GLake

蚂蚁高性能显存管理器

张锐
蚂蚁集团高级开发工程师
AI infra项目核心成员

DLRover
Deep Learning Rover

目录

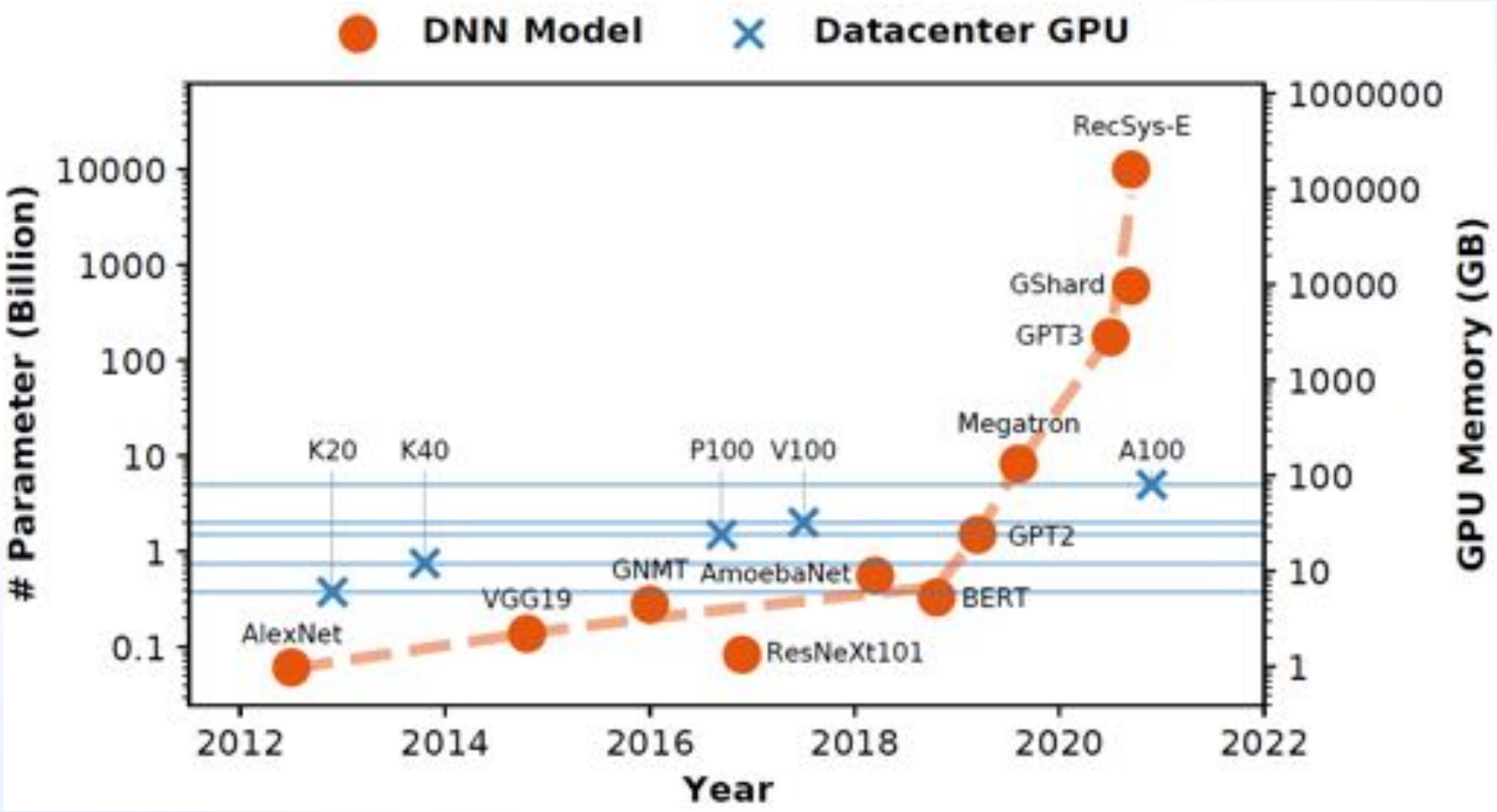
CONTENTS

DLRover
Deep Learning Rover

01. 项目背景

项目背景

大模型显存需求 vs 显存容量

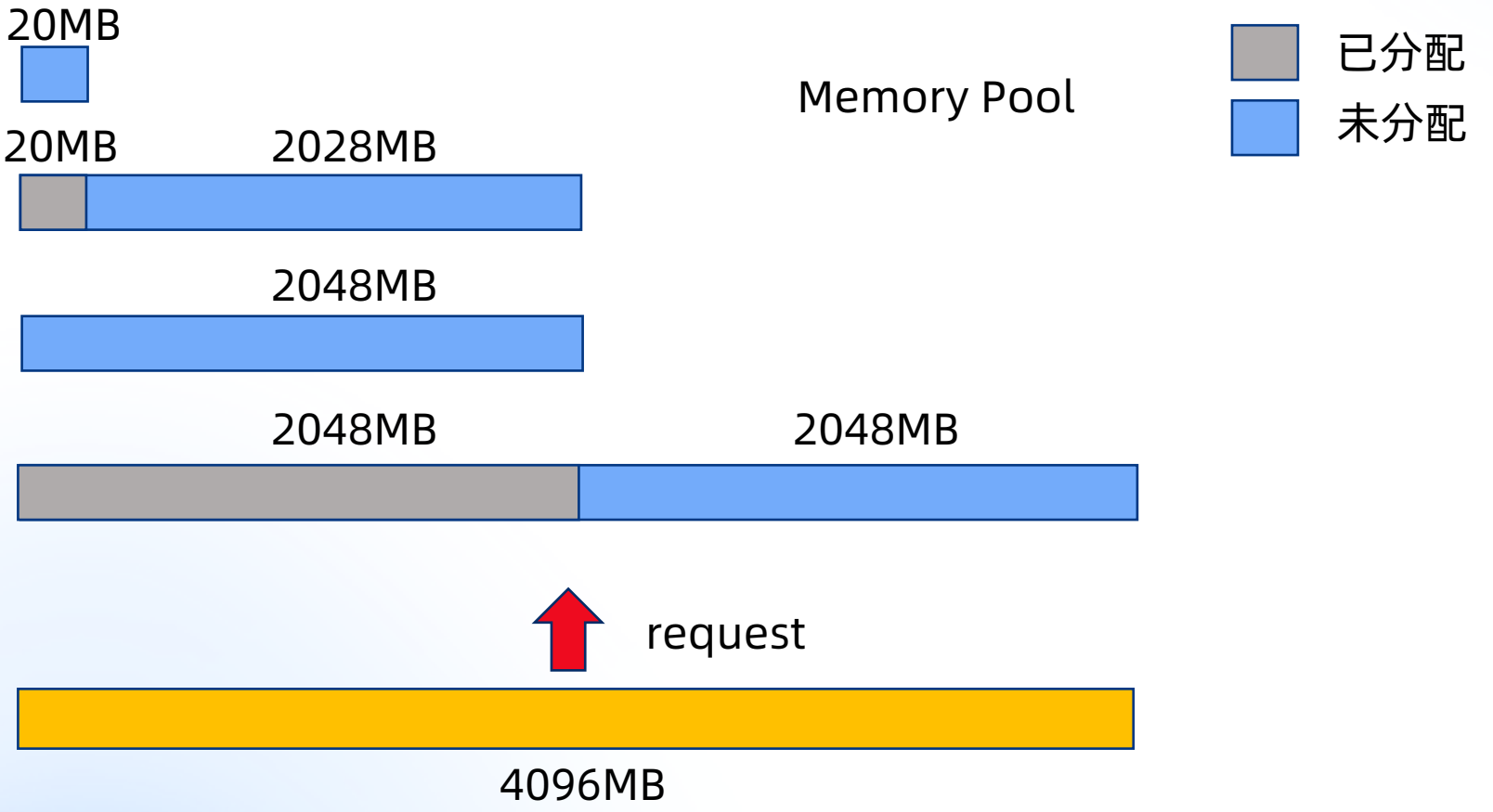


src: Harmony: Overcoming the hurdles of GPU memory capacity to train massive DNN models on commodity servers, VLDB, 2022

项目背景

Pytorch显存碎片

- 什么是显存碎片：系统中显存够用，但是分配不到
- 原因：显存地址空间连续性约束，剩余显存资源只有地址连续才能被使用



项目背景

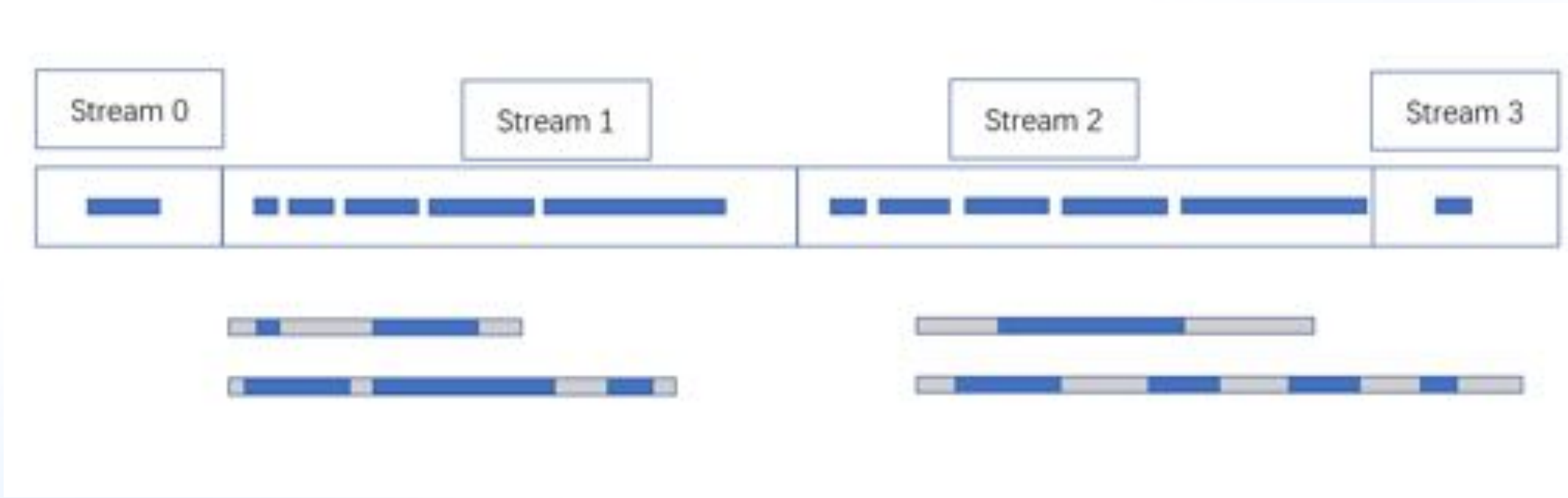
Pytorch分配方案

- **Size best fit：最开始从GPU中分配合适大小的显存块**

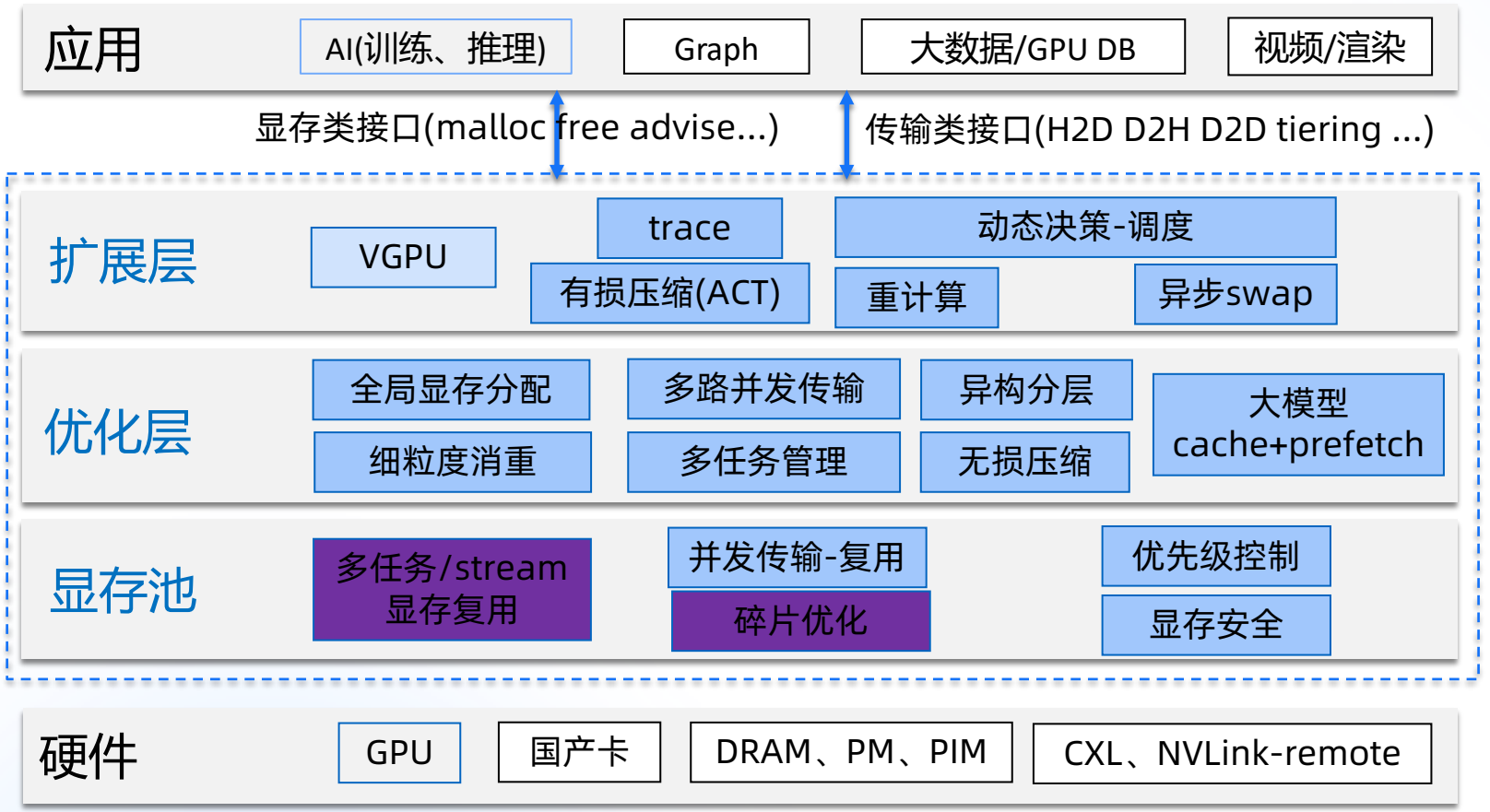
项目背景

Pytorch分配方案

- Size best fit：从已经还回显存池的块做split



GLake架构



- **解决的问题**
- 模型在训练和推理过程中碎片化导致显存没法高效利用的问题，也可能导致性能下降
- 多个stream之间显存使用不均衡导致显存浪费的问题

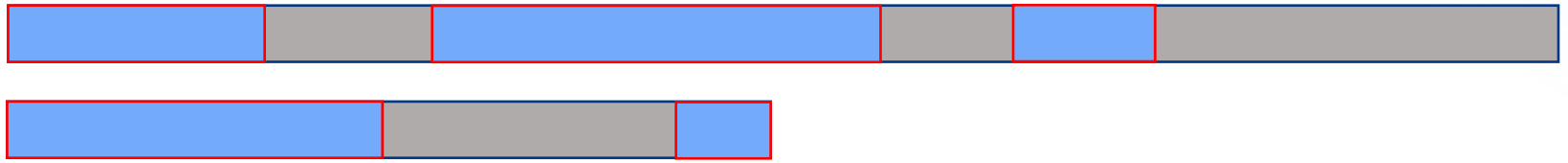
DLRover
Deep Learning Rover

02. GLake项目关键技术

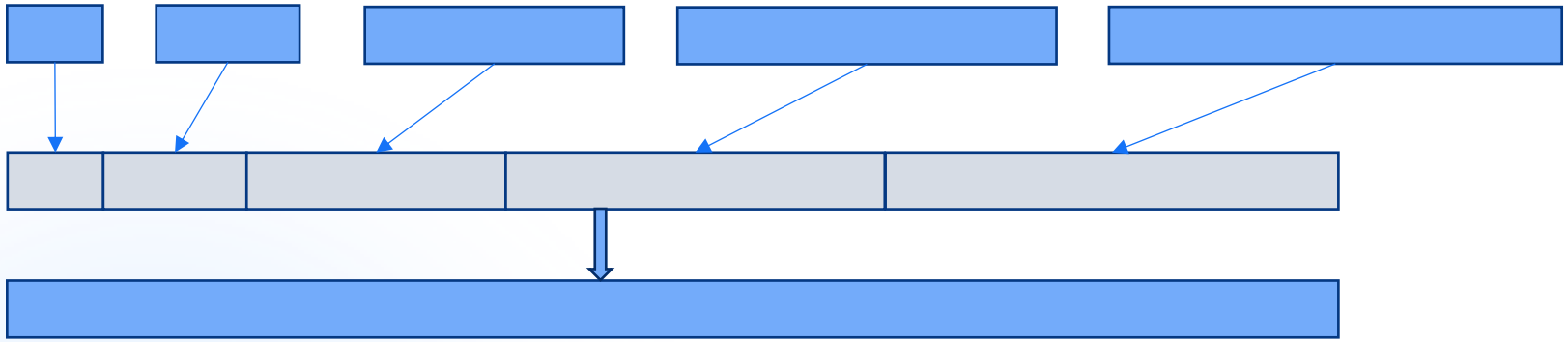
GLake关键技术

离散显存粘合

- 将多个不连续的显存粘合在一起



Original block Pool



Fused block Pool

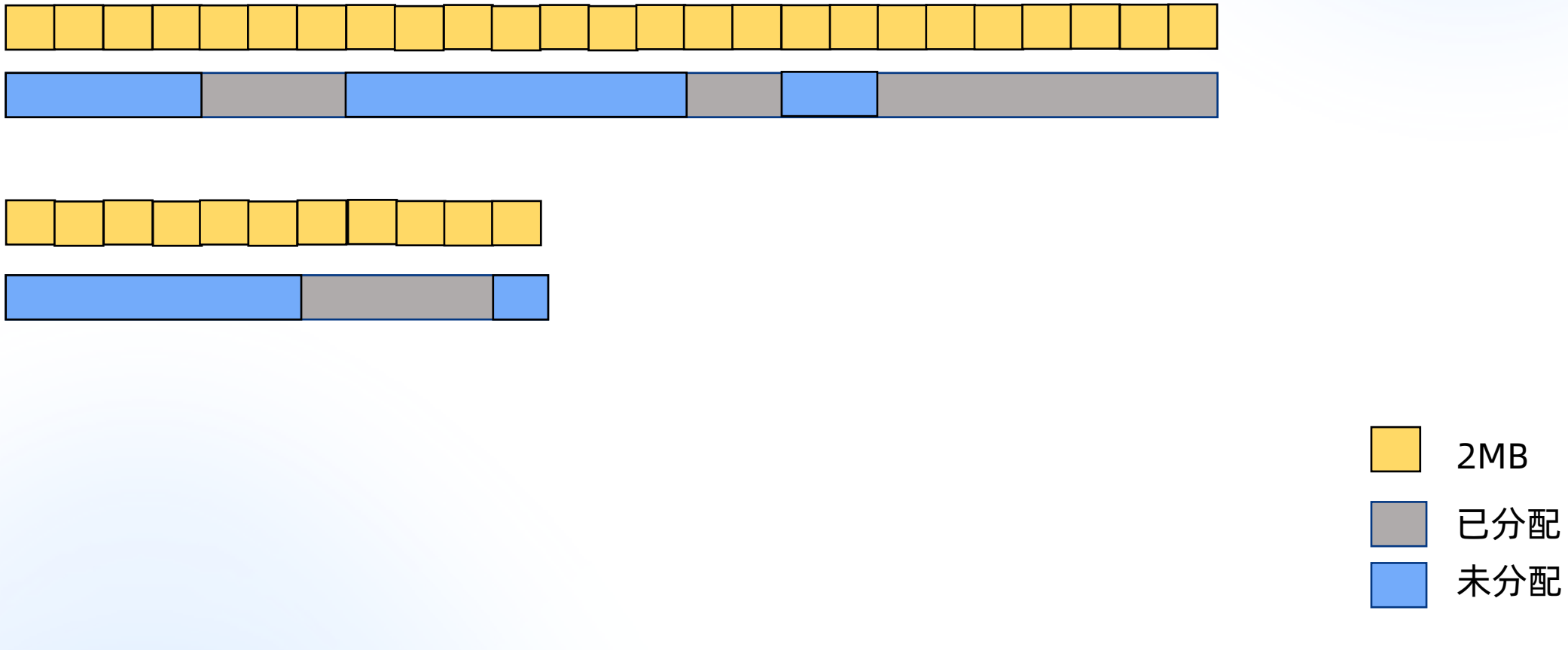
已分配

未分配

GLake关键技术

离散显存粘合

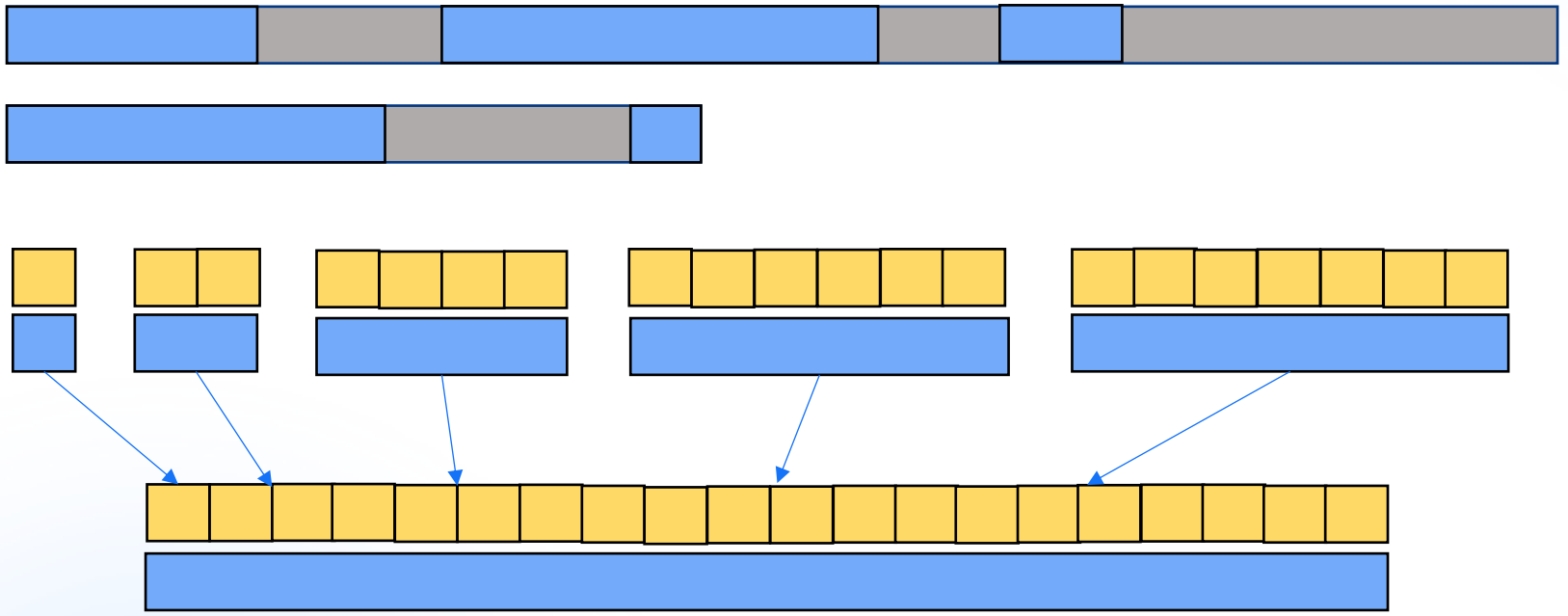
- 每个block由2MB粒度的若干个Physical Block组合在一起



GLake关键技术

离散显存粘合

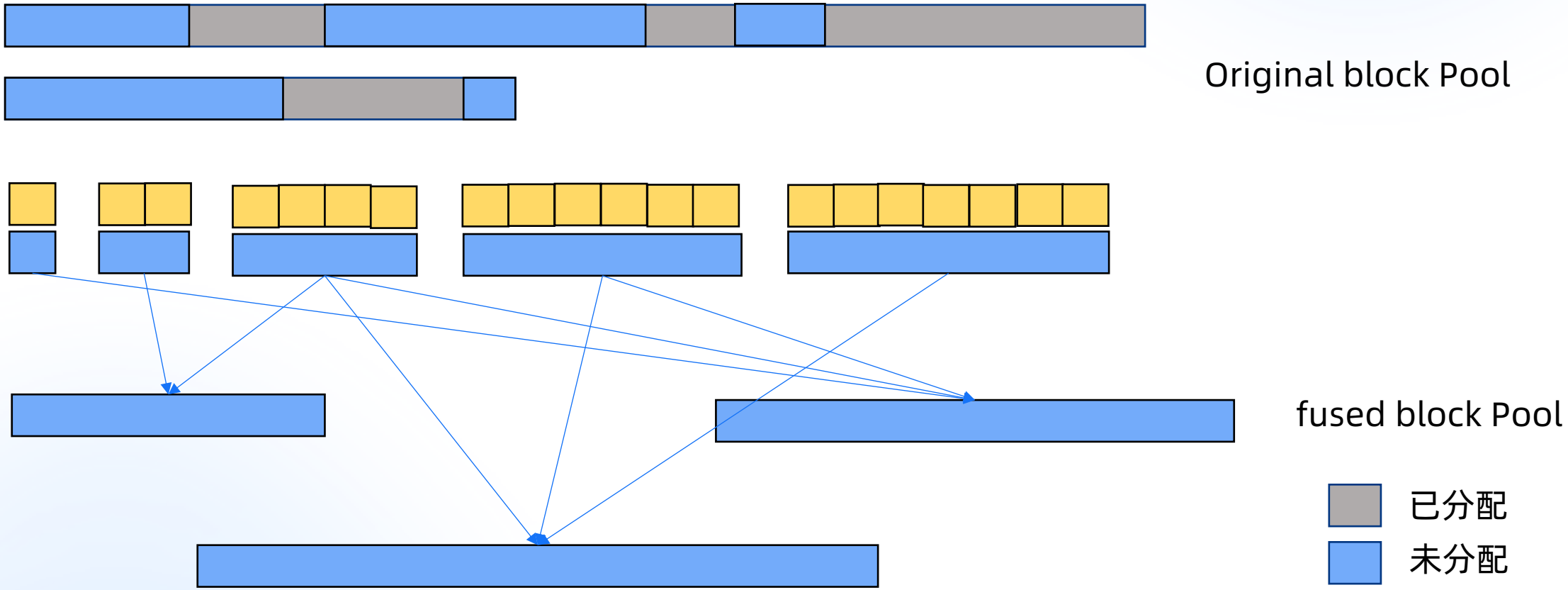
- 有显存碎片时，收集空闲Physical Block，映射到新的虚拟地址



GLake关键技术

Physical Block分时复用

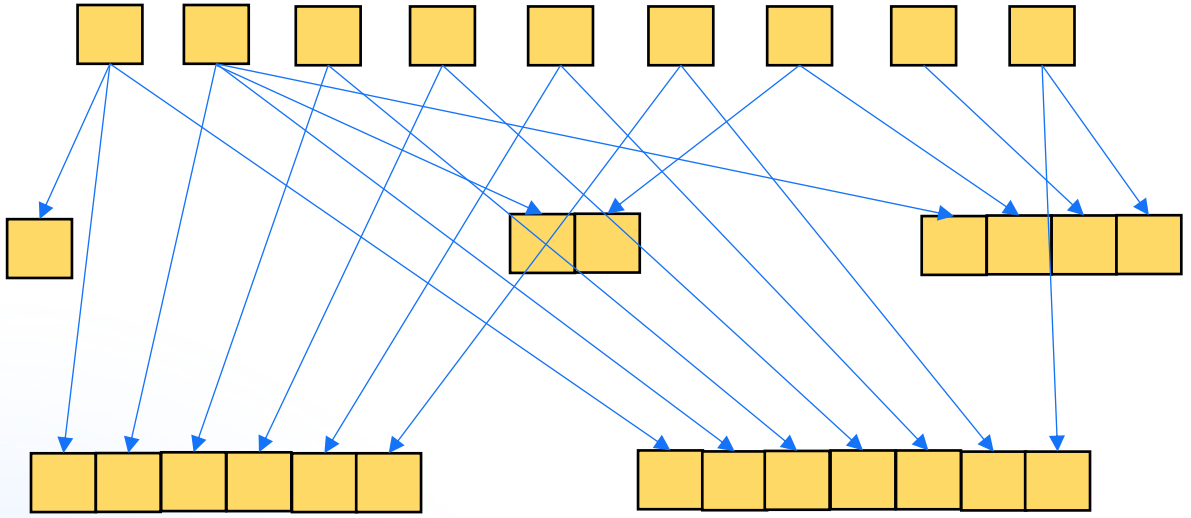
- 多个虚拟地址共享若干个Physical Block，做到高效分时复用，有效减少显存占用



GLake关键技术

Physical Block分时复用

- 站在Physical Block角度来看：打破了地址空间连续性约束，显存作为一种资源在空间上以不同的形态随意组合



模型测试

Pytorch1.13.1

DLRover
Deep Learning Rover

- Stable diffusion推理实测

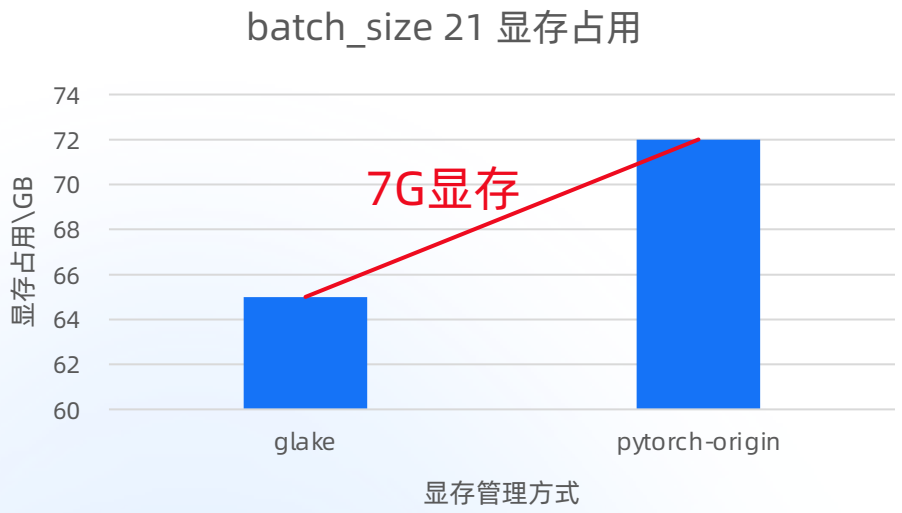
	Peak Allocated memory	Peak Reserved memory	Fragments
pytorch allocator	12564 MB	16978 MB	4414 MB(26%)
our allocator	12564 MB	12580 MB	16 MB(0.1%)

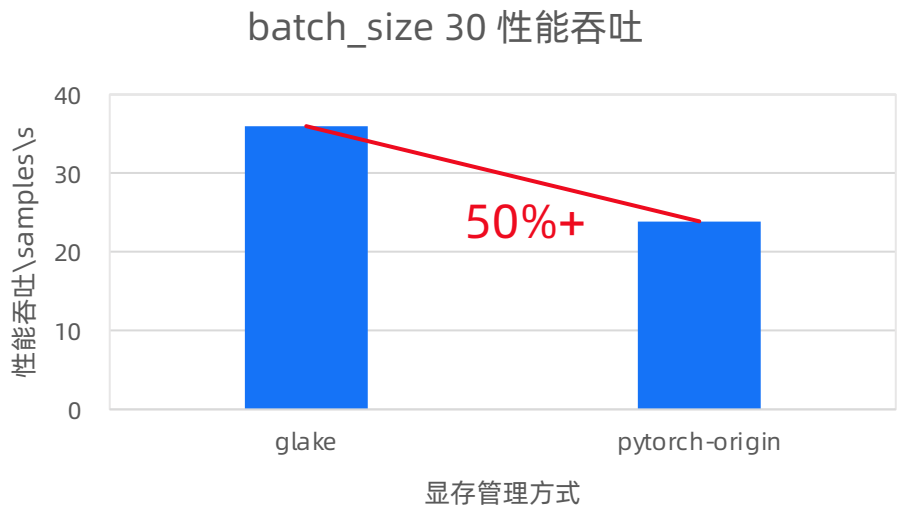
	峰值显存	时间性能
off	16978 MB	24608.35ms/step
on	12580 MB	23670.07ms/step

模型测试

Pytorch1.13.1

- **LLM百亿模型**
  - **测试环境**： 8卡A100 80G
  - 开启pytorch FSDP

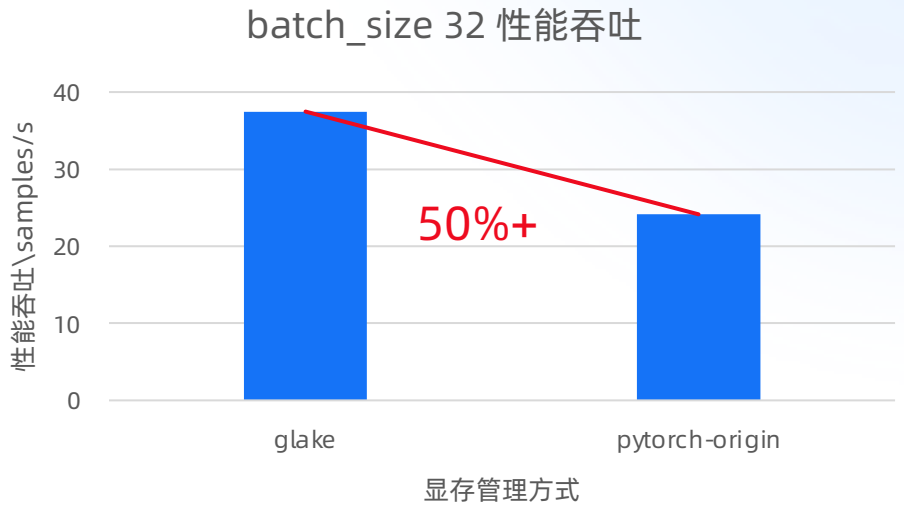


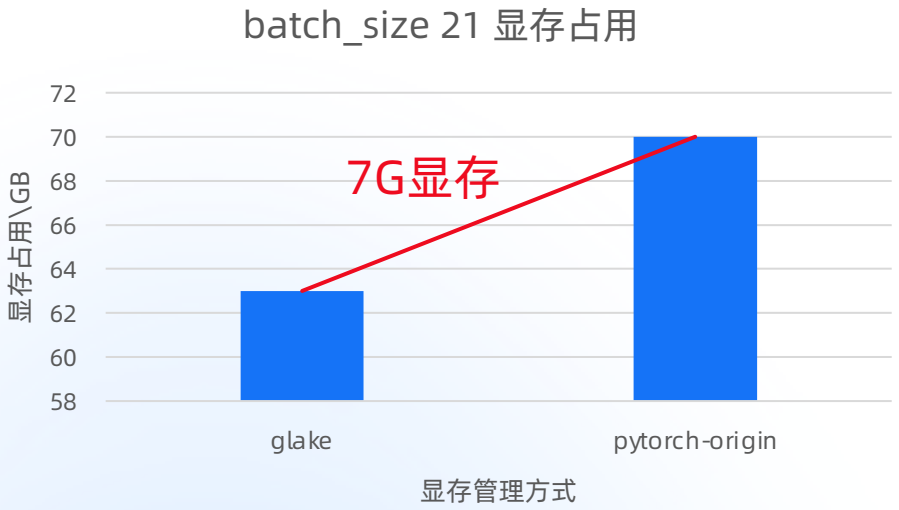


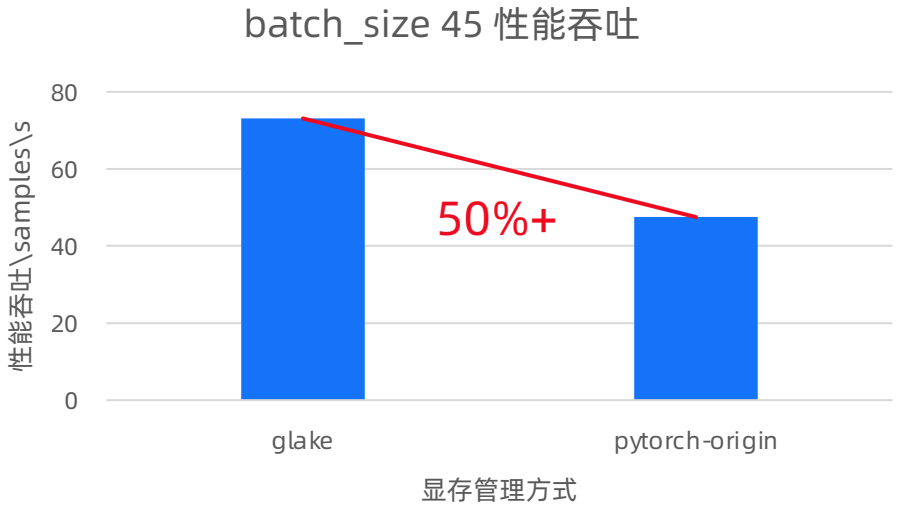
模型测试

Pytorch2.0

- **LLM百亿模型**
  - **测试环境**： 8卡A100 80G
  - 开启pytorch FSDP





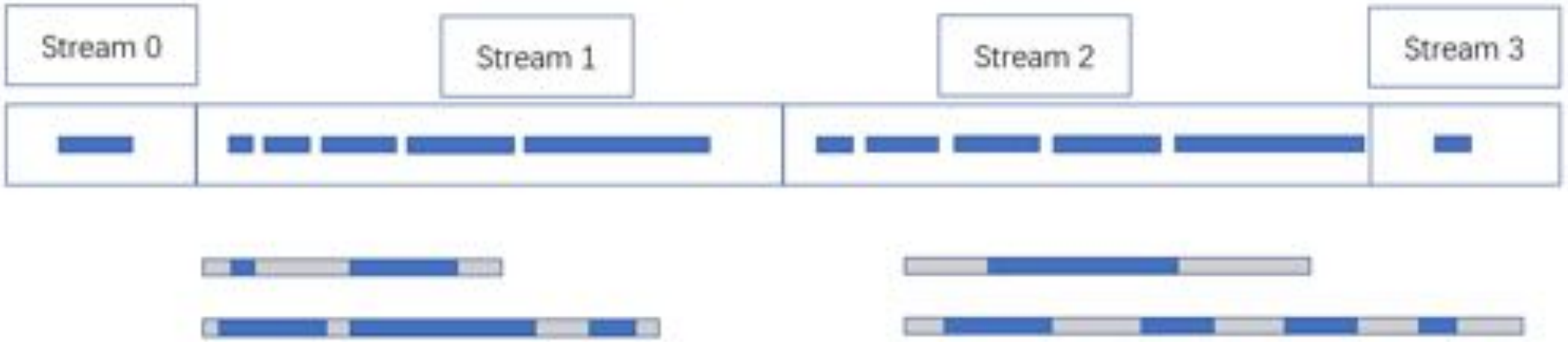


GLake关键技术

多stream之间显存复用

DLRover
Deep Learning Rover

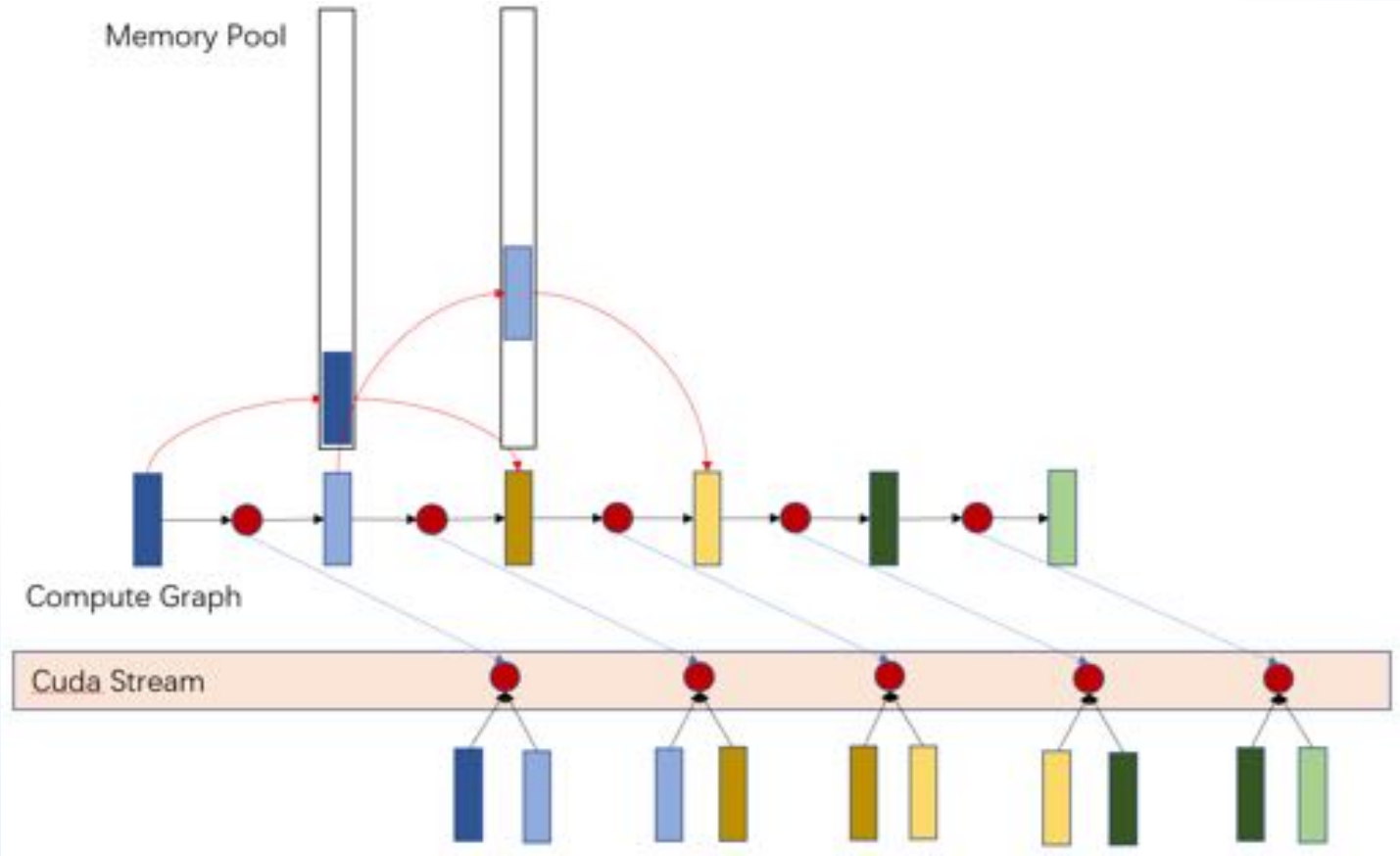
- **Pytorch当前实现：完全隔离不同stream间的显存block**
  - **优点**： 绝对安全，不会发生数据race
  - **缺点**：多个stream之间无法互相复用，当一个stream中剩余较多显存时另一个stream无法复用，导致显存浪费，可能会导致产生很多gc操作，影响性能



GLake关键技术

多stream之间显存复用

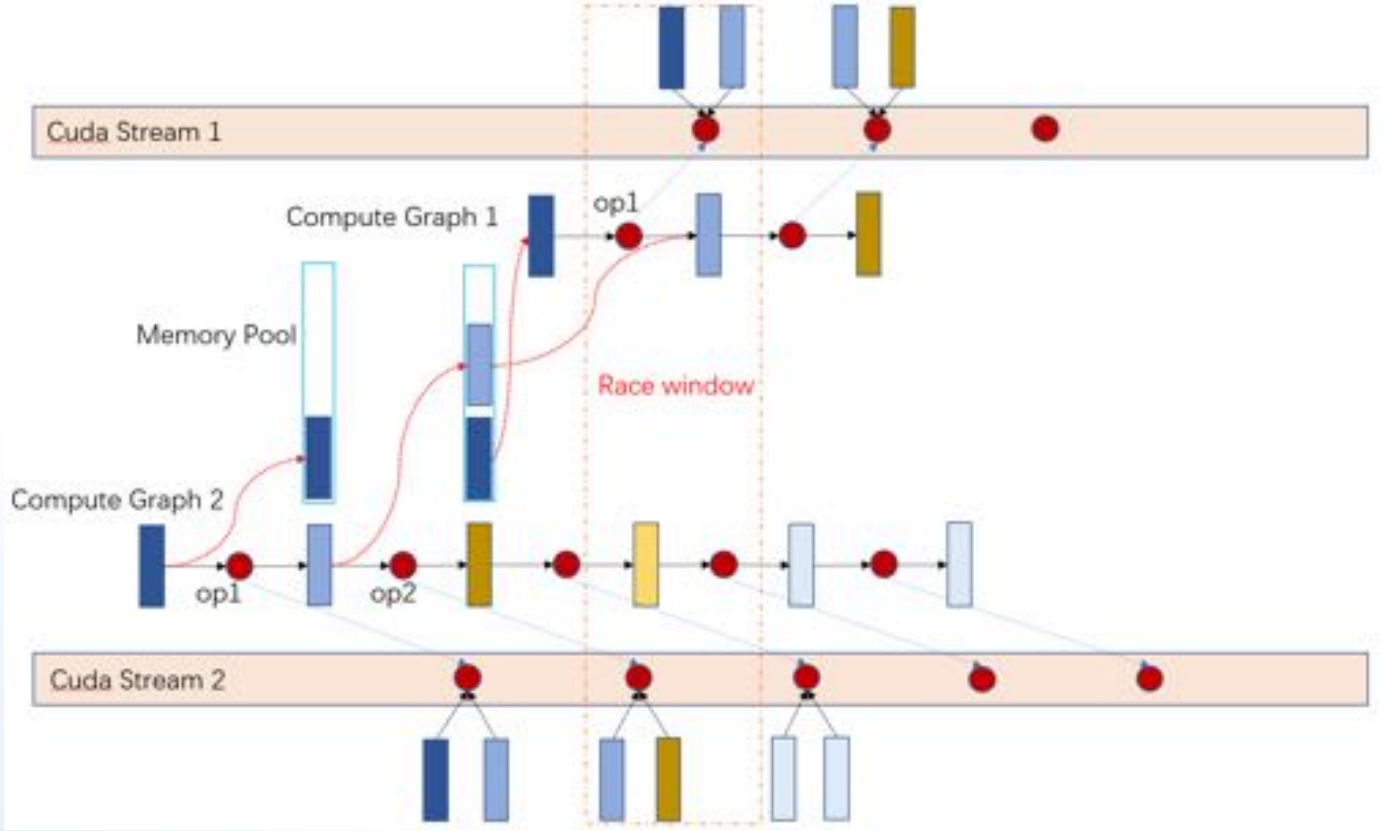
- Pytorch当前实现：相同stream内部的显存可以异步复用



GLake关键技术

多stream之间显存复用

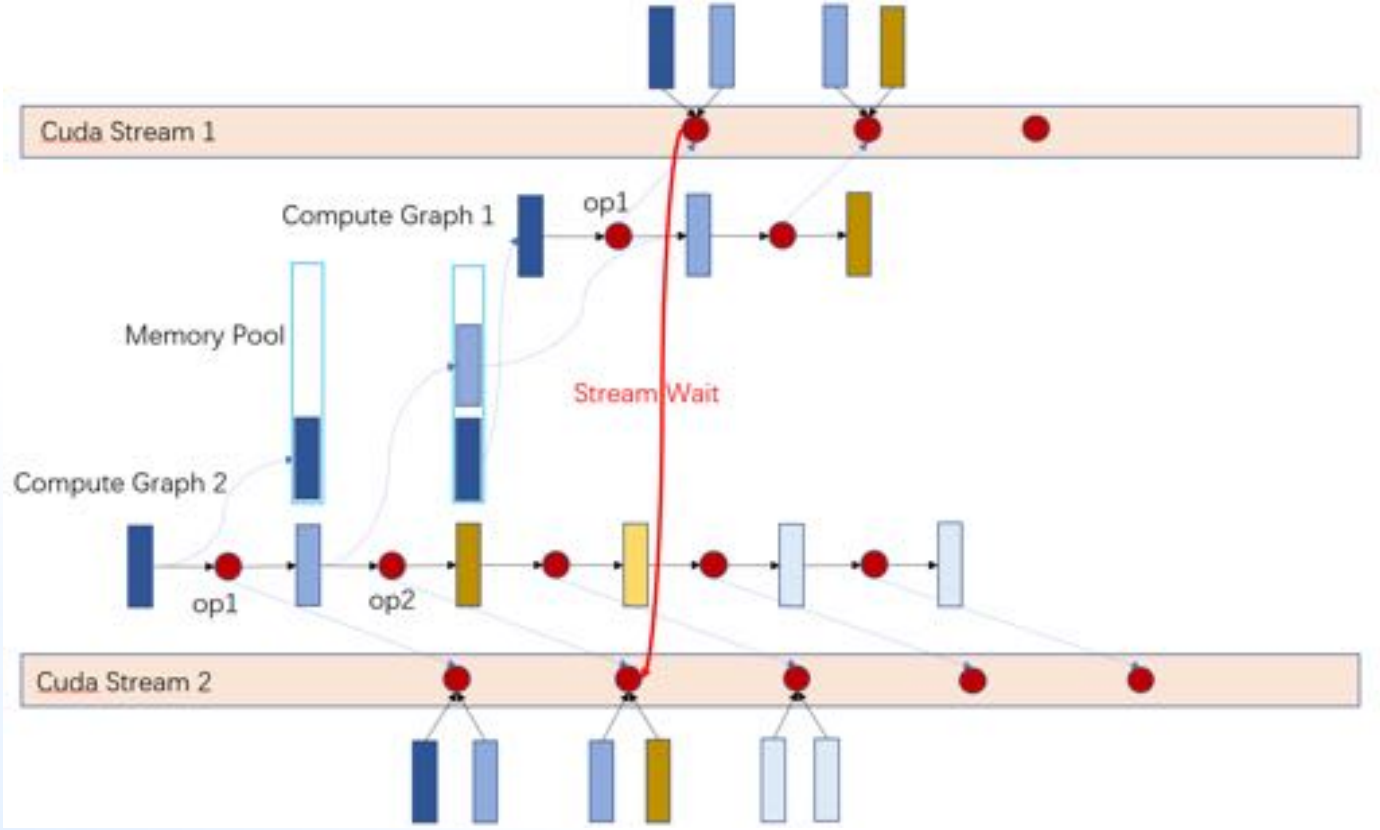
- 不同stream之间的显存异步复用会发生数据race



GLake关键技术

多stream之间显存复用

- 安全借用other stream的显存：借用stream去异步wait源stream中释放该显存块的op



GLake关键技术

多stream之间显存复用

- 将block按照时间顺序排列：单个stream的block按照释放顺序粘和

GLake关键技术

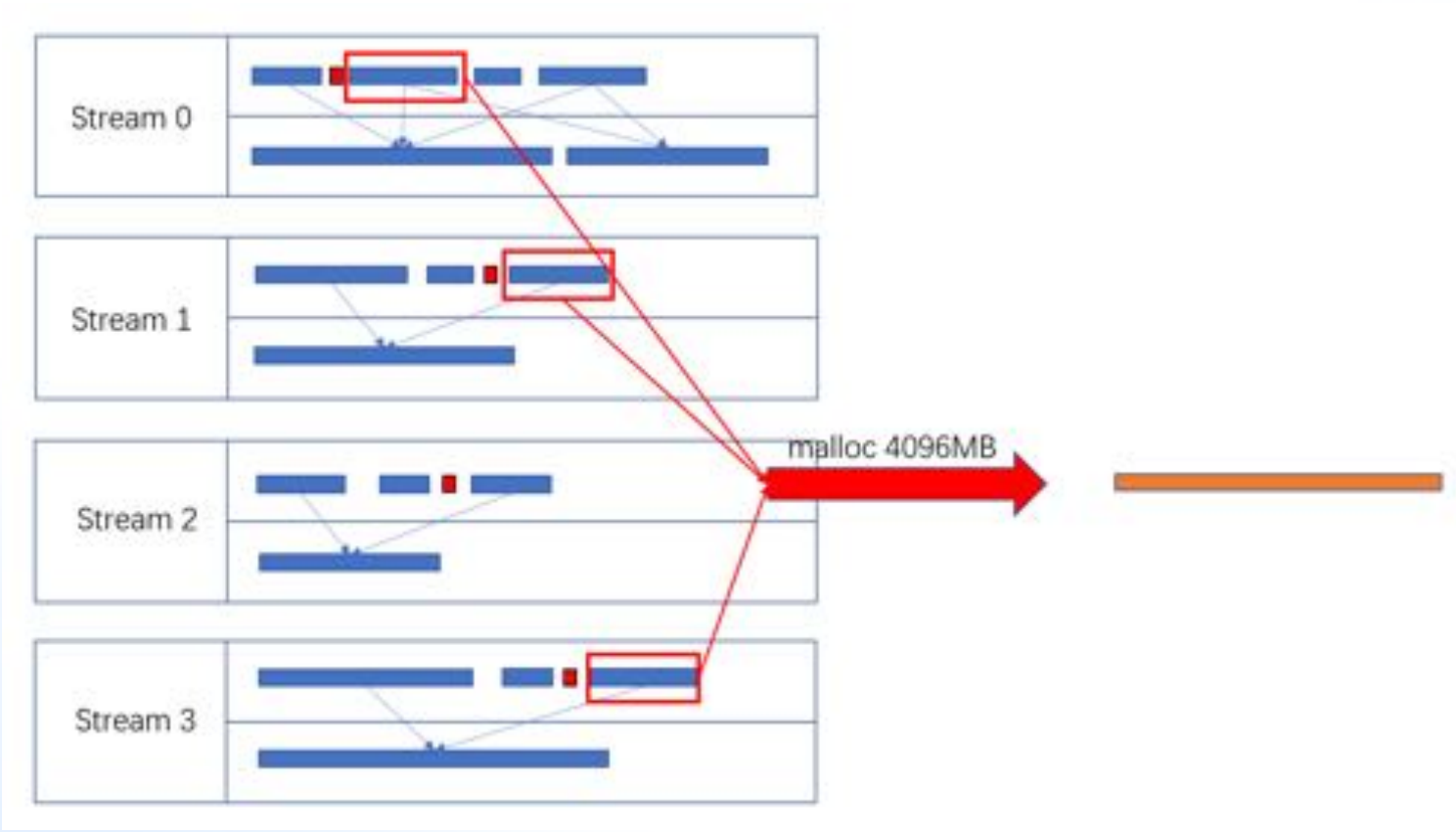
多stream之间显存复用

- 将block按照时间顺序排列：other stream中event已经done的block

GLake关键技术

多stream之间显存复用

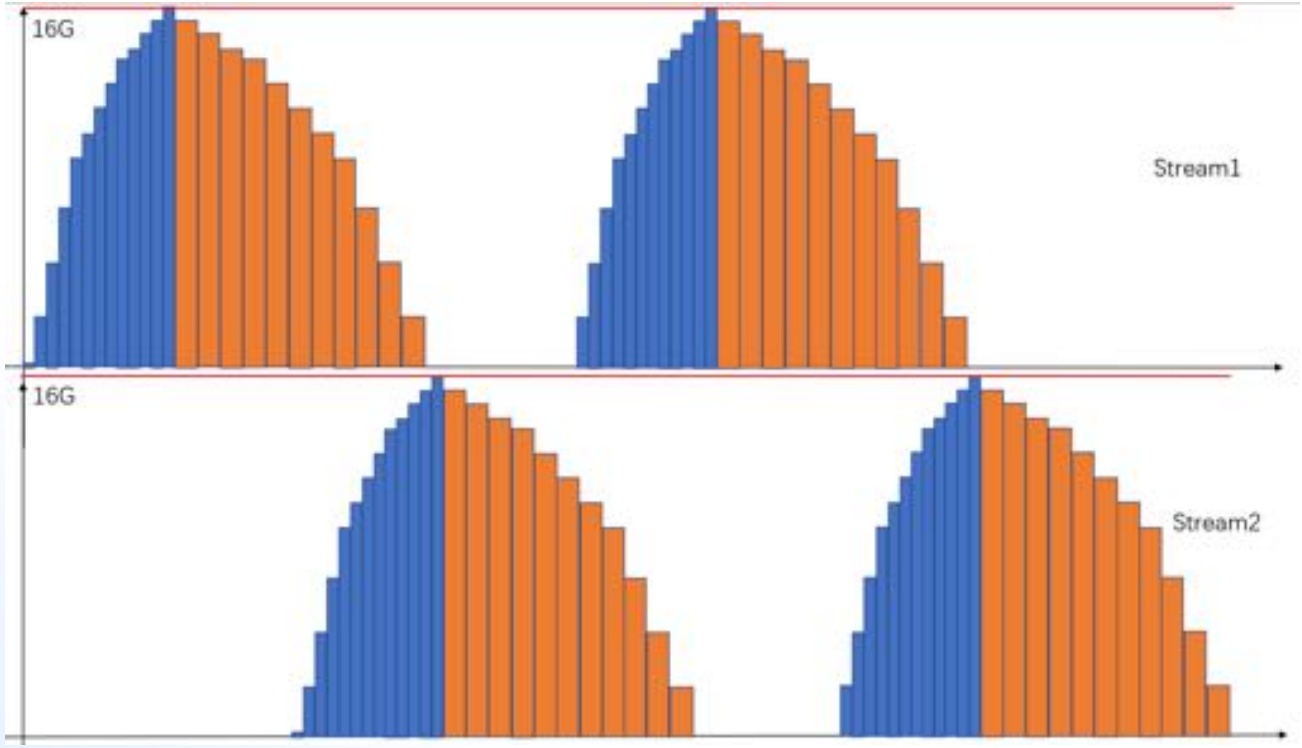
- 将block按照时间顺序排列：other stream中的event还没有done的block，加event依赖做同步



GLake关键技术

多stream之间显存复用

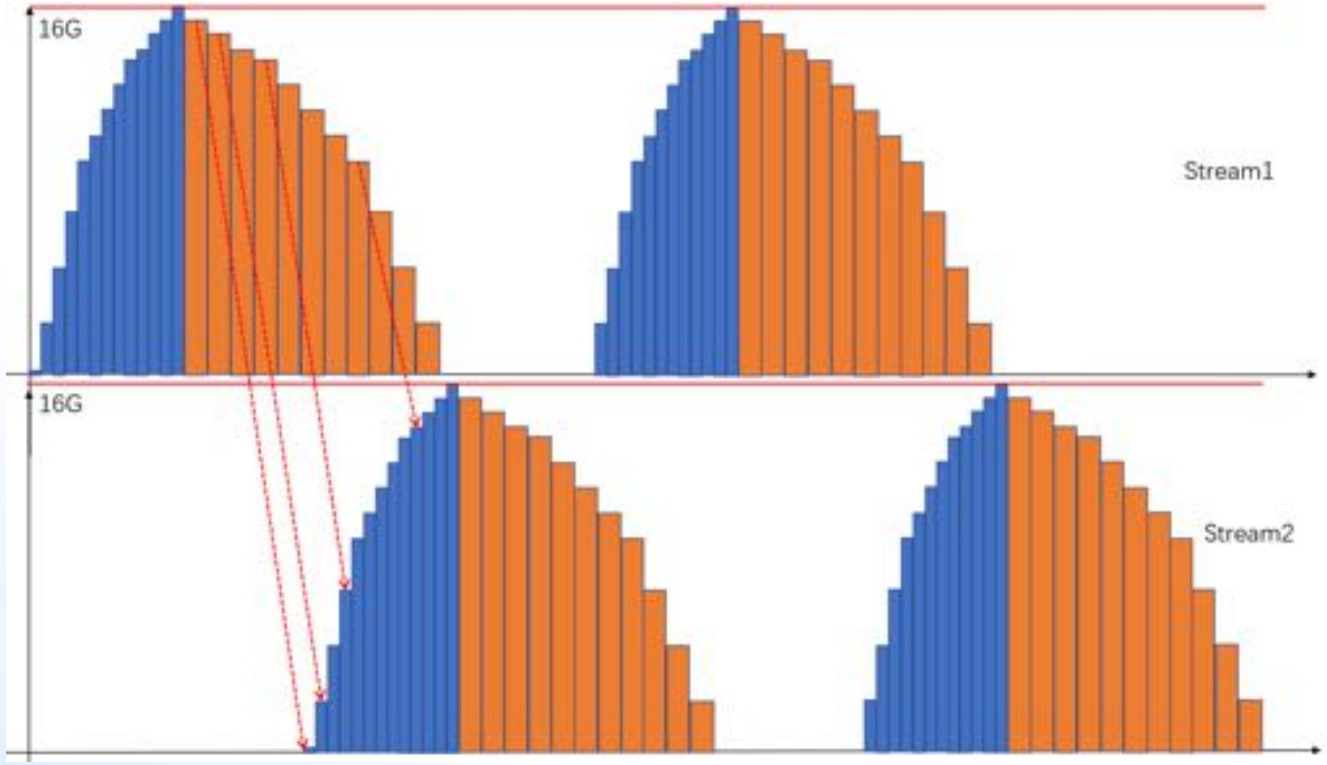
- **验证场景**
  - Wavelet场景： WAVELET: EFFICIENT DNN TRAINING WITH TICK-TOCK SCHEDULING, MLSYS21
  - Idea：显存动态变化，不是一直占满，利用训练过程中显存的动态变化产生的波峰波谷做overlap



GLake关键技术

多stream之间显存复用

- **验证场景**
  - 两个stream并发跑两个batch的训练，每路都可以几乎占满显存，一路训练的forward借用另一路训练backward释放的显存



GLake关键技术

多stream之间显存复用

- **验证场景**
  - Wavelet测试数据：huggingface bert-base-cased两路训练

	显存占用	训练吞吐量
单路batch=12	14.76G	1.136 batch/sec
两路batch=12 wavelet并发	15.10G	1.294 batch/sec

GLake关键技术

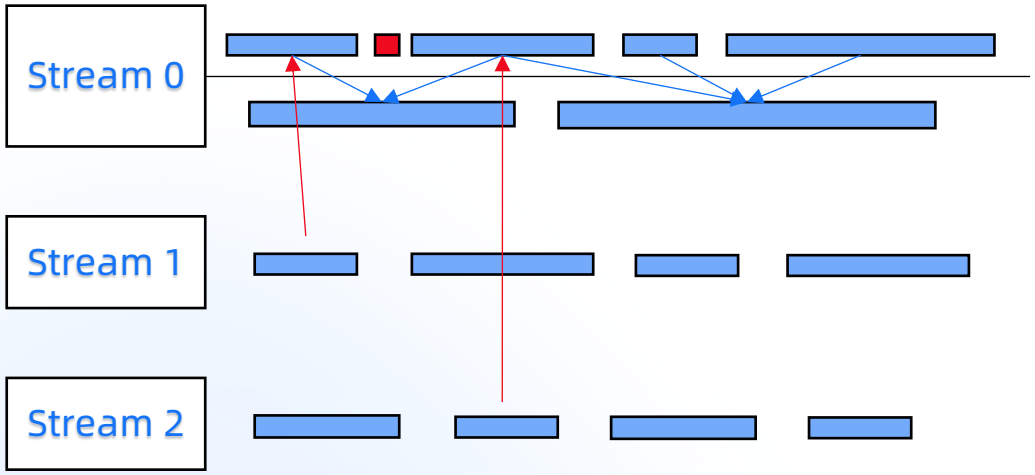
多stream之间显存复用

- **多stream之间显存互相复用**
  - **优点**：更高效利用显存
  - **缺点**：多个stream之间互相借用时，会频繁调用event wait接口，影响训练性能
- **模型训练观察**
  - Deepspeed、fsdp等训练框架中存在的分配显存的stream数量较少
  - 主要都是stream 0在分配显存，其他stream中单个stream内显存分配不太均衡
  - 其他stream主要用于通信stream中使用，分配规律以及使用频繁

GLake关键技术

多stream之间显存复用

- Other stream来复用stream 0中的显存
- 优先复用stream 0中event已经done的block
- Event仍然没有done的block，加event做依赖同步



GLake关键技术

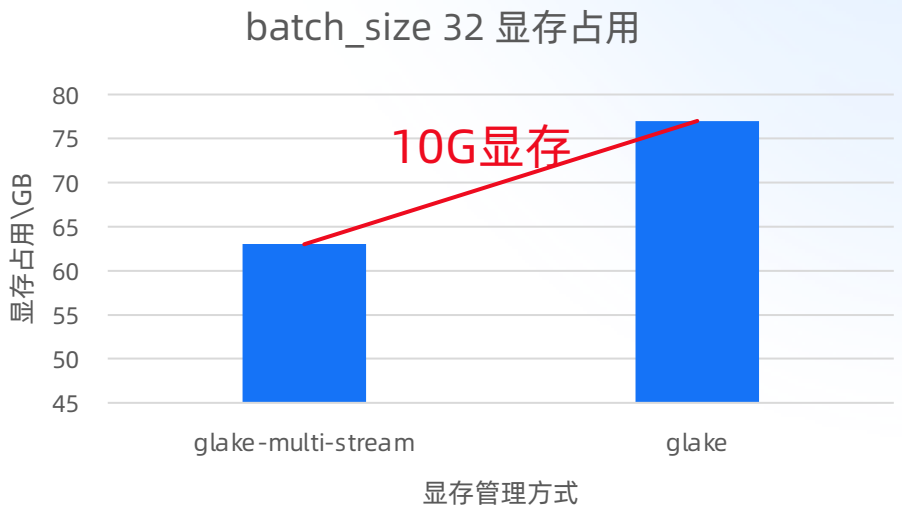
多stream之间显存复用

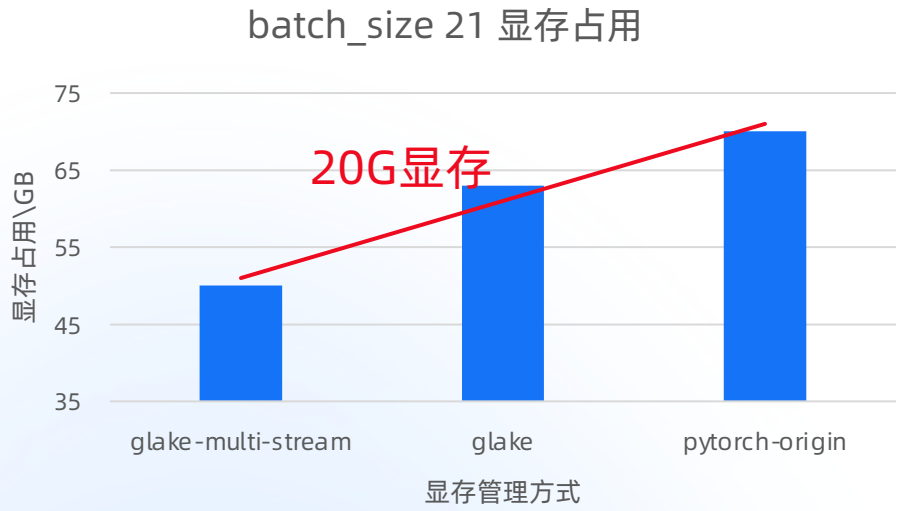
- **复用策略**
  - 选择较早释放的block，保证wait时间相对较短
  - Other stream不允许拼接stream 0中的空闲Physical block
  - 限制Other stream复用stream 0中的block，当空闲差值较大时不允许复用
  - Other stream复用stream 0中的block不允许做split
  - Fuse block时限制使用other stream中的空闲block
  - Other stream中的block限制使用fuse好的block

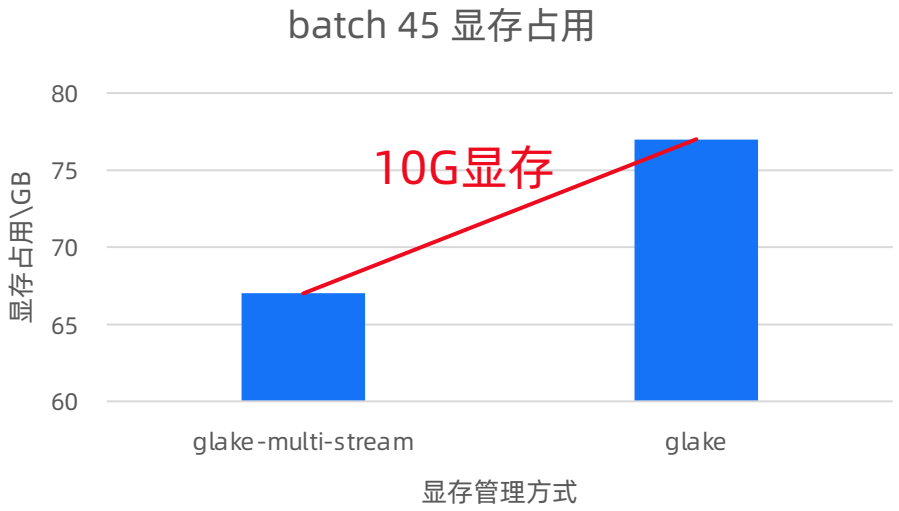
模型测试

Pytorch2.0

- **LLM百亿模型**
  - **测试环境**： 8卡A100 80G
  - 开启pytorch FSDP







DLRover
Deep Learning Rover

03. GLake项目展望

GLake项目展望

DLRover
Deep Learning Rover

- **GLake性能优化**
  - 完善GLake组件中的碎片拼接功能，可以节约更多的显存，针对多个模型可以减少碎片拼接的开销
  - 进一步完善GLake组件中的多stream借用block的功能，节约更多的显存，帮助模型增大batch_size，提高吞吐
  - 结合多个分布式训练框架的特点，优化GLake在模型训练和推理中的显存占用，进一步优化性能

- **GLake开源/开放**
  - 作为DLRover的一部分，计划今年Q3开源
  - 相关技术细节最终会整理成论文


DLRover
THANKS