实验10Java工程性能测试

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 组 别 |  | 实验人员 | KAFLE SAMRAT | | | | |
| 实验项目名称 | Java工程性能测试 | | | | | | |
| 实 验 类 别 | 验证型□ 设计型□ 综合型□ | | | | | | |
| 实 验 工 具 | Jmeter、eclipse | | | 实验地点 | J13-132 | 实验日期 | 2020.12 |
| 学生实验小结 | 通过Jmeter对java工程的性能测试，我发现jmeter是一款强大的性能测试软件，可以通过模拟多个用户同时对java工程进行测试，通过检测处理时间来分析java项目性能好坏，结果清晰明了。 | | | | | | |
| 教师评语 | 指导教师签名：  年 月 日 | | | | | | |
| 实验成绩 |  | | | | | | |

1实验内容和目标

1.1实验内容

本实验主要是在eclipse中编写jmeter辅助函数，形成jar包放入jmeter中，并根据要求设置多个虚拟用户，对Java工程进行性能测试。

1.2实验目标

（1）熟悉jmeter辅助函数；

（2）掌握对Java工程性能测试。

2实验方案设计

2.1实验环境

JMeter+eclipse

2.2实验方法设计

通过eclipse编写测试代码，通过jmeter对java请求进行测试。

3实验步骤和结果

Understand the JMeter helper functions

(1) getDefaultParameters(): It is mainly used to obtain interface parameters, and can also set the incoming parameters.

(2) setupTest(): This method is used to do the necessary work before the test, such as initializing the variables needed in the test script.

(3) runTest(): This is the main method of the test, and the logic to be tested should be written here. Each thread executes this method in a loop, and of course how many threads are executing this method, for how long, and how many times, depending on the Settings in the thread group.

(4) tearDownTest (): This method executes after each thread has finished all of its test work, kind of like a finally function. For example, if I open a connection to a database, I drop it after all the threads have finished their work. Really have no what can do, do not write of course also can. This is a brief introduction to Figure 1

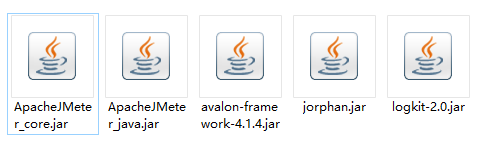
The order of execution is:

getDefaultParameters() --> setupTest(JavaSamplerContext context) --> runTest(JavaSamplerContext context) --> teardownTest(JavaSamplerContext context)

Write JMeter helper functions

(1) Create a new Java project in Eclipse called jmeter\_test, and create a package called com.jmeter\_test

(2) Import the JAR package sent by the teacher to be used in the experiment.



(3) Create a new jmeter\_test class and write it

**package** com.jmeter.test;

**import** java.io.FileNotFoundException;

**import** org.apache.jmeter.protocol.java.sampler.JavaSamplerContext;

**import** org.apache.jmeter.samplers.SampleResult;

**import** com.sun.org.slf4j.internal.Logger;

**import** com.sun.tools.javac.main.Arguments;

**public** **class** jmeter\_test {

**private** Logger log = getLogger();

**private** SampleResult result;

**private** String testStr;

**private** String resultData;

**public** Arguments getDefaultParameters()

{

log.info("execute getDefaultParameter");

org.apache.jmeter.config.Arguments params = **new** Arguments();

params.addArgument("testStr","aaa");

params.addArgument("testString","abc");

**return** params;

}

**public** **void** setupTest(JavaSamplerContext context)

{

log.info("start execute setupTest...");

result = **new** SampleResult();

testStr = context.getParameter("testStr");

**if** (testStr != **null** && testStr.length()>0)

result.setSamplerData(testStr);

}

**public** SampleResult runTest(JavaSamplerContext context)

{

log.info ("execute runTest..");

String a = context.getParameter("testStr");

String b = context.getParameter("testString");

result.setSampleLabel("java请求");

result.sampleStart();

**if** (a.equals(b))

resultData = "两字符串是一样的";

**else**

resultData = "两字符串是一样";

result.setResponseData("结果是： "+resultData, **null**);

result.setDataType(SampleResult.***TEXT***);

**if** (testStr.length()<0)

{

log.info("fail...");

result.SetSuccessful(**false**);

}

**else**

{

log.info("TestString is \t"+testStr);

result.setSuccessful(**true**);

}

**return** result;

}

**public** **static** **void** main(String[] args) **throws** FileNotFoundException {

// **TODO** Auto-generated method stub

org.apache.jmeter.config.Arguments params = **new** Arguments();

params.addArgument("testStr","你好");

JavaSamplerContext context = **new** JavaSamplerContext(params);

jmeter\_test test = **new** jmeter\_test();

test.setupTest(context);

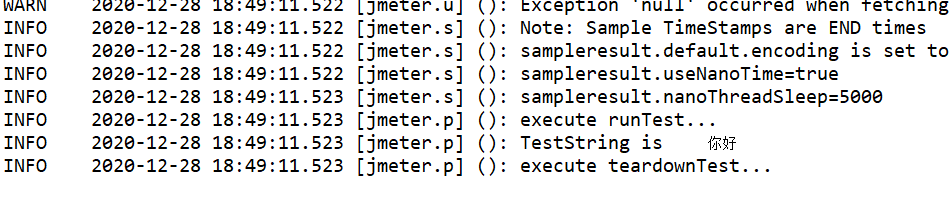
test.runTest(context);

test.teardownTest(context);

}

}

(4) Test the console of the written program through Main function

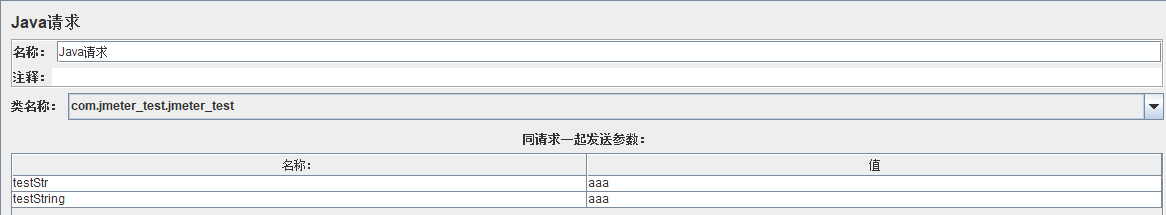


(5) Export the project as a JAR package and save it in the lib path of JMeter

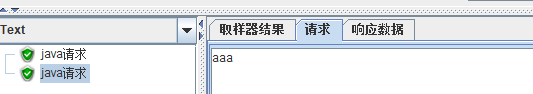
Perform performance tests on Java projects

(1) Add and set thread group.

Right-click on the thread group and add a Java request from Sampler



(3) Add "view result tree" and view test results：



By writing the Java project test script, writing the code to be tested in runTest(), and performing the performance test using JMeter, the task prints out the parameters we will pass.

1. On the basis of the code in Experiment 3, add a string testString and add it to params in getDefaultParameters() method, so that JMeter can get the interface parameter testString and output the result to the response data by adding a method to compare test strings in the runTest() method.

**public** SampleResult runTest(JavaSamplerContext context)

{

log.info ("execute runTest..");

String a = context.getParameter("testStr");

String b = context.getParameter("testString");

result.setSampleLabel("java请求");

result.sampleStart();

**if** (a.equals(b))

resultData = "两字符串是一样的";

**else**

resultData = "两字符串是一样";

result.setResponseData("结果是： "+resultData, **null**);

result.setDataType(SampleResult.***TEXT***);

**if** (testStr.length()<0)

{

log.info("fail...");

result.SetSuccessful(**false**);

}

**else**

{

log.info("TestString is \t"+testStr);

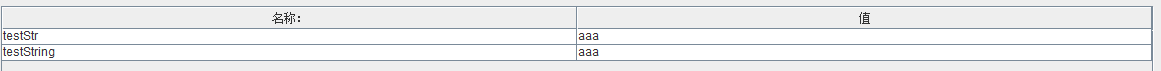
result.setSuccessful(**true**);

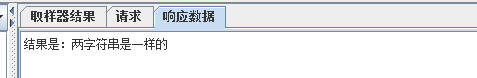
}

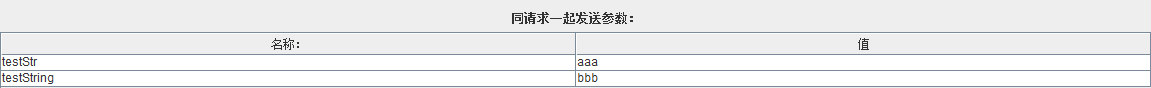
**return** result;

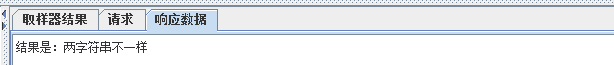
}

1. Generate the project JAR package and test it in Jmeter：









实验结果分析

通过Jmeter对java工程的性能测试，我发现jmeter是一款强大的性能测试软件，可以通过模拟多个用户同时对java工程进行处理，检测处理时间来分析java项目性能好坏，结果清晰明了。