

<天上白玉京>

博客园 | 首页 | 新随笔 | 联系 | 订阅  | 管理

昵称：<天上白玉京>

园龄：10个月

粉丝：2

关注：0

+加关注

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- mongodb(2)
- redis(1)
- String 对象(1)
- StringBuilder(1)
- websocket(1)
- 拆箱(1)
- 队列(1)
- 泛型(1)
- 进程通信(1)
- 内存资源(1)
- 更多

随笔档案

- 2018年2月 (3)
- 2018年1月 (4)
- 2017年10月 (2)

最新评论

- 1. Re:使用websocket  
实现在线聊天功能  
以聊天室 功能 写的话  
如果可以 让整个聊天室  
的人互相 直接 连接，那

## MONGODB批量导入及简单的性能优化

今天简单分享一下MongoDB使用过程中的一些性能优化，其实并不只适用MongoDB,其他数据库多少也可适用。

首先先随机导入一千万条数据。这里我分段导入的，因为mongo的BsonDocument一次导入的数据有限制，之前有一次最多导入20w左右，当然那次的对象字段要多很多，

所以本次测试每次导入为10w。咻咻咻咻咻咻咻咻咻咻的一声就导完了。



```
/// <summary>
/// 批量导入
/// </summary>
public void ImportBatch()
{
    string[] nameArr = { "周", "吴", "郑", "王" };
    string[] addressArr = { "浙江省杭州市", "浙江杭州", "浙江省杭州市滨江区", "北京", "上海", "广州", "深圳" };
    int[] ageArr = { 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 };
    int[] sexArr = { 0, 1 };
    //总条数一千万条
    int count = 10000000;
    //每次导入
    int size = 100000;
    //分num次导入
    var num = (int)Math.Ceiling((double)count / (double)size);
    for (int i = 0; i < num; i++)
    {
        //最后一批导入
        List<BsonDocument> docs = new List<BsonDocument>();
        for (int j = i * size; j < (i + 1) * size; j++)
        {
            var user = new User()
            {
                Name = GetStrRandomNumber(nameArr) + j,
                Age = GetIntRandomNumber(ageArr),
                Address = GetStrRandomNumber(addressArr),
                Sex = GetIntRandomNumber(sexArr),
            };
            var json = JsonHelper.SerializeObject(user);
            BsonDocument document = BsonDocument.Parse(json);
```

么服务器那边的效率会高很多。

--SoarNo1

2. Re:使用websocket实现在线聊天功能  
百度了一下SuperSocket的一些特性：高性能的事件驱动通信;非常简单易用;你只需要通过创建几个类就能获得一个健壮的 Socket 服务器;内置的命令行协议让你能够迅速创建一个网络命令行接口服务.....

--<天上白玉京>

3. Re:使用websocket实现在线聊天功能  
不懂

--~雨落忧伤~

4. Re:使用websocket实现在线聊天功能  
主要还是看服务器的性能啊，确实性能堪忧

--Visitors

5. Re:使用websocket实现在线聊天功能  
/dx

--CooMark

### 阅读排行榜

- 1. 使用websocket实现在线聊天功能(2548)
- 2. 简单使用redis实现ss o单点登录(600)
- 3. C#操作MongoDB的简单实例(327)
- 4. 使用redis实现生产者消费者模式(280)
- 5. MongoDB批量导入及简单的性能优化(224)

### 评论排行榜

- 1. 使用websocket实现在线聊天功能(15)

### 推荐排行榜

- 1. 使用websocket实现在线聊天功能(3)

```
        docs.Add(document);
    }
    //导入mongodb
    mongoServer.ImportBatch(collName, docs);
}
}
/// <summary>
/// 随机获取int数组的值
/// </summary>
/// <param name="a"></param>
/// <returns></returns>
static int GetIntRandomNumber(int[] a)
{
    Random rnd = new Random();
    int index = rnd.Next(a.Length);
    return a[index];
}
/// <summary>
/// 随机获取string数组的值
/// </summary>
/// <param name="a"></param>
/// <returns></returns>
static string GetStrRandomNumber(string[] a)
{
    Random rnd = new Random();
    int index = rnd.Next(a.Length);
    return a[index];
}
}
/// <summary>
/// 批量导入
/// </summary>
/// <param name="collectionName"></param>
/// <param name="docs"></param>
public void ImportBatch(string collectionName, List<BsonDocument> docs)
{
    var collection = database.GetCollection<BsonDocument>(collectionName);
    collection.InsertMany(docs);
}
}
```



然后进行测试，先去看下索引db.getCollection('users').AgetIndexes()，可以看到主键\_id是索引

New Connectionlocalhost:27017UsersCenter

db.getCollection('users').AgetIndexes()

0 sec.

Key	Value	Type
▼ (1)	[ 1 element ]	Array
▼ [0]	{ 4 fields }	Object
v	2	Int32
> key	{ 1 field }	Object
name	_id_	String
ns	UsersCenter.users	String

然后随便找一条数据测试。

db.getCollection('users').find({"Name":"王27"})		
users	5.638 sec.	
Key	Value	
(1) ObjectId("5a5df70862a72330a4448d31")	{ 5 fields }	
_id	ObjectId("5a5df70862a72330a4448d31")	
Name	王27	
Sex	1	
Age	25	
Address	深圳	

  

db.getCollection('users').find({_id:ObjectId("5a5df70862a72330a4448d31")})		
users	0.002 sec.	
Key	Value	Type
(1) ObjectId("5a5df70862a72330a4448d31")	{ 5 fields }	Object
_id	ObjectId("5a5df70862a72330a4448d31")	ObjectId
Name	王27	String
Sex	1	Int32
Age	25	Int32
Address	深圳	String

可以看到用主键\_id(0.002s)性能比Name(5.638)明显快很多。然后给Name建立个索引，然后再用Name(0.042)做搜索条件。

db.getCollection('users').ensureIndex({"username":1})		
48.944 sec.		
Key	Value	Type
(1)	{ 4 fields }	Object
createdCollectionAutomatically	false	Boolean
numIndexesBefore	1	Int32
numIndexesAfter	2	Int32
ok	1.0	Double

再测试下Age字段，用Age倒序排，取前100条。

db.getCollection('users').find({}).sort({"Age":-1}).limit(100)		
users	23.175 sec.	
Key	Value	Type
> (13) ObjectId("5a5df70862a72330a4448d23")	{ 5 fields }	Object
> (14) ObjectId("5a5df70862a72330a4448d24")	{ 5 fields }	Object
> (15) ObjectId("5a5df70862a72330a4448d25")	{ 5 fields }	Object
> (16) ObjectId("5a5df70862a72330a4448d26")	{ 5 fields }	Object
> (17) ObjectId("5a5df70862a72330a4448d27")	{ 5 fields }	Object
> (18) ObjectId("5a5df70862a72330a4448d28")	{ 5 fields }	Object
> (19) ObjectId("5a5df70862a72330a4448d29")	{ 5 fields }	Object
> (20) ObjectId("5a5df70862a72330a4448d2a")	{ 5 fields }	Object
> (21) ObjectId("5a5df70862a72330a4448d2b")	{ 5 fields }	Object

建立索引之后。db.getCollection('users').ensureIndex({"Age":-1})

```
db.getCollection('users').find({}).sort({"Age":-1}).limit(100)|
```

users0.023 sec.

Key	Value	Type
> (1) ObjectId("5a5df70862a72330a4448d17")	{ 5 fields }	Object
> (2) ObjectId("5a5df70862a72330a4448d18")	{ 5 fields }	Object
> (3) ObjectId("5a5df70862a72330a4448d19")	{ 5 fields }	Object
> (4) ObjectId("5a5df70862a72330a4448d1a")	{ 5 fields }	Object
> (5) ObjectId("5a5df70862a72330a4448d1b")	{ 5 fields }	Object
> (6) ObjectId("5a5df70862a72330a4448d1c")	{ 5 fields }	Object
> (7) ObjectId("5a5df70862a72330a4448d1d")	{ 5 fields }	Object
> (8) ObjectId("5a5df70862a72330a4448d1e")	{ 5 fields }	Object
> (9) ObjectId("5a5df70862a72330a4448d1f")	{ 5 fields }	Object

上面主要测试索引的效率。当然要避免"\$nin"，模糊查询等一系列全文档扫描的查询条件，会很影响效率

```
db.getCollection('users').find({"Age":{"$nin:[19,20,21]}}).count()
```

6.979 sec.

8119908

如果就要通过地址字段模糊查询，那样也可以根据地址进行分库，分表的处理，可以根据数据量大热门城市建立user\_beijing,user\_shanghai，user\_hangzhou等处理方案。  
具体就不多说了。

纸上得来终觉浅，绝知此事要躬行。

标签: mongodb, 性能优化, 索引, 批量数据导入

好文要顶

关注我

收藏该文

园友

<天上白玉京>

关注 - 0

粉丝 - 2

+加关注

« 上一篇：C#操作MongoDB的简单实例  
» 下一篇：简单使用redis实现sso单点登录

发表于 2018-01-16 22:11 <天上白玉京> 阅读(224) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。



最新IT新闻：

- 癌症新靶向疗法：化疗药物被包裹在“纳米气泡” 并通过X射线按需释放
  - 法国队世界杯夺冠 华帝在天猫兑现“退全款”承诺
  - 三星让供应商感动到哭：发2000万美元奖金鼓舞士气
  - 途牛"屠牛"：单方面裁员降薪500人背后 3年累亏46亿
  - 以色列有望成第四个将探测器送上月球的国家，拟于年底发射
- » 更多新闻...



最新知识库文章：

- 危害程序员职业生涯的三大观念
  - 断点单步跟踪是一种低效的调试方法
  - 测试 | 让每一粒尘埃有的放矢
  - 从Excel到微服务
  - 如何提升你的能力？给年轻程序员的几条建议
- » 更多知识库文章...