

# 性能测试

--性能测试工具JMeter使用（关联、检查点、事务、定时器）

# 内容回顾

---

- JMeter介绍
- JMeter安装（需要有JDK8以上的环境）
- JMeter基本使用：体系构成、运行原理、基本元件类型
- 脚本产生的方法
  - 手动书写
  - Badboy录制
  - 使用代理录制

# 内容回顾

---

## ■ JMeter参数化

- 什么是参数化
- 为什么进行参数化
- 怎样进行参数化
  - **\_CSVRead()** : 读取CSV数据函数
  - **CSV Data Set Config** CSV 数据控件
  - **User Defined Variables**: 用户自定义变量

# 内容回顾

---

- Thred Group 中配置项的含义
- HTTP Request中配置项含义
- CSV Data Set Config注意：
  - 当参数不够用时，怎样处理
  - 当参数多于需要请求的数量时，怎样处理

# 目录

---

- 关联
- 检查点
- 事务
- 定时器

# 关联

## ■ 思考

- 什么是关联
- 什么时候需要使用关联
- 关联怎样实现

# 关联

## ■ 定义（LoadRunner中）

- 脚本回放过程中，客户端发出请求，通过一些规则，在服务器所响应的内容中查找，得到相应的值，以参数的形式替换录制时的静态值，从而向服务器发出正确的请求，这种动态获得服务器响应内容的方法被称作关联

## ■ 定义（JMeter中）

- 利用JMeter元件来获取服务器返回的动态数据

# 关联

## ■ 什么情况需要使用关联

- 当脚本中的数据每次回放都发生变化时，并且这个**动态数据在后面的请求中需要**发送给服务器，那么这个内容需要通过关联来询问服务器，获得该数据的变化结果。例如：

- 登录字符串。带有会话 ID 或时间戳等动态数据的登录字符串
- 日期/时间戳。使用日期或时间戳或者其他用户凭据的任意字符串
- 常见前缀：如 SessionID 或 CustomerID



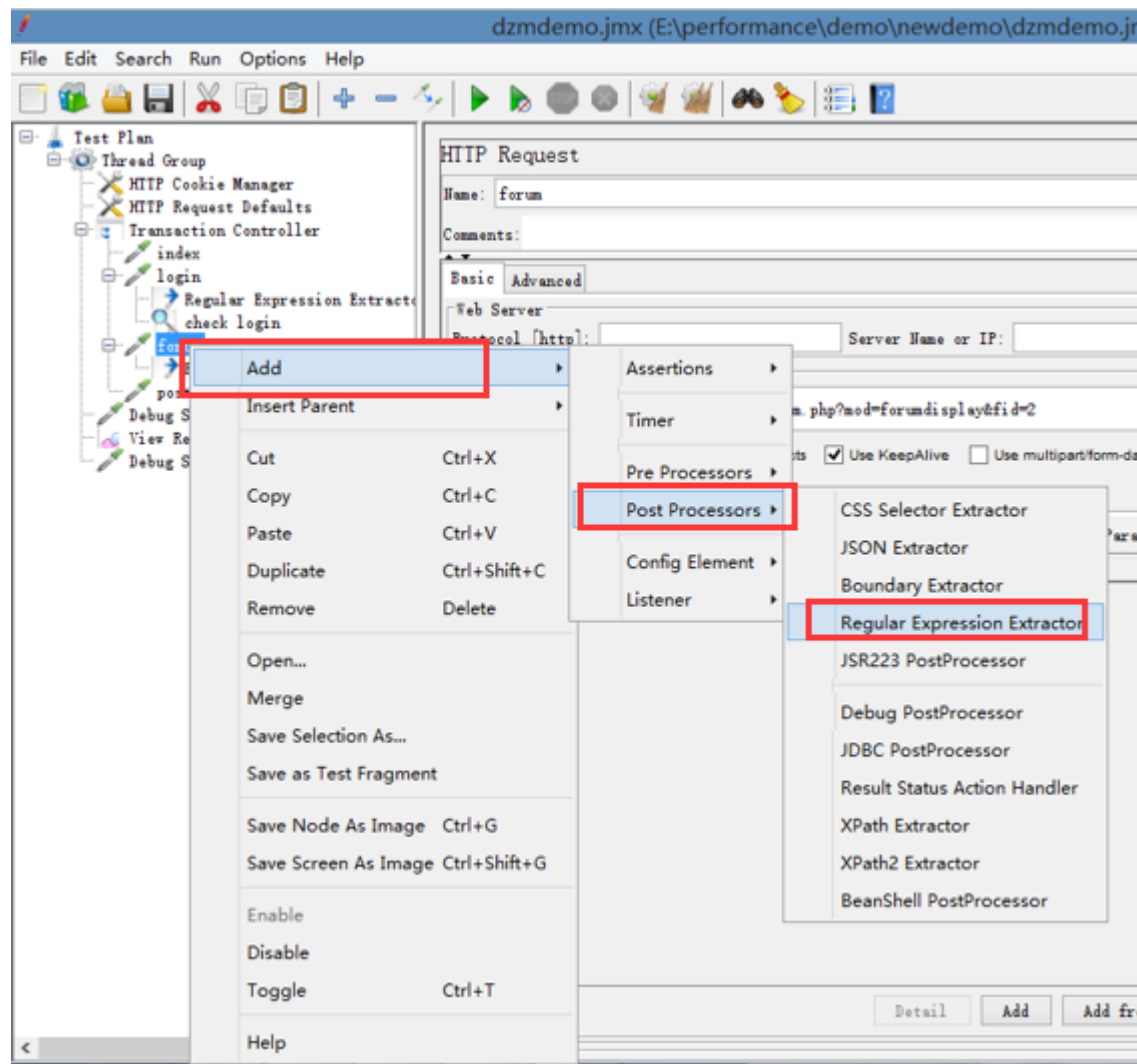
# 关联

## ■ JMeter中关联怎样实现

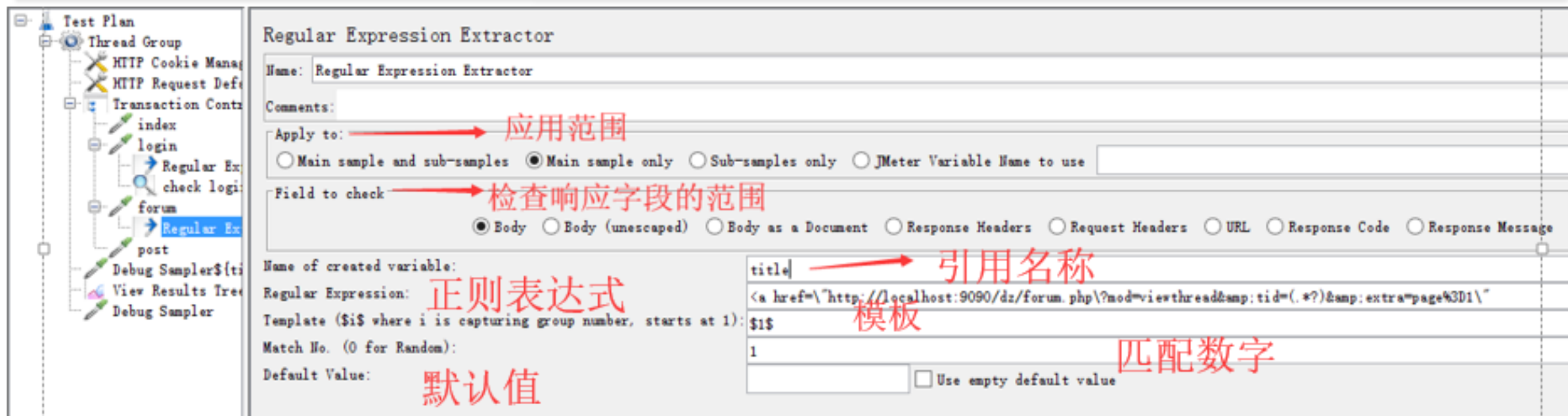
- 通过JMeter中的**正则表达式提取器**所定义的规则，在服务器所响应的内容中查找，得到相应的值，以变量的形式替换静态值，从而向服务器发出正确的请求

# 关联怎样实现

## ■ 添加关联的步骤



# 关联怎样实现



The screenshot shows the JMeter Regular Expression Extractor configuration window. Red arrows and text annotations highlight key fields:

- 应用范围** (Application Scope): Points to the "Apply to:" section, where "Main sample only" is selected.
- 检查响应字段的范围** (Check response field scope): Points to the "Field to check:" section, where "Body" is selected.
- 引用名称** (Reference Name): Points to the "Name of created variable:" field, which contains "title".
- 正则表达式** (Regular Expression): Points to the "Regular Expression:" field, which contains a regex pattern.
- 模板** (Template): Points to the "Template (\$i\$ where i is capturing group number, starts at 1):" field, which contains "\$1\$".
- 匹配数字** (Match Number): Points to the "Match No. (0 for Random):" field, which contains "1".
- 默认值** (Default Value): Points to the "Default Value:" field, which is empty.

- 引用名称：下一个请求要引用的参数名称，如填写title，则可用 `${title}` 引用它
- 正则表达式中()括起来的部分就是要提取的

# 关联怎样实现

---

## ■ 正则表达式:

- 使用单个字符串来描述、匹配一系列符合某个句法规则的字符串

# 正则表达式

具有特殊含义的字符：

( 和 )：界定期望获取字符串的匹配模式

. (字符点)：匹配任何单个字符

+：一次或者多次

\*：匹配0次或多次的项目

?：找到匹配的结果后立刻停止查找

\：转义字符

[]：匹配符合 [] 内的字符

# 正则表达式

具有特殊含义的字符：

[0-9] 匹配所有数字字符

[a-z] 匹配所有小写字母字符

[^0-9] 匹配所有非数字字符

[^a-z] 匹配所有非小写字母字符

^ 匹配字符开头的字符

\$ 匹配字符结尾的字符


## 关联怎样实现

- 模板：用\$\$引用起来，如果在正则表达式中有多个正则表达式（多个括号括起来的部分），则可以是\$2\$，\$3\$等等，表示解析到的第几个值给title
- 匹配数字：0代表随机，-1代表所有，其余正整数代表将在检查的内容中，第几个匹配的内容提取出来

# 关联怎样实现

## ■ 实例一：练习正则表达式的使用，从文件中获取Session ID

### — 文件内容：

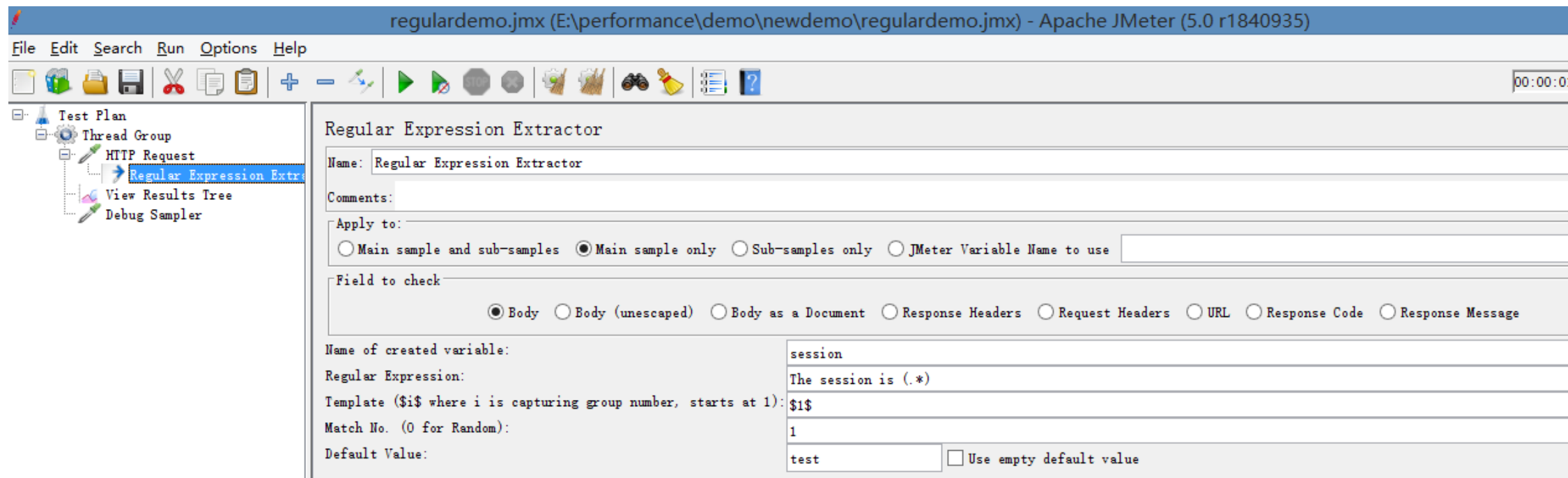


```
This is a file.  
The session is kfdskf45873ktjkrtj48594  
this is a file.  
that is a file.  
the second session is 12345.
```



# 关联怎样实现

## ■ 配置正则表达式的方式：



# 关联怎样实现

## ■ 需求：

- 1 手动书写访问、登录Discuz论坛脚本
- 2 登录成功后，请求默认板块页面
- 3 在默认板块里，点击某一个帖子名称，查看其内容（请求查看帖子详情）

备注：正则表达式书写是否正确，可以借助于工具Register验证工具

demo:classdzmcorr

# 关联怎样实现

dzmdemo.jmx (E:\performance\demo\newdemo\dzmdemo.jmx) - Apache JMeter (5.0 r1840935)

File Edit Search Run Options Help

Test Plan

- Thread Group
  - HTTP Cookie Manager
  - HTTP Request Defaults
  - Transaction Controller
    - index
    - login
    - Regular Expression E:
      - check login
      - forum
        - Regular Expression E:
          - post
      - Debug Sampler\${title}
      - View Results Tree
      - Debug Sampler

Regular Expression Extractor

Name: Regular Expression Extractor

Comments:

Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only ☐ JMeter Variable Name to use

Field to check

☒ Body ☐ Body (unescaped) ☐ Body as a Document ☐ Response Headers ☐ Request Headers ☐ URL ☐ Response Code ☐ Response Message

Name of created variable: title

Regular Expression: <a href=\"http://localhost:9090/dz/forum.php?mod=viewthread&tid=(\${title})&extra=page%3D1\"

Template (\$i\$ where i is capturing group number, starts at 1): \$1\$

HTTP Request

Name: post

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP:

HTTP Request

Method: GET Path: /dz/forum.php?mod=viewthread&tid=\${title}&extra=page%3D1

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

# 目录

---

- 关联
- 检查点
- 事务
- 定时器

# 检查点

## ■ 什么是检查点

- 检查服务器返回的信息是否正确

## ■ 什么情况用检查点

- 验证服务器返回是否正确

## ■ JMeter中怎样使用检查点

- 断言组件实现此功能

## 检查点

---

- 需求：验证用户是否成功登录Discuz论坛
- 分析：用户登录成功后，服务器响应内容有哪些数据
- 解析：使用响应断言对服务器的响应数据进行规则匹配

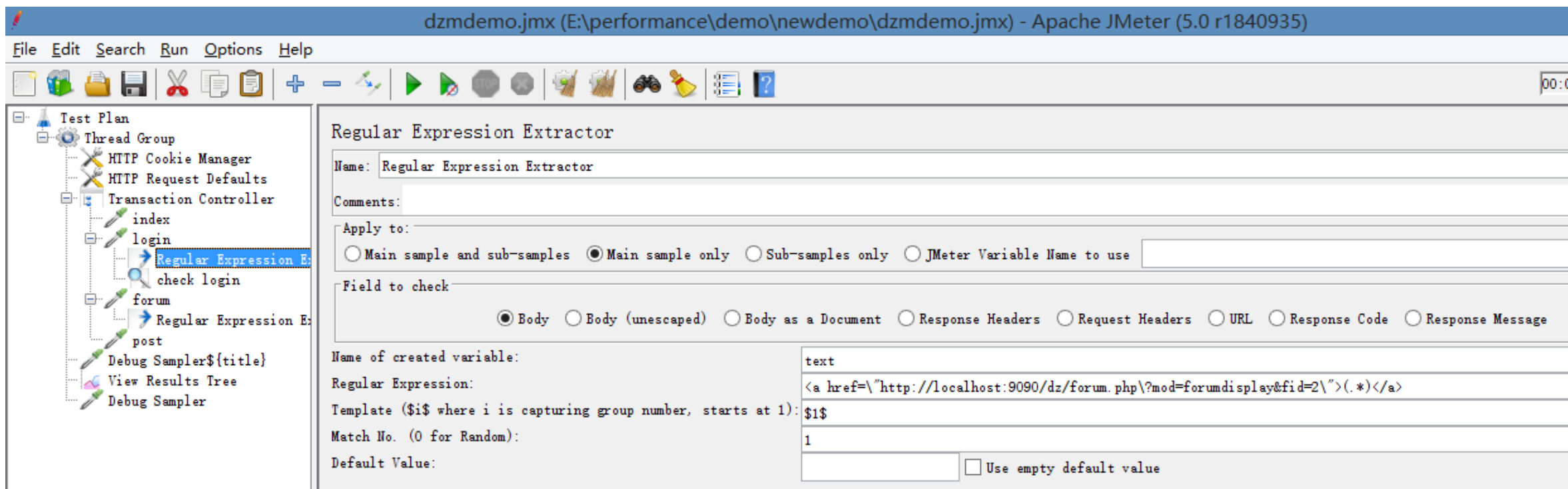
# 检查点

## ■ 检查点设置步骤

- 1 在相应请求后，添加正则表达式，将匹配数据存入某个变量
- 2 添加响应断言，响应断言中引用正则表达式的变量名称
- 3 添加断言结果监听器

# 检查点

- 通过正则表达式找出响应数据中的关键字存入变量text中，使用断言判断是否匹配到





# 检查点

## ■ 添加响应断言

dzmdemo.jmx (E:\performance\demo\newdemo\dzmdemo.jmx) - Apache JMeter (5.0 r1840935)

File Edit Search Run Options Help

Test Plan

- Thread Group
  - HTTP Cookie Manager
  - HTTP Request Defaults
  - Transaction Controller
    - index
    - login
      - Regular Expression E...
      - check login
    - forum
      - Regular Expression E...
    - post
    - Debug Sampler\${title}
    - View Results Tree
    - Debug Sampler

Response Assertion

Name: check login

Comments:

Apply to:

☒ Main sample and sub-samples ☐ Main sample only ☐ Sub-samples only ☐ JMeter Variable Name to use

Field to Test

☒ Text Response ☐ Response Code ☐ Response Message ☐ Response Headers

☐ Request Headers ☐ URL Sampled ☐ Document (text) ☐ Ignore Status

☐ Request Data

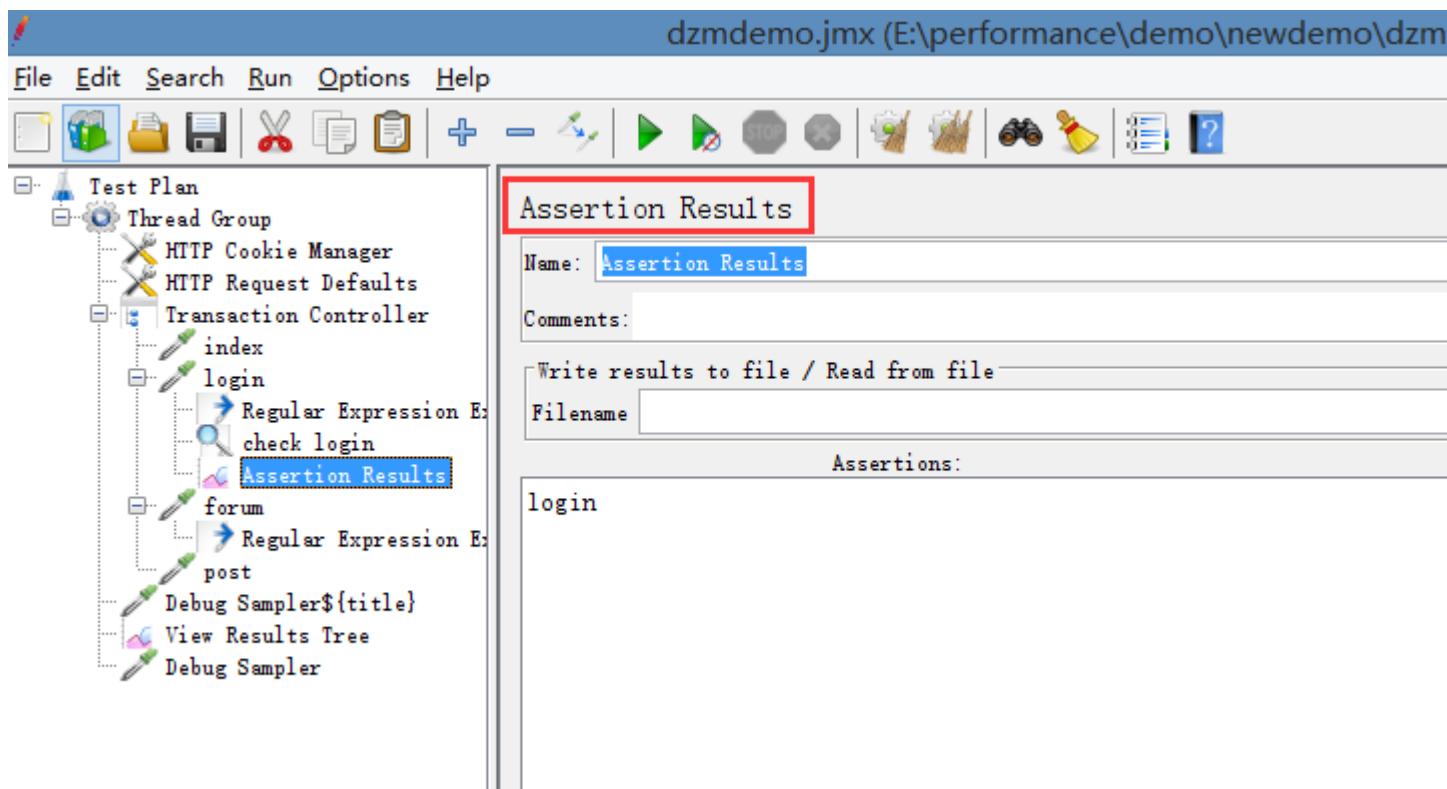
Pattern Matching Rules

☐ Contains ☐ Matches ☐ Equals ☒ Substring ☐ Not ☐ Or

Patterns to Test

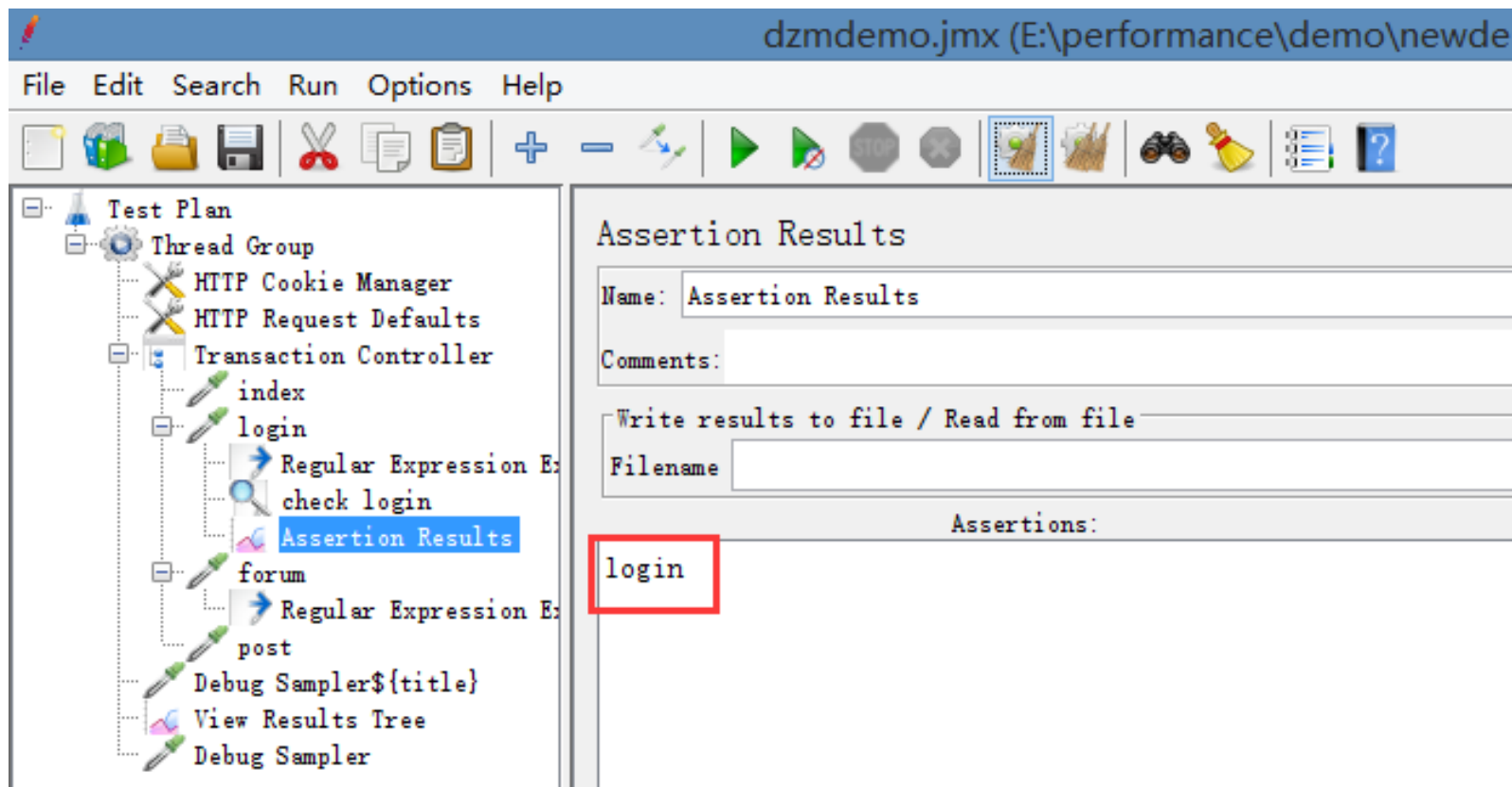
	Patterns to Test
1	\${text}

## ■ Listener---Assertion Results



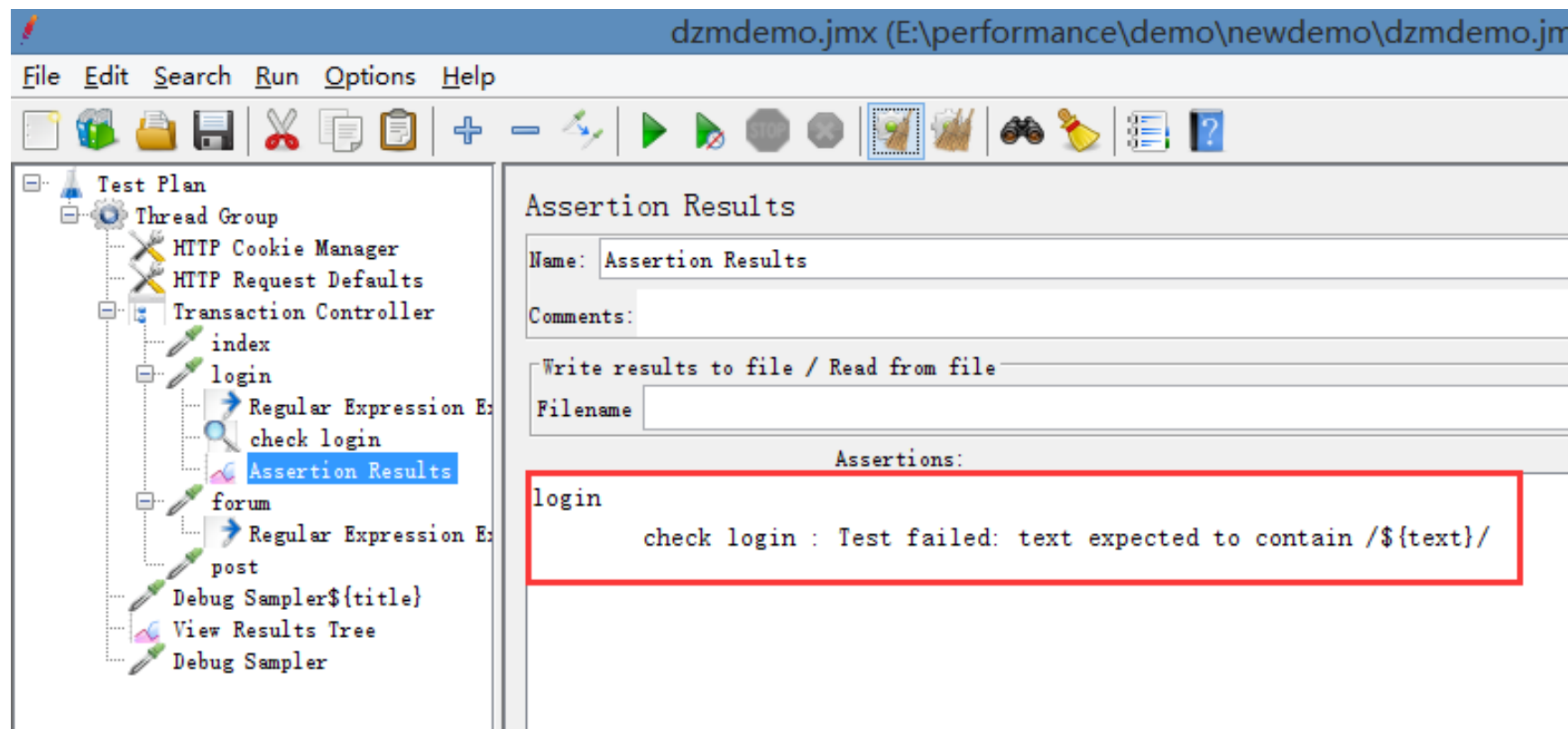
# 检查点

## ■ 断言成功



# 检查点

## ■ 断言失败



# 断言配置项解释

## ■ 配置项解释

- **Apply to:** 添加的断言可以用在哪
- **Main sample and sub-samples** （主请求和子请求都适用）
- **Main sample only** （仅主请求）
- **Sub-samples only** （仅子请求）
- **JMeter Variable** （JMeter的变量）

# 断言配置项解释

## ■ Response Field to Test（要测试的响应字段）

- **Text Response**, 在响应数据中进行校验, 排除Headers其他选项说明
- **Response Code**:匹配HTTP状态码
- **Response Message**:匹配响应信息, 如处理成功返回“成功”、OK”字样
- **Response Headers**:匹配响应头信息
- **Request Headers**:匹配请求头信息

# 断言配置项解释

- **Document(text):** 匹配文档内容
- **URL Sampled:**匹配URL链接
- **Ignore Status:**一个请求有多个响应断言，其中第一个响应断言选中此项，当第一个响应断言失败时可以忽略此响应结果，继续进行下一个断言，如果下一个断言成功还是可以判断事务成功的
- **Request Data:** 匹配请求数据

# 断言配置项解释

## ■ Patten Matching Rules（模式匹配规则）

- 选择**Contains**（包括），只要响应数据中包含需要匹配的内容代表响应成功，支持正则表达式其他选项说明
- **Matches**:响应数据需要完全匹配需要匹配的内容即代表响应成功，支持正则表达式



## 断言配置项解释

- **Equals:**响应数据需要完全等于需要匹配的内容才代表响应成功，需要匹配的内容是字符串，不支持正则表达式
- **Substring:**响应数据包含需要匹配的内容才代表响应成功，需要匹配的内容是字符串，不支持正则表达式
- **Not:**匹配字符串时，选择此项，对大小写不敏感
- **Or:** 多个要匹配的模式默认是以AND（逻辑与）连接起来的；勾选Or则是以逻辑或连接

# 断言配置项解释

---

## ■ Patterns to Test（要测试的模式）

- 填入需要匹配的字符串或正则表达式

# 其他断言类型

## ■ JSON断言

- 如果服务器响应返回的是JSON格式的内容,这时最佳的断言验证类型就是使用

## ■ XPath断言

- 如果服务器响应返回的是XML格式的内容,这时最佳的断言验证类型就是使用

# 其他类型断言

---

## ■ Duration to Assert

- 允许的响应时间的最大值，即断言的持续时间

## ■ Size to Assert

- 对于返回结果文件大小的标准定义

# 目录

---

- 关联
- 检查点
- 事务
- 定时器

# 事务

---

## ■ 什么是事务

- 模拟用户的一个相对完整的、有意义的业务操作过程

## ■ 什么情况下使用事务

- 查看某系列操作的使用时间

## ■ 怎样使用事务

## ■ 对比LoadRunner中事务的使用

# 事务

## ■ 怎样使用事务

### — 事务控制器

- 生成一个**额外的样本**，用于测量执行嵌套测试元素所花费的**总时间**
- **The Transaction Controller generates an additional sample which measures the overall time taken to perform the nested test elements (官方文档)**

# 事务控制器怎样使用

## ■ 举例：

- 添加线程组
- 添加HTTP 请求采样器（两个或更多）
- 添加结果查看树



# 事务控制器怎样使用

transactiondemoa.jmx (E:\performance\demo\newdemo\transactiondemoa.jmx) - Apache JMeter (5.0 r1840935)

File Edit Search Run Options Help

Test Plan  
Thread Group  
Transaction Controller  
one  
two  
Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
one	1	147	147	147	0.00	0.00%	6.8/sec	18.47	0.75	2781.0
two	1	3300	3300	3300	0.00	0.00%	18.2/min	204.90	0.03	692402.0
Transaction Con...	1	3447	3447	3447	0.00	0.00%	17.4/min	196.78	0.06	695183.0
TOTAL	3	2298	147	3447	1522.17	0.00%	52.2/min	393.56	0.13	463455.3

取样器名称 事务数量 请求事务平均响应时间 响应时间的标准方差

# 事务控制器怎样使用

## ■ 实例

- 勾选 **Generate parent sample**
- 增加等待时间 (**Constant Timer**)
- 勾选 **Include duration of timer and pre-post processors in generated sample**

# 事务控制器怎样使用

## ■ 勾选 Generate parent sample

- 将该事务控制器作为一个父样本
- 勾选了之后，在聚合报告和Summary Report这两个监听器中，只会显示事务控制器的统计信息，不会显示子请求的统计信息。不过在察看结果树这个监听器里面，还是会显示所有组件的执行信息的，勾选了这个选项后，事务时间统计时不会包含浪费时间

# 事务控制器怎样使用

- 勾选 Include duration of timer and pre-post processors in generated sample
  - 事务控制器的时间包含各种时钟的时间和前置处理器执行的时间
  - 一般来说这个选项是不勾选的，我们在统计事务时间时，是不需要统计思考时间，集合等待时间和前置处理时间等的

# 事务控制器怎样使用

- 如果不勾选Generate parent sample，则事务时间中会统计浪费时间
  - 如：Java请求1耗时245毫秒，Java请求2耗时255毫秒，但是事务控制器统计出来的时间是502毫秒，比 $245+255=500$ 毫秒多出来了两毫秒
  - 原因：
    - JMeter运行事务控制器、Java请求本身是要消耗时间的

# 事务控制器怎样使用

- 如果发现事务控制器的时间比所有子组件的时间之和差距过大
  - 那么就说明JMeter本身的性能出现问题了
- 可以考虑通过如下三种方式进行处理：
  - 1、修改JMeter.bat文件，调整JVM参数，将heap和permsize值适当设置大一点
  - 2、联机负载，减少单台机器上的负载线程数
  - 3、采用命令模式运行测试

# 目录

---

- 关联
- 检查点
- 事务
- 定时器

# 定时器

---

- 什么叫定时器
- 常用定时器的使用
  - 固定定时器 (Constant Timer)
  - 同步定时器 (Synchronizing Timer)



# 什么叫定时器

---

## ■ 定时器 (Timer)

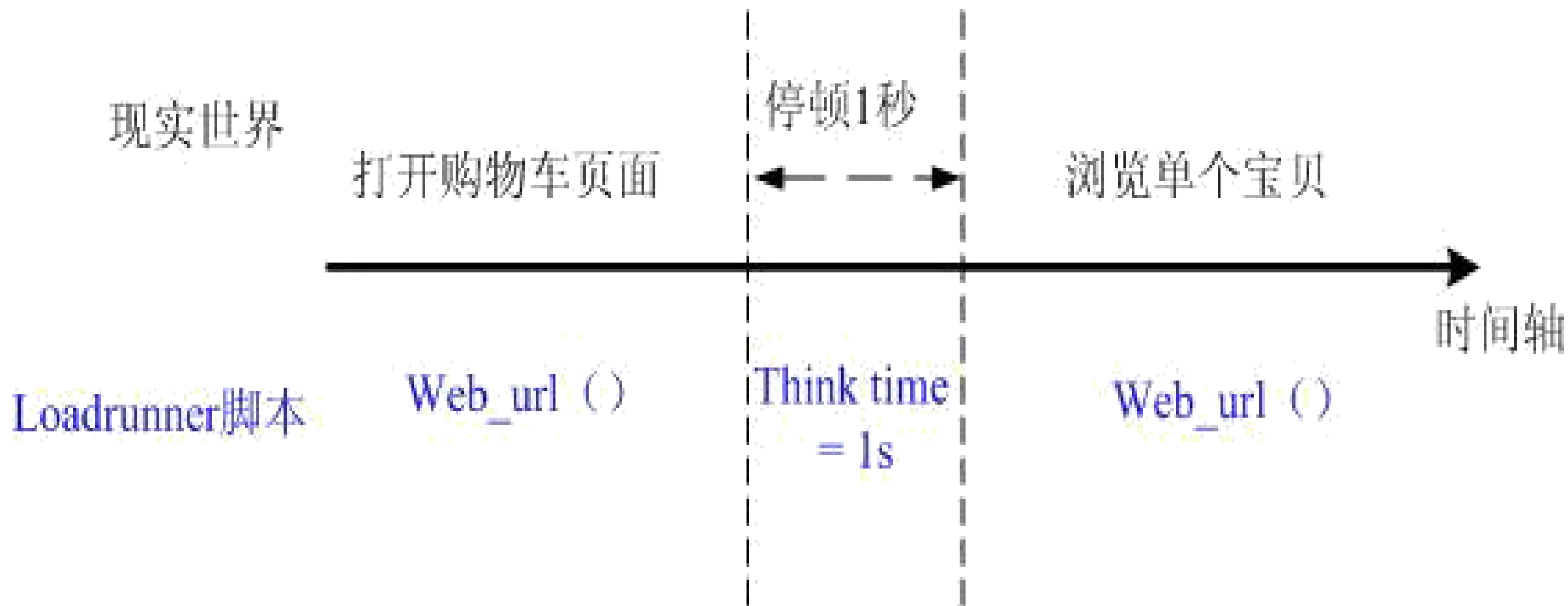
- 负责定义请求之间的延迟间隔

# 定时器

## ■ 常用定时器

- 固定定时器
- 同步定时器（集合点）
- 高斯随机定时器
- 泊松随机定时器
- BeanShell 定时器
- .....

## 固定定时器的使用



# 固定定时器的使用

Constant Timer

Name: Constant Timer

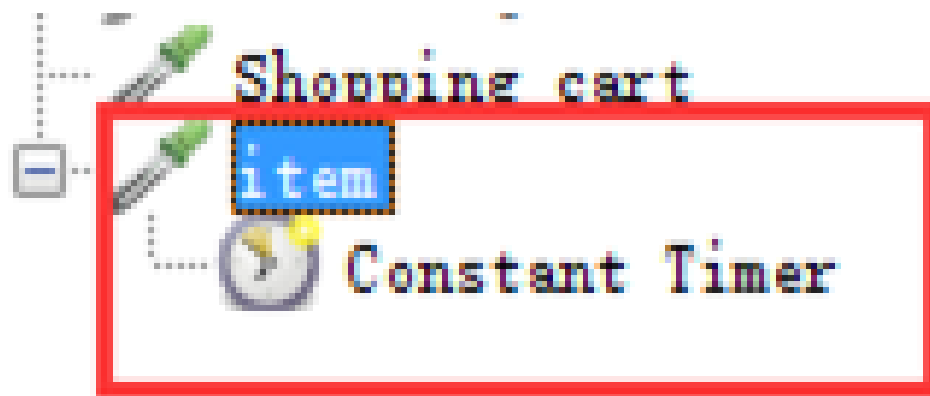
Comments:

Thread Delay (in milliseconds): 2000

# 固定定时器

## ■ 放在什么位置

- 在Thread Group中添加，所有请求发出前都等待设置的那些时间
- 需要让哪个请求等待，则作为其子节点加入



# 同步定时器

## ■ 同步定时器

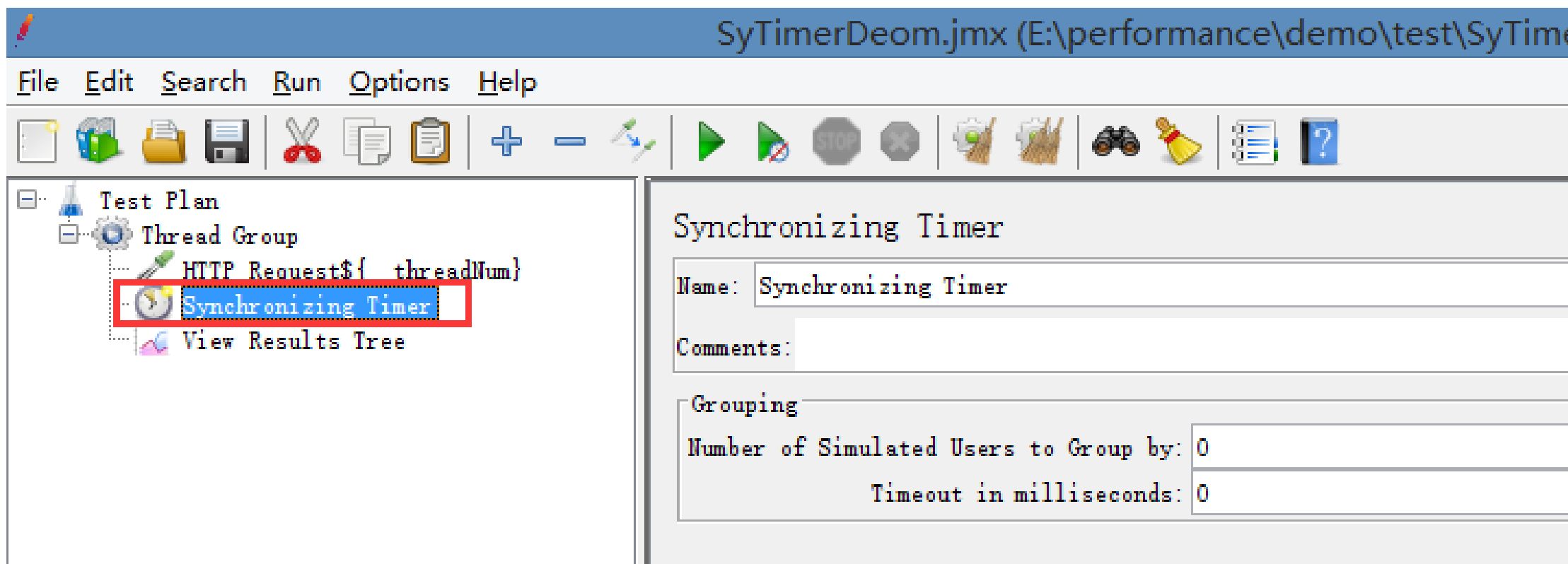
- 用于同步虚拟用户恰好在某一时刻执行任务，确保用户更准确、集中的进行某个指定操作，达到更理想的负载模拟效果

## ■ 为什么用同步定时器

- 更有针对性地对某个可能存在性能问题的模糊或子系统施压，以便找到性能瓶颈

# 同步定时器的使用

## ■ 直接在Thread Group上添加即可



# 同步定时器使用

## ■ 实例演示

- 0 添加线程组（500个线程）
- 1 不使用同步定时器，设置Http Request Sample,向10.7.1.9:8686/zentao/发请求
- 2 使用同步定时器，重复步骤1
- 两个做的过程中都监控10.7.1.9上资源利用率以及请求结果
  - 设置同步定时器后有部分请求不能请求成功，说明理论上所有线程都准备好后，一齐请求服务器，瞬间产生较大压力



# 同步定时器使用

## ■ Number of Simulated Users to Group by:每次释放的线程数量

- 设置为0，等同于设置为线程组中的线程数量
- 设置为10，表示等待10个用户到达后，再一起并发请求

# 同步定时器使用

## ■ Timeout in milliseconds: 默认为0

- 设置为0, Timer将会等待线程数达到了"Number of Simultaneous Users to Group"中设置的值才释放
- 大于0, 那么如果超过Timeout in milliseconds中设置的最大等待时间(毫秒为单位)后还没达到"Number of Simultaneous Users to Group"中设置的值, Timer将不再等待, 释放已到达的线程
- 如果设置为0, 且线程数量无法达到"Number of Simultaneous Users to Group by"中设置的值, 那么Test将无限等待, 除非手动终止

# 同步定时器的使用

- 同步也是相对的，即使进行了设置，假设1000个用户提交订单也无法在几毫秒内完成
  - 可能需要几十毫秒甚至几百毫秒完成
  - 不进行设置，1000个用户提交订单可能在几秒甚至几十秒才完成

# 内容回顾

---

## ■ 关联

- 基本概念、什么情况使用、怎样使用

## ■ 检查点

- 基本概念、什么情况使用、怎样使用

## ■ 事务

- 基本概念、什么情况使用、怎样使用

## ■ 定时器

- 基本概念、什么情况使用、怎样使用



# Question

---