分布式数据库中间件

常见问题

文档版本	05
发布日期	2022-08-18

华为技术有限公司

商标声明

HUAWEI和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 DDM 通用类

1.1 DDM 提供哪些高可靠保障

数据完整性

DDM实例故障不会影响数据的完整性。

- 业务数据存储在数据节点分片中，DDM不存储业务数据。
- 逻辑库与逻辑表等配置信息存储在DDM数据库中，DDM数据库主备高可用。

高可用机制

DDM采用多个无状态节点集群式部署模式，通过弹性负载均衡地址提供服务。

- DDM节点自身宕机类故障，对于已建立在故障节点上的连接会断连报错，DDM集群整体服务不受影响，通常情况下可在5秒内将故障节点从集群中剔除。
- 下挂数据节点故障，通常情况下可以在下挂数据节点恢复后30秒内完全恢复正常服务能力。

1.2 如何选择和配置安全组

DDM实例采用了VPC和安全组等网络安全保护措施，以下内容帮助您正确配置安全组。

通过 VPC 内网访问 DDM 实例

DDM实例的访问和使用，包括客户端所在ECS访问DDM实例，以及DDM实例访问其关联的数据节点。

除了ECS、DDM实例、数据节点必须处于相同VPC之外，还需要他们的安全组分别配置了正确的规则，允许网络访问。

1. 建议ECS、DDM、数据节点配置相同的安全组。安全组创建后，默认包含同一安全组内网络访问不受限制的规则。
2. 如果配置了不同安全组，可参考如下配置方式：

说明

- 假设ECS、DDM、RDS分别配置了安全组：sg-ECS、sg-DDM、sg-RDS。
- 假设DDM实例服务端口为5066，RDS for MySQL实例服务端口为3306。
- 以下规则，远端可使用安全组，也可以使用具体的IP地址。

ECS所在安全组需要增加图1-1中的规则，以保证客户端能正常访问DDM实例：

图 1-1 ECS 安全组策略

DDM所在安全组需要增加图1-2和图1-3中的规则，以保证能访问数据节点，且被客户端访问。

图 1-2 DDM 安全组入方向配置

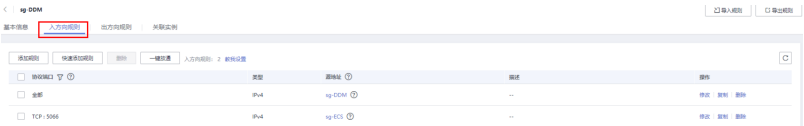
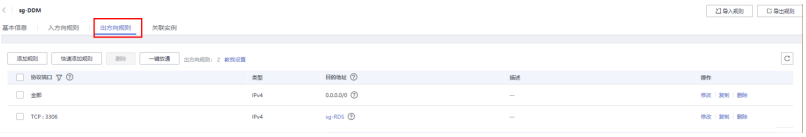


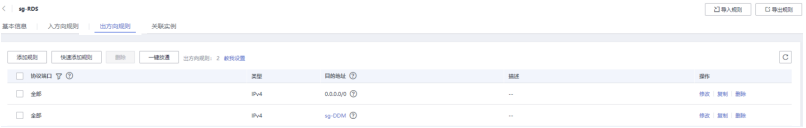
图 1-3 DDM 安全组出方向配置



数据节点所在安全组需要增加图1-4和图1-5中的规则，以保证能被DDM访问。

图 1-4 RDS 安全组入方向配置

图 1-5 RDS 安全组出方向配置



1.3 数据库时间与北京时间相差 13 或 14 小时该如何解决

问题现象

数据库时区设置为北京时间时，通过JDBC连接DDM，查询到的时间与北京时间相差13或14小时。

原因分析

JDBC驱动连接DDM时会向DDM查询数据库时区设置，DDM返回时区为CST(中国标准时间)。

CST有4种含义：

- 美国中部时间 Central Standard Time (USA) UTC-06:00
- 澳大利亚中部时间 Central Standard Time (Australia) UTC+09:30
- 中国标准时 China Standard Time UTC+08:00
- 古巴标准时 Cuba Standard Time UTC-04:00

在JDBC驱动中，会将CST时间解析为美国中部时间，与北京时间相差了13或14个小时。

解决方法

在JDBC连接数据库的字符串中添加时区配置项：

jdbc:mysql://*xxx*:3306/*database_name*?serverTimezone=Asia/Shanghai&useUnicode=true&characterEncoding=utf8

1.4 一个 DDM 实例关联的不同数据节点之间是否可以共享数据

一个DDM实例关联的不同数据节点之间数据是互相独立的，无法共享。

1.5 DDM 实例关联的数据节点需要满足什么条件

DDM实例关联的数据节点需满足以下条件：

- 数据节点的状态为正常运行中。
- 数据节点和DDM实例处于相同VPC。
- 数据节点没有被其它DDM实例关联。

2 DDM 使用类

2.1 DDM 如何进行分片

在分布式数据库中，可以通过分片存储方式，轻松解决大数据量单表容量达到单机数据库存储上限的瓶颈，因此创建逻辑库和逻辑表时，需要根据实际情况确定逻辑表是否进行分片以及按什么规则分。

说明

分片存储后，需要尽量避免跨库JOIN操作带来的性能与资源消耗问题。

- 逻辑表是否分片

  DDM逻辑表支持全局表、拆分表、单表三种类型。用户可以按照数据表的实际使用需求，选择最合适的逻辑表类型创建，实际操作请参考创建表。

  - 单表只在第一个分片创建表以及存储数据。
  - 全局表在每一个分片创建表并且存储全量数据。
  - 拆分表在每一个分片创建表，数据按照拆分规则分散存储在分片中。
- 按什么规则分

  逻辑表的拆分键选择非常重要。建议按实际业务场景选择拆分键，不同逻辑表，如果具有E-R关系，建议选择相同字段做拆分键，避免跨库JOIN操作。

在实际使用中，请结合以下建议评估是否进行分片：

- 数据量在1000万条以下的表，不建议分片。
- 数据量在1000万条以上的表，建议分片。将数据分片存储后，既能解决单张表容量过大带来的性能瓶颈，同时提高并发支持。注意要选择合适的拆分键，提前做好规划。
- 业务读取尽量少用多表JOIN，同一个事务避免跨分片。
- 查询条件尽量带上拆分键，避免全拆分表扫描。

2.2 如何解决 JDBC 驱动方式连接 DDM 异常问题

MySQL驱动（ JDBC ）通过Loadbalance方式连接DDM，在某些场景下连接切换时会陷入死循环，最终导致栈溢出。

问题定位

1. 查看APP日志，定位异常原因。

   例如，从以下日志中分析出异常最终原因为栈溢出。

```
Caused by: java.lang.StackOverflowError
    at java.nio.HeapByteBuffer.<init>(HeapByteBuffer.java:57)
    at java.nio.ByteBuffer.allocate(ByteBuffer.java:335)
    at java.nio.charset.CharsetEncoder.encode(CharsetEncoder.java:795)
    at java.nio.charset.Charset.encode(Charset.java:843)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:2362)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:2344)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:568)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:626)
    at com.mysql.jdbc.Buffer.writeStringNoNull(Buffer.java:670)
    at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2636)
```

2. 分析溢出源。

   例如，从以下日志可以分析出，溢出原因为驱动内部陷入死循环。

```
at
com.mysql.jdbc.LoadBalancedConnectionProxy.pickNewConnection(LoadBalancedConnectionProxy.java:
344)
at
com.mysql.jdbc.LoadBalancedAutoCommitInterceptor.postProcess(LoadBalancedAutoCommitIntercepto
r.java:104)
at com.mysql.jdbc.MysqlIO.invokeStatementInterceptorsPost(MysqlIO.java:2885)
at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2808)
at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2483)
at com.mysql.jdbc.ConnectionImpl.setReadOnlyInternal(ConnectionImpl.java:4961)
at com.mysql.jdbc.ConnectionImpl.setReadOnly(ConnectionImpl.java:4954)
at com.mysql.jdbc.MultiHostConnectionProxy.syncSessionState(MultiHostConnectionProxy.java:381)
at com.mysql.jdbc.MultiHostConnectionProxy.syncSessionState(MultiHostConnectionProxy.java:366)
at
com.mysql.jdbc.LoadBalancedConnectionProxy.pickNewConnection(LoadBalancedConnectionProxy.java:
344)
```

3. 查看使用的MySQL版本，为5.1.44。

   查看该版本源代码，发现获取连接时，**LoadBalance**会根据负载均衡策略更新连接，并将老连接的配置复制给新连接，在新连接**AutoCommit**为**true**，新连接部分参数和老连接不一致，**loadBalanceAutoCommitStatementThreshold**参数没有配置的场景下，会陷入死循环，更新连接函数调用同步参数函数，同步参数又调用更新连接，最终导致栈溢出。

解决方法

在连接DDM的URL添加**loadBalanceAutoCommitStatementThreshold=5&retriesAllDown=10**参数。

```
//使用负载均衡的连接示例
//jdbc:mysql:loadbalance://ip1:port1,ip2:port2..ipN:portN/{db_name}
String url = "jdbc:mysql:loadbalance://192.168.0.200:5066,192.168.0.201:5066/db_5133?
loadBalanceAutoCommitStatementThreshold=5&retriesAllDown=10";
```

- loadBalanceAutoCommitStatementThreshold：表示连接上执行多少个语句后会重新选择连接。

  假设loadBalanceAutoCommitStatementThreshold设为5，则当执行5个sql后（Queries或者updates等），将会重新选择连接。若为0表示“粘性连接，不重新选择连接”。关闭自动提交时（autocommit=false）会等待事务完成再考虑是否重新选择连接。

2.3 如何选择 JDBC 驱动方式的版本和参数

DDM暂不支持使用5.1.46版本的JDBC驱动连接DDM。

JDBC驱动下载地址：**https://dev.mysql.com/doc/index-connectors.html**。

JDBC URL中推荐参数如**表2-1**所示。

表 2-1 参数

参数名称	参数说明	推荐取值
ip:port	连接地址和端口，用于连接DDM。	在DDM管理控制台DDM实例管理中查看连接地址。
db_name	连接逻辑库名称。	在DDM管理控制台，DDM实例管理 > 逻辑库管理下查看逻辑库名称。
loadBalanceAutoCommitStatementThreshold	表示连接上执行多少个语句后会重新选择连接。 ● 若取值为5，则当执行5个sql后（Queries或者updates等），将会重新选择连接。 ● 若取值为0，则表示“粘性连接，不重新选择连接”。 关闭自动提交时（autocommit=false）会等待事务完成再考虑是否重新选择连接。	5
loadBalanceHostRemovalGracePeriod	设置主机从负载均衡连接中移除的宽限时间。	15000
loadBalanceBlacklistTimeout	设置服务器在全局黑名单中存留的时间。	60000
loadBalancePingTimeout	使用负载均衡连接时，等待每个负载均衡连接ping响应的毫秒数。	5000
retriesAllDown	当所有的连接地址都无法连接时，轮询重试的最大次数。 重试次数达到阈值仍然无法获取有效连接，将会抛出SQLException。	10
connectTimeout	和数据库服务器建立socket连接时的超时。 单位：毫秒，0表示永不超时，适用于JDK 1.4及更高版本。	10000
socketTimeout	socket操作（读写）超时。 单位：毫秒， 0表示永不超时	根据业务实际情况合理配置。

2.4 使用 mysqldump 从 MySQL 导出数据非常缓慢的原因

mysqldump客户端的版本和DDM所支持的MySQL版本不一致，可能会导致从MySQL导出数据非常缓慢。

建议版本保持一致。

2.5 导入数据到 DDM 过程中出现主键重复

在DDM中创表时设置自增起始值，并确保起始值大于导入数据自增键的最大值。

2.6 如何处理数据迁移过程中自增列报错：主键重复

重新设置自增主键的初始值为大于当前已有数据的最大值，执行如下语句：

```
ALTER SEQUENCE 库名.SEQ名 START WITH 新初始值
```

2.7 如何处理配置参数未超时却报错

建议可将参数SocketTimeOut值调整或者去掉，默认为0则不断开连接。

2.8 如何处理 DDM 逻辑库与 RDS 实例的先后关系

DDM逻辑库与关联的RDS强相关，不允许直接删除关联的RDS，这会导致业务不可用且逻辑库也会删除失败。如果需要删除，先删除逻辑库再删除RDS。

2.9 DDM 逻辑库删除后，数据节点里面残留着部分预留的 DDM 数据库和一些 DDM 的账户，这些是否需要手动删除

如果不需要了，可以直接手动删除，释放空间。

3 SQL 语法类

3.1 DDM 是否支持分布式 JOIN

DDM支持分布式JOIN。

- 表设计时，增加字段冗余
- 支持跨分片的JOIN，主要实现的方式有三种：广播表，ER分片和ShareJoin。
- DDM目前禁止多个表的跨库update和delete。

3.2 如何进行 SQL 优化

- 尽量避免使用LEFT JOIN或RIGHT JOIN，建议使用INNER。
- 在使用LEFT或RIGHT JOIN时，ON会优先执行，WHERE条件在最后执行，所以在使用过程中，条件尽可能在ON语句中判断，减少WHERE的执行。
- 尽量少用子查询，改用JOIN，避免大表全表扫描。

3.3 DDM 是否支持数据类型强制转换

数据类型转换属于高级用法，DDM对SQL的兼容性会逐步完善，如有需要请提工单处理。

3.4 如何处理 INSERT 语句批量插入多条数据时报错

解决方案

建议拆分为多条INSERT语句插入。

4 RDS 相关类

4.1 数据库表名是否区分大小写

DDM默认对databaseName、tableName、columnName不区分大小写。

4.2 RDS for MySQL 哪些高危操作会影响 DDM

RDS for MySQL相关高危操作如表4-1所示。

表 4-1 RDS for MySQL 高危操作

操作类别	操作	操作影响
RDS for MySQL控制台操作类	删除RDS for MySQL实例	RDS for MySQL实例删除后，DDM关联该RDS for MySQL实例的逻辑库、逻辑表都无法使用。
	切换RDS for MySQL主备实例	切换主备实例可能造成短时间内的RDS for MySQL服务闪断，并有可能在主备同步时延过大的情况下，导致少量数据丢失。 ● RDS for MySQL实例主备切换过程中，DDM将无法进行创建逻辑库、创建表等操作。 ● RDS for MySQL实例主备切换后，DDM中RDS for MySQL实例ID不变。
	重启实例	重启过程中，RDS for MySQL实例将不可用，DDM业务将会受影响。
	重置密码	RDS for MySQL重置密码后，DDM这边创建逻辑库时输入重置后的密码即可。

操作类别	操作	操作影响
	修改参数模板	其中如下参数为固定值，如果修改，将会影响DDM正常运行。 ● 数据表名和序列名称不区分大小写，“lower_case_table_names”固定为“1”。 ● 扩容场景，必须将“local_infile”配置为“ON”。
	修改安全组	将导致DDM服务无法连接RDS for MySQL实例。
	修改VPC	DDM实例与RDS for MySQL实例不在同一VPC中将导致无法互通。
	恢复	恢复数据可能会破坏数据完整性。
RDS for MySQL客户端类	删除DDM创建的物理库	删除物理库后，原数据将会丢失，新数据将无法写入。
	删除DDM创建的物理帐号	删除物理帐号后将无法在DDM上创建逻辑表。
	删除DDM创建的物理表	删除物理表后，将导致DDM数据丢失，DDM后续无法正常使用该逻辑表。
	修改DDM创建的物理表名	将导致DDM无法获取该逻辑表的数据，且后续无法正常使用。
	修改记录	如修改全局表记录，将会影响各分片数据一致性。
	修改白名单	需要确保DDM服务在RDS for MySQL实例的白名单内，否则DDM服务将无法访问RDS for MySQL实例。

4.3 如何处理表中存在主键重复的数据

场景

DDM实例的逻辑表中已存在主键数据类型边界值的记录，如果插入的数据超过主键数据类型的范围，表中会出现主键重复的数据。

处理方法

步骤1 登录云服务管理控制台。

步骤2 在RDS for MySQL的“实例管理”页面，查找DDM实例对应的RDS for MySQL实例，单击目标RDS for MySQL实例名称，进入实例的“基本信息”页面。

步骤3 在基本信息页面的左侧导航栏中选择“参数修改”。

步骤4 在“参数”页签搜索“sql_mode”，单击“值”列中的下拉框，勾选“STRICT_ALL_TABLES”或“STRICT_TRANS_TABLES”方式，单击“保存”。

说明

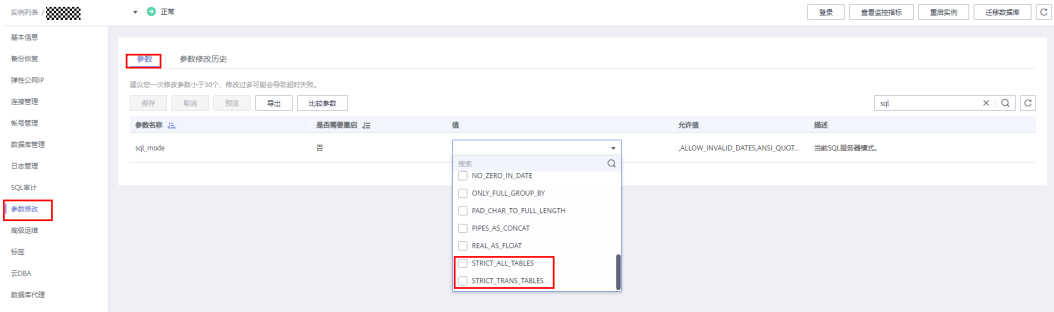
“STRICT_ALL_TABLES”和“STRICT_TRANS_TABLES”方式属于严格模式。严格模式控制MySQL如何处理非法或丢失的输入值。

- 非法：数据类型错误或超出范围。
- 丢失：如果某列定义为非空列且没有DEFAULT值，当新插入的行不包含该列时，该行记录丢失。
- 在进行扩容时，若DDM的实例版本低于2.4.1.3。在选择MySQL实例的参数sql_mode时，请不要选择ANSI_QUOTES。不能使用双引号来引用文字字符串，因为它们被解释为标识符。

  例如：select * from test where tb = "logic"。

关于“sql_mode”更多信息，请参考**Server SQL Modes**。

图 4-1 修改实例参数



步骤5 在“DDM实例管理”页面，重启DDM实例。

----结束

4.4 如何通过 show full innodb status 指令查询 RDS for MySQL 相关信息

通过MySQL客户端连接DDM实例后，可直接输入show full innodb status指令查询该DDM实例所关联的RDS for MySQL实例信息。可查询信息如：

- 当前的时间及自上次输出以来经过的时长。
- 可以使用命令show full innodb status来查看master thread的状态信息。
- 如果有高并发的工作负载，您需关注SEMAPHORES信号量，它包含了两种数据：事件计数器以及可选的当前等待线程的列表，如果有性能上的瓶颈，可使用这些信息来找出瓶颈。

5 连接管理类

5.1 MySQL 连接 DDM 时出现乱码如何解决

MySQL连接的编码和实际的编码不一致，可能导致DDM解析时出现乱码。

通过“default-character-set=utf8”指定客户端连接的编码即可。

如下所示：

```
mysql -h 127.0.0.1 -P 5066 -D database --default-character-set=utf8 -u ddmuser –p password
```

6 资源冻结/释放/删除/退订

DDM 资源为什么被释放了?

客户在华为云购买产品后，如果没有及时的进行续费或充值，将进入宽限期。如宽限期满仍未续费或充值，将进入保留期。在保留期内资源将停止服务。保留期满仍未续费或充值，存储在云服务中的数据将被删除、云服务资源将被释放。请参见资源停止服务或逾期释放说明。

DDM 资源为什么被冻结了?

资源冻结的类型有多种，最常见类型为欠费冻结。

实例被冻结了，还可以备份数据吗?

不支持，如果是欠费冻结，需要您先续费解冻DDM实例后才能备份数据。

怎样将资源解冻?

欠费冻结：用户可通过续费或充值来解冻资源，恢复DDM正常使用。欠费冻结的DDM允许续费、释放 或删除；已经到期的包周期DDM不能发起退订，未到期的包周期DDM可以退订。

冻结、解冻、释放资源时对业务的影响

- 资源冻结时：
  - 资源将被限制访问和使用，会导致您的业务中断。例如DDM被冻结时，会使得用户无法再连接至数据库。
  - 包周期资源被冻结后，将被限制进行变更操作。
  - 资源被冻结后，可以手动进行退订/删除。
- DDM底层的DN节点被冻结会导致DDM与DN节点无法正常通信，进而使DDM功能不可用。
- 资源解冻时：资源将被解除限制，用户可以连接至数据库。
- 资源释放时：资源将被释放，实例将被删除。

怎样续费?

包年/包月方式购买的DDM到期后，请在管理控制台**续费管理**页面进行续费操作。详细操作请参考**续费管理**。

资源被释放了能否恢复? /退订错了可以找回吗?

实例被删除，无法找回。

退订资源前请一定要仔细确认资源信息。如果退订错了建议重新购买使用。

怎样删除 DDM 实例?

- 按需实例，请参见**删除按需实例**。
- 包周期实例，请参见**退订包周期实例**。
- 包周期实例，请参见退订包周期实例。

A 修订记录

发布日期	修订记录
2022-08-18	第五次正式发布。 修改**如何处理表中存在主键重复的数据**。
2022-08-12	第四次正式发布。 新增**DDM实例关联的数据节点需要满足什么条件**。
2022-08-08	第三次正式发布。 新增**资源冻结/释放/删除/退订**。
2021-10-29	第二次正式发布。 ● 新增**数据库时间与北京时间相差13或14小时该如何解决**。 ● 新增**一个DDM实例关联的不同数据节点之间是否可以共享数据**。
2020-10-20	第一次正式发布。