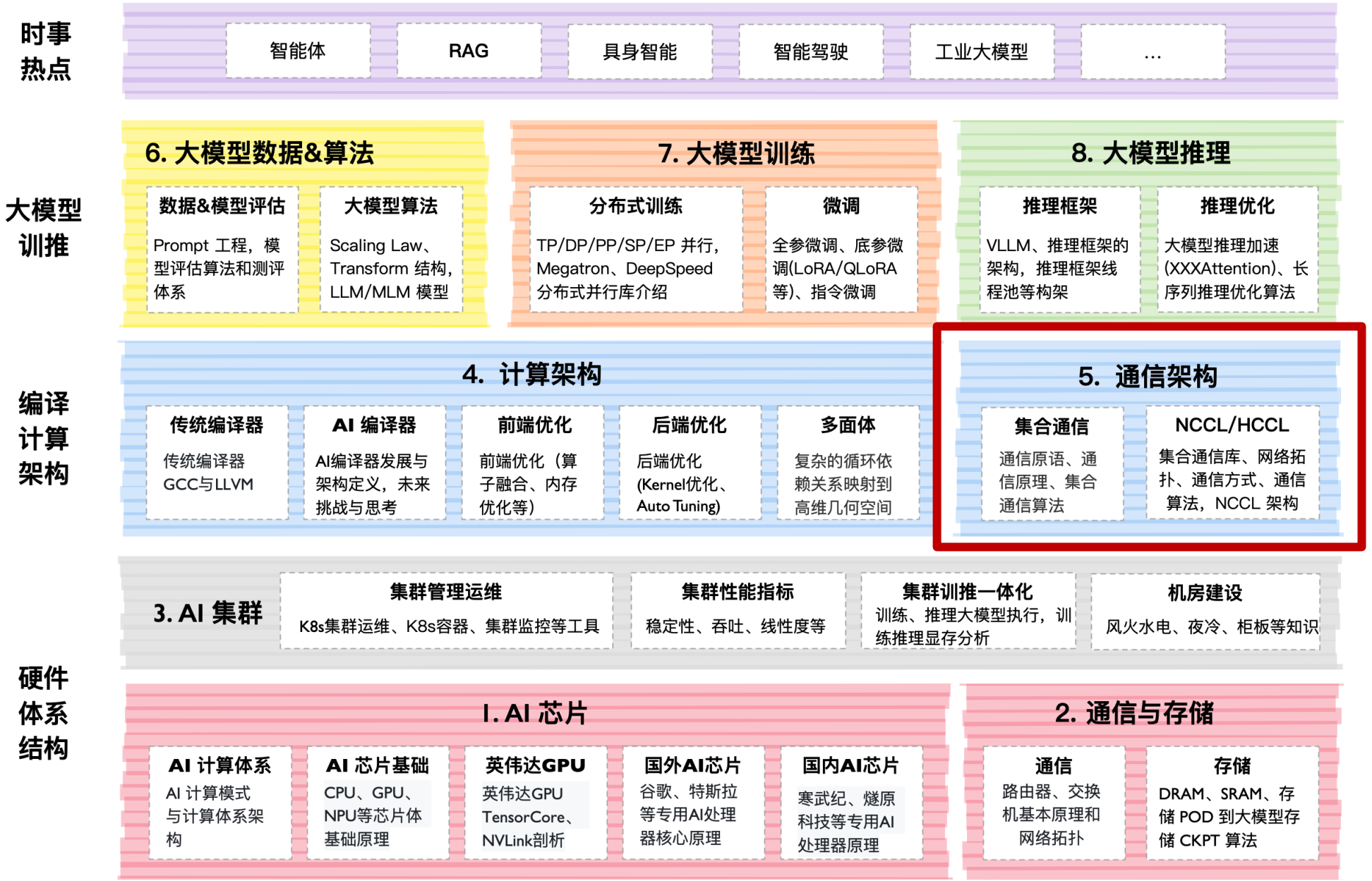
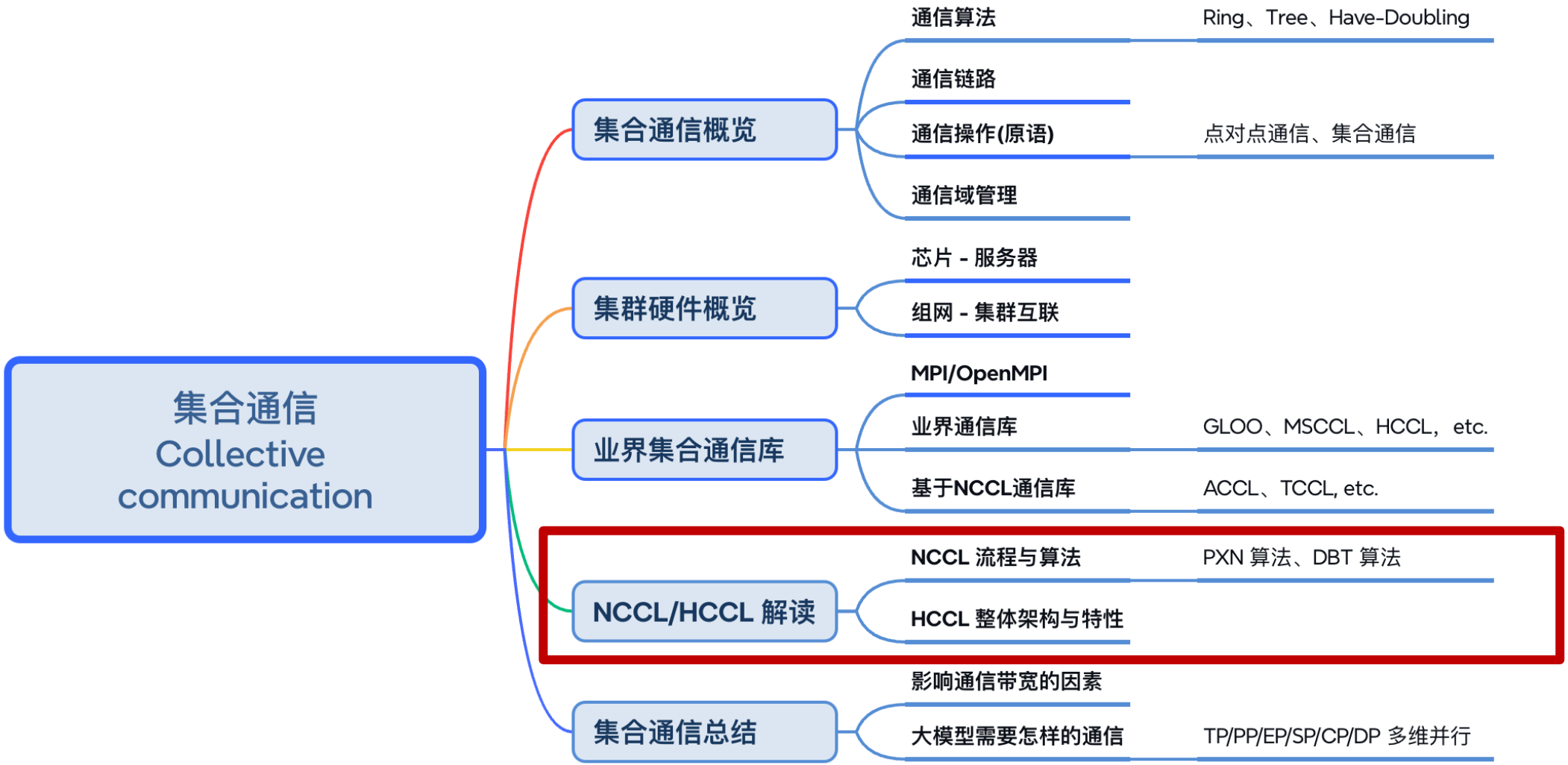
大模型系列 – 集合通信库

华为HCCL架构介绍

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
业界集合通信库
MPI/OpenMPI
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

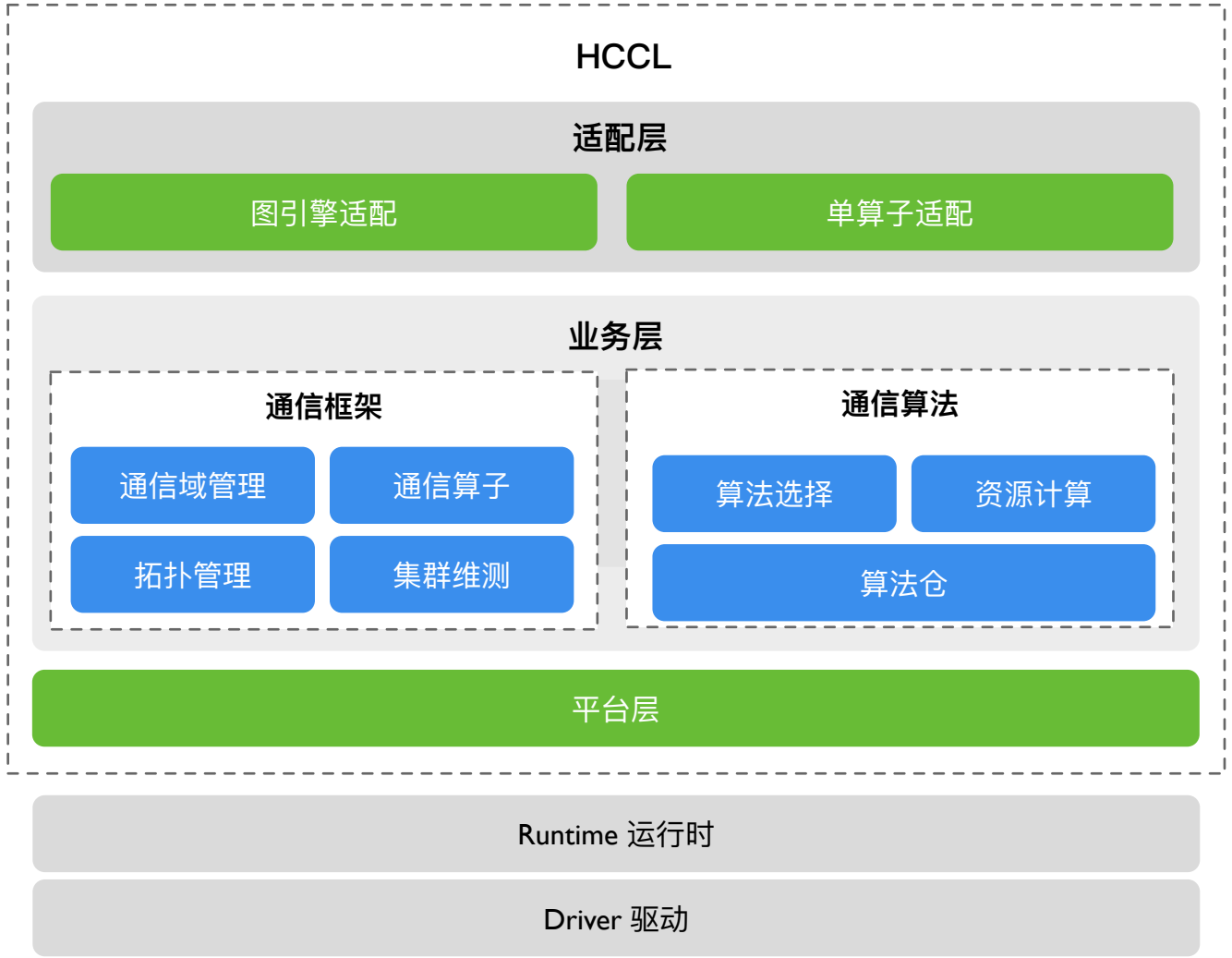
1. 华为 HCCL 终于开源了，国内唯一的开源集合通信库，到底有什么东西?

01. HCCL

基本介绍

HCCL 基本介绍

- 华为集合通信库（Huawei Collective Communication Library，HCCL）基于昇腾AI处理器高性能集合通信库；
- HCCL 分为通信框架、通信算法与通信平台三个模块；
- 开源源码仓中包含其中紫色底纹所示的“通信框架”与“通信算法”两个模块源码。



华为昇腾AI处理器介绍

再不了解昇腾 AI服务器就要被公关掉了，随时删库跑路！ #大模型 #

8783 6-22

你居然？敢说昇腾310/910 SOC处理器架构！#昇腾 #AI芯片

9349 6-23

昇腾的达芬奇内核架构，终于有人说明白了！#昇腾 #AI芯片

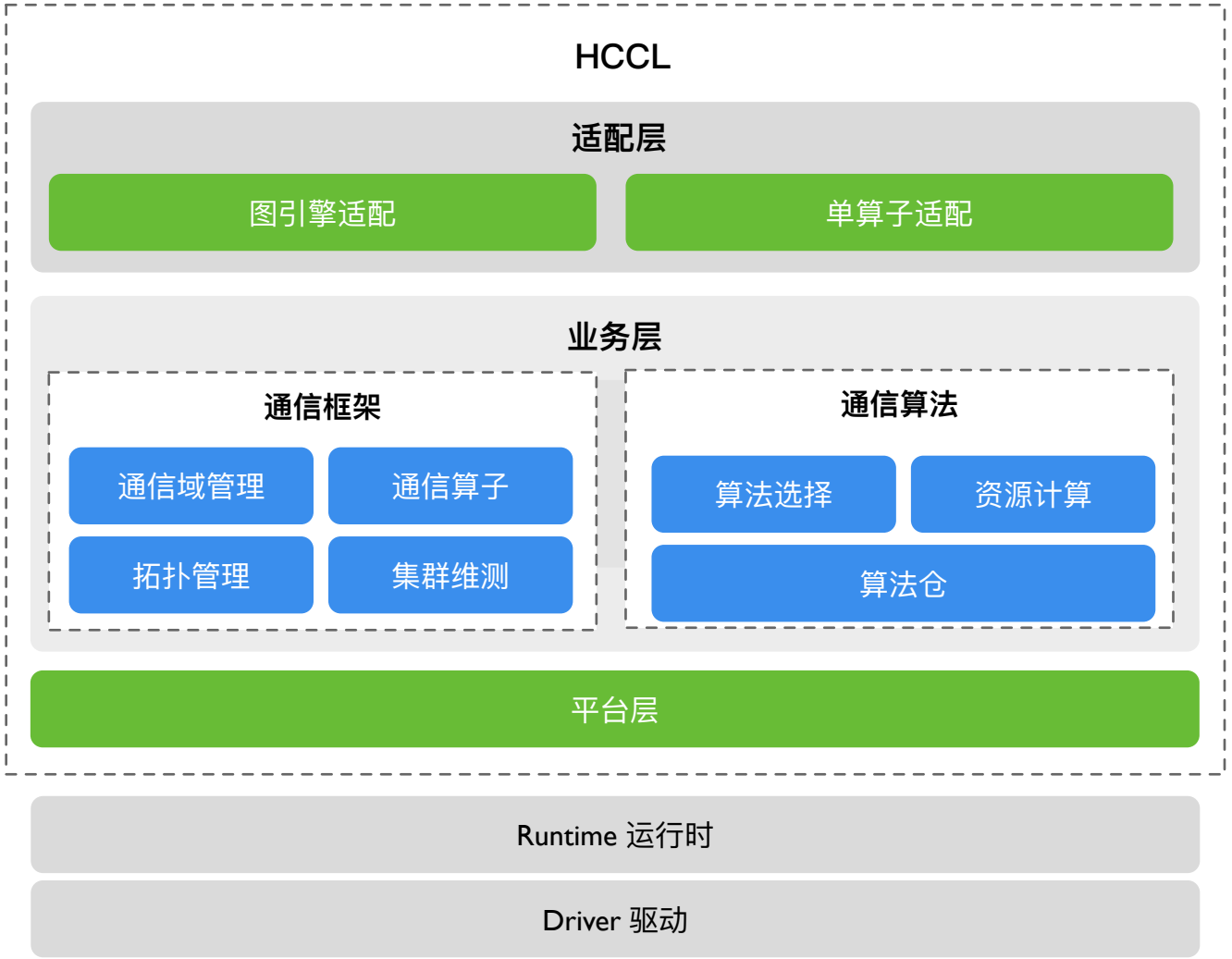
8166 6-24

昇腾AICore快速计算矩阵的秘密被打开了！#昇腾 #AI芯片

3949 6-29

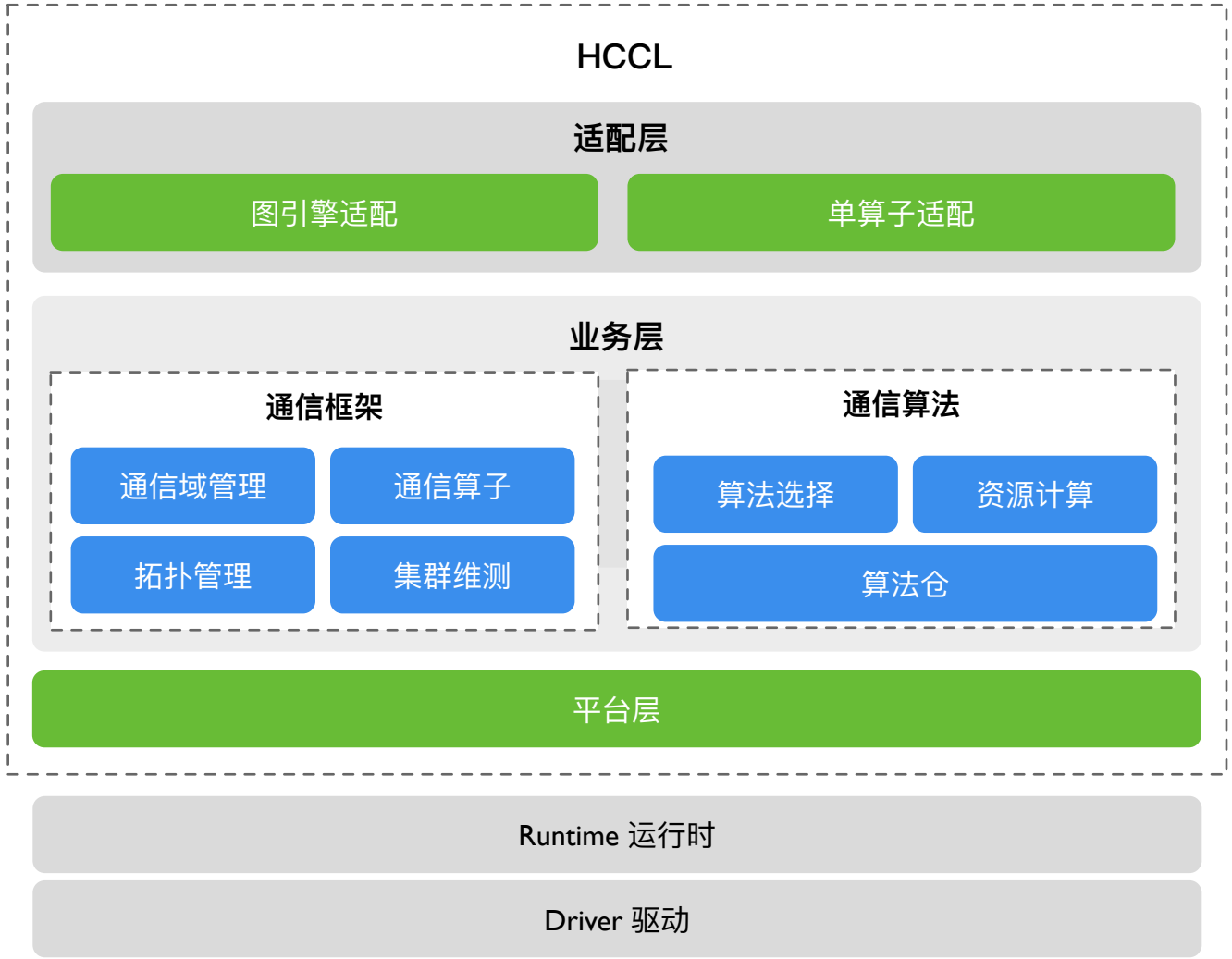
HCCL 基本介绍

- **适配层：**
  - 图引擎与单算子适配，进行通信切分寻优等操作。
- **集合通信平台层：**
  - 提供 NPU 之上与集合通信关联的资源管理，并提供集合通信维测等能力。

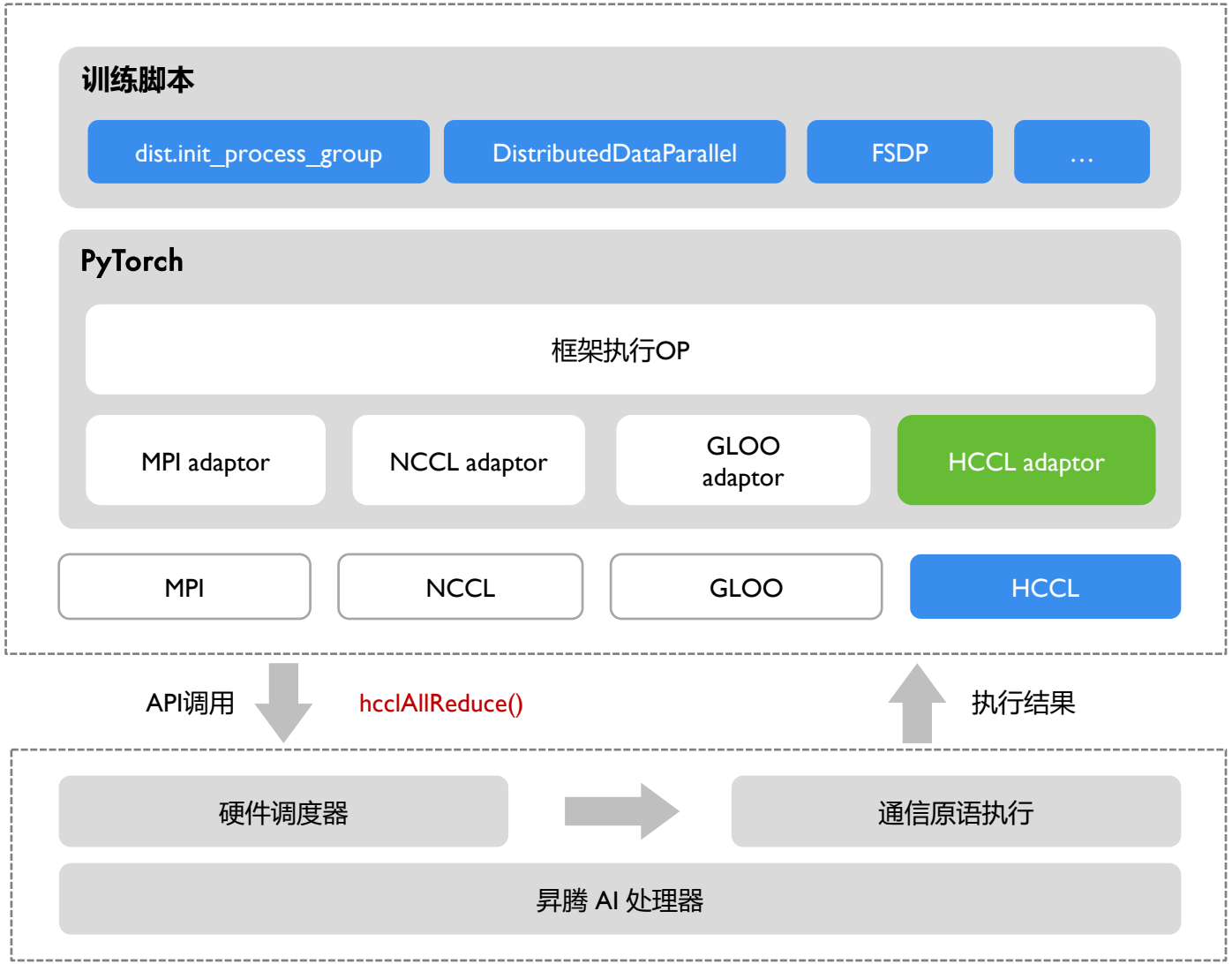


HCCL 基本介绍

- **集合通信业务层**：
  - **通信框架**：通信域 Rank 管理，通信算子的业务串联，协同通信算法模块完成算法选择，协同通信平台模块完成资源申请并实现集合通信任务的下发。
  - **通信算法**：作为集合通信算法的承载模块，提供特性集合通信操作的资源计算，并根据通信域信息完成通信任务编排。



HCCL在 AI 框架中位置

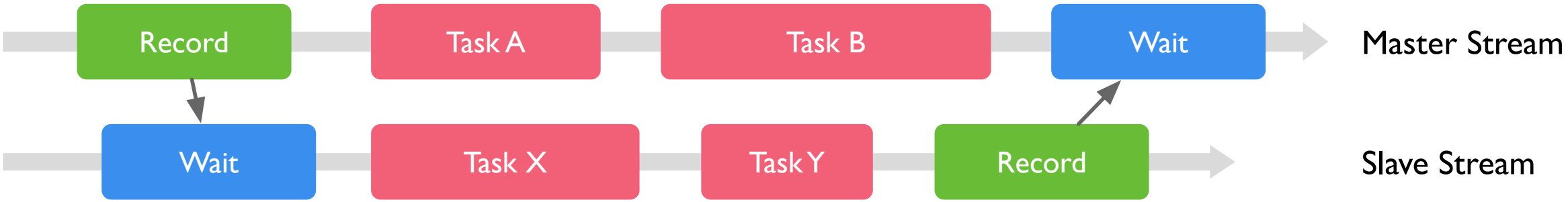


02. HCCL

通信基础概念

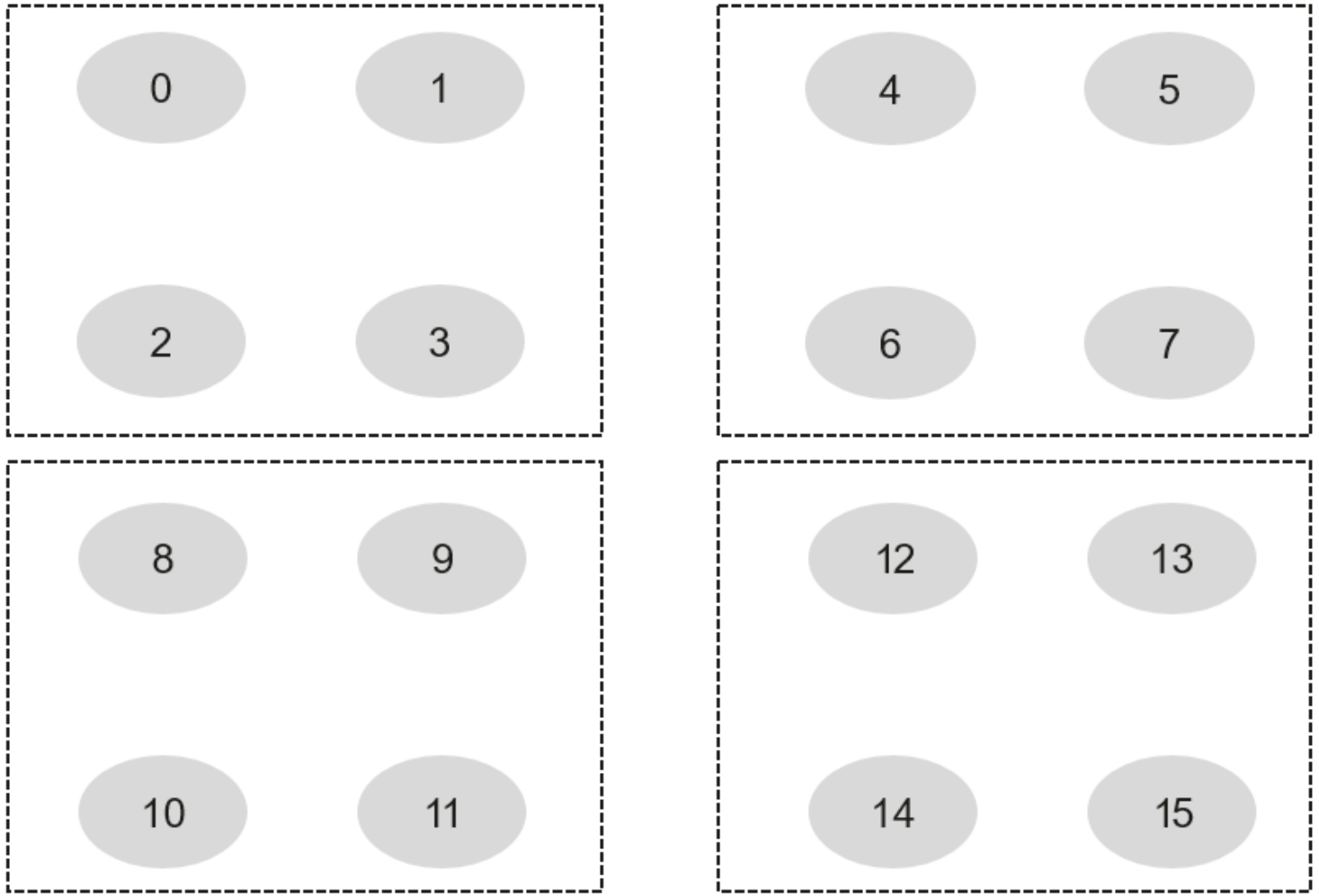
Stream

1. 流（Stream）属于硬件资源，承载待执行 Task 任务序列，其中Task 可以是 DMA 操作、同步处理 or 算子执行等。
2. Stream 流上 Task 序列按顺序执行，流间通过 Record/Wait 两个匹配 Task 进行同步，保持多 Stream 之间的同异步。



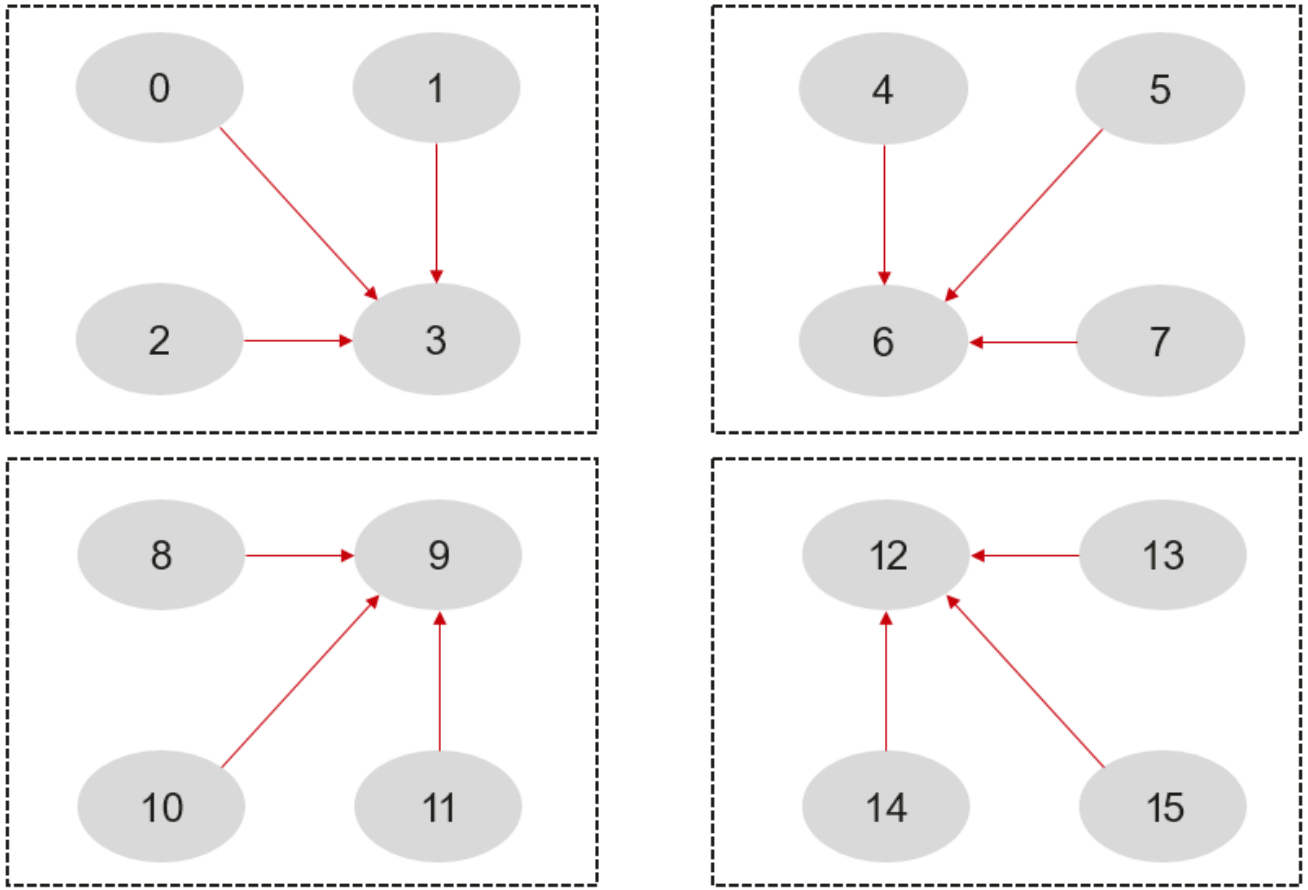
通信域 & 子通信域

- 假设有一个 4*4 通信域，一个方框表示一个节点（Server），一个节点内有 4 个 Rank（NPU），Rank 编号从 0 ~ 15。
- 按照 2 层 All-Reduce 算法，节点内和节点间进行分层，Server 内有一个子通信域，Server 间有一个子通信域。



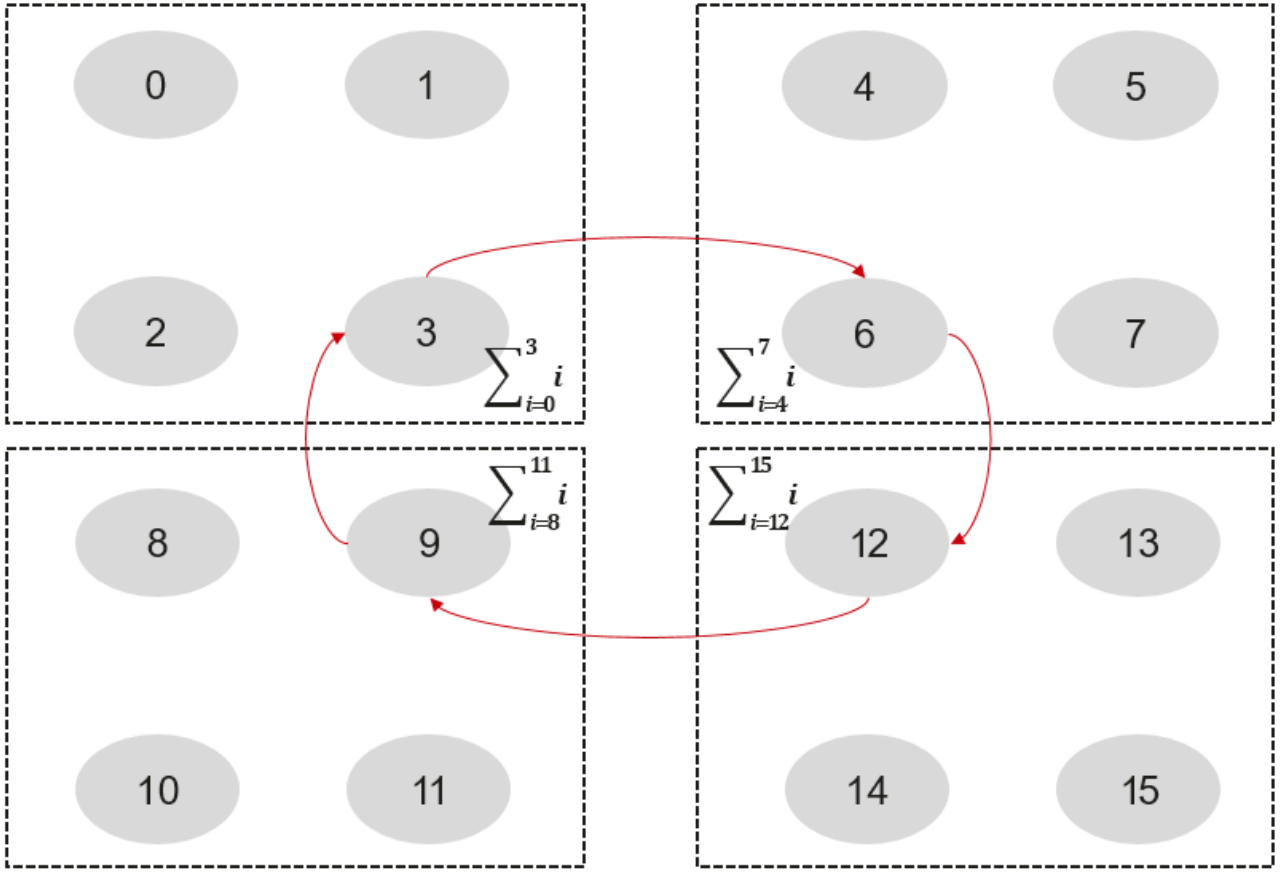
2 层 Ring All Reduce 算法

- 1、先在 Server 内执行 Reduce 操作



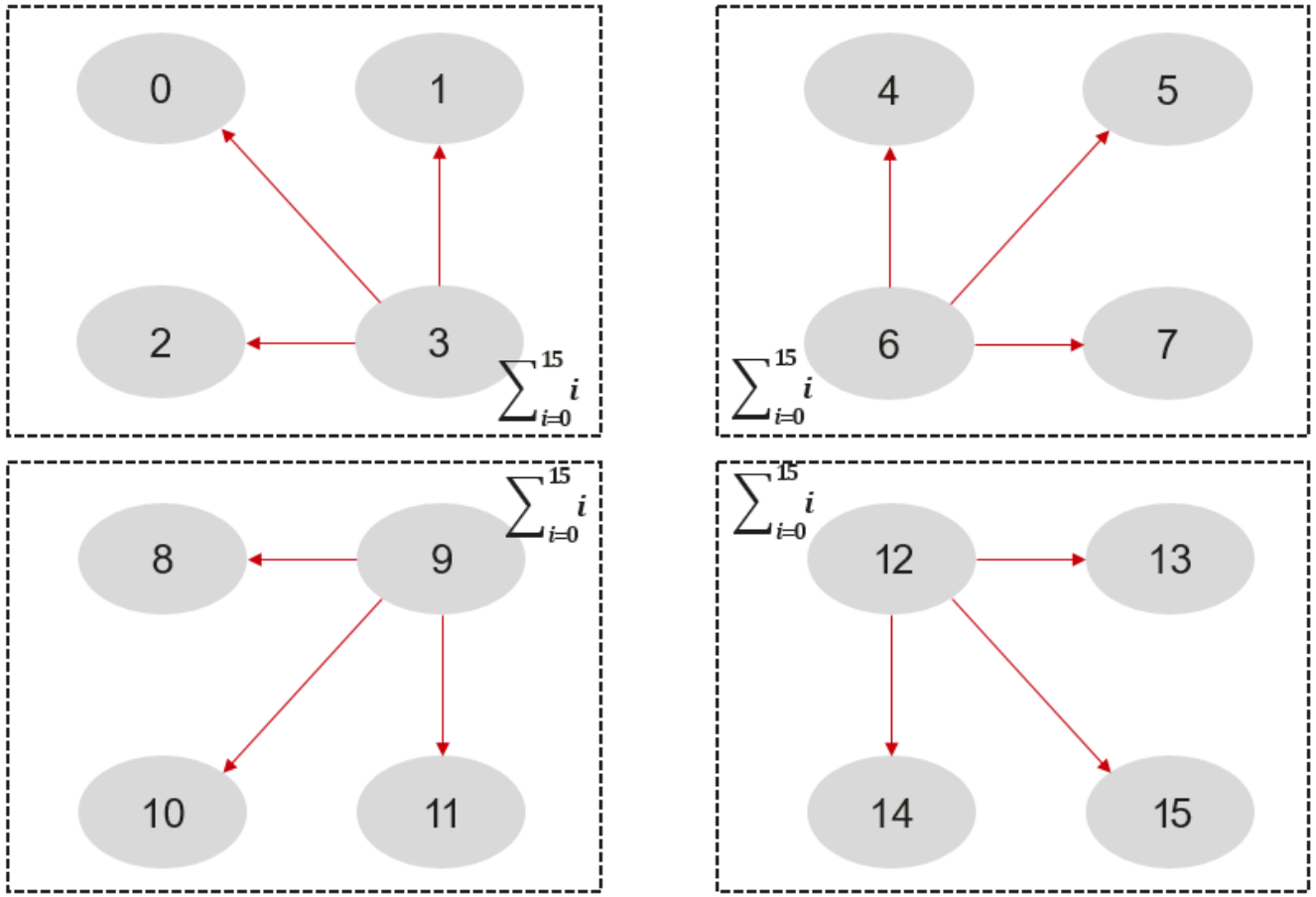
2 层 Ring All Reduce 算法

- 2、Server 间执行 Ring 算法 All Reduce



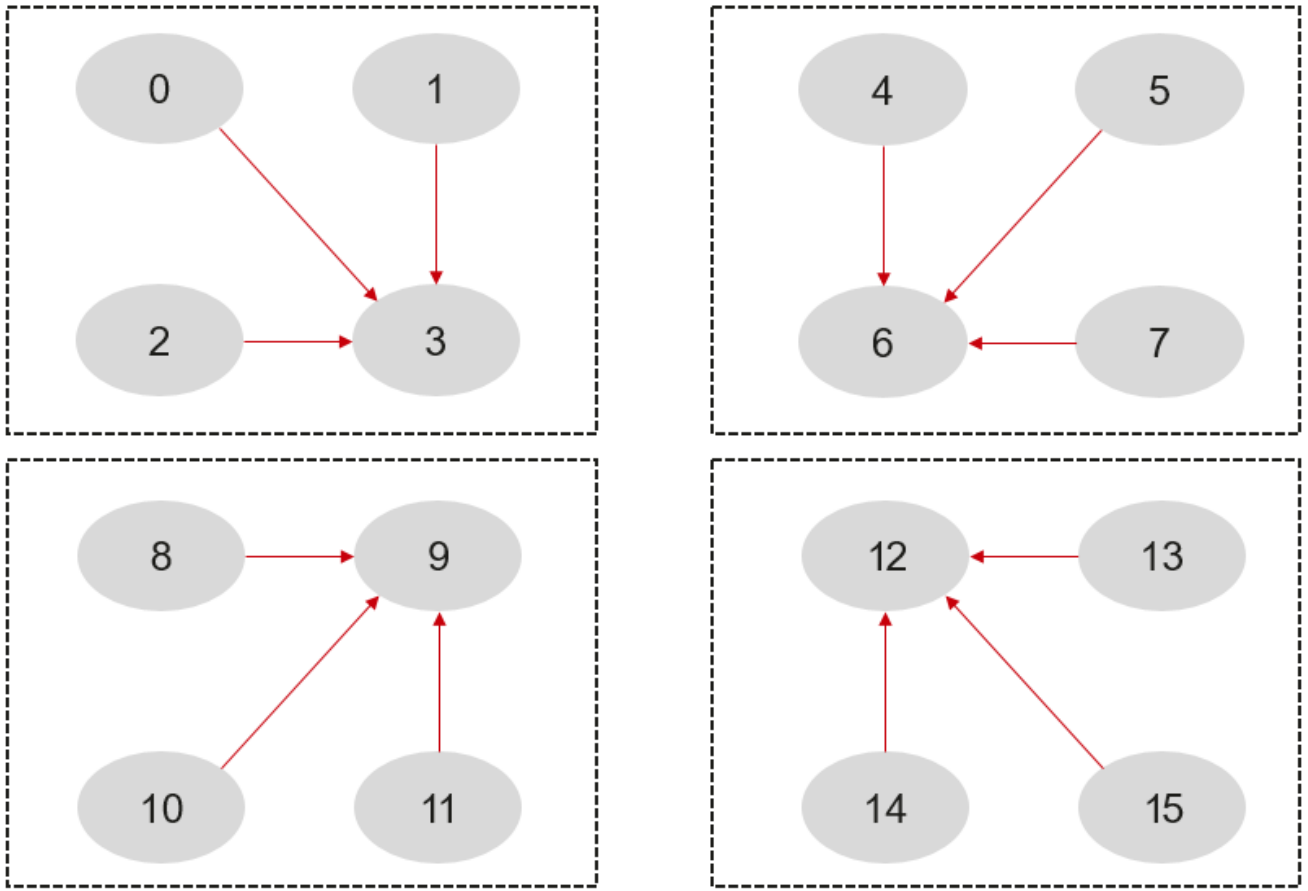
2 层 Ring All Reduce 算法

- 3、Server 内执行 BroadCast



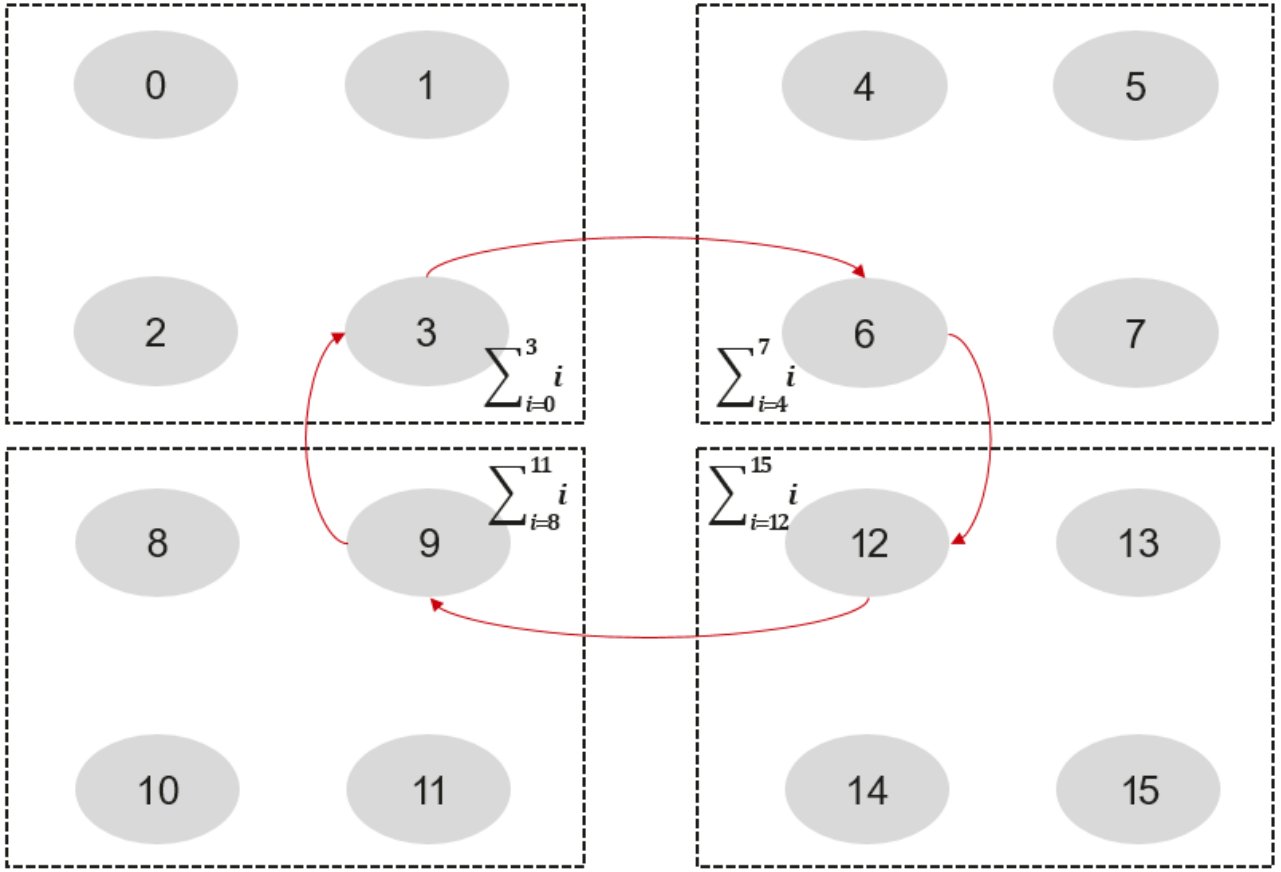
2 层 Ring All Reduce 算法

- 三步执行后，4*4 All Reduce 执行完成，每个 Rank 上都获取 0~15 共 16 个 rank所有数据。



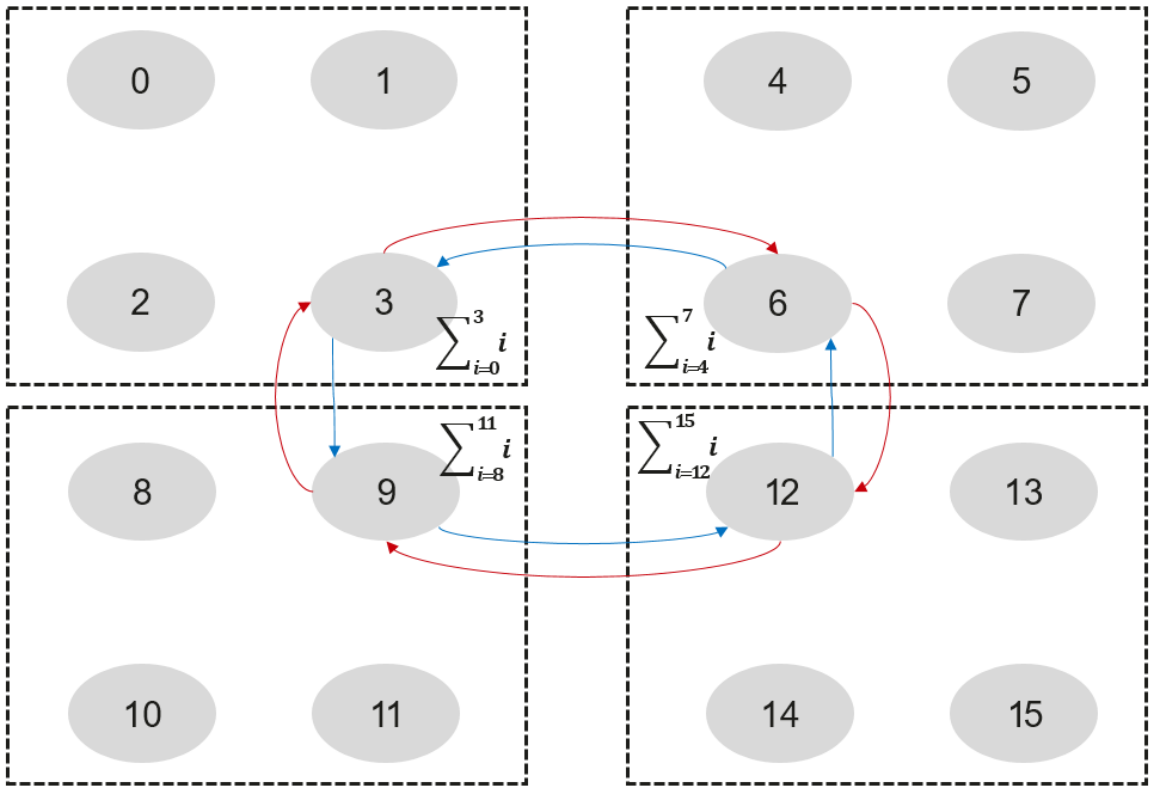
2 层 Ring All Reduce 算法

- 第二步执行节点间算法使用一个 Ring 环



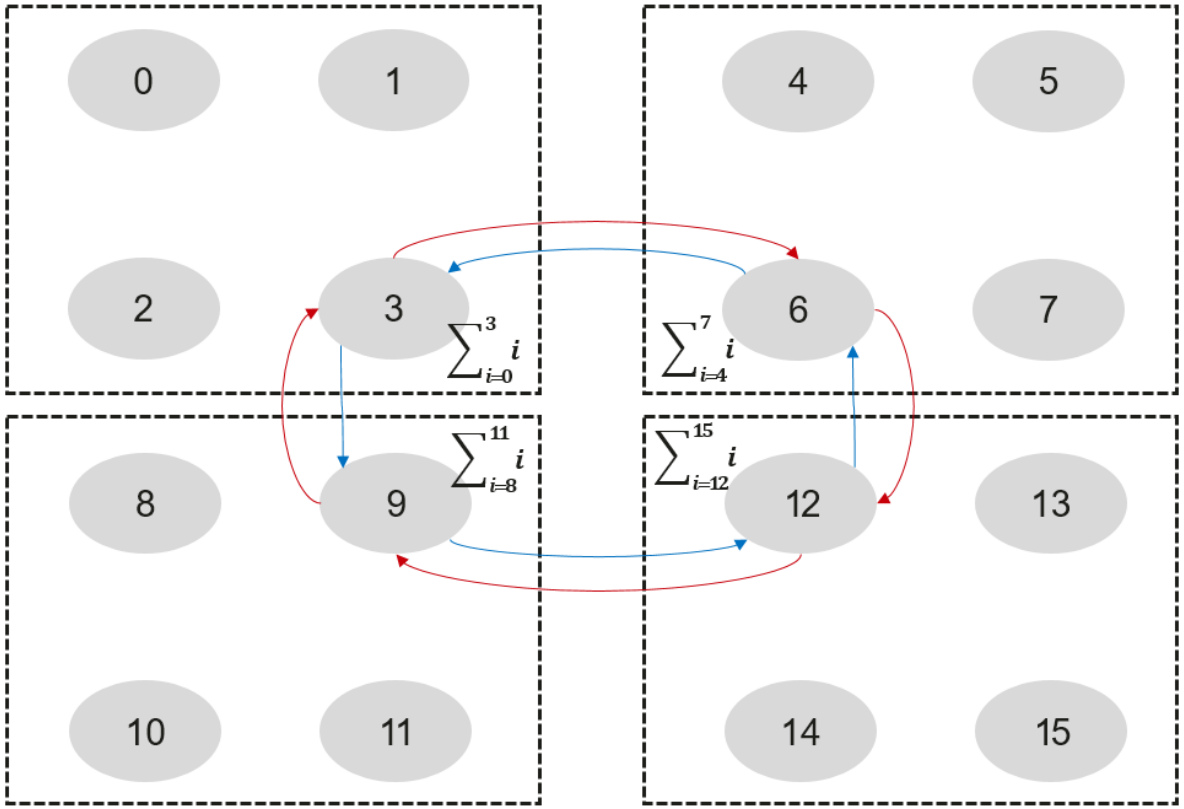
优化 Ring All Reduce 算法

- 将通信域链路另一个方向也利用起来，组成两个 Ring 环。待传输数据分 1/2，分别在两个 Ring 环上传输；



优化 Ring All Reduce 算法

- 此时节点间存在两个子通信域。HCCL 子通信域内，每个 Rank 有一个子通信域内编号（subCommRank），如 3、6、9、12 四个 userRank 对应 subCommRank 分别是0、1、2、3。



03. HCCL

开发流程

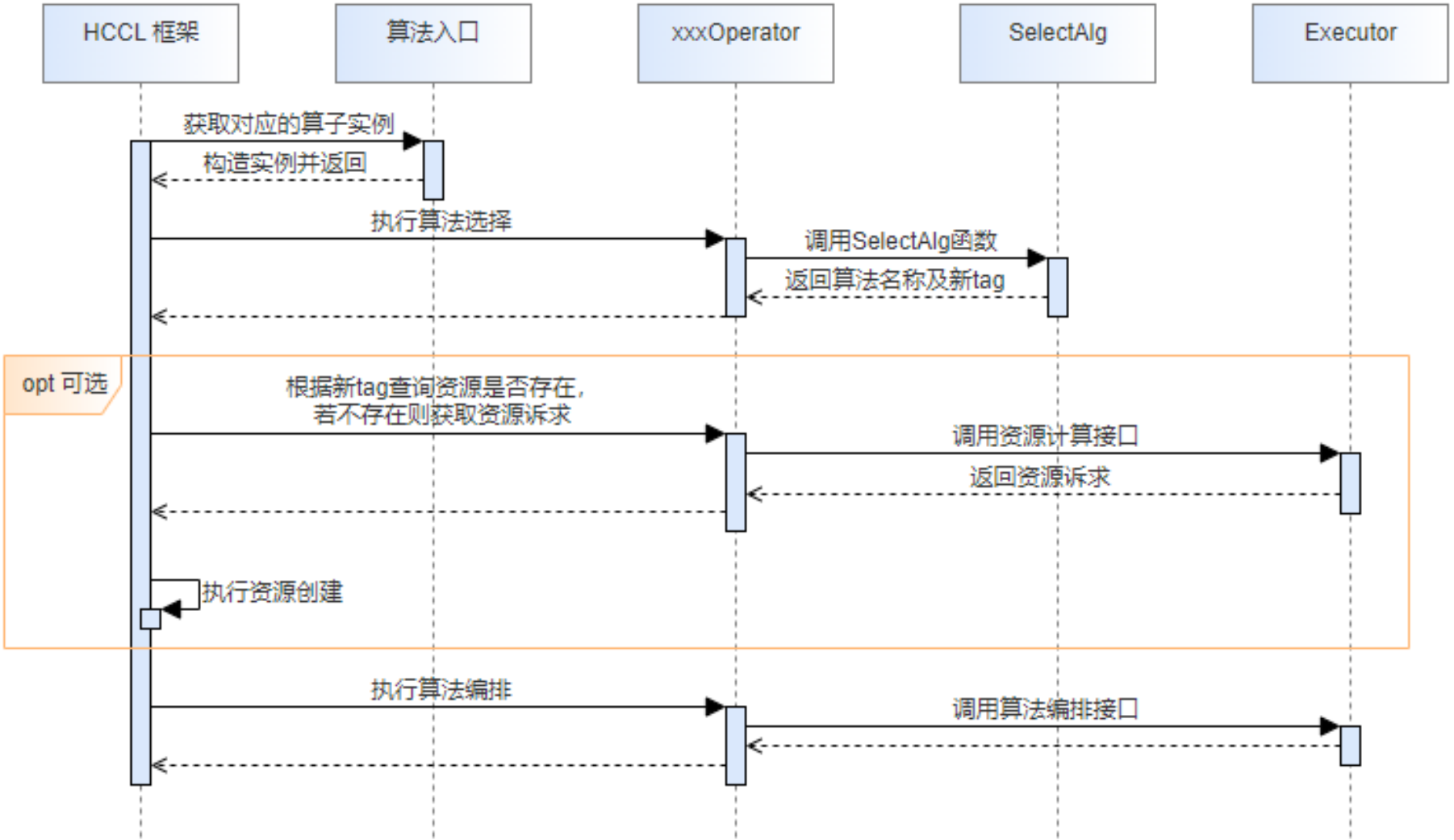
HCCL 通信算法开发流程

- 通信算法主体执行流程可划分为 5 步：

1. **实例化**：HCCL 根据通信原语的类型构造出对应实例
2. **算法选择**：HCCL 调用实例进行通信算法选择，并返回将要执行算法名字和标志资源 newTag 字符串
3. **查询资源**：HCCL 根据标志查询对应资源（调用实例资源计算接口获取需要资源诉求，HCCL 根据诉求创建资源）
4. **算法编排**：HCCL 传入通信算法执行需要的资源，并执行对应 Stream 的算法编排
5. **提交 Task**：算法编排执行过程中，通过集合通信平台层 API，提交要执行 Task 任务

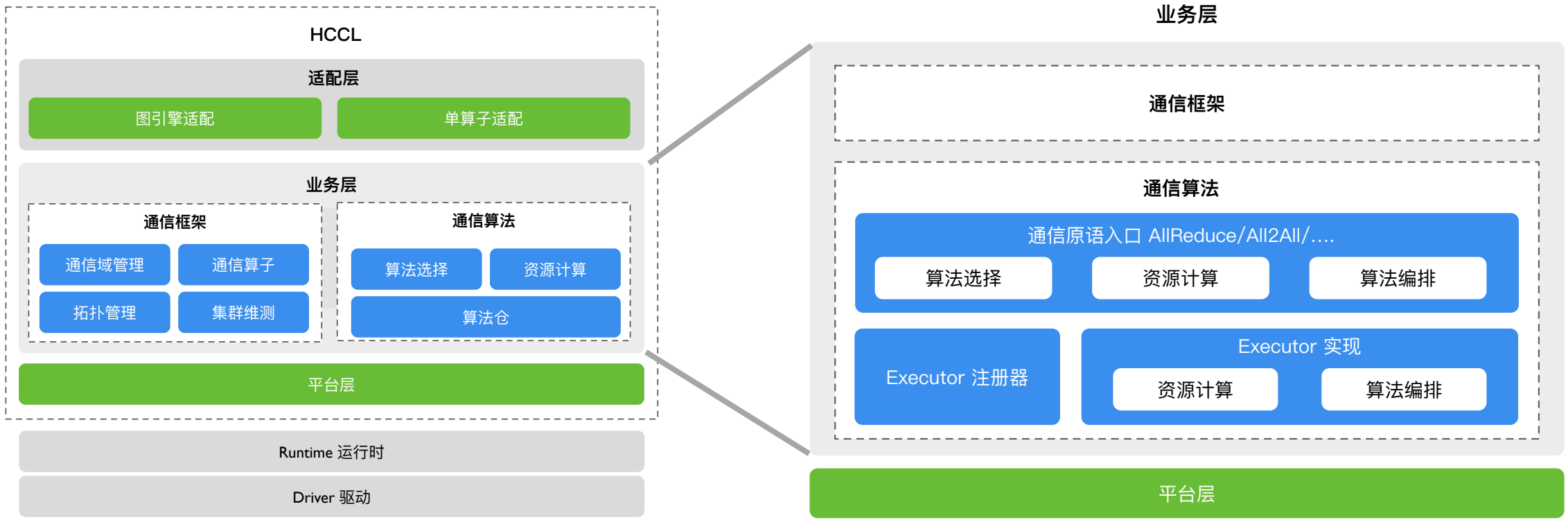
- 上述流程 Step 2 ~ 4，是执行通信算法必经步骤。添加新算法，对应需要修改代码对应步骤。

HCCL 通信算法开发流程



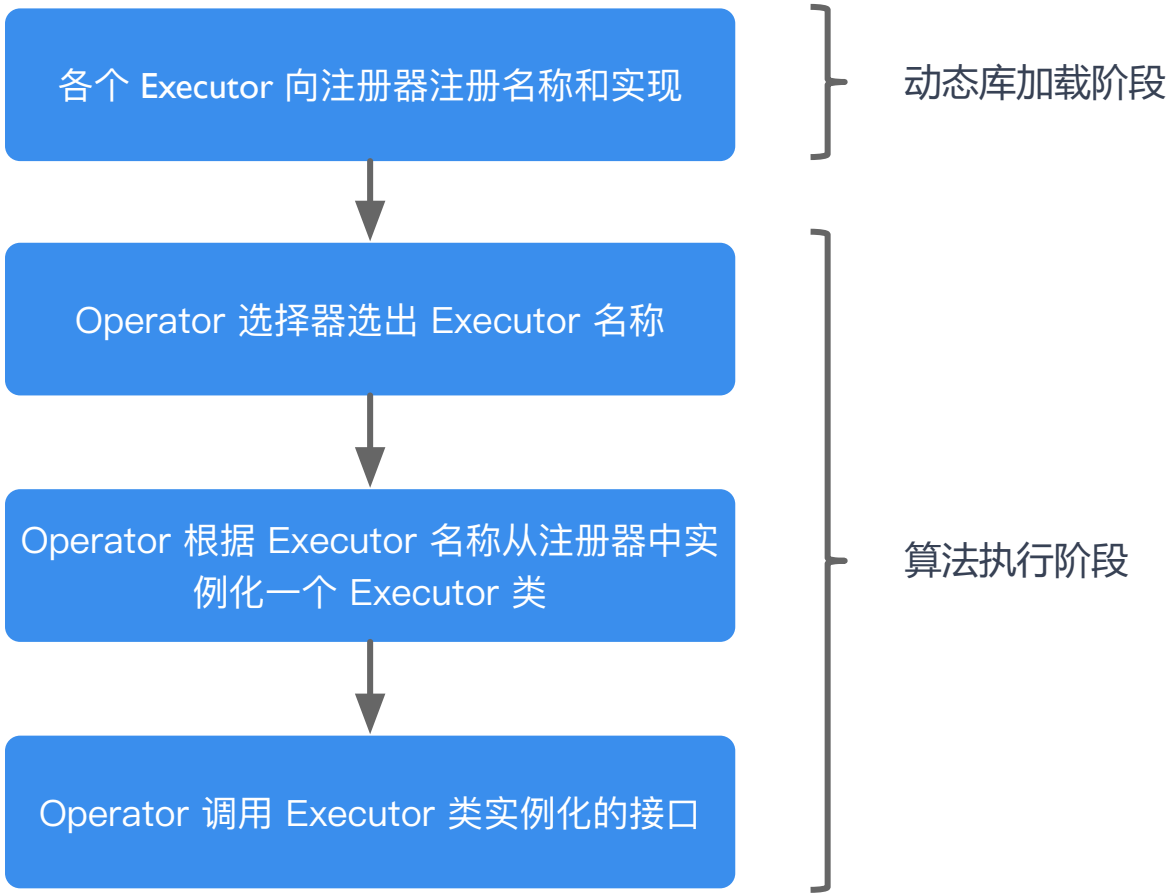
HCCL 通信算法开发流程

- HCCL 算法库向上对接通信框架，向下调用集合通信平台层API。算法库通过 Operator 对象来对接框架，Operator对象对框架主要提供三个 API：算法选择 、资源计算、算法编排。



HCCL 通信算法开发流程

- 算法库提供 统一 Executor 注册器，内置算法 or 自定义算法都通过 Executor 向算法库进行注册。



HCCL 通信算法开发流程代码 review

- **注册宏代码**
  - src/domain/collective_communication/algorithm/impl/coll_executor/registry/coll_alg_exec_registry.h
- **使用示例**
  - REGISTER_EXEC("AllGatherComm", AllGatherComm, CollAllGatherCommExecutor);

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