目标: 测试最大并发数量

测试成功标识: 服务器返回信息超时，cpu过高，内存泄漏，IO异常等

测试环境:

服务器配置

cpu: Intel(R) Core(TM) i3-2100 3.10GHz

Memory: 4G

带宽: 局域网内 理论为100M，实际效果为 上行带宽: 4.93Mbps 下行带宽: 25.50Mbps

软件配置:

OS: WIN10专业版

JDK: 1.7.0.8  
TOMCAT: Tomcat7.0

数据库: Mysql

测试工具 [Apache HTTP Server benchmarking Tool(ab)](http://httpd.apache.org/docs/2.2/programs/ab.html)

画图工具 gnuplot　(脚本文件 run1.dem)



### 测试1: 登录并发最大数量，maxResponse=5s

ab -n 5000 -c 5000 -p "data/data.txt" -g apache.tsv -T "application/json" "http://192.168.1.119:8080/api/user/login"

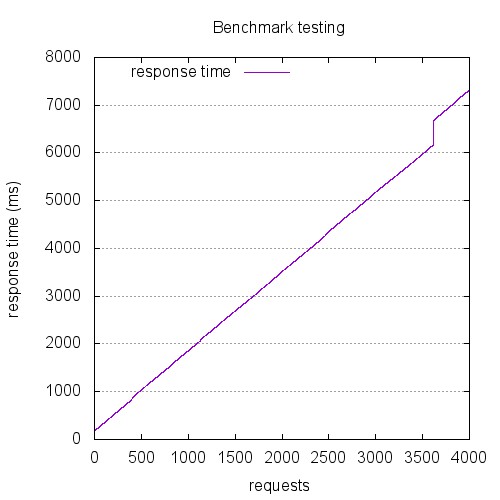
POST Json数据 (data.txt)

{

"email": "chen@qq.com",

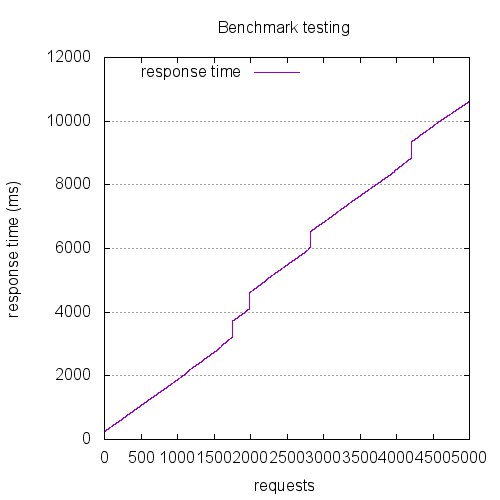
"password": "appmima1"

}



(图1.1 n=4000, c=4000)

用户数n=4000，并发数c达到4000时，max=3000



(图1.2 n=5000， c=5000)

用户数n=5000，并发数c达到5000时，max=2500

### 测试2: 同时搜索最大并发数量，maxResponse=5s

ab -n 2000 -c 2000 -p "data/data.txt" -g apache.tsv -T "application/json" "http://192.168.1.119:8080/api/restaurant/query"

POST Json数据 (data.txt)

{

"lng":114.0468853422,

"lat":22.5437818091,

"restaurantType":7,

"page":1,

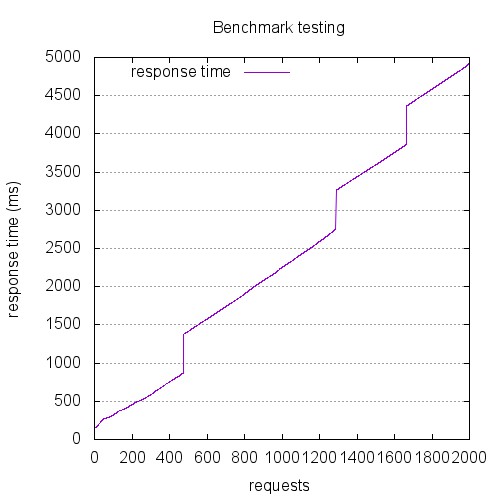
"foodstylesList":[

{"id":0, "appId":1, "foodstyle":"rice"},

{"id":0, "appId":1, "foodstyle":"chicken"}

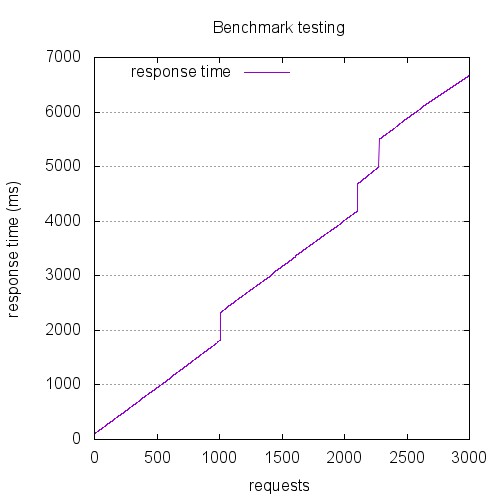
]

}



(图2.1 n=2000, c=2000)

用户数n=2000，并发数c达到2000时



(图2.2 n=3000, c=3000)

用户数n=3000，并发数c达到3000时 max=2200

### 测试3: 同时下单最大并发数量，maxResponse=5s

ab -n 2000 -c 2000 -p "data/data.txt" -g apache.tsv -T "application/json" "http://192.168.1.119:8080/api/order/init"

POST Json数据 (data.txt)

{

"order":{

"tableId":1,

"customerId":1111,

"customerName":"我压力测试勒",

"deliveryArriveTime":"03:25 PM",

"takeoutTime":"03:25 PM",

"personNum":2,

"tableName":"Delivery",

"waiterName":"Customer",

"cashierName":"Customer",

"kitchenRemark":"nothing",

"subTotal":30,

"amount":40,

"qty":2,

"customerPhone":"125800336",

"orderType":6,

"companyId":1,

"companyName":"淘宝",

"companyPhone":"1234",

"deliveryAddress":"Gghh Fbhn Bbnn",

"orderStatus":"2",

"orderPayments":[{

"amount":40,

"paymentMethodName":"masterpass",

"paymentMethodType":-1,

"cashierName":"Customer",

"changeAmt":0,

"paid":40

}],

"orderItems":[{

"categoryName":"test",

"categoryId":1,

"categorySequence":1,

"qty":2,

"itemId":1,

"itemPrice":16,

"itemName":"rice",

"itemCost":10,

"originalPrice":12

},

{

"categoryName":"test",

"categoryId":1,

"categorySequence":1,

"qty":2,

"itemId":2,

"itemPrice":26,

"itemName":"chicken",

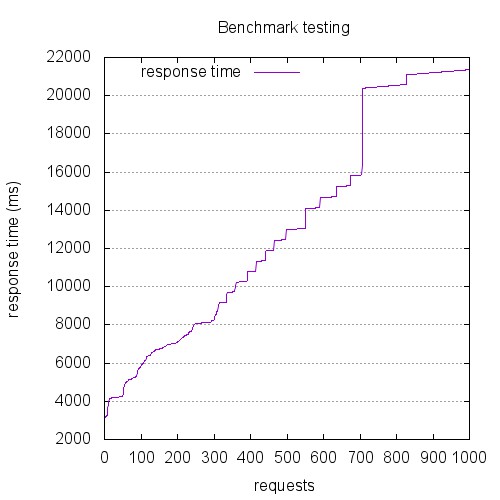
"itemCost":20,

"originalPrice":42

}]

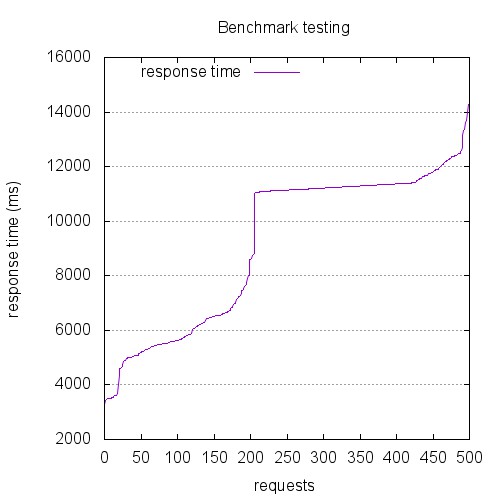
}

}



(图3.1 n=1000, c=1000)

用户数n=1000，并发数c达到1000时 max=100



(图3.2 n=500, c=500)

用户数n=500，并发数c达到500时 max=100

总结:

因为查询时总的来说没有一次性拿很多数据，比如搜索和获取附近餐厅，都是一次拿[10, 20] 条数据，而下单属于向数据库中插入数据，IO压力大，最大最大并发为100。

后续测试为真实服务器测试，因为带宽，内存的原因估测为现在的3/4\*max