MongoDB中MapReduce介绍与使用

一、简介

在用MongoDB查询返回的数据量很大的情况下，做一些比较复杂的统计和聚合操作做花费的时间很长的时候，可以用MongoDB中的MapReduce进行实现

MapReduce是个非常灵活和强大的数据聚合工具。它的好处是可以把一个聚合任务分解为多个小的任务，分配到多服务器上并行处理。MongoDB也提供了MapReduce，当然查询语肯定是JavaScript。

MongoDB中的MapReduce主要有以下几阶段：

- Map：把一个操作Map到集合中的每一个文档
- Shuffle: 根据Key分组对文档，并且为每个不同的Key生成一系列(>=1个)的值表(List of values)。
- Reduce: 处理值表中的元素，直到值表中只有一个元素。然后将值表返回到Shuffle过程，循环处理，直到每个Key只对应一个值表，并且此值表中只有一个元素，这就是MR的结果。
- Finalize：此步骤不是必须的。在得到MR最终结果后，再进行一些数据“修剪”性质的处理。

MongoDB中使用emit函数向MapReduce提供Key/Value对。

Reduce函数接受两个参数：Key,emits. Key即为emit函数中的Key。 emits是一个数组，它的元素就是emit函数提供的Value。

Reduce函数的返回结果必须要能被Map或者Reduce重复使用，所以返回结果必须与emits中元素结构一致。

Map或者Reduce函数中的this关键字，代表当前被Mapping文档。

二、介绍

```
db.runCommand({
    mapreduce:<collection>,
    map:<mapfunction>,
    reduce:<reducefunction>,
    [,query:<query filter object>]
    [,sort:<sorts the input objects using this key.Useful for optimization,like sorting by the emit key for fewer reduces>]
    [,limit:<number of objects to return from collection>]
    [,out:<see output options below>]
    [,keeptemp:<true|false>]
    [,finalize:<finalizefunction>]
    [,scope:<object where fields go into javascript global scope>]
    [, jsMode : boolean, default true]
    [,verbose:true]
});
```

参数说明：

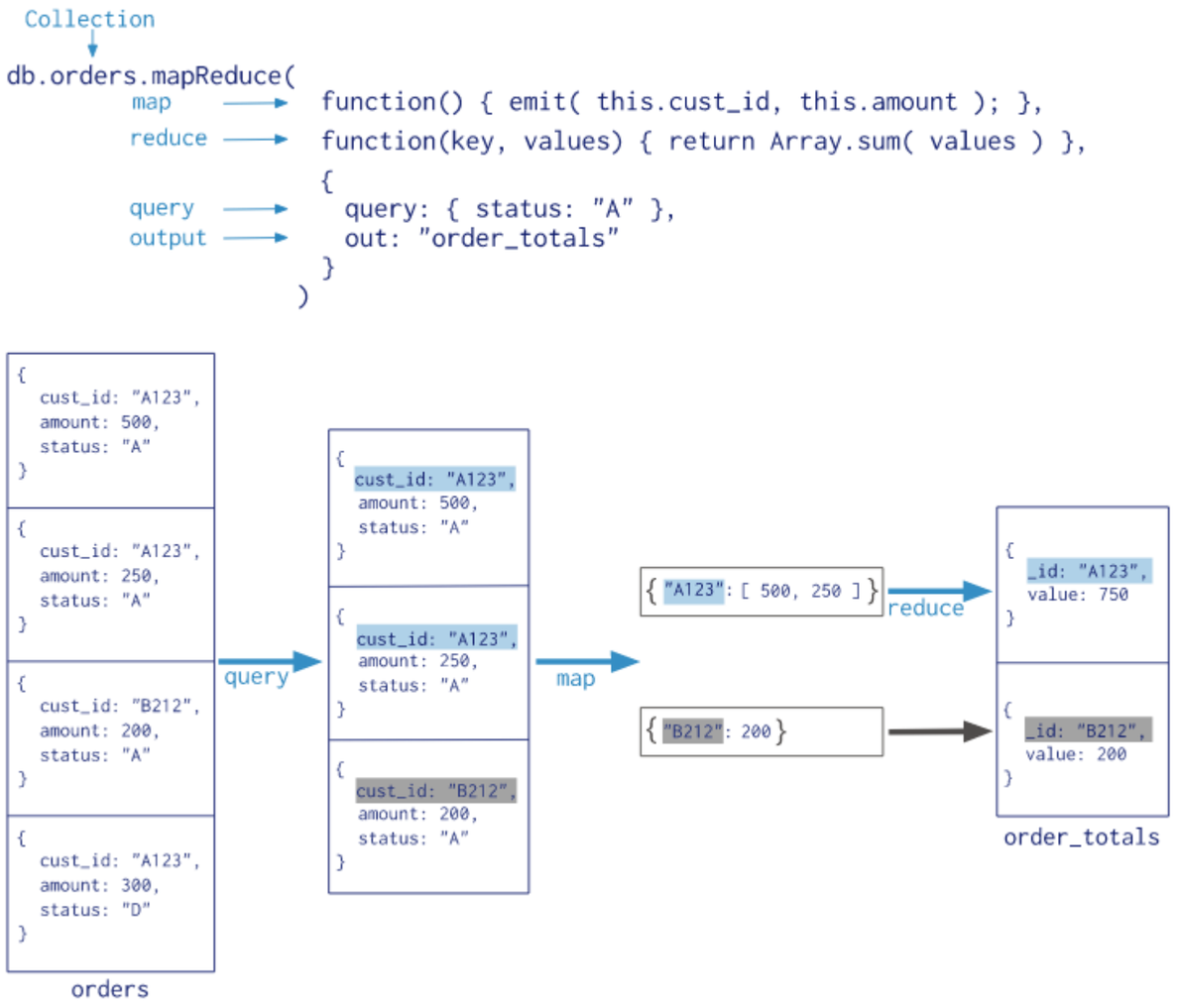
- Mapreduce：要操作的目标集合
- Map：映射函数（生成键值对序列，作为reduce函数参数）
- Reduce：统计函数
- Query：目标记录过滤
- Sort：目标记录排序
- Limit：限制目标记录数量
- Out：统计结果存放集合（不指定使用临时集合，在客户端断开后自动删除）
- Keeptemp：是否保留临时集合
- Finalize：最终处理函数（对reduce返回结果进行最终整理后存入结果集合）
- Scope：向map、reduce、finalize导入外部变量
- jsMode说明:为false时 BSON-->JS-->map-->BSON-->JS-->reduce-->BSON,可处理非常大的mapreduce,为true时 BSON-->js-->map-->reduce-->BSON
- Verbose：显示详细的时间统计信息

行查询的步骤

- MapReduce对指定的集合Collection进行查询
- 对A的结果集进行mapper方法采集
- 对B的结果执行finalize方法处理
- 最终结果集输出到临时Collection中
- 断开连接，临时Collection删除或保留

需要注意的

以下是来自文档的图，可以清楚的说明 Map-Reduce 的执行过程。



In this map-reduce operation, MongoDB applies the *map* phase to each input document (i.e. the documents in the collection that match the query condition). The map function emits key-value pairs. For those keys that have multiple values, MongoDB applies the *reduce* phase, which collects and condenses the aggregated data. MongoDB then stores the results in a collection. Optionally, the output of the reduce function may pass through a *finalize* function to further condense or process the results of the aggregation.

All map-reduce functions in MongoDB are JavaScript and run within the `mongod` process. Map-reduce operations take the documents of a single collection as the *input* and can perform any arbitrary sorting and limiting before beginning the map stage. `mapReduce` can return the results of a map-reduce operation as a document, or may write the results to collections. The input and the output collections may be sharded.

NOTE

For most aggregation operations, the Aggregation Pipeline provides better performance and more coherent interface. However, map-reduce operations provide some flexibility that is not presently available in the aggregation pipeline.

Map-Reduce 的执行过程是先 `map` 然后 `reduce` 么？仔细再看一遍上文的图，不是每次 `map` 都有 `reduce` 的！如果 `map` 的结果不是数组，mongodb 就不会执行 `reduce`。很合理的处理逻辑。

对于 `map` 到的数据，如果在 `reduce` 时希望做统一的处理，一定会发现数据结果是不完整的。

三、查询分析

测试数据：

```
> db.test.find()
{ "_id" : ObjectId("5a1d45ab893253f4d2e4bf91"), "name" : "yy1", "age" : "22" }
{ "_id" : ObjectId("5a1d45b1893253f4d2e4bf92"), "name" : "yy2", "age" : "23" }
{ "_id" : ObjectId("5a1d45c5893253f4d2e4bf93"), "name" : "yy3", "age" : "24" }
{ "_id" : ObjectId("5a1d45d4893253f4d2e4bf94"), "name" : "yy5", "age" : "25" }
{ "_id" : ObjectId("5a1d45f7893253f4d2e4bf95"), "name" : "yy6", "age" : "26" }
{ "_id" : ObjectId("5a1d45ff893253f4d2e4bf96"), "name" : "yy4", "age" : "25" }
```

1、查询年龄大于23岁的

map：

```
var m  = function(){if(this.age > 23) emit(this.age,{name:this.name})};
```

reduce：

```
var r = function(key,values){return JSON.stringify(values);}
```

或者：

```
var r = function(key,values){ var ret={names:values};return ret;}
```

生成结果集：

```
var res = db.runCommand({mapreduce:"test",map:m,reduce:r,out:"emp_res"})
```

查询：

```
> db.emp_res.find()
{ "_id" : "24", "value" : { "name" : "yy3" } }
{ "_id" : "25", "value" : "[{\"name\":\"yy5\"},{\"name\":\"yy4\"}]" }
{ "_id" : "26", "value" : { "name" : "yy6" } }
```

或者：

```
> db.emp_res.find()
{ "_id" : "24", "value" : { "name" : "yy3" } }
{ "_id" : "25", "value" : { "names" : [ { "name" : "yy5" }, { "name" : "yy4" } ] } }
{ "_id" : "26", "value" : { "name" : "yy6" } }
```

最后，还可以编写finalize函数对reduce的返回值做最后处理：

```
var f=function(key,rval){ if(key==24){ rval.msg="do somethings";} return rval }
```

生成结果集：

```
> var f=function(key,rval){ if(key==24){ rval.msg="do somethings";} return rval }
> db.emp_res.find()
{ "_id" : "24", "value" : { "name" : "yy3", "msg" : "do somethings" } }
{ "_id" : "25", "value" : { "names" : [ { "name" : "yy5" }, { "name" : "yy4" } ] } }
{ "_id" : "26", "value" : { "name" : "yy6" } }
```

```
>
```

2、过滤出来age=25的

方法1：

```
> var m  = function(){ emit(this.age,{name:this.name})};
> var r = function(key,values){ var ret={names:values};return ret;}
> var res = db.runCommand({mapreduce:"test",map:m,reduce:r,finalize:f,query:{age:"25"},out:"emp_res"})
> db.emp_res.find()
{ "_id" : "25", "value" : { "names" : [ { "name" : "yy5" }, { "name" : "yy4" } ] } }
```

方法2：

```
> var m  = function(){ emit(this.age,{p:[this.name]})};
> var r = function(key, values) {
    var ret = {p:[]};
    for(var i = 0; i < values.length; i++){
        ret.p.push(values[i].p[0]);
    }
    return ret;
};
> var res = db.runCommand({mapreduce:"test",map:m,reduce:r,finalize:f,query:{age:"25"},out:"emp_res"})
```

方法3：

```
> var m = function() {
    emit(this.age, {name:[this.name]});
};
> var r = func   tion(key, values) {
    var ret = {locs:[]}
    for(var i = 0; i < values.length; i++){
        ret.locs.push(values[i].locs[0]);
    }
    return ret;
};
> var res = db.runCommand({mapreduce:"test",map:map,reduce:reduce,finalize:f,query:{age:"25"},out:"emp_res"})
```

这个过程中遇到很多坑，需要多练习，多debug

参考

- MongoDB官方文档https://docs.mongodb.org/manual/core/map-reduce/ 解释很详细，图片到位，简单易懂
- http://thejackalofjavascript.com/mapreduce-in-mongodb/ MapReduce使用的例子
- http://stackoverflow.com/questions/8175015/mongodb-mapreduce-reduce-multiple-not-supported-yethttp://stackoverflow.com/questions/8175015/mongodb-mapreduce-reduce-multiple-not-supported-yet
- http://stackoverflow.com/questions/13963483/how-to-get-print-output-for-debugging-map-reduce-in-mongoid
- http://stackoverflow.com/questions/7527126/mongodb-how-to-debug-map-reduce-on-mongodb-shell

- http://stackoverflow.com/questions/8099991/rejoining-split-mapreduce-arrays-in-mongohttp://stackoverflow.com/questions/8099991/rejoining-split-mapreduce-arrays-in-mongo

分类: Mongodb

标签: mongo, mongodb, MapReduce

« 上一篇：Centos下MongoDB的安装与配置
» 下一篇：MongoDB中聚合工具Aggregate等的介绍与使用

posted @ 2017-11-29 00:27 踏雪无痕SS 阅读(408) 评论(0) 编辑 收藏

最新IT新闻:
· 俞敏洪：不需要有个清晰的梦想才去做，需要的是每一天都要前行
· 纳德拉：Office 365的增长机会大过公司历史上任何产品
· 三星官宣Galaxy Note 9：预装Bixby2.0 9月发布
· 微软量子开发套件更新 现已支持macOS和Linux
· 谷歌旗下DeepMind教AI预测死亡
» 更多新闻...

最新知识库文章:
· 和程序员谈恋爱
· 学会学习
· 优秀技术人的管理陷阱
· 作为一个程序员，数学对你到底有多重要
· 领域驱动设计在互联网业务开发中的实践
» 更多知识库文章...

历史上的今天:
2016-11-29 分布式搜索引擎Elasticsearch性能优化与配置
2015-11-29 【msql】关于redo 和 undo log