

AI 存储：初次探秘

主讲人：孙亮

Content

目录

1. AI 需要什么样的存储
2. 存储如何满足 AI 的需求
3. AI 存储的发展

Part 01

AI 需要什么样的存储

■ 企业数据中心的发展历程和趋势

传统数据中心



- 设备独立
- 硬件孤岛
- 厂商异构
- 管理复杂
- 数据存储

虚拟化数据中心



- 资源池化
- 管理统一
- 横向扩展
- 降低成本
- 数据利用

云化数据中心



- 按需配置
- 自动部署
- 服务标准化
- 业务平台化
- 数据应用

AI算力数据中心



- CPU + GPU
- 分布式计算
- 高性能网络
- 高性能存储
- 高效调度
- 数据深度学习

■ 存储发展历程

外部市场需求变化

第一阶段

- 计算机普及：
 - 更加亲民的设计需配备相应的存储设备
 - 计算机的推广拉高存储设备销量

第二阶段

- 互联网发展
 - 互联网方便了不同用户的数据信息共享，从而要求存储提供基于不同连接方式支持
 - 网速增快需要存储传输速度匹配

第三阶段

- 云计算出现
 - 基于IT基础设施资源共享有效提高了利用率，并降低用户IT部署门槛，数据爆发
 - 数字化进入公众视野，商业模式变革带来更多新兴数据创新需求

存储特点

DAS直连存储

- 数据存储量小；主机数量少
- 共享需求弱

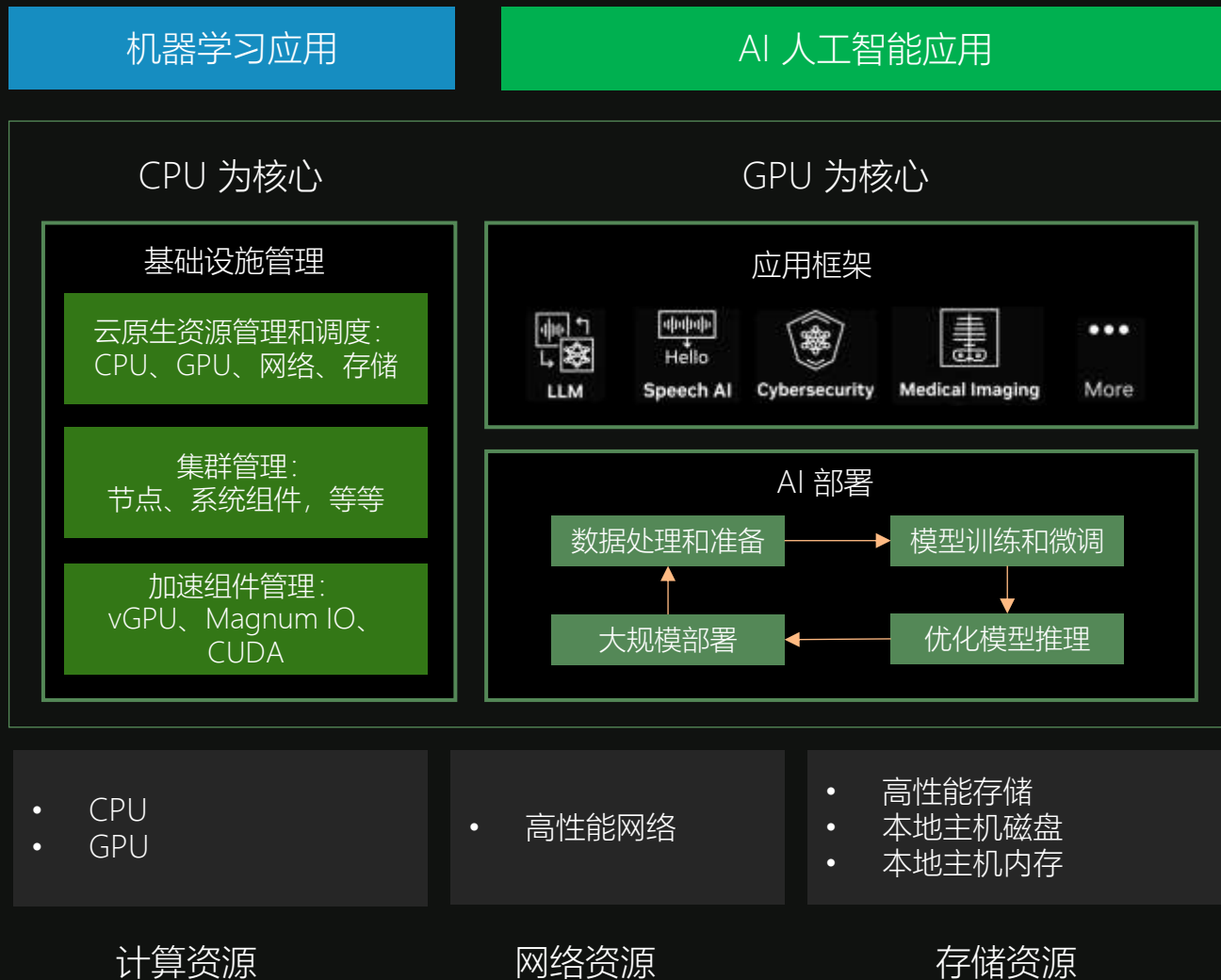
共享网络存储SAN & NAS

- 外置存储共享需求
- 互联网下客户端与服务器交互
- 存储扩展需求

统一存储、分布式存储

- 不同文件协议类型、块数据等需统一管理
- 并发数据快速访问
- 非结构化数据增长，快速应变、弹性拓展要求更高
- 分布式存储、HCI出现，更加适合云环境

■ 第四阶段： AI 算力中心对存储要求



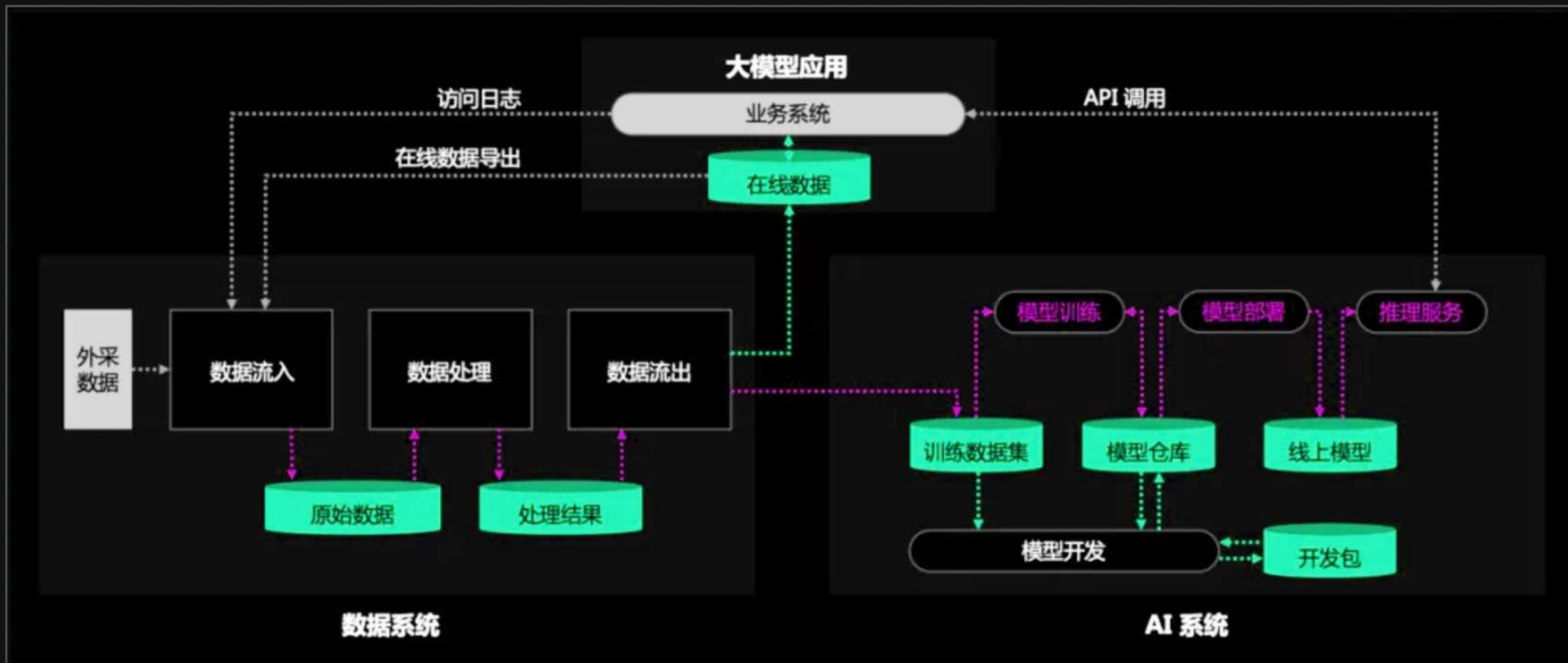
高性能存储 - 要求:

1. 符合 POSIX 协议的文件系统，提供高性能的并发读写
2. 支持高性能网络，例如 InfiniBand、RDMA
3. 利用本地内存缓存数据
4. 利用本地磁盘缓存大量数据集

用户存储 - 要求:

1. 能提供高性能的元数据访问性能，高 IOPS，以及关键的企业级能力，例如：checkpointing
2. 提供冗余的数据访问网络，避免故障

■ 大模型应用的数据流程



■ 大模型应用数据流转过程对存储的需求

关键词	海量数据，持续更新	快速开发，效率为王	时间即金钱！拒绝等待，拒绝失败	规模再大些，部署再快点
	 数据存储和处理	 模型开发	 模型训练	 模型推理
业务场景	数据采集和导入	实验管理	数据集读取	模型分发部署
	数据清洗、转换、标注	交互式开发	checkpoint 保存	
	数据共享和导出	效果评估	checkpoint 加载	
	数据长期归档			
存储需求	生态互通 高吞吐 大容量	POSIX 兼容 可共享 高可靠	数据集：读得快、少等待 checkpoint：高吞吐、少耗时	高并发、高吞吐、高效率

Part 02

存储如何满足 AI 的需求

■ 典型的大模型训推过程是如何产生对存储的要求的？

// 一个训练迭代多轮 Epoch

Repeat (for each epoch):

// 随机打散数据集

List files in the dataset and shuffle

// 读取一批数据开始训练

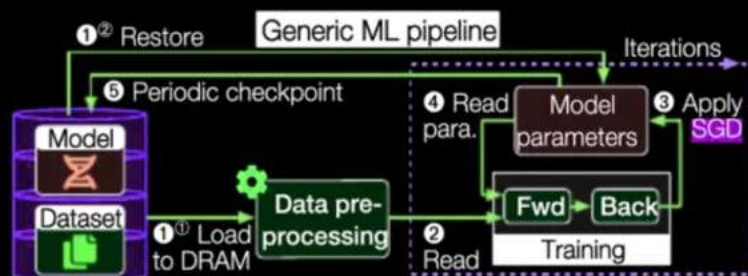
Repeat (for each batch):

Read files of the batch

Training

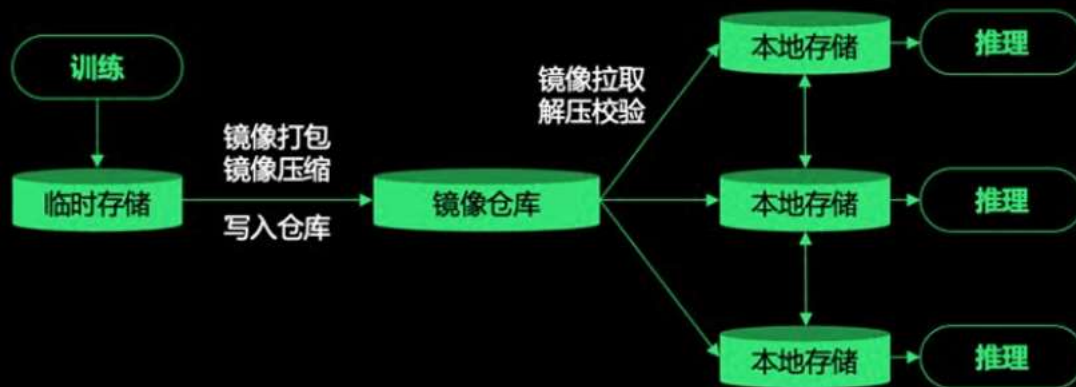
// 周期 checkpoint 用于故障恢复

Checkpoint if necessary



- 训练数据加载 – 数据加速
- 中间状态保存 – checkpoint缓存
- 训练结果模型保存 – 镜像仓库

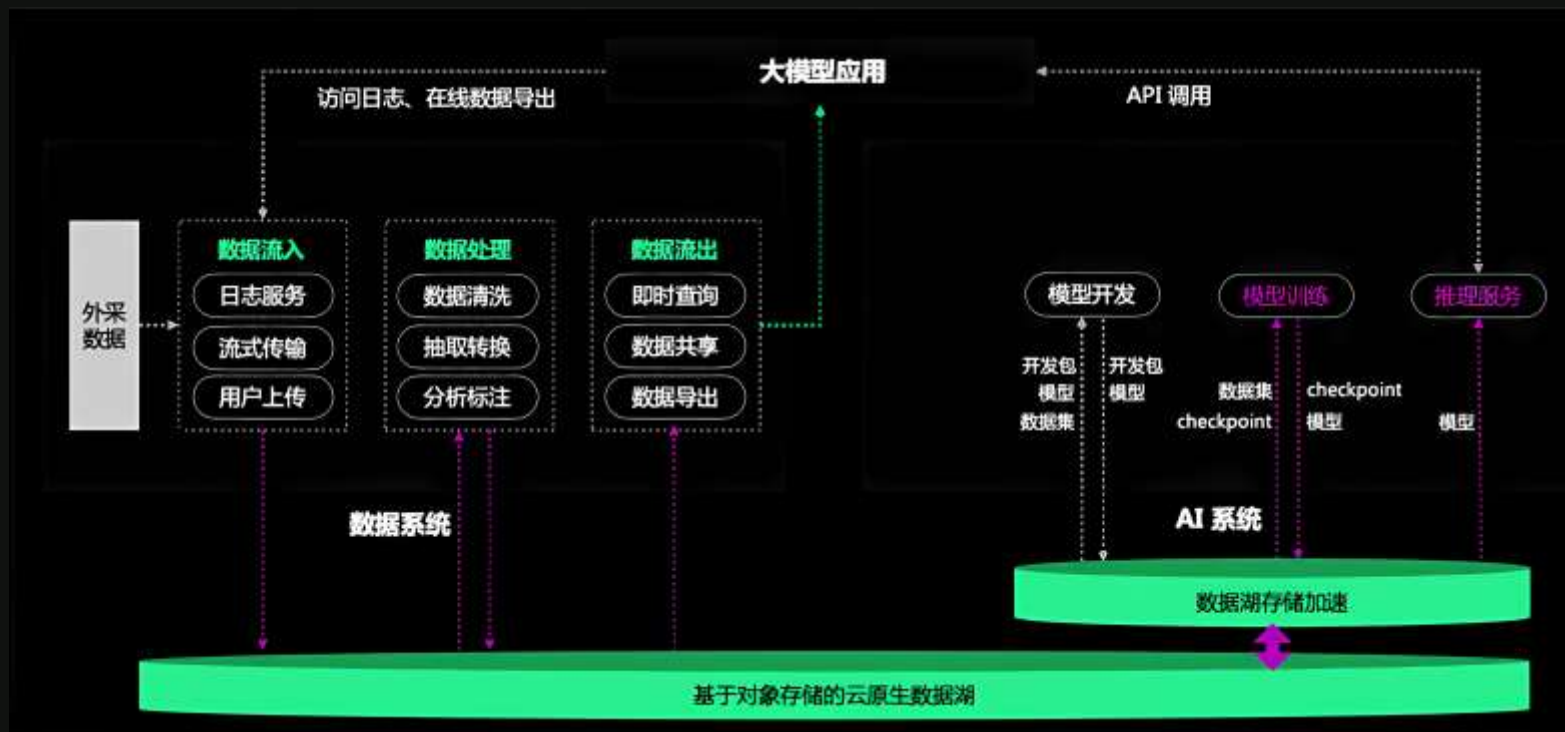
□ 模型镜像的分发



■ 大模型训推过程中常见的存储方案

方案一：缓存系统 + 外部存储

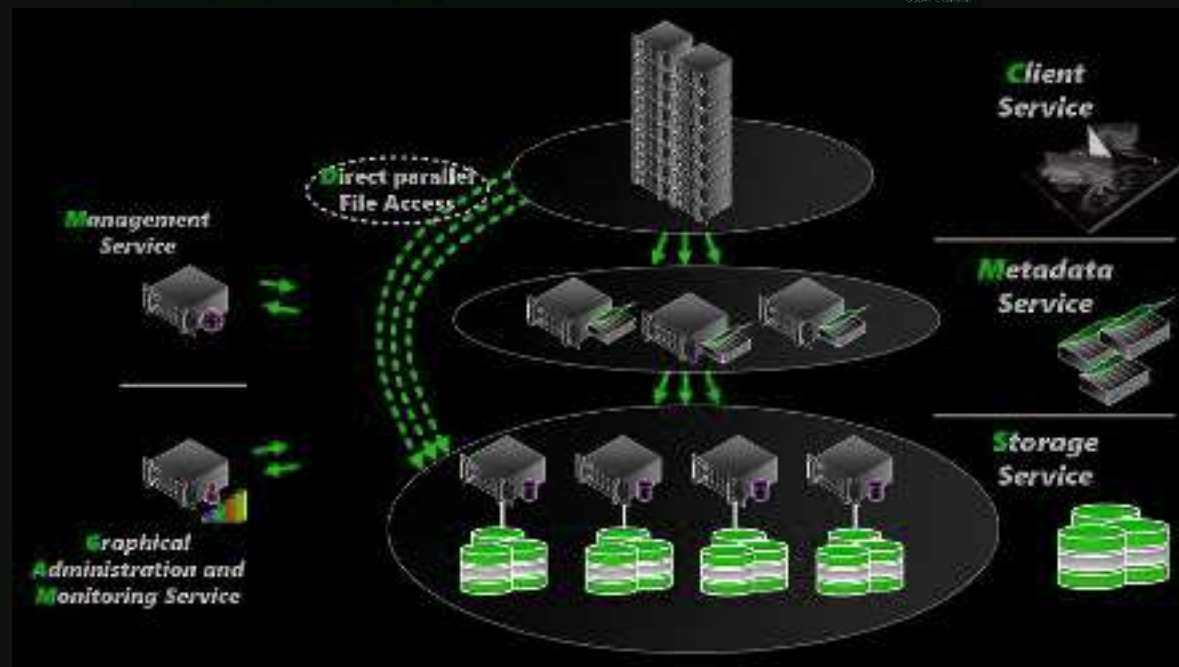
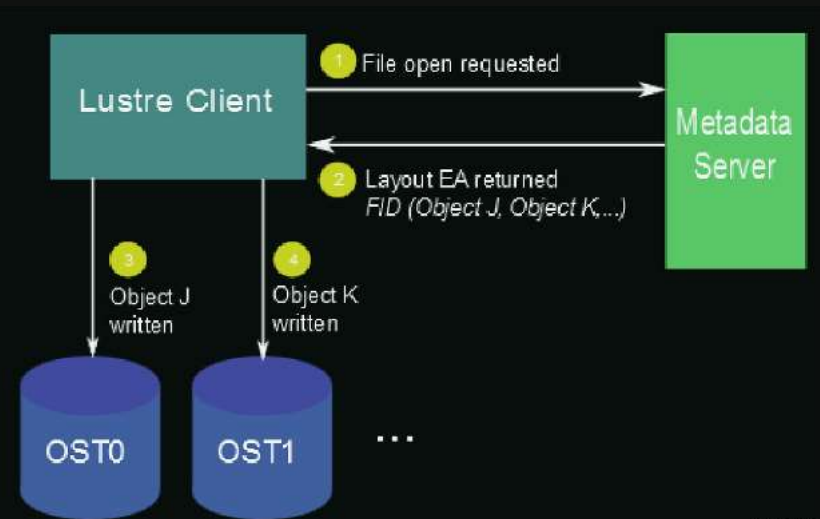
- 典型代表：JuiceFS + 对象存储，Alluxio + 第三方存储，HwameiStor + 第三方存储，等等
- 特点：
 - 置于外部存储与计算应用之间的独立系统
 - 缓存数据，提供高性能数据访问
 - 更适合高速数据读取



■ 大模型训推过程中常见的存储方案

方案二：高性能存储

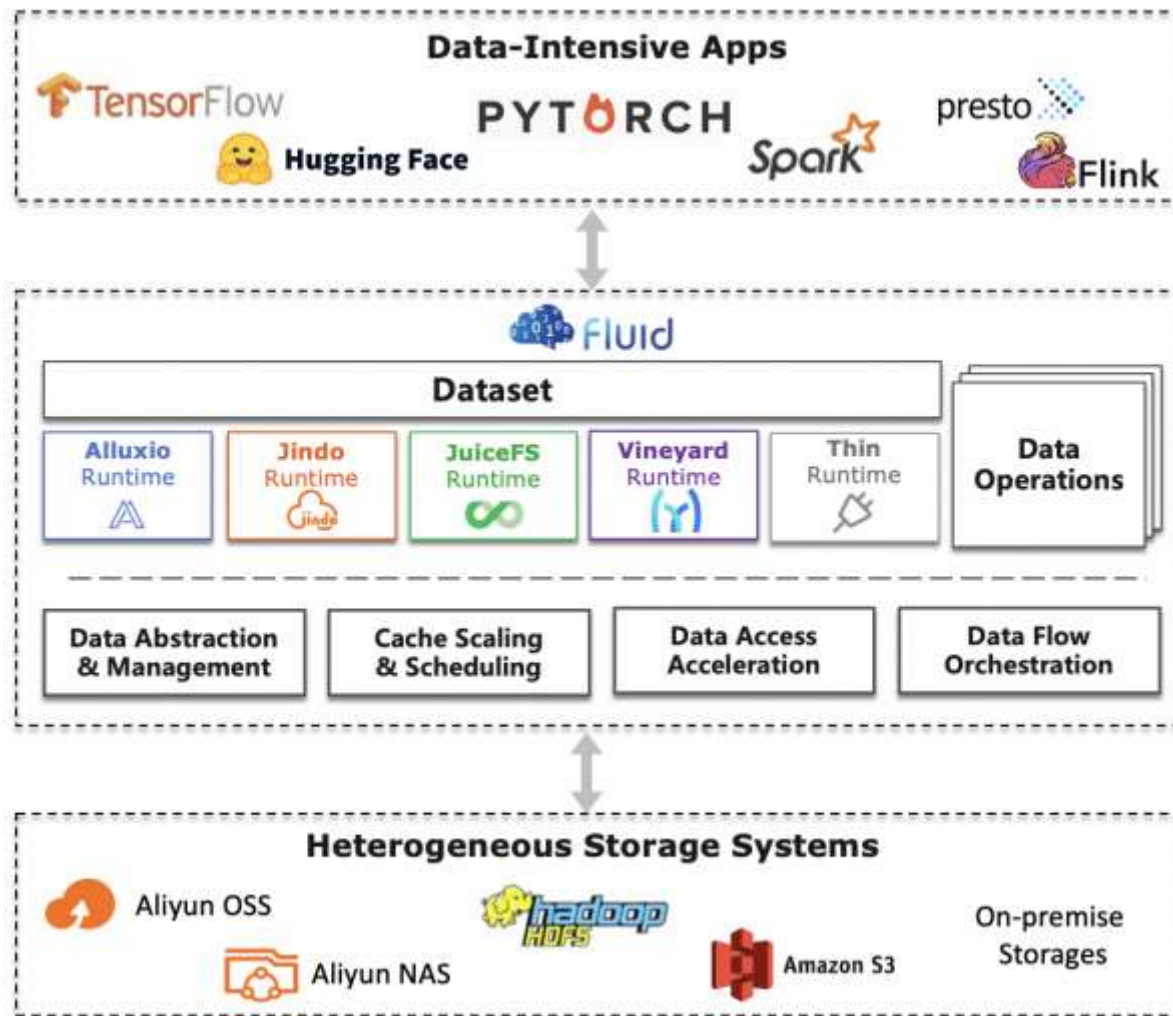
- 典型代表：Lustre, BeeGFS, GPFS, 等等
- 特点：
 - 提供高性能的并行数据
 - 适合高速数据读写



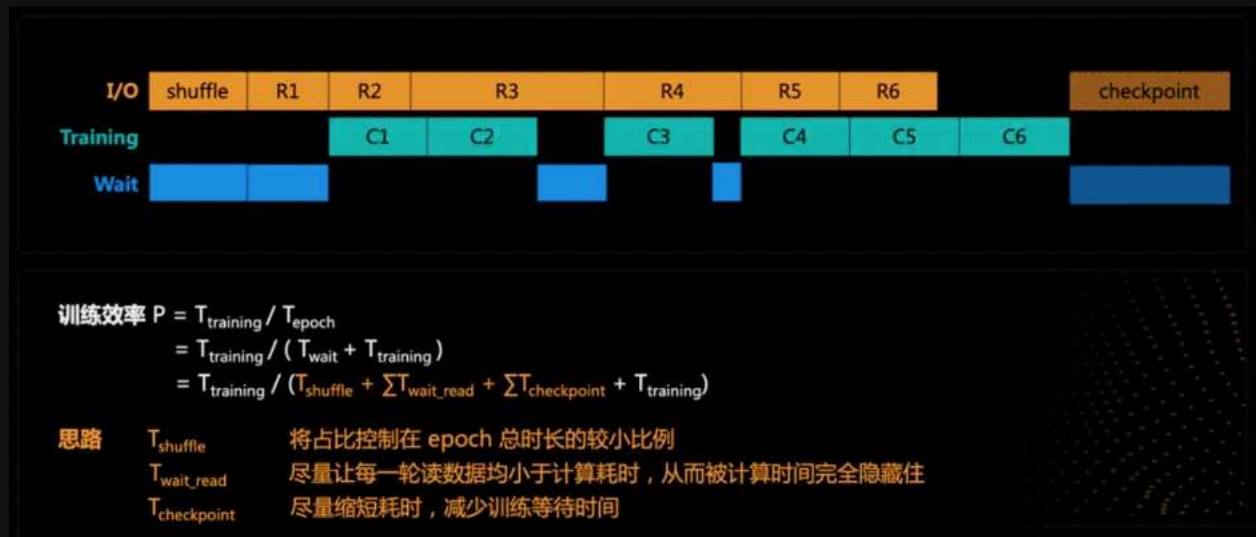
■ 通过 Fluid 实现存储方案一的加速部署

Fluid 的目标是为 AI 与大数据云原生应用提供一层高效便捷的数据抽象，将数据从存储抽象出来，以便实现：

- 通过数据亲和性调度和分布式缓存引擎加速，实现数据和计算之间的融合，从而加速计算对数据的访问。
- 将数据独立于存储进行管理，并且通过 Kubernetes 的命名空间进行资源隔离，实现数据的安全隔离。
- 将来自不同存储的数据联合起来进行运算，从而有机会打破不同存储的差异性带来的数据孤岛效应。



大模型训练过程中如何进行数据加速?



大模型训练数据访问特点:

1. 将数据集分片
2. 多轮次训练
3. 准备数据 -> 训练数据

数据加速方案:

1. 减少GPU空闲时间
2. 训练时可以加载/准备下一轮训练数据
3. 训练任务流水线化

Task1 和 Task2 竞争同一 GPU 资源,
且均需要从大容量存储/数据湖准备数据



调度策略 1: Task1 和 Task2 作为整体调度



调度策略 2: Task1 和 Task2 流水线化调度



通过 k8s fluid 组件 pipeline 数据准备

- 将数据集作为一个抽象实体进行管理
- 一个训练任务除了真正执行训练的部分外, 可能存在其它处理环节
- 通过让不竞争算力资源的环节流水线化, 可以提高计算资源利用率

■ 大模型训练过程中检查点 Checkpoint 保存

检查点是 AI 模型训练中不可或缺的一环：

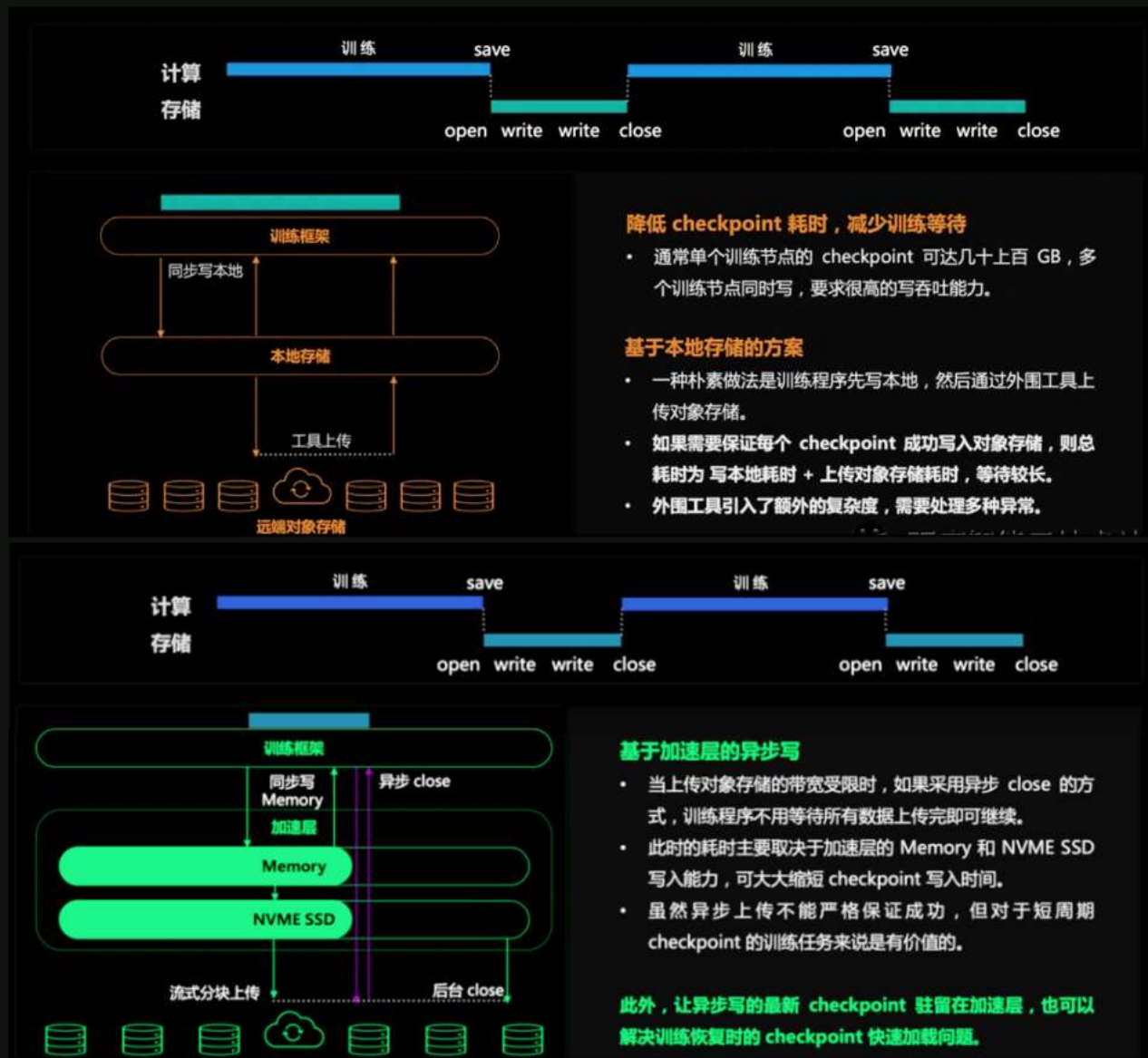
- 定期保存：模型权重、优化器状态等
- 用于 1) 确保训练进度；2) 模型调试评估；3) 监控模型训练过程

检查点保存过程：

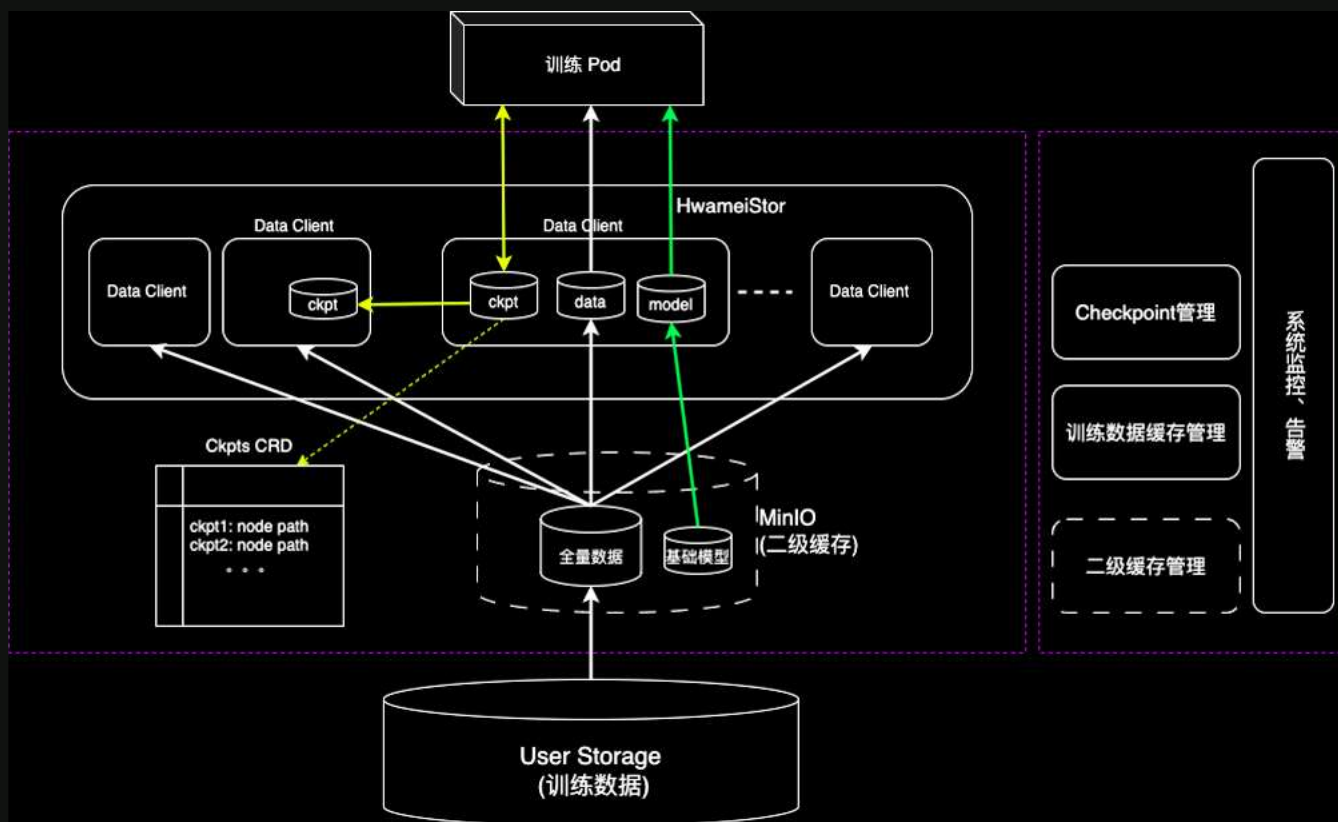
1. 暂停训练，模型状态从 GPU 转至系统内存；
2. 模型状态序列化；
3. 写入持久化存储；

特点：

- 周期性产生检查点，例如：30 分钟
- 数据量巨大、写入存储时间长
- 大多数检查点数据不需要长久保存



■ HwameiStor 如何利用本地磁盘加速 AI 训练



训练数据加速:

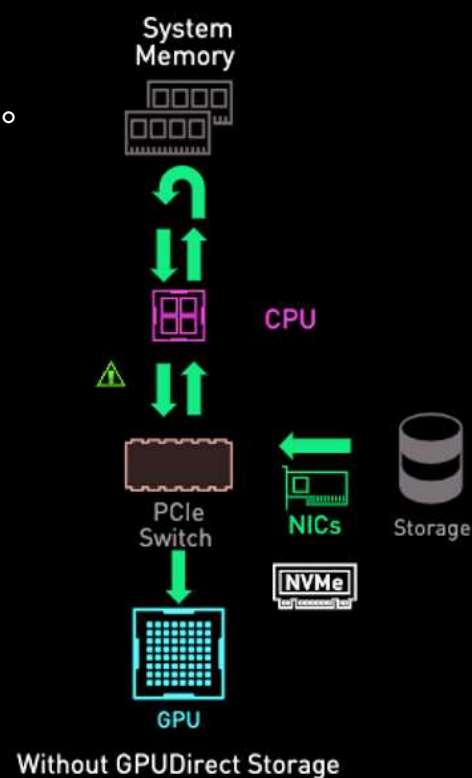
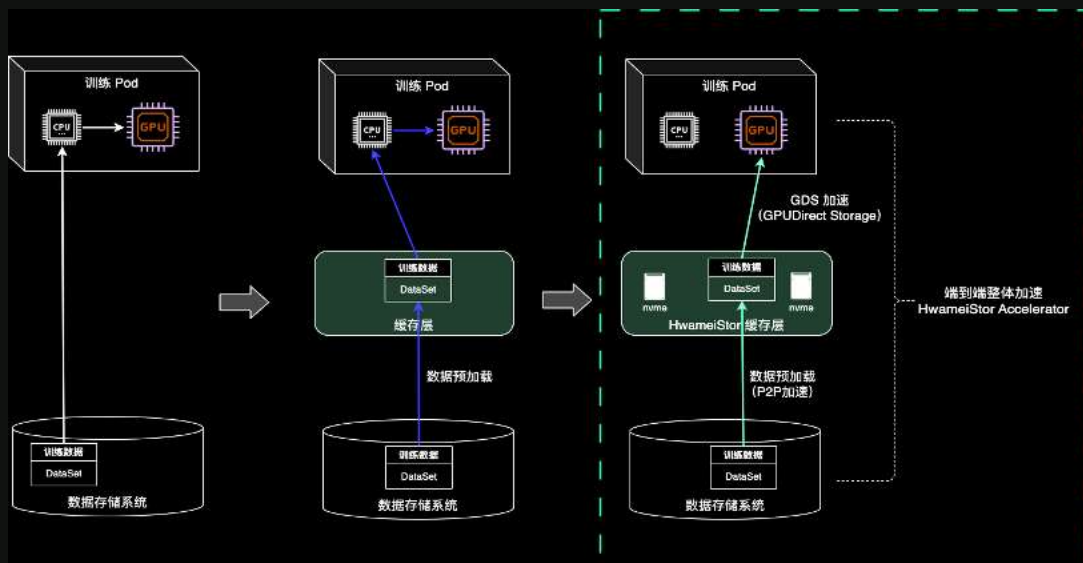
1. 采用 P2P 技术将远程数据集加载到本地磁盘
2. 训练任务从本地快速读取训练数据
3. 可以选择构建二级缓存, 避免多次加载远程数据

检查点保持加速:

1. 训练任务将检查点写入本地磁盘
2. 利用全局检查点 CRD 记录各个检查点信息
3. 可以将检查点异步保存至外部存储

■ HwameiStor 支持 GDS，实现数据访问进一步加速

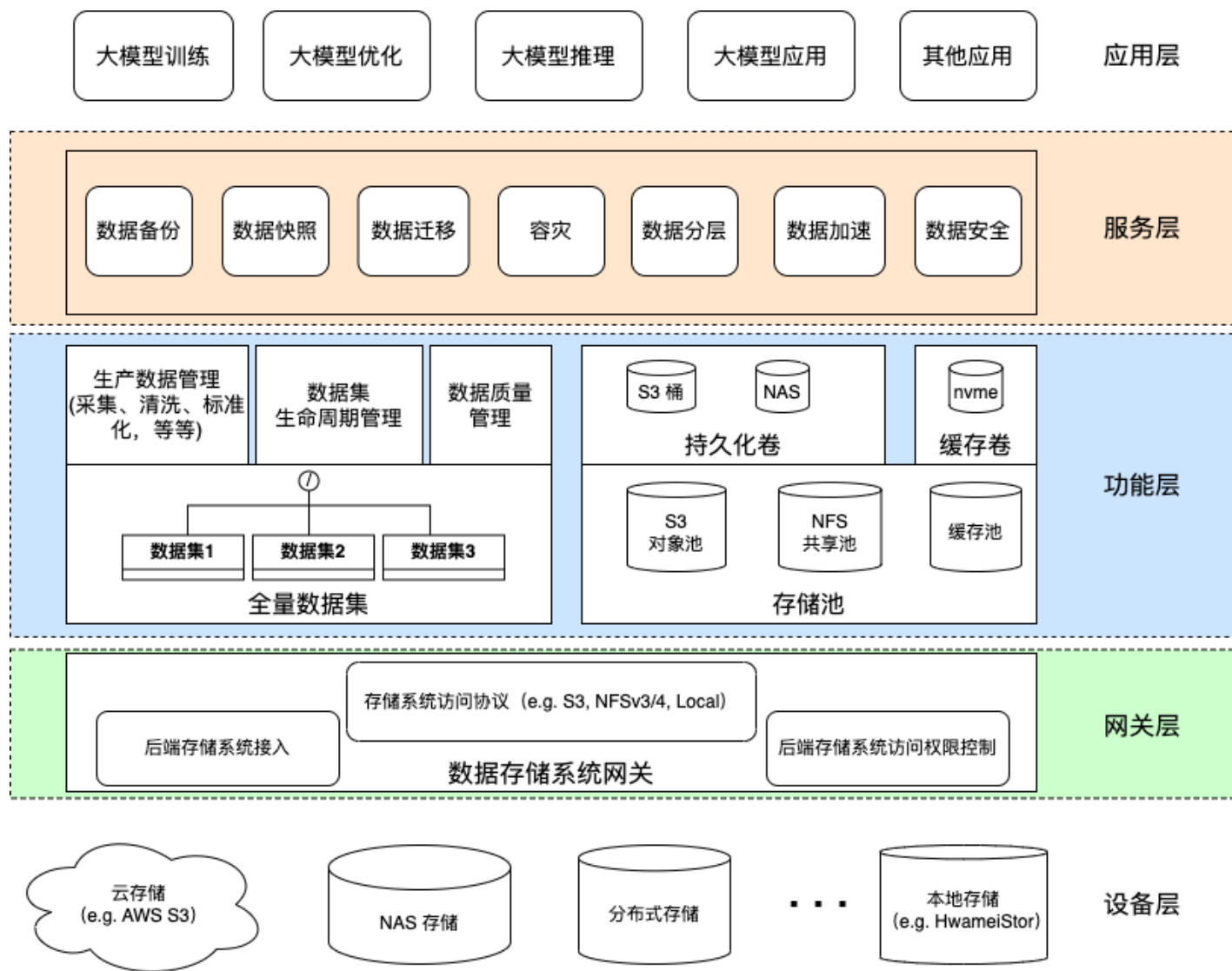
- NVIDIA 提出的优化 I/O 访问路径的一种方式
- 主要加速数据从存储系统到 GPU 显存之间读取和写入。
- 存储设备直通 GPU 显存，减少了数据的二次拷贝
- 使整个 I/O 链路绕行 CPU，从而降低了对 CPU 的依赖。



Part 03

AI 存储的发展

■ AI 存储展望 - 数据存储平台



- 将各种数据能力标准化，为各种场景应用提供标准的数据服务，包括 AI 应用、非 AI 应用

- 统一管理 AI 大模型训练数据集
- 为实时生产数据提供统一处理框架
- 提供数据质量管理框架
- 分别为持久化数据和临时数据提供存储能力

- 统一管理各种存储系统
- 统一接入云原生平台并使用
- 提供统一访问协议

Thanks.



欢迎扫码入群探讨并获取课件哦