

HBase 多租户隔离技术：RegionServer Group 介绍及实战

背景

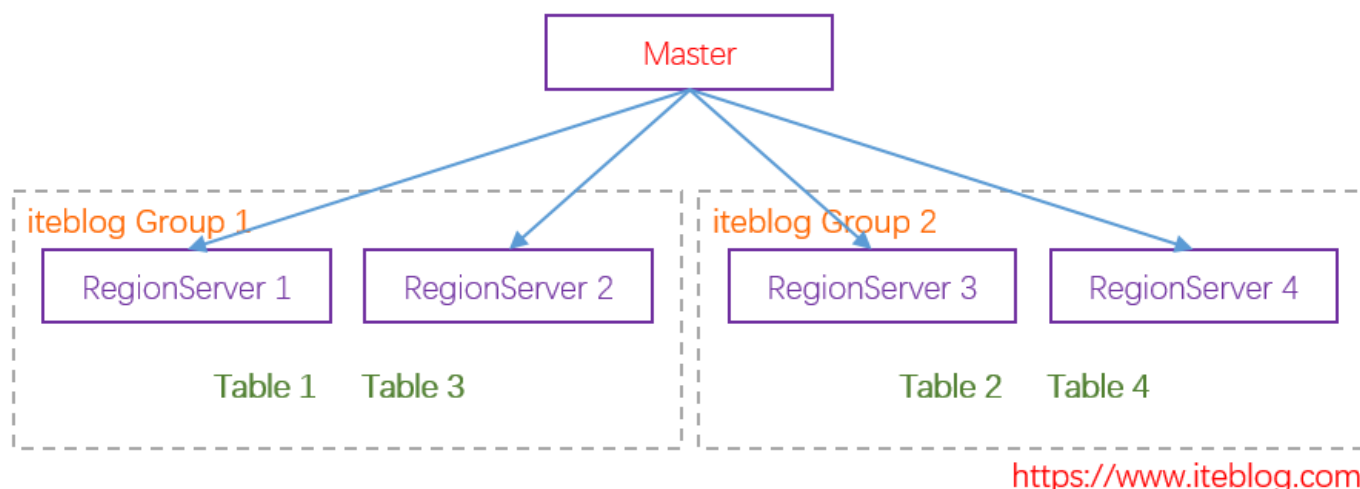
随着 Apache HBase 在各个领域的广泛应用，在 HBase 运维或应用的过程中我们可能会遇到这样的问题：

- 同一个 HBase 集群使用的用户越来越多，不同用户之间的读写或者不同表的 compaction、region splits 操作可能对其他用户或表产生了影响。将所有业务的表都存放在一个集群的好处是可以很好的利用整个集群的资源，只需要一套运维系统。
- 如果一个业务或者一个部门使用一个 HBase 集群，这样会导致 HBase 集群的数量越来越多，直接导致了运维成本的增加。而且集群的分离也会导致资源的浪费，有些集群资源过剩，有些集群资源不足，这种情况我们无法充分利用不同集群的资源。将集群按照业务或部门分开的好处是可以很好的隔离不同表、不同用户之间的影响。

上面两种情况均存在不足，如果我们能够多租户共用一套集群，而且能够使不同用户之间进行隔离，在上层用户看来好像是独享一套 HBase 集群。为了解决这个问题，来自雅虎的 Francis Liu 提出了多租户隔离技术：RegionServer Group，详情请参见 [HBASE-6721](#)。这个技术的目标就是只需维护一个 HBase 集群，同时满足各个表在性能上、操作上的互不干扰，又支持访问安全隔离。

RegionServer Group

RegionServer Group 技术是通过将 RegionServer 进行分组，不同的 RegionServer 分到不同的组。每个组可以按需挂载不同的表，并且当组内的表发生异常后，Region 不会迁移到其他的组。这样，每个组就相当于一个逻辑上的子集群，通过这种方式达到资源隔离的效果，降低管理成本，不必为每个高 SLA 的业务线单独搭集群。



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

从上图可以看出，RegionServer 1 和 RegionServer 2 同属于 iteblog Group 1，而且管理 Table 1 和 Table 3 两张表；RegionServer 3 和 RegionServer 4 同属于 iteblog Group 2，而且管理 Table 2 和 Table 4 两张表。从用户角度上看，RegionServer 1 和 RegionServer 2 看起来是属于一个集群；而 RegionServer 3 和 RegionServer 4 同属于一个集群，这两个组之间均不互相影响。但是对于集群运维人员来说，这就是一个 HBase 集群，我们只需要运维这一个 HBase 集群即可，大大降低了运维成本。

技术简要介绍

在引入 RegionServer Group 技术之前，HBase 默认使用 StochasticLoadBalancer 策略（通过 hbase.master.loadbalancer.class 参数实现，参见[HBASE-9555](#)）将表的 Region 移动到 RegionServer 里面去。这种策略被没有考虑到 RegionServer Group 的信息，所以如果要对 RegionServer 分组，我们就需要在移动 Region 的时候考虑到这些信息。

基于这些信息，社区开发出能够识别出 RegionServer Group 信息的 RSGroupBasedLoadBalancer，这个类和上面的 StochasticLoadBalancer 都是实现了 LoadBalancer 接口的，详见 [HBASE-6721](#)。

在默认情况下，所有的表和 RegionServer 都属于 default 组。RSGroupBasedLoadBalancer 类具有识别组信息的能力，所以在移动表的 Region 到 RegionServer 的时候会考虑到 RegionServer Group 信息的。为了启用这个特性，我们需要在 master 节点的 hbase-site.xml 文件加入以下的配置：

```
<property>
  <name>hbase.coprocessor.master.classes</name>
  <value>org.apache.hadoop.hbase.rsgroup.RSGroupAdminEndpoint</value>
</property>

<property>
  <name>hbase.master.loadbalancer.class</name>
  <value>org.apache.hadoop.hbase.rsgroup.RSGroupBasedLoadBalancer</value>
</property>
```

设置完之后需要重启 master 节点。

RegionServer Group 技术新引入的命令

RegionServer Group 技术为我们带来了以下13个新的 HBase shell 命令。使用下面的命令一定需要在 master 节点上的 hbase-site.xml 文件配置好上面两个属性。否则将会出现 UnknownProtocolException: No registered Master Coprocessor Endpoint found for RSGroupAdminService 异常。如下所示：

```
hbase(main):002:0> list_rsgroups
NAME                                SERVER / TABLE
```

```
ERROR: org.apache.hadoop.hbase.exceptions.UnknownProtocolException: No registered Master Coprocessor Endpoint found for RSGroupAdminService. Has it been enabled?
at org.apache.hadoop.hbase.master.MasterRpcServices.execMasterService(MasterRpcServices.java:802)
at org.apache.hadoop.hbase.shaded.protobuf.generated.MasterProtos$MasterService$2.callBlockingMethod(MasterProtos.java)
at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:413)
at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:130)
at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:324)
at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:304)
```

List all RegionServer groups. Optional regular expression parameter can be used to filter the output.

Example:

```
hbase> list_rsgroups
hbase> list_rsgroups 'abc.*'
```

Took 1.1077 seconds

- `list_rsgroups` : 列出所有的 RegionServer groups , 我们可以在这个命令的后面使用正则表达式来过滤一些我们要的信息。

```
hbase> list_rsgroups
hbase> list_rsgroups 'iteblog.*'
```

- `get_rsgroup` : 获取某个 RegionServer group 的信息。

```
hbase> get_rsgroup 'default'
```

- `add_rsgroup` : 创建一个新的 RegionServer Group。

```
hbase> add_rsgroup 'iteblog_group'
```

- `remove_rsgroup` : 删除某个 RegionServer Group。

```
hbase> remove_rsgroup 'iteblog_group'
```

- `balance_rsgroup`：对某个 RegionServer Group 进行 balance 操作。

```
hbase> balance_rsgroup 'iteblog_group'
```

- `move_servers_rsgroup`：将 RegionServers 从一个组移动到另一个组。RegionServer 中的所有 Region 都将移动到另一个 RegionServer 中。

```
hbase> move_servers_rsgroup 'dest',['server1:port','server2:port']
```

- `move_tables_rsgroup`：将表从一个 RegionServer Group 移动另一个 RegionServer Group。

```
hbase> move_tables_rsgroup 'dest',['table1','table2']
```

- `move_namespaces_rsgroup`：将指定命名空间的表从一个 RegionServer Group 移动另一个 RegionServer Group。

```
hbase> move_namespaces_rsgroup 'dest',['ns1','ns2']
```

- `move_servers_tables_rsgroup`：将 RegionServers 和 Tables 从一个 RegionServer Group 移动另一个 RegionServer Group。

```
hbase> move_servers_tables_rsgroup 'dest',['server1:port','server2:port'],['table1','table2']
```

- `move_servers_namespaces_rsgroup`：将指定命名空间的 RegionServers 和 Tables 从一个 RegionServer Group 移动另一个 RegionServer Group。

```
hbase> move_servers_namespaces_rsgroup 'dest',['server1:port','server2:port'],['ns1','ns2']
```

- `get_server_rsgroup`：获取给定 RegionServer 所属的 RegionServer Group。

```
hbase> get_server_rsgroup 'server1:port1'
```

- `get_table_rsgroup`：获取给定表所属的 RegionServer Group。

```
hbase> get_table_rsgroup 'iteblog_Table'
```

- `remove_servers_rsgroup` : 从 RegionServer Group 中删除已停用的 Region。处于 Dead/recovering/live 状态的 Region 将无法操作。

```
hbase> remove_servers_rsgroup ['server1:port','server2:port']
```

如何使用 RegionServer Group

我们前面说了，如果没有创建 RegionServer Group 的话，HBase 默认的组只有 default，而且所有的表和 RegionServer 都属于 default 组，如下：

```
hbase(main):001:0> list_rsgroups
NAME                                SERVER / TABLE
default                             server 192.168.1.103:16020
                                     server 192.168.1.103:16021
                                     table hbase:meta
                                     table hbase:namespace
                                     table hbase:rsgroup

2 row(s)
Took 1.4986 seconds
```

现在我们使用 `add_rsgroup` 命令创建了一个名为 `iteblog_group` 的组：

```
hbase(main):020:0> add_rsgroup 'iteblog_group'
Took 0.0444 seconds
hbase(main):021:0> list_rsgroups
NAME                                SERVER / TABLE
iteblog_group                       server 192.168.1.103:16020
default                             server 192.168.1.103:16021
                                     table hbase:meta
                                     table hbase:namespace
                                     table hbase:rsgroup

2 row(s)
Took 0.0101 seconds
```

下面命令我们将 192.168.1.103:16021 RegionServer 移到了名为 iteblog_group 组里面：

```
hbase(main):022:0> move_servers_rsgroup 'iteblog_group', ['192.168.1.103:16021']
Took 1.0220 seconds
hbase(main):023:0> list_rsgroups
NAME                                SERVER / TABLE
iteblog_group                       server 192.168.1.103:16021
default                             server 192.168.1.103:16020
                                     table hbase:meta
                                     table hbase:namespace
                                     table hbase:rsgroup

2 row(s)
Took 0.0102 seconds
```

我们已经看到了地址为 192.168.1.103:16021 的 RegionServer 已经移到名为 iteblog_group 的组了。现在我们来创建一张表 iteblog_table，并且把这张表移到名为 iteblog_group 的组里面。为了演示 RegionServer Group 的作用，这里我现在创建一张名为 iteblog_table 的表，并且设置了预分区，如下：

```
hbase(main):011:0> create 'iteblog_table', 'cf', SPLITS=>['10','20','30','40']
Created table iteblog_table
Took 1.3884 seconds
=> Hbase::Table - iteblog_table
hbase(main):013:0> get_table_rsgroup 'iteblog_table'
default
1 row(s)
Took 0.0281 seconds
```

从上面的输出可以看到，创建表的时候，默认是属于 default 组，虽然我们有两个 RegionServer，而且设置了预分区，按正常情况，iteblog_table 的 Region 应该是会分布在这两个 RegionServer 上的。但实际上因为这两个 RegionServer 的组不一样，而且 iteblog_table 的组属于 default，所有 iteblog_table 的 Region 全部位于 192.168.1.103:16020 上。正如下图所示：

← → ↻ ① 不安全 | 192.168.1.103:16010/table.jsp?name=iteblog_table&showwhole=true ☆ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳ ㉑ ㉒ ㉓ ㉔ ㉕ ㉖ ㉗ ㉘ ㉙ ㉚ ㉛ ㉜ ㉝ ㉞ ㉟ ㊀ ㊁ ㊂ ㊃ ㊄ ㊅ ㊆ ㊇ ㊈ ㊉ ㊊ ㊋ ㊌ ㊍ ㊎ ㊏ ㊐ ㊑ ㊒ ㊓ ㊔ ㊕ ㊖ ㊗ ㊘ ㊙ ㊚ ㊛ ㊜ ㊝ ㊞ ㊟ ㊠ ㊡ ㊢ ㊣ ㊤ ㊥ ㊦ ㊧ ㊨ ㊩ ㊪ ㊫ ㊬ ㊭ ㊮ ㊯ ㊰ ㊱ ㊲ ㊳ ㊴ ㊵ ㊶ ㊷ ㊸ ㊹ ㊺ ㊻ ㊼ ㊽ ㊾ ㊿

APACHE HBASE Home Table Details Procedures Process Metrics Local Logs Log Level Debug Dump Metrics Dump HBase Configuration

Sort As **RegionName** Ascending ShowDetailName&Start/End Key Reorder

Name(5)	Region Server	ReadRequests (0)	WriteRequests (0)	StorefileSize (0 B)	Num.Storefiles (0)	MemSize (0 B)	Locality	Start Key	End Key
iteblog_table,,1541134693533.647ec21c0525a4e11d7c2dbd97c2bb48.	192.168.1.103:16030	0	0	0 B	0	0 B	0.0		10
iteblog_table,10,1541134693533.5f02958d38da1f3cb47daf65ad136d28.	192.168.1.103:16030	0	0	0 B	0	0 B	0.0	10	20
iteblog_table,20,1541134693533.e887290f8f0fa1e0983ed24d1b5871d4.	192.168.1.103:16030	0	0	0 B	0	0 B	0.0	20	30
iteblog_table,30,1541134693533.0b8201d354a266d45cd505ef0d53b2d7.	192.168.1.103:16030	0	0	0 B	0	0 B	0.0	30	40
iteblog_table,40,1541134693533.5edc01776353c59979ef8048a142cafc.	192.168.1.103:16030	0	0	0 B	0	0 B	0.0	40	

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

现在我们将表 iteblog_table 移到名为 iteblog_group 的组里面，那属于 iteblog_table 的 Region 也应该全部移到 iteblog_group 里面的：

```
hbase(main):007:0> move_tables_rsgroup 'iteblog_group', ['iteblog_table']
Took 5.6177 seconds
```

```
hbase(main):008:0> get_table_rsgroup 'iteblog_table'
iteblog_group
1 row(s)
Took 0.0202 seconds
```

我们已经看到 iteblog_table 所属的组为 iteblog_group，从下图也可以看到，iteblog_table 的所有 Region 全部移到了 192.168.1.103:16021 上。

← → ↻ ① 不安全 | 192.168.1.103:16010/table.jsp?name=iteblog_table&showwhole=true ☆ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳ ㉑ ㉒ ㉓ ㉔ ㉕ ㉖ ㉗ ㉘ ㉙ ㉚ ㉛ ㉜ ㉝ ㉞ ㉟ ㊀ ㊁ ㊂ ㊃ ㊄ ㊅ ㊆ ㊇ ㊈ ㊉ ㊊ ㊋ ㊌ ㊍ ㊎ ㊏ ㊐ ㊑ ㊒ ㊓ ㊔ ㊕ ㊖ ㊗ ㊘ ㊙ ㊚ ㊛ ㊜ ㊝ ㊞ ㊟ ㊠ ㊡ ㊢ ㊣ ㊤ ㊥ ㊦ ㊧ ㊨ ㊩ ㊪ ㊫ ㊬ ㊭ ㊮ ㊯ ㊰ ㊱ ㊲ ㊳ ㊴ ㊵ ㊶ ㊷ ㊸ ㊹ ㊺ ㊻ ㊼ ㊽ ㊾ ㊿

APACHE HBASE Home Table Details Procedures Process Metrics Local Logs Log Level Debug Dump Metrics Dump HBase Configuration

Table Regions

Sort As **RegionName** Ascending ShowDetailName&Start/End Key Reorder

Name(5)	Region Server	ReadRequests (0)	WriteRequests (0)	StorefileSize (0 B)	Num.Storefiles (0)	MemSize (0 B)	Locality	Start Key	End Key
iteblog_table,,1541134693533.647ec21c0525a4e11d7c2dbd97c2bb48.	192.168.1.103:16031	0	0	0 B	0	0 B	0.0		10
iteblog_table,10,1541134693533.5f02958d38da1f3cb47daf65ad136d28.	192.168.1.103:16031	0	0	0 B	0	0 B	0.0	10	20
iteblog_table,20,1541134693533.e887290f8f0fa1e0983ed24d1b5871d4.	192.168.1.103:16031	0	0	0 B	0	0 B	0.0	20	30
iteblog_table,30,1541134693533.0b8201d354a266d45cd505ef0d53b2d7.	192.168.1.103:16031	0	0	0 B	0	0 B	0.0	30	40
iteblog_table,40,1541134693533.5edc01776353c59979ef8048a142cafc.	192.168.1.103:16031	0	0	0 B	0	0 B	0.0	40	

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

我们还可以为命名空间设置 RegionServer 组，这样属于这个命名空间的表都属于这个 RegionServer 组：

```
hbase(main):012:0> create_namespace 'iteblog_namespace', {'hbase.rsgroup.name'=>'iteblog_group'}
Took 0.3406 seconds
hbase(main):013:0> create 'iteblog_namespace:mytable', 'f'
Created table iteblog_namespace:mytable
Took 0.8126 seconds
=> Hbase::Table - iteblog_namespace:mytable
hbase(main):014:0> get_table_rsgroup 'iteblog_namespace:mytable'
iteblog_group
1 row(s)
Took 0.0080 seconds
```

好了，到这里我们已经了解了 HBase RegionServer Group 技术，已经如何使用了。剩下的命令我就不再介绍，感兴趣的同学可以自己再去学习。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: **【】**（ ）