TiDB Release Notes：从 v8.2.0 到 v8.5.0 的变更

PingCAP

20241219

本文档包含 TiDB v8.2.0-DMR、v8.3.0-DMR、v8.4.0-DMR 和 v8.5.0-LTS 的 release notes。当从 TiDB v8.1.x 升级到 v8.5.0 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 v8.5.0 的详细信息和使用文档，请参考 TiDB v8.5 文档。

目录

1 TiDB 8.5.0 Release Notes

发版日期：2024 年 12 月 19 日

TiDB 版本：8.5.0

试用链接：快速体验 | 生产部署 | 下载离线包

TiDB 8.5.0 为长期支持版本 (Long-Term Support Release, LTS)。

相比于前一个 LTS（即 8.1.0 版本），8.5.0 版本包含 8.2.0-DMR、8.3.0-DMR, 和 8.4.0-DMR 中已发布的新功能、提升改进和错误修复。当你从 8.1.x 升级到 8.5.0 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 8.1.0 到 8.5.0 的一些关键特性：

分类	功能	描述
可扩展性与性能	多维度降低数据处理延迟	TiDB 不断优化数据处理细节，持续提升性能，以更好地满足金融领域对 SQL 处理低延迟的高要求。关键更新包括： 并行排序（从 v8.2.0 开始引入） 优化 KV 请求批处理策略（从 v8.3.0 开始引入） 并行获取 TSO（从 v8.4.0 开始引入） 降低 DELETE 操作的资源开销（从 v8.4.0 开始引入） 优化缓存表场景性能（从 v8.4.0 开始引入） Hash Join 算法优化（实验特性，从 v8.4.0 开始引入）

分类	功能	描述
	TiKV MVCC 内存引擎（从 v8.5.0 开始引入）	TiKV MVCC 内存引擎 (In-Memory Engine, IME) 将最新写入的 MVCC 版本的数据缓存到内存中，帮助 TiKV 快速跳过旧版本，直接检索最新数据。在数据记录频繁更新或历史版本保留时间较长的场景下，该特性可以显著提升数据扫描性能。
	通过 Active PD Follower 提升 PD Region 信息查询服务的扩展能力（从 v8.5.0 开始成为正式功能）	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 节点数较多和 Region 数较多的集群中，该特性可以提升 PD 集群处理 GetRegion 和 ScanRegions 请求的能力，减轻 PD leader 的 CPU 压力。在 v8.5.0，Active PD Follower 成为正式功能。
	实例级执行计划缓存（实验特性，从 v8.4.0 开始引入）	实例级执行计划缓存允许同一个 TiDB 实例的所有会话共享执行计划缓存。与现有的会话级执行计划缓存相比，实例级执行计划缓存能够在内存中缓存更多执行计划，减少 SQL 编译时间，从而降低 SQL 整体运行时间，提升 OLTP 的性能和吞吐，同时更好地控制内存使用，提升数据库稳定性。
	分区表全局索引（从 v8.4.0 开始成为正式功能）	全局索引可以有效提高检索非分区列的效率，并且消除了唯

分类	功能	描述
		一键必须包含分区键的限制。该功能扩展了 TiDB 分区表的使用场景，提升了分区表的性能，降低了分区表在一些查询场景的资源消耗。
	默认允许将 Projection 算子下推到存储引擎（从 v8.3.0 开始引入）	Projection 算子下推可以将负载分散到存储节点，同时减少节点间的数据传输。这有助于降低部分 SQL 的执行时间，提升数据库的整体性能。
	统计信息收集忽略不必要的列（从 v8.3.0 开始引入）	在保证优化器能够获取到必要信息的前提下，加快了统计信息收集的速度，提升统计信息的时效性，进而保证选择最优的执行计划，提升集群性能。同时也降低了系统开销，改善了资源利用率。
稳定性与高可用	提升超大规模集群的稳定性	对于使用 TiDB 运行多租户应用或者 SaaS 应用的公司，经常需要存储大量的表，TiDB 在 v8.5.0 着力增强了大规模集群的稳定性。 Schema 缓存控制以及设置统计信息缓存使用内存的上限成为正式功能，减少了内存过度消耗带来的稳定性问题。 PD 通过 Active Follower 可应对大量 Region 带来的压力，并将 PD 所承担的服务逐步解耦，独立部署。

分类	功能	描述
		PD 优化提升 Region 心跳处理的性能，并支持集群中千万级 Regions 的规模。 通过增加并发度，以及减少收集对象的数量，统计信息收集和加载效率得到提升，保证了大集群执行计划的稳定性。
	Runaway Queries 支持更多触发条件，并能够切换资源组（从 v8.4.0 开始引入）	Runaway Queries 提供了有效的手段来降低突发的 SQL 性能问题对系统产生的影响。v8.4.0 中新增 Coprocessor 处理的 Key 的数量 (PROCESSED_KEYS) 和 Request Unit (RU) 作为识别条件，并可以将识别到的查询置入指定资源组，对 Runaway Queries 进行更精确的识别与控制。
	支持为资源管控的后台任务设置资源使用上限 (实验特性，从 v8.4.0 开始引入）	为资源管控的后台任务设置百分比上限，针对不同业务系统的需求，控制后台任务的消耗，从而将后台任务的消耗限制在一个很低的水平，保证在线业务的服务质量。
	增强并扩展 TiProxy 的使用场景	作为 TiDB 高可用的重要组件，TiProxy 除了提供 SQL 流量接入和转发功能外，新增了对集群变更的评估能力。主要包括： TiProxy 流量捕获和回放（实验特性，从 v8.4.0 开始引入）

分类	功能	描述
		TiProxy 内置虚拟 IP 管理（从 v8.3.0 开始引入） TiProxy 支持多种负载均衡策略（从 v8.2.0 开始引入）
	并行 HashAgg 算法支持数据落盘（从 v8.2.0 开始成为正式功能）	HashAgg 是 TiDB 中常用的聚合算子，用于快速聚合具有相同字段值的行。TiDB v8.0.0 引入并行 HashAgg 作为实验特性，以进一步提升处理速度。当内存资源不足时，并行 HashAgg 可以将临时排序数据落盘，避免因内存使用过度而导致的 OOM 风险，从而提升查询性能和节点稳定性。该功能在 v8.2.0 成为正式功能，并默认开启，用户可以通过 tidb_executor_concurrency 安全地设置并行 HashAgg 的并发度。
SQL	外键约束（从 v8.5.0 开始成为正式功能）	外键 (Foreign Key) 是数据库中的一种约束，用于建立表与表之间的关联关系，确保数据一致性和完整性。它可以确保子表中引用的数据必须存在于主表中，防止无效数据插入。同时，外键支持级联操作（如删除或更新时自动同步），简化了业务逻辑的实现，减少了手动维护数据关联的复杂性。
	支持向量搜索功能（实验特性，从 v8.4.0 开始引入）	向量搜索是一种基于数据语义的搜索方法，可以提供更相关的搜索结果。作为 AI 和大语

分类	功能	描述
		言模型 (LLM) 的核心功能之一，向量搜索可用于检索增强生成 (Retrieval-Augmented Generation, RAG)、语义搜索、推荐系统等多种场景。
数据库管理与可观测性	在内存表中显示 TiKV 和 TiDB 的 CPU 时间（从 v8.4.0 开始引入）	将 CPU 时间合入系统表中展示，与会话或 SQL 的其他指标并列，方便你从多角度对高 CPU 消耗的操作进行观测，提升诊断效率。尤其适用于诊断实例 CPU 飙升或集群读写热点等场景。
	按表或数据库维度聚合 TiKV 消耗的 CPU 时间（从 v8.4.0 开始引入）	当热点问题不是由个别 SQL 语句引起时，利用 Top SQL 中按表或者数据库聚合的 CPU 时间，能够协助用户快速发现造成热点的表或者应用程序，从而大大提升热点问题和 CPU 消耗问题的诊断效率。
	Backup & Restore (BR) 启用 AWS SDK for Rust 访问外部存储（从 v8.5.0 开始引入）	BR 使用 AWS Rust SDK 替换掉原有的 Rusoto 库，从 TiKV 访问 Amazon S3 等外部存储，以更好地兼容 AWS 的 IMDSv2 以及 EKS Pod Identity 等新特性。
安全	快照备份数据和日志备份数据支持客户端加密（从 v8.5.0 开始成为正式功能）	在上传快照备份和日志备份到备份存储之前，你可以对备份数据进行加密，确保数据在存储和传输过程中的安全性。

1.1 功能详情

1.1.1 可扩展性

- Schema 缓存可用的内存上限成为正式功能 (GA)，当表的数量达到几十万甚至上百万时，可以显示减少 Schema 元数据的内存占用 #50959 @tiancaiamao @wjhuang2016 @gmhdbjd @tangenta

  在一些 SaaS 场景下，当表的数量达到几十万甚至上百万时，Schema 元数据会占用较多的内存。开启该功能后，系统将使用 Least Recently Used (LRU) 算法缓存和淘汰相应的 Schema 元数据信息，有效减少内存占用。

  从 v8.4.0 开始，该功能默认开启，默认值为 536870912（即 512 MiB），你可以通过系统变量 tidb_schema_cache_size 按需调整。

  更多信息，请参考用户文档。

- 通过 Active PD Follower 提升 PD 上 Region 信息查询服务的扩展能力 (GA) #7431 @okJiang

  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 Active PD Follower 作为实验特性，以提升 PD 上 Region 信息查询服务的扩展能力。在 v8.5.0 中，该功能成为正式功能 (GA)。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。

1.1.2 性能

- TiDB 加速建表成为正式功能 (GA)，显著缩短数据迁移和集群初始化时间 #50052 @D3Hunter @gmhdbjd

  TiDB v7.6.0 引入加速建表功能作为实验特性，并通过系统变量 tidb_ddl_version 控制。从 v8.0.0 开始，该系统变量更名为 tidb_enable_fast_create_table。

  在 v8.5.0 中，TiDB 加速建表功能成为正式功能 (GA) 并默认开启。在数据迁移或集群初始化时，该功能支持快速创建百万级规模的表，从而显著缩短相关操作的耗时。

  更多信息，请参考用户文档。

- TiKV 支持 MVCC 内存引擎 (In-memory Engine, IME)，可加速需要扫描大量 MVCC 历史版本的查询 #16141 @SpadeA-Tang @glorv @overvenus

  当频繁更新记录或者需要 TiDB 保留较长时间的历史版本（例如 24 小时）数据时，堆积的 MVCC 版本会导致扫描性能下降。TiKV 的 MVCC 内存引擎可以将最新的 MVCC 版本缓存在内存中，并通过快速的 GC 机制删除内存中的历史版本，从而提升扫描性能。

  从 v8.5.0 开始，TiKV 引入 MVCC 内存引擎。当 TiKV 集群中的 MVCC 版本堆积导致扫描性能下降时，你可以通过设置 TiKV 参数 in-memory-engine.enable 来开启 TiKV MVCC 内存引擎，提升扫描性能。

  更多信息，请参考用户文档。

1.1.3 稳定性

- 支持限制 PD 处理请求的最大速率和并发度 #5739 @rleungx

当突然有大量请求发送到 PD 时，这些请求可能导致 PD 工作负载过高，进行影响 PD 性能表现。从 v8.5.0 开始，你可以使用 pd-ctl 来限制 PD 处理请求的最大速率和并发度，提升 PD 的稳定性。

更多信息，请参考用户文档。

1.1.4 SQL 功能

- 外键成为正式功能 (GA) #36982 @YangKeao @crazycs520

  在 v8.5.0 中，TiDB 的外键功能成为正式功能 (GA)，支持使用外键约束提升数据一致性和保障完整性。你可以轻松创建表间的外键关联，实现级联更新和删除操作，使得数据管理更加便捷。这一功能为复杂数据关联的应用场景提供了更好的支持。

  更多信息，请参考用户文档。

- 引入 ADMIN ALTER DDL JOBS 语法，支持在线修改 DDL 任务参数 #57229 @fzzf678 @tangenta

  从 v8.3.0 开始，TiDB 支持在会话级别设置变量 tidb_ddl_reorg_batch_size 和 tidb_ddl_reorg_worker_cnt，因此通过 Global 设置这两个变量已不再影响所有运行中的 DDL 任务。如需更改这些变量的值，需要先取消 DDL 任务，调整变量取值后再重新提交。

  从 v8.5.0 开始，引入了 ADMIN ALTER DDL JOBS 语句。你可以在线调整指定的 DDL 任务的变量值，以便灵活平衡资源消耗与性能，并将变更限定于单个任务，使影响范围更加可控。例如：

  - ADMIN ALTER DDL JOBS job_id THREAD = 8;：在线调整该 DDL 任务的 tidb_ddl_reorg_worker_cnt
  - ADMIN ALTER DDL JOBS job_id BATCH_SIZE = 256;：在线调整该 DDL 任务的 tidb_ddl_reorg_batch_size

- ADMIN ALTER DDL JOBS job_id MAX_WRITE_SPEED = '200MiB';：在线调整写入每个 TiKV 节点的索引数据流量大小

更多信息，请参考用户文档。

1.1.5 安全

- BR 支持在客户端进行快照备份数据和日志备份数据加密 (GA) #28640 #56433 @joccau @Tristan1900
  - 通过快照备份数据的客户端加密功能（在 TiDB v5.3.0 中以实验特性引入），你可以使用自定义的固定密钥在客户端加密备份数据。
  - 通过日志备份数据的客户端加密功能（在 TiDB v8.4.0 中以实验特性引入），你使用以下方式之一在客户端加密日志备份数据：
    - 使用自定义的固定密钥加密
    - 使用本地磁盘的主密钥加密
    - 使用 KMS（密钥管理服务）的主密钥加密

  从 v8.5.0 开始，这两个加密功能都成为了正式功能 (GA)，进一步增强了客户端数据的安全性。

  更多信息，请参考加密备份数据和加密日志备份数据。

- TiKV 静态加密支持 Google Cloud Key Management Service (Google Cloud KMS) (GA) #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。在 v8.0.0 中，TiKV 静态加密以实验特性的形式支持了基于 Google Cloud KMS 的主密钥管理。

  从 v8.5.0 起，基于 Google Cloud KMS 的静态加密成为正式功能 (GA)。要使用该功能，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。

1.2 兼容性变更

注意：

以下为从 v8.4.0 升级至当前版本 (v8.5.0) 所需兼容性变更信息。如果从 v8.3.0 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 行为变更

- 为了与 MySQL 兼容，在非严格模式下 (sql_mode = '') 向非 NULL 列插入 NULL 值会报错。 #55457 @joechenrh
- 不再支持 ALTER TABLE ... DROP FOREIGN KEY IF EXISTS ... 语句。 #56703 @YangKeao

1.2.2 系统变量

变量名	修改类型	描述
tidb_enable_fast_create_table	修改	经进一步的测试后，默认值从 OFF 修改为 ON，即默认开启 TiDB 加速建表。
tidb_ddl_reorg_max_write_speed	新增	限制每个 TiKV 节点写入的带宽，仅在开启添加索引加速功能时生效（由变量 tidb_ddl_enable_fast_reorg 控制）。例如，当该值设置为 200MiB 时，最大写入速度限制为 200 MiB/s。

1.2.3 配置参数

配置文件或组件	配置项	修改类型	描述
TiDB	deprecate-integer-display-length	修改	从 v8.5.0 开始，整数显示宽度功能已废弃，该配置项的默认值从 false 修改为 true。
TiKV	raft-client-queue-size	修改	默认值从 8192 修改为 16384。

配置文件或组件	配置项	修改类型	描述
PD	patrol-region-worker-count	新增	控制 checker 检查 Region 健康状态时，创建 operator 的并发数。
BR	--checksum	修改	默认值从 true 修改为 false，即 BR 进行全量备份时，默认不计算表级别的校验和，以提升备份性能。

1.3 操作系统支持变更

升级 TiDB 前，请务必确保你的操作系统版本符合操作系统及平台要求。

- 根据 CentOS Linux EOL，CentOS Linux 7 的上游支持已于 2024 年 6 月 30 日终止。从 v8.4.0 版本开始，TiDB 已结束对 CentOS 7 的支持，建议使用 Rocky Linux 9.1 及以上的版本。如果将运行在 CentOS 7 上的 TiDB 集群升级到 v8.4.0 或之后版本，将导致集群不可用。
- 根据 Red Hat Enterprise Linux Life Cycle，Red Hat Enterprise Linux 7 的 Maintenance Support 已于 2024 年 6 月 30 日终止。从 v8.4.0 版本开始，TiDB 已结束对 Red Hat Enterprise Linux 7 的支持，建议使用 Rocky Linux 9.1 及以上的版本。如果将运行在 Red Hat Enterprise Linux 7 上的 TiDB 集群升级到 v8.4.0 或之后版本，将导致集群不可用。

1.4 移除功能

- 以下为已移除的功能：
    - TiDB Binlog 在 v8.4.0 中被移除。从 v8.3.0 开始，TiDB Binlog 被完全废弃。如需进行增量数据同步，请使用 TiCDC。如需按时间点恢复 (point-in-time recovery, PITR)，请使用 PITR。在将 TiDB 集群升级到 v8.4.0 或之后版本前，务必先切换至 TiCDC 和 PITR。
- 以下为计划在未来版本中移除的功能：

    - 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。

1.5 废弃功能

以下为计划将在未来版本中废弃的功能：

- TiDB 在 v8.0.0 引入了系统变量 tidb_enable_auto_analyze_priority_queue，用于控制是否启用优先队列来优化自动收集统计信息任务的排序。在未来版本中，优先队列将成为自动收集统计信息任务的唯一排序方式，该系统变量将被废弃。
- TiDB 在 v7.5.0 引入了系统变量 tidb_enable_async_merge_global_stats，用于设置 TiDB 使用异步方式合并分区统计信息，以避免 OOM 问题。在未来版本中，分区统计信息将统一使用异步方式进行合并，该系统变量将被废弃。
- 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 在 v8.0.0 引入了系统变量 tidb_enable_parallel_hashagg_spill，用于控制 TiDB 是否支持并行 HashAgg 进行落盘。在未来版本中，该系统变量将被废弃。
- TiDB 在 v5.1 引入了系统变量 tidb_partition_prune_mode，用于设置是否开启分区表动态裁剪模式。从 v8.5.0 开始，将该变量设置为 static 或 static-only 时会产生警告。在未来版本中，该系统变量将被废弃。
- 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 v6.3.0 开始，分区表默认使用动态裁剪模式，相比静态裁剪模式，动态裁剪模式支持 IndexJoin、Plan Cache 等特性，性能表现更好。在未来版本中，静态裁剪模式将被废弃。

1.6 改进提升

- TiDB
    - 提升关闭分布式执行框架时，ADD INDEX 加速功能对任务取消的响应速度 #56017 @lance6716
    - 提升小表加索引的速度 #54230 @tangenta
    - 新增系统变量 tidb_ddl_reorg_max_write_speed，用于限制加索引时 ingest 阶段速度的上限 #57156 @CbcWestwolf
    - 提升某些情况下查询 information_schema.tables 的性能 #57295 @tangenta
    - 支持动态调整更多 DDL 任务参数 #57526 @fzzf678
    - 支持全局索引包含分区表达式中的所有列 #56230 @Defined2014
    - 支持 List 分区表在 Range 查询的场景中进行分区裁剪 #56673 @Defined2014
    - 默认开启 FixControl#46177，修复在某些情况下错误地选择了全表扫描，而没有选择索引范围扫描的问题 #46177 @terry1purcell
    - 改进内部估算逻辑，使其能够更充分地利用多列多值索引的统计信息，提升某些涉及多值索引的查询的估算精度 #56915 @time-and-fate
    - 提高特定情况下全表扫描的代价估算，减少错误地选择全表扫描的概率 #57085 @terry1purcell
    - 优化统计信息同步加载所需的数据量，提升加载性能 #56812 @winoros
    - 优化特定情况下，OUTER JOIN 含有唯一索引且有 ORDER BY ... LIMIT 子句时的执行计划，提高执行效率 #56321 @winoros
- TiKV

    - 利用单独的线程清理副本，保证 Raft 读写关键路径的延迟稳定 #16001 @hbisheng
    - 向量距离函数支持 SIMD 以提升性能 #17290 @EricZequan
- PD
    - 支持 tso 服务在微服务模式和非微服务模式之间动态切换 #8477 @rleungx
    - 优化 pd-ctl config 输出中部分字段的大小写 #8694 @lhy1024
    - Store limit v2 成为正式功能 (GA) #8865 @lhy1024
    - 支持配置 Region 巡检的并行度（实验特性）#8866 @lhy1024
- TiFlash
    - 提升聚簇索引表在后台回收过期数据的速度 #9529 @JaySon-Huang
    - 提升数据更新场景下向量搜索的查询性能 #9599 @Lloyd-Pottiger
    - 新增关于构建向量索引 CPU 使用率的监控指标 #9032 @JaySon-Huang
    - 提升逻辑运算符的执行效率 #9146 @windtalker
- Tools
    - Backup & Restore (BR)
        - 减少备份过程中无效日志的打印 #55902 @Leavrth
        - 优化加密密钥 --crypter.key 的错误提示信息 #56388 @Tristan1900
        - 增加 BR 创建数据库时的并发度，以提升数据恢复性能 #56866 @Leavrth
        - 在进行全量备份时，默认不计算表级别的 checksum (--checksum=false) 以提升备份性能 #56373 @Tristan1900
        - 新增对每个存储节点的连接超时进行独立追踪和重置的机制，增强了对慢节点的处理能力，从而避免备份卡住的问题 #57666 @3pointer
    - TiDB Data Migration (DM)

- 在 DM 集群启动时，增加 DM-worker 连接到 DM-master 的重试 #4287 @GMHDBJD

1.7 错误修复

- TiDB
    - 修复当从 PD 返回的 Region 元数据中未包含 Leader 信息时，TiDB 未自动重试请求可能导致执行报错的问题 #56757 @cfzjywxk
    - 修复写冲突时 TTL 任务可能无法取消的问题 #56422 @YangKeao
    - 修复取消 TTL 任务时，没有强制 Kill 对应 SQL 的问题 #56511 @lcwangchao
    - 修复集群从 v6.5 升级到 v7.5 或更高版本后，已有 TTL 任务执行意外频繁的问题 #56539 @lcwangchao
    - 修复 INSERT ... ON DUPLICATE KEY 语句不兼容 mysql_insert_id 的问题 #55965 @tiancaiamao
    - 修复 TTL 在未选用 TiKV 作为存储引擎时可能失败的问题 #56402 @YangKeao
    - 修复通过 IMPORT INTO 导入数据后，AUTO_INCREMENT 字段没有正确设置的问题 #56476 @D3Hunter
    - 修复执行 ADD INDEX 时，未检查索引长度限制的问题 #56930 @fzzf678
    - 修复执行 RECOVER TABLE BY JOB JOB_ID; 可能导致 panic 的问题 #55113 @crazycs520
    - 修复由于 stale read 未对读操作的时间戳进行严格校验，导致 TSO 和真实物理时间存在偏移，有小概率影响事务一致性的问题 #56809 @MyonKeminta
    - 修复 DDL owner 节点切换后，无法按照切换前的进度继续执行 Reorg DDL 任务的问题 #56506 @tangenta
    - 修复分布式执行框架的监控面板上部分指标不准确的问题 #57172 @fzzf678 #56942 @fzzf678

- 修复某些情况下 REORGANIZE PARTITION 无法显示报错原因的问题 #56634 @mjonss
- 修复查询 INFORMATION_SCHEMA.TABLES 时由于大小写敏感导致返回结果错误的问题 #56987 @joechenrh
- 修复当公共表表达式 (CTE) 有多个数据消费者时，如果某个消费者在未读取数据时就退出，可能导致非法内存访问的问题 #55881 @windtalker
- 修复 INDEX_HASH_JOIN 在异常退出时可能卡住的问题 #54055 @wshwsh12
- 修复 TRUNCATE 语句在处理 NULL 值时返回结果错误的问题 #53546 @tuziemon
- 修复由于类型推导错误导致 CAST AS CHAR 函数结果不正确的问题 #56640 @zimulala
- 修复由于类型推导错误导致部分函数的输出结果中字符串被截断的问题 #56587 @joechenrh
- 修复 ADDTIME() 和 SUBTIME() 函数的第一个参数为日期类型时返回错误结果的问题 #57569 @xzhangxian1008
- 修复在非严格模式下 (sql_mode = '') ，非法空值能被插入的问题 #56381 @joechenrh
- 修复 UPDATE 语句更新 ENUM 类型的值时更新错误的问题 #56832 @xhebox
- 修复开启 tidb_low_resolution_tso 变量后，执行 SELECT FOR UPDATE 语句出现资源泄漏的问题 #55468 @tiancaiamao
- 修复 JSON_TYPE() 函数未检查其参数类型导致传入非 JSON 数据类型时未报错的问题 #54029 @YangKeao
- 修复在 PREPARE 语句中使用 JSON 函数可能导致执行失败的问题 #54044 @YangKeao
- 修复将数据从 BIT 类型换为 CHAR 类型时可能导致 TiKV 崩溃的问题 #56494 @lcwangchao

- 修复在 CREATE VIEW 语句中使用变量或参数时未报错的问题 #53176 @mjonss
- 修复 JSON_VALID() 函数返回结果错误的问题 #56293 @YangKeao
- 修复关闭 tidb_ttl_job_enable 变量后 TTL 任务未被取消的问题 #57404 @YangKeao
- 修复同时使用 RANGE COLUMNS 分区函数和 utf8mb4_0900_ai_ci 排序规则时，查询结果错误的问题 #57261 @Defined2014
- 修复执行以换行符开头的 Prepared 语句会导致数组越界的运行错误 #54283 @Defined2014
- 修复 UTC_TIMESTAMP() 函数中的精度相关问题，比如精度设置得太大 #56451 @chagelo
- 修复 UPDATE、INSERT、DELETE IGNORE 语句中没有忽略外键错误的问题 #56678 @YangKeao
- 修复查询 information_schema.cluster_slow_query 表时，如果不加时间过滤条件，则只会查询最新的慢日志文件的问题 #56100 @crazycs520
- 修复 TTL 表的内存泄漏问题 #56934 @lcwangchao
- 修复 write_only 状态的表外键约束未生效的问题，以避免使用 non-public 状态的表 #55813 @YangKeao
- 修复使用 NATURAL JOIN 或者 USING 子句之后，再使用子查询可能会报错的问题 #53766 @dash12653
- 修复如果 CTE 包含 ORDER BY、LIMIT、SELECT DISTINCT 子句，并且被另外一个 CTE 的递归部分所引用时，可能被错误地 inline 导致执行报错的问题 #56603 @elsa0520
- 修复 VIEW 中定义的 CTE 被错误 inline 的问题 #56582 @elsa0520
- 修复使用 PLAN REPLAYER 导入含有外键的表结构时可能报错的问题 #56456 @hawkingrei

- 修复使用 PLAN REPLAYER 导入含有 Placement Rule 的表结构时可能报错的问题 #54961 @hawkingrei
- 修复使用 ANALYZE 收集表的统计信息时，如果该表包含虚拟生成列的表达式索引，执行会报错的问题 #57079 @hawkingrei
- 修复 DROP DATABASE 语句没有正确触发统计信息对应变更的问题 #57227 @Rustin170506
- 修复在 CTE 中解析数据库名时，返回错误的数据库名的问题 #54582 @hawkingrei
- 修复在使用 DUMP STATS 将统计信息转为 JSON 的过程中，直方图上下界数据受损的问题 #56083 @hawkingrei
- 修复 EXISTS 子查询的结果继续参与代数运算时，结果和 MySQL 不一致的问题 #56641 @windtalker
- 修复无法为带别名的多表删除语句 DELETE 创建执行计划绑定的问题 #56726 @hawkingrei
- 修复优化器在简化复杂谓词时，由于没有考虑字符集及排序规则，可能导致执行报错的问题 #56479 @dash12653
- 修复 Grafana 中 **Stats Healthy Distribution** 面板的数据可能错误的问题 #57176 @hawkingrei
- 修复在包含聚簇索引的表上执行向量搜索查询时，结果可能错误的问题 #57627 @winoros

- TiKV
  - 修复读线程在从 Raft Engine 中的 MemTable 读取过时索引时出现的 panic 问题 #17383 @LykxSassinator
  - 修复当大量事务在排队等待同一个 key 上的锁被释放且该 key 被频繁更新时，TiKV 可能因死锁检测压力过大而出现 OOM 的问题 #17394 @MyonKeminta
  - 修复资源管控后台任务 CPU 使用率可能被重复统计的问题 #17603 @glorv

- 修复由于 CDC 内部任务堆积过多导致 TiKV OOM 的问题 #17696 @3AceShowHand
- 修复 raft-entry-max-size 设置过高时，写入 batch 可能过大引起性能抖动的问题 #17701 @SpadeA-Tang
- 修复 Region Split 后可能无法快速选出 Leader 的问题 #17602 @LykxSassinator
- 修复使用 RADIANS() 或 DEGREES() 函数时可能导致 TiKV panic 的问题 #17852 @gengliqi
- 修复所有休眠的 Region 被集中唤醒时，可能导致写入抖动的问题 #17101 @hhwyt

- PD
  - 修复热点缓存中可能存在的内存泄露问题 #8698 @lhy1024
  - 修复资源组 (Resource Group) 选择器对所有面板都未生效的问题 #56572 @glorv
  - 修复已删除的资源组仍然出现在监控面板中的问题 #8716 @AndreMouche
  - 修复 Region syncer 加载过程中日志描述不清晰的问题 #8717 @lhy1024
  - 修复 label 统计中的内存泄露问题 #8700 @lhy1024
  - 修复配置 tidb_enable_tso_follower_proxy 为 0 或 OFF 时无法关闭 TSO Follower Proxy 特性的问题 #8709 @JmPotato
- TiFlash
  - 修复 SUBSTRING() 函数不支持部分整数类型的 pos 和 len 参数导致查询报错的问题 #9473 @gengliqi
  - 修复在存算分离架构下，扩容 TiFlash 写节点后向量搜索性能可能下降的问题 #9637 @kolafish
  - 修复当 SUBSTRING() 函数的第二个参数为负数时，可能返回错误结果的问题 #9604 @guo-shaoge

- 修复当 REPLACE() 函数的第一个参数为常数时，可能报错的问题 #9522 @guo-shaoge
- 修复当 LPAD() 和 RPAD() 函数在某些情况下返回错误结果的问题 #9465 @guo-shaoge
- 修复在创建向量索引后，如果 TiFlash 内部用于构建向量索引的任务被意外中断，可能造成 TiFlash 写入损坏数据并无法重启的问题 #9714 @JaySon-Huang

- Tools
  - Backup & Restore (BR)
    - 修复备份过程中还未备份完成的 range 空洞过多时出现 OOM 的问题，减少了预先分配的内存 #53529 @Leavrth
    - 修复备份时无法备份全局索引的问题 #57469 @Defined2014
    - 修复日志可能打印加密信息的问题 #57585 @kennytm
    - 修复 advancer 无法处理锁冲突的问题 #57134 @3pointer
    - 升级 k8s.io/api 库的版本以修复潜在的安全漏洞 #57790 @BornChanger
    - 修复当集群存在大量表但实际数据量较小时，PITR 数据恢复任务可能出现 Information schema is out of date 报错的问题 #57743 @Tristan1900
    - 修复日志备份在 advancer owner 切换时可能会异常进入暂停状态的问题 #58031 @3pointer
    - 修复通过 tiup br restore 命令进行库表级别恢复时，遗漏检查目标集群中表是否已存在，可能会覆盖已有表的问题 #58168 @RidRisR
  - TiCDC
    - 修复使用 Debezium 协议时 Kafka 消息中缺少 Key 的问题 #1799 @wk989898
    - 修复 redo 模块无法正确上报错误的问题 #11744 @CharlesCheung96

        - 修复 TiDB DDL owner 变更导致 DDL 任务的 schema 版本出现非递增时，TiCDC 错误丢弃 DDL 任务的问题 #11714 @wlwilliamx
    - TiDB Lightning
        - 修复 TiDB Lightning 因 TiKV 发送的消息过大而接收失败的问题 #56114 @fishiu
        - 修复使用物理导入模式导入数据后，AUTO_INCREMENT 值设置过大的问题 #56814 @D3Hunter

1.8 贡献者

感谢来自 TiDB 社区的贡献者们：

- dash12653（首次贡献者）
- chagelo（首次贡献者）
- LindaSummer
- songzhibin97
- Hexilee

2 TiDB 8.4.0 Release Notes

发版日期：2024 年 11 月 11 日

TiDB 版本：8.4.0

试用链接：快速体验 | 下载离线包

在 8.4.0 版本中，你可以获得以下关键特性：

分类	功能/增强	描述
可扩展性和性能	实例级执行计划缓存（实验特性）	实例级执行计划缓存允许同一个 TiDB 实例的所有会话共享执行计划缓存。与现有的会话级执行计划缓存相比，实例级执行计划缓存能够在内存中缓存更多执行计划，减少 SQL 编译时间，从而降低 SQL 整体运行时间，提升 OLTP 的性能和吞吐，同时更好地控制内存使用，提升数据库稳定性。

分类	功能/增强	描述
	分区表全局索引成为正式功能	全局索引可以有效提高检索非分区列的效率，并且消除了唯一键必须包含分区键的限制。该功能扩展了 TiDB 分区表的使用场景，避免了数据迁移过程中的一些应用修改工作。
	TiDB 并行获取 TSO	在高并发场景下，并行获取 TSO 能够有效降低等待获取 TSO 的时间，提升集群的吞吐。
	提升缓存表的查询性能	优化了缓存表索引扫描的查询性能，部分场景可提升 5.4 倍。在需要对小表进行高速查询的场景下，利用缓存表可大幅提升整体性能。
稳定性与高可用	Runaway Queries 支持更多触发条件，并能够切换资源组	Runaway Queries 提供了有效的手段来降低突发的 SQL 性能问题对系统产生的影响。v8.4.0 中新增 Coprocessor 处理的 Key 的数量 (PROCESSED_KEYS) 和 Request Unit (RU) 作为识别条件，并可以将识别到的查询置入指定资源组，对 Runaway Queries 进行更精确的识别与控制。
	支持为资源管控的后台任务设置资源使用上限	为资源管控的后台任务设置百分比上限，针对不同业务系统的需求，控制后台任务的消耗，从而将后台任务的消耗限制在一个很低的水平，保证在线业务的服务质量。
	TiProxy 流量捕获和回放（实验特性）	在进行集群升级、迁移或部署变更等重要操作之前，使用 TiProxy 捕获 TiDB 生产集群的真实负载，并在测试的目标集群中重现该工作负载，从而验证性能，确保变更成功。
	自动统计信息收集任务支持并发	使用系统变量 tidb_auto_analyze_concurrency 控制单个自动统计信息收集任务内部的并发度，TiDB 会根据节点规模和硬件规格自动确定扫描任务的并发度。该功能通过充分利用系统资源，提高统计信息收集效率，从而减少手动调优，并确保集群性能稳定。
SQL	支持向量搜索功能（实验特性）	向量搜索是一种基于数据语义的搜索方法，可以提供更相关的搜索结果。作为 AI 和大语言模型 (LLM) 的核心功能之一，向量搜索可用于检索增强生成 (Retrieval-Augmented Generation, RAG)、语义搜索、推荐系统等多种场景。
数据库管理和	在内存表中显示 TiKV 和 TiDB 的 CPU 时间	将 CPU 时间合入系统表中展示，与会话或 SQL 的其他指标并列，方便你从多角度对高 CPU 消耗的操作进行观测，提升诊断效率。尤其适用于诊断实例 CPU 飙升或集群读写热点等场景。

分类	功能/增强	描述
可观测性	按表或数据库维度聚合 TiKV 消耗的 CPU 时间	当热点问题不是由个别 SQL 语句引起时，利用 Top SQL 中按表或者数据库聚合的 CPU 时间，能够协助用户快速发现造成热点的表或者应用程序，从而大大提升热点问题和 CPU 消耗问题的诊断效率。
	支持对开启了 IMDSv2 服务的 TiKV 实例做备份	目前 AWS EC2 的默认元数据服务是 IMDSv2。TiDB 支持从开启了 IMDSv2 的 TiKV 实例中备份数据，协助你更好地在公有云服务中运行 TiDB 集群。
安全	日志备份数据支持客户端加密（实验特性）	在上传日志备份到备份存储之前，你可以对日志备份数据进行加密，确保数据在存储和传输过程中的安全性。

2.1 功能详情

2.1.1 性能

- 新增 TSO 请求的并行批处理模式，降低获取 TSO 的延迟 #54960 #8432 @MyonKeminta

  在 v8.4.0 之前，TiDB 向 PD 请求 TSO 时会将一段时间内的请求汇总起来并以串行的方式进行批处理，以减少 RPC (Remote Procedure Call) 请求数量，从而降低 PD 负载。对于延迟敏感的场景，这种串行模式的性能并不理想。

  在 v8.4.0 中，TiDB 新增 TSO 请求的并行批处理模式，并提供不同的并发能力。并行模式可以降低获取 TSO 的延迟，但可能会增加 PD 的负载。你可以通过 tidb_tso_client_rpc_mode 变量设定获取 TSO 的 RPC 模式。

  更多信息，请参考用户文档。

- 优化 TiDB Hash Join 算子的执行效率（实验特性）#55153 #53127 @windtalker @xzhangxian1008 @XuHuaiyu @wshwsh12

在 v8.4.0 中，TiDB 对 Hash Join 算子的实现方法进行了优化，以提升其执行效率。目前，优化版的 Hash Join 仅对 Inner Join 和 Outer Join 操作生效，且默认关闭。如需开启优化版的 Hash Join，可以将系统变量 tidb_hash_join_version 设置为 optimized。

更多信息，请参考用户文档。

- 支持下推以下日期函数到 TiKV #56297 #17529 @gengliqi
    - DATE_ADD()
    - DATE_SUB()
    - ADDDATE()
    - SUBDATE()

  更多信息，请参考用户文档。

- 支持实例级执行计划缓存（实验特性）#54057 @qw4990

  实例级执行计划缓存允许同一个 TiDB 实例上的所有会话共享执行计划缓存。该功能可以大幅降低 TiDB 的查询响应时间，提升集群吞吐，减少执行计划突变的可能性，并保持集群性能的稳定。相比会话级执行计划缓存，实例级执行计划缓存具有以下优势：

    - 消除冗余，在相同的内存消耗下缓存更多执行计划。
    - 在实例上分配固定大小的内存区域，更有效地限制内存使用。

  在 v8.4.0 中，实例级执行计划缓存仅支持对查询的执行计划进行缓存，且默认关闭。你可以通过系统变量 tidb_enable_instance_plan_cache 开启该功能，并通过系统变量 tidb_instance_plan_cache_max_size 设置其最大内存使用量。开启该功能之前，请关闭会话级别的 Prepare 语句执行计划缓存和非 Prepare 语句执行计划缓存。

  更多信息，请参考用户文档。

- TiDB Lightning 的逻辑导入模式支持预处理语句和客户端语句缓存 #54850 @dbsid

通过开启配置项 logical-import-prep-stmt，TiDB Lightning 逻辑导入模式中执行的 SQL 语句将通过使用预处理语句和客户端语句缓存，降低 TiDB SQL 解析和编译的成本，提升 SQL 执行效率，并有更大机会命中执行计划缓存，提升逻辑导入的速度。

更多信息，请参考用户文档。

- 分区表的全局索引成为正式功能 (GA) #45133 @mjonss @Defined2014 @jiyfhust @L-maple

  之前版本的分区表，因为不支持全局索引有较多的限制，比如唯一键必须包含分区表达式中用到的所有列，如果查询条件不带分区键，查询时会扫描所有分区，导致性能较差。从 v7.6.0 开始，引入了系统变量 tidb_enable_global_index 用于开启全局索引特性，但该功能当时处于开发中，不够完善，不建议开启。

  从 v8.3.0 开始，全局索引作为实验特性正式发布。你可通过关键字 GLOBAL 为分区表显式创建一个全局索引，从而去除分区表唯一键必须包含分区表达式中用到的所有列的限制，满足灵活的业务需求。同时基于全局索引也提升了非分区列的查询性能。

  在 v8.4.0 中，全局索引成为正式功能 (GA)。你无需再设置系统变量 tidb_enable_global_index 开启全局索引特性，可以直接使用关键字 GLOBAL 创建全局索引。从 v8.4.0 开始，该系统变量被废弃，并总是设置为 ON。

  更多信息，请参考用户文档。

- 优化缓存表在部分场景下的查询性能 #43249 @tiancaiamao

  优化缓存表的查询性能，在使用 IndexLookup 执行 SELECT ... LIMIT 1 时，性能最高提升 5.4 倍。同时，提升 IndexLookupReader 在全表扫描和主键查询场景下的性能。

2.1.2 稳定性

- 超出预期的查询 (Runaway Queries) 新增处理行数和 Request Unit 作为阈值 #54434 @HuSharp

  从 v8.4.0 开始， TiDB 可以依据处理行数 (PROCESSED_KEYS) 和 Request Unit (RU) 定义超出预期的查询。和执行时间 (EXEC_ELAPSED) 相比，新增阈值能够更准确地定义查询的资源消耗，避免整体性能下降时发生识别偏差。

  支持同时设置多个条件，满足任意条件即识别为 Runaway Queries。

  可以观测 Statement Summary Tables 中的几个对应字段 (RESOURCE_GROUP、MAX_REQUEST_UNIT_WRITE、MAX_REQUEST_UNIT_READ、MAX_PROCESSED_KEYS)，根据历史执行情况决定条件值的大小。

  更多信息，请参考用户文档。

- 超出预期的查询 (Runaway Queries) 支持切换资源组 #54434 @JmPotato

  v8.4.0 新增支持将 Runaway Queries 切换到指定资源组。在降低优先级 (COOLDOWN) 仍旧无法有效降低资源消耗的情况下，你可以创建一个资源组 (Resource Group)并限制其资源上限，通过配置参数 SWITCH_GROUP 指定将识别到的查询切换到该资源组中，会话的后续查询仍在原资源组中执行。切换资源组的行为能够更精确地限制资源使用，对 Runaway Queries 的资源消耗做更加严格的控制。

  更多信息，请参考用户文档。

- 支持使用系统变量 tidb_scatter_region 设置集群级别的 Region 打散策略 #55184 @D3Hunter

  在 v8.4.0 之前，系统变量 tidb_scatter_region 仅支持设置为开启或者关闭。开启后，建表时会使用表级别打散策略。在批量快速建表，且表的数量达到

几十万张后，该策略会导致 Region 集中分布在其中几个 TiKV 节点，导致这些 TiKV 节点 OOM。

从 v8.4.0 开始，该系统变量改为字符串类型，且新增支持集群级别的打散策略，避免上述场景下导致 TiKV OOM 的问题。

更多信息，请参考用户文档。

- 支持为资源管控的后台任务设置资源上限 #56019 @glorv

  TiDB 资源管控能够识别并降低后台任务的运行优先级。在部分场景下，即使有空闲资源，用户也希望后台任务消耗能够控制在很低的水平。从 v8.4.0 开始，你可以使用参数 UTILIZATION_LIMIT 为资源管控的后台任务设置最大可以使用的资源百分比，每个节点把所有后台任务的使用量控制在这个百分比以下。该功能可以让你精细控制后台任务的资源占用，进一步提升集群稳定性。

  更多信息，请参考用户文档。

- 优化资源组资源分配策略 #50831 @nolouch

  TiDB 在 v8.4.0 部分调整了资源分配策略，更好地满足用户对资源管控的预期。

  - 控制大查询在运行时的资源分配，避免超出资源组限额。配合 Runaway Queries 的 COOLDOWN 动作，识别并降低大查询并发度，降低瞬时资源消耗。
  - 调整默认的优先级调度策略。当不同优先级的任务同时运行时，高优先级的任务获得更多资源。

2.1.3 高可用

- TiProxy 支持流量回放功能（实验特性）#642 @djshow832

从 TiProxy v1.3.0 开始，你可以使用 tiproxyctl 连接 TiProxy 实例，捕获 TiDB 生产集群中的访问流量，并在测试集群中按照指定的速率回放这些流量。通过该功能，你可以在测试环境中重现生产集群的实际工作负载，从而验证所有 SQL 的执行结果和性能表现。

流量回放适用于以下场景：

- TiDB 版本升级前验证
- 执行变更前影响评估
- TiDB 扩缩容前性能验证
- 集群性能上限测试

更多信息，请参考用户文档。

2.1.4 SQL 功能

- 支持向量搜索功能（实验特性）#54245 #17290 #9032 @breezewish @Lloyd-Pottiger @EricZequan @zimulala @JaySon-Huang @winoros @wk989898

  向量搜索是一种基于数据语义的搜索方法，可以提供更相关的搜索结果。作为 AI 和大语言模型 (LLM) 的核心功能之一，向量搜索可用于检索增强生成 (Retrieval-Augmented Generation, RAG)、语义搜索、推荐系统等多种场景。

  从 v8.4.0 开始，TiDB 支持向量数据类型和向量搜索索引，具备强大的向量搜索能力。TiDB 的向量数据类型最多可支持 16383 维度，并支持多种距离函数，包括 L2 距离（欧式距离）、余弦距离、负内积和 L1 距离（曼哈顿距离）。

  在使用时，你只需要创建包含向量数据类型的表，并插入向量数据，即可执行向量搜索查询，也可进行向量数据与传统关系数据的混合查询。

此外，你可以创建并利用向量搜索索引来提升向量搜索的性能。需要注意的是，TiDB 的向量搜索索引依赖于 TiFlash。在使用向量搜索索引之前，需要确保 TiDB 集群中已部署 TiFlash 节点。

更多信息，请参考用户文档。

2.1.5 数据库管理

- 日志备份数据支持客户端加密（实验特性）#55834 @Tristan1900

  在之前的版本中，仅快照备份数据支持客户端加密。从 v8.4.0 起，日志备份数据也支持客户端加密。在上传日志备份到备份存储之前，你可以选择以下方式之一对日志备份数据进行加密，从而确保备份数据的安全性：

  - 使用自定义的固定密钥加密
  - 使用本地磁盘的主密钥加密
  - 使用 KMS（密钥管理服务）的主密钥加密

  更多信息，请参考用户文档。

- BR 降低了从云存储服务系统恢复数据的权限要求 #55870 @Leavrth

  在 v8.4.0 之前，BR 在恢复过程中会将恢复进度的检查点信息写入到备份存储系统。当恢复过程出现中断时，这些检查点使中断的恢复操作能够快速恢复。从 v8.4.0 开始，BR 将恢复检查点信息写入到目标 TiDB 集群中。这意味着 BR 在恢复时只需要具备对备份目录的读取权限。

  更多信息，请参考用户文档。

2.1.6 可观测性

- 在系统表中显示 TiDB 和 TiKV 消耗的 CPU 时间 #55542 @yibin87

  TiDB Dashboard 的 Top SQL 页面能够展示 CPU 消耗高的 SQL 语句。从 v8.4.0 开始，TiDB 将 CPU 时间消耗信息加入系统表展示，与会话或 SQL

的其他指标并列，方便你从多角度对高 CPU 消耗的操作进行观测。在实例 CPU 飙升或集群读写热点的场景下，这些信息能够协助你快速发现问题的原因。

  - STATEMENTS_SUMMARY 增加 AVG_TIDB_CPU_TIME 和 AVG_TIKV_CPU_TIME，显示单个 SQL 语句在历史上消耗 CPU 的平均时间。
  - INFORMATION_SCHEMA.PROCESSLIST 增加 TIDB_CPU 和 TIKV_CPU，显示会话当前正在执行 SQL 的累计 CPU 时间消耗。
  - 慢日志中增加字段 Tidb_cpu_time 和 Tikv_cpu_time，显示被捕捉到的 SQL 语句消耗 CPU 的时间。

  其中，TiKV 的 CPU 时间默认显示。采集 TiDB 的 CPU 时间会引入额外开销（约 8%），因此仅在开启 Top SQL 特性时，TiDB 的 CPU 时间才会显示为实际值，否则始终显示为 0。

  更多信息，请参考 INFORMATION_SCHEMA.PROCESSLIST 和 INFORMATION_SCHEMA.SLOW_QUERY。

- Top SQL 支持按表或数据库维度查看 CPU 时间的聚合结果 #55540 @nolouch

  在 v8.4.0 之前，Top SQL 以 SQL 为单位来聚合 CPU 时间。如果 CPU 时间不是由少数几个 SQL 贡献，按 SQL 聚合并不能有效发现问题。从 v8.4.0 开始，你可以选择 **By TABLE** 或者 **By DB** 聚合 CPU 时间。在多系统融合的场景下，新的聚合方式能够更有效地识别来自某个特定系统的负载变化，提升问题诊断的效率。

  更多信息，请参考用户文档。

2.1.7 安全

- BR 支持 AWS IMDSv2 #16443 @pingyu

在 Amazon EC2 上部署 TiDB 时，BR 支持 AWS 的 Instance Metadata Service Version 2 (IMDSv2)。你可以在 EC2 实例上进行相关配置，使 BR 可以使用与实例关联的 IAM 角色以适当的权限访问 Amazon S3。

更多信息，请参考用户文档。

2.1.8 数据迁移

- TiCDC Claim-Check 支持仅发送 Kafka 消息的 value 部分到外部存储 #11396 @3AceShowHand

  在 v8.4.0 之前，如果开启了 Claim-Check 功能 （将 large-message-handle-option 设置为 claim-check），TiCDC 在处理大型消息时会将 key 和 value 都进行编码并存储在外部存储系统中。

  从 v8.4.0 开始，TiCDC 支持仅将 Kafka 消息的 value 部分发送到外部存储，该功能仅适用于非 Open Protocol 协议。你可以通过设置 claim-check-raw-value 参数控制是否开启该功能。

  更多信息，请参考用户文档。

- TiCDC 引入 Checksum V2 算法校验 Update 或 Delete 事件中 Old Value 数据 #10969 @3AceShowHand

  从 v8.4.0 开始，TiDB 和 TiCDC 引入 Checksum V2 算法，解决了 Checksum V1 在执行 ADD COLUMN 或 DROP COLUMN 后无法正确校验 Update 或 Delete 事件中 Old Value 数据的问题。对于 v8.4.0 及之后新创建的集群，或从之前版本升级到 v8.4.0 的集群，启用单行数据 Checksum 正确性校验功能后，TiDB 默认使用 Checksum V2 算法进行 Checksum 计算和校验。TiCDC 支持同时处理 V1 和 V2 两种 Checksum。该变更仅影响 TiDB 和 TiCDC 内部实现，不影响下游 Kafka consumer 的 Checksum 计算校验方法。

  更多信息，请参考用户文档。

2.2 兼容性变更

注意：

以下为从 v8.3.0 升级至当前版本 (v8.4.0) 所需兼容性变更信息。如果从 v8.2.0 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 系统变量

变量名	修改类型	描述
log_bin	删除	从 v8.4.0 开始，TiDB Binlog 被移除。该变量表示是否使用 TiDB Binlog，从 v8.4.0 开始被删除。
sql_log_bin	删除	从 v8.4.0 开始，TiDB Binlog 被移除。该变量表示是否将更改写入 TiDB Binlog，从 v8.4.0 开始被删除。
tidb_enable_global_index	废弃	从 v8.4.0 开始，该变量被废弃。其值将固定为默认值 ON，即默认启用全局索引。你只需在执行 CREATE TABLE 或 ALTER TABLE 时给对应的列加上关键字 GLOBAL 即可创建全局索引。
tidb_enable_list_partition	废弃	从 v8.4.0 开始，该变量被废弃。其值将固定为默认值 ON，即默认启用 List 分区。
tidb_enable_table_partition	废弃	从 v8.4.0 开始，该变量被废弃。其值将固定为默认值 ON，即默认启用分区表。
tidb_analyze_partition_concurrency	修改	取值范围从 [1, 1844674407370955161 5] 修改为 [1, 128]。
tidb_enable_inl_join_inner_multi_pattern	修改	默认值从 OFF 修改为 ON。从 v8.4.0 开始，当内表上有 Selection、Projection 或 Aggregation 算子时，默认支持 Index Join。
tidb_opt_prefer_range_scan	修改	默认值从 OFF 修改为 ON。对于没有统计信息的表（伪统计信息）或空表（零统计信息），优化器将优先选择区间扫描而不是全表扫描。
tidb_scatter_region	修改	在 v8.4.0 之前，该变量为布尔型，仅支持开启或关闭，且开启后新建表的 Region 只支持表级别打散。从

变量名	修改类型	描述
		v8.4.0 开始，增加 SESSION 作用域，类型由布尔型变更为枚举型，默认值由原来的 OFF 变更为空，表示不打散表 Region，并增加了可选值 TABLE 和 GLOBAL。支持集群级别的打散策略，避免快速批量建表时由于 Region 分布不均匀导致 TiKV OOM 的问题。
tidb_schema_cache_size	修改	默认值从 0 修改为 536870912（即 512 MiB），表示默认开启该功能，且最小值允许设置为 67108864（即 64 MiB）。
tidb_auto_analyze_concurrency	新增	设置单个自动统计信息收集任务内部的并发度。在 v8.4.0 之前，该并发度固定为 1。你可以根据集群资源情况提高该并发度，从而加快统计信息收集任务的执行速度。
tidb_enable_instance_plan_cache	新增	控制是否开启 Instance Plan Cache 功能。
tidb_enable_stats_owner	新增	设置该 TiDB 实例是否可以运行统计信息自动更新任务。
tidb_hash_join_version	新增	控制 TiDB 是否使用 Hash Join 算子的优化版。默认值为 legacy，表示不使用优化版。如果设置为 optimized，TiDB 在执行 Hash Join 算子时将使用其优化版，以提升 Hash Join 性能。
tidb_instance_plan_cache_max_size	新增	设置 Instance Plan Cache 的最大内存使用量。
tidb_instance_plan_cache_reserved_percentage	新增	控制内存驱逐后 Instance Plan Cache 的空闲内存百分比。
tidb_pre_split_regions	新增	在 v8.4.0 之前，要设置新建表默认的行分裂分片数，需要在每个 CREATE TABLE SQL 语句里声明 PRE_SPLIT_REGIONS，一旦需要同样配置的表数量较多，操作复杂。为解决这些问题，引入了该变量。你可以在 GLOBAL 或 SESSION 级别设置该系统变量，提升易用性。

变量名	修改类型	描述
tidb_shard_row_id_bits	新增	在 v8.4.0 之前，要设置新建表默认的行 ID 的分片数，需要在每个 CREATE TABLE 或 ALTER TABLE 的 SQL 语句里声明 SHARD_ROW_ID_BITS，一旦需要同样配置的表数量较多，操作复杂。为解决这些问题，引入了该变量。你可以在 GLOBAL 或 SESSION 级别设置该系统变量，提升易用性。
tidb_tso_client_rpc_mode	新增	设置 TiDB 向 PD 发送 TSO RPC 请求时使用的模式。这里的模式将用于控制 TSO RPC 请求是否并行，调节获取 TS 时消耗在请求攒批阶段的时间，从而在某些场景中减少执行查询时等待 TS 阶段的时间。

2.2.2 配置参数

配置文件或组件	配置项	修改类型	描述
TiDB	grpc-keepalive-time	修改	增加最小值 1。
TiDB	grpc-keepalive-timeout	修改	在 v8.4.0 之前，该参数为 INT 类型，且最小值仅支持设置为 1。从 v8.4.0 开始，数据类型修改为 FLOAT64，且最小值支持设置为 0.05。在网络抖动比较频繁的场景中可以适当调小该值，通过减少重试间隔，来减少网络抖动带来的性能影响。
TiDB	tidb_enable_stats_owner	新增	表示该 tidb-server 是否可以运行统计信息自动更新任务。
TiKV	region-split-keys	修改	默认值从 "960000" 修改为 "2560000"。
TiKV	region-split-size	修改	默认值从 "96MiB" 修改为 "256MiB"。
TiKV	sst-max-size	修改	默认值从 "144MiB" 修改为 "384MiB"。

配置文件或组件	配置项	修改类型	描述
TiKV	pessimistic-txn.in-memory-instance-size-limit	新增	控制单个 TiKV 实例内存悲观锁的内存使用上限。超过此限制时，悲观锁将回退到持久化方式写入磁盘。
TiKV	pessimistic-txn.in-memory-peer-size-limit	新增	控制单个 Region 内存悲观锁的内存使用上限。超过此限制时，悲观锁将回退到持久化方式写入磁盘。
TiKV	raft-engine.spill-dir	新增	指定 TiKV 实例存储 Raft 日志文件的辅助目录，用于支持多盘存储 Raft 日志文件。
TiKV	resource-control.priority-ctl-strategy	新增	配置低优先级任务的管控策略。TiKV 通过对低优先级的任务添加流量控制来确保优先执行更高优先级的任务。
PD	cert-allowed-cn	修改	从 v8.4.0 开始，支持设置多个 Common Name。在 v8.4.0 之前，只能设置一个 Common Name。
PD	max-merge-region-keys	修改	默认值从 200000 修改为 540000。
PD	max-merge-region-size	修改	默认值从 20 修改为 54。
TiFlash	storage.format_version	修改	TiFlash 底层存储格式的默认版本从 5 修改为 7，以支持向量索引的构建与存储。由于该格式修改，升级 TiFlash 到 v8.4.0 或更高版本后，不支持原地降级到之前的版本。
TiDB Binlog	--enable-binlog	删除	从 v8.4.0 开始，TiDB Binlog 被移除。该参数用于开启或关闭

配置文件或组件	配置项	修改类型	描述
			TiDB 中 binlog 的生成，从 v8.4.0 开始被删除。
TiCDC	claim-check-raw-value	新增	控制 TiCDC 是否仅将 Kafka 消息的 value 部分发送到外部存储，该功能仅适用于非 Open Protocol 协议。
TiDB Lightning	logical-import-prep-stmt	新增	在逻辑导入模式下，该参数控制是否使用预处理语句和语句缓存来提高性能。默认值为 false。
BR	--log.crypter.key	新增	设置日志备份数据的加密密钥，十六进制字符串格式，aes128-ctr 对应 128 位（16 字节）密钥长度，aes192-ctr 为 24 字节，aes256-ctr 为 32 字节。
BR	--log.crypter.key-file	新增	设置日志备份数据的密钥文件，可直接将存放密钥的文件路径作为参数传入，此时 log.crypter.key 不需要配置。
BR	--log.crypter.method	新增	设置日志备份数据的加密算法，支持 aes128-ctr、aes192-ctr 和 aes256-ctr 三种算法，缺省值为 plaintext，表示不加密。
BR	--master-key	新增	设置日志备份数据的主密钥，可以是基于本地磁盘的主密钥或基于云 KMS (Key Management Service) 的主密钥。
BR	--master-key-crypter-method	新增	设置日志备份数据基于主密钥的加密算法，支持 aes128-ctr、aes192-ctr 和 aes256-ctr 三种算法，缺省值为 plaintext，表示不加密。

2.3 离线包变更

从 v8.4.0 开始，TiDB-community-toolkit 二进制软件包中移除了以下内容：

- pump-{version}-linux-{arch}.tar.gz
- drainer-{version}-linux-{arch}.tar.gz
- binlogctl
- arbiter

2.4 操作系统支持变更

升级 TiDB 前，请务必确保你的操作系统版本符合操作系统及平台要求。

- 根据 CentOS Linux EOL，CentOS Linux 7 的上游支持已于 2024 年 6 月 30 日终止。从 v8.4.0 版本开始，TiDB 已结束对 CentOS 7 的支持，建议使用 Rocky Linux 9.1 及以上的版本。如果将运行在 CentOS 7 上的 TiDB 集群升级到 v8.4.0 或之后版本，将导致集群不可用。
- 根据 Red Hat Enterprise Linux Life Cycle，Red Hat Enterprise Linux 7 的 Maintenance Support 已于 2024 年 6 月 30 日终止。从 v8.4.0 版本开始，TiDB 已结束对 Red Hat Enterprise Linux 7 的支持，建议使用 Rocky Linux 9.1 及以上的版本。如果将运行在 Red Hat Enterprise Linux 7 上的 TiDB 集群升级到 v8.4.0 或之后版本，将导致集群不可用。

2.5 移除功能

- 以下为从 v8.4.0 开始已移除的功能：
    - TiDB Binlog 在 v8.4.0 中被移除。从 v8.3.0 开始，TiDB Binlog 被完全废弃。如需进行增量数据同步，请使用 TiCDC。如需按时间点恢复 (point-in-time recovery, PITR)，请使用 PITR。在将 TiDB 集群升级到 v8.4.0 或之后版本前，务必先切换至 TiCDC 和 PITR。
- 以下为计划将在未来版本中移除的功能：
    - 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。

2.6 废弃功能

以下为计划将在未来版本中废弃的功能：

- TiDB 在 v8.0.0 引入了系统变量 tidb_enable_auto_analyze_priority_queue，用于控制是否启用优先队列来优化自动收集统计信息任务的排序。在未来版本中，优先队列将成为自动收集统计信息任务的唯一排序方式，系统变量 tidb_enable_auto_analyze_priority_queue 将被废弃。
- TiDB 在 v7.5.0 引入了系统变量 tidb_enable_async_merge_global_stats，用于设置 TiDB 使用异步方式合并分区统计信息，以避免 OOM 问题。在未来版本中，分区统计信息将统一使用异步方式进行合并，系统变量 tidb_enable_async_merge_global_stats 将被废弃。
- 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 在 v8.0.0 引入了系统变量 tidb_enable_parallel_hashagg_spill，用于控制 TiDB 是否支持并行 HashAgg 进行落盘。在未来版本中，系统变量 tidb_enable_parallel_hashagg_spill 将被废弃。
- 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 v6.3.0 开始，分区表默认使用动态裁剪模式，相比静态裁剪模式，动态裁剪模式支持 IndexJoin、Plan Cache 等特性，性能表现更好。在未来版本中，静态裁剪模式将被废弃。

2.7 改进提升

- TiDB

- 优化扫描大量数据时构造 BatchCop Task 的效率 #55915 #55413 @wshwsh12
- 优化事务的缓存，以降低事务中的写操作延时与 TiDB CPU 使用 #55287 @you06
- 优化系统变量 tidb_dml_type 为 "bulk" 时 DML 语句的执行性能 #50215 @ekexium
- 支持使用 Optimizer Fix Control 47400 控制优化器是否将 estRows 的最小值限制为 1，与 Oracle 和 DB2 等数据库的行为保持一致 #47400 @terry1purcell
- 为日志表 mysql.tidb_runaway_queries 增加写入控制，降低大量并发写入引发的开销 #54434 @HuSharp
- 当内表上有 Selection、Projection 或 Aggregation 算子时默认支持 Index Join #47233 @winoros
- 在某些场景下减少 DELETE 操作从 TiKV 获取的列信息数量，降低 DELETE 操作的资源开销 #38911 @winoros
- 支持通过系统变量 tidb_auto_analyze_concurrency 设置单个自动统计信息收集任务内部的并发度 #53460 @hawkingrei
- 优化一个内部函数的处理逻辑，提升查询大量列的表时的性能 #52112 @Rustin170506
- 支持将形如 a = 1 AND (a > 1 OR (a = 1 AND b = 2)) 的过滤条件简化为 a = 1 AND b = 2 #56005 @ghazalfamilyusa
- 在选中不优执行计划风险较高的场景中，提高代价模型中全表扫描的代价，使得优化器更倾向于使用索引 #56012 @terry1purcell
- TiDB 支持 MID() 函数的两参数版本，即 MID(str, pos) #52420 @dveeden
- 支持对主键为非 binary 类型的表拆分 TTL 任务 #55660 @lcwangchao
- 优化系统元数据相关语句性能 #50305 @ywqzzy @tangenta @joechenrh @CbcWestwolf

- 采用新的优先级队列处理自动收集统计信息操作，以提高收集性能并减少重建队列的开销 #55906 @Rustin170506
- 引入 DDL 通知程序，允许统计信息模块订阅 DDL 事件 #55722 @fzzf678 @lance6716 @Rustin170506
- TiDB 升级期间强制新版 TiDB 节点接管 DDL Owner，避免旧版本 TiDB 节点接管引发的兼容性问题 #51285 @wjhuang2016
- 支持集群级别的 Scatter Region 打散 #8424 @River2000i

- TiKV
  - Region 的默认值由 96 MiB 提升到 256 MiB，避免 Region 数量过多导致额外开销 #17309 @LykxSassinator
  - 支持指定单个 Region 或 TiKV 实例的内存悲观锁的内存上限，在热点写导致大量悲观锁加锁时，可以通过修改配置提高内存上限，避免悲观锁落盘导致的 CPU/IO 开销 #17542 @cfzjywxk
  - Raft Engine 新增 spill-dir 配置，支持 Raft 日志的多磁盘存储。当主目录 dir 所在磁盘的容量不足时，Raft Engine 会自动将新日志写入 spill-dir，从而确保系统的持续运行 #17356 @LykxSassinator
  - 优化存在大量 DELETE 版本时 RocksDB 的 compaction 触发机制，以加快磁盘空间回收 #17269 @AndreMouche
  - 支持在线更改写入流量控制 (flow-control) 的相关配置 #17395 @glorv
  - 优化空表和小 Region 场景下 Region Merge 的速度 #17376 @LykxSassinator
  - Pipelined DML 不会长时间阻塞 resolved-ts #17459 @ekexium
- PD
  - 支持 TiKV 节点在 TiDB Lightning 导入数据期间优雅下线 (graceful offline) #7853 @okJiang
  - 在 pd-ctl 命令中将 scatter-range 重命名为 scatter-range-scheduler #8379 @okJiang

    - 为 grant-hot-leader-scheduler 添加冲突检测 #4903 @lhy1024
- TiFlash
    - 优化 LENGTH() 和 ASCII() 函数执行效率 #9344 @xzhangxian1008
    - 减少处理存算分离请求时创建的线程数，避免 TiFlash 计算节点在处理大量请求时崩溃 #9334 @JinheLin
    - 改进 Pipeline Model 执行模型下任务的等待机制 #8869 @SeaRise
    - 改进 JOIN 算子的取消机制，使得 JOIN 算子内部能及时响应取消请求 #9430 @windtalker
- Tools
    - Backup & Restore (BR)
        - 当集群的 split-table 和 split-region-on-table 配置项为默认值 false 时，BR 在恢复数据到该集群的过程中不会按照 table 分裂 Region，以提升恢复速度 #53532 @Leavrth
        - 默认不支持使用 SQL 语句 RESTORE 全量恢复数据到非空集群 #55087 @BornChanger

2.8 错误修复

- TiDB
    - 修复当 tidb_restricted_read_only 变量设置为 true 时可能死锁的问题 #53822 #55373 @Defined2014
    - 修复 TiDB 优雅关闭时不等待 auto commit 事务完成的问题 #55464 @YangKeao
    - 修复在 TTL 任务执行过程中，减小 tidb_ttl_delete_worker_count 的值导致任务无法完成的问题 #55561 @lcwangchao
    - 修复当一张表的索引中包含生成列时，通过 ANALYZE 语句收集这张表的统计信息时可能报错 Unknown column 'column_name' in 'expression' 的问题 #55438 @hawkingrei

- 废弃统计信息相关的无用配置，减少冗余代码 #55043 @Rustin170506
- 修复执行一条包含关联子查询和 CTE 的查询时，TiDB 可能卡住或返回错误结果的问题 #55551 @guo-shaoge
- 修复禁用 lite-init-stats 可能导致统计信息同步加载失败的问题 #54532 @hawkingrei
- 修复当 UPDATE 或 DELETE 语句包含递归的 CTE 时，语句可能报错或不生效的问题 #55666 @time-and-fate
- 修复当一条 SQL 绑定涉及窗口函数时，有一定概率不生效的问题 #55981 @winoros
- 修复统计信息初始化时，使用非二进制排序规则的字符串类型列的统计信息可能无法正常加载的问题 #55684 @winoros
- 修复当查询条件为 column IS NULL 访问唯一索引时，优化器将行数错误地估算为 1 的问题 #56116 @hawkingrei
- 修复当查询包含形如 (... AND ...) OR (... AND ...) ... 的过滤条件时，优化器没有使用最优的多列统计信息估算行数的问题 #54323 @time-and-fate
- 修复当一个查询有索引合并 (Index Merge) 执行计划可用时，read_from_storage hint 可能不生效的问题 #56217 @AilinKid
- 修复 IndexNestedLoopHashJoin 中存在数据竞争的问题 #49692 @solotzg
- 修复 INFORMATION_SCHEMA.STATISTICS 表中 SUB_PART 值为空的问题 #55812 @Defined2014
- 修复 DML 语句中包含嵌套的生成列时报错的问题 #53967 @wjhuang2016
- 修复带有最小显示宽度的 integer 类型的数据参与除法运算时，可能导致除法结果溢出的问题 #55837 @windtalker
- 修复 TopN 算子之后的算子无法在内存超限时触发回退操作的问题 #56185 @xzhangxian1008

- 修复 Sort 算子中的 ORDER BY 列如果包含常量会卡住的问题 #55344 @xzhangxian1008
- 修复在添加索引期间，kill PD leader 后出现 8223 (HY000) 报错，且表中数据不一致的问题 #55488 @tangenta
- 修复当请求历史 DDL 任务信息时，DDL 历史任务过多导致 OOM 的问题 #55711 @joccau
- 修复当 Region 大小超过 96 MiB 时，启动全局排序后执行 IMPORT INTO 卡住的问题 #55374 @lance6716
- 修复在临时表上执行 IMPORT INTO 会导致 TiDB crash 的问题 #55970 @D3Hunter
- 修复添加唯一索引出现 duplicate entry 报错的问题 #56161 @tangenta
- 修复当 TiKV 停机超过 810 秒后时，TiDB Lightning 未 ingest 所有 KV 对，导致表中数据不一致的问题 #55808 @lance6716
- 修复无法对缓存表使用 CREATE TABLE LIKE 语句的问题 #56134 @tiancaiamao
- 修复 CTE 中 FORMAT() 表达式的警告信息混乱的问题 #56198 @dveeden
- 修复 CREATE TABLE 与 ALTER TABLE 建立分区表时对列的类型限制不一致的问题 #56094 @mjonss
- 修复 INFORMATION_SCHEMA.RUNAWAY_WATCHES 表中时间类型不正确的问题 #54770 @HuSharp

- TiKV
  - 修复当主密钥存储于 KMS (Key Management Service) 时无法轮换主密钥的问题 #17410 @hhwyt
  - 修复删除大表或分区后可能导致的流量控制问题 #17304 @Connor1996
  - 修复过期副本处理 Raft 快照时，由于分裂操作过慢并且随后立即删除新副本，可能导致 TiKV panic 的问题 #17469 @hbisheng

- TiFlash
    - 修复当表里含 Bit 类型列并且带有表示非法字符的默认值时，TiFlash 无法解析表 schema 的问题 #9461 @Lloyd-Pottiger
    - 修复当多个 Region 并发进行副本同步时，可能错误触发 Region overlap 检查失败而导致 TiFlash panic 的问题 #9329 @CalvinNeo
    - 修复一些 TiFlash 不支持的 JSON 函数被错误地下推到 TiFlash 的问题 #9444 @windtalker
- Tools
    - Backup & Restore (BR)
        - 修复 TiDB 节点停止时，监控中 PITR checkpoint 间隔显示异常增大，与实际情况不符的问题 #42419 @YuJuncen
        - 修复备份过程中由于 TiKV 没有响应导致备份任务无法结束的问题 #53480 @Leavrth
        - 修复开启日志备份时，BR 日志可能打印权限凭证敏感信息的问题 #55273 @RidRisR
        - 修复当 PITR 日志备份任务失败时，用户停止了该任务后，PD 中与该任务相关的 safepoint 未被正确清除的问题 #17316 @Leavrth
    - TiDB Data Migration (DM)
        - 修复多个 DM-master 节点可能同时成为 Leader 导致数据不一致的问题 #11602 @GMHDBJD
        - 修复 DM 在处理 ALTER DATABASE 语句时未设置默认数据库导致同步报错的问题 #11503 @lance6716
    - TiDB Lightning
        - 修复两个实例同时并行开始导入任务时，由于分配到的任务 ID 相同导致 TiDB Lightning 报 verify allocator base failed 错误的问题 #55384 @ei-sugimoto

2.9 贡献者

感谢来自 TiDB 社区的贡献者们：

- ei-sugimoto
- eltociear
- guoshouyan（首次贡献者）
- JackL9u
- kafka1991（首次贡献者）
- qingfeng777
- samba-rgb（首次贡献者）
- SeaRise
- tuziemon（首次贡献者）
- xyproto（首次贡献者）

3 TiDB 8.3.0 Release Notes

发版日期：2024 年 8 月 22 日

TiDB 版本：8.3.0

试用链接：快速体验 | 下载离线包

在 8.3.0 版本中，你可以获得以下关键特性：

分类	功能/增强	描述
可扩展性和性能	分区表全局索引（实验特性）	全局索引能够有效提升对非分区列的检索效率，同时也解除了唯一键 (Unique Key) 必须要包含分区键 (Partition Key) 的限制，扩展了 TiDB 分区表的使用场景，也能够避免数据迁移可能遇到的部分应用改造工作。
	默认允许将 Projection 算子下推到存储引擎	Projection 算子下推可以将负载分散到存储节点，同时减少节点间的数据传输。这有助于降低部分 SQL 的执行时间，提升数据库的整体性能。
	统计信息收集忽略不必要的列	在保证优化器能够获取到必要信息的前提下，加快了统计信息收集的速度，提升统计信息的时效性，进而保证选择最优的执行计

分类	功能/增强	描述
		划，提升集群性能。同时也降低了系统开销，改善了资源利用率。
稳定性与高可用	TiProxy 内置虚拟 IP 管理	TiProxy 内置了虚拟 IP 管理功能，配置后支持自动切换虚拟 IP，而无需依赖外部平台或工具。这简化了 TiProxy 的部署，降低了数据库接入层的复杂度。

3.1 功能详情

3.1.1 性能

- 优化器默认允许将 Projection 算子下推到存储引擎 #51876 @yibin87

  将 Projection 算子下推到存储引擎可以减少计算引擎和存储引擎之间的数据传输量，从而提升 SQL 执行效率。这在处理包含 JSON 查询类函数或 JSON 值属性类函数 的查询时尤其有效。从 v8.3.0 开始，TiDB 默认开启 Projection 算子下推功能，控制该功能的系统变量 tidb_opt_projection_push_down 的默认值从 OFF 修改为 ON。启用该功能后，优化器会自动将符合条件的 JSON 查询类函数、JSON 值属性类函数等下推到存储引擎。

  更多信息，请参考用户文档。

- 优化 KV（键值）请求的批处理策略 #55206 @zyguan

  TiDB 通过向 TiKV 发送 KV 请求读取数据。将多个 KV 请求攒批并进行批处理，可以有效提高执行效率。在 v8.3.0 之前，TiDB 的批处理策略效率不高。从 v8.3.0 开始，TiDB 在现有的 KV 请求批处理策略基础上，引入更高效的策略。你可以通过配置项 tikv-client.batch-policy 设置不同的批处理策略，以适应不同的业务场景。

  更多信息，请参考用户文档。

- TiFlash 新增 HashAgg 聚合计算模式，提升高 NDV 数据的聚合计算性能 #9196 @guo-shaoge

  在 v8.3.0 之前，TiFlash 在 HashAgg 聚合计算中处理高 NDV (number of distinct values) 数据时，第一阶段的聚合计算效率较低。从 v8.3.0 开始，TiFlash 引入多种 HashAgg 聚合计算策略，以提升不同特征数据的聚合计算性能。你可以通过系统变量 tiflash_hashagg_preaggregation_mode 设置所需的 HashAgg 聚合计算策略。

  更多信息，请参考用户文档。

- 统计信息收集忽略不必要的列 #53567 @hi-rustin

  当优化器生成执行计划时，只需要部分列的统计信息，例如过滤条件上的列、连接键上的列、聚合目标用到的列。从 v8.3.0 起，TiDB 会持续观测 SQL 语句对列的使用历史，默认只收集有索引的列，以及被观测到的有必要收集统计信息的列。这将会提升统计信息的收集速度，避免不必要的资源浪费。

  从 v8.3.0 之前的版本升级到 v8.3.0 或更高版本时，TiDB 默认保留原有行为，即收集所有列的统计信息。如果要启用该功能，需要手动将系统变量 tidb_analyze_column_options 设置为 PREDICATE，新部署的集群默认开启该功能。

  对于随机查询比较多的偏分析型系统，可以将系统变量 tidb_analyze_column_options 设置为 ALL 来收集所有列的统计信息，以保证随机查询的性能。对于其余类型的系统，推荐保留 tidb_analyze_column_options 的默认设置 PREDICATE，只收集必要列的统计信息。

  更多信息，请参考用户文档。

- 提升部分系统表的查询性能 #50305 @tangenta

之前的版本中，当集群规模变大、表数量较多时，查询系统表性能较差。

在 v8.0.0 优化了以下 4 个系统表的查询性能：

  - INFORMATION_SCHEMA.TABLES
  - INFORMATION_SCHEMA.STATISTICS
  - INFORMATION_SCHEMA.KEY_COLUMN_USAGE
  - INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS

在 v8.3.0 版本优化了以下系统表的查询性能，相比 v8.2.0，性能有数倍的提升：

  - INFORMATION_SCHEMA.CHECK_CONSTRAINTS
  - INFORMATION_SCHEMA.COLUMNS
  - INFORMATION_SCHEMA.PARTITIONS
  - INFORMATION_SCHEMA.SCHEMATA
  - INFORMATION_SCHEMA.SEQUENCES
  - INFORMATION_SCHEMA.TABLE_CONSTRAINTS
  - INFORMATION_SCHEMA.TIDB_CHECK_CONSTRAINTS
  - INFORMATION_SCHEMA.TiDB_INDEXES
  - INFORMATION_SCHEMA.TIDB_INDEX_USAGE
  - INFORMATION_SCHEMA.VIEWS

- 分区表达式使用 EXTRACT(YEAR_MONTH...) 函数时，支持分区裁剪，提升查询性能 #54209 @mjonss

  之前的版本中，当分区表达式使用 EXTRACT(YEAR_MONTH...) 函数时，不支持分区裁剪，导致查询性能较差。从 v8.3.0 开始，当分区表达式使用该函数时，支持分区裁剪，提升了查询性能。

  更多信息，请参考用户文档。

- 批量建表 (CREATE TABLE) 的性能提升了 1.4 倍，批量建库 (CREATE DATABASE) 的性能提升了 2.1 倍，批量加列 (ADD COLUMN) 的性能提升了 2 倍 #54436 @D3Hunter

v8.0.0 引入了系统变量 tidb_enable_fast_create_table，用于在批量建表的场景中提升建表的性能。在 v8.3.0 中，通过 10 个 session 在单个库内并发提交建表的 DDL，相比 v8.2.0，性能有 1.4 倍的提升。

在 v8.3.0 中，逻辑 DDL (General DDL) 在批量执行时的性能相比 v8.2.0 也均有提升，其中通过 10 个 session 并发批量建库 (CREATE DATABASE) 的性能相比 v8.1.0 提升了 19 倍，相比 v8.2.0 提升了 2.1 倍。10 个 session 对同个库内的多个表批量加列 (ADD COLUMN) 的性能相比 v8.1.0 提升了 10 倍，相比 v8.2.0 提升了 2 倍。

更多信息，请参考用户文档。

- 分区表支持全局索引 (Global Index)（实验特性）#45133 @mjonss @Defined2014 @jiyfhust @L-maple

  之前版本的分区表，因为不支持全局索引有较多的限制，比如唯一键必须包含分区表达式中用到的所有列，如果查询条件不带分区键，查询时会扫描所有分区，导致性能较差。从 v7.6.0 开始，引入了系统变量 tidb_enable_global_index 用于开启全局索引特性，但该功能当时处于开发中，不够完善，不建议开启。

  从 v8.3.0 开始，全局索引作为实验特性正式发布。你可通过关键字 Global 为分区表显式创建一个全局索引，从而去除分区表唯一建必须包含分区表达式中用到的所有列的限制，满足灵活的业务需求。同时基于全局索引也提升了非分区列的查询性能。

  更多信息，请参考用户文档。

3.1.2 稳定性

- 支持以流式获取游标的结果集（实验特性）#54526 @YangKeao

当应用代码通过 Cursor Fetch 获取结果集时，TiDB 通常会先将完整结果保存至 TiDB 内存，再分批返回给客户端。如果结果集过大，可能会触发落盘临时将结果写入硬盘。

从 v8.3.0 开始，如果将系统变量 tidb_enable_lazy_cursor_fetch 设置为 ON，TiDB 不再把所有数据读取到 TiDB 节点，而是会随着客户端的读取逐步将数据读到 TiDB 节点。在处理较大的结果集时，这将减少 TiDB 节点的内存使用，提升集群的稳定性。

更多信息，请参考用户文档。

- 增强 SQL 执行计划绑定 #55280 #55343 @time-and-fate

  在 OLTP 负载环境中，绝大部分 SQL 的最优执行计划是固定不变的。对业务中的重要 SQL 实施执行计划绑定，可以降低执行计划变差的几率，提升系统稳定性。为了满足创建大量 SQL 执行计划绑定的场景需求，TiDB 对 SQL 绑定的能力和体验进行了增强，其中包括：

  - 用单条 SQL 语句从多个历史执行计划中创建 SQL 执行计划绑定，提升创建绑定的效率。
  - SQL 执行计划绑定支持更多的优化器提示，并优化了对复杂执行计划的转换方法，使得绑定能够更稳定地还原执行计划。

  更多信息，请参考用户文档。

3.1.3 高可用

- TiProxy 内置虚拟 IP 管理功能 #583 @djshow832

  在 v8.3.0 之前，当使用主从模式以保证高可用性时，TiProxy 需要额外的组件管理虚拟 IP。从 v8.3.0 开始，TiProxy 内置虚拟 IP 管理功能。在主从模式下，当主节点发生切换时，新的主节点会自动绑定指定的虚拟 IP，确保客户端始终能通过虚拟 IP 连接到可用的 TiProxy。

要启用虚拟 IP 管理功能，需要通过 TiProxy 配置项 ha.virtual-ip 指定虚拟 IP 地址，并通过 ha.interface 指定绑定虚拟 IP 的网络接口。只有这两个配置项都设置时，TiProxy 实例才能绑定虚拟 IP。

更多信息，请参考用户文档。

3.1.4 SQL 功能

- 支持将 SELECT LOCK IN SHARE MODE 升级为排它锁 #54999 @cfzjywxk

  TiDB 暂不支持 SELECT LOCK IN SHARE MODE。从 v8.3.0 开始，TiDB 支持将 SELECT LOCK IN SHARE MODE 升级为排它锁，实现对 SELECT LOCK IN SHARE MODE 语法的支持。你可以使用系统变量 tidb_enable_shared_lock_promotion 控制是否启用该功能。

  更多信息，请参考用户文档。

3.1.5 可观测性

- 展示初始统计信息的加载进度 #53564 @hawkingrei

  TiDB 在启动时要加载基础统计信息，在表或者分区数量很多的情况下，该过程要耗费一定时间。当配置项 force-init-stats 设置为 ON 时，初始统计信息加载完成前，TiDB 不会对外提供服务。在这种情况下，你需要对加载过程进行观测，从而预估服务开启时间。

  从 v8.3.0 开始，TiDB 会在日志中分阶段打印初始统计信息的加载进度，以便了解运行情况。为了给外部工具提供格式化的结果，TiDB 增加了额外的监控 API，以便能够在启动阶段随时获取初始统计信息的加载进度。

- 添加 Request Unit (RU) 配置监控指标 #8444 @nolouch

3.1.6 安全

- 增强 PD 日志脱敏 #8305 @JmPotato

TiDB v8.0.0 增强了日志脱敏功能，支持控制是否使用 ‹ › 包裹 TiDB 日志中的用户数据。基于标记后的日志，你可以在展示日志时决定是否对被标记信息进行脱敏处理，从而提升日志脱敏功能的灵活性。在 v8.2.0 中，TiFlash 实现了类似的日志脱敏功能增强。

在 v8.3.0 中，PD 实现了类似的日志脱敏功能增强。要使用该功能，可以将 PD 配置项 security.redact-info-log 的值设置为 "marker"。

更多信息，请参考用户文档。

- 增强 TiKV 日志脱敏 #17206 @lucasliang

  TiDB v8.0.0 增强了日志脱敏功能，支持控制是否使用 ‹ › 包裹 TiDB 日志中的用户数据。基于标记后的日志，你可以在展示日志时决定是否对被标记信息进行脱敏处理，从而提升日志脱敏功能的灵活性。在 v8.2.0 中，TiFlash 实现了类似的日志脱敏功能增强。

  在 v8.3.0 中，TiKV 实现了类似的日志脱敏功能增强。要使用该功能，可以将 TiKV 配置项 security.redact-info-log 的值设置为 "marker"。

  更多信息，请参考用户文档。

3.1.7 数据迁移

- TiCDC 支持通过双向复制模式 (Bi-Directional Replication, BDR) 同步 DDL 语句 (GA) #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 不支持双向复制 DDL 语句，因此要使用 TiCDC 双向复制必须将 DDL 语句在两个 TiDB 集群分别执行。有了该特性，在为一个集群分配 PRIMARY BDR role 之后，TiCDC 可以将该集群的 DDL 语句复制到 SECONDARY 集群。

  在 v8.3.0，该功能成为正式功能 (GA)。

更多信息，请参考用户文档。

3.2 兼容性变更

注意：

以下为从 v8.2.0 升级至当前版本 (v8.3.0) 所需兼容性变更信息。如果从 v8.1.0 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

- 为了避免命令错误使用，pd-ctl 取消了前缀匹配的机制，例如 store remove-tombstone 不能通过 store remove 来调用 #8413 @lhy1024

3.2.2 系统变量

变量名	修改类型	描述
tidb_ddl_reorg_batch_size	修改	增加 SESSION 作用域。
tidb_ddl_reorg_worker_cnt	修改	增加 SESSION 作用域。
tidb_gc_concurrency	修改	从 v8.3.0 起，该变量可以控制垃圾回收 (GC) 过程中 Resolve Locks（清理锁）和 Delete Range（删除区间）的并发线程数。在 v8.3.0 之前，该变量只能控制 Resolve Locks（清理锁）的线程数。
tidb_low_resolution_tso	修改	增加 GLOBAL 作用域。
tidb_opt_projection_push_down	修改	增加 GLOBAL 作用域，变量值可以持久化到集群。经进一步的测试，默认值从 OFF 修改为 ON，即默认允许优化器将 Projection 算子下推到 TiKV。
tidb_schema_cache_size	修改	取值范围修改为 0 或 [536870912, 9223372036854775807]，其中最小值为 536870912 bytes（即 512 MiB），避免设置了过小的 cache size 导致性能下降。

变量名	修改类型	描述
tidb_analyze_column_options	新增	控制 ANALYZE TABLE 语句的行为。将其设置为默认值 PREDICATE 表示仅收集 predicate columns 的统计信息；将其设置为 ALL 表示收集所有列的统计信息。
tidb_enable_lazy_cursor_fetch	新增	这个变量用于控制 Cursor Fetch 功能的行为。
tidb_enable_shared_lock_promotion	新增	控制是否启用共享锁升级为排他锁的功能。默认值为 OFF，表示不启用共享锁升级为排他锁的功能。
tiflash_hashagg_preaggregation_mode	新增	控制下推到 TiFlash 的两阶段或三阶段 HashAgg 在第一阶段采用哪种预聚合策略。

3.2.3 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	tikv-client.batch-policy	新增	控制 TiDB 向 TiKV 发送请求时的批处理策略。
PD	security.redact-info-log	修改	支持将 PD 配置项 security.redact-info-log 的值设置为 "marker"，使用标记符号 ‹ › 标记出敏感信息，而不是直接隐藏，以便你能够自定义脱敏规则。
TiKV	security.redact-info-log	修改	支持将 TiKV 配置项 security.redact-info-log 的值设置为 "marker"，使用标记符号 ‹ › 标记出敏感信息，而不是直接隐藏，以便你能够自定义脱敏规则。
TiFlash	security.redact-info-log	修改	支持将 TiFlash Learner 配置项 security.redact-info-log 的值设置为 "marker"，使用标记符号 ‹ › 标记出敏感信息，而不是直接隐藏，以便你能够自定义脱敏规则。
BR	--allow-pitr-from-incremental	新增	控制增量备份和后续的日志备份是否兼容。默认值为 true，即增量备份兼容后续

配置文件	配置项	修改类型	描述
			的日志备份。兼容的情况下，增量恢复开始前会对需要回放的 DDL 进行严格检查。

3.2.4 系统表

- 在系统表 INFORMATION_SCHEMA.PROCESSLIST 和 INFORMATION_SCHEMA.CLUSTER_PROCESSLIST 中新增 ROWS_AFFECTED 字段，用于显示 DML 语句当前影响的数据行数。#46889 @lcwangchao

3.3 废弃功能

- 以下为从 v8.3.0 开始已废弃的功能：
    - 从 v7.5.0 开始，TiDB Binlog 的数据同步功能被废弃。从 v8.3.0 开始，TiDB Binlog 被完全废弃，并计划在未来版本中移除。如需进行增量数据同步，请使用 TiCDC。如需按时间点恢复 (point-in-time recovery, PITR)，请使用 PITR。
    - 从 v8.3.0 开始，系统变量 tidb_enable_column_tracking 被废弃。TiDB 默认收集 predicate columns 的统计信息。更多信息，参见 tidb_analyze_column_options。
- 以下为计划将在未来版本中废弃的功能：
    - TiDB 在 v8.0.0 引入了系统变量 tidb_enable_auto_analyze_priority_queue，用于控制是否启用优先队列来优化自动收集统计信息任务的排序。在未来版本中，优先队列将成为自动收集统计信息任务的唯一排序方式，系统变量 tidb_enable_auto_analyze_priority_queue 将被废弃。
    - TiDB 在 v7.5.0 引入了系统变量 tidb_enable_async_merge_global_stats，用于设置 TiDB 使用异步方式合并分区统计信息，以避免 OOM 问题。在未来版本中，分区统计

信息将统一使用异步方式进行合并，系统变量 tidb_enable_async_merge_global_stats 将被废弃。

- 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 在 v8.0.0 引入了系统变量 tidb_enable_parallel_hashagg_spill，用于控制 TiDB 是否支持并行 HashAgg 进行落盘。在未来版本中，系统变量 tidb_enable_parallel_hashagg_spill 将被废弃。
- 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。

- 以下为计划将在未来版本中移除的功能：
  - 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。

3.4 改进提升

- TiDB
  - 支持 SELECT ... STRAIGHT_JOIN ... USING (...) 语句 #54162 @dveeden
  - 支持为形如 ((idx_col_1 > 1) or (idx_col_1 = 1 and idx_col_2 > 10)) and ((idx_col_1 < 10) or (idx_col_1 = 10 and idx_col_2 < 20)) 的过滤条件构造更精准的索引访问 Range #54337 @ghazalfamilyusa
  - 支持形如 WHERE idx_col_1 IS NULL ORDER BY idx_col_2 的 SQL 查询利用索引顺序避免额外排序操作 #54188 @ari-e
  - 在系统表 mysql.analyze_jobs 中显示被 ANALYZE 过的索引 #53567 @hi-rustin

- EXPLAIN 语句支持应用 tidb_redact_log，并进一步优化了日志记录的处理逻辑 #54565 @hawkingrei
- 支持在多值索引的 IndexRangeScan 上生成 Selection，以提高查询效率 #54876 @time-and-fate
- 支持在设定的自动 ANALYZE 时间窗口外终止正在执行的自动 ANALYZE 任务 #55283 @hawkingrei
- 当某个统计信息完全由 TopN 构成，且对应表的统计信息中修改行数不为 0 时，对于未命中 TopN 的等值条件，估算结果从 0 调整为 1 #47400 @terry1purcell
- TopN 算子支持数据落盘功能 #47733 @xzhangxian1008
- TiDB 节点支持执行包含 WITH ROLLUP 修饰符和 GROUPING 函数的查询 #42631 @Arenatlx
- 系统变量 tidb_low_resolution_tso 增加全局 (GLOBAL) 作用域 #55022 @cfzjywxk
- GC（垃圾回收）支持并发 Delete Range（删除区间）以提升处理效率，可以通过 tidb_gc_concurrency 控制并发线程数 #54570 @ekexium
- 提升批量 DML 执行方式 (tidb_dml_type = "bulk") 的性能 #50215 @ekexium
- 提升 schema 信息缓存相关接口 SchemaByID 的性能 #54074 @ywqzzy
- 提升 schema 信息缓存开启时部分系统表的查询性能 #50305 @tangenta
- 优化添加唯一索引时冲突键的报错信息 #53004 @lance6716

- PD
  - 支持通过 pd-ctl 修改 evict-leader-scheduler 的 batch 配置，以提升驱逐 Leader 的速度 #8265 @rleungx
  - Grafana 上的 **Cluster > Label distribution** 面板新增 store_id 监控指标，用于显示不同 label 对应的 store ID #8337 @HuSharp

    - 当指定的资源组不存在时，支持回退到默认资源组 #8388 @JmPotato
    - 在 pd-ctl 的 region 命令输出的 Region 信息中，新增 approximate_kv_size 字段 #8412 @zeminzhou
    - 优化调用 PD API 删除 TTL 配置时的输出信息 #8450 @lhy1024
    - 优化大查询读请求消耗 RU (Request Unit) 的行为，以减少对其他请求的影响 #8457 @nolouch
    - 优化 PD 微服务设置错误时返回的错误信息 #52912 @rleungx
    - PD 微服务新增 --name 启动参数，以便部署时更精确地显示服务名称 #7995 @HuSharp
    - 支持通过 Region 数量动态调整 PatrolRegionScanLimit，以减少扫描 Region 所需的时间 #7963 @lhy1024
- TiKV
    - 优化 async-io 下写 Raft 日志的攒批策略，减少对磁盘 I/O 带宽资源的使用 #16907 @LykxSassinator
    - 重新设计了 TiCDC 的 delegate 和 downstream 模块以更好地支持 Region 局部订阅 #16362 @hicqu
    - 减少单个慢日志的大小 #17294 @Connor1996
    - 增加监控指标 min safe ts #17307 @mittalrishabh
    - 减少 peer message channel 的内存使用 #16229 @Connor1996
- TiFlash
    - 支持生成 SVG 格式的堆内存分析结果 #9320 @CalvinNeo
- Tools
    - Backup & Restore (BR)
        - 在第一次进行按时间点恢复 (Point-in-time recovery, PITR) 前，新增对全量备份是否存在的检查；如果未找到全量备份，会终止恢复并返回错误 #54418 @Leavrth

- 在恢复快照备份的数据之前，新增对 TiKV 和 TiFlash 是否有足够的磁盘空间的检查；如果空间不足，会终止恢复并返回错误 #54316 @RidRisR
- 在 TiKV 下载每个 SST 文件之前，新增对 TiKV 是否有足够的磁盘空间的检查；如果空间不足，会终止恢复并返回错误 #17224 @RidRisR
- 支持通过环境变量设置阿里云访问身份 #45551 @RidRisR
- 使用 BR 进行备份恢复时，会根据 BR 进程的可用内存自动设置环境变量 GOMEMLIMIT，避免出现 OOM #53777 @Leavrth
- 使增量备份兼容按时间点恢复 (PITR) #54474 @3pointer
- 支持备份和恢复 mysql.column_stats_usage 表 #53567 @hi-rustin

3.5 错误修复

- TiDB
    - 通过重置 PipelinedWindow 的 Open 方法中的参数，修复当 PipelinedWindow 作为 apply 的子节点使用时，由于重复的打开和关闭操作导致重用之前的参数值而发生的意外错误 #53600 @XuHuaiyu
    - 修复由于查询超出 tidb_mem_quota_query 设定的内存使用限制，导致终止查询时可能卡住的问题 #55042 @yibin87
    - 修复 HashAgg 算子在并行计算过程中因落盘导致查询结果不正确的问题 #55290 @xzhangxian1008
    - 修复从 YEAR 转换为 JSON 格式时 JSON_TYPE 错误的问题 #54494 @YangKeao
    - 修复系统变量 tidb_schema_cache_size 的取值范围错误的问题 #54034 @lilinghai

- 修复当分区表达式为 EXTRACT(YEAR FROM col) 时没有分区裁剪的问题 #54210 @mjonss
- 修复表较多的情况下 FLASHBACK DATABASE 失败的问题 #54415 @lance6716
- 修复库较多的情况下 FLASHBACK DATABASE 死循环的问题 #54915 @lance6716
- 修复使用索引加速模式添加索引可能失败的问题 #54568 @lance6716
- 修复 ADMIN CANCEL DDL JOBS 可能导致 DDL 失败的问题 #54687 @lance6716
- 修复来自 DM 同步的表超过索引列最大长度 max-index-length 时，同步失败的问题 #55138 @lance6716
- 修复开启 tidb_enable_inl_join_inner_multi_pattern 时，执行 SQL 语句可能报错 runtime error: index out of range 的问题 #54535 @joechenrh
- 修复 TiDB 在统计信息初始化的过程中，无法通过 Control+C 的方式退出 TiDB 的问题 #54589 @tiancaiamao
- 修复 INL_MERGE_JOIN Optimizer Hint 返回错误结果的问题，将其废弃 #54064 @AilinKid
- 修复关联子查询中包含 WITH ROLLUP 时 TiDB 可能 panic 并报错 runtime error: index out of range 的问题 #54983 @AilinKid
- 修复当 SQL 查询的过滤条件中包含虚拟列，且执行条件中包含 UnionScan 时，谓词无法正常下推的问题 #54870 @qw4990
- 修复开启 tidb_enable_inl_join_inner_multi_pattern 时，执行 SQL 语句可能报错 runtime error: invalid memory address or nil pointer dereference 的问题 #55169 @hawkingrei
- 修复包含 UNION 的查询语句可能返回错误结果的问题 #52985 @XuHuaiyu

- 修复 mysql.stats_histograms 表的 tot_col_size 列可能为负数的潜在风险 #55126 @qw4990
- 修复 columnEvaluator 无法识别输入 chunk 中的列引用，导致执行 SQL 报错 runtime error: index out of range 的问题 #53713 @AilinKid
- 修复 STATS_EXTENDED 变成保留关键字的问题 #39573 @wddevries
- 修复 tidb_low_resolution 开启时，select for update 可以被执行的问题 #54684 @cfzjywxk
- 修复 tidb_redact_log 开启时，内部 SQL 在慢日志里无法显示的问题 #54190 @lcwangchao
- 修复事务占用的内存可能被多次重复统计的问题 #53984 @ekexium
- 修复使用 SHOW WARNINGS; 获取警告时可能导致 panic 的问题 #48756 @xhebox
- 修复加载索引统计信息可能会造成内存泄漏的问题 #54022 @hi-rustin
- 修复当排序规则为 utf8_bin 或 utf8mb4_bin 时意外消除 LENGTH() 条件的错误 #53730 @elsa0520
- 修复统计数据在遇到主键重复时没有更新 stats_history 表的问题 #47539 @Defined2014
- 修复递归 CTE 查询可能导致无效指针的问题 #54449 @hawkingrei
- 修复某些连接在握手完成之前退出导致 Grafana 监控指标中的连接数 (Connection Count) 不正确的问题 #54428 @YangKeao
- 修复使用 TiProxy 和资源组 (Resource Group) 功能时，每个资源组的连接数 (Connection Count) 显示不正确的问题 #54545 @YangKeao
- 修复当查询包含非关联子查询和 LIMIT 子句时，列剪裁可能不完善导致计划不优的问题 #54213 @qw4990
- 修复针对 SELECT ... FOR UPDATE 复用了错误点查询计划的问题 #54652 @qw4990

- 修复当第一个参数是 month 并且第二个参数是负数时，TIMESTAMPADD() 函数会进入无限循环的问题 #54908 @xzhangxian1008
- 修复慢日志中内部语句中的 SQL 默认被脱敏为空的问题 #54190 #52743 #53264 @lcwangchao
- 修复可以生成 _tidb_rowid 的点查 (PointGet) 执行计划的问题 #54583 @Defined2014
- 修复从 v7.1 升级后 SHOW IMPORT JOBS 报错 Unknown column 'summary' 的问题 #54241 @tangenta
- 修复当视图定义中使用子查询作为列定义时，通过 information_schema.columns 获取列信息返回告警 Warning 1356 的问题 #54343 @lance6716
- 修复可以创建非严格自增的 RANGE 分区表的问题 #54829 @Defined2014
- 修复当 SQL 异常中断时，INDEX_HASH_JOIN 无法正常退出的问题 #54688 @wshwsh12
- 修复使用分布式框架添加索引期间出现网络分区可能导致数据索引不一致的问题 #54897 @tangenta

- PD
    - 修复将角色 (role) 绑定到资源组时未报错的问题 #54417 @JmPotato
    - 修复资源组在请求 token 超过 500 ms 时遇到超出配额限制的问题 #8349 @nolouch
    - 修复 INFORMATION_SCHEMA.RUNAWAY_WATCHES 表中时间类型不正确的问题 #54770 @HuSharp
    - 修复资源组 (Resource Group) 在高并发场景下无法有效限制资源使用的问题 #8435 @nolouch
    - 修复获取表属性时错误调用 PD API 的问题 #55188 @JmPotato

- 修复开启 scheduling 微服务后，扩缩容进度显示错误的问题 #8331 @rleungx
- 修复加密管理器在使用前未初始化的问题 #8384 @rleungx
- 修复部分日志未脱敏的问题 #8419 @rleungx
- 修复开启 PD 微服务时，重定向可能 panic 的问题 #8406 @HuSharp
- 修复反复修改 split-merge-interval 的值（例如从 1s 改为 1h，再改回 1s）可能导致该配置不生效的问题 #8404 @lhy1024
- 修复设置 replication.strictly-match-label 为 true 导致 TiFlash 启动失败的问题 #8480 @rleungx
- 修复在 ANALYZE 大规模分区表时获取 TSO 慢导致 ANALYZE 性能下降的问题 #8500 @rleungx
- 修复了大规模集群下可能发生数据竞争的问题 #8386 @rleungx
- 修复 TiDB 在判断查询是否为 Runaway Queries 时，只统计了 Coprocessor 侧的时间消耗但未统计 TiDB 侧的时间消耗，导致一些查询未被识别为 Runaway Queries 的问题 #51325 @HuSharp

- TiFlash
  - 修复使用 CAST() 函数将字符串转换为带时区或非法字符的日期时间时，结果错误的问题 #8754 @solotzg
  - 修复跨数据库对含空分区的分区表执行 RENAME TABLE ... TO ... 后，TiFlash 可能 panic 的问题 #9132 @JaySon-Huang
  - 修复开启延迟物化后，部分查询在执行时可能报列类型不匹配错误的问题 #9175 @JinheLin
  - 修复开启延迟物化后，带有虚拟生成列的查询可能返回错误结果的问题 #9188 @JinheLin
  - 修复将 TiFlash 中 SSL 证书配置项设置为空字符串会错误开启 TLS 并导致 TiFlash 启动失败的问题 #9235 @JaySon-Huang

    - 修复数据库创建后短时间内被删除时，TiFlash 可能 panic 的问题 #9266 @JaySon-Huang
    - 修复 TiFlash 与任意 PD 之间发生网络分区（即网络连接断开），可能导致读请求超时报错的问题 #9243 @Lloyd-Pottiger
    - 修复在存算分离架构下，TiFlash 写节点可能重启失败的问题 #9282 @JaySon-Huang
    - 修复在存算分离架构下，TiFlash 写节点的读快照可能没有被及时释放的问题 #9298 @JinheLin
- TiKV
    - 修复在清理旧 Region 时可能会误删数据的问题 #17258 @hbisheng
    - 修复 Grafana TiKV 组件中的 Ingestion picked level 和 Compaction Job Size(files) 显示不正确的问题 #15990 @Connor1996
    - 修复 cancel_generating_snap 错误地更新 snap_tried_cnt 导致 TiKV panic 的问题 #17226 @hbisheng
    - 修复 Ingest SST duration seconds 统计信息说明错误的问题 #17239 @LykxSassinator
    - 修复 CPU profiling flag 在出现错误时没有正确重置的问题 #17234 @Connor1996
    - 修复早期版本（早于 v7.1）和之后的版本的 bloom filter 无法兼容的问题 #17272 @v01dstar
- Tools
    - Backup & Restore (BR)
        - 修复增量恢复过程中 ADD INDEX、MODIFY COLUMN 等需要回填的 DDL 可能无法正确恢复的问题 #54426 @3pointer
        - 修复备份恢复时进度条卡住的问题 #54140 @Leavrth
        - 修复备份恢复的断点路径在一些外部存储中不兼容的问题 #55265 @Leavrth
    - TiCDC

        - 修复当下游 Kafka 无法访问时，Processor 可能卡住的问题 #11340 @asddongmen
    - TiDB Data Migration (DM)
        - 修复 schema tracker 无法正确处理 LIST 分区表导致 DM 报错的问题 #11408 @lance6716
        - 修复当索引长度超过 max-index-length 默认值时数据同步中断的问题 #11459 @michaelmdeng
        - 修复 DM 无法正确处理 FAKE_ROTATE_EVENT 的问题 #11381 @lance6716
    - TiDB Lightning
        - 修复 TiDB Lightning 获取 keyspace 失败时输出的 WARN 日志可能引起用户混淆的问题 #54232 @kennytm
        - 修复 TiDB Lightning 的 TLS 配置影响集群证书的问题 #54172 @ei-sugimoto
        - 修复使用 TiDB Lightning 导入数据时报事务冲突的问题 #49826 @lance6716
        - 修复导入大量库表时 checkpoint 文件过大导致性能下降的问题 #55054 @D3Hunter

3.6 贡献者

感谢来自 TiDB 社区的贡献者们：

- ari-e
- ei-sugimoto
- HaoW30
- JackL9u
- michaelmdeng
- mittalrishabh
- qingfeng777
- SandeepPadhi
- yzhan1

4 TiDB 8.2.0 Release Notes

发版日期：2024 年 7 月 11 日

TiDB 版本：8.2.0

试用链接：快速体验 | 下载离线包

在 8.2.0 版本中，你可以获得以下关键特性：

分类	功能/增强	描述
稳定性与高可用	TiProxy 支持多种负载均衡策略	在 TiDB v8.2.0 中，TiProxy 支持从多个维度（包括状态、连接数、健康度、内存、CPU 和地理位置）对 TiDB 节点进行评估和排序，并支持通过 policy 配置项配置这些负载均衡策略的优先级。TiProxy 将根据 policy 动态选择最优 TiDB 节点执行数据库操作，从而优化 TiDB 节点的整体资源使用率，提升集群性能和吞吐。
	并行 HashAgg 算法支持数据落盘成为正式功能 (GA)	HashAgg 是 TiDB 中常用的聚合算子，用于快速聚合具有相同字段值的行。TiDB v8.0.0 引入并行 HashAgg 作为实验特性，以进一步提升处理速度。当内存资源不足时，并行 HashAgg 可以将临时排序数据落盘，避免因内存使用过度而导致的 OOM 风险，从而提升查询性能和节点稳定性。该功能在 v8.2.0 成为正式功能，并默认开启，用户可以通过 tidb_executor_concurrency 安全地设置并行 HashAgg 的并发度。
	统计信息加载效率提升 10 倍	对于拥有大量表和分区的集群，比如 SaaS 或 PaaS 服务，统计信息加载效率的提升能够解决 TiDB 实例启动缓慢的问题，同时也能提升统计信息动态加载的成功率，从而减少由于统计信息加载失败造成的性能回退，提升集群的稳定性。
数据库管理与可观测性	为切换资源组引入权限控制	随着资源管控功能被广泛应用，对资源组切换操作的权限控制能够避免数据库用户对资源的滥用，强化管理员对整体资源使用的保护，从而提升集群的稳定性。

4.1 功能详情

4.1.1 性能

- 支持下推以下 JSON 函数到 TiKV #50601 @dbsid
    - JSON_ARRAY_APPEND()
    - JSON_MERGE_PATCH()
    - JSON_REPLACE()

  更多信息，请参考用户文档。

- TiDB 支持并行排序 #49217 #50746 @xzhangxian1008

  在 v8.2.0 之前，TiDB 只能以非并行的方式执行排序计算，当需要对大量数据进行排序时，查询性能会受到影响。

  从 v8.2.0 开始，TiDB 支持并行排序功能，显著提升了排序计算的性能。该功能无需手动开启，TiDB 会根据系统变量 tidb_executor_concurrency 的值自动选择并行或非并行排序。

  更多信息，请参考用户文档。

- TiDB 的并行 HashAgg 算法支持数据落盘成为正式功能 (GA) #35637 @xzhangxian1008

  TiDB v8.0.0 以实验特性引入了并行 HashAgg 算法支持数据落盘功能。在 v8.2.0 中，该功能成为正式功能 (GA)。TiDB 在使用并行 HashAgg 算法时，将根据内存使用情况自动触发数据落盘，从而兼顾查询性能和数据处理量。该功能默认开启，控制该功能的变量 tidb_enable_parallel_hashagg_spill 将在未来版本中废弃。

  更多信息，请参考用户文档。

4.1.2 稳定性

- 统计信息加载效率提升 10 倍 #52831 @hawkingrei

SaaS 或 PaaS 类业务应用中可能存在大量的数据表，这些表不但会拖慢初始统计信息的加载速度，也会增加高负载情况下同步负载的失败率。TiDB 的启动时间以及执行计划的准确性都会受到影响。在 v8.2.0 中，TiDB 从并发模型、内存分配方式等多个角度优化了统计信息的加载过程，降低延迟，提升吞吐，避免由于统计信息加载速度过慢，影响业务扩容。

新增支持自适应的并行加载。默认情况下，配置项 stats-load-concurrency 的值为 0，统计信息加载的并行度会根据硬件规格自动选择。

更多信息，请参考用户文档。

4.1.3 高可用

- TiProxy 支持多种负载均衡策略 #465 @djshow832 @xhebox

  TiProxy 是 TiDB 的官方代理组件，位于客户端和 TiDB server 之间，为 TiDB 集群提供负载均衡和连接保持功能。在 v8.2.0 之前，TiProxy 默认使用 v1.0.0 版本，仅支持基于 TiDB server 状态和连接数的负载均衡策略。

  从 v8.2.0 开始，TiProxy 默认使用 v1.1.0 版本，新增多种负载均衡策略，除了状态和连接数，还支持根据健康度、内存、CPU、地理位置对 TiDB 集群的连接进行动态负载均衡调度，提高整个 TiDB 集群的稳定性。

  你可以通过 TiProxy 配置项 policy 配置负载均衡策略的组合和优先级，具体策略包括：

  - resource：资源优先策略，优先级顺序依次为基于状态、健康度、内存、CPU、地理位置、连接数的负载均衡。
  - location：地理优先策略，优先级顺序依次为基于状态、地理位置、健康度、内存、CPU、连接数的负载均衡。
  - connection：最小连接数策略，优先级顺序依次为基于状态、连接数的负载均衡。

  更多信息，请参考用户文档。

4.1.4 SQL 功能

- TiDB 支持 JSON Schema Validation 函数 #52779 @dveeden

  在 v8.2.0 之前，你需要依赖外部工具或自定义验证逻辑进行 JSON 数据验证，开发和维护比较复杂，开发效率低。从 v8.2.0 版本开始，引入了 JSON_SCHEMA_VALID() 函数。通过在 CHECK 约束中使用 JSON_SCHEMA_VALID()，可以避免插入不符合要求的数据，而不是事后检查数据。你可以在 TiDB 中直接验证 JSON 数据的有效性，提高数据的完整性和一致性，提升了开发效率。

  更多信息，请参考用户文档。

4.1.5 数据库管理

- TiUP 支持部署 PD 微服务 #5766 @rleungx

  PD 从 v8.0.0 开始支持微服务模式。该模式通过将 PD 的时间戳分配和集群调度功能拆分为独立的服务进行部署和管理，可以更好地控制资源的使用和隔离，减少不同服务之间的相互影响。但是，在 v8.2.0 之前的版本中，PD 微服务仅支持通过 TiDB Operator 进行部署。

  从 v8.2.0 开始，PD 微服务支持通过 TiUP 进行部署。你可以在集群中单独部署 tso 微服务和 scheduling 微服务，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。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。

- 为切换资源组的操作增加权限控制 #53440 @glorv

  TiDB 允许用户使用命令 SET RESOURCE GROUP 或 Hint RESOURCE_GROUP() 切换到其他资源组，这可能会造成部分数据库用户对资源组的滥用。TiDB v8.2.0 增加了对资源组切换行为的管控，只有被授予

动态权限 RESOURCE_GROUP_ADMIN 或者 RESOURCE_GROUP_USER 的数据库用户，才能切换到其他资源组，以加强对系统资源的保护。

为了维持兼容性，从旧版本升级到 v8.2.0 及之后版本的集群维持原行为不变。通过设置新增变量 tidb_resource_control_strict_mode 为 ON，来开启上述的增强权限控制。

更多信息，请参考用户文档。

4.1.6 可观测性

- 记录执行计划没有被缓存的原因 #50618 @qw4990

  在一些场景下，用户希望多数执行计划能够被缓存，以节省执行开销，并降低延迟。目前执行计划缓存对 SQL 有一定限制，部分形态 SQL 的执行计划无法被缓存，但是用户很难识别出无法被缓存的 SQL 以及对应的原因。因此，从 v8.2.0 开始，为系统表 STATEMENTS_SUMMARY 增加了新的列 PLAN_CACHE_UNQUALIFIED 和 PLAN_CACHE_UNQUALIFIED_LAST_REASON，来解释计划无法被缓存的原因，协助用户进行性能调优。

  更多信息，请参考用户文档。

4.1.7 安全

- 增强 TiFlash 日志脱敏 #8977 @JaySon-Huang

  TiDB v8.0.0 增强了日志脱敏功能，支持控制是否使用标记符号 ‹ › 包裹 TiDB 日志中的用户数据。基于标记后的日志，你可以在展示日志时决定是否对被标记信息进行脱敏处理，从而提升日志脱敏功能的灵活性。在 v8.2.0 中，TiFlash 进行了类似的日志脱敏功能增强。要使用该功能，可以将 TiFlash 配置项 security.redact_info_log 的值设置为 marker。

  更多信息，请参考用户文档。

4.1.8 数据迁移

- 对齐不同 changefeed 的 Syncpoint #11212 @hongyunyan

  在 v8.2.0 之前，对齐多个 changefeed 的 Syncpoint 很有挑战性。在创建 changefeed 时，必须谨慎选择 changefeed 的 startTs，以便与其他 changefeed 的 Syncpoint 对齐。从 v8.2.0 开始，为 changefeed 创建的 Syncpoint 是 changefeed 的 sync-point-interval 配置的倍数。这个调整可以让你对齐具有相同 sync-point-interval 配置的多个 changefeed 的 Syncpoint，简化和提高了对齐多个下游集群的能力。

  更多信息，请参考用户文档。

- TiCDC Pulsar Sink 新增支持 pulsar+http 和 pulsar+https 连接协议 #11336 @SandeepPadhi

  在 v8.2.0 之前，TiCDC Pulsar Sink 仅支持使用 pulsar 和 pulsar+ssl 协议进行连接。从 v8.2.0 开始，TiCDC Pulsar Sink 新增支持使用 pulsar+http 和 pulsar+https 协议进行连接。这个调整可以让你更加灵活地配置 Pulsar Sink 的连接方式。

  更多信息，请参考用户文档。

4.2 兼容性变更

注意：

以下为从 v8.1.0 升级至当前版本 (v8.2.0) 所需兼容性变更信息。如果从 v8.0.0 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 行为变更

- 使用 TiDB Lightning 导入 CSV 文件时，如果设置了严格格式 strict-format = true 将一个大 CSV 文件切分为多个小 CSV 文件来提升并发和导入性能，

需要显式指定行结束符 terminator，参数的取值为 \r、\n 或 \r\n。如果没有指定行结束符，可能导致 CSV 文件数据解析异常。#37338 @lance6716

- 使用 IMPORT INTO 导入 CSV 文件时，如果指定 SPLIT_FILE 参数将一个大 CSV 文件切分为多个小 CSV 文件来提升并发和导入性能，需显式指定行结束符 LINES_TERMINATED_BY，参数的取值为 \r、\n 或 \r\n。如果没有指定行结束符，可能导致 CSV 文件数据解析异常。#37338 @lance6716

- 在 BR v8.2.0 之前的版本中，当集群存在 TiCDC 同步任务时，BR 不支持进行数据恢复。从 BR 8.2.0 起，BR 数据恢复对 TiCDC 的限制被放宽：如果所恢复数据的 BackupTS（即备份时间）早于 Changefeed 的 CheckpointTS（即记录当前同步进度的时间戳），BR 数据恢复可以正常进行。考虑到 BackupTS 的时间通常较早，此时可以认为绝大部分场景下，当集群存在 TiCDC 同步任务时，BR 都可以进行数据恢复。#53131 @YuJuncen

4.2.2 MySQL 兼容性

- 在 v8.2.0 之前，执行带有 PASSWORD REQUIRE CURRENT DEFAULT 选项的 CREATE USER 语句会返回错误，因为 TiDB 不支持且无法解析该选项。从 v8.2.0 开始，TiDB 支持解析并忽略该选项，以便与 MySQL 兼容 #53305 @dveeden

4.2.3 系统变量

变量名	修改类型	描述
tidb_analyze_distsql_scan_concurrency	修改	最小值从 1 改为 0。当设置为 0 时，TiDB 会根据集群规模自适应调整执行 ANALYZE 时 scan 操作的并发度。
tidb_analyze_skip_column_types	修改	从 v8.2.0 开始，默认设置下，TiDB 不会收集类型为 MEDIUMTEXT 和 LONGTEXT 的列，避免潜在的 OOM 风险。
tidb_enable_historical_stats	修改	默认值从 ON 修改为 OFF，即默认关闭历史统计信息，避免潜在的稳定性问题。

变量名	修改类型	描述
tidb_executor_concurrency	修改	新增支持对 sort 算子的并发度进行设置。
tidb_sysproc_scan_concurrency	修改	最小值从 1 改为 0。当设置为 0 时，TiDB 会根据集群规模自适应调整执行内部 SQL 语句时 scan 操作的并发度。
tidb_resource_control_strict_mode	新增	SET RESOURCE GROUP 和优化器 RESOURCE_GROUP() Hint 权限控制的开关。

4.2.4 配置文件参数

配置文件	配置项	修改类型	描述
TiDB	stats-load-concurrency	修改	默认值从 5 修改为 0，最小值从 1 修改为 0。0 为自动模式，根据服务器情况，自动调节并发度。
TiDB	token-limit	修改	最大值从 18446744073709551615（64 位平台）和 4294967295（32 位平台）修改为 1048576，代表同时执行请求的 session 个数最多可以设置为 1048576，避免设置过大导致 TiDB Server OOM。
TiKV	max-apply-unpersisted-log-limit	修改	默认值从 0 修改为 1024，代表允许 apply 已经 commit 但尚未持久化的 Raft 日志的最大数量为 1024，用于降低 TiKV 节点上因 I/O 抖动导致的长尾延迟。
TiKV	server.grpc-compression-type	修改	该配置项现在也会影响 TiKV 向 TiDB 发送的响应消息的压缩算法。开启压缩可能消耗更多 CPU 资源。
TiFlash	security.redact_info_log	修改	可选值新增 marker 选项。当配置项的值设置为 marker 时，日志中的用户数据会被标记符号 ‹ › 包裹。

4.2.5 系统表

- 在系统表 INFORMATION_SCHEMA.PROCESSLIST 和 INFORMATION_SCHEMA.CLUSTER_PROCESSLIST 中新增 SESSION_ALIAS 字段，用于显示当前连接的别名。#46889 @lcwangchao

4.2.6 编译器版本

- 为了提升 TiFlash 的开发体验，编译和构建 TiDB 所需的 LLVM 的最低版本从 13.0 升级到了 17.0。如果你是 TiDB 开发者，为了保证顺利编译，请对应升级你的 LLVM 编译器版本。#7193 @Lloyd-Pottiger

4.3 废弃功能

- 以下为从 v8.2.0 开始已废弃的功能：
    - 从 v8.2.0 开始，TiDB 的配置项 enable-replica-selector-v2 被废弃。向 TiKV 发送 RPC 请求时，默认使用新版本的 Region 副本选择器。
    - 从 v8.2.0 开始，BR 快照恢复参数 --concurrency 被废弃。作为替代，你可以通过 --tikv-max-restore-concurrency 配置快照恢复阶段单个 TiKV 节点的任务最大并发数。
    - 从 v8.2.0 开始，BR 快照恢复参数 --granularity 被废弃，粗粒度打散 Region 算法默认启用。
- 以下为计划将在未来版本中废弃的功能：
    - TiDB 在 v8.0.0 引入了系统变量 tidb_enable_auto_analyze_priority_queue，用于控制是否启用优先队列来优化自动收集统计信息任务的排序。在未来版本中，优先队列将成为自动收集统计信息任务的唯一排序方式，系统变量 tidb_enable_auto_analyze_priority_queue 将被废弃。
    - TiDB 在 v8.0.0 引入了系统变量 tidb_enable_parallel_hashagg_spill，用于控制 TiDB 是否支持并行 HashAgg 进行落盘。在未来版本中，系统变量 tidb_enable_parallel_hashagg_spill 将被废弃。

    - TiDB 在 v7.5.0 引入了系统变量 tidb_enable_async_merge_global_stats，用于设置 TiDB 使用异步方式合并分区统计信息，以避免 OOM 问题。在未来版本中，分区统计信息将统一使用异步方式进行合并，系统变量 tidb_enable_async_merge_global_stats 将被废弃。
    - 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。
- 以下为计划将在未来版本中移除的功能：
    - 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。

4.4 改进提升

- TiDB
    - 支持并行执行逻辑 DDL 语句 (General DDL)。相比 v8.1.0，在使用 10 个会话并发提交不同 DDL 语句的场景下，性能提升了 3 到 6 倍 #53246 @D3Hunter
    - 改进形如 ((a = 1 and b = 2 and c > 3) or (a = 4 and b = 5 and c > 6)) and d > 3 的表达式匹配多列索引的逻辑，使其能生成更加精准的 Range #41598 @ghazalfamilyusa
    - 优化对大数据量的表进行简单查询时获取数据分布信息的性能 #53850 @you06

- 聚合的结果集能够作为 IndexJoin 的内表，使更多的复杂查询可以匹配到 IndexJoin，从而可以通过索引提升查询效率 #37068 @elsa0520
- 通过批量删除 TiFlash placement rule 的方式，提升对分区表执行 TRUNCATE、DROP 后数据 GC 的处理速度 #54068 @Lloyd-Pottiger
- 升级 Azure Identity Libraries 和 Microsoft Authentication Library 的版本，增强安全性 #53990 @hawkingrei
- 将 token-limit 的最大值设置为 1048576，避免设置过大导致 TiDB Server OOM #53312 @djshow832
- 改进对于 MPP 执行计划的列裁剪功能，以提升 TiFlash MPP 的执行性能 #52133 @yibin87
- 优化 IndexLookUp 算子在回表数据量较多（大于 1024 行）时的性能开销 #53871 @crazycs520
- 在 MPP 负载均衡时移除不包含任何 Region 的 Store #52313 @xzhangxian1008

- TiKV
  - 增加 **Compaction Job Size(files)** 指标来呈现单个 compaction job 涉及的 SST 文件数 #16837 @zhangjinpeng87
  - 默认开启提前 apply 特性，开启后，Raft leader 在多数 peer 完成 Raft log 持久化之后即可进行 apply，不再要求 leader 自身完成 Raft log 的持久化，降低少数 TiKV 抖动对写请求延迟的影响 #16717 @glorv
  - 增加 **Raft dropped messages** 事件的可观测性，以便定位写入慢的根本原因 #17093 @Connor1996
  - 增加对 ingest file 的延迟可观测性，以便排查集群的延迟问题 #17078 @LykxSassinator
  - 利用单独的线程来进行副本清理工作，保证 Raft 读写关键路径的延迟稳定 #16001 @hbisheng

- 提升正在进行 apply 的 Raft 快照数量的可观测性 #17078 @hbisheng

- PD
  - 优化提升 Region 心跳处理的性能 #7897 @nolouch @rleungx @JmPotato
  - pd-ctl 支持通过 byte 或 query 维度来查询热点 Region #7369 @lhy1024
- TiFlash
  - 减少数据高并发读取下的锁冲突，优化短查询性能 #9125 @JinheLin
  - 消除 Join 算子中对于 Join Key 的冗余拷贝 #9057 @gengliqi
  - 将 HashAgg 算子中转换两级哈希表的过程并行化 #8956 @gengliqi
  - 移除 HashAgg 算子的冗余的聚合函数以减少计算开销 #8891 @guo-shaoge
- Tools
  - Backup & Restore (BR)
    - 优化备份功能，提升在大量表备份过程中遇到节点重启、扩容或网络抖动时的备份性能和稳定性 #52534 @3pointer
    - 优化恢复过程中对 TiCDC Changefeed 的细粒度检查，如果 Changefeed 的 CheckpointTS 晚于数据的备份时间，则不会影响恢复操作，从而减少不必要的等待时间，提升用户体验 #53131 @YuJuncen
    - 为 BACKUP 语句和 RESTORE 语句添加了多个常用参数选项，例如 CHECKSUM_CONCURRENCY #53040 @RidRisR
    - 去掉除了 br log restore 子命令之外其它 br log 子命令对 TiDB domain 数据结构的载入，降低内存消耗 #52088 @Leavrth
    - 支持对日志备份过程中生成的临时文件进行加密 #15083 @YuJuncen

        - 在 Grafana 面板中新增 tikv_log_backup_pending_initial_scan 监控指标 #16656 @3pointer
        - 优化 PITR 日志的输出格式，并在日志中新增 RestoreTS 字段 #53645 @dveeden
    - TiCDC
        - 支持当下游为消息队列 (Message Queue, MQ) 或存储服务时直接输出原始事件 #11211 @CharlesCheung96

4.5 错误修复

- TiDB
    - 修复当 SQL 语句包含 Outer Join，且 Join 条件包含 false IN (column_name) 表达式时，查询结果缺少部分数据的问题 #49476 @ghazalfamilyusa
    - 在收集表中 PREDICATE COLUMNS 的统计信息时，不再收集系统表中列的统计信息 #53403 @hi-rustin
    - 修复系统变量 tidb_persist_analyze_options 为 OFF 时，系统变量 tidb_enable_column_tracking 未生效的问题 #53478 @hi-rustin
    - 修复在 (*PointGetPlan).StatsInfo() 执行过程中可能遇到数据竞争的问题 #49803 #43339 @qw4990
    - 修复在包含数据修改操作的事务中查询带有虚拟列的表时，查询结果可能错误的问题 #53951 @qw4990
    - 修复在自动收集统计信息时，系统变量 tidb_enable_async_merge_global_stats 和 tidb_analyze_partition_concurrency 未生效的问题 #53972 @hi-rustin
    - 修复查询 TABLESAMPLE 时可能遇到 plan not supported 报错的问题 #54015 @tangenta
    - 修复执行 SELECT DISTINCT CAST(col AS DECIMAL), CAST(col AS SIGNED) FROM ... 查询时结果出错的问题 #53726 @hawkingrei

- 修复在客户端读取数据超时后查询无法被终止的问题 #44009 @wshwsh12
- 修复 Longlong 类型在谓词中溢出的问题 #45783 @hawkingrei
- 修复窗口函数中有某些子查询时可能会 panic 的问题 #42734 @hi-rustin
- 修复 TopN 算子可能被错误地下推的问题 #37986 @qw4990
- 修复在聚簇索引作为谓词时 SELECT INTO OUTFILE 不生效的问题 #42093 @qw4990
- 修复 information schema 缓存未命中导致 stale read 查询延迟上升的问题 #53428 @crazycs520
- 修复 YEAR 类型的列与超出范围的无符号整数进行比较导致错误结果的问题 #50235 @qw4990
- 修复重启 TiDB 后，主键列统计信息中的直方图和 TopN 未被加载的问题 #37548 @hawkingrei
- 修复 Massively Parallel Processing (MPP) 中 final AggMode 和 non-final AggMode 无法共存的问题 #51362 @AilinKid
- 修复执行谓词总是为 true 的 SHOW ERRORS 语句导致 TiDB panic 的问题 #46962 @elsa0520
- 修复在递归 CTE 中无法使用视图的问题 #49721 @hawkingrei
- 修复 TiDB 启动加载统计信息时可能因为 GC 推进报错的问题 #53592 @you06
- 修复使用 PREPARE/EXECUTE 方式执行带 CONV 表达式的语句，且 CONV 表达式包含 ? 参数时，多次执行可能导致查询结果错误的问题 #53505 @qw4990
- 修复将数据从 FLOAT 类型转换为 UNSIGNED 类型时结果错误的问题 #41736 @guo-shaoge
- 修复创建带有外键的表时，TiDB 未创建对应的统计信息元信息 (stats_meta) 的问题 #53652 @hawkingrei

- 修复查询中的某些过滤条件可能导致 planner 模块发生 invalid memory address or nil pointer dereference 报错的问题 #53582 #53580 #53594 #53603 @YangKeao
- 修复并发执行 CREATE OR REPLACE VIEW 可能报错 table doesn't exist 的问题 #53673 @tangenta
- 修复 INFORMATION_SCHEMA.TIDB_TRX 表中 STATE 字段的 size 未定义导致 STATE 显示为空的问题 #53026 @cfzjywxk
- 修复关闭 tidb_enable_async_merge_global_stats 时，分区表的全局统计信息中的 Distinct_count 信息可能错误的问题 #53752 @hawkingrei
- 修复使用 Optimizer Hints 时，可能输出错误的 WARNINGS 信息的问题 #53767 @hawkingrei
- 修复对时间类型执行取负操作结果不正确的问题 #52262 @solotzg
- 修复 REGEXP() 函数对空模式参数未显式报错的问题 #53221 @yibin87
- 修复将 JSON 转换为时间格式在某些情况下可能会丢失精度的问题 #53352 @YangKeao
- 修复 JSON_QUOTE() 函数在某些情况下返回结果不正确的问题 #37294 @dveeden
- 修复执行 ALTER TABLE ... REMOVE PARTITIONING 后可能导致数据丢失的问题 #53385 @mjonss
- 修复使用 auth_socket 认证插件时，TiDB 在某些情况下未能拒绝不符合身份认证的用户连接的问题 #54031 @lcwangchao
- 修复 JSON 相关函数在某些情况下报错信息与 MySQL 不一致的问题 #53799 @dveeden
- 修复分区表在 INFORMATION_SCHEMA.PARTITIONS 中的 INDEX_LENGTH 列显示不正确的问题 #54173 @Defined2014

- 修复 INFORMATION_SCHEMA.TABLES 中 TIDB_ROW_ID_SHARDING_INFO 列显示不正确的问题 #52330 @tangenta
- 修复生成列返回非法时间戳的问题 #52509 @lcwangchao
- 修复通过分布式执行框架添加索引时，设置 max-index-length 导致 TiDB panic 的问题 #53281 @zimulala
- 修复某些情况下可以创建非法的 DECIMAL(0,0) 列类型的问题 #53779 @tangenta
- 修复使用 CURRENT_DATE() 作为列默认值时查询结果错误的问题 #53746 @tangenta
- 修复 ALTER DATABASE ... SET TIFLASH REPLICA 语句错误地给 SEQUENCE 表添加 TiFlash 副本的问题 #51990 @jiyfhust
- 修复 INFORMATION_SCHEMA.KEY_COLUMN_USAGE 表中 REFERENCED_TABLE_SCHEMA 字段显示不正确的问题 #52350 @wd0517
- 修复 AUTO_ID_CACHE=1 时，单条语句插入多行数据导致 AUTO_INCREMENT 列不连续的问题 #52465 @tiancaiamao
- 修复弃用警告的格式问题 #52515 @dveeden
- 修复 TRACE 命令在 copr.buildCopTasks 中丢失的问题 #53085 @time-and-fate
- 修复 memory_quota Hint 在子查询中可能不生效的问题 #53834 @qw4990
- 修复在某些情况下，元数据锁使用不当可能导致使用 plan cache 时写入异常数据的问题 #53634 @zimulala
- 修复在事务内的语句被 OOM 终止之后，如果在当前事务内继续执行下一条语句，可能报错 Trying to start aggressive locking while it's already started 并发生 panic 的问题 #53540 @MyonKeminta

- TiKV

- 修复将 JSON_ARRAY_APPEND() 函数下推至 TiKV 导致 TiKV panic 的问题 #16930 @dbsid
- 修复 leader 未及时清理发送失败的 snapshot 文件的问题 #16976 @hbisheng
- 修复高并发的 Coprocessor 请求可能导致 TiKV OOM 的问题 #16653 @overvenus
- 修复在线变更 raftstore.periodic-full-compact-start-times 配置项可能会导致 TiKV panic 的问题 #17066 @SpadeA-Tang
- 修复执行 make docker 和 make docker_test 失败的问题 #17075 @shunki-fujita
- 修复 **gRPC request sources duration** 在监控中显示错误的问题 #17133 @King-Dylan
- 修复设置 gRPC 消息的压缩算法 (grpc-compression-type) 对 TiKV 发送到 TiDB 的消息不起作用的问题 #17176 @ekexium
- 修复 tikv-ctl 的 raft region 命令的输出中未包含 Region 状态信息的问题 #17037 @glorv
- 修复 advance-ts-interval 配置未被用于限制 CDC 和 log-backup 模块中 check_leader 操作的 timeout，导致在某些情况下 TiKV 正常重启时 resolved_ts lag 过大的问题 #17107 @MyonKeminta

- PD
  - 修复 ALTER PLACEMENT POLICY 无法修改 placement policy 的问题 #52257 #51712 @jiyfhust
  - 修复写热点调度可能会违反放置策略 (placement policy) 约束的问题 #7848 @lhy1024
  - 修复使用 Placement Rules 的情况下，down peer 可能无法恢复的问题 #7808 @rleungx
  - 修复取消资源组查询导致大量重试的问题 #8217 @nolouch
  - 修复手动切换 PD leader 可能失败的问题 #8225 @HuSharp
- TiFlash

- 修复在含空分区的分区表上执行查询时，可能会超时的问题 #9024 @JinheLin
- 修复在存算分离架构下，DDL 新增带有 not null 属性的列后，查询可能返回错误的 null 值的问题 #9084 @Lloyd-Pottiger
- 修复函数 SUBSTRING_INDEX() 可能导致 TiFlash Crash 的问题 #9116 @wshwsh12
- 修复通过 BR 或 TiDB Lightning 导入数据后，FastScan 模式下可能读到大量重复行数据的问题 #9118 @JinheLin

- Tools
  - Backup & Restore (BR)
    - 修复由于 EndKey 为空导致恢复事务 KV 集群失败的问题 #52574 @3pointer
    - 修复 PD 连接失败导致日志备份 advancer owner 所在的 TiDB 可能崩溃的问题 #52597 @YuJuncen
    - 修复日志备份在 advancer owner 发生迁移后可能被暂停的问题 #53561 @RidRisR
    - 修复在恢复过程中，由于多层重试导致 BR 无法正确识别错误的问题 #54053 @RidRisR
    - 修复用于获取 TiKV 配置的连接可能未被关闭的问题 #52595 @RidRisR
    - 修复测试用例 TestStoreRemoved 不稳定的问题 #52791 @YuJuncen
    - 修复 PITR 恢复过程中 TiFlash 崩溃的问题 #52628 @RidRisR
    - 修复增量备份过程中扫描 DDL 作业的效率较低的问题 #54139 @3pointer
    - 修复断点备份过程中查找 Region leader 中断导致备份性能受影响问题 #17168 @Leavrth
  - TiCDC

        - 修复 Grafana 监控中的 **Kafka Outgoing Bytes** 面板显示不准确的问题 #10777 @asddongmen
        - 修复在多节点环境下进行大量 UPDATE 操作时，反复重启 Changefeed 可能导致的数据不一致问题 #11219 @lidezhu
    - TiDB Data Migration (DM)
        - 升级 go-mysql 以修复连接阻塞的问题 #11041 @D3Hunter
        - 修复同步 MariaDB 数据时 SET 语句导致 DM panic 的问题 #10206 @dveeden
    - TiDB Lightning
        - 修复 TiDB Lightning 导入 zstd 压缩文件时可能报错的问题 #53587 @lance6716
    - Dumpling
        - 修复 Dumpling 在同时导出表和视图时报错的问题 #53682 @tangenta
    - TiDB Binlog
        - 修复开启 TiDB Binlog 后，在 ADD COLUMN 执行过程中删除行可能报错 data and columnID count not match 的问题 #53133 @tangenta

4.6 贡献者

感谢来自 TiDB 社区的贡献者们：

- CabinfeverB
- DanRoscigno（首次贡献者）
- ei-sugimoto（首次贡献者）
- eltociear
- jiyfhust
- michaelmdeng（首次贡献者）
- mittalrishabh
- onlyacat

- qichengzx（首次贡献者）
- SeaRise
- shawn0915
- shunki-fujita（首次贡献者）
- tonyxuqqi
- wwu（首次贡献者）
- yzhan1