服务器集群

Apache 和 nginx（web服务器）

1. 多台集群机器联合处理一个任务。
2. 一台机器处理不同的多个任务。

特点：

处理静态数据速度快（自动html静态数据的缓存）

并发数一般要比J2EE服务器要高。

安装apache：

192.168.1.10

Apache

Weblogic

Tomcat

1、简单安装：yum install httpd

2、源码安装

1、安装apr

* + - 1. 解压
      2. ./configure --prefix=/usr/local/apr 指定安装目录
      3. Make && make install
    1. 安装apr-util
       1. 解压
       2. ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
       3. Make && make install
    2. 源码安装apache
       1. 解压

./configure --prefix=/usr/local/apache --sysconfdir=/etc/httpd --enable-so --enable-ssl --enable-cgi --enable-rewrite --with-zlib --with-pcre --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util --enable-modules=most --enable-mpms-shared=most --with-mpm=event

* + - 1. Make && make install
  1. Apache和tomcat的连接

apache 连接tomcat

连接方法：两种

基于ajp协议方式连接，基于二进制

基于HTTP协议连接

两种连接工具：

Proxy:mod\_proxy.so

Connectors：mod\_jk.so

1、mod\_jk :使用的是ajp连接器。

mod\_jk 1x

mod\_jk 2x :丢弃

ajp协议：

基于二进制，速度快，兼容。安全。

1、下载mod\_jk

a、tomcat-conentors ，需要安装之后才能得到mod\_jk.so

b、mod\_jk.so

2、mod\_jk.so放到apache目录下mudules目录

3、编辑apache配置文件

LoadModule jk\_module modules/mod\_jk.so

JkWorkersFile conf/workers.properties

JkLogFile logs/mod\_jk.log

JkLogLevel info

JkMount /\* controller

新建一个workers.properties

worker.list=controller

worker.tomcat1.port=8009

worker.tomcat1.host=192.168.197.128

worker.tomcat1.type=ajp13

worker.tomcat1.lbfactor=1

worker.tomcat2.port=8009

worker.tomcat2.host=192.168.197.129

worker.tomcat2.type=ajp13

worker.tomcat2.lbfactor=1

worker.controller.type=lb #lb:表示负载均衡器

worker.controller.balance\_workers=tomcat2,tomcat1

worker.controller.sticky\_session=0 # 表示apache不处理session，每次传入sessionID

worker.controller.sticky\_session\_force=0

2、proxy\_mod：支持ajp和http连接器。 （官方推荐）

是apache自带的一个模块。

1、检查Proxy\_mod是否开启。

2、编辑apache配置文件：httpd.cnf

ProxyVia Off #设置响应时是否出现主机信息。

ProxyRequests Off #:是否开启apache的正向代理功能，正反向不能同时使用

ProxyPreserveHost Off #:是否在反向代理中支持虚拟主机

ProxyPass / ajp://localhost:8009/

ProxyPassReverse / ajp://localhost:8009/ #表示用户响应是可能使用重定向也需要经过代理服务器

3、负载均衡器

1、硬件负载均衡

2、软件负载均衡

在apache配置文件中

<Proxy balancer://mybl>

BalancerMember ajp://192.168.197.128:8009 loadfactor=1 route=tomcat1

BalancerMember ajp://192.168.197.129:8009 loadfactor=1 route=tomcat2

ProxySet lbmethod=byrequests

</Proxy>

ProxyPass / balancer://mybl/ 使用负载均衡器

ProxyPassReverse / balancer://mybl/

mybl:表示负载均衡器的名字

BalancerMember:代表一个后端应用服务器

loadfactor=权重 ，决定节点压力大小。

route：后端应用服务器的名字（负载均衡时必须要）

在tomcat的配置文件中定义名字

server.xml

ProxySet lbmethod=byrequests ,根据请求的数量进行负责均衡

lbmethod=byrequests 按照请求次数均衡(默认)

lbmethod=bytraffic 按照流量均衡

lbmethod=bybusyness 按照繁忙程度均衡(总是分配给活跃请求数最少的服务器)

stickysession=JSESSIONID|jsessionid nofailover=On

apache虚拟主机

1. 一台机器运行多个项目。
   1. 项目a –》tomcat1：8080和项目 b ---》tomcat2：8081
   2. 使用apache配置虚拟主机

<VirtualHost \*:80> \* ：任意的客户端

DocumentRoot "/usr/local/apache/docs/dummy-host2.example.com" 静态资源路径

ServerName peiyou.com（该项目访问的域名，每个虚拟主机要不同）

ErrorLog "logs/dummy-host2.example.com-error\_log"

CustomLog "logs/dummy-host2.example.com-access\_log" common

#ProxyVia Off

#ProxyPreserveHost Off

#ProxyRequests Off

#ProxyPass / ajp://h1:8009/

#ProxyPassReverse / ajp://h1:8009/

</VirtualHost>

session：的处理

1、session复制

tomcat 本身带有复制session的功能。（不讲）

2、共享session

需要专门管理session的软件，

memcached 缓存服务，可以和tomcat整合，帮助tomcat共享管理session。

安装memcached

1、安装libevent

2、安装memcached

3、启动memcached

memcached -d -m 128m -p 11211 -l 192.168.197.128 -u root -P /temp/

-d:后台启动服务

-m:缓存大小

-p：端口

-l:IP

-P:服务器启动后的系统进程ID，存储的文件

-u:服务器启动是以哪个用户名作为管理用户

3、拷贝jar到tomcat的lib下

4、配置tomcat，每个tomcat里面的context.xml中加入：

<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"

memcachedNodes="n1:192.168.197.128:11211"

sticky="false"

lockingMode="auto"

sessionBackupAsync="false"

requestUriIgnorePattern=".\*\.(ico|png|gif|jpg|css|js)$"

sessionBackupTimeout="1000" transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"

/>

<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"

memcachedNodes="n1:host1.yourdomain.com:11211,n1:host2.yourdomain.com:11211"

sticky="false"

sessionBackupAsync="false"

lockingMode="uriPattern:/path1|/path2"

requestUriIgnorePattern=".\*\.(ico|png|gif|jpg|css|js)$"

transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"

/>

httpSession

session.setAttribute(“key”,value);

动态代理

Get

T1

T2

T1