

异步远程数据插入效率问题解决方案：

效率瓶颈：

大量数据进行远程传输，大量的时间花费在数据串行化和web io远程传输中

基本原理：

1. MongoDB的高速备份和恢复机制来处理，利用压缩文件的传输一次性传递多条数据，减少大量数据远程传输

2. C/C++原生（MongoDB提供）数据备份和恢复功能

实现方案：

两个关键函数原生接口 mongodump 和 mongorestore

实现流程：

1. 每一个结点插入本机的数据库实例中，（这里要求一个Node对应一个数据库实例，强调，每一个Node对应一个）

2. mongodump导出到一个文件，压缩这个文件

3. 多机器scp传输到一台机器（主数据库实例发布的机器）

4. 解压这个文件，mongorestore到指定数据库

5. 接数据库去重复的操作。

例子：这个工程就是简单的例子，你们将这个弄到现在的工程中，记得先起一个新的分支，不影响现阶段的测试。

这个例子的运行是这样的，在本机创建两个数据库（1和2）

将1和2导出，然后合并到3中。

如下图所示。

```
> show dbs
baby                0.078GB
db_prefromance_test1 0.078GB
db_prefromance_test2 0.078GB
db_prefromance_test3 0.078GB
local               0.031GB
> use db_prefromance_test3
switched to db db_prefromance_test3
> db.prefromance.find().pretty()
{ "_id" : ObjectId("58b67b52b03db3c925b1977f"), "test" : "yangyuan" }
{ "_id" : ObjectId("58b689cbb03db3c925b19781"), "test" : "sunfei" }
> use db_prefromance_test1
switched to db db_prefromance_test1
> db.prefromance.find().pretty()
{ "_id" : ObjectId("58b67b52b03db3c925b1977f"), "test" : "yangyuan" }
> use db_prefromance_test2
switched to db db_prefromance_test2
> db.prefromance.find().pretty()
{ "_id" : ObjectId("58b689cbb03db3c925b19781"), "test" : "sunfei" }
>
```

