核对线程、每线程内存、临时文件和I/O

从samtools sort的-m与-@含义入手，区分内存申请、临时盘写入和输入异常，再完成坐标排序及输出验证。

samtools sort 因 OOM 退出时，先确认是内存、临时盘还是输入问题。最容易误判的是把 -m 当成整个任务的内存上限：它约束每个排序线程的内存，不等于进程或作业的硬上限；-@ 会增加排序和压缩线程。并发任务、缓冲区和系统开销还会继续占用资源，因此不能只看参数乘积就断定安全。

先把失败定位到具体环节

记录 samtools 版本、完整命令、BAM 路径、退出码、作业申请资源和失败时间。检查标准错误日志是否提到内存分配失败、临时文件写入失败或输入读取错误；再查看调度器的作业记录和节点日志。集群环境要以作业实际分配的内存为准，不能拿登录节点的空闲内存代替。

表现	先查什么	下一步
作业被 OOM 终止	作业峰值内存、同节点并发、-m、-@	降低并发或线程，跑小样本试验
写临时文件报错	-T 所在文件系统、空间、inode、配额	换到有足够空间且可写的临时目录
输入读取错误	路径、权限、BAM 完整性和上游日志	先修复或重做输入，再排序

确认临时目录和输出不会互相覆盖

sort 可能把中间块写到临时文件，因此临时盘空间不足也会失败。把 -T 指向项目专属目录，先用 df -h 和 df -i 查看空间与 inode，并确认目录有写权限。输出文件放到新路径；失败后不要把未完成的 BAM 当成成功结果。排序完成后用 quickcheck 做快速完整性检查，并按下游需要建立索引。

用小样本逐步调整，而不是一次加满

先固定输入、线程和 -m，记录耗时、峰值内存、临时盘峰值和退出码；只改一个参数再跑对照。若日志指向 OOM，先减少同时运行的 sort 数量或线程，再在作业资源允许时调整每线程内存。增加 -m 可能减少临时块，却会提高内存需求；减小它可能增加临时文件和 I/O。选择能稳定完成的组合，而不是只追求一次运行更快。

```
set -euo pipefail
: "${IN_BAM:?set input BAM}"
: "${OUT_BAM:?set new output BAM}"
: "${TMP_ROOT:?set writable project scratch}"
: "${THREADS:?set threads}"
: "${SORT_MEM:?set per-thread -m, e.g. 2G}"
test -r "$IN_BAM" && test -d "$TMP_ROOT"
test ! -e "$OUT_BAM"
samtools --version | head -n 1
samtools quickcheck -v "$IN_BAM"
df -h "$TMP_ROOT"; df -i "$TMP_ROOT"
/usr/bin/time -v samtools sort -@ "$THREADS" -m "$SORT_MEM" \
  -T "$TMP_ROOT/sample-sort" -o "$OUT_BAM" "$IN_BAM"
samtools quickcheck -v "$OUT_BAM"
samtools index "$OUT_BAM"
```

这段命令假设输入需要坐标排序，且输出准备供区域查询使用；若下游要求按 read name 排序，应先确认其排序要求，因为 name sort 不能按常规方式建立坐标索引。批处理时为每个样本使用唯一临时前缀，避免文件冲突。下次复现时保存命令、软件版本、作业资源和日志，才能判断调整是否真的有效。

把日志留成可复查的证据

建议每次运行单独保存标准输出、标准错误、版本、命令行、作业编号和资源申请值。作业被调度器杀掉时，程序自身日志可能来不及写出最后一行，因此还要查看调度器的峰值内存、退出原因和节点事件。若错误集中在临时目录，检查实际挂载点和项目配额，而不是只查看系统根盘剩余空间。调整参数前保留原始日志和失败文件清单，确认没有其他作业仍在使用这些临时文件后再清理。这样下次重跑时能解释为何改了线程、每线程内存或临时路径，也能避免把不同原因的失败都归为“内存不够”。

参考资料

samtools sort 官方手册（samtools 1.25） · 官方资料

samtools quickcheck 官方手册 · 官方资料

samtools 官方手册 · 官方资料