# 大数据场景下的mongo架构

1. **场景**

随着互联网应用的广泛普及，用户的行为记录越来越重要。现在的应用APP，保留用户行为采用了多种方法，无论是PV(页面浏览量),UV，ip,还是用户详细的动作指令，都纷纷记录在库中。为大数据的分析，绘制用户行为及画像，提供了精确的数据基础。给运营人员精确转化真实用户带来了明确目标，有了大量的数据作为可靠依据。

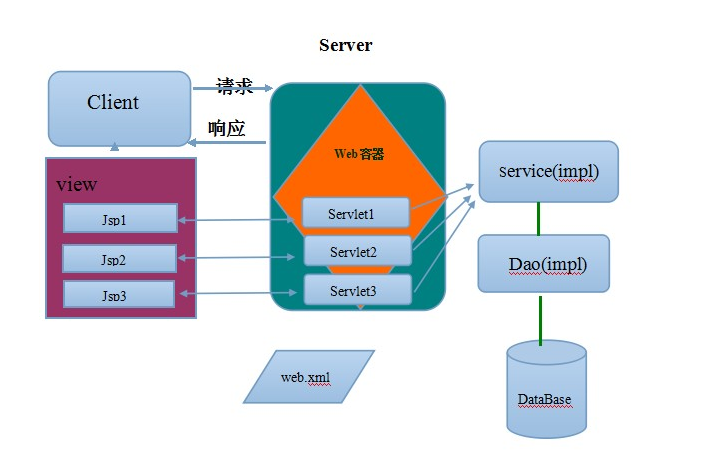
但每天产生几百万，几千万，甚至几十亿的PV无疑对数据库造成了相当高的负载，对于系统的稳定性和扩展性造成了极大的问题。此类数据的采集使用，又不像正常数据一样经常被查询，更新使用。通过集群方案，读写分离，负载均衡，控制了并发量，读取数据的速度，但是付出的代价是繁杂而且昂贵。这时候就需要一种简单而实用的框架来满足该需求。

pv的计算：当一个访问者访问的时候，记录他所访问的页面和对应的IP，然后确定这个IP今天访问了这个页面没有。如果你的网站到了23点，单纯IP有60万条的话，每个访问者平均访问了3个页面，那么pv表的记录就要有180万条。

什么是uv 　　uv(unique visitor)，指访问某个站点或点击某条新闻的不同[IP地址](http://www.so.com/s?q=IP%E5%9C%B0%E5%9D%80&ie=utf-8&src=wenda_link)的人数。 　　在同一天内，uv只记录第一次进入网站的具有独立IP的访问者，在同一天内再次访问该网站则不计数。独立IP访问者提供了一定时间内不同观众数量的统计指标，而没有反应出网站的全面活动。

1. 解决方案

1、普通 spring MVC的架构设计



service是定义接口，serviceimpl是定义接口的实现，是功能代码编写地方，它从DaoImpl接口中得到数据，而dao是对数据库进行操作的代码。简而言之，action调用serviceimpl，service实现serviceimpl，serviceimpl调用daoimpl，dao实现daoimpl。

JSP全名为Java Server Pages，中文名叫java服务器页面，其根本是一个简化的Servlet设计

[jsp](http://www.so.com/s?q=jsp&ie=utf-8&src=wenda_link)页面是[表示层](http://www.so.com/s?q=%E8%A1%A8%E7%A4%BA%E5%B1%82&ie=utf-8&src=wenda_link)View，DaoImpl是最后操作数据库的，如果你用[ibatis](http://www.so.com/s?q=ibatis&ie=utf-8&src=wenda_link)的话，用Service、ServiceImpl调用Dao、DaoImpl，然后数据库的操作语句的调用写在Dao、DaoImpl，最后真正数据库操作语句写在相应的xml中

web容器是一种服务程序，在服务器一个端口就有一个提供相应服务的程序，而这个程序就是处理从[客户端](http://baike.so.com/doc/4889711-5107810.html)发出的请求，如JAVA中的Tomcat容器，ASP的IIS或PWS都是这样的容器。一个服务器可以多个容器。

**Servlet**(Server Applet)，全称**Java Servlet**，未有中文译文。是用Java编写的服务器端程序。其主要功能在于交互式地浏览和修改数据，生成动态Web内容。狭义的Servlet是指Java语言实现的一个接口，广义的Servlet是指任何实现了这个Servlet接口的类，一般情况下，人们将Servlet理解为后者。

Servlet运行于支持Java的应用服务器中。从实现上讲，Servlet可以响应任何类型的请求，但绝大多数情况下Servlet只用来扩展基于HTTP协议的Web服务器。

最早支持Servlet标准的是JavaSoft的Java Web Server。此后，一些其它的基于Java的Web服务器开始支持标准的Servlet。

DispatcherServlet接口：（对应server）

        Spring提供的前端控制器，所有的请求都有经过它来统一分发。在DispatcherServlet将请求分发给Spring Controller之前，需要借助于Spring提供的HandlerMapping定位到具体的Controller。

HandlerMapping接口：

        能够完成客户请求到Controller映射。

Controller接口：

        需要为并发用户处理上述请求，因此实现Controller接口时，必须保证线程安全并且可重用。Controller将处理用户请求，这和Struts Action扮演的角色是一致的。一旦Controller处理完用户请求，则返回ModelAndView对象给DispatcherServlet前端控制器，ModelAndView中包含了模型（Model）和视图（View）。从宏观角度考虑，DispatcherServlet是整个Web应用的控制器；从微观考虑，Controller是单个Http请求处理过程中的控制器，而ModelAndView是Http请求过程中返回的模型（Model）和视图（View）。

ViewResolver接口：

        Spring提供的视图解析器（ViewResolver）在Web应用中查找View对象，从而将相应结果渲染给客户。

SpringMVC入门教程及其原理讲解和实例代码下载

SpringMVC框架介绍

Spring框架提供了构造Web应用程序的全能MVC模块。Spring MVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行制定。是一个标准的MVC框架。

Client:访问者， DAO:数据访问对象， DATE BASE 数据库 IMPL:初始微程序装载，servlet:小服务程序 redis 再区分 nginx 服务器

此类架构：一般为中小型的网站架构。在数据操作上都很少实现对数据库的IO。

数据量一旦上升，直接击穿容器和数据库，导致网站瘫痪。

1. 中型网站的架构设计

redis

Client

nginx

请求

service(impl)

Web容器

web.xml

servlet

view

分布式Server Server2 Server3

Dao(impl)

Write

DataBase

Read

DataBase

servlet

jsp

Jsp(JAVA服务器页面)

此分布式架构特点：从访问到数据库底层，以金字塔模型的方式，逐渐分解压力。在起初nginx做反向代理，轮询服务器以起到负载均衡的作用。在redis提供共享session(会议)，以及大数据量的频繁查询。给数据库减少了大量的负担。在数据库集群上的读写IO分离的情况下，保障数据安全，防灾等，起到很好的作用。然后，缺点在于 不是很常用的大量PV，用户操作等数据的保存，长期占用消耗整个数据库的资源，在维护上将花费大量的人力物力。

[反向代理](https://www.baidu.com/s?wd=%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmyRsmWDLuHf3nj01uWNB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nWDLPHDzPjmYPWn4nHRYr0)，是把一些静态资源存储在服务器上，当用户有请求的时候，就直接返回[反向代理](https://www.baidu.com/s?wd=%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmyRsmWDLuHf3nj01uWNB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nWDLPHDzPjmYPWn4nHRYr0)服务器上的资源给用户，而如果[反向代理](https://www.baidu.com/s?wd=%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmyRsmWDLuHf3nj01uWNB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nWDLPHDzPjmYPWn4nHRYr0)服务器上没有的资源，就转发给后面的[负载均衡服务器](https://www.baidu.com/s?wd=%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmyRsmWDLuHf3nj01uWNB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nWDLPHDzPjmYPWn4nHRYr0)，[负载均衡服务器](https://www.baidu.com/s?wd=%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvmyRsmWDLuHf3nj01uWNB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nWDLPHDzPjmYPWn4nHRYr0)再将请求分发给后端的web服务器。 区别就是：反向代理服务器是需要存储资源的，让用户更快速的接收到资源 负载均衡就是，为了保证后端web服务器的高可用，高并发，是不需要要存储资源，只需要转发用户的请求。

通常的[代理服务器](http://baike.baidu.com/view/751.htm)，只用于代理内部网络对Internet的连接请求，客户机必须指定代理服务器,并将本来要直接发送到Web服务器上的http请求发送到代理服务器中。由于外部网络上的[主机](http://baike.baidu.com/view/23880.htm)并不会配置并使用这个代理服务器，普通代理服务器也被设计为在Internet上搜寻多个不确定的服务器,而不是针对Internet上多个客户机的请求访问某一个固定的服务器，因此普通的Web代理服务器不支持外部对内部网络的访问请求。当一个代理服务器能够代理外部网络上的主机，访问内部网络时，这种代理服务的方式称为反向代理服务。此时代理服务器对外就表现为一个Web服务器，外部网络就可以简单把它当作一个标准的Web服务器而不需要特定的配置。不同之处在于，这个服务器没有保存任何网页的真实数据，所有的静态网页或者CGI程序，都保存在内部的Web服务器上。因此对反向[代理服务器](http://baike.baidu.com/view/751.htm)的攻击并不会使得网页信息遭到破坏，这样就增强了Web服务器的安全性。

1. mongodb的架构很好的解决了这个问题。

redis

Client

nginx

请求

service(impl)

Web容器

web.xml

servlet

view

Server Server2 Server3

servlet

jsp

jsp

Dao(impl)

PV

mongodb

Write

DataBase

Read

DataBase

za1

此mongoDB的架构中，很好的解决了不常用的数据快速存取。对mysql的数据库的压力进行了释放。主要的信息仍然用MySQL进行保存。PV,日志，动作详情等用mongodb进行保存。这样分库带来了，维护的便利。mongoDB又是文档型的NOSQL，缓存机制，数据分片。通过SQL索引查询有关信息。完全满足了需求。

**MySQL 是一个**[**关系型数据库**](http://baike.so.com/doc/5294416.html)**，**NoSQL，泛指非关系型的数据库

[关系数据库](http://www.so.com/s?q=%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93&ie=utf-8&src=wenda_link)的原理就是[二维表](http://www.so.com/s?q=%E4%BA%8C%E7%BB%B4%E8%A1%A8&ie=utf-8&src=wenda_link)非关系数据库有层次数据库、[网状数据库](http://www.so.com/s?q=%E7%BD%91%E7%8A%B6%E6%95%B0%E6%8D%AE%E5%BA%93&ie=utf-8&src=wenda_link)等大概就是这样

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

1. mongodb原理

介绍：mongodb是一个基于分布式文件存储的数据库，由C++语言编写，在为WEB应用提供可扩展的高性能数据存储解决方案。

特点：高性能、易部署、易使用，存储数据非常方便。

主要功能特性有：

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

使用原理

所谓“面向集合”，意思是数据被分组存储在数据集中，被称为一个集合。每个集合在数据库都有一个唯一的标识名，并且可以包含无限数目的文档。集合的概念类似关系型数据库里的表，不同的是它不需要定义任何模式。

模式自由，意味着对于存储在mongodb数据库中的文件，我们不需要知道它的任何结构定义。如果需要的话，你完全可以把不同结构的文件存储在同一个数据库里。存储在集合中的文档，被存储为键值对的形式。键用于唯一标识一个文档，为字符串类型，而值则可以是各种复杂的文件类型。我们称这种存储形式为BSON。

mongodb的缺点：无事物机制。

如果需要将mongodb作为后端db来代替mysql使用，即这里mysql与mongodb 属于平行级别，那么，这样的使用可能有以下几种情况的考量： (1)mongodb所负责部分以文档形式存储，能够有较好的代码亲和性，json格式的直接写入方便。(如日志之类) (2)从data models设计阶段就将原子性考虑于其中，无需事务之类的辅助。开发用如nodejs之类的语言来进行开发，对开发比较方便。 (3)mongodb本身的failover机制，无需使用如MHA之类的方式实现。

原子性:指整个事务是不可以分割的工作单元。只有事务中所有的操作执行成功，才算整个事务成功，事务中任何一个SQL语句执行失败，那么已经执行成功的SQL语句也必须撤销，数据库状态应该回到执行事务前的状态。

1. 将mongodb作为类似redis ，memcache来做缓存db，为mysql提供服务，或是后端日志收集分析。 考虑到mongodb属于nosql型数据库，sql语句与数据结构不如mysql那么亲和 ，也会有很多时候将mongodb做为辅助mysql而使用的类redis memcache 之类的缓存db来使用。 亦或是仅作日志收集分析。
2. 数据测试

在项目利用第3种架构，分布式发布服务3台。mysql数据库只需要主备。部署如上。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1台nginx+3台tomcat+mysql主备 |  |  |  |  |
| 总请求数量 | 并发用户数 | 每秒处理请求数(#/sec) | 平均请求等待时间(ms) | mysql (I/O)CPU占用 |
| 1500 | 50 | 45.16 | 1107.163 | 35% |
| 1500 | 100 | 49.64 | 2014.515 | 50% |
| 1500 | 200 | 47.20 | 4237.442 | 65% |
| 1500 | 300 | 47.44 | 6324.362 | 77% |
| 1500 | 400 | 47.39 | 8441.283 | 77% |
| 1500 | 500 | 47.73 | 10554.630 | 77% |
| 1500 | 600 | 47.88 | --- |  |

在用MySQL进行数据保存，在并发用户300度，连续发请求的时候，mysql的CPU占用就一直不变，也就是说IO已经到瓶颈了。后续来的请求占用连接池需要进行排队等候。因此处理速度就慢，占用资源也多。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1台nginx+3台tomcat+mysql主备+mongdb |  |  |  |  |
| 总请求数量 | 并发用户数 | 每秒处理请求数(#/sec) | 平均请求等待时间(ms) | mongodb ( I/O) CPU占用 |
| 1500 | 50 | 78.16 | 607.128 | 35% |
| 1500 | 100 | 109.01 | 1414.219 | 41% |
| 1500 | 200 | 125.20 | 2137.442 | 51% |
| 1500 | 300 | 160.44 | 3024.362 | 62% |
| 1500 | 400 | 207.39 | 4041.283 | 69% |
| 1500 | 500 | 249.73 | 6554.630 | 71% |
| 1500 | 600 | 356.18 | 8632.512 | 72% |

在mongodb进行数据保存的时候，发现处理请求数是MYSQL的5倍，在数据量暴增的时候，尤其明显。mongodb能够处理更多的数据。而且没有遇到很大的瓶颈。