

[博客](#)[学院](#)[下载](#)[GitChat](#)[论坛](#)[问答](#)[商城](#)[写博客](#)[发Chat](#)[登录](#)[注册](#)

chszs的专栏

LQ的小窝

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

MongoDB GridFS最佳应用概述

标签：[mongodb](#) [gridfs](#) [nosql数据库](#) [gridfs-fuse](#) [gridfs-nginx](#)

2014年02月28日 10:21:11

15121人阅读

[评论\(1\)](#)

[收藏](#)

[举报](#)

分类：[NoSQL数据库 \(25 \)](#)

版权声明：本文为博主chszs的原创文章，未经博主允许不得转载。

《MongoDB GridFS最佳应用概述》

作者：chszs，转载需注明。博客主页：<http://blog.csdn.net/chszs>

GridFS是MongoDB数据库之上的一个简单文件系统抽象。如果你熟悉Amazon S3的话，那么GridFS与之相似。为什么像MongoDB这样的NoSQL数据库会提供这样的一个文件层抽象呢？

一、使用GridFS的理由

理由如下：

1) 存储用户产生的文件内容

大多数Web应用都允许用户上传文件。当用户使用关系数据库时，这些用户产生的文件会存储在文件系统中，与数据库相隔离，而不是放在数据库内。这就带来了一些问题。如何将文件复制到所有需要文件的服务器上？当文件删除后，怎样删除所有的拷贝？怎样保障文件的安全以及做灾备呢？GridFS很好地解决了这些问题，你可以利用你的数据库备份来备份你的文件。而且由于MongoDB自身的复制技术，在MongoDB集群中的每一个副本处都有你的文件拷贝。删除文件跟删除数据库中的对象一样简单。

2) 访问文件内容的分区

当把文件上传到GridFS后，文件会被分割成大小为256KB的块，并单独存放。因此当你需要读文件中的某个范围的字节时，只需把相应的文件块载入内存，而无需把整个文件加载到内存。这一点对于选择读或编辑尺寸很大的媒体内容文件时非常有用。

3) 在MongoDB中存储16MB以上的文件

MongoDB默认的文件大小上限为16MB。所以，如果你的文件超过了16MB，那么你就应该使用GridFS。

4) 克服文件系统的限制

如果你需要存储大量的文件，你就需要考虑文件系统自身的限制，因为文件系统对目录下的文件数量是有要求的。而使用GridFS后，你无需再担心这个问题。GridFS和MongoDB的分片使得你的文件可以分布到多个服务器上，而且没有增加操作的复杂性。

二、深入GridFS

GridFS使用了两种集合Collection来存储数据

```
[javascript]
1. > show collections;
2. fs.chunks
3. fs.files
4. system.indexes
5. >
```

fs.files集合包含了文件的元数据，而fs.chunks集合则存储实际的以256KB尺寸进行分割的文件块。如果你有分片的集合，那么文件块会分布到多台服务器上，或许能获得比文件系统更好的性能。

```
[javascript]
1. > db.fs.files.findOne();
2. {
3.   "_id" : ObjectId("530cf1bf96038f5cb6df5f39"),
4.   "filename" : "./conn.log",
5.   "chunkSize" : 262144,
6.   "uploadDate" : ISODate("2014-02-25T19:40:47.321Z"),
7.   "md5" : "6515e95f8bb161f6435b130a0e587ccd",
8.   "length" : 1644981
9. }
10. >
```

MongoDB还在files_id和文件块数中创建了复合索引，以帮助快速访问这些文件块

```
[javascript]
1. > db.fs.chunks.getIndexes();
2. [
3.   {
4.     "v" : 1,
5.     "key" : {
6.       "_id" : 1
7.     },
8.     "ns" : "files.fs.chunks",
9.     "name" : "_id_"
10.  },
11.  {
12.    "v" : 1,
13.    "key" : {
```

```
14.   "files_id" : 1,
15.   "n" : 1
16. },
17. "ns" : "files.fs.chunks",
18. "name" : "files_id_1_n_1"
19. }
20. ]
21. >
```

三、GridFS实例

MongoDB有一个内建的工具mongofiles，可以帮助练习实际使用GridFS的场景。请参阅相关的Driver文档，查看如何使用GridFS。

```
[javascript]
1. Put
2. #mongofiles -h -u -p --db files put /conn.log
3. connected to: 127.0.0.1
4. added file: { _id: ObjectId('530cf1009710ca8fd47d7d5d'), filename: "./conn.log", chunkSize: 262144, uploadDate: new Date(13933570576
5. done!
6.
7. Get
8. #mongofiles -h -u -p --db files get /conn.log
9. connected to: 127.0.0.1
10. done write to: ./conn.log
11.
12. List
13. # mongofiles -h -u -p list
14. connected to: 127.0.0.1
15. /conn.log 1644981
16.
17. Delete
18. [root@ip-10-198-25-43 tmp]# mongofiles -h -u -p --db files delete /conn.log
19. connected to: 127.0.0.1
20. done!
```

四、GridFS的模块

如果你想把存储在MongoDB的GridFS的文件直接服务于Web服务器或文件系统，那么你可以使用下面的GridFS插件：

- 1) GridFS-Fuse：让GridFS的文件直接服务于文件系统
- 2) GridFS-Nginx：让GridFS的文件直接服务于Nginx

五、GridFS的局限性

GridFS也并非十全十美的，它也有一些局限性：

1) 工作集

伴随数据库内容的GridFS文件会显著地搅动MongoDB的内存工作集。如果你不想让GridFS的文件影响到你的内存工作集，那么可以把GridFS的文件存储到不同的MongoDB服务器上。

2) 性能

文件服务性能会慢于从Web服务器或文件系统中提供本地文件服务的性能。但是这个性能的损失换来的是管理上的优势。

3) 原子更新

GridFS没有提供对文件的原子更新方式。如果你需要满足这种需求，那么你需要维护文件的多个版本，并选择正确的版本。

- [上一篇](#) 数据清洗工具OpenRefine
- [下一篇](#) NoSQL世界还会有DBA存在吗