句不再生效，TiDB 将在一个事务中提交所有行。

  从 v7.6.0 开始，LOAD DATA 在事务中与其它普通 DML 的处理方式一致，特别是和 MySQL 的事务行为一致。事务内的 LOAD DATA 语句本身不再自动提交当前事务，也不会开启新事务，并且事务内的 LOAD DATA 语句可以被显式提交或者回滚。此外，LOAD DATA 语句会受 TiDB 事务模式设置（乐观/悲观）影响。这些改进简化了数据从 MySQL 到 TiDB 的迁移过程，使得数据导入体验更加统一和可控。

  更多信息，请参考用户文档。

3.1.6 数据库管理

- 闪回功能支持精确 TSO #48372 @BornChanger

  TiDB v7.6.0 提供了更加强大和精确的闪回功能 FLASHBACK CLUSTER，不仅支持回溯到过去指定的时间点，还可以通过 FLASHBACK CLUSTER TO TSO 精确地指定要恢复的 TSO 时间戳，实现更加灵活的数据恢复。例如，与 TiCDC 结合使用时，该功能允许下游 TiDB 集群在暂停数据同步、开启预上

线读写测试后，快速且优雅地回溯到暂停同步时的 TSO 时间戳，并继续通过 TiCDC 同步数据，从而简化了预上线验证流程和数据管理。

```
FLASHBACK CLUSTER TO TSO 445494839813079041;
```

更多信息，请参考用户文档。

- 支持自动终止长时间未提交的空闲事务 #48714 @crazycs520

  在网络异常断开或应用程序故障时，COMMIT/ROLLBACK 语句可能无法正常传送到数据库。这种情况可能导致数据库锁未能及时释放，进而引起事务锁等待以及数据库连接数快速增加。这类问题在测试环境中较常见，但在线上环境也会偶尔发生，并且有时难以迅速诊断。为有效防止此类问题的发生，TiDB v7.6.0 引入 tidb_idle_transaction_timeout 系统变量，可以自动终止长时间运行的空闲事务。当用户会话处于事务状态且空闲时间超过该变量设定的值时，TiDB 会自动强制结束该事务的数据库连接并回滚事务。

  更多信息，请参考用户文档。

- 简化执行计划绑定的语法 #48876 @qw4990

  TiDB v7.6.0 简化了创建执行计划绑定的语法。在创建执行计划绑定的命令中无需提供原 SQL 语句，TiDB 可以根据带 hint 的语句识别出对应的原 SQL。这一改进提高了创建执行计划绑定的便利性。例如：

  ```
  CREATE GLOBAL BINDING
  USING
  SELECT /*+ merge_join(t1, t2) */ * FROM t1, t2 WHERE t1.id = t2.id;
  ```

  更多信息，请参考用户文档。

- 支持动态调整 TiDB 单行记录大小限制 #49237 @zyguan

  在 v7.6.0 之前，事务中单行记录的大小受 TiDB 配置项 txn-entry-size-limit 限制。如果单行记录的大小超出此限制，TiDB 将返回 entry too large 错误。此时，用户需要修改 TiDB 配置文件并重启 TiDB 才能够生效。为降低

用户的管理成本，TiDB v7.6.0 新增系统变量 tidb_txn_entry_size_limit，支持动态修改 txn-entry-size-limit 配置项的值。该变量的默认值为 0，表示默认使用 txn-entry-size-limit 配置项的值作为限制。当设置为非 0 值时，TiDB 优先使用该变量的值作为事务中的单行记录大小的限制。这一改进旨在提高用户调整系统配置的灵活性，无需重启 TiDB 即可生效。

更多信息，请参考用户文档。

- BR 默认恢复用户账号等系统表数据 #48567 @BornChanger #49627 @Leavrth

  从 br v5.1.0 开始，快照备份时默认自动备份 **mysql schema** 下的系统表数据，但恢复数据时默认不恢复系统表数据。在 v6.2.0 中，br 增加恢复参数 --with-sys-table 支持恢复数据的同时恢复部分系统表相关数据，提供更多的操作灵活性。

  为了进一步降低用户的管理成本，并提供更直观的默认行为。从 v7.6.0 开始，br 默认开启恢复参数 --with-sys-table，并支持恢复 user 为 cloud_admin 的用户数据。这意味着 br 默认支持恢复数据的同时恢复部分系统表相关数据，特别是用户账号和表的统计信息数据。这一改进旨在使备份恢复操作更加直观，减轻手动配置的负担，从而提升整体的操作体验。

  更多信息，请参考用户文档。

3.1.7 可观测性

- 增强资源管控相关的观测性 #49318 @glorv @bufferflies @nolouch

  随着越来越多用户利用资源组对业务应用进行隔离，资源管控提供了更丰富的基于资源组的数据，协助你观测资源组负载、资源组设置，确保出现问题时能够快速发现并精准诊断。其中包括：

  - 慢查询日志增加资源组名称、RU 消耗、以及等待资源耗时。

  - Statement Summary Tables 增加资源组名称、RU 消耗、以及等待资源耗时。
  - 在变量 tidb_last_query_info 中增加了 SQL 的 RU 消耗信息 ru_consumption，你可以利用此变量获取会话中上一条语句的资源消耗。
  - 增加基于资源组的数据库指标：QPS/TPS、执行时间 (P999/P99/P95)、失败次数、连接数。
  - 增加系统表 request_unit_by_group 记录资源组每天的历史资源消耗。

  更多信息，请参考慢查询日志、Statement Summary Tables、资源管控 (Resource Control) 监控指标详解。

3.1.8 数据迁移

- Data Migration (DM) 支持迁移 MySQL 8.0 的功能成为正式功能（GA） #10405 @lyzx2001

  之前 DM 迁移 MySQL8.0 仅为实验特性，不能用于生产环境。TiDB v7.6.0 增强了该功能的稳定性、兼容性，可在生产环境帮助你平滑、快速地将数据从 MySQL 8.0 迁移到 TiDB。在 v7.6.0 中，该功能正式 GA。

  更多信息，请参考用户文档。

- TiCDC 支持通过双向复制模式 (Bi-Directional Replication, BDR) 同步 DDL 语句（实验特性）#10301 #48519 @okJiang @asddongmen

  从 v7.6.0 开始，TiCDC 支持在配置了双向复制的情况下同步 DDL 语句。以前，TiCDC 不支持复制 DDL 语句，因此要使用 TiCDC 双向复制必须将 DDL 语句分别应用到两个 TiDB 集群。有了该特性，TiCDC 可以为一个集群分配 PRIMARY BDR role，并将该集群的 DDL 语句复制到下游集群。

  更多信息，请参考用户文档。

- TiCDC 支持查询 changefeed 的下游同步状态 #10289 @hongyunyan

  从 v7.6.0 起，TiCDC 引入了一个新的 API GET /api/v2/changefeed/{changefeed_id}/synced，用于查询指定同步任务 (changefeed) 的下游同步状态。通过此 API，你可以判断 TiCDC 是否已将所接收到的上游数据完全同步到下游。

  更多信息，请参考用户文档。

- TiCDC 支持将 CSV 格式中的 delimiter 设置为 3 个字符 #9969 @zhangjinpeng87

  从 v7.6.0 开始，你可以将 TiCDC 输出的 CSV 格式中的 delimiter 设置为 1 到 3 个字符。例如，你可以指定 TiCDC 使用 2 个字符的 delimiter（例如 || 或 $^）或 3 个字符的 delimiter（例如 |@|）分隔字段。

  更多信息，请参考用户文档。

3.2 兼容性变更

注意：

以下为从 v7.5.0 升级至当前版本 (v7.6.0) 所需兼容性变更信息。如果从 v7.4.0 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 MySQL 兼容性

- 在 TiDB v7.6.0 之前，LOAD DATA 操作在一个事务中提交所有行，或者以批量方式提交事务，和 MySQL 的行为略有不同。从 TiDB v7.6.0 开始，LOAD DATA 的事务行为与 MySQL 的事务行为一致，包括事务内的 LOAD DATA 语句本身不再自动提交当前事务，也不会开启新事务，并且事务内的 LOAD DATA 语句可以被显式提交或者回滚。此外，LOAD DATA 语句会受 TiDB 事务模式设置（乐观/悲观）影响。#49079 @ekexium

3.2.2 系统变量

变量名	修改类型	描述
tidb_auto_analyze_partition_batch_size	修改	经进一步的测试后，该变量默认值从 1 修改为 128。
tidb_sysproc_scan_concurrency	修改	在大规模集群里，scan 操作的并发度可以调整的更高，以满足 ANALYZE 的需要，因此将该变量最大值由 256 修改为 4294967295。
tidb_analyze_distsql_scan_concurrency	新增	用于设置执行 ANALYZE 时 scan 操作的并发度。默认值为 4。
tidb_ddl_version	新增	用于控制是否开启

变量名	修改类型	描述
		TiDB DDL V2。将该变量的值设置为 2 可以开启该功能，设置为 1 关闭该功能。默认值为 1。开启后，将使用新版本的实现执行 DDL 语句。TiDB DDL V2 对 DDL 功能做了提升，建表 DDL 的执行速度相比 V1 版本提升 10 倍。
tidb_enable_global_index	新增	用于控制是否支持对分区表创建 Global index。默认值为 OFF。

变量名	修改类型	描述
		Global index 当前正处于开发阶段，**不推荐修改该变量值**。
tidb_idle_transaction_timeout	新增	用来控制用户会话中事务的空闲超时。当用户会话处于事务状态且空闲时间超过该变量设定的值时，会话会被 Kill 掉。默认值 0 表示没有时间限制。
tidb_opt_enable_fuzzy_binding	新增	用于控制是否开启跨数据库绑定执行计划功能，默认值 OFF 表示关闭。

变量名	修改类型	描述
tidb_txn_entry_size_limit	新增	用于动态修改 TiDB 配置项 performance.txn-entry-size-limit，即限制 TiDB 单行数据的大小。默认值为 0，表示默认使用配置项的值。当设置为非 0 值时，优先使用该变量的值作为 txn-entry-size-limit 的值。
pd_enable_follower_handle_region	新增	用于控制是否开启 Active PD Follower（实验特性）。当该值为 OFF 时，TiDB 仅从 PD leader

变量名	修改类型	描述
		获取 Region 信息。当该值为 ON 时，TiDB 在获取 Region 信息时会将请求均匀地发送到所有 PD 节点上，因此 PD follower 也可以处理 Region 信息请求，从而减轻 PD leader 的 CPU 压力。

3.2.3 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	tls-version	修改	默认值为空，TiDB 默认支持的 TLS 版本从 TLS1.1 及更高提升为 TLS1.2 及更高。
TiKV	blob-file-compression	修改	设置 Titan 中 value 所使用的压缩算法。从 v7.6.0 开始，默认采用 zstd 压缩算法。

配置文件	配置项	修改类型	描述
TiKV	rocksdb.defaultcf.titan.min-blob-size	修改	从 TiDB v7.6.0 开始，新建集群默认值为 32KB。对于已有集群升级到 v7.6.0 版本的情况，默认值为 1KB 保持不变。
TiKV	rocksdb.titan.enabled	修改	开启 Titan 开关。v7.5.0 及更早的版本默认值为 false。从 v7.6.0 开始，新建集群默认值是 true，已有集群升级到 v7.6.0 或更高版本则会维持原有的配置。
TiKV	gc.num-threads	新增	设置当 enable-compaction-filter 为 false 时 GC 的线程个数。默认值为 1。
TiKV	raftstore.periodic-full-compact-start-times	新增	设置 TiKV 启动周期性全量数据整理 (Compaction) 的时间。默认值 [] 表示默认情况下禁用周期性全量数据整理。
TiKV	raftstore.periodic-full-compact-start-max-cpu	新增	设置 TiKV 执行周期性全量数据整理时的 CPU 使用率阈值，默认值为 0.1。
TiKV	zstd-dict-size	新增	指定 zstd 字典大小，默认值为 0KB，表示关闭 zstd 字典压缩。

配置文件	配置项	修改类型	描述
TiFlash	logger.level	修改	为减少日志打印的开销，默认值由 "debug" 改为 "INFO"。
TiDB Lightning	tidb.pd-addr	修改	配置 PD Server 的地址，从 v7.6.0 开始支持设置多个地址。
TiDB Lightning	block-size	新增	控制物理导入模式 (backend='local') 中本地文件排序的 I/O 区块大小。默认值为 16KiB。当 IOPS 成为瓶颈时，可以调大该参数的值以缓解磁盘 IOPS，从而提升数据导入性能。
BR	--granularity	新增	通过设置 --granularity="coarse-grained" 启用粗粒度的 Region 打散算法（实验特性）进行恢复，加快大规模 Region 场景下的 Region 恢复速度。
TiCDC	compression	新增	设置 redo log 文件的压缩行为。
TiCDC	sink.cloud-storage-config	新增	设置同步数据到对象存储时自动清理历史数据的功能。

3.2.4 系统表

- 新增系统表 INFORMATION_SCHEMA.KEYWORDS 用来展示 TiDB 支持的所有关键字的信息。
- 在系统表 INFORMATION_SCHEMA.SLOW_QUERY 中增加了以下资源管控 (Resource Control) 相关的字段：
  - Resource_group：语句执行所绑定的资源组。
  - Request_unit_read：执行语句消耗的总读 RU。
  - Request_unit_write：执行语句消耗的总写 RU。
  - Time_queued_by_rc：执行语句过程中等待可用资源的总耗时。

3.3 离线包变更

从 v7.6.0 开始，TiDB-community-server 二进制软件包中新增代理组件 TiProxy 的安装包 tiproxy-{version}-linux-{arch}.tar.gz。

3.4 废弃功能

- TiDB v7.6.0 废弃了对 TLSv1.0 和 TLSv1.1 协议的支持，并将从 TiDB v8.0.0 开始移除对这两个协议的支持。请升级 TLS 至 TLSv1.2 或 TLSv1.3。
- 执行计划的自动演进绑定（实验特性）将从 TiDB v8.0.0 开始废弃，等同的功能将会在后续版本中重新设计。
- 系统变量 tidb_disable_txn_auto_retry 将从 TiDB v8.0.0 开始废弃，废弃后将不再支持乐观事务的自动重试。

3.5 改进提升

- TiDB
  - 当使用非二进制排序规则并且查询条件中包含 LIKE 时，优化器可以生成 IndexRangeScan 以提升执行效率 #48181 #49138 @time-and-fate

  - 增强特定情况下 OUTER JOIN 转 INNER JOIN 的能力 #49616 @qw4990
  - 提升分布式执行框架任务在节点重启场景下的均衡性 #47298 @ywqzzy
  - 允许多个快速加索引 DDL 任务排队执行，而非回退为普通加索引任务 #47758 @tangenta
  - 增强 ALTER TABLE ... ROW_FORMAT 的兼容性 #48754 @hawkingrei
  - 将 CANCEL IMPORT JOB 命令调整为同步命令 #48736 @D3Hunter
  - 提升空表加索引的速度 #49682 @zimulala
  - 当关联子查询的列未被上层算子引用时，可以直接消除该关联子查询 #45822 @King-Dylan
  - EXCHANGE PARTITION 操作会触发统计信息的维护更新 #47354 @hi-rustin
  - TiDB 支持构建符合联邦信息处理标准 (FIPS) 要求的二进制文件 #47948 @tiancaiamao
  - 改进 TiDB 在处理部分类型转换时的实现，并修复相关问题 #47945 #47864 #47829 #47816 @YangKeao @lcwangchao
  - 在获取 schema 版本时，默认使用 KV timeout 特性读取，减少 meta Region leader 读取慢对 schema 版本更新的影响 #48125 @cfzjywxk
- TiKV
  - 增加查询异步任务的 API endpoint /async_tasks #15759 @YuJuncen
  - 给 gRPC 监控增加优先级的标签，从而显示资源管理中的各个不同优先级的资源组的数据 #49318 @bufferflies
  - 支持动态调整参数 readpool.unified.max-tasks-per-worker 的值，可根据优先级单独核算正在运行的任务数 #16026 @glorv
  - 支持动态调整 GC 的线程数，默认值为 1 #16101 @tonyxuqqi
- PD

  - 提升 PD TSO 在磁盘抖动时的可用性 #7377 @HuSharp
- TiFlash
  - 降低磁盘性能抖动对读取延迟的影响 #8583 @JaySon-Huang
  - 减少后台数据 GC 任务对读、写任务延迟的影响 #8650 @JaySon-Huang
  - 支持在存算分离架构下通过合并相同数据的读取操作，提升多并发下的数据扫描性能 #6834 @JinheLin
  - 优化 JOIN ON 条件中仅包含 JOIN KEY 等值条件时，半连接 (SEMI JOIN) 及 LEFT OUTER SEMIJOIN 的执行性能 #47424 @gengliqi
- Tools
  - Backup & Restore (BR)
    - 新增全量备份恢复阶段对 Amazon S3 session-token 以及 assume-role 的认证支持 #39832 @3pointer
    - 新增 PITR 对 delete range 场景的集成测试，提升 PITR 稳定性 #47738 @Leavrth
    - 提升了 RESTORE 语句在大数据量表场景下的建表性能 #48301 @Leavrth
    - 重构 BR 异常处理机制，提高对未知错误的容忍度 #47656 @3pointer
  - TiCDC
    - 通过增加并行，优化了 TiCDC 同步数据到对象存储的性能 #10098 @CharlesCheung96
    - 支持通过在 sink-uri 中设置 content-compatible=true 使 TiCDC Canal-JSON 兼容 Canal 官方输出的内容格式 #10106 @3AceShowHand
  - TiDB Data Migration (DM)
    - 为 DM OpenAPI 增加了全量物理导入的相关配置 #10193 @GMHDBJD

- TiDB Lightning
    - 支持配置多个 PD 地址以增强稳定性 #49515 @mittalrishabh
    - 支持通过配置参数 block-size 来控制 TiDB Lightning 本地文件排序的 I/O 区块大小，提升数据导入性能 #45037 @mittalrishabh

3.6 错误修复

- TiDB
    - 修复 TiDB panic 并报错 invalid memory address or nil pointer dereference 的问题 #42739 @CbcWestwolf
    - 修复当 DDL jobID 恢复为 0 时 TiDB 节点 panic 的问题 #46296 @jiyfhust
    - 修复某些情况下相同的查询计划拥有不同的 PLAN_DIGEST 的问题 #47634 @King-Dylan
    - 修复 UNION ALL 第一个子节点是 DUAL Table 时，执行可能报错的问题 #48755 @winoros
    - 修复当 tidb_max_chunk_size 值较小时，包含公共表表达式 (CTE) 的查询出现 runtime error: index out of range [32] with length 32 错误的问题 #48808 @guo-shaoge
    - 修复使用 AUTO_ID_CACHE=1 时 Goroutine 泄漏的问题 #46324 @tiancaiamao
    - 修复 MPP 计算 COUNT(INT) 时结果可能出错的问题 #48643 @AilinKid
    - 修复当分区列类型为 DATETIME 时，执行 ALTER TABLE ... LAST PARTITION 失败的问题 #48814 @crazycs520
    - 修复数据中包含后导空格时，在 LIKE 中使用 _ 通配符可能会导致查询结果出错的问题 #48983 @time-and-fate

- 修复 tidb_server_memory_limit 导致内存长期压力较高时，TiDB CPU 利用率过高的问题 #48741 @XuHuaiyu
- 修复 ENUM 类型列作为 join 键时，查询结果错误的问题 #48991 @winoros
- 修复当内存使用超限时包含 CTE 的查询非预期卡住的问题 #49096 @AilinKid
- 修复 TiDB server 在使用企业插件审计日志时可能占用大量资源的问题 #49273 @lcwangchao
- 修复特定情况下优化器将 TiFlash 选择路径错误转化为 DUAL Table 的问题 #49285 @AilinKid
- 修复包含递归 (WITH RECURSIVE) CTE 的 UPDATE 或 DELETE 语句可能会产生错误结果的问题 #48969 @winoros
- 修复包含 IndexHashJoin 算子的查询由于内存超过 tidb_mem_quota_query 而卡住的问题 #49033 @XuHuaiyu
- 修复在非严格模式下 (sql_mode = ")，INSERT 过程中产生截断仍然会报错的问题 #49369 @tiancaiamao
- 修复 CTE 查询在重试过程中可能会报错 type assertion for CTEStorageMap failed 的问题 #46522 @tiancaiamao
- 修复在嵌套的 UNION 查询中 LIMIT 和 ORDER BY 可能无效的问题 #49377 @AilinKid
- 修复在解析 ENUM 或 SET 类型的非法值时会导致 SQL 语句报错的问题 #49487 @winoros
- 修复构造统计信息时因为 Golang 隐式转换算法导致统计信息误差过大的问题 #49801 @qw4990
- 修复在某些时区下夏令时显示有误的问题 #49586 @overvenus
- 修复在有大量表时，AUTO_ID_CACHE=1 的表可能造成 gRPC 客户端泄漏的问题 #48869 @tiancaiamao

- 修复 TiDB server 在优雅关闭 (graceful shutdown) 时可能 panic 的问题 #36793 @bb7133
- 修复 ADMIN RECOVER INDEX 在处理包含 CommonHandle 的表时报错 ERROR 1105 的问题 #47687 @Defined2014
- 修复执行 ALTER TABLE t PARTITION BY 时指定 Placement Rules 报错 ERROR 8239 的问题 #48630 @mjonss
- 修复 INFORMATION_SCHEMA.CLUSTER_INFO 中 START_TIME 列类型不合理的问题 #45221 @dveeden
- 修复 INFORMATION_SCHEMA.COLUMNS 中 EXTRA 列类型不合理导致报错 Data Too Long, field len 30, data len 45 的问题 #42030 @tangenta
- 修复 IN (...) 语句导致 INFORMATION_SCHEMA.STATEMENTS_SUMMARY 中的 PLAN_DIGEST 不同的问题 #33559 @King-Dylan
- 修复 TIME 类型转换为 YEAR 类型时，返回的结果混合了 TIME 和年份的问题 #48557 @YangKeao
- 修复关闭 tidb_enable_collect_execution_info 导致 Coprocessor Cache panic 的问题 #48212 @you06
- 修复 shuffleExec 意外退出导致 TiDB 崩溃的问题 #48230 @wshwsh12
- 修复静态 CALIBRATE RESOURCE 依赖 Prometheus 数据的问题 #49174 @glorv
- 修复在日期中加上数值较大的 Interval 时返回错误结果的问题。修复后，带有无效前缀或字符串 true 的 Interval 将被视为零值，与 MySQL 8.0 保持一致 #49227 @lcwangchao
- 修复 ROW 函数对 null 类型推断有误导致意外报错的问题 #49015 @wshwsh12
- 修复在某些情况下 ILIKE 函数可能导致数据竞争的问题 #49677 @lcwangchao

- 修复由于 STREAM_AGG() 错误处理 CI 导致查询结果有误的问题 #49902 @wshwsh12
- 修复将字节转换为 TIME 时出现编码失败的问题 #47346 @wshwsh12
- 修复 CHECK 约束的 ENFORCED 选项的行为与 MySQL 8.0 不一致的问题 #47567 #47631 @jiyfhust
- 修复 CHECK 约束的 DDL 卡住的问题 #47632 @jiyfhust
- 修复由于内存不足导致 DDL 快速加索引失败的问题 #47862 @GMHDBJD
- 修复在执行加索引的过程中升级集群可能导致数据与索引不一致的问题 #46306 @zimulala
- 修复更新 tidb_mem_quota_query 系统变量后执行 ADMIN CHECK 报错 ERROR 8175 的问题 #49258 @tangenta
- 修复 ALTER TABLE 修改外键引用列的类型时，DECIMAL 精度发生变化没有报错的问题 #49836 @yoshikipom
- 修复 ALTER TABLE 修改外键引用列的类型时，INTEGER 长度发生变化误报错的问题 #47702 @yoshikipom
- 修复某些场景下表达式索引没有发现除数是 0 的问题 #50053 @lcwangchao
- 缓解当要处理的表的数量过多时，TiDB 节点 OOM 的问题 #50077 @zimulala
- 修复集群滚动重启时 DDL 卡在运行中状态的问题 #50073 @tangenta
- 修复使用 PointGet 或 BatchPointGet 算子访问分区表的全局索引时，结果可能出错的问题 #47539 @L-maple
- 修复当生成列上的索引设置为可见时，可能无法选中 MPP 计划的问题 #47766 @AilinKid
- 修复 LIMIT 可能无法推入到 OR 类型的 Index Merge 的问题 #48588 @AilinKid

- 修复 BR 导入后，mysql.bind_info 表中可能存在重复的内置 (builtin) 行的问题 #46527 @qw4990
- 修复删除分区后，分区表的统计信息更新行为不合理的问题 #48182 @hi-rustin
- 修复并发合并分区表的全局统计信息时可能遇到报错的问题 #48713 @hawkingrei
- 修复在具有补齐空格 (PADDING SPACE) 的列上使用 LIKE 运算符进行索引范围扫描时，查询结果可能出错的问题 #48821 @time-and-fate
- 修复生成列可能触发对内存的并发读写导致数据竞争的问题 #44919 @tangenta
- 修复当指定 WITH 0 TOPN（即不收集 TopN 的统计信息）时，ANALYZE TABLE 仍然可能收集 Top1 的统计信息的问题 #49080 @hawkingrei
- 修复不合法的优化器 hint 可能会导致合法 hint 不生效的问题 #49308 @hawkingrei
- 修复对 Hash 类型的分区表进行分区的增删重组或 TRUNCATE 操作时，统计信息没有对应更新的问题 #48235 #48233 #48226 #48231 @hi-rustin
- 修复设置统计信息自动更新的时间窗口后，时间窗口外统计信息仍然可能更新的问题 #49552 @hawkingrei
- 修复从分区表转为非分区表时，旧统计信息不会自动删除的问题 #49547 @hi-rustin
- 修复当使用 TRUNCATE TABLE 清空非分区表的数据时，旧统计信息不会自动删除的问题 #49663 @hi-rustin
- 修复当查询使用了会强制排序的优化器 hint（例如 STREAM_AGG()）且其执行计划包含 IndexMerge 时，强制排序可能会失效的问题 #49605 @AilinKid
- 修复直方图的边界包含 NULL 时，直方图统计信息可能无法解析成可读字符串的问题 #49823 @AilinKid

  - 修复查询语句包含 GROUP_CONCANT(ORDER BY) 语法时，执行可能出错的问题 #49986 @AilinKid
  - 修复当未使用严格的 SQL_MODE 时，UPDATE、DELETE、INSERT 语句返回溢出错误而非警告的问题 #49137 @YangKeao
  - 修复当表中存在由多值索引和非 BINARY 类型字符串组成的复合索引时，数据无法插入的问题 #49680 @YangKeao
  - 修复多级嵌套的 UNION 查询中 LIMIT 无效的问题 #49874 @Defined2014
  - 修复当使用 BETWEEN ... AND ... 条件查询分区表时结果有误的问题 #49842 @Defined2014
  - 修复无法在 REPLACE INTO 语句中使用 hint 的问题 #34325 @YangKeao
  - 修复在查询 Hash 分区表时 TiDB 可能选择错误的分区导致结果有误的问题 #50044 @Defined2014
  - 修复使用 MariaDB Connector/J 并配置启用压缩时发生连接错误的问题 #49845 @onlyacat
- TiKV
  - 修复损坏的 SST 文件可能会扩散到其他 TiKV 节点导致 panic 的问题 #15986 @Connor1996
  - 修复 Online Unsafe Recovery 时无法处理 merge abort 的问题 #15580 @v01dstar
  - 修复扩容时可能导致 DR Auto-Sync 的 joint state 超时问题 #15817 @Connor1996
  - 修复 Titan blob-run-mode 无法在线更新的问题 #15978 @tonyxuqqi
  - 修复 Resolved TS 可能被阻塞两小时的问题 #11847 #15520 #39130 @overvenus
  - 修复在 Flashback 时遇到 notLeader 或 regionNotFound 时卡住的问题 #15712 @HuSharp

  - 修复如果 TiKV 运行极慢，在 Region Merge 之后可能 panic 的问题 #16111 @overvenus
  - 修复 GC 扫描过期 lock 时无法读取内存悲观锁的问题 #15066 @cfzjywxk
  - 修复 Titan 监控中 blob 文件大小不正确的问题 #15971 @Connor1996
  - 修复 TiCDC 同步大表可能导致 TiKV OOM 的问题 #16035 @overvenus
  - 修复 TiDB 和 TiKV 处理 DECIMAL 算术乘法截断时结果不一致的问题 #16268 @solotzg
  - 修复 cast_duration_as_time 可能返回错误结果的问题 #16211 @gengliqi
  - 修复巴西和埃及时区转换错误的问题 #16220 @overvenus
  - 修复 gRPC threads 在检查 is_shutdown 时可能出现 panic 的问题 #16236 @pingyu
- PD
  - 修复 PD 内 etcd 健康检查没有移除过期地址的问题 #7226 @iosmanthus
  - 修复 PD Leader 切换且新 Leader 与调用方之间存在网络隔离时，调用方不能正常更新 Leader 信息的问题 #7416 @CabinfeverB
  - 将 Gin Web Framework 的版本从 v1.8.1 升级到 v1.9.1 以修复部分安全问题 #7438 @niubell
  - 修复在不满足副本数量需求时，删除 orphan peer 的问题 #7584 @bufferflies
- TiFlash
  - 修复当查询遇到内存限制后发生内存泄漏的问题 #8447 @JinheLin
  - 修复在执行 FLASHBACK DATABASE 后 TiFlash 副本的数据仍会被 GC 回收的问题 #8450 @JaySon-Huang
  - 修复慢查询导致内存使用显著增加的问题 #8564 @JinheLin

  - 修复在 CREATE TABLE、DROP TABLE 频繁执行的场景下，部分 TiFlash 副本数据无法通过 RECOVER TABLE 或 FLASHBACK TABLE 恢复的问题 #1664 @JaySon-Huang
  - 修复在查询带有类似 ColumnRef in (Literal, Func...) 的过滤条件时，查询结果出错的问题 #8631 @Lloyd-Pottiger
  - 修复在 TiDB 执行并发 DDL 遇到冲突时 TiFlash panic 的问题 #8578 @JaySon-Huang
  - 修复存算分离架构下，可能无法正常选出对象存储数据 GC owner 的问题 #8519 @JaySon-Huang
  - 修复 lowerUTF8/upperUTF8 不允许大小写字符占据不同字节数的错误 #8484 @gengliqi
  - 修复 TiFlash 错误处理 ENUM 偏移量为 0 的问题 #8311 @solotzg
  - 修复表达式 INET_NTOA() 中的兼容性问题 #8211 @solotzg
  - 修复在 stream 读时扫描多个分区表可能导致潜在的 OOM 问题 #8505 @gengliqi
  - 修复成功执行的短查询打印过多信息日志的问题 #8592 @windtalker
  - 修复 TiFlash 在停止时可能崩溃的问题 #8550 @guo-shaoge
  - 修复 GREATEST 或 LEAST 函数在包含常量字符串参数时，可能发生的随机无效内存访问的问题 #8604 @windtalker
- Tools
  - Backup & Restore (BR)
    - 修复生成外部存储文件 URI 错误的问题 #48452 @3AceShowHand
    - 修复在任务初始化阶段出现与 PD 的连接错误导致日志备份任务虽然启动但无法正常工作的问题 #16056 @YuJuncen
    - 修复日志备份任务可能出现内存泄露以及备份任务启动后无法正常运行的问题 #16070 @YuJuncen

- 修复 PITR 恢复过程中插入数据到系统表 mysql.gc_delete_range 出错的问题 #49346 @Leavrth
- 修复从旧版本的备份恢复数据时报错 Unsupported collation 的问题 #49466 @3pointer
- 修复在部分场景下通过快照恢复用户表后权限更新不及时的问题 #49394 @Leavrth

– TiCDC

- 修复某些场景下在同步 DELETE 语句时，WHERE 条件没有采用主键作为条件的问题 #9812 @asddongmen
- 修复同步数据到对象存储时，可能会出现 TiCDC Server panic 的问题 #10137 @sdojjy
- 修复 kv-client 初始化过程中可能出现数据竞争的问题 #10095 @3AceShowHand
- 修复在某些特殊场景下，TiCDC 错误地关闭与 TiKV 的连接的问题 #10239 @hicqu
- 修复上游在执行有损 DDL 时，TiCDC Server 可能 panic 的问题 #9739 @hicqu
- 修复数据同步到下游 MySQL 时可能出现 checkpoint-ts 卡住的问题 #10334 @zhangjinpeng87

– TiDB Data Migration (DM)

- 修复 DM 遇到 “event type truncate not valid” 错误导致升级失败的问题 #10282 @GMHDBJD
- 修复 GTID 模式同步时性能可能会下降的问题 #9676 @feran-morgan-pingcap
- 修复下游表结构包含 shard_row_id_bits 时同步任务报错的问题 #10308 @GMHDBJD

3.7 贡献者

感谢来自 TiDB 社区的贡献者们：

- 0o001（首次贡献者）
- bagechengzi（首次贡献者）
- feran-morgan-pingcap（首次贡献者）
- highpon
- jiyfhust
- L-maple
- lkshminarayanan（首次贡献者）
- lyang24（首次贡献者）
- mittalrishabh
- morgo
- nkg-（首次贡献者）
- onlyacat
- shawn0915
- Smityz
- szpnygo（首次贡献者）
- ub-3（首次贡献者）
- xiaoyawei（首次贡献者）
- yorkhellen
- yoshikipom（首次贡献者）
- Zheaoli