> 最后更新时间：2026 年 09 月

ECDN 资源测试流程

1 与厂商沟通确认要上机的资源类型，然后由我方提供：

- 测试程序包
- 上机需求区域(建议)
- 分配 provider、基础 provider_id 程序启动使用。注意每个厂商一般会分配一个基础 id.但对应不同资源类型(如大、小节点、盒子)id 会略微变化。（详见下面资源类型说明）
- 分配 dbid，供拉取接口时使用(可正式上量后再分配)

2 厂商先参阅程序包里的程序说明文档，按合同议定资源类型上机测试，上 1-2 台设备进行功能性测试。

- 该过程主要是通过文档了解跑量的基本背景流程、程序的部署及设备要求等(一般 1-2 天左右)。

3 功能验证完后尽快按提供的需求区域上机(按想上的资源类型资源尽量多区域分散上一些)，提供 15～20G 左右资源线上跑 2～4 周(看指标运行情况决定是否要延期)。

- 该过程主要是验证下资源的长期稳定性和质量，中间进一步熟悉出现常见问题的处理方式及日常运营。如果测试过程中质量一直较差，将会延长测试周期。

4 测试周期结束后我方会邮件同步测试结果通知商务、并通知厂商。后续由商务推进对接招标、签订合同等后续流程。确定起始计费日期后按需求表正常上量即可。

程序启动参数说明

点播程序

点播程序标准启动命令类似如下，启动后可以看到俩个进程（daemon+worker）：

```
./nexusplugin \
  --provider_id=12 \
  --provider=testProv \
  --guid=test_device_guid:001 \
  --multi_line_speed=12.5 \
  --path=/disk1,/disk2 \
  --nat=0 \
  --storage_total=3000000 \
  --limited_area=0 \
  --log_level=2 \
  --log_path=/disk/log \
  --log_size=100 \
  --nic_out=ppp0,ppp1,ppp2,ppp3,ppp4 \
  --nic_in=ppp0,ppp1,ppp2,ppp3,ppp4 \
  --tcp_conf_path=/disk/etc/ \
  --tcp_nic_port=all:18800 \
  --upnp_port=18000
```

- --provider=testProv 厂商分配简称，各种类型区分见下文“资源要求说明->资源分类”，如--provider=testProv；这里注意专线的有区别加个后缀_zx 对应--provider=testProv_zx。设备类型一定不要填错，不然影响计费。
- --provider_id=12 厂商分配的唯一 id，厂商区分标识将影响计费不要填错。同一厂商不同类型资源 id 有所区别。如汇聚 12，专线+100 为 112；x86 小节点+200 为 212；arm+300 为 312。专线汇聚，小盒子之类的分配方式可参照下面资源类型说明。
- --guid=xxxxx 厂商分配的机器唯一 id 标识，确定上报后不能随便变更，否则跟可能同一台设备当日被当作不同机器最终在线时间不满足触发 sla。且 guid 和缓存绑定，改变后会被识别为新机器产生异常。 guid 具体命名方式见下文。
  - 为便于比较方便直接的识别出某节点厂商，区域，类型等，建议厂商上机传 guid 时尽量遵守以下命名规则。
  - 另外建议统一使用小写(一定不要大小写混用，以免出现两者机器 guid 一样)，支持字母、下划线`_`、冒号`：`、以及数字。

Plain Text

```
guid 命名方式：{厂商 provider}_{区域名拼音}_{运营商}_{资源线路类型}_{厂商自定义设备 id}:{预留编号}
    guid 里只能使用：小写字母，数字，下划线，冒号。

- 厂商 provider：分配给厂商的 provider 名称，
 如 bdy，aly，txy 等
- 区域名拼音：区域省份的汉语拼音(或首字母缩写)：
 如 beijing 北京(bj)，guangdong 广东(gd),shanxi 山西，shaanxi 陕西。
- 运营商及专线汇聚：(isp 对应： ct 电信 cu 联通 cm 移动)
- 资源线路类型：(acdn 代表专线类型的 x86 大节点 x86 代表汇聚类型的 x86 大节点 arm 代表 arm 盒子资源 sx86 代表 x86 小节点[x86 盒子])
- 厂商自定义设备 id:厂商对设备自定义编号以区分，厂商自行灵活配置(只能使用字母数字下划线，建议不要太长串能简单自行标识区分即可)
 如江苏盐城设备厂商可自定义为 yc001，yc888 等；如深圳南山区 2 号机房 ns02_001;自建设备 zj002 等。
- 预留编号：继承旧的命名方式，防止后续仍有单机多实例情况。目前单实例统一使用:000
 如：000，001，002 区分同一台机器上的不同实例。一台设备的 guid 应该是唯一的， 即同一台设备{:预留编号}前应该是一样的，只是为了防止后面有多实例情况预留该字段区分

eg：好的命名
    hwy_guangdong_cu_x86_szft01:000：代表华为云 广东 联通 x86 设备 深圳福田机房 01
    aly_neigmenggu_ct_acdn_eeds_02_032:000： 代表阿里云 内蒙古 电信 acdn 设备 鄂尔多斯 02 号机房 32 号设备
    txy_liaoning_cm_sx86__02_012:000： 代表腾讯云 辽宁 移动 x86 小节点 02 号机房 12 号设备
```

- --nat=0 只接受 nat 为 0 或 1 的机器，按照 stun 协议探测。若厂商启动不传，则程序会自行探测，为避免受防火墙端口或个别线路影响，建议厂商启动时都需要自行探测配置好 nat 参数，否则对上报异常（nat!=0,1）的可能会无差别拉黑。可自行使用 pystun 探测。

- --multi_line_speed=12.5 multi_line_speed: 每根线路的单线最大上行带宽，注意单位 MB/s,支持带小数位.必填。无论是专线还是汇聚，都是传单线路带宽，程序会自行探测设备线路数并计算整机带宽。(注意这里不要虚报，若发现虚报带宽实际不能跑到这么多，会视情况拉黑或者直接清退)
  - 如果每根线路出口 1Gbps，则这里参数值为 1Gbps / 8 = 125,单位 MB；同理 100Mb 的线路则位 100Mb/8=12.5，单位 MB；
  - 设置的值不能偏差太大，如 10MB 的点设置了 100MB，冷启动初始化权重按照 100MB 运行，这种偏差太大的话检测到会直接拉黑清退。
  - 只有 multi_line_speed>=125MB 且公网 ip(其他 nat 类型不算)的才能被视为专线机器，其他如果 nat1 的即便单线 1Gbps 以上也是属于 x86 汇聚，这里会影响到计费需注意上报不要搞错。
- --path=/data1,/data2,/data3 每个磁盘都会下载缓存文件，优先选取剩余存储空间最大的下载，即剩余空间 1T 的磁盘和 2T 的磁盘会优先使用 2T 的磁盘下载。
  - 重点说明：有一些元信息会存储在第一个磁盘路径(比如这里是/data1)下，如 user.ini.enc(节点元信息)，read_test.txt(记录当前实际探测到的线路数)，所以 path 后的路径顺序一定不能随便变更，也不要乱删缓存，不然会当成新机器上报出错。
  - 设备上的 etc、user.ini.enc、缓存文件禁止删除
- --storage_total=100000 本机分配总缓存大小，单位 MB，小于总磁盘空间。建议设置到总磁盘空间的 70-80%左右或者按 io 能力适配，保证 io 质量即可。 在不影响 io 的情况下缓存一般越多越好，种子多了被命中返回调度的概率越大，**支持跨省资源每 G 带宽对应 1.5～2T 缓存（纯跑本省的要求每 G 带宽搭配 3~4T 缓存）**，太低的比如 400MB/Gb~500MB/Gb 这种不符合线上要求的，需要清退处理。
- --tcp_nic_port=all:33520 设置的 tcp 端口(端口自行设定)，保证不跟其他程序冲突。这里暂时统一设为 all,不要填其他网卡名 。
- --tcp_conf_path=./etc/ 条件必选，开启 tcp 后需要指定加载配置路径，如程序使用绝对路径启动，请使用绝对路径。
- --tcp_enable=<mode> TCP 跑量模式：0=禁用 TCP, 1=OnlyIPv4, 2=IPv4+IPv6, 3=OnlyIPv6。默认值为 2，以拨测结果决定是否可跑 TCP。
- --udp_enable=<mode> UDP 跑量模式：0=禁用 UDP, 1=OnlyIPv4。默认 arm 和 x86_small 开启 IPv4 跑量，x86 和 zx_x86 关闭 UDP。需要特殊控制是否开启 UDP 跑量时可指定此参数，否则保持默认即可，不用在启动参数中指定该参数。（该参数与 tcp_enable **不可**同时关闭）
- --dispatch_ip_version=<VER> 请求调度使用的 IP 协议版本，0=不限制, 4=IPv4, 6=IPv6。默认取值和--tcp_enable 保持一致。
- --download_ip_version=<VER> 回源下载使用的 IP 协议版本，0=不限制, 4=IPv4, 6=IPv6。默认取值和--tcp_enable 保持一致。
- --upnp_port=18000 udp 使用起始端口；保证范围内端口不被别的程序使用(跟 tcp_nic_port 端口错开)；如 A 机器有 10 条线路,开始端口为 18000，则端口依次指定为 18000-18010。
- --log_level=2 日志级别：(3:error, 2:warn: 1:info: 0:debug) 支持使用--log_level=2 或--log_level=warn 两种方式设置。
- --log_path=/data1/log/ 日志路径：如果担心系统盘被打爆，可以将日志路径设置在数据盘。

- --log_size=100 日志文件：注意日志文件大小单位是 MB。
- --log_backups=3 日志文件备份个数，默认备份 3 个。
- --nic_out=ppp1,ppp2,ppp3 程序可用于上传的网卡列表。
- --nic_in=ppp1,ppp2,ppp3 程序可用于下载的网卡列表。
- --limited_area 该参数标明资源的区域，参数值包括【0:全国 1:本省 2:大区】。
- --limit_cross_isp 本省资源是否限制跨网请求，此参数只在--limited_area=1 条件下生效【0:否 1:是】
- --isp 该参数指示跨网的目的运营商，参数值包括【1:电信 2:联通 3:移动】，对于跨网设备必须正确地设置该参数值。非跨网设备启动时不带这个参数。
- --province_blacklist=11,12,13 该参数用于显式指定拉黑某些省份跑量的省份列表。
- --province=11 该参数用于指定同省资源跨省调度，必须同时设置--limited_area=1，且不可和--province_blacklist 参数一起使用。
- --resource_code 该参数指定资源分时段资源类型，**需要与快手侧沟通后才能使用**，新的资源类型需要快手侧配置添加。

如果是机房机器且有多台，可以设置以下命令行参数把多个节点组成内网，内网节点可共享磁盘存储（组网时需务必确保内网带宽足够）：

- --intranet_group=<分组名> 建议设置为 <provider-机房名-分组编号> 这种格式，组内的所有节点指定的组名必须完全一致。
- --intranet_nic=<内网网卡> 必须有内网 IPv4 地址的网卡。
- --intranet_port=<内网端口> 内网节点信息同步和数据拉取服务端口，TCP/UDP 都必须可用。
- --intranet_peers=<IP1:Port1,IP2:Port2,...> 罗列出组内的所有节点的内网 IP 和端口，必须包含自己的 IP:Port。

注：程序启动后，会自动在 nexusplugin::worker 进程的启动参数加上 --reboot 参数，表示程序重启的次数。注意这是程序启动后自己生成的参数，程序启动时不要带入此参数。对于 reboot 大于 0 尤其较大时说明一直在重启，应及时检查设备是否有问题。常见一般是 oom 引起，可先自行检查内存与缓存是否配比合理。一般 1TB 缓存建议 8G 内存，不然容易触发 oom。

程序启动时尽量按以上建议的参数配置启动，不要自行更改添加或剔除某参数。

直播程序

```
./nexuslive \
  --log_level=2 \
  --log_path=/disk/log \
  --log_size=100 \
  --provider_id=100 \
  --provider=testProv \
  --guid=test_device_guid:001 \
  --path=/disk1 \
```

```
--multi_line_speed=12.5 \
--max_line_download_speed=12.5 \
--nat=0 \
--nic_out=ppp0,ppp1,ppp2,ppp3,ppp4 \
--nic_in=ppp0,ppp1,ppp2,ppp3,ppp4 \
--tcp_conf_path=/disk/etc/ \
--tcp_nic_port=all:18800
```

直播业务属于新上线的业务场景，独立于点播程序，拥有单独的一套部署程序。目前因总放量较少，一般都是会按大区测试放量同步进行。

由于直播业务对设备网络要求及稳定性比较高，目前不能随便自助上机，还是人工管理偏多，厂商有要跑直播业务的提供节点后必须保证节点稳定，不能随便变更及上下线。每次有新的直播放量需求会单独通知到对应厂商在某区域部署多少量级。 直播程序目前使用单独一套程序版本，部署时需区分开点播程序。启动参数设置**严格按照**上述启动示例进行。

直播与点播设置存在差异的参数，具体解释如下。其余参数含义参考点播参数解释部分。

- --max_line_download_speed 直播设备除了标注设备上行能力外，还需标明设备的单线下行能力（单位：MB/s，和单线上行能力 multi_line_speed 类似）.该参数仅限直播场景部署时使用。
- --path 缓存路径。直播中只需要设置一个路径即可，如：--path=/data_live。

注意事项：

- 注意单台机器上只能跑一个任务，**一定不能**同时跑点播和直播 2 套程序。
- 直播业务比较敏感，注意一定不要随意重启变更，有特殊情况要下(停)机必须提前同步，这边先拉黑踢流量后再操作下机。
- 注意同一台机器在点播直播来回切换的时候， 直播的路径一定要和点播彻底区分开（否则两者的缓存信息在同一个盘下会冲突覆盖，同一台机器点播和直播切换程序时，会覆盖彼此元数据信息导致有可能不跑量）

设备系统环境配置

系统参数

- 开启 bbr，尽量使用较新内核版本如 4.9 以后版本，以支持 bbr 算法。

Plain Text
```
  echo "net.core.default_qdisc=fq" >> /etc/sysctl.conf
  echo "net.ipv4.tcp_congestion_control=bbr" >> /etc/sysctl.conf
```

保存生效：`sysctl -p` 执行：`sysctl net.ipv4.tcp_available_congestion_control` 如果结果中有`bbr`, 则证明你的内核已开启 bbr。
`net.ipv4.tcp_available_congestion_control = reno bbr cubic`

- 文件描述符打开数目限制： 目前部分机器因连接数过多打开文件到达上限，需调大文件打开上限。可用 ulimit -n 查看，建议设置到 **1024000** 以上。
- nf_conntrack : 该参数容易引发“ nf_conntrack:table full，drop packet”类似错误，主要是 iptables 的连接跟踪状态表满了导致丢弃部分包。
- core： 主要是程序崩溃时便于调试，可用 ulimit -c 查看若为 0 则表示未开启 core，可改为 unlimit -c unlimited 不限制但需注意 core 文件一般较大容易占满磁盘。
- 若使用 docker 启动，务必在容器内装好一些基本调试查看工具，便于出问题时排查。

除上面这些外，一般常推荐的系统内核参数可以参考如下：

```
Plain Text
#!/bin/sh

sysctl -w net.core.somaxconn=10000
sysctl -w net.ipv4.tcp_max_syn_backlog=100000
sysctl -w net.ipv4.tcp_slow_start_after_idle=0
sysctl -w net.core.wmem_default=327680
sysctl -w net.core.wmem_max=3276800
sysctl -w net.core.rmem_default=327680
sysctl -w net.core.rmem_max=327680

sysctl -w net.ipv4.tcp_retries2=5
sysctl -w net.ipv4.tcp_no_metrics_save=1
sysctl -w net.ipv4.tcp_no_delay_ack=1
sysctl -w net.ipv4.tcp_wmem="4096 262144 524288"
sysctl -w net.ipv4.tcp_rmem="4096 524288 524288"

sysctl -w net.ipv4.tcp_tw_recycle=0

ip route change `ip r | grep default` initcwnd 44

ip tcp_metric flush

sysctl -w vm.dirty background ratio=1

#echo 100000 >  /proc/sys/fs/aio-max-nr

#建议关闭，如果一定要开，建议设大
sysctl -w net.netfilter.nf_conntrack_max=25000000
sysctl -w net.netfilter.nf_conntrack_buckets=3000000
```

常用工具软件

目前一些机器有的使用 docker，有的直接使用二进制物理机直接启动。(优先推荐物理机启动，不推荐跑 docker & 云主机)，即便跑 docker，也需保证单机只跑一个 docker 实例。**各个厂商的机器镜像，系统版本及预装工具不一，导致平时登陆命令排查问题很不方便**。请务必在机器上统一安装以下一些常用排查工具或命令，尤其是 docker 封装的机器很多基础命令没有，查问题很不方便。

vnstat: 可实时动态观测网卡上下行带宽. 安装方式：yum install vnstat -y

htop: 进程查看工具： 安装方式： yum install -y htop

pystun：可用来探测网络 nat 类型：安装方式：yum install python-pip; pip install pytun

Perf: 安装方式：yum install perf -y

若使用 docker 镜像的，docker 里尽量打包好一些基础的常用命令及网络工具。nc,telnet,tcpdump 等

资源要求说明

资源分类

- ●x86 大节点
  - ○ x86 汇聚：单线线路带宽>= 100Mbps, 分配使用 --proverder=ksy --provider_id=12
  - ○ acdn ： 单线线路带宽大于 1Gbps 且为公网 ip, (id+100,加个`_zx`), 分配使用 --proverder=ksy_zx --provider_id=112
- ●盒子类资源：
  - ○ x86 小节点： 单线线路带宽>=30Mb, (id+200)分配使用 --provider=ksy --provider_id=212
  - ○ arm: 单线线路带宽>=20Mb, (id+300)分配使用 --proverder=ksy --provider_id=312

以厂商 ksy 为例，假设分配基础 ID 为 12，启动时厂商需按分配的基础 id(+100,+200,+300),自行根据节点类型指定启动参数--provider 和--provider_id； 假设部署 1.1.80.719 版本，对应 provider，provider_id 启动参数如下 (程序会自动补齐架构[x86 还是 arm]及版本号）：

provider(启动时配置)	provider_id(启动时配置)	version(程序自动生成)	节点类型	节点允许单线带宽范围（不符合时启动失败）
ksy	12	ksy_x86_1.1.80.719	x86 大节点汇聚	100Mbps<=bw<=25Gbps
ksy_zx	112	ksy_zx_x86_1.1.80.719	x86 大节点专线 acdn	1000Mbps<=bw<=25Gbps
ksy	212	ksy_x86_small_1.1.80.719	x86 小节点	30Mbps<=bw<=25Gbps
ksy	312	ksy_arm_1.1.80.719	arm 盒子	20Mbps<=bw<=1Gbps

版本发布

● 变更窗口期问题

- ○ 变更的时间窗口期：工作日（一般为周一至周四）10:00 -11:00 14:00-17:00，若遇节假日有特殊调整会另行通知；无特殊说明情况下只能在规定时间段内变更 ，无论是上下机、重启、版本变更等一切线上操作。
- ○ 特殊情况需要提前报备。必须在非窗口期内操作的，需提前同步确认后进行，未报备的会纳入 SLA 考核。

● 版本升级问题

- ○ 版本升级需遵循灰度流程，无特殊情况下，新版本 2 周内全量，历史长尾版本 1 个月内完成下线；
  - ■ 版本变更按照 1%、5%、10%、20%、50%、100%节奏发布，有问题及时反馈，严禁无灰度批量操作；
  - ■ 如无特殊说明，灰度到 10%量级至少需要跑 1 个晚高峰，灰度到 50%至少跑一个周末，版本全量需在 2 周内完成；
  - ■ 每次新版本升级后我们会规划一定实际内停掉旧版本的跑量(不每次另行通知)

● 其他说明：

- ○ 严禁无报备随便删除机器缓存 ，包括磁盘缓存数据，程序配置数据等，
- ○ 做好版本备份，至少保留历史 5 个版本，必要时需要做版本回滚；

正常每次发版提供程序时也会提供对应多个程序版本，主要包含以下 6 个程序包：

- x86 大节点
  - ○ 32 位
  - ○ 64 位
- 盒子
  - ○ x86 小节点 32 位
  - ○ x86 小节点 32 位
  - ○ arm 32 位
  - ○ arm 64 位
  - ○ android 32 位
  - ○ android 64 位
  - ○ mips 32 位

厂商自行根据自己机器所属架构选用版本部署即可：注意不同类型的设备程序一定不要部署其他版本，否则会影响计费数据及结果需自行承担。

资源约束(重要)

以下要求适应所有类型资源：

- 机器网络要求：招收 nat=0，1，2，3。如果 nat=4，需要有 ipv6 地址（专线资源要求必须是公网/全锥）
- 所有机器必须在物理机部署交付(可使用容器化部署)，不能使用虚拟机 vm 设备。
- 带宽大小：见上资源类型<节点允许单线带宽范围>，不能低于最低要求。
- 只接受三大运营商(电信联通移动)网络，其他（城际网，网通，铁通等）一律暂不接收。
- 必须是服务器直接拨号，不接收软路由。
- 单机只能部署单实例，即使是汇聚多线的设备也是只部署单个实例，程序会自动识别所有线路，不能每条线对应部署一个实例，否则会引发异常。

ps: *支持跨省资源要求每 G 带宽至少应配备 1.5T 缓存，**本省资源每 1Gbps 带宽需要 3TB 以上存储**，否则跑量会有影响（理论上越多越好，但缓存多了也可能增大 io 压力）；同时每 1T 缓存须对应有 8g 以上内存，不然容易触发 oom 或者 coredump；*

推荐硬件配置

ps: *目前以下机器配置是基于以往线上主流机器硬件推荐值，不作为强制上机要求。但在因硬件低于推荐配置导致出现指标问题、或者资源冗特别富余的情况下，我们会优先清退处理掉这批设备*。

- x86 大节点
  - 专线

单 IP 带宽	IP 地址	端口	CPU	内存	网卡	运营商	硬盘	IO	root 权限	操作系统
1Gbps~2Gbps	公网静态 IP 公网拨号 IP	指定	32 核	32GB	万兆	电信/联通/移动	SSD 盘 2 块，总容量≥ 2T	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机
2Gbps~3Gbps	公网	指	32 核	32GB	万兆	电信	SSD 盘 2-4	4K 随机读写	必须提	指定内核版本

	静态 IP 公网拨号 IP	定				/联通/移动	块；总容量≥3T	跑满网卡单次 IO 延时小于 5ms	供	centos，不支持虚拟机
3Gbps~4Gbps	公网静态 IP	指定	48 核	64GB	万兆	电信/联通/移动	SSD 盘 4-6 块；总容量≥5T	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机
4Gbps~5Gbps	公网静态 IP	指定	48 核	64GB	万兆	电信/联通/移动	NVME 盘 4 块；总容量≥6T	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机
5Gbps~6Gbps	公网静态 IP	指定	64 核	64GB	万兆	电信/联通/移动	NVME 盘 4 块；总容量≥6T	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机
10Gbps 以上	公网静态 IP	指定	64 核	96GB	≥10G	电信/联通/移动	NVME 盘 4-6 块；总容量≥15T	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机

- 汇聚

单线带宽	NAT 类型	端口	CPU	内存	汇聚线路数	网卡	运营商	硬盘	IO	root 权限	操作系统	补充说明
100Mbps	公网全锥	指定	32 核	64GB	大于 10 条小于	万兆	电信/联通/移动	SSD 或 NVME 盘，1G 带宽对应 1.5T 存储，	4K 随机读写跑满网卡单次 IO 延时	必须提供	指定内核版本 centos，不支持虚拟机	汇聚设备优先 4-5G，单线带宽以晚高峰压测

					50 条			下同	小于 5ms			&tcp 重传率小于 5%为准
200Mbps	公网全锥	指定	32 核	64GB	大于 10 条小于 30 条	万兆	电信/联通/移动	SSD 或 NVME 盘，1G 带宽对应 1.5T 存储，下同	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机	汇聚设备优先 4-5G，单线带宽以晚高峰压测 &tcp 重传率小于 5%为准
500Mbps	公网全锥	指定	48 核	64GB	大于 2 条小于 10 条	万兆	电信/联通/移动	SSD 或 NVME 盘，1G 带宽对应 1.5T 存储，下同	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机	汇聚设备优先 4-5G，单线带宽以晚高峰压测 &tcp 重传率小于 5%为准
其他	公网全锥	指定	48 核	64GB	小于 50 条	万兆	电信/联通/移动	SSD 或 NVME 盘，1G 带宽对应 1.5T 存储	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	指定内核版本 centos，不支持虚拟机	汇聚设备优先 4-5G，单线带宽以晚高峰压测 &tcp 重传率小于 5%为准

- 盒子
  - x86 小节点
  - arm 盒子

盒子类资源硬件设备配置建议：(Ps:cpu 内存为单线标准，线路增加需对应比例扩增硬件)

类型	单线最小带宽	NAT 类型	端口	CPU	最低内存	汇聚线路数	运营商	硬盘	IO	root 权限	操作系统	补充说明

ARM	20Mbps	公网/全锥	指定	2 核	1Gb	单线	电信/联通/移动	SSD 或 NVME 盘，1G 带宽对应 1.5T 存储，下同	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	linux 内核版本 4.10 以上，不支持虚拟机	重点关注 cpu,io 和 tcp 重传率小于 5%为准
X86	30Mbps	公网/全锥	指定	2 核	1GB	小于 50 条	电信/联通/移动	SSD 或 NVME 盘，1G 带宽对应 1.5T 存储，	4K 随机读写跑满网卡单次 IO 延时小于 5ms	必须提供	linux 内核版本 4.10 以上，不支持虚拟机	重点关注 cpu,io 和 tcp 重传率小于

省份运营商代码

Plain Text

```
运营商：

isp = 0  '未知'

isp = 1  '电信'

isp = 2  '联通'

isp = 3  '移动'


省份代码及大区划分

id     省份 归属大区

11     北京 华北

12     天津 华北

13     河北 华北

14     山西 华北

15     内蒙古 华北

21     辽宁 东北

22     吉林 东北

23     黑龙江 东北

31     上海 华东

32     江苏 华东

33     浙江 华东
```

```
34    安徽 华东
35    福建 华东
36    江西 华东
37    山东 华东
41    河南 华中
42    湖北 华中
43    湖南 华中
44    广东 华南
45    广西 华南
46    海南 华南
50    重庆 西南
51    四川 西南
52    贵州 西南
53    云南 西南
54    西藏 西南
61    陕西 西北
62    甘肃 西北
63    青海 西北
64    宁夏 西北
65    新疆 西北
```

日常运营说明(重要)

- 日常巡检

  尽量按照我方提供的指标接口数据对机器进行日常巡检，目前需多关注的可能影响跑量的指标主要涉及：

  - nat 类型：目前主要支持公网和全锥型机器，否则可能因为打洞不通而不跑量；
  - 运营商：只支持三大运营商，其他的识别有问题的自行处理下掉；不要一直留着不管；
  - 硬件方面：cpu 使用率，cpu_iowait，cpu 软中断占比；io 延时。
  - 网络方面：tcp 线路传输速度(指线路的实际传输速率，而不是指该线路上限带宽)，tcp 重传等。

以上可自行在每日的数据接口(主要是设备指标接口，拉黑设备接口)里获取，对于比较明显的差指标需尽快整改。否则将加大压制幅度或完全拉黑，并在本月的质量评分中获取低分。

- 设备带宽限制问题

- 正常情况下往一台机器调度请求，当出现指标问题时就会开始限制请求继续调度到该机器。 这种被压制(程度相对严重)的机器每天会统计进入限制列表(每日 blacklist 接口可自行拉取)，比如 5g 的机器跑到 2g 就出现重传问题，则这台机器实际会被限制按 2g 跑。正常限制机器 1-2 天后会自动探测恢复，若放开后仍有问题则继续限制，如此循环。 若设备长期(3~5 天不定视资源冗余稀缺程度)指标质量差会考虑直接永久拉黑该机器清退。

●缓存操作问题：

- 注意当前机器缓存不要随便删除。设备缓存中信息 绑定了机器初始信息，包括：guid、运营商、省份、厂商、设备类型(如专线、汇聚，盒子)等。故以上信息如果变了，就会跟缓存不匹配导致设备跑量异常。 故一台机器上机后 不应随便操作变更这些基础信息。特殊情况变更了导致不能跑量的可每周二通过提交白名单过来。注意只针对库存机器，对于用新机器冒认白名单机器的行为将记录违规记录。

●变更窗口期

- 变更的时间窗口期：工作日（一般为周一至周五）10:00 -11:00 14:00-17:00(若遇节假日有特殊调整会另行通知)，没有特殊情况只能在这个规定时间段内变更 ，无论是上下机重启等一切线上操作皆适用。特殊情况需要提前说明。必须在非窗口期内操作的，需提前同步确认后进行，这边同时会将非窗口期的操作频率记录进稳定性考核。

● 版本升级问题

- 每次新版本升级后我们会规划一定实际内停掉旧版本的跑量(不每次另行通知)，后续每次给的新版本需按要求在规定时间内及时升级完(注意灰度节奏逐渐扩大，不要集中在某一时间点内大批量升级，若有问题及时反馈)，否则由于未及时升级引起不跑量自行负责；

●厂商评分管理

- 请严格遵守日常运营操作规范，我们会根据各家厂商的平均成本、资源情况（冗余/稀缺区域资源占比，拉黑设备平均占比，质量情况等）、资源稳定性和日常服务配合支持度等几项指标进行综合评估打分，并作出一定奖惩措施。

sla 触发条件

午晚高峰时间 11:00-14:00 17:30-24:00

Plain Text

> 高峰时段在线率：　　午、晚高峰时段，设备正常服务时长单时间段内离线累计不超过 15min，各时间段单独计算。

> 重传率：　午晚高峰时段每 1min 取一个点计算重传率，由高到低排列取第 31 个点作为该时间段重传率（认为有超过半小时的时间重传偏高），各时间段单独计算。

> io 高负载　午晚高峰时段 io 高负载时长占比， IO 读延时大于 10ms 或 IO 写延时大于 100ms 定义为高负载，高负载出现时间超过 15min 认为触发当日 sla。各时间段单独计算。

> cpu 高负载　午晚高峰时段 cpu 高负载时长占比，CPU 利用率超 90%定义为高负载，高负载出现时间超过 15min 认为触发当日 sla。各时间段单独计算。

> 线路频繁离线（针对汇聚） 每天午晚高峰时段，每 1min 取一个点，线路离线大于 5 或超过总线路数 1/3 的次数超过 60 个点，认为当日线路离线频繁。各时间段单独计算。

> 在线时长 每日机器在线时长,离线时间不超过 12h.统计整日 24h。

> 压测带宽 不定期进行压测，压测带宽与当时实际上报带宽进行对比，若有差距则认为当日虚报带宽.

> 资源调整 资源新增/下线超过 50g 时需提前沟通告知，否则认为是擅自变更资源。每月以上月末资源为基准，本月每日资源取均值，月末资源变更占比

> 内容安全 1、乙方保证不对所提供的互联网出口带宽流量进行任何劫持、修改，对流量内容不得进行任修改、添加或者安插广告；2、乙方保证不对设备进行超卖，包括但不限如混跑，超额部署实例等

> 服务响应 "1、乙方应有 7*24 小时联系方式和响应，提供电话，邮箱等方式"

> 质量指标反馈 我方会提供接口供查询指标异常的节点，经双方核实若乙方无明确反对，则需尽快 3 天内处理或下线该节点

常见影响跑量指标场景

Plain Text

（以下为常见指标因素,非所有。 对于该类指标差的设备会对应减少调度，故需自行按接口文档及时修复问题机器）

tcp_retran_limit tcp 重传限制

cpu_limit cpu 负载高，排查硬件 cpu 好坏，系统是否其他任务拖垮系统负载，或者检查是否未设置均衡软中断等。

mix_run 混跑，当网卡带宽比业务层超出一定时，怀疑是其他业务等因素往网卡堆带宽导致远高于业务层带宽。 多网卡设备也可能是由于未正确配置 v4/v6 策略路由导致。

app_bigger_than_net_card 业务带宽高于网卡带宽 20%，可能是网卡不好，或者被限速导致业务包堆积不能及时发出去。

io_limit io 延时 ，程序自行计算的从收到请求到读取到请求文件的时间，io 延时大，不一定完全是磁盘 io 影响，中间读取过程中也可能受 cpu 是否忙碌等影响。

tcp_speed_limit tcp 包传输速度，为业务程序自行计算的单位时间内传输的流量，与上报带宽值没有关系。该指标是全局设备统一标准，各设备网络环境有差别引发原因不一，这个指标差的话估计就是哪里网络有问题或者限制，需自行检测。

machine_band_limit 整机带宽限制

数据接口

目前接口 api 统一前缀为 `{$domain}/provider/${厂商 dbid}/${date}`,后面加各类信息路径。每个厂商会单独分配一个 dbid（非 provider_id），假设有厂商 ksy，分配 dbid： ksy123456789，则：

`curl -O {$domain}/provider/ksy123456789/20211008/sla/ksy_20211008_sla.txt`

若返回文档 txt 读取到 null，则表示未筛选到相关设备。

目前获取该接口需厂商测提供 ip 白名单(2 个固定 ip 或网段)，只有在白名单内的 ip 才可拉取对应自己 dbid 下接口数据。

sla 信息

Plain Text

每日 sla 统计数据获取： curl {$domain}/provider/ksy123456789/{日期}/sla/{厂商}_{日

每日设备清单

Plain Text

每日机器列表状态及调度压制信息数据：

curl {$domain}/provider/{厂商 dbid}/{日期}/allNodeList/{厂商简称}_{日期}_nodeList.txt

如：

curl {$domain}/provider/ksy123456789/20211008/allNodeList/ksy_20211008_nodeList.txt
{$domain}/provider

txt 文件列依次表示：

guid
uid
version

运营商

大区

区域

在线线路

离线线路

上报带宽(Gb)

实际在线带宽(Gb)　去除离线线路后实际带宽

在线时间(min) 记录每个设备当日的在线时长，标准 1440min。

设备异常指标每日巡检

Ps：正常情况下，这里的机器都是严格意义上有某项指标问题或者不太符合我们要求的设备；故越少越好。

- 日常指标巡检：
- 新指标合到一个文件里边，各项指标均有参考线上平均水平及给出一定范围容错空间。仍有异常的已经是远超出线上设备平均水平，目标应保证 excel 里第 6 列之后指标尽可能都无异常。

Plain Text

质量指标获取地址：

```
curl {$domain}/provider/厂商 db_id/年月日时间/时间段/文件 id
eg:
 curl {$domain}/provider/ksy123456789/20211122/night/allIndex.txt
 curl {$domain}/provider/ksy123456789/20211122/night/allIndex.csv
```

该指标提供 txt 和 csv 俩种格式下载。

说明：

1 该接口每天会过滤午晚高峰时间段内一些重要指标不理想的设备，请厂商每天巡检。中午 15 点（凌晨 2 点）后可获取前一个高峰期时间段内的指标数据。每个数据获取一次，不要频繁下载！！

2 厂商 db_id：单独分配

3 目前含以下指标，后续会视实际情况增减；当前各指标标准阈值是根据线上大致平均值计算取得，已考虑一定容错区间。仍出现异常的认为是质量异常设备，会不定时被拉黑限制。

数据格式：

各列依次为：guid uid isp region version reportOffLine coredump cachePerG cpuUsage cpuSoftIrq cpuIoWait hotPushTotalFiles ispIdentifyError natIdentifyError err500 tcpRetran avgLineTcpSpeed tcpstatus multiLineTcpRetran。

guid：自分配，必须全局唯一，不区分大小写，勿大小写混用；同一机器不能随便更改，否则易出发 sla 离线；

uid：程序分配唯一 id，生成后在--path 下第一个磁盘位置下生成，不能随便删除或更换磁盘位置；

isp：运营商 仅限三大运营商；

region：国内省市代码(身份证首 2 位)，个别不能识别的自行调整或停掉；

version：统计时该设备的版本。除新发布要求灰度的版本外，线上需保持所有设备版本尽量到最新稳定版；

reportOffLine：通过网卡上报失败次数，一般表示网卡能检测到，但是未能获取到 ip 或者通过该线路网络上报异常，这种一般导致该线路离线完全不跑量，需及时检查相关线路联通性及网络是否正常。否则直接影响跑量且长期出现会视情况拉低调度优先级。

coredump：可先检查是否是 内存，cpu，磁盘等瓶颈导致程序崩溃。否则若程序仍频繁出现 core 时可及时反馈；

cachePerG:缓存带宽比（每 Gb 带宽对应下载缓存），同等情况下缓存越多有助于提高返点命中率提升跑量，除新上机器下载缓存中外，正常机器应稳定在 1000G/1Gb 左右或以上。5g 以下带宽不得少于 800GB/1Gb 带宽，否则会降低调度优先级。

cpuUsage：cpu 使用率，高峰期取按高到低排序取第 16 个点（给予 15min 容错防止突刺等），高于 80%的需检查 cpu 是否不足。否则会降低优先级或直接拉黑。

cpuSoftIrq：cpu 软中断，高峰期取按高到低排序取第 16 个点（给予 15min 容错防止突刺等），高于 20%的需检查是否软中断是否均衡，或者检查网卡队列等是否符合要求。否则会降低优先级或直接拉黑。

cpuIoWait:cpu 等待 io 占比，高峰期取按高到低排序取第 16 个点（给予 15min 容错防止突刺等），高于 20%的需检查是否软中断是否均衡，或者检查网卡队列等是否符合要求。否则会降低优先级或直接拉黑。

ioDelay:io 延时，注意这里不是指磁盘 io 的延时，而是程序自行计算的从收到请求到读取到请求文件首片的时间，io 延时大，不一定完全是磁盘 io 影响，中间读取过程中也可能受 cpu 是否忙碌等影响。

appGtNicBw：业务带宽大于网卡带宽的次数，这里已经考虑一定容忍度，还是出现在这里的已经是比较明显的问题了。很可能是网卡或者设备限速了导致出现次问题。

hotPushTotalFiles:高峰期推送文件，程序会自动取拉取热点文件，必须保证高峰期可正常拉取文件数且不能数量过少(500 以下认为过少，可

能存在限制磁盘缓存等作弊行为），否则认为服务异常设备，可降低调度优先级或直接拉黑；

ispIdentifyError：只限三大运营商，以外的线路（如一些城际网通，铁通等）此类机器应及时直接下掉。此类机器会直接影响调度跑量或者被拉黑。

natIdentifyError：nat 类型识别异常；若程序启动未设置 nat 类型则程序会自行探测（建议都自行探测好传入），否则受开放端口网络等影响可能会探测异常。目前线上支持 nat0，1，2，3.专线（acdn）机器必须全部为 nat0 和 nat1。

err500：data 请求 500 错误率，设备异常比例：单位：百分比，如 1%表示该高峰期内有一段时间，5%的请求全部异常，正常控制在 1%以内（越低越好）。可能是某些指标异常比如带宽、CPU、内存超限（负载），客户端为保证已有带宽运行，处理不了接下来的请求就会拒绝掉。此类机器会浪费有效请求会降低调度优先级或直接拉黑。

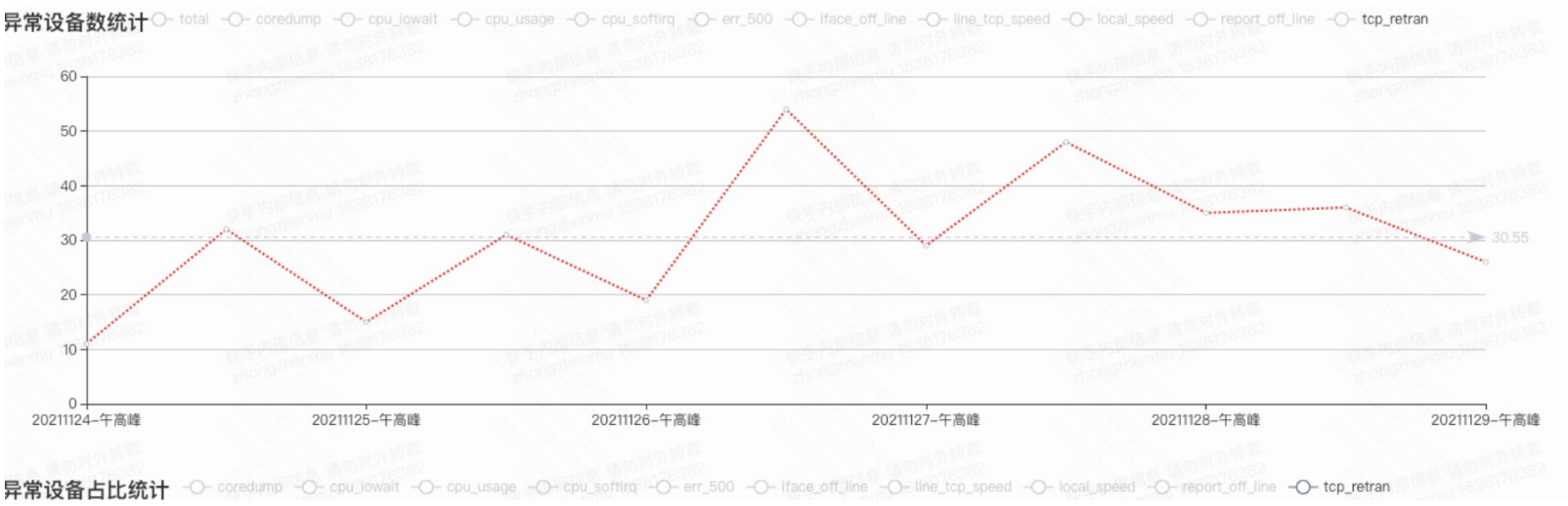
avgLineTcpSpeed:指从收到请求建立 tcp 链接到最后断开 期间发送的数据量/发送消耗的时间的计算得到的线路 tcp 平均速度，目前全局平均 50Mbps 以上，越高越好。低于 15 的会直接影响跑量，或者被调度压制拉黑

tcpRetran：机器纬度重传率，高峰期取按高到低排序取第 21 个点（给予 20min 容错防止突刺等）>5%的认为网络质量较差，会被调度压制或者被拉黑。

tcpstatus：指该机器的拨测状态，一般该项出现异常的话就表示拨测异常。很可能是机器网络出了啥问题导致后端服务器到该节点 tcp 拨测失败。这种情况下基本不会跑量。可自行检查下是否网络策略是否被限制或者防火墙等是否拦截。

multiLineTcpRetran：会显示单线重传高于 5%的线路，计算规则类似上面重传，但该指标不是整机维度而是程序基于每条线路计算 socket 实际传输数据得出。对于个别机器整机重传不高，但其实是某些线重传高 被重传低的线路拉低摊平了。这类会直接影响该条线路的调度导致整机跑量。

目前各项指标并非最低标准值，后续会根据线上实际情况调整，理论上出现在表格里的设备都是在某些方面指标不合格的设备，应及时处理，不要连续或长期出现，这边也会每日巡检各项指标异常数及占比。



供厂商加入每日巡检，每日及时把表格中右图区域异常指标设备尽量清理干净。

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
date	period	guid	uid	region	isp	version	reportOffLir	coredump	cachePerG	cpuUsage	cpuSoftIrq	cpuIoWait	hotPushTot	ispIdentifyE	natIdentifyE	err500	tcpRetran	avgLineTcp	tcpstatus	multiLineTcpRetran	
20220704	night	ianjin_	###	12	2	mal	-	-	-	-	-	-	-	-	-	-	8.37	-	-	-	
20220704	night	anjin_	###	12	2	mal	-	-	-	-	-	-	-	-	-	-	-	9	-	-	
20220704	night	angxi_	###	36	1	l	-	-	-	-	-	-	-	-	-	-	-	8.5	-	-	
20220704	night	uizho	###	52	3	l	-	-	-	-	-	-	-	-	-	-	6.11	4.5	0	-	
20220704	night	uang	###	44	1	l	-	-	-	-	-	-	-	-	-	-	-	12.1	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	58	-	-	-	-	5.63	-	-	-	
20220704	night	uang	###	0	0	mal	-	1	-	-	22	-	-	-	-	-	-	13.9	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	24	-	-	-	-	-	3.3	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	23	-	-	-	-	-	-	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	-	-	-	-	-	-	6.9	-	-	
20220704	night	ngxi_	###	36	1	mal	-	-	-	-	33	-	-	-	-	-	-	-	-	-	
20220704	night	uang	###	0	0	mal	-	117	-	-	-	25	314	未知	-	-	-	0	0	-	
20220704	night	eilong	###	23	1	mal	-	-	-	-	-	-	-	-	-	-	-	7.1	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	-	-	-	-	-	-	10.2	-	-	
20220704	night	anjin_	###	12	2	l	-	-	-	-	-	-	-	-	-	-	-	0	0	-	
20220704	night	ebei_(	###	13	3	l	-	-	-	-	-	-	-	-	-	-	-	9.6	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	-	-	-	-	-	-	0	0	-	
20220704	night	uang	###	0	0	mal	-	13	-	-	-	-	-	-	-	-	-	-	-	-	
20220704	night	uizho	###	52	3	l	-	-	-	-	-	-	-	-	-	-	-	3.5	-	-	
20220704	night	eijing	###	11	2	l	-	-	-	-	-	-	-	-	-	-	-	-	-	wan1_ad:5.6	
20220704	night	uang	###	44	3	l	-	-	-	-	-	-	-	-	-	-	-	4.8	-	-	
20220704	night	angxi_	###	36	1	mal	-	-	-	-	-	24	-	-	-	-	-	-	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	-	-	-	-	-	-	13.1	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	37	-	-	-	-	-	-	-	-	
20220704	night	uang	###	44	1	mal	-	-	-	-	-	-	-	-	-	-	-	13.5	-	-	
20220704	night	uang	###	44	1	l	-	-	-	-	-	-	-	-	-	-	-	13.3	-	-	
20220704	night	hangh	###	31	1	l	-	-	-	-	-	47	-	-	-	-	-	-	-	-	
20220704	night	uang	###	44	1	l	-	-	-	-	-	21	-	-	-	-	-	-	-	-	
20220704	night	uang	###	44	1	l	-	-	-	-	-	-	-	-	-	-	-	1.8	-	-	
20220704	night	uang	###	0	0	l	-	1	-	-	-	-	-	-	-	-	-	12.5	-	-	
20220704	night	uang	###	0	0	mal	-	1	-	-	-	-	-	-	-	-	-	14.2	-	-	
20220704	night	uang	###	0	0	mal	-	17	-	-	-	30	-	-	-	-	-	-	-	-	
20220704	night	ebei_t	###	0	0	mal	-	27	-	-	-	37	-	-	-	-	-	-	-	-	

压测实际带宽限制列表

Plain Text

该名单中的机器主要是指我们压测实际跑量过程中，设备因为某些原因或指标不好，认为当前设备能实际提供的带宽能力不足以达到上报带宽，这时会取该设备按跑量临界点（即刚好出问题时刻设备能达到的峰值带宽）的带宽来调度跑量。若为 0 则表示该机器质量极差几乎完全不能使用。

每日上午 10 点以后可通过以下接口获取参考带宽名单机器数据：

curl {$domain}/provider/{厂商 dbid}/{日期}/limitNodeList/limitNodeList.txt

如：curl {$domain}/provider/ksy123456789/20211008/limitNodeList/limitNodeList.txt

txt 文件列依次表示：

日期

guid

version 版本

线路数

单线带宽(Mb)

上报带宽(Gb)

跑量临界值(Gb)

跑不上去原因： 一般是随着跑量上升到某个点时因某个质量指标问题触发了调度压制(参照文档压制因子说明)，或者指标数据监测到多天出现同一问题导致跑不上去但一直未及时处理。这里提供影响跑量的主要因素，表示该设备必定存在该问题且影响了调度导致跑量上不去。

ps：

- 对于 acdn 设备，即单线 1g 以上设备但实际压测出来参考带宽达不到 1g 专线要求的，属于 x86 汇聚设备虚报成 acdn，这类设备需及时整改，否则后续会对该节点完全限制不跑。
- 限制时长看设备每日指标严重程度或持续时间综合考虑一般 3 天以上，每隔天会试放开限制探测是否已恢复（若恢复正常会自行放开），虚报总量较大且持续出现时会同步商务处理。

Ps:日常的设备排查过程中，nodeList 接口主要是作为机器在线列表信息，质量问题不用参考这里的 xx_limit。重点是处理解决 limitNodeList，和 allIndex.txt 接口里的设备。

需求缺口表

```
Plain Text
为防止某些区域资源变化较大，不能及时同步。该接口目前正常 2～4 周更新一次，一般视当月实际波动情况在月初或月中会更新生成一份新的需求表，会附日期说明。
- 值大于 0 的表示有缺口可以上机，值为 0 的为过渡区域已相对冗余不建议大批量上新机； 为 N 的表示禁止上机区域（上机完全不跑）。
- 上机量一定不能超过区域缺口值；单次超 50g 以上必须提前报备确认；
- 若违背以上不保证利用率、且视为违规上机在影响其他设备时我们会直接对其批量拉黑清退；
每周一晚上可通过以下接口获取需求表：
curl {$domain}/provider/{厂商 dbid}/public/resgap.txt
如：curl {$domain}/provider/ksy123456789/public/resgap.txt
```

常见问题

低跑量排查

1 一般程序启动后即便没多少需求多少都会有一些流量，若几乎接近零跑量。

一般是本地环境患者程序启动，网络是否有问题。建议先排查机器本身，或者如果是前短时间一直质量不好，也可能是被拉黑。所以需关注日常跑量指标。

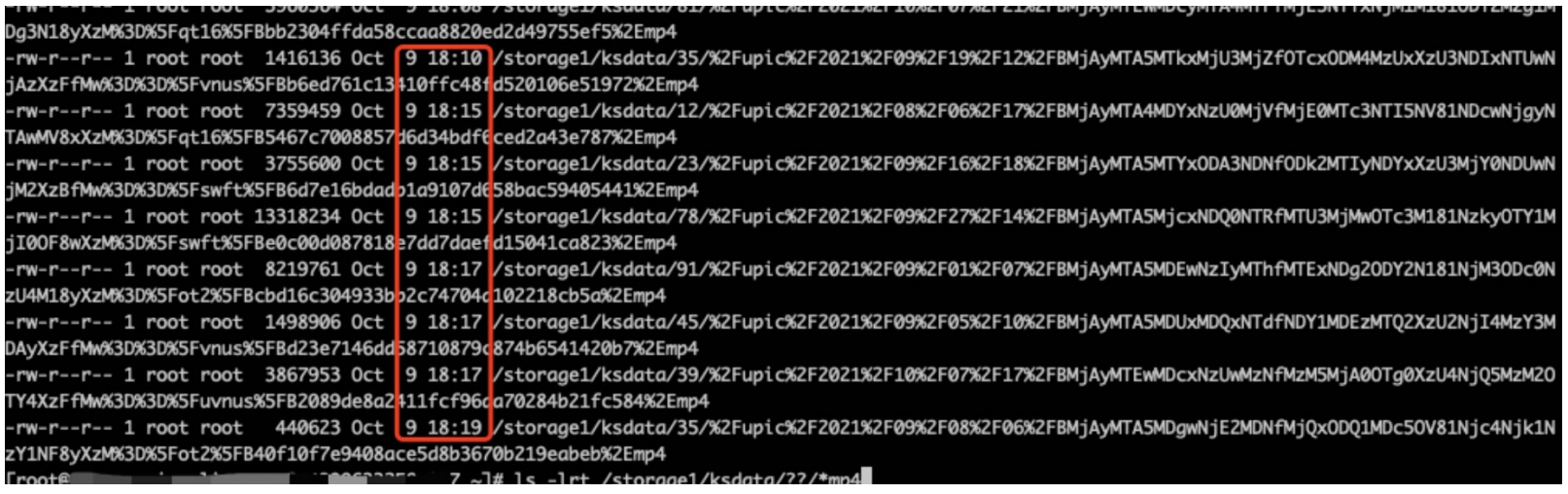
2 平时一般周末或节假日相对于工作日跑量会偏高，平时在一定范围内有回降的话属正常现象。

3 目前调度一般是专线优于汇聚，本省优于跨省，基本不会跨运营商调度。故同一个省内不同机器调度跑量可能也会有些偏差。

4 可继续查询机器列表接口，里边含降权原因说明。若无明显说明，则可能是机器需求或者调度问题，可进一步追踪反馈。

观测程序是否正常启动

- ps -ef|grep nexusplugin 观测进程是否正常启动：正常单机启动后为一个 daemon 进程和一个 worker 进程
- 观测 path 下的缓存是否有正常更新，过滤下过去几 min 内是否有新的文件下载判断。如下假设存储路径为/storage1/ksdata/下，` ls -lrt /storage1/ksdata/??/*mp4`，晚高峰前查下说明文件是有正常更新的。



管理及业务线路跨运营商

目前机器上程序一般是走默认路由解析及探测运营商，故需保证管理线路(默认路由)同业务线路是同一运营商，否则跨运营商往往容易导致一些未知问题，如运营商拦截，跨 isp 调度异常等。

Plain Text

管理线路：`curl myip.ipip.net`

业务线路：`curl myip.ipip.net --interface ppp2`



该问题一般参考解决方案有：

Plain Text

调整线路使运营商保持一致；

关闭管理线路网卡，默认路由走到业务线路上；

更改机器路由，把业务线路路由挪到管理线路之前；

具体机器环境下各自按实际情况调整.

多网卡 ipv4/v6 策略路由问题

多网卡设备，需要正确配置策略路由。确保客户端请求都能从原网卡进行回包。

支持 v4/v6 的设备，v4/v6 的策略路由都需要配置。

汇聚机器离线线路定位

在 path 的第一个路径下 read_test.txt 文件，取 line 字段 。若该值与期望的 nic 数目不匹配则表示有链路离线。

机器不跑 tcp 流量问题

目前线上已全量下发跑 tcp（不限制带宽，100Mb 也可以跑），只要 tcp 开关开启 tcp 相关启动参数也正常的机器肯定会跑 tcp。一点不跑的大概率是异常机器 比如 tcp 端口外部探测不通畅等才会转跑 udp。 目前 tcp 速度这个是程序自行统计计算的，

本地观察是否跑 tcp：

可在日志目录下看到会生成 tcp_0.log 等相关日志，log-level 设为 1；

netstat｜grep 设定的端口看有没 tcp 连接，正常本地启动的 ks 进程端口可以看到有 tcp 连接建立。

自行统计 tcp 带宽流量，太低一般可能就有异常。

排查方式：

1 检查程序启动对应 tcp 参数是否存在：如 `--tcp_conf_path=/etc/ --tcp_nic_port=all:18800`；

2 Netstat |grep 端口 检查程序指定的端口是否成功启动，

3 检查 nat 类型是否符合，要求为 nat=0，1，2，3，nat5 不支持

4 专线或者 nat0 的可以直接拨测下 ip 端口。nat1 的暂时自行排查是否有外部防火墙或者其他策略拦截。

nat 类型探测

目前线上机器普遍要求 nat=0，1，2，3，nat5 不支持。可自行探测加入日常巡检：

1 安装 pystun: yum install python-pip -y ; pip install pytun；

2 pystun -i $ip 即会展示结果；注意防火墙端口要放开，不然可能会拦截探测端口导致结果错误；

正常的为 Open Internet 或者 Full Clone 。其他类型的基本都是；

如下例：

```
--
ppp28: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  m
        inet 124.119.45.195  netmask 255.255.255.255  destin
        ppp  txqueuelen 3  (Point-to-Point Protocol)
        RX packets 151928700  bytes 166359634434 (154.9 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
--
ppp29: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  m
        inet 124.119.44.255  netmask 255.255.255.255  destin
        ppp  txqueuelen 3  (Point-to-Point Protocol)
        RX packets 152603455  bytes 167171270336 (155.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
[root@ ~]# pystun -i 124.119.47.138
NAT Type: Symmetric NAT
External IP: 124.119.47.2
External Port: 23968
```

磁盘更换

磁盘目前写缓存策略是 会优先选用剩余可用空间最大的磁盘优先写入，故有时盘大小不一时可能会看着写入不太均衡，此时相差不大就行。应尽量保证一台设备上磁盘大小一致. 否则也有可能某块盘缓存多集中到上面读写导致整体 io 变慢。

另需注意--path 下不要有放其它文件占用空间，目前有出现--storage_total 参数设置不会超过总磁盘空间，磁盘看着已经用满，但是其中因为有别的文件占用了空间，导致最终程序的实际缓存其实一直很少。

更换坏盘或加新盘时，注意--path=xx 缓存路径顺序不能随意改变，磁盘下有一些程序存储的元数据，丢失了后会被当成新节点重新上线。这在拉黑区域时尤其注意，不然改了磁盘后就不能上机了。

- 增减磁盘基本步骤：先停止进程，然后操作增减磁盘，在--path 最后添加新的磁盘路径序列(注意--path 路径不要随意变更顺序)，重新计算好--storage_total 参数 ，启动进程即可。

uid not match 问题

日志中有时会出现大量类似` target_uid is not match my uid `这种错误. 该错误一般是由于设备更换了--path 参数下的磁盘信息，导致跟后台记录的信息不对称导致。这种的可以删掉旧缓存重新生成，正常如果忽略此错误 一般会需要等待一段时间预计 30min-1h 后也会恢复。

```
2021-11-29 15:37:57:951917 [0x35][0x7f6f098d5700][ksp2p-server][ERR][punch_service.cpp:358][ProcessPunchRelayReq] [00] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
2021-11-29 15:37:58:278251 [0x35][0x7f736cff9700][ksp2p-server][ERR][upload_thread.cpp:357][CheckUid] [00] target_uid(315196822941364224) is not match my uid(316223214227652608) !!!
2021-11-29 15:37:58:278307 [0x35][0x7f736cff9700][ksp2p-server][ERR][upload_thread.cpp:305][ProcessHelloReq] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
2021-11-29 15:37:58:312868 [0x35][0x7f6f098d5700][ksp2p-server][ERR][upload_thread.cpp:357][CheckUid] [00] target_uid(315196822941364224) is not match my uid(316223214227652608) !!!
2021-11-29 15:37:58:312914 [0x35][0x7f6f098d5700][ksp2p-server][ERR][punch_service.cpp:358][ProcessPunchRelayReq] [00] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
2021-11-29 15:38:00:405263 [0x35][0x7f754a7fc700][ksp2p-server][ERR][upload_thread.cpp:357][CheckUid] [00] target_uid(315196822941364224) is not match my uid(316223214227652608) !!!
2021-11-29 15:38:00:405298 [0x35][0x7f754a7fc700][ksp2p-server][ERR][upload_thread.cpp:305][ProcessHelloReq] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
2021-11-29 15:38:00:445355 [0x35][0x7f6f098d5700][ksp2p-server][ERR][upload_thread.cpp:357][CheckUid] [00] target_uid(315196822941364224) is not match my uid(316223214227652608) !!!
2021-11-29 15:38:00:445410 [0x35][0x7f6f098d5700][ksp2p-server][ERR][punch_service.cpp:358][ProcessPunchRelayReq] [00] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
2021-11-29 15:38:00:526054 [0x35][0x7f75497fa700][ksp2p-server][ERR][upload_thread.cpp:357][CheckUid] [00] target_uid(315196822941364224) is not match my uid(316223214227652608) !!!
2021-11-29 15:38:00:526085 [0x35][0x7f75497fa700][ksp2p-server][ERR][upload_thread.cpp:305][ProcessHelloReq] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
2021-11-29 15:38:00:560056 [0x35][0x7f6f098d5700][ksp2p-server][ERR][upload_thread.cpp:357][CheckUid] [00] target_uid(315196822941364224) is not match my uid(316223214227652608) !!!
2021-11-29 15:38:00:560097 [0x35][0x7f6f098d5700][ksp2p-server][ERR][punch_service.cpp:358][ProcessPunchRelayReq] [00] client req target_uid(315196822941364224) is not match my uid(316223214227652608)
^C
```

login error with no unauthorized

这种一般是登陆拒绝。一般是上机器碰巧在需求表里的禁止上机区域 故被拒绝登陆后端。或者就是上面所提到的硬件某个关键信息变了或者删了缓存，导致硬件和缓存信息不匹配当成新机器，落在了禁止区域，也会被拒绝。若是纯粹新机则该区域已然无需求需下线处理。