创新设计方向

1.设计方向名称（建议）

AI-aware Tiered Tensor Store（AAT-TS）——面向模型推理/微调的智能分层存储与预取中间件

2.背景与出发点（为什么新）

1. 近年来模型参数 / 激活 / checkpoint 数量爆炸（从几十MB到几百GB），模型运行（尤其推理冷启动与在线小批量推理）对 随机/准实时 I/O 敏感。
2. 传统存储层只做静态冷热分层（例如 HDD/SSD），缺少对 AI 工作负载访问模式（tensor 级别、按 layer/step 访问）的理解与主动优化。
3. 存储与 AI 框架之间的"语义鸿沟"：框架知道要哪个 tensor / layer，但存储层不知道；中间件可以承担语义桥接，做语义驱动的缓存/预取/压缩/分层迁移，改善延时与吞吐、降低整体存储成本。
4. 现有大型方案（模型并行/ZeRO 等）关注算与通信，本项目聚焦 存储层对推理/微调延时与成本的优化，且做成轻量软件中间件，便于学生实现、演示与测评。

3.核心创新点（可写入参赛文档的"创新性说明"）

（1）语义级预取器：通过采集模型运行 Trace（例如哪些 layer、哪类 tensor 在何时被访问），用轻量预测器（Markov / small RNN / 协程统计）预测下步需要的 tensor 并提前把它们从慢层迁移到快层。
（2）透明 POSIX/对象层拦截：以用户态 FUSE 或对象存储代理的方式对接 AI 框架，无需改写模型代码即可受益（可演示 PyTorch / HuggingFace 模型）。
（3）按 tensor 特性自适应策略：小而频繁访问的 activation 与 embedding 放入高速缓存（RAM/pmem/Redis），大且少访问的 checkpoint 存到冷存（MinIO / 本地磁盘）并可选择压缩。
（4）成本-性能可配置策略：提供策略引擎（规则+学习器）在"性能优先 ↔ 成本优先"之间动态切换，适合云端/边缘不同场景。
（5）轻量可复现的开源原型：以 Python 为主，降低实现门槛，便于在比赛中演示性能收益与工程实现细节。

4.实际应用情形

1. 在线小批量推理服务（SaaS）：用户请求频繁但每次请求只触发部分 layer 的读取，AAT-TS 提升冷启动/首次响应延时。
2. 边缘微调/个性化：边缘节点存储资源有限，AAT-TS 帮助把热子集缓存到本地快速层、把历史 checkpoint 压到远端冷层。
3. 科研/训练环境的模型版本管理：快速访问最近训练产生的 checkpoint，历史版本压缩归档，节省空间。

AAT-TS：一个面向 AI 工作负载的智能分层存储中间件，通过语义感知的缓存与预取，把频繁访问的 tensor 放在快速层，冷数据放在低成本层，从而降低延时并节约存储成本。