用样本试跑测内存峰值与临时盘，再匹配租赁方案

正式租用Cell Ranger计算环境前，可以先用一个获准的代表性样本测量内存、运行时间和工作目录增长。测量要以有效结果为前提：任务因输入错误提前退出，得到的低峰值没有选型价值。本文给出一个count示例的试跑协议，并说明怎样把测量转成批次与服务器配置。

Cell Ranger资源试跑只有在任务跑完且输出符合预期时才有参考价值。输入路径错、reference不匹配或FASTQ不完整导致的提前失败，可能只留下很低的内存峰值，不能据此选小机器。本文以一个已确认适用count的代表性样本为例，展示如何固定参数、记录资源和检查临时盘；其他文库应换成对应pipeline及官方输入要求。

试跑前固定输入与输出位置

写下Cell Ranger版本、样本ID、FASTQ文件清单、参考版本、pipeline和本次参数。输出目录使用新的唯一ID，避免覆盖旧pipestance。确认参考和FASTQ对运行用户可读，工作目录有写权限，并在FASTQ、参考、pipestance和日志所在的各个挂载点检查可用空间及配额。压缩FASTQ大小不是运行时临时空间峰值。

```
set -o pipefail
cellranger --version | tee version.txt
run_id="pilot_$(date +%Y%m%d_%H%M%S)"
mkdir -p "$run_id"
cd "$run_id"
/usr/bin/time -v -o ../resources.txt \
  cellranger count --id=sample_pilot \
  --transcriptome=/path/to/reference \
  --fastqs=/path/to/fastqs --sample=sample \
  --localcores=8 --localmem=48 \
  2>&1 | tee run.log
echo "pipeline_exit=$?"
```

示例里的8核和48 GB只是命令格式示意，必须替换为本次作业实际可分配且适合该数据的额度；不能复制后就视为推荐规格。GNU time输出提供进程运行统计，Cell Ranger会启动多个子阶段，因此同时保存调度器或容器的作业级资源记录更可靠。记录完整命令、启动结束时间和退出状态，不能只抄峰值数字。

记录运行中磁盘和作业峰值

启动前记录工作盘容量、inode和项目目录大小。运行期间每隔固定时间记一次对应挂载点的剩余容量，或使用集群提供的作业级文件系统监测；运行后保存pipestance和最终输出大小。周期采样可能错过短时峰值，应把这一局限写明。若运行在cgroup v2或容器中，管理员可根据权限读取对应作业的内存峰值；登录节点的free结果不能替代任务配额数据。

指标	建议记录	判断用途
退出和结果	退出码、日志末尾、web summary	区分成功与提前失败
CPU与时间	墙钟时间、分配核心、CPU利用情况	识别排队、计算或I/O等待
内存	作业级峰值及申请额度	预留波动，避免只看主机RAM
磁盘	各挂载点运行前后及采样记录	估算并行任务的工作空间

用成功结果推算批量配置

先确认试跑显示pipestance成功结束，检查web summary、预期输出文件、日志错误和样本身份，再接受这次资源数据。若第二个样本的规模或文库不同，就不能简单按样本个数等比例推算。批量运行时，一批中多个任务的内存可能重叠；以单任务峰值乘并发数作为保守起点，并留出系统、I/O缓存及其他使用者的空间。若观察到换页、被终止或耗时反而上升，降低并发重新测，不要只扩大核心数。

最终交付的选型依据应包括试跑版本、输入规模、完整成功证据、资源记录、计划并发和数据保留策略。本文未在真实课题FASTQ上做基准；任务规模变化或软件升级后应重新测量。这样可以把服务器需求说明白，也能避免为输入问题或不恰当并发多租不需要的资源。

如果集群提供Slurm统计，优先保存作业最大内存、运行时间、退出状态和分配节点；Cell Ranger会启动多个子阶段，单看一个用户进程可能漏掉子进程占用。没有作业级统计时，也注明观测来源和采样间隔。把“测到的峰值”和“为了波动预留的余量”分开写，后续加样本才知道配置调整依据。

参考资料

Cell Ranger v10.1系统要求（10x官方） · 官方资料

Cell Ranger count pipeline（10x官方） · 官方资料

Linux cgroup v2内存控制文档 · 官方资料