性能测试方案

# 1.文档介绍

## 1.1测试目的

本测试方案将对SDK广告平台系统的测试工具、测试流程、测试方法、测试范围、测试的软件硬件环境以及风险与应急进行详细的定义和整体的描述。

## 1.2术语及解释

* 性能测试：在一定约束条件下（指定的软件、硬件和网络环境等）确定系统所能承受的最大负载压力的测试过程。
* 场景：一种文件，用于根据性能要求定义在每一个测试会话运行期间，发生的事件。
* 虚拟用户：在场景中，LoadRunner用虚拟用户代替实际用户。模拟实际用户的操作来使用应用程序。一个场景可以包含几十、几百甚至上千个虚拟用户。
* 虚拟用户脚本：用于描述虚拟用户在场景中执行的操作。
* 事务：表示要度量的最终用户业务流程。

## 1.3测试目标

通过对SDK广告平台的性能测试，在测试范围内可以达到如下的目的：

* 了解业务系统在各种业务场景下的性能表现；
* 了解业务系统的稳定性情况；
* 通过各种业务场景的测试实施，为系统调优提供数据参考；
* 预估业务系统的业务容量

## 1.4测试范围

主要为客户端向服务器发送请求，获取广告相关信息等接口。

# 2.测试环境

## 2.1测试环境

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CPU | 内存 | 硬盘 | 操作系统 | 配置环境 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 2.2测试工具

负载性能测试工具的原理通常是通过录制、回放脚本、模拟多用户同时访问被测试系统，制造负载，产生并记录各种性能指标，生成分析结果，从而完成性能测试的任务。

### 2.2.1压力负载工具

HP-Mercury LoadRunner 是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner 能够对整个企业架构进行测试。通过使用LoadRunner 企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。

Virtual User Generator 用来创建虚拟用户脚本；

Controller 用来创建、运行实际场景工具。

### 2.2.2性能监控工具

Spotlight on MySQL是一款用于监视MySQL数据库服务器的工具，软件是Quest公司发布，SpotLight监控MySQL的基本原理与LoadRunner监控类似，通过获取MySQL的数据字典和动态性能视图，然后把性能数据按直观的方式展现出来。

### 2.2.3其他

对于Linux的性能监控，除了利用spotlight on mysql 获取性能数据外，结合Linux本身的命令，如iostat、vmstat、top等命令来查看服务器的资源使用情况。

对于Mysql的性能监控结合其自身的show status等命令来获取mysql的锁、慢查询等相关性能信息。

# 3测试流程

性能测试通常由五个阶段组成。计划、脚本创建及调试、场景定义、场景执行和结果分析。

* 计划部分：定义性能测试要求，例如并发用户量、典型业务流程和所需响应时间等；
* 脚本创建及调试：将最终用户的操作转化成脚本，包括录制及手动编写等；
* 场景定义：使用测试工具设置负载测试环境；
* 场景执行：通过测试工具驱动、管理和监控负载测试；
* 结果分析：分析测试执行过程中收集到的性能数据，进行性能分析。包括：响应时间、TPS、点击率、吞吐量和系统资源消耗情况等。

# 4测试方法

## 4.1测试点选取

目前会对服务器造成性能压力的接口包括：①用户打开客户端时发送给服务器的获取广告列表信息请求会对服务器造成压力影响；②用户访问指定广告，如图片、视频及声音等，也会向服务器发出请求，亦会对服务器造成压力。

## 4.2测试发起策略

本次测试采用HP Loadrunner压力测试工具，模拟手机客户端APP像广告平台接口发送请求。通信协议为HTTP协议。

## 4.3测试场景策略

性能测试场景一般包括基准场景、混合场景、容量测试场景和稳定性测试场景。不同的测试场景对应的测试目的也不相同。

|  |  |
| --- | --- |
| 场景类型 | 测试目的 |
| 基准场景 | 获取单事务的处理能力，验证是否存在并发性问题 |
| 混合场景 | 验证相互关联的一组交易之间是否存在性能冲突 |
| 容量测试场景 | 模拟实际生产环境中在业务处理高峰时期系统的压力情况，得到此时的系统性能表现数据，为系统的实际上线运行提供参考 |
| 稳定性场景 | 验证系统在长时间运行情况下的稳定性 |

本次性能测试主要选取基准场景、混合场景、容量测试场景合稳定性场景，基准场景及容量测试过程中，每组测试运行30分钟。根据app客户端监控得来的数据，混合场景测试的时间为xx分钟。系统的运行时间为7\*24小时，但大部分用户使用客户端的时间在12小时之内，故稳定性场景将持续运行12小时。

## 4.4数据准备

* 系统数据准备

系统后台增加图片、GIF、视频广告至少3个广告位。数据库中至少有1年的业务数据量。因为如果测试环境的数据量级与生产系统相差很远的情况下，其测试的结果的真实性和准确性是要打折扣的。

* 测试数据准备

测试脚本编写过程中涉及到的需要在系统中提前准备好的业务数据。

* 测试脚本数据准备

针对选取的接口，编写及调试相对应的测试脚本。

## 4.5用户比例

测试过程中采用阶梯的方式增加虚拟用户数，查看系统在不同虚拟用户数下的性能状况。如500-1000-1500-2000-2500-3000这几个阶段。

## 4.6监控策略

测试结果分析、测试报告用到的数据来源于测试监控与测试结果统计，所以测试监控是性能测试必不可少的环节。一般遵循如下原则

必要的监控指标优先，如：CPU利用率、可用内存、磁盘IO等。

监控命令优先：一般若系统自带的监控工具或者命令可以满足监控需求，就不用第三方监控工具。因为系统自带的监控工具是最简单、最直接的监控采样器，很多第三方工具都是到系统自带的监控工具中获取相关监控指标，监控采样比较耗时、耗资源，为了不影响性能测试结果的可靠性，提高性能测试的准确率，尽可能避免不必要的间接采样交互，故此建议系统自带监控工具（或命令）。

监控工具尽量避免网络交互：网络丢包、延迟等风险很多，因此尽量选择不受网络影响的工具。

选择对系统占用资源较少的工具：有些监控工具非常全面，但是监控同时本身耗费的资源过大，导致影响了被测系统的处理能力。

对采样点、采样时间间隔合理设置，综合考虑：采样点的详细程度、采样时间间隔都会影响到被测系统的性能表现，如果短时间的调优测试，采用即时监控方式，那么采样点应尽量详细，采样时间间隔尽量短，目的在于排查、调优，如果是稳定性测试则建议采样点尽量减少，采样时间适当长一些。关注点不同策略也要相对变更。

## 4.6数据收集

* 事务响应时间；
* 服务器CPU利用率；
* 服务器可用物理内存；
* 服务器网卡数据流量；
* 并发用户数；
* 数据库慢查询等。

# 5风险与应急

预测测试计划中的风险，规定对各种风险的应急措施（如延期的测试项可能需要加班来赶上规定的进度）。

## 5.1风险因素

对测试执行过程中，可能存在的影响测试按时完成的风险因素进行分析。在测试执行过程中，通常可能存在如下因素影响测试的进行：

* 被测试产品的重大Bug，以致测试无法继续，需要研发进行额外的调试和修改才能继续；
* 硬件、软件或网络环境出现故障灯。

## 5.2应急措施

对上述潜在风险因素的应急措施逐项给以明确规定。通常的应急措施有：

* 通过适当加班来保证测试的按时完成；
* 如果是由于被测试产品存在重大错误而影响测试进行，则考虑暂停测试。