基于HBase模糊查询的RowKey设计

目录

[一、概述 1](#_Toc420403721)

[二、业务场景 1](#_Toc420403722)

[三、RowKey设计 1](#_Toc420403723)

[四、过滤器选择 2](#_Toc420403724)

[五、进一步说明 3](#_Toc420403725)

# 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，但是在非RowKey字段的条件查询上，性能表现就不是太理想，本文探讨一种通过对RowKey的设计，提升非RowKey字段模糊查询性能的设计方法。

# 二、业务场景

医院一般检验流程：检验申请单(bill)→检验报告（report），现在要求能够通过申请单号以及检验指标对检验报告进行快速模糊查询。其中申请单号(OrderNo)以及检验指标代码(TestCode)是检验报告的两个字段。

# 三、RowKey设计

如果直接对检验报告的两个字段进行模糊匹配查询，HBase需要扫描整个数据集，逐条进行筛选，性能表现可想而知。

为了提升查询性能，我们要想办法把这两个字段组合到表的RowKey里，初步想法：

|  |  |  |  |
| --- | --- | --- | --- |
| RowKey | f1 | | |
| OrderNo | TestCode | … |
| 0001\_2003\_\* | 1 | 2003 |  |
| 0002\_1025\_\* | 2 | 1025 |  |

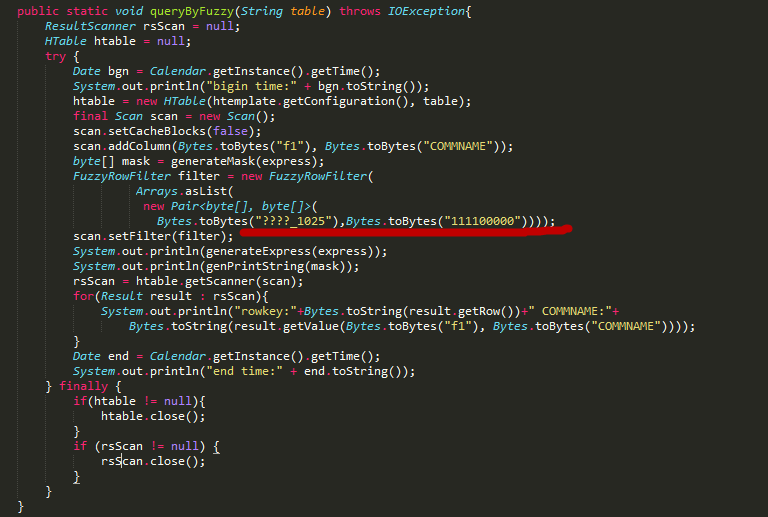
RowKey由申请单号和检验指标代码组合而成，当然这两个字段外还会有其它拼接值，比如时间戳、序号等，这些就不再展开说。申请单号作为RowKey的前缀，如果有基于申请单号的查询，可以直接走HBase的前缀查询，性能会非常高，第二字段是检验指标代码，如果有基于检验指标代码的查询，可以走HBase的模糊查询，两个字段的组合情况，就不一一说了，因为最终它都能落地到以上两种方式的查询。

# 四、过滤器选择

前缀查询用PrefixFilter，很简单，直接上代码：



模糊查询用FuzzyRowFilter，相对复杂一点，假如，我们要查询检验指标代码是1025的所有检验报告，代码如下：



FuzzyRowFilter需要把模糊字段值组织成Pair对，firt是定长的RowKey串，其中模糊匹配部分用?或[\\x00](file:///\\x00)占位，second是first的掩码，模糊匹配部分是1，固定部分是0，first和second长度相同。

# 五、进一步说明

前提条件：

1. 在HBase装入数据前，需要有RowKey的设计规划。
2. HBase RowKey的每个组合部分都要固定长度，不够长度的要有填充策略。