
目录

1.	CQ 性能测试整体实施方案概述.....	- 3 -
1.1	性能测试实施五个阶段.....	- 3 -
1.1	现阶段性能测试的问题.....	- 3 -
1.3	前期预计需要解决问题.....	- 4 -
1.4	后期逐步需要解决问题.....	- 4 -
2	CQ 性能测试需求.....	- 5 -
2.1	前期性能指标：（需与研发共同确定）.....	- 6 -
	后期性能指标（待新的测试方案完全建立后，与研发共同确定）.....	- 7 -

1. CQ 性能测试整体实施方案概述

CQ 整体性能测试方案概述，此次规划是 CQ 性能测试的初期规划。本次规划的目的是针对现有测试流程及测试方法，制定新的更加规范的测试体系优化测试方法提高测试效率。作为初期的性能测试规划，我们将初步分为以下四个阶段进行实施。从根本解决现有方案的瓶颈问题，规范测试步骤使测试数据更加准确、真实、可靠。

1.1 五个阶段（CMM 质量成熟度模型相似）：

1. 模式分析阶段（使用原始测试方案，提出新的测试需求，提出可逐步改善性能测试需求模型，尝试性的使用初步确立的新测试方案）
2. 模式改进阶段（使用新的测试方案，提出符合现有测试流程改善建议重新定义更佳完善的性能指标，逐步确定和完善性能测试需求模型）
3. 模式持续改进阶段（针对不同测试目的，使用较为固定的测试方案，确立可控的测试需求，定义较为固定的性能测试模型、提出可持续改进的过程，建立持续改进机制。
4. 模式定性阶段（针对一个时期内的测试需求模型进行较为准确统计与归纳，确立成型的测试体系，深入的划分可持续改进的区域，形成规范的测试体系，提出有效的度量方法）
5. 模式探索阶段（针对单位年时间内测试需求模型改进数据统计，探索下一个单位年预计可持续改进区域，建立性能测试产品价值链模型（为产品生命周期的各个阶段提供可靠性能测试数据支持）

1.2 现阶段性能测试的问题（排名不分先后）：

1. GameServer 服务器异常死机
2. GameServer 服务器周期性死机
3. 如何通过几组性能测试数据，确定 GameServer 服务器是否能稳定持续运行 144 单个工作目（既：24（小时/天）×6（天））

-
4. GameServer 服务器负载测试（机器人登陆的最大瓶颈、负载的峰值[数值、时间]、效率[cpu、Memory、磁盘的 I/O 消耗情况]）
 5. GameServer 服务器排查性验证(启动服务时不加载某些 model、不开启某些地图等)
 6. ……

1.3 前期预计需要解决问题:

1. 常规性测试:

- GameServer 服务器负载测试
- GameServer 服务器稳定性测试（既：模拟数据增长到 144 小时所需要的数据量，从而在短时期内对额定单位时间进行预估）
- GameServer 服务器异常死机（Cpu 瓶颈、内存泄漏问题[如：[obj 对象总数、每个地图能承受的 obj 最大总数、obj 对象开销细分统计]、磁盘 I/O 瓶颈。
- ……

2. 排查性测试:

- 验证新功能及对某个模块进行加载或关闭操作，对 GameServer 服务器造成的性能影响
- GameServer 周期性异常时间（根据：一个阶段的收集的数据，整理服务器在现有的发布周期内[如:每周五发布新的活动与修改 Bug 内容]服务器负载的变化情况，从而进一步确定产品的周期内的性能质量情况。（为周质量数据控制提供参考依据）
- ……

1.4 后期逐步需要解决问题:

- 定义在统一的性能测试模型 (PGMT) 下，针对不同的测试类型建立可控的测试场景，形成较为规范的测试文档（阶段性的输入、输出文档）与体系。
- 在前期的性能测试基础上，为产品研发部、质量控制小组提供多方位性能测试数据统计与分析，给出较有意义的测试报告。力图帮助各单位小组尽可能的减少产品的实施成本，缩短单位人日消耗。
- GameServer 压力性测试(如:组织模拟真实用户进行并发操作,以得到现阶段服务器性能质量参考值,对服务器健康情况进行估计)
- RoleDBServer 压力性测试(疑问:客户端哪些指令和操作(动作)需要与角色服务器或 Pays 服务器进行数据交换)
- GameServer 容错性测试（如：模拟意外断线、亚健康网络主干环境、

- 大数据量验证、数据包攻击（假设：存在外挂或恶意数据）、错误数据（如：模拟一段时间内发生意外宕机的错误代码，以验证周期错误的回归比率、为产品版本基线控制改进提供理论依据）等）
-

2. CQ 性能测试需求

备注：采用研发需要哪些性能参量，前期与后期两个部分，指标（性能）与测试场景（阶段）的关系。由于现有的测试方案，主要通过使用 Linux 命令来对 GameServer 服务器进行监测。由此，在未启动自动化测试方案前，我们还需要通过原始的测试方案进行数据监控，这样我们可参考的性能指标就较自动化方案略微减少。

2.1 前期性能指标：（需与研发共同确定）

①System	%User Time	系统上所有处理器执行非内核操作的平均响应时间的百分比，该值反映了用户有作业的时间比率。
	CPU context switches	CPU 上下文切换。在 vmstat 的结果中该值显示为 cs。
	CPU us	CPU 的实际消耗百分比（CPU 占有率）。
①Memory	Free (KB)	可用物理内存数
	Swap (KB)	已使用的虚拟内存数量
	Cache (KB)	文件系统缓存
②Process	%CPU Usage	被处理消耗的处理时间数量。（PID 对应的消耗）
	Resident Size (KB)	进程保留的使用内存量。如果该值在测试过程中持续增长，很可能意味着该版本发生了内存泄漏。
②Processor	%Idle Time	CPU 总的空闲时间。如果该值持续低于 10%，表明瓶颈可能使 CPU。
	%User Time	非内核操作的 CPU 时间。如果系统使用大量算法或较复杂的计算，该值可能会比较大
	%IOWait Time	CPU 消耗在等待 I/O 处理上的时间，此值结合 I/O 的计数器考虑
①③Physical Disk	The Number of disk operations per second	显示每个磁盘每秒的被操作次数。
	Percent of time the is busy	指所选磁盘驱动器忙于为读写或写入请求提供服务所用的时间的百分比

	Reads (Writes) per sec	物理磁盘上每秒读、写的次数。两者相加应小于磁盘设备最大容量。在 iostat 的结果中，该值显示为 r/s 和 w/s
	Average Number of Transactions waiting for serviced	指读取（写入）请求（列队）的平均数。在 iostat 的结果中，显示为 wait.

- ①该计数器的值来源于 vmstat 应用的输出结果
- ②该计算器的值来源于 top 命令的输出结果
- ③该计算器的值来源于 iostat 命令的输出结果

前期性能指标（在特地的测试类型、测试场景的指标和阶段对应关系，与研发共同确定）

- 1. 启动期（关注的的数据、计算的数据、收集的数据）
- 2. 稳定期（关注的的数据、计算的数据、收集的数据）
- 3. 结束期（关注的的数据、计算的数据、收集的数据）

后期性能指标（待新的测试方案完全建立后，与研发共同确定）

- 1. 总体目标（假设：性能测试开展的标准、性能测试结束的标准）
- 2. 技术方案（采用：工具与方法，现有的方案与优化后的方案，对于方案预想，建立持续改进的过程）
- 3. 测试流程（采用：流程控制模型、性能测试相关的配置管理）
- 4. 测试数据（基准测试环境[服务器的基本配置[硬件与软件]、系统标准配置参量（涉及到后期的优化，如：profile 中的 export 等]、基准测试数据建立与约定[需要研发配合]、思考如何保证数据的真实性，需要研发一起约定实际标准文档已做限制）
- 5. 测试范围约定（每进行一次测试，必须明确测试目的，预期的数据范围与已有数据进行横向或纵向的对比测试关系，如：测试类型[对比测试、新功能验证、排查性测试、系统压力测试、系统负载测试、破坏性测试、容错性测试、稳定性测试）
- 6. 测试活动安排
 - 前期测试活动安排：常规的测试、阶段性验证测试（每周、每月的版本性能对比测试、以日版本为单位进行的性能对比测试）
 - 后期测试活动安排：
 - a) 新功能上线前的预先测试—考察新功能程序设计框架是否合理，达到人数上线需求
 - b) 根据已有的性能测数据进行估算，对后期游戏各个阶段游戏性能进行估计（如：服务器 N 年增长下的服务器健康情况、在网络条件不稳定的情况对服务器的健康影响等）

.....