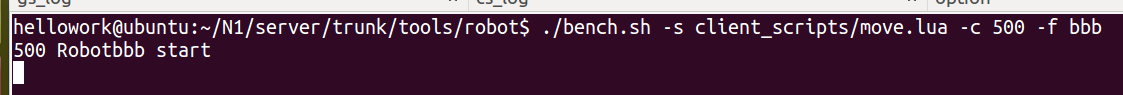
**压测流程：**

1. **程序评估需压测的功能模块性能，写好相应的模拟客户端脚本。(具体参考《模拟客户端使用》)**
2. **申请压测服(pressure-dev)、压测模拟客户端服(client-dev-1,client-dev-2,client-dev-3)。**
3. **同步代码至压测服pressure-dev以及各客户端服(同步指令pressure\_update.sh)，清除相关数据库数据，避免旧数据干扰，重启gs、cs进程。**
4. **利用三台压测模拟客户端服，每次bench.sh模拟大量玩家登录以及相应功能模块行为。**

**例如，如下指令是测试场景走路压力**

**-s 调用脚本 –c 机器人数量 –f 标识名称**

**(备注：了解top指令中相应参数含义，及时查看客户端服top指令，若CPU0核的繁忙程度过高，表明此时线程数过多，运行中存在大量线程切换调度，此时无法真实模拟玩家的操作频繁度，所以需要保证cpu0不能过于繁忙)**

1. **解决gs、cs报错问题，避免阻断真实性能损耗。**
2. **游戏客户端登录压测服(尽量用客户端build包，unity客户端过多的协议打印会比较卡，可尝试游戏中各模块功能玩法，感受响应速度)**

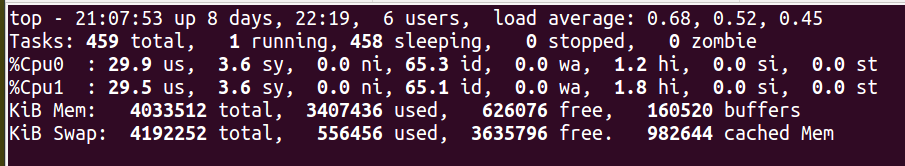
**同时游戏指令监控游戏服在线人数，以及各场景人数负载均衡**

**例如n1在线人数指令onlinestat，场景详细信息指令showscenestat**

1. **继续bensh.sh增加模拟客户端玩家数。**

**(debug\_console 主要关注每个服务stat(任务数，协程数)，cmem (c层内存),mem lua(虚拟机内存))**

**(密切关注top指令，若在游戏进程占总CPU百分比相对较低情况下，CPU核存指标占比过高情况，需要注意是否业务功能代码中存在问题)**

****

**us ：用户空间占用CPU百分比**

**sy ：内核空间占用CPU百分比**

**ni ：用户进程空间内改变过优先级的进程占用CPU百分比**

**id ：空闲CPU百分比**

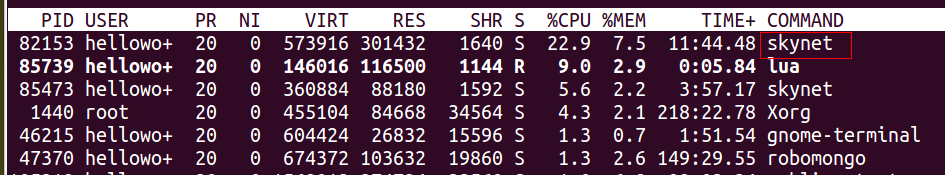
**wa：等待输入输出的CPU时间百分比**

**hi ：CPU服务于硬中断所耗费的时间总额**

**si ：CPU服务于软中断所耗费的时间总额**

**Mem ：total 物理内存总量 、used使用的物理内存总量、free空闲内存总量 、buffer用作内核缓存的内存量**

**Swap交换区参数，可自行了解**

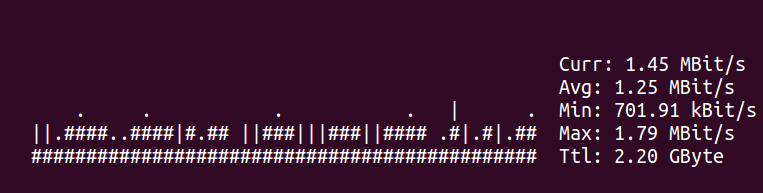
****

**CPU：CPU时间占用百分比**

**MEM：进程使用的物理内存百分比**

**其余可以自行了解**

**(流量监控(nload指令)，监控入站流量和出站流量，例如在测试场景的承载人数压力时，场景服务最大的消耗在于同步视野信息，此时我们可以通过nload来监控流量的大小)**

****

**8、最后确认在较高在线玩家数情况下，CPU总占相对较高且稳定情况下，CPU各个核占比类似，内存情况能在长时间内相对稳定。**

**压测规范：**

我们目前的压测设施是H7 N1共用，是在公司公用的DELL服务器上分配的VM。压测流程和指标希望大家都能熟悉，这对于项目健康发展和排除安全隐患有很大意义，另外对于自身成长也有较大意义，可以问下组里老司机，尤其主城需要特别关注项目的各种指标。由于是公用的设施，有些流程大家注意：

1、开发过程中，对于功能或者玩法，自己或者主城判断是否需要进行压测（CPU 内存 流量 IO等）。自行完成对应的压测脚本，并在本机自行压测完全再申请使用正式压测环境

2、有正式压测环境需求（或者体验），跟主城知会一声，然后在LUA服务端群知会一声，譬如：

需要使用压测环境：

目的：体验

时长：60分钟（一般需求不超过120分钟）

项目：H7

PS。外服上线使用的是云主机商提供的VM，会与目前正式压测环境有些差异，譬如在全公司服务器计算资源紧张情况下会发现CPU争用较为激烈，但按大部分时间来说应该够用