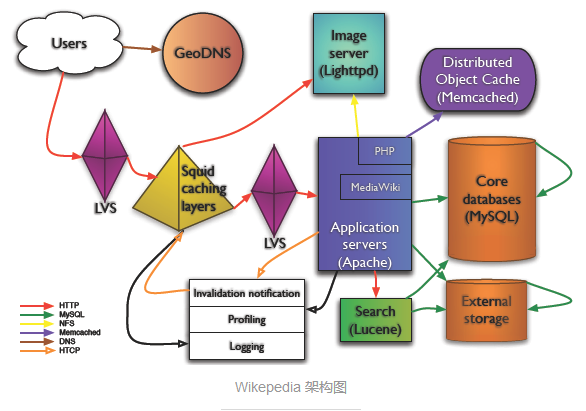
# 维基百科架构分析

1. 架构简单介绍  
   Wikipedia网站建立在LAMP（Linux+apache+MySQL+PHP）之上，其它技术组件也基本全部采用免费的开源软件。Wikipedia由于其非盈利性，所以不得不榨尽系统所有资源的利用价值，用最少的资源实现复杂功能，其结构如下图：  
     
   架构主要组成部分如下：  
   GeoDNS：基于开源域名服务器软件BIND的增强版本，可将域名解析到离用户最近的服务器。  
   LVS：基于Linux的开源负载均衡服务器。  
   Squid：基于Linux的开源反向代理服务器。  
   Lighttpd：开源应用服务器，较主流的Apache服务器更轻量、更快速。许多网站使用Lighttpd作为图片服务器。  
   PHP：免费的web应用程序开发语言。  
   Memcached：无中心高性能的开源分布式缓存系统，稳定、可靠、历久弥新，是网站分布式缓存服务必备的。  
   Lucene：由Apache出品的，Java开发的开源全文搜索引擎。  
   MySQL：开源的关系型数据库管理系统，虽被Oracle收购，但开源社区将其继续开源发展的决心不动摇。
2. 架构性能分析
3. 前端架构核心——反向代理服务器squid集群。Squid集群，大约部署了数十台服务器，请求通过LVS负载均衡地发布到每台Squid服务器，热点词条被缓存在这里，大量请求可直接响应，请求不用发送到Apache服务器，减轻应用负载压力，不能命中，则再发送请求。如词条有更新，那么应用服务器会使用Invalidation Notification服务通知Squid缓存失效，重新访问应用服务器更新词条。
4. “圣杯”——CND服务。CND服务位于反向代理Squid之前，用户查询词条大部分集中在比重很小的热点词条，没错，CND主要存储的词条，就是那些。此外，CND离用户浏览器最近，这样能保证响应速度，同时减轻squid的压力，节约资源，以更快处理CND未能处理的请求。
5. 服务器端性能优化。服务器端是业务逻辑的核心部分，处理事物复杂，需耗更多的资源，这里也配置最好的服务器，硬件改善性能，同时还优化应用层。
6. 使用APC——PHP的字节码缓存模块，加速代码执行，减少资源损耗。
7. 使用Imagemagick进行图片处理和转化。
8. 使用Tex进行文本格式化，尤其是将科学公式内容转化成图片格式。
9. 重构PHP的字符串查找函数strtr()。
10. 后端性能优化。主要使用手段是使用缓存，将热点数据缓存在分布式系统的内存中，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。主要做法如下：

* 读取率极高的数据直接缓存到应用服务器的本地内存中，能保证快速的访问。
* 缓存数据尽可能采用应用服务器可直接使用的格式，以减少解析或解码的代价。
* 使用缓存服务器存储session对象。

1. 存储核心——MySQL数据库优化。

* 使用较大服务器内存。增加内存，能够最大限度的改善MySQL性能。
* 采用RAID0磁盘阵列，虽然降低了数据的持久可靠性，但却解决了性能问题。此外，可靠性能通过其它手段解决，例如：MySQL主从复制、数据异步备份等。
* 将数据库事务一致性设置在较低水平，加快宕机恢复速度。
* 约束业务，这将能获得更多的技术方案。例如在master数据库宕机时，立即将应用切换到salve数据库，同时关闭数据写功能。

综上分析，Wikipedia架构中保证性能主要采用了类似计算机系统中的cache机制，其使用到的策略有以下几个：增加资源、维护多个计算副本和维护多个数据副本的策略、减少日常开销的策略、调度资源策略、增加资源有效性。