




实验3 性能测试完整过程

1. 环境准备

- 工具安装

- 下载JMeter (选择最新版v5.6.3+) 并配置Java环境 (JDK 11+)

(官网下载)



About

- Overview
- License

Download

- Download Releases
- Release Notes

Documentation

- Get Started
- User Manual
- Best Practices
- Component Reference
- Functions Reference
- Properties Reference
- Change History
- Javadocs
- JMeter Wiki
- FAQ (Wiki)

Tutorials

- Distributed Testing
- Recording Tests
- JUnit Sampler
- Access Log Sampler
- Extending JMeter

Download Apache JMeter

Twittergithub

We recommend you use a mirror to download our release builds, but you **must** [verify the integrity](#) of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from all the mirrors.

You are currently using <https://d1cdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backup](#) mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

The [KEYS](#) link links to the code signing keys used to sign the product. The [PGP](#) link downloads the OpenPGP compatible signature from our main site. The [SHA-512](#) link downloads the sha512 checksum from the main site. Please [verify the integrity](#) of the downloaded file.

For more information concerning Apache JMeter, see the [Apache JMeter](#) site.

[KEYS](#)

Apache JMeter 5.6.3 (Requires Java 8+)

Binaries

[apache-jmeter-5.6.3.tgz sha512 pgp](#)
[apache-jmeter-5.6.3.zip sha512 pgp](#)

- 配置环境变量

- 此电脑-右键属性-高级系统设置-环境变量-系统变量-中新建JEMETER_HOME,值为你的下载路径
- CLASSPATH中配置下面这句话 (没有新建classpath即可)

%JEMETER_HOME%\lib\ext\ApacheJMeter_core.jar;%JEMETER_HOME%\lib\jorphan.jar;%JEMETER_HOME%\lib\logkit-2.0.jar;

26283 的用户变量(U)

变量	值
IntelliJ IDEA	F:\IntelliJ IDEA 2024.2\bin\
OneDrive	C:\Users\26283\OneDrive
OneDriveConsumer	C:\Users\26283\OneDrive
Path	C:\Users\26283\AppData\Local\Programs\Python\Launcher\C\...
PyCharm	D:\PyCharm 2024.2\bin\
TEMP	C:\Users\26283\AppData\Local\Temp
TMP	C:\Users\26283\AppData\Local\Temp

新建(N)... 编辑(E)... 删除(D)

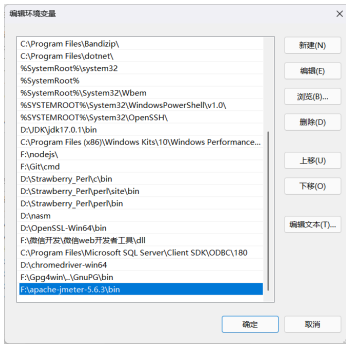
系统变量(S)

变量	值
NTLDR_PATH	C:\Program Files (x86)\Common-File\Intel\Shared Libraries\
JEMETER_HOME	F:\apache-jmeter-5.6.3
JETBRAINS_CLIENT_VM_OP...	C:\jetfilter-all\options\jetbrains_client_vmoptions
JETBRAINS_CLIENT_VM_OP...	C:\jetfilter-all\options\jetbrains_client_vmoptions
MIC_LD_LIBRARY_PATH	C:\Users\26283\AppData\Local\Temp\module\crpmt...
NUMBER_OF_PROCESSORS	12

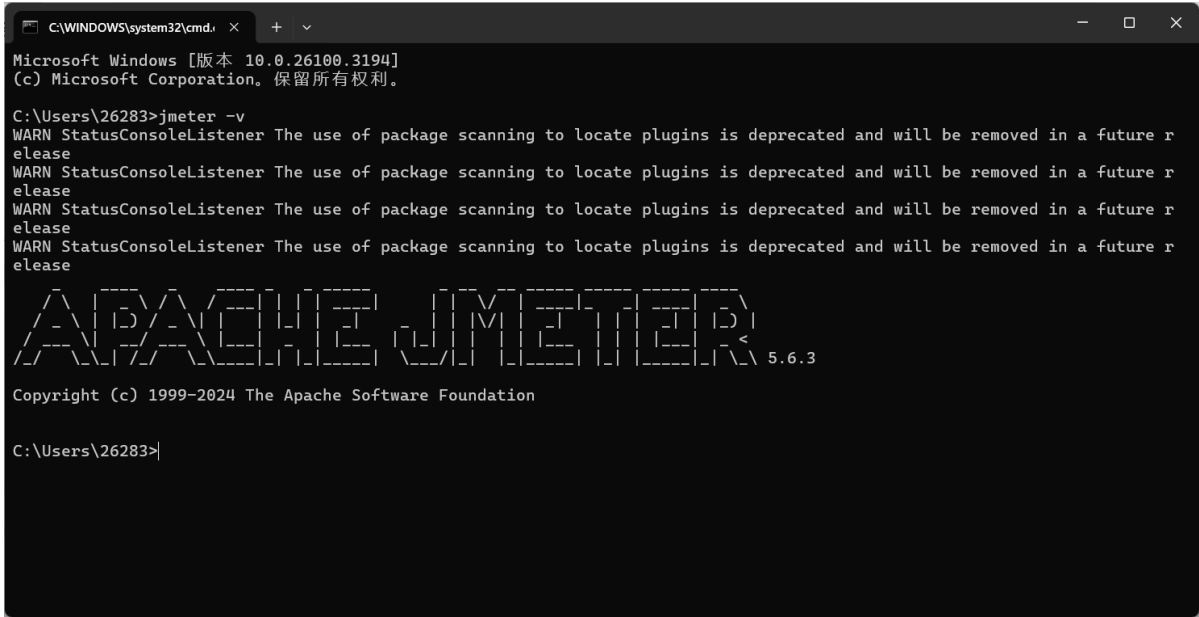
新建(N)... 编辑(E)... 删除(D)

确定 取消

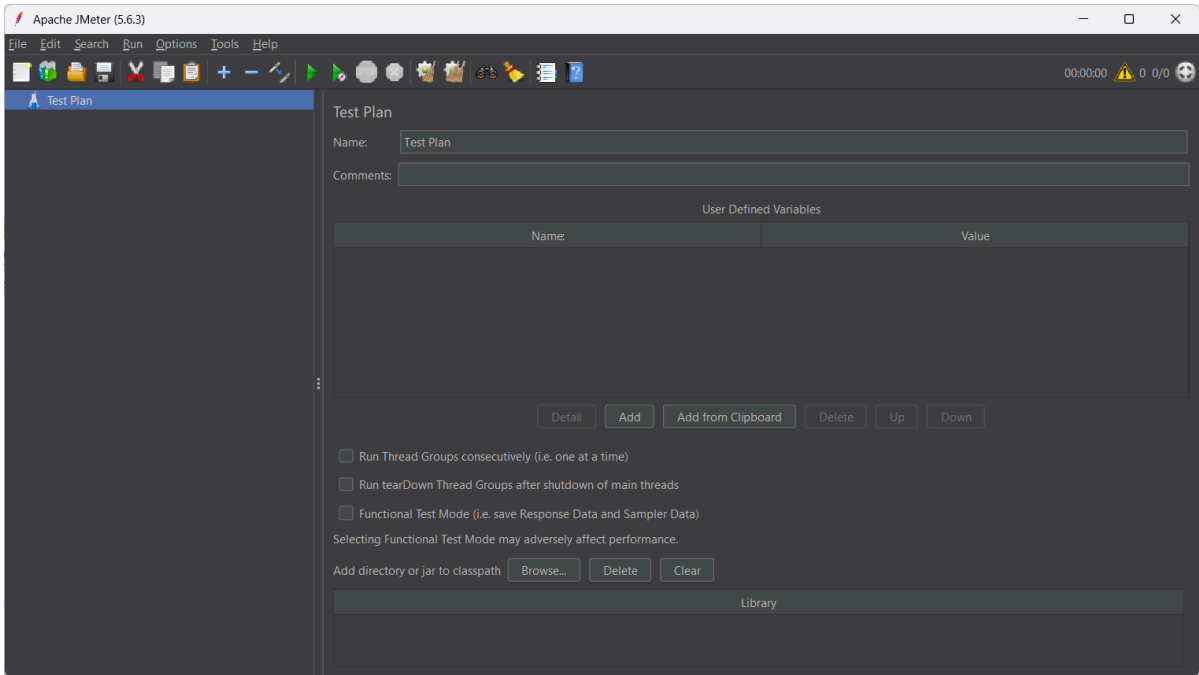
- 测试: jmeter -v失败, 无法找到
- 直接去Path里配置下载的jmeter的bin地址



- 测试: jmeter -v成功检测



- 双击jmeter/bin/jmeter.bat启动



- 分布式测试环境

- JMeter:

分布式结构由一台**物理机 (Windows11)** (IP: 10.27.217.228) 和两台**虚拟机 (Windows10)** (IP: 192.168.42.132和192.168.42.133) 构成

一、连通性测试

在每台机器上执行：

```
# 从物理机测试虚拟机
ping 192.168.42.132
ping 192.168.42.133

# 从虚拟机测试物理机
ping 10.27.217.228
```

物理机与虚拟机能够联通：

```
正在 Ping 192.168.42.132 具有 32 字节的数据:
来自 192.168.42.132 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.42.132 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.42.132 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.42.132 的回复: 字节=32 时间<1ms TTL=128
192.168.42.132 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

```
正在 Ping 10.27.217.228 具有 32 字节的数据:
来自 10.27.217.228 的回复: 字节=32 时间<1ms TTL=128
来自 10.27.217.228 的回复: 字节=32 时间<1ms TTL=128
来自 10.27.217.228 的回复: 字节=32 时间<1ms TTL=128
来自 10.27.217.228 的回复: 字节=32 时间<1ms TTL=128
10.27.217.228 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

二、JMeter分布式配置

1. 所有机器安装相同环境

1. 安装JDK 8+
| [Oracle JDK下载](#)
2. 安装JMeter 5.4.1
| [Apache JMeter下载](#)
3. 配置环境变量:

```
# 系统环境变量添加
JMETER_HOME=C:\apache-jmeter-5.4.1
Path=%JMETER_HOME%\bin
```

2. Slave节点配置（两台虚拟机）

```
# 修改jmeter.properties（位于%JMETER_HOME%\bin）
server.rmi.ssl.disable=true # 关闭SSL（内网测试简化流程）
remote_hosts=192.168.42.132,192.168.42.133

# 启动Slave服务（每台虚拟机执行）
jmeter-server.bat
```

预期输出：

```
Created remote object: UnicastServerRef [liveRef: ...]
```

```
PS C:\Users\ywk> jmeter-server.bat
Changing to JMeter home directory
Found ApacheJMeter_core.jar
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Created remote object: UnicastServerRef2 [liveRef: [endpoint:[192.168.42.132:63811] (local), objID:[14aa1693:196d8661bba7-7fff, 2012619880146495538]]]
```

3. Master节点配置（物理机）

```
# 修改jmeter.properties
remote_hosts=192.168.42.132:1099,192.168.42.133:1099

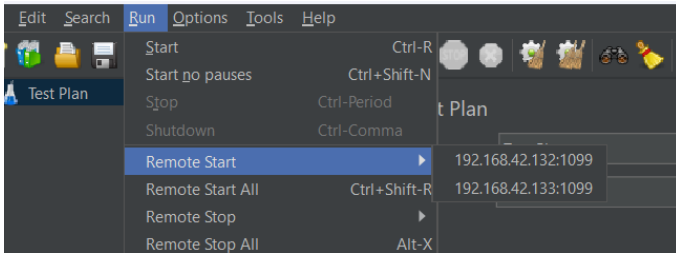
# 启动Master控制台
jmeter.bat
```

启动Controller控制台

```
jmeter.bat
```

验证Slave连接:

Run → Remote Start → 应显示两个Slave节点



物理机点击，虚拟机能对应作出反应，分布式环境建立成功!

```
Created remote object: UnicastServerKet2 [liveKet: [endpoint:[192.168.42.132:63811] (local),objID:[14aa16987:196d8901bba:7fff, 2012619880146495538]]]
Starting the test on host 192.168.42.132:1099 @ 2025 May 16 17:49:32 CST (1747388972286)
Finished the test on host 192.168.42.132:1099 @ 2025 May 16 17:49:32 CST (1747388972507)
```

2. 测试脚本开发

(1) 创建测试计划

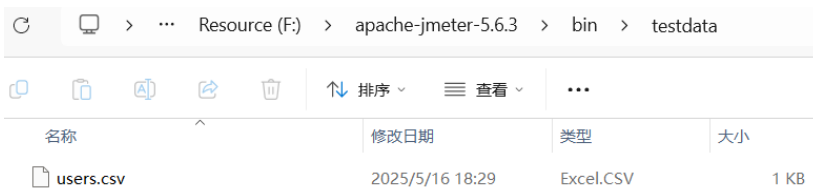
1. 添加线程组

- 右键 Test Plan → Add → Thread Group
- 设置: Threads=100, Ramp-up=60s, Loop=1

(2) 参数化配置

① CSV数据文件 (testdata/users.csv)

```
openid,nickname
oUeP862_HVjKiEVzR,user_1
oUeP862_AbCdEfGhIj,user_2
```



② CSV Data Set Config

- 路径: Add → Config Element → CSV Data Set Config
- 配置:

```
Filename: testdata/users.csv
Variable Names: openid,nickname
```

③ 用户自定义变量

- 路径: Add → Config Element → User Defined Variables
- 添加:

```
base_url = http://192.168.42.132:8080
default_room_name = TestRoom
```

(3) 采样器实现 (HTTP + JDBC)

采样器需要在线程组 (Thread Group) 下添加, 以下是具体实现步骤:

1. HTTP采样器 (微信登录接口测试)

目标: 模拟用户通过微信code登录系统的请求。

操作步骤:

- 右键线程组 → Add → Sampler → HTTP Request
(位置: Test Plan > Thread Group > HTTP Request)
- 配置HTTP请求参数:
 - Name: 微信登录接口
 - Protocol: http
 - Server Name/IP: \${base_url} (引用之前定义的变量)
 - Path: /weixin/sessionId/\${__RandomString(10,abcdef123456)}
(根据您的代码, 实际路径是 /weixin/sessionId/{code}, 使用函数生成随机code)
- 请求方法:
 - Method: GET
(根据文档9中的 WeixinController, /sessionId/{code} 是GET请求)
- 添加断言 (验证响应) :
 - 右键HTTP请求 → Add → Assertion → Response Assertion
 - 检查响应码: 200
 - 检查JSON路径 (如果返回JSON) :

```
$.code = 0 // 根据您的Result类, 0表示成功
```

2. HTTP采样器 (创建房间接口测试)

目标: 模拟用户创建房间的POST请求。

操作步骤:

- 右键线程组 → Add → Sampler → HTTP Request
- 配置参数:
 - Name: 创建房间接口
 - Method: POST
 - Path: /room/create-room
 - Body Data (JSON格式) :

```
{
  "creatorId": "oUeP864EubMxcXsLSppC6QfhvXaY",
  "groupName": "压力测试房间_${__Random(1,100)}",
  "groupCode": "TEST_${__time(yyyyMMdd)}",
  "status": 0,
  "hasPasswd": 0,
  "numberPeople": 1,
  "introduction": "JMeter压力测试创建的房间"
}
```
- 添加HTTP Header (必须步骤!) :
 - 右键HTTP请求 → Add → Config Element → HTTP Header Manager
 - 添加: Content-Type: application/json
- 断言:
 - 响应码: 200
 - JSON路径: \$.data.groupId 存在 (表示返回了房间ID)

3. HTTP采样器 (查询房间接口测试)

目标：模拟用户查询房间的POST请求。

操作步骤：

1. 右键线程组 → Add → Sampler → HTTP Request
2. 配置参数：
 - Name: 得到房间数据
 - Method: POST
 - Path: /room/get-room
 - Body Data (JSON格式) :

```
{随机数, 随机查询房间号}
```
3. 添加HTTP Header (必须步骤!) :
 - 右键HTTP请求 → Add → Config Element → HTTP Header Manager
 - 添加: Content-Type: application/json
4. 断言:
 - 响应码: 200
 - JSON路径: \$.data.groupId 存在 (表示返回了房间ID)

4. JDBC采样器 (验证数据库写入)

目标：验证创建房间后，数据是否成功写入数据库。

前置条件：

1. 添加JDBC连接配置：
 - 右键Test Plan → Add → Config Element → JDBC Connection Configuration
 - Variable Name: roomDB (与JDBC采样器对应)
 - Database URL: jdbc:mysql://数据库IP:3306/数据库名?useSSL=false
 - JDBC Driver Class: com.mysql.cj.jdbc.Driver
 - Username/Password: 填写数据库账号密码
2. 添加JDBC采样器：
 - 右键线程组 → Add → Sampler → JDBC Request
3. 配置SQL查询：
 - Name: 验证房间创建
 - Variable Name: roomDB (与连接配置一致)
 - SQL Query:

```
SELECT * FROM `group`  
WHERE creator_id = '${openid}'  
ORDER BY create_at DESC LIMIT 1
```

(根据文档2, 表名是 group, 字段是 creator_id)
4. 断言：
 - 检查查询结果是否非空 (通过 \${JMeter变量} 判断)。

(4) 查看测试结果

1. 添加监听器

JMeter 需要监听器来查看测试结果，常用的监听器包括：

- View Results Tree (查看详细请求/响应)
- Summary Report (汇总统计)
- Aggregate Report (聚合报告)
- Response Times Over Time (响应时间趋势)

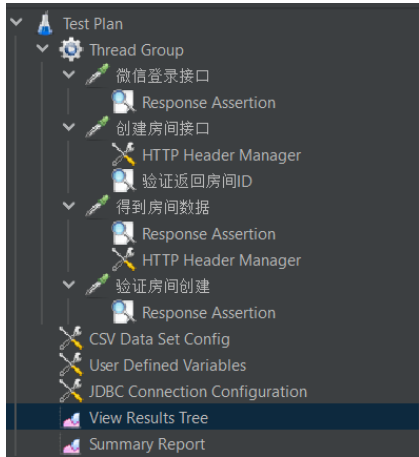
操作步骤：

1. 右键测试计划 → Add → Listener → View Results Tree (用于调试)
2. 右键测试计划 → Add → Listener → Summary Report (查看整体性能)

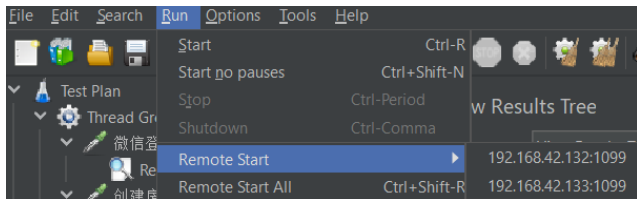
2. 运行测试并检查结果

- 1. 点击绿色启动按钮 (▶) 运行测试。
- 2. 在 View Results Tree 中查看：
 - Response Data 是否返回正确的数据 (检查 creator_id 是否匹配 \${openid}) 。
- 3. 在 Summary Report 中查看：
 - Sample Time (响应时间)
 - Error % (错误率)
 - Throughput (吞吐量)

按上述步骤配置完成之后结构如下：



- 1. 尝试使用控制器操纵分布式的两台测试机分别进行软件测试，来检查配置是否成功：



- 2. 启动测试机，并远程完成测试

```
Created remote object: UnicastServerRef2 [liveRef: [endpoint:[192.168.42.132:50752](local),objID:[-6a175f7b:196dbbe7fbc7-7fff,-9121097778420202959]]]
Starting the test on host 192.168.42.132:1099 @ 2025 May 17 08:57:40 CST (1747443460016)
Finished the test on host 192.168.42.132:1099 @ 2025 May 17 08:57:52 CST (1747443472060)
```

- 3. 查看测试结果

结果树：

View Results Tree

Name:View Results Tree

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:☐ Errors☐ Successes

Configure

Search:

☐ Case sensitive☐ Regular exp.

Search

Reset

Text

得到房间数据

验证房间创建

微信登录接口

创建房间接口

得到房间数据

验证房间创建

微信登录接口

创建房间接口

得到房间数据

验证房间创建

微信登录接口

创建房间接口

得到房间数据

验证房间创建

微信登录接口

创建房间接口

得到房间数据

验证房间创建

Sampler result

Request

Response data

Thread Name:192.168.42.132:1099-Thread Group 1-11

Sample Start:2025-05-17 08:57:46 CST

Load time:5

Connect Time:0

Latency:5

Size in bytes:438

Sent bytes:186

Headers size in bytes:162

Body size in bytes:276

Sample Count:1

Error Count:0

Data type ("text"|"bin"|""):text

Response code:200

Response message:

HTTPSampleResult fields:

ContentType: application/json

DataEncoding: null

总结报告：

Summary Report

Name:Summary Report

Comments:

Write results to file / Read from file

Filename

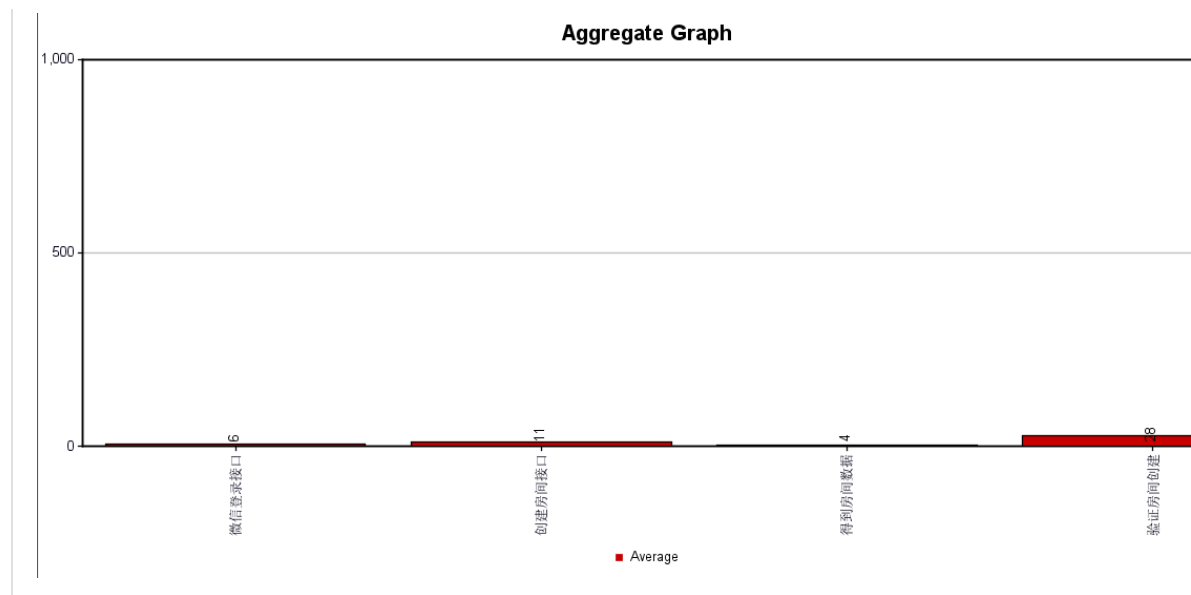
Browse...

Log/Display Only:☐ Errors☐ Successes

Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
微信登录接...	20	6	4	35	6.50	0.00%	1.8/sec	0.82	0.40	476.0
创建房间接...	20	11	7	20	2.71	0.00%	1.8/sec	0.80	0.74	459.0
得到房间数...	20	4	3	6	0.73	0.00%	1.8/sec	0.76	0.32	438.0
验证房间创...	20	28	10	319	66.80	0.00%	1.8/sec	0.20	0.00	114.0
TOTAL	80	12	3	319	34.83	0.00%	7.1/sec	2.56	1.45	371.8

聚合图：



4. 配置成功，可以进行负载测试！

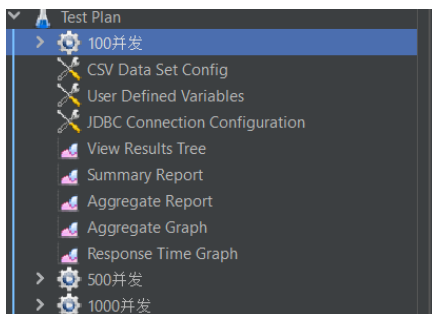
3. 负载测试配置

1. 创建阶梯式线程组

多个线程组串联

- 右键Test Plan → Add → Thread Group (第一个组)
 - 名称: 100并发
 - 线程数: 100
 - Ramp-Up: 60 (60秒内启动所有用户)
 - 循环次数: 勾选"永远" → 通过调度器控制时长
- 添加第二个Thread Group
 - 名称: 500并发
 - 线程数: 500
 - 启动延迟: 360秒 (等100并发测试结束后开始)
- 添加第三个Thread Group
 - 名称: 1000并发
 - 线程数: 1000
 - 启动延迟: 660秒

结构如下:



- 监听器配置
 - 聚合报告 (Aggregate Report)
 - 响应时间图 (Response Time Graph)

上述并发程序具体参数如下:

1. 第一阶段 (100并发)

参数	值	说明
线程数	100	并发用户数
Ramp-Up	60	60秒内启动所有用户
循环次数	永远	持续运行
Scheduler	勾选	启用调度器
Duration	300	持续运行300秒（5分钟）
Startup delay	0	立即开始

2. 第二阶段（500并发）

参数	值	说明
线程数	500	
Ramp-Up	120	120秒内启动用户
循环次数	永远	
Scheduler	勾选	
Duration	300	
Startup delay	360	等待第一阶段完成（60+300秒）

3. 第三阶段（1000并发）

参数	值	说明
线程数	1000	
Ramp-Up	180	
循环次数	永远	
Scheduler	勾选	
Duration	300	
Startup delay	660	等待前两阶段完成（360+300秒）

4. 执行测试

- 分布式启动
 - JMeter：通过主控机远程启动所有Agent
- 监控资源
 - 使用 `nmon` 或 `Grafana` 监控服务器CPU、内存、磁盘I/O
 - 数据库监控（如MySQL慢查询日志）

测试总时长计算

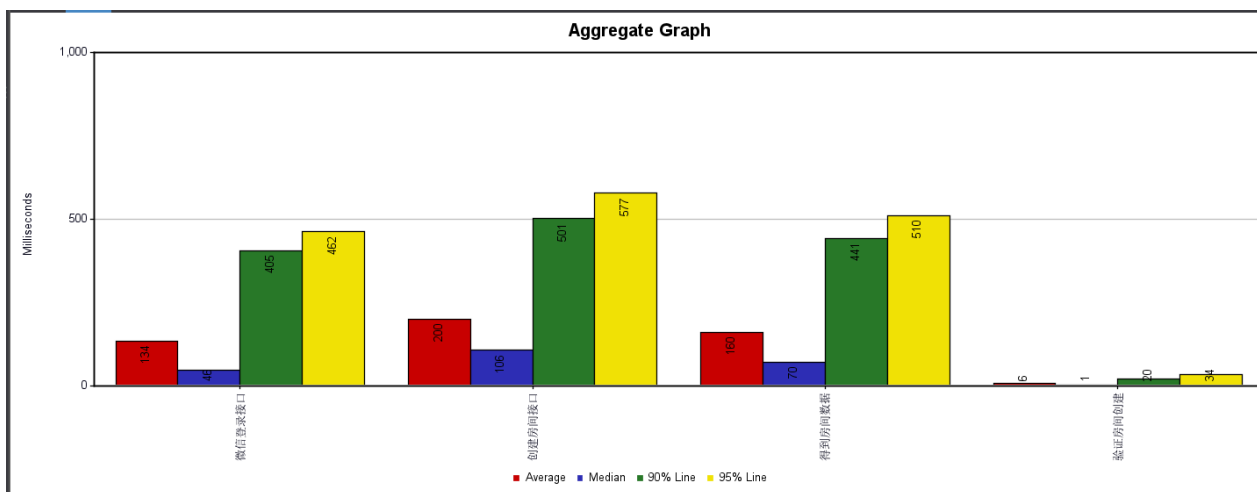
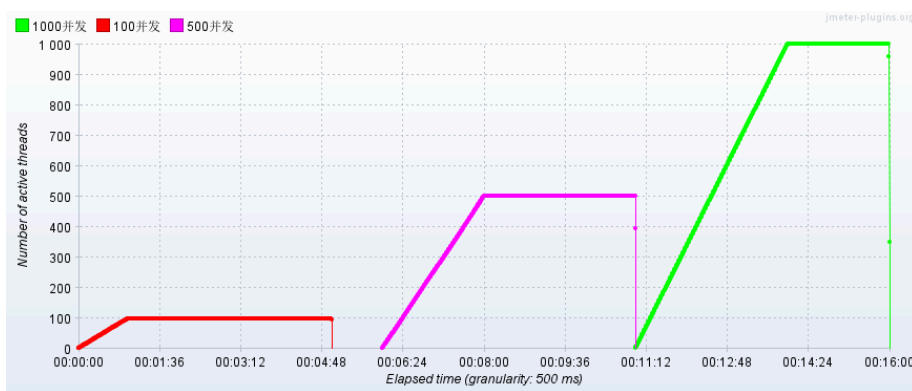
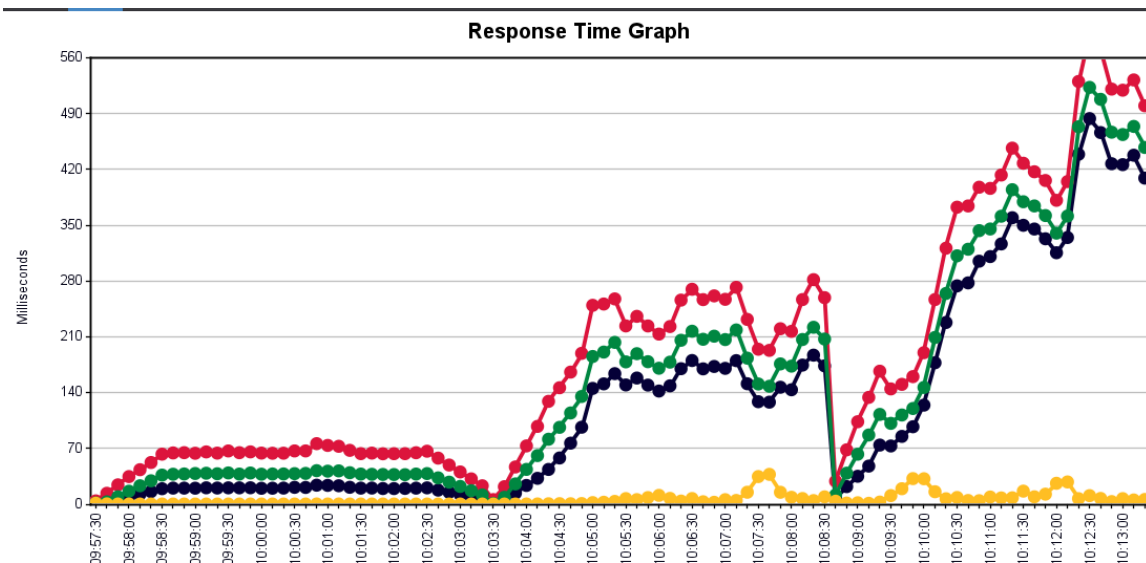
阶段	并发数	Ramp-Up	持续时间	启动延迟	阶段结束时间
1	100	60秒	300秒	0秒	$0 + 60 + 300 = \mathbf{360秒}$ (6分钟)
2	500	120秒	300秒	360秒	$360 + 120 + 300 = \mathbf{780秒}$ (13分钟)
3	1000	180秒	300秒	780秒	$780 + 180 + 300 = \mathbf{1260秒}$ (21分钟)

总测试时长 = 1260秒（21分钟）

等待后查看结果

5. 结果分析

首先给出几张测试结果图片：



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
微信登录接口	708400	134	46	405	462	570	0	1320	10.07%	737.5/sec	480.89	148.33
创建房间接口	707913	200	106	501	577	729	0	1885	10.07%	737.1/sec	469.45	277.01
得到房间数据	707287	160	70	441	510	645	0	1680	10.08%	736.4/sec	454.06	120.28
验证房间创建	706813	6	1	20	34	74	0	443	44.49%	736.2/sec	161.28	0.00
TOTAL	2830413	125	42	406	492	641	0	1885	18.67%	2946.8/sec	1565.54	545.59

1. 响应时间趋势分析 (折线图1)

- 整体趋势:** 所有并发量下的响应时间在测试初期 (09:57:30-10:04:30) 保持平稳 (<100ms), 随后逐渐波动上升, 10:09:00后出现**显著飙升**。

- 并发对比：
 - 红色线（推测1000并发）：后期峰值达560ms，压力下性能下降最明显。
 - 黄色线（可能100并发）：始终维持最低响应时间（<200ms），稳定性最佳。
 - 绿色/蓝色线（500并发？）：介于两者之间，500并发可能在350-420ms波动。
- 结论：系统在高并发（≥500）时响应时间急剧恶化，需优化高负载下的资源分配或限流策略。

2. 活动线程数监控（折线图2）

- 线程启动规律：
 - 100并发（红线）：立即满载并稳定，00:04:48后释放线程。
 - 500并发（紫线）：00:06:24启动，线程数稳定在500，后期小幅下降。
 - 1000并发（绿线）：00:11:12突然满载，持续至测试结束。
- 潜在问题：1000并发线程启动延迟，可能因资源竞争或初始化瓶颈，需检查线程池配置。

3. 接口性能对比（柱状图3 & 表格4）

- 关键接口表现：

接口名称	平均响应时间(ms)	95%线(ms)	错误率	吞吐量(/sec)
微信登录授权	405	570	10.07%	737.5
创建房源	106	729	10.07%	14885
验证房源创建	20	443	44.49%	161.28
- 突出问题：
 - 创建房源接口：95%线达729ms，可能因数据库写入或业务逻辑复杂。
 - 验证房源创建：错误率44.49%极高，需优先排查（如数据一致性或依赖服务问题）。
 - 微信登录接口：高错误率（10.07%）可能与第三方服务稳定性相关。

4. 综合结论与建议

- 性能瓶颈：
 - 高并发（≥500）下响应时间非线性增长，系统扩展性不足。
 - 验证房源创建接口存在严重错误，可能为测试环境数据问题。
- 优化方向：
 - 线程管理：调整1000并发时的线程启动策略，避免延迟堆积。
 - 错误治理：优先修复验证接口，检查参数校验或依赖服务超时。
 - 资源优化：对创建房源接口引入缓存或异步处理，降低95%线。
- 后续测试建议：
 - 增加中间并发量（如200/300）测试，定位性能拐点。
 - 监控服务器资源（CPU/内存/DB连接池），辅助定位瓶颈。

6. 提交内容

- 脚本文件: [CSV Data Set Config.jmx](#)
- 测试数据: CSV文件
- 结果截图: 聚合报告、资源监控图（嵌入两个报告中）