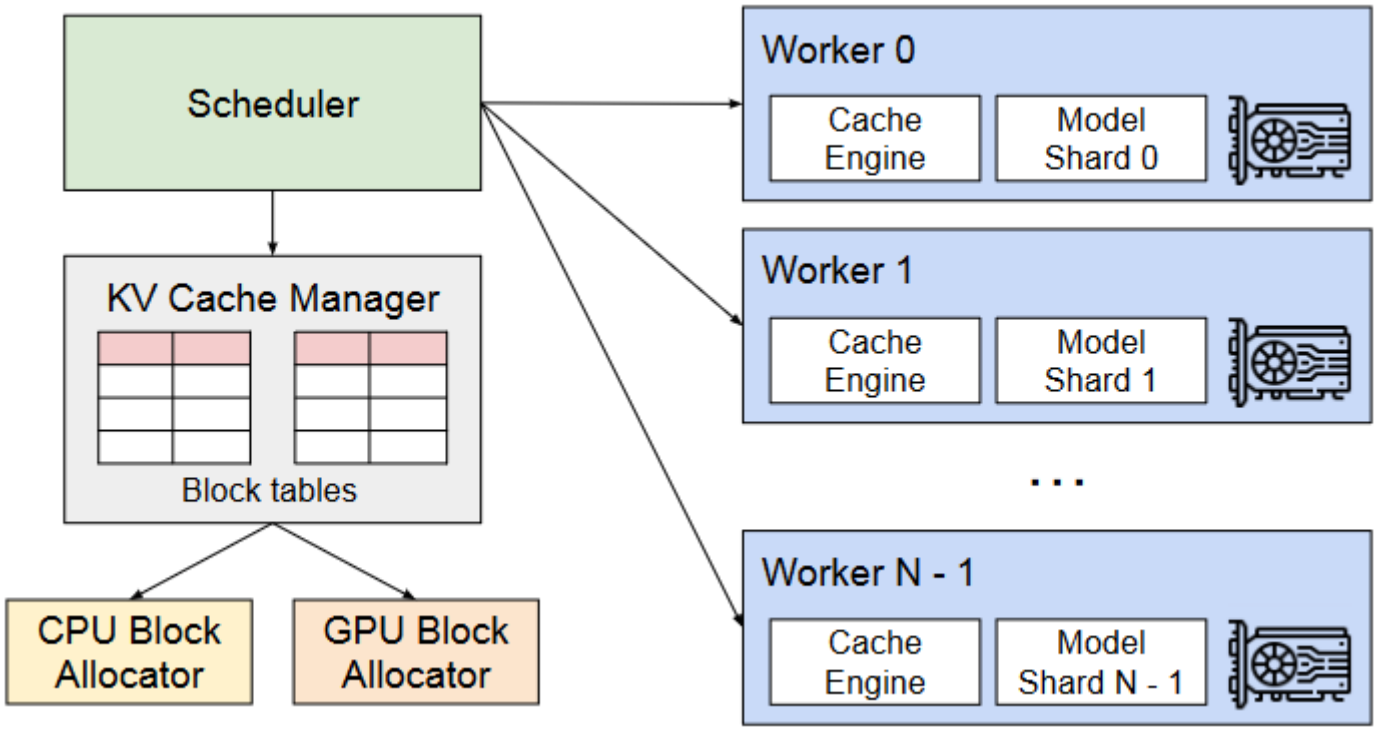
Pageattention

6.3.3 PagedAttention

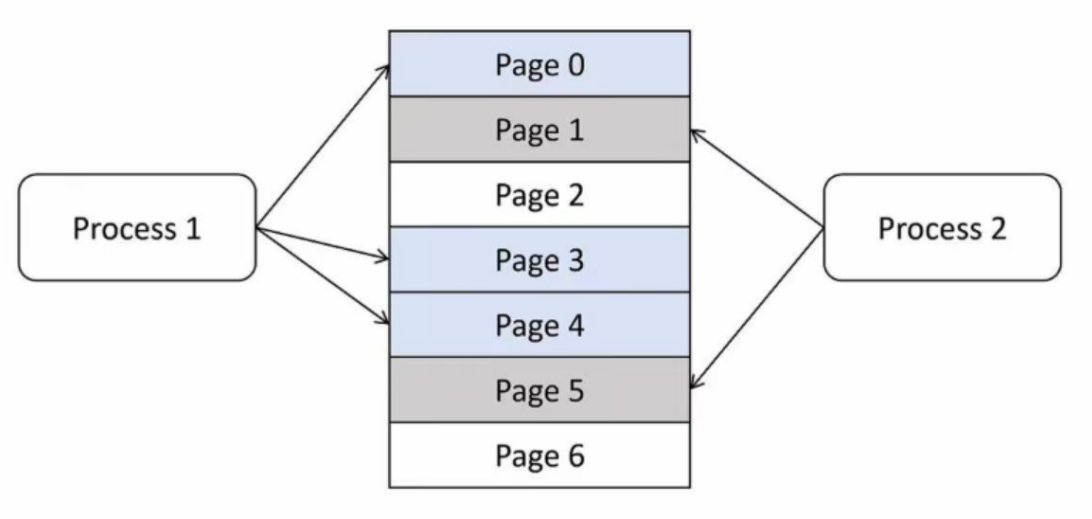
vLLM 采用一种**集中式调度器（scheduler）**来协调分布式 GPU 工作器（worker）的执行。**KV 缓存管理器**由 **PagedAttention** 驱动，能以**分页**方式有效管理 KV 缓存。具体来说，KV 缓存管理器通过集中式调度器发送的指令来管理 GPU 工作器上的物理 KV 缓存内存。



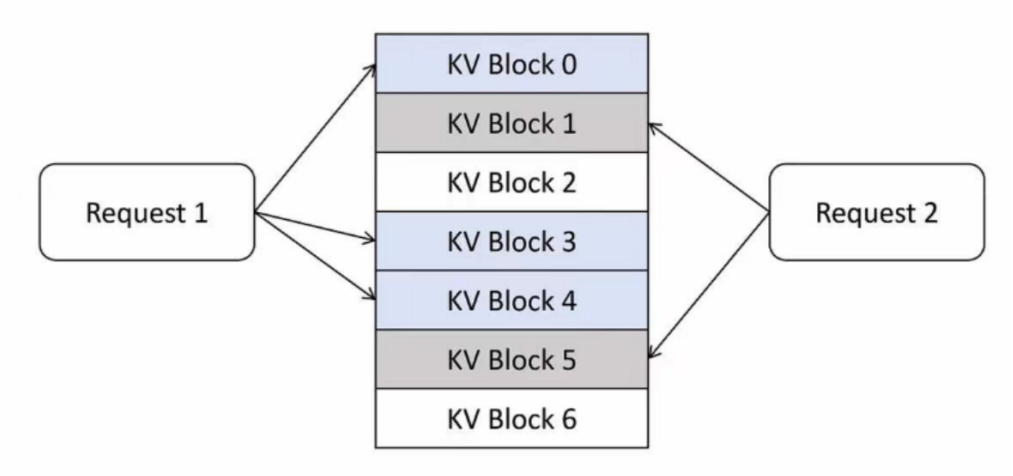
回忆操作系统中的虚拟内存和页管理技术，需要解决以下问题：

- **操作系统给每个程序怎么分配内存?**
- **要不要预分配内存?**
- **程序关闭后怎么回收内存?**
- **内存碎片怎么处理?**
- **怎么管理内存?**

虚拟内存以页为最小单位来分配内存，物理内存被划分为很多页。



PagedAttention 允许**在非连续的内存空间**中**存储连续的键和值**。具体来说，PagedAttention 将每个序列的 KV 缓存划分为块，每个块包含固定数量 token 的 key 和 value：

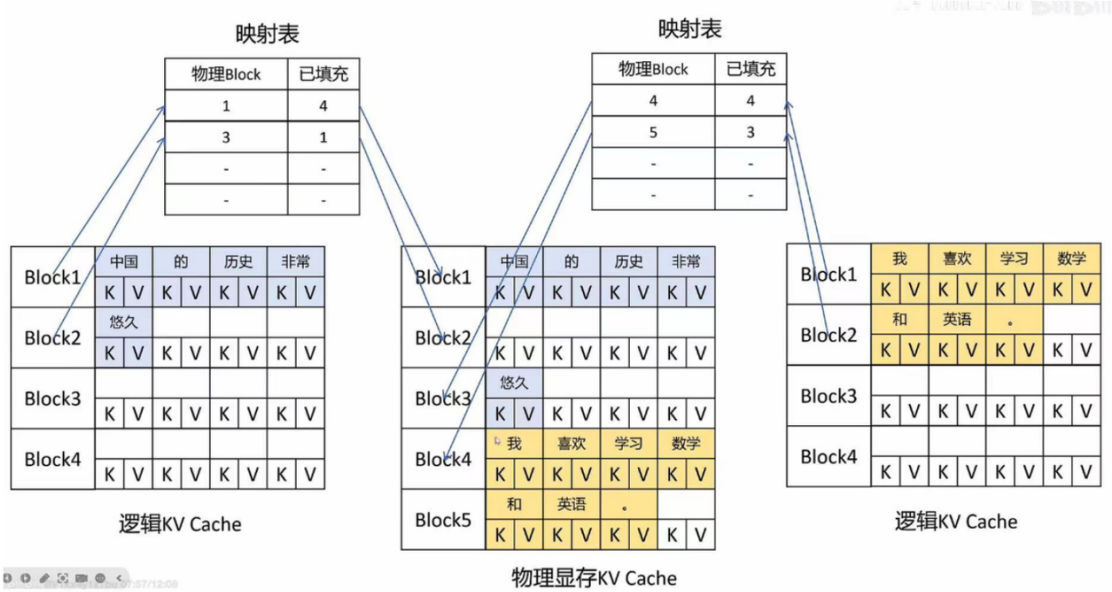


例如，每个kv block可以缓存4个token的k、v向量

1、**按需分配**，不提前预分配

2、**按block分配**，解决了显存碎片的问题（因为碎片最大也只有3个token的大小）

3、**虚拟内存**：维护了逻辑kv cache，让系统看上去是使用的连续的显存（**逻辑连续，物理不连续**），通过映射表来实现，方便block的调用。



对 KV 缓存的请求会被表示成一系列逻辑 KV 块，在生成新 token 和它们的 KV 缓存时从左向右填充。最后一个 KV 块中未填充的位置留给未来填充。通过pagedattention的优化，kv cache的显存利用率从20%-40%提升到了96%，因为浪费只会发生在最后一个块中。采用vLLM能将更多序列进行批处理，提高 GPU 使用率，显著提升吞吐量。

📌 PagedAttention的优点

1. **页按需分配：**PagedAttention 将每个请求的 KV Cache 拆分为若干固定大小的页，并仅在生成新 tokens 时按需分配物理页，无需提前为最大序列长度预留内存，从而大幅降低 KV Cache 内存占用。
2. **跨请求 KV Cache 共享：**相同前缀的多个请求可共用 KV pages，大幅降低内存占用，适用于beam search、A/B test等场景。
3. **消除zero-padding：**传统方法往往需将所有序列填充到同一最大长度，导致短序列浪费大量内存和计算。PagedAttention 只在实际需要的 blocks 上分配内存，短序列不会因对齐而被额外填充，从而**zero-padding**问题被根本杜绝。
4. **提升推理吞吐：**由于减少了内存浪费和碎片，GPU 可以在同等显存下加载更多并发请求，vLLM 的实测表明 PagedAttention 能带来 2–4× 的整体吞吐提升，且在中长序列下优势更明显。