1JMeter介绍

　　JMeter是Apache组织的开放源代码项目，它是功能和性能测试的工具，100%的用[**java**](javascript:;)实现。JMeter可以用于测试静态或者动态资源的性能（文件、Servlets、Perl脚本、java对象、数据库和查询、ftp服务器或者其他的资源）。JMeter用于模拟在服务器、网络或者其他对象上附加高负载以测试他们提供服务的受压能力，或者分析他们提供的服务在不同负载条件下的总性能情况。

　　2.启动JMeter

　　进入JMeter的bin目录，然后执行：

　　sudo ./jmeter.sh

　　3.原始的测试方法

在没有使用JMeter前，我对hetty的性能测试，都是通过自己写多线程代码去完成的，相当苦逼，相当麻烦，不过也能锻炼自己的编码能力，我先贴出比较原始的测试方法，如下：

|  |
| --- |
| public class RpcHessianClient {  public static void main(String[] args) {  String url = "http://localhost:8081/apis/hello";  HessianProxyFactory factory = new HessianProxyFactory();  ExecutorService es = Executors.newFixedThreadPool(10);  int size = 1000000;  final CountDownLatch cdl = new CountDownLatch(size);  try {  long start = System.currentTimeMillis();  factory.setUser("client1");  factory.setPassword("client1");  factory.setOverloadEnabled(true);  final Hello basic = (Hello) factory.create(Hello.class，  url);  for (int i = 0; i < size; i++) {  es.submit(new Runnable() {  @Override  public void run() {  String u=basic.hello("guolei");  //System.out.println(u);  cdl.countDown();  }  });  }  cdl.await();  long time = System.currentTimeMillis() - start;  System.out.println("SayHello:");  System.out.println("耗时：" + (double) time / 1000 + " s");  System.out.println("平均:" + ((double) time) / size +" ms");  System.out.println("TPS:" + (double) size / ((double) time / 1000));  // System.out.println("Hello， " + s.getMail());  } catch (MalformedURLException e) {  e.printStackTrace();  } catch (InterruptedException e) {  e.printStackTrace();  }finally{  es.shutdown();  }  } |