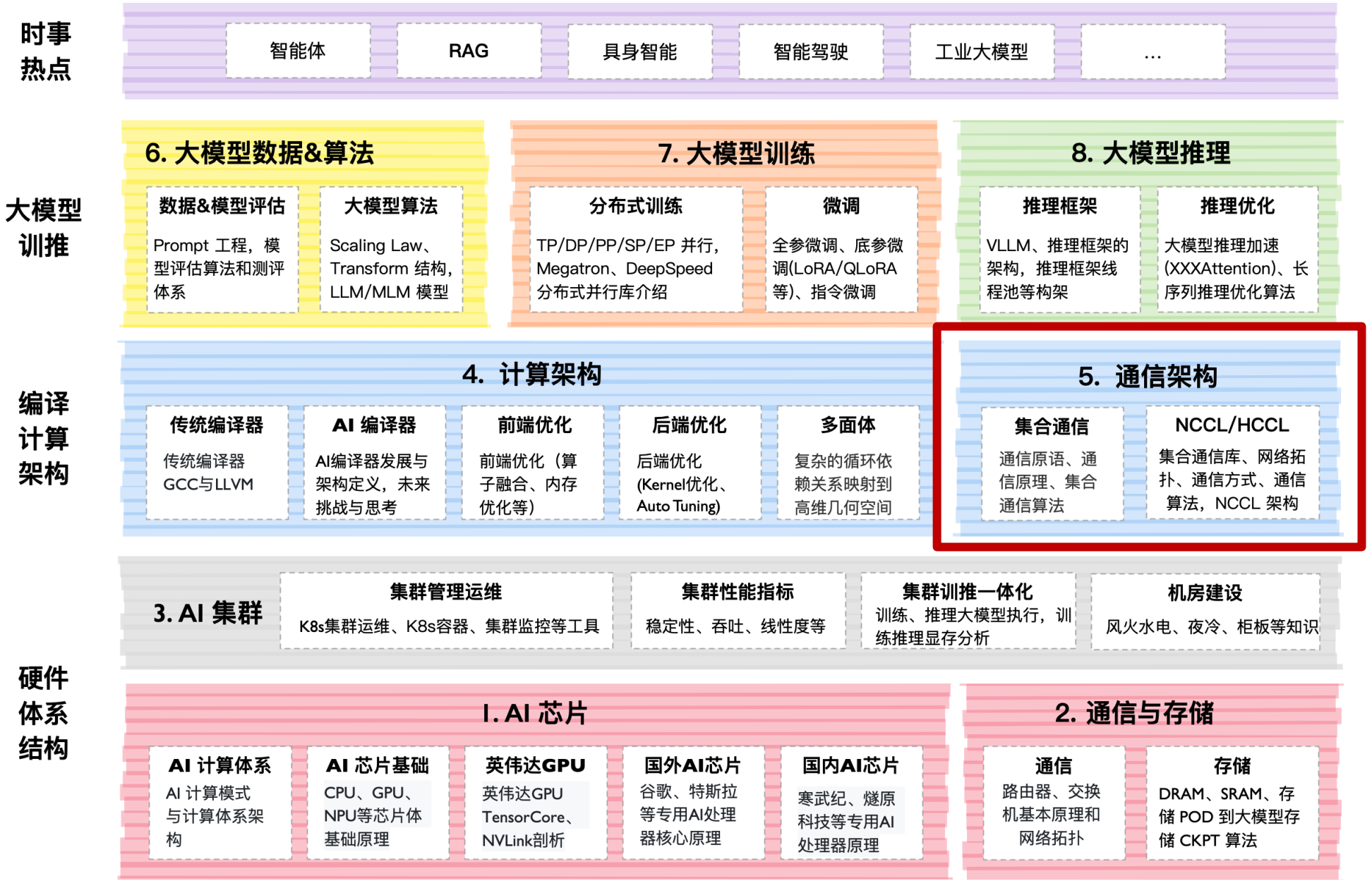
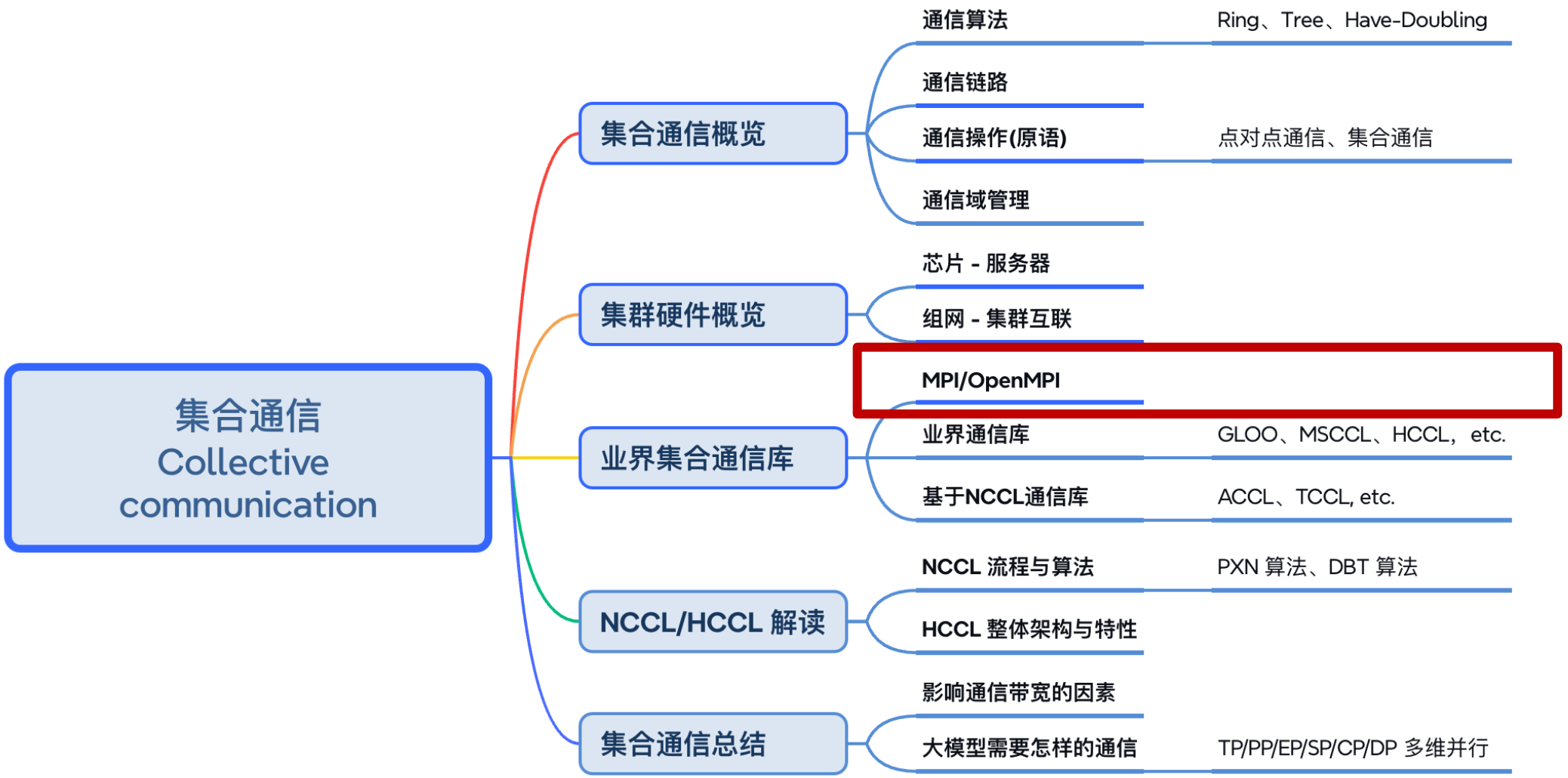
大模型系列 – 集合通信库

MPI 通信与通信库

ZOMI


时事热点
智能体
RAG
具身智能
智能驾驶
工业大模型
...
大模型训推
6. 大模型数据&算法
数据&模型评估
Prompt 工程，模型评估算法和测评体系
大模型算法
Scaling Law、Transform 结构，LLM/MLM 模型
7. 大模型训练
分布式训练
TP/DP/PP/SP/EP 并行，Megatron、DeepSpeed 分布式并行库介绍
微调
全参微调、底参微调(LoRA/QLoRA 等)、指令微调
8. 大模型推理
推理框架
VLLM、推理框架的架构，推理框架线程池等构架
推理优化
大模型推理加速(XXXAttention)、长序列推理优化算法
编译计算架构
4. 计算架构
传统编译器
传统编译器 GCC与LLVM
AI 编译器
AI编译器发展与架构定义，未来挑战与思考
前端优化
前端优化（算子融合、内存优化等）
后端优化
后端优化(Kernel优化、Auto Tuning)
多面体
复杂的循环依赖关系映射到高维几何空间
5. 通信架构
集合通信
通信原语、通信原理、集合通信算法
NCCL/HCCL
集合通信库、网络拓扑、通信方式、通信算法，NCCL 架构
硬件体系结构
3. AI 集群
集群管理运维
K8s集群运维、K8s容器、集群监控等工具
集群性能指标
稳定性、吞吐、线性度等
集群训推一体化
训练、推理大模型执行，训练推理显存分析
机房建设
风火水电、夜冷、柜板等知识
1. AI 芯片
AI 计算体系
AI 计算模式与计算体系架构
AI 芯片基础
CPU、GPU、NPU等芯片体基础原理
英伟达GPU
英伟达GPU TensorCore、NVLink剖析
国外AI芯片
谷歌、特斯拉等专用AI处理器核心原理
国内AI芯片
寒武纪、燧原科技等专用AI处理器原理
2. 通信与存储
通信
路由器、交换机基本原理和网络拓扑
存储
DRAM、SRAM、存储 POD 到大模型存储 CKPT 算法

思维导图 XMind


集合通信
Collective
communication
集合通信概览
通信算法
Ring、Tree、Have-Doubling
通信链路
通信操作(原语)
点对点通信、集合通信
通信域管理
集群硬件概览
芯片 - 服务器
组网 - 集群互联
MPI/OpenMPI
业界集合通信库
业界通信库
GLOO、MSCCL、HCCL，etc.
基于NCCL通信库
ACCL、TCCL, etc.
NCCL/HCCL 解读
NCCL 流程与算法
PXN 算法、DBT 算法
HCCL 整体架构与特性
集合通信总结
影响通信带宽的因素
大模型需要怎样的通信
TP/PP/EP/SP/CP/DP 多维并行

Question

1. 为什么要了解 MPI？ MPI 是集合通信库 XCCL 的基础，包含了很多基本概念和基础 API 定义，是了解 NV NCCL 和 Huawei HCCL 的最好入门。

本节内容

1. MPI 通信基本概念：什么是 MPI
2. P2P 通信：Per2Per
3. 集合通信：CC
4. 程序运行：PM

1. MPI通信基本概念

MPI Introduction

MPI通信基本概念

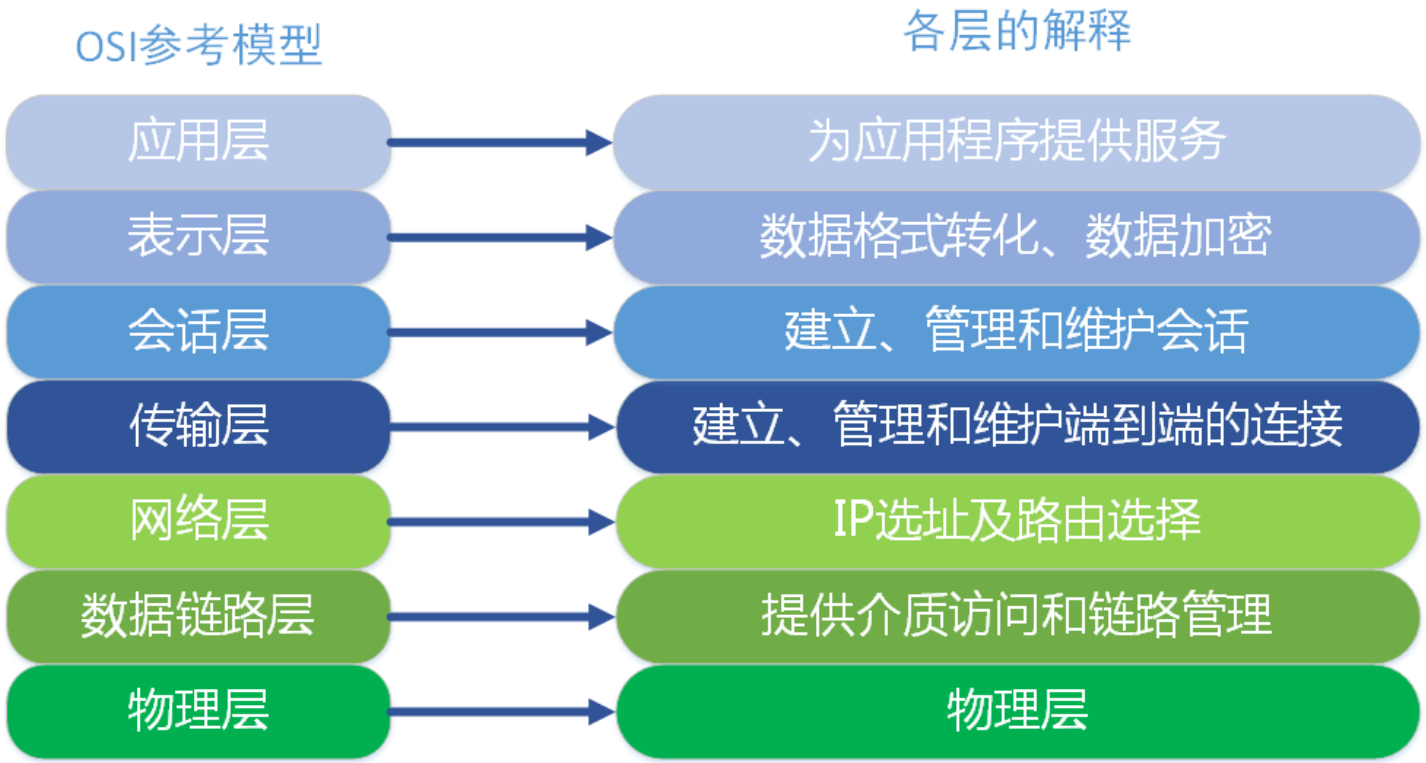
- MPI（message passing interface）跨语言的通讯协议 or 范式标准，提供了应用程序接口 API，包括协议和通信语义。支持语义丰富的消息通信机制，包括点对点、组播和多播模式。
- MPI标准规定了基于消息传递的并行编程 API的调用规范和语义，不同的实现（如 mpich / openmpi）采用不同优化策略。

MPI通信基本概念

- MPI 与网络协议本身没有直接关系，MPI 属于 OSI 参考模型第五层 or 更高，MPI 实现可以通过传输层 sockets 和 TCP。大部分 MPI 实现由指定 API 组成。

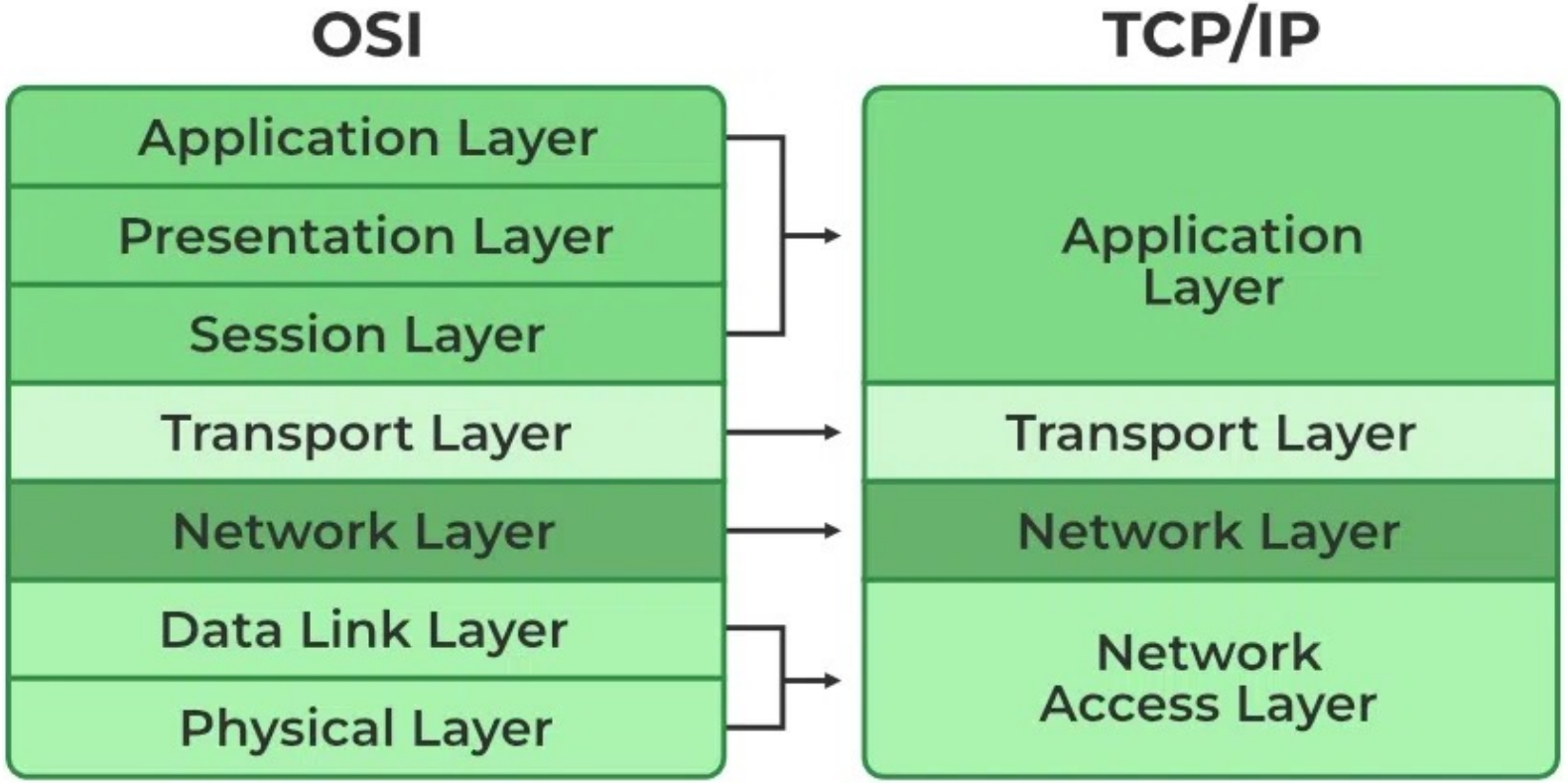
OSI 模型

- OSI模型，即开放式通信系统互联参考模型(Open Systems Interconnection Reference Model)，国际标准化组织(ISO) 提出一个试图使各种计算机在世界范围内互连的网络标准框架，简称 OSI。



TCP/IP 协议

- 但 OSI 模型并没有提供可实现方法，只是概念描述，用来协调进程间通信。即 OSI 模型并不是标准，而是制定标准时所使用的概念性框架。事实上网络标准是 TCP/IP。

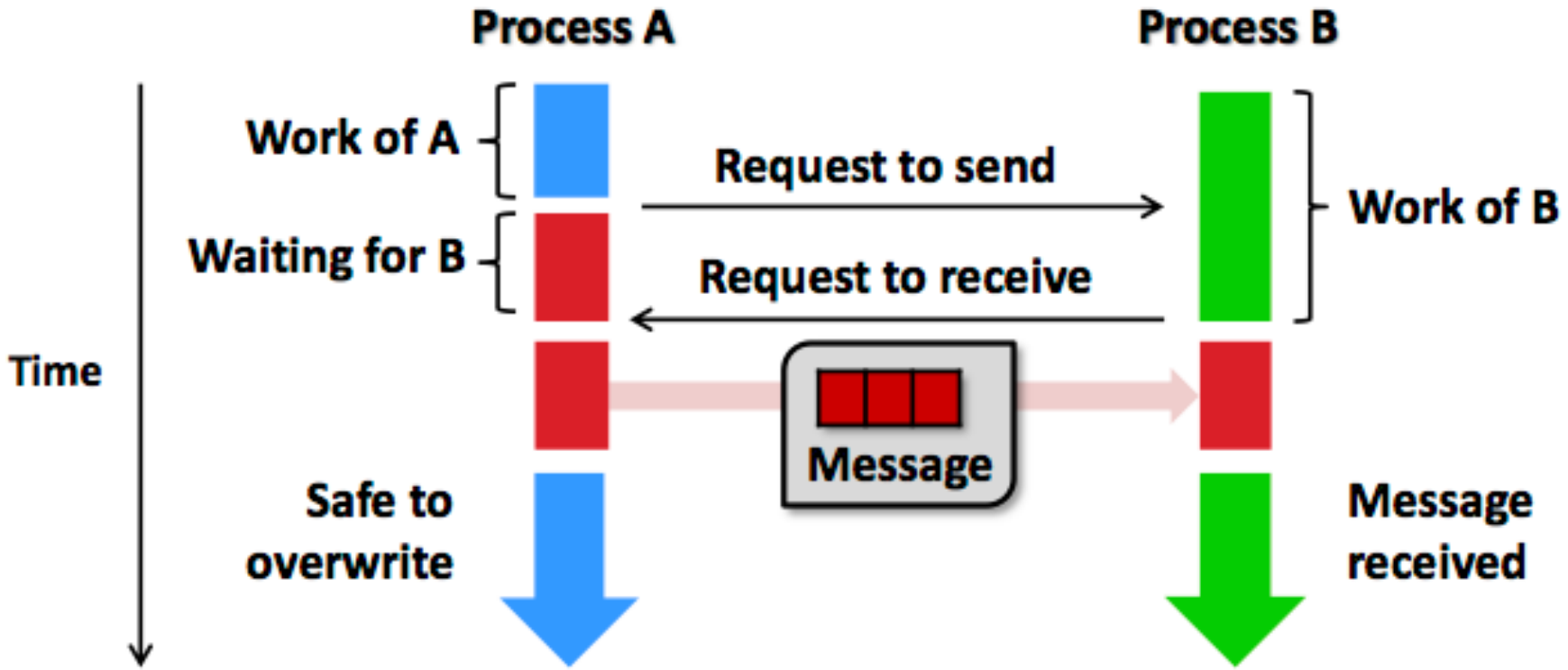


2. P2P 通信

Per2Per

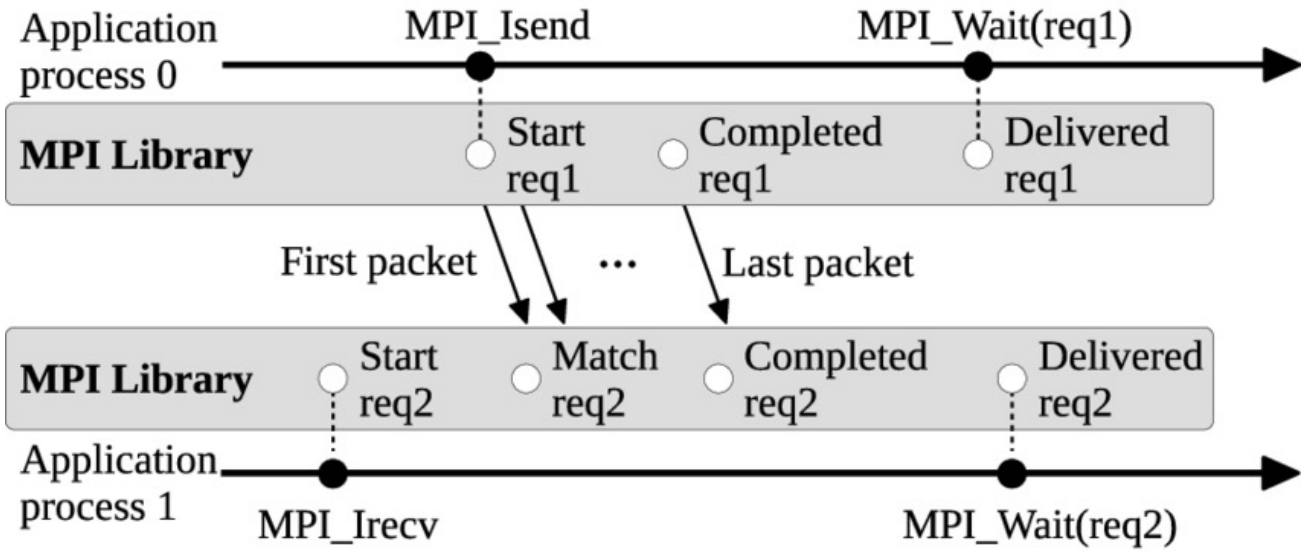
Per2Per

- **点对点通信：**两个进程间通信，用于控制同步或者数据传输，如 MPI_Send 和 MPI_Recv。
- **通信方式：**两进程 Process 计算结束后，相互交换消息前需要请求访问，再进行下一阶段计算。



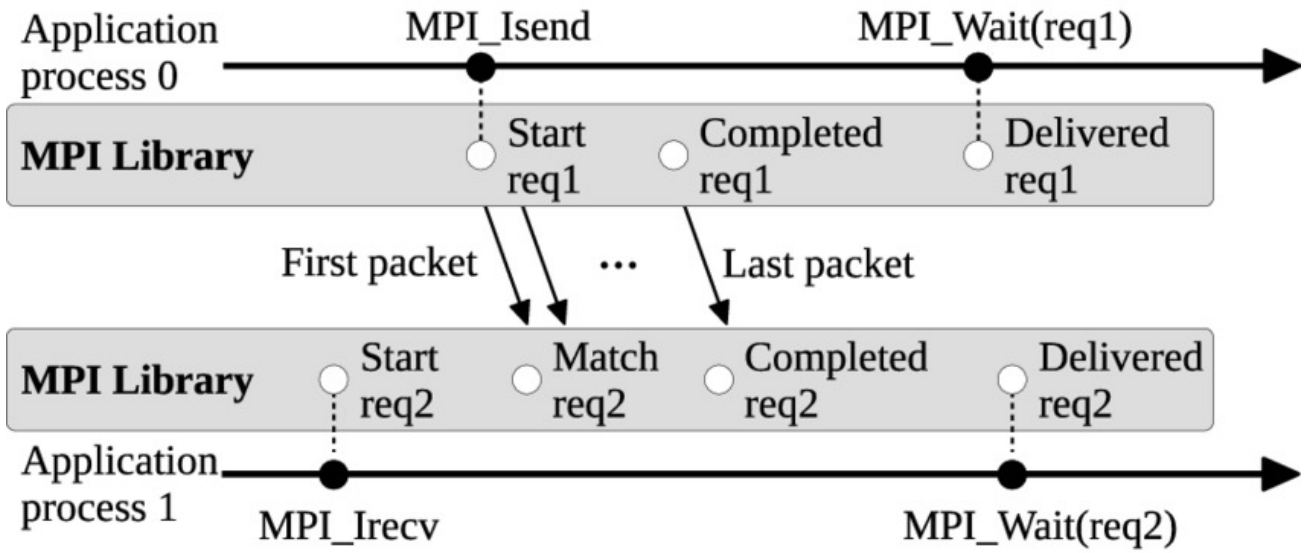
Per2Per 同步阻塞

- 点对点通信分为同步（阻塞blocking）和异步（非阻塞non-blocking）。**同步阻塞：**
  - MPI_Send 返回意味进程发送数据结束，进程缓冲可以重用/覆盖，但不代表 Receiver 收到数据。
  - MPI_Recv 返回意味着数据已经接收到进程的缓冲区，可以使用。



Per2Per异步非阻塞

- 点对点通信分为同步（阻塞blocking）和异步（非阻塞non-blocking）。**异步非阻塞：**
  1. 意义在于进程 send / receive 操作马上返回，继续后续阶段的计算
  2. 当程序要求操作必须确认完成时，调用相应测试接口（e.g. MPI_Wait）阻塞等待操作完成
  3. 异步编程相对复杂，但使得计算和通信可以一定程度并行，降低数据同步带来的运行时开销



3. 集合通信

Collective Communication

集合通信

- 集合通信包括了一对多、多对一和多对多的通信方式，常用于一组进程之间的数据交换。

类型	函数名	含义
通信	MPI_Bcast	一对多广播相同消息
	MPI_Gather	多对一收集各进程消息
	MPI_Gatherv	MPI_Gather 一般化
	MPI_Allgather	全局收集
	MPI_Allgatherv	MPI_Allgather 一般化
	MPI_Scatter	一对多散播不同消息
	MPI_Scatterv	MPI_Scatter 一般化
	MPI_Alltoall	多对多全局交换信息
	MPI_Alltoallv	MPI_Alltoall 一般化
规约	MPI_Reduce	多对一规约
	MPI_Allreduce	MPI_Reduce 一般化
	MPI_Reduce_scatter	MPI_Reduce 一般化
	MPI_Scan	前缀和
同步	MPI_Barrier	路障同步

集合通信原语的含义

https://space.bilibili.com/517221395/channel/collectiondetail?sid=3130927

大模型的集合通信内容介绍 #大模型 #通信 #集合通信

3654 6-2

为什么需要集合通信？NCCL的架构是什么样？ #大模型 #通信 #集合通

3439 6-3

集合通信的操作/原语/算子是什么？ #大模型 #通信 #集合通信

2830 6-5

AI 对集合通信算法的诉求有什么？集合通信算法是啥？ #大模型 #通

2342 6-12

大模型并行的集合通信算法具体实现细节纰漏！#大模型 #集合通信

2156 6-14

通信域是什么概念？PyTorch 如何实现集合通信？#大模型 #集合通信

2383 6-15

研究大模型在 AI 集群的通信，还要了解芯片内互联技术？Yes！#大模

3008 6-30

终于到了大模型集群互联，看昇腾 Atlas 900集群细节！#大模型 #集

3882 7-2

进程启动与收发数据顺序：Broadcast

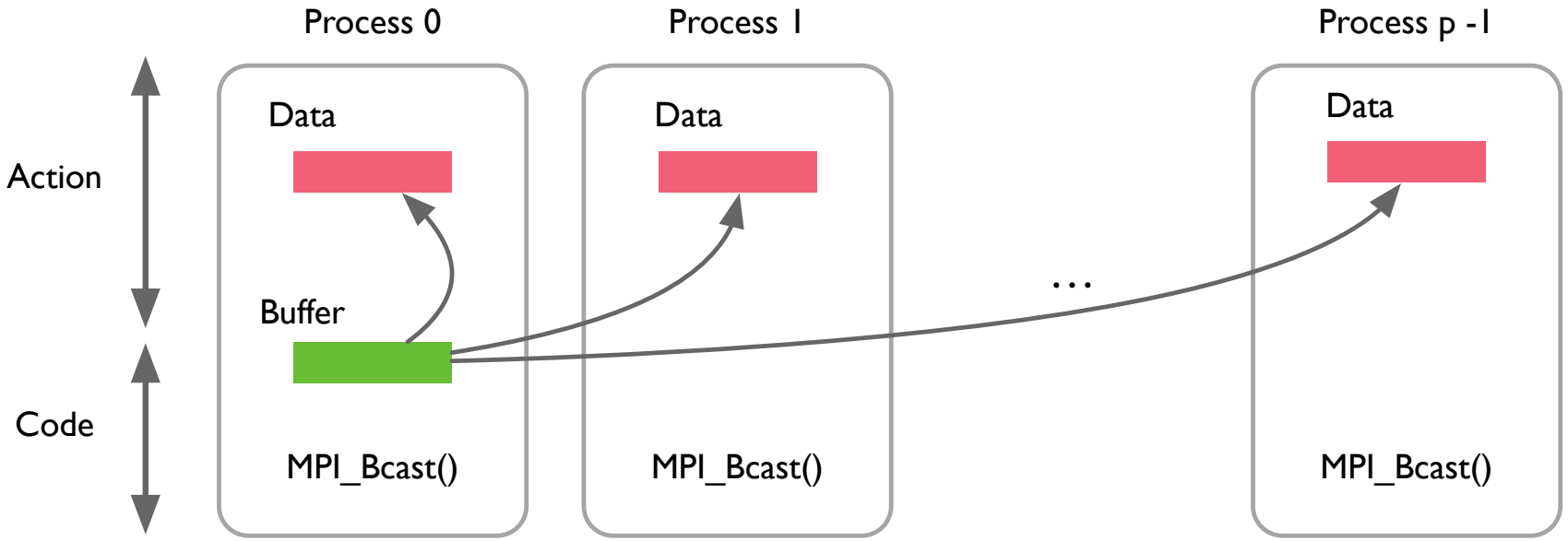
- 当 MPI 进程启动后，每个 Process 会分配唯一序号 Rank。集合通信需要指定一个协调者（e.g. Rank 0 Process，一般称为 ROOT），由其负责将数据发送给所有进程。
- 以接口 MPI_BCAST 为例，它将数据从根进程发送到所有其它进程。所有进程都调用 MPI_BCAST。虽然每个进程都调用 MPI_Bcast，但根进程负责广播数据，其它进程接收数据。

```
## Broadcast 100 ints from process 0 to every process in the group.

MPI_Comm comm;
int array[100];
int root=0;
...
MPI_Bcast(array, 100, MPI_INT, root, comm);
...
```

进程启动与收发数据顺序：Broadcast

- 当 MPI 进程启动后，每个 Process 会分配唯一序号 Rank。集合通信需要指定一个协调者（e.g. Rank 0 Process，一般称为 ROOT），由其负责将数据发送给所有进程。
- 以接口 MPI_BCAST 为例，它将数据从根进程发送到所有其它进程。所有进程都调用 MPI_BCAST。虽然每个进程都调用 MPI_Bcast，但根进程负责广播数据，其它进程接收数据。

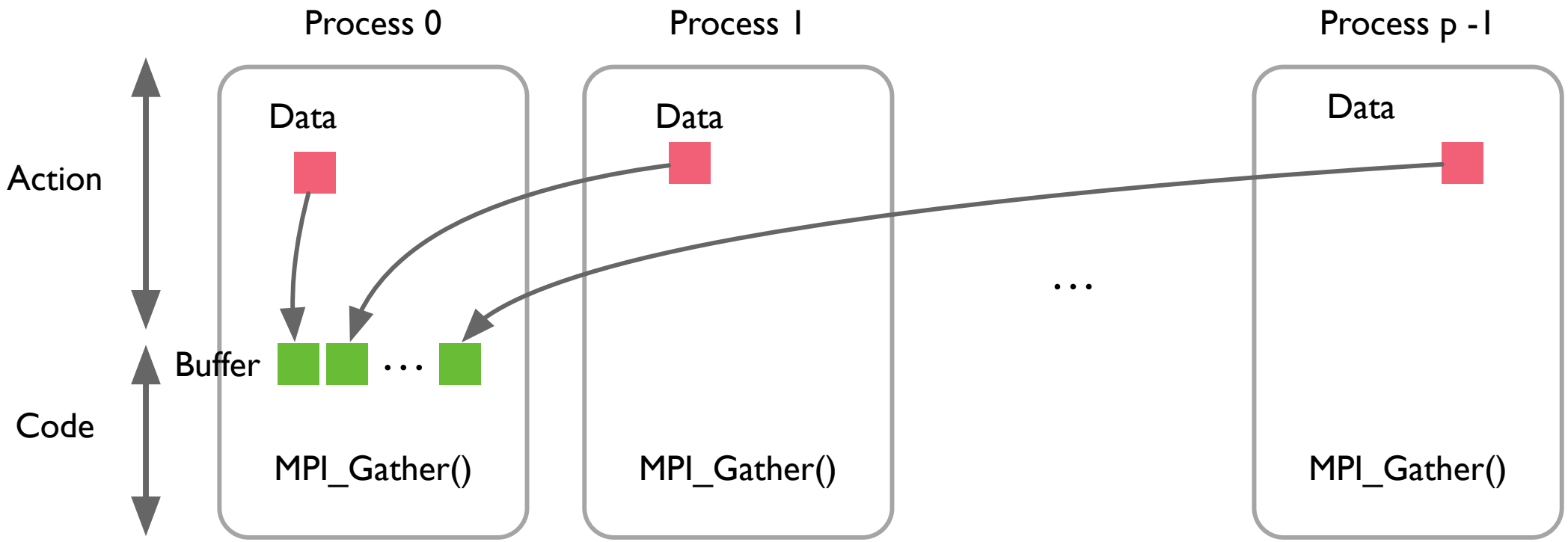


进程启动与收发数据顺序：Gather

- MPI_Gather 行为恰好相反，每个进程将数据发送给根进程。如下代码实现一个典型聚合数据过程：

```
MPI_Comm comm;
int gsize,sendarray[100];
int root, myrank, *rbuf;
...
MPI_Comm_rank(comm, &myrank);
if (myrank == root) {
    MPI_Comm_size(comm, &gsize);
    rbuf = (int *)malloc(gsize*100*sizeof(int));
}
MPI_Gather(sendarray, 100, MPI_INT, rbuf, 100, MPI_INT, root, comm);
...
```

进程启动与收发数据顺序：Gather

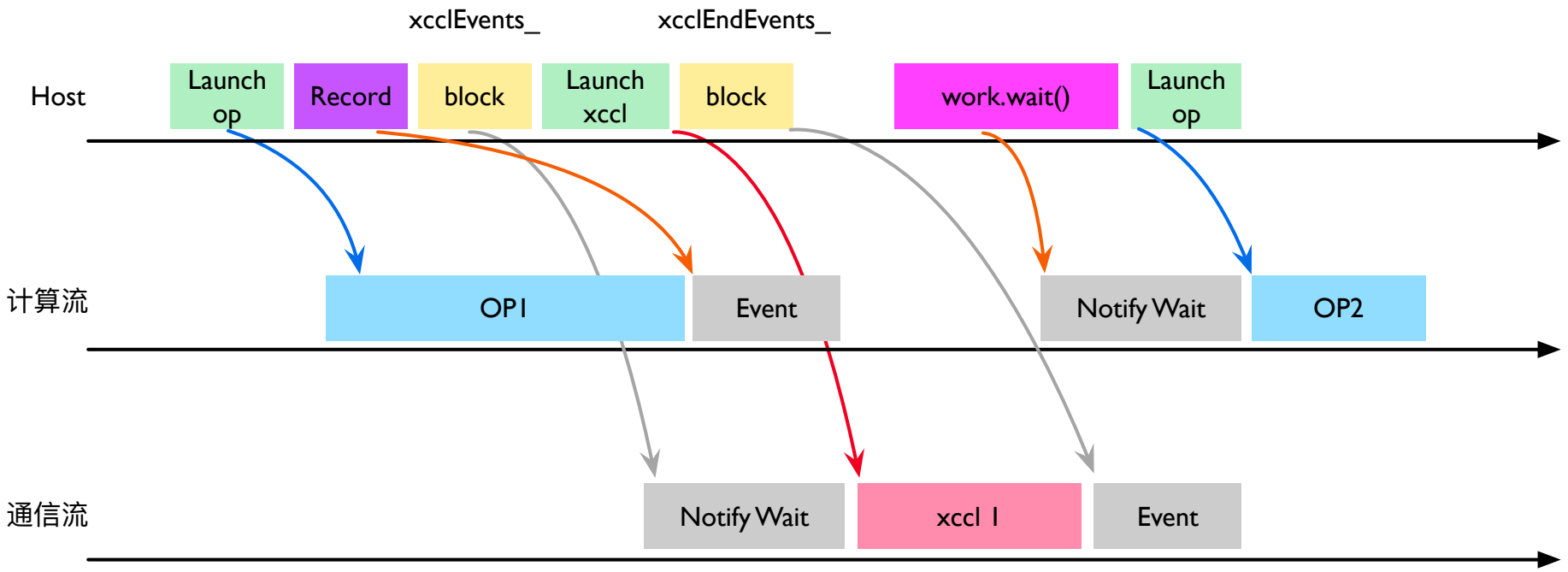


MPI集合通信编程模式

- 每个程序独立完成计算后，到达交汇点，同时调用集合通信原语（Primitive）完成数据通信，然后根据结果进行后续计算。
- 当计算规模较大，集合通信性能非常关键，不同 MPI 实现框架有不同优化方案。实际工程应用中，往往采用更复杂拓扑结构来提升性能，e.g. 树形结构、环形结构。

同步 Barrier

- 某些场景下，多个进程需要协调同步进入某个过程。MPI提供了同步原语例如MPI_Barrier。所有进程调用MPI_Barrier，阻塞程序直到所有进程都开始执行这个接口，然后返回。
- Barrier 作用就是让所有进程确保 MPI_Barrier 之前的工作都已完成，同步进入下一个阶段。

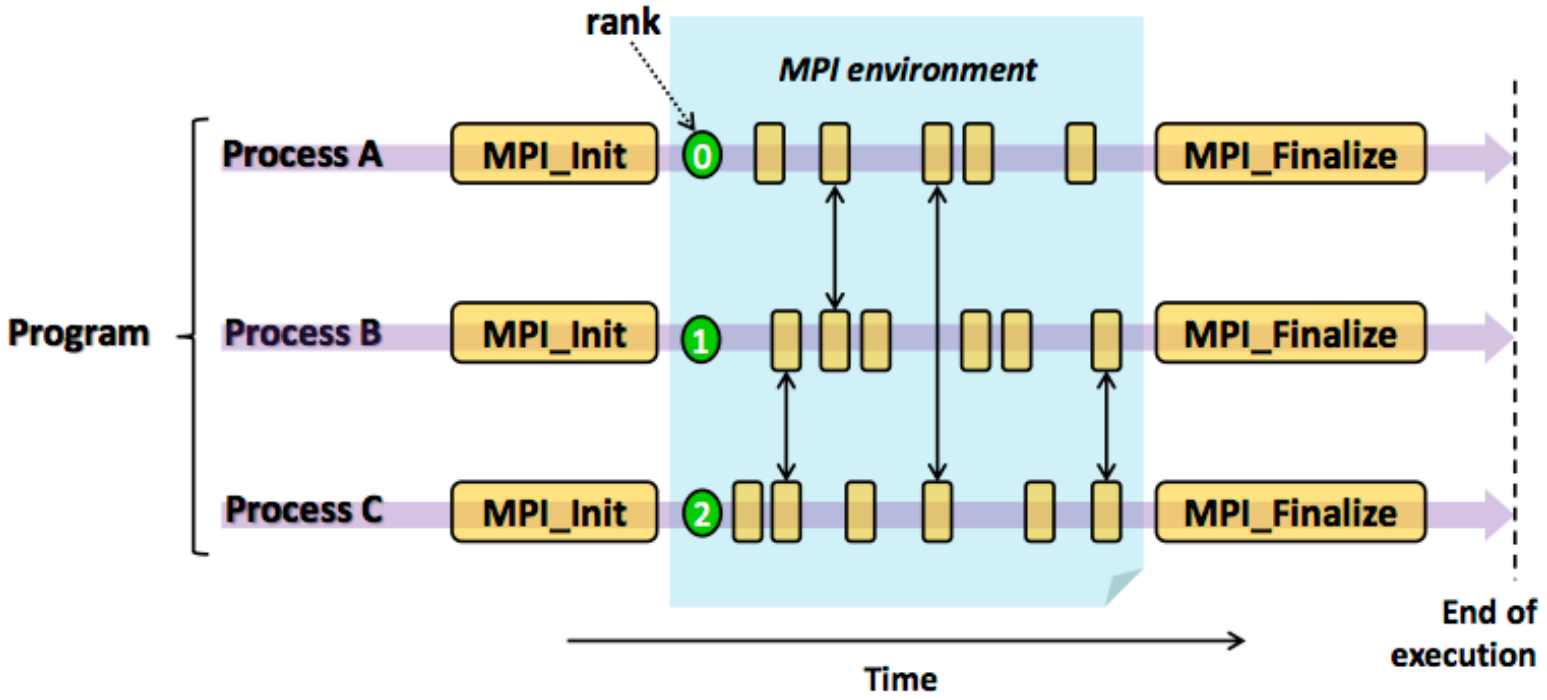


4.MPI 程序运行

Programing Model

编程模型 Programming Model

- MPI 程序编程模式为迭代式“计算 + 通信”，程序可以分为计算块和通信块。
  1. 每个程序可以独立完成计算块，计算完成后进行交互（通信 or 同步）
  2. 交互后后进入下一阶段计算，直到所有任务完成，程序退出

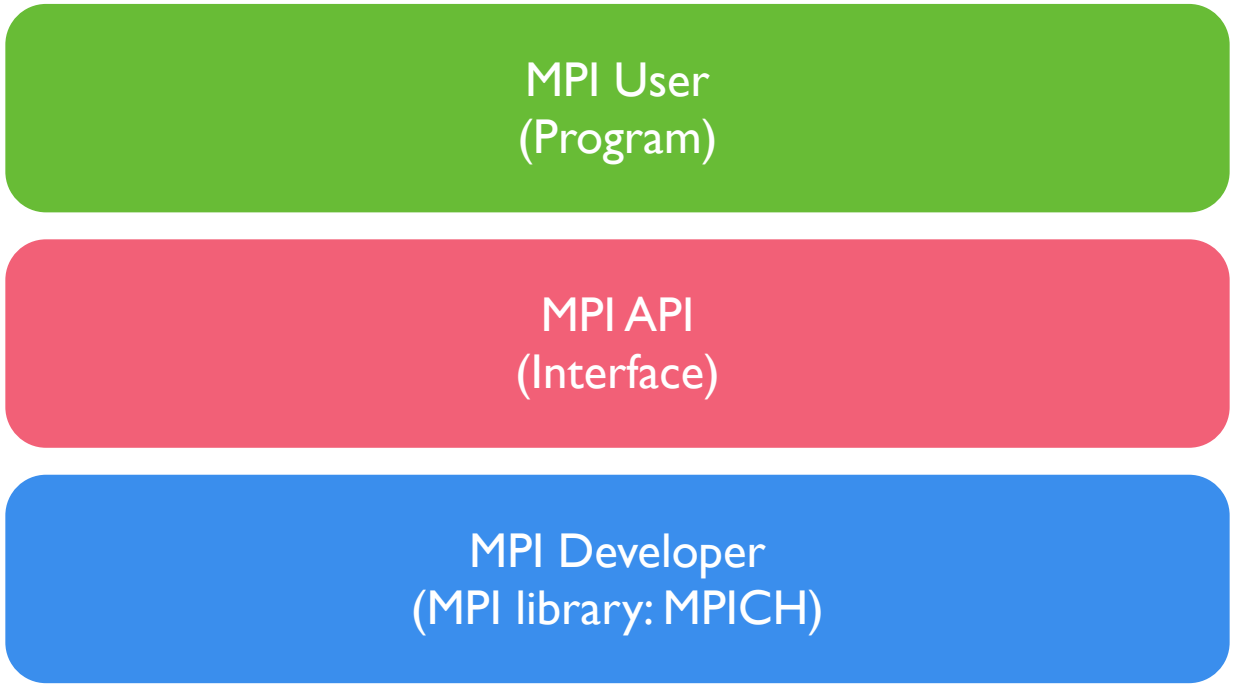


MPI 程序任务分解

- MPI框架只提供通信机制 —— 任务间同步和通信手段，与计算无关。
- 计算任务如何分解和实现，数据如何划分，任务如何合并等问题由程序开发者决定。
- MPI 框架在程序启动时为每个程序副本分配唯一 Rank ID。程序通过获取和根据 Rank 确定任务。

MPI 程序任务分解

- 一个典型 MPI 程序由用户程序部分（MPI User）链接 MPI 库（MPI Interface）构成，计算任务本身的算法实现、任务分解和合并实现在用户程序部分，与MPI无关，也不受MPI限制。SO MPI 提供给开发者灵活性，实现了最小封装。



5 小结与思考

Question

1. MPI 是集合通信库 XCCL 的基础，包含了很多基本概念和 基础 API 定义，是了解 NV NCCL 和 Huawei HCCL 的最好入门。
2. OpenMPI 作为早期的开源集合通信库，定义了P2P 通信、集合通信和对应的程序运行。

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/AIFoundation