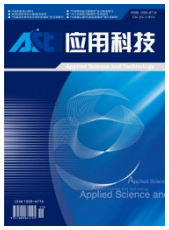

应用科技
Applied Science and Technology
ISSN 1009-671X,CN 23-1191/U

《应用科技》网络首发论文

题目：　弹性部署的分布式 AI 计算架构系统研究
作者：　李茜萌，陈春雨，何恒翔
收稿日期：　2020-07-28
网络首发日期：　2020-10-20
引用格式：　李茜萌，陈春雨，何恒翔．弹性部署的分布式 AI 计算架构系统研究．应用科技. https://kns.cnki.net/kcms/detail/23.1191.U.20201020.1327.002.html

应　用　科　技
Applied Science and Technology

DOI: 10.11991/yykj.202006001

弹性部署的分布式 AI 计算架构系统研究

李茜萌，陈春雨，何恒翔

哈尔滨工程大学 信息与通信工程学院，黑龙江 哈尔滨市 150001

摘　要：为了了解决嵌入式前端运行人工智能的现实需求与其自身性能不足的矛盾，本文提出了一种可弹性部署的分布式人工智能（AI）计算架构系统。本系统通过一套经过精心设计的网络通讯程序，将嵌入式设备需要计算的数据发送至云端，然后利用云端的 GPU 工作站集群进行高速计算。本系统能高效管理云端各个工作站的运行，可弹性部署的特性使其能依据计算负载的不同，方便快捷地增加或者减少算力，实现节能与算力的最优化。分布式的特性使本系统具有一定的灾备冗余特性，有效避免了因某几个计算节点崩溃导致整个系统瘫痪，并且运用了集群计算模式，大幅减少了系统对公网 IP 数量的要求。

关键词：人工智能；分布式计算；深度学习；云计算；嵌入式设备；网络通讯；工作站集群；神经网络

中图分类号：TP18　　**文献标志码：**A

Research on distributed AI computing architecture system with flexible deployment

LI Ximeng, CHEN Chunyu, HE Hengxiang

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Abstract: In order to resolve the contradiction between the practical needs of artificial intelligence (AI) algorithm running on the front end of embedded device and the insufficient performance, a distributed AI computing architecture system with flexible deployment is proposed in this paper. The system sends the data that embedded devices need to calculate to the cloud through a well-designed network communication program, then GPU workstation clusters in the cloud compute at high-speed. The system can efficiently manage each workstation in the cloud. Because the system can be deployed flexibly, it is capable to increase or decrease the computing power conveniently according to different computing loads. The characteristic of distributed architecture makes sure the system has certain redundancy attribute for disaster recovery. Thus, it effectively avoids that the collapse of some computing nodes causes the entire system crash. In addition, because the system works in cluster computing mode, it largely reduces the number of public network IP required by the system.

Keywords: artificial intelligence (AI); distributed computing; deep learning; cloud computing; embedded device; network communication; workstation cluster; neural networks

近年来，以 ARM 阵营为代表的嵌入式计算平台的性能高速增长带来了嵌入式产业的繁荣，ARM 芯片开始进入越来越多的平台之中，如无人机、平板、智能机器人之中。ARM 芯片内部集成了 CPU、GPU、ISP 等模块，部分芯片内甚至集成了通讯基带，这种系统级芯片（system-on-a-chip，Soc）的封装特点在于大大简化了嵌入式设备的硬件研发难度的同时，为嵌入式平台带来了强大的图像、音频及其他传感器数据获取的能力[1]。在具备了可靠的数据高速获取能力之后，如何更加高速可靠地处理这些数据则对平台性能提出了更高的要求。当前，人

收稿日期：2020-07-28
基金项目：国家自然科学基金项目(61871142)；基于人工智能架构的多传感器信息融合与决策系统的研究与实现(KY10800180032)；中央高校基本科研业务费资助项目(No.3072020CFT0803).
作者简介：李茜萌，女，硕士研究生；
陈春雨，男，副教授，博士.
通信作者：李茜萌，E-mail：liximeng@hrbeu.edu.cn.

工智能技术的第三次繁荣，其在计算机视觉(CV，Computer Vision)、自然语言处理(NLP, Natural Language Processing)等领域取得了比传统算法更加优良的效果，部分领域甚至实现了大幅度的超越。但是深度学习技术对于计算力的需求极高，很多网络需要高性能的 GPU 或者 TPU 才能运行，无法直接将其移植至嵌入式端[2]。本系统利用现代社会已经实现完善部署的 Wi-Fi 网络及蜂窝网络，将嵌入式设备需要计算的图像数据传送至云端，云端通过部署的服务器、工作站集群进行高速计算并返回结果，充分利用了嵌入式设备体积小巧、适合部署于前端的特性，以及云计算高性能、高可靠、存储容量大的特性，实现了成本及性能的最优化配置，对多种现代流行的 AI 框架都能很好的兼容。大量嵌入式设备的同时接入，对系统的计算能力也提出了巨大的挑战，本文将探究一种能够解决以上矛盾的可弹性部署的分布式 AI 计算架构系统。

1　系统各模块介绍

在传统方法中，为了让嵌入式设备采集到的数据能被高速处理，多采用有线连接的模式在其附近部署一套工作站，这种方式虽然初步实现了深度学习计算的功能，但是其能耗高、发热量大，在部分工业生产环境中是不被允许的，特别是部分现场潮湿、多尘、电压不稳的情况，这些不利因素直接导致了工作站不能稳定工作。并且因为数据并非时刻需要计算，那么这种方案也将导致计算资源长时间处于闲置状态，是一种极大的成本和资源的浪费。本文结合一些分布式设计思路[3-5]，提出了一种框架，将计算平台部署于云端，工作站集群部署于环境适宜的机房内，为设备长时间高负荷运行提供了有效保障[6]。分布式计算、弹性部署的特点能为大量嵌入式设备的计算需求提供有效的计算性能保障，同时也减少了计算资源的限制和浪费。与单纯利用嵌入式设备进行计算对比，本方案具有极大的性能优势，以 Mask R-CNN[7]网络为例，相较在树莓派 4 平台直接运行，速度提升几十倍。

本系统由中央服务器、集群服务器、GPU 工作站集群、嵌入式端 4 部分组成。中央处理器是整个网络的核心节点，负责综合调度各个设备进行工作，接收到嵌入式端的请求后会为其分配当前最空闲的计算集群的连接方式；然后嵌入式设备连接集群服务器后开始传送数据，集群服务器会进一步将数据转发给集群内运行对应模型，且最为空闲的 GPU 工作站进行计算并接收计算结果；最后将结果回传给嵌入式终端，如图 1 所示。

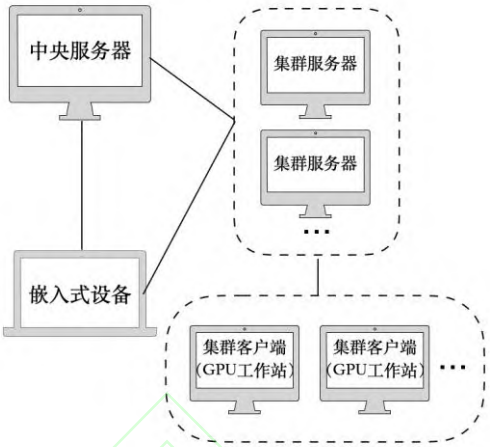

图 1　系统架构

中央服务器在整个系统中担当的是核心调度的工作，其对设备的稳定性要求极高[8]，故采用了阿里云服务器作为中央服务器的硬件载体。云服务器具有安全、可靠、稳定及可弹性增加容量等特点[9]。本系统中的云服务器配置为单核 Xeon 处理器，主频 2.4 GHz，2 GB 内存，40 GB 硬盘，1 Mb 固定 IP 带宽，操作系统为 Windows Server 2013。

为了实现多地分布式部署，一个系统中允许存在多个计算工作站集群，每个集群中都有一台专门的服务器作为集群服务器，由其与中央服务器以及嵌入式设备进行通信，然后再转发给与它连接的各台 GPU 工作站。集群服务器需要运行大量的通信进程以及调度进程，同时需要保存大量原始数据和计算结果，故对 CPU 核心数量、内存容量以及存储空间要求较高，本系统中集群服务器使用了一台搭载 i7-8700k 处理器、64 GB DDR4 内存、1 TB 固态硬盘、12 TB 机械硬盘的工作站，并为其开通了 50 Mb 带宽的固定 IP，操作系统为 Ubuntu 18.04。

一个集群内部会部署多台 GPU 工作站，GPU 作为 AI 计算的实际主体，为了尽可能多部署 GPU，每台 GPU 工作站内部都拥有多块显卡[10]。本系统采用的 GPU 工作站配置为：i9-10900k 处理器、128 GB DDR4 内存、4 块 2080Ti 显卡、2 TB 固态硬盘、12 TB 机械硬盘，操作系统为 Ubuntu 18.04。深度学习框架支持主流的 Tensorflow、Keras、Pytorch、Caffe 等[11]。

嵌入式端采用最新推出的树莓派 4B 作为主控板，其搭载了 1.5 GHz 四核 A72 核心 CPU，500 MHz 的 GPU，2 GB DDR4 内存，802.11ac 无线网卡，千兆有线网卡，USB3.0 接口。四核心的高性能 ARM 核心搭配大容量的内存，使其能运用多线程技术来

很好地支持数据采集工作，千兆网口和 802.11ac 无线网卡的配备使其能很方便地接入有线网络或 Wi-Fi 网络。再结合小巧的体积和低功耗、低发热设计，完善的操作系统适配，使其极其适合作为本系统的嵌入式端核心板，操作系统采用其官方系统（基于 Debian 开发）。

2 系统网络架构介绍

本系统是一套面向实际应用的可弹性部署的分布式 AI 计算系统，当前整套系统跨越了 Windows、Ubuntu、Debian 这 3 个操作系统，且未来很可能会移植代码至不同的操作系统进行运行。为了实现代码能方便地进行跨系统开发，我们选定了 C++作为主要开发语言，并且采用 Boost 这个实际意义上的 C++标准库来实现代码与操作系统无关的特性。同时，在深度学习模型调用部分采用 Python 编写，采用 C++调用 Python 的方法将两者融合。C++和 Python 两种语言都具有很好的跨平台适配能力，C++拥有极高的运行速度，多用于系统、平台框架开发；Python 具有开发方便、工具包众多等优点，在人工智能等领域被广泛采用。C++与 Python 的混合开发，使整个系统兼具运行速度快和开发方便的优点。

在通讯程序部分，本系统采用了 Boost.Asio 作为通讯框架，以 TCP 为通信协议。本项目需要传送大量文件数据，TCP 相比 UDP 拥有更好的可靠性，能很好地支持系统工作。通讯模式采用异步非阻塞模式，以实现在尽可能少占用资源的情况下进行大量设备的连接和数据收发[12]。在本系统中，服务器端和客户端各自拥有一个处理交互类，在该类中，定义了一系列收发处理函数还有超时检测等函数。因异步非阻塞通信的特殊性，为了实现错误重发和文件收发等功能，在类内定义了一个保存状态的标志位。默认状态标志是 normal，服务器在该状态可以接收客户端发送的字符串信息。接收到字符串后，按照指定格式解码字符串，从而提取到客户端的需求，如：登陆、心跳检测、信息传输、文件发送请求等。接着服务器端调用相信函数处理客户端需求后并回传结果信息。以收发图像为例，如客户端发送的字符串属于请求发送图像，则该字符串内同时会包含图像文件大小等信息，此时服务器的状态和客户端的状态标志都会改变为收发文件状态。下一步客户端会在收到服务器回信后发送文件，服务器端则会启动一个循环接收的程序，直至收到的文件大小等于上一步中客户端发送的文件大小值，收发成功后，服务器和客户端的状态再次回归 normal，如出错则再次进行文件传送。

2.1 中央服务器

中央服务器是整个系统的核心节点，其拥有固定公网 IP 且开发特定端口允许设备连接，其核心功能由 3 部分构成：与集群服务器通讯程序、与嵌入式端通讯程序、与其他服务器通讯程序，

如图 2 所示。中央服务器支持负载均衡功能，能为用户设备提供最合适的算力分配。中央服务器内部拥有一个储存各个集群服务器实时状态信息的 vector，该 vector 内的信息每秒会更新一次。当中央服务器收到嵌入式端计算请求后，会遍历该 vector，首先检测哪几个集群服务器能提供满足该计算需求的计算能力，接着对比这几个集群服务器负载压力，找出压力最小的那个服务器集群并将其连接方式转发给嵌入式端，以此实现负载均衡，平衡各个集群的计算压力。

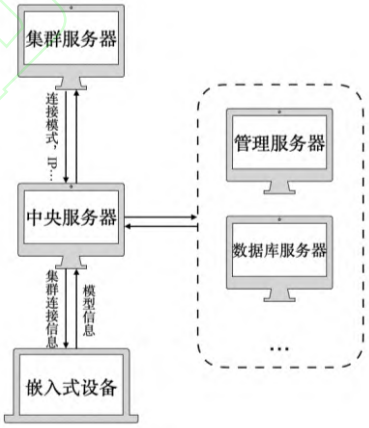

图 2 中央服务器

2.1.1 与集群服务器通信

该模块能管理接入的集群服务器信息，其内部拥有一个 vector 用于存储集群服务器的信息，包括连接模式、IP 地址、端口、当前负载、支持运行的模型名称及其版本等。

其首先会启动一个异步非阻塞模式的通讯服务器端用于接收集群服务器的连接。当一台集群服务器进行接入时，服务器端会先接收到一条带有 login 标志为用户名及密码的字符串，可以以此进行身份校验，防止恶意接入。身份校验通过后，服务器会接收到一条字符串信息，解码该字符串即可获取其连接方式，该信息将用于转发给嵌入式端。以上步骤完成后，服务器端会每秒和集群服务器通信一次，接收其发送的实时负载状态并更新 vector 中对应位

置的信息。

2.1.2　与嵌入式端通讯

该模块负责与嵌入式设备进行通讯，工作模式为异步非阻塞。当嵌入式设备接入时，同样会首先进行账号和密码验证，确认无误后进一步接收设备需要计算的模型信息。接着该模块开始检索存储有集群服务器信息的 vector 列表，找出支持该模型计算的、且当前负荷最小的那个集群，进一步取出该集群的连接方式等信息后转发给嵌入式端，待嵌入式端接收完毕后，中央服务器会断开连接以减少整个系统的负荷。

2.1.3　与其他服务器通讯

此模块主要与其他服务器，如数据库服务器、管理服务器进行通讯和信息交互，通讯模式为同步阻塞模式，属于客户端。此模块属于对外交互模块，主要用于上传状态信息、计算结果等信息给其他服务器，同时可以帮助中转其他服务器的指令给嵌入式端。

2.2　集群服务器

集群服务器扮演着一个计算集群的管理者的角色，也是整个集群对外交互的唯一设备。其作为一个客户端与中央服务器进行通讯，用于上传自身的连接方式和实时状态信息，同时其自身也作为一个服务器端与嵌入式端进行通讯，将需要计算的数据分发给各个工作站，如图 3 所示。

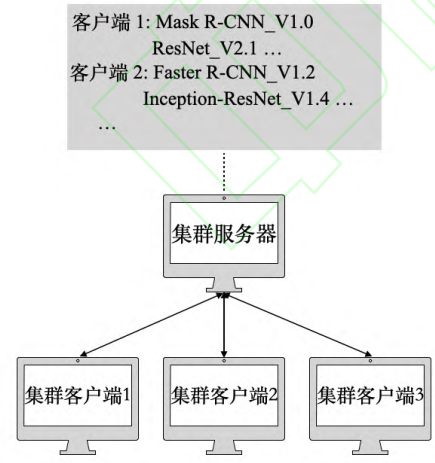

图 3　集群服务器

由于部分复杂任务需要依次经过多个模型进行计算才能完成，如先经过一个 Mask R-CNN 模型，再经过一个 ResNet[13]才能输出结果，所以集群服务器内部维护着一个上下文列表，这个上下文列表记录着每个任务需要依次调用的模型以及模型的版本。当一个计算任务到达时，集群服务器会先检索这个上下文列表，寻找到该任务需要进行的计算步骤，然后依次将该数据发送给进行对应任务的 GPU 工作站进行计算，并取回结果。

集群服务器管理着一个集群内部的所有 GPU 工作站，它会建立一个通讯服务器程序，当 GPU 工作站上的各种计算程序接入后，就会被集群服务器进行集中管理，实时监控各个任务的计算负荷、模型名称和版本等数据。

2.3　GPU 工作站集群

GPU 工作站集群由多台 GPU 工作站构成，每台都安装有多块高性能 GPU，每一块 GPU 视部署的模型大小可以运行一个或多个任务进程，每一个进程都是一个独立的程序。

当一个计算进程启动后，会首先调用 Python 加载深度学习模型，同时读取该模型的记录文档，获取模型名称和版本信息。以上步骤完成后，该进程会启动一个 Python 版的 Socket 程序主动连接集群服务器，身份校验以及字符串解码等格式与本系统中其他部分的通信程序一致，不同的是此处采用同步阻塞模式，可以大大减轻编程复杂度。依托通信程序，GPU 进程能告知集群服务器自身的模型版本以及负载压力等信息，后续当收到集群服务器发送的图像数据后，GPU 立刻开始计算并返回计算结果。为了提高软硬件运行效率，每一个计算进程都支持数据缓存队列，会提前接收后续数据放入队列中，这样可以减少每次计算的数据等待时间。

当前存在很多热门的深度学习框架，而一个任务很可能在不同的处理阶段需要依赖不同的框架进行计算。为了实现对多种深度学习框架的支持，本系统允许各个计算进程在单独的 Python 环境中运行。即利用 Anaconda 生成多个独立的深度学习环境，分别安装不同的深度学习框架[14]，然后再在各个框架内建立任务进程。各个进程都依靠网络通讯单独与集群服务器建立连接，再依赖集群服务器的上下文管理功能进行任务调度，这种模式有效避免了不同深度学习框架依赖库版本不同而导致的系统混乱，完美实现了一个系统内多种深度学习框架的共存。

2.4　嵌入式端

嵌入式端是用户使用的设备，当用户有数据需要云端进行计算时，其首先会以客户端的身份连接中央服务器的通讯端口，同时上传自身的身份验证和任务需求，在接收到云服务器返回的计算集群连接信息后，断开与中央服务器之间的连接。接着又以客户端的身份连接集群服务器，支持以 IP 和网址的形式进行连接，在连接到集群服务器后，同样是先进行身份验证，然后再上传任务需求和所需计算

的数据，如图 4 所示。因为嵌入式端需要计算的往往是图像数据，以项目中经常需要计算的 3D 图像和 RGB 高清图像数据为例，单份数据的体积往往能达到 3 MB。为了尽可能地减少网络资源占用及降低通讯延时，同时为了防止数据在公网进行传输时发生泄密，需要在数据发送前对其进行一定压缩及加密[15]。最后，嵌入式端会等待集群服务器回传计算结果，一次完整的计算过程便完成了。

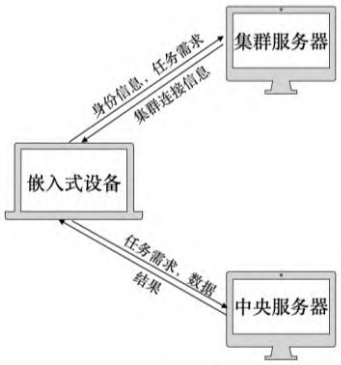

图 4　嵌入式端

3　系统性能分析

3.1　性能指标

为了使本套系统能符合项目使用需要，故列出如表 1 所示性能指标。

表 1　系统性能指标

性能	具体指标
通讯模式	异步非阻塞
客户端连接数最大值	500（服务器线程数为 8 时）
并行处理客户端最大值	50（服务器线程数为 8 时）
数据传输效率	大于 40 MB/s (Wi-Fi 环境测试，766 Mb 带宽)
编程语言支持	C++、Python
操作系统支持	Ubuntu、Mac OS、Windows
深度学习框架支持	Tensorflow、Pytorch

本系统采用 Boost.Aasio 作为核心通信框架，相比经典的 Socket 通信程序，Boost.Asio 可以看成是 Boost 库对 Socket 的进一步封装，能更好地支持异步通信模式。相比 ACE 等通信框架，鉴于 Boost 库已经成为事实上的 C++标准库，其自带的 Asio 拥有比 ACE 框架更加容易安装和使用的优势。

3.2　实际测试性能

在实际测试场景中，中央服务器带宽为 1 Mb，计算集群带宽为 50 Mb，嵌入式端上行带宽受用户实际使用场景不同而不同。每份数据经过压缩后约为 0.6 MB，因为每个设备不是连续发送数据的，经测试，整个系统可以同时为超过 50 台设备提供服务，同一时刻并发接收 8 个设备上传数据，基本接近了网络带宽极限。在客户端连接数量方面，实测最大可保持连接数可保持 500 以上，短时可保持 1 000 以上。

模型计算方面以 GPU 集群中部署的 Mask R-CNN 模型为例，计算一张图像的平均时间约为 0.15 s，数据传输时间受用户上行带宽不同而不同，按平均 0.2 s 计算，设备端完整运行一次云计算的时间花费约 0.35 s，完全满足用户需求。

经测试，整个程序可完美兼容 Ubuntu、Mac OS、Windows 这三款主流操作系统，支持 C++和 Python 混合编程，对 Tensorflow 和 Pytorch 这两款学术界热门的深度学习框架能很好地支持。

中央服务器工作状态如图 5 所示。

图 5　中央服务器工作状态

4　结论

随着嵌入式产业与人工智能产业的蓬勃发展，以及最新 5G 及 Wi-Fi 6 的入局，为大量嵌入式设备提供 AI 计算云服务成为了一个重要的研究方向。

1）本系统构建了一套可弹性扩容、分布式部署的 AI 计算系统；

2）该系统试验期间稳定运行无崩溃现象；

3）本系统是智能养殖场景下，使用深度学习方法进行体尺计算的一种创新型方法；

4）本系统具有硬件资源利用率高、成本低、安全可靠的特点，有效解决了嵌入式设备对快速可靠的 AI 计算的需求。

5）本系统很容易扩展至其他应用场景。

参考文献:

[1] 谢铖. 多内核构件化嵌入式操作系统的研究[D]. 杭州: 浙江大学, 2006.

[2] 姜典坤. 基于异构处理器的深度卷积神经网络加速系统设计与实现[D]. 北京: 北京交通大学, 2018.

[3] 刘艺. 基于 Spark 平台的分布式物流配送优化算法研究

[D]. 成都: 电子科技大学, 2020.
[4] 洪栋斌. 基于云计算的故障装备大数据技术研究[D]. 北京: 北京邮电大学, 2019.
[5] 汤闻达. 支持云雾端应用集成的资源调度策略及其优化技术[D]. 南京: 南京大学, 2019.
[6] 谈佩文. 云存储技术在食品安全视频监控领域的应用[D]. 南京: 南京邮电大学, 2017.
[7] HE Kaiming, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 42(2): 386-397.
[8] 柳赛男. 基于 Web 的制造业仓库管理物流平台关键技术及其应用研究[D]. 杭州: 浙江大学, 2007.
[9] 张焕兰. 基于 433MHz 频段的无线安全智能锁系统设计[D]. 杭州: 杭州电子科技大学, 2018.
[10] 肖汉. 基于CPU+GPU 的影像匹配高效能异构并行计算研究[D]. 武汉: 武汉大学, 2011.
[11] 舒娜, 刘波, 林伟伟, 等. 分布式机器学习平台与算法综述[J]. 计算机科学, 2019, 46(3): 9-18.
[12] 顾昕. 分布式流式计算框架关键技术的研究与实现[D]. 北京: 北京邮电大学, 2012.
[13] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016.
[14] 杜珂. 基于深度学习的图像分类关键技术研究[D]. 长沙: 国防科学技术大学, 2016.
[15] 徐通. 智慧协同网络组件层高速移动业务族群安全技术研究[D]. 北京: 北京交通大学, 2018.

本文引用格式:

李茜萌, 陈春雨, 何恒翔, 等. 弹性部署的分布式 AI 计算架构系统研究[J]. 应用科技, DOI: 10.11991/yykj.202006001.

LI Ximeng, CHEN Chunyu, HE Hengxiang, et al. Research on Distributed AI Computing Architecture with Elastic Deployment [J]. Applied science and technology, DOI: 10.11991/yykj. 202006001.