

性能测试

--性能测试工具JMeter使用—场景设计和运行

目录

- 场景设计
- 场景运行
 - GUI方式运行
 - 非GUI方式运行

场景设计

■ 回顾LoadRunner中场景的设置

- 目标场景
- Group
- Scenario
 - Real-world schedule（真实场景模式）
 - Basic shedule

场景设计

■ 目标场景

- Thread Group中设置
- 添加Aggregate Report（聚合报告）
- 预期结果与实际结果对比

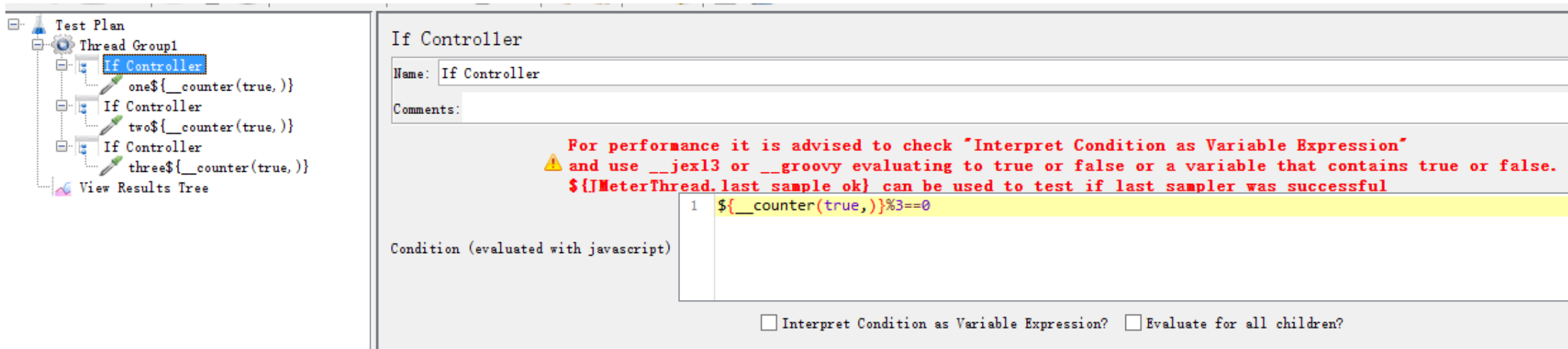
场景设计

- LoadRunner中Group方式，可以设置多少数量的用户做某个操作，或多少百分比的用户做某些操作
- 在JMeter中如何实现设置一部分用户做A操作，一部分用户做B操作，一部分用户做C操作.....

场景设计

- 加多个Thread Group，每个Thread Group设置计划设置的虚拟用户数
- If控制器配合线程迭代计数函数（`$_counter(true,)`）来控制
 - 实现每次调用计数器都会产生一个新值，从1开始每次累加1
 - true: 针对每个虚拟用户是独立的
 - false: 所有虚拟用户是公用的
- 举例：A，B，C三种类型的操作，30%的情况执行A操作，20%的情况执行B操作，50%情况执行C操作，加入脚本共迭代10次，如何设置？

场景设计



The screenshot shows the JMeter GUI. On the left, a tree view shows a 'Test Plan' containing a 'Thread Group' with three 'If Controller' elements. The first 'If Controller' is selected. The main panel shows the configuration for this 'If Controller'. The 'Name' field is 'If Controller'. The 'Comments' field is empty. A red warning message is displayed: 'For performance it is advised to check "Interpret Condition as Variable Expression" and use __jexl3 or __groovy evaluating to true or false or a variable that contains true or false. \${JMeterThread.last_sample_ok} can be used to test if last sampler was successful'. Below this, a list box contains a single item: '1 \${__counter(true,)}%3==0'. The 'Condition (evaluated with javascript)' label is visible. At the bottom, there are two checkboxes: 'Interpret Condition as Variable Expression?' (unchecked) and 'Evaluate for all children?' (unchecked).

Test Plan

- Thread Group1
 - If Controller
 - one\${__counter(true,)}
 - If Controller
 - two\${__counter(true,)}
 - If Controller
 - three\${__counter(true,)}
 - View Results Tree

If Controller

Name: If Controller

Comments:

For performance it is advised to check "Interpret Condition as Variable Expression" and use __jexl3 or __groovy evaluating to true or false or a variable that contains true or false. \${JMeterThread.last_sample_ok} can be used to test if last sampler was successful

1 \${__counter(true,)}%3==0

Condition (evaluated with javascript)

☐ Interpret Condition as Variable Expression? ☐ Evaluate for all children?

■ $\${__counter(true,)}\%5==0$

■ $\${__counter(true,)}\%2==0$

场景设计

The screenshot displays the JMeter interface during a test execution. On the left, the Test Plan tree shows a Thread Group with three If Controllers and a View Results Tree listener. The If Controllers are configured with the following code snippets:

- one\${__counter(true,)}
- two\${__counter(true,)}
- three\${__counter(true,)}

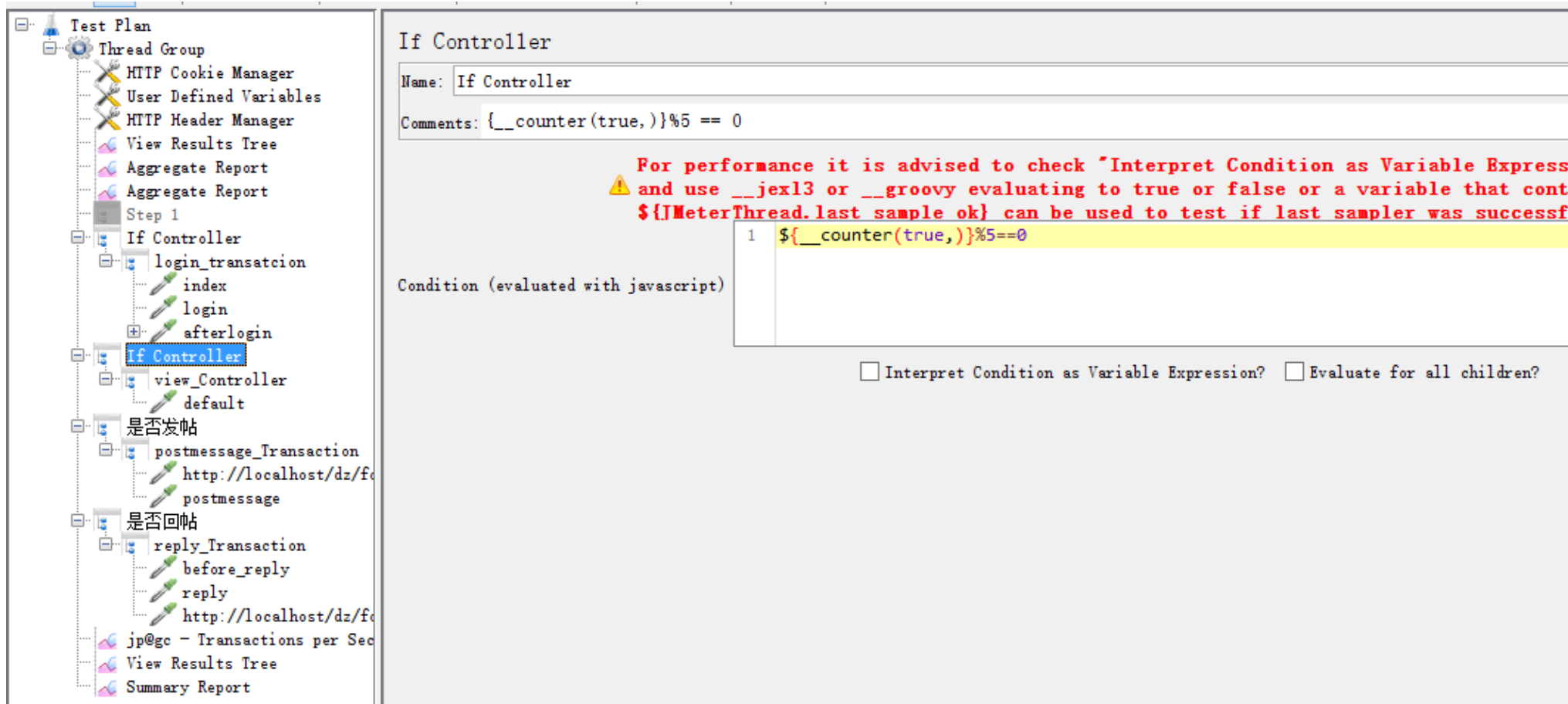
The View Results Tree listener is highlighted in blue. On the right, the View Results Tree panel shows the test results. The Name field is set to "View Results Tree". The Comments field is empty. The "Write results to file / Read from file" checkbox is checked. The Filename field is empty. The Search field is empty, and the Case sensitive and Regular checkboxes are unchecked. The Test Results table shows 10 samples, all of which are successful (indicated by green checkmarks). The results are grouped by thread:

Test	Sampler result
three1	
one1	
three2	
two1	
one2	
three3	
three4	
one3	
two2	
three5	

场景设计—实例

- 10%用户做登录, 20%用户浏览帖子, 50%用户发帖, 20%用户回帖

帖



The screenshot shows the JMeter Test Plan configuration. The left pane displays the Test Plan structure, including a Thread Group, HTTP Cookie Manager, User Defined Variables, HTTP Header Manager, View Results Tree, Aggregate Report, and Step 1. Step 1 contains an If Controller, which is expanded to show its children: login_transacton (with index, login, and afterlogin samplers), view_Controller (with default sampler), 是否发帖 (with postmessage_Transaction), and 是否回帖 (with reply_Transaction). The right pane shows the configuration for the selected If Controller. The Name is 'If Controller' and the Comments are '\$_counter(true,)%5 == 0'. A warning message is displayed: 'For performance it is advised to check "Interpret Condition as Variable Expression" and use __jexl3 or __groovy evaluating to true or false or a variable that contains the value of the last sample ok. \${JMeterThread.last_sample_ok} can be used to test if last sampler was successful.' The condition is set to '1 \$_counter(true,)%5==0'. The 'Interpret Condition as Variable Expression?' checkbox is checked, and the 'Evaluate for all children?' checkbox is unchecked.

Test Plan

- Thread Group
 - HTTP Cookie Manager
 - User Defined Variables
 - HTTP Header Manager
 - View Results Tree
 - Aggregate Report
 - Aggregate Report
 - Step 1
 - If Controller
 - login_transacton
 - index
 - login
 - afterlogin
 - If Controller
 - view_Controller
 - default
 - 是否发帖
 - postmessage_Transaction
 - http://localhost/dz/fc
 - postmessage
 - 是否回帖
 - reply_Transaction
 - before_reply
 - reply
 - http://localhost/dz/fc
 - jp@gc - Transactions per Sec
 - View Results Tree
 - Summary Report

If Controller

Name: If Controller

Comments: \$_counter(true,)%5 == 0

For performance it is advised to check "Interpret Condition as Variable Expression" and use __jexl3 or __groovy evaluating to true or false or a variable that contains the value of the last sample ok. \${JMeterThread.last_sample_ok} can be used to test if last sampler was successful.

1 \$_counter(true,)%5==0

Condition (evaluated with javascript)

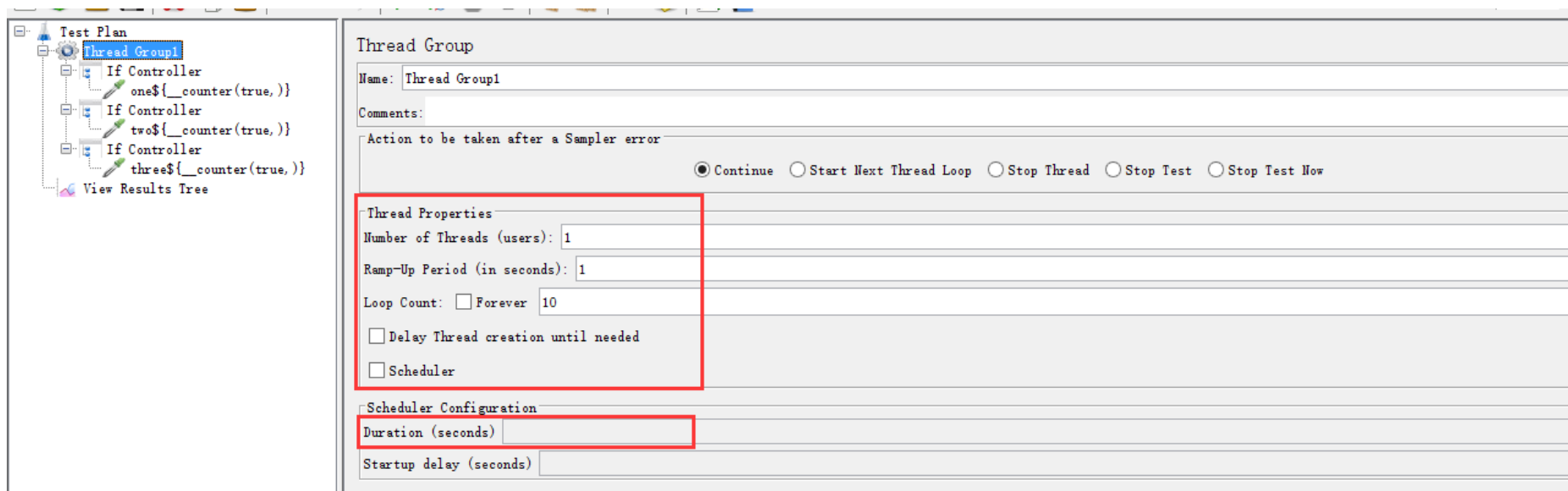
☒ Interpret Condition as Variable Expression? ☐ Evaluate for all children?

场景设计

■ Scenario

- 多长时间内启动多少线程
- 线程持续运行多长时间

■ 使用Thread Group设置项进行



场景设计

■ 举例：

- 10秒钟内启动100个线程
- 线程持续运行5分钟时间

目录

■ 场景设计

■ 场景运行

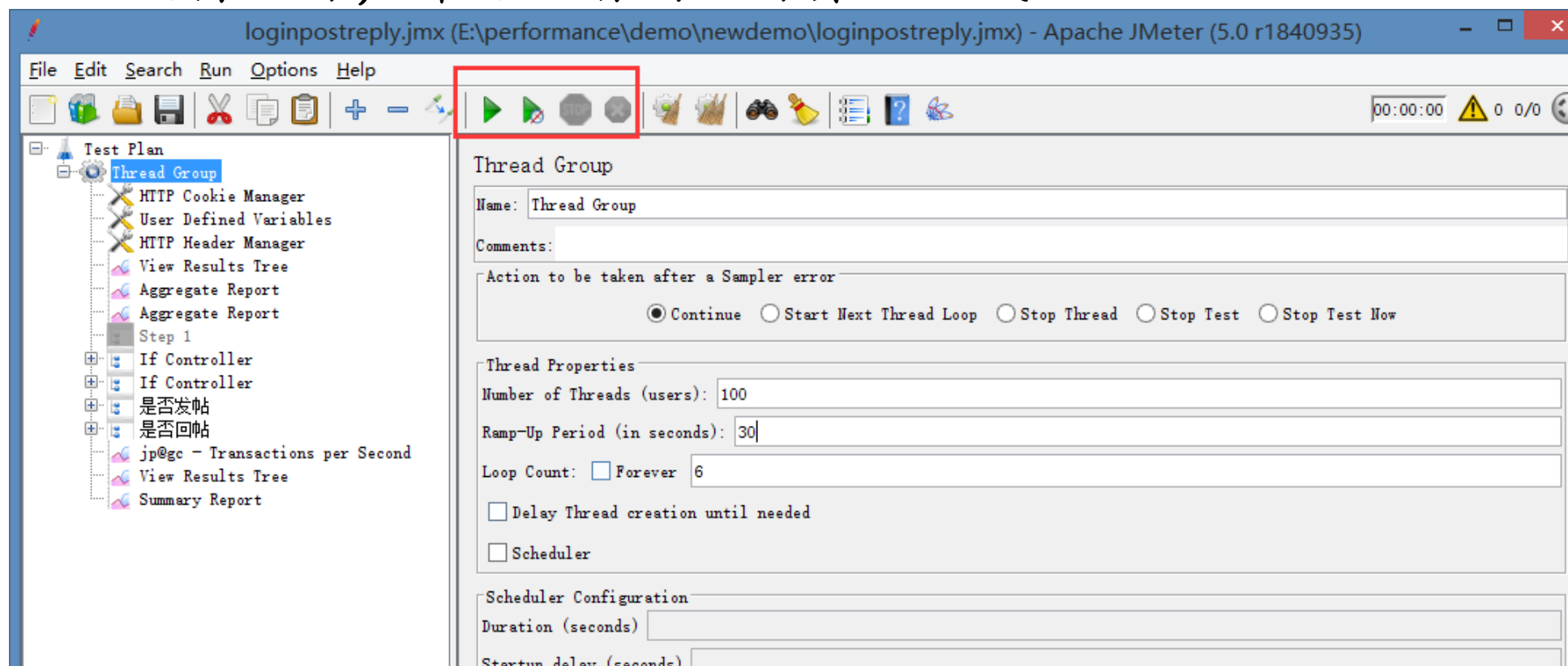
- GUI方式运行

- 非GUI方式运行

GUI方式运行

■ GUI方式

— 视窗运行，即可以看到运行界面方式



非GUI方式运行

■ 什么是非GUI方式

- 在命令窗口通过命令行的方式运行场景，没有图形界面

■ 为什么使用非GUI方式

- GUI方式会占用更多的系统资源
 - 影响负载量的增加
 - 响应时间增加（GUI方式本身占用一定时间）
- 性能测试自动化时更容易控制
- 非视窗（如：Linux）系统下命令方式运行更方便

非GUI方式运行

■ JMeter非GUI运行的命令

- 前提：系统环境变量设置jmeter_home，环境变量值是JMeter中bin所在的路径
- 命令：**java -jar %JMETER_HOME%\bin\ApacheJMeter.jar -n -t %JMETER_HOME%\script\login.jmx -l E:\performtest\result.jtl**
 - **java -jar**：运行某个JAR文件
 - **%JMETER_HOME%\bin**：ApacheJMeter.jar所在的目录
 - **-n**：非GUI方式运行

非GUI方式运行

- **-t:** 要运行的jmeter脚本
- **-l:**记录测试结果到文件，指定地址与名称，可以是相对路径，也可以是绝对路径
- **%JMETER_HOME%\script\login.jmx:** 要执行的文件路径和名称

非GUI方式运行

- **-r:**开启远程负载机，远程负载机列表在jmeter.properties中指定

```
254 # Remote hosts and RMI configuration
255 #-----
256
257 # Remote Hosts - comma delimited
258 remote_hosts=127.0.0.1,192.168.1.12
259 #remote_hosts=localhost:1099,localhost:2010
260
261 # RMI port to be used by the server (must start rmiregistry with same port)
262 #server_port=1099
```

- **-l:** 记录测试结果到文件，指定文件目录与名称，可以是相对路径，也可以是绝对路径

非GUI方式运行

■ 其他常用参数

- **-R**: 开启远程负载机, 可以指定负载机IP
- **-X**: 停止远程执行
- **-h**: 调出帮助

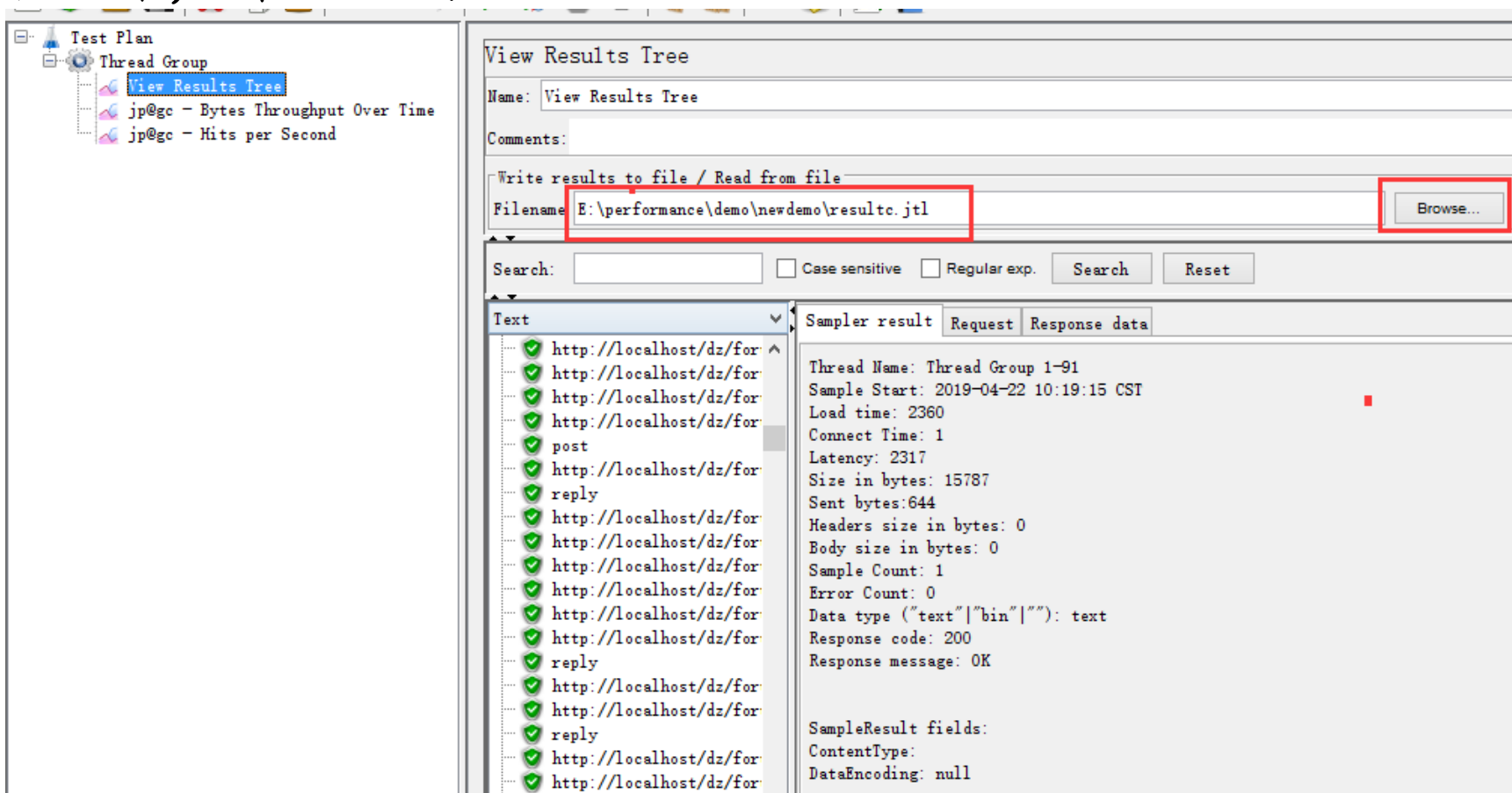
非GUI方式运行

- **jmeter -n -t d:\demo\test.jmx -l d:\result\test.jtl -R 192.168.1.100:1099,192.168.1.101:1099 -X**
 - 使用非GUI模式在远程主机192.168.1.100和192.168.1.101主机上运行test.jmx这个脚本，并将测试结果存入test.jtl文件内，脚本运行完自动退出远程主机
 - 注意：需要在远程主机上开启jmeter-server.bat文件

非GUI方式运行

■ 生成的jtl文件怎样查看

— 使用监听器打开文件，即可查看



内容总结

■ 场景设计

■ 非GUI方式运行

- 什么是非GUI方式运行
- 为什么使用非GUI方式运行
- 怎样使用非GUI方式运行
 - 命令
 - 各参数含义
 - 生成的结果文件怎样查看



Question
