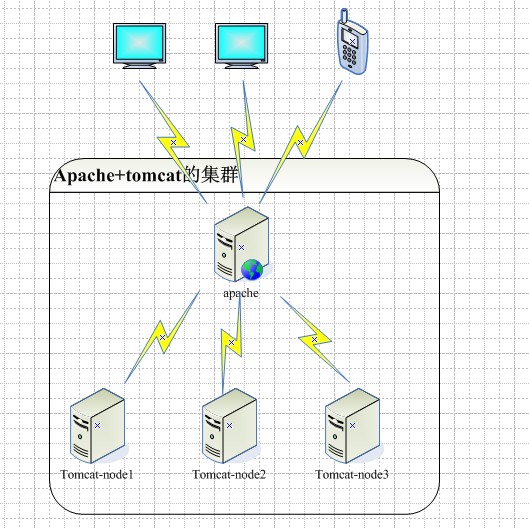
### [浅谈web应用的负载均衡、集群、高可用(HA)解决方案](http://aokunsang.iteye.com/blog/2053719)

****1、熟悉几个组件****  
  
****1.1、apache****  
     —— 它是Apache软件基金会的一个开放源代码的跨平台的网页服务器，属于老牌的web服务器了，支持基于Ip或者域名的虚拟主机，支持代理服务器，支持安全Socket层(SSL)等等，目前互联网主要使用它做静态资源服务器，也可以做代理服务器转发请求(如：图片链等)，结合tomcat等servlet容器处理jsp。  
****1.2、ngnix****  
     —— 俄罗斯人开发的一个高性能的 HTTP和反向代理服务器。由于Nginx 超越 Apache 的高性能和稳定性，使得国内使用 Nginx 作为 Web 服务器的网站也越来越多，其中包括新浪博客、新浪播客、网易新闻、腾讯网、搜狐博客等门户网站频道等，在3w以上的高并发环境下，ngnix处理能力相当于apache的10倍。  
     参考：apache和tomcat的性能分析和对比(http://blog.s135.com/nginx\_php\_v6/)  
****1.3、lvs****  
     —— Linux Virtual Server的简写，意即Linux虚拟服务器，是一个虚拟的服务器集群系统。由毕业于国防科技大学的章文嵩博士于1998年5月创立，可以实现LINUX平台下的简单负载均衡。了解更多，访问官网：http://zh.linuxvirtualserver.org/。

****1.4、HAProxy****

     —— HAProxy提供****高可用性****、****负载均衡****以及基于TCP和HTTP应用的代理，****支持虚拟主机****，它是免费、快速并且可靠的一种解决方案。HAProxy特别适用于那些负载特大的web站点， 这些站点通常又需要会话保持或七层处理。HAProxy运行在当前的硬件上，完全可以支持数以万计的并发连接。并且它的运行模式使得它可以很简单安全的整合进您当前的架构中， 同时可以保护你的web服务器不被暴露到网络上.  
****1.5、keepalived****  
     —— 这里说的keepalived不是apache或者tomcat等某个组件上的属性字段，它也是一个组件，可以实现web服务器的高可用(HA high availably)。它可以检测web服务器的工作状态，如果该服务器出现故障被检测到，将其剔除服务器群中，直至正常工作后，keepalive会自动检测到并加入到服务器群里面。实现主备服务器发生故障时ip瞬时无缝交接。它是LVS集群节点健康检测的一个用户空间守护进程，也是LVS的引导故障转移模块（director failover）。Keepalived守护进程可以检查LVS池的状态。如果LVS服务器池当中的某一个服务器宕机了。keepalived会通过一 个setsockopt呼叫通知内核将这个节点从LVS拓扑图中移除。  
****1.6、memcached****  
     —— 它是一个高性能分布式内存对象缓存系统。当初是Danga Interactive为了LiveJournal快速发展开发的系统，用于对业务查询数据缓存，减轻数据库的负载。其守护进程(daemon)是用C写的，但是客户端支持几乎所有语言(客户端基本上有3种版本[memcache client for java;spymemcached;xMecache])，服务端和客户端通过简单的协议通信；在memcached里面缓存的数据必须序列化。  
****1.7、terracotta****  
     —— 是一款由美国Terracotta公司开发的著名开源Java集群平台。它在JVM与Java应用之间实现了一个专门处理集群功能的抽象层，允许用户在不改变系统代码的情况下实现java应用的集群。支持数据的持久化、session的复制以及高可用(HA)。详细参考：http://topmanopensource.iteye.com/blog/1911679  
  
****2、关键术语****  
****2.1、负载均衡（load balance）****  
   
在互联网高速发展的时代，大数据量、高并发等是互联网网站提及最多的。如何处理高并发带来的系统性能问题，最终大家都会使用负载均衡机制。它是根据某种负载策略把请求分发到集群中的每一台服务器上，让整个服务器群来处理网站的请求。  
公司比较有钱的，可以购买专门负责负载均衡的硬件（如：F5）,效果肯定会很好。对于大部分公司，会选择廉价有效的方法扩展整个系统的架构，来增加服务器的吞吐量和处理能力，以及承载能力。  
  
****2.2、集群（Cluster）****  
  
 用N台服务器构成一个松耦合的多处理器系统(对外来说，他们就是一个服务器)，它们之间通过网络实现通信。让N台服务器之间相互协作，共同承载一个网站的请求压力。  
  
****2.3、高可用（HA）****  
  
 在集群服务器架构中，当主服务器故障时，备份服务器能够自动接管主服务器的工作，并及时切换过去，以实现对用户的不间断服务。ps：这里我感觉它跟故障转移(failover)是一个意思，看到的网友给个解释，谢谢？  
  
****2.4、session复制/共享****  
  
 在访问系统的会话过程中，用户登录系统后，不管访问系统的任何资源地址都不需要重复登录，这里面servlet容易保存了该用户的会话(session)。如果两个tomcat(A、B)提供集群服务时候，用户在A-tomcat上登录，接下来的请求web服务器根据策略分发到B-tomcat，因为B-tomcat没有保存用户的会话(session)信息，不知道其登录，会跳转到登录界面。  
这时候我们需要让B-tomcat也保存有A-tomcat的会话，我们可以使用tomcat的session复制实现或者通过其他手段让session共享。

****3、常用web集群****  
  
****3.1、tomcat集群方案****  
 apache+tomcat；ngnix+tomcat；lvs+ngnix+tomcat；大家比较熟悉的是前两种。(lvs负责集群调度，nginx负责静态文件处理，tomcat负责动态文件处理[最优选择])。 以apache+tomcat集群为例，简单说一下：  
  1、他们之间的通信有三种方式：ajp\_proxy、mod\_jk链接器、http\_proxy。具体参考：http://www.ibm.com/developerworks/cn/opensource/os-lo-apache-tomcat/  
  2、apache的分发策略有4种。权重(默认)、流量(bytraffic)、请求次数(byRequests)、繁忙程度(byBusyness根据活跃请求数的多少)  
  3、apache支持stickysession(粘性session)，即为：访问用户访问了A-tomcat，那么他的所有请求都会转发到A-tomcat，而不会到B-tomcat。[这样的负载均衡效果不好，适用于小型网站，下面说非粘性session]  
  4、它们之间的架构如图1：



问题1：只有一个web服务器，明显的单点故障。如果该apache出现问题，整个网站就会瘫痪。

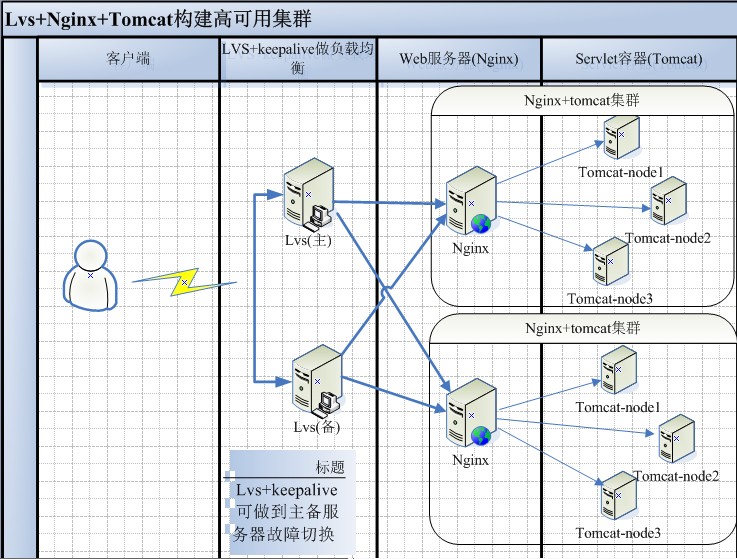
****3.2、session复制****

  如果不采用stickysession(粘性session)，那么我们可以采用tomcat的session复制使所有节点tomcat的会话相同，tomcat使用组播技术，只要集群中一个tomcat节点的session发生改变，会广播通知所有tomcat节点发生改变。  
  
问题2：据网友测试，当tomcat节点数达到4个以上时候，集群性能呈线性下滑；另外当用户访问量大到一定程度，会话内容随之增多，tomcat节点相互之间通信产生大量的网络消耗，产生网络阻塞，整个集群的吞吐量不能再上升。

****4、高可用(HA)和session共享(解决上面提到的两个问题)****

****4.1、使用lvs+keepalive实现集群高可用，达到更健壮的LB****  
 我们可以做前端使用lvs来做负载均衡，根据lvs的8种调度算法(可设置)，分发请求到对应的web服务器集群上。lvs做双机热备，通过keepalived模块能够达到故障自动转移到备份服务器，不间断提供服务，结构如图2：

   
说明：据查询了解，一般在WEB端使用的负载均衡比较多的是HAProxy+keepalived+nginx；数据库mysql集群使用Lvs+keepalived+mysql实现。因为HAProxy和nginx一样是工作在网络7层之上，并且前者弥补了nginx的一些缺点如session的保持，cookie的引导等，且它本身是个负责均衡软件，处理负载均衡上面必然优于nginx；lvs比较笨重，对于比较庞大的网络应用实施比较复杂，虽然它运行在网络4层之上，仅做分发没有流量产生，但是它不能做正则处理也不能也不能做动静分离，所以一般用lvs+keepalived或heatbeat做数据库层的负载均衡。



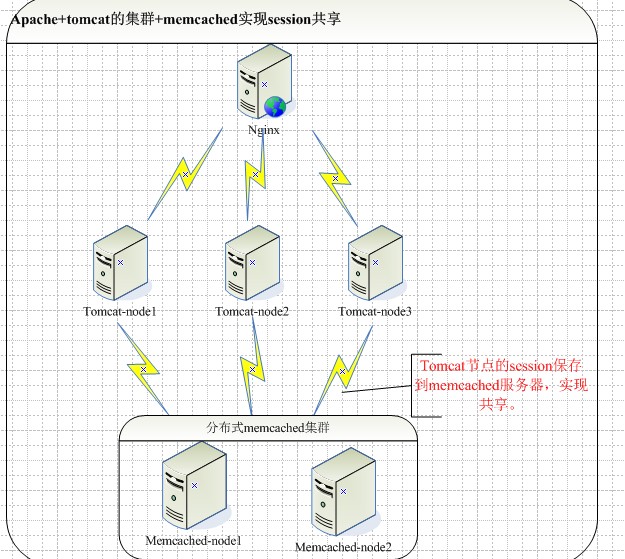
[LVS、HAProxy、Nginx做负载均衡的比较](http://www.blogjava.net/ivanwan/archive/2013/12/25/408014.html" \t "http://aokunsang.iteye.com/blog/_blank)  
****4.2、使用terracotta或者memcached使session共享****

****4.2.1、terracotta是jvm级别的session共享****

 它基本原理是对于集群间共享的数据，当在一个节点发生变化的时候，Terracotta只把变化的部分发送给Terracotta服务器，然后由服务器把它转发给真正需要这个数据的节点，并且共享的数据对象不需要序列化。

****4.2.2、通过memcached实现内存级session共享****

通过memcached-session-manager（msm）插件，通过tomcat上一定的配置，即可实现把session存储到memcached服务器上。注意：tomcat支持tomcat6+，并且memcached可以支持分布式内存，msm同时支持黏性session（sticky sessions）或者非黏性session（non-sticky sessions）两种模式，在memcached内存中共享的对象需要序列化。结构如图3：



  通过一定的配置，可以实现故障转移(只支持对非粘性session)。如：

**Xml代码**



1. **<Context>**
2. ...
3. **<Manager** className="de.javakaffee.web.msm.MemcachedBackupSessionManager"
4. memcachedNodes="n1:host1.yourdomain.com:11211,n2:host2.yourdomain.com:11211"
5. failoverNodes="n1"
6. requestUriIgnorePattern=".\*\.(ico|png|gif|jpg|css|js)$"
7. transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"
8. **/>**
9. **</Context>**

 说明：failoverNodes：故障转移节点，对非粘性session不可用。属性failoverNodes="n1"的作用是告诉msm最好是把session保存在memcached "n2"节点上，只有在n2节点不可用的情况下才把session保存在n1节点。这样即使host2上的tomcat宕机，仍然可以通过host1上的tomcat访问存放在memcached "n1" 节点中的session。  
   
****4.2.3、其他方案****  
通过cookie保存用户信息(一般是登录信息)，每一个请求到达web应用的时候，web应用从cookie中取出数据进行处理（这里尽量对cookie做加密处理）；  
另外一种是把用户信息的关键属性保存到数据库，这样就不需要session了。请求过来从数据库查询关键属性数据，做相应处理。***缺点：加大了数据库的负载，使数据库成为集群的瓶颈。***