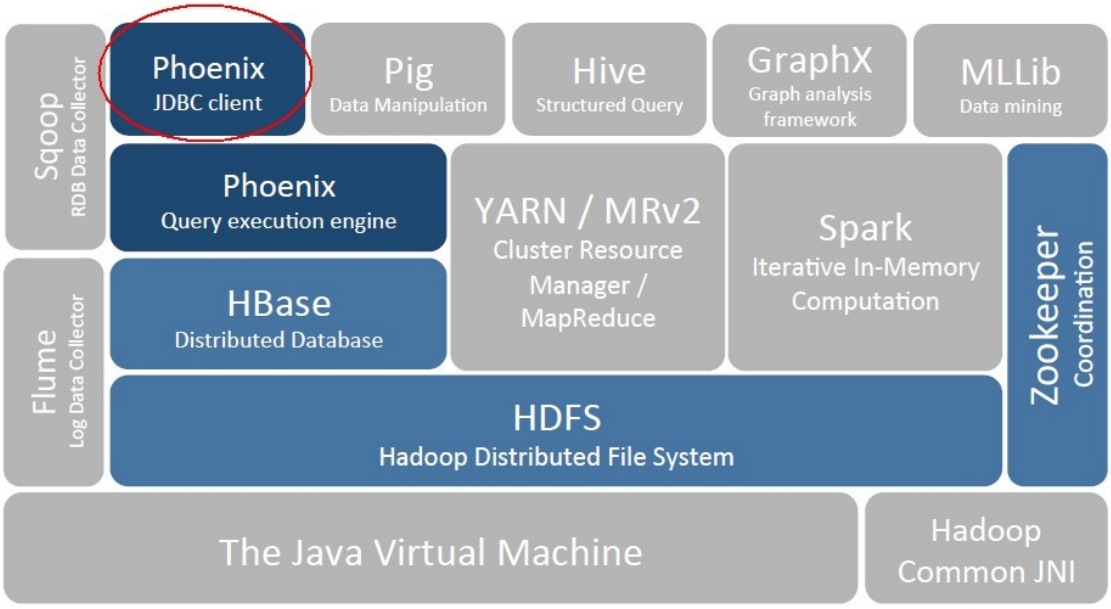
1. Phoenix On HBase

官网：http://phoenix.apache.org/

1.1. Phoenix简介

Phoenix 是 HBase 的开源 SQL 中间层，它允许你使用标准 JDBC 的方式来操作 HBase 上的数据。在 Phoenix 之前，如果你要访问 HBase，只能调用它的 Java API，但相比于使用一行 SQL 就能实现数据查询，HBase 的 API 还是过于复杂。 Phoenix 的理念是 we put sql SQL back in NOSQL ，即你可 以使用标准的 SQL 就能完成对 HBase 上数据的操作。同时这也意味着你可以通过集成 Spring Data JPA 或 Mybatis 等常用的持久层框架来操作 HBase。 其次 Phoenix 的性能表现也非常优异， Phoenix 查询引擎会将 SQL 查询转换为一个或多个 HBase Scan，通过并行执行来生成标准的 JDBC 结果集。它通过直接使用 HBase API 以及协处理器和自定义过 滤器，可以为小型数据查询提供毫秒级的性能，为千万行数据的查询提供秒级的性能。同时 Phoenix 还 拥有二级索引等 HBase 不具备的特性，因为以上的优点，所以 Phoenix 成为了 HBase 最优秀的 SQL 中间层。



1.2. Phoenix安装

我们可以按照官方安装说明进行安装，官方说明如下：

```
1、download and expand our installation tar

2、copy the phoenix server jar that is compatible with your HBase installation
into the lib directory of every region server

3、restart the region servers

4、add the phoenix client jar to the classpath of your HBase client

5、download and setup SQuirrel as your SQL client so you can issue adhoc SQL
against your HBase cluster
```

1.2.1. 下载并解压

官方针对 Apache 版本和 CDH 版本的 HBase 均提供了安装包，按需下载即可。官方下载地址：http://phoenix.apache.org/download.html

```
# 下载
wget https://mirrors.tuna.tsinghua.edu.cn/apache/phoenix/apache-phoenix-5.0.0-
HBase-2.0/bin/apache-phoenix-5.0.0-HBase-2.0-bin.tar.gz
# 解压
tar -zxvf apache-phoenix-5.0.0-HBase-2.0-bin.tar.gz
```

1.2.2. 拷贝Jar包

按照官方文档的说明，需要将 phoenix server jar 添加到所有 Region Servers 的安装目录的 lib 目录下。

这里由于我搭建的是 HBase 伪集群，所以只需要拷贝到当前机器的 HBase 的 lib 目录下。如果是真实集群，则使用 scp 命令分发到所有 Region Servers 机器上。

```
cp ~/apps/apache-phoenix-5.0.0-HBase-2.0-bin/phoenix-5.0.0-HBase-2.0-server.jar
~/apps/hbase-2.2.5/lib
```

1.2.3. 重启 Region Servers

```
# 停止Hbase
stop-hbase.sh
# 启动Hbase
start-hbase.sh
```

1.2.4. 启动Phoenix

在 Phoenix 解压目录下的 bin 目录下执行如下命令，需要指定 Zookeeper 的地址：

- 如果 HBase 采用 Standalone 模式或者伪集群模式搭建，则默认采用内置的 Zookeeper 服务，端口为 2181；
- 如果是 HBase 是集群模式并采用外置的 Zookeeper 集群，则按照自己的实际情况进行指定。

```
./sqlline.py bigdata02:2181
```

1.2.5. 启动结果

启动后则进入了 Phoenix 交互式 SQL 命令行，可以使用 !table 或 !tables 查看当前所有表的信息

1.3. Phoenix 简单使用

1.3.1. 创建表

```
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));
```

新建的表会按照特定的规则转换为 HBase 上的表，关于表的信息，可以通过 Hbase Web UI 进行查看：

1.3.2. 插入数据

Phoenix 中插入数据采用的是 UPSERT 而不是 INSERT ,因为 Phoenix 并没有更新操作，插入相同主键的数据就视为更新，所以 UPSERT 就相当于 UPDATE + INSERT

```
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);
```

1.3.3. 修改数据

```
-- 插入主键相同的数据就视为更新
UPSERT INTO us_population VALUES('NY','New York',999999);
```

1.3.4. 删除数据

```
DELETE FROM us_population WHERE city='Dallas';
```

1.3.5. 查询数据

```
SELECT state as "州",count(city) as "市",sum(population) as "热度"
FROM us_population
GROUP BY state
ORDER BY sum(population) DESC;
```

1.3.6. 退出命令

```
!quit
```

1.3.7. 扩展

从上面的操作中可以看出，Phoenix 支持大多数标准的 SQL 语法。关于 Phoenix 支持的语法、数据类型、函数、序列等详细信息，因为涉及内容很多，可以参考其官方文档，官方文档上有详细的说明：

- 语法 (Grammar)： https://phoenix.apache.org/language/index.html
- 函数 (Functions)： http://phoenix.apache.org/language/functions.html
- 数据类型 (Datatypes)： http://phoenix.apache.org/language/datatypes.html
- 序列 (Sequences) :http://phoenix.apache.org/sequences.html
- 联结查询 (Joins)： http://phoenix.apache.org/joins.html

1.4. Phoenix Java API

因为 Phoenix 遵循 JDBC 规范，并提供了对应的数据库驱动 PhoenixDriver ，这使得采用 Java 语言对其进行操作的时候，就如同对其他关系型数据库一样，下面给出基本的使用示例。

1.4.1. 引入Phoenix core JAR包

如果是 maven 项目，直接在 maven 中央仓库找到对应的版本，导入依赖即可：

```
<!-- https://mvnrepository.com/artifact/org.apache.phoenix/phoenix-core -->
<dependency>
    <groupId>org.apache.phoenix</groupId>
    <artifactId>phoenix-core</artifactId>
    <version>5.0.0-HBase-2.0</version>
</dependency>
```

如果是普通项目，则可以从 Phoenix 解压目录下找到对应的 JAR 包，然后手动引入：

1.4.2. 简单的Java API实例

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class PhoenixJavaApi {
    public static void main(String[] args) throws Exception {
        // 加载数据库驱动
        Class.forName("org.apache.phoenix.jdbc.PhoenixDriver");
        /*
         * 指定数据库地址,格式为 jdbc:phoenix:Zookeeper 地址
         * 如果 HBase 采用 Standalone 模式或者伪集群模式搭建，则 HBase 默认使用内置的
Zookeeper，默认端口为 2181
         */
        Connection connection =
            DriverManager.getConnection("jdbc:phoenix:192.168.200.226:2181");
        PreparedStatement statement = connection.prepareStatement("SELECT * FROM
us_population");
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            System.out.println(resultSet.getString("city") + " " +
resultSet.getInt("population"));
        }
        statement.close();
        connection.close();
    }
}
```