DOI:10. 20079/j. issn. 1001-893x. 221208003

一种 FPGA 集群轻量级深度学习计算架构设计及实现*

刘红伟[1,2],潘 灵[1,2],吴明钦[1,2],韩毅辉[1,2],侯 云[1,2],席国江[1,2]

(1. 敏捷智能计算四川省重点实验室,成都 610036;2. 中国西南电子技术研究所,成都 610036)

摘 要:传感器技术的发展带来了边缘、端设备功能的迅速迭代升级,也带来了战场前端的数据量成倍增长。针对边缘、端设备数据量的急剧增长和芯片计算处理能力的矛盾,结合 Map/Reduce 框架,提出了一种基于现场可编程门阵列(Field Programmable Gate Array,FPGA)计算集群资源的深度学习架构,能够实现多个深度学习算法的并行快捷部署和应用。该轻量级深度学习计算架构同时满足军事应用对“端”的智能处理能力提出的新要求,即不仅局限于数据采集和智能的应用,还必须具备分布式并行智能实时计算的能力。该 FPGA 集群轻量级深度学习计算框架部署不同类型算法容易,实时性高(ms 级任务响应),可扩展性好,在多种类异构传感器、大场景大数据吞吐量的军事场景及森林防火等民用场景有广泛的应用前景。

关键词:深度学习;边缘计算;端设备;海量数据;实时处理

开放科学(资源服务)标识码(OSID):

中图分类号:TN971;TP18 **文献标志码**:A **文章编号**:1001-893X(2024)01-0014-08

Design and Implementation of Lightweight Deep Learning Computing Architecture for FPGA Cluster

LIU Hongwei[1,2], PAN Ling[1,2], WU Mingqin[1,2], HAN Yihui[1,2], HOU Yun[1,2], XI Guojiang[1,2]

(1. Sichuan Key Laboratory of Agile Intelligent Computing, Chengdu 610036, China;
2. Southwest China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: With the development of sensor technology, the functions of edge or terminal equipment are rapidly upgraded, and the data quantity of front-end battlefield increases exponentially. According to the contradiction between the inability of chips and the sharp growth of data volume on edge and terminal equipment, combined with the Map/Reduce framework, a deep learning architecture based on field programmable gate array (FPGA) computing cluster resources is proposed, which can deploy multiple applications with deep learning algorithms and can be widely used in military scenes and civilian scenes such as forest fire prevention.

Key words: deep learning; edge computing; terminal equipment; massive data; real-time processing

0 引 言

随着信息技术在军事领域的广泛应用,战争形态发生重大变化,世界范围内的新军事变革日益深化。战场信息已经成为获取战场主动权的关键[1]。在军事领域,战场态势瞬息万变,最新、最重要的态势信息往往来源于最靠近战场的“端”(例如无人机平台等),作战要求“端”必须具备对战场的快速反应能力[2]。而战场中,作战平台与云端(即脑)的网络连接情况非常不确定,甚至存在无法联网的情况,因此一大部分的数据处理、决策智能需要在“湾”甚

* **收稿日期**:2022-12-08;**修回日期**:2023-03-13
基金项目:四川省重点研发计划项目(2022YFG0231);四川省自然科学基金项目(2023NSFSC0497)
通信作者:刘红伟 Emai:weimeng01234@126. com

至“端”处做出,不得不弱化对云端智能的依赖性。另外,随着传感器技术的发展,战场前端的数据量成倍增长,例如,雷达和传感器领域的技术发展对于保持先进的情报、监视和侦察任务能力具有至关重要的作用,同时也带来了海量数据实时处理的需求[3-6]。美国空军研究实验室信息研究局利用两个合成孔径雷达系统实现 20 km 广域持久监视能力,每 2 s 需要生成 3 140 亿像素的图像[7]。军事应用对“端”的智能处理能力要求较高,不仅局限于数据采集和智能的应用,还必须具备分布式并行智能计算的能力。这对小型化、轻量级、可分布式计算的智能作战平台提出了迫切需求。

当前云计算、大数据、智能化处理其基础都是依赖于高性能的大规模并行计算技术。云计算实际上也是一种并行计算技术,而大数据处理是将大的数据段分割成若干小的数据段进行并行计算处理。模式识别、神经网络等智能化计算在很大程度上也是依赖于并行计算模式,其每个神经元就是一个计算节点。

在并行计算框架方面,先后经历了 MPI、Pthread、OpenMap、OpenCL, MapReduce。Google 推出的 MapReduce[8]给大数据并行处理带来了巨大的革命性影响,使其成为事实上的大数据处理的工业标准。作为一种面向大规模数据处理的并行计算模型、框架、平台和方法,MapReduce 将一堆杂乱无章的数据按照某种特征归纳,处理并得到最后的结果,已广泛应用于大规模的算法图形处理、文字处理、数据挖掘、机器学习、统计机器翻译等领域和 Google 内部需要大规模并行计算的应用程序。

如今,来自云、移动和物联网设备的海量流数据正在不断高速地产生,而云计算模型已经不能高效及时地处理这些数据,满足不了许多物联网应用的实时性需求,因此近年来学术界正在研究基于边缘计算模式的流数据的处理。被 CPU 统一调度的多片 FPGA 作为计算单元的集群,简称 FPGA 集群,可以实现资源实时在线重配置、内存外扩、端到端高速通信模式,而且能够灵活部署深度学习算法,因此更加适合需要智能算法处理时延低、吞吐量大的场景,也适用于对实时性要求高、信号计算更为复杂的产生大量数据的边缘、端处理领域[9]。华中科技大学的胡蝶[10]对 FPGA 在加速分布式流处理系统方面进行了深入研究[10],实现了基于 Storm[11-12]框架的异构环境下的加速器优先的 CPU-FPGA 混合调度策略,但业务数据通过 CPU 外挂的 PCIE 接口进行数据注入,FPGA 仅仅作为 CPU 的加速器进行工作,两者耦合性太大,受限于 CPU 外挂的 PCIE 控制器个数,不易大规模部署多个 FPGA 节点,扩展性不好。

本文基于 FPGA 计算集群资源实现资源管理、MapReduce 框架下应用的部署,能够针对实时计算的边缘、端场景实现数据的采集、数据流的流式管理和人工智能算法的快捷部署。本文主要创新是系统管理拓扑和业务数据流的解耦,数据流直接通过 FPGA 外挂的高速总线(SRIO 或以太网)进行业务数据的注入和结果的输出,CPU 仅实现 FPGA 执行器的节点管理(如可执行文件的注入加载,业务网络端口通信管理等),因此更加轻量级,对 CPU 的性能依赖更小,系统中仅一个双核 ARM 处理器(部署 Linux 操作系统)和 2 GB 内存即可实现对 4 片 Xilinx V7 系列 FPGA 的控制。同时,业务数据流和系统管理拓扑解耦,更便于计算节点的扩展。

1　FPGA 集群轻量级深度学习计算架构设计

1.1　总体架构设计

本文提出了一种基于 FPGA 计算集群资源的深度学习架构,结合 Map/Reduce 框架实现深度学习算法的快捷部署和应用。

系统从任务实现方面划分为网页前端、资源池和外部系统,如图 1 所示。网页前端进行分布式任务编排、资源监控和应用部署;资源池为 FPGA 节点组成的计算集群阵列和系统控制节点,以及算法库与分布式应用库。从总线层面划分,系统分为管理调度总线和实时总线,管理调度总线负责资源的分配、查询、任务的接收和部署,实时总线负责资源池中 FPGA 节点之间的实时高速数据传输。

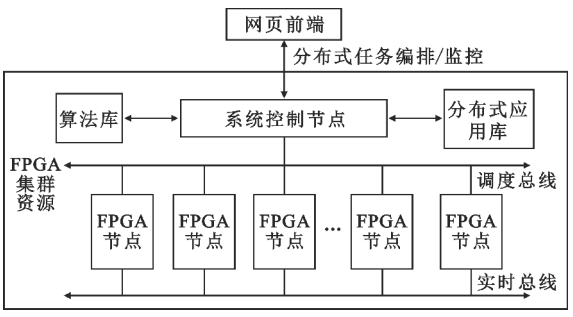

图 1　系统总体架构

Fig. 1 System architecture

根据图计算的理论和方法,在总体方案设计上,可以将系统分为若干软件配置项,各自完成并行计算中与之相关的任务。总的来说,软件配置项主要分为调度类的配置项与计算类的配置项,如图 2 所示。图中没有体现出网管的配置项,因为网络管理与通信软件进行了解耦,只需进行节点间路径的配置,而不负责逻辑通道的配置,相当于实现了一个实时数据传输的网络交换机,Worker 之间通信的建立都是 Worker 调用通信软件的接口,根据下发的参数主动与其他 Worker 建立关系。

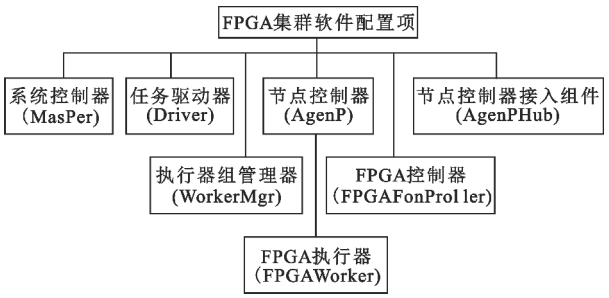

图 2 系统软件配置项

Fig. 2 System software configuration items

1.2 基本工作流程

深度学习计算应用开发和提交任务运行的基本工作流程如图 3 所示。

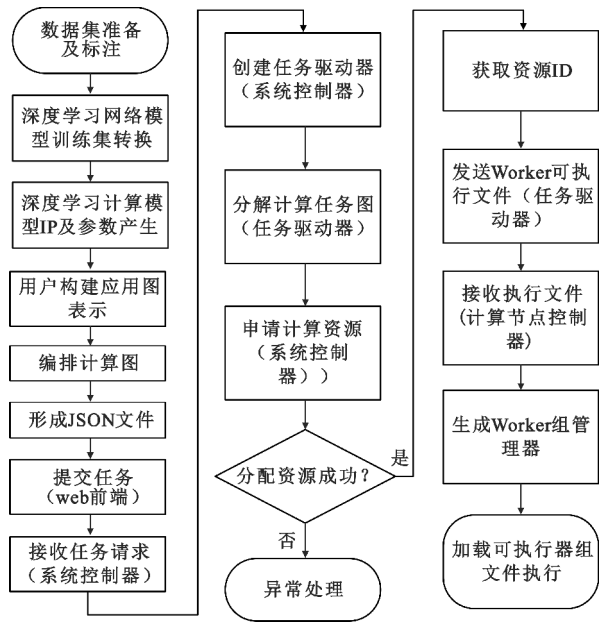

图 3 战术级深度学习并行计算平台基本工作原理

Fig. 3 Basic principle of a lightweight deep learning parallel computing platform

用户根据应用场景,经过数据集准备及标注、深度学习网络模型训练及转换、深度学习计算模型 IP 在 FPGA 上的实现产生模型及参数,之后在战术级深度学习并行计算平台上将产生的深度学习计算模型及参数完成部署,部署过程中按照 Map-Reduce 编程模型构建计算图(用户可编写 Map( ) 和 Reduce( ) 函数或指定系统提供的 Map( ) 和 Reduce( ) 函数),编排计算图形成 JSON 文件,并存放在用户程序管理端。

在任务需要运行时,通过用户界面(如浏览器网页)向系统控制器提交任务请求。系统控制器接收到新任务请求,即创建一个任务驱动器进程,任务驱动器对任务的计算图进行分解(为执行器组),形成对计算和通信资源的需求,并向系统控制器提起资源请求,系统控制器根据当前资源状况进行资源分配并返回分配到的资源 ID 给任务驱动器。任务驱动器获得资源 ID 后,即通知资源组所在的节点控制器(资源可能分配至多个节点,即对应多个节点控制器),并发送资源部署信息和执行器(Worker)文件。节点控制器根据资源部署信息创建执行器组管理器,执行器组管理器对管辖的组内执行器(Worker)进行参数注入。上述工作完成以后,即任务部署和资源调度已完成,等待任务启动。

1.3 任务提交工作流程

任务通过管理界面提交后,系统控制器(Master)会创建一个任务驱动器(Driver)子进程,通过进程的标准输入管道,将解压后的并行计算相关文件目录传递给 Driver。

Driver 启动后,将计算图分解为多个执行器组,每个执行器组里包含了一个或多个 FPGA 执行器(Worker),Driver 将以执行器组为单位,向 Master 申请计算资源。值得注意的是,每个执行器组作为一个可以调度运行的分组,必须调度到一个计算节点上运行,因此,对一个执行器申请资源时,Master 将为该执行器组都分配唯一的一个计算资源,返回该计算资源的位置信息。

Driver 在申请成功所有执行器组的计算资源后,并发地向每个执行器组对应的节点管理器(Agent)提交执行请求。接下来,节点管理器(Agent)为一个执行器组创建一个执行器组管理器,执行器组管理负责了该执行器组整个生命周期的运行管理,它通过 AgentHub 继承下来的通信连接与 FPGA 控制器(FPGAController)进行通信,传递控制信息。

1.4　任务销毁流程

在运行的任务结束,或需要主动停止正在运行的任务时,可通过网页操作接口,选中指定的任务,点击取消按钮,就触发了任务的销毁流程。销毁过程中,任务驱动器(Driver)根据当前执行情况,获取本任务内所有执行器(Worker)的信息,并分别向节点管理器(Agent)发送停止对应执行器(Worker)的指令,节点执行器(Agent)找到对应的执行器组管理器(WorkerMgr)并向其发送停止指令;执行器组管理器(WorkerMgr)停止其管辖的所有 FPGA 执行器(Worker)后退出并释放资源,节点管理器(Agent)更新资源使用状况,并向系统控制器(Master)上报资源状态。

2　FPGA 集群深度学习计算架构实现

该深度学习计算框架主要包括通用调度域和实时计算域两个维度,通用调度域包含系统控制器(Master)、任务驱动器(Driver)、节点控制器(Agent)、执行器组管理器(WorkerMgr)几个功能模块,实时计算域中是多个 FPGA 执行器(Worker),承担实质的实时计算任务。计算架构内部逻辑关系图如图 4 所示。

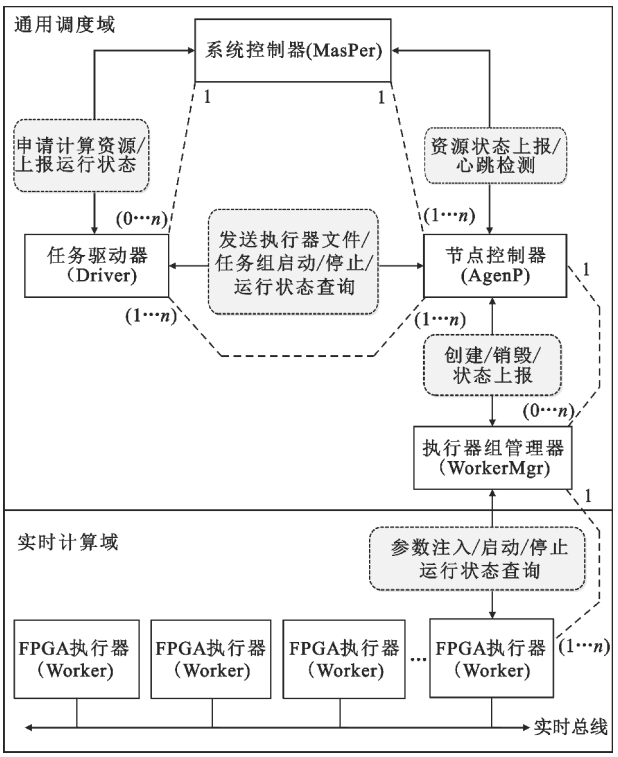

图 4　计算架构内部逻辑关系

Fig. 4 Internal logical relationship of the computing architecture

系统控制器复杂系统初始化时启动,负责系统总的并行计算资源管理、任务提交、任务监控等功能,提供了与浏览器接口的功能。

任务驱动器是并行计算框架实时计算域任务部署、管理、执行过程监控的核心部件,起承上启下作用,系统控制器下发的每个任务对应一个任务驱动器,系统中可同时存在多个任务驱动器。每个任务驱动器在新任务下发后由系统控制器产生,负责任务部署和任务执行整个生命周期的管理,在任务结束后销毁。任务驱动器部署在系统控制器物理节点或独立物理节点上。

节点控制器处在通用调度域中主节点和从节点之间,与主节点的系统控制器和任务驱动器均有信息的交互,同时控制着执行器组管理器的运行。节点控制器上行与系统控制器和任务驱动器信息交互,下行与执行器组管理器交互。

执行器组管理器负责一个任务组中各个 FPGA 执行器的生命周期管理,具备应用程序加载、参数注入、启动或停止算法运行以及执行器运行状态查询功能。

2.1　架构硬件平台实现

该 FPGA 集群平台硬件实物为一个 6U 机箱,该机箱由 1 个系统管控单元和 6 个嵌入式深度学习计算单元组成。为保障轻量级深度学习计算框架软件运行,系统管控单元对处理器的选择需要满足 CPU 具有千兆网协议控制器,处理器内核频率不小于 1 GHz,内存容量不低于 4 GB,内存访问数据速率不小于 12.5 GB/s 等指标;为保障 FPGA 具备充足的资源来进行深度学习算法模型部署,FPGA 至少选用 Xilinx 的 V7 系列芯片作为 FPGA 大规模逻辑处理单元。随着 FPGA 芯片的工艺提升,可以选用更大逻辑资源的 FPGA 系列。

2.2　FPGA 执行器软件实现

FPGA 执行器作为深度学习算法驻留和实际运行者,深度学习算法模型在 FPGA 执行器上实现过程如图 5 所示。

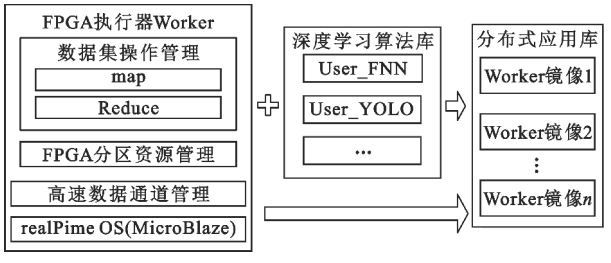

图 5　深度学习算法模型在 FPGA 执行器上实现软件架构

Fig. 5 Software structure of the implement for deep learning models deployed on the FPGA workers

FPGA 执行器软件通过实时操作系统 MicroBlaze 进行高速数据通道管理、FPGA 分区资源管理以及数据集操作管理。FPGA 执行器根据数据集管理、数据集操作需求产生基础接口源码,并预留深度学习算法模型嵌入函数。深度学习算法库按照 FPGA 执行器需要的接口形式添加深度学习算法模型源码,之后进行镜像生产,生产过程分为综合部分和布线部分,综合部分生成网表和资源需求信息文件,根据资源需求文件对已划分好的分区进行自动合并,并生成不同分区位置的镜像文件。生成的 Worker 镜像放置在分布式应用库中,可以被 web 前端调用。

2.3 深度学习算法针对 FPGA 的 IP 实现

执行 CNN 运算的 FPGA IP 是基于 RTL 代码完成的,该 IP 的设计思想是以卷积层作为核心来构成整个框架的,因此解析 CNN 模型也按照该思路来完成。

以 Caffe 深度学习计算框架下 CNN 深度学习算法来描述 FPGA 实现过程。Caffe 的网络模型由两部分组成:一个是 *. prototxt 文件,描述了网络模型的结构信息;另一个是 *. caffemodel,存储了网络中每层计算需要的参数。在 Caffe 中,一个完整的 CNN 模型就是一个 Caffe Net,是由不同的 Layers 组成的有向无环图。一个典型的 Net 从 data Layer 开始输入数据(在 CNN 中一般为图片数据),经过各个 Layer 的处理,最后在 Loss Layer(或者其他 Layer 并附加某些处理算法)完成计算目标任务。Net 是由一些 Layers 和它们之间的相互连接构成的,并且用文本建模语言来描述这种结构。

Layer 是 Caffe 模型的本质内容和执行计算的基本单元,可以进行多种运算,比如卷积、全连接、池化、激活函数、归一化、缩放、softmax 等。在 Caffe 的 Layer catalogue 层目录中可以查看所有的操作,也可以查看到绝大部分目前最前沿的深度学习任务的 Layer 类型。例如,一个典型 Layer 通过 bottom 连接层接收数据,通过 top 连接层输出数据。

整个实现软件按照语言被分成了两大块,主模块是作为软件的入口完成所有的调度,其中 Config 解析模块负责解析 *. prototxt 文件,Params 解析模块负责解析 *. caffemodel 文件,如图 6 所示。在导入 *. prototxt 和 *. caffemodel 之后,首先基于 protobuf 的 API 和 numpy 对 CNN 模型文件按照 Caffe 的设计结构完成层层的分解,然后调用 C 库做进一步的处理,计算和整理成 FPGA 可以识别的数据结构,最后保存文件,形成算法库,之后可以根据应用加载给 FPGA 来使用。

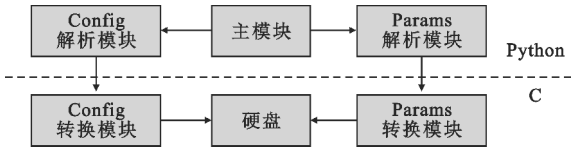

图 6 CNN 模型解析软件的结构框图

Fig. 6 Diagram of CNN model analysis software

2.4 深度学习应用在 FPGA 执行器中实现

FPGA 人工神经网络加速系统支持采用各种开源开发环境建立的神经网络模型,还支持用户自定义的神经网络模型。当神经网络模型的算法更新时,只需要重新配置神经网络模型的参数,而无需更改硬件设计。以 CNN 神经网络为例,基于 FPGA 实现加速目标识别包括以下步骤:

1)神经网络模型及模型参数加载;

2)输入待识别的数据(图像),并进行预处理,以适配模型的输入标准;

3)根据配置参数进行卷积计算加速得到卷积结果;

4)卷积结果进行 BatchNorm 函数、Scale 函数、ReLU 函数、Pooling 函数等模型函数计算,进入下一次迭代循环,迭代的中间过程数据存放在 FPGA 外挂的 DDR3 内存颗粒中;

5)输出推理结果。

实现过程如图 7 所示。

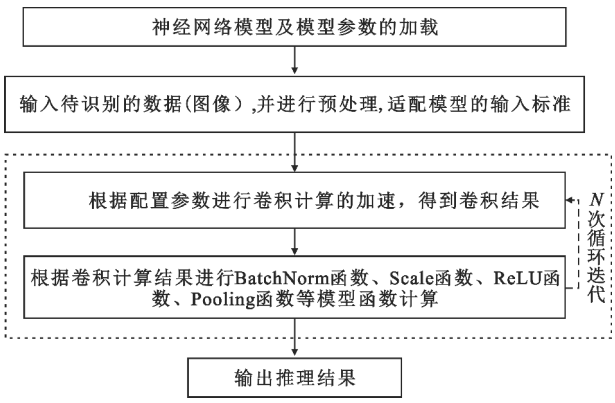

图 7 深度学习应用在 FPGA 执行器中实现

Fig. 7 The implementation on the workers for deep learning algorithms

2.5 多个深度学习应用在 FPGA 集群中的部署

试验使用的 FPGA 集群平台包含 6 个嵌入式深

度学习计算单元,每个单元包含 4 片 Xilinx 的 XC7VX690T 芯片,每片 FPGA 能够例化 4 个 DDR3 内存控制总线,因此一个单元能够加载部署 16 个 CNN 神经网络模型同时进行推理。本文采用基于 Map/Reduce 框架的实时计算图远程并行加载技术进行多个深度学习应用在 FPGA 集群中进行部署。计算图远程并行加载关键技术是一种基于 RPC 网络远程调用框架,通过节点控制器和执行器组管理器分层部署策略,将计算图中的多个操作部署到硬件资源域中的大规模数量的计算核上的应用技术。该技术能够实现多 FPGA 计算单元分区粒度的程序部署。实时计算图的运行过程就是计算图中多操作到硬件资源的部署过程,如图 8 所示。大规模计算图或者多计算图并行运行时,这些图的运行需要将计算图中的多操作并行部署到计算域中通用的计算资源中去。

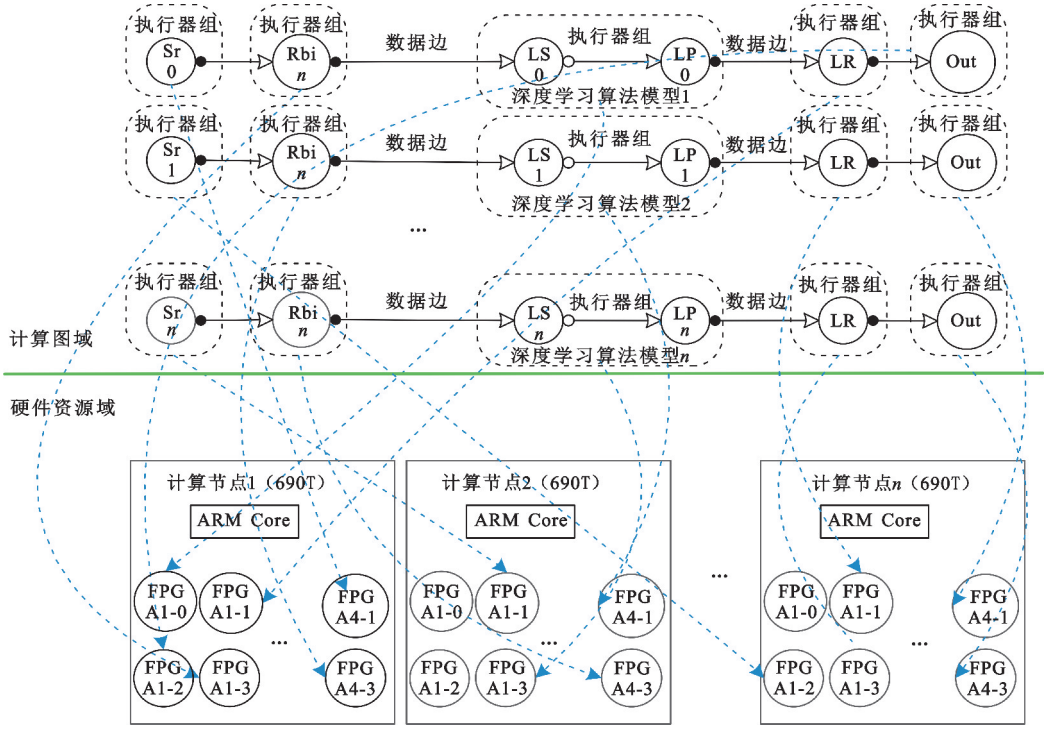

图 8　多个深度学习应用在 FPGA 集群中的部署

Fig. 8 The process of deploying multiple deep learning applications in an FPGA cluster

3　架构实现验证

实验使用的测试平台级深度学习计算单元如图 9 所示。

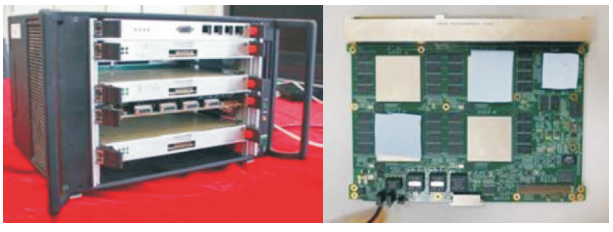

图 9　测试平台及深度学习计算单元

Fig. 9 The test platform and deep learning computing module

选取 MSTAR 数据集中的 T72、2S1、D7 三个类别数据进行训练测试。MSTAR 数据集是美国 DARPA 组织支持的 MSTAR 计划所公布的实测 SAR 地面静止目标数据,采集该数据集的传感器为高分辨率的聚束式合成孔径雷达。对采集到的图像经过处理后得到像素大小为 128 pixel×128 pixel 的静止军事车辆图像。训练集为 2 747 张,测试集为 456 张。

采用 YOLOv4 的预训练模型在服务器上进行迁移学习训练,神经网络模型的图像输入尺寸为 608 pixel×608 pixel。模型通过阶跃衰减学习率调度策略,批量大小为 8。实验以 500 000 个迭代次数进行训练,初始学习率为 0.01,分别在第 400 000 以及第 450 000 个迭代步骤之后,学习率下降为原来的 0.1 倍。此外,实验采用 0.9 的动量参数和 0.000 5 的权重衰减参数。

实验采用 MS COCO 数据集常用的评价标准中的 AP50(IoU 阈值为 0.50 的 AP)来评价性能,如表

1 所示。

表 1 部署模型测试指标

Tab. 1 Test results of the model

目标类别	AP_{50}
T72	0. 990
2S1	0. 980
D7	0. 980

对测试集的图像进行推理并进行反标,如图 10 所示,示例图片识别 T72 的准确度为 0. 99,识别 2S1 的准确度为 0. 98,D7 的准确度为 0. 98。经测试,搭建的测试平台,文中部署的 YOLOv4 模型吞吐量为 2 880 frame/min,文献[13]提到的 GTX1080 的吞吐量为 2 700 frame/min,两者性能可比。

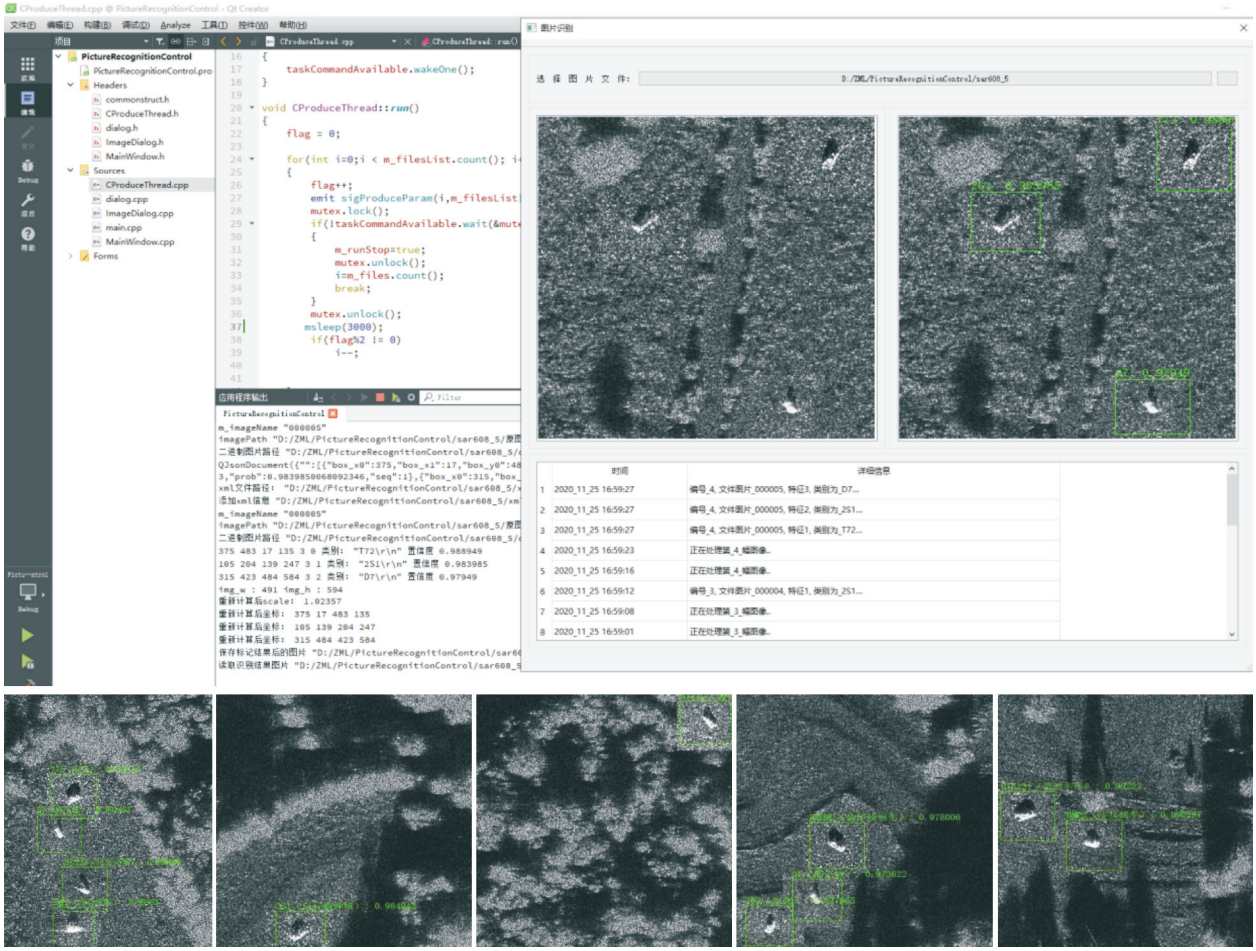

图 10 测试结果

Fig. 10 Test results

4 结 论

本文针对边缘、端设备数据量的急剧增长和芯片计算处理能力的矛盾,结合 Map/Reduce 框架,提出了一种基于 FPGA 计算集群资源的深度学习架构,能够实现多个深度学习算法的并行快捷部署和应用。同时,基于主流深度学习框架 Caffe,能够实现多种神经网络的 FPGA 解析部署。以 MSTAR 数据集进行训练和测试,在测试平台上并行部署 YOLOv4 模型,经测试,YOLOv4 模型的吞吐量为 2 880 frame/min,目标类型准确度超过 98%。

该 FPGA 集群轻量级深度学习计算框架部署不同类型算法容易,实时性高(ms 级任务响应),可扩展性好,在多种类异构传感器,大场景大数据吞吐量的情报、监视和侦察等军事场景及森林防火等民用场景有广泛的应用前景。

参考文献:

[1] 李伦平,刘达. 机载光电侦察吊舱综合信息处理技术发展与分析[J]. 光学与光电技术,2017,15(6):29-35.

[2] 费华莲. 美军电子战装备及能力发展趋势分析[J]. 飞航导弹,2020,13(6):64-69.

[3] PERRY B. New demands on ISR require new technologies and systems[EB/OL]. [2023-05-15]. https://militaryembedded. com/unmanned/isr/new -require-new-technologies-systems.

[4] 魏恒东,何丽莎. 国外电子任务飞机现状与发展趋势[J]. 电讯技术,2022,62(7):1000-1005.

[5] CHEN F. UAVs for ISR: capabilities and limitations [EB/OL]. (2015-06-17)[2023-05-15]. https://figshare. com/articles/journal_contribution/UAVs_for_ISR_Capabilities_and_Limitations/1450805.

[6] 薛俊杰,肖吉阳,祝捷. 美国军用无人机情报侦察监视应用现状研究[J]. 无人机,2020,16(11):57-62.

[7] MARK T. Global horizons final report[R]. Washington DC:United States Air Force,2013.

[8] STEVEN J. Spark structured streaming [EB/OL]. (2022-11-28)[2023-05-15]. http://spark. apache. org/streaming.

[9] 肖春华,黄樟钦,李达. 一种面向高性能计算的多 FPGA 互连结构及划分方法[J]. 计算机应用研究,2015,32(1):150-155.

[10] 胡蝶. 面向边缘的基于 FPGA 加速的分布式流处理系统[D]. 武汉:华中科技大学,2019.

[11] 张鹏,李鹏霄,任彦,等. 面向大数据流分布式流处理技术综述[J]. 计算机研究与发展,2014(2):1-9.

[12] TOSHNIWAL A,TANEJA S,SHUKLA A,et al. Storm@twitter[C]//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. New York:ACM Press,2014:1-7.

[13] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4:optimal speed and accuracy of object detection [EB/OL]. (2020-04-23)[2023-05-15]. https://arxiv. org/abs/2004. 10934.

作者简介:

刘红伟　男,1986 年生于湖北十堰,2014 年获博士学位,现为高级工程师,主要从事人工智能、信号处理等方面的研究。

潘　灵　男,1981 年生于四川仁寿,2009 年获硕士学位,现为高级工程师,主要从事嵌入式系统架构方面的研究。

吴明钦　男,1986 年生于重庆忠县,2010 年获硕士学位,现为高级工程师,主要从事信号处理等方面的研究。

韩毅辉　男,1991 年生于山西襄汾,2019 年获博士学位,现为工程师,主要从事计算数学、人工智能方面的研究。

侯　云　男,1987 年生于山西朔州,2021 年获博士学位,现为工程师,主要从事图像处理、机器学习方面的研究。

席国江　男,1992 年河北平山,2020 年于新加坡国立大学获应用数学博士学位,现为工程师,从事科学计算、并行计算方面的研究。