##### Environment

**jdk11**

**apache-tomcat-9.0.53**

**go1.21.1**

**ec2 2c4g**

##### framework

│ assignment.docx

│

├─client

│ │ pom.xml

│ └─src

│ └─main

│ ├─java

│ │ └─client

│ │ │ PlotUtil.java

│ │ │ ThreadWork.java

│ │ │

│ │ ├─part1

│ │ │ Part1Client.java

│ │ │

│ │ └─part2

│ │ Part2Client.java

│ │

│ └─resources

│ nmtb.png

│

├─goServer

│ go.mod

│ go.sum

│ Readme.md

│ server.go

│

└─javaServer

│ nmtb.png

│ pom.xml

│

└─src

└─main

├─Java

│ └─com

│ └─album

│ AlbumException.java

│ AlbumInfo.java

│ AlbumServlet.java

│ ImageMetaData.java

│

└─webapp

│ index.jsp

│

└─WEB-INF

web.xml

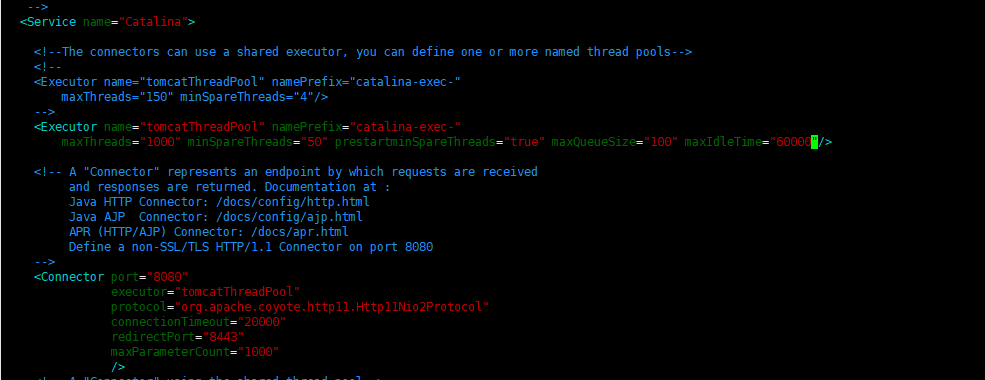
client 创建线程池，多线程发起http请求压测客户端，通过CountDownLatch等待线程池执行完，计算服务器接口的响应时间

goServer golang gin web框架提供post/get服务接口处理

javaServer java servlet提供doGet，doPost分别处理客户端发送的post/get请求

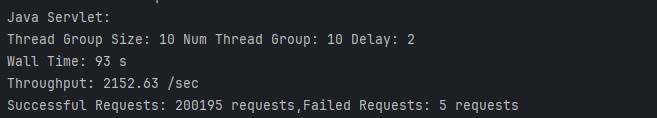
1. The client creates a thread pool, and the multi-thread initiates HTTP requests to stress test the client, and waits for the thread pool to be executed through CountDownLatch to calculate the response time of the server interface
2. The goServer golang gin web framework provides POST/GET service interface processing
3. The javaServer java servlet provides doGet, and doPost handles post/get requests sent by clients

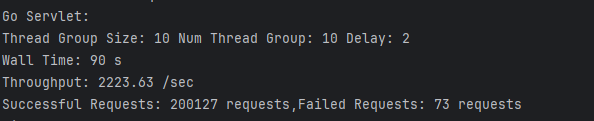
##### Tomcat Server配置



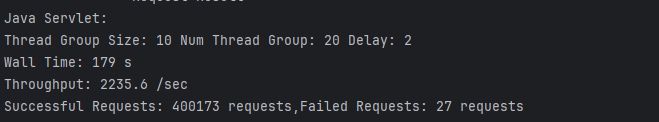
##### Part1

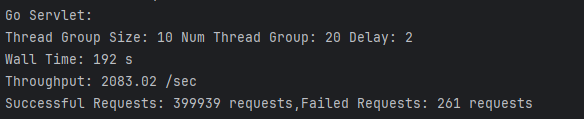
**threadGroupSize = 10, numThreadGroups = 10, delay = 2**



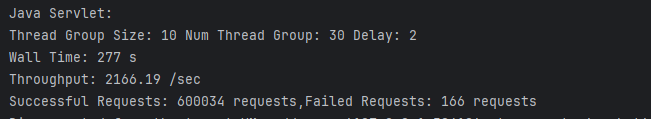


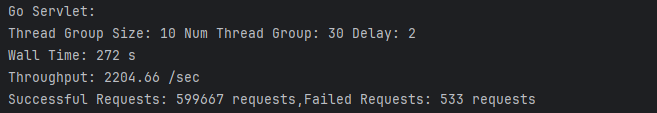
**threadGroupSize = 10, numThreadGroups = 20, delay = 2**





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

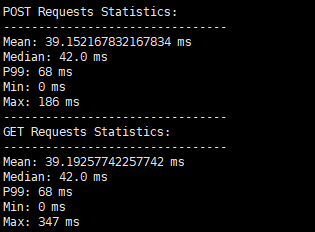


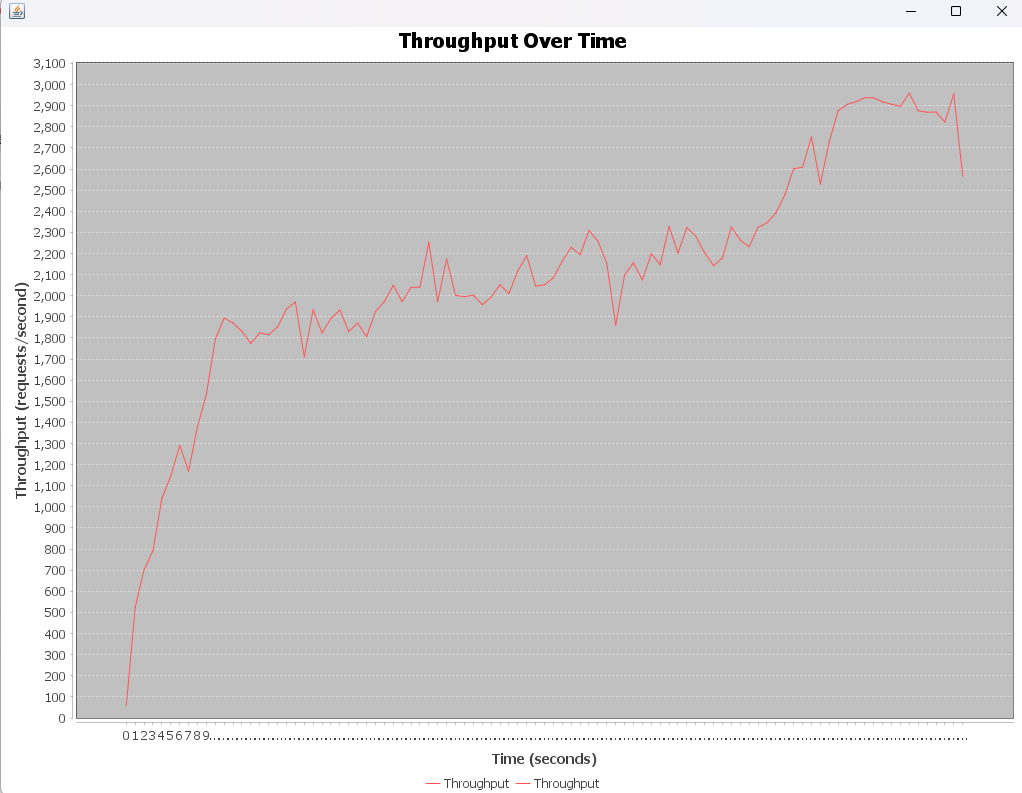


##### Part2

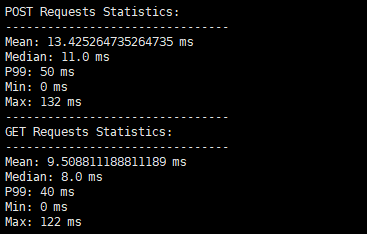
**threadGroupSize = 10, numThreadGroups = 10, delay = 2**

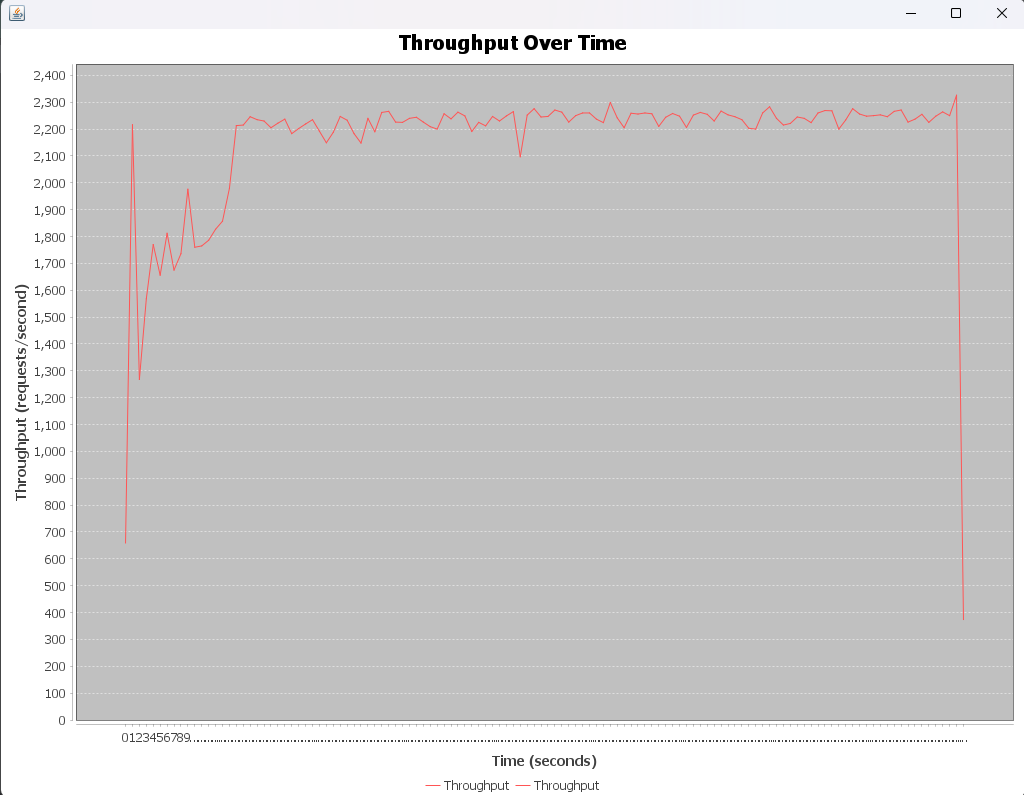
**Java Requests Statistics**





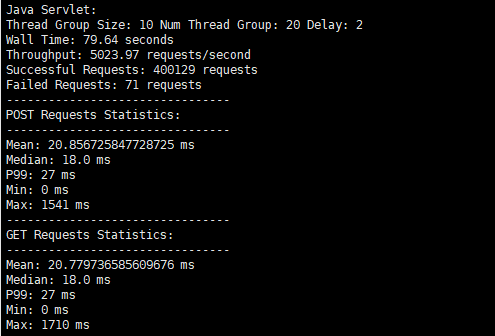
**Go Requests Statistics**

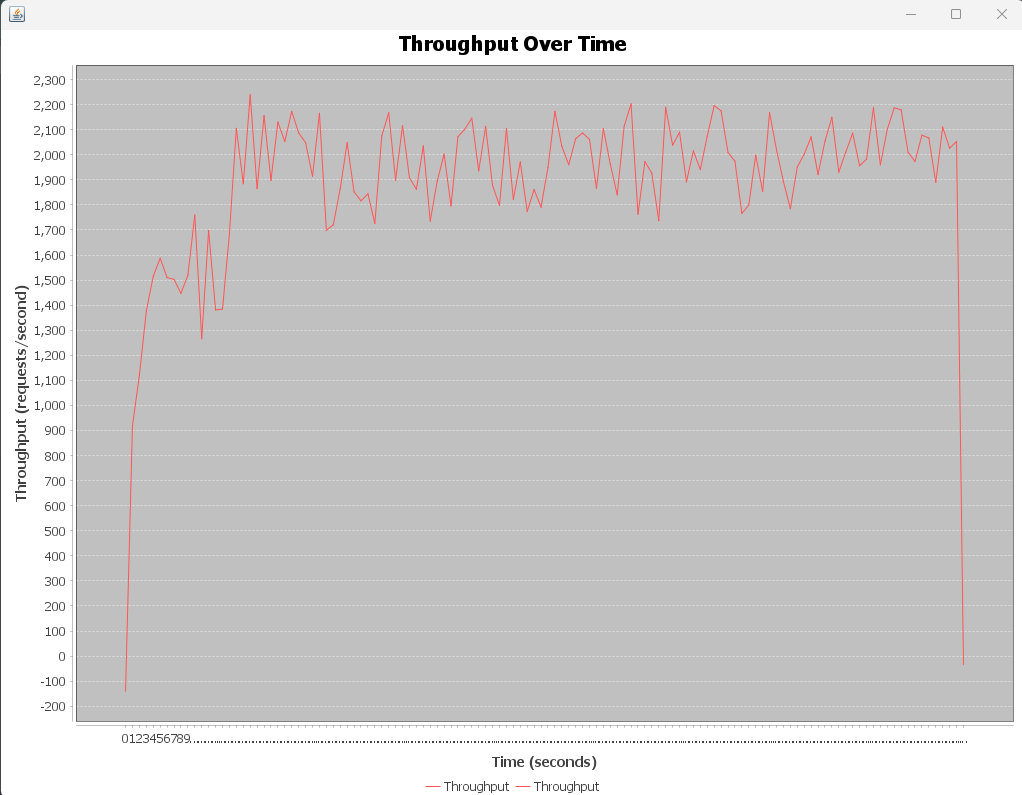




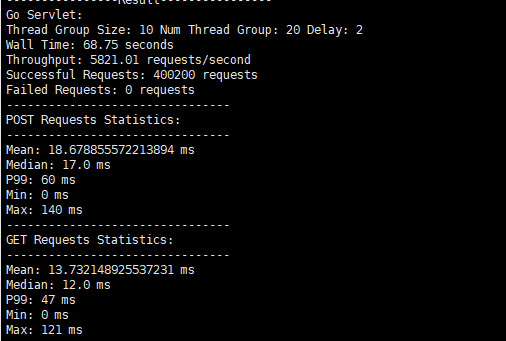
**threadGroupSize = 10, numThreadGroups = 20, delay = 2**

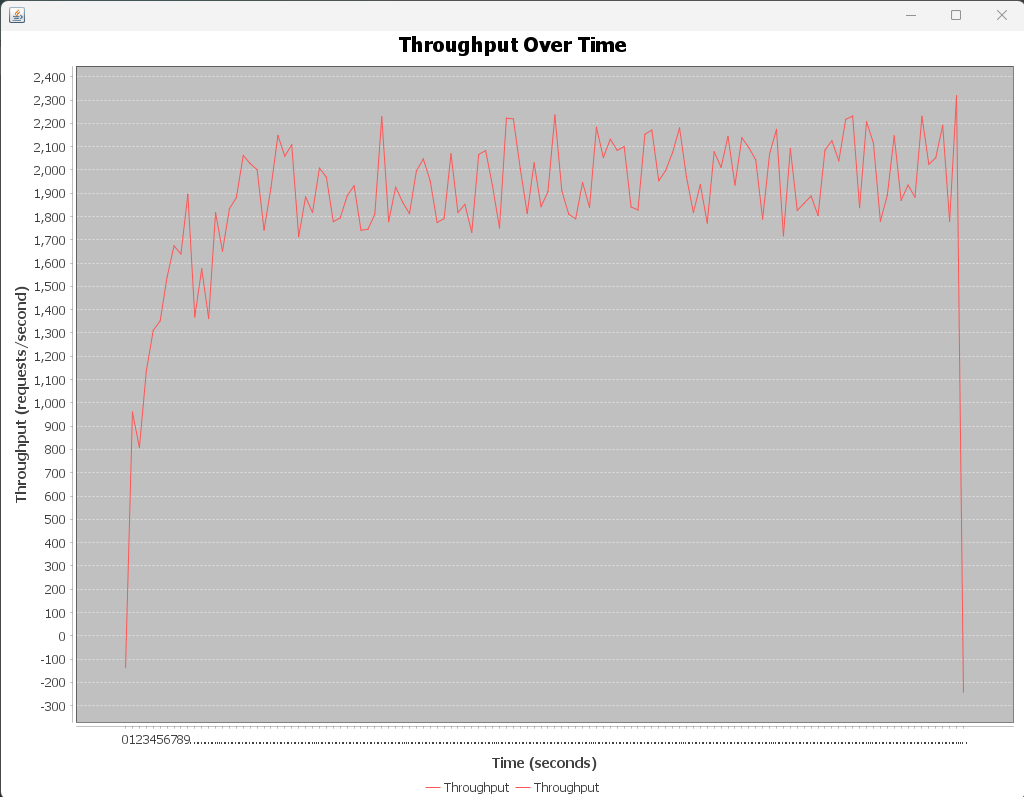
**Java Requests Statistics**





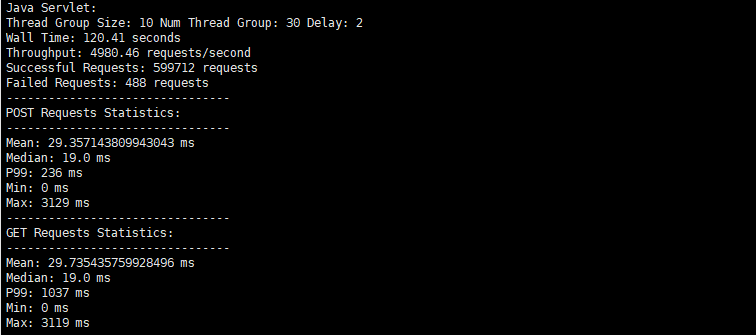
**Go Requests Statistics**

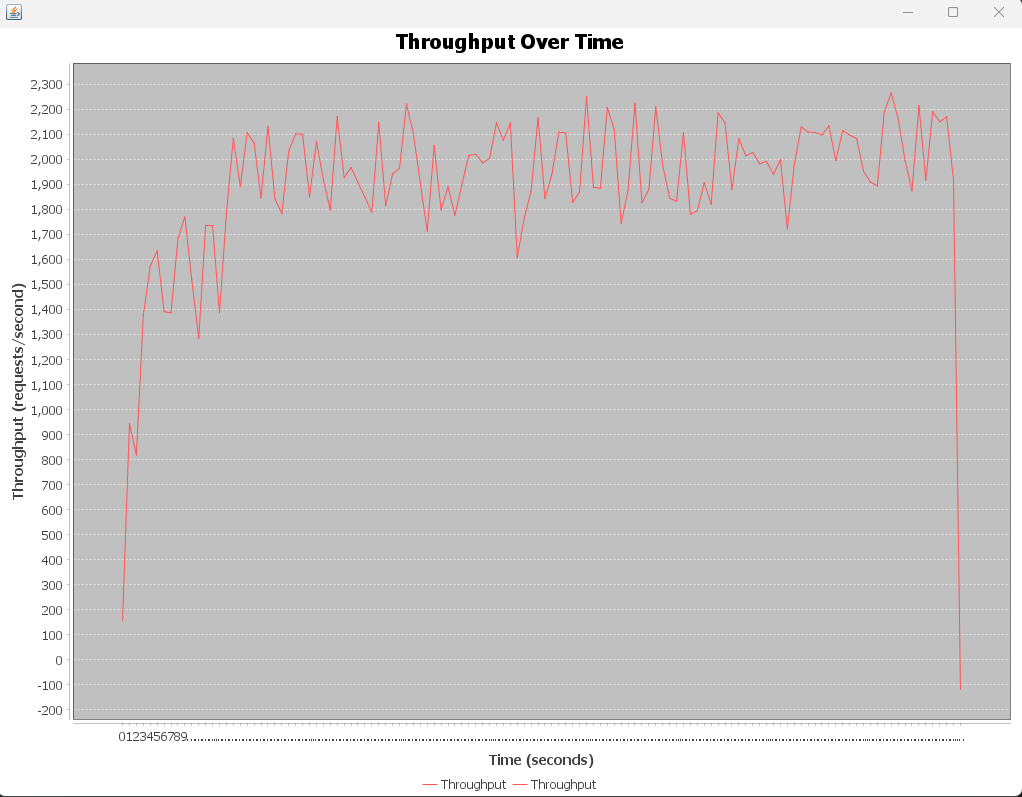




**threadGroupSize = 10, numThreadGroups = 30, delay = 2**

**Java Requests Statistics**





**Go Requests Statistics**

