##### Environment preparation

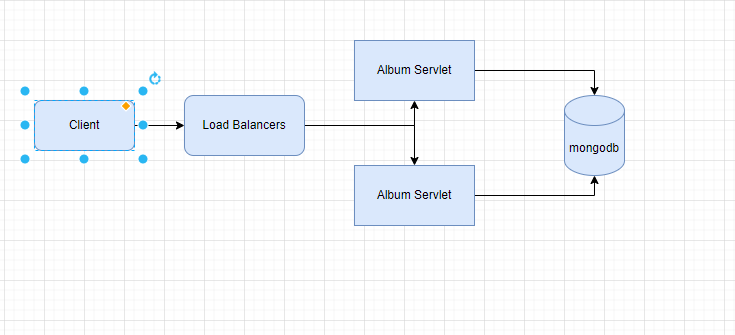
**jdk11**

**apache-tomcat-9.0.53**

**ec2 2c4g**

**mongo dB**

##### Project Framework



├─client

│ │ nmtb.png

│ │ pom.xml

│ │ ├─src

│ │ └─main

│ │ ├─java

│ │ │ └─client

│ │ │ │ AppMainTest.java

│ │ │ │ ImageMetaData.java

│ │ │ │ PlotUtil.java

│ │ │ │ ThreadWork.java

│ │ │ │

│ │ │ ├─part1

│ │ │ │ Part1Client.java

│ │ │ │

│ │ │ └─part2

│ │ │ Part2Client.java

│ │ │

│ │ └─resources

│ │ nmtb.png

│ │

└─javaDbAlbum

│ pom.xml

│

├─.idea

│ │ .gitignore

│ │ compiler.xml

│ │ encodings.xml

│ │ jarRepositories.xml

│ │ misc.xml

│ │ vcs.xml

│ │ workspace.xml

│ │

│ └─codeStyles

│ codeStyleConfig.xml

│ Project.xml

│

├─src

│ └─main

│ ├─Java

│ │ └─org

│ │ └─example

│ │ AlbumInfo.java

│ │ AlbumServlet.java

│ │ ImageMetaData.java

│ │ MongoTool.java

│ │

│ ├─resources

│ │ nmtb.png

│ │

│ └─webapp

│ │ index.jsp

│ │ nmtb.png

│ │

│ └─WEB-INF

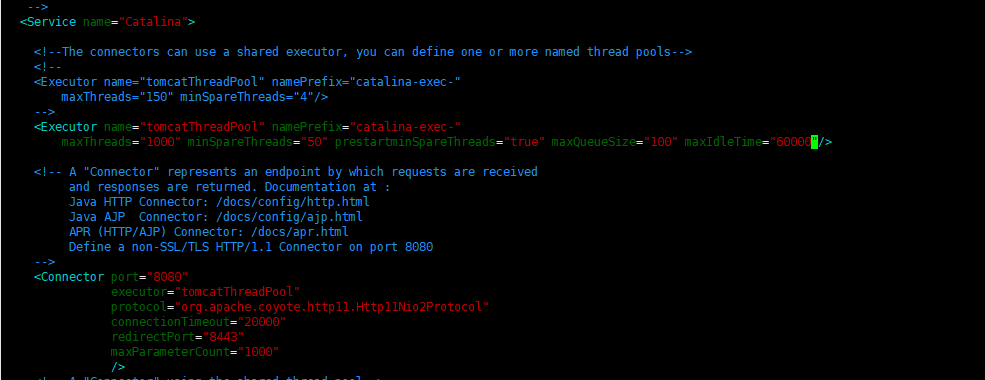
│ web.xml

1. The client creates a thread pool, and the multi-thread initiates HTTP requests to stress test the client, and waits for the thread pool to be executed through CountDownLatch to calculate the response time of the server interface
2. The javaDbAlbum java servlet provides doGet, which handles post/get requests sent by clients and reads and writes to databases

(client 创建线程池，多线程发起http请求压测客户端，通过CountDownLatch等待线程池执行完，计算服务器接口的响应时间

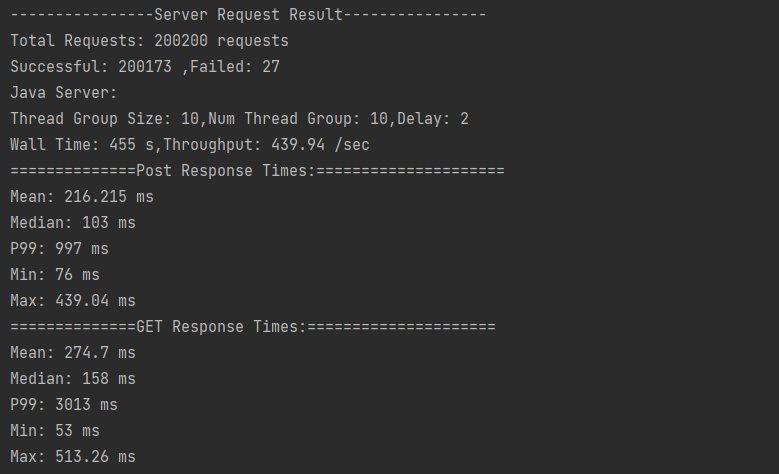
javaDbAlbum java servlet提供doGet，doPost分别处理客户端发送的post/get请求,读写数据库)

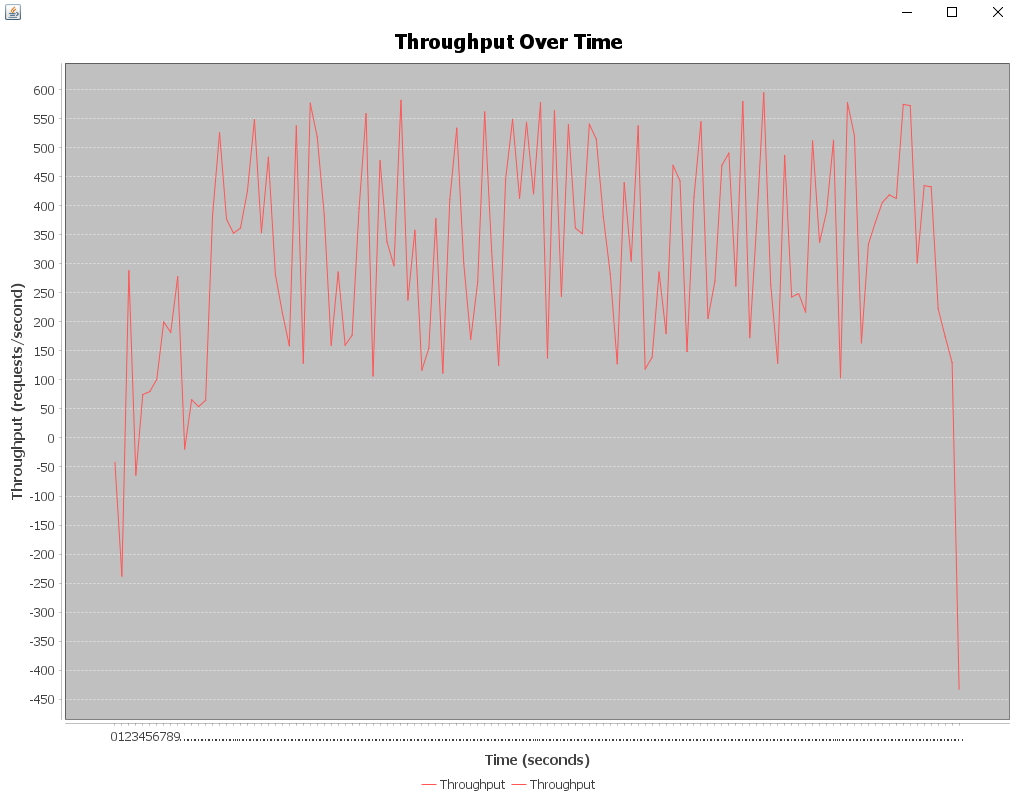
##### Tomcat Server setting



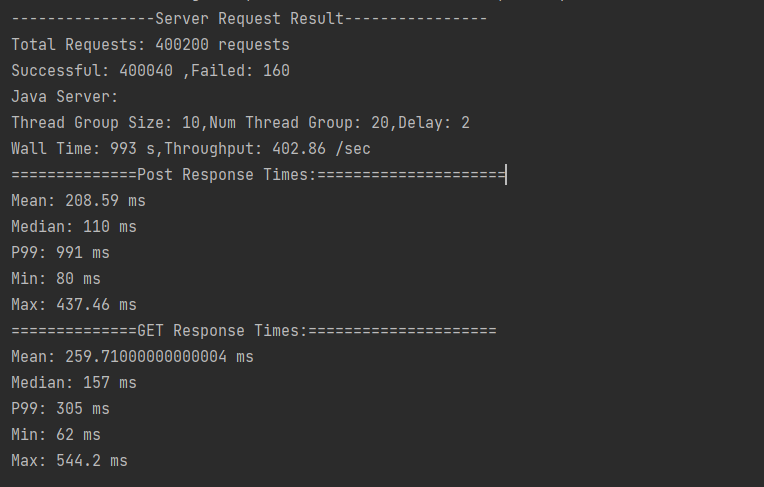
##### Single Server DB

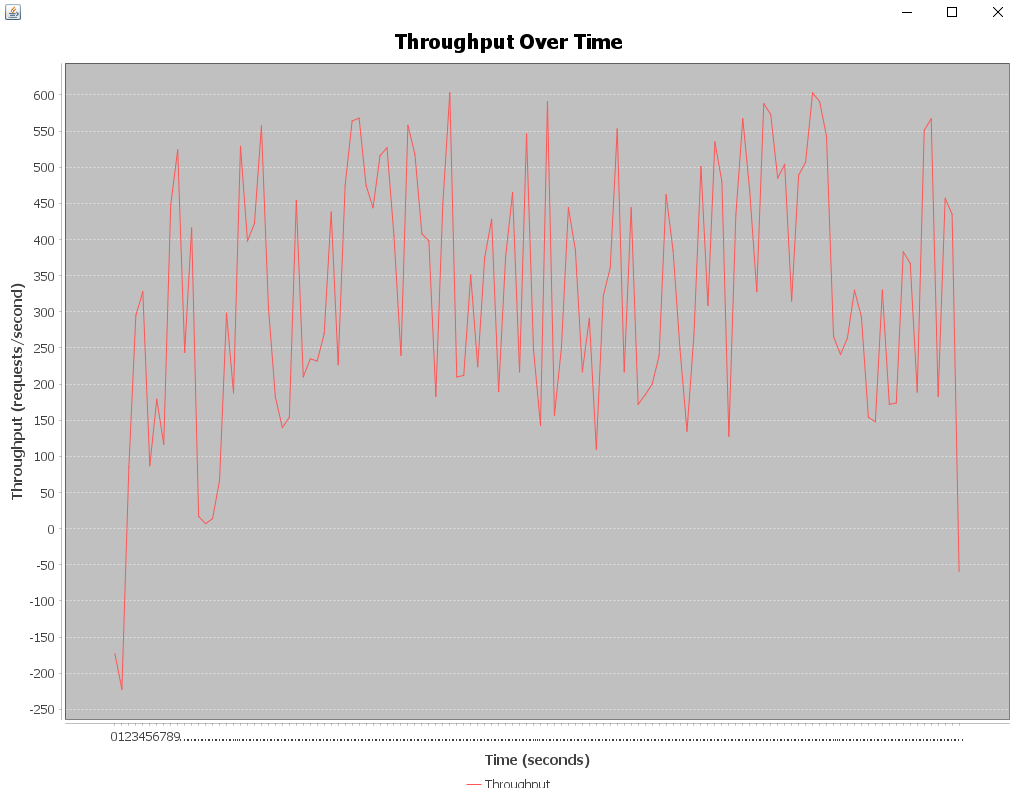
**threadGroupSize = 10, numThreadGroups = 10, delay = 2**

****

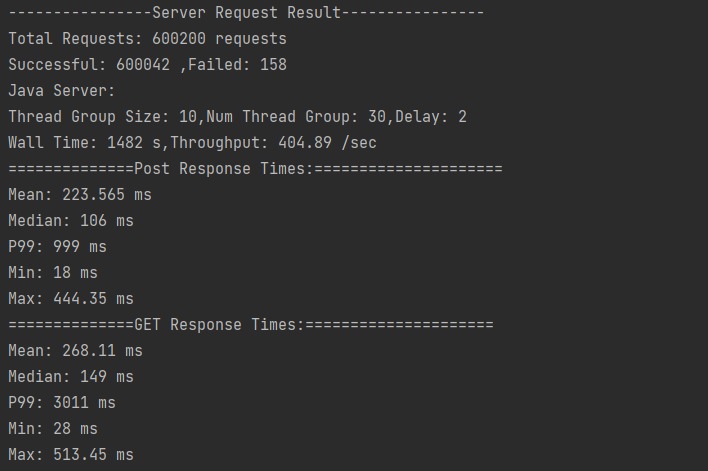
****

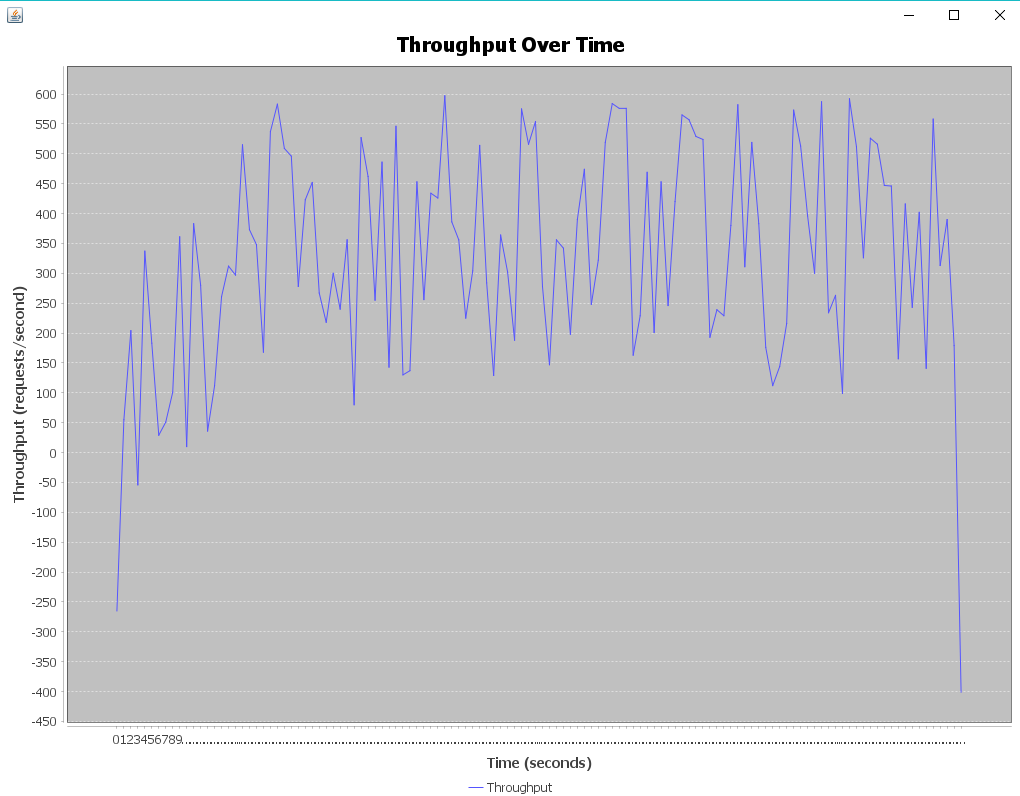
**threadGroupSize = 10, numThreadGroups = 20, delay = 2**

****

****

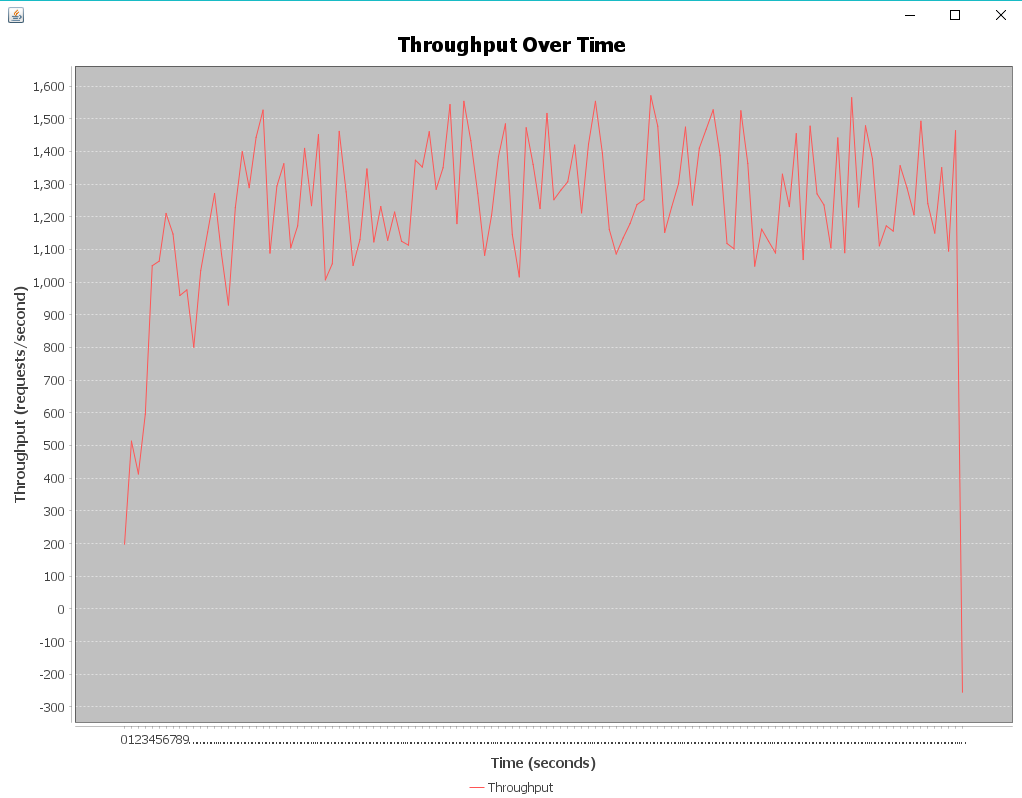
**threadGroupSize = 10, numThreadGroups = 30, delay = 2**

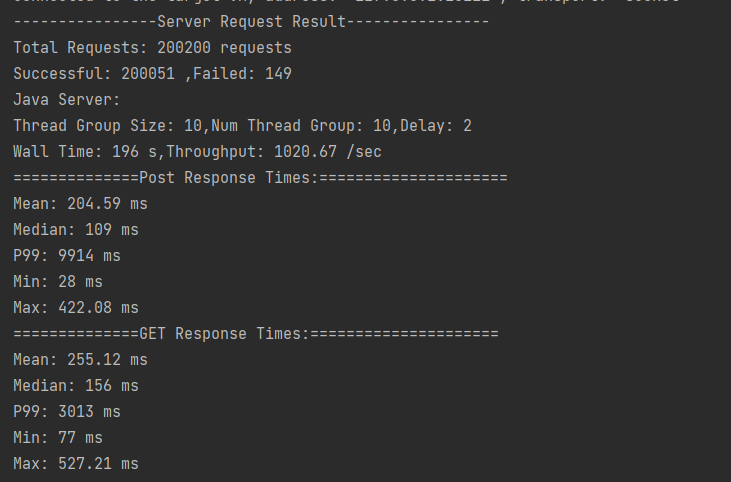
****



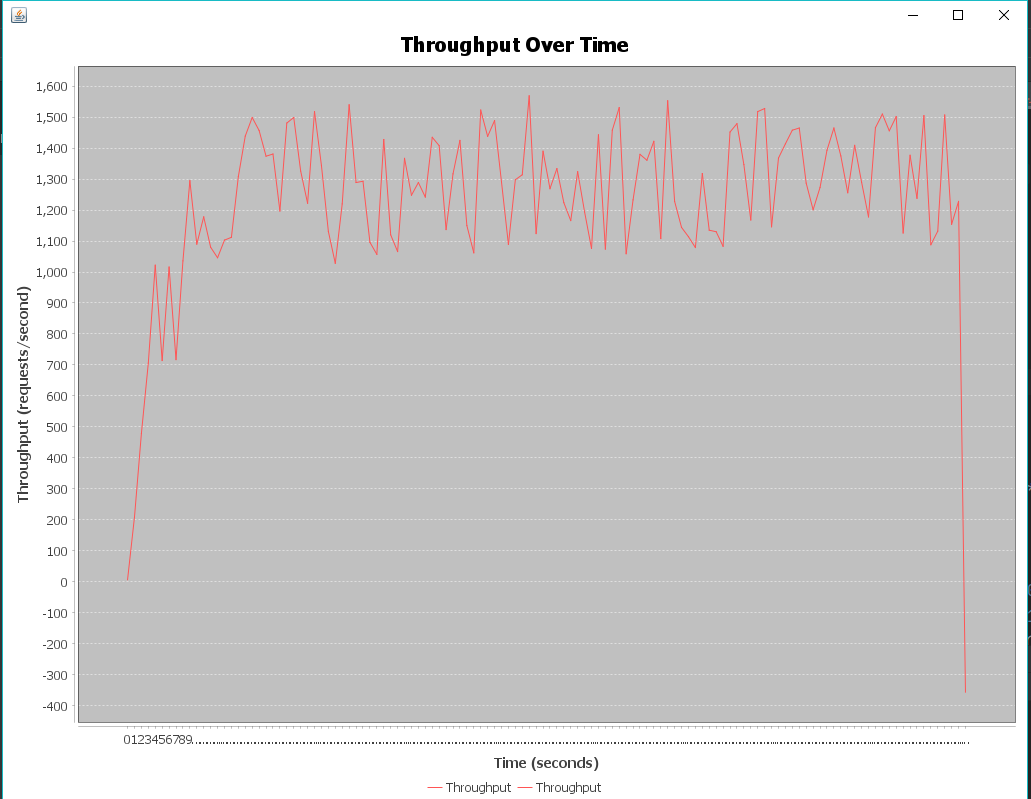
##### Load Balancers

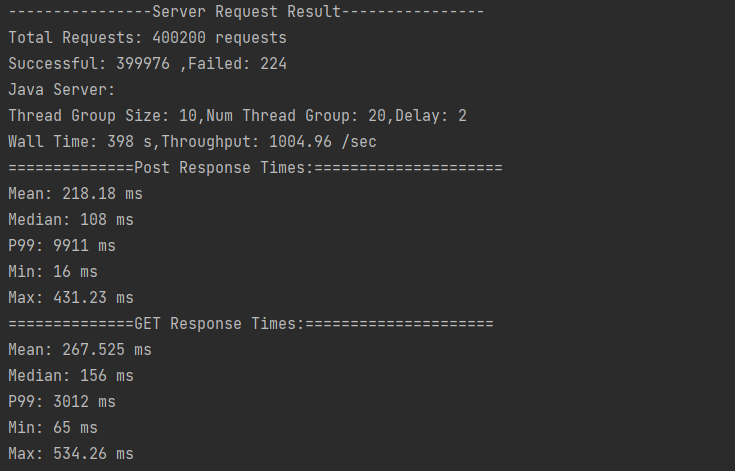
**threadGroupSize = 10, numThreadGroups = 10, delay = 2**



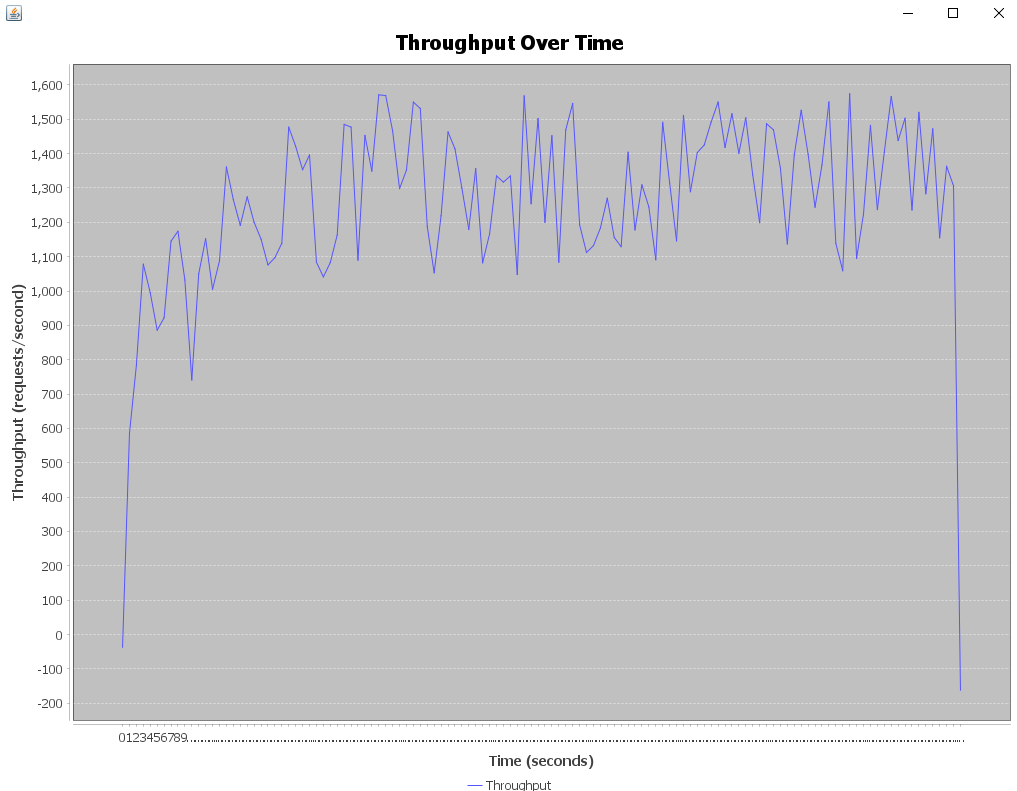


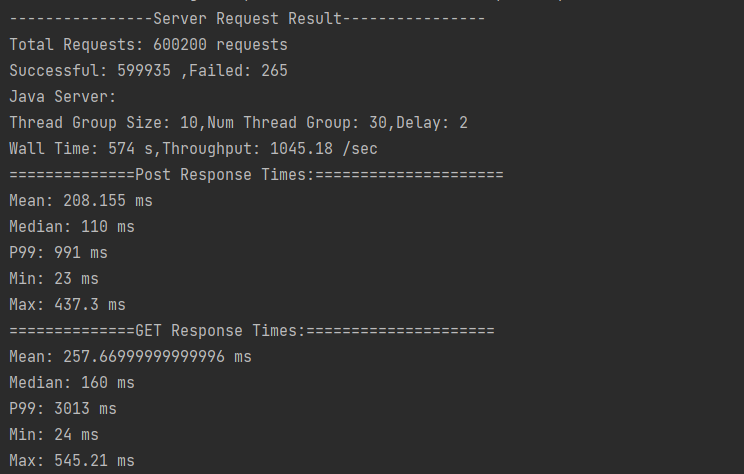
**threadGroupSize = 10, numThreadGroups = 20, delay = 2**





**threadGroupSize = 10, numThreadGroups = 30, delay = 2**

****

****