我们知道，常见的方法调用都是同步调用，这种调用方式是一种阻塞式的调用方式，即客户端（主调用方）代码一直阻塞等待直到被服务端（被调用方）返回 为止。这种调用方式相对比较直观，也是大部分编程语言直接支持的一种调用方式。但是，如果我们面对是基于粗粒度的服务组件，面对的是一些需要比较长时间才 能有响应的应用场景，那么我们就需要一种非阻塞式调用方式，即异步调用方式。

SCA编程模式提供了三种方式的异步调用，它们分别是：

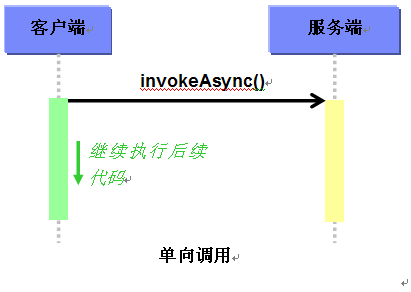
1． 单向调用方式。

2． 延迟响应方式。

3． 请求回调方式。

**单向调用**

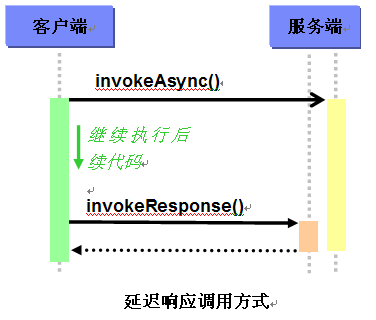
单向调用方式是最为简单的异步调用方式，在这种调用方式中，客户端发出请求之后就不再关心服务端的情况，包括是否执行成功，返回值是什么等等。我们可以用下面的图 4示来描述这种单向调用方式：

**图 4: 单向调用**  


单向调用方式是一种不管调用结果的方式，但是在很多情况下我们是需要知道调用结果的。我们需要知道调用是否成功，需要知道调用的结果，就算调用失败我们也希望知道错误代码等信息。在这种情况下，延迟响应和请求回调就是两种能够让我们知道调用结果的方式。

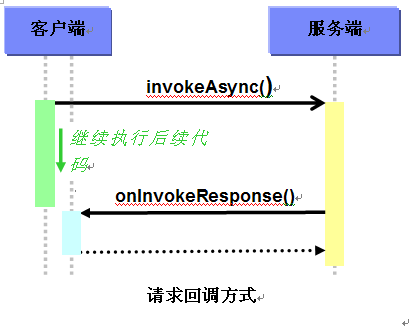
**延迟响应方式**

延迟响应方式是指客户端在发出调用请求之后继续执行，但是经过一段时间之后，客户端再调用相应的方法去检索返回结果，并通过参数指定如何根据调用的 结果而执行进一步动作。由于是异步调用方式，因此，在第一次发出调用请求的时候，服务端需要返回一个称为票据（Ticket）的对象。这个对象会作为第二 次发出检索结果请求时的一个参数。显然，这个Ticket对象的作用与WEB编程的SessionID非常类似。我们可以用图 5 来表示延迟相应调用方式：

**图 5：延迟响应调用方式**  


**请求回调**

与延迟响应方式类似，请求回调方式也能得到服务端的响应，但是不同的是这个响应是由服务端通过回调方式来触发的，而不像延迟响应方式由客户端来主动 检索的。请求回调方式的原理与许多编程语言中的回调机制类似，不同的是这里实现的层次比较高一点。我们可以用图6来表示请求调用方式：

**图 6：请求回调方式**  
  
http://www.ibm.com/developerworks/cn/webservices/ws-sca/