

单维路由

单维路由根据单个路由字段值判断SQL应当分发到哪个分库执行

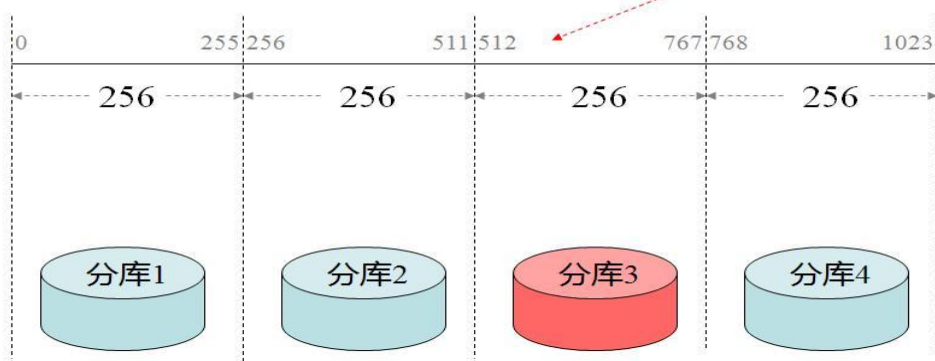
- 将[0-1023]分为不同的区段，每个区段代表一个分库，其中区段的个数以及每个区段的长度由用户自定义；
- 用SQL中的拆分字段值对1024取模，结果落入哪个区段，SQL便被路由到那个分库执行（如果拆分字段不是数值类型，则首先做一次hash）

基本示例

假设现在需要将一张表平均拆分到4个分库，路由字段为字符串类型。

- 首先将[0-1023]平均分为4个区段，[0-255],[256-511],[512-767],[768-1023]，
- 然后对字符串（或子串，由用户自定义）做hash，hash结果对1024取模
- 最终得出的结果落入那个区段，便路由到哪个分库。

`hash(pavarott) = 3170972965401 % 1024 = 537`



在Cobar配置中，我们把区段的个数成为partitionCount，每个区段的长度为partitionLength，满足 $\sum(\text{partitionCount} * \text{partitionLength}) = 1024$ 。

function定义如下：

```
<!-- 定义function名称，如果路由字段是字符串类型，对应class路径为
com.alibaba.cobar.route.function.PartitionByString-->
<function name="func"
class="com.alibaba.cobar.route.function.PartitionByString">
  <!-- 该路由函数将1024分成多少个区段-->
  <property name="partitionCount">4</property>
  <!-- 每一区段的长度，满足4 * 256 = 1024-->
  <property name="partitionLength">256</property>
  <!-- 字符串需要先取hash值再做映射，这里定义取字符串的哪一部分子串
  做hash-->
  <!-- :8表示子串的下标，意思是取路由字段的前8个字符做hash，该语法
  与python取字符串下标方式相同-->
  <property name="hashSlice">:8</property>
</function>
```

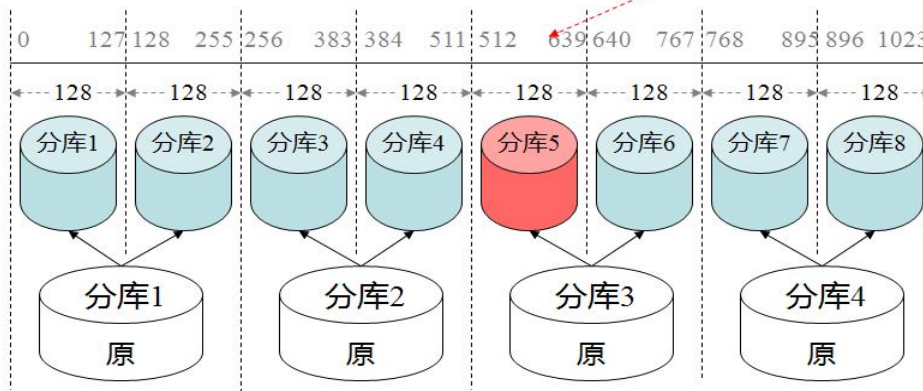
假如路由字段是数值类型，按照如下配置

```
<!-- 数值类型路由函数对应class路径为
com.alibaba.cobar.route.function.PartitionByLong-->
<function name="func"
class="com.alibaba.cobar.route.function.PartitionByLong">
  <property name="partitionCount">4</property>
  <property name="partitionLength">256</property>
</function>
```

扩容

这种路由方法可以很容易实现扩容，假设现在需要将4个分库扩容为8个分库，只需要将[0-1023]平分为8个区段即可，如下：

$\text{hash}(\text{pavarott}) = 3170972965401 \% 1024 = 537$

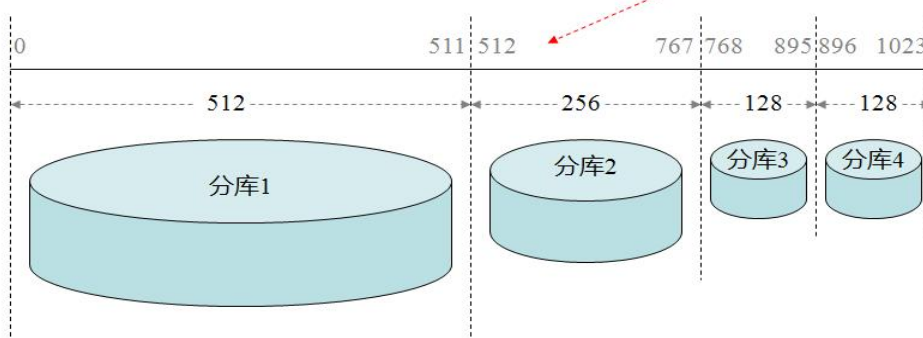


```
<!--这里仅列出按照数据类型字段路由的扩容方式，按照字符串路由的情况类似-->
<function name="func"
class="com.alibaba.cobar.route.function.PartitionByLong">
  <!--满足128 * 8 = 1024-->
  <property name="partitionCount">8</property>
  <property name="partitionLength">128</property>
</function>
```

不均匀分布

如果需要不均匀分库，则将[0-1023]分成不均匀的区段即可

$\text{hash}(\text{pavarott}) = 3170972965401 \% 1024 = 537$



```
<function name="func"
class="com.alibaba.cobar.route.function.PartitionByLong">
  <!--这两行配置一一对应，表示512长度的区段1个，256长度的区段1个，128长度的区段2个-->
  <!--满足1*512 + 1*256 + 2*128 = 1024-->
  <property name="partitionCount">1,1,2</property>
  <property name="partitionLength">512,256,128</property>
</function>
```

多维路由

自定义路由算法

用户可以自定义路由算法，具体可以参照`com.alibaba.cobar.route.function.PartitionByLong`或者`com.alibaba.cobar.route.function.PartitionByString`编写代码