上周我提了全量微调的想法，后面开始了实验，快两天的时间跑了0.99个epoch.https://wandb.ai/jerrylikespython-bnu-hkbu-united-international-college/uncategorized/runs/gc8fusve/overview这是实验的实时进度和结果。不知道为什么在0.99epoch卡住了，目前已经运行133小时还是没有返回新的结果。检查可能是处理的逻辑有问题。

pytorch存在锁内存的情况。经常会报错目前需要申请xxxmb内存，但pytorch占用25gb的内存，剩余xxxgb被pytorch占用不能分配。

尝试修改batch\_size。从16修改成4。但是内存占用大小没有改变。

看到知乎的一个解法说dataloader\_num\_workers=0#这个不为零会导致copy-on-access的现象（pytorch内存（显存）泄漏的常见原因和解决方法 - 吴业涛的文章 - 知乎

<https://zhuanlan.zhihu.com/p/672921675>的问题3）。

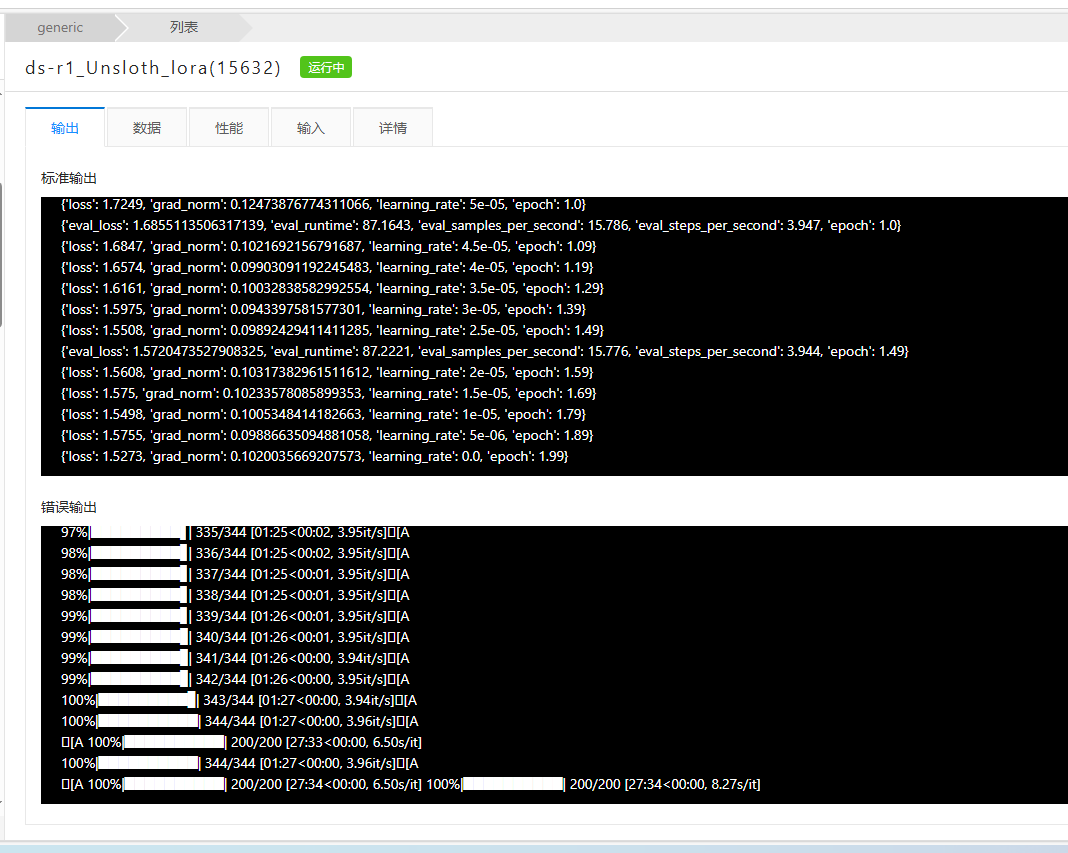
最后查找到问题，是数据分片未对齐造成的，最后一个batch的数据没有对齐，会造成部分进程提前退出，参考<https://support.huaweicloud.com/intl/zh-cn/usermanual-standard-modelarts/modelarts_trouble_0108.html>的博客发现的问题。然后采取最后一个batch不足直接丢弃。

内存会有25是因为cuda有个将一半的内存虚拟化为显存的技术。

🧩 如果你使用的是 Transformers 训练器（Trainer + DeepSpeed）

你只需要在命令行传：

--deepspeed ds\_config.json

在2080ti和v100都出现了类似的错误，推测是数据分片未对齐

之前提交脚本一直没有返回错误就退出

log下没有任何输出，由于刚接触hpc不太会调试，一直想通过打印的方法，但是一直没有任何输出，我使用了三个文件。作业脚本.sh文件，训练代码.py文件和deepspeed设置.json文件。我反复替换想找到问题，未果，最后认为是deepspeed问题，可能是不兼容。排查环境脚本意外发现当前cuda为12.8，集群cuda12中只支持12.2和12.1。然后开始在shell中申请gpu和调试，意外发现shell中调用.sh居然会有输出。排查了一圈，最后发现罪魁祸首居然是wandb.wandb有交互式登录，导致 DeepSpeed 初始化后看似“一直没输出”——程序正等待我在终端输入选项。你的 SLURM 脚本里 export WANDB\_API\_KEY="..." 已经把 key 放到环境变量里了，但 WandB 默认的 wandb.init() 依旧会在没有显式登录的情况下弹出交互提示。要彻底消除这个交互，你需要在 Python 代码里 显式调用 wandb.login（而不是仅靠 wandb.init），并且在初始化时指定 mode="online" 或 mode="disabled"

集群队列有多个节点，部分节点有不同的卡闲置，我希望申请多个节点然后使用deepspeed实现不同节点之间的通信。但Deepspeed进程的端口一直报错29500被占用，我想直接修改成10086,发现输出还是29500，想直接kill进程回到shell发现lsof -t -i :29500有noting个进程在运行。经过大量排查发现服务器会强制覆盖脚本冲突环境，通信端口默认29500并且占用，用户不能自己修改。需要将自己获取端口之类的代码删干净。节点数量改成1。多个进程运行会争抢端口再报错，所以进程数量改成1

最终成功运行只能使用如下格式

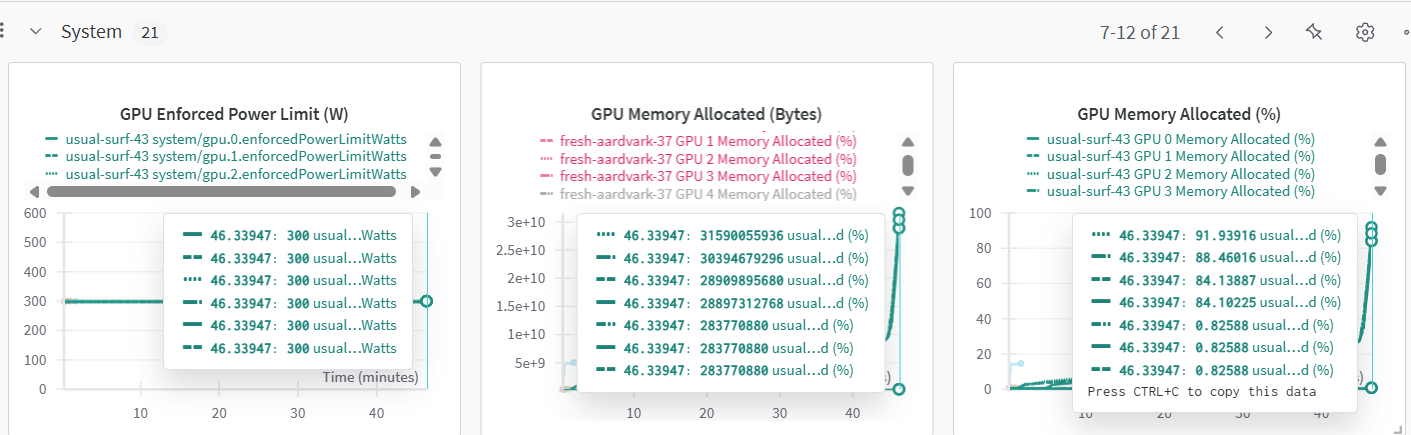
#SBATCH --nodes=1 # 单节点

#SBATCH --ntasks-per-node=1 # 每节点进程数，通常等于GPU数

#SBATCH --gres=gpu:4 # 每节点使用4块GPU

#SBATCH --cpus-per-task=6

使用4张v100训练1.5b



torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 560.00 MiB. GPU 3 has a total capacity of 31.74 GiB of which 21.38 MiB is free. Including non-PyTorch memory, this process has 31.71 GiB memory in use. Of the allocated memory 30.93 GiB is allocated by PyTorch, and 78.23 MiB is reserved by PyTorch but unallocated.

大力出奇迹

