**项目实施案例分析及调优**

**移动APP**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档编号： |  | 文档名称： |  |
| 编 写： |  | 审 核： |  |
| 批 准： |  | 批准日期： |  |

**PTS 产品中心**

文件修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| **修改日期** | **版本号** | **修改描述** | **作者** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

目录

[1 背景 1](#_Toc412706422)

[2 测试目标 1](#_Toc412706423)

[3 架构 1](#_Toc412706424)

[4 测试指标 2](#_Toc412706425)

[5 业务模型 2](#_Toc412706426)

[5.1 分析 2](#_Toc412706427)

[5.2 模型 4](#_Toc412706428)

[5.2.1 模型1 4](#_Toc412706429)

[5.2.2 模型2 4](#_Toc412706430)

[5.2.3 模型3 5](#_Toc412706431)

[5.2.4 模型4 5](#_Toc412706432)

[6 脚本设计 6](#_Toc412706433)

[7 测试结果 6](#_Toc412706434)

[7.1 容量测试 6](#_Toc412706435)

[7.1.1 测试场景 6](#_Toc412706436)

[7.1.2 测试结果及分析 6](#_Toc412706437)

[7.1.3 测试结论 9](#_Toc412706438)

[7.2 线上线下资源消耗对比测试 9](#_Toc412706439)

[7.2.1 测试场景 9](#_Toc412706440)

[7.2.2 测试结果及分析 9](#_Toc412706441)

[7.2.3 测试结论 10](#_Toc412706442)

[7.3 线上线下存储访问时间对比测试 10](#_Toc412706443)

[7.3.1 测试场景 10](#_Toc412706444)

[7.3.2 测试结果及分析 10](#_Toc412706445)

[7.3.3 测试结论 10](#_Toc412706446)

[7.4 突变测试 11](#_Toc412706447)

[7.4.1 测试场景 11](#_Toc412706448)

[7.4.2 测试结果及分析 11](#_Toc412706449)

[7.4.3 测试结论 12](#_Toc412706450)

[7.5 恢复性测试 12](#_Toc412706451)

[7.5.1 测试场景 12](#_Toc412706452)

[7.5.2 测试结果及分析 13](#_Toc412706453)

[7.5.3 测试结论 14](#_Toc412706454)

[7.6 稳定性测试 14](#_Toc412706455)

[7.6.1 测试场景 14](#_Toc412706456)

[7.6.2 测试结果及分析 14](#_Toc412706457)

[7.6.3 测试结论 17](#_Toc412706458)

[8 风险及建议 17](#_Toc412706459)

# 背景

随着客户业务发展，目前系统架构已不能满足业务发展需要，因此急需将服务器托管到阿里云上，并进行扩容；迁移到阿里云上以后，系统资源消耗是否比目前线上环境结果要好。因此在上线前需要进行性能测试，测试是否满足各项性能指标。

# 测试目标

本次测试目标如下：

* 容量测试：核心业务(核心业务1+核心业务2)+非核心业务基线（非核心业务1+非核心业务2+非核心业务3+非核心业务4+非核心业务5+非核心业务6）混合交易容量
* 稳定性测试：混合交易稳定性
* 突变测试：非核心业务突变3倍，对核心业务的影响
* 对比测试：和线上同等压力下，线上和线下资源消耗和响应时间对比。
* 恢复性测试：模拟网络攻击

# 架构

系统架构主要有如下服务器：

* HTTP服务器：核心业务1和核心业务2业务
* TCP服务器：核心业务使用人员终端心跳业务
* MongoDB服务器：非结构化数据库存储
* Redis服务器：信息推送
* MySQL服务器：结构化数据库存储

# 测试指标

* 容量测试：核心业务1 TPS>=600笔/秒,核心业务2 TPS>=1200笔/秒
* 稳定性测试：至少在核心业务1 TPS等于300笔/秒和核心业务2 TPS等于600笔/秒能稳定运行8小时
* 突变测试：非核心业务突变3倍，基本对核心业务无影响
* 线上线下资源消耗对比测试：在跟线上核心业务1 TPS等于150笔/秒和核心业务2 TPS等于120笔/秒同等压力下，测试环境的MonogoDB和Redis CPU Load小于0.5%,磁盘利用率小于0.1%
* 线上线下存储访问时间对比测试：在核心业务1 TPS等于200笔/秒和核心业务2 TPS等于400笔/秒的情况下，应用观察到的存储访问平均耗时不超过4ms，最大耗时不超过100ms。
* 恢复性测试：系统能恢复，TPS无变化

# 业务模型

## 分析

通过生产上高峰业务量分析得出，核心业务1和核心业务2除了双12外，比例占比1：1.5左右，通过系统整个趋势观察，发现核心业务2业务量有明显增长趋势，因此核心业务1和核心业务2的占比为1：2。高峰时候核心业务总的TPS只有50~200笔／秒。

核心业务量：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 时间点 | 业务 | 合计 | 占比 | 平均 | 占比 | 最大值 | 占比 | 最大值TPS |
| 2014/12/12 17：00~18：00 | 核心业务1 | 351348 | 0.601096299 | 5855 | 0.605043 | 6403 | 0.604513 | 106.716667 |
| 核心业务2 | 233164 | 0.398903701 | 3822 | 0.394957 | 4189 | 0.395487 | 69.8166667 |
| 　 | 合计 | 584512 | 　 | 9677 | 　 | 10592 | 　 | 　 |
| 　 | 　 | 　 | 　 | 　 | 　 | 　 | 　 | 　 |
| 时间点 | 业务 | 合计 | 占比 | 平均 | 占比 | 最大值 | 占比 | 　 |
| 2015/01/04 07：00~08：00 | 核心业务1 | 143323 | 0.245201125 | 2349 | 0.242741 | 3425 | 0.430385 | 57.0833333 |
| 核心业务2 | 245052 | 0.419242034 | 4084 | 0.422032 | 4533 | 0.569615 | 75.55 |
| 　 | 合计 | 388375 | 　 | 6433 | 　 | 7958 | 　 | 　 |
| 　 | 　 | 　 | 　 | 　 | 　 | 　 | 　 | 　 |
| 时间点 | 业务 | 合计 | 占比 | 平均 | 占比 | 最大值 | 占比 | 　 |
| 2015/01/06 07：00~08：00 | 核心业务1 | 126989 | 0.217256446 | 2116 | 0.218663 | 3123 | 0.360499 | 52.05 |
| 核心业务2 | 257128 | 0.439902004 | 4215 | 0.435569 | 5540 | 0.639501 | 92.3333333 |
| 　 | 合计 | 384117 | 　 | 6331 | 　 | 8663 | 　 | 　 |
| 　 | 　 | 　 | 　 | 　 | 　 | 　 | 　 | 　 |
| 时间点 | 业务 | 合计 | 占比 | 平均 | 占比 | 最大值 | 占比 | 　 |
| 2014/11/11 17：00~18：00 | 核心业务1 | 81702 | 0.13977814 | 1361 | 0.140643 | 1632 | 0.400294 | 27.2 |
| 核心业务2 | 111120 | 0.190107303 | 1852 | 0.191382 | 2445 | 0.599706 | 40.75 |
| 　 | 合计 | 192822 | 　 | 3213 | 　 | 4077 | 　 | 　 |

非核心业务量：

非核心业务1+非核心业务2+非核心业务3+非核心业务4+非核心业务5+非核心业务6

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **业务** | **TPS** | **占比** |
| 1 | 非核心业务1 | 400 | 16.4% |
| 2 | 非核心业务2 | 500 | 20.5% |
| 3 | 非核心业务3 | 833 | 34.2% |
| 4 | 非核心业务4 | 210 | 8.6% |
| 5 | 非核心业务5 | 300 | 12.3% |
| 6 | 非核心业务6 | 190 | 8% |
| 合计 |  | 2433 | 100% |

## 模型

### 模型1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **业务类型** | **业务** | **占比** | **备注** |
| 1 | 核心业务 | 核心业务1 | 33.3% | 采用梯度施压测试，测出容量 |
| 2 | 核心业务2 | 66.7% |
| 3 | 非核心业务 | 非核心业务1 | 16.4% | 非核心基线，总的TPS为2433笔/秒，按照占比进行分配 |
| 4 | 非核心业务2 | 20.5% |
| 5 | 非核心业务3 | 34.2% |
| 6 | 非核心业务4 | 8.6% |
| 7 | 非核心业务5 | 12.3% |
| 8 | 非核心业务6 | 8% |

此模型用于容量测试、稳定性测试和恢复性测试。

### 模型2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **业务类型** | **业务** | **占比** | **备注** |
| 1 | 核心业务 | 核心业务1 | 33.3% | 按照测试出来的容量的50%压力运行 |
| 2 | 核心业务2 | 66.7% |
| 3 | 非核心业务 | 非核心业务1 | 16.4% | 非核心基线突变3倍，总的TPS为7299笔/秒，按照占比进行分配 |
| 4 | 非核心业务2 | 20.5% |
| 5 | 非核心业务3 | 34.2% |
| 6 | 非核心业务4 | 8.6% |
| 7 | 非核心业务5 | 12.3% |
| 8 | 非核心业务6 | 8% |

此模型用于突变测试。

### 模型3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **业务类型** | **业务** | **占比** | **备注** |
| 1 | 核心业务 | 核心业务1 | 55.5% | 按照核心业务1 150TPS和核心业务2 120TPS情况，资源消耗对比 |
| 2 | 核心业务2 | 44.5% |
| 3 | 非核心业务 | 非核心业务1 | 16.4% | 非核心基线，总的TPS为2433笔/秒，按照占比进行分配 |
| 4 | 非核心业务2 | 20.5% |
| 5 | 非核心业务3 | 34.2% |
| 6 | 非核心业务4 | 8.6% |
| 7 | 非核心业务5 | 12.3% |
| 8 | 非核心业务6 | 8% |

此模型用于线上线下资源消耗对比测试

### 模型4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **业务类型** | **业务** | **占比** | **备注** |
| 1 | 核心业务 | 核心业务1 | 33.3% | 总的TPS为600笔/秒，方法耗时 |
| 2 | 核心业务2 | 66.7% |
| 3 | 非核心业务 | 非核心业务1 | 16.4% | 非核心基线，总的TPS为2433笔/秒，按照占比进行分配 |
| 4 | 非核心业务2 | 20.5% |
| 5 | 非核心业务3 | 34.2% |
| 6 | 非核心业务4 | 8.6% |
| 7 | 非核心业务5 | 12.3% |
| 8 | 非核心业务6 | 8% |

此模型用于线上线下存储访问时间对比测试

# 脚本设计

经过调研，发送后台的业务均是URL+自定义Body方式，因此在PTS里面，新增一个脚本，上传参数化文件，定义事务，设置连接和Body就行了，注意尽可能多的进行参数化。



# 测试结果

## 容量测试

### 测试场景

按照模型１，设置用户数比例和步调时间(保持业务占比，不偏模型)，运行20分钟，进行负载测试。

### 测试结果及分析

* 第一轮测试

按照核心业务1 1000笔/秒和核心业务2 2000笔/秒目标发起压力，发现不能达到目标，TPS曲线不稳定，运行到１分钟的时候，下降非常厉害，抖动也非常厉害，通过监控，发现FULL GC非常频繁，达到１秒１次，经过与架构师沟通，这是由于实现机制导致的，核心业务1的机制是将内容放到队列里面，队列长度是2147483647,后台只有32个线程（不能修改）在消化，消费者（消化）处理速度的比生产者（核心业务1）慢，导致队列长度越来越大，内存很快被消化完了，导致FULL GC频繁，这属于架构问题，不能进行修改。

 　核心业务2：



* 第二轮测试

按照核心业务1 600笔/秒和核心业务2 1200笔/秒发起压力，运行20分钟，TPS基本保持稳定，通过监控发现，order应用连接MongoDB连接数报已满的异常错误、logserver IO过高、MongoDB locked DB值高于75%。

按照核心业务1 800笔/秒和核心业务2 1600笔/秒目标发起压力，不能达到此目标，TPS曲线非常不稳定。

* 第三轮测试

mongoDB 只有表锁没有行锁，导致locked值非常高，这个是产品问题，没办法进行调优。

将order应用MongoDB连接数从250调到1000;logserver　磁盘换成效率更高SSD磁盘；重新按照核心业务1 800笔/秒和核心业务2 1600笔/秒目标发起压力，运行20分钟，TPS曲线基本稳定。

核心业务1：



核心业务2：



### 测试结论

系统的容量为核心业务1 800笔/秒和核心业务2 1600笔/秒，满足核心业务1 600笔/秒和核心业务2 1200笔/秒目标要求。

## 线上线下资源消耗对比测试

### 测试场景

按照模型3发起压力，在核心业务1 150TPS和核心业务2 120TPS压力情况下，运行20分钟，资源消耗对比。

### 测试结果及分析

MongoDB 和 Redis CPU Load均小于0.5, CPU 利用率均小于10%,磁盘利用率均小于0.1%,　这些指标结果比线上资源消耗结果略好。

### 测试结论

在跟线上同等压力的情况下，阿里云环境各项指标结果略好于目前线上环境资源消耗。

## 线上线下存储访问时间对比测试

### 测试场景

按照模型４发起压力，在核心业务1 200笔/秒和核心业务2 400笔/秒的压力下，运行20分钟，观察存储访问的时间。

### 测试结果及分析

在xflush上面观察到的存储耗时值小于3ms，最大值不超过100ms

### 测试结论

满足目标平均耗时不超过4ms，最大耗时不超过100ms的需求。

## 突变测试

### 测试场景

按照模型２，在核心业务1 TPS 400笔/秒和核心业务2 TPS 800笔/秒的情况下，平稳运行５分钟后，将非核心业务按照基线的３倍进行突变，运行５分钟，观察核心业务TPS曲线的变化，然后将非核心业务恢复到基线，观察核心业务TPS曲线的变化。

### 测试结果及分析

核心业务1：



核心业务2：



从图中可以看出，当非核心业务突变3倍以后，对核心业务1和核心业务2有轻微的影响（核心业务1和核心业务2 TPS下降），但马上能恢复，突变的整个过程对核心业务基本无影响。

### 测试结论

非核心业务突变３倍对核心业务基本无影响，满足目标要求。

## 恢复性测试

### 测试场景

按照模型１，在核心业务1 800笔/秒和核心业务2 1600笔/秒的压力下，平稳运行５分钟后，断开所有mysql服务网络5秒，观察核心业务TPS曲线变化，然后恢复mysql网络，观察核心业务TPS曲线变化，接着断开所有MongoDB服务网络5秒，观察核心业务TPS曲线变化，然后恢复所有MongoDB服务网络，观察核心业务TPS曲线变化。

### 测试结果及分析

核心业务1：



核心业务2：



从图中可以看出，断开MySQL和MongoDB网络5秒的瞬间，核心业务1和核心业务2的TPS有轻微的下降，随后能恢复到正常水平，因此对核心业务基本没有影响。

### 测试结论

模拟网络攻击，对核心业务基本没有影响，满足目标要求。

## 稳定性测试

### 测试场景

按照模型1和最大容量的80%左右发起压力(核心业务1:600笔/秒和核心业务2：1200笔/秒)，运行8小时，观察系统是否能稳定运行。

### 测试结果及分析

核心业务1：



核心业务2：



运行到35分钟后，核心业务1和核心业务2 TPS开始有轻微大幅度波动，运行到45分钟后，核心业务1和核心业务2 TPS开始大幅度波动，比较频繁，并且不能恢复到初始水平（过一段时间，TPS逐渐在下降），经过分析发现是FULL GC导致，详见7.1.2测试结果及分析。

因此将压力降为一半（核心业务1：300笔/秒，核心业务2：600笔/秒），重新运行稳定性测试。

核心业务1：



核心业务2：



系统在核心业务1 300笔/秒和核心业务2 600笔/秒的压力下，基本能稳定运行8小时，但随着时间推移，Full GC次数越来越多，长时间运行下去将会导致系统处理能力大幅度下降（详见7.1.2测试结果及分析）

### 测试结论

在核心业务1 300笔/秒和核心业务2 600笔/秒的压力下，系统基本能稳定运行8小时，满足目标要求。

# 风险及建议

经过多次分析、调优及测试，基本能满足各业务场景的目标要求，但系统处理能力不能再继续上升的瓶颈主要体现在系统架构上，因此随着未来业务量猛增，超过系统处理能力的时候，将会产生处理能力急需下降、客户体现差甚至宕机的风险，建议针对系统架构进行修改，并进行架构类性能测试，满足日益增长的业务需要。