基于云渲染的项目测试方案

该方案从 1. 测试目标 2. 数据准备 3. 测试环境 4. 测试流程 5. 测试人员分工 六个方面来叙写。

1. **测试目标**

在服务器端显卡数量配置的不同下，模拟进行客户端的实际操作。得出在满足客户使用需求（实时、流畅… 即：帧率>= 15FPS）的情况下，服务器能够支持的并发客户端数量。

用户体验最直接的感受是使用是否流畅，对应的最核心的技术参数是操作响应时间，响应时间很难测量，也很难有评判标准，只能测量其他容易测量的相关资源指标（如渲染服务器的CPU、GPU占用率，RAM和GM使用量 等）以推算系统的性能。

1. 数据准备（用于测试的模型）
   1. 模型生成

ModelGenerator参数说明：

Allowed options:

-h [ --help ] produce help message

--db arg sqlite .db file for output

--spanX arg (=100000) spanX

--spanY arg (=200000) spanY

--spanZ arg (=10000) spanZ

--cylinderNum arg (=0) cylinder num

--sphereNum arg (=0) sphere num

--ovalNum arg (=0) oval num

--torusNum arg (=0) torus num

--srand arg (=0) random algorithm seed

--equipRow arg (=1) equipment matrix row number

--equipCol arg (=1) equipment matrix col number

--spaceScale arg (=1.5) space scale of span(X&Y) between equipments

示例：ModelGenerator --db "C:\Users\zzh\AppData\Roaming\Fulongtech Kernel\ModelCache\data\kernel.db" --equipCol 3 --equipRow 5 --cylinder 150000 --srand 123 --spanY 330000 –spaceScale 1.5

解释：在指定db生成模型，有15套装置（3×5），每套装置有15万圆柱体，每套装置之间的间隔为1.5倍。

使用上述方法生成db之后，再使用Unlink处理db后才可以使用v2和v3加载。

注：由于只生成了PDSOFT格式的模型，经Unlink处理后，V/v2/v3都可以正常加载。

* 1. 地形生成

（地形生成工具写完之后再补充）

1. **测试环境**
   1. 硬件需求：

多块显卡，用于分层次（装不同块数的显卡）对服务器进行性能测试。

其它参数实地具体咨询并记录。

* 1. **软件需求：**（所有软件均需**64位版本**）
     1. 服务器端（应用服务器和云渲染服务器）：
        1. 应用服务器：安装Oracle 11g数据库、apache-tomcat-7.0.42、jdk 8（最新版并配置好环境变量）。
        2. 云渲染服务器：安装云渲染服务系统，并配置好相应参数。
        3. 云渲染服务器数据记录工具： CPU平均使用率、GPU平均使用率、内存占用量、显存占用量磁盘平均读写速率 的测试工具。
     2. 客户机端（多台客户机）：

多台客户机（屏幕分辨率为1920×1080）作为客户端，用于模拟实际客户进行并发访问。

1. **测试流程**（本次测试有2块K2可用，有20—30台客户机可用）
   1. 客户机端

按照正常系统部署方法，运行IE之后，对模型进行查看，记录在不同频率不同操作下，系统所对应的各项参数。

* + 1. 操作内容：模型的旋转、平移、缩放、高亮 等。部分功能的使用（例如：目录树定位、查找） 等。（某种操作的性能问题）（录制脚本原则）
    2. 操作要求：以实际使用情景为基准，尽量保证每次操作后都能看到新的模型模块被加载出来，以达到资源的最大化利用。
    3. 建议：每进行一次或几次操作后有短暂的停顿（1-3秒），用以保证模型加载。
    4. 对应于服务器端显卡的不同配置，首先将并发访问的客户端数量定为：理论支持最多客户端数量的1/2（即：8台客户机），测试并记录各项参数。理论支持最多数量：1块K1：32台，1块K2：16台。
    5. 将客户端数量增加/减少，测试并记录各项参数。

增加/减少策略：采用折半（首次为 最大值/2，之后为 上一次/2）的方式。例如：对于1块K2，理论上最多可支持16个客户端，可以先将其增加到8个，若满足需求，再增加4个到12个；若不满足需求，可以减少4个到4个。以此类推，具体数量根据显卡块数及实际情况确定。

测试时间

* 1. **服务器端**

1）安装1块显卡，进行客户机端的测量与记录。

2）安装2块显卡，进行客户机端的测量与记录。

3）服务器端记录数据

|  |  |
| --- | --- |
| 参数 | 数据（平均、最大） |
| 客户端数量 |  |
| CPU平均使用率<核数，频率，利用率> |  |
| GPU平均使用率 |  |
| 内存占用 |  |
| 显存占用 |  |
| 带宽占用 |  |
| 网络延迟 |  |
| 磁盘平均读写速率 |  |

**CPU**：CPU跟应用软件和使用场景强相关。CPU主频越高，用户体验越好；如果应用软件在多核下有优化，给VM分配更多的虚拟机核，用户体验也会提升；VM中看到的CPU占用率低于70%，体验较好；物理机的CPU占用率跟多VM并发率有关，如果一台物理机承载的VM多，多人同时抢占物理CPU，会导致所有人的体验都差，服务器物理CPU低于70%时，所有人的体验感觉都会较好。

**磁盘IO**：一般的程序在进行磁盘读写时，程序都是暂停等待，等待的时间就是核心的体验指标----IO响应时延，但这个指标也是很难测量，跟应用相关，也跟硬盘类型相关。容易测量的指标是IOPS（每秒IO次数），一般来讲，IO次数越多，存储负荷越重，体验越差。

1. **测试人员分工**

（可以使用“按键精灵”记录操作脚本，之后进行自动运行，情景重现，代替测试人员）。

寻找压力平衡配置（时间）

**附录**

1. 测试时的操作方式与加载模式也有关系，也与验证的瓶颈有关系。
   1. 在V1加载模式下，如果要验证对CPU、RAM、GPU的压力，则不需要什么操作，找一个较复杂场景，高亮一个模型就可以了；验证最大显存使用量则需要尽量让所有模型至少显示一次，也就需要在场景中到处游览才能得到最大内存需求量。
   2. 在V3加载模式下，主要性能问题出现在动态加载功能上。操作方式为：经常变换视角，并在变换视角后停顿一下以便模型进行加载，加载完成之后或几乎完成时就再次变换视角。
2. 各部件压力如何测定

在云桌面模式下，RAM和GM可以直接查看服务器端的各项参数。

CPU和GPU的压力除了要看服务器端的负载，也要在客户端上查看CPU和GPU各自负责的渲染阶段的处理时间，可以在客户端按再次s键，其中Event+Update+Cull+Draw为CPU处理时间，GPU为GPU处理时间。

CPU负载情况判定表：

|  |  |  |
| --- | --- | --- |
| CPU利用率  CPU处理时间 | 高 | 低 |
| 高 | 瓶颈（之一） | 算法有问题 |
| 低 | 还不错 | 完全不是问题 |

GPU同理。

1. 当各部件压力不均衡时的调整方法（应该不太容易调整到一个较平衡的负载状态）
   1. CPU：使用V1可使CPU利用率大幅下降，但内存和显存会大幅增加。
   2. RAM：可以使用V3（无妨：服务器增加内存较方便和廉价）。
   3. GPU：此时可以计算每个显卡可以支撑几个客户端（当然GM也是一个考查点）。
   4. GM：如果是贴图太大，可以使用TextureOptimizer处理；其它情况只能是使用了V1或缓存所有模型的V3模式。
2. 测试方法建议

服务器只有一个，各项参数的监测和记录只需要一个人。

客户端有多个，操作可以按键精灵来进行。如果按键精灵脚本写的好的话，数据的记录和观察只需要一到两个人，各自实际操作和监测一个客户端，进行直观的操作体验感受