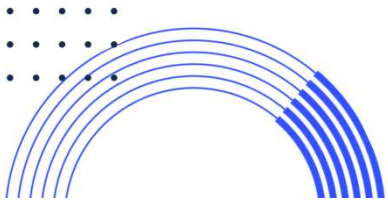
PingCAP
EDUCATION

数据库 OOM 问题诊断及处理之 TiDB Server



Copyright ©2022 PingCAP. All Rights Reserved.

课程目标

PingCAP EDUCATION

- 掌握判断 `TiDB Server OOM` 现象的方法
- 诊断 `TiDB Server OOM` 原因
- 掌握缓解 `TiDB Server OOM` 的措施

Copyright ©2022 PingCAP. All Rights Reserved.

TiDB Server OOM 对业务的影响

PingCAP EDUCATION

- `TiDB Server` 上的业务 `SQL` 会失败
- 业务响应时间升高
- 前端体验变差

Copyright ©2022 PingCAP. All Rights Reserved.

TiDB Server OOM 的诊断方法（一）

PingCAP EDUCATION

- 客户端应用
- 日志
  - `dmesg -T | grep tidb-server` 结果中有事故发生附近时间点的 `OOM-killer` 的日志
  - `tidb.log`（事故发生后附近时间的 `"Welcome to TiDB"` 的日志 < 即 `TiDB server` 发生重启 >）
  - `tidb_stderr.log`（`grep` 到 `fatal error`: `"runtime: out of memory"` 或 `"cannot allocate memory"`）

```
ERROR 2013 (HY000): Lost connection to MySQL server during query
```

Copyright ©2022 PingCAP. All Rights Reserved.

- tidb_stderr.log
  - grep 到 fatal error: "runtime: out of memory" 或 "cannot allocate memory"。

TiDB Server OOM 的诊断方法（二）

PingCAP EDUCATION

- Grafana 监控
  - TiDB --> Server --> Memory Usage

Copyright ©2022 PingCAP. All Rights Reserved.

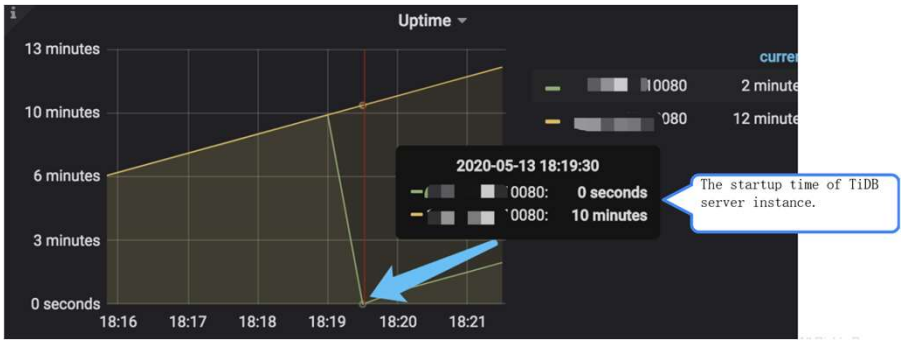
- Grafana 监控
  - TiDB --> Server --> Memory Usage ，可以观察到 TiDB server 实例，呈现内存迅速升高，又急速下降的现象

  - TiDB --> Server --> Uptime ，可以观察到 TiDB server 实例的启动时间，可以和 『Memory Usage』配合一起使用确认 TiDB Server 实例的运行情况，如是否在内存使用率飙高的同时，TiDB Server 发生的重启现象

TiDB Server OOM 的诊断方法（三）

PingCAP EDUCATION

- **Grafana** 监控
- **TiDB --> Server --> Uptime**，和 **『Memory Usage』** 配合一起使用，确认 **TiDB Server** 实例的运行情况，观察是否同时有内存使用率飙高的情况



Copyright ©2022 PingCAP. All Rights Reserved.

- Grafana 监控
  - TiDB --> Server --> Memory Usage ，可以观察到 TiDB server 实例，呈现内存迅速升高，又急速下降的现象

  - TiDB --> Server --> Uptime ，可以观察到 TiDB server 实例的启动时间，可以和 『Memory Usage』配合一起使用确认 TiDB Server 实例的运行情况，如是否在内存使用率飙高的同时，TiDB Server 发生的重启现象

造成 TiDB Server OOM 的原因

PingCAP EDUCATION

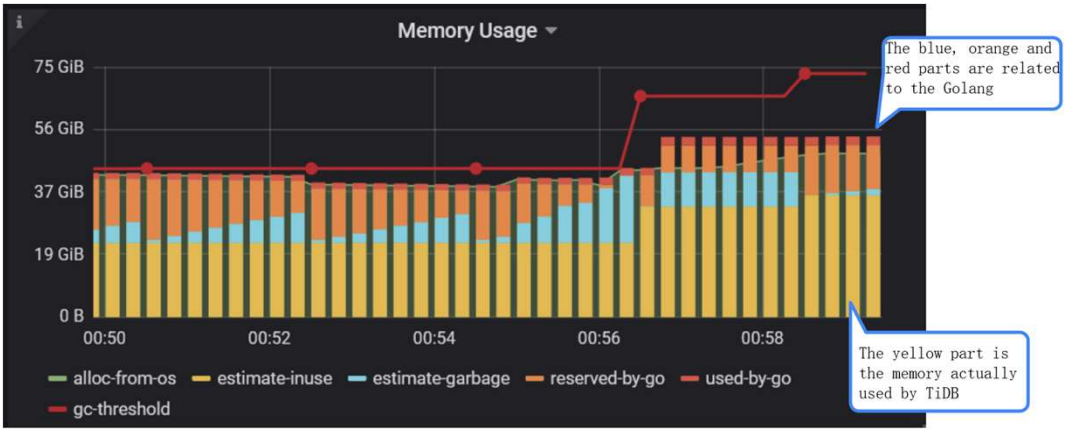
- SQL 语句的执行
- Golang 内存释放机制

Copyright ©2022 PingCAP. All Rights Reserved.

Grafana TiDB-Runtime 监控

PingCAP EDUCATION

- **TiDB-Runtime --> Memroy Usage**，确认造成 **TiDB Server OOM** 的原因



Copyright ©2022 PingCAP. All Rights Reserved.

- 最底部的黄色部分是 TiDB 真正使用的内存，TiDB 能够自行管控的也只有这一部分，除此之外。剩下的用量，TiDB 没有任何手段去干预。如果是黄色的部分用量不符合预期，可以按照正常的方式去排查，比如大查询，heap profile 等等。

除此之外，任何其他部分的用量高都是 Go 的责任，TiDB 自身完全无法控制，以下分情况介绍原因和可能的处理方案
- 蓝色部分，表示垃圾部分，这个部分接受 TiDB 控制，即使用内存越多，这部分也会越多。用堆空间的放大，降低回收器压力
- 橙色部分，这个部分一般和历史上最大内存用量有关系，但是当堆增长的时候会复用，所以不会引起 OOM 可以忽略。
- 红色部分 (used-by-go)，这部分也和历史最大堆大小有关，但是这部分不是堆内存，也不会释放。

定位内存占用大的 SQL

PingCAP EDUCATION

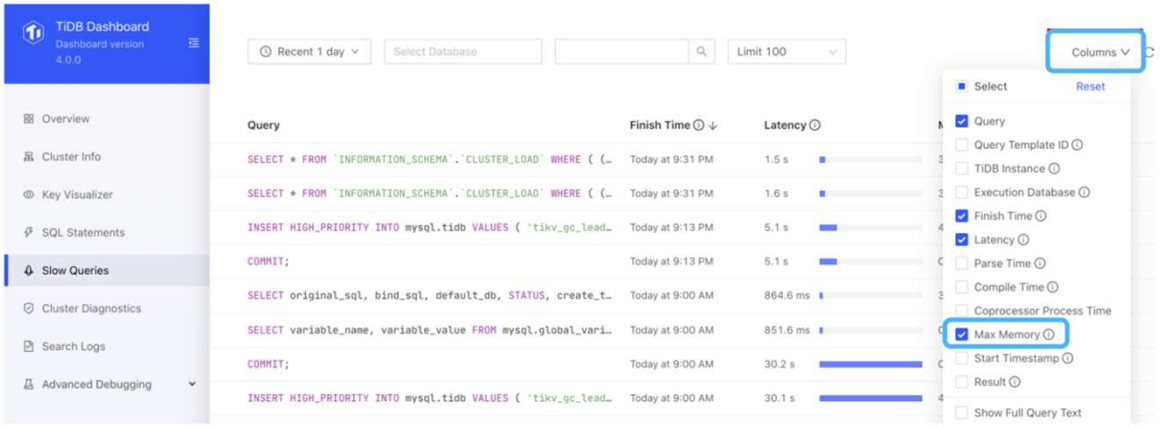
- **TiDB Dashboard** 慢查询
- **TiDB Dashboard SQL** 语句分析
- **TiDB Server** 日志中的 **expensive query**

Copyright ©2022 PingCAP. All Rights Reserved.

- TiDB Dashboard 慢查询
- TiDB Dashboard SQL 语句分析
- TiDB Server 日志中的 expensive query

TiDB Dashboard 慢查询

- 将时间调整到 **TiDB Server OOM** 前，并按照 **Max Memory** 进行排序，可以找到集群中消耗内存较大的 **SQL** 语句

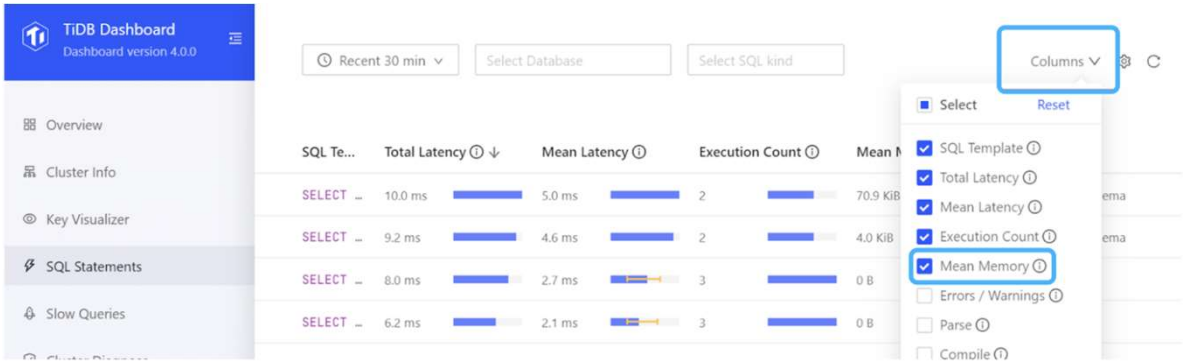


Copyright ©2022 PingCAP. All Rights Reserved.

将时间调整到 TiDB Server OOM 前，并按照 Max Memory 进行排序。这样，我们就可以找到集群中消耗内存量的 SQL list。

TiDB Dashboard SQL 语句分析

- 将时间调整到 **TiDB Server OOM** 前，并按照 **Max Memory** 进行排序，可以找到集群中消耗内存较大的 **SQL** 语句



Copyright ©2022 PingCAP. All Rights Reserved.

TiDB Server 日志 - expensive query

PingCAP EDUCATION

- 当一条语句在执行过程中达到资源使用阈值时（执行时间< **tidb_expensive_query_time_threshold** >/使用内存量< **mem-quota-query** >），**TiDB** 会即时将这条语句的相关信息写入 **tidb-server** 日志

```
[2020/02/05 15:32:25.096 +08:00] [WARN] [expensivequery.go:167] [expensive_query] [cost_time=60.008338935s]
[wait_time=0s] [request_count=1] [total_keys=70] [process_keys=65] [num_cop_tasks=1] [process_avg_time=0s]
[process_p90_time=0s] [process_max_time=0s] [process_max_addr=10.0.1.9:20160] [wait_avg_time=0.002s]
[wait_p90_time=0.002s] [wait_max_time=0.002s] [wait_max_addr=10.0.1.9:20160] [stats=t:pseudo] [conn_id=60026]
[user=root] [database=test] [table_ids="[122]"] [txn_start_ts=414420273735139329] [mem_max="1035 Bytes (1.0107421875
KB)"] [sql="insert into t select sleep(1) from t"]
```

Copyright ©2022 PingCAP. All Rights Reserved.

缓解 TiDB Server OOM 的措施

PingCAP EDUCATION

- SQL 优化，减少非必要返回的数据量
- 减少大事务，将大事务拆分为小事务
- 调整 TiDB Server 相关参数，来限制单条 SQL 内存的使用
- 其他缓解 TiDB Server OOM 的方法

Copyright ©2022 PingCAP. All Rights Reserved.

缓解 TiDB Server OOM 的措施

- SQL 优化，减少非必要返回的数据量
  - SQL 业务逻辑优化，确认数据库端返客户端的数据量是否合理
  - SQL 执行计划优化，确认 SQL 执行计划的准确性，如是否存在非必要的全表扫描，以及表关联方式是否合理等
- 减少大事务，将大事务拆分为小事务
- 调整 TiDB Server 参数，如调整单条 SQL 可使用的内存参数等
- 其他缓解 TiDB Server OOM 的方法，如调整 GO 内存释放机制等

SQL 优化

PingCAP EDUCATION

- SQL 业务逻辑优化
  - TiDB Dashboard 慢查询
  - TiDB Dashboard SQL 语句分析
  - TiDB Server 日志中的 expensive query
- 执行计划优化
  - 优化器没有选择最优执行计划
  - 未创建必要索引
  - 执行计划准确，但内存占用高

Copyright ©2022 PingCAP. All Rights Reserved.

需要跟业务开发侧的相关人员确认返回大量数据的合理性，以及是否可在满足业务要求的情况下，缩小数据范围。

- SQL 执行计划优化

通过前面 SQL 优化的相关课程，我们已经学习了 TiDB SQL 优化相关的内容，故此处不再赘述，只做简要的说明，如有疑问可再次学习 SQL 优化的课程。

  - 步骤 1：通过 TiDB Dashboard 慢查询以及 SQL 语句分析功能模块，可查看集群中内存占用大的 SQL 的执行计划，以及执行计划的个数等信息
  - 步骤 2：通过步骤 1，基本可以判断执行计划是否准确性

    - 优化器选择了效率低的执行计划
      - 可通过下面的方式干预优化器选择成本低，但访问效率高的执行计划：
        - Hint，需要改写 SQL，对前端业务有一定的入侵性
        - SPM，在不修改 SQL 语句的情况下干预执行计划的选择，自动捕获绑定、自动演进绑定等
      - 检查统计信息的健康度
        - SQL 执行计划中，如果出现了 pesudo 关键字时，表示无可用统计信息，需要对表或索引进行 analyze
        - 如果统计信息健康度在 100%，那么可能是 CBO 优化器成本估算出现了不准确的情况，可使用 SQL Binding 在非改写 SQL 的情况下，固定该 SQL 的执行计划

    - 未创建必要索引
      - 根据业务访问目标表的行为特点，创建必要的索引，在提高 SQL 访问效率的同时，减少非必要的数据返回，降低内存占用

    - 执行计划准确，但内存占用高
      - 在统计信息，以及优化器选择均准确的情况下，也有可能出现内存占用高的情况，如

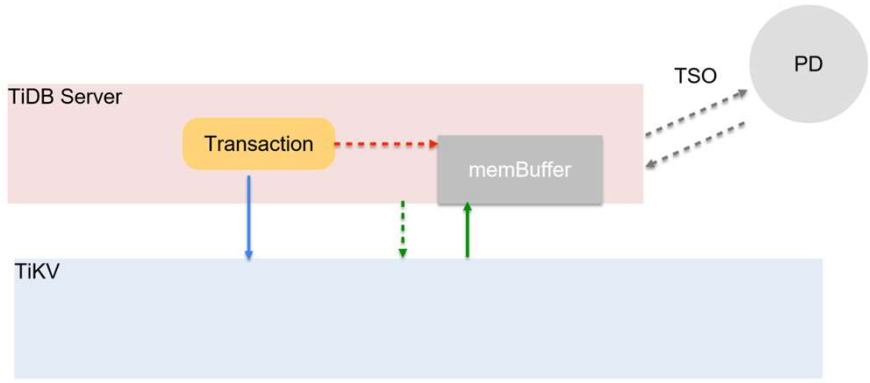
Hash Join 的 inner 表占用内存比较大，此时可考虑 Merge Join

- Merge Join 是一种特殊的 Join 算法。当两个关联表要 Join 的字段需要按排好的顺序读取时，适用 Merge Join 算法。Merge Join 占用的内存要远低于 Hash Join，但 Merge Join 不能并发执行，并且执行时间会更久。当数据量太大，或系统内存不足时，建议尝试使用。

拆分大事务

PingCAP EDUCATION

- 建议将大事务拆分为小事务，在一定程度上，提高事务提交成功的概率



Copyright ©2022 PingCAP. All Rights Reserved.

建议将大事务拆分为小事务，在一定程度上，提高事务提交成功的概率，以及降低事务回滚所带来的代价

调整 TiDB Server 参数（一）

PingCAP EDUCATION

- 单条 SQL 占用的最大内存 & 超出最大内存后 SQL 的行为
  - mem-quota-query
  - oom-action

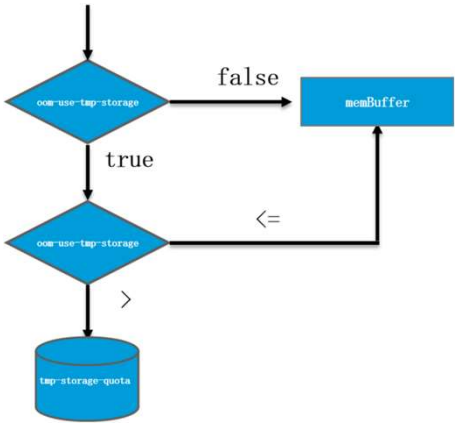
Copyright ©2022 PingCAP. All Rights Reserved.

调整 TiDB Server 参数
单条 SQL 占用的最大内存 & 超出最大内存后 SQL 的行为
- mem-quota-query
  - 单条 SQL 语句可以占用的最大内存阈值，单位为字节，默认值为 1073741824
  - 超过该值的请求会被 oom-action 定义的行为所处理，并且该值作为系统变量 tidb_mem_quota_query 的初始值
- oom-action
  - 当 TiDB 中单条 SQL 的内存使用超出 mem-quota-query 限制且不能再利用临时磁盘时的行为
  - 默认值："cancel"。设置为 "log" 时，仅输出日志。设置为 "cancel" 时，取消执行该 SQL 操作，并输出日志。

调整 TiDB Server 参数（二）

PingCAP EDUCATION

- 启用临时磁盘
  - **oom-use-tmp-storage**
  - **tmp-storage-path**
  - **tmp-storage-quota**



Copyright ©2022 PingCAP. All Rights Reserved.

- oom-use-tmp-storage
  - 设置是否在单条 SQL 语句的内存使用超出 mem-quota-query 限制时为某些算子启用临时磁盘。默认为 true
- tmp-storage-path
  - 单条 SQL 语句的内存使用超出 mem-quota-query 限制时，某些算子的临时磁盘存储位置。
  - 默认值为：<操作系统临时文件夹>/<操作系统用户 ID>_tidb/<host>:<port>/<statusHost>:<statusPort> 的 base64 编码结果/tmp-storage。
- tmp-storage-quota
  - tmp-storage-path 存储使用的限额，单位是 Byte。当单条 SQL 语句使用临时磁盘，导致 TiDB Server 的总体临时磁盘总量超过 tmp-storage-quota 时，当前 SQL 操作会被取消，并返回 Out Of Global Storage Quota! 错误。
  - 当 tmp-storage-quota 小于 0 时则没有上述检查与限制。

其他减少内存占用的方法

PingCAP EDUCATION

- 调整 GO 内存释放策略
  - GO 的 Runtime 在释放内存返回到内核时，在 Linux 平台下有以下两种策略：
    - MADV_DONTNEED
    - MADV_FREE
  - 在 TiDB Server 启动前添加 GODEBUG=madvdontneed=1 环境变量
- 滚动重启 TiDB Server

Copyright ©2022 PingCAP. All Rights Reserved.

其他减少内存占用的方法
通过 Grafana 里的 TiDB-Runtime --> Memroy Usage 面板，确认如果是非 SQL 造成的内存使用异常问题，即非黄色部分占用异常，造成可尝试使用下述方法：

- 调整 GO 内存释放策略
GO 的 Runtime 在释放内存返回到内核时，在 Linux 平台下有以下两种策略：
  - MADV_DONTNEED：会让 RSS 数量下降得很快，但是效率比较低
  - MADV_FREE ：Runtime 在释放内存时，使用了更加高效的 MADV_FREE ，即内核会等到内存紧张时才会释放。在释放之前，这块内存依然可以复用。这个特性从 Linux 4.5 版本内核开始支持
在 TiDB Server启动前添加 GODEBUG=madvdontneed=1 环境变量，GC 时立即将内存返还给操作系统，降低内存占用。

- 滚动重启 TiDB Server

课堂测试

PingCAP EDUCATION

- 1) 以下哪些因素，会引起 `TiDB Server` 内存使用异常？（请选 3 项）
  - A.大事务造成内存放大
  - B. `SQL` 执行计划错乱造成返回给 TiDB 大量非必要数据
  - C. `GO` 的内存回收机制，也可能导致内存占用量呈现异常的情况
  - D. `ADD Index` 操作，在 `write reorg` 阶段也会大量占用内存

Copyright ©2022 PingCAP. All Rights Reserved.

答案：a、b、c

课堂测试

PingCAP EDUCATION

- 2) 在 TiDB Server 中，调整哪些参数，可以限制单条 SQL 的内存用量？（请选 2 项）
  - A. mem-quota-query 来设置单条 SQL 语句可以占用的最大内存阈值
  - B. oom-use-tmp-storage 开启临时磁盘功能
  - C. rocksdb.defaultcf.block-cache-size 调整 RocksDB defaultcf 的内存占用量
  - D. rocksdb.writecf.block-cache-size 调整 RocksDB writecf 的内存占用量

Copyright ©2022 PingCAP. All Rights Reserved.

答案：a、b

课程小结

PingCAP EDUCATION

- 掌握判断 `TiDB Server OOM` 现象方法
- 诊断 `TiDB Server OOM` 原因
- 掌握缓解 `TiDB Server OOM` 的措施

Copyright ©2022 PingCAP. All Rights Reserved.

PingCAP EDUCATION

Thanks

Copyright ©2022 PingCAP. All Rights Reserved.