中国科学技术大学 University of Science and Technology of China

USTC School of Computer Science and Technology 计算机科学与技术学院

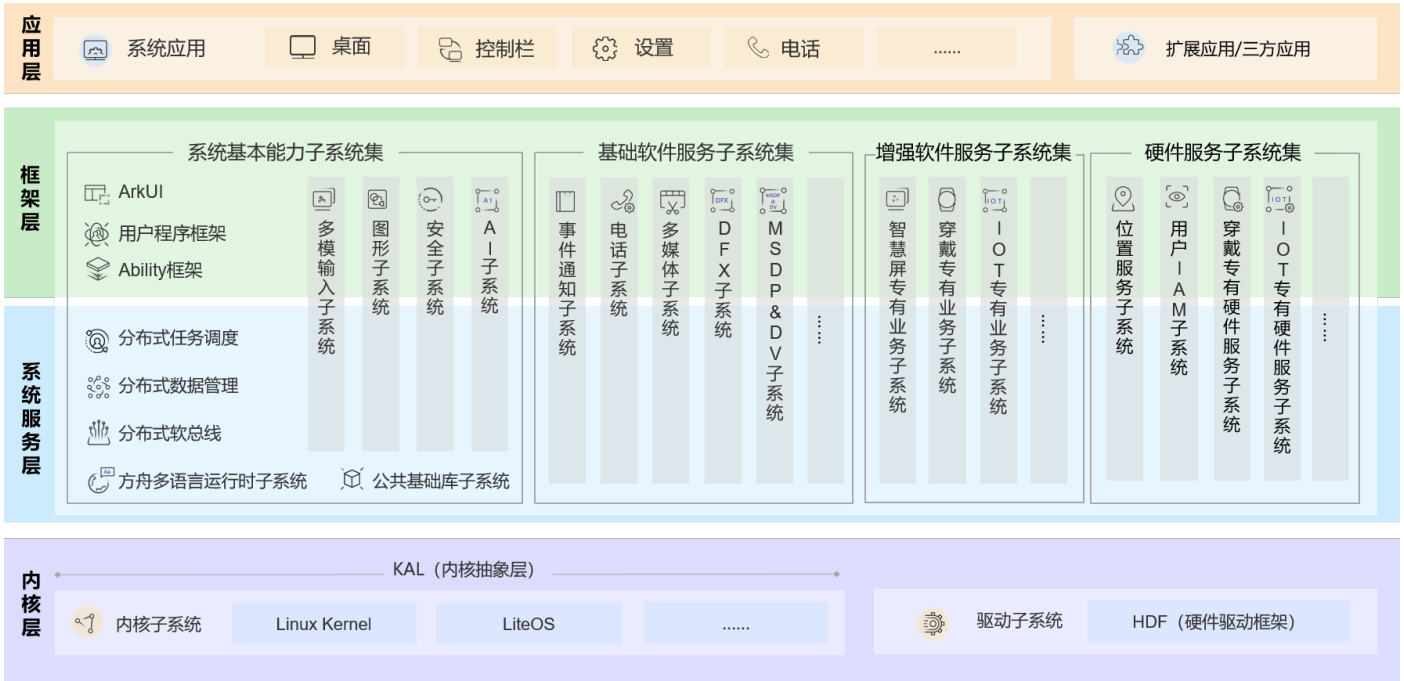
关于课题研究内容的了解学习

组会日程内容汇报

董若扬

计算机科学与技术学院, USTC

2024-07-11



我认为学习的重点是系统服务层的(分布式)数据管理与系统基本能力子系统集中的 AI 子系统。其中研究的重点应该是在分布式数据管理上做改进使其能充分利用 AI 子系统的能力。

- 跨应用数据管理（DataShare）：提供了数据提供者 provider、数据消费者 consumer 以及同设备跨应用数据交互的增、删、改、查以及订阅通知等能力。
- 统一数据管理框架（UDMF）：提供了数据跨应用、跨设备交互标准，实现高效的数据跨应用、跨设备共享。
- 数据管理服务（DatamgrService）：提供其它部件的同步及跨应用共享能力.

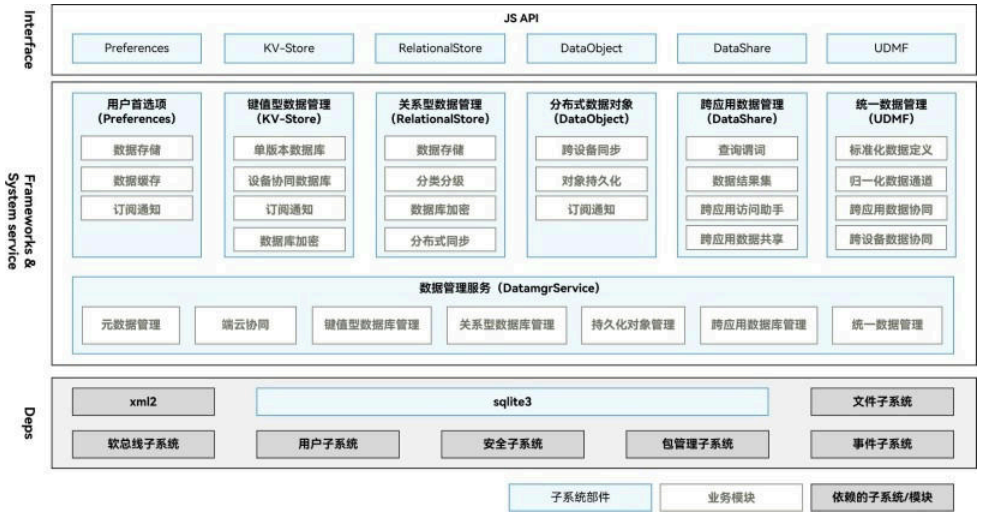


Figure 2: OpenHarmony 数据管理架构图

标准化数据定义

当出现多应用数据共享时，每个应用都有自己的数据定义和数据格式，应用间的的数据交互需做大量的数据格式适配工作。为了降低应用/业务数据交互成本，提出标准化数据定义作为 OpenHarmony 统一的数据语言，构建 OpenHarmony 数据跨应用交互标准。

一对多跨应用数据共享

基于跨应用一对多数据共享的场景，可通过 DataShare 实现。数据提供方无需进行繁琐的封装，可直接使用 DataShare 向其他应用共享数据；对数据访问方来说，因 DataShare 的访问方式不会因数据提供的方式而不同，只需要学习和使用一套接口即可，大大减少了学习时间和开发难度。

数据提供方无需进行繁琐的封装，可直接使用 DataShare 向其他应用共享数据；对数据访问方来说，因 DataShare 的访问方式不会因数据提供的方式而不同，只需要学习和使用一套接口即可，大大减少了学习时间和开发难度。

DataShare 实现跨应用数据共享有两种方式：

- 使用 DataShareExtensionAbility 实现数据共享
- 通过数据管理服务实现数据共享静默访问

多对多跨应用数据共享

通过标准化数据通路实现数据共享。应用可以根据 UDMF 标准化数据通路提供的数据接入与读取接口，将符合标准化数据定义的数据写入 UDMF 不同的数据共享通路，并提供多应用进行读取。写入 UDMF 中的数据依据应用定义的权限、数据通路定义的权限以及整个 UDMF 框架定义的权限管理逻辑进行管理。

PowerInfer-2: Fast Large Language Model Inference on a Smartphone

目前在移动设备上可运行的模型相对较小，并且消耗大量内存，严重限制了大型模型的应用场景。

PowerInfer-2 的主要见解是通过将传统的矩阵计算分解为细粒度的神经元簇计算来利用智能手机中的异构计算、内存和 I/O 资源。PowerInfer-2 具有多态神经元引擎，可针对推理的各个阶段 LLM 调整计算策略。引入了分段神经元缓存和细粒度神经元簇级流水线，可有效最小化和隐藏 I/O 操作导致的开销。

PowerInfer-2 要解决的挑战：

- 利用当代智能手机中存在的高度异构 XPU（如 CPU，GPU，NPU）
- 缓存未命中导致的不可避免的 I/O 开销

PowerInfer-2 的解决方案：

- 将推理中 LLM 典型的粗粒度矩阵计算分解为细粒度神经元簇计算。神经元簇由多个神经元组成，其数量由 XPU、内存和 I/O 的特性决定。PowerInfer-2 设计了一个多态神经元引擎，为推理过程的预填充和解码阶段提供不同的计算模式。在预填充阶段，将所有神经元合并到一个神经元簇中，以最大限度发挥 NPU 在处理大型矩阵计算方面的优势。相反，在解码阶段，其批处理大小为 1 并表现出显着的稀疏性，使用小神经元簇来利用 CPU 内核的灵活性来执行这项相对较轻的计算任务。
- PowerInfer-2 引入了一个在神经元粒度中运行的分段缓存。该缓存针对不同的 LLM 权重类型设计了特定的缓存策略，有效提升了缓存命中率。此外，为了减少 I/O 操作导致的计算延迟，PowerInfer-2 提出了一种细粒度的神经元簇级流水线技术，该技术将 I/O 操作与神经元簇计算重叠。此方法可显著减少与 I/O 延迟相关的等待气泡。

Efficient Memory Management for Large Language Model Serving with PagedAttention

大型语言模型的高吞吐量服务需要一次批处理足够多的请求。然而每个请求的键值缓存（KV 缓存）内存都很大，并且会动态增长和收缩，则碎片和冗余重复可能会严重浪费内存，从而限制批处理大小。

现有的 LLM 服务系统未能有效管理 KV 缓存内存。这主要是因为将请求的 KV 缓存存储在连续的内存空间中。

KV 缓存随着模型生成新标记，它会随着时间的推移动态增长和收缩，并且其生存期和长度是先验未知的。不可避免地会导致内部碎片（预分配了具有请求最大长度的连续内存块）和外部碎片（每个请求的预分配大小可能不同）。

LLM 服务为每个请求生成多个输出，请求由多个序列组成，这些序列可以部分共享其 KV 缓存。由于现有系统的序列的 KV 缓存存储在单独的连续空间中因此无法共享内存。

PagedAttention：受操作系统中经典虚拟内存和分页技术启发的注意力算法。

- 将请求的 KV 缓存划分为多个块，每个块可以包含固定数量令牌的注意力键和值。
- KV 缓存的块不一定存储在连续空间中。

PagedAttention 的抽象：

- Block->Page;
- Token->Byte;
- Request->Process.

- KV 缓存大小随着请求数量的增加而快速增长，即使将所有可用内存都分配给 KV 缓存，也只能容纳几十个请求。GPU 的计算速度增长速度快于内存容量的增长速度，内存将成为一个越来越重要的瓶颈。
- 复杂的解码算法。KV 缓存共享的程度取决于所采用的特定解码算法。
- LLM 服务的请求在其输入和输出长度上表现出可变性要求内存管理系统适应各种提示长度。

现有系统中的内存管理：

- 要求张量存储在连续内存中
- 根据请求的最大可能序列长度为请求静态分配内存块，而不考虑请求的实际输入或最终输出长度

PagedAttention 算法允许将 KV 块存储在非连续物理内存中，从而在 vLLM 中实现更灵活的分页内存管理。vLLM 可以动态增长 KV 缓存内存，而无需提前为所有位置保留它，从而消除了现有系统中的大部分内存浪费。

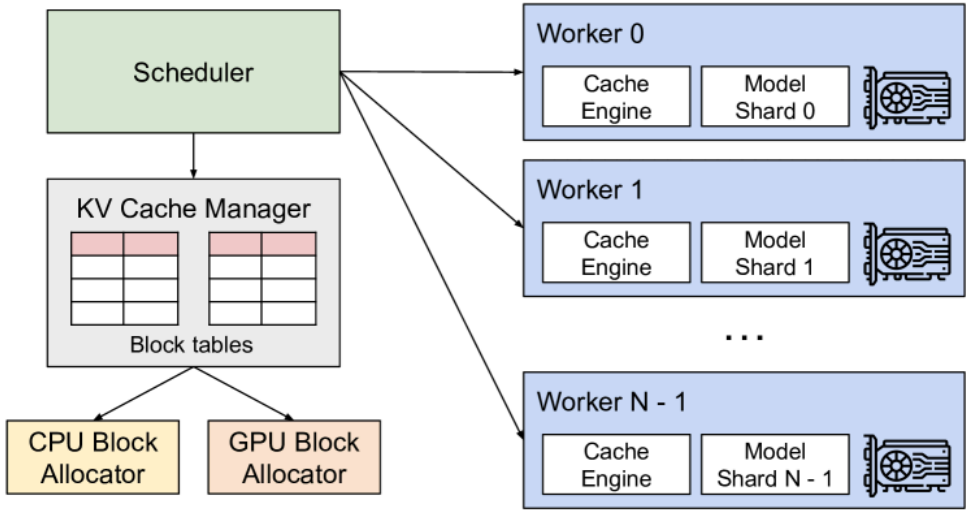


Figure 3: vLLM 采用集中式调度器来协调分布式 GPU Worker 的执行

在 Parallel sampling 中，一个请求生成多个序列，通过其 PagedAttention 和分页内存管理，vLLM 可以轻松实现这种共享并节省内存。

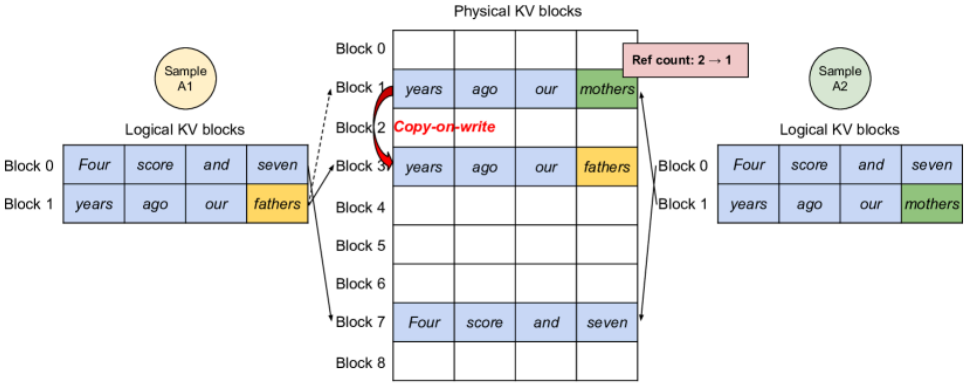


Figure 4: Parallel sampling example.

在 Beam search 中，不仅共享初始提示块，还共享不同候选者之间的其他块，并且共享模式随着解码过程的推进而动态变化。通过 vLLM 的物理块共享而不用频繁地复制 KV 缓存可以显著降低内存拷贝开销。

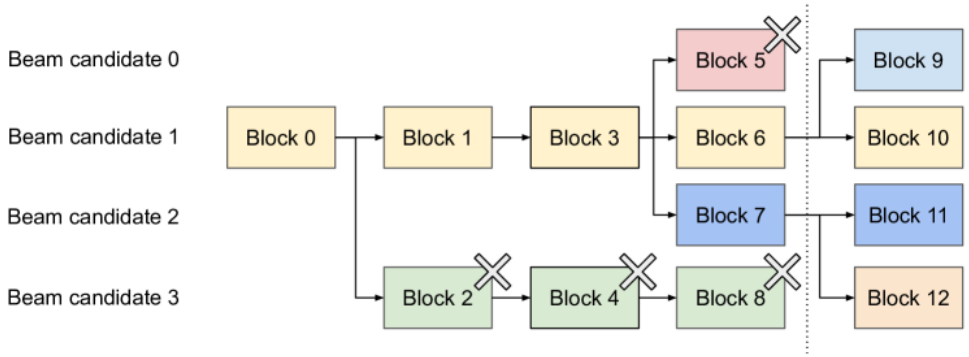


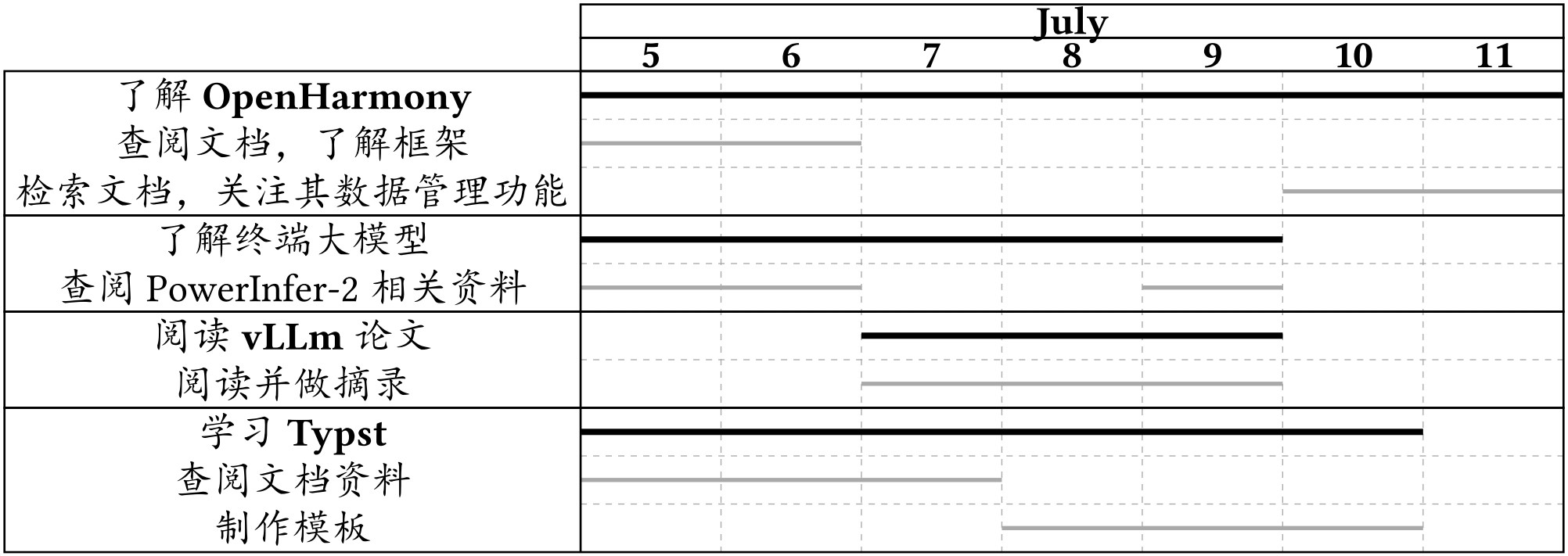
Figure 5: Beam search example.

在以上 PowerInfer-2 和 vLLM 的论文中，都着重关注了设备的内存资源有限这一情况。PowerInfer-2 通过细粒度神经元簇计算和分段缓存技术，有效利用了智能手机中的异构计算、内存和 I/O 资源。vLLM 通过 PagedAttention 算法实现了更灵活的分页内存管理，减小了内存的外部碎片和内部碎片，同时有效地在不同请求之间分享页面。

与计算任务的规模相比，终端设备的内存远远不足以支持 LLM 的推理，整个过程会伴随着大量的调度行为，一些来不及完成的请求需要置换到交换区，因此与常规终端的外部存储相比，交换区相对文件区的占比要相对较大一些。同时为了避免 I/O 操作造成的计算时延，要尽可能减少 I/O 的频率和开销，需要在操作系统的调度上以及文件管理系统上做出改进。同时尽可能的将 I/O 操作于其他请求的计算操作重叠，减少等待气泡。与通用机相比，为更好地支持 LLM，终端设备需要在操作系统的多个模块做出支持 LLM 的改进。

同时在操作系统层面可以为资源受限的终端设备重新设计内核以减少频繁系统调用的开销，减少内核态和用户态的切换次数，减低进程间通信开销等举措。

日程安排


July
5
6
7
8
9
10
11
了解 OpenHarmony
查阅文档，了解框架
检索文档，关注其数据管理功能
了解终端大模型
查阅 PowerInfer-2 相关资料
阅读 vLLm 论文
阅读并做摘录
学习 Typst
查阅文档资料
制作模板