链滴

Apache Kudu 初体验

作者：flowaters

原文链接：https://ld246.com/article/1527948895529

来源网站：链滴

背景

Apache Kudu是Cloudera贡献的开源项目，主要特点是：

- Streamlined Architecture
- Faster Analytics
- Open for Contributions

Kudu是在HBase出现之后，根据Hbase的思想开发的。

HBase一般将数据持久化在HDFS中，使用HDFS的数据一致性；而Kudu是存储在本地，通过raft协来保证强一致和高可靠性。

Kudu的设计目标是快速分析，除了和Hbase一样支持随机读写外，在数据扫描(Scan)的性能比Hbas要好，适合OLAP场景。

另外，Kudu核心模块由C++开发，没有gc风险；而Hbase由Java开发，内存紧张时有gc风险。

详细对比见参考1。

与Parquet, Hbase比较

整理自The Columnar Era: Leveraging Parquet, Arrow and Kudu for High-Performance Analytics

	分析(快速Scan)	随机读写
Parquet	最好	最差
KUDU	较好	较好
HBase	最差	最好

安装

Kudu可以单独安装，也可以在安装CDH时做为一个组件自动安装。

这里单独安装仅仅为了测试，生产环境中还是使用的CDH安装。

单独安装方法，来自于参考官方文档。

另外，本文单独安装方法安装的是kudu 1.4。官方最新的版本是kudu 1.7。

RHEL 7

增加源

wget http://archive.cloudera.com/kudu/redhat/7/x86_64/kudu/cloudera-kudu.repo

保存到文件夹/etc/yum.repos.d/下

可以使用阿里云的镜像，见 http://mirrors.aliyun.com/

安装源

```
sudo yum install kudu                 # Base Kudu files
sudo yum install kudu-master            # Kudu master init.d service script and default confi
uration
sudo yum install kudu-tserver           # Kudu tablet server init.d service script and default c
nfiguration
sudo yum install kudu-client0           # Kudu C++ client shared library
sudo yum install kudu-client-devel        # Kudu C++ client SDK
```

启动服务

```
[root@localhost fw]# /etc/init.d/kudu-tserver start
Started Kudu Tablet Server (kudu-tserver):                 [  确定  ]
[root@localhost fw]# /etc/init.d/kudu-master start
Started Kudu Master Server (kudu-master):                  [  确定  ]
```

确认进程 kudu-tserver 和 kudu-master 都存在

- Master: kudu-master的页面为 http://localhost:8051
- Tablet Server: kudu-tserver的页面为http://localhost:8050

如果远程连接不上的话，确认一下linux的防火墙是不是屏蔽了对应的请求。

配置文件

```
/etc/kudu/conf/master.gflagfile
/etc/kudu/conf/tserver.gflagfile
```

一般不用修改

但是如果是单机测试的话，修改一下最小分片数到1吧。

分别修改这两个配置文件，增加：

```
--unlock_unsafe_flags=true
--allow_unsafe_replication_factor=true
--default_num_replicas=1
--rpc_negotiation_timeout_ms=9000
```

然后重启两个服务。

Python使用

由简单到复杂，先来看python的使用吧。

具体参考了官方kudu-python示例和pydoc kudu帮助手册。

安装 kudu-python

先安装pip，如果已经有了请略过

```
sudo yum -y install epel-release python2-pip python-devel
```

pip安装kudu-python

```
## 准备活动
pip install --upgrade pip
pip install --upgrade setuptools

## 安装包. 对于 kudu1.4，要选择1.2.0的kudu-python
pip install Cython kudu-python==1.2.0
```

查看所有可以安装的版本:

指定个不存在的版本，就会提示出所有可选的版本。

```
pip install kudu-python==99.99.99
Collecting kudu-python==99.99.99
  Could not find a version that satisfies the requirement kudu-python==99.99.99 (from versio
s: 0.1.0, 0.1.1, 0.2.0, 0.3.0, 1.1.0, 1.2.0, 1.7.0)
No matching distribution found for kudu-python==99.99.99
```

使用

连接

```
>>> import kudu
>>> client = kudu.Client("127.0.0.1:7051")
```

查看表

```
>>> client.list_tables()
['java_sample-1527939906616', 'java_sample-1527939874474', 'java_sample-1527938481750',
java_sample-1527938216058']
```

删除表

```
>>> client.delete_table('java_sample-1527939906616')
>>> client.list_tables()
['java_sample-1527939874474', 'java_sample-1527938481750', 'java_sample-1527938216058']
```

创建表

创建schema

```
>>> builder = kudu.schema_builder()
>>> builder.add_column('key', kudu.int32, nullable=False)
>>> builder.add_column('value', kudu.string, nullable=False).compression('lz4')
>>> builder.set_primary_keys(['key'])
>>> schema = builder.build()
```

创建partition

```
>>> from kudu.client import Partitioning
>>> partitioning=Partitioning().add_hash_partitions(['key'], 2)
```

创建表

```
>>> client.create_table('table_demo',schema,partitioning)
```

查看表结构

```
>>> table = client.table('java_sample-1527939874474')
>>> table.schema
kudu.Schema {
  key    int32 NOT NULL
  value  string NOT NULL
  PRIMARY KEY (key)
}
```

查看表内容

```
>>> scanner = table.scanner()
>>> scanner.open()
<kudu.client.Scanner object at 0x7fb7f78982d0>
>>> tuples = scanner.read_all_tuples()
>>> tuples
[(0, 'value 0'), (1, 'value 1'), (2, 'value 2')]
```

插入数据到表

```
>>> session = client.new_session()
>>> op = table.new_insert()
>>> op['key']=3
>>> op['value']='value 3'
>>> session.apply(op)
>>> session.flush()
```

验证插入成功

```
>>> scanner = table.scanner()
>>> scanner.open()
<kudu.client.Scanner object at 0x7fb7f7898960>
>>> scanner.read_all_tuples()
[(0, 'value 0'), (1, 'value 1'), (2, 'value 2'), (3, 'value 3')]
```

其它操作

其它操作有：

- new_delete: 删除数据
- new_update: 更新数据
- new_upsert: 如果存在则更新数据，不存在则插入数据
- rename: 更改表名

Java使用

这里java依赖了1.4的kudu-client，示例代码见github。

Java依赖

```
<dependencies>
  <dependency>
    <groupId>org.apache.kudu</groupId>
    <artifactId>kudu-client</artifactId>
    <version>1.4.0</version>
  </dependency>
</dependencies>
```

示例代码

演示了创建表，写入数据，扫描数据，和删除表的过程。

代码来自github

```
import java.util.ArrayList;
import java.util.List;

import org.apache.kudu.ColumnSchema;
import org.apache.kudu.Schema;
import org.apache.kudu.Type;
import org.apache.kudu.client.CreateTableOptions;
import org.apache.kudu.client.Insert;
import org.apache.kudu.client.KuduClient;
import org.apache.kudu.client.KuduScanner;
import org.apache.kudu.client.KuduSession;
import org.apache.kudu.client.KuduTable;
import org.apache.kudu.client.PartialRow;
import org.apache.kudu.client.RowResult;
import org.apache.kudu.client.RowResultIterator;

public class Sample {

  private static final String KUDU_MASTER = System.getProperty("kuduMaster", "192.168.199
137:7051");

  public static void main(String[] args) {
    System.out.println("-----------------------------------------------");
    System.out.println("Will try to connect to Kudu master at " + KUDU_MASTER);
    System.out.println("Run with -DkuduMaster=myHost:port to override.");
    System.out.println("-----------------------------------------------");
    String tableName = "java_sample-" + System.currentTimeMillis();
    KuduClient client = new KuduClient.KuduClientBuilder(KUDU_MASTER).build();

    try {
      // 建表
      List<ColumnSchema> columns = new ArrayList(2);
      columns.add(new ColumnSchema.ColumnSchemaBuilder("key", Type.INT32).key(true).
```

```
uild());
            columns.add(new ColumnSchema.ColumnSchemaBuilder("value", Type.STRING).build()
;
            List<String> rangeKeys = new ArrayList<>();
            rangeKeys.add("key");

            Schema schema = new Schema(columns);
            client.createTable(tableName, schema, new CreateTableOptions().setRangePartitionCo
umns(rangeKeys));

            // 插入数据
            KuduTable table = client.openTable(tableName);
            KuduSession session = client.newSession();
            for (int i = 0; i < 3; i++) {
                Insert insert = table.newInsert();
                PartialRow row = insert.getRow();
                row.addInt(0, i);
                row.addString(1, "value " + i);
                session.apply(insert);
            }

            // 查询数据
            List<String> projectColumns = new ArrayList<>(1);
            projectColumns.add("value");
            KuduScanner scanner = client.newScannerBuilder(table).setProjectedColumnNames(p
ojectColumns).build();
            while (scanner.hasMoreRows()) {
                RowResultIterator results = scanner.nextRows();
                while (results.hasNext()) {
                    RowResult result = results.next();
                    System.out.println(result.getString(0));
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                // 删除表
                client.deleteTable(tableName);
            } catch (Exception e) {
                e.printStackTrace();
            } finally {
                try {
                    client.shutdown();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

运行结果

```
-----------------------------------------------
Will try to connect to Kudu master at 192.168.199.137:7051
Run with -DkuduMaster=myHost:port to override.
-----------------------------------------------
[New I/O worker #2] WARN org.apache.kudu.util.NetUtil - Slow DNS lookup! Resolved IP of `l
calhost' to localhost/127.0.0.1 in 3168448ns
value 0
value 1
value 2
```

参考

- Apache Kudu 1.4.0 中文文档
- 官方安装文档
- kudu和hbase的区别和联系
- KUDU安装
- kudu 1.4 单机安装后 运行java-sample
- can not install kudu-python
- getting started with the cloudera kudu storage engine in python
- github cloudera kudu
- The Columnar Era: Leveraging Parquet, Arrow and Kudu for High-Performance Analytics