私塾在线《高级软件架构师实战培训阶段二》

——跟着cc学架构系列精品教程

本部分课程概览

- 根据实际的应用需要，学习Web表现层性能优化的补充知识，大致包括：

1：Tomcat集群

2：集群后的Session统一管理

3：处理静态化页面中的动态内容

Tomcat集群

- Tomcat集群概述

所谓Tomcat集群，就是让多个Tomcat组合在一起，协同来完成工作。要做集群的话，面临的问题就是Http Session如何能在多个Tomcat中都有效。

- 常见的集群方案介绍

1：使用tomcat自带的cluster方式，在多个tomcat间自动实时复制session信息。优点是配置起来很简单；缺点是这个方案的效率低下，高并发下表现较差

2：利用前端的反向代理服务器的某些策略，比如Nginx的基于IP的Hash路由策略，来保证相同的IP始终被路由到同一个Tomcat上。优点是配置简单；缺点是容易造成热点访问，而且存在单点风险

3：把多个Tomcat的Session集中管理，存储到公共的缓存中，比如Memcached，前端再利用反向代理服务器来做负载均衡，优点是系统能很容易的进行水平扩展，也能保证较高的性能；缺点是需要自行修改应用，或者是添加其它管理和同步会话的应用，比如：memcached-session-manager，简称msm

MSM-1

- MSM（ memcached-session-manager ）是什么

  MSM是一个高可用的Tomcat集群中Session共享解决方案。

- MSM特性

  支持黏性、非黏性Session

  可处理Tomcat故障转移

  可处理Memcached故障转移

  插件式session序列化

  允许异步保存session

- 黏性、非黏性Session的方案

  最大的区别在于：

1：黏性Session的方案，适合于前端的负载均衡器能保证，每个用户的请求都路由到同一个Tomcat上这种场景

2：黏性Session的方案，会把Memcached中的数据同步存放到本地的Http Session区。

- SESSIONID的格式

  由于MSM知道Memcached节点列表，因此SESSIONID中会带着这些节点，比如：

  09D01FE736F0C682A2AA71D464EC5ED9-n1

MSM-2

- MSM安装配置

1：环境是Tomcat8，需要的jar包参见

2：把这些jar包添加到Tomcat8/lib，或者是应用自身的lib里面

3：修改context.xml或server.xml文件中的<Context>节点，示例如下：

（1）使用non-sticky session，memcached缓存

```
<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"
 memcachedNodes="n1:localhost:11211,n2:192.168.1.106:2222"
     sticky="false"
     sessionBackupAsync="false"
     lockingMode="auto"
     requestUrilgnorePattern=“.*\.(ico|png|gif|jpg|css|js)$”
    transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFac
    tory" />
```

（2）使用sticky session，memcached缓存

a：添加jvmroute参数，每个Tomcat配置的这个值要不一样，如下：

```
<Engine name="Catalina"defaultHost="localhost“ jvmRoute="tomcat1">
```

b：把上面配置中的sticky设置为true

MSM-3

c：不用配置lockingMode，这个是只有非黏性session才使用，默认值为none
none：从不对session进行锁定
all：session将一直被锁定，直到请求结束
auto：对于只读请求，session将不会被锁定，如果是非只读请求，则会被锁定
uriPattern:<regexp>：通过正则表达式的方式来对请求uri以及查询字符串进行匹配，只有匹配上的才会被锁定。

d：添加failoverNodes配置，用来指定备用的memcached节点

4：配置后启动Tomcat，如果没有报错就表示配置好了

5：可以自定义序列化的方案，除了默认的Java序列化外，常见的还有：kryo、javolution、xstream、flexjson等，可以下载相应的jar，并修改transcoderFactoryClass。
建议使用maven来管理并下载需要的jar

MSM-4

- 查看Memcached的数据

1：连接到Memcached服务器：telnet 192.168.1.106 2222
2：查看多有的Item：stats items
3：输出Item：stats cachedump itms后面的数据 0 ，0表示显示全部数据
4：输出具体的数据：get key ，key就是上面item后面的数据

- Tomcat故障转移

- Memcached故障转移

处理静态化页面中的动态内容

- 问题描述

- 常见的解决方案

1：在服务端处理，使用Varnish的ESI

2：在客户端处理，使用JS来异步加载内容

- 方案选择

1：对于重要的、需要页面一展示就应该出现的内容，宜采用方案一

2：对于耗时的、非重要的、不需要页面一展示就出现的内容，宜采用方案二