

NoSQL数据库实战

BestTest

课程大纲

- 非关系型数据库简介
- Mongodb实战
- Redis实战
- Memcache实战

非关系型数据库

- 非关系型数据库也叫Nosql数据库，全称是not noly sql。
- 2009年初，Johan Oskarsson举办了一场关于开源分布式数据库的讨论，Eric Evans在这次讨论中提出了NoSQL一词，用于指代那些非关系型的，分布式的，且一般不保证遵循ACID原则的数据存储系统。Eric Evans使用NoSQL这个词，并不是因为字面上的“没有SQL”的意思，他只是觉得很多经典的关系型数据库名字都叫“**SQL”，所以为了表示跟这些关系型数据库在定位上的截然不同，就是用了“NoSQL”一词。
- 非关系型数据库提出另一种理念，例如，以键值对存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，这样就不会局限于固定的结构，可以减少一些时间和空间的开销。使用这种方式，用户可以根据需要去添加自己需要的字段，这样，为了获取用户的不同信息，不需要像关系型数据库中，要对多表进行关联查询。仅需要根据id取出相应的value就可以完成查询。

关系型数据库与非关系型数据库的区别

- 关系型数据库通过外键关联来建立表与表之间的关系，非关系型数据库通常指数据以对象的形式存储在数据库中，而对象之间的关系通过每个对象自身的属性来决定。

nosql数据库的特点

- 模式自由
 - 不需要定义表结构，数据表中的每条记录都可能有不同的属性和格式。
- 逆规范化
 - 不遵循范式要求，去掉完整性约束，减少表之间的依赖
- 弹性可扩展
 - 可在系统运行的过程中，动态的删除和增加节点。
- 多副本异步复制
 - 数据快速写入一个节点，其余节点通过读取写入的日志来实现异步复制。
- 弱事务
 - 不能完全满足事务的ACID特性，但是可以保证事务的最终一致性。

什么时候用nosql数据库

- 数据库表schema经常变化
- 数据库表字段是复杂数据类型
- 高并发数据库请求
- 海量数据的分布式存储

课程大纲

- 非关系型数据库简介
- **Mongodb实战**
- Redis实战
- Memcache实战

Mongodb实战

- **Mongodb简介**
- Mongodb安装
- Mongodb增删改查
- Mongodb监控

Mongodb简介

- MongoDB.inc 公司研发的一款nosql类型的文档型数据库。
- 功能强大、使用灵活、性能卓越且易于扩展的数据库。
- 2009年发布第一个稳定版本，是当前非常热门的开源的nosql类型数据库。
- 官方网站：<https://www.mongodb.org>
- 开源项目：<http://github.com/mongodb>

Mongodb特点

- 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

Mongodb安装

- 1、下载mongodb安装包，
- 2、在/usr/local目录下创建一个mongodb文件夹，然后把mongodb的安装包解压到mongodb文件中
- 3、创建一个存放数据的文件夹data和日志文件logs
 - `cd /usr/local/mongodb`
 - `mkdir data`
 - `touch logs`
- 4、启动mongodb
 - `/usr/local/mongodb/bin/mongod --dbpath=/usr/local/mongodb/data --logpath=/usr/local/mongodb/logs --logappend --port=27017 --fork`

Mongodb启动参数和启动脚本

- 使用Mongodb安装目录下的mongod文件来启动mongodb
- 常用的启动参数：
 - dbpath的意思是指定存储数据的文件夹
 - logpath的意思是指定日志存储文件
 - logappend的意思是日志以增加方式产生
 - port指定端口，如果不写的话，默认是27017
 - fork 代表后台运行
- 也可以把这些参数都写到一个配置文件中，然后读取配置文件

Mongodb启动脚本

- 启动脚本:

```
#!/bin/bash
pid=`ps -ef|grep /usr/local/mongodb/bin/mongod|grep -v "grep" |`
awk ' {print $2}' `
if [[ $pid ]]
then
echo "mongodb is running..."
else
/usr/local/mongodb/bin/mongod --dbpath=/usr/local/mongodb/data --
logpath=/usr/local/mongodb/logs --logappend --port=27017 --fork
echo 'mongodb started!'
fi
```


停止mongodb和停止脚本

- Mongodb停止的时候不能直接kill掉，如果kill掉的话，下次启动会有问题
- 停止mongodb的时候要用mongod 后面加上shutdown参数，并且指定数据库文件
- 停止脚本

```
#!/bin/bash
pid=`ps -ef|grep /usr/local/mongodb/bin/mongod |grep -v
"grep" | awk '{print $2}'`
if [[ $pid ]]
then
/usr/local/mongodb/bin/mongod --shutdown --dbpath
/usr/local/mongodb/data/
echo 'mongodb stoped'
else
echo "mongodb is not running..."
fi
```

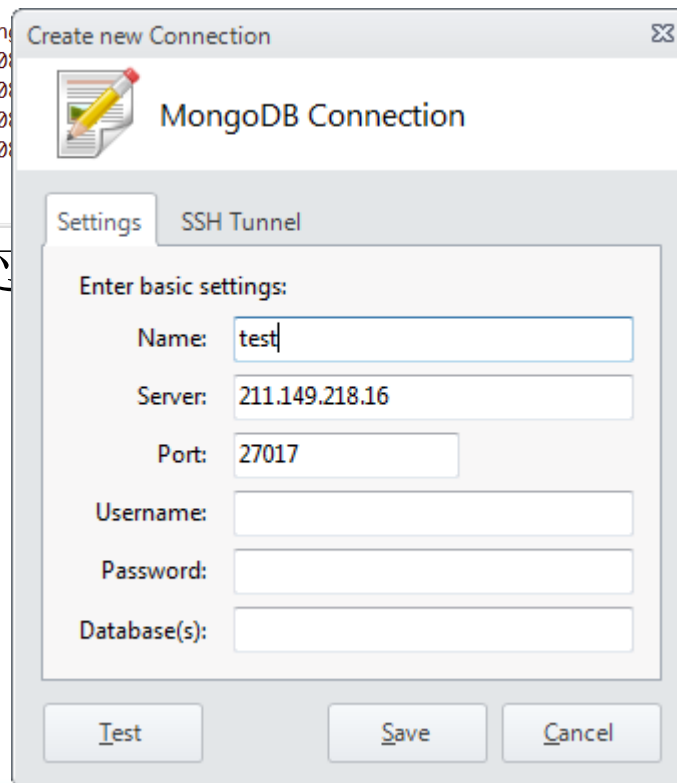
重启mongodb和重启脚本

- 重启的过程就是先停止，后启动，我们已经写好了启动的脚本和停止的脚本，依次调用即可
- `#!/bin/bash`
- `/usr/local/mongodb/stop_mongodb.sh`
- `/usr/local/mongodb/start_mongodb.sh`

连接mongodb

- Linux下可以直接使用mongo进入mongod命令行操作

```
mongod --help ...  
[root@ebs-32534 mongod]# mongo  
MongoDB shell version: 2.6.7  
connecting to: test  
Server has startup warning:  
2016-01-03T01:59:02.452+0800  
2016-01-03T01:59:02.452+0800  
2016-01-03T01:59:02.452+0800  
2016-01-03T01:59:02.452+0800  
>
```



Create new Connection

MongoDB Connection

Settings SSH Tunnel

Enter basic settings:

Name: test

Server: 211.149.218.16

Port: 27017

Username:

Password:

Database(s):

Test Save Cancel

/mongodb/data is set to 4096KB
b 256KB (512 sectors) or less
g/core/readahead

- 也可以通过图形化工具来连接

Mongodb增删改查

- 概念： 集合==表 文档==数据
- 命令关键词：
 - Show dbs: 查看数据库
 - Show collectios: 查看集合
 - Create collection: 创建一个集合
 - use: 切换数据库
 - insert: 插入数据
 - Find: 查找数据
 - Update: 修改数据
 - Remove: 删除数据

Mongodb创建和删除数据库

- Mongodb中使用use关键字来创建一个数据库
- use besttest;#创建了一个数据库
- db;#查看当前的数据库
- db.dropDatabase();

插入数据

- Mongodb中使用insert关键字来插入数据
`db.collectios.insert(doc);`
- 创建一个besttest库，插入一条班级信息
- `use besttest;`
- `db.classes.insert({"name":"乔巴", "nums":100, "course":["mysql", "linux", "性能测试项目实战", "性能调优", "安全测试"], "teacher":"安大叔"});`
- 也可以通过定义变量的方式来插入数据
- `new_class=({"name":"索隆", "nums":150, "course":["mysql', 'nosql', 'linux', '前端性能测试'], "teacher":"andashu"});`
- `db.classes.insert(new_class);`

更新数据

- Mongodb中使用update关键字来更新数据, `db.collection.update({条件},{更新的值})`
- 把刚才插入的乔巴班的信息加一个状态, `status`, 上课中
- `use besttest;`
- 所有字段加全
 - `db.classes.update({"name":"乔巴"}, {"name":"乔巴", "nums":100, "course":["mysql","linux","性能测试项目实战","性能调优","安全测试"], "teacher":"安大叔", "status":"上课中"});`
- `$set`方式只更新指定的字段
 - `db.classes.update({"name":"乔巴"}, {$set:{"createtime":"20150810"}});`
- 更新所有匹配的值
 - `db.classes.update({"teacher":"andashu"}, {$set:{"teacher":"niuhy"}}, {multi:true});`
- 如果更新的值不存在的话, 插入一条
 - `db.classes.update({"teacher":"fenglou"}, {$set:{"teacher":"niuhy"}}, {upsert:true});`
- `$inc`, 在原来的值上增加值, 只适用于数字型
 - `db.classes.update({"name":"乔巴"}, {$inc:{"nums":20}});`

查询数据

- Mongodb中使用find关键字来查询数据,`db.collection.find()`;
- `db.collection.find()`;查询所有的数据
- `db.collection.find().pretty()`;已格式化的方式显示数据
- `db.collection.find({"xx":"xx"})`;`#`指定条件查询
- `db.collection.find({条件1},{条件2})``#`and操作
- `db.collection.find({$or,[{条件1},{条件2}]})``#`or操作
- `db.collection.find({条件1,$or[{条件2},{条件3}]})`;`#`and和or合用
- `db.collection.find().count()`;`#`查询所有的行数
- `db.collection.find().sort(KEY:1)``#`排序

Mongodb中的条件表达式

操作	格式	范例	RDBMS中的类似语句
等于	{<key>:<value>}	db.classes.find({"teacher": "安大叔"}).pretty()	where teacher = '安大叔'
小于	{<key>:{<\$lt:<value>}}	db.classes.find({"nums":{\$lt:50}}).pretty()	where nums < 50
小于或等于	{<key>:{<\$lte:<value>}}	db.classes.find({"nums":{\$lte:50}}).pretty()	where nums <= 50
大于	{<key>:{<\$gt:<value>}}	db.classes.find({"nums":{\$gt:50}}).pretty()	where nums > 50
大于或等于	{<key>:{<\$gte:<value>}}	db.classes.find({"nums":{\$gte:50}}).pretty()	where nums >= 50
不等于	{<key>:{<\$ne:<value>}}	db.classes.find({"nums":{\$ne:50}}).pretty()	where nums != 50

删除数据

- Mongodb中使用remove关键字来删除数据
`,db.collectios.remove();`
- `db.collectios.remove({});`删除所有的数据
- `db.collections.remove({条件1},1)`#如果有多条匹配的，只删除一条

Redis实战

- Redis简介
- Redis安装
- Redis增删改查
- Redis监控

Redis简介

- redis是一个开源的、使用C语言编写的、支持网络交互的、可基于内存也可持久化的Key-Value数据库。
- redis的官网地址，非常好记，是redis.io。
- Redis和mongodb的区别是redis数据全部存储在内存中，使用磁盘仅用于数据的持久化，而mongodb数据是存储在磁盘上。

Redis安装

- 1、下载redis安装包

```
wget http://download.redis.io/releases/redis-2.8.12.tar.gz
```

- 2、安装redis依赖

```
yum -y install gcc*  
yum -y install make  
yum -y install tcl
```

- 3、编译安装

```
tar xvf redis-2.8.12.tar.gz  
mv redis-2.8.12 redis #改名  
mv redis /usr/local #移动  
cd /usr/local/redis  
make
```

- 4、使用redis的可执行命令

将redis命令移动到/usr/local/bin目录下，把redis命令放到/usr/local/bin目录下后就可以直接使用redis命令，如果不放到/usr/local/bin也是可以的，直接到/usr/local/redis/src目录下使用即可。

```
cd /usr/local/redis/src  
cp -rf redis-benchmark redis-server redis-cli redis-check-  
dump redis-check-aof /usr/local/bin
```

Redis启动

- 启动redis

- 介绍两种启动方式，一种是命令启动，一种是使用配置文件启动。
- `redis-server &` #这种是命令启动，默认端口6379，&表示后台运行
- `redis-server /usr/local/redis/redis.conf` #配置文件启动的，可以在配置文件中修改端口
- 由于redis默认带的配置文件不是后台运行的，咱们不用它，新建一个`redis.conf`配置文件，把原来那个默认的删掉，新的配置文件内容如下：

```
daemonize  yes #代表后台启动
pidfile    /usr/local/redis/redis.pid #pid文件
logfile    /usr/local/redis/redis.log #日志文件
port       6379#端口
dir        /usr/local/redis/redisData #持久化文件存放位置
```

Redis启动脚本

- 为了方便管理，现在写一个redis的启动脚本和停止脚本
- 启动脚本：

```
#!/bin/bash
pid=`ps -ef|grep redis-server|grep -v "grep"|awk '{print $2}'`
if [[ "$pid" ]]
then
echo "redis is running...."
else
/usr/local/bin/redis-server /usr/local/redis/redis.conf
echo "redis started! "
```


Redis停止脚本

- 停止脚本:

- `#!/bin/bash`
- `pid=`cat /usr/local/redis/redis.pid``
- `new_pid=`ps -ef|grep $pid|grep -v "grep"|awk '{print $2}'``
- `if [[$new_pid]]`
- `then`
- `kill -9 $pid`
- `echo "redis is stoped!"`
- `else`
- `echo "redis is not running..."`
- `fi`

Redis重启脚本

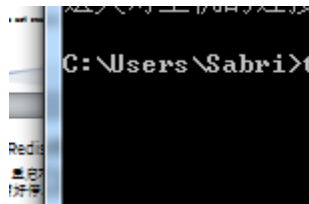
- 重启脚本，重启不过就是先停止，后启动，我们已经写好停止和启动的脚本，依次调用即可：
 - `#!/bin/bash`
 - `/usr/local/redis/stop_redis`
 - `/usr/local/redis/start_redis`

连接redis

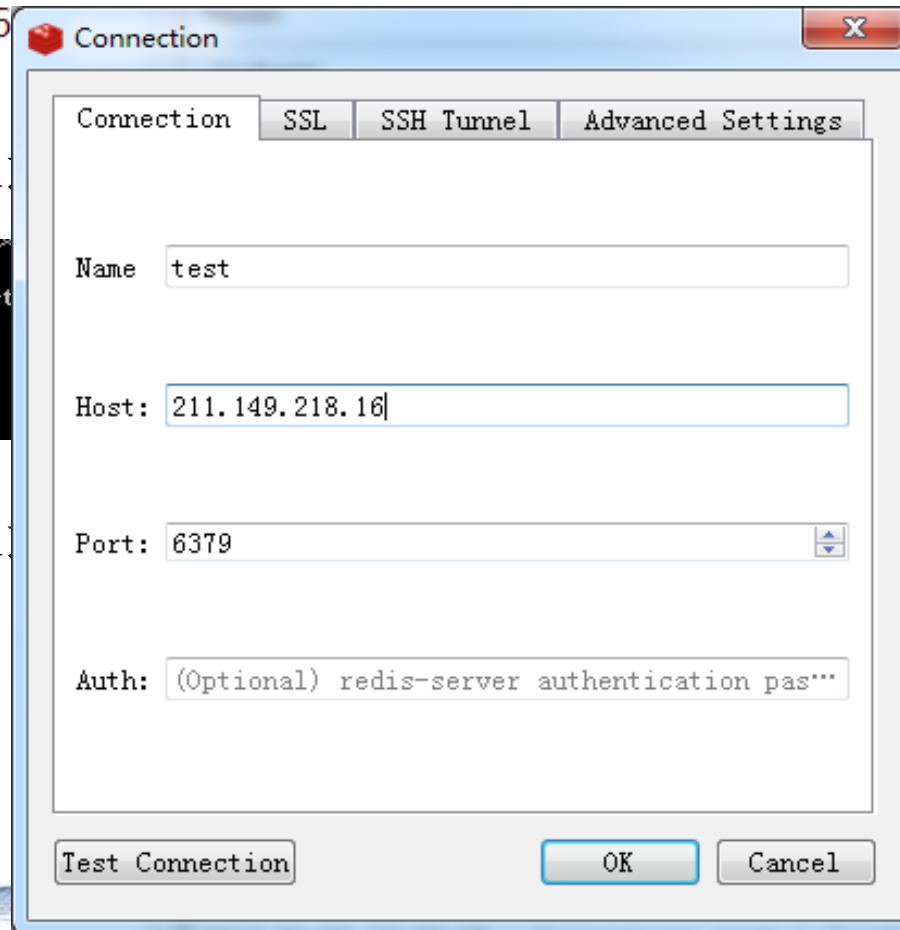
- Linux下可以直接使用redis-cli进入redis命令行操作

```
[root@ebs-32534 memcached]# redis-cli  
127.0.0.1:6379
```

- 也可以通



- 也可以通

A screenshot of the 'Connection' dialog box in Redis Desktop Manager. The dialog has tabs for 'Connection', 'SSL', 'SSH Tunnel', and 'Advanced Settings'. The 'Connection' tab is active, showing fields for 'Name' (test), 'Host' (211.149.218.16), 'Port' (6379), and 'Auth' ((Optional) redis-server authentication pas...). At the bottom are buttons for 'Test Connection', 'OK', and 'Cancel'.

Connection	SSL	SSH Tunnel	Advanced Settings
Name	test		
Host:	211.149.218.16		
Port:	6379		
Auth:	(Optional) redis-server authentication pas...		

来连接

Redis的增删改查

- 选择数据库，使用select关键字
 - select 1, 选择第一个数据库
- 使用set关键字插入数据
 - set name andashu
- 使用get 关键字获取数据
 - get name
- 使用del 关键字 删除数据
 - del name

Memcached实战

- Memcached简介
- Memcached安装
- Memcached增删改查
- Memcached监控

Memcached简介

- Memcached是一个高性能的分布式的内存对象缓存系统，目前全世界不少人使用这个缓存项目来构建自己大负载的网站，来分担数据库的压力，通过在内存里维护一个统一的巨大的hash表，它能够用来存储各种格式的数据，包括图像、视频、文件以及数据库检索的结果等。简单的说就是将数据调用到内存中，然后从内存中读取，从而大大提高读取速度。

Memcached安装

- 1、安装依赖包libevent

```
yum -y install libevent
```

- 2、编译安装

```
tar xvf memcached-1.2.0.tar.gz
```

```
cd memcached-1.2.0
```

```
./configure --prefix=/usr/local/memcached
```

```
make
```

```
make install
```

Memcached启动参数

- -p 监听的端口
- -c 最大同时连接数，默认是1024
- -m 最大内存使用，单位MB。默认64MB
- -P 设置保存Memcache的pid文件
- -d 后台运行
- -u 运行Memcache的用户，仅在以root运行的时候有效

Memcached启动脚本

- `#!/bin/bash`
- `pid=`ps -ef|grep memcached|grep -v "grep"|awk ' {print $2}``
- `if [[$pid]]`
- `then`
- `echo "memcached is running..."`
- `else`
- `/usr/local/bin/memcached -d -m 10 -p 11211 -u root -c 256 -P`
`/tmp/memcached.pid >> /tmp/memcached.log`
- `echo 'memcached started pidfile path is /tmp/memcached.pid'`
- `fi`

Memcached停止脚本

- `#!/bin/bash`
- `pid=`cat /tmp/memcached.pid``
- `new_pid=`ps -ef|grep $pid|grep -v "grep"|awk ' {print $2}'``
- `if [[$new_pid]]`
- `then`
- `kill -9 `cat /tmp/memcached.pid``
- `echo "memcached stoped..."`
- `else`
- `echo "memcached is not runing...."`
- `fi`

Memcached重启脚本

- 重启脚本也和前面的一样，先停止再启动
- `#!/bin/bash`
- `/usr/local/memcached/stop_mem.sh`
- `/usr/local/memcached/start_mem.sh`

Memcached连接方式

- Memcached没有可视化工具，只能通过Telnet这种方式来连接
- telnet ip 端口
- telnet 211.149.218.16 11211
- telnet上去之后，可以输入stats

```
C:\Users\Sabri>telnet 211.149.218.16 11211
```

```
Telnet 211.149.218.16
ERROR
stats
STAT pid 32588
STAT uptime 131
STAT time 1451756835
STAT version 1.2.0
STAT pointer_size 64
STAT rusage_user 0.012943
STAT rusage_system 0.021179
STAT curr_items 0
STAT total_items 0
STAT bytes 0
STAT curr_connections 2
STAT total_connections 3
STAT connection_structures 3
STAT cmd_get 0
STAT cmd_set 0
STAT get_hits 0
STAT get_misses 0
STAT bytes_read 13
STAT bytes_written 7
STAT limit_maxbytes 10485760
END
```


Memcached的增删改查

- 增加使用add 关键字
 - add key 0 存放时间 数据大小
 - add name 0 30 5
- 查询使用get 关键字
 - get key
 - get name
- 修改使用set或者replace关键字，set和replace的区别是set一个不存在的key时，会新增，replace一个不存在key时，会报错。
 - set key 0 存放时间 数据大小
 - replace key 0 存放时间 数据大小
 - set name1 0 50 6
 - replace name 0 70 5
- 删除使用delete关键字
 - delete key
 - delete name1