手机实物抽奖算法设计方案

修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **文件编号** | **版本号** | **拟制人/修改人** | **拟制/修改日期** | **更改理由** | **主要更改内容**  **（写要点即可）** |
|  | 0.1 | 石金普 | 2020/11/24 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[手机实物抽奖算法设计方案 1](#_Toc57126957)

[1. 场景与需求 3](#_Toc57126958)

[2. 赛题业务说明 3](#_Toc57126959)

[3. 方案设计 4](#_Toc57126960)

[3.1 概述 4](#_Toc57126961)

[3.2公式 4](#_Toc57126962)

[3.3 架构设计 5](#_Toc57126963)

[3.4详细流程 6](#_Toc57126964)

[3.5后期扩展 6](#_Toc57126965)

[4. 接口: 7](#_Toc57126966)

[4.1首轮初始化接口: 7](#_Toc57126967)

[4.2当前轮数参数查询接口: 8](#_Toc57126968)

[4.3抽奖接口 8](#_Toc57126969)

[4.4下一轮接口 9](#_Toc57126970)

[5. 接口性能压测 10](#_Toc57126971)

[5.1接口压测(单机) 10](#_Toc57126972)

[6. 业务指标测试 13](#_Toc57126973)

[7. 表设计 15](#_Toc57126974)

[8. 测试环境 16](#_Toc57126975)

# 场景与需求

场景: 运营商为了提高用户的粘性，一般会有会员积分，会员权益；会员积分可以用来购买商品，抽奖等，通过积分消费可以增加趣味性和提高运营商的品牌。 手机实物抽奖，就是积分消费的一个使用场景

需求: 实现抽奖概率固定,按照抽奖参与的人数和概率进行抽奖的算法

# 赛题业务说明

对于每种**手机类型的奖品**，有 （当天获奖基数）， （当天允许的最大获奖人数）, （中奖率）三个参数，其中 （当天获奖基数）每天是固定的，（当天允许的最大获奖人数） 在活动第一天与（当天获奖基数）相等，后续动态调整；(中奖率=当天允许的最大获奖人数/第一天定义的抽奖次数)，具体规则如下：

每个抽奖活动会预先设置首次（**预期抽奖总次数**），接下来每天的**预期抽奖总次数**是前一天的**实际的抽奖次数**；

每日**获奖基数**不变，当天未抽完的奖品滚动到第二天（即：第二天的**允许最大获奖人数**）；

设置每日获奖基数（例如100）。活动开始第一天最大获奖人数=每日获奖基数（也就是100），当天实际中奖人数（例50），那么需要结转到第二天的中奖数量（100-50=50），这样第二天的最大获奖人数就是 150（100+50=150）

**中奖概率（G） =当天允许最大获奖人数** 除以 **当天预期抽奖总次数；**

**实际中奖人数(E)** ，要充分考虑随机性进行抽取，从概率学计算获奖人数为：实际抽奖次数\*中奖概率，同时最大值不能超过 **当天允许最大获奖人数**。 （**算法为本系统的核心，要保证有人中奖，而且要符合概率统计，无限趋近实际抽奖次数\*中奖概率**）

如下表:



# 方案设计

## 3.1 概述

本方案采用分布式架构进行提供服务,方便业务峰值时进行横向扩展与无状态更新升级。分布式业务事务采用redis锁进行实现, 节点缓存数据更新通过redis订阅发布模型进行信号通知, 每一轮结果保存至mysql数据库方便进行数据分析展示。

## 3.2公式

**术语说明:**

A 预期抽奖总次数

B 实际抽奖次数

C 获奖基数

D 当天允许最大获奖人数

E 实际中奖人数

F 当天剩余奖品

G 中奖概率

**业务关系公式:**

D = C + F

E = B\*G OR D

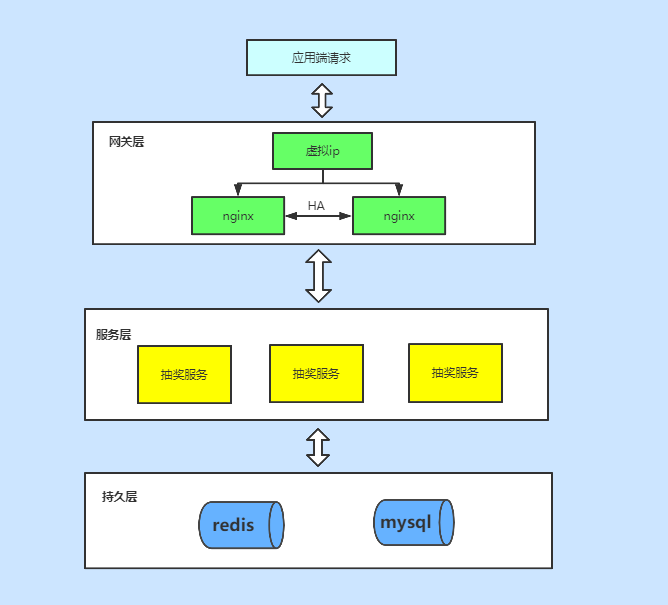
F = D – E

G = D / A

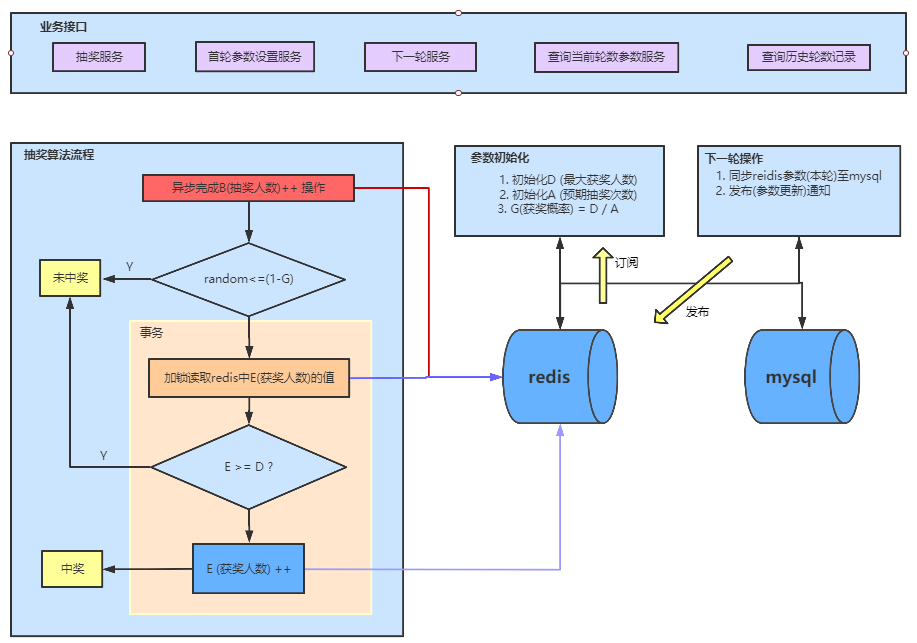
**限定条件:**

E <= D

## 3.3 架构设计



## 3.4详细流程



## 3.5后期扩展

限流, 熔断, 服务质量监控

# 接口:

## 4.1首轮初始化接口:

请求类型: GET

接口: /api/param/initFirstDayParam

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 例子 |
| baseWinNum | 获奖基数 C | 100 |
| maxWinNum | 当天允许最大获奖人数 D | 100 |
| predictNum | 预期抽奖总次数 A | 200000 |

例子:

127.0.0.1:9099/api/param/initFirstDayParam?baseWinNum=100&maxWinNum=100&predictNum=200000

响应:

{

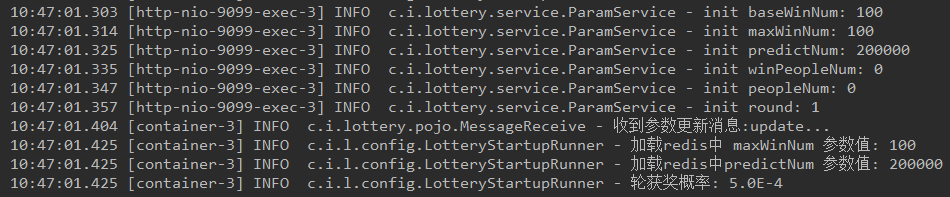
    "code": 200,

    "data": **null**,

    "message": **null**

}

后台日志:



## 4.2当前轮数参数查询接口:

请求类型: GET

接口: /api/param/queryParam

参数: 无

例子:

127.0.0.1:9099/api/param/queryParam

响应:

{

    "code": 200,

    "data": {

        "winPeopleNum": 0,

        "peopleNum": 0,

        "baseWinNum": 100,

        "maxWinNum": 100,

        "predictNum": 200000,

        "round": 1

    },

    "message": **null**

}

## 4.3抽奖接口

请求类型: GET

接口: /api/lottery/play

参数: 无

{

    "code": 200,

    "data": 0,

    "message": "未中奖"

}

或者

{

    "code": 200,

    "data": 1,

    "message": "中奖"

}

## 4.4下一轮接口

请求类型: GET

接口: /api/param/nextLottery

参数: 无

例子:

127.0.0.1:9099/api/param/nextLottery

响应:

{

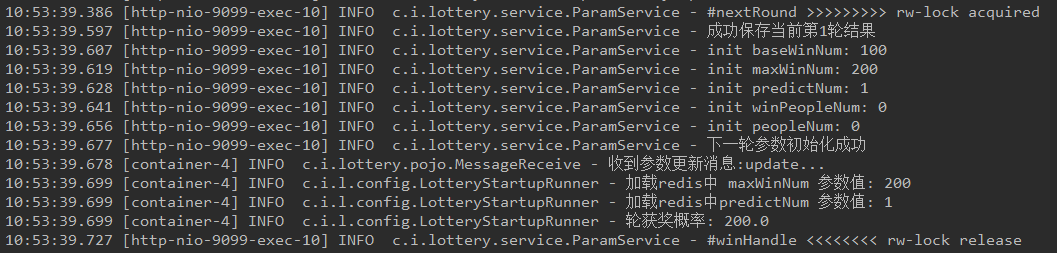
    "code": 200,

    "data": **null**,

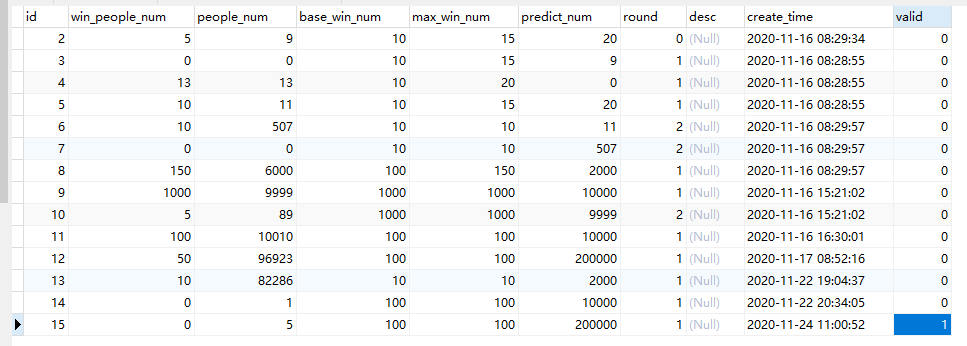
    "message": **null**

}

后台日志:



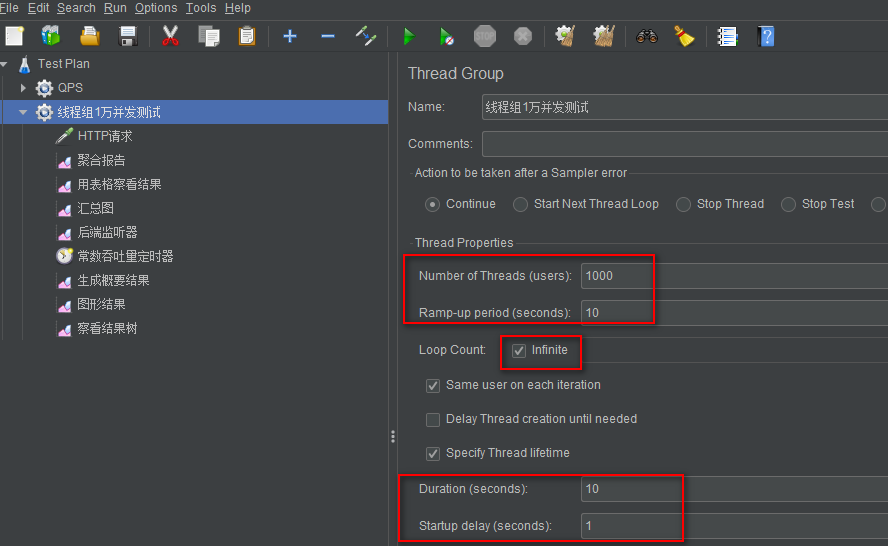
mobile\_lottery表结果:

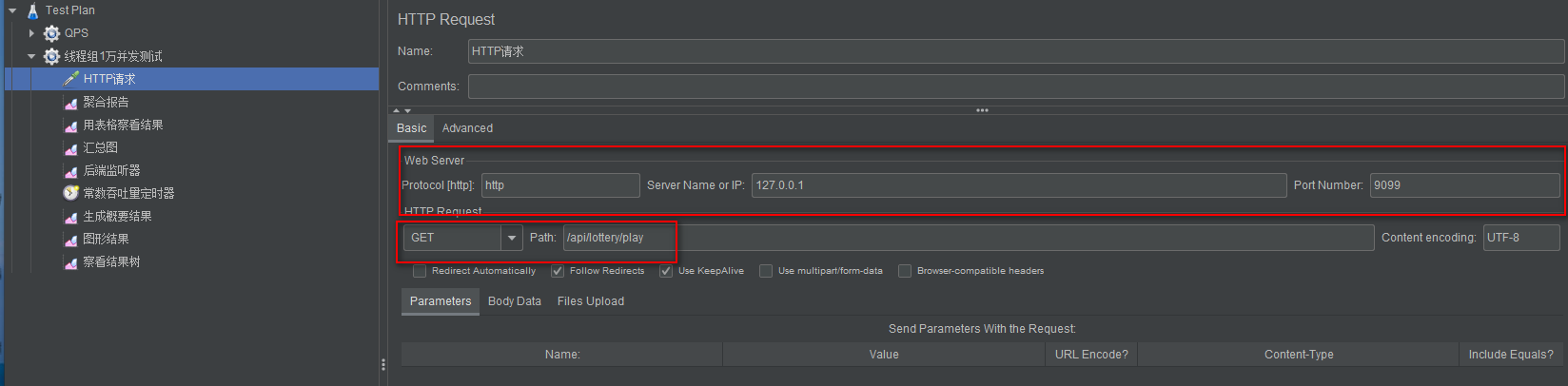


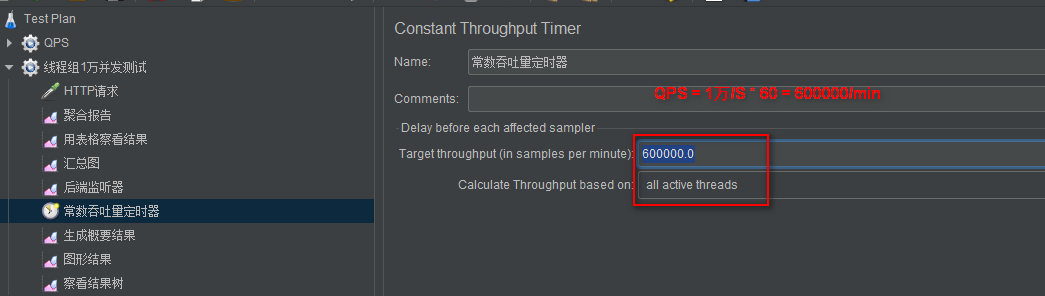
# 接口性能压测

## 5.1接口压测(单机)

**jmeter 参数设置**

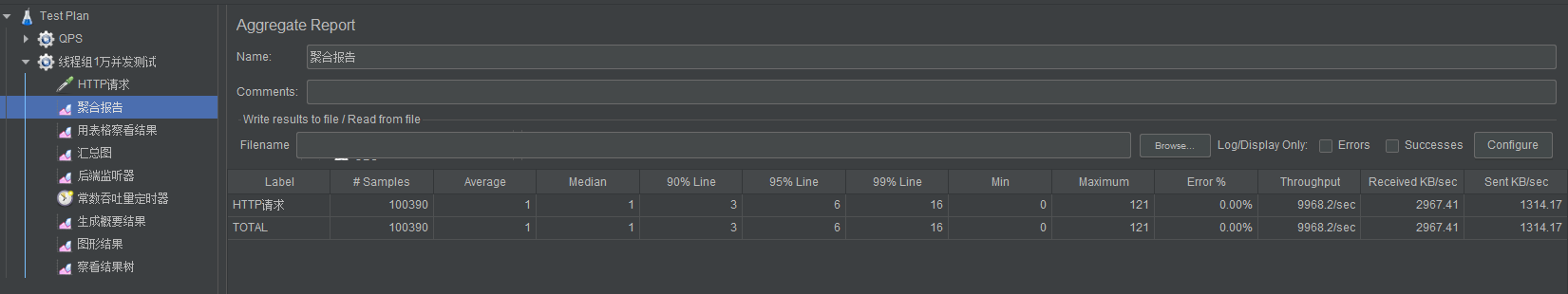


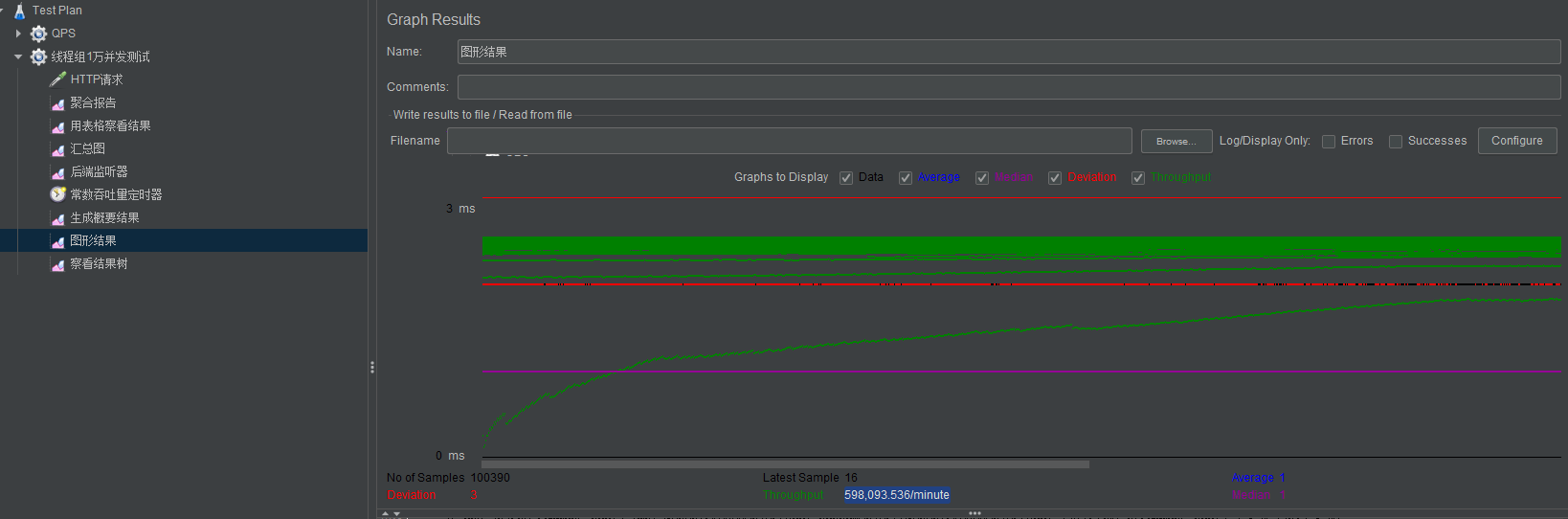




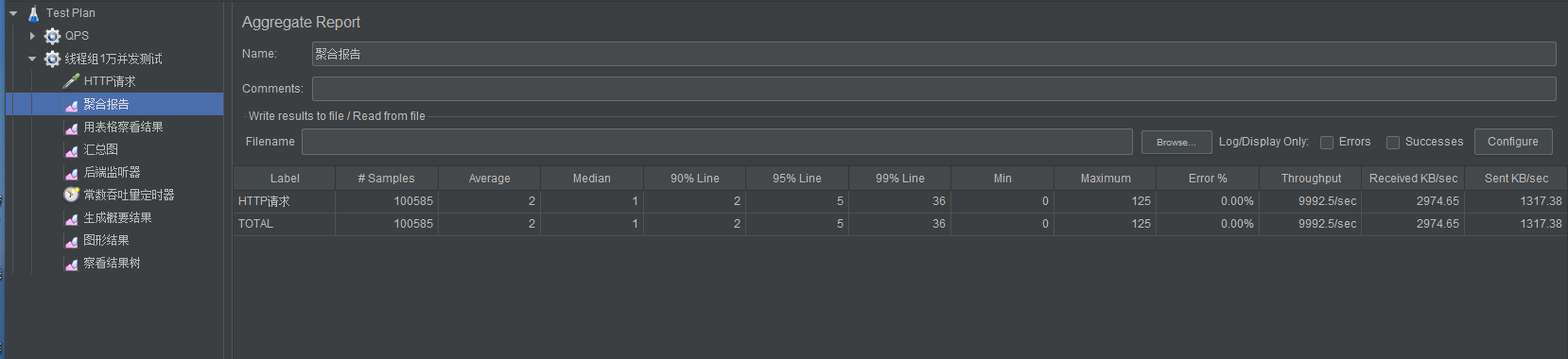
结果:

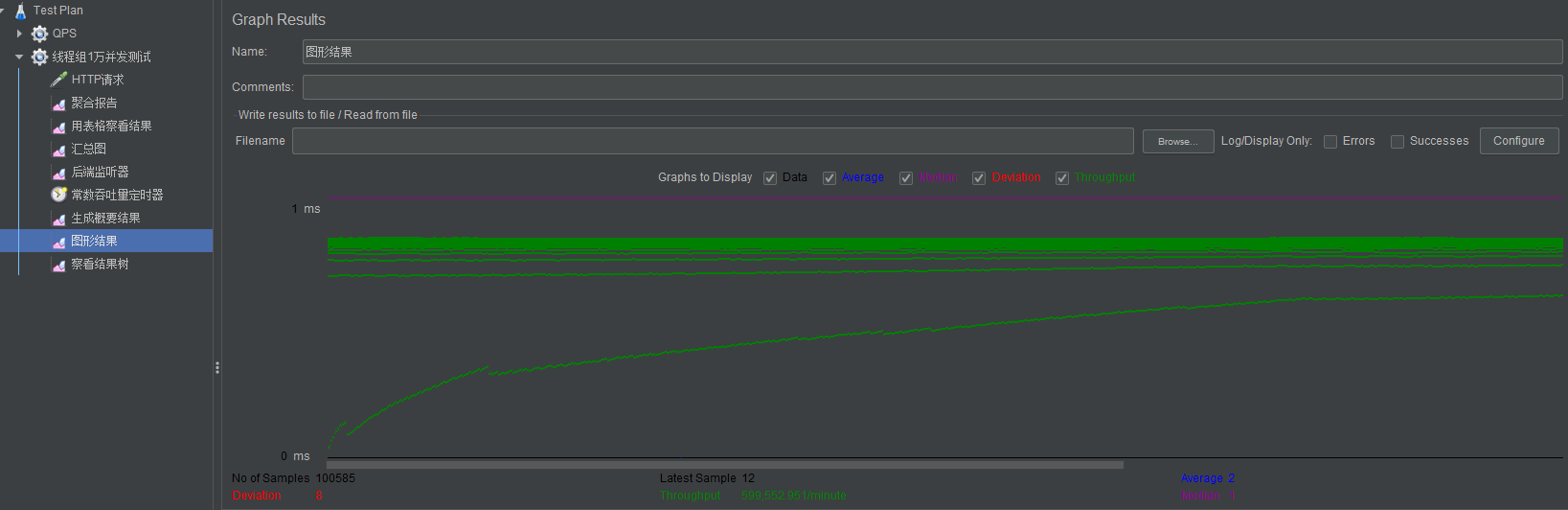
第一轮压测 QPS=9968.2/S

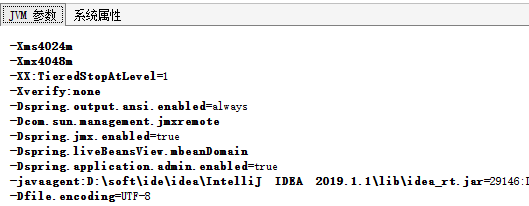




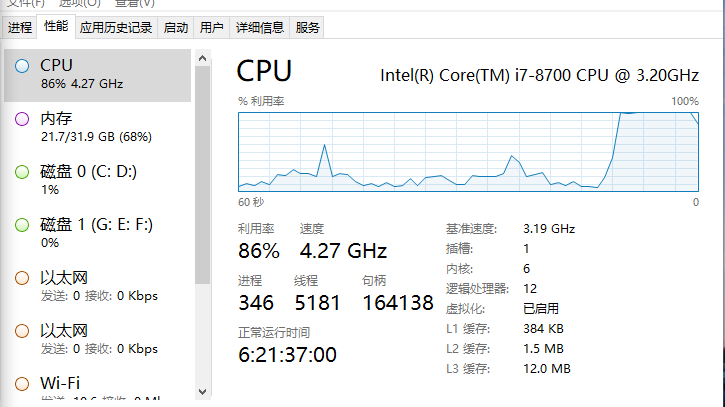
第二轮压测: QPS=9992.5/S

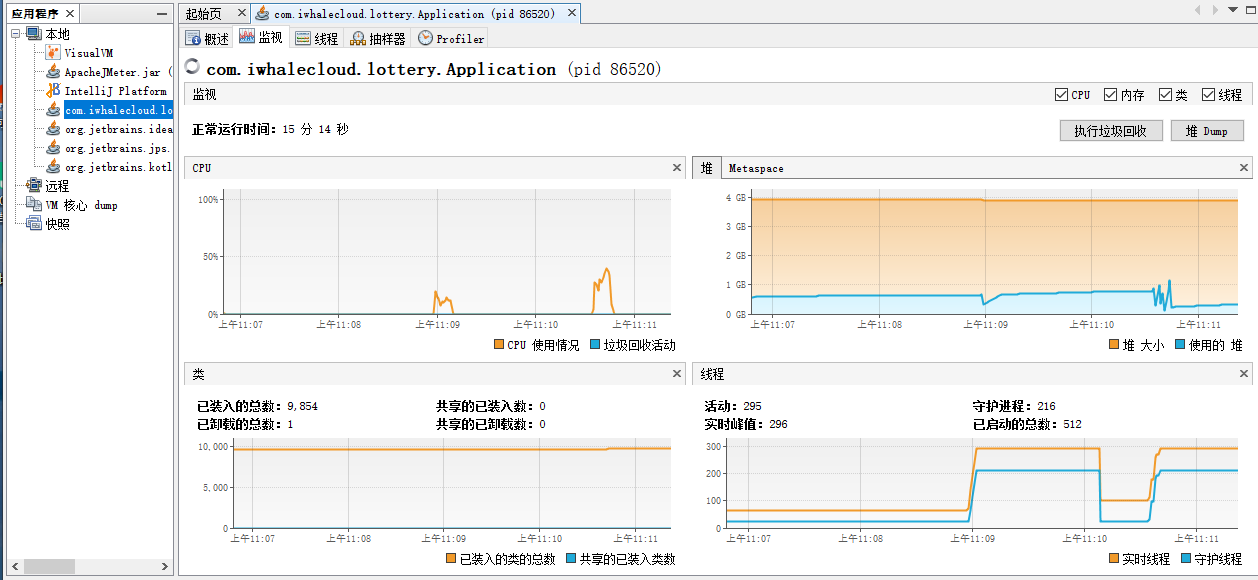






**单机服务,瓶颈在于CPU**

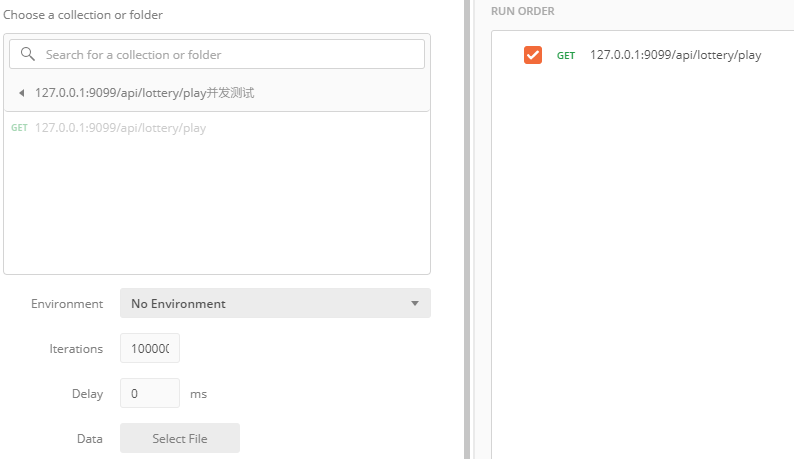




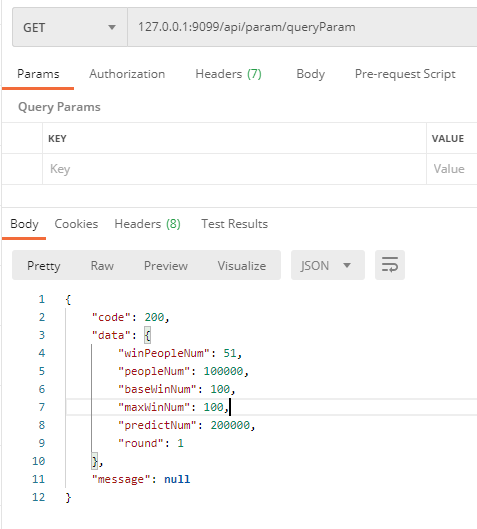
# 业务指标测试

第一轮测试:

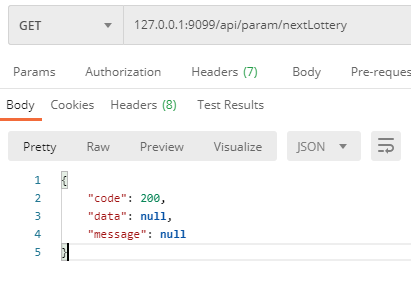
10万请求



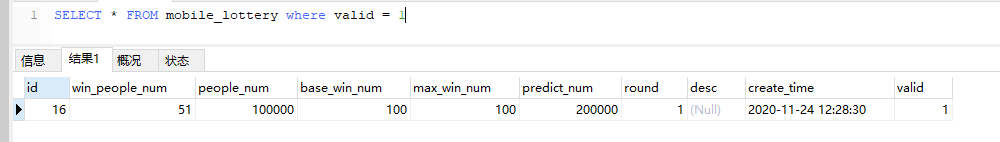
查看当前轮参数



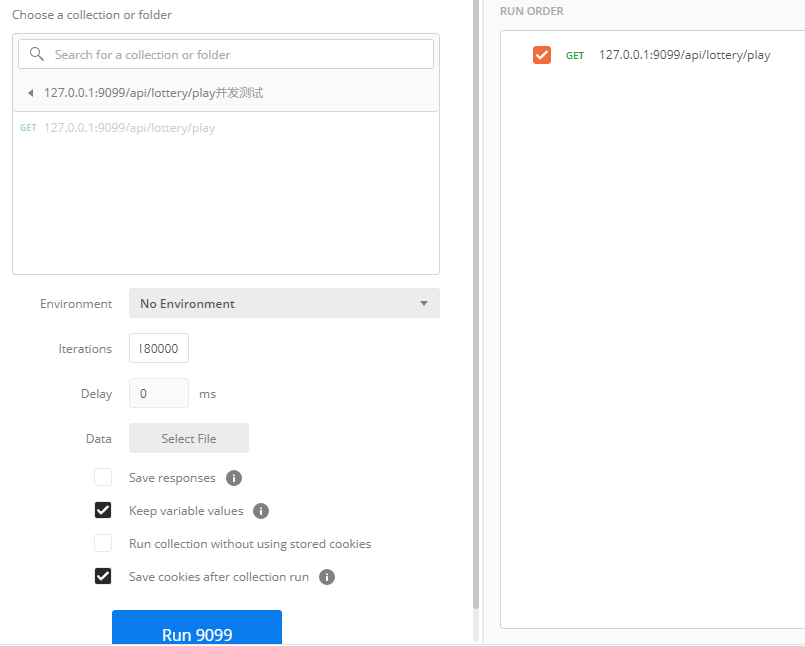
下一轮测试:

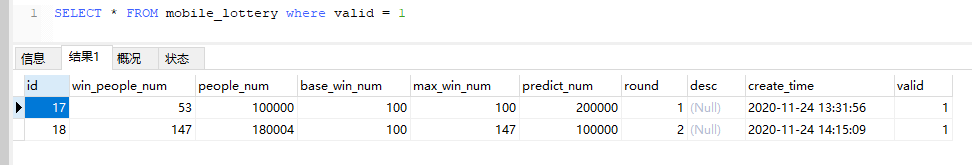


查看结果:

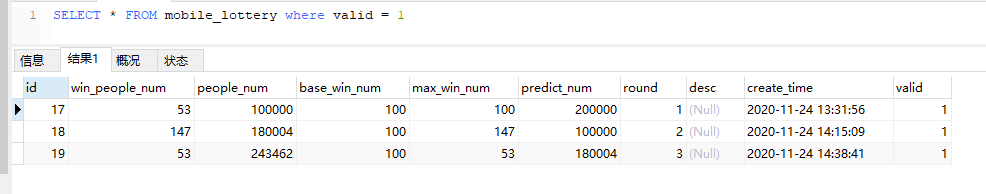


第二轮测试: 18万请求





第三轮:



# 表设计

1. 新建mysql数据库 lottery库
2. 新建mobile\_lottery表

CREATE TABLE `mobile\_lottery` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`win\_people\_num` int(11) DEFAULT '0' COMMENT '获奖人数',

`people\_num` int(11) DEFAULT '0' COMMENT '抽奖人数',

`base\_win\_num` int(11) DEFAULT NULL COMMENT '获奖基数',

`max\_win\_num` int(11) DEFAULT NULL COMMENT '最大获奖人数',

`predict\_num` int(11) DEFAULT NULL COMMENT '预测抽奖人数',

`round` int(11) DEFAULT NULL COMMENT '第几轮',

`desc` varchar(255) DEFAULT NULL,

`create\_time` datetime DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP,

`valid` int(11) NOT NULL DEFAULT '1' COMMENT '0无效1有效',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=16 DEFAULT CHARSET=utf8mb4;

# 测试环境

前端测试地址: <http://10.45.53.111:9099/index.html>

