按单样本实测安排线程、批次与租用资源

固定HUMAnN输入与数据库，在新目录比较线程设置，记录完整流程和中间盘，再用重叠任务验证批量预算。

HUMAnN 跑得慢或内存不稳时，先区分两个设置：--threads 控制线程/进程数量，--memory-use 选择软件内部尽量少占内存还是多用内存的模式。它们不是一回事，也不能代替集群的 CPU 和内存申请。数据库规模、样本读数、临时目录所在磁盘及同节点并发都会改变实际运行表现，因此配置应由代表性样本试跑得出。

先把输入、数据库和输出路径记录清楚

记录 HUMAnN 版本、输入文件大小和校验值、数据库版本、完整命令以及作业分配资源。确认输入格式被当前版本识别，输出目录是新目录且可写，数据库路径存在。若使用共享数据库，不要在多个作业同时构建或修改数据库；每个样本应使用独立输出与临时工作目录，避免中间文件互相覆盖。

先固定 memory-use，再比较线程

用一份有代表性的样本，固定数据库、输入和 --memory-use，先比较较低线程数，再逐步增加。每次只改一个设置，记录总耗时、调度器峰值内存、CPU 使用率、I/O 等待和输出是否完整。若线程提高后 CPU 利用率不升反降，可能受磁盘、数据库读取或共享节点负载限制；继续加线程可能只增加资源竞争。样本大小不同，不能用一个小样本的耗时简单线性推算全部任务。

观察结果	优先核对	调整思路
CPU 不高而耗时长	数据库和临时目录所在存储、I/O 等待	先检查存储吞吐和并发读写
作业被 OOM 终止	调度器峰值、memory-use、同节点任务	先降低并发或采用更省内存模式
加线程却没有加速	节点是否实际分配相应 CPU、外部程序瓶颈	小步试跑后选择稳定值
结果文件缺失	HUMAnN 日志和最后完成的阶段	先解决具体阶段错误，不盲目重复全流程

用同一份输入做一次有记录的试跑

以下示例适用于当前 HUMAnN 支持的单文件输入。将输入格式和数据库配置按实际环境确认；paired-end reads 如何组织应遵循所用 HUMAnN 版本及实验流程说明，不能把 R1、R2 随意拼接成一条序列。先在集群中申请与 THREADS 对应的 CPU，再提交命令。

```
set -euo pipefail
: "${READS:?set input reads file}"
: "${OUTDIR:?set a new output directory}"
: "${THREADS:?set allocated thread count}"
test -r "$READS"
test ! -e "$OUTDIR"
humann --version
humann --input "$READS" --output "$OUTDIR" \
  --threads "$THREADS" --memory-use minimum --verbose \
  2> "${OUTDIR}.stderr.log"
```

示例选择 minimum 作为较省内存的试跑起点；若实际目标是提高速度且节点余量充足，可以按本机 humann --help 中允许的模式对照运行。maximum 会让部分对象尽量保留在内存中，可能缩短反复读写，但峰值内存会提高；不要仅凭“服务器内存很大”就切换。

用完成度而非进程退出判断结果

检查日志是否完整经过 prescreen、核酸比对、翻译搜索和结果汇总等实际启用阶段，再检查预期的通路丰度、通路覆盖度及基因家族输出。保存退出码、HUMAnN 日志、数据库路径与版本、输入哈希、线程和 memory-use 设置。若失败在某个外部工具阶段，应先查看该阶段错误和资源峰值。最终线程数应满足任务稳定完成、结果完整且峰值不超过作业限制；批量运行还要按同时活跃的样本数核算节点总资源。

参考资料

HUMAnN 官方项目与用户文档 · 官方资料

HUMAnN 官方命令参数实现（threads 与 memory-use） · 官方资料

GNU time 资源记录手册 · 官方资料