

ALIBABA CLOUD

Alibaba Cloud

云原生分布式数据库 PolarDB-X
最佳实践

文档版本：20201111

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.如何选择实例规格	05
2.如何选择拆分键	07
3.如何选择分片数	09
4.如何高效扫描数据	10

1. 如何选择实例规格

DRDS实例与RDS实例按照CPU的处理能力、内存容量和磁盘空间等来划分实例的规格，并提供多种不同规格的实例供选择，规格越高代表实例的处理能力越强。本文介绍如何选择实例规格。

DRDS实例类型与规格的选择

DRDS实例均为专享实例，每个实例最少提供2个节点保证高可用。DRDS提供的是计算密集型的服务，处理能力主要与CPU相关，并以QPS为衡量指标。选择实例时，应参照不同规格支持的最大QPS，并结合估算的业务最大QPS来进行选择。

系列	实例规格（CPU和内存）	特点
入门版	8核16 GB	面向初期的业务开发、测试场景，不具备复杂查询加速能力。
	12核24 GB	
	16核32 GB	
	20核40 GB	
	24核48 GB	
	28核56 GB	
	32核64 GB	
标准版	16核32 GB	规格丰富，性价比高。面向具备超高并发、复杂查询及轻量分析的在线业务场景。默认提供Parallel Query并行查询能力，可有效提升对于在线业务的多表关联、聚合排序等复杂查询的执行效率。
	24核48 GB	
	32核64 GB	
	40核80 GB	
	48核96 GB	
	56核112 GB	
	64核128 GB	
	72核144 GB	
	80核160 GB	
	88核176 GB	
	96核192 GB	
	112核224 GB	
	120核240 GB	
128核256 GB		

系列	实例规格 (CPU和内存)	特点
企业版	32核64 GB	大规格资源，面向具备企业级超高并发、大规模数据复杂查询、加速分析的业务场景。默认提供Parallel Query并行查询能力，可大幅提升海量数据下复杂查询、报表分析的执行效率。
	48核96 GB	
	64核128 GB	
	96核384 GB	
	112核224 GB	
	128核256 GB	
	144核288 GB	
	160核320 GB	
	176核352 GB	
	192核384 GB	

RDS实例规格的选择

- 预估1~2年内的业务增长量，判断需要的最大磁盘空间。
- 估算一个RDS实例需要的最大IOPS。

 **说明** 建议购买多个中小规格的RDS实例，方便后期当遇到存储瓶颈时可对RDS进行快速升配。

2. 如何选择拆分键

本文将介绍如何在DRDS中选择合适的拆分键。

背景信息

拆分键即分库或分表字段，是水平拆分过程中用于生成拆分规则的数据表字段。DRDS将拆分键值通过拆分函数计算得到一个计算结果，然后根据这个结果将数据分拆到RDS实例上。

数据表拆分的首要原则是尽可能找到数据所归属的业务逻辑实体，并确定大部分（或核心的）SQL操作或者具有一定并发的SQL都是围绕这个实体进行，然后可使用该实体对应的字段作为拆分键。

示例

业务逻辑实体通常与应用场景相关，下面的一些典型应用场景都有明确的业务逻辑实体（以此类推，其它应用场景也能找到合适的业务逻辑实体），其标识型字段可用来做拆分键。

- 面向用户的互联网应用，围绕用户维度来做各种操作，那么业务逻辑实体就是用户，可使用用户ID作为拆分键。
- 侧重于卖家的电商应用，围绕卖家维度来做各种操作，那么业务逻辑实体就是卖家，可使用卖家ID作为拆分键。
- 游戏类在线应用，围绕玩家维度来做各种操作，那么业务逻辑实体就是玩家，可使用玩家ID作为拆分键。
- 车联网在线应用，围绕车辆维度来做各种操作，那么业务逻辑实体就是车辆，可使用车辆ID作为拆分键。
- 税务类在线应用，围绕纳税人来进行前台业务操作，那么业务逻辑实体就是纳税人，可使用纳税人ID作为拆分键。

例如某面向卖家的电商应用，需要对如下单表进行水平拆分：

```
CREATE TABLE sample_order (  
  id INT(11) NOT NULL,  
  sellerId INT(11) NOT NULL,  
  trade_id INT(11) NOT NULL,  
  buyer_id INT(11) NOT NULL,  
  buyer_nick VARCHAR(64) DEFAULT NULL,  
  PRIMARY KEY (id)  
)
```

确定业务逻辑实体为卖家，那么选择字段 `sellerId` 作为拆分键，则您可以使用如下分布式DDL语句建表：

```
CREATE TABLE sample_order (  
  id INT(11) NOT NULL,  
  sellerId INT(11) NOT NULL,  
  trade_id INT(11) NOT NULL,  
  buyer_id INT(11) NOT NULL,  
  buyer_nick VARCHAR(64) DEFAULT NULL,  
  PRIMARY KEY (id)  
) DBPARTITION BY HASH(sellerId)
```

如果确实找不到合适的业务逻辑实体作为拆分键，特别是传统企业级应用，那么可以考虑以下方法来选择拆分键。

- 根据数据分布和访问的均衡度来考虑拆分键，尽量将数据表中的数据相对均匀地分布在不同分表中，DRDS推出了全局强一致二级索引和Parallel Query能够提高在此场景下SQL并发度并缩短响应时间。
- 按照数字（字符串）类型与时间类型字段相结合作为拆分键，进行分库和分表，适用于日志检索类的应用场景。

例如某日志系统记录了用户的所有操作，现需要对如下日志单表进行水平拆分：

```
CREATE TABLE user_log (  
  userId INT(11) NOT NULL,  
  name VARCHAR(64) NOT NULL,  
  operation VARCHAR(128) DEFAULT NULL,  
 actionDate DATE DEFAULT NULL  
)
```

此时可以选择用户标识与时间字段相结合作为拆分键，并按照一周七天进行分表，则您可以使用如下分布式DDL语句建表：

```
CREATE TABLE user_log (  
  userId INT(11) NOT NULL,  
  name VARCHAR(64) NOT NULL,  
  operation VARCHAR(128) DEFAULT NULL,  
 actionDate DATE DEFAULT NULL  
) DBPARTITION BY HASH(userId) TBPARTITION BY WEEK(actionDate) TBPARTITIONS 7
```

更多拆分键的选择和分表形式，请参见[CREATE TABLE](#)和[拆分函数概述](#)。

3. 如何选择分片数

本文将介绍如何为DRDS中选择分片数（即水平拆分时的物理分表数）。

背景信息

DRDS中的水平拆分包含了分库和分表两个层次。若您在[创建数据库](#)时，选择拆分模式为水平拆分，则DRDS为默认为每个RDS实例创建8个物理分库，每个物理分库上可以创建一个或多个物理分表，而分表数通常也被称为分片数。

计算公式

一般情况下，建议单个物理分表的容量不超过500万行数据。通常可以预估1~2年内的数据增长量，用估算出的总数据量除以总的物理分库数，再除以建议的单个物理分表的最大数据量（即500万），即可得出每个物理分库上需要创建的物理分表数。

$$\text{物理分库上的物理分表数} = \text{向上取整} \left(\frac{\text{估算的总数据量}}{(\text{RDS实例数} \times 8) / 5,000,000} \right)$$

因此，若计算出的物理分表数等于1时，当前分库即可满足需求，您无需再进一步分表，保持当前每个物理分库上一个物理分表即可。若计算结果大于1，则建议既分库又分表，即每个物理分库上再创建多个物理分表。

示例

- 假设预估一张表在2年后的总数据量约为1亿行，您已购买了4个RDS实例，那么按照分片数公式进行如下计算：

$$\text{物理分库上的物理分表数} = \text{CEILING}(100,000,000 / (4 * 8) / 5,000,000) = \text{CEILING}(0.625) = 1$$

结果为1，那么您只需要分库而无需分表，即保持当前每个物理分库上1张物理分表即可。

- 假设预估一张表在2年后的总数据量约为1亿行，但您只购买了1个RDS实例，那么按照分片数公式进行如下计算：

$$\text{物理分库上的物理分表数} = \text{CEILING}(100,000,000 / (1 * 8) / 5,000,000) = \text{CEILING}(2.5) = 3$$

结果为3，那么建议您既分库又分表，即需要在每个物理分库上再创建3张物理分表。

4. 如何高效扫描数据

DRDS支持高效的数据扫描方式，并支持在全表扫描时使用聚合函数进行统计汇总。本文介绍如何高效扫描DRDS数据。

常见的扫描场景

- 没有分库分表：DRDS会把原SQL传递到后端MySQL执行。这种情况下DRDS支持任何聚合函数。
- 非全表扫描：SQL经过DRDS路由后，发送到单个MySQL库上执行。比如说拆分键在WHERE中是等于关系时，就会出现非全表扫描。此时同样可以支持任何聚合函数。
- 全表扫描：目前支持的聚合函数有COUNT、MAX、MIN和SUM。另外在全表扫描时同样支持LIKE、ORDER BY、LIMIT以及GROUP BY语法。
- 并行的全表扫描：如果需要从所有库导出数据，可以通过SHOW指令查看表拓扑结构，针对分表并行处理。

通过HINT进行表遍历

1. 执行 `SHOW TOPOLOGY FROM TABLE_NAME` 指令获取表拓扑结构。

```
mysql> SHOW TOPOLOGY FROM DRDS_USERS;
+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+
| 0 | DRDS_00_RDS | DRDS_USERS |
| 1 | DRDS_01_RDS | DRDS_USERS |
+-----+-----+-----+
2 rows in set (0.06 sec)
```

说明

- 非分库分表的表默认存储在第0个分库。
- 推荐每次扫描前执行SHOW TOPOLOGY FROM TABLE_NAME获取最新的表拓扑结构。

2. 针对TOPOLOGY进行单表遍历。

- 第0个分库运行当前SQL。

```
!/TDDL:node='DRDS_00_RDS'*/SELECT * FROM DRDS_USERS;
```

- 第1个分库运行当前SQL。

```
!/TDDL:node='DRDS_01_RDS'*/SELECT * FROM DRDS_USERS;
```

并行扫描

DRDS支持mysqldump指令导出数据。但如果想要更快地扫描数据，可以针对每个分表开启多个会话的方式并行加速扫描。

```
mysql> SHOW TOPOLOGY FROM LJLTEST;
```

```
+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+
| 0 | TDDL5_00_GROUP | ljltest_00 |
| 1 | TDDL5_00_GROUP | ljltest_01 |
| 2 | TDDL5_00_GROUP | ljltest_02 |
| 3 | TDDL5_01_GROUP | ljltest_03 |
| 4 | TDDL5_01_GROUP | ljltest_04 |
| 5 | TDDL5_01_GROUP | ljltest_05 |
| 6 | TDDL5_02_GROUP | ljltest_06 |
| 7 | TDDL5_02_GROUP | ljltest_07 |
| 8 | TDDL5_02_GROUP | ljltest_08 |
| 9 | TDDL5_03_GROUP | ljltest_09 |
| 10 | TDDL5_03_GROUP | ljltest_10 |
| 11 | TDDL5_03_GROUP | ljltest_11 |
+-----+-----+-----+
12 rows in set (0.06 sec)
```

如上所示该表有四个分库，每个分库有三个分表。使用以下的SQL对TDDL5_00_GROUP库上的分表进行操作。

```
!/TDDL:node='TDDL5_00_GROUP'*/select * from ljltest_00;
```

② 说明

- HINT 中的TDDL5_00_GROUP与SHOW TOPOLOGY指令结果中的GROUP_NAME列相对应。
- SQL中的表名为分表名。
- 此时可开启最多12个会话（分别对应12张分表）并行处理数据。