给计算任务加项目标识并汇总资源与存储记录

完整操作教程 · 参考固定 · 往返恢复验收

CPU核时、峰值内存和存储占用是不同指标，不能只拿运行时长或某一时刻的目录大小做项目成本。本文给出可审计的记录口径和一段轻量采集示例。

运行边界：资源数值取决于调度器记账与采样配置；示例记录不是实测用量或账单。

先统一“算了什么”，再汇总项目

课题组按项目统计资源时，最先要统一指标口径。CPU核时通常按实际消耗的CPU时间累计，不等于墙钟运行时间乘申请核数；单线程跑10小时约消耗10 CPU小时，8核任务运行10小时的理论上限约80 CPU小时，但实际占用取决于线程利用率。峰值内存是任务在某个时间点达到的最大驻留内存，不能把多个任务峰值简单相加后当成课题组的真实同时峰值。GPU时长、显存峰值、读写流量和存储容量也应单独记录。

存储指标同样要写清楚是逻辑文件总量、文件系统已分配块、个人/项目配额，还是备份后的容量。稀疏矩阵、压缩FASTQ、硬链接、快照和对象存储会让这些数值不同。`du`看目录占用会受硬链接和权限影响；`df`显示整个文件系统使用率，不能直接归属到某个项目。应明确采样时点、单位、是否包含临时文件和备份副本。月报里既给原始数值，也给口径说明。

项目标识要从任务提交开始就固定。可以用项目ID作为调度器account、作业名标签、输出目录前缀和日志字段；不要依赖用户手工写的自由文本，因为同一课题可能出现多个拼写。多人共同跑同一项目时，另记提交用户与任务ID，便于负责人核对。调度器如果已配置作业计费，应优先使用其accounting数据作为CPU、内存和GPU资源来源，并先确认运维启用了相应采集插件和存储后端。

指标	常见来源	汇总时的解释
CPU核时	调度器实际CPUTime或进程计时	区分申请核数与已用核时
峰值RSS	调度器MaxRSS或进程监控	注明单任务与采样粒度
GPU时长/显存	设备和作业级监控	记录设备编号及采集缺口
项目存储	配额、目录清单、对象存储统计	注明快照、备份和采样时点

给每个任务分配稳定标签并保存原始记录

建议每个任务在启动时输出一条不可变的任务记录：project_id、job_id、提交用户、命令摘要、开始和结束时间、退出码、资源申请、测得CPU时间、最大RSS、输入输出目录清单和工具版本。命令行可能含访问令牌或个人路径，不能未经检查写入团队可见日志。可以先对参数做脱敏，只保留能解释工作量的关键选项。输入文件不必复制进统计表，记录相对路径、大小和SHA256即可。

如果没有调度器，GNU time可以为单个进程记录elapsed、user/system CPU time和最大RSS，但不同操作系统和`time`实现的字段不同；并行程序、子进程和容器的资源统计也需要核实是否覆盖整个进程树。目录大小可以按周或月由管理员采集，不建议每个用户每次提交任务都遍历全盘。统计程序要写入临时文件再原子替换，避免多人并发时互相截断。记录失败任务也很重要，它能区分资源已消耗但没有产出、排队等待以及用户取消。

示例：用项目标签记录任务及CPU时间

下面的脚本适合在GNU/Linux环境中演示日志字段。真实使用前，先确认`/usr/bin/time`是GNU time、项目ID来自受控列表、日志目录由管理员设置权限。脚本不会捕获命令输出到日志，避免误存样本信息；执行命令仍需由课题组的作业系统和安全策略管理。

```
#!/usr/bin/env bash
set -euo pipefail
PROJECT_ID="${PROJECT_ID:?set approved project id}"
JOB_ID="${SLURM_JOB_ID:-manual-$(date +%s)}"
LOG_DIR="${RESOURCE_LOG_DIR:-./resource-log}"
mkdir -p "$LOG_DIR"
case "$PROJECT_ID" in (*[!A-Za-z0-9_-]*|'') echo 'invalid project id' >&2; exit 2;; esac
printf 'project_id\tjob_id\tuser\tstart\tcommand\n' > "$LOG_DIR/$JOB_ID.tsv.tmp"
printf '%s\t%s\t%s\t%s\t%s\n'   "$PROJECT_ID" "$JOB_ID" "$(id -un)" "$(date -Is)"
"analysis-command"   >> "$LOG_DIR/$JOB_ID.tsv.tmp"
# In a production wrapper, capture GNU time to a restricted sidecar file.
if /usr/bin/time -v your-analysis-command --explicit-arguments 2> "$LOG_DIR/$JOB_
ID.time.txt"; then
  rc=0
else
  rc=$?
fi
printf 'exit_code\t%s\n' "$rc" >> "$LOG_DIR/$JOB_ID.tsv.tmp"
mv "$LOG_DIR/$JOB_ID.tsv.tmp" "$LOG_DIR/$JOB_ID.tsv"
```

示例中需注意一个shell细节：启用`set -e`后，分析命令失败会立即退出，后面的退出码记录不会执行。用于正式采集时应在受控的`if command; then rc=0; else rc=$?; fi`结构中捕获退出码，再写入临时文件并原子改名。本文代码块展示字段与隔离日志的思路，正式版本应按实际调度器接口封装，并对并发写入增加文件锁。不要把申请的核数误写成消耗核时，也不要把脚本估算替代调度器实测。

从明细到月度报表，保留不确定项

汇总时按项目ID分组，先校验任务ID唯一，再处理重试与数组作业。一次任务可能重试多次，既要保留每次尝试，也要标记最终成功的一次；若只保留成功记录，会低估资源消耗。CPU时间缺失时标“未采集”，不要填零。内存记录是采样值还是调度器最大值也要区分。存储应报告项目目录清单与采样日期，并在同一张表中注明是否包含归档和备份。

给课题组看的月报适合分为“已确认用量”“系统级指标”和“待核对项”。例如GPU监控没覆盖某节点，就列出节点和缺失时段；配额统计无法区分硬链接，就说明计量口径。资源核算用于规划共享容量、发现异常任务和沟通成本，不应把粗略统计包装成精确计费。如果团队有收费或成本分摊安排，需另行约定单价、舍入规则、闲置与失败任务的处理方式和数据申诉流程。

权限、保留周期和复核方法

资源日志可能包含用户名、项目名、路径和命令信息，按最小权限保存。PI或项目负责人通常只需要本项目汇总，不必查看其他团队的原始命令。设定保存周期，到期后归档或删除；若需长期留存论文复现信息，保留的是必要的环境与参数元数据，而不是无限期保留全部运行日志。每月抽样核对几条任务ID与调度器记录，随机比对墙钟时间、CPU时间、退出码和输出目录。发现标签缺失，先补标规则并在报表中披露范围，避免人工猜测归属。

用调度器记录核时，别把申请量当实际量

如果使用Slurm，可以在管理员启用作业记账的前提下查询任务明细，例如按JobID查看AllocCPUS、CPUTime、Elapsed、MaxRSS、State和ExitCode。对数组任务和多步骤作业，要确认`sacct`结果包含的是父任务、子任务还是step记录，避免重复求和。一个作业可能申请8个CPU但只使用1个CPU；将Elapsed乘AllocCPUS得到的是理论申请核时上限，不是CPU实际时间。相反，多进程任务累计CPUTime可能高于Elapsed。月报应选定一个主口径，并保留明细字段，不能把多个算法算出的数混成一个精确值。

采集策略要考虑集群是否真的开启了JobAcctGatherType和AccountingStorageType。若没有开启，`sacct`可能无数据或只显示部分字段，应先向管理员确认，不要把空值解释成零使用。MaxRSS取决于采样和任务step，GPU资源还可能依赖GRES/TRES计量配置。字段含义和采集方式需要写在报告的注释中。对重要项目抽取几条任务与作业日志、执行时间和输出目录交叉核对，检查重复重试是否被保留、取消任务是否计入、失败后重新提交是否生成新JobID。

不同研究组对成本的关注点不一样：PI可能关心每月总核时和盘占用，分析人员关心单次任务的异常峰值，管理员关心全系统容量和排队。可以给不同角色提供不同聚合视图，但底层明细应由授权人员保管。面向课题组的视图可以隐藏用户名

或内部节点名，仅展示本项目的数据；负责人需要解释资源异常时，再通过受控工单查看必要的任务记录。不要在公共表格中公开其他组的项目名、路径和命令。

把存储统计拆成在用、归档和备份

存储月报至少区分活跃工作目录、冷归档、快照和备份。项目目录删除文件后，文件系统快照或备份可能仍占空间；硬链接文件也可能让逻辑文件大小与实际磁盘占用不一致。配额可能按用户、组或项目目录统计，含义取决于文件系统配置。对对象存储还要区分对象版本、删除标记和生命周期策略。应在固定采样日和固定时区记录可复查的原始输出，写明统计命令、权限限制及是否遵循符号链接。

项目存储责任人可以先制定生命周期：原始数据何时转只读、分析临时文件何时清理、最终结果如何归档、备份保留几份、谁批准删除。不要根据一次`du`输出自动删除目录；临时文件可能仍被运行进程使用，路径也可能通过软链接指向其他项目。清理前生成待删清单、文件数量和摘要，并由项目负责人确认。存储统计用于容量规划和风险发现，不代表项目数据的科学价值，也不能自动代替课题组的保存义务。

资源报告还应解释缺失值和估算区间。若某些交互式Notebook任务无法关联项目ID，可按用户和时间列为“未归属”，要求下月补充标签，而不是平均分给现有项目。若GPU时长缺采集，报告CPU和存储时应仍可用，同时明确GPU项不完整。想把资源用量转成内部费用时，另行讨论闲置成本、备份成本、软件许可、公共数据库下载和失败重跑的分摊方法；不应把系统采集指标直接称为财务结算金额。

并发、重试和成本口径要能解释给项目组

当多人同时跑同一项目时，单个用户目录的大小并不代表项目总存储。应以经过确认的项目目录或调度器account为分组依据，再决定公共参考库、共享缓存和个人工作目录如何归属。公共数据库可能被多个项目只读使用，不能在每个项目账本里重复算成完整存储；但若系统要求分摊，可记录采用的比例和依据。共享缓存也可能被系统清理，使用者需要知道缓存不是永久保存的数据归档。

重试需要保留每次attempt的资源消耗。失败任务消耗的CPU时间和临时空间仍然是系统成本，但它可能没有形成可交付结果；报告可以分开列成功运行资源、失败/取消资源和排队时间。任务重跑时引用同一父任务ID或实验编号，记录重跑原因，如输入修正、节点故障或参数变更。不要将排队等待算作CPU用量，也不要把进程存活时间直接当CPU核时。解释这些差异比给一张看似精确的总价表更有价值。

月度核对可设置三项简单检查：任务明细中每个JobID是否唯一，项目归属是否来自受控映射，无法归属项占比是否有记录。抽取高耗资源任务联系责任人确认分析是否预期；若出现异常循环或输出暴涨，先暂停新增任务并保留日志，不要在未确认数据状态前清理运行目录。项目负责人可以据此调整并发或分批策略，管理员据全系统容量安排资源。容量规划与科研判断分开记录，避免把低资源占用误解为工作质量低。

配套练习包

本PDF对应的ZIP包含本文代码、主题模板或合成数据、README和SHA256校验文件。运行前核对包内清单，并按文章说明在独立目录验证。

参考资料

Slurm sacct 作业统计字段 · 来源资料

Slurm 资源记账配置要求 · 来源资料

GNU Time 手册 · 来源资料

操作建议：先在合成小样本上验证流程，再依据项目参考、标签合同、存储策略与获配资源验收真实数据。