通过MapReduce对HBase数据进行统计分析

目录

[一、概述 1](#_Toc418520567)

[二、应用场景 1](#_Toc418520568)

[三、Map程序 2](#_Toc418520569)

[四、Reduce程序 2](#_Toc418520570)

[五、Combiner程序 3](#_Toc418520571)

[六、作业执行 3](#_Toc418520572)

# 一、概述

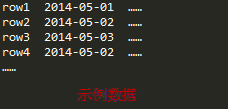
HBase作为一种kv数据库，能够很好的面对高吞吐率的在线数据读写服务，尤其是写操作，但是在非rowkey多条件查询、数据分析、统计等场景下，HBase表现的就不是很好了，这些场景下就比较适合MapReduce出场了。

# 二、应用场景

这里就通过一个示例程序，一步一步带着大家学习 MapReduce的开发。

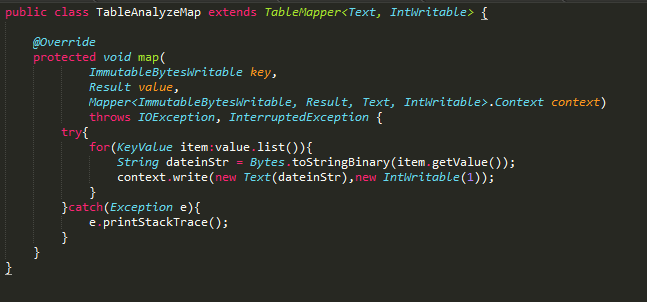
假设我们有一张业务表（biz\_detail），它有一列是业务发生日期（ts），每行数据就是一条业务发生，现在我们要统计业务发生日期每天的业务量（数据行数），对于

这样一个简单统计需求，我们可以通过MapReduce程序来实现。



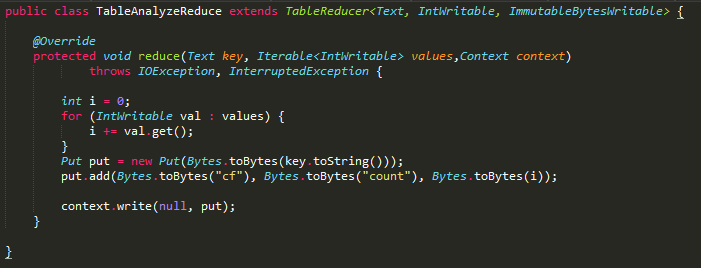
# 三、Map程序

TableAnalyzeMap继承HBase的TableMapper基类，把ts看做文本，行数自然是整数，所以key-value输出类型自然是<Text,IntWritable>，具体代码如下：



# 四、Reduce程序

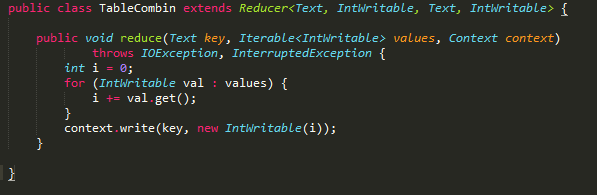
TableAnalyzeReduce继承HBase的TableReducer基类，这里我们把最终聚合后的结果写到目标表中，rowkey对应ts，行数对应目标表的cf:count列，具体代码如下：



# 五、Combiner程序

分析map程序，我们可以看到，map阶段处理后的数据是没有任何合并的，key为日期，

value都是1，这样的话，如果直接进入reduce阶段，要分发的数据量还是比较大的，会造成网络负担，针对这个问题，我们可以在map阶段后，做一下本地reduce，这样进入reduce的阶段的数据量会大大减少。



# 六、作业执行

程序接收四个参数，依次为：业务表名称、日期字段列族、日期字段列名、目标表名。

当然你也可以使用Apache Commons CLI类解析命令行参数，这里就不做说明了。

其中scan.setCaching(500)设置每次读取行数，根据实际情况进行配置，scan.setCacheBlocks(false)告诉HBase本次扫描的数据不要放入缓存中，其他就不再详述，通过代码就能够看明白。



这样，整个mapreduce程序我们已经开发完成，生成相应jar包后，我们就可以执行了。



当然，如果要在hbase集群外执行的话，作业执行程序里就要配置hbase必要环境信息。