

[Cassandra教程] (十) 二级索引(Secondary Index)

2016/9/16 0 Comments 2,926 Views 0 Times

1. 什么是二级索引?

我们前面已经介绍过Cassandra之中有各种Key, 比如Primary Key, Cluster Key 等等。如果您对这部分概念并不熟悉, 可以参考之前的文章: [\[Cassandra教程\] \(四\) 使用Key的正确姿势](#) 对于Cassandra来说, 一级索引就是Primary Key. 因为查询的时候, 可以直接根据Key算出token然后直接获取对应的记录。

而二级索引, 作为辅助索引就是为了找到一级索引。然后再通过一级索引找到真正的值

原文链接: <http://www.flyml.net/2016/09/16/cassandra-tutorial-secondary-index/2>. 二级索引的原理

Cassandra之中的索引的实现相对MySQL的索引来说就要简单粗暴很多了。他实际上是自动偷偷新创建了一张表格, 同时将原始表格之中的索引字段作为新索引表的Primary Key! 并且存储的值为原始数据的Primary Key

因此, 什么样的数据、字段适合做二级索引, 也就很清楚了。

我们翻译一下官方的解释:

参考网址: https://docs.datastax.com/en/cql/3.1/cql/ddl/ddl_when_use_index_c.html

原文链接: <http://www.flyml.net/2016/09/16/cassandra-tutorial-secondary-index/3>. 什么时候不适合用索引

1. High-cardinality 列。 相当于这一列的值很多很多的时候。
 1. 因为查询了很多结果只能取出一小部分数据集
2. counter 类型的列
3. 删除、更新太过频繁的列
 1. Cassandra删除、更新数据都会给老数据设置一个Tombstone (墓碑)。当Tombstone的数据查过10K的时候, 就会报错
 2. 再加上需要同步的更新索引表, Tombstone本身的标记也会很消耗资源
4. 数据集值太多
 1. 原文英文没怎么看懂, 这是笔者自己的理解。
 2. 再次强调一下: 二级索引里面存储的是原始数据的Primary Key。因此如果一次查询的数据过多就会遇到超时异常


总结一下:

就是索引对应的数据值不能太多也不能太少。 太多就超时, 太少就浪费资源 (需要创建太多的Primary Key)。 同时索引的列还要稳定, 不能频繁的删除或者更新~

所以个人认为，能不用Cassandra之中的索引就不要用，还不如自己显示的创建一个

原文链接: <http://www.flyml.net/2016/09/16/cassandra-tutorial-secondary-index/4>. 如何使用索引

索引的使用非常简单，参考下面的代码以及相应的注释：

MySQL  -- 创建索引
CREATE INDEX
原文链接: <http://www.flyml.net/2016/09/16/cassandra-tutorial-secondary-index/>

本文为原创文章，转载请注明[出处](#)原文链接: <http://www.flyml.net/2016/09/16/cassandra-tutorial-secondary-index/>

