

云数据库 HBase

HBase SQL服务(Phoenix)

HBase SQL服务(Phoenix)

HBase SQL(Phoenix)与Spark的选择

Phoenix 与 Spark的区别

ApsaraDB Phoenix是ApsaraDB HBase提供的SQL层，主要为了解决高并发、低延迟、简单查询场景，当然也可以解决一定的分析需求。必须命中索引且命中后返回的数据较少，如果是join，则join任意一则返回的数据量在10w以下，且另一侧必须命中索引。为了保障集群稳定性，一些复杂的sql及耗时的sql会被平台拒绝运行。

ApsaraDB Spark是ApsaraDB HBase提供的分析引擎，满足低并发，高延迟，复杂计算场景。不管怎么复杂的SQL，都可以完成。另外Spark可以支持sql、scala、java、python语言，支持流、OLAP、离线分析、数据清洗、支持多源(HBase、MongoDB、Redis、OSS等)。(Spark Streaming支持准实时的在线流，不在此讨论访问内)

对比项目	Phoenix	Spark
SQL复杂度	简单查询，必须命中索引且命中后返回的数据较少，如果是join，则join任意一则返回的数据量在10w以下，且另一侧必须命中索引。为了保障集群稳定性，一些复杂的sql及耗时的sql会被平台拒绝运行。	全部支持执行完成，支持Spark映射到Phoenix，做到Spark在简单SQL查询能到Phoenix同样的性能，不过Spark定位为分析的场景，与Phoenix纯TP有本质的区别
集群	HBase共享一个集群，本质是HBase提供的SQL	Spark需要单独购买集群
并发	单机 1w-5w左右	Spark最高不超过100
延迟	延迟在ms级别，一些命中较多的数据的sql会到秒	一般延迟在300ms以上，大部分sql需要秒，分钟，甚至小时
更新	Phoenix支持	Spark不支持
支持业务	在线业务	离线业务 或者 准在线业务

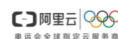
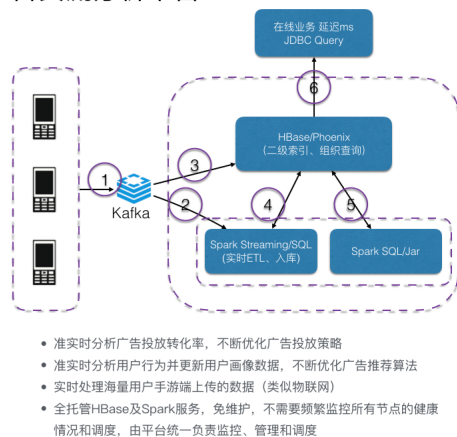
所以：

- 简单查询、高并发、低延迟、在线业务 选择 Phoenix
- 复杂计算、低并发、高延迟、离线业务、准在线业务 选择 Spark

案例

可以看出：spark主要做流ETL及数据的二次加工，在线的查询还是通过Phoenix完成的

广告买流分析平台



一、数据源

(1)：捕获用户行为事件或者手机端上传的数据并写入Kafka集群；

二、实时入库及ETL

(3)：从Kafka读取手机端上传的数据然后直接写入HBase/Phoenix；
(2)、(4)：Spark streaming实时读取Kafka数据，在做ETL的同时读取HBase/Phoenix的维表数据；并将结果数据写入HBase/Phoenix对外提供查询；

三、复杂分析

(5) Spark SQL对HBase/Phoenix中的数据做复杂分析，并把结果写入HBase/Phoenix；

四、在线查询

(6) BI或个性化推荐系统通过标准JDBC接口，实时查询HBase/Phoenix中的用户画像数据

HBase SQL(Phoenix) 4.x 使用说明

Phoenix是什么

Phoenix查询引擎支持使用SQL进行HBase数据的查询，会将SQL查询转换为一个或多个HBase API，协同处理器与自定义过滤器的实现，并编排执行。使用Phoenix进行简单查询，其性能量级是毫秒。

更多的信息可以参考官网：<http://phoenix.apache.org/>

Ali-Phoenix 说明

1. 兼容开源客户端(开源4.12)
2. 支持公网访问
3. 修复多个开源BUG
4. 新增功能和性能优化

参考：版本说明

Ali-Phoenix 客户端下载地址

Phoenix-4.12.0-AliHBase-1.1-0.9 下载 (JDK7编译)

使用说明

准备工作

准备一个内网的ECS，需要和HBase处在同一个网络内。

例如HBase是在经典网络的，那么就准备一个经典网络的ECS，如果HBase是在VPC的，那么就在需要在同一个VPC内的ECS

按照下载地址下载 Phoenix 客户端

在这台ECS上下载HBase对应版本的Phoenix客户端，这里以4.12.0-AliHBase-1.1-0.9版本举例

```
wget https://hbase-opt.oss-cn-hangzhou.aliyuncs.com/ali-phoenix-4.12.0-AliHBase-1.1-0.9.tar.gz
```

3. 解压缩压缩包

```
tar zxvf ali-phoenix-4.12.0-AliHBase-1.1-0.9.tar.gz
```

在HBase的网络白名单中开启访问节点的IP白名单

查看这台ECS的内网IP

```
hostname -i
```

然后把他加到HBase的网络白名单中。加入的方法请参考HBase白名单控制。

启动sqlline

在HBase产品的集群详情页面查看ZooKeeper的连接地址，然后使用如下的方式启动。启动命令sqlline.py在bin目录下

```
./sqlline.py hb-bp19142ir9xxxxx-001.hbase.rds.aliyuncs.com,hb-bp19142ir9ruxxxxx-002.hbase.rds.aliyuncs.com,hb-bp19142ir9ruxxxxx-004.hbase.rds.aliyuncs.com
```

界面如果显示出类似jdbc:xxxx>这样的一个提示的时候，就表示启动成功了。

验证

我们在这个命令行中输入

```
!tables
```

如果看到一个表的列表，那么就说明我们配置成功了。可以开始更加深入的使用了。

退出

使用quit命令退出Phoenix

```
!quit
```

更多资料

Phoenix的入门教学Phoenix的深入使用

HBase SQL(Phoenix) 5.x 使用说明

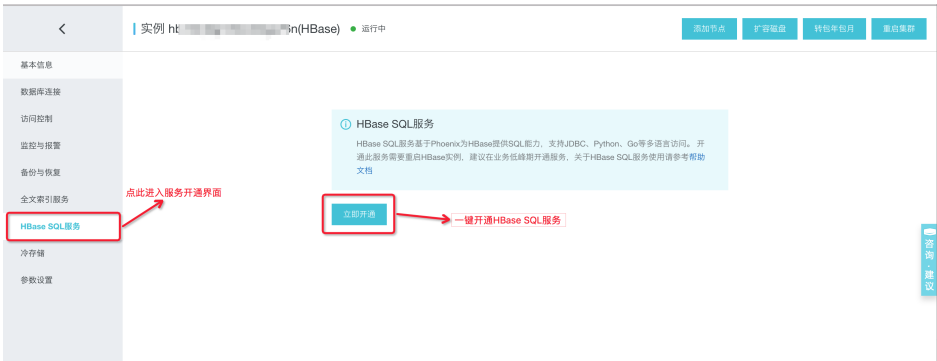
Phoenix 5.x 说明

HBase SQL服务基于Phoenix 5.x为HBase 2.x提供SQL能力，通过轻客户端即可快速连接访问。SQL服务挂载了SLB负载均衡，通过round robin模式将请求均匀分发给每个query server节点。此外轻客户端还支持Python、Go等多语言访问。

使用步骤

开通HBase SQL服务

HBase实例开通后，进入控制台管理界面，在左侧菜单栏可以点击HBase SQL服务项->立即开通，即可开通HBase SQL服务。



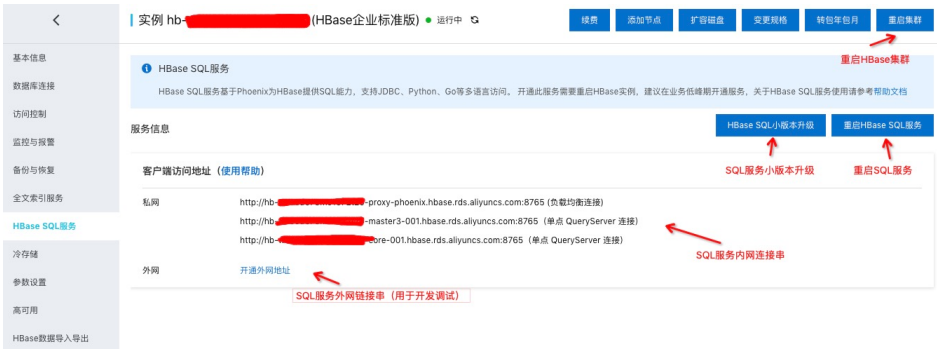
开通HBase

SQL服务时需要重启HBase实例，并占用部分内存资源，请在业务低峰期进行开通，开通过程需要十分钟左右，节点数越多时间会相对越长。

开通之后，后续对SQL服务重启和小版本升级也在该界面，点击相应按钮即可。

SQL服务管理

服务开通后，在HBase SQL服务栏即可看到如下界面：



用户可以在上图所示控制台中对SQL服务进行日常管理，包括SQL服务小版本升级，SQL服务重启，开通公网访问。

需要说明的是，SQL服务连接地址分为负载均衡连接和单点连接，分别在不同的场景中使用：

- 负载均衡连接：使用SLB进行负载均衡，对于高并发读写请求能够均发至后端的多个QueryServer中处理，提升集群的整体吞吐能力。
- 单点连接：使用大表创建索引或使用UPSERT...SELECT导出数据以及复杂查询等单点请求场景，由于请求时间过长使用负载均衡连接会出现连接超时断掉情况，需使用单点QueryServer连接。

客户端准备

准备一台ECS，需要和HBase处在同一个网络内，建议是同一个vpc内部的ECS实例，否则需要打通网络，或者使用外网地址连接。

查看这台ECS的内网IP

```
hostname -i
```

然后把他加到HBase的网络白名单中。加入的方法请参考HBase白名单控制。

客户端访问HBase SQL服务

- 在准备好的ECS客户机上下载最新版本的HBase SOL客户端：

```
wget http://hbase-opt.oss-cn-hangzhou.aliyuncs.com/ali-phoenix-5.2.0-HBase-2.x.tar.gz
```

- 解压客户端压缩包

```
tar zxvf ali-phoenix-5.2.0-HBase-2.x.tar.gz
```

- 启动轻客户端工具

```
bin/sqlline-thin.py http://xxx-proxy-phoenix.hbase.rds.aliyuncs.com:8765
```

- 验证

我们在这个命令行中输入：

```
!tables
```

如果看到一个表的列表，那么就说明我们配置成功了。可以开始更加深入的使用了。

- 退出

使用quit命令退出Phoenix

```
!quit
```

注意：首次连接会创建meta表，需等待一段时间。

Phoenix 5.x vs Phoenix-4.x

1. Phoenix-5.x对于时区的处理逻辑做了统一优化，请参考云栖文章：[Phoenix关于时区的处理方式说明](#)。
2. Phoenix-5.x使用轻客户端模式，而Phoenix-4.x默认提供重客户端模式。轻客户端和重客户端模式请参考云栖文章：[Phoenix客户端优化之由重到轻](#)。

HBase SQL(Phoenix) 入门

Phoenix 入门

本篇介绍简单的Phoenix使用方法，进行初步的数据查询。

在开始之前，请先确认您已经准备好了Phoenix的运行环境，若还没有，请参考[这里](#)

快速开始

1. 创建一个us_population表

```
CREATE TABLE IF NOT EXISTS us_population (  
  state CHAR(2) NOT NULL,  
  city VARCHAR NOT NULL,  
  population BIGINT  
  CONSTRAINT my_pk PRIMARY KEY (state, city));
```

写入数据

```
UPSERT INTO us_population VALUES('NY','New York',8143197);  
UPSERT INTO us_population VALUES('CA','Los Angeles',3844829);  
UPSERT INTO us_population VALUES('IL','Chicago',2842518);  
UPSERT INTO us_population VALUES('TX','Houston',2016582);  
UPSERT INTO us_population VALUES('PA','Philadelphia',1463281);  
UPSERT INTO us_population VALUES('AZ','Phoenix',1461575);  
UPSERT INTO us_population VALUES('TX','San Antonio',1256509);  
UPSERT INTO us_population VALUES('CA','San Diego',1255540);  
UPSERT INTO us_population VALUES('TX','Dallas',1213825);  
UPSERT INTO us_population VALUES('CA','San Jose',912332);
```

查询SQL

```
SELECT state as "State",count(city) as "City Count",sum(population) as "Population Sum"  
FROM us_population  
GROUP BY state  
ORDER BY sum(population) DESC;
```

结果验证

```
csv columns from database.
CSV Upsert complete. 10 rows upserted
Time: 0.159 sec(s)

St              City Count              Population Sum
-----
NY              1              8143197
CA              3              6012701
TX              3              4486916
IL              1              2842518
PA              1              1463281
AZ              1              1461575
Time: 0.16 sec(s)
```

API访问Phoenix JDBC

Phoenix 5.x SDK maven依赖

```
<dependency>
<groupId>com.aliyun.phoenix</groupId>
<artifactId>ali-phoenix-queryserver-client</artifactId>
<version>5.2.1-HBase-2.x</version>
</dependency>
```

Phoenix 4.x SDK maven依赖

```
<dependency>
<groupId>com.aliyun.phoenix</groupId>
<artifactId>ali-phoenix-core</artifactId>
<version>${参考FAQ中最新版本说明}</version>
</dependency>
```

代码示例请参考：云HBase Demo

使用前必读

Phoenix定位为OLTP和操作型分析（operational analytics），大多用于在线业务，稳定性要求第一位。Phoenix的功能很强大，也很灵活，驾驭好对用户来说还是有一定难度，本文结合大部分用户的经验，给出一些使用建议，希望能帮助更多用户少走弯路。

一. 用户手册

Phoenix SQL基于SQL-92标准，但是还是有很多方言，软件架构也跟传统单机数据库有较大区别，用户在使用前务必阅读相关文档资料。以下罗列系列资料供用户参考：

1、Phoenix社区官方文档

<http://phoenix.apache.org/>，其中经常会翻到的链接如下：

Phoenix SQL语法

Phoenix数据类型

Phoenix内置函数

SQLLine 手册

2、阿里云HBase团队云栖中文系列文章和文档

Phoenix用户手册

HBase进化之从NoSQL到NewSQL，凤凰涅槃成就Phoenix

阿里云HBase SQL (Phoenix) 服务深度解读

二. 使用建议

1、二级索引使用指南

二级索引是Phoenix中非常重要的功能，用户在使用前需要深入了解，可以参考资料：二级索引社区文档，二级索引中文文档，全局索引设计实践。

下面介绍一些在使用二级索引过程中的注意事项：

1) 是否需要使用覆盖索引？

覆盖索引需要将查询返回字段加入到索引表中，这样在命中索引时，只需要查询一次索引表即可，非覆盖索引，要想拿到完整结果则需要回查主表。不难理解，覆盖索引查询性能更好，但是会浪费一定存储空间，影响一定写性能。非覆盖索引使用时，有时执行计划并不能默认命中索引，此时，用户需要加索引Hint。

2) 应该使用local Index还是global Index？实现上,一个global index表对应着一个hbase 表，local index是在主表上新增一列存储索引数据。

适用场景上，global index 适用于多读的场景，但存在同步索引时带来网络开销较大的问题。而local由于和原数据存储在一张表中同步索引数据会相对快一点。虽然local index也有一定适用场景，但仍然推荐使用global index, 其原因有以下几点：

1. 当前版本的phoneix的local index的实现相对global index不太完善，问题较多，使用存在一定的风险。
2. local index不太完善，大的改动后，可能会存在不兼容，升级流程比较复杂。
3. 在大数据量下，原始数据和索引数据放在一起会加剧region分裂，且分裂后索引数据的本地性也会丧失。

因此，在阿里云HBase SQL服务中**LOCAL INDEX功能已经被禁止**！

3) 索引表最多可以创建多少个？

索引会保证实时同步，也会引来写放大问题，一般建议不超过10个，如果超过建议使用HBase全文索引功能。

4) 构建索引需要注意哪些事项？

使用创建索引语句（CREATE INDEX）时，如果指定async参数，则为异步构建，语句完成时，会在SYSTEM.CATALOG表中简历索引表的元信息，并建立跟主表的关系，但是状态是building，索引表中没有数据，也不可查，需要后续用REBUILD语句来构建，或者借助MR工具来构建索引，并将状态更新为active。如果不指定，则为同步构建，一般建议数据量小于1亿行都使用同步构建即可，比较简单。

2、加盐注意事项

注意：本特性有副作用，不建议使用。使用前请确保您已经充分了解其原理与适用场景。如果您无法确定您的场景是否适合，请钉钉与我们联系，我们会帮您进行评估。

加盐通常用来解决数据热点和范围查询同时存在的场景。原理介绍可以参考：Phoenix社区文档和中文文档。

加盐有较强的适用场景要求，场景不合适将适得其反：

- 写热点或写不均衡：比如以时间作为第一列主键，永远写表头或者表尾
- 需要范围查询：要按第一列主键进行范围查询，故不能使用hash打散

有热点就要打散，但打散就难以做范围查询。因此，要同时满足这对**相互矛盾的需求**，必须有一种**折中**的方案：既能在一定程度上打散数据，又能保证有序。这个解决方案就是加盐，亦称分桶(salt buckets)。数据在桶内**保序，桶之间随机**。写入时按桶个数取模，数据随机落在某个桶里，保证写请求在桶之间是均衡的。查询时读取所有的桶来保证结果集的有序和完备。

一般来说，严格满足上述条件的业务场景并不常见。大多数场景都可以找到其他的业务字段来协助散列。考虑到其严重的副作用，我们不建议使用这个特性。

副作用：

- **写瓶颈**：一般全表只有buckets个region用于承担写。当业务体量不断增长时，因为无法调整bucket数量，不能有更多的region帮助分担写，会导致写入吞吐无法随集群扩容而线性增加。导致写瓶颈，从而限制业务发展。
- **读扩散**：select会按buckets数量进行拆分和并发，每个并发都会在执行时占用一个线程。select本身一旦并发过多会导致线程池迅速耗尽或导致QueryServer因过高的并发而FGC。同时，本应一个RPC完成的简单查询，现在也会拆分成多个，使得查询RT大大增加。

这两个副作用会制约业务的发展，尤其对于大体量的、发展快速的业务。因为桶个数不能修改，写瓶颈会影响业务的扩张。读扩散带来的RT增加也大大降低了资源使用效率。

常见的使用误区：

- **预分区**：用分桶来实现建表的预分区最常见的误用。这是因为Phoenix提供的split on预分区语法很难使用。目前可用hbase shell的建表，指定预分区，之后关联为Phoenix表。在海量数据场景，合理的预分区是一个很有挑战的事情，我们后续会对此进行专题讨论，敬请期待。
- **伪热点**：写入热点或不均衡大多数情况都是假象，通常还有其他字段可用于打散数据。比如监控数据场景，以metric名字的hash值做首列主键可有效解决写入均衡的问题。

一定不要为了预分区而使用加盐特性，要结合业务的读写模式来进行表设计。如果您无法做出准确的判断，可以钉钉联系云HBase值班咨询。

Buckets个数跟机型配置和数据量有关系，可以参考下列方式计算，其中 N 为 Core/RS 节点数量：

- 单节点内存 8G: $2*N$
- 单节点内存 16G: $3*N$
- 单节点内存 32G: $4*N$
- 单节点内存 64G: $5*N$
- 单节点内存 128G: $6*N$

注意：索引表默认会继承主表的盐值；bucket的数目不能超过256；一个空的Region在内存中的数据结构大概2MB，用户可以评估下单个RegionServer承载的总Region数目，有用户发生过在低配置节点上，建大量加盐表直接把集群内存耗光的问题。

3、慎用扫全表、OR、Join和子查询

虽然Phoenix支持各种Join操作，但是Phoenix主要还是定位为在线数据库，复杂Join，比如子查询返回数据量特别大或者大表Join大表，在实际计算过程中十分消耗系统资源，会严重影响在线业务，甚至导致OutOfMemory异常。对在线稳定性和实时性要求高的用户，建议只使用Phoenix的简单查询，且查询都命中主表或者索引表的主键。另外，建议用户在运行SQL前都执行下explain，确认是否命中索引，或者主键，参考explain社区文档和explain中文文档。

4、Phoenix不支持复杂查询

Phoenix的二级索引本质还是前缀匹配，用户可以建多个二级索引来增加对数据的查询模式，二级索引的一致性是通过协处理器实现的，索引数据可以实时可见，但也会影响写性能，特别是建多个索引的情况下。对于复杂查询，比如任意条件的and/or组合，模糊查找，分词检索等Phoenix不支持，建议使用SearchIndex（全文索引）。需要说明的是，相比于二级索引，SearchIndex采用后台异步索引，对在线业务影响较小，但会引入一定延迟，默认10s。

5、Phoenix不支持复杂分析

Phoenix定位为操作型分析（operational analytics），对于复杂分析，比如前面提到的复杂join则不适合，这种建议用Spark这种专门的大数据计算引擎来实现，参考X-Pack Spark分析服务，HBase SQL(Phoenix)与

Spark的选择。

6、Phoenix是否支持映射已经存在的HBase表？

支持，参考社区相关文档。用户可以通过Phoenix创建视图或者表映射已经存在的HBase表，注意如果使用表的方式映射HBase表，在Phoenix中执行DROP TABLE语句同样也会删除HBase表。另外，由于column family和列名是大小写敏感的，必须一一对应才能映射成功。另外，Phoenix的字段编码方式大部分跟HBase的Bytes工具类不一样，一般建议如果只有varchar类型，才进行映射，包含其他类型字段时不要使用映射。

备注：其他使用中常见问题请多翻看FAQ文档

HBase SQL(Phoenix) FAQ

1. ali-phoenix 最新版本在maven中央仓库发布了哪些jar包？

- ali-phoenix 4.x发布包如下：

GroupId	ArtifactId	Latest Version
com.aliyun.phoenix	ali-phoenix	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-hive	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-spark	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-pherf	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-queryserver	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-queryserver-client	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-pig	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-flume	4.12.0-AliHBase-1.1-0.9
com.aliyun.phoenix	ali-phoenix-core	4.12.0-AliHBase-1.1-0.9

- ali-phoenix 5.x发布包如下：

GroupId	ArtifactId	Latest Version
com.aliyun.phoenix	ali-phoenix-queryserver-client	5.1.0-HBase-2.0
com.aliyun.phoenix	ali-phoenix-shaded-thin-client	5.1.0-HBase-2.0

避免轻客户端中包与业务三方包冲突，推荐使用轻客户端shade包。

2. ali-phoenix JDBC URL格式是什么样的？

在启用query-server的时候需要使用轻客户端，否则使用重客户端。*ali-phoenix*不支持以*keytab*的方式访问云HBASE，所以此处和*apache phoenix*的JDBC格式有所差异

重客户端DRIVE

URL语法

```
jdbc:phoenix:[ZK_HOST1:port, ZK_HOST2:port, ZK_HOST3:port | comma-separated ZooKeeper Quorum  
[:port] [:hbase root znode] ]
```

简单URL示例

```
jdbc:phoenix:localhost  
jdbc:phoenix:localhost:123:/hbase  
jdbc:phoenix:v1,v2,v3:123:/hbase  
jdbc:phoenix:v1:2181,v2:2181,v3:2181:/hbase  
jdbc:phoenix:v1:2181,v2:2181,v3:2181
```

轻客户端DRIVE

URL语法

```
jdbc:phoenix:thin:[key=value[:key=value...]]
```

简单URL示例

```
jdbc:phoenix:thin:url=http://localhost:8765;serialization=PROTOBUF
```

3. 是否支持QueryServer?

HBase1.x版本使用Phoenix4.x重客户端模式，需要用户自行搭建QueryServer服务。

HBase2.0版本增加HBase SQL服务，默认开启QueryServer。

4. 是否支持Tracing Web Application ?

当前云HBASE上的ali-phoenix此不支持，此功能正在开发中

5.构建同步的二级索引超时怎么办？

HBase1.0上Phoenix4.x版本，需要在客户加上如下配置，并重启客户端。

```
<property>
<name>hbase.rpc.timeout</name>
<value>60000000</value>
</property>
<property>
<name>hbase.client.scanner.timeout.period</name>
<value>60000000</value>
</property>
<property>
<name>phoenix.query.timeoutMs</name>
<value>60000000</value>
</property>
```

HBase2.0上Phoenix5.x版本，可在控制台参数管理中修改上述参数，并重启HBase SQL服务，注意不需要重启HBase，仅重启HBase SQL服务(Phoenix)即可。

6.开通Namespace Mapping

4.x开通步骤：

1). 在客户端增加以下配置， 2).找云 HBase 答疑开通 Server 端参数， 3). 重启 HBase 集群。

```
<configuration>
<property>
<name>phoenix.schema.isNamespaceMappingEnabled</name>
<value>true</value>
</property>
<property>
<name>phoenix.schema.mapSystemTablesToNamespace</name>
<value>true</value>
</property>
</configuration>
```

5.x开通步骤：

1). 控制台 -> 参数配置，配置phoenix.schema.isNamespaceMappingEnabled 和 phoenix.schema.mapSystemTablesToNamespace 为 true。 2). 重启 HBase 集群。 3). 重启 SQL 服务。

注意： 这里需要配置两组相同的参数，参数描述是不相同的，一组是 query server 的，一组是 HBase server 端的，都需要设置。

7.是否支持连接池

Phoenix4.x最新版本的4.12.0.X版本支持，具体参考PhoenixConnectionPool.java

Phoenix5.x版本基于轻客户端实现，PhoenixConnectionPool.java不能使用，建议使用社区第三方连接池，比如mybatis，可参考链接池demo

8.执行创建索引时间太长能否断开客户端链接？

不能断开客户端链接。执行create index主要有两个步骤，第一步在server端同步源表数据到索引表，第二步在客户端发起请求修改索引表状态设置为active。其中第一步一般是客户端发起请求在server端完成后。

9.创建同步索引表，同步索引数据的速度怎么样？

一般情况下1000W数据创建索引需要5-20min, 具体情况视集群配置和资源使用情况而定。

10.创建索引由于时间太长，客户端断开了链接怎么办？

一般情况下当前索引表的状态是building状态的（可以在sqlline中使用 !table命令查看），只有当索引表状态变为active才算真正完成了索引构建。此时有两种解决方法：第一、通过alter index命令rebuild索引。第二、删除building状态的索引表，配置更大的客户端超时时间，重新创建索引。

11.关系型数据库怎么导入云HBASE的phoenix表中？

通过dax(<https://github.com/alibaba/DataX>)或者CDP, Phoenix4.x可利用hbase11xsqlwriter插件写入到Phoenix表中，其中zk的zookeeper.znode.parent配置值为/hbase。Phoenix5.x可使用hbase20xsqlwriter插件写入到Phoenix表中。

12. 查询时发生遇到ERROR 599(42912): Default enable Force index, please set phoenix.force.index=false to disable it....，应该怎么处理？

为了避免查询扫全表，会在SQL编译阶段检查查询条件是否有主键或者索引列作为过滤条件，如果没有会产生此异常。如果查询确实需要非主键或非索引列作为过滤条件的列，phoenix4.x版本可以在客户端的hbase-site.xml文件中配置phoenix.force.index为false，重新打开客户端，即可生效。Phoenix5.x在控制台参数管理中修改phoenix.force.index为false，重启HBase SQL服务即可。

13. 通过springboot使用durid连接池报java.sql.SQLException: java.lang.IllegalArgumentException: Connection is null or closed.

由于Phoenix内部会缓存链接，上层再使用一层连接池时，会出现部分链接被关闭的情况，所以目前不推荐使用此类方式。你可以尝试当前版本自带的连接池，参考问题10。

14. 使用python客户端连接QueryServer时，连接闲置一段时间后再进行读写时报错phoenixdb.errors.InternalError: ('', None, None, None)

由于开源python客户端未实现连接空闲超时重建机制，通过SLB负载均衡连接超时后再次请求发送到其他QueryServer节点导致。

解决办法：

1. 下载阿里phoenix-python客户端：

phoenix-python-client下载

2. 如果已安装phoenix-python驱动，需要进行删除

```
rm -rf /usr/lib/python${version}/site-packages/phoenixdb*
```

3. 解压后在phoenixdb目录安装驱动

```
python setup.py install
```

15、Phoenix是否支持多租户？

暂时不支持，包括grant，revoke命令。