

海涛的CSDN博客

当处在低谷时一定要清醒，要忍耐，要淡定。这是生活给你一个难得的自省机会，利用这些时间多学习、暗地里提高自己，为即将到来的高峰做准备！

目录视图


摘要视图

RSS 订阅

联系方式

欢迎讨论技术问题，请通过邮件联系。我的邮箱：  
719343891@qq.com

个人资料



海涛 zht666

访问：852165次

积分：7386

等级：

BLOG 6

排名：第1460名

原创：63篇

转载：64篇

译文：2篇

评论：172条

文章搜索

- 文章分类
- MyBatis (6)

Spring (9)

C# (16)

Java EE (50)

C/C++ (2)

Oracle (7)

MySQL (12)

OpenGL/DirectX (0)

算法学习 (1)

设计模式 (1)

HTML5 (16)

Javascript (3)

jQuery Mobile (2)

jQuery (1)

学院APP首次下载，可得50C币！

CSDN网站系统升级公告

免费学技术，就是如此任性

CSDN诚征代码英才

Nginx+Tomcat7+Mencached负载均衡集群部署笔记

标签：Nginx Memcached Tomcat7 集群部署 负载均衡

2014-08-12 15:00

7347人阅读

评论(10)

收藏

举报

分类：CentOS/Linux (1)

nginx (4)

MySQL (11)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

Nginx+Tomcat+Memcached负载均衡集群服务搭建

操作系统：CentOS6.5

本文档主要讲解，如何在CentOS6.5下搭建Nginx+Tomcat+Memcached负载均衡集群服务器，Nginx负责负载均衡，Tomcat负责实际服务，Memcached负责同步Tomcat的Session，达到Session共享的目的。

**本文档下载地址：**<http://download.csdn.net/detail/zht666/7744157>  
**Tomcat与Memcached配置官方文档：**<https://code.google.com/p/memcached-session-manager/wiki/SetupAndConfiguration>

**注意文档中修正Tomcat配置部分：**  
打开server.xml文件，在<Engine>节点的中添加jvmRoute="tomcatRoute1"，这个是为了指定Tomcat集群的路由。Tomcat2设置成jvmRoute="tomcatRoute2"，Tomcat3设置成jvmRoute="tomcatRoute3"，依次类推。

1.安装Nginx

Nginx官网：<http://nginx.org/>  
下载最新稳定版本。在安装Nginx之前，需要先安装gcc、openssl、pcre和zlib软件库。

1.1安装gcc、gcc-c++  
安装命令：  
#sudo yum install gcc  
# sudo yum install gcc-c++

1.2安装openssl  
openssl官网：<http://www.openssl.org/>  
安装版本：openssl-1.0.1i.tar.gz  
安装命令：  
#tar -zxvf openssl-1.0.1i.tar.gz  
#cd openssl-1.0.1i  
#sudo ./config --prefix=/usr/local/openssl-1.0.1i #prefix指定安装目录  
#sudo make  
#sudo make install

【注意】：此处使用的是config命令，而不是平常的configure命令  
安装完成后，到usr/local/下查看是否安装成功。如果安装出错，需要重新解压缩，重新安装。

1.3安装pcre

心情与感悟 (3)

Ubuntu/Linux (18)

Struts (7)

DB2 (2)

编程感悟 (5)

nginx (5)

CentOS/Linux (2)

Android (2)

Maven (2)

SQL Server (1)

Android (0)

Eclipse (2)

数据库 (5)

jQuery EasyUI (0)

Windows特殊命令 (1)

IntelliJ IDEA (1)

IDEA (0)

文章存档

2015年10月 (1)

2015年08月 (1)

2015年06月 (2)

2015年05月 (1)

2015年04月 (1)

展开

阅读排行

IntelliJ IDEA 12创建Mav (73260)

移动设备、手机浏览器Ja (57254)

Spring MVC返回json数据 (43381)

Eclipse搭建Android开发 (38737)

MyBatis官方下载地址 (37847)

Java中@Override的作用 (36429)

jQuery Mobile 手动显示 (29378)

jQuery Mobile动态刷新 (27804)

Spring3.0.6定时任务task (25548)

解决Failed to load class (23964)

评论排行

IntelliJ IDEA 12创建Mav (30)

Java中@Override的作用 (12)

Eclipse搭建Android开发 (11)

mybatis连接mysql数据库 (11)

Nginx+Tomcat7+Menca (10)

移动设备、手机浏览器Ja (9)

ASP.NET MVC4使用Htr (8)

IntelliJ IDEA 12调试窗口 (6)

Html5饼图绘制 (统计图 (5)

Ubuntu下MySQL中文乱 (4)

推荐文章

\*没有躲过的坑--正则表达式截取字符串

\*CardView完全解析与RecyclerView结合使用(三十二)

\*And roid 高仿微信发朋友圈浏览图片效果

pcre官网: <http://www.pcre.org/>

安装版本: pcre-8.35.tar.gz

安装命令:

```
#tar -zxvf pcre-8.35.tar.gz
#cd pcre-8.35
#sudo ./configure --prefix=/usr/local/pcre-8.35 #prefix指定安装目录
#sudo make
#sudo make install
```

安装完成后, 到/usr/local/下查看是否安装成功。如果安装出错, 需要重新加压缩, 重新安装。

【注意】: 如果没有安装c++编译器, 这个软件的安装会报错!

1.4安装zlib

zlib官网: <http://www.zlib.net/>

安装版本: zlib-1.2.8.tar.gz

安装命令:

```
#tar -zxvf zlib-1.2.8.tar.gz
#cd zlib-1.2.8
#sudo ./configure --prefix=/usr/local/zlib-1.2.8 #prefix指定安装目录
#sudo make
#sudo make install
```

安装完成后, 到/usr/local/下查看是否安装成功。如果安装出错, 需要重新加压缩, 重新安装。

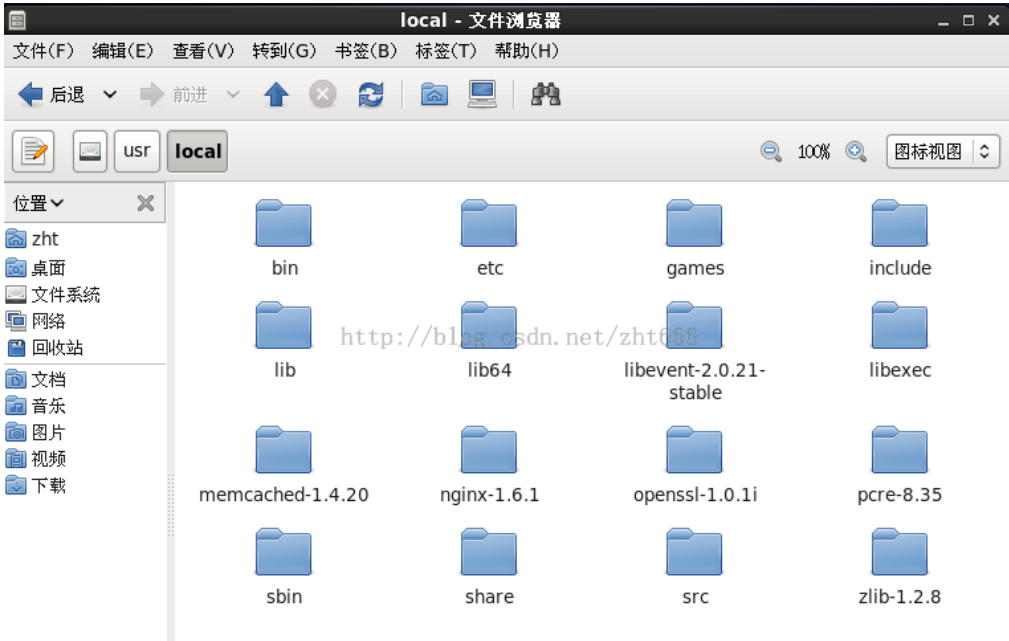
1.5安装Nginx

安装版本: nginx-1.6.1.tar.gz

安装命令:

```
#tar -zxvf nginx-1.6.1.tar.gz
#cd nginx-1.6.1
#sudo ./configure
--prefix=/usr/local/nginx-1.6.1 #prefix指定安装目录
--with-openssl=/home/zht/src/openssl-1.0.1i #指的是openssl源码路径
--with-pcre=/home/zht/src/pcre-8.3.5 #指的是pcre的源码路径
--with-zlib=/home/zht/src/zlib-1.2.8 #指的是zlib 的源码路径
--with-http_ssl_module
#sudo make
#make install
```

安装完成后, 到/usr/local/下查看是否安装成功。如果安装出错, 需要重新加压缩, 重新安装。



1.5.1 配置Nginx

\*通过Ajax的方式执行GP服务

\*编译器架构的王者LLVM——  
(6) 多遍翻译的宏翻译系统

\*【笨木头Unity】入门之旅  
010 (完结) : Demo之四处找死  
(五) \_UI

## 最新评论

移动设备、手机浏览器Javascr  
飞侠:  
<https://github.com/shipeifei/moveto>  
大家可以看一下这...

Nginx+Tomcat7+Memcached负  
战斗鸡: 非常感谢楼主! 按你的文  
档, 第一次弄这个, 半天多一点  
而就把环境搭建好了! 灰常感  
谢, 威武!

Html5饼图绘制 (统计图)  
baidu\_32679445: 我想问下, 饼  
图大小不要随Canvas容器高度自  
动调整, 我就想固定圆饼大小,  
然后固定放在画布的某个方位...

Java中@Override的作用  
lhmm77521: 懂了

mybatis连接mysql数据库插入中  
huanxin2008: 参数之间用&amp;  
分割

mybatis连接mysql数据库插入中  
huanxin2008: 【出现问题】加了  
参数后报错: 对实体  
"characterEncoding" 的引用必须  
以 '分...

mybatis连接mysql数据库插入中  
Jimyang1ssa: 给力, 解决了我的  
问题!

数据库尽量不要存储较大的二进  
夕阳下的奔跑: 你要这么做也是  
可以的。不过呢可以考虑另外建  
一张表, 用关联字段。这样不影  
响你原表读取数据。等需要照  
片...

Java中@Override的作用  
jackfong\_123: thanks

Java Struts2 POI创建Excel文件  
皆-空: 学习了

配置文件目录: **/usr/local/nginx-1.6.1/conf/nginx.conf**

**# cd /usr/local/nginx-1.6.1/conf**

**# sudo vi nginx.conf**

【修改后的配置文件如下】:

**#创建进程的用户和用户组**

**user zht zht;**

**#服务进程数量, 一般等于CPU数量**

**worker\_processes 1;**

**#全局错误日志定义, 建议开启error级别日志.[ debug | info | notice | warn | error | crit ]**

**error\_log logs/error.log error;**

**#error\_log logs/error.log notice;**

**#error\_log logs/error.log info;**

**#记录进程ID的文件**

**#pid logs/nginx.pid;**

**events {**

**#epoll是多路复用IO(I/O Multiplexing)中的一种方式,但是仅用于linux2.6以上内核,可以大大提高nginx的性  
能.Linux建议使用epoll, FreeBSD建议使用kqueue.**

**useepoll;**

**#一个worker\_process允许的最近并发连接数量**

**worker\_connections 1024;**

**}**

**http {**

**include mime.types;**

**default\_type application/octet-stream;**

**#log\_format main '\$remote\_addr - \$remote\_user [\$time\_local] "\$request" '**

**# '\$status \$body\_bytes\_sent "\$http\_referer" '**

**# "\$http\_user\_agent" "\$http\_x\_forwarded\_for";**

**#access\_log logs/access.log main;**

**sendfile on;**

**#tcp\_nopush on;**

**#http连接的持续时间**

**keepalive\_timeout 65;**

**#gzip压缩设置**

**gzip on; #开启gzip**

**gzip\_min\_length 1k; #最小压缩文件大小**

**gzip\_buffers 4 16k; #压缩缓冲区**

**#http的协议版本(1.0/1.1),默认1.1, 前端如果是squid2.5请使用1.0**

**gzip\_http\_version 1.1;**

**#gzip压缩比, 1压缩比最小处理速度最快, 9压缩比最大但处理速度最慢(传输快但比较消耗cpu)**

**gzip\_comp\_level 2;**

**#和http头有关系, 加个vary头, 给代理服务器用的, 有的浏览器支持压缩, 有的不支持, 所以避免浪费不支  
持的也压缩, 所以根据客户端的HTTP头来判断, 是否需要压缩**

**gzip\_vary on;**

**#gzip压缩类型, 不用添加text/html, 否则会有警告信息**

```
gzip_types text/plain text/javascript text/css application/xmlapplication/x-javascript application/json;
```

#设定负载均衡的服务器列表，可以设置多个upstream，但mysvr名字要区分

```
upstreammyClusterServer1 {
```

```
#weight参数表示权值，权值越高被分配到的几率越大
```

```
#本机上的Squid开启3128端口
```

```
server 127.0.0.1:8081 weight=5;
```

```
server 127.0.0.1:8082 weight=5;
```

```
server 127.0.0.1:8083 weight=5;
```

```
}
```

```
server {
```

```
#nginx监听的端口号
```

```
listen 80;
```

```
#域名可以有多个，用空格隔开
```

```
server_name 127.0.0.1;
```

```
#字符编码方式
```

```
charset utf-8;
```

```
#设定本虚拟主机的访问日志。关闭日志可以减少IO，提高性能。
```

```
#access_log logs/host.access.log main;
```

```
#默认请求
```

```
location / {
```

```
#定义服务器的默认网站根目录位置
```

```
root html;
```

```
#定义首页索引文件的名称
```

```
index index.html index.htmindex.jsp;
```

```
#请求转向mysvr 定义的服务器列表
```

```
proxy_pass http://myClusterServer1;
```

```
proxy_redirect default;
```

#跟代理服务连接的超时时间，必须留意这个time out时间不能超过75秒，当一台服务器当掉时，过10秒转发到另外一台服务器。

```
proxy_connect_timeout 10;
```

```
}
```

```
#error_page 404 /404.html;
```

```
#redirect server 500 the static page /50x.html
```

```
#
```

```
error_page 500 /50x.html;
```

```
location = /50x.html {
```

```
root html;
```

```
}
```

```
#proxy the PHP scripts to Apache listening on 127.0.0.1:80
```

```
#
```

```
#location ~ \.php$ {
```


```
# proxy_pass http://127.0.0.1;
```

```
#}
```

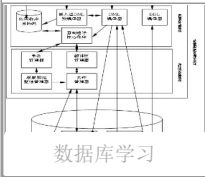
```
#pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
```

```
#
```

关闭



宣传视频



数据库学习

c++游戏开发

集群系统

计算机集群

北京租房

数据库集群技术

mysql

ME /scripts\$fastcgi\_script\_name;

cache's document root

```
#
#location ~ /\.ht {
#    deny all;
#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen    8000;
#    listen    somename:8080;
#    server_name somename alias another.alias;

#    location / {
#        root html;
#        index index.html index.htm;
#    }
#}

# HTTPSServer
#
#server {
#    listen    443 ssl;
#    server_name localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

#    ssl_session_cache    shared:SSL:1m;
#    ssl_session_timeout  5m;

#    ssl_ciphers  HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers on;

#    location / {
#        root html;
#        index index.html index.htm;
#    }
#}

}
```

1.5.2启动与关闭Nginx

(1)启动

```
# sudo /usr/local/nginx-1.6.1/sbin/nginx
```

确保系统的 80 端口没被其他程序占用

重启:

```
# sudo /usr/local/nginx-1.6.1/sbin/nginx -s reload
```

(2)关闭:

```
ps -ef | grep nginx
```

找到主进程ID, 然后kill即可, 如:

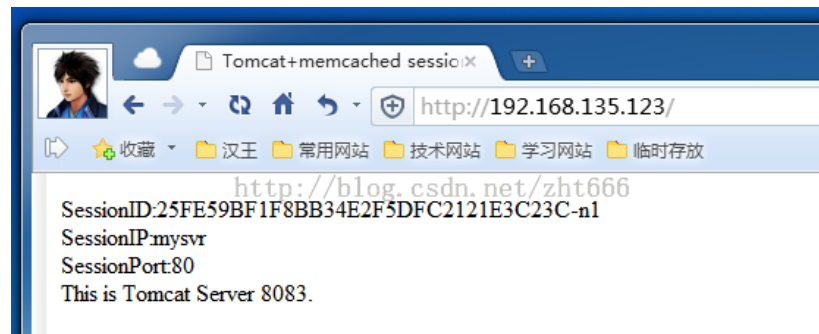
```
# kill -9 [进程号]
```

```
zht@VirtualBox-CentOS6:/usr/local/nginx-1.6.1
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[zht@VirtualBox-CentOS6 nginx-1.6.1]$ ps -ef | grep nginx
root      13237      1    0 05:26 ?        00:00:00 nginx: master process sbin/nginx
zht       13270    13237    0 05:28 ?        00:00:00 nginx: worker process
zht       13273    13213    0 05:28 pts/2    00:00:00 grep nginx
[zht@VirtualBox-CentOS6 nginx-1.6.1]$
```

(3)检查是否启动成功:

```
netstat -ano | grep 80 有结果输入说明启动成功
```

打开浏览器访问此机器的 IP, 如果浏览器出现Welcome to nginx! 则表示 Nginx 已经安装并运行成功。如果已经配置了负载均衡服务器, 则会看Tomcat中的网站页面, 如下图:



## 2. Memcache安装

Memcached官网: <http://memcached.org/>

安装memcached需要先安装libevent, libevent官网: <http://libevent.org/>

本次安装版本:

memcached-1.4.20.tar.gz

libevent-2.0.21-stable.tar.gz

2.1安装livevent

查看是否已安装: `# rpm qa | grep libevent`

如果已安装且版本低于1.3, 则先通过:

```
# rpm -e libevent --nodeps 进行卸载。
```

```
# tar zxvf libevent-2.0.21-stable.tar.gz
```

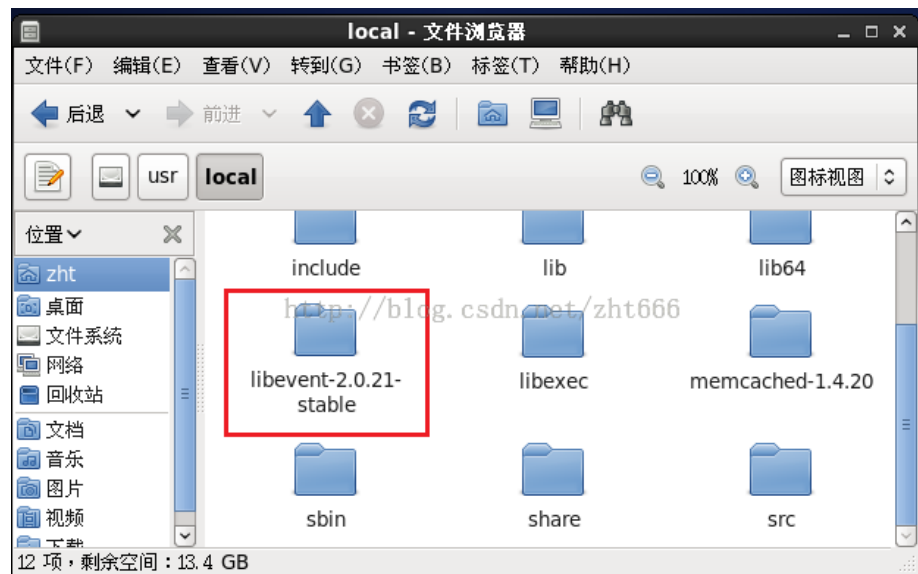
```
# cd libevent-2.0.21-stable
```

```
# sudo ./configure --prefix=/usr/local/libevent-2.0.21-stable #prefix指定安装路径
```

```
# sudo make
```

```
# sudo make install
```

安装完成后, 到prefix指定的目录下, 看看是否存在libevent-2.0.21-stable目录, 如下图所示。

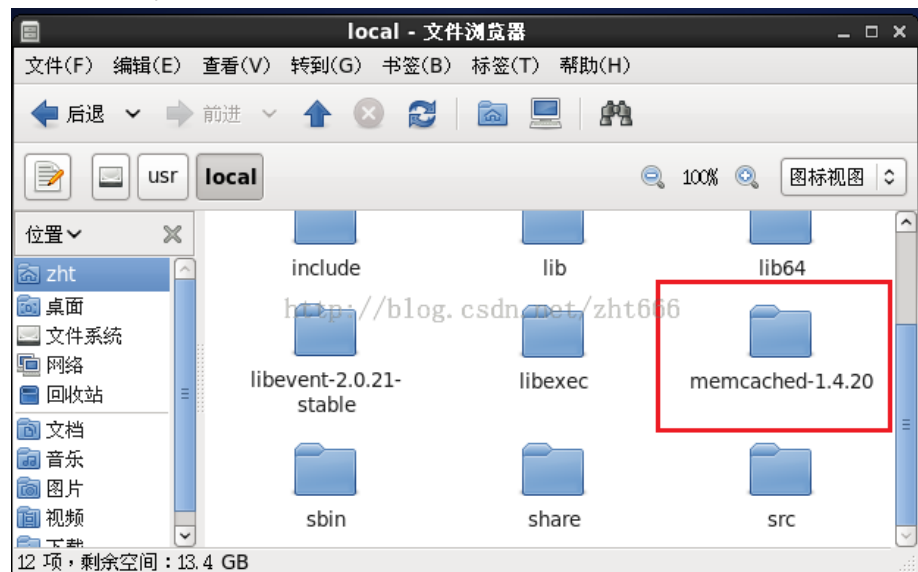


## 2.2 安装Memcached

```
# tar zxvf memcached-1.4.20.tar.gz
# cd memcached-1.4.20
# sudo ./configure
--prefix=/usr/local/memcached-1.4.20
--with-libevent=/usr/local/libevent-2.0.21-stable
# sudo make
# sudo make install
```

### 2.2.1 检查安装情况

安装完成后，到prefix指定的目录下查看是否有memcached-1.4.20目录，如下图所示。



### 2.2.2 查看memcached和libevent版本信息

首先定位到Memcached的bin目录下：

```
# cd /usr/local/memcached-1.4.20/bin
```

执行命令：

```
# sudo ./memcached -i
```

### 2.2.3 启动memcached

```
# sudo ./memcached -d -v -p 12000 -m 512 -u zht
```

解释：-d表示以守护进程方式运行memcached；-v表示输出浸膏和错误信息；-p指定监听的端口号；-m指定能使用的最大内存，单位MB；-u指定运行memcached的账户，非root用户。

使用# ps -ef | grep memcached查看进程。



关注基本选项:	说明
-p <num>	监听的TCP端口 (缺省: 11211)
-d	以守护进程方式运行memcached
-u <username>	运行memcached的账户, 非root用户
-m <num>	最大的内存使用, 单位是MB, 缺省是 64 MB
-c <num>	软连接数量, 缺省是1024 (最大并发连接数)
-v	输出警告和错误信息
-vv	打印客户端的请求和返回信息
-h	打印帮助信息
-i	打印memcached和libevent的版权信息

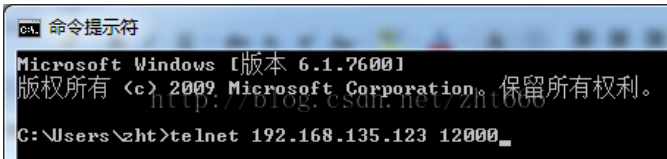
2.2.4使用telnet验证服务是否可用

Win7安装telnet: 控制面板-> 程序和功能->打开或安装Windows功能->勾选Telnet服务端、Telnet客户端。



CentOS安装telnet客户端: # sudo yum install telnet

Windows命令:



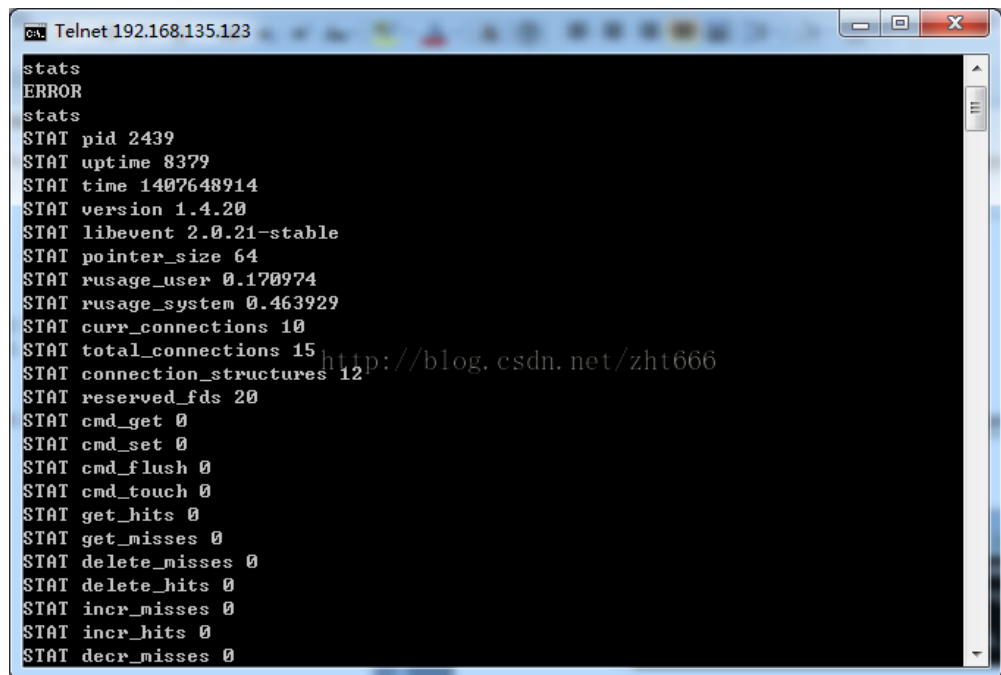
CentOS (Linux) 下命令:

# telnet 127.0.0.1 12000  
Trying 127.0.0.1...  
Connected to localhost (127.0.0.1).  
Escape character is '^['.

连接成功后, 手动输入命令: stats

即可看到如下信息:



A screenshot of a Telnet window titled 'Telnet 192.168.135.123'. The window shows a series of commands and their outputs. The command 'stats' is entered, followed by 'ERROR', and then 'stats' again. The output of the second 'stats' command is a list of statistics for the memcached service, including pid, uptime, time, version, libevent version, pointer size, memory usage, connections, and various hit/miss counts. A watermark 'http://blog.csdn.net/zht666' is visible across the middle of the screenshot.

```
stats
ERROR
stats
STAT pid 2439
STAT uptime 8379
STAT time 1407648914
STAT version 1.4.20
STAT libevent 2.0.21-stable
STAT pointer_size 64
STAT rusage_user 0.170974
STAT rusage_system 0.463929
STAT curr_connections 10
STAT total_connections 15
STAT connection_structures 12
STAT reserved_fds 20
STAT cmd_get 0
STAT cmd_set 0
STAT cmd_flush 0
STAT cmd_touch 0
STAT get_hits 0
STAT get_misses 0
STAT delete_misses 0
STAT delete_hits 0
STAT incr_misses 0
STAT incr_hits 0
STAT decr_misses 0
```

手动输入: quit //退出

Connection closed by foreign host

2.2.5停止memcached服务

# ps -ef | grep memcached

找到memcached进程ID号, 然后强制杀死:

kill -9 2439

### 3. 安装Tomcat+配置memcached

Tomcat官网: <http://tomcat.apache.org/>

3.1安装Tomcat

本次使用Tomcat版本: apache-tomcat-7.0.55.tar.gz

将Tomcat解压到任意目录下。在同一台机器上安装, 多个Tomcat时, 端口号需要设置成不一样的, 编辑三个Tomcat的server.xml将端口号都改成不一样的, 我的三个Tomcat端口号分别是8081,8082,8083。Tomcat8081的server.xml文件内容如下(去掉了注释部分, 其它两个做类似的修改):



### 3.2为Tomcat配置memcached

#### 3.2.1为Tomcat添加库文件

Tomcat要支持memcached管理Session，需要调用一些jar库文件：

- asm-3.2.jar
- kryo-1.04.jar
- kryo-serializers-0.11.jar
- memcached-session-manager-1.8.2.jar
- memcached-session-manager-tc6-1.8.2.jar
- memcached-session-manager-tc7-1.8.2.jar
- memcached-session-manager-tc8-1.8.2.jar
- minlog-1.2.jar
- msm-kryo-serializer-1.8.2.jar
- reflectasm-1.01.jar
- spymemcached-2.11.1.jar

msm1.6.5依赖了Couchbase，需要添加couchbase-client的jar包，否则启动会报：

java.lang.NoClassDefFoundError: com/couchbase/client/CouchbaseClient。【注意】：

Tomcat6和Tomcat7使用不同msm支持包：memcached-session-manager-tc6-1.6.5.jar和memcached-session-manager-tc7-1.6.5.jar，只可选一，否则启动报错。

msm源码中的lib包版本太低：spymemcached需要使用2.10.2，否则启动tomcat报错：

```
java.lang.NoSuchMethodError:net.spy.memcached.MemcachedClient.set(Ljava/lang/String;ILjava/lang/Object;)L
atde.javakaffee.web.msm.BackupSessionTask.storeSessionInMemcached(BackupSessionTask.java:227)
```

kryo-serializers需要使用0.10版本，否则报错：

Caused by:java.lang.ClassNotFoundException: de.javakaffee.kryoserializers.DateSerializer

部分文件下载地址：<http://code.google.com/p/memcached-session-manager/downloads/list>

其他的文件自己找。

下载后，将这些库文件放到tomcat\lib目录下。

### 3.2.2为Tomcat配置memcached

配置文件目录：

**tomcat\conf\context.xml**

**tomcat\conf\server.xml**

打开context.xml文件，在<Context>...</Context>节点中添加如下内容：

```
<ManagerclassName="de.javakaffee.web.msm.MemcachedBackupSessionManager"
    memcachedNodes="n1:127.0.0.1:12000"
    sticky="false"
    requestUriIgnorePattern=".*\.(png|gif|jpg|css|js|ico|jpeg)$"
    sessionBackupAsync="false"
    sessionBackupTimeout="100"
    transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"/>
```

【参数说明】：

docBase: 与<Host>中的appBase一致，网站部署目录。

memcachedNodes: memcached服务器信息，多个服务器时，使用空格分开，如：

n1:127.0.0.1:12001 n2:127.0.0.1:12002 n3:127.0.0.1:12003

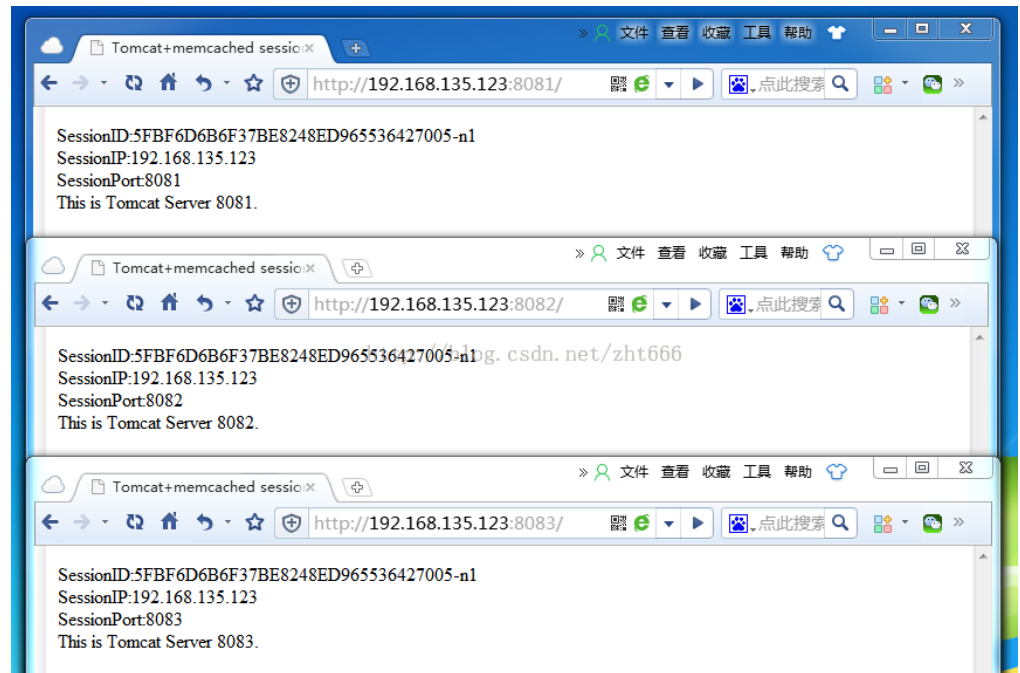
打开server.xml文件，在<Engine>节点的中添加jvmRoute="tomcatRoute1"，这个是为了指定Tomcat集群的路由。Tomcat2设置成jvmRoute="tomcatRoute2"，Tomcat3设置成jvmRoute="tomcatRoute3"，依次类推。

### 3.2.3测试Session共享

测试JSP代码如下：index.jsp

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Tomcat+memcached共享session测试</title>
</head>
<body>
SessionID:<%=session.getId()%>
<BR>
SessionIP:<%=request.getServerName()%>
<BR>
SessionPort:<%=request.getServerPort()%>
<BR>
<%
out.println("This is Tomcat Server 8081.");
%>
</body>
</html>
```

同时启动多个Tomcat，我部署了3个，打开浏览器去访问第一个Tomcat，然后在访问第二个和第三个Tomcat，页面Session信息如下：



由图可以看到，三个Tomcat的SessionID都是一样的：5FBF6D6B6F37BE8248ED965536427005-n1，只要不关闭浏览器，不管怎么刷新，SessionID都是不变了。由此可以，三个Tomcat通过memcached实现了Session信息共享。

### 3.2.3 Manager标签属性说明

className：此属性是必须的。

memcachedNodes：此属性是必须的。这个属性必须包含你所有运行的memcached节点。每个节点的定义格式为<id>:<host>:<port>。多个之间用空格或半角逗号隔开（如：

memcachedNodes="n1:localhost:11211,n2:localhost:11212"）。如果你设置单个memcache节点<id>是可选的，所以它允许设置为<host>:<port>（memcachedNodes="localhost:11211"）。

failoverNodes：可选项，属性只能用在非粘连Session机制中。此属性必须包含memcached节点的Id，此节点是Tomcat作为备份使用。多个之间用空格或逗号隔开。

memcachedProtocol：可选项，默认为text。出属性指明memcached使用的存储协议。只支持text或者binary。

sticky：可选项，默认为true。指定使用粘性的还是非粘性的Session机制。

lockingMode：可选项，此属性只对非粘性Session有用，默认为none。指定非粘性Session的锁定策略。它的值的只有：

- (1) none：从不加锁
- (2) all：当请求时对Session锁定，直到请求结束
- (3) auto：对只读的request不加锁，对非只读的request加锁
- (4) uriPattern:<regex>：使用正则表达式来比较requestRUI + "?" + queryString来决定是否加锁，

requestUriIgnorePattern：可选项，此属性是那些不能改备份Session的请求的正则表达式。如果像css、javascript、图片等静态文件被同一个Tomcat和同一个应用上下文来提供，这些请求也会通过memcached-session-manager。但是这些请求在一个http会话中几乎没什么改变，所以他们没必要触发Session备份。所以那些静态文件没必要触发Session备份，你就可以使用此属性定义。此属性必须符合java regex正则规范。

sessionBackupAsync：可选项，默认true。指定Session是否应该被异步保存到Memcached中。如果被设置为true，backupThreadCount设置起作用，如果设置false，通过sessionBackupTimeout设置的过期时间起作用。

backupThreadCount：可选项，默认为CPU内核数。用来异步保存Session的线程数(如果sessionBackupAsync="true")。

**sessionBackupTimeout:** 可选项，默认100，单位毫秒。设置备份一个Session所用的时间，如果操作超过时间那么保存失败。此属性只在**sessionBackupAsync="false"**起作用。默认100毫秒。

**sessionAttributeFilter:** 可选项从1.5.0版本有。此属性是用来控制Session中的那个属性值保存到Memcached中的正则表达式。正则表达式被用来匹配Session中属性名称。如

**sessionAttributeFilter="^(userName|sessionHistory)\$"** 指定了只有"userName"和"sessionHistory"属性保存到Memcached中。依赖于选择的序列化策略。

**transcoderFactoryClass:** 可选，默认为de.javakaffee.web.msm.JavaSerializationTranscoderFactory，此属性值是创建序列化和反序列化保存到Memcached中的Session的编码转换器的工厂类名。这个指定的类必须实现了de.javakaffee.web.msm.TranscoderFactory和提供一个无参的构造方法。例如其他的有效的实现在其他packages/jars中提供如：msm-kryo-serializer,msm-xstrea-serializer和msm-javolution-serializer。

**copyCollectionsForSerialization:** 可选项，默认false。

**customConverter:** 可选项

**enableStatistics:** 可选项，默认true用来指定是否进行统计。

**enabled:** 可选项，默认true。指定Session保存到Memcached中是否可用和是否可以通过JMX进行改变。只用于粘性Session。

## 4. 安装Samba共享文件服务

查看是否已经安装samba:

```
# rpm -qa | grep samba
```

### 4.1 安装samba

使用网络安装快速方便，命令如下:

```
# sudo yum install samba samba-client
```

### 4.2 配置共享文件夹

(1)修改配置文件:

```
# sudo vi /etc/samba/smb.conf
```

[SharedFolder]

```
path = /home/zht/SharedFolder      #共享目录的路径
```

```
writable = yes                      #是否允许写
```

```
browsable = yes                    #是否允许以浏览目录内容
```

```
valid user = zht                   #访问目录的账户名
```

(2)添加账户

添加zht账户: **# sudosmbpasswd -a zht**

按照提示输出访问密码。

### 4.3 开启端口号

Port 137 (UDP) --NetBIOS name server and nmbd

Port 138 (UDP) --NetBIOS datagram service

Port 139 (TCP) --File and printer sharing and smbd

Port 389 (TCP) --for LDAP(Active Directory Mode)

Port 445 (TCP) --NetBIOS was moved to 445 after 2000 and beyond(CIFS)

Port 901 (TCP) --for SWAT

(1) CentOS使用图形化添加端口号: **sudo setup;**

(2) 使用以下命令开启:

```
# iptables -I INPUT -p udp --dport [端口号] -j ACCEPT
```

```
# iptables -I INPUT -p tcp --dport [端口号] -j ACCEPT
```

保存:

```
# sudo service iptables save
```

#### 4.4关闭SELinux

查看状态: `# getenforce`

关闭: `# setenforce 0`

SELinux几种状态:

**enforcing:** 强制模式, 代表SELinux运行中且正确限制;

**permissive:** 宽容模式, 代表SELinux运行中, 不过仅有警告信息, 并不实际限制;

**disable:** 关闭, SELinux未运行。

永久关闭方法:

```
# sudo vi /etc/selinux/config
```

```
# SELINUX = enforcing    注释掉
```

```
# SELINUXTYPE = targeted  注释掉
```

在末尾添加一行:

```
SELINUX = disable
```

保存关闭: `:wq`

重启系统

#### 4.5启动与停止服务

(1) 查看运行状态:

```
# sudo service smb status
```

(2) 启动和停止服务:

```
# sudo /etc/init.d/smb start/stop/restart
```

或者

```
# sudo service smb start/stop/restart
```

(3) 设置开启启动:

```
# sudo chkconfig --level 35 smb on
```

在3,5级别上自动运行smb服务

(4) 验证服务

```
# smbclient -L //192.168.101.249 -U rxxy
```

或者

```
# smbclient //192.168.101.249/sharedFolder -U rxxy
```

## 5安装配置JDK

本次使用JDK版本为: `jdk-7u60-linux-x64.tar.gz`

#### 5.1卸载已安装的JDK

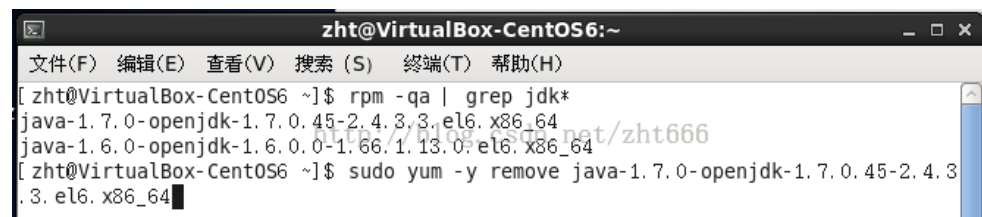
查看已经安装的JDK

```
# rpm -qa | grep jdk*
```

卸载JDK:

```
# sudo yum -y remove [包名]
```

如下图所示:



安装新的JDK: 将`jdk-7u60-linux-x64.tar.gz`解压缩到指定目录下即可。

## 5.2配置Java环境变量

```
# sudo vi /etc/profile
```

在末尾添加:

```
#JDK
```

```
export JAVA_HOME=/home/zht/BalanceServer/Java/jdk1.7.0_60
```

```
export JRE_HOME=$JAVA_HOME/jre
```

```
export CLASSPATH=$CLASSPATH:.$JRE_HOME/lib:$JAVA_HOME/lib
```

```
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

## 6安装MySQL数据库

### 6.1卸载已安装的mysql

查看已经安装的mysql数据库:

```
# rpm -qa | grep mysql
```

卸载已安装的mysql数据库:

```
# rpm -e [包名] //普通卸载, 可能因为依赖无法卸载
```

```
# rpm -e --nodeps [包名] //强制卸载, 有依赖也能卸载
```

或者

```
# yum -y remove [包名] //卸载时自动处理依赖
```

### 6.2安装mysql

安装MySQL数据库需要安装4个组件: mysql--shared-compat、mysql-server、mysql-client和mysql-workbench, 安装方法如下。

(1) 安装mysql共享库:

```
# rpm -ivh MySQL-shared-compat-5.6.20-1.el6.x86_64.rpm
```

(2) 安装mysql服务端: :

```
# rpm -ivh MySQL-server-5.6.20-1.el6.x86_64.rpm
```

(3) 安装mysql客户端:

```
# rpm -ivh MySQL-client-5.6.20-1.el6.x86_64.rpm
```

(4) 安装mysql-workbench工具:

```
# rpm -ivh mysql-workbench-community-6.1.7-1.el6.x86_64.rpm
```

【说明】: RPM安装包, 在CentOS下可以直接双击运行安装, 系统会自动处理依赖。

另附: yum安装方法

查看yum上可安装的版本: # yumlist | grep mysql

安装: # yum install -y mysql-server mysql mysql-devel

### 6.3配置mysql

配置文件目录: /usr/my.cnf或者/etc/mysql/my.cnf

打开文件后, 在[mysqld]后面添加如下配置:

```
#字符编码
```

```
character-set-server=utf-8
```

```
#存储引擎
```

```
default-storage-engine=INNODB
```

```
innodb_flush_log_at_trx_commit=2
```

```
#缓冲池
```

```
innodb_additional_mem_pool_size=16M
```

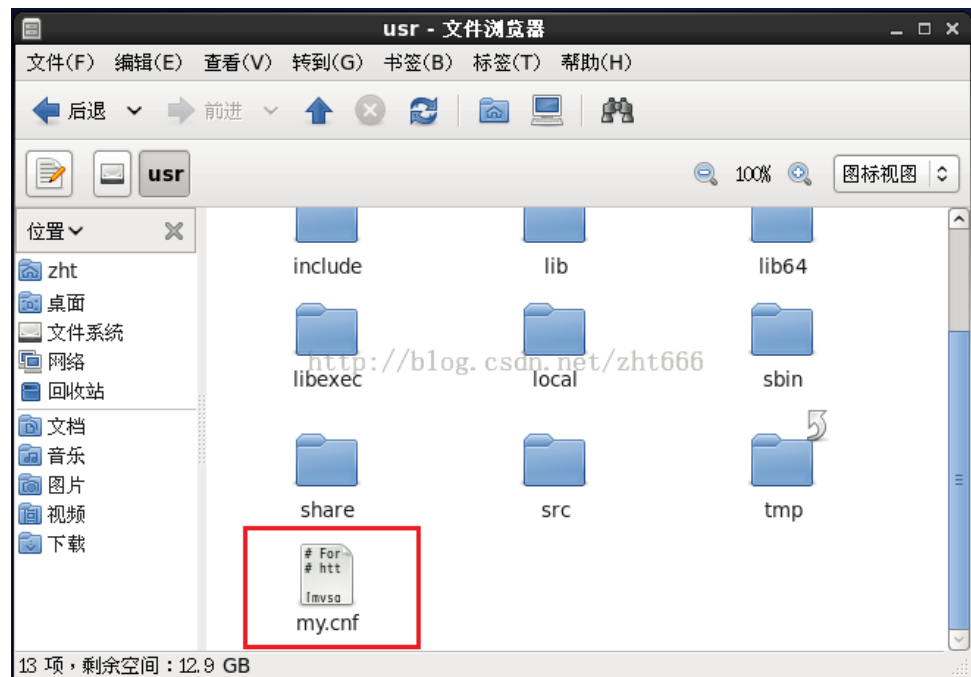
```
innodb_buffer_pool_size=1024M
```

```
#最大连接数
```

```
max_connections=1000
```

```
#允许的最大包大小 (例如一个Blob字段)
```

```
max_allowed_packet=16M
```



#### 6.4 启动mysql

(1) 启动mysql:

```
# sudo service mysql start/stop/restart
```

或

```
# sudo /etc/rc.d/init.d/mysql start/stop/restart
```

(或 # sudo /etc/rc.d/init.d/mysqld start/stop/restart)

启动mysql出错时, 请到/var/lib/mysql/xxx.err查看错误信息。



(2) 登录MySQL

```
# mysql -u root -p
```

# 输入密码

```
# mysql> show databases;
```

```
# mysql> show variables like '%character_set%'
```

```
# mysql> ...
```

(3) 创建远程连接用户

启用sxzl用户在本地 (localhost) 的登录权限, 密码hymmlr.

```
# mysql> grant all privileges on *.* to sxzl@localhost identified by 'hymmlr.' with grant option;
```

启用sxzl用户远程 (%) 登录权限, 密码hymmlr.

```
# mysql> grant all privileges on *.* to sxzl@%" identified by 'hymmlr.' with grant option;
```

刷新权限

```
# mysql> flush privileges;
```

第一次安装需要重设mysql密码, 详见如下。

#### 6.5 重设mysql密码 (官网方法)

启动mysql, 然后执行如下命令:

```
# ps -ef | grep -i mysql
```

查看mysqld\_safe的路径, 例如: /usr/bin/mysqld\_safe





停止mysql，然后以安全方式启动mysql:

# sudo /usr/bin/mysqld\_safe --skip-grant-tables >/dev/null 2>&1 &

5秒钟后执行:

# sudo /usr/bin/mysql -u root mysql

重设root用户的密码:

# mysql> update user SET PASSWORD=PASSWORD('root')where user='root'

刷新权限

# mysql> flush privileges;

# mysql> exit;

接着再次登录mysql:

# mysql -u root -p

# 输入密码

然后执行:

# mysql> show databases;

报错: You must SET PASSWORD before executing thisstatement.

解决办法, 重设一次密码:

# mysql> SET PASSWORD=PASSWORD('root');

# mysql> flush privileges;

设置完毕后, 就可以打开mysql-workbench连接到数据库了。

## 7命令说明: yum

yum安装选项说明:

yum -y install 包名 (支持\*) : 自动选择y, 全自动

yum install 包名 (支持\*) : 手动选择y or n

yum -y remove [package name] : 自动处理依赖

yum remove 包名 (不支持\*)

rpm -ivh 包名 (支持\*) : 安装rpm包

rpm -e 包名 (不支持\*) : 卸载rpm包

本文文档下载地址: <http://download.csdn.net/detail/zht666/7744157>

上一篇 SVN三种合并类型

下一篇 JNA结构体数组

顶

3

踩

0

我的同类文章

CentOS/Linux (1)	nginx (4)	MySQL (11)

- CentOS6.4下安装配置Samba

主题推荐      tomcat      nginx

猜你在找

- |   |                                      |
|---|--------------------------------------|
| Html5服务器推送事件                              | nginx负载均衡 tomcat集群 memcache共享session |
| 基于Servlet+JDBC+Bootstrap+MySQL+AJAX权限管理系统 | Nginx+tomcat配置集群负载均衡                 |
| 高并发集群架构超细精讲                               | 图文解说Nginx + tomcat配置集群负载均衡           |
| MySQL SQL优化及高可用公开课视频分享                    | nginx + tomcat 配置集群负载均衡转             |
| 高并发之Memcached实战                           | 图文解说Nginx+tomcat配置集群负载均衡             |

100offer  
互联网人才拍卖

告别盲目投简历

让海量好机会主动来找你

了解更多

查看评论

9楼 战斗鸡 2015-11-16 17:22发表



非常感谢楼主！按你的文档，第一次弄这个，半天多一点而就把环境搭建好了！灰常感谢，威武！

8楼 MasqueJava 2015-04-13 12:56发表



现在明显是两个在一台服务器上的才集群了  
我采用tomcat的session复制机制也是这样  
求指教

7楼 MasqueJava 2015-04-13 12:55发表



B: 04  
SessionID:AE54C77D509EEE3F2BFD2B1F68FDE515-n1.tomcatRoute3  
SessionIP:192.168.17.246  
SessionPort:9002  
This is Tomcat Server 9002.

6楼 MasqueJava 2015-04-13 12:54发表



B:03  
SessionID:AE54C77D509EEE3F2BFD2B1F68FDE515-n1.tomcatRoute3  
SessionIP:192.168.17.246  
SessionPort:9001  
This is Tomcat Server 9001.

5楼 MasqueJava 2015-04-13 12:54发表



A: 02  
SessionID:B5AAADC7D5F41B3CF49F0117F3510A35-n1.tomcatRoute2  
SessionIP:localhost  
SessionPort:8082  
This is Tomcat Server 8082.

4楼 MasqueJava 2015-04-13 12:52发表



A :01  
SessionID:B5AAADC7D5F41B3CF49F0117F3510A35-n1.tomcatRoute2  
SessionIP:localhost  
SessionPort:8081  
This is Tomcat Server 8081.

3楼 MasqueJava 2015-04-13 12:52发表



现在我在两台电脑上分别按照了2个实例  
A: 01,02  
B: 03,04

2楼 冲天 2014-08-17 00:08发表



您能帮我看下吗,我的qq号是865111840,谢谢

Re: 海涛zht666 2014-08-17 10:05发表



回复冲天: Tomcat需要修改server.xml文件, 修改端口号才行, 我已经在文章中第3点添加了Tomcat修改端口号的配置。

1楼 冲天 2014-08-17 00:07发表



您好,我在安装tomcat后 无法打开localhost:8081 localhost:8082 localhost:8083

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker  
OpenStack   VPN   Spark   ERP   IE10   Eclipse   CRM   JavaScript   数据库   Ubuntu   NFC  
WAP   jQuery   BI   HTML5   Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML  
LBS   Unity   Splashtop   UML   components   Windows Mobile   Rails   QEMU   KDE   Cassandra  
CloudStack   FTC   coremail   OPhone   CouchBase   云计算   iOS6   Rackspace   Web App  
SpringSide   Maemo   Compuware   大数据   aptech   Perl   Tornado   Ruby   Hibernate   ThinkPHP  
HBase   Pure   Solr   Angular   Cloud Foundry   Redis   Scala   Django   Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服   杂志客服   微博客服   webmaster@csdn.net   400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

