先测文件数量与元数据开销，再比较打包和增量迁移

完整操作教程 · 参考固定 · 往返恢复验收

先统计数量和元数据开销；冻结目录后比较单包搬迁与增量rsync，并在目标端重新校验。

运行边界：迁移脚本仅覆盖冻结后的普通文件；ACL、权限映射与特殊文件需另行验证。

几百万个小文件迁移慢，常见瓶颈是目录遍历、文件创建和元数据操作，带宽没有跑满并不一定是网络故障。先决定这些文件是否都需要保留，再比较打包传输与逐文件增量。本篇处理已冻结、不再写入的普通文件目录，目标是内容完整迁移；不负责复制原系统账号或整机权限。

用小批样例建立两条路线

记录文件数、总字节、目录层级、源目标文件系统和完整墙钟。抽取有代表性的子目录，分别测直接rsync与先打包再传，包含目标端解包及校验时间，不能只比较传输阶段。已有目标大部分相同时增量可能合适，初次搬迁密集小文件时打包可减少传输侧逐文件往返。

路线	适用情况	额外成本
直接rsync	后续更新，目标已有多数文件	两端仍需遍历和比较文件
tar普通文件包	冻结目录的首次搬迁	包与解包目录可能同时占空间
权限/链接归档	确有ACL、硬链接或特殊文件要求	需另订权限和恢复合同，本例不覆盖

冻结后生成NUL文件清单与校验表

以下在GNU工具环境运行。根目录由本人控制，禁止其他任务继续改写；这里只纳入普通文件，空目录不影响内容验收。先拒绝符号链接及其他特殊对象，文件名中的空格或换行由NUL清单处理；不要改成按空格切分的for循环。清单与归档放在源目录之外。

```
#!/usr/bin/env bash
set -euo pipefail
source_dir=/data/frozen-project
test -d "$source_dir"
mkdir migration-001
evidence="$PWD/migration-001"
cd "$source_dir"
unusual="$(find . ! -type d ! -type f -print -quit)"
test -z "$unusual"
find . -type f -print0 | LC_ALL=C sort -z > "$evidence/files.nul"
test -s "$evidence/files.nul"
xargs -0 -r sha256sum < "$evidence/files.nul" > "$evidence/files.sha256"
/usr/bin/time -v -o "$evidence/tar-time.txt" \
  tar --create --file="$evidence/files.tar" --null --verbatim-files-from \
      --files-from="$evidence/files.nul"
cd "$evidence"
sha256sum files.tar files.nul files.sha256 > transport.sha256
```

用已有SSH配置将migration-001传到目标的独立接收目录，例如rsync -a --partial migration-001/ target:/incoming/P001/。不加delete或remove-source-files。先在目标接收目录执行sha256sum -c transport.sha256，再创建空的恢复目录，使用tar -xf接收目录中的files.tar -C恢复目录，最后在恢复目录执行sha256sum -c接收目录中的files.sha256。命令中的路径替换为目标端实际绝对路径。

比校验通过再多检查一步

SHA256表证明清单内文件内容相同，但不会自动报告目标多出来的旧文件。使用新建空目录能避免这类混入，仍需用相同find/sort规则生成目标files.nul并与源清单cmp；数量、相对路径和每个内容都一致才记内容迁移完成。另抽查下游程序能按新路径找到必需输入。

rsync的常规快速检查主要看大小和修改时间，不应当作独立密码学校验；checksum也需要读取数据。迁移期间源文件改变会破坏冻结假设，此时停止交付、重新生成清单和包，不用忽略“file changed”警告继续验收。链接、ACL、属主或空目录有意义时，应另行测试相应保留选项，不能套用本篇普通文件合同。

计划容量要同时放得下源包、目标接收包、恢复文件及排序临时量。打包不一定要压缩，已有gzip数据通常不应盲目再压缩。先完成代表目录对照，再按实测元数据吞吐安排批次；内容和路径验收、保留周期确认后再单独清理源副本。

先测文件系统元数据成本

小文件迁移的总字节数并不能描述工作量。几百万条目录项需要大量stat、open、create和close调用；源端元数据服务器或目标端inode可能先成为瓶颈。样例应按目录深度、文件大小分布、文件名特点和权限情况分层抽取，既包括大量极小文件，也包括较大的代表文件。记录总文件数、总字节、目录数、最长路径、异常对象和两端文件系统类型，样例抽取规则要保存下来。

对比路线时分开记源扫描、清单生成、归档打包、网络传输、目标解包、校验和下游访问的时间。端到端时间才是任务需要的结果；单看传输速度会忽略打包和解包阶段。尽量在相近负载时段重复测试小样例，并记录是否命中缓存。不要在共享生产文件系统上启动未获批准的大规模find或校验扫描，以免影响其他课题组。

冻结源目录并建立相对路径合同

文件清单以迁移根目录为起点，记录相对路径、字节数和SHA256。原目录必须冻结或有快照一致性保证，否则扫描与打包期间文件变化会使清单失去意义。若需要ACL、所有者、硬链接、扩展属性、稀疏文件或符号链接，应单独定义并演练相应归档选项；普通文件的内容校验不等于这些元数据已保留。还应检查文件名编码、换行和前导减号等特殊情况，始终用NUL分隔或可靠清单格式。

目标恢复到一个新建、空的、权限合适的目录。这样源目录的相对路径可逐一对应，目标中不会夹入同名旧文件。传输完成后再次生成目标清单，比较文件数、路径和哈希；仅校验压缩包本身只能证明包在传输途中没有改变，不能证明解包内容符合预期。

评估tar与增量路线的真实取舍

tar将大量元数据操作聚合成更少的传输对象，可能减少网络协议往返和目标端小文件写入的频率，但创建归档依旧要遍历源文件，解包仍需创建所有目标文件。归档文件通常会和源目录同时存在，若启用压缩，CPU时间和临时空间也会增加。磁盘空间紧张时，先计算源文件、归档、目标副本和校验清单可能同时存在的峰值，再决定放在哪个挂载点。

rsync适合反复增量更新或目标已有大部分内容的情形，但首次同步仍需扫描和创建小文件。常规快速检查依赖大小与修改时间；如需内容级一致性需采用经过评估的校验流程，但这会额外读取源与目标数据。对冻结目录的一次性迁移，打包路线可能更简单；若需频繁更新，保存清单并做分批增量可能省去重复传输。结论取决于文件数量、网络往返、文件系统和任务频率，不能只根据“打包通常更快”下结论。

控制并发，观察元数据与空间瓶颈

同时创建多个大归档或并发解包会增加磁盘寻道、inode分配和缓存争用。先以一个小目录测出单任务的端到端表现，再逐步增加并发，观察吞吐和平均单文件处理时间是否改善。若CPU不忙、网络低利用率但文件创建很慢，可能是元数据服务或目标盘限制；若传输快但解包慢，瓶颈已转到目标文件系统。监控可用空间、inode、I/O等待和系统负载，勿仅依据网卡带宽判断。

迁移中断时保留状态清单和已验证归档，不用`--delete`或`--remove-source-files`自动清理源端。重跑前核对源目录是否仍冻结、目标是否为预期空目录、已完成分片的哈希是否有效。归档分片要在清单中记录顺序和覆盖路径，避免缺一片仍进入解包阶段。

迁移验收与回滚计划

验收分两层：第一层检查迁移对象数、相对路径和内容哈希；第二层在目标环境用实际下游程序打开代表输入，确认权限、目录布局和文件格式可用。若保留ACL或链接，则按合同分别检查。记录未迁移对象及理由，不能把忽略列表藏在脚本参数里。确认项目负责人接受结果后，才安排源端保留期结束后的清理。

正式切换前应保留一段回滚窗口：旧路径仍只读可访问，新路径通过抽样和应用检查后再通知团队更改工作流。若发现路径缺失、哈希不符或权限差异，停止删除并从已验证源端重新迁移。迁移报告记录工具版本、命令、源和目标文件系统、并发数、各阶段时间、峰值空间、校验结果与例外。若专用工作盘或inode容量影响时间表，再据这些实际指标评估计算环境；仅仅网络带宽有余量并不能说明要增加CPU或GPU。

特殊文件和权限需要独立合同

本例先拒绝符号链接和非普通文件，是为了避免在不了解目标语义时把链接追踪到根目录外、复制设备文件或丢失元数据。如果项目确实需要ACL、所有者、组、硬链接、符号链接、时间戳或扩展属性，应逐项写清谁有权恢复这些属性、目标账号映射规则、归档工具选项及验收方法。权限位数字相同也不意味着目标用户和组存在；高校不同服务器的UID/GID常不一致。

归档文件最好记录创建工具及版本、命令参数、源端快照标识、文件数、总字节、校验清单、归档哈希和目标恢复日志。tar包成功退出不证明每个应用都能按原路径打开数据。若迁移涉及数据库、正在写入的结果、软链接、特殊权限或多文件间的一致性，应使用应用感知的备份/迁移方式，不能直接套用静态普通文件脚本。

为超大目录分片时保持无遗漏

若单一归档超出可用工作盘或传输窗口，可按相对路径前缀或稳定清单分片，并为每片生成独立文件列表、记录数和哈希。分片规则必须互斥且覆盖完整目录；最后合并各片清单，检查重复路径、遗漏路径和总记录数。不要按当前遍历顺序随意切片，因为文件变化或排序差异会令重跑边界漂移。清单可以按稳定排序的NUL路径流与明确分片边界建立。

分片传输时目标仍恢复到空的隔离目录。每一片校验通过后再解包，避免未校验包写入最终路径；但如果多片路径重叠，解包可能覆盖先前文件，因此在清单阶段先检测路径交集。失败只重传未验证分片，全部完成后仍要对目标全量文件清单做一次最终校验。

报告瓶颈需要同口径数据

比较rsync和tar时，在同一源样例、同一目标挂载点、相近系统负载和同一校验强度下记录阶段耗时。rsync首次同步与后续增量应分开；tar是否压缩也应单独标记。记录源端扫描时间、传输字节、目标端创建文件数量、校验I/O和失败重试。没有权限读取文件系统元数据指标时，在报告中注明不可得，不将推测写成测量结果。

若小文件迁移慢，代表性试验可帮助区分源扫描、网络往返和目标文件创建。网络利用率低并不足以证明网络无问题；文件系统延迟、服务端节流或连接数限制也会影响观察。只有当多个样例显示瓶颈稳定地落在元数据操作且目标侧能承受归档展开负载时，才考虑批量打包路线。否则保持小批次并向存储管理员确认服务能力。

配套练习包

本PDF对应的ZIP包含本文代码块、主题练习模板或合成数据、README和SHA256校验文件。请从同一教程页面下载练习包；先核对校验值，再在独立目录复现。

参考资料

rsync官方手册 · 来源资料
GNU tar NUL文件清单 · 来源资料
GNU Coreutils sha256sum · 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。