

调研报告

一·概述

维基百科是一个基于维基技术的多语言百科全书写作计划，用多种语言编写的网络百科全书，所谓的维基技术是Ward Cunningham在1995年为了方便模式社群的交流建立的一个工具，作为一种超文本系统支持面向社群的协作式写作，可以基于Web对wiki文本进行浏览，创建和更改，同时Wiki系统还支持面向社群的协作式写作，最后所有对Wiki进行编辑的作者自然的构成了一个社群，Wiki就是这个社群简单的交流工具。

维基百科是一个基于维基技术的全球性多语言百科全书协作计划，同时也是一部用多种语言编成的网络百科全书，其目标及宗旨是为全人类提供自由的百科全书——用他们所选择的语言来书写而成的，是一个动态的、可自由访问和编辑的全球知识体。并且在许多国家相当普及。其口号为“维基百科，自由的百科全书。”

二 详细介绍

2.1 支持语言

维基百科作为一个全球性的百科全书，支持280种语言对词条的编辑，正是由于它的全球性，会导致来自全球各地的贡献者可能在浏览相同语言版本的维基百科时却使用不同的方言，又或者受到不同国家的习惯用语影响而使得彼此用语出现些微差异，或者是受到不同地点社会环境的影响下使得对于条目内容的观点不一。另外尽管在各种语言版本的维基百科之中也有如同“中立的观点”般普遍施行的方针，然而许多语言版本的维基百科仍然必须依照自己主要使用用户的国家其法律限制，这使得各个维基百科计划在方针和作法上并非一致。其中最为明显的例子，则是各个维基百科必须依照相关法律限制决定是否能够根据许可而采纳自由内容或者合理使用的内容。

2.2 涉及范围

维基百科涉及到的内容包括传记，历史，地理，社会，文化，科学，技术，食品和数学等主题的基本内容。且用户创建的某些词条可能经过一定时间发展成为维基百科的“优良条目”，“特色条目”。优

良条目和特色条目是通过投票选举产生。大多数特色条目是由优良条目升级达到的。

2.3 传播媒介

由于维基百科的内容是根据开放式许可来作为许可协议，这意味着任何人都可以重复发表维基百科内的文章，同时重新发布内容时也不需要付费。除了维基百科自身网站之外，维基百科的内容不论是于在线或者是脱机都曾经使用不同媒体来向外发表，这包括有：

网站内容：数以千计的“镜像网站”纷纷再次发布维基百科的条目内容，有些网站甚至也能够提供维基百科中所引用的参考来源。另外在维基百科推出与移动设备友好界面前，另外有非官方的Wapedia来传播维基百科的内容。

应用程序：维基百科本身也推出许多于移动设备上运作的应用程序来推广维基百科的内容，包括Android和iOS等用户都能够借此浏览维基百科的条目。

搜索引擎：一些互联网搜索引擎便会借由Powerset等技术，以在自身搜索结果中能够将维基百科的内容列为特殊选项，这其中包括有Bing和DuckDuckgo等。

各式光盘：维基百科已经多次将自身条目集结推出光盘。

书籍刊物：许多维基百科的用户期望能够将自身的条目内容集结，并且尝试以传统出版书籍的方式来推广。其中包括美国出版公司Books LLC以及德国出版商VDM Publishing便曾与维基百科进行合作，而自2009年以来包括英语、德语、俄语、法语维基百科的用户纷纷将自身条目采用随选打印的方式出版各式书籍刊物。

语义查询：自2007年开始，DBpedia开始自英语维基百科的“infoboxes”和“页面分类”提取数据数据，借由资源描述框架的技术创建可供查询的语义网计划。同时直接通过包括Semantic MediaWiki等转换技术的协助，使得维基百科的数据数据能够在网站上扩展成为可供阅读的语句。而这些导往其他网站的数据信息也能够重新使用于维基百科计划，除了能够在同一语言版本的维基百科文章之中引用，同时也能够提供信息给其他不同语言版本的维基百科使用。

然而维基百科对于其他网站重新使用维基百科的完整条目并非完全鼓励，特别是反对他人直接通过网络蜘蛛来复制文章内容。然而这些以

数据库转存储技术过去只能够转移维基百科的文章内容，一直到2007年以后才有相关技术能够同时给予图像转移的服务。另外数个语言版本的维基百科也多设置有“询问处”的服务，由志愿者借由维基百科内的条目来回答其他网络用户所提出的各种问题。

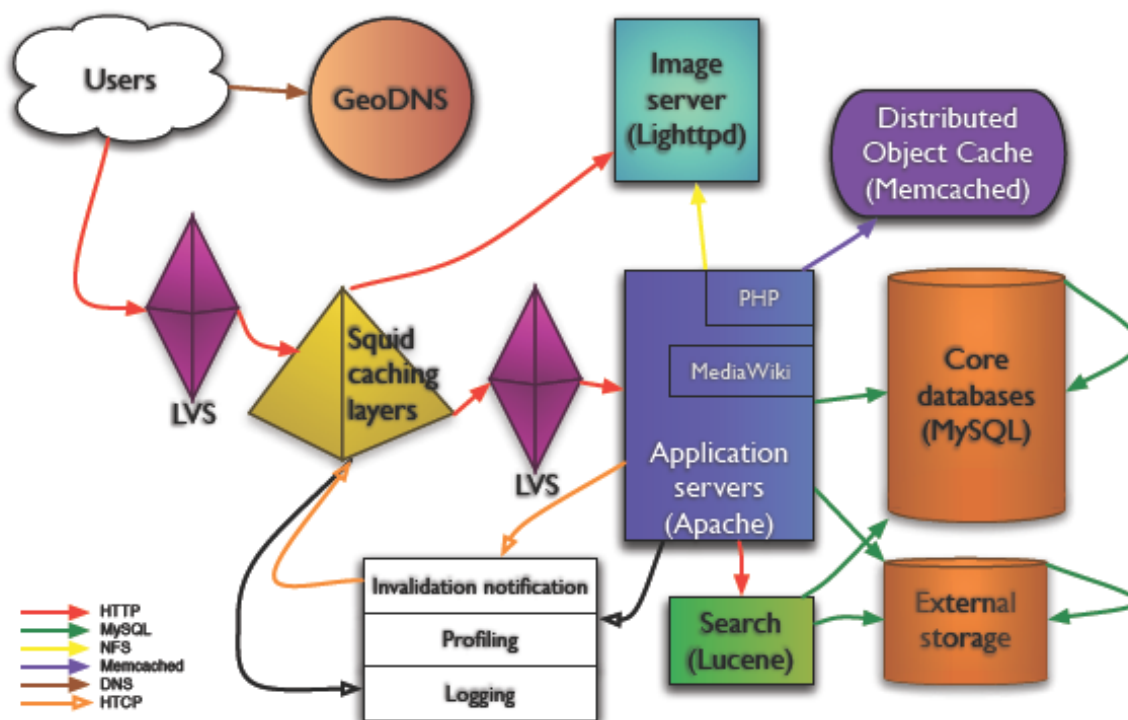
2.4 软硬件技术

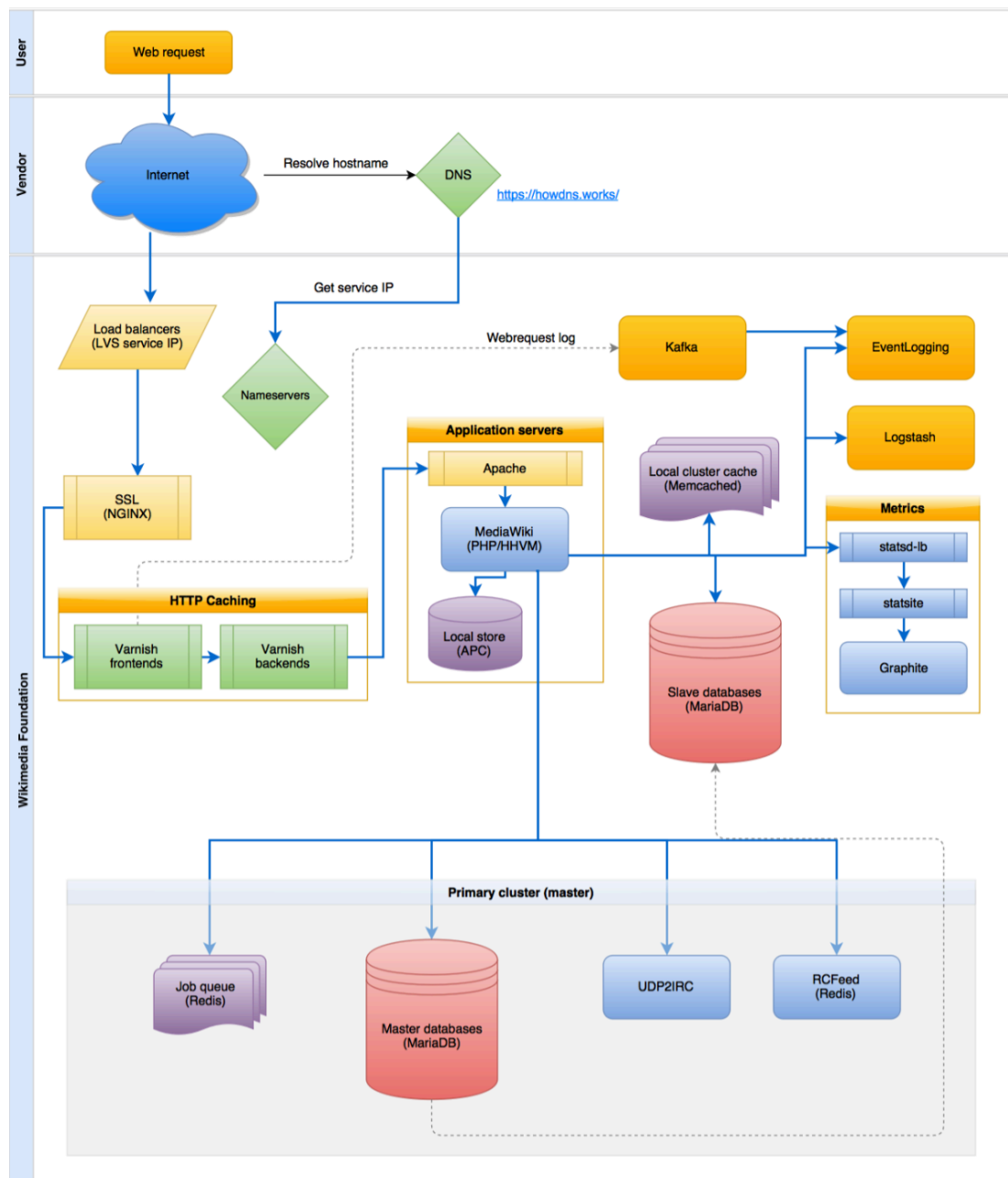
维基百科开始的时候使用PHP+mysql实现，后来由于使用的人流增加，网站速度明显下降，几乎到了连文章都无法编辑的地步，所以整个程序被重写，使用新的软件运行程序以及数据库，这个新的程序被称为MediaWiki，数据表示在峰值每秒钟有3万个HTTP请求。

2.5 审核机制

维基百科的大部分的内容，由一般的维基人讨论、修改，通常为民主的形式。维基百科的系统里同时有资深的维基人担当管理员，负责清除破坏及封锁恶意破坏者的帐户。

三 维基百科的架构介绍





如上图是维基百科的技术架构图和系统架构图。

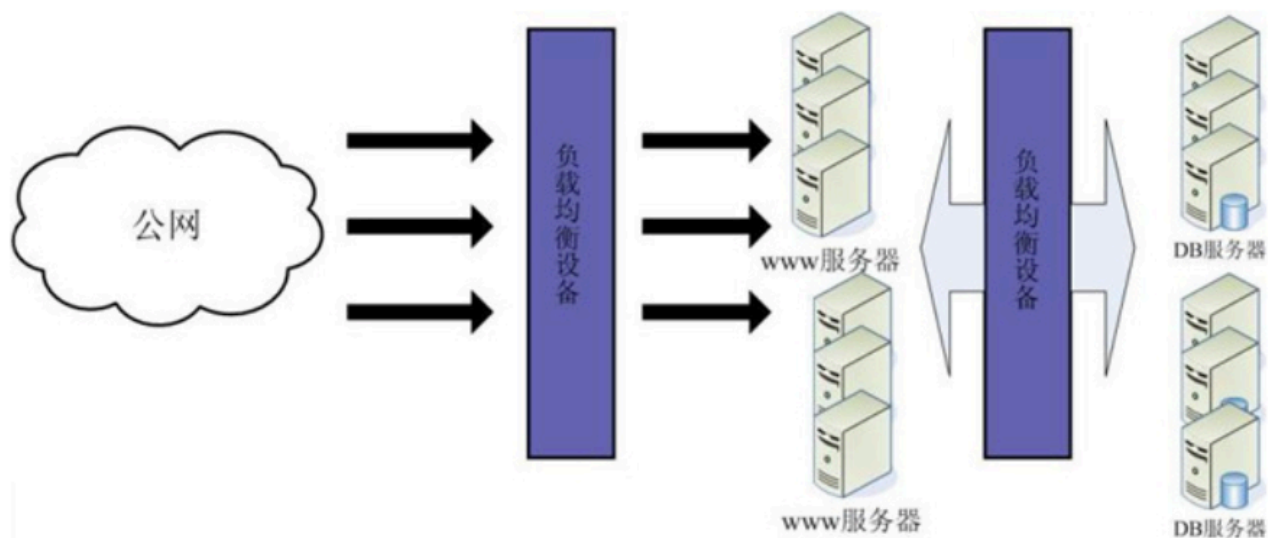
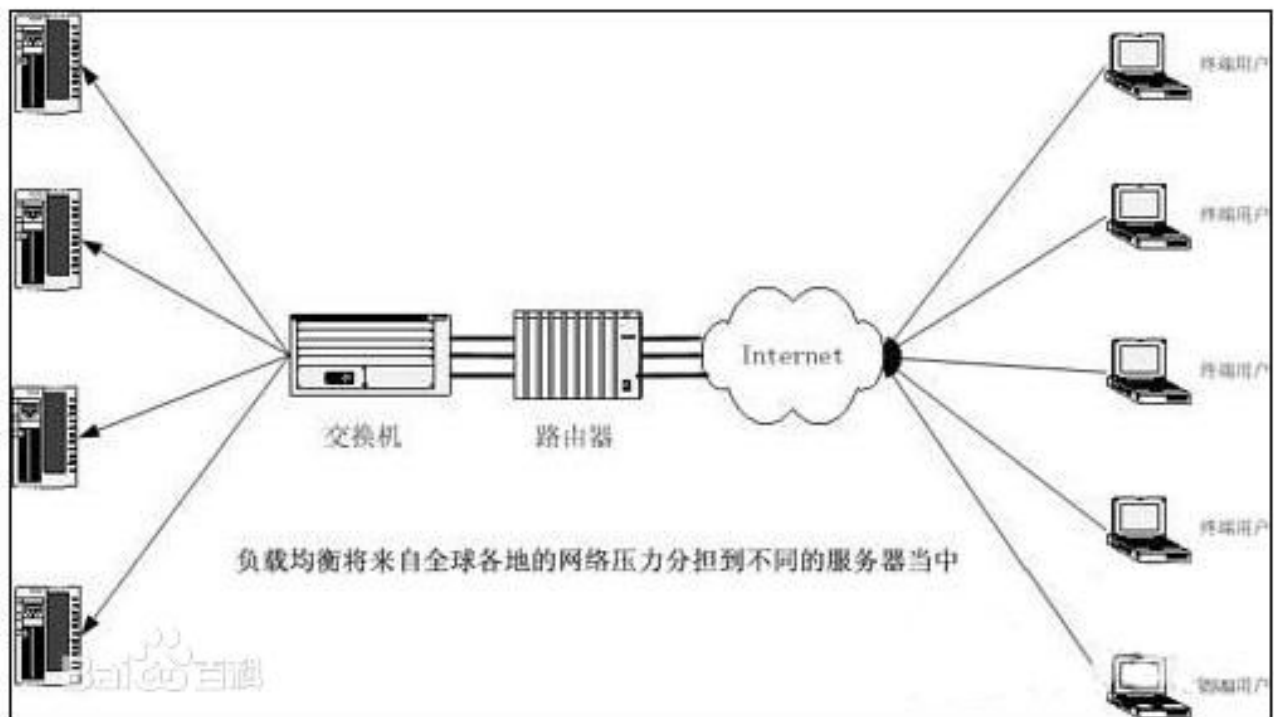
3.1 GeoDNSA

基于开源域名服务器软件BIND(Berkeley Internet Name Domain) 的增强版本，可将域名解析到离用户最近的服务器。所以GeoDNSA在维基百科架构中的任务是把用户带到最近的服务器。

3.2 LVS

基于Linux的开源负载均衡服务器，所谓负载均衡就是建立在现有网络结构之上提供的一种廉价有效透明的方法扩展网络设备和服务器的带宽，

增加吞吐量，加强网络数据处理能力，提高网络的灵活性和可用性。由多台服务器以对称的方式组成一个服务器集合，每台服务器都具有等价的地位，都可以单独对外提供服务，通过负载分担技术，将外部发送来的请求均匀分配到对称结构中的某一台服务器上，而接收到请求的服务器独立地回应客户的请求。均衡负载能够平均分配客户请求到服务器列阵，以此提供快速获取重要数据，解决大量并发访问服务问题。

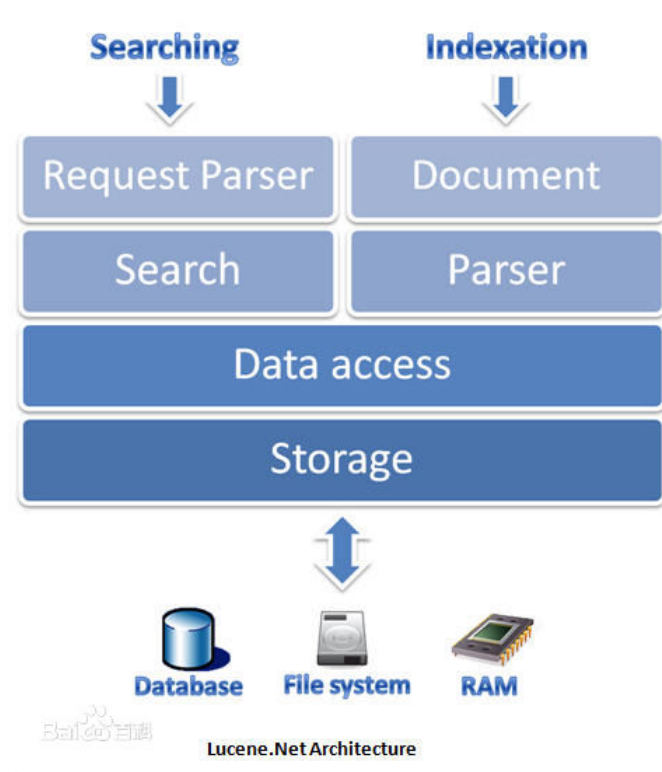
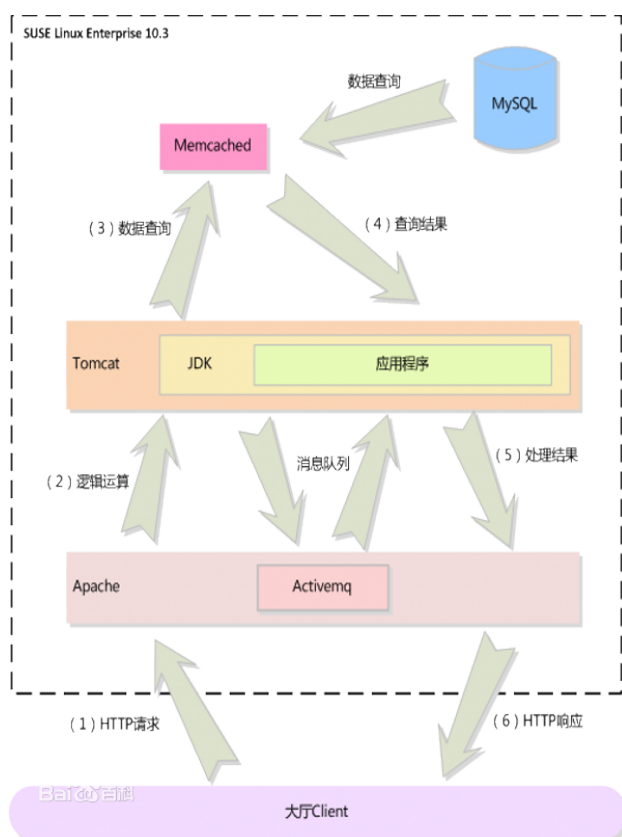


3.3 图片服务器 Lighttpd

就目前来说Lighttpd已经成为了标准的图片服务器的，作为一个开源的Web服务器软件，齐根本目的是提供一个专门针对高性能网站，安全，快速，兼容性海并且灵活的Web server环境，具有非常低的内存开销、cpu占用率低、效能好以及丰富的模块等特点。

3.4 Wiki软件：MediaWiki

MediaWiki就是在前面介绍中说到的后来重写的程序，运行于PHP+MYSQL环境，是建立wiki网站的首选后台程序，可以直接使用MediaWiki来建立自己的知识库。



3.5 Memcached

维基百科成功的关键要素就是Cache，Memcached作为维基百科所用的cache的一种，是一个高性能的分布式内存对象缓存系统，用于动态Web应用以减轻数据库负载。它通过在内存中缓存数据和对象来减少读取数据库的次数，从而提高动态、数据库驱动网站的速度。Memcached基于一个存储键/值对的hashmap。

3.6 Lucene

在维基百科中主要用于搜索，Lucene是apache出品的一个开放源代码的全文检索引擎工具包，为开发人员提供一个简单的工具包，以方便在目标系统中实现全文检索的功能。

3.7 Squid

基于Linux的开源反向代理软件，是一个高性能的代理缓存服务器，Squid支持FTP、gopher和HTTP协议。和一般的代理缓存软件不同，Squid用一个单独的、非模块化的、I/O驱动的进程来处理所有的客户端请求。缓存在实际的应用中意义很大，好的缓存技术会提高命中率，来增加访问速度和减少资源的占用率。

3.8 CDN技术

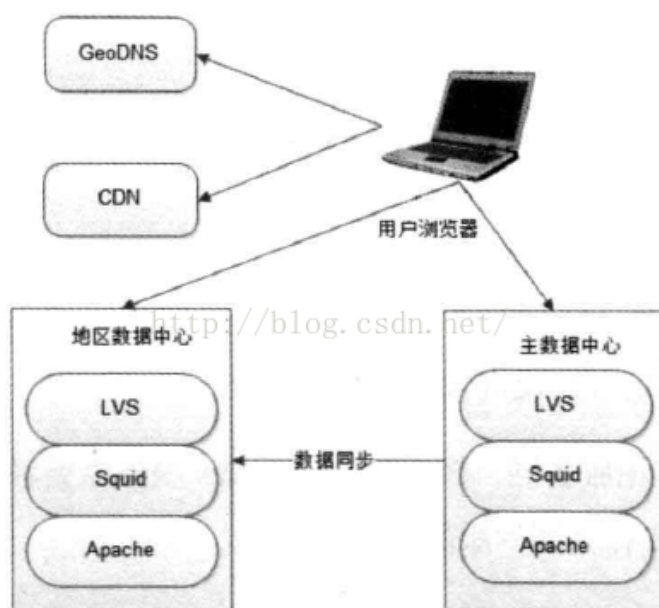
CDN是构建在网络之上的内容分发网络，依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率。CDN的关键技术主要有内容存储和分发技术。CDN的基本原理是广泛采用各种缓存服务器，将这些缓存服务器分布到用户访问相对集中的地区或网络中，在用户访问网站时，利用全局负载技术将用户的访问指向距离最近的工作正常的缓存服务器上，由缓存服务器直接响应用户请求。

维基在美国、荷兰和韩国分别设有群集，维基CDN会根据来访IP位置的不同选择指向最近的群集。

数据存储：元数据，比如文章修改历史，文章的链接和用户资料等内容被存放于主数据库；正文存于外部存储；用户上传的图片等信息则单独存放于图片服务器。主数据库服务器一共有15台，配置为内存4GB~16GB，6块73~146GB的硬盘和双CPU。数据库中除了有一个主数据库，还有许多复制的从数据库，这些主从数据库并不是按照服务器个数来划分的，数据库都是跨服务器运行的。至于外部存储，维基将文章正文存到外部存储群集。维基目前采用了3个复制的MySQL群集来负责外部存储。这样可以节省主数据库的昂贵空间。为了节省费用，那些应用服务器的剩余空间也被用来存储。此外，图片服务器单独存放了1.3TB的图片和另外4百万个文件。

四 维基百科的性能优化策略

所谓网站前端是指应用服务器(也就是PHP服务器)之前的部分,包括DNS服务、CDN服务、反向代理服务、静态资源服务等。对wikipedia而言,80%以上的用户请求可以通过前端服务返回,请求根本不会达到应用服务器,这也就使得网站最复杂、最有挑战的应用服务端和存储端压力骤减。



维基百科的前端架构的核心是反向代理服务器Squid集群,大约部署有数十台服务器,请求通过LVS负载均衡地分发到每台Squid服务器,热点词条缓存在这里,大量请求可直接返回响应,请求无需发送到Apache服务器,减轻应用负载压力。Squid缓存不能命中的请求再通过LVS发送到Apache应用服务器集群,如果有词条信息更新,应用服务器使用Invalidation Notification服务通知Squid缓存失效,重新访问应用服务器更新词条。

而在反向代理Squid之前,则是被维基百科技术团队称为"圣杯"的CDN服务,CDN服务对于维基百科性能优化居功至伟。因为用户查询的词条大部分集中在比重很小的热点词条上,将这些词条内容页面缓存在CDN返回,响应速度非常快,这些请求甚至根本不会到达维基百科数据中心的Squid服务器,服务器压力减小,节省的资源可以更快地处理其他未被CDN缓存的请求。

维基百科 CDN缓存的几条准则为：

- 1.内容页面不包含动态信息，以免页面内容缓存失效或者包含过时信息。
- 2.每个内容页面有唯一的REST风格的URL，以便CDN快速查找并避免重复缓存。
- 3.在HTML响应头写入缓存控制信息，通过应用控制内容是否缓存及缓存有效期等。

五 维基百科服务端性能优化策略

服务端主要是PHP服务器，这里是网站业务逻辑的核心部分，运行的模块都比较复杂笨重，需要消耗较多的资源，wikipedia 将最好的服务器部署在这里（和数据库配置一样的服务器），从硬件上改善性能。

除了硬件改善，维基百科还使用许多其他开源组件对应层进行如下优化。

- 1.使用APC，这是一个PHP字节码缓存模块，可以加速代码执行减少资源消耗。
- 2.使用Imagemagick进行图片处理和转化。
- 3.使用Tex进行文本格式化，特别是将科学公式内容转化成图片格式。
- 4.替换PHP的字符串查找函数strtr(), 使用更优化的算法重构。

六 维基百科后端性能优化策略

包括缓存、存储、数据库等被应用服务器依赖的服务都可以归类为后端服务。后端服务通常是一些有状态的服务，即需要提供数据存储服务，这些服务大多建立在网络通信和磁盘操作基础上，是性能的瓶颈，也是性能优化的重灾区。

后端优化最主要的手段是使用缓存，将热点数据缓存在分布式缓存系统的内存中，加速应用服务器的数据读操作速度，减轻存储和数据库服务器的负载。维基百科的缓存使用策略如下：

- 1.热点特别集中的数据直接缓存到应用服务器的本地内存中，因为要占用应用服务器的内存且每台服务器都需要重复缓存这些数据，因此这些数据量很小，但是读取频率极高。

- 2.缓存数据的内容尽量是应用服务器可以直接使用的格式，比如HTML格式，以减少应用服务器从缓存中获取数据后解析构造数据的代价。
- 3.使用缓存服务器存储session对象。
- 4.相比数据库，Memcached的持久化连接非常廉价，如有需要就创建一个Memcached连接。

作为存储核心资产的MySQL数据库，维基百科也做了如下优化：

- 1.使用较大的服务器内存。在wikipedia应用场景中，增加内存比增加其他资源更能改善MySQL性能。
- 2.使用RAID0磁盘阵列以加速磁盘访问，RAID0虽然加速磁盘访问，但是却降低了数据库持久可靠性（一块盘坏了，整个数据库的数据都不完整了）。显然维基百科认为性能问题迫在眉睫，而数据可靠性问题可以通过其他手段解决（如MySQL主从复制，数据一步备份等）。
- 3.将数据库事务一致性设置在较低水平，加快系统恢复速度；
- 4.如果Master数据库死机，立即将应用切换到Slave数据库，同事关闭数据写服务，这意味着关闭词条编辑功能。维基百科通过约束业务获得更大的技术方案选择余地，很多时候业务后退一小步，技术就可以前进一大步。

