Cambricon

寒 武 纪 科 技

MLU100/MLU100+ FAQ

版本记录

文档名称	MLU100/MLU100+ FAQ
版本号	V2.3
创建日期	2018 年 8 月 7 日

更新历史

顺序	日期	版本号	作者	更新说明
0	2018.8.7	V1.0	程归鹏	初始版本
1	2018.8.10	V1.1	冯云凯/谈家桐	增加部分详细内容
2	2018.8.15	V1.2	张赟龙	增加视频/图片解码相关的数据
3	2018.8.16	V1.3	程归鹏	删除和重新回答部分问题，整理格式
4	2018.8.16	V1.4	宋琎/陈光/吴林阳等	部分有关算子，性能，融合等问题答复。
5	2018.8.27	V2.0	刘少礼/刘道福/郭崎/孟小甫等	共同修订
6	2018.9.7	V2.1	卞景山	更新解码性能
7	2018.10.23	V2.2	季雨娇	增加部分详细内容
8	2018.12.13	V2.3	程归鹏	修正/删除部分不准确描述

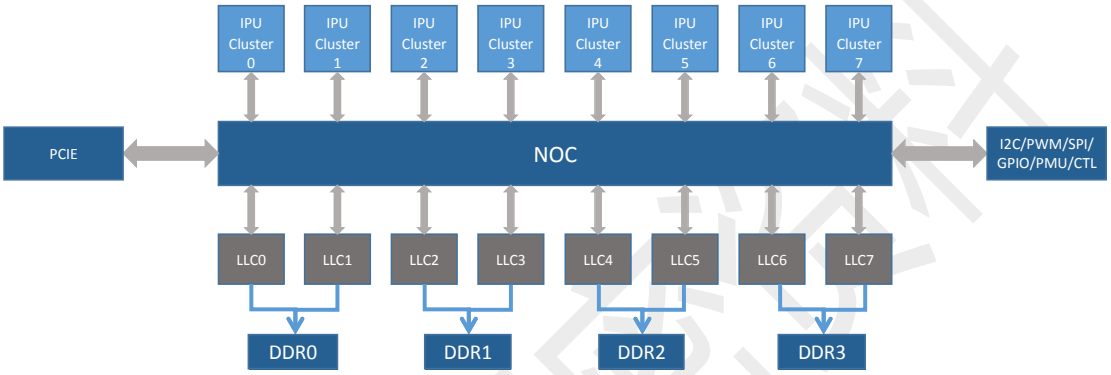
目录

1.芯片架构

1.1 MLU100/MLU100+是否支持训练和推理？

MLU100/MLU100+支持推理和训练需求，尤其是侧重推理。2019 年即将推出的 MLU200，支持训练性能更优。

1.2 MLU100/MLU100+所使用的芯片是什么架构？



MLU100/MLU100+所使用的芯片架构如上图所示，共有 32 个核心，4 个 DDR4 通道。MLU100+板卡在 MLU100 的基础上增加了图片、视频编解码芯片，并实现了 cnstream 框架，可以将解码后的图片和视频直接送给 MLU100 芯片而不需要经过 Host。

1.3 什么是 BLOCK 及 UNION2 模式？

MLU100/MLU100+集成了 32 个 1H16 高性能核心。BLOCK 为单核模式，即 Kernel 函数每次执行需要一个计算核心；UNION2 为 8 核模式，即 Kernel 函数每次执行需要八个计算核心。用户可选择不同模式进行运算。

1.4 MLU100/MLU100+内存容量及带宽分别是多少？

当前 MLU100/MLU100+分为两个版本，工程送测版和量产板。工程送测版规格为全高全长、双 slot，内存容量 32G。MLU100 量产版规格为半高半长、单 slot，内存容量有 8GB/16GB，型号分别对应 MLU100-D3 和 MLU100-D4；MLU100+量产版规格为全高全长、单 slot，内存容量有 8GB/16GB，型号分别对应 MLU100-C3 和 MLU100-C4。

所有 MLU100/MLU100+内存带宽均为 102.4GB/S。

1.5 MLU100/MLU100+支持几种运算精度？

MLU100/MLU100+支持 float16（简称，fp16）和 int8 两种运算精度。

1.6 MLU100/MLU100+ fp16/int8 和 fp32 运算精度的误差是多少？

MLU100/MLU100+ 的 fp16 运算，数值范围需要在 fp16 的表示范围之内；一般地，在常见网络级的输入（算子规模和数值范围）下，单算子的精度误差一般在 1%左右；网络级测试的精度误差一般在 1%以内。

MLU100/MLU100+ 的 fix8 运算涉及 MLP 全连接层，Conv 卷积层，LRN 局部响应归一化层。一般地，在常见网络级的输入下，单算子的精度误差在 2%左右；网络级测试的精度误差一般在 1%以内。

1.7 稀疏/int8 实际能提供多大的性能提升？

MLU100/MLU100+ 的稀疏模式只会影响 MLP 全连接层和 Conv 卷积层的性能。实际提升幅度与权值、神经元稀疏度都相关，理论上稀疏度越高的性能越好，当输入和权值稀疏度均达到 50%以上时，性能会较好。但在实际网络中，当稀疏度达到某些阈值后，该稀疏网络层的性能瓶颈不再是计算量，而是 IO 的开销（例如 1x1 的卷积层）。一般地，对于计算量较大的层，稀疏度较高的条件下，性能提升较为明显，能达到 40%或更高。

MLU100/MLU100+ 的 fix8 模式的算子目前只涉及 MLP 全连接层，Conv/Deconv 层，LRN 局部响应归一化层。fix8 模式相比 fp16 模式，MLP 和 Conv 计算性能增加一倍。同样地，在计算量不大的网络层，fix8 模式下性能瓶颈不再是计算量，而是 IO 的开销。

1.8 MLU 里面有多少个核？

32 核。

1.9 MLU 架构类似 TPU 吗？

架构与 TPU 不同，但有些基础的思想是类似的，比如更高算力密度，以及通用性。

1.10 怎么看 DSP？MLU 优势？

DSP 的能效比不好，相比 MLU100 需要占用大量指令开销，不是为深度学习设计的。MLU 的计算效率更高。

1.11 MLU 是专用深度学习架构 ASIC 芯片？

是

1.12 寒武纪软件平台的训练和推理架构一样吗？

软件训练与推理框架一致，都会支持主流深度学习框架。

1.13 怎么看 FPGA？MLU 优势？FPGA 是可编程，这不是相对 MLU 的优势吗？

FPGA 优势：开发时间短，一次性开发成本低（不需流片费用）

劣势：能效比相对于 MLU 低很多，性能不如 MLU，量产后成本大大高于 MLU。

FPGA 的可编程是指可以重新烧制硬件。MLU 指的是可编程的指令集。

1.14 全长全高的 C3 与 NV P4 相比，单台服务器内部的 AI 加速卡密度大幅降低，竞争力减弱吗？

我们的解码能力尤其 jpeg 解码能力优于 P4，2019 年会有半高半长带解码能力的板卡问世。

1.15 稀疏化具体是怎么实现的？ 软硬件分别做了什么工作？开稀疏化与不开稀疏化的四倍差距是如何计算出来的？

稀疏化就是在保证计算精度的前提下，筛选出计算中的非零的权值和神经元进行计算，用来提高有效数据的计算效率。因为目前最大提取非零单元比例为 3/4，所以最高只能有 4 倍提升。

1.16 寒武纪板卡的视频/图片解码可以支持哪些格式？

寒武纪板卡是硬件实现的视频/图片编解码。视频解码格式包括：H264，H265，MJPEG，MPEG4；图片解码格式包括 JPEG。

1.17 寒武纪板卡的存储是片内还是片外的，如果是片内，那寒武纪的高效复用存储模式是怎么实现的？

寒武纪板卡存储既有片内又有片外。片内的通过相邻算子间的片内存储复用减少片外缓存的访问来达到提高计算速度；片外通过层间地址复用节省内存占用，以支持更大的任务。

1.18 MLU200 的硬件架构与 100 相比在哪些方面做了改进？ 是如何做到更适合训练的？

"MLU200 的改进：1. 更灵活的指令集；2. 更灵活的位宽；3. 大幅提升能效比；4. 降低带宽需求；5. 更强的视频编解码能力；6. 支持训练；

如何支持训练：支持 FP32 运算"。

2. 性能指标

2.1 各网络性能指标与 p4 的对比

详见各 release 版本的《MLU100 performance 测试数据表》。

2.2 解码性能指标(分辨率，速率，延时)

H264/H265 解码数据：

输入	输出	通道	单通道速率	解码延时（ms）
H264 1920x1080	1920x1080 YUV420SP	16	30	160
		20	24	192
		32	15	285
	1080x1080 YUV420SP	28	30	160
		36	24	194
H264 1280x720	1280x720 YUV420SP	36	30	153
		46	24	186
		64	15	267
	720x720 YUV420SP	60	30	148
		64	24	169
H265 1920X1080	1920x1080 YUV420SP	16	30	130
		20	24	182
		32	15	244
	1080x1080 YUV420SP	28	30	191
		34	24	162
H265 1280x720	1280x720 YUV420SP	36	30	180
		44	24	149
		64	15	241
	720x720 YUV420SP	56	30	143
		64	24	163

JPEG 解码数据：

输入	输出	通道	总张数	单通道张数	解码延时(ms)
Jpeg 1920x1080	1920x1080 YUV420SP	32	384	12	68
	1080x1080 YUV420SP	32	384	12	67
Jpeg 1280x720	1280x720 YUV420SP	32	768	24	41
	720x720 YUV420SP	32	768	24	40
Jpeg 256x256	256x256 YUV420SP	64	2112	33	22

2.3 矩阵乘性能及规模(MAC 利用率,模型并行后的性能提升)

matri_mult 规模			性能（时间计量） mluLaunchKernel Time per core				加速比		
m	k	n	fp16 (us)	50%filter 稀疏 (us)	50%input +50%filter 稀疏 (us)	p4 fp16 (us)	p4_fp16 / MLU100 dense (%)	p4_fp16 / MLU100 filter_50% (%)	p4_fp16 / MLU100 filter_50% +input_50% (%)
32	832	2048	21.091	15.380	15.206	90.698	430.030%	589.723%	596.476%
32	2048	2048	48.784	34.849	34.788	143.261	293.665%	411.086%	411.813%
32	2048	9024	214.593	157.144	157.046	394.834	183.992%	251.256%	251.413%
64	832	2048	29.306	22.264	22.245	111.687	381.113%	501.639%	502.081%
64	2048	2048	61.868	48.157	48.196	189.346	306.049%	393.184%	392.871%
64	2048	9024	271.830	216.820	216.785	587.253	216.037%	270.848%	270.892%
128	832	2048	44.853	35.533	35.499	165.382	368.720%	465.432%	465.884%
128	2048	2048	87.853	74.448	74.443	318.319	362.330%	427.575%	427.603%
128	2048	9024	387.246	333.478	333.467	1094.950	282.753%	328.342%	328.353%
256	832	2048	132.837	129.407	129.408	282.325	212.535%	218.168%	218.167%
256	2048	2048	298.741	289.259	289.244	561.141	187.835%	193.993%	194.003%
256	2048	9024	1323.310	1285.220	1285.180	2146.080	162.175%	166.982%	166.987%
512	832	2048	258.489	254.667	254.654	488.751	189.080%	191.918%	191.927%
512	2048	2048	580.192	569.614	569.642	1087.660	187.466%	190.947%	190.937%
512	2048	9024	2572.250	2573.660	2575.231	4021.350	156.336%	156.250%	156.155%

说明：mluLauchkernel Time per core 是实际运行时，mluLauchKernelTime 平均到每个 core（MLU100 一共有 32 个）上的大小。

2.4 Batch size 实际含义

单核一次性处理多少张图片。我们测试的表格里面所指的 batch32 是将 32 张图片均分到 32 个核上，相当于 batch size 是 1，因此在网络的 nchw 里体现的 n 的值是 1。

2.5 我司解码采用的是什么芯片，为什么实际解码性能不佳？

MLU100 解码采用了单独的芯片。针对 1080p 输入，当前解码速率可以达到 32 路，每路 16fps，或 20 路 25fps。

2.6 提供参考测试报告 benchmark report（典型网络下）

FP16 是 P4 两倍，Int8 和 P4 相当。具体细节详见《MLU100 performance 测试报告》

2.7 提供 MLU100 Resnet50 Fix8 dense/sparse 优化后的性能数据以及如何提升精度？

当前版本 ResNet50 fix8 dense 性能为 1177.86fps（32batch）；ResNet50 fix8 sparse 性能为 1396.17fps（32batch）。

2.8 离线模式对哪些模型的支持度最好？ 在线模式性能如何、相比离线模式性能下降多少？

从用户友好程度来说，离线模式对于端到端模型（即所有操作都可以在 MLU 上运行）支持最好，用户不需要编写额外的 CPU 代码。在线模式性能相比离线模式性能的差距与框架本身开销有关。当前 Caffe/MxNet 在线与离线差距在 10%左右，TensorFlow 框架在线与离线性能差距较大（数量级差距），研发正在优化中。

2.9 高性能模式是传统意义上的超频吗？

高性能模式并不是简单的超频。采用高性能模式后 MLU100 最高频率变高，同时会根据实际功耗动态调整频率。但不推荐使用。

3. 编程相关

3.1 MLU100/MLU100+支持哪几种深度学习框架？

MLU100/MLU100+目前支持 Caffe，Caffe2，TensorFlow，MXNet，ONNX 等主流的深度学习框架。

3.2 MLU100/MLU100+支持的网络种类有多少？

MLU100/MLU100+支持主流深度学习神经网络的加速，如

分类网络：AlexNet/VGG/Inception 系列/ResNet 系列

检测网络：yolov1/v2/v3，SSD，Faster RCNN 等

循环神经网络：RNN/LSTM 等

3.3 MLU100/MLU100+是否支持稀疏化和 int8 运算？

MLU100/MLU100+支持稀疏化和 int8 运算，部分网络在进行稀疏化和 int8 处理的时候需要使用我们提供的工具进行重新训练，用以提高网络精度。

3.4 MLU100/MLU100+支持哪些算子？

详见 Caffe/TensorFlow/CNML 算子列表，其中 CNML 是机器学习库，全称是 Cambricon Neuware Meachine Learning。

3.5 MLU100/MLU100+的在线和离线模式是什么含义？

在线模式是基于深度学习框架的 API 接口进行“前向”（Forward）计算的模式。

离线模式就是利用寒武纪提供的工具把用户训好的模型，编译生成寒武纪格式的模型，运行时直接调用运行时接口 CNRT（全称是 Cambricon Neuware Run Time），脱离原生框架运行。离线模型里减少了指令生成的时间以及框架开销，对于训练好的固定模型能大幅度提高运行性能。

3.6 离线模型中 batch size 如何修改？

离线模型中的 batch size 是固定的。修改 batch size 需要通过修改原生模型中的 batch size，并且需要重新生成离线模型。

3.7 什么是数据或者模型并行编程模式？

数据并行：多个核同时处理相同模型，或者多个核处理不同模型，但每个核处理不同的输入数据。

模型并行：同一个模型或者输入可以分拆到不同核上执行。

数据并行主要面向高吞吐的场景，模型并行主要面向低延迟的场景，两者可以结合使用。

3.8 如何理解 Stream 的概念？

Stream 是串行执行的任务流。同一个 stream 内的任务串行执行，不同 stream 之间任务可以并行执行。通过创建多个 stream 并把可并行的任务放在不同的 stream 中可以实现多个任务的并行执行。

3.9 自定义算子如何实现

在 cnml 中添加自定义算子的流程如下：首先需要将算子的计算流程进行拆分，然后用 cnml 中已提供的基础算子拼接出自定义算子的整个计算流程，进而实现在 MLU 上可以运行的自定义算子。我们后续会提供高级语言用于添加用户自定义算子。

3.10 内存复用及优化思路

在线模式下，模型数据（权值）的复用通过共享 cnmlBaseOp_t / cnmlFusionOp_t 数据结构来实现；

离线模式下，模型的复用通过共享 cnrtFunction_t 数据结构来实现。

输入输出的复用由用户来实现。cnml 提供获取 cnmlTensor 大小的接口以及内存分配的接口，用户通过分析数据依赖关系，然后通过调用这些接口给复用的多个数据块分配同一个 MLU 地址，并共享这块地址来实现内存复用。

3.11 与 FPGA 相比，MLU100 可编程性如何，如果出现一些新网络模型，如何工作？

对于新出现的网络与模型，如果其中算子我们的高性能库均支持，可直接编程实现。如果出现不支持算子，有两种方式：1）使用我们的高性能库提供的基本算子拼接而成；2）2019/Q1 我们提供的编程语言直接实现相应算子。

3.12 寒武纪的 mlisa 语言和 bang 语言目前的成熟度如何，能否正常实现所有需要的算子？

MLISA 和 Bang 语言已经 release 给部分客户，功能上可以实现任意算子。

3.13 寒武纪支持的算子列表？支持程度？

我们自己有一套支持算子列表。详见各编程框架及算子手册。

3.14 寒武纪板卡对语音及自然语言处理的支持如何？

寒武纪的板卡有很强的灵活性，可以很好的支持语音和自然语言处理任务。

4. 工具使用

4.1 MLU100/MLU100+是否有监控工具？

cnmon 可以监控设备的实时功耗、温度、ECC Error 统计、内存/MLU 的利用率等信息。

4.3 MLU100/MLU100+是否提供性能分析工具

CNPERF 是一款针对用户层程序的性能分析工具，可用于分析 CPU 和 MLU 的性能。CNPERF 提供了以下功能：

- 精确获得用户程序及部分动态链接库中函数的执行时间；
- 获得函数调用栈信息；
- 获得 CNML 库中 MLU 算子的运算性能和访存性能；
- 获得 CPU 与 MLU 之间数据拷贝的数量以及速度；
- 获得 CPU\MLU 使用内存量（malloc、free）。

4.4 Docker 使用

目前 MLU100/MLU100+采用 docker 的方式来进行交付，保证环境一致。同时我们也即将推出针对 MLU100/MLU100+的 Cambricon-dokcer 的安装包，方便用户进行大规模的云端部署。

4.5 P4 训练后如何在 MLU 上推理？

我们提供模型转换工具支持将主流模型进行转换。

4.6 边缘计算也带着软件平台框架跑吗？

边缘计算可以依托于我们提供的离线模型运行，脱离框架减少整体开销。

4.7 为什么不采用和 P4 相同的内存颗粒而使用 DDR4？

P4 为 GDDR5，MLU100 为 DDR4 不同的协议，无法使用相同的颗粒。

5. 其他

5.1 MLU100/MLU100+对系统配置有什么要求（如 cpu，内存等）？

建议使用 Intel i7 处理器或者 Intel 服务器处理器，或者同等性能的 AMD 处理器，内存容量大于 16GB。

5.2 MLU100/MLU100+支持的 OS 及内核？

Ubuntu 16.04	Linux 内核版本 4.4 及以后均可
Debian 9.x	Linux 内核版本 4.9.x
CentOS 7.x	Linux 内核版本 3.10.x

目前 MLU100/MLU100+已经对客户开放驱动代码，可以根据系统及内核进行适配。

5.3 MLU100/MLU100+是否支持 Windows？

Windows 版本驱动、CNML、CNRT 等正在开发中。

5.4 MLU100/MLU100+的产品形态是什么？

MLU100/MLU100+是一款面向云端（服务端）的智能加速卡，暂无终端或嵌入式的形态。

5.5 类似于 TX1 的边缘端模块或芯片，寒武纪是否有 Roadmap？

2019 年下半年会有低功耗边缘端的产品上市。

5.6 寒武纪没有 emulator，客户只想用软件，也不想上机/上板子，怎么办？

未来会考虑提供云端板卡，debug 云，以及 x86 上纯软件 cambricon 适配后的框架三种 solution。

5.7 寒武纪指令集不开放、寄存器不开放，何时开放更多？

暂时还没有开放计划。

5.8 寒武纪编译器何时开放？

寒武纪编译器预计明年 Q1 开放。

5.9 寒武纪产品线未来两年的 Roadmap（官方）

保持每年推出一代新的芯片和 IP 的节奏。2019 年会推出训练产品。

5.10 提供半高半长板卡做测试（C3/D3 到货时间）

C3 11 月底会有板卡提供客户小批量测试，D3 在 12 月中旬会提供客户小批量测试，2019/1 中旬会正式量产。

5.11 与 MLU100 成功适配过的服务器整机型号

已在联想、曙光、浪潮的一些服务器上进行了适配。

5.12 原厂直供 or 代理商？支不支持账期？交期一般多久？

目前量大采用原厂直供，未来会考虑引入代理商。如果是服务器产品可以直接通过我们的服务器合作伙伴购买。小量 2~3 周，大量供货需要 3 个月。

5.13 在相关金融项目会大量用到 Pagerank，目前我们是否有相关 schedule 去完善支持？

暂时没有支持计划。

5.14 单 slot 测试卡（C3E/D3E）到卡时间以及正式量产时间

C、D 系列在 2019/1/15 正式量产。

5.15 MLU100 双 slot 和未来批量单 slot 目前软件版本（驱动）是否还在更新中？如果还在更新，最终定型版本时间？

目前仍然在按版本更新，已经可以 release。11 月底 RC 版，1 月初公测版。

5.16 目前寒武纪板卡的功耗并不比 P4 低，有没有可能做到功耗的大幅度降低？何时？

2019 年下半年会有低功耗边缘端的产品上市。

5.17 适配某些版本 Linux 出现 soft_lockup 情况如何解决？

这个问题与 DDR 训练有关，在 1 月份量产版中会解决。