对代表性样本试跑并确定CPU内存与工作盘

在独立目录保留完整组装日志，分开记录进程、任务组与工作盘峰值，并用明确的FASTA检查验收交付文件。

metaSPAdes 提交后被 OOM 杀掉，或运行许久没有输出时，先核对它实际申请了多少 CPU 与内存。当前 SPAdes 官方参数说明中，-t 是线程数，-m 是以 GB 为单位的内存上限；程序到达该限制会终止，实际 RAM 通常低于限制。官方手册的默认值是 16 线程、250 GB，不能把默认值直接当作当前节点的可用资源。

作业申请值和软件参数要同时匹配

先看调度器任务记录里的 CPU、内存、退出原因和峰值，再对照命令的 -t 与 -m。软件内存上限应按作业分配值设置并留出系统及其他进程余量；提高 -m 不会自动为作业多申请内存。并发运行多个样本时，节点总需求还要加总各任务的实测峰值。

症状	核对证据	先采取的动作
作业被系统 OOM 结束	调度器峰值、同节点并发任务、-m 值	降低并发或调整申请，再用代表样本试跑
SPAdes 自身达到内存限制退出	日志阶段、程序提示、当前 -m	确认可申请资源后再调整上限
纠错阶段临时盘不足	输出目录、--tmp-dir 挂载点和空间	将临时目录放到可写且空间充足的文件系统
线程变多但耗时没下降	CPU 利用率、I/O、内存峰值	定位瓶颈，避免盲目增加线程

先用代表数据测阶段峰值

选择读数质量、覆盖和复杂度接近正式样本的数据做试跑。保存版本、命令、reads 校验值、作业资源、退出码、日志、峰值内存和临时盘峰值。不同 K 值迭代或纠错阶段可能呈现不同资源曲线，所以不要只在任务刚启动时观察一次。若试跑数据规模明显小于正式数据，结果只能作为低估风险的参考。

```
set -euo pipefail
: "${R1:?set forward reads}"
: "${R2:?set reverse reads}"
: "${OUT:?set new output directory}"
: "${THREADS:?set threads allocated by scheduler}"
: "${MEM_GB:?set SPAdes memory cap in GB}"
test -r "$R1" && test -r "$R2"
test ! -e "$OUT"
spades.py --meta -1 "$R1" -2 "$R2" \
  -t "$THREADS" -m "$MEM_GB" \
  --tmp-dir "$OUT.tmp" -o "$OUT"
```

谨慎使用降低内存的模式

SPAdes 的 --frugal 模式尝试用组装连续性换取较低内存，官方提示其结果影响不可预测，主要面向菌株变异较高的复杂宏基因组。采用后应重新检查 contig 长度、关键标记和下游分箱，而不是把它当成不影响结果的“省内存开关”。同样，减少线程可能降低并行资源需求，但不一定解决输入质量、临时盘或单阶段峰值造成的失败。

正式配置应以能稳定完成并满足研究质量标准为准。保留完整日志和资源统计，若需继续失败任务，使用 SPAdes 支持的 checkpoint/continue 机制并遵循其约束；修改输入或关键参数时，应先备份需要的结果，再按官方说明决定继续或重跑。

不要把 OOM 等同于“内存上限设得太小”

程序达到 -m 上限而正常报错，与调度器因作业超出申请量强制终止，是两种不同情况。前者可在有资源余量时评估是否调整 SPAdes 参数；后者需要先查看集群作业记录并调整资源申请或并发数。若实际资源高于申请值，单纯把 -m 调大反而会增加被系统终止的概率。对 metaSPAdes，是否可以续跑取决于停止阶段和已有 checkpoint；官方 continue 模式对可更改选项有限制，不能一边换输入或参数一边假设旧中间结果仍然有效。

每次试跑只调整线程数、内存上限或临时盘中的一个因素，并把运行版本和输入哈希保存在结果目录。达到资源限制后先定位失败时处于纠错还是组装阶段，再选择降低并发、换更大作业资源或重新安排任务。若启用 --frugal，应另行检查连续性和目标标记，因为它会对结果结构产生影响，而不仅是运行时间变化。

参考资料

MEGAHIT 官方项目与使用说明 · 官方资料

MEGAHIT 内存设置官方 Wiki · 官方资料

SPAdes 官方运行参数说明 · 官方资料