三种框架对比

面试常见问题之ZF、TP、CI等框架的区别

HELLO WORLD性能测试

http://www.ttlsa.com/php/yii-yaf-ci-php/

Zend Framework简称ZF

ZF是Zend官方产品，

代码严谨，

采用了完全面向对象的模式，

可使用命令脚本创建项目，

纯PHP5环境，

使用了大量的接口、异常、抽象。

ZF各模块松散 耦合，非常灵活。

自带了非常多的library，

MVC设计，

比较简洁，

支持多种格式的配置文件（.ini、.php、.xml），

缓存功能比较强大，

尤其是后端缓存支持Memcache、APC、SQLite、文件等方式；

支持各种数据库驱动，默认是PDO方式，

ZF的

View层实现简单，

没有采用模板 引擎。框架比较大，略显臃肿，适合中大型项目，运行效率一般。

CodeIgniter简称CI

配置简单，

上手很快，

全部的配置使用PHP脚本来配置，

没有使用很多太复杂的设计模式，(MVC设计模式)

执行性能和代码可读性上都不错，

执行效率比较高，具有基本的MVC 功能.

快速简洁，代码量少，框架简单，容易上手，自带了很多简单好用的library，

框架适合中小型项目，

大型项目也不是不可以，

只是扩展能力稍差。

ThinkPHP简称TP

TP借鉴了Java思想，基于PHP5，

充分利用了PHP5的特性，

部署简单只需一个入口文件，一切搞定，简单高效，

中文文档齐全，入门超级简单。

自带模板引擎，具有独特的数据验证和自动填充功能，框架更新速度比较迅速，目前最新版本是3.x。

Yii框架：

Yii是国际化的项目，纯面向对象的框架

支持php的命名空间和自定义autoload的方法

支持多配置文件

自带环境监测脚本

框架特点：

Yii的组件思路是非常不错的，用起来十分地舒服。从session到cache，你可以无缝地更换所有的组件而无需重构项目。而且Yii的延迟加 载也做得比较彻底，每个组件都是用到的时候才加载。比如，TP中，如果配置了session自动打开，则TP在应用初始化的时候执行 session_start()。而Yii则是你用到session的时候才打开session。

CodeIgniter（CI）

优点：

1. 配置简单，全部的配置使用PHP脚本来配置，执行效率高；具有基本的路由功能，能够进行一定程度的路由；具有初步的Layout（布局）功能，能够制作一定程度的界面外观；数据库层封装的不错，具有基本的MVC功能

2. 快速简洁，代码不多，执行性能高，PHP框架简单，容易上手，学习成本低，文档详细；自带了很多简单好用的library，框架适合小型应用

缺点：

1. 把Model层简单的理解为数据库操作

2. PHP框架略显简单，只能够满足小型应用，略微不太能够满足中型应用需要

评价：

总体来说，拿CodeIgniter来完成简单快速的应用还是值得，同时能够构造一定程度的layout（布局），便于模板的复用，数据操作层来说封装的不错，并且CodeIgniter没有使用很多太复杂的设计模式，执行性能和代码可读性上都不错。至于附加的 library 也还不错，简洁高效。

CakePHP

优点：

1. CakePHP是最类似于RoR的PHP框架，包括设计方式，数据库操作的Active Record方式；设计层面很优雅，没有自带多余的library，所有的功能都是纯粹的框架，执行效率还不错；数据库层的 hasOne, hasMany 功能很强大，对于复杂业务处理比较合适；路由功能，配置功能还不错；自动构建脚手架（scaffold）很强大；适合中型应用；基本实现过了MVC每一 层；具有自动操作命令行脚本功能；

2. 文档比较全，在国内推广的比较成功，大部分都知道CakePHP，学习成本中等

缺点：

1. CakePHP非常严重的问题是把Model理解为数据库层操作，严重影响了除了数据库之外的操作能力

2. CakePHP的cache功能略显薄弱，配置功能稍嫌弱；CakePHP不适合大型应用，只适合中型应用，小型应用来说略微的学习成本高了点

评价：

总体来说CakePHP框架代表了PHP框架很重要的一个时代和代表，并且目前发挥着很重要的作用，不少自己写的框架都模仿了CakePHP的方式，是个里程碑式的产品；CakePHP透露着RoR的敏捷开发方式和把数据库操作认为是唯一Model的设计思想，作为开发快速应用和原型是绝好的工具；同样，用来做Web2.0网站的开发框架，也是值得选择的。

Zend Framework（ZF）

优点：

1. 官方出品，自带了非常多的 library，框架本身使用了很多设计模式来编写，架构上很优雅，执行效率中等；MVC设计中，比较简洁，具有路由功能，配置文件比较强大（能够处理 XML和php INI），各种 library 很强大，是所有PHP框架中各种功能最全面的，包括它不仅是一个PHP框架，更是一个大类库（取代PEAR），这是它的主要特色；能够直观的支持除数据库操 作之外的Model层（比 CodeIgniter 和 CakePHP 强），并且能够很轻易的使用Loader功能加载其他新增加的Class；Cache功能很强大，从前端Cache到后端Cache都支持，后端 Cache支持Memcache、APC、SQLite、文件等等方式；数据库操作功能很强大，支持各种驱动（适配器）

2. 文档很全，在国内社区很成熟，并且目前不少Web 2.0网站在使用，学习成本中等

缺点：

1. MVC功能完成比较弱，View层简单实现（跟没实现一样），无法很强大的控制前端页面

2. 没有自动化脚本，创建一个应用，包括入口文件，全部必须自己手工构建，入门成本高

3. Zend Framework 作为一个中型应用框架问题不大，也能够勉强作为大型应用的PHP框架，但是作为一个很成熟的大型PHP框架来说，还需要一些努力

评价：

作为官方出品的框架，Zend Framework的野心是可以预见的，想把其他框架挤走，同时封装很多强大的类库，能够提供一站式的框架服务，并且他们的开发团队很强大，完全足够有能 力开发很强大的产品出来，所以基本可以确定的是Zend Framework前途无量，如果花费更多的时间去完善框架。同样的，Zend Framework架构本身也是比较优雅的，说明Zend官方是有很多高手的，设计理念上比较先进，虽然有一些功能实现的不够完善，比如View层，自动化脚本等等，这些都有赖于未来的升级。总体来说Zend Framework是最值得期待的PHP框架，当然，你目前要投入你的项目中使用也是完全没问题的。

Symfony

面试常见问题之ZF、TP、CI等框架的区别 - 最强大脑 - 最强大脑的博客

优点

1. Symfony 是我了解的PHP框架中 功能最强大的，而且我使用时间比较长，但是很多功能还是没有挖掘出来；它完整实现了MVC三层，封装了所有东西，包括 $_POST，$_GET 数据，异常处理，调试功能，数据检测；包含强大的缓存功能，自动加载Class（这个功能很爽），强大的i18n国家化支持；具有很强大的view层操 作，能够零碎的包含单个多个文件；非常强大的配置功能，使用yml配置能够控制所有框架和程序运行行为，强大到让人无语；能够很随意的定义各种自己的 class，并且symfony能够自动加载（auto load）这些class，能够在程序中随意调用；包含强大的多层级项目和应用管理：Project --> Application --> Module --> Action，能够满足一个项目下多个应用的需要，并且每层可以定义自己的类库，配置文件，layout；非常强大的命令行操作功能，包括建立项目、建立 应用、建立模块、刷新缓存等等；

2. Symfony绝对是开发大型复杂项目的首选，因为使用了Symfony，将大大节约开发成本，并且多人协作的时候，不会出现问题，在Project级别定义好基础Class以后，任何模块都能够重用，大大复用代码

缺点：

1. 数据库操作model采用了重量级的propel和creole，不过在我测试的版本中已经把他们移到了addon里，可用可不用

2. 缓存功能无法控制，每次开发调试总是缓存，需要执行 symfony cc, symfony rc 来清除和重建缓存；

3. 效率不是很高，特别是解析模板和读取配置文件的过程，花费时间不少；

4. 学习成本很高，并且国内没有成熟的社区和文档，连中文手册都没有，相应的要掌握所有功能，需要花费比较多的时间

评价：

Symfony绝对是企业级的PHP框架，唯一能够貌似能够跟Java领域哪些强悍框架抗衡的东西；强悍的东西，自然学习复杂，但是相应的对项目开发 也比较有帮助，自然是推荐复杂的项目使用Symfony来处理，觉得是值得，后期的维护成本比较低，复用性很强。相应的如果使用Symfony的应该都是 比较复杂的互联网项目，那么相应的就要考虑关于数据库分布的问题，那么就需要抛弃Symfony自带的数据库操作层，需要自己定义，当然了，Symfony支持随意的构造model层。

总结

以上数款PHP框架，各有特色，而且都是开源项 目，不过框架针对的项目不一样，一般来说 CodeIngiter 比较适合小型项目，CakePHP 和 Zend Framework 比较适合中型项目，Symfony 比较适合大型重量级项目，在项目选型的时候，要充分考虑框架的可以定制性、扩展性，因为每个项目都无法确定你是否会随着需求的变化进行改变。

相对来说，Zend Framework 和 Symfony 应对变化的能力比较强，特别是能够随意定制 model 层的Class，能够非常方便增加自己业务或者数据处理类，我是个人比较推荐在中大型项目中使用的PHP框架。

CodeIngiter 和 CakePHP 在中小型项目中同样能够发挥重大作用，快速开发和原型构建，非常适合目标不清晰的原型项目的开发。

Yii和ThinkPHP

这**不是**一篇评测文章。只是我的喃喃碎语，不计较真。而且，下面的内容真的会很杂，不全面，而且你不可能有和我一样的开发经历。所以对于某些我醉心的特性，你可能不会理解。同样的，我也不可能全部理解你为何对某一项特性十分喜欢。

关于ThinkPHP（以下简称TP）和Yii Framework（以下简称Yii）的背景、作者和速度方面就不涉及了。因为速度是一个很复杂的问题，牵扯的因素很多。我不得不承认ThinkPHP是 一个是国内框架运营方面的榜样（当FleaPHP/QeePHP最火的那阵，我说过FleaPHP/QeePHP会倒的）。

运行环境：

- Windows NT ACER 5.1 build 2600 (Windows XP Professional Service Pack 3) i586
- Apache/2.2.14 (Win32) DAV/2 mod_autoindex_color PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
- Mysql 5.1.4

下载的代码：

- ThinkPHP 2.1，带扩展、示例和文档完整包，发布于2011年5月1日，下载地址是 http://thinkphp.cn/Down/download/178
- Yii 1.1.7，内含框架、实例和环境检测，下载地址 http://yii.googlecode.com/files/yii-1.1.7.r3135.tar.gz 文档需要另外下载

中文化方面，TP和Yii都可以满足中文用户的需求。但是由于Yii是国际化的项目，所以代码的注释仍旧是英文。不过两个框架的作者都是国人（没错，Yii作者的国籍仍旧是中国），所以交流起来还是很方便的。

是Yii自带了一个环境检测脚本，可以告诉你当前的主机环境是否满足Yii的需求。检测的内容也比较详细。我觉得这点比较方便。TP最低需要PHP5.0支持，而Yii最低需要PHP 5.1.0支持。由于我使用PHP 5.3，所以对我来说没有什么区别。

Yii是纯面向对象的的框架，而TP提供了一系列单字母函数。相比之下我更喜欢Yii的方式，因为可以避免项目之间的冲突。

TP在以前的版本的基类Base类，当时就和一个整合Ucenter时的类冲突了，一度很苦恼。现在TP的各种基类仍旧是直接命名，如Think 类。在项目开发过程中就会体会命名冲突的痛苦之处。Yii则在框架的类都加上了C前缀（接口是I前缀），有效地避免了这个问题。Yii中的 CComponent是所有类的基类，可以看看CComponent的代码，很有用。

说到命名问题了，就不得不说自动导入的问题。TP的类导入和Yii的代码风格差不多。但是Yii还支持PHP的命名空间和自定义autoload方法。

TP有个特色叫项目编译。我觉得与其使用项目编译，还不如使用APC。在Yii中也有个yiilite.php文件，里面就包含了Yii的所有核心类。Yii作者表示在没有APC的情况下，还是不要使用这个“编译”好的文件，因为反而会增加系统开销。

TP中还在第一次访问的时候自动生成项目，我觉得这一点和自动编译一样，都是我不喜欢的。我对每添加一个if都很敏感，这种判断让我很纠结。比如说 TP在每次运行的时候都要检测PHP版本，而Yii则单独做了一个内容更详细的环境监测脚本。我既然要用这个框架，我在第一次使用的时候，肯定就知道能不 能在当前环境上使用了，为什么要每次都要检测呢。当时我就说过，TP为用户做了太多事情。比如旧版本中的TopN函数。

Yii的组件思路是非常不错的，用起来十分地舒服。从session到cache，你可以无缝地更换所有的组件而无需重构项目。而且Yii的延迟加 载也做得比较彻底，每个组件都是用到的时候才加载。比如，TP中，如果配置了session自动打开，则TP在应用初始化的时候执行 session_start()。而Yii则是你用到session的时候才打开session。

说到项目的配置文件，TP要求是config.php，而Yii则比较灵活，支持多配置文件。

当初TP很推崇自己的ThinkAjax，现在也改用JQuery。这一点是进步。

TP做了很多小实例，这一点值得Yii学习。Yii在这一方面正在有一个叫yii playground的实例网站在开发中（http://code.google.com/p/yiiplayground/）。

TP的动态模型可以实现不需要定义Model。但是在实际的项目中，我更倾向于使用Yii的方式。顺便说一句，将label定义在model中，为我的日常开发带来了许多方便之处。

刚才提到TP的项目自动生成，Yii中也有这种工具。而且比起TP，Yii的工具更加强大而且可扩展。

从TP的代码中，有人可以看出其作者熟悉JAVA。而从Yii的代码中，有人会发现其作者熟悉.Net。这常常是我身边人看到代码的时候发生的小插曲。

Yii封装了大量的页面控件和类库，也是Yii如此吸引我的一点。这是TP短期无法比拟的，在TP的使用过程中总遇到这样那样的问题，让我感觉TP对我反而是阻碍。而Yii真的是，舒服，实在是太好用了！

无论从代码规范、设计思路、类库丰富程度上来说，TP都远远不及Yii。有人说你看TP多简洁，Yii太臃肿了。错了！简单和简洁不是一回事。TP 那叫简单，你读读Yii的代码吧，那才叫简洁。至于臃肿，去看看Zend Framework就知道了。（顺便说一句，我很喜欢Zend Framework，它是学习设计的典范）

说到读代码。对于程序员真的很难吗？读写得好的代码应该是一种享受才对。Yii的学习曲线是比TP高那么一点点，但是对比Yii的

巨大优势而言不算什么了。而且，我认为在遇到学习困难就退缩或者认为Yii就像天书一样的人，还是转行吧。

以上是应一篇评论所写的。对比TP1，现在的TP2的确有了很多进步，但是还是存在一些问题。对比Yii……，TP真的没有可比的能力。抱歉让TP的fans失望了。

那就下定论了吗？不，不是的。从类库到框架，再到解决方案。什么是最好的？每一个人都有不同发说法，这是因为每一个人的思维习惯不同，遇到的问题不同，问题所在的环境也不同。怎么能奢求所有人都有同一个选择呢？

还是那句，适合，就是最好的。对我来说，Yii是最好的。
