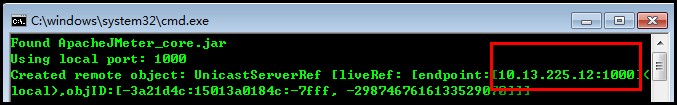
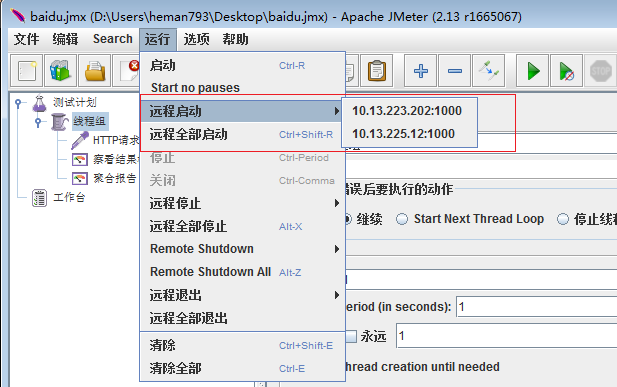
在使用Jmeter进行性能测试时，如果并发数比较大(比如最近项目需要支持1000并发)，单台电脑的配置(CPU和内存)可能无法支持，这时可以使用Jmeter提供的分布式测试的功能。  
**一、Jmeter分布式执行原理：**  
　　1、Jmeter分布式测试时，选择其中一台作为调度机(master)，其它机器做为执行机(slave)。  
　　2、执行时，master会把脚本发送到每台slave上，slave 拿到脚本后就开始执行，slave执行时不需要启动GUI，我理解它应该是通过命令行模式执行的。  
　　3、执行完成后，slave会把结果回传给master，master会收集所有slave的信息并汇总。  
**二、执行机（slave）配置：**  
　　1、slave机上需要安装Jmeter，具体如何安装这里不详细介绍了。  
　　2、添加环境变量：JMETER\_HOME=D:B\_TOOLSapache-jmeter-2.13，此处为你Jmeter的路径  
　　3、启动bin目录下的：jmeter-server.bat，启动成功如下图：



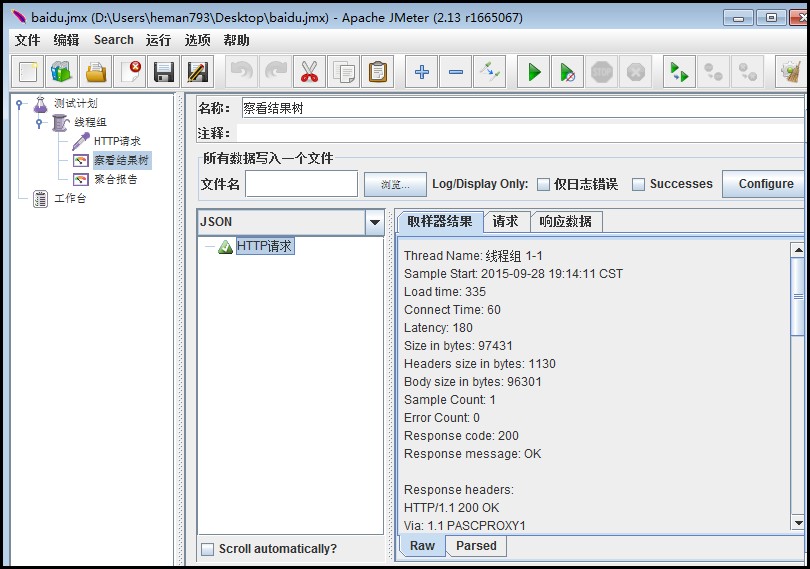
　　4、上图上标红的IP和端口会在master里配置时用到。IP就是slave机器IP，端口默认是1099，端口也可以自定义，这里我自定义为1000，这个后面会讲。  
　　5、多台slave的话，重复1~4步骤就好。   
　**三、调度机（master）配置：**  
　　1、脚本：简单的一个访问百度的脚本：



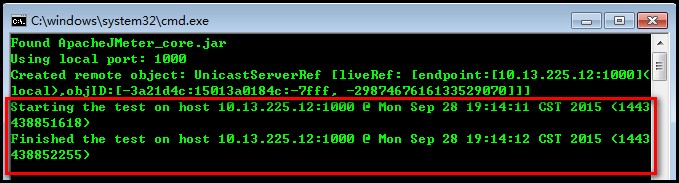
　　2、找到Jmeter的bin目录下jmeter.properties文件，修改如下配置，IP和Port是slave机的IP以及自定义的端口（这里端口我自定义为100，后面会讲如何自定义）：  
　　remote\_hosts=10.13.223.202:1000，10.13.225.12:1000  
　　多台slave之前用"，"隔开，我这配置了2台，可以看到标红的这个就是上面截图slave的IP和Port.  
　　3、打开Jmeter，选择运行，有运程启动、运程全部启动两个选项：



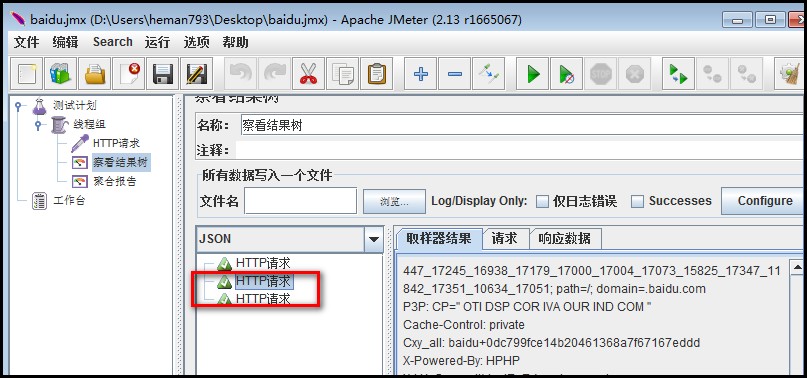
　　4、选择远程启动-->10.13.225.12:1000  
　　a) master结果，这里我只启动了10.13.225.12:1000这一台slave，所以只有一个结果(线程数和循环次数都是1)：



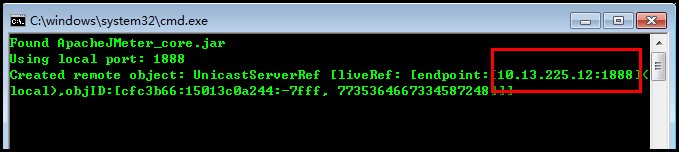
　　b) slave控制台信息：



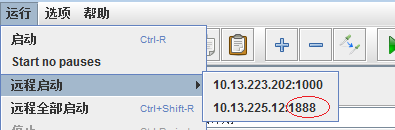
5、选择远程启动-->远程全部启动：  
　　a) master结果，全部启动，我配置了2台slave，所以有两次执行结果：



**四、自定义端口：**  
　　上面其实已经实现了Jmeter的分布式测试，这部分主要介绍下如何自定义slave端口：  
　　1、slave：在slave机的Jmeter的bin目录下，找到jmeter.properties文件，修改如下两个配置项，比如我这里修改为1888：  
　　server\_port=1888  
　　server.rmi.localport=1888  
　　2、启动slave机上的jmeter-server.bat，如下图，端口已经修改为：1888



　　3、master：修改master机器的jmeter.properties文件：  
　　remote\_hosts=10.13.223.202:1000，10.13.225.12:1888  
　　4、重启jmeter.bat，如下图，端口已经变了:



**五、其它说明：**  
　　1、调度机(master)和执行机(slave)最好分开，由于master需要发送信息给slave并且会接收slave回传回来的测试数据，所以mater自身会有消耗，所以建议单独用一台机器作为mater。  
　　2、参数文件：如果使用csv进行参数化，那么需要把参数文件在每台slave上拷一份且路径需要设置成一样的。  
　　3、每台机器上安装的Jmeter版本和插件最好都一致，否则会出一些意外的问题。