

嵌入式软件开发的三个趋势

何小庆¹, 何灵渊²

¹嵌入式系统联谊会, 北京

²博通公司, 森尼韦尔 美国

收稿日期: 2024年7月4日; 录用日期: 2024年7月25日; 发布日期: 2024年8月5日

摘要

智能时代的嵌入式系统离不开高性能、高效的软件和先进的软件开发方式。本文介绍了近期嵌入式软件开发的三个趋势: 1) 边缘计算作为一种在本地处理和分析数据的方式正在快速发展, 边缘计算与人工智能的结合正将智能计算从以云为中心的模型中转移出来; 2) 虚拟化技术是今天高算力多核处理器计算机系统采用的全新解决方案, 容器技术则可通过简化嵌入式软件开发、部署和维护来助力复杂嵌入式系统的管理; 3) DevOps的概念和实践正在逐步渗透进入嵌入式软件开发中, 助力加快软件交付速度, 提高应用程序质量和稳定性。

关键词

嵌入式系统, 边缘计算, 人工智能, 虚拟化, 软件开发

Three Trends in Embedded Software Development

Xiaoqing He¹, Lingyuan He²

¹Embedded System Association, Beijing

²Broadcom Inc., Sunnyvale, USA

Received: Jul. 4th, 2024; accepted: Jul. 25th, 2024; published: Aug. 5th, 2024

Abstract

High-performance efficient software and advanced software development techniques are keys to embedded systems in the era of intelligence. This paper describes three current trends in embedded software development: 1) Edge Computing is progressing quickly as a method to process and analyze data locally. Edge Artificial Intelligence is changing the cloud-central method of intelligent

文章引用: 何小庆, 何灵渊. 嵌入式软件开发的三个趋势[J]. 嵌入式技术与智能系统, 2024, 1(1): 1-10.

DOI: 10.12677/etis.2024.11001

computing; 2) Virtualization is the new solution for high-performance multi-core processor systems. The container technology is streamlining the management of complex embedded systems by simplifying software development, deployment, and maintenance; 3) The concept and practice DevOps is breaking ground in embedded software development. It speeds up delivery, elevates quality, and promotes stability throughout the development process.

Keywords

Embedded System, Edge Computing, Artificial Intelligence, Virtualization, Software Development

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 产业背景

2024年2月IDC发布的数据显示,2023年中国物联网(IoT)连接量超66亿个,未来5年复合增长率约16.4% [1]。随着物联网设备数量的增加和应用场景的扩展,嵌入式系统在连接和管理大规模物联网设备方面将发挥更加重要的作用。嵌入式系统需要能够处理大规模的数据,并提供安全、可靠的连接和通信功能,以支持各种物联网应用,包括智能城市、工业自动化、智能健康等。

在物联网发展过程中,边缘计算正在快速兴起。随着对实时性和隐私性要求的加强,边缘计算作为一种在本地处理和分析数据的方式变得越来越受欢迎。边缘计算能够实现快速响应和即时决策,同时减少对云端资源的依赖。嵌入式系统在边缘设备上正扮演着重要角色,边缘计算的发展趋势要求嵌入式系统的性能和能效不断提升,以满足边缘计算场景下的需求。

随着人工智能(AI)和机器学习(ML)在各个领域的应用不断深入,物联网和边缘计算的普及,嵌入式系统将有能力更加普遍地整合人工智能技术。这种能力包括在嵌入式设备上实现复杂的AI算法和模型推理,以提供更智能、更自适应的功能和服务,例如在自动驾驶汽车、智能机器人和医疗设备等领域应用。

2. 近期市场研究

关于嵌入式软件开发趋势的研究和讨论一直在积极进展中,VDC Research的多位分析师撰写了一份关于2024年物联网、嵌入式和工业技术预测的报告,报告概述了影响硬件和软件市场的关键主题和趋势,并对未来的技术发展做出了一系列预测[2]。报告中预测AI正在从云端数据中心稳步迁移到网络边缘,包括嵌入式设备。嵌入式和边缘AI市场对特定行业应用部署的优化需求日益增长,这种趋势推动边缘AI硬件变得更加面向应用。报告还预测云原生(Cloud Native)开发解决方案未来将得到普及,新冠疫情期间云原生软件开发解决方案取得了巨大的成功,嵌入式工程社区正迅速适应这一开发模式,逐渐摆脱传统的本地开发方式。

知名的嵌入式软件专家和培训师,Beningo Embedded Group的创始人Jacob Beningo,在2024年1月嵌入式软件的五个趋势的文章中指出,嵌入式软件团队正在快速采用DevOps。它为团队提供了自动化构建、测试和部署过程的方法[3]。Integrated Computer Solutions (ICS)的CEO Peter Winston在2024年3月一篇博客文章中指出:未来六大趋势对嵌入式开发将产生深远影响,它们是网络安全要求、设备的共存和互操作性、新芯片架构、连接技术、AI/ML集成,和面向服务的架构。这些趋势肯定会在短期内迅速重塑产品开发过程,并在未来持续产生影响[4]。

综合以上的观点, 在本文中我们将重点讨论 AI/ML 集成与边缘智能, 虚拟化/容器和混合部署, 以及 CI/CD 和 DevOps 这三大嵌入式软件开发趋势。

3. 趋势之一: AI/ML 集成与边缘智能

3.1. 边缘计算与边缘 AI

微软对边缘计算的定义是: 边缘计算是一种分布式计算框架, 允许 IoT 设备快速处理网络边缘的数据并对其采取行动[5]。微软给出了一个边缘计算示例: 远程仓库中的安全相机使用 AI 识别可疑活动, 并且仅将该特定数据发送到主数据中心进行即时处理。相机不会不断传输拍摄的全部视频片段, 而是只发送相关的片段, 避免每天 24 小时传输给网络带来负担, 从而释放公司的网络带宽和计算处理资源用于其他用途。

边缘 AI 正在将智能计算从以云为中心的模型中转移出来, 并使其更接近数据源。推动这一转变的动机是减少网络流量, 允许时间关键型应用(例如制造、自主系统)进行实时决策, 以及通过本地处理数据来加强隐私性。

边缘 AI 减少了对超大规模 AI 提供商的依赖, 并促进了更广泛的 AI 应用。它在医疗保健、汽车和机器人技术领域具有推动变革的潜力, 能够重塑这些行业的运营范式。展望未来, AI 将对不同类型的边缘产生不同的影响。不同类型的边缘计算如图 1 所示[6]。

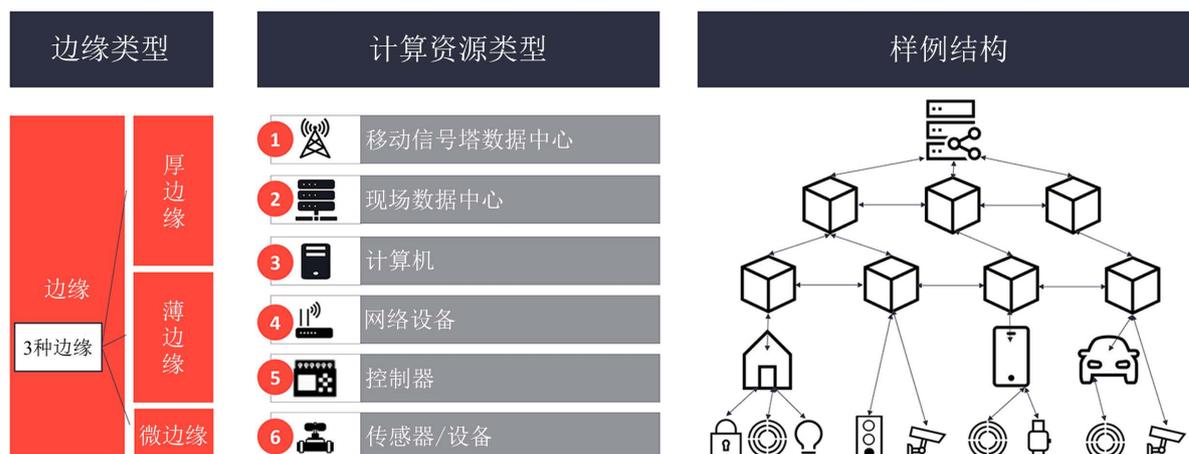


Figure 1. Three types of edges and common edge devices

图 1. 三种类型的边缘和常见的相关设备

厚边缘(Thick Edge)是指配备了高性能计算部件(例如高端中央或图形处理单元)的计算资源, 通常位于数据中心, 旨在处理计算密集型任务/工作负载, 例如数据存储和分析。薄边缘(Thin Edge)是指智能控制器、网络设备和计算机, 它们聚合了来自传感器的数据。微边缘(Micro Edge)是指生成数据的智能传感器和设备。

3.2. 边缘 AI 的细分

边缘 AI 是在边缘计算的设备上部署 AI 模型, 从而在不依赖云连接的情况下实现 AI 推理和决策。边缘 AI 也分为厚边缘 AI、薄边缘 AI 和微边缘 AI。

厚边缘 AI 的定位是在边缘服务器支持执行多个 AI 推理模型, 包括针对本地敏感数据场景的 AI 模型训练或再训练。薄边缘 AI 则是利用网关、IPC (工业计算机)和 PLC (可编程逻辑控制器)在网络边缘进行

AI 处理, 增强现有传感器和设备的智能化能力。微边缘 AI 支持将 AI 直接集成到传感器中, 提高智能系统的可扩展性, 并使日常设备能够做出自主决策。

在微边缘, 微型机器学习(Tiny Machine Learning, TinyML)是重要的技术。这个技术让在 MCU 等资源受限(如内存、计算和能耗限制)的小芯片上运行机器学习模型和算法成为可能。区别于现在流行的大模型, TinyML 的尺寸远远小于一般的机器学习模型, 和大模型更是相差甚远。现在很多的物联网设备上都已经用上了 TinyML 这项技术, 例如智能家居(恒温器和智能灯光系统)和健康监测(心率和睡眠质量)等[7]。

在 Embedded World 2024 展会上 TinyML 基金会展区展示的一个案例中, TinyML 被集成到日常物品和工具中, 使他们能够自主执行决策功能, 而无需云连接(如图 2 所示)。这种方法能增强隐私性和数据安全性。

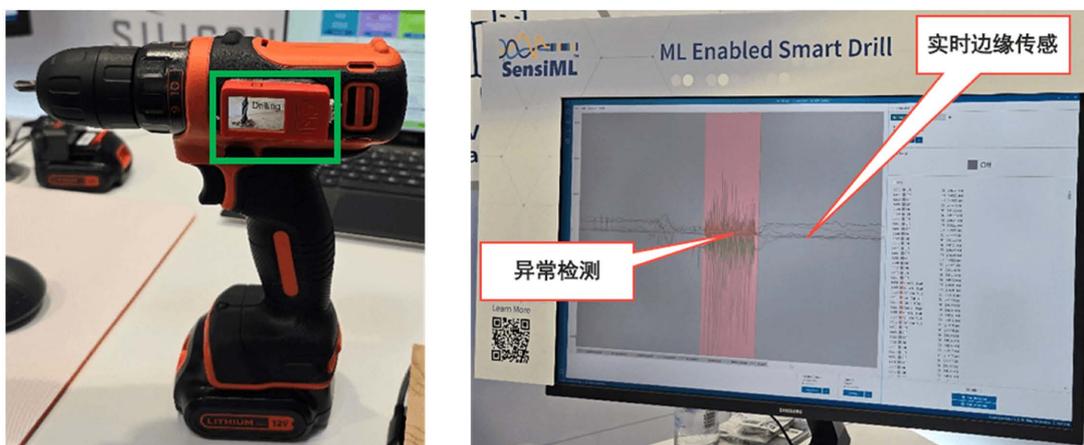


Figure 2. An electric drill and the TinyML development tool in an ESP32-based edge sensing IoT kit
图 2. 基于 ESP32、用于边缘传感的物联网开发套件中的电钻和 TinyML 开发工具

3.3. 边缘 AI 的三个趋势

今天边缘 AI 呈现出三大发展趋势: 厚边缘 AI 训练、薄边缘和微边缘 NPU (神经处理单元)加速, 以及微边缘 TinyAI 赋能。第一个趋势是 AI 模型训练正在从云转移到数据中心或微服务器等厚边缘位置。从技术上看这是可能的, 因为边缘微服务器集成了高性能 CPU 和 GPU, 可在边缘实现强大的计算, 包括 AI 训练和多种 AI 推理功能。人工智能训练也可以在地端数据中心进行, 这样做有四大好处: 减少对云基础设施的依赖, 降低成本, 增强隐私性, 以及提高人工智能应用程序在边缘设备上的响应能力。

在边缘设备中集成专用 NPU 可大大增强 AI 推理能力。NPU 可以降低功耗, 改进能耗管理, 并助推高效的多任务处理, 这使 AI 能够在功耗敏感和延迟的关键型应用(如可穿戴设备和传感器节点)中部署。Arm 的数据表明, 用于 AI 推理的 ARM Cortex A55 配置和 Arm Cortex A55 + Arm Ethos U65 NPU 比较, 后者将 70% 的 AI 推理从 CPU 转移到 NPU, 推理性能提高了 11 倍。

微边缘和薄边缘 AI 实现了自主决策的本地化。将支持 AI 的芯片组直接集成到蜂窝物联网设备中的趋势正在兴起, 这标志着物联网设备正在向能够进行本地化决策的智能、自主物联网系统转变。这一趋势预计会对智慧城市和工厂等行业产生重大影响, 并带来显著优势, 包括实时数据处理、减少延迟以及通过更小的外形尺寸提高效率。

嵌入式开发者对人工智能的开发非常重视。2023 年 Embedded.com 嵌入式开发现状调查报告显示, 针对在嵌入式开发中利用先进的技术能力这一问题, 嵌入式人工智能和机器学习最受关注, 紧随其后的是嵌入式视觉和语音功能(如图 3 所示) [8]。

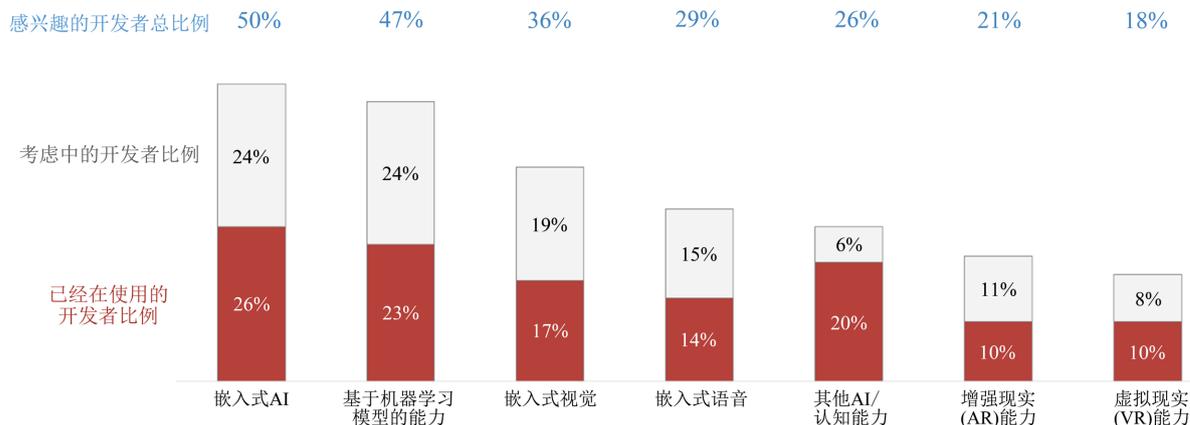


Figure 3. Using advanced technologies in embedded system development
图 3. 在嵌入式开发中利用先进的技术能力

4. 趋势之二：虚拟化、容器和混合部署

4.1. 虚拟化

虚拟化技术是操作系统的一个重要技术，在嵌入式系统中呈现蓬勃发展的趋势。借助于对底层处理器内核、内存和外设的抽象，这种技术使得多个虚拟机可以运行在同一个物理处理器上。虚拟化提供了多操作系统的运行环境，例如可以在同一个设备中同时运行高实时性操作系统(如 FreeRTOS)和通用的操作系统(如 Linux)。虚拟化技术是今天高算力多核处理器计算系统采用的全新解决方案，它平衡了通用性与可靠性两方面需求。嵌入式虚拟化的典型应用在智能汽车电子系统中，比如在一个智能座舱系统中同时运行高实时性操作系统和人机交互操作系统[9]。

智能工业场景下的混合关键系统应用可以借助多核处理器系统以及虚拟化技术部署，一个国产的虚拟化操作系统解决方案见图 4。该方案基于 Intewell 实时操作系统，是一个针对数控系统、面向低成本硬件的可配置混合异构系统解决方案[10]。

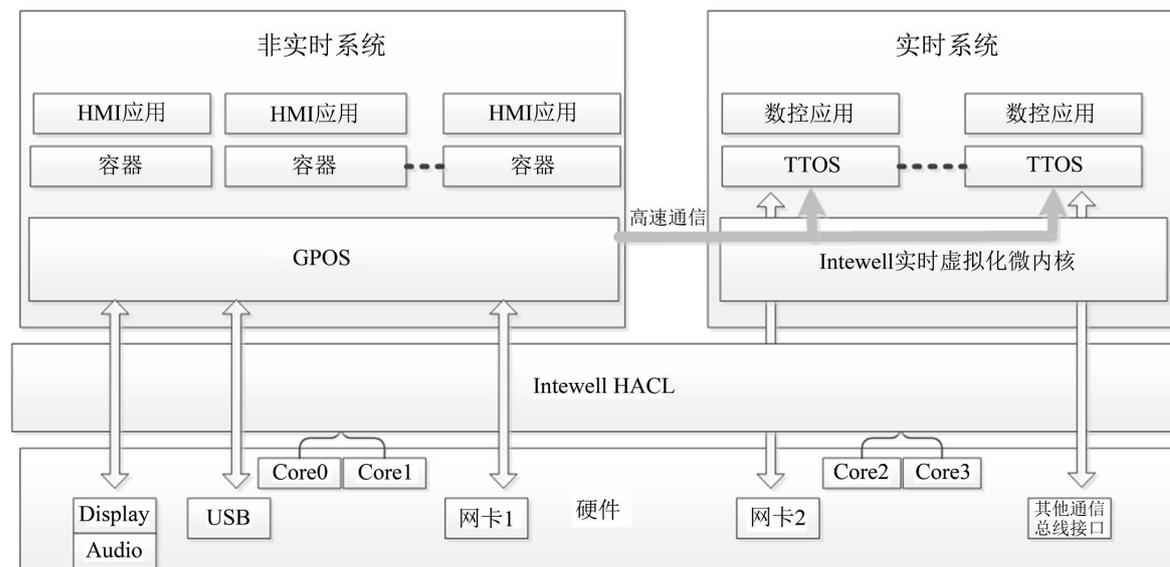


Figure 4. Intewell configurable hybrid heterogeneous system architecture
图 4. Intewell 可配置混合异构系统体系结构

4.2. 容器

容器是一种轻量级、可移植、隔离的软件环境。通过容器技术, 开发人员可以将应用程序及其依赖项打包在一起, 确保应用在不同平台上的一致性运行, 简化应用开发、部署和管理流程。容器和虚拟机具有相似的资源隔离和分配功能, 容器虚拟化软件环境, 而虚拟化则是虚拟化硬件。

编排器在容器技术中起到重要的作用, 像 Kubernetes 这样的编排器旨在自动执行容器化应用程序的部署、扩展和管理。Docker 是 Linux 环境下最常见的容器平台, 随着云计算和大数据的广泛应用, 一部分虚拟化正在被轻量级的容器技术替代, 编排器的作用也变得越来越重要。有关于编排器性能和能耗的研究受到关注, 例如郑忠斌等提出利用线性规划在提高资源利用率的同时降低碳足迹[11]。

4.3. 实时容器及其应用

实时容器是基于实时操作系统(RTOS)实现的容器机制, 贴近 Docker 用户习惯, 符合开放容器规范(OCI—Open Container Initiative), 实现轻量化、安全和便捷的容器部署。

目前, 国内针对实时容器对外公开的研究较少, 比较有代表性的有翼辉基于 SylixOS 开发的实时容器 ECS (Edge Container Stack) [12]。ECS 是轻量级实时容器技术, 使用 ECS 实时容器的 SylixOS 内核依然有着与标准版本 SylixOS 相当的性能。ECS 实时容器提供给每个容器隔离的运行环境。完整独立的容器运行环境保证了容器内的应用免受环境和其它应用的影响, 使容器的运行环境符合预期, 提升容器内应用的安全性。

邓广宏等指出, 受限于开发时间以及软件生态支持, ECS 实时容器尚不成熟, 难以应对构建大型容器云的需求[13]。国外对于实时容器的研究相对丰富, 大多基于实时 Linux 补丁与 Docker 的组合实现实时容器。NXP 的专家[14]重点研究了资源和成本受限的 MCU/MPU 领域的容器解决方案, 提出了借鉴 Android 容器技术的一种解决方案——MICROEJ VEE, 该方案可用于运动手表、智能洗衣机、智能电表和工业打印机等场景, 其软件架构如图 5 所示。



Figure 5. A small real-time container solution: MICROEJ VEE

图 5. 一种小型实时容器方案: MICROEJ VEE

总之, 容器技术可通过简化嵌入式软件开发、部署和维护来助力复杂嵌入式系统的管理。开发者在使用容器时需要注意平衡容器的优势和容器之间通信带来的复杂性和性能损失。

4.4. 混合关键系统部署

什么是混合关键系统(Mixed Criticality Systems, 简称 MCS)? 英国约克大学的专家给出了下面的定义: 混合关键系统是指包含两个或更多不同关键性级别的组件的系统[15], 例如安全关键、任务关键和非关键组件。这些系统通常存在于汽车和航空电子领域中的复杂嵌入式系统中, 它们正在向 MCS 发展, 以满足与成本、空间、重量和功耗等非功能需求相关的严格标准。

混合关键系统的研究和工程实践旨在通过部署、隔离和调度等技术手段, 实现多系统混合部署, 并实现系统间彼此隔离保护, 通过调度提升资源利用率。具体在技术层面, 学术界侧重调度方法研究, 西北工业大学的一项研究中作者提出了一种基于异构多核系统的混合关键任务调度算法[16]。工业界更多关注时空隔离下的混合关键性部署, 保证各子系统间互相隔离, 资源采用静态分配, 因此整体资源利用率不高。为了解决这一问题, 华为提出基于 openEuler 的混合关键性系统解决方案[17], 其架构如图 6 所示。



Figure 6. A mixed criticality system architecture

图 6. 混合关键性系统架构

混合关键系统的实现可以依托弹性融合底座, 底座的核心是容器和虚拟化技术。该解决方案中推荐使用轻量级容器 isula, 嵌入式虚拟化推荐使用 ZVM 和 Jailhouse。ZVM (Zephyr-based Virtual Machine) 是基于 Zephyr 实时操作系统实现虚拟化功能的开源项目, 该项目是由湖南大学嵌入式与网络计算湖南省重点实验室主导, 目前可以实现同时启动一个 Linux 与多个 Zephyr RTOS, 从而在同一硬件平台上实现混合内核部署[18]。混合关键系统目前主要应用在智能制造、机器人、能源、军工和航空航天等强实时和高安全领域。

5. 趋势之三: CI/CD 和 DevOps

5.1. 概述

DevOps 是一种软件开发实践, 可促进开发与运维之间的协作, 从而更快、更可靠地交付软件。DevOps 通常被理解一种文化, 将开发者、流程和方法连接在一起提供持续价值[19]。DevOps 在 IT 产业被广泛地使用, 其具体做法是 CI/CD (持续集成/持续交付), 这是一种通过应用开发阶段引入自动化技术快速迭代向客户交付应用的方法。与 CI/CD 关联的步骤通常被统称为 CI/CD 管道(Pipeline), 由开发和运维团队以敏捷开发方式协同完成。

随着云计算、容器、微服务等技术的蓬勃发展, 云原生的概念已被市场普遍接受。云原生技术是 DevOps 实践的核心组成部分, 通过使用云原生工具和流程, 可以实现自动化的应用程序开发、测试和部署, 从而加快软件交付速度, 提高应用程序质量和稳定性。DevOps 一站式平台则将云原生开发模式融合到产品中, 为广大开发者提供好用易用的云原生研发管理解决方案, 包含敏捷项目管理、代码管理、自动化测试管理, CI/CD 流水线等功能, 让开发、测试、部署全流程与云原生底座平台无缝结合, 降低嵌入式开发者上手的门槛, 加快 CI/CD 和 DevOps 云原生技术在嵌入式开发中的落地。

5.2. CI/CD 和 DevOps 在嵌入式开发中的实践

苏黎世应用科技大学嵌入式系统学院(InES)在工业协议认证测试中, 采用基于云原生的 CI/CD 管道技术的自动化编排方法。具体步骤有以下几个: 测试用例与测试环境与 Azure DevOps CI/CD 结合, Pipelines (编排)的集成, PROFINET (工业以太网)的概念验证, 以及创建、验证单元测试环境。最后的结论是这种方法改变了传统的 V 模型(验证模型)开发流程, 即从瀑布式转变到敏捷开发, 加快开发周期, 提高效率[20]。

西门子对当前汽车电子产业的软件工程开发做了详尽分析, 西门子的研究员指出车厂 OEM 和 OEM 的许多一级供应商正在采用软件工厂的开发方法, 比如从传统的汽车开发中分离出来的大众集团软件公司 CARIAD 和捷豹路虎的爱尔兰香农和英国曼彻斯特软件开发团队[21]。汽车软件开发正在拥抱 CI/CD, 支持多个车型和平台上整个生命周期的持续的软件更新。典型的 CI/CD 开发过程如图 7 所示。

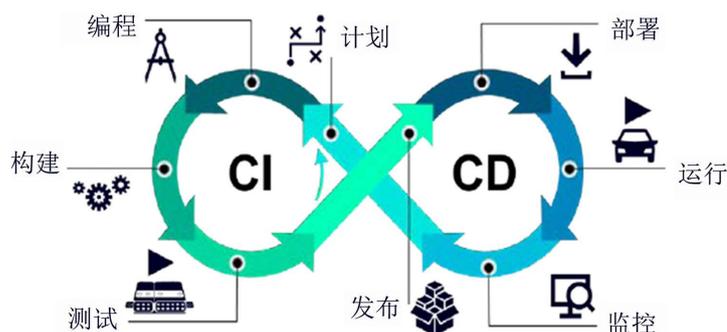


Figure 7. A typical CI/CD development process
图 7. 典型的 CI/CD 开发过程

5.3. 嵌入式 CI/CD 和 DevOps 工具

传统的嵌入式软件开发工具是一个集成开发环境, 将编辑、编译、构建和调试合成在一个以工程包为核心的开发环境中, 具有入门门槛低、集成度高以及针对性强等特点, 缺点也很明显: 跨平台、升级和第三方集成不方便。典型的传统开发工具如 IAR Embedded Workbench。

过去十年来, 在高端微控制器中运行的软件的复杂性显着增加, 比如实时操作系统(RTOS)和通信堆栈已经取得了广泛应用。随着物联网的普及, 附加软件层通信协议(MQTT)和高级安全性管理(例如基于云的设备管理)在物联网应用中是必需的。同时, GUI 库、DSP 库、机器学习软件框架等中间件也增加了手动集成软件组件的复杂性。

为了改善嵌入式开发过程, Arm 开发了 CMSIS-Toolbox 实用程序, 它是一个开源项目, 也是 Open-CMSIS-Pack 的一部分[22]。Arm 于 2014 年推出了 CMSIS-Pack, 它是 CMSIS (通用微控制器软件接口标准)版本 4 的一部分。CMSIS-Toolbox 支持使用不同工具和不同工作流程, 帮助嵌入式软件开发人员更轻松地协作, 如图 8 所示。

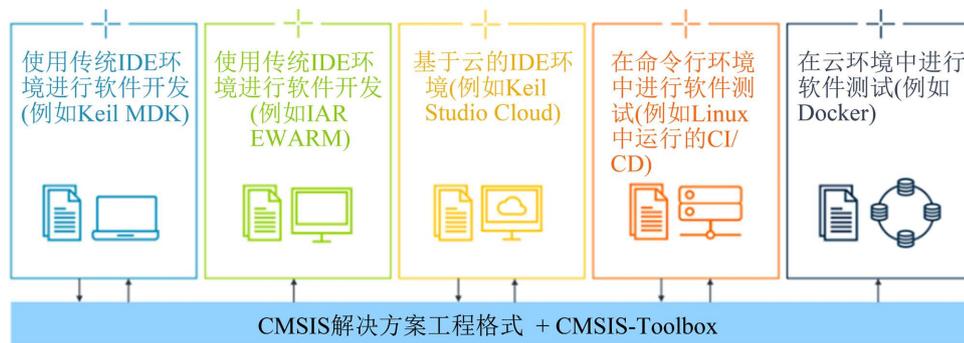


Figure 8. An open workflow involving CMSIS-Toolbox
图 8. CMSIS-Toolbox 开放的工作流程

市面上已经出现了整合云和 CI/CD 的嵌入式开发平台, 如 Wind River Studio (下文中简称为 Studio)。Studio 是一个云原生工具集, 用于开发、部署、运营和服务关键任务智能系统[23]。它为开发嵌入式设备软件平台和应用程序而构建, 包括交叉编译系统和嵌入式设备测试系统, 可查看端到端工作流程的状态, 以便诊断定制化流水线中的复杂问题。Studio 在基于云的基础设施上提供一套高度集成的工具, 实现嵌入式软件开发的自动化, 并针对平台开发人员、应用程序开发人员和测试人员实现自定义的 CI/CD 流程。

Wind River (风河)是全球知名的嵌入式软件企业, 产品包括边缘设备的实时操作系统、实时 Linux 和虚拟化技术, 以及边缘开发和云计算解决方案, 应用覆盖航空、航天、工业、汽车和通信。风河在嵌入式系统耕耘数十年, 是嵌入式操作系统业界的常青树[24]。

6. 结语

在智能系统时代, 嵌入式系统重要挑战来自软件开发。过去 40 年, 集成电路在摩尔定律推动下快速发展, 遗憾的是软件发展远远滞后了。当前, 嵌入式软件开发面临巨大挑战。以自动驾驶为代表的智能应用, 一方面必须以高安全等级为基础, 另一方面, 由于高算力多核芯片的超速发展, 基础软件需不断更新以适配新型异构多核、包含 AI/GPU 等不同芯片的处理器体系架构。此外, 微内核操作系统发展提速, 混合部署应用需要虚拟化和容器技术支撑[25]。

展望未来, 业界需要研究嵌入式软件开发趋势, 把握电子信息产业最新需求, 梳理软件工程实践经验, 改进嵌入式软件开发方法, 勇于采用新技术新平台, 从而迎接人工智能时代嵌入式软件的新机遇。

参考文献

- [1] IDC. 5G 引领物联网连接增长, IDC 发布中国物联网连接量预测[EB/OL]. <https://www.idc.com/getdoc.jsp?containerId=prCHC51842824>, 2023-02-02.
- [2] Rommel, C., Hoffenberg, S., Mandell, D., *et al.* (2024) 2024 IoT, Embedded & Industrial Technology Predictions. <https://www.vdcresearch.com/vdcc/wp-content/uploads/2024/01/2024-IoT-Embedded-Industrial-Tech-VDC.pdf>
- [3] Beningo, J. (2024) 5 Embedded Software Trends to Watch in 2024. <https://www.designnews.com/embedded-systems/5-embedded-software-trends-to-watch-in-2024>
- [4] 微软公司. 什么是边缘计算? [EB/OL]. <https://azure.microsoft.com/zh-cn/resources/cloud-computing-dictionary/what-is-edge-computing/>, 2024-01-03.
- [5] Winston, P. (2024) 6 Embedded Software Development Trends to Watch in 2024. <https://www.ics.com/blog/6-embedded-software-development-trends-watch-2024>
- [6] IoT Business News (2024) The Top 6 Edge AI Trends—As Showcased at Embedded World 2024. <https://iotbusinessnews.com/2024/04/30/34354-the-top-6-edge-ai-trends-as-showcased-at-embedded-world-2024>
- [7] Lin, J., Zhu, L.G., Chen, W.-M., *et al.* (2024) Tiny Machine Learning Projects. <https://hanlab.mit.edu/projects/tinyml>

-
- [8] Embedded.com (2023) The Current State of Embedded Development. <https://www.embedded.com/wp-content/uploads/2023/05/Embedded-Market-Study-For-Webinar-Recording-April-2023.pdf>
- [9] 何小庆. 嵌入式实时操作系统的昨天、今天和明天[J]. 中国计算机学会通讯, 2023(2): 80-85.
- [10] 郭建川, 殷灿菊. 面向数控机床异构系统架构设计的操作系统[J]. 单片机与嵌入式系统应用, 2022(3): 8-10.
- [11] 郑忠斌, 李世强, 费海平. 一种基于 Kubernetes 的工业物联网的新型调度[J]. 单片机与嵌入式系统应用, 2021, 21(6): 8-10.
- [12] 翼辉信息. ECS 实时容器[EB/OL]. <https://www.acoinfo.com/product/5328/?category=42&subCategory=7350&curCategory=5328>, 2022-05-07.
- [13] 邓广宏, 张棋恒. 基于混合关键系统的容器调度架构设计[J]. 计算机科学, 2023, 50(z1): 901-905.
- [14] Patel, M. (2024) Containers for Cost Optimized MCUs and MPUs: A Game-Changer for Embedded Systems. https://www.linkedin.com/posts/maulin-patel-6a927a9_containers-for-cost-optimized-mcus-and-mpus-activity-7193335080805715971-BeMp
- [15] Burns, A. and Davis, R.I. (2017) A Survey of Research into Mixed Criticality Systems. *ACM Computing Surveys (CSUR)*, **50**, 1-37. <https://doi.org/10.1145/3131347>
- [16] 赵瑞姣, 朱怡安, 李联. 基于异构多核系统的混合关键任务调度算法[J]. 计算机工程, 2018, 44(2): 51-55.
- [17] 余德钊. 实时内核 UniProton 及其混合关键性部署的实践[EB/OL]. https://www.esbf.org/wp-content/uploads/2023/08/202308_YDZ.pdf, 2023-08.
- [18] openEuler. openEuler 开源新项目, 嵌入式实时虚拟机 ZVM 介绍[EB/OL]. <https://www.openeuler.org/zh/blog/20230325-ZVM/20230325-ZVM.html>, 2023-03-23.
- [19] 微软公司. DevOps 教程——简介[EB/OL]. <https://azure.microsoft.com/zh-cn/solutions/devops/tutorial/>, 2024-07-01.
- [20] Lone, O., Stasiak, T., Meisterhan, J., *et al.* (2024) The Future of Industrial Protocol Certification Testing: CI/CD Pipelines, Cloud-Based, Automated and Orchestrated. *Embedded World Conference*, Nuremberg, 9-11 April 2024, 399-402.
- [21] Morris, B. (2024) What Is Coming Next with Software Defined Vehicles? An Examination of the Trends Predicted over the Coming Years. *Embedded World Conference*, Nuremberg, 9-11 April 2024, 304-309.
- [22] Yiu, J. (2024) Simplifying the Integration of Software Components in Modern Microcontroller Systems. *Embedded World Conference*, Nuremberg, 9-11 April 2024, 29-35.
- [23] Wind River. Wind River Studio. <https://www.windriver.com/studio>
- [24] 何小庆. 嵌入式操作系统风云录——历史演进与物联网未来[M]. 北京: 机械工业出版社, 2016: 18-19.
- [25] 何小庆. AIoT 时代的嵌入式技术与人才培养[J]. 单片机与嵌入式系统应用, 2020, 20(9): 6.