

MongoDB 查询分析

MongoDB 查询分析可以确保我们所建立的索引是否有效，是查询语句性能分析的重要工具。

MongoDB 查询分析常用函数有：`explain()` 和 `hint()`。

使用 `explain()`

`explain` 操作提供了查询信息，使用索引及查询统计等。有利于我们对索引的优化。

接下来我们在 `users` 集合中创建 `gender` 和 `user_name` 的索引：

```
>db.users.ensureIndex({gender:1,user_name:1})
```

现在在查询语句中使用 `explain`：

```
>db.users.find({gender:"M"},{user_name:1,_id:0}).explain()
```

以上的 `explain()` 查询返回如下结果：

```
{
  "cursor" : "BtreeCursor gender_1_user_name_1",
  "isMultiKey" : false,
  "n" : 1,
  "nscannedObjects" : 0,
  "nscanned" : 1,
  "nscannedObjectsAllPlans" : 0,
  "nscannedAllPlans" : 1,
  "scanAndOrder" : false,
  "indexOnly" : true,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 0,
  "indexBounds" : {
    "gender" : [
      [
        "M",
        "M"
      ]
    ],
    "user_name" : [
      [
        {
          "$minElement" : 1
        }
      ]
    ]
  }
}
```

```
{
  "$maxElement" : 1
}
]
}
}
```

现在，我们看看这个结果集的字段：

- **indexOnly**: 字段为 true，表示我们使用了索引。
- **cursor**：因为这个查询使用了索引，MongoDB 中索引存储在B树结构中，所以这也是使用了 BtreeCursor 类型的游标。如果没有使用索引，游标的类型是 BasicCursor。这个键还会给出你所使用的索引的名称，你通过这个名称可以查看当前数据库下的system.indexes集合（系统自动创建，由于存储索引信息，这个稍微会提到）来得到索引的详细信息。
- **n**：当前查询返回的文档数量。
- **nscanned/nscannedObjects**：表明当前这次查询一共扫描了集合中多少个文档，我们的目的是，让这个数值和返回文档的数量越接近越好。
- **millis**：当前查询所需时间，毫秒数。
- **indexBounds**：当前查询具体使用的索引。

使用 hint()

虽然MongoDB查询优化器一般工作的很不错，但是也可以使用 hint 来强制 MongoDB 使用一个指定的索引。

这种方法某些情形下会提升性能。一个有索引的 collection 并且执行一个多字段的查询(一些字段已经索引了)。

如下查询实例指定了使用 gender 和 user_name 索引字段来查询：

```
>db.users.find({gender:"M"},{user_name:1,_id:0}).hint({gender:1,user_name:1})
```

可以使用 explain() 函数来分析以上查询：

```
>db.users.find({gender:"M"},{user_name:1,_id:0}).hint({gender:1,user_name:1}).explain()
```

← MongoDB 覆盖索引查询

MongoDB 原子操作 →

 点我分享笔记

