　　Appache JMeter 以及 SOAP 协议简述  
　　开源测试工具：Appache JMeter  
　　JMeter 是 Apache 基金会 Jakarta 上的一个纯 Java 开源项目，起初用于基于 Web 的压力测试（pressure test），后来其应用范围逐渐扩展到对文件传输 FTP， 大型数据库（JDBC 方式），脚本程序（CGI， Perl 等），Web Services，Java 应用系统等方面的测试。JMeter 本身主要用于性能测试，如系统压力等。除此之外，JMeter 能够对应用系统做功能测试和回归测试，并且能够通过使用带有断言的脚本程序来验证系统然后返回用户期望的结果。为了提高工具的应用灵活性，JMeter 允许使用正则表达式创建断言。正是由于它的灵活性和可扩展性，JMeter 逐渐成为流行的开源测试工具。  
　　消息传递协议：SOAP  
　　SOAP（Simple Object Access Protocol）称为简单对象访问协议， 是 W3C 定义的一种标准消息传递协议，而它通常被认为是 Web Services 的事实标准。SOAP 协议使用 XML 语言来描述，SOAP 消息格式是由 XML Schema 模式定义，因而通过使用 XML 命名空间使得 SOAP 具有很强的可扩展性。  
　　SOAP 是在去中心化（Decentralized）分布式（Distributed）环境中用来信息交换的一个轻量级协议。SOAP 本身并不定义像程序模型或实施声明等形式的语法，而只定义了一种简单机制：通过提供模块化的包装模型编码机制来传输应用信息。  
　　SOAP 基本结构：  
　　1) 信封 Envelope Envelope 元素是 SOAP 中的根元素，并且定义为在 SOAP 消息中必须出现。Envelope 元素中可以包含多可选的 Header 元素，但同时必须要包含一个 Body 元素。  
　　2) 消息头 Header Header 可能出现在 SOAP 消息中，是一个可选元素。如果出现在消息中，那么 Header 一定要是 SOAP 中的第一个元素。SOAP Header 在 Web Services 中的应用越来越广泛，例如在应用程序的安全性事物中使用标准的消息头文件，因而成为扩展 SOAP 协议的一个非常有效的方法。  
　　3）消息体 Body Body 元素是 SOAP 中必须出现的一个元素，它要包含应用程序中的传输数据或者反馈消息。 应用程序中的传输数据可以是任意形式的 XML 数据。SOAP 消息接收者最终来处理 SOAP Body 体。  
　　JMeter 调用 SOAP 框架机制  
　　SOAP 使用 RPC（远程过程调用）和消息传递来建立通信服务，SOAP RPC 定义了用于表示远程过程调用和应答的协议。SOAP 协议本身仅仅定义了消息的交换结构，它可以和许多现存因特网协议结合在一起使用，其中包括超文本传输协议（ HTTP），多用途网际邮件扩充协议（MIME），Java 消息服务（JMS）以及简单邮件传输协议（SMTP）等。目前与 SOAP 应用最为广泛的是 HTTP 协议和 JMS 协议，而与之相对应的两种应用就是 SOAP Over HTTP 和 SOAP Over JMS。  
　　根据 JMS 的规范，消息交换有 2 种方式：消息发布 / 订阅方式和点对点方式。由这两种交换方式所建立的消息收发系统都是异步的，即 JMS 客户机可以发送消息而不必等待回应。如果应用程序测试者或测试脚本开发者希望每一条消息都能够被处理并且消息总是能够被传送到指定的位置，那么应该使用点对点消息模型而不是消息发布 / 订阅模型。  
　　HTTP（超文本传送协议）是属于应用层的面向对象的协议，是万维网 (WWW) 的基础，由于其简单快速、灵活、无连接、无状态的方式，适用于分布式网络信息系统。SOAP Over HTTP 应用就是指的是遵守 SOAP 编码规则的 HTTP 请求 / 响应，我们可以用简单的公式来对此作一个描述：HTTP + XML = SOAP。  
　　JMeter 也同样提供了两种 Sampler 分别建立对这两种服务的调用：Web Services (SOAP) Request 和 JMS Point-to-Point。前者使用互联网中最为广泛的超文本传输协议（ HTTP）而后者使用 JMS 协议，JMS 是 Java 平台面向消息中间件的技术规范，用它来提供创建、发送、接收、读取消息的服务。许多厂商目前都支持 JMS，包括 BEA 的 WebLogic JMS service，IBM 的 MQSeries 和 Progress 的 SonicMQ。

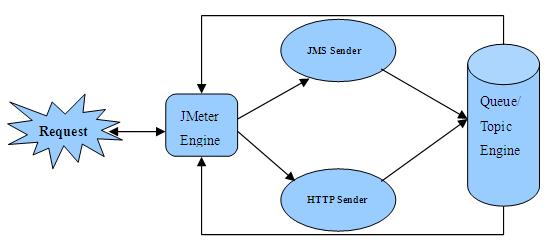


图 1.JMeter 框架基于上述两种不同的协议对 SOAP 消息的一次简单调用机制流程

　　准备测试环境  
　　当精心编写好测试脚本满怀信心的去运行测试计划时，发现所有的测试脚本都 failed 掉了，原因可能是你的测试环境中并没有完全准备好。下面给出了准备测试环境的详细步骤：  
　　1.环境变量设置：JMeter 运行在 JRE/JDK 之上，在所有开始之前要设置 JMeter 自动检测的环境变量 JAVA\_HOME=#JAVA INSTALL DIRECTORY#.  
　　2.JMeter 安装：本文下面下载栏提供了 Apache JMeter 下载地址，首先要取得最新版本的 JMeter 测试工具，JMeter 最新版本包含了构建和运行绝大部分测试类型的文件，包括 Web (HTTP/HTTPS)， FTP， JDBC， LDAP， Java， 和 JUnit 等。  
　　3.准备 jar 包：JMeter 虽然提供了对 SOAP Over HTTP 以及 SOAP Over JMS 测试的 Sampler，但是出于对 licence 的考虑它本身并没有提供 JMS 需要使用的 jar 包。因此，在运行测试之前需要将这些包复制到 JMeter 的 lib 目录下，下面列表对测试所需 jar 包作了详细说明。  
　　4.BeanShell 脚本处理：如果在测试用例中用到了 BeanShell 脚本，则需要将 BeanShell 包拷贝到 JMeter bin 目录下。BeanShell 是一种兼容 Java 语言的轻量级脚本语言，JMeter 脚本中可能会经常用它来做日志处理，正则表达式后处理（Post- Process）等。如果在测试用例中用到了 Mail Visualiser， Mail Reader 以及 Web Services (SOAP) sampler，则需要将 MAIL 包拷贝到 JMeter bin 目录下。如果在测试用例中用到了 JMS 相关的 sampler，则需要将 JMS 包拷贝到 JMeter bin 目录下。  
　　下面的列表列出了不同的测试用例所需要的 jar 包，以及其下载地址：  
　　bsh-2.0b4.jarhttp://www.beanshell.org/  
　　mail.jar http://java.sun.com/products/javamail/index.jsp  
　　jms.jarhttp://java.sun.com/products/jms/docs.html  
　　调试脚本中非常有用的信息日志：jmeter.log 在脚本的调试和运行过程中，所以的日志信息都会记录在 jmeter.log 中，因此你会在这个文件中找到比较有用的信息。  
　　注意事项  
　　如果 JMeter 在执行测试脚本过程中应该修改 jmeter.bat 文件中的一些参数，参数大小可以根据测试计划合理确定：  
　　HEAP=-Xms256m – Xmx1024m  
　　NEW=-XX:NewSize=128m -XX:MaxNewSize=128m  
　　TENURING=-XX:MaxTenuringThreshold=2  
　　EVACUATION=-XX:MaxLiveObjectEvacuationRatio=20%  
　　PERM=-XX:PermSize=64m -XX:MaxPermSize=64m  
　　DEBUG=-verbose:gc -XX:+PrintTenuringDistribution  
　　此外，在搭建测试环境时还需要更多注意的地方：  
　　JMeter 使用兼容 JKD1.4 或者更高版本  
　　JMeter 无法识别 zip 格式的包文件，所以需要的包文件均要求以 .jar 结尾  
　　JMeter 会自动在 JMETER\_HOME/lib 和 ext 目录下寻找需要的类  
　　对于使用 CSVDataSet， 那么不要勾选 "Memory Cache"否则数据无法迭代  
　　使用 JMeter 连接 SOAP Over HTTP 服务  
　　JMeter 提供了 Web Service (SOAP) sampler，用以调用基于 HTTP 的 Web 服务。下面详细说明 SOAP Over HTTP 服务调用的各个属性。

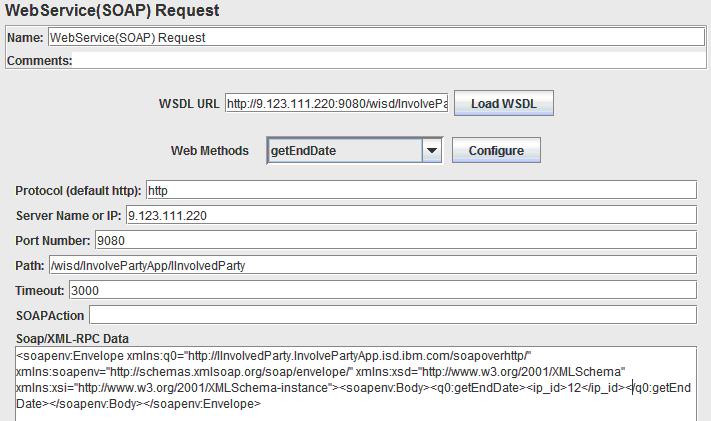


图 2.SOAP Over HTTP 服务调用的各个属性

　　SOAP Over HTTP 服务调用的各个属性说明：  
　　WSDL URL:指定 WSDL 文件的目标地址  
　　Web Methods:选择本次请求调用的方法  
　　Protocol:指定使用的协议，默认为 HTTP  
　　Server Name Or IP:服务的地址（服务器名或 IP 地址）  
　　Path:调用方法所在的位置  
　　Timeout:设置请求超时限制  
　　SOAPAction:存在于 WSDL 文件中的调用方法，默认不必填写  
　　Soap/XML-RPC Data:请求数据  
　　下面是一次完整的 HTTP 请求与 HTTP 响应 SOAP 数据：  
HTTP Request  
<soapenv:Envelope>  
<soapenv:Body>  
<q0:getEndDate>  
<ip\_id>12</ip\_id>  
</q0:getEndDate>  
</soapenv:Body>  
</soapenv:Envelope>  
HTTP Response  
<soapenv:Envelope>  
<soapenv:Header/>  
<soapenv:Body>  
<p928:getEndDateResponse>  
dstSavings=3600000，useDaylight=true，startYear=0，startMode=3，startMonth=2，  
startDay=8，startDayOfWeek=1，startTime=7200000，startTimeMode=0，endMode=3，  
endMonth=10，endDay=1，endDayOfWeek=1，endTime=7200000，endTimeMode=0]]，  
firstDayOfWeek=1，minimalDaysInFirstWeek=1，ERA=1，YEAR=2005，MONTH=8，  
WEEK\_OF\_YEAR=37，WEEK\_OF\_MONTH=2，DAY\_OF\_MONTH=7，DAY\_OF\_YEAR=250，DAY\_OF\_WEEK=4，  
DAY\_OF\_WEEK\_IN\_MONTH=1，AM\_PM=0，HOUR=0，HOUR\_OF\_DAY=0，MINUTE=0，SECOND=0，  
MILLISECOND=0，ZONE\_OFFSET=-18000000，DST\_OFFSET=3600000]  
</p928:getEndDateResponse>  
</soapenv:Body>  
</soapenv:Envelope>  
　　使用 JMeter 连接 SOAP Over HTTP 服务  
　　JMeter 提供了 Web Services (SOAP) sampler，用以调用基于 HTTP 的 Web 服务。下面详细说明 SOAP Over HTTP 服务调用的各个属性。

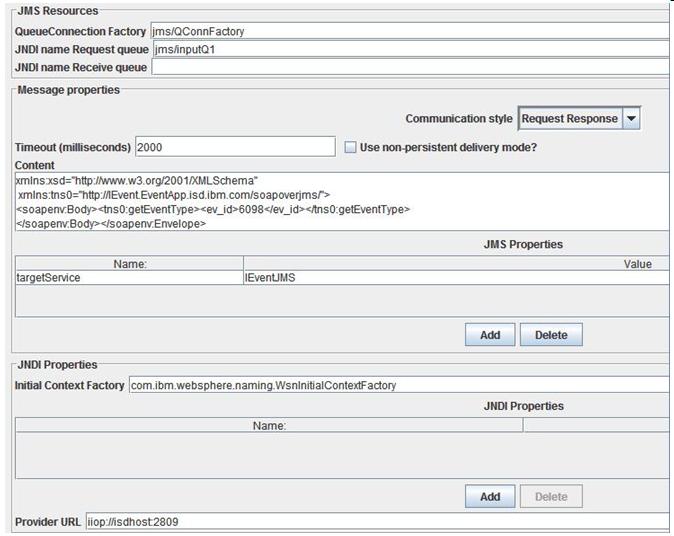


图 3.SOAP Over HTTP 服务调用的各个属性

　　SOAP Over JMS 服务调用的各个属性说明：  
　　QueueConnectionFactory:连接工厂的默认 JNDI 实体  
　　JNDI name Request queue:JNDI 请求队列名字  
　　JNDI name Receive queue:JNDI 接收队列名字  
　　Timeout:请求超时设置  
　　Communication style:通讯形式（包括仅仅请求和请求应答）  
　　Content:请求信封  
　　JMS Properties:JMS 的一些属性设置（对于 IBM WAS 必须要有 targetService 属性）  
　　Initial Context Factory:JNDI 的初始会话工厂  
　　Provider URL:服务提供地址  
　　下面是一次完整的 JMS 请求与 JMS 响应 SOAP 数据：  
　　JMS Request  
　　<soapenv:Envelope>  
　　<soapenv:Body>  
　　<tns0:getAuEmpPositionId>  
　　<ev\_id>6098</ev\_id>  
　　</tns0:getAuEmpPositionId>  
　　</soapenv:Body>  
　　</soapenv:Envelope>  
　　JMS Response  
　　<soapenv:Envelope>  
　　<soapenv:Header/>  
　　<soapenv:Body>  
　　<p150:getAuEmpPositionIdResponse>  
　　<getAuEmpPositionIdReturn xsi:nil="true"/>  
　　</p150:getAuEmpPositionIdResponse>  
　　</soapenv:Body>  
　　</soapenv:Envelope>  
　　设计高效的测试用例集  
　　压力测试或者系统测试不同于功能测试，测试的重点不在系统产品是不是满足设计需求。它所看重的是系统在大的用户量和负载情况下的可靠性以及系统响应 ， 它目标是测试系统的执行效率，特别是在较短时间内系统负载快速增长时系统的相应速度。在实际的测试过程中，大量用户同时访问的系统节点也可能成为产品潜在的效率瓶颈。因此 ， 压力测试和系统测试也往往是在功能测试之后进行。  
　　对于普通的软件系统 ， 产品的瓶颈可能会在数据库服务器上，Web 服务器上，而对于 SOAP 服务系统测试，Web Services 服务器和 JMS 服务器是客户端请求的主要节点 ， 同时，主要业务逻辑的处理也都分布在这些节点上，它们很有可能成为系统访问的瓶颈，如果这些节点出现问题，那么对整个系统的效率会有致命的影响，也是压力测试和系统测试要优先考虑的。  
　　改进测试策略、测试方法、测试过程，使用高效的测试用例集，从而保证产品质量。这个是主要目的，也是最直接的目的。一个高效的测试用例集应包含以及适应如下要素：  
　　在什么时候确定要执行系统测试  
　　如何去检测并解决系统性能和负载问题  
　　收集监视服务器性能数据（I/O，CPU，MEM）  
　　尽量减少因为个人配置和某些测试用例而造成系统出现错误和瓶颈  
　　所有测试工作都得到有效协调并目标一致  
　　当已经确定了所需的 JMeter Samplers，并且在此基础上设计出一个通用的测试计划，那么就可以构建我们的测试脚本了。本文的测试用例以及最终的测试计划也是建立在这些要素之上。  
　　测试计划（Test Plan）描述了测试运行过程中 JMeter 的执行顺序、过程以及步骤，一个完整的测试计划包括一个或者多个线程组 (Thread Groups)、循环控制器（Loop Controllers）、监听器 (Listener)、逻辑控制器（Logic Controller）、定时器（Timer）、断言（Assertions）、配置信息（Config Elements）等。  
　　在测试计划中添加一个用户定义变量配置元素（User Defined Variables）， 可以在里面定义服务器地址，日志路径，超时限制等变量，提供脚本重用。同时添加两个用户组，一个是 SOAP Over HTTP Group，一个是 SOAP Over JMS Group。在每个用户组下面分别添加一个总的循环控制器（Loop Controller），用以控制脚本循环次数。在总循环控制器下面添加随机选择器（Random Selector）用以随机选择运行测试脚本。下图是我们整个的 Test Plan。

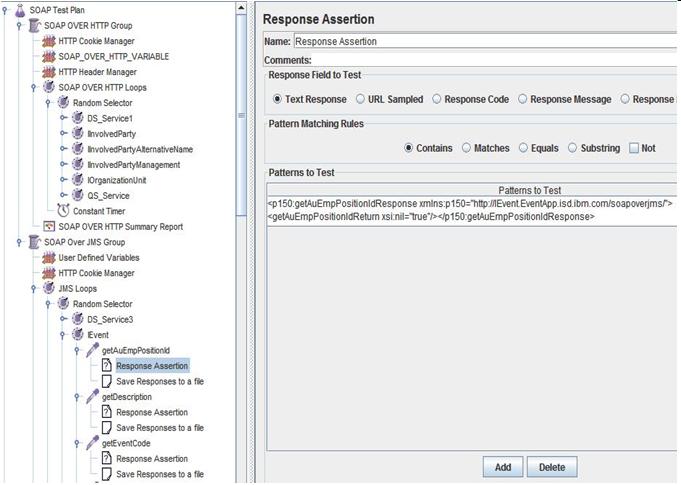


　图 4. 设计完成之后的 SOAP 测试计划

　　启动 SOAP 服务测试  
　　当准备好我们的测试计划之后就可以启动执行压力测试了，为了记录测试结果和信息，要增加 Listener 来完成这个任务。JMeter 提供了可视化的界面以及统计报表来供我们选择。这里我们使用表格（Summary Report）的形式来查看和分析测试结果。  
　　你可以通过下面的步骤来给每个 Group 增加 Summary Report 监视器 :  
　　1. 选中 Test Plan 中要添加 Listener 的 Group 节点，这里我们选择 SOAP Over JMS Group。  
　　2. 右击选择 Add-->Listener-->Summary Report， 界面右边会相应的出现我们选择的 Listener 的设置信息。  
　　在经过一系列工作之后，已经完成了整个 Test Plan，现在可以选择 JMeter 菜单 run-->start 来启动我们的压力测试了。下图是运行过程中测试统计数据的实时跟新信息。为了增加请求负载和获得更有价值的数据，我们可以更改线程数、等待时间和循环次数。

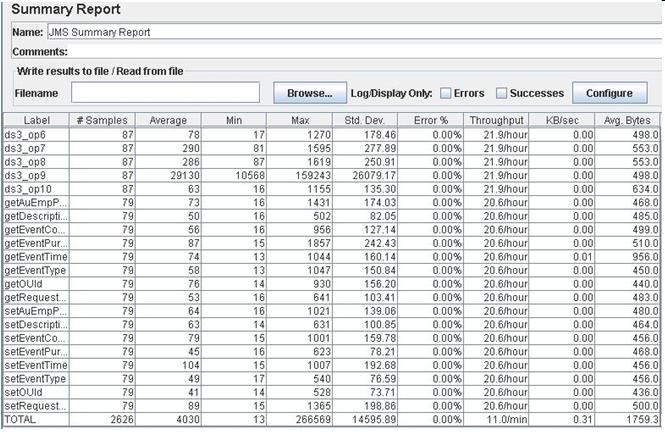


图 5. 基于吞吐量的测试结果报表（Summary Report）

　　获得的经验  
　　总结：  
　　使用 JMeter 来作为测试工具对 SOAP 协议的服务进行压力和系统测试是一个很好选择，选择 JMeter 来进行 SOAP 测试具有以下显著的优点：首先 JMeter 提供了强大全面的 SOAP 请求 / 接收以及监视功能，允许你执行、捕获在客户端和服务器端的 SOAP 流量分析。其次，可以使用 JMeter 可以设计出高效、易维护的测试用例甚至测试计划。最后，我们可以选择 JMeter 提供的符合我们情况的结果 Listener，并且可以从这些 Listener 中很容易的分析出系统或者是服务存在的问题和瓶颈。总体上讲，我们在 JMeter 测试框架中构建的 SOAP 测试计划很好的完成了对 SOAP 协议的系统测试。下面详细列出了我们在本次测试过程中获得的技巧以及经验。  
　　测试工具的选择  
　　测试工具在软件和产品测试中是必不可少的，包括系统测试，压力测试，性能测试以及功能测试。它也会与要测试的产品，测试的领域以及测试的重点有很大的关系。因此，选择一款合适的测试工具对高效的完成测试是至关重要的。  
　　设计高效的测试计划  
　　一个高效的测试用例集可以快速的诊断出系统的性能瓶颈。 为此应该全面的分析了解要测试系统的架构与应用，尽量避免盲目或者重复的测试用例，最终来构建效率尽可能高的测试用例集。  
　　尽量全面的系统监控  
　　软件缺陷和系统性能瓶颈的诊断可能会需要各个方面的检测数据，它们对问题的解决会提供很大的帮助，因此测试过程中应该有全面的系统监控，包括服务器的各项数据（CPU，I/O，MEM）， 后台数据库的各项数据，相应时间以及网络流量等。  
　　关注 SOAP 请求的超时（Timeout）  
　　基于 SOAP 协议的请求，无论是 SOAP Over HTTP 还是 SOAP Over JMS 都会有请求超时（Timeout），引起请求超时的原因可能是多方面的（服务器的响应速度，效率，网络带宽等），合理的分析以及设置请求超时能更准确的掌握产品的性能情况。