Atitit 延迟绑定架构法attilax总结

[后期绑定和前期绑定 1](#_Toc2300)

[延迟调用 2](#_Toc10729)

[用 Java 语言延迟绑定 2](#_Toc31690)

[什么是推迟绑定 C++ 2](#_Toc28189)

[配置文件的延迟绑定 3](#_Toc3443)

何为延迟绑定呢。这个也比较好理解。libc库中有很多的函数，但是我们编写程序的时候，并不一定会调用libc库中的每个函数，更多的情况下，我们只调用了极少数函数，如果我们将每个函数的地址都解析出来，其实是一种浪费。所以采用的方法是用到函数时再进行对函数的位置进行定位。 这种技术就叫做延迟定位。

 动态链接技术，严格的说分成两类，一种是 Load-Time Relocation，这种技术容易理解，但是缺点也比较致命，不能共享，起不到节省内存的目的，目前X86\_64已经不提供这种方式；另外一种属于主流的位置无关（PIC）动态库

[Position Independent Code (PIC) in shared libraries](http://eli.thegreenplace.net/2011/11/03/position-independent-code-pic-in-shared-libraries/" \t "http://blog.chinaunix.net/_blank)

## 后期绑定和前期绑定

编程语言能够将对函数（或在面向对象语言中的方法）的声明从其调用中分离出来。可以声明一个方法并使用单独的语法调用这个方法，但最终系统需要将这两者绑到一起。将调用和实现绑到一起的过程叫做绑定。前期先绑定到类型再绑定到实现，还是后期先绑定到类型再绑定到实现，这对一门给定语言的编程体验来说有着显著的影响。大多数面向对象的语言都在后期绑定到实现，从而允许多态性 ，该功能让您能够将许多不同的子类型表示为一种类型。Java 代码和 C 主要在前期的一个编译的步骤里绑定到一种类型。使用此策略，编译器就有足够的信息可以捕获许多不同类型的 bug，比如说方法参数或返回值之间类型的不兼容。

## 延迟调用

在静态语言中，编译器在编译时直接将调用绑定到实现。动态语言则有些不同。Ruby、Smalltalk 和 Self 依赖于消息传送来延迟绑定。客户机使用消息传送来指定目标对象、消息和参数集。这完全是一个运行机制。所以动态语言有效地添加了一级间接寻址。它们将消息名绑定到一个对象上，

## 用 Java 语言延迟绑定

Java 社区对静态类型检查的迷恋程度令人惊讶，Java 程序员们正在不遗余力地寻找延迟绑定的方式。有些方法是成功的。诸如 Spring 等框架的存在主要是为了延迟绑定，它有助于减缓客户机和服务之间的耦合。面向方面的编程通过提供能够扩展类的功能（甚至可以超出其当前的功能）的服务来实现延迟绑定。像 Hibernate 这样的框架也可以延迟绑定，

# 什么是推迟绑定 C++

[九公主的悲哀](http://www.baidu.com/p/%E4%B9%9D%E5%85%AC%E4%B8%BB%E7%9A%84%E6%82%B2%E5%93%80?from=zhidao" \t "http://zhidao.baidu.com/_blank) | 浏览 339 次  2013-01-14 22:05

2015-08-09 16:39最佳答案

1、为每个含有虚函数的类（基类以及派生类）都创建一张虚函数表(VTable，存储于常量区)，依次存放虚函数的地址。对于派生类来说，如果没有重写其基类的虚函数，那么，将会在VTable中存放基类的虚函数地址。  
2、为每个含有虚函数的类的对象，创建一个指针(VPtr)，指向这个类的虚函数表（所以说，同类对象的  
VPtr的值是一样的，也就是说，虚函数，在这一点上，类似于类的static函数，是所有对象共有的。）  
3、通过强制转换将派生类对象的地址/引用 赋给基类指针/变量（也就是向上类型转换UpCasting）  
4、然后，通过基类指针/引用对象 来调用虚函数（即：polymorphic call），就会通过此时基类对象的VPtr指针在所指向虚函数表中寻址(一般是从表头地址开始，加上一定的偏移量)找到相应的函数地址，也就是之前派生类VPtr指针指向的虚函数表的函数地址

 晚捆绑（出自《C++编程思想》）或者动态类型。 意思是：一个表达式或者对象的类型直到运行的时候才确定。 与之对应的是晚捆绑（静态类型），意思是一个表达式的类型或者对象的类型在编译的时候就应经确定了。 C++实现玩捆绑使用的是：virtual关键字。virtual 关键字告诉编译器对某个类的某个函数采用晚捆绑机制

对应java里的泛型概念，不管推迟、晚、迟、动态，重点都是在于抽象和“编译器运行时”的对象绑定、理解成实例化也可以。

# 配置文件的延迟绑定

跨越边界\_ 延迟绑定.html