MongoDB 基础教程

第一部分

翻译原因

网上的相关说明别说中文了连英文都很少，而且有的基本上就是一些简单的配置，加上最近时间比较多，有点想做，所以就做了。

致谢

先感谢国家，谢谢国家对我的培养，再感谢组织，感谢我的公司给我了研究 MongoDB 的机会，让我有时间去研究了 MongoDB，最后感谢父母的养育之恩。

温馨提示

我以下所有示例都是建立在 Java 语言的基础上，只作 java 语言的介绍，其他语言类似。

概述

MongoDB 是一个基于分布式文件存储的数据库。由 C++语言编写。旨在为 WEB 应用提供可护展的高性能数据存储解决方案。

它的特点是高性能、易部署、易使用，存储数据非常方便。主要功能特性有：

*面向集合存储，易存储对象类型的数据。

*模式自由。

*支持动态查询。

*支持完全索引，包含内部对象。

*支持查询。

*支持复制和故障恢复。

*使用高效的二进制数据存储，包括大型对象（如视频等）。

*自动处理碎片，以支持云计算层次的扩展性

*支持 RUBY，PYTHON，JAVA，C++，PHP 等多种语言。

*文件存储格式为 BSON（一种 JSON 的扩展）

*可通过网络访问

所谓“面向集合”（Collenction-Orented），意思是数据被分组存储在数据集中，被称为一个集合（Collenction)。每个集合在数据库中都有一个唯一的标识名，并且可以包含无限数目的文档。集合的概念类似关系型数据库（RDBMS）里的表（table），不同的是它不需要定义任何模式（schema)。

模式自由（schema-free)，意味着对于存储在 mongodb 数据库中的文件，我们不需要知道它的任何结构定义。如果需要的话，你完全可以把不同结构的文件存储在同一个数据库里。

存储在集合中的文档，被存储为键-值对的形式。键用于唯一标识一个文档，为字符串类型，而值则可以是各中复杂的文件类型。我们称这种存储形式为 BSON（Binary Serialized dOcument Format）。

MongoDB 服务端可运行在 Linux、Windows 或 OS X 平台，支持 32 位和 64 位应用，默认端口为 27017。推荐运行在 64 位平台，因为 MongoDB 在 32 位模式运行时支持的最大文件尺寸为 2GB。

注意：以上概述部分摘自百度百科 http://baike.baidu.com/view/3385614.htm

第二部分

基础知识（重点）

快速入门

快速入门 Unix 系统

安装 MongoDB

先根据自己的系统下载自己需要的版本，然后解压即可

32 位 Linux

```
$ curl http://downloads.mongodb.org/linux/mongodb-linux-i686-1.4.3.tgz > mongo.tgz
$ tar xzf mongo.tgz
```

64 位 Linux

```
$ curl http://downloads.mongodb.org/linux/mongodb-linux-x86_64-1.4.3.tgz > mongo.tgz
$ tar xzf mongo.tgz
```

其他版本去下载页面 http://www.mongodb.org/display/DOCS/Downloads 找与之对应的版本安装即可。

下载完了之后解压。

创建数据存放目录。

默认情况下 MongoDB 的数据存放目录是 /data/db，但是 MongoDB 不会自己去创建目录，所以需要你手工创建：

```
$ sudo mkdir /data/db/
```

你也可以使用不同的路径，在启动的时候知道路径参数—dbpath 即可。

启动 MongoDB

```
$ ./mongodb-xxxxxxx/bin/mongod
```

连接 MongoDB

```
$ ./mongodb-xxxxxxx/bin/mongo
> db.foo.save( { a : 1 } )
> db.foo.find()
```

恭喜你，到此为止，你的第一条数据已经保存进了 Linux 系统的 MongoDB 了。

快速入门 windows 系统

下载，安装同 Linux 的下载，安装，根据自己需要的版本去下载页面下载，下载完了之后解压，在这里不多罗嗦了。

创建数据目录

在 windows 下的默认数据目录是 C:\data\db，但是 MongoDB 同样不会自己去创建该目录，需要自己的去创建，当然通 Linux 系统一下，也可以指定别的目录。在启动的时候告诉他即可。

启动 MongoDB

```
mongodb-xxxxxxx\bin\mongod.exe
```

连接 MongoDB

```
mongodb-xxxxxxx\bin\mongo.exe
> db.foo.save( { a : 1 } )
> db.foo.find()
```

恭喜你，到此为止，你的第一条数据已经保存进了 windows 系统的 MongoDB 了。

MongoDB 的下载

http://www.mongodb.org/display/DOCS/Downloads

MongoDB 驱动下载

Java 驱动下载地址 http://github.com/mongodb/mongo-java-driver/downloads

PS：我正在写这个的时候 MongoDB 的最高版本是 1.3

对应的 API 地址：http://api.mongodb.org/java/2.0/index.html

小试牛刀

Java 的驱动提供了一个 DBObject 接口，保存到数据库的对象需要实现该接口。请看下面这个例子：

```
public class Tweet implements DBObject {
    /* ... */
}
```

然后你的保存可以这么写：

```
Tweet myTweet = new Tweet();
myTweet.put("user", userId);
myTweet.put("message", msg);
myTweet.put("date", new Date());

collection.insert(myTweet);
```

当一个文件从数据库检索出来，他自动转换成 DBObject，为了转换成你所需要的类型，使用 DBCollection.setObjectClass()：

```
collection.setObjectClass(Tweet);

Tweet myTweet = (Tweet)collection.findOne();
```

PS：可能很多朋友测试不成功，不知道这个 collection 怎么来的，在这里你可以这样来获得 collection：

```
DBAddress dba = new DBAddress("localhost", 27017, "TestDB");
DB db = Mongo.connect(dba);
DBCollectioncollection = db.getCollection("testcollection");
```

具体是为什么，后面会具体说到。

MongoDB 的使用

MongoDB 的启动

//在这里主要说在 Linux 的下的问题，windows 差不多，但不说了

普通启动

(1) 默认启动 bin/mongod
(2) 指定端口和数据目录 bin/mongod –dbpath=/var/data/ –port=556600
(3) 带有鉴权的启动 bin/mongod –auth

分布式配置

您需要启动的至少两个 MongoDB 文档数据库。

服务器 1：192.168.1.10/Linux1
服务器 2：192.168.1.11 /Linux2

启动服务器 1：

bin/mongod –slave –source=192.168.1.11:556600 –dbpath=/var/db/ –port=556611 –slavedelay 10 &

启动服务器 2：

bin/mongod –slave –source=192.168.1.10:556601 –dbpath=/var/db/ –port=556610 –slavedelay 10 &

PS:这里的意思是启动该服务器 1，并且把服务器 2 作为主服务器，每 10 秒与主服务器 2 同步一次。启动服务器 2 的意思是，把服务器 1 作为主服务器，每 10 秒与服务器 1 同步一次。这样服务器 1 与服务器 2 就构成了同步了。
在这里我曾经在公司测试过，是可以的，但是我现在是放假在家里，不知道其中是否有错误，你自己鉴别。

MongoDB 在 Java 中的应用

建立连接

要建立 MongoDB 的连接，你只要指定要连接到的数据库就可以。这个数据库不一定存在，如果不存在，MongoDB 会先为你建立这个库。同时，在连接时你也可以具体指定要连接到的网络地址和端口。(以下示例连接到 Linux 192.168.1.10)

```
//创建一个连接地址
DBAddress dba = new DBAddress("192.168.1.10", 27017, "mydb1");
```

```
//获得数据库
DB db = Mongo.connect(dba);

//鉴权(当鉴权启动时是必须的),成功返回 true，否则返回 false
boolean auth = db.authenticate("username","passwd".toCharArray());
```

基本操作

获取集合列表(*ps:你可以把集合理解成关系型数据库中的表*)

```
Set<String> colls = db.getCollectionNames();
for(String s : colls){
  System.out.println(s);
}
```

获得一个集合,其中 testCollection 可以是集合列表中的一个，也可以是一个新的集合名字，当该集合不存在时自动创建该集合，获得之后即和对该集合进行增删查改

```
DBCollection    coll = db.getCollection("testCollection");
```

插入文档(插入的文档需要是 json 格式的，类似于 key-value，每个键值对之间用逗号分隔)

Eg:

```
{
  "name" : "MongoDB",
  "type" : "database",
  "count" : 1,
  "info" : {
     x : 203,
     y : 102
    }
}
```

代码实现：

```
BasicDBObject doc = new BasicDBObject();
doc.put("name", "MongoDB");
doc.put("type", "database");
doc.put("count", 1);

BasicDBObject info = new BasicDBObject();
info.put("x", 203);
info.put("y", 102);
doc.put("info", info);

coll.insert(doc);
```

获得文档总数

```
Long i = coll.getCount();
```

查找文档(查找第一个文档：findOne()；查找所有文档：find()；根据条件查询文档：find(DBObject))

findOne()的用法:

```
DBObject obj = coll.findOne();
System.out.println(obj);
```

find()的用法:

```
DBCursor cur = coll.find();
    while(cur.hasNext()){
        System.out.println(cur.next());
    }
```

find(DBOject)用法:

```
//查询 count = 1 ， name= “MongoDB”的数据
BasicDBObject query = new BasicDBObject();
query.put("count", 1);
query.put("name ", “MongoDB”);
DBCursor cur = coll.find(query);
System.out.println(cur.count());    //查询结果个数
while(cur.hasNext()){
 System.out.println(cur.next());
}

//引申：根据 _id 查询
BasicDBObject query = new BasicDBObject();
query.put("_id", new ObjectId("4bb29ce82075bcce6102573d"));
DBCursor cur = coll.find(query);
System.out.println(cur.count());    //查询结果个数
while(cur.hasNext()){
 System.out.println(cur.next());
}
```

模糊查询

MongoDB 支持正则表达式查询，可以代替关系数据库中的模糊查询。由于文档中写代码格式不是很好看，所以不在这里写了。具体请查看下下来的代码文件。

删除数据

```
coll.remove(DBObject);
```

创建索引

MongoDB 支持索引，而且很容易在集合上增加索引。要创建索引，只需要指定要加索引的属性，并且指定升序（1）或降序即可（-1）。

```
coll.createIndex(new BasicDBObject("i", 1));
```

获取索引列表

```
List<DBObject> list = coll.getIndexInfo();
for(DBObject o : list){
  System.out.println(o);
}
```

MongoDB 连接池

任何一种数据在做数据库连接的时候都是非常耗时间的，所以需要连接池，MongoDB 也不例外，目前我还没有发现 MongoDB 的第三方连接池，所以我自己写了一个很烂的连接池，放在了代码里面。写的很烂，请别笑话！

第三部分

补充说明

由于在 word 中写代码我个人感觉很别扭，而且很难写好，所以代码的例子就写这么几个，具体的代码请查看一起下载下来的代码文件，同时欢迎加入 NOSQL 交流群 67472265 一起讨论交流。我的英语不是很好，加上我对 MongoDB 刚接触，所以错误是再说难免的，如果在用的过程中有什么问题，发现有什么错误，我的邮件地址是：xcwang@vip.qq.com，期待收到你的邮件。谢谢！

免责申明

我所写的不一定是正确，我所写的只是一些常用的部分，如果你按照我所写的去做，遇到一切风险，我不承担任何责任。