PHP相关概念和配置

客户端动态和服务端动态，如JavaScript和java

编程语言：

静态型语言：编译型语言

C， C++, JAVA(又编译又解释，先编译成class字节码，边执行边解释)

动态型语言：解释性语言

bash， shell, perl, python

对比：

编译型语言错误追踪，调试没有解释性语言好，更适合开发底层软件，实时性和性能要求高，如操作系统，驱动程序等，数据库系统等，做的很底层，直接操作硬件，以至于用户接口没有解释性语言做得好。

解释性语言有各式各样的轮子，拿来用。因为他最接近用户上层。但是像C , C++ 做得过于底层，和硬件打交道，轮子就不是那么丰富，而且我做的轮子别人未必可以用。

编译语言每改动一次都需要重新生成编译，相对开发周期长，维护成本大。

他们各自有各自的引用场景。

PHP is Hypertext preprocessor

## 一、PHP简介

PHP是通用服务器端脚本编程语言，其主要用于web开发以实现动态web页面，它也是最早实现将脚本嵌入HTML源码文档中的服务器端脚本语言之一。同时，php还提供了一个命令行接口，因此，其也可以在大多数系统上作为一个独立的shell来使用。

Rasmus Lerdorf于1994年开始开发PHP，它是初是一组被Rasmus Lerdorf称作“Personal Home Page Tool” 的Perl脚本， 这些脚本可以用于显示作者的简历并记录用户对其网站的访问。后来，Rasmus Lerdorf使用C语言将这些Perl脚本重写为CGI程序，还为其增加了运行Web forms的能力以及与数据库交互的特性，并将其重命名为“Personal Home Page/Forms Interpreter”或“PHP/FI”。此时，PHP/FI已经可以用于开发简单的动态web程序了，这即是PHP 1.0。1995年6月，Rasmus Lerdorf把它的PHP发布于comp.infosystems.www.authoring.cgi Usenet讨论组，从此PHP开始走进人们的视野。1997年，其2.0版本发布。

1997年，两名以色列程序员Zeev Suraski和Andi Gutmans重写的PHP的分析器(parser)成为PHP发展到3.0的基础，而且从此将PHP重命名为PHP: Hypertext Preprocessor。此后，这两名程序员开始重写整个PHP核心，并于1999年发布了Zend Engine 1.0，这也意味着PHP 4.0的诞生。2004年7月，Zend Engine 2.0发布，由此也将PHP带入了PHP5时代。PHP5包含了许多重要的新特性，如增强的面向对象编程的支持、支持PDO(PHP Data Objects)扩展机制以及一系列对PHP性能的改进。

## 二、PHP Zend Engine

Zend Engine是开源的、PHP脚本语言的解释器，它最早是由以色列理工学院(Technion)的学生Andi Gutmans和Zeev Suraski所开发，Zend也正是此二人名字的合称。后来两人联合创立了Zend Technologies公司。

Zend Engine 1.0于1999年随PHP 4发布，由C语言开发且经过高度优化，并能够做为PHP的后端模块使用。Zend Engine为PHP提供了内存和资源管理的功能以及其它的一些标准服务，其高性能、可靠性和可扩展性在促进PHP成为一种流行的语言方面发挥了重要作用。

Zend Engine的出现将PHP代码的处理过程分成了两个阶段：首先是分析PHP代码并将其转换为称作Zend opcode的二进制格式(类似Java的字节码)，并将其存储于内存中；第二阶段是使用Zend Engine去执行这些转换后的Opcode。

## 三、PHP的Opcode

Opcode是一种PHP脚本编译后的中间语言，就像Java的ByteCode,或者.NET的MSL。PHP执行PHP脚本代码一般会经过如下4个步骤(确切的来说，应该是PHP的语言引擎Zend)：

1、Scanning(Lexing) —— 将PHP代码转换为语言片段(Tokens)

2、Parsing —— 将Tokens转换成简单而有意义的表达式

3、Compilation —— 将表达式编译成Opocdes

4、Execution —— 顺次执行Opcodes，每次一条，从而实现PHP脚本的功能

## 四、php的加速器

基于PHP的特殊扩展机制如opcode缓存扩展也可以将opcode缓存于php的共享内存中，从而可以让同一段代码的后续重复执行时跳过编译阶段以提高性能。由此也可以看出，这些加速器并非真正提高了opcode的运行速度，而仅是通过分析opcode后并将它们重新排列以达到快速执行的目的。

常见的php加速器有：

1、APC (Alternative PHP Cache)

遵循PHP License的开源框架，PHP opcode缓存加速器，目前的版本不适用于PHP 5.4。项目地址，http://pecl.php.net/package/APC。

2、eAccelerator

源于Turck MMCache，早期的版本包含了一个PHP encoder和PHP loader，目前encoder已经不在支持。项目地址， http://eaccelerator.net/。

3、XCache

快速而且稳定的PHP opcode缓存，经过严格测试且被大量用于生产环境。项目地址，http://xcache.lighttpd.net/

4、Zend Optimizer和Zend Guard Loader

Zend Optimizer并非一个opcode加速器，它是由Zend Technologies为PHP5.2及以前的版本提供的一个免费、闭源的PHP扩展，其能够运行由Zend Guard生成的加密的PHP代码或模糊代码。 而Zend Guard Loader则是专为PHP5.3提供的类似于Zend Optimizer功能的扩展。项目地址，http://www.zend.com/en/products/guard/runtime-decoders

5、NuSphere PhpExpress

NuSphere的一款开源PHP加速器，它支持装载通过NuSphere PHP Encoder编码的PHP程序文件，并能够实现对常规PHP文件的执行加速。项目地址，http://www.nusphere.com/products/phpexpress.htm

## 五、PHP源码目录结构

PHP的源码在结构上非常清晰。其代码根目录中主要包含了一些说明文件以及设计方案，并提供了如下子目录：

1、build —— 顾名思义，这里主要放置一些跟源码编译相关的文件，比如开始构建之前的buildconf脚本及一些检查环境的脚本等。

2、ext —— 官方的扩展目录，包括了绝大多数PHP的函数的定义和实现，如array系列，pdo系列，spl系列等函数的实现。 个人开发的扩展在测试时也可以放到这个目录，以方便测试等。

3、main —— 这里存放的就是PHP最为核心的文件了，是实现PHP的基础设施，这里和Zend引擎不一样，Zend引擎主要实现语言最核心的语言运行环境。

4、Zend —— Zend引擎的实现目录，比如脚本的词法语法解析，opcode的执行以及扩展机制的实现等等。

5、pear —— PHP 扩展与应用仓库，包含PEAR的核心文件。

6、sapi —— 包含了各种服务器抽象层的代码，例如apache的mod\_php，cgi，fastcgi以及fpm等等接口。

7、TSRM —— PHP的线程安全是构建在TSRM库之上的，PHP实现中常见的\*G宏通常是对TSRM的封装，TSRM(Thread Safe Resource Manager)线程安全资源管理器。

8、tests —— PHP的测试脚本集合，包含PHP各项功能的测试文件。

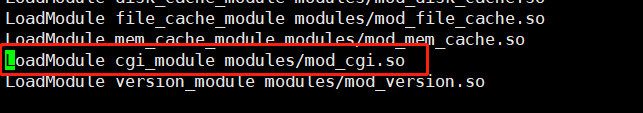
9、win32 —— 这个目录主要包括Windows平台相关的一些实现，比如sokcet的实现在Windows下和\*Nix平台就不太一样，同时也包括了Windows下编译PHP相关的脚本。

执行原理

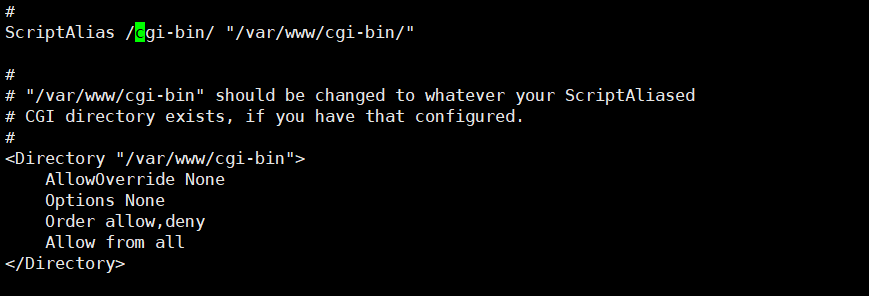
CGI: Common Gateway Interface

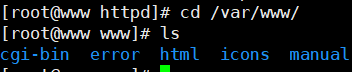
Apache进程 <-----> CGI <-----> PHP解释器

Httpd加载了mod\_cgi.so 模块

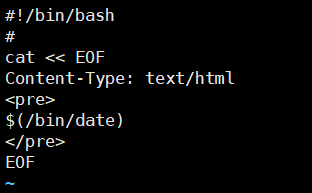


ScriptAlias

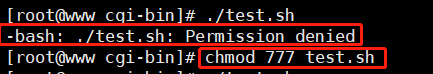


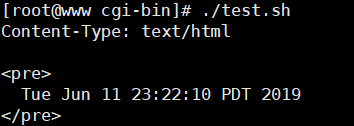


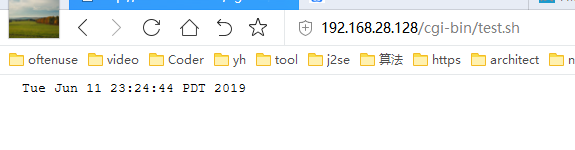
在/cgi-bin/目录下创建test.sh脚本



执行一下，如果提示没有权限，那就改下权限







DSO：dynamic shared object 动态模块分析

PHP中有 php\_mod 在httpd进程中就可以使用了。

fastCGI: php5.4以后有 FPM。

Apache+php结合有3种：

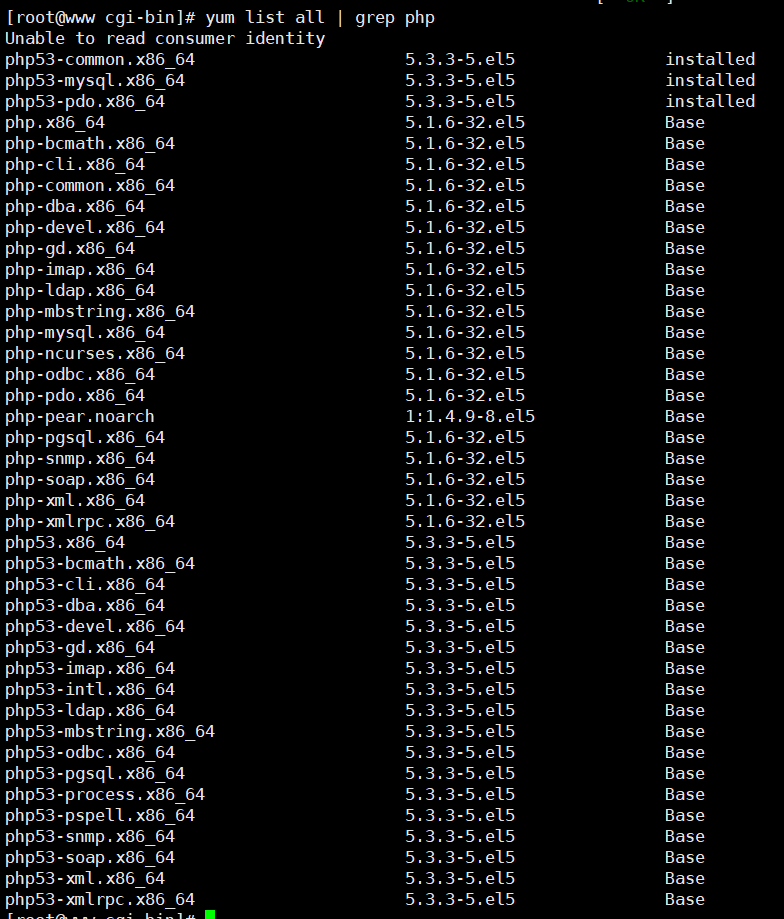
1. CGI
2. Module
3. FastCGI 性能最好，配置麻烦

最简单的第二种。

处理静态内容Nginx比Apache要好很多。Nginx+FPM

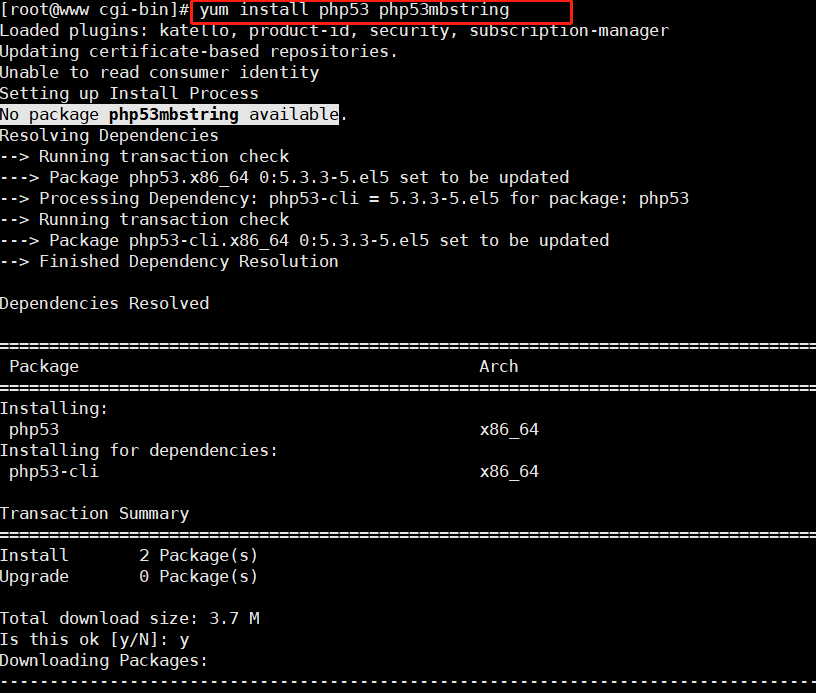
## 安装PHP

Yum list all | grep php

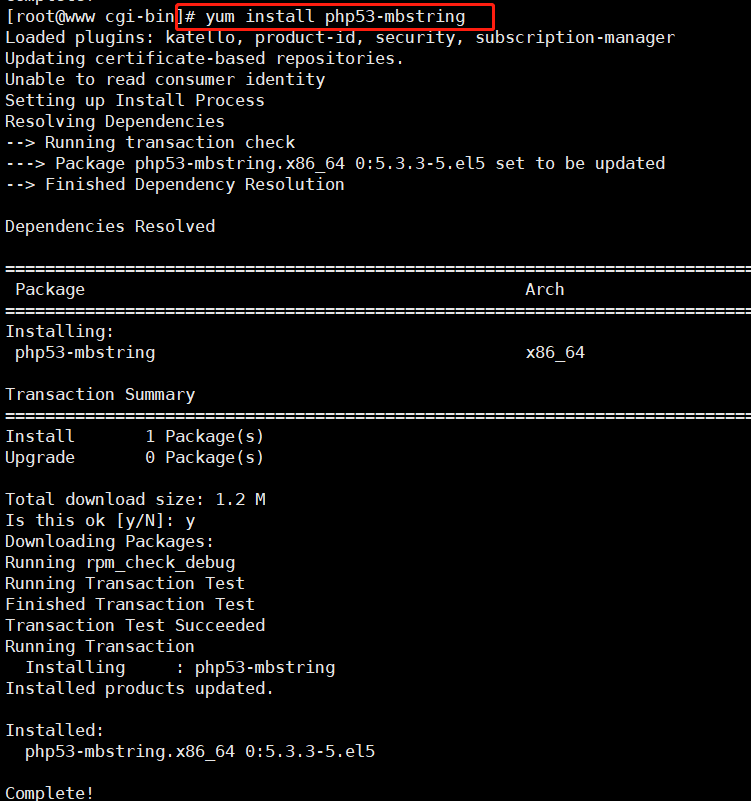


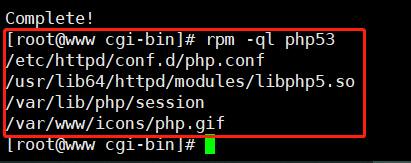
两个版本， 我们安装53

Yum install php53

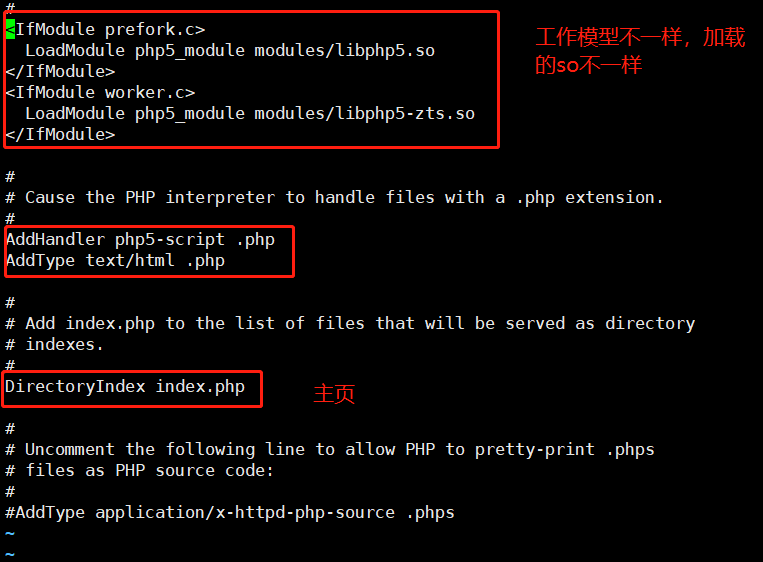


Yum install php53-mbstring





vi /etc/httpd/conf.d/php.conf



在/www/hc1.org/目录下提供一个index.php

