推理引擎 OpenPPL 实战训练营

商汤 sensetime

商汤自研 AI 推理引擎 OpenPPL 的实践之路

2021年12月10日星期五

课程安排

课程安排	主讲人	课程时间
第一期：商汤自研AI推理引擎 OpenPPL 的实践之路	高洋	2021年12月07日
第二期：编程工作坊：基于 OpenPPL 的模型推理与应用部署	欧国宇	2021年12月16日
第三期：OpenPPL 性能优化：通用架构下的性能优化概要	许志耿	2021年12月28日
第四期：模型大小与推理速度的那些事儿	田子宸	2022年01月06日
第五期：性能调优实战（x86篇）	梁杰鑫	敬请期待
第六期：性能调优实战（CUDA篇）	李天健	敬请期待
第七期：OpenPPL+RISC-V 指令集初探	焦明俊/杨阳	敬请期待
第八期：OpenPPL 在 ARM Server 上的技术实践	许志耿/邱君仪	敬请期待
第九期：量化工具实践	纪喆	敬请期待

实战训练营项目

项目亮点

- **全面讲解**：商汤资深研究员倾情讲授，基础知识一应俱全
- **项目实践**：拒绝纸上谈兵！多个**课程体验 Demo**，学完即可直接上手实操
- **实时答疑**：课程期间设有答疑环节，更有互动社群随时交流
- **专属社群**：9 期课程专属群，主讲人**实时解答**，更有多种互动好礼等你

社群有礼

- **实名社群**：亮出你的身份才能在社群内交到更多朋友哦
- **社群互动**：配合训练营安排，小助手将按时**提醒进展、分享资料、发布任务、解答疑问**
- **打榜好礼**：打卡课程、体验 Demo、参与互动均可收获 **“P 币”**。训练营期间，群内将定期送出**互动好礼**；结营之时，P 币累积在**总排行榜前十**的同学，更会收到**商汤精美定制大礼包**一份

实战训练营

我已经给你安排好了

主讲人介绍

社区ID：高叔叔

高　洋

商汤科技高性能计算技术执行总监

- 目前在商汤担任数据与计算平台高性能计算团队负责人，带领团队研发了业界知名的深度学习推理系统 PPL，并于今年 6 月对社区开源
- 曾任职阿里巴巴核心系统研发部和百度系统部，研发互联网高性能计算基础设施
- 曾任职中国工程物理研究院高性能数值模拟软件中心，研发高性能科学计算基础框架

课程提纲

第一期课程将介绍 PPL 的发展历程、基本结构、典型的优化方法、简单的使用方法以及在业务线落地中的使用体验，并将简述 OpenPPL 作为 PPL 的开源版本和未来面向社区的发展规划。

- PPL 简介
- PPL 的技术实践
- PPL 支持的明星产品业务
- OpenPPL 的未来发展
- Q & A

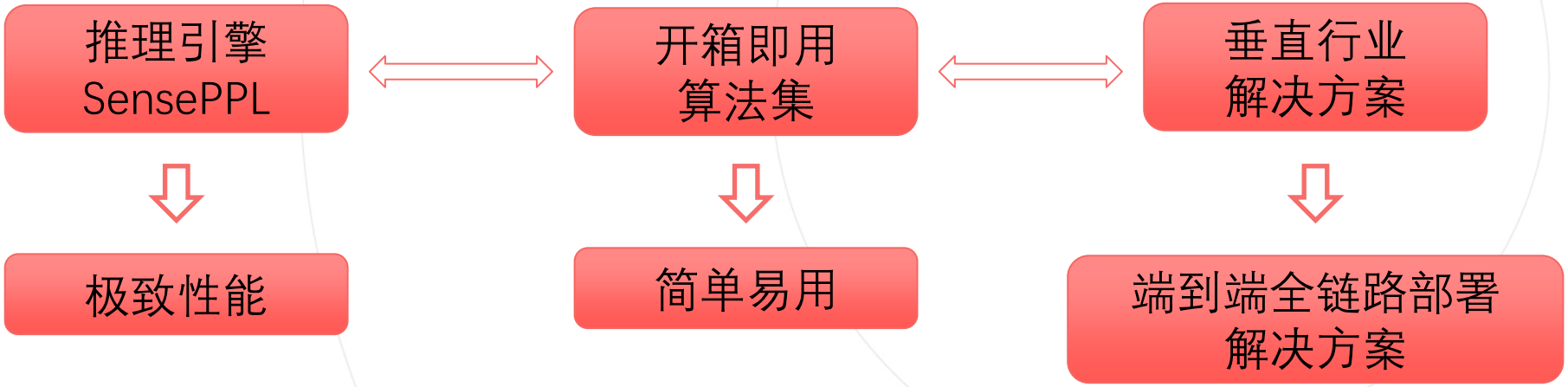
OpenPPL 的前世今生

商汤自研 AI 推理引擎的实践之路

什么是 PPL

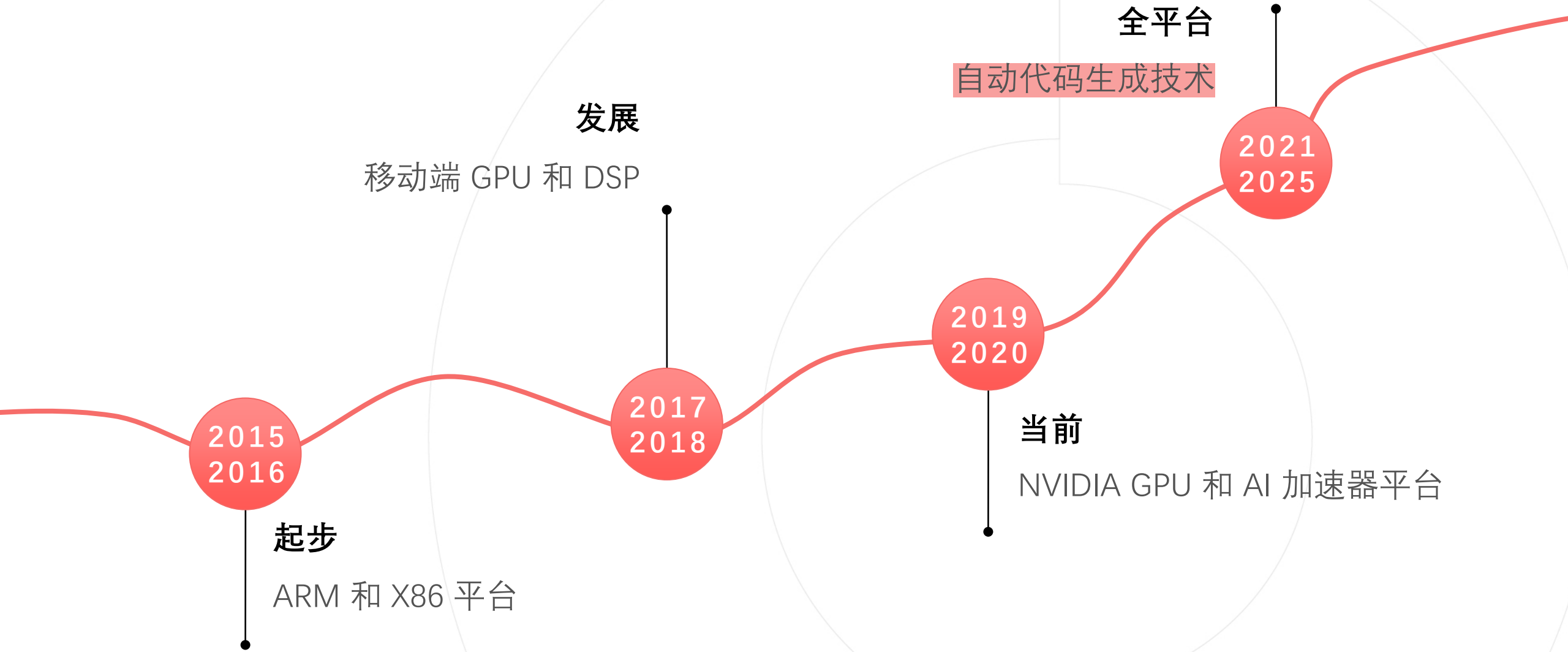
SensePPL 是商汤自主研发的深度学习推理框架平台。

它能够让人工智能应用高效可靠地运行在现有的 CPU，GPU，DSP 和 NPU 等计算平台上，覆盖市面上很大部分的主流芯片产品。在人工智能加速发展和算法训练相对成熟的时代，在产品业务侧的推理部署，成为各家公司落地人工智能的重要技术基础。目前 SensePPL 为至少 5 亿用户提供人工智能推理服务，覆盖安防，金融，手机，娱乐互联网，智能硬件和智能驾驶等广泛的应用场景。



GitHub: https://github.com/openppl-public

PPL 的技术发展历程

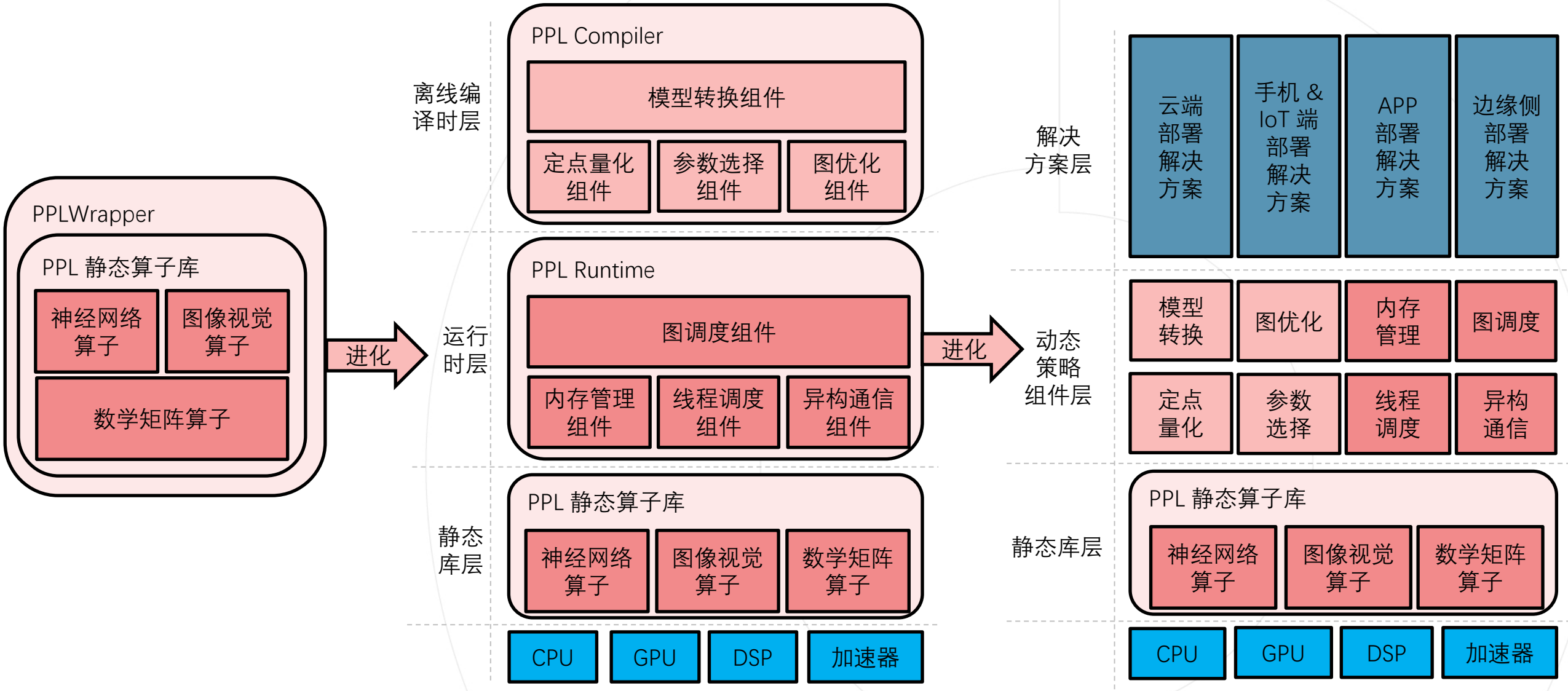


GitHub: https://github.com/openppl-public

为什么要开源 OpenPPL

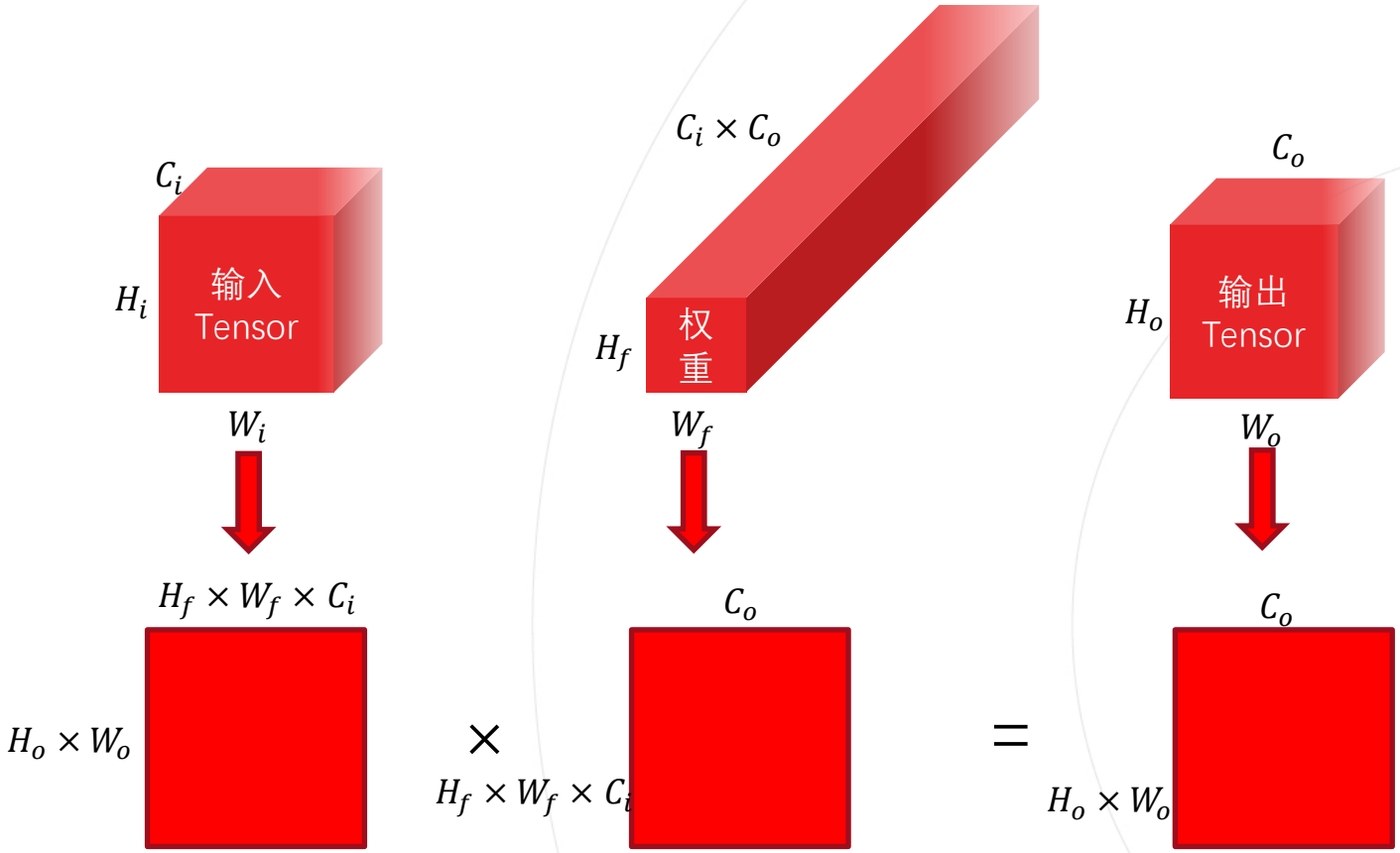
- 将商汤多年在模型推理部署的经验反馈给业界，提供多一种选择
- 一站式的 AI 模型部署解决方案工具链
- 对外接口遵照业界标准的接口
- 多后端支持
  - CPU，GPU，DSP，DSA
- 模型从训练到部署的无痛转换
  - 保证精度
  - 保证工程指标（速度，内存，存储）
- 作为平台寻求和各方面潜在客户的合作

PPL 的框架设计演进



GitHub: https://github.com/openppl-public

卷积与矩阵乘法的关系



- 输入输出格式：NCHW
- 其他参数
  - Stride：St_h，St_w
  - Padding：P_h，P_w

几种卷积高性能计算方法

- 基于矩阵乘的算法
  - 基本思路是将卷积计算展开成矩阵乘法
  - Caffe经典算法
    1. Im2col
    2. Gemm
  - Tile-based gemm卷积
- 本质是对 $H\_i \times W\_i$ 这一维做向量化

- 直接计算卷积算法
  - 顾名思义，直接计算卷积
  - 受限于体系结构
  - 对 cache 不甚友好
- 本质是对 C_o 这一维做向量化

- 减少运算量的算法
  - FFT 卷积算法
  - Winograd 卷积算法
- 基本步骤
  - 输入转换
  - 多 batch 矩阵乘法
  - 输出转换
- 本质在于多 batch 矩阵乘法步骤中的乘加运算量小于传统算法

都是基于矩阵乘法的算法
实现框架非常类似

PPL 在 ARM CPU 上的优化

优化模型推理性能

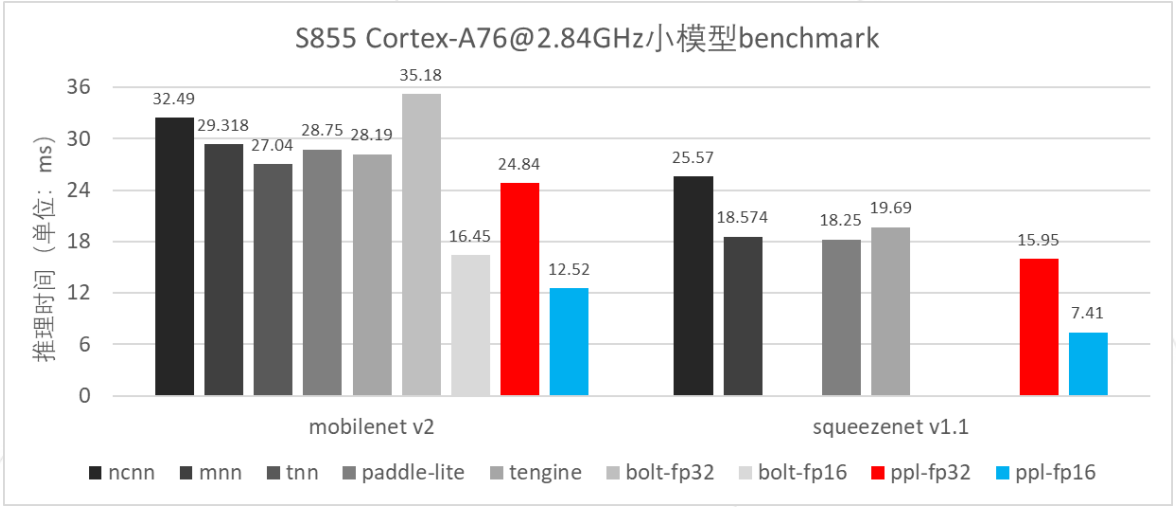
- 针对不同的硬件微架构，设计不同的底层汇编代码，在不同的平台（armv7, armv8…）都能发挥极致性能；
- 深入分析执行逻辑，合并同类操作，减少非计算的 overhead；
- 多种不同的算法，以应对不同场景下的需求。

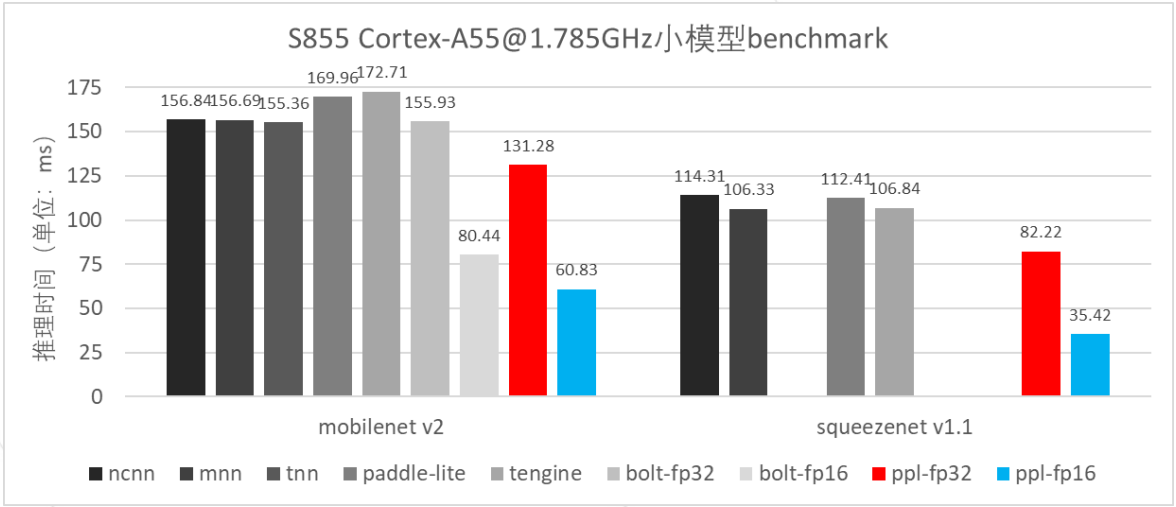
NN/CV算子添加及优化

- 算子性能相比业内开源实现有明显优势，从 20% 到 100% 不等；
- 添加大量商汤自研的自定义算子，以支持内部特殊的使用场景。

测试工具及文档完善

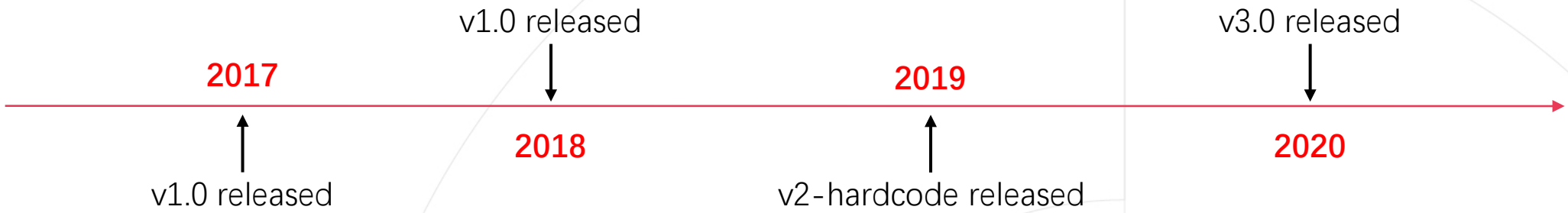
- 完善错误检查等机制；
- 提供高自由度的调用方式以应对不同的使用环境；
- 提供详细的算子实现/使用方式等的说明文档。





GitHub: https://github.com/openppl-public

PPL OCL：商汤自研移动端 GPU 推理框架



支持的硬件平台		
Qualcomm Adreno	Adreno 4xx	~5
	Adreno 5xx	~10
	Adreno 6xx	~20
ARM Mali	Midgard	~5
	Bifrost	~5
	Valhall	~5
Imagination PowerVR	Series6XT	~5
	Series7XT	~5
	Series8XE	~5
	Series9XM	~1

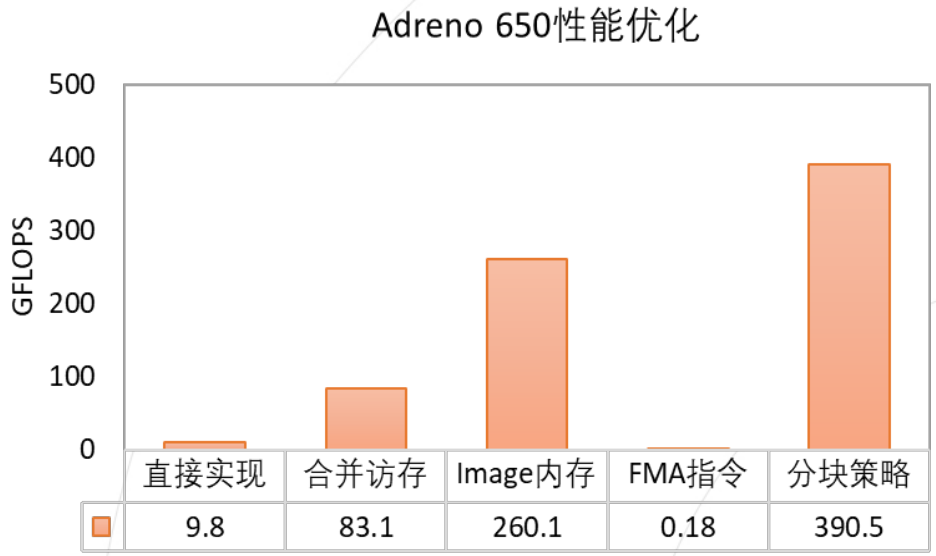
支持的 OpenCL 版本
OpenCL 1.1 full
OpenCL 1.2 full
OpenCL 1.2 embedded profile
OpenCL 2.0 full

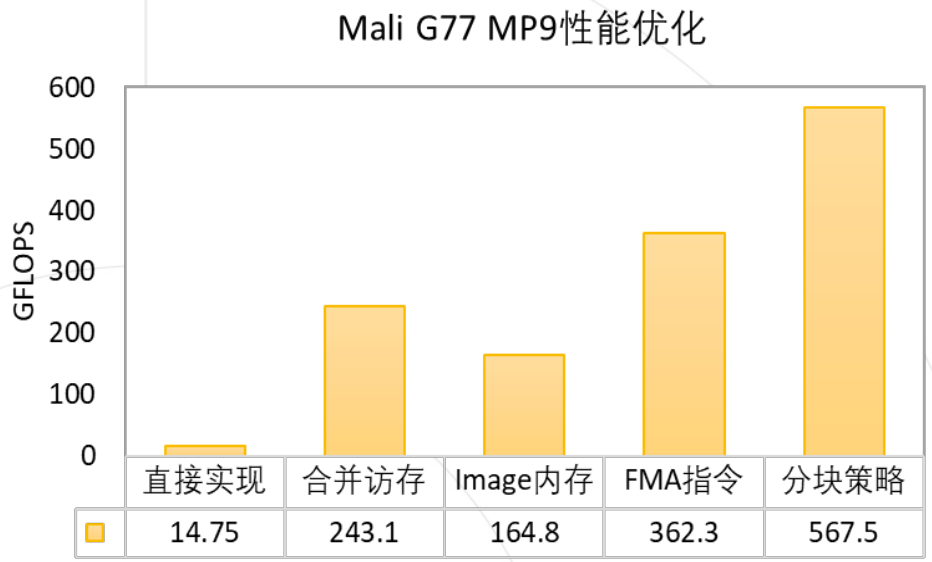
支持的算子	
神经网络	40+
图像处理	50+
代数运算	10+

GitHub: https://github.com/openppl-public

PPL OCL 计算优化：算子调优

- 矩阵乘法优化：
  - 合并访存
  - 使用 Image 内存
  - 使用 FMA 指令
  - 分块策略





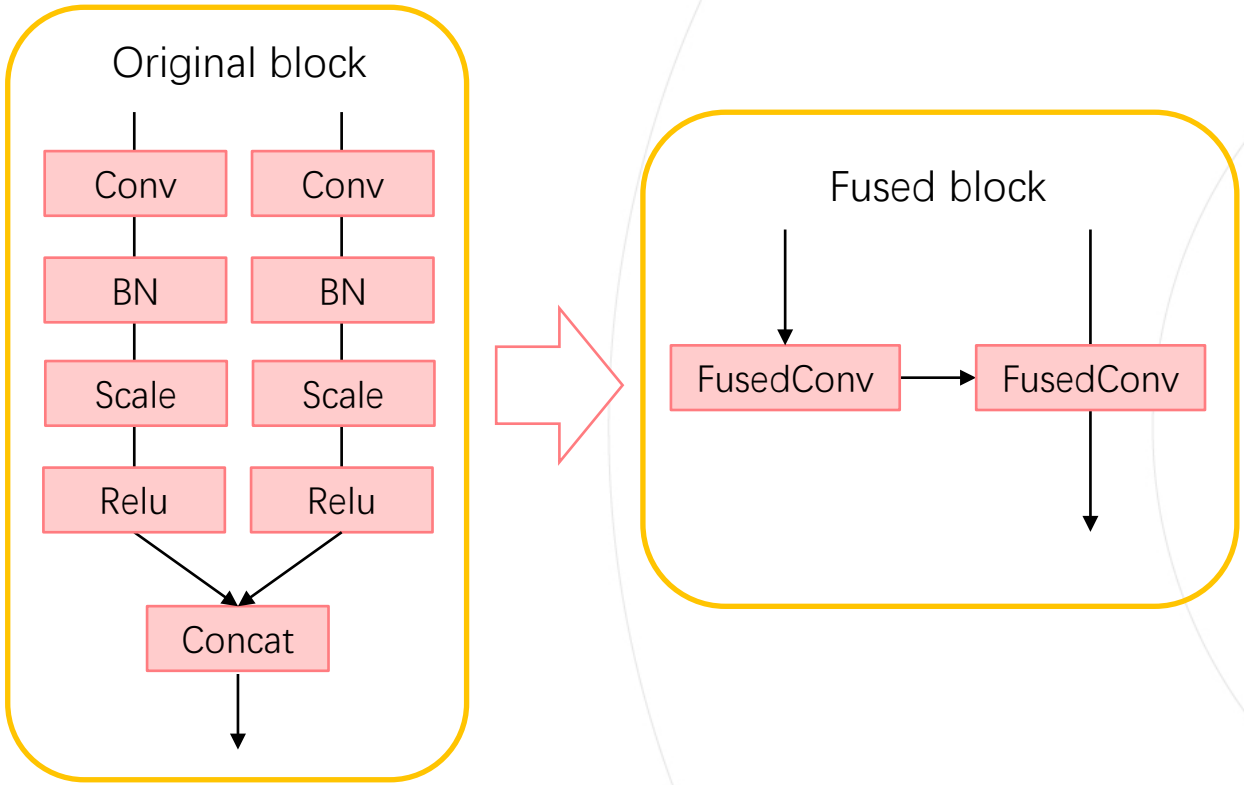
- 卷积优化：

卷积规模		卷积算法
图（w、h）	通道（cin、cout）	
大	大	Winograd
大	小	Direct Conv，Implict GEMM
小	大	Winograd，Inverse GEMM
小	小	Implict GEMM
大/小	group=cin=cout	Depthwise Conv
畸形（如大 w 小 h）	大/小	Implict GEMM

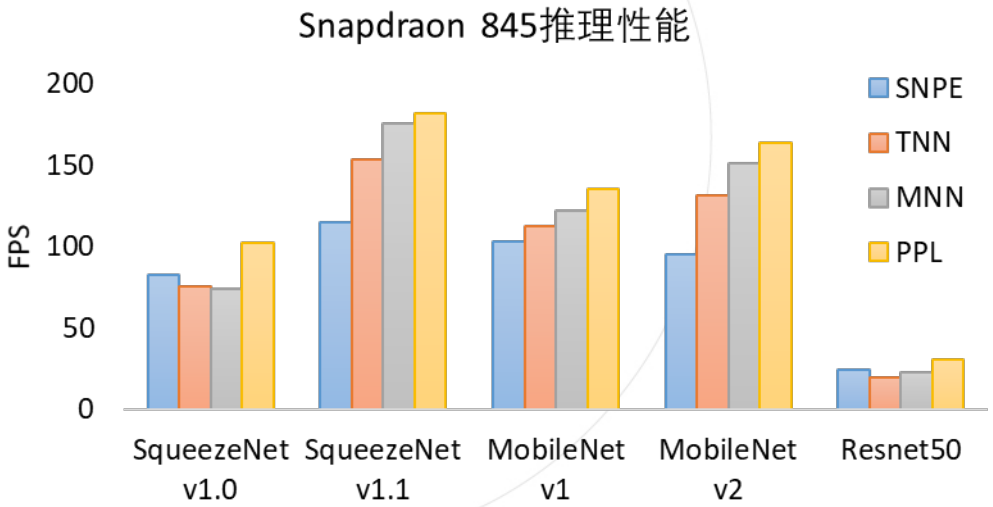
GitHub: https://github.com/openppl-public

PPL OCL 计算优化：框架调优

- 丰富的图融合模式
  - 减少全局访存，减少 kernel launch
  - BN, Scale, Relu, PRelu, Relu6, Eltwise, Concat...



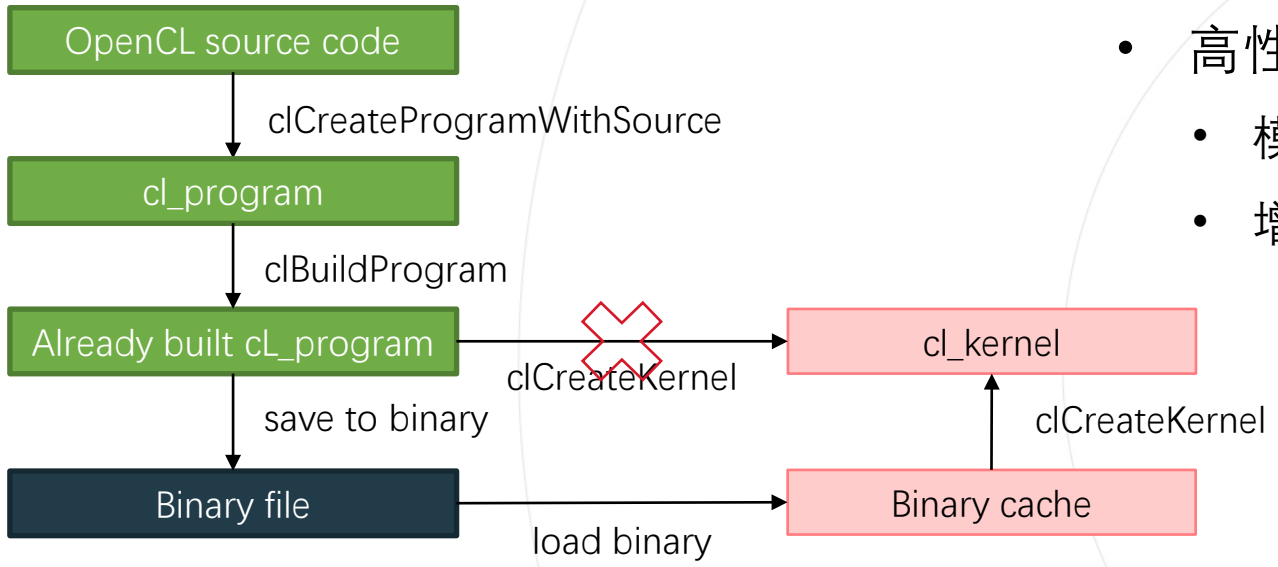
- 利用 3d Image 数据排布消除 padding
  - 减少全局访存，减少 kernel launch
  - 数据排布：nchw4，channel 对齐到 4，每 4 个 channel 对应于 CL_RGBA 格式的 image 元素
  - Sampler：进行 clamp 或 mirror 等模式的边界处理



GitHub: https://github.com/openppl-public

PPL OCL 预处理优化

- Binary Cache
  - 消除 OpenCL 运行时编译机制对初始化时间的影响



- 离线算法选择
  - 目标机 Try-run：选择最优算法，最优分块策略；
  - 数据排布转换：filter channel 补齐、转置等。
- 高性能vs.通用性
  - 模型初始化时间从数 s 降低到数十 ms；
  - 增加模型版本管理的复杂性。

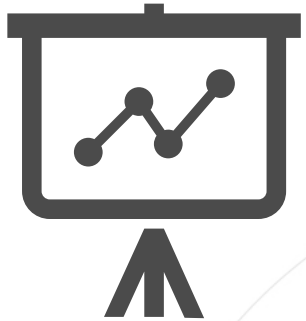
2018 年 PPL 在 ARM CPU 和移动 CPU 支持过的业务场景

GitHub: https://github.com/openppl-public

PPL CUDA 推理引擎

功能全

- **全自研：**计算库、数学库、推理库、CV 库等全部自研
- **全覆盖：**支持Pascal/Volta/Turing 等多种架构，支持 FP32/FP16/IN8 等多种数据类型，支持检测/分割/分类/超分等多种算法模型



高性能

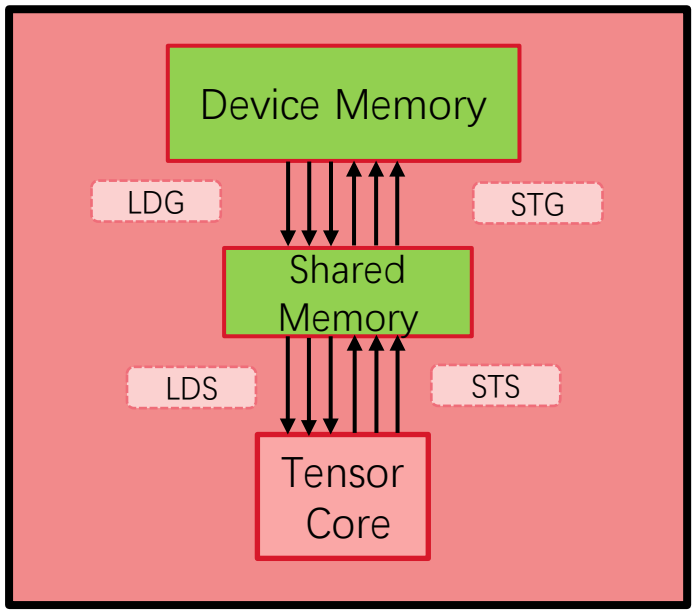
- **算子极致优化：**
  - Conv/Gemm 等算子优化
  - 访存密集型算子
  - 业务网络算子
- **网络层面优化：**
  - 层间融合
  - 最优排布选择

易用性

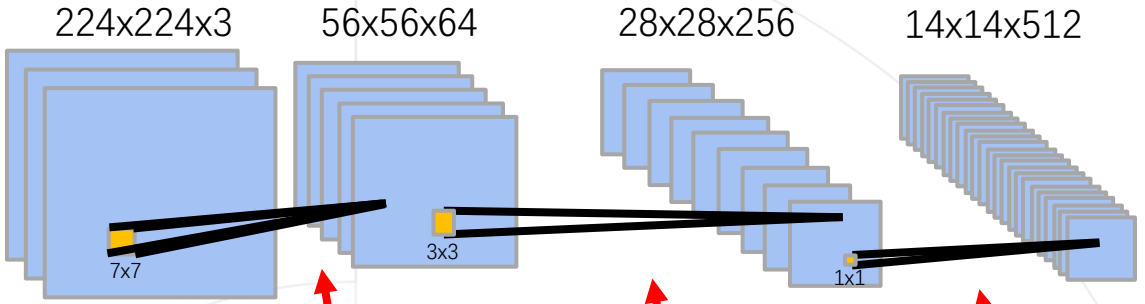
- **精度灵活配置：**推理引擎提供逐算子配置精度接口
- **减少初始化时间：**提供序列化功能
- **减少库体积：**提供库裁剪功能
- **降低显存占用：**内存共享技术

GitHub: https://github.com/openppl-public

TensorCore 上高性能卷积/矩阵乘法的实现



Volta/Turing Tensor Core 单元



架构研究

算法设计

TensorCore上高性能卷积/矩阵乘法的实现

代码生成

- **覆盖各种卷积 shape**
  - 多种分块尺寸的 kernel
  - 多种数据排布支持
  - 多种精度支持

大图少通道类型：
Memory-bound 特点
Indexed_no_smem 算法

大图多通道类型：
Compute-bound 特点
Swizzle_gemm 算法

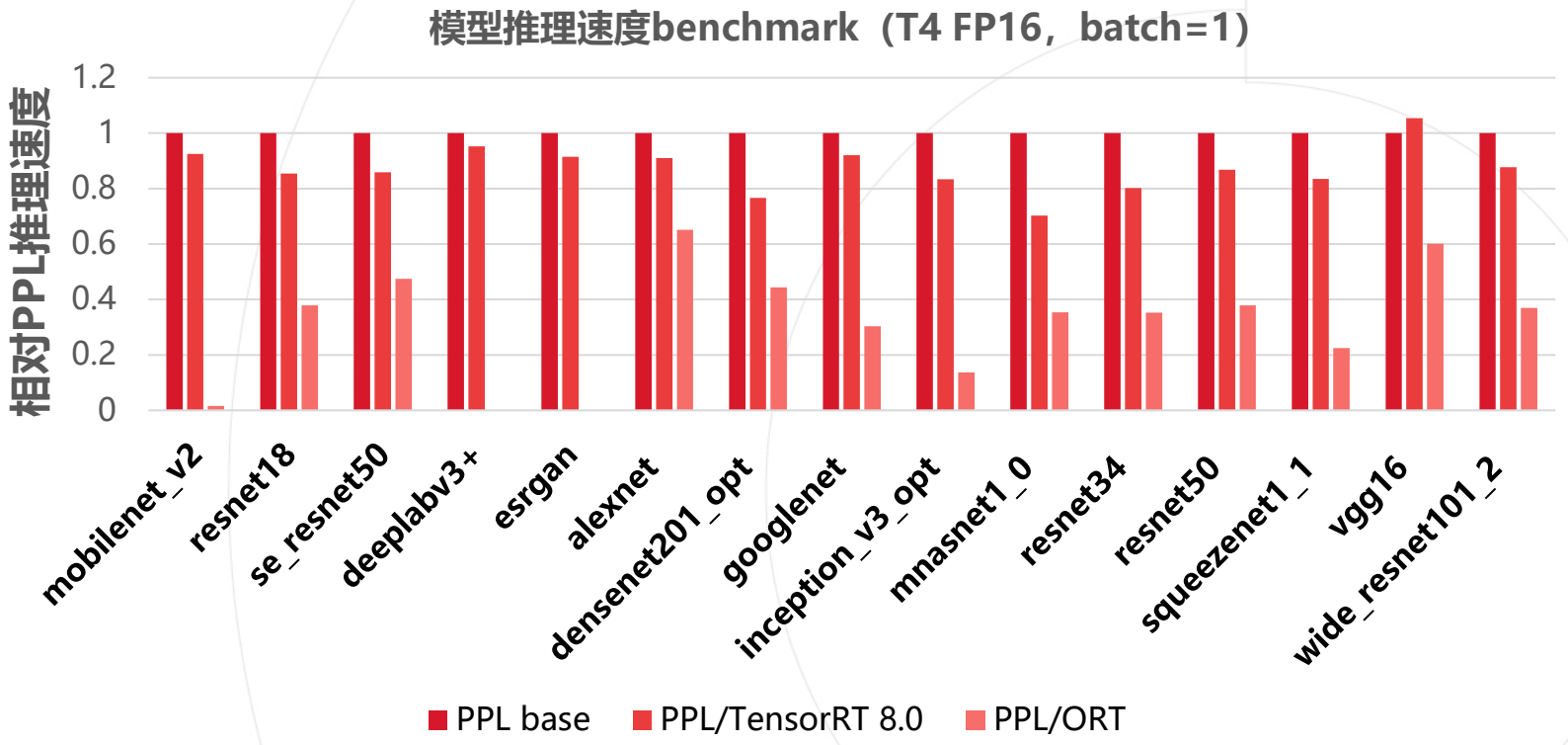
小图多通道类型：
通道数目多特点
2Level_splitk 算法

GitHub: https://github.com/openppl-public

PPL CUDA 网络性能

【常见公开网络推理性能结果】

T4 FP16 的推理性能相较于 TensorRT 8.0，**平均提升 20-30%。**

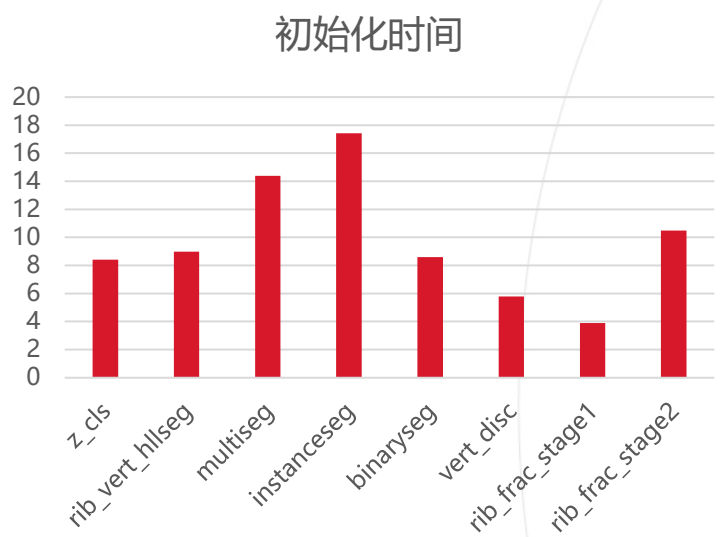


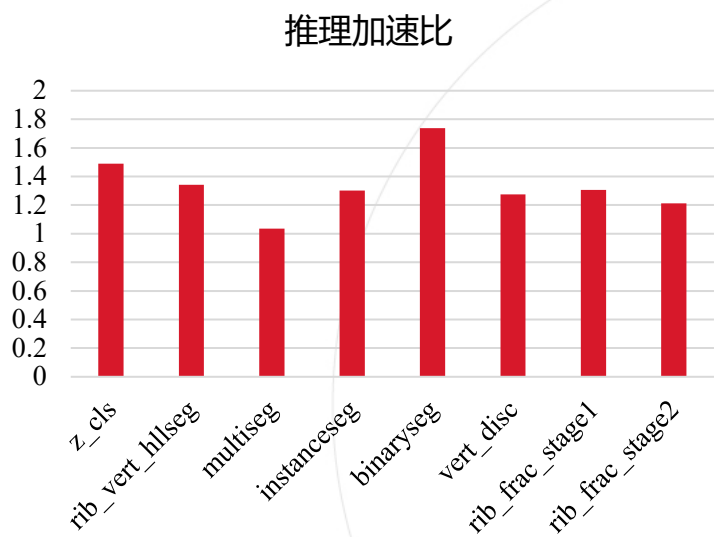
（更多详细结果：https://github.com/openppl-public/ppl.nn/blob/master/docs/en/cuda-doc/benchmark_tool.md）

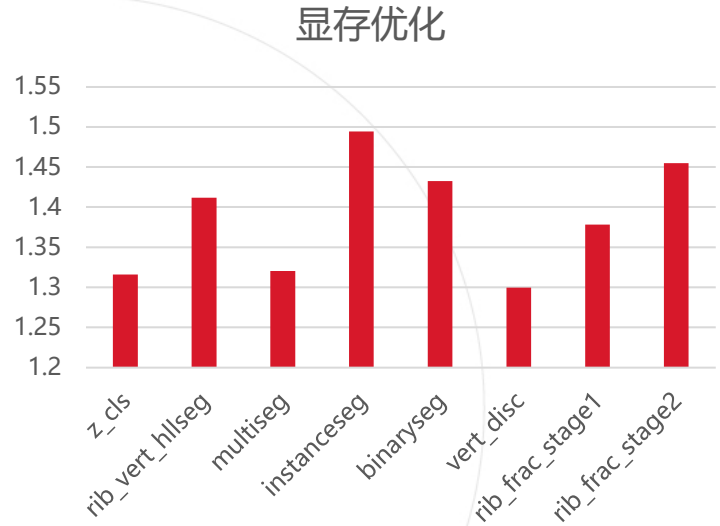
PPL CUDA 业务落地

PPL CUDA 在医疗 SenseCare 中落地，支持肺、肋骨等多个项目检测。

其中**初始化时间降低 4 倍以上，推理速度提升 10-70%，显存降低 30-50%**





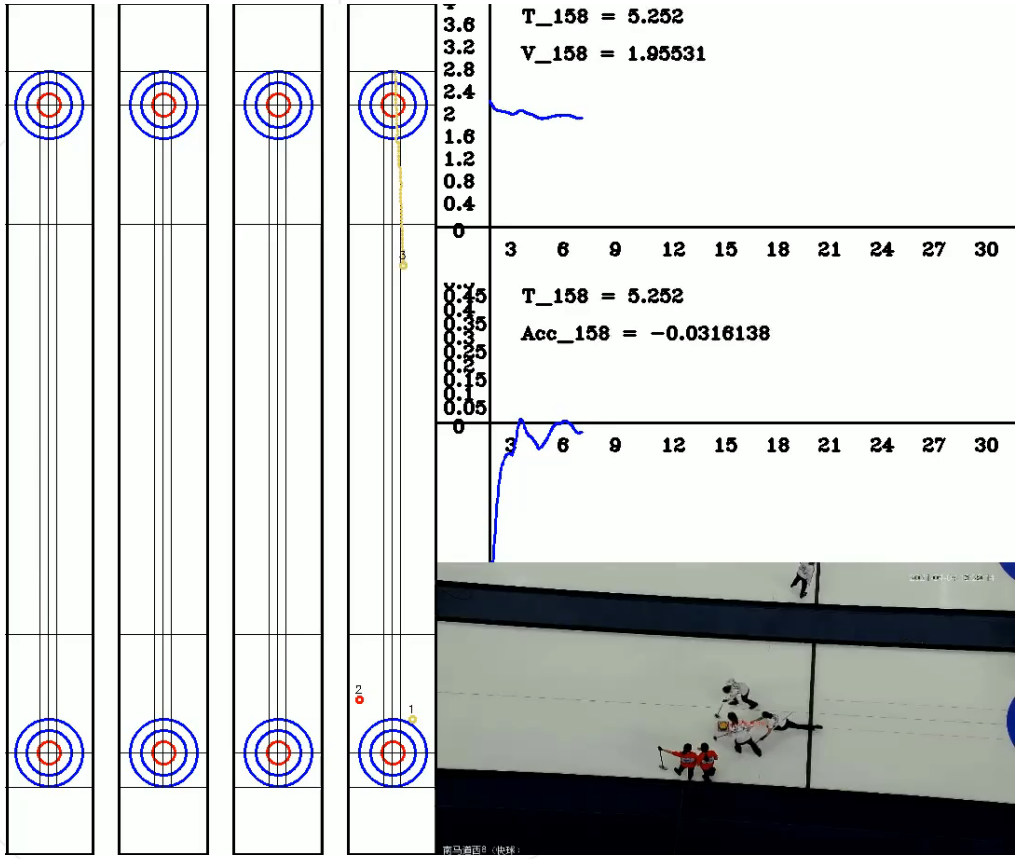


GitHub: https://github.com/openppl-public

PPL CUDA 业务落地

PPL CUDA 在水立方 AR、冬奥会冰壶追踪等多个项目中落地。通过多精度混合优化，水立方特征提取耗时减少 44%，冰壶检测追踪 4Batch 耗时减少 68%，且精度损失极小。

下面是 AR 效果及冰壶追踪的视频片段：



GitHub: https://github.com/openppl-public

PPL Hexagon DSP HVX 技术路线

1. 丰富的中低比特量化方案

- 采用 8bit、16bit 非对称线性量化，支持 8/16bit 混合精度推理，满足各类型 AI 任务精度需求；
- 支持 Per-Layer、Per-Channel 量化方案。

2. 自研算子与汇编级性能调优

- 基于内部量化方案，自研 HVX 神经网络算子；
- VLIW 汇编级细粒度指令编排，结合自研数据排布，发挥 vrmpyz 指令性能；
- 高度复用片上 VTCM/L2 等各级存储层次，辅以细粒度预取；
- 面向 AI 画质等业务场景的细致优化，提供优于 SNPE 的推理性能。

3. 极低的CPU资源占用

- 全部基于 HVX 实现量化/反量化/重量化，不依赖 CPU；
- 精细的设备端内存池管理，最小化内存占用；
- 基于外部 ION Buffer 的零拷贝输入/输出，无需实际数据搬运。

4. 支持神经网络/前后处理端到端加速

- 自研 PPL HVX CV 图像处理库，性能优于高通 Fast CV；
- 结合 PPL HVX 推理引擎，提供前后处理 + 模型推理端到端 HVX 加速支持。

GitHub: https://github.com/openppl-public

PPL Hexagon DSP HVX 业务实践

- 为 AI 画质、智能驾驶、AR 娱乐等业务线提供低功耗、高性能推理支持
- 以智能驾驶 ADAS 产品为例，使用 PPL 后的 SenseDriver 模块 A (Driver)、B (FaceID) 的性能与资源占用均得到大幅优化

同比 SNPE-DSP HVX		
指标 / 产品	Driver	FaceID
CPU 占用	降低 79%	降低 90%
内存占用	降低 11%	降低 13%
端到端推理性能	提升 1.67 倍	提升 1.14 倍

PPL 加速器优化技术实践 —— PPL Ascend

1. 基于分层 IR 的加速器工具链生态接入与扩展

- 构建从 Graph 至 PPL IR、华为 Ascend IR 的分层 IR 系统;
- 算子库覆盖厂商公开算子库与内部扩展自定义算子库;
- 一个 PPL 算子可对应有限个厂商算子/内部自定义算子组合
- 支持多层次自定义图优化。

2. 结合离线编译与轻量运行时系统，发挥推理性能

- 离线编译阶段进行数据预处理、tiling/ 调度策略搜索、序列化等工作;
- 运行时轻量加载、高效推理。

3. 扩展支持前后处理端到端加速，并支持运行时动态特性

- 在 Ascend 设备端支持并优化前后处理操作，实现全流程端到端加速;
- 从运行时管理及算子实现层面切入，支持运行时任意尺寸输入，解决现有 Ascend 框架对动态特性支持有限的问题。

4. 建立统一高效的算子开发机制，支持业务迭代

- 基于 native 编程接口（tik），抽象封装易用、高效的计算接口，支持算子快速开发;
- 构建多种 tiling、计算模式模板，可快速实现满足片上缓冲限制与性能要求的自定义算子。

GitHub: https://github.com/openppl-public

PPL Ascend 业务实践

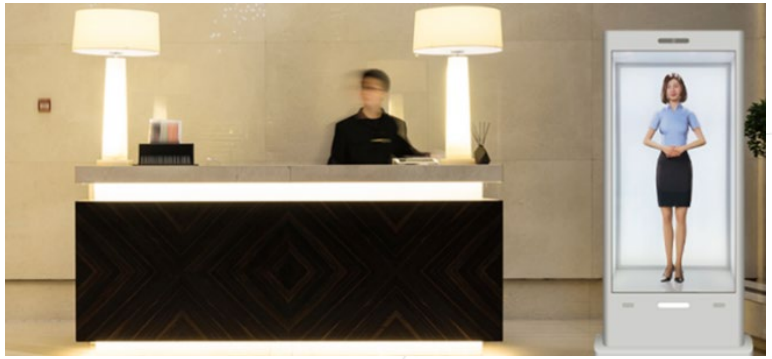
1. **满足 AI 业务的国产化替代需求，落地安防、视频大数据、智能驾驶等场景，提供模型推理、前后处理端到端加速支持；**
2. **覆盖 Ascend310（例如 Atlas200DK、Altas300、Atlas800 等）与 Ascend610（例如MDC610）等平台；**
3. **以自研可控方案，解决业务侧落地问题，例如：**

- 解除前后处理受 CPU 主机的性能影响，支持端到端加速，安防目标检测场景整体性能提升从 20% 到 200% 不等；
- 解决部分官方算子性能与精度问题，如双线性插值提升 5 倍左右；
- 支持大量小算子/长尾算子的自定义算子融合，实现模型性能的数量级提升；
- 自定义算子需求，支持数十种自定义业务算子。

PPL 支持的明星产品业务

疫情防控 - 火神系列产品

智慧文旅（AR, 数字人系列产品）

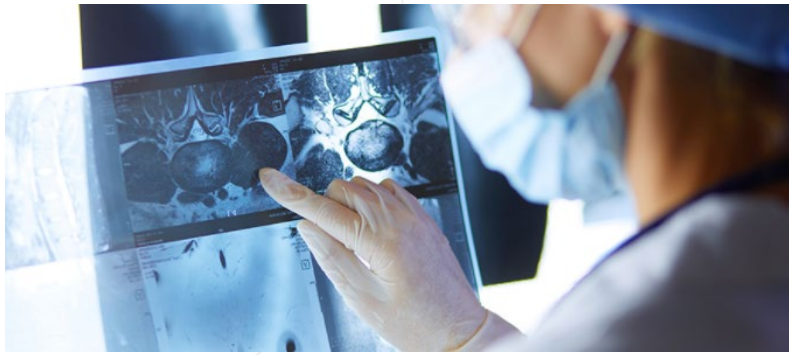


智能手机（超分，人脸解锁等）

智能车舱

夜景固化项目

智慧健康



GitHub: https://github.com/openppl-public

OpenPPL 的未来发展之路

1. 模型格式：onnx
   a. 较完善较完善的模型转换工具
   b. 支持 OpenMMLab 大多数模型
   c. 量化工具链开源
   d. 前后处理 pipeline 优化
2. 架构层面：初步支持动态图推理
3. 支持后端
   a. x86：FMA/AVX512F 指令集
   b. Nvidia GPU：Turing 架构TensorCore fp16 推理
4. 性能
   a. X86 可以超越多数市面上现有推理引擎
   b. GPU 可以在单 batch 下超越TensorRT，其他情况达到 80%-90% 的性能
5. PPL.cv：初步支持

Ver 0.1 版

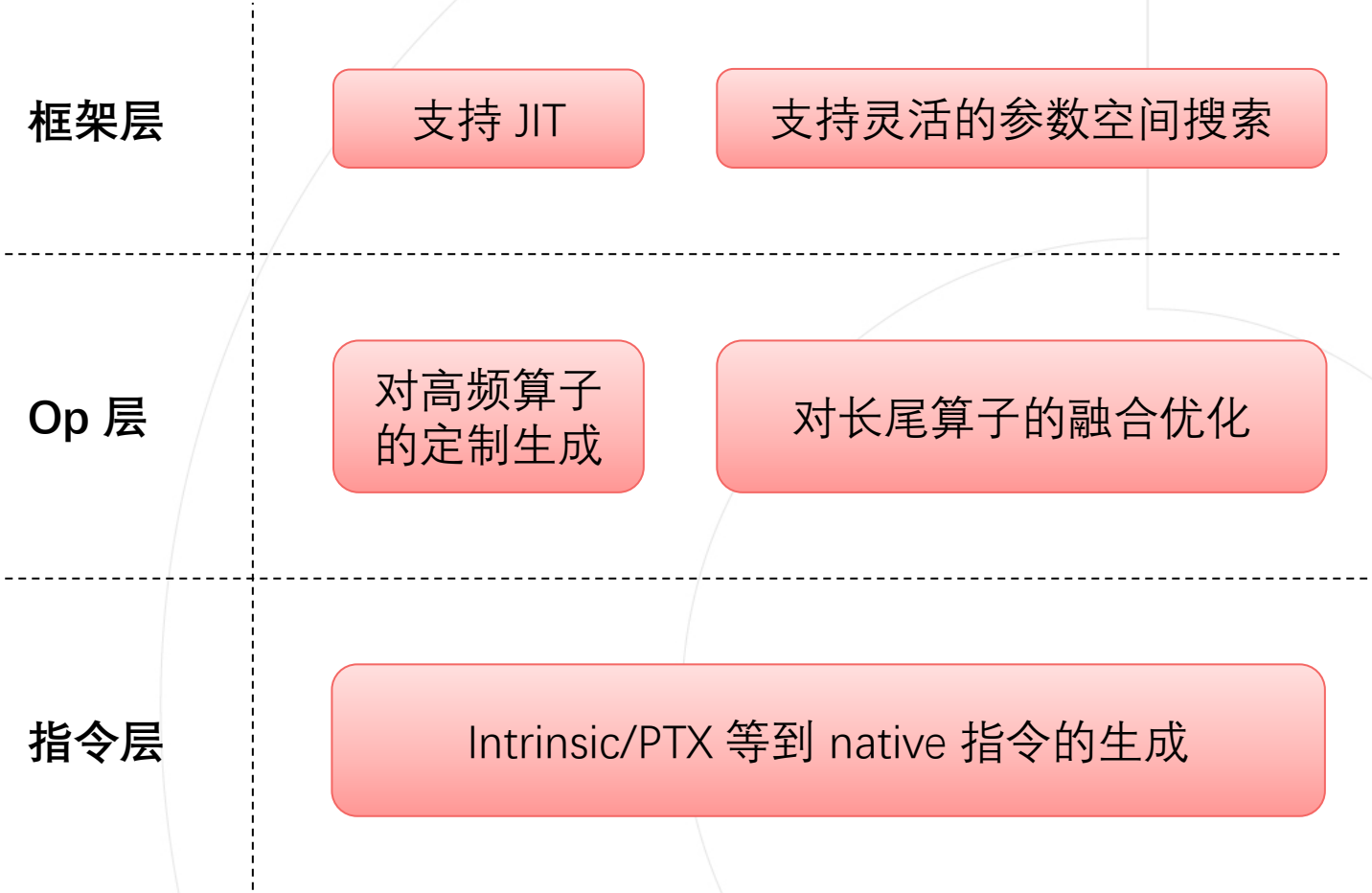
预计一年时间

1. 模型格式：onnx
   a. 较完善较完善的模型转换工具
   b. 支持 OpenMMLab 大多数模型
   c. 量化工具链开源
   d. 前后处理 pipeline 优化
2. 架构层面
   a. 支持动静分离
   b. 更多的融合策略
3. 支持后端
   a. x86：SSE/VNNI 指令集
   b. Nvidia GPU
      图灵，伏达和安培架构 & fp16/int8/int4 混合精度推理
   a. Arm server：arm v8.x 指令
   b. RISC-V：对 v 指令
4. 性能
   a. GPU 可以在多数情况超过 TensorRT，其他情况至少持平
   b. CPU 做到各自架构的 SOTA
5. PPL.cv：完备支持

Ver 1.0 版

GitHub: https://github.com/openppl-public

OpenPPL 对自动编译优化的支持



THANK YOU

QUESTIONS?

Website: https://openppl.ai/ Website OpenPPL

Discuss on Zhihu: OpenPPL Discuss on Zhihu

Thanks for listening!

Q&A Time