LNMP架构：linux + Nginx + mysql + php；在静态页面上处理，nginx完胜apache，基本可以超出一倍以上，对于高性能，高并发来说，nginx可以针对100000并发（实际可能要低些），nginx的io机制使用的是epoll，而apache采用的是select机制。Nginx--php-fpm—mysql，替换之前的apache—mod-php—mysql；

Nginx的安装：使用干净的系统，卸载之前的apache，要编译nginx先要安装编译软件，然后下载stable version的源码，./configure，make，

在configure过程中“./configure: error: the HTTP rewrite module requires the PCRE library.”，可以先安装pcre，pcre库是实现正则表达式的基础，如果缺少此库，nginx无法支持HTTP中的URL重写功能。如果你不需要此功能，可以在执行编译配置脚本时加入“--without-http\_rewrite\_module”，还有zlib，这个记着去官网下源码编译，我就出错在zlib的1.2.3，而版本已经更新到1.2.8，而且pkgs上居然也是1.2.3。。。

安装的nginx在/usr/local/nginx目录下，命令文件/sbin，配置nginx.conf， 然后启动./nginx，what，又出错了：Nginx: error while loading shared libraries: libpcre.so.1，这个主要是加载的时候找不到libpcre.so.1，我们在这里做一个链接即可，在/lib目录下可以发现：libpcre.so.0  libpcre.so.0.0.1，这里我们将libpcre.so.1作为libpcre.so.0.0.1的链接然后执行nginx，就可以正常访问了。

Php-fpm的安装：先下载php的源代码，编译前的工作，安装repoforge的yum源，yum clean all，yum list，刷新yum源；卸载php相关程序，

Yum list | grep php，一般安装的话我们要卸载yum remove php php-devel php-mysql，然后安装库文件（编译需要），yum install –y libxml2-devel libjpeg-devel libpng-devel freetype-devel openssl-devel libcurl-devel libmcrypt-devel，安装mysql，然后进入下载的php，远程登录，在configure的时候注意参数，安装的路径/usr/local/php，查看sbin，执行文件，关于配置文件，我们需要一个php.ini文件，可以从源代码的php-ini-production获得，然后重命名这两个文件，启动php-fpm, 我们下面要做的就是配置nginx.conf，加入对php-fpm的支持，php scripts to FastCGI，把注释取消，把/scripts改成我们网站根目录的地址/usr/local/nginx/html，还有一点要注意，将localtion下的index加上index.php，并且chmod –R 777 html/，然后killall nginx，重新启动nginx，创建index.php，查看能否正常运行。

LNMP性能测评：搭建wordpress，对其进行测试（ab），可以发现requests per second有所提高但并不是很高。

使用eAccelerator用来加速lnmp的站点：下载解压缩eAccelerator的源文件，然后使用/usr/local/php/bin/phpize创建configure文件（无法找到php-conf的位置可以手工指定/usr/local/php/bin/php-config），然后编译安装，重新配置php.ini文件，加入eAccelerator的配置即可，killall nginx，php-fpm，之后重启；查看eAccelerator是否安装上了。（这里我是用的是php5.5.10，但貌似5.5以后不能使用eAccelerator了，好悲催，现在php-fpm都不能启动了，只能把前面php.ini中的eAccelerator注释掉）。

优化php-fpm：php-fpm.conf可以修改端口号，可以设置pm.max\_children = 5+，pm.start\_server=2+，还有其他的也可以调整，然后重启php-fpm。这里主要改善了延迟的时间，requests per second 没有多大的提高。