视频网站实验技术报告

实验设计分析

整体设计

- 首先,对于整体的结构，本项目包含了两个前端，对应的两个后端以及一个独立的后端应用，具有一定的独立性,可以较好的进行系统维护和扩展。但是基于类似微服务的思想，后端应用仍然可以进一步的细分，在本实验的实现过程，视频的信息，图片以及文件本身几乎都是绑定在一起管理，在本实验中文件占据主要地位，信息和图片属于少量附加信息因此弊端不是特别明显。但如果附属数据增加，视频和附属数据的压力均由相同的服务器来承担会大大增加系统复杂度，减少系统效率。例如现在视频网站都会具有的缩略图效果，各种各样的弹幕，这些效果与视频本身并不具有完全的绑定关系，完全可以使用不同的服务，部署在不同的机器上，不仅可以分摊压力而且可以降低维护难度，使得每个服务只要专注于所要关注的功能即可。
- 其次，对于系统响应速度，一个系统想要提升响应速度几乎都是考虑以下几点，系统处理的优化，服务器扩展以及缓存机制。其中关于系统处理优化，本实验多使用了一些第三方软件，相比自己编码要进行了部分优化，对于专业的互联网设计应有针对自身特点更进一步的系统优化，而且还可以考虑尽量压缩或分摊数据，降低web平均传输数据量，进行代码压缩，多采用css描述使用ajax以及其他针对浏览器数据传输[1]，以及线程和线程池，减少web服务请求周期[2]等技术以及其他方面的优化。关于服务器扩展，软件设计相对独立，互相之间工作没有影响，不需要特定的数据传输协调工作，因此都可以独立进行扩展。最后即为缓存机制，本实验没有采取缓存机制，一是本实验计算密集型任务较少，二是最主要的计算任务--视频压制--的计算结果几乎无法进行复用，将其结果缓存意义较少，不过仍有可以考虑的缓存方法，CDN[3]就是其中之一。CDN的全称是 Content Distribution Network，即内容分发网络，不同于传统的缓存但是基于缓存的思想，就是将常访问的内容放置在能快速访问到的位置。CDN采用的是将一片区域的用户经常访问的内容缓存在离此区域最近的服务端，其工作流程也大致于缓存技术相同，用户访问的url会被dns解析指向到一个全局负载均衡，全局均衡器会返回给用户一个局部均衡器，最终选择一台服务器为用户提供服务，如果服务器没有所需服务则向源服务器请求，并根据缓存策略选择是否缓存此数据[4]。CDN相当于将地方服务器作为源服务器的缓存，存储在用户附近，在互联网设计中CDN可以大大改善用户体验，降低响应速度，并且可以进行一定的压力分摊，增加系统受压能力。

具体设计

- 文件与信息上传存储，文件存储使用了nfs，网络文件系统，与服务器运行状态无关，属于一个独立的文件存储系统，可以进行单独的管理维护，但是存储本身仍然采取了直接写入文件的方式，而且文件备份和播放文件本身存储在同一机器上备份的效果较差。大型互联网应用往往采用分布式的文件和数据存储，例如Yutube采用了Google的big table的分布式数据存储。
- 视频播放和图片传输，采用将上传的文件转换为静态文件，直接进行传输播放，设计较为简单，难以承受各种各样的访问压力，可以使用其余第三方的web服务器来提供视频服务，例如使用lighttpd，lighttpd是一个资源占用非常低的web服务器软件，目的是提供一个安全、快速、兼容性好并且灵活的web server环境[5], 其在传输静态文件时有较快的传输速度且资源占用较少，可以作为视频以及图片传输服务的重要工具。
- 信息传递和存储， 使用rabbitmq进行少量的信息传递，使得两个系统相互独立，而且采用被压式系统设计，将压力主要转移给存储少量信息的rabbitmq，而不是cpu占用率较高的解码器，具有良好的压力均衡。并且可以在访问url中加入负载均衡器，降低了单个服务器的访问数量，使得水平扩展能的到更好的效果。信息存储采用了mysql，具有良好的性能和效果。

大规模用户下系统设计

系统架构的演变[6]

在软件系统发展过程中，用户的数量正在不断的增加，为了满足日益增长的数据量访问，软件系统架构也发生了很大的变化，从集中式架构，到垂直拆分，分布式服务，服务治理，再到微服务，这些变化都是为了更好的满足大规模的用户量访问。

- 集中式架构：所有功能集中在一起，只需要一个应用，容易部署。缺点是难以开发维护，对于特定功能难以进行针对性的改进和优化，水平扩展较难。
- 垂直拆分：对任务进行拆分，形成独立的模块，方便了扩展，分担了负载压力，但是容易造成重复开发
- 分布式服务：随着程序复杂度升高，拆分功能使得应用之间交互不可避免，将核心业务抽取出来，作为独立的服务，逐渐形成稳定的服务中心，同时也增加了耦合度，分布式调用使得关系复杂较难维护
- 服务治理（SOA）：当服务不断增加时，小服务增多，使得系统利用率和性能评估效果下降，需要添加一个资源管理和调度中心，进行服务注册和服务管理
- 微服务：将服务进行更细粒度的拆分，只需关心服务暴露的接口，相互独立，互不干扰。

大型网站框架[7]

首先观察一下Yutube的数据量，"一天的YouTube流量相当于发送750亿封电子邮件.", 2006 年中就有消息说每日 PV 超过 1 亿,现在? 更夸张了,"每天有10亿次下载以及6,5000次上传"[8],在这个2011年的文章里是这样写的。不难想象又经过了大约十年的发展，其数据量达到了何种地步，因此其框架设计也更值得我们研究。其中很主要的思想就是切分服务，多采用合适的第三方服务。

- 视频服务：采用了“迷你集群”，每个视频都有多个机器存储，在线备份，而且多硬盘具有更快的速度。使用了第三方的lighttpd进行视频服务，使用CDN来分布式缓存各种常访问的资源。
- 缩略图：缩略图无论是数量还是访问次数都远远大于视频，对服务端带来的压力很大，因此Yutube采用了将此功能分离出去，使用单独的服务器分摊压力，并且做了专门的cache和硬盘io优化，使用分布式数据存储系统big table管理小文件达到更快的访问效果。
- 数据库：早期采用mysql具有很多问题，主节点和工作节点是异步的，主节点为大机器多线程可以快速处理多个任务，备份异步导致子节点往往跟不上主节点，而且更新会导致缓存失效。后期采用了分区的思想，将数据库分成shards，不同的用户指定到不同的shards ，扩散读写。

整体来看就是建立多级的划分系统，不断的划分服务，以用户聚集为区域，将服务划分到用户附近，数据也进行不断的划分，使得扩展更新影响造成的波动尽可能地减少。

参考文献：

[1] 陈晓林. 中小型Web服务器系统优化策略[J]. 电脑知识与技术(学术交流), 2007(02):338-339.

[2]鲍剑洋, 沈群. Web服务器性能优化[J]. 计算机工程与应用, 1999, 035(006):81-85.

[3]Peng G . CDN: Content Distribution Network[J]. Research Proficiency Exam Report, 2003:1-6.

[4]程序员都应了解的 CDN 是什么？ https://mp.weixin.qq.com/s?src=11×tamp=1594449289&ver=2453&signature=QUf7of3KrRBDhGsZY8tMVg1GLYQK6TB71GpLEf2fYs1f940RNXfkn*QlYpQxHppmVjwoBEBAXo3Qf5RZpAdd2IWRcWTmj67H7627Mfqp95yUCYmysI7R5LIrgT0UAoz8&new=1

[5]搜狗百科-lighttpd https://baike.sogou.com/v3839570.htm?fromTitle=Lighttpd

[6]系统架构演变 https://blog.csdn.net/qq_41234832/article/details/84637227

[7]大型视频网站YOUTUBE的技术架构 https://blog.csdn.net/iteye_9806/article/details/81898952

[8]大型视频网站的技术架构方案 https://blog.csdn.net/iteye_18051/article/details/82200276