从课程样例试跑到并发演练，规划教学账号与资源

从学生权限样例到全班并发演练，准备账号、环境、结果核查和课后数据安排。

几十名学生同时上机，先选一份真实课程样例，再测登录、启动和计算并发。人数乘一个想象的配置，无法反映共享盘、环境导入与同时提交任务的高峰。本篇按既定课时和教学流程制定账号、样例、并发演练及课后交接，不假设服务器租赁自动包含教学平台。

把每个教学环节变成验收项

明确学生人数、SSH或Notebook入口、课程软件、输入数据、每节执行步骤和预计提交时点。教师先在学生同等权限下完整跑一次，保留固定版本、数据指纹及预期结果；测试管理员账号成功，不代表学生能访问环境或保存作业。大样例与课堂缩小样例的科学范围也要说明。

对象	本例约定	验收要点
登录	每人独立账号与恢复方式	提前测试校园网和实际入口
环境	同一课程版本和只读样例	个人输出、缓存和Notebook分开
资源	完整单次任务加并发演练	同时导入、启动与写盘都要测
课程后	作业回传、账号期限与重置	先按约定保留数据，再执行清理

先运行课程提供的同一小样例

下面在学生自己的可写目录执行。run-pilot.sh是教师已验收的实际课程脚本，sample-data.sha256对应课件目录内的固定样例；路径按课程部署替换。GNU Time记录完整运行，成功标记只说明命令链结束，后面还须核对课程定义的结果。

```
#!/usr/bin/env bash
set -euo pipefail
pilot=/course/lesson/run-pilot.sh
test -s "$pilot"
(
  cd /course/lesson
  sha256sum -c sample-data.sha256
)
umask 077
mkdir course-pilot-001
root="$PWD/course-pilot-001"
cd "$root"
/usr/bin/time -v -o resources.txt bash "$pilot" > run.log 2>&1
printf '%s\n' 'command chain ended; inspect lesson outputs' > COMMAND_OK
```

脚本输出位置也应使用当前个人目录，若课程脚本硬写到公共目录，先修部署而不是让学生共用写权限。核对结果文件、内容、软件版本及作业提交路径，分别测试失败重试和断线恢复；不能只确认Jupyter页面打开就算实操课准备完成。

分层增加并发，记录高峰

按少量测试账号、半班、预期峰值逐步演练，同步执行真实登录、内核启动、包导入和最重计算步骤。记录完成时间分布、错误率、总RAM、I/O和磁盘峰值，而不只看一个任务CPU。数据可以共享只读，个人缓存、临时文件和输出要核对并发安全。

例如某样例单次测得2核和2GB主机RAM，20份同时任务的粗略基数是40核和40GB，还需结合实测并发、服务开销及共享阶段修正；这是教学算例，不是通用规格。若分批提交或存在等待阶段，不能把班级总人数直接当每一时刻的活动任务数。

验证限制、支持与课后安排

若采用JupyterHub，内存和CPU限制是否生效取决于实际Spawner及部署；写入mem_limit或cpu_limit不代表所有实现都会强制执行。SSH入口也要核对实际作业限制。用学生账号做越额与恢复演练需在获准测试范围内安排，避免正式课堂才发现一个任务拖垮全班。

开课前发明确的入口、样例目录、输出目录、故障反馈及助教分工，给首次登录和网络问题留缓冲。课后按约定导出作业、关闭账号及重置测试环境；不在未确认回传前统一删除学生数据。最终容量与租期依据课程日期、任务高峰和演练结果确定，平台部署及教学支持范围另行写清。

参考资料

JupyterHub官方Spawner资源控制说明 https://jupyterhub.readthedocs.io/en/stable/reference/spawners.html

JupyterHub各Spawner支持边界 https://jupyterhub.readthedocs.io/en/stable/reference/api/spawner.html

GNU Time完整任务记录 https://www.gnu.org/software/time/