



未来网络白皮书

——无服务器边缘计算网络白皮书 (2021版)



第五届未来网络发展大会组委会
2021年6月

版权声明

本白皮书版权属于网络通信与安全紫金山实验室所有并受法律保护，任何个人或是组织在转载、摘编或以其他方式引用本白皮书中的文字、数据、图片或者观点时，应注明“**来源：网络通信与安全紫金山实验室**”。否则将违反中国有关知识产权的相关法律和法规，对此网络通信与安全紫金山实验室有权追究侵权者的相关法律责任。

编写说明

编写单位：

网络通信与安全紫金山实验室、北京邮电大学、

参与单位：（排序不分先后）

中国信息通信研究院、广东省新一代通信与网络创新研究院、中国电信股份有限公司研究院、中国联合网络通信有限公司研究院、中国通信标准化协会 TC610 SDN/NFV/AI 标准与产业推进委员会、EdgeGallery 边缘计算社区

主要编写人员：（排序不分先后）

谢人超、贾庆民、刘辉、黄韬、黄文浩、丁宇卿、姜伟、马军峰、刘芷若、朱伏生、卢华、段雪飞、雷波、曹畅、张岩、陈道清

前 言

无服务器边缘计算网络是融合边缘计算、无服务器计算等先进技术，高效、低成本利用边缘侧有限异构资源，基于大量无服务器边缘计算节点形成的一体化计算网络，实现计算节点内的转算存融合及节点间的网络协同，支撑海量多样智能终端、用户在边缘侧的差异化计算诉求，满足广泛接入、高吞吐和低时延的网络诉求。

本白皮书首先介绍了无服务器边缘计算网络的发展背景、基本概念、参考架构、关键技术，同时分析了无服务器边缘计算网络的典型应用场景，并探讨了无服务器边缘计算网络的生态建设。由于无服务器边缘计算网络仍处于快速发展之中，因此本白皮书还存在需要不断完善的地方，真诚地企盼读者批评指正。

第五届未来网络发展大会白皮书

目 录

前 言.....	I
一、无服务器边缘计算网络的发展背景	1
1.1 边缘计算的蓬勃发展	1
1.2 无服务器计算的逐渐兴起	3
1.3 无服务器边缘计算网络的提出及发展	6
二、无服务器边缘计算网络架构及关键技术	10
2.1 无服务器边缘计算网络参考架构	10
2.2 无服务器边缘计算网络关键技术	12
三、无服务器边缘计算网络的应用场景	18
3.1 智能制造	19
3.2 物联网.....	21
3.3 车联网.....	22
3.4 AR/VR.....	23
3.5 智慧交通	25
四、无服务器边缘计算网络生态建设与产业发展	28
4.1 开源生态发展	28
4.2 产业发展	39
4.3 标准化进展	51
五、总结与展望	54

附录 A：术语与缩略语.....	55
参考文献.....	56

第五届未来网络发展大会白皮书

一、无服务器边缘计算网络的发展背景

无服务器边缘计算网络是融合边缘计算、无服务器计算等先进技术，高效、低成本利用边缘侧有限异构资源，基于大量无服务器边缘计算节点形成的一体化计算网络，实现计算节点内的转算存融合及节点间的网络协同，支撑海量多样智能终端、用户在边缘侧的差异化计算诉求，满足广泛接入、高吞吐和低时延的网络诉求。本章对无服务器边缘计算网络的发展背景进行介绍，首先介绍和分析无服务器边缘计算网络的两项核心支撑技术，即边缘计算和无服务器计算，然后对无服务器边缘计算网络的进展进行介绍。

1.1 边缘计算的蓬勃发展

随着网络技术的不断发展，海量智能终端、移动设备产生的数据越来越多，对网络系统也提出了更高要求，例如低时延、实时处理、事件驱动开发、高效部署等；同时，各类互联网服务和应用（例如AR/VR，高清视频等）的快速发展，也给数据高效存储和处理带来新的挑战。

考虑到云计算数据中心距离用户较远，在云计算数据中心中进行数据的存储、处理、分析存在传输时延大、带宽成本高的问题。因此，业界提出了边缘计算的概念[1]，即在靠近用户终端的网络边缘部署云计算环境，就近处理数据和计算任务。然而，边缘计算的资源通常

是受限的，因此，加强边缘计算节点之间的协同，共享分布式的边缘计算资源具有重要意义。目前，边缘计算在国内外的学术、标准、技术、产业等方面均获得长足发展，其在大幅降低业务时延、减少对传输网带宽压力、降低传输成本、提高内容分发效率、提升用户体验、增强个人隐私和数据安全等方面均有显著优势，在智慧医疗、智慧工厂、智慧电网、智慧城市等行业，以及车联网、高清视频、AR/VR 等技术领域被广泛应用。

近年，国家大力推动工业互联网、5G 和物联网等方向的发展和建设，边缘计算作为上述重大战略方向的共性关键技术在国内的研究亦如火如荼，国家也出台了大量的政策，例如，2017 年 11 月，国务院发布《国务院关于深化“互联网+先进制造业”发展工业互联网的指导意见》。2020 年 3 月工业和信息化部发布的《关于推动工业互联网加快发展的通知》。2017 年 8 月中国自动化学会边缘计算专委会成立，推动相关标准及技术发展，并在多类业务场景中开展应用。2020 年 8 月，华为、中国信通院、中国移动、中国联通、腾讯、紫金山实验室等八家创始成员发起的一个 MEC 边缘计算开源项目——EdgeGallery，该项目为业界首个 5G 边缘计算开源平台，目的是打造一个符合 5G 边缘“联接+计算”特点的边缘计算公共平台，实现网络能力开放的标准化和 MEC 应用开发、测试、迁移和运行等生命周期流程的通用化[2]。2020 年 11 月，中国通信学会成立边缘计算委员会，旨在凝聚产学研各方共识，推广边缘计算成果，推动边缘计算蓬勃发展，促进行业数字化转型发展。

1.2 无服务器计算的逐渐兴起

无服务器计算（Serverless Computing）也是当前云计算领域的热点技术[3]，据云原生计算基金会（Cloud Native Computing Foundation, CNCF）定义，无服务器计算是指在构造和运行应用程序时无需管理服务器的一种计算范式。它描述了细粒度部署模型，由一个或多个函数组成的应用可上传到平台，并执行、扩缩容和基于实际运行时的资源消耗进行计费。

像使用水电一样按需或按使用付费的理念可追溯到 2006 年，云计算的出现为达到上述理念提供了一种方式，实现了自动扩缩容的按需资源分配，极大降低了管理成本，提升了用户服务质量。然而云计算的客户不得不基于分配的资源来付费，而非实际消费的资源；另外，他们还需实现自己的自动扩缩容方法，或利用云供应商提供的非应用高效的通用自动扩缩容方法。云计算使用的重型虚拟化技术本身也造成了较大的计算负担而影响一定的应用性能。因此，基于虚拟化技术的云计算尚未达到理想的目标。

同时，在开发侧，应用软件从单体架构过渡到面向服务的架构，直到当前流行的微服务架构，其基本原则即小的松耦合系统更易部署、管理和监控。从而应用被解构为更小的代码片段，即函数，形成了当前的函数即服务（Function-as-a-Service, FaaS）。当然，函数可被组合编排形成更复杂的服务。基于该思路，事件驱动的编程被引入微服务架构，赋能微服务更细粒度的开发和交互。同一时期，在部署交付侧，

为解决云计算所依赖的重型虚拟化技术带来的计算负担和性能损失，容器（集群）技术、DevOps 理念（持续集成和部署交付 CICD）等被提出。这也使得微服务架构更加流行。虽然如此，这些最新技术进展并没有弥合云计算在按需付费和灵活扩缩容上的缺失，直到 Serverless 计算的到来。

Serverless 融合了微服务、FaaS、事件驱动编程、容器化和纯粹的 pay-per-use 模型以及易扩缩容等最新技术进展。它是从 IAAS 到 PAAS 间抽象的一大步，允许在不提供任何依赖 OS 或虚拟方式（类 VM/容器）条件下执行软件。Serverless 计算并不意味着无需服务器来托管和运行代码；也不意味着运维工程师将会失业。相反，它指的是一种理念，无服务器计算的消费者无需在服务器供应、维护、更新、扩缩容和容量规划上花费时间和资源。相反，这些职责都由无服务器计算平台承担，而从开发者/IT 运维团队中完全抽离。

无服务器计算在以下三方面扮演着重要角色：

- **开发者：**基于无服务器计算平台进行服务或业务的编程，开发者可专注于业务程序的开发和优化，而无需关注系统平台的运维。
- **使用者：**按照使用的资源或调用服务的次数计费，真正做到按需使用和按需付费，极大降低使用者的成本。
- **提供方：**作为平台服务提供方，采用无服务器计算技术之后，可以进一步的提升系统资源的扩缩性能，实现更加灵活敏捷的扩缩容，进而最大化的利用基础设施资源。

无服务器计算平台包括两个方面：

- **Functions-as-a-Service (FaaS)**: 提供事件驱动的计算。开发者基于函数运行和管理应用代码,函数被事件或 HTTP 请求触发。开发者部署小的代码单元到 FaaS, 作为离散行为按需执行, 无需管理服务器或其他任何潜在基础设施并实现扩缩容。
- **Backend-as-a-Service (BaaS)**: 基于第三方 API 的服务, 代替应用中的常用功能。因为这些 API 以能透明自动扩缩容和运维的服务提供, 这对开发者来说就是 Serverless 的。

最先使用 Serverless 术语的是 2012 年 Iron.io 公司提出的 IronWorker 产品, 一个基于容器的分布式按需工作平台。从此, 在公/私有云上出现了越来越多的 serverless 实现。首先是 BaaS 平台, 如 2011 年的 Parse, 2012 年的 Firebase (分别被 Facebook 和 Google 收购)。其次是 FaaS 平台, 2014 年 11 月 AWS Lambda 项目启动, 2016 年初 IBM Bluemix 上的 OpenWhisk (即现在的 IBM Cloud Functions, 其核心开源项目成为 Apache OpenWhisk)、Google 的 Cloud Functions、微软的 Azure Functions 等也对外发布。2017 年, 华为推出 Function Stage, Oracle 开源 Fn。除了产业界 Serverless 平台外, 学术界也开发了很多简单易用的平台, 如 Openfaas、Fission、Kubeless、Knative、IronFunctions 等。当然还有很多开源 Serverless 框架。每个框架, 不管公开还是私有, 都有唯一的语言运行时集合, 和处理事件/数据的服务集合。2019 年 2 月, UCB 发表论文《Cloud Programming Simplified: A Berkeley View on Serverless Computing》, 指出 Serverless 将成为下

一代云服务的主流形态。

1.3 无服务器边缘计算网络的提出及发展

因此，随着边缘计算和无服务器计算的发展，业界逐渐意识到边缘计算和无服务器计算融合的价值和必要性，无服务器边缘计算网络的理念也随之提出。无服务器边缘计算网络将具有如下优势特性：

(1) 纯粹按需使用计费：通过 Serverless，收取费用是由真正触发的事件来驱动的，即分配的资源 and 函数触发的次数，可以极大地降低用户的使用成本。

(2) 分布式细粒度扩缩容：边缘设备上的挑战之一是资源受限。传统虚拟机内存占用大，难于扩缩容等，不是一个完美的解决方案。传统虚拟机也导致了大量重复数据带来的费用。基于轻量级抽象（含容器、Unikernel 等），Serverless 会满足较小的占用空间，细粒度自动扩缩容，因为创建/终止副本的开销相对于成熟的虚拟机来说是非常小的。当 Serverless 将函数遵循微服务架构最新进展的计算原则，代替应用作为一个完整黑盒，这将进一步得到的保证。

(3) 事件驱动的应用，在基于事件的传感器以及基于动作的 IoT 应用场景下，Serverless 可以完美适应他们。在典型的云服务下，即移动应用的后台系统下，如 API 调用、数据存储等事件在函数调用方面起到关键作用。另一方面，大部分 IoT 应用都是事件驱动的。例如，在智能农场里，有关农场管理的决策支持是传感驱动应用的典型案例；管理系统由温度、湿度变化时驱动，否则它什么都不做。在 Serverless

里，这意味着传感器基于本地数据获取后是否触发函数调用链做了基本的决策。如果没被触发，在 Serverless 平台就应该没有资源消耗。在包含云和一个边缘系统的例子里，远程传感器可能总是触发边缘系统上既定函数的执行（基于湿度/温度数据），随后后者将决策是否通过触发云平台的一个函数来升级到管理系统。在所有案例下，Serverless 的设计非常适合 Sense-Decide-Act 模式，这种模式在实践中的很多 IoT 应用里是非常典型的。

(4) 无状态生命周期，在为事件驱动的 IoT 应用设计的边缘里，被调函数执行任务，如图像处理，这种任务不知道，也不需要以往执行的知识经验。在路-车监控系统里，monitor 会调用图像供应函数来分析输入图片里的许可车牌号。该函数不需要知道过去任务的情况，仅需载入重复的代码和库，就像从头开始一样。无状态函数仅有一个任务去执行，就像 IoT 事件所需一样。无状态因此是 Serverless 到边缘的一个基本迁移。可加速这种迁移的就是 Serverless 中潜在的资源抽象，基于容器和 Unikernel。这种抽象也故意构造为无状态的。

(5) 突发工作负载，基于工作负载，81%的 Serverless 用例组成了间歇性负载，即峰值或大量涌入工作负载。这主要由 Serverless 具有的容易且细粒度扩缩容能力导致的，这避免了较长延时并提供精确资源供给。基于此，潜在遇到间歇工作负载的 IoT 用例将很适合与 Serverless 部署。流量或人群控制系统等用例都会持续监控，并暴露出间歇性事件。不能及时响应会导致不可察觉的重要事件发生。传统 IAAS/PAAS 系统可能仅通过提前分配资源（如根据一直活跃的服务

副本) 来处理爆发事件, 因为他们自动扩缩容功能会产生延时而不能满足应用需求。在边缘领域会进一步恶化, 因为边缘节点边云环境资源更少, 因此提前预留是不合适的。Serverless 针对突发流量效应的用例带来了重要好处, 因为 Serverless 允许通过空闲时收缩到 0 而峰值时快速扩容的方式探索突发工作负载的统计多路复用。基于此, 边缘计算编排将利用 Serverless 功能而非 lazy 重型虚拟化以及低效的资源预留解决方案来解决这类突发性。

鉴于无服务器边缘计算网络的显著优势, 业界纷纷对无服务器边缘计算网络开展了相关的研究, 并推出了相关的产品。2017 年, 意大利的研究机构首次将 Serverless 计算融入了边缘计算架构, 以赋能低时延应用。随后, 该研究机构进一步研究了基于端-边-云连续空间的函数管理框架, 基于资源可用性及服务质量等方面实现 Serverless 函数在端-边-云连续性空间上的适时分配。该研究机构于 2019 年在基于移动通信环境的雾计算场景(含有大量高密度分布式异构的雾计算节点)下提出 Serverless 边缘计算的一些关键特性(如低时延计算卸载、平台间协同、延时优化、随机数据分析、边缘节点间业务协同、有状态划分等), 实现了相关基于开源无服务器计算 Openwhisk 的边缘计算平台原型系统, 并做了相关评估。在产业界, 无服务器边缘计算的第一个商业尝试为 AWS 于 2017 年发布的 Lambda@Edge。在研究领域, 基于 OpenFaas 进行裁剪发布了一个基于少量资源运行的轻量级可移植的版本 faasd。

在国内, 无服务器边缘计算网络的研究刚刚起步, 成果相对较少,

但当前边缘计算集成 Serverless 已经吸引了产业界和学术界的极大兴趣。同时，无服务器边缘计算网络也面临着一些技术挑战，例如冷启动产生的长延时，不实用的为云设计的成本效益模型，持续工作负载和边缘 AI 应用（无 GPU 支持）的不适用性，缺失位置和能源感知，未考虑分布式网络，无效的数据迁移，间断资源说明，可靠性和容错考虑，无效的资源调用的可能性，安全问题，不成熟的函数触发，缺失仿真工具等。

第五届未来网络发展大会白皮书

二、无服务器边缘计算网络架构及关键技术

随着自动驾驶、智能制造等新型业务的发展，边缘计算和无服务器计算技术得到了快速发展，同时边缘计算与无服务器计算的融合也成为互联网领域发展的重要趋势。因此，介绍分析无服务器边缘计算网络的架构和关键技术对于推动网络产业发展、驱动未来网络创新具有重要意义。本章首先对无服务器边缘计算网络的参考架构进行介绍分析，然后分别对基于 Serverless 的服务管理技术、资源管理编排技术、网络控制管理技术等关键技术进行介绍。

2.1 无服务器边缘计算网络参考架构



图 2-1 无服务器边缘计算网络的参考架构

在本小节，主要介绍无服务器边缘计算网络参考架构，该架构主要五个部分：分布式边缘计算网络基础设施资源、边缘网络控制器及边缘编排管理器、基于 Serverless 的服务管理系统、智能业务与应用。无服务器边缘计算网络的参考架构包括的基础功能如下：

(1) 智能业务与应用：该部分主要是部署在无服务器边缘计算网络中的新型业务与应用，这些新型的智能业务与应用，可以通过相应的函数 API 接入并调用平台提供的函数服务；同时，通过底层网络的控制和资源的编排，实现跨网跨域的函数服务调用，进而具备分布式弹性可扩展的能力。

(2) 基于 Serverless 的服务管理系统：基于 Serverless 的服务管理系统可以认为是该架构的“大脑”，实现了对 Serverless 服务的管控，并向上支撑智能服务与应用，向下连通网络及资源。基于 Serverless 的服务管理子系统，具备 Serverless 任务分解、Serverless 服务部署、Serverless 服务发现、Serverless 服务调度（负载均衡）、协同调度等机制能力。

(3) 边缘网络控制器及边缘编排管理器：网络控制管理子系统，连接各个分布式的边缘计算节点，实现对边缘网络集中统一的管理和控制，并保证时延确定性和路径最优化，构建敏捷、高效、可定制的边缘智能网络。边缘资源编排器主要对分布式的计算资源和存储资源进行集中统一的管理和编排，包括对资源负载状况的实时监控，为函数的计算处理提供弹性的资源，具备灵活的资源扩缩能力；对接异构基础设施资源，屏蔽异构资源对应用部署的影响，实现多级泛在算力

的抽象和管理；同时对接分布式基础设施资源，实现层级式管控。

(4) 分布式边缘计算网络基础设施资源：分布式基础设施资源主要由边缘计算节点和边缘网络设备组成，为无服务器边缘计算网络参考架构提供计算、存储和网络设施资源。

基于以上无服务器边缘计算网络参考架构以及功能组成，无服务器边缘计算网络的核心特征可以概括如下：

- **使用者：**按需使用、按需付费、降低使用者的成本，避免为闲置资源买单；
- **开发者：**专注业务开发，无需关注底层基础设施资源；
- **边缘计算系统：**极致的资源弹性，高效的分布式资源协同，敏捷的分布式服务调度，最大化的利用分布式的边缘计算资源；
- **边缘计算网络：**超低的边缘网络传输时延，超高的边缘网络传输可靠性，灵活的边缘网络管控能力，边缘计算站点间网络传输时延、抖动可保证；

2.2 无服务器边缘计算网络关键技术

本小节主要介绍无服务器边缘计算网络参考架构中的关键技术，包括基于 Serverless 的服务管理技术、资源管理编排技术、网络控制管理技术等。

2.2.1 基于 Serverless 的服务管理技术

基于 Serverless 的服务管理技术主要为分布式函数计算提供支撑；

其中，无服务器计算技术（Serverless Computing）是基于 Serverless 的服务管理技术的基础。在无服务器计算中，开发者在构建和运行应用时无需管理服务器等基础设施；无服务器计算构建了一个更细粒度的部署模型，在该模型中应用被拆解为细粒度的函数并被上传到一个平台，然后根据当前所需执行、扩展和计费；在无服务器边缘计算网络中，用户不需要知道服务的部署位置或者资源的部署情况，只要网络系统能够满足用户的计算需求和服务体验即可。基于 Serverless 的服务管理技术主要具备基于 Serverless 的服务部署、基于 Serverless 的服务发现和基于 Serverless 的服务调度等能力。其中，基于 Serverless 的服务部署主要解决函数服务在哪里部署的问题以及部署在何种计算载体中的问题；基于 Serverless 的服务发现主要解决函数服务对外暴露的问题，便于计算任务的请求；基于 Serverless 的服务调度主要解决函数服务处理负载均衡的问题。

2.2.2 资源管理编排技术

资源管理编排是无服务器边缘计算网络参考架构的关键技术之一，主要通过统一协同管理分布式的边缘计算节点资源，实现对计算资源、网络资源和存储资源的管理和编排，以保证资源的按需供应、弹性灵活，进而提升分布式计算和存储资源的利用率。其中，对容器、Unikernal 等计算载体进行管理编排是资源管理编排技术的主要内容。接下来，本小节主要针对容器和 Unikernal 这两种计算载体的管理编排进行介绍分析。

当前，容器（Container）技术作为主流的轻量级操作系统虚拟化技术，通过将应用程序及其运行依赖环境打包封装到标准化、强移植的镜像中，并通过容器引擎提供进程隔离、资源可限制的运行环境，实现应用与 OS 平台及底层硬件的解耦，一次打包，随处运行。容器基于镜像运行，可部署在物理机或虚拟机上，通过容器引擎与容器编排调度平台实现容器化应用的生命周期管理。其中，Docker 技术是容器的主流技术手段。当前，业界普遍采用 Kubernetes(K8S)实现 Docker 的高效管理编排。Kubernetes 是 Google 开源的容器集群管理系统。K8S 在 Docker 技术的基础上，为容器化的应用提供部署运行、资源调度、服务发现和动态伸缩等一系列完整功能，提高了大规模容器集群管理的便捷性。Kubernetes 是一个完备的分布式系统支撑平台，具有完备的集群管理能力，多扩多层次的安全防护和准入机制、多租户应用支撑能力、透明的服务注册和发现机制、内建智能负载均衡器、强大的故障发现和自我修复能力、服务滚动升级和在线扩容能力、可扩展的资源自动调度机制以及多粒度的资源配额管理能力。同时 Kubernetes 提供完善的管理工具，涵盖了包括开发、部署测试、运维监控在内的各个环节。

另一方面，随着边缘计算的不断发展，很多边缘计算服务朝着单一核心的方向发展。为解决以传统虚拟机为载体向用户提供应用服务时所带来的资源开销大、启动速度慢、攻击面大的问题，业界提出了 Unikernal 作为新型的计算载体。Unikernal 作为一种精简、专属、全系统优化的解决方案，具有小型、快速、安全的优势，是继容器之后

的轻量级微服务系统架构。虽然基于 Docker 的容器技术解决了传统 VM 存在的系统庞大、启动速度慢的问题，且易于使用，但是大家最关心的安全问题仍然还未得到真正的解决。因此，人们仍然需要一种能够提供高级别的安全性、具有体系架构精简、高效率和高性能特性的解决方案。Unikernal 还可充分利用虚拟机管理程序的资源管理与隔离技术，提供更好的安全性。Unikernal 还在跨平台环境、大数据分析 and 横向扩展云计算服务中展现出极大的灵活性、高速度和多功能性；Unikernal 有望成为超越容器的下一代云技术；考虑到目前的云数据中心大部分都是使用强大的通用操作系统来部署应用服务，这些系统不仅消耗大量资源，还具有相当大的攻击面。而 Unikernal 推陈出新，试图将嵌入式编程技术应用于数据中心，消除了几乎所有的多余组件，在很大程度上减少了资源占用空间和攻击面。

无服务器边缘计算使得服务对单位资源的需求更加轻量化，可以引入更多的计算资源接入边缘网络来贡献闲置算力，从而形成边缘算力服务与交易过程，对于资源管理来说，可以建立依托于区块链的去中心化、低成本、保护隐私的可信交易技术。在未来泛在边缘计算场景中，无服务器边缘计算网络可以将算力资源作为透明和公开的服务能力提供给用户。在边缘算力交易过程中，算力的贡献者与算力的使用者分离，通过可拓展的区块链技术和容器化编排技术，整合算力贡献者的零散算力资源，为算力使用者和边缘服务的其他参与方提供经济、高效、去中心化、实时便捷的算力服务[8]。

2.2.3 网络控制管理技术

网络控制管理技术主要采用软件定义网络（Software Defined Network, SDN）技术以及时间敏感网络（Time-Sensitive Networking, TSN）技术，实现分布式边缘计算节点网络连接的可管可控，保证时延确定性和路径确定性。在无服务器边缘计算网络中，可以基于 SDN 技术构建边缘网络的控制面，通过要求边缘计算节点以及网络设备定期上报计算、网络和存储相关状态信息，构建计算、网络和存储的状态视图，从而实现在控制面集成计算资源的感知、网络资源感知、内容资源感知等功能。当终端设备发起服务请求时，基于 SDN 的边缘网络控制平面可以根据实时的资源状态状况，将服务请求调度至最匹配的边缘计算节点，实现任务调度的灵活敏捷和高效。

另一方面，边缘计算节点之间的计算任务分发通常要求极低的时延，以保证终端设备能做出及时的动作响应，这就要求无服务器边缘计算网络支持确定性的网络传输，以保证超低时延、超高可靠的传输。其中，通过确定性网络技术，实现边缘计算节点之间、终端设备与边缘计算节点间的“准时、准确、快速”的数据传输，进而控制并降低端到端时延，为时间敏感型网络业务提供确定性的网络传输服务。目前，可应用于无服务器边缘计算网络的确定性技术主要包括 FlexE、AVB/TSN、DetNet[4]。FlexE 应用于物理层与数据链路层之间，它通过时分复用分发机制，将多个 Client 接口的数据按照时隙方式调度并分发至多个不同的子通道，使网络即具备类似于 TDM（时分复用）

的独占时隙、隔离性好的特性，又具备以太网统计复用、网络效率高的特性。AVB/TSN 应用于数据链路层，该技术首先将网络中需求不同的流量分成不同的优先级流，将有确定性需求的流量与其余流量区分开，然后以类似“时分复用”的思想，通过不同的流量整形机制为高优先级流量提供确定的传输“时隙”，以保证时间敏感流量有一条确定的传输路径。DetNet 应用于网络层，该技术的目标是在第 2 层桥接和第 3 层路由段上实现确定传输路径，这些路径可以提供延迟、丢包和抖动的最坏情况界限，以此提供确定的延迟。

第五届未来网络发展大会

三、无服务器边缘计算网络的应用场景

无服务器计算作为新一代云计算的思想及理念，其核心是将提供服务资源的基础设施抽象成各种服务，通过 API 接口的方式提供给用户调用，落到具体技术上主要有函数即服务（FaaS）以及后端即服务（BaaS）等。其具备如下优势：第一，底层透明化，应用开发、部署无需关注底层基础设施情况，专注于业务的自身逻辑层面即可，提升业务开发效率；第二，支持服务的弹性伸缩，按需使用，适用与业务量不确定性，或者具备潮汐效应的一些场景。但不可避免的是，作为云计算的一部分，存在着与最终用户距离过远，依赖数据传送网的稳定高效等诸多因素。MEC（Multi-Access Edge Computing）即多接入边缘计算，相对于云计算集中部署的架构，边缘计算是一种分布式计算部署架构，将计算能力，业务，以及部分 5G 网络能力部署到网络边缘乃至用户侧，提供一种可靠、高效的业务体验。其具备如下优势：第一，低时延的就地数据处理；第二，敏感数据本地处理，更可靠。但是，边缘侧也就意味着单一节点所能提供服务的能力有限，无法向云计算那样无限的横向扩展。

因此，基于两者的优势结合，构建无服务器边缘计算网络可以更好的适配那些低频、频次不确定性同时时间敏感的业务场景，如：智能制造、物联网、车联网、智慧交通等。

3.1 智能制造

智能制造不仅仅是无人工厂，而应该是贯穿于产品设计、生产、管理、销售、服务等各个环节，利用人工智能、大数据等新技术，以达到提升企业洞察力、提高生产效率、强化产品竞争力为目标的完整体系。目前，世界主要发达国家政府及组织高度重视，积极出台相关战略政策，提升工业智能化水平已成为全球共识与趋势。我国近期发布的《中共中央关于制定国民经济和社会发展第十四个五年规划和二〇三五年远景目标的建议》中就明确提出了，要把发展经济的着力点放在实体经济之上，给予高端制造和智能制造大力关注。其特别强调，要坚定不移建设制造强国、质量强国、网络强国、数字中国，推进产业基础高级化、产业链现代化，提高经济质量效益和核心竞争力。同时坚持创新在我国现代化建设全局中的核心地位，加速实现关键核心技术的重大突破。近年来，我国智能制造及相关产业发展迅速，也取得了一定的成绩，但是，与世界发达国家相比，仍有不小的差距。目前来说，制约智能制造发展的因素有很多，例如：生产模式与材料缺失、信息融合困难、平台纷杂、管理方式落后、人才不足等方面。就信息融合困难这一点来说，就存在着数据采集困难、协议复杂不一、信息化与自动化融合困难，数据算法缺失等诸多问题亟待解决。

截至目前，我国已建成全球最大、最为完善的 5G 网络，为工业设备、数据的互联互通提供了高质量的传输通道。此外，飞速发展的云计算、大数据、人工智能等新技术，也在不断地尝试服务于传统制

造业，帮助企业智能化转型升级。综合行业需求及各项技术的发展，我们认为无服务器边缘计算网络能够在靠近工业现场、数据源头侧，构建融合网络、计算、存储、应用核心能力的分布式开放体系，为企业智能制造就近提供边缘智能相关服务，满足传统制造业在敏捷联接、实时业务、数据优化、应用智能、安全与隐私保护等方面的关键需求，从而实现 OT、CT 和 IT 三者跨界协作、推动信息互通融合，实现知识的模型化以及开展端到端的产业各环节协作推动制造业的智能化发展，能够有效地解决了企业转型过程中地诸多阻碍。



图 3-1 智能制造

以智慧工厂为例，传统制造工厂存在设备数量大、多次代、封闭等诸多问题，想要以较小的代价实现智能化转型，则需要打破壁垒，实现设备互联互通，数据统一融合处理，利用人工智能等技术，实现生产全流程的智能化。无服务器边缘计算网络融合 5G 网络及计算能力，首先从数据快速采集回传、协议适配统一等方面解决了设备互联

互通的问题，并能够实现数据不出园区，充分保障安全于隐私；其次，提供现场级的实时算力、极致的业务弹性支撑，实时匹配生产经营全流程的智能化需求，协助工厂以最小的代价换来制造能力的巨大提升；最终，作为工厂智能升级改造的基础平台，打造工业智能应用生态，打破传统设备的封闭性，实现 OICT 统一融合。

3.2 物联网

物联网（Internet of Things, IoT）的概念至今已为人广泛理解，作为互联网的延伸和扩展，实现了物与物、物与人的互联互通。我国物联网产业发展迅速，各类终端的数量和移动性显著增加，所产生的数据也随之越来越多、越来越复杂，对于数据传输网络及数据处理能力提出了更高的要求。作为数据传输网的补充、云计算的延伸，无服务器边缘计算网络将进一步保障智能物联网的快速发展。具体来说，首先物联网中的终端、应用的数据传输需要可靠低时延的通信网络，5G 的成熟应用及广泛部署，可以满足这一要求，而边缘计算网络则在数据源头就对数据进行处理，进一步降低时延，提升可靠性。其次某些低复杂度的终端及应用，比如远程抄表、井盖定位等，存在节点多、数据量不大、周期性采集等因素，如果将这些数据全部采用云计算方式来实现，会导致云中心需要预留大量资源来应对周期性的的大量的数据处理请求，而这些数据经过无服务器边缘计算网络处理，只需要将最终结果上报中心即可。而对于那些复杂度较高的终端，如智能工业设备、高端可穿戴设备等所产生的大量数据如果都需要传输到中

央服务器进行保存、分析、模型训练和决策，需要消耗大量的网络资源和存储资源。因此，可以在边缘处执行数据整理和分析的边缘计算可以减轻 IT 架构的负担，降低网络消耗。

例如在传统井盖定位场景下，终端定期发送状态信息至管理部门的中心进行分析，中心对相关数据进行处理分析，得出结论。但是此类场景下，终端数量巨大，活跃时间不一致，如果想通过中心的后台监测，则需要中心的设备 7*24 小时服务在线，同时考虑到大并发量，还需要保持对应服务的伸缩架构，对于管理者的建设及维护来说并不友好。随着无服务器边缘计算网络的逐步成熟，可以将相关任务在边缘侧完成处理，当终端设备上报状态，在边缘端收到此类结构化的消息后直接触发无服务计算函数，异常信息转发到后端服务做进一步业务处理，例如：发推送消息给网格管理员，同时进行集中存储等。

3.3 车联网

车联网（Internet of Vehicles, IoV）作为物联网的延伸，是以车内网、车际网和车载移动互联网为基础，按照约定的通信协议和数据交互标准，在车-X（X：车、路、行人及互联网等）之间，进行无线通讯和信息交换的大系统网络，是能够实现智能化交通管理、智能动态信息服务和车辆智能化控制的一体化网络，是物联网技术在交通系统领域的典型应用。

随着技术的发展和进步，通过传感器、GPS、摄像头等装置，车辆感知自身状态和周围环境，并通过网络实现车辆与服务器，以及车

辆和车辆之间的通信。在收集和分析了大量的实时数据之后，可以根据当前车辆的状况、路况甚至是天气等信息，得出车辆的最佳行驶路线，最终实现智能交通和自动驾驶等功能，提高人们生活的智能化程度。车联网是目前十分热门的网络技术应用场景。在车联网环境下，终端每隔 200 微秒就要发送一条信息，如果指令丢失或者延迟过大都会造成十分严重的后果。因此车联网对网络服务提出了更高的要求，包括更低的延迟、更快的数据收集和处理能力、网络架构高扩展性以及能够满足海量接入的能力。要实现毫秒级的稳定网络服务，现有的端-管-云网络架构已经无法满足需求，对现有的网络架构进行变革，未来网络将会发挥重要的作用。

低延迟和快速响应是保证车联网能够平稳运行的重要因素，5G 网络和无服务器边缘计算网络作为核心技术，在降低车联网骨干网络压力、降低业务交换延迟等方面发挥巨大的作用。端到端的时延由多段路径上的时延叠加而成，仅仅依靠局部优化无法实现毫秒级时延的需求。通过 5G 降低空口传输时延和边缘计算尽可能减少数据在网转发跳数，保障车联网业务的低延迟。通过无服务器计算，在边缘端快速响应海量车联网业务请求，满足车辆在高速移动过程中不断切换边缘节点时的业务请求响应速度。

3.4 AR/VR

虚拟现实（Virtual Reality，VR），增强现实（Augmented Reality，AR）是指通过计算将图形进行转换，后通过耳机或眼镜等可穿戴设

备创建一个身临其境的虚拟世界，如 VR 游戏、VR 旅游、AR 教育等，已经给人们的生活、娱乐、工作等方面带来了全新的体验。随着计算机技术、图像处理、计算机视觉、仿真技术、显示技术等技术的快速发展，必将真正地实现虚拟现实，从而引起整个人类生活与发展的很大变革。

AR/VR 技术的诞生已经有几十年的历史了，发展至今，国内外诸多企业、机构纷纷推出自己的相关产品，如 Facebook 的 Oculus、Google 的 VR Glass、腾讯的 TenVR 等，但是当前制约 AR/VR 产业无法大规模发展的原因还有很多，除了 AR/VR 技术的成熟度以为外，最大的因素莫过于 AR/VR 技术所需计算能力以及数据传输带来的网络时延两方面。

AR/VR 现阶段依赖的可穿戴设备算力有限，无法完成或者说快速完成图片渲染、图像转码等工作，同时频繁的计算会使得终端设备能源消耗过快，发热量巨大，无法长期使用。云端渲染转码，即将图片渲染、转码的工作放到计算资源充足的云上去处理，存储，用户使用时再从云端将内容下载到本地显示即可，但这也将带来另一个问题，即数据的传输时延问题。要想获得极致的、沉浸式的 VR 体验，尤其在实时 AR/VR 场景下，业务的端到端时延需要控制在 25ms 以内，才能满足高刷新率下，高清图像显示、复杂场景的再现。而当时延过大时，在用户侧则体现为画面显示不完整，导致头晕、目眩等不良反应，大大降低了用户体验。



图 3-2 AR/VR

当下已有部分 AR/VR 服务提供商尝试通过将服务部署在边缘计算节点以取得更好得用户体验，也取得一定得效果。例如腾讯在其 CDN 边缘节点上试水云游戏等，但是随着 AR/VR 业务的快速发展、普及，需要渲染、转码的的图像将越来越多，传统边缘计算如果通过普通服务器来处理任务，要保证所有视频、图像在预期时间内完成转码的话，则需要按照最大并发量配置相应的服务器资源，成本投入是相当巨大的。如果将分散的边缘计算节点融合成一个统一的网络，并叠加无服务器计算能力，形成无服务器边缘计算网络，当有视频、图像上传到边缘网络中，则会直接触发相关服务，实时处理，满足处理大量视频、图像数据时的极致扩容要求。

3.5 智慧交通

智慧交通是指在传统交通行业中充分运用物联网、大数据、人工智能、自动控制等新技术，致力于打造一个保障公众出行安全、提升

交通运输效率,支撑交通建设管理的交通体系,成为国家经济的快速、可持续发展的一大保障。

随着我国城市化水平的快速提升,城市交通容量急剧攀升,大量的道路监控辅助设备带来海量的数据,如高清的监控图片、视频等,这些数据处理当下还较为依赖后台低效的人工,同时将这些数据全部回传至分析中心,又会造成巨大的网络带宽浪费,在智能交通实时调度等领域,还存在实时性无法保障等问题。因此,在智慧交通场景下,以无服务器边缘计算网络为依托,在交通网络局部范围内,实现交通状况动态感知、自适应调整、智能优化。

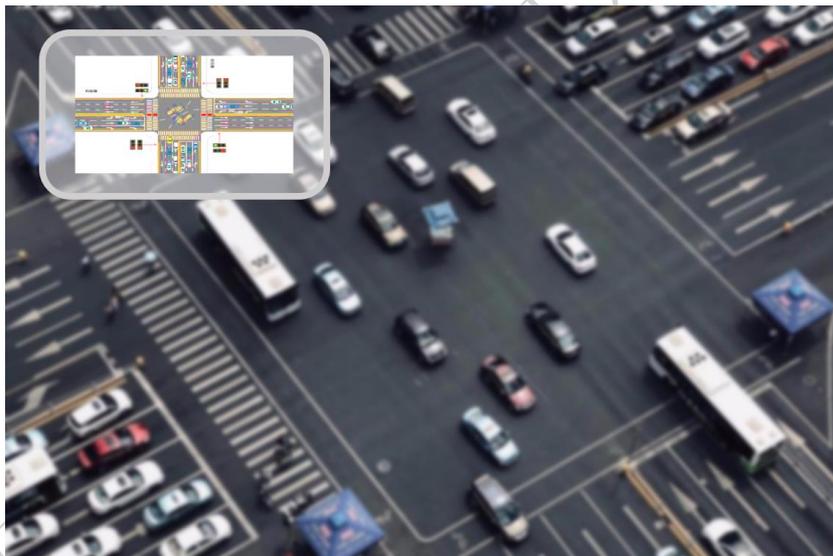


图 3-3 智慧交通

以道路拥塞场景下的车流调度场景为例,全流程涉及数据采集、数据分析、结果判定,多路口协同策略选择、路面实时控制等多个环节,而每个环节对于算力的要求各不相同。总体来说,无服务器计算网络满足了该场景下,基于既定模型,对图像、视频进行实时 AI 分析,避免了大量数据回传中心带来的延迟,同时面对城市交通的特殊

性，在早晚高峰期以及突发状况下，快速响应大量的调度请求，交通信号按需实时控制，降低交管中心信息基础建设投资。

第五届未来网络发展大会白皮书

四、无服务器边缘计算网络生态建设与产业发展

目前越来越多信息通信网络相关企业进入到边缘计算和无服务器计算相关研究中。边缘计算作为边缘网络的汇聚和控制节点，将会影响到整个信息通信网络行业的生态建设。而无服务器计算引发了企业运营思维的转变，提高了技术服务的准确度和影响力，未来的无服务器计算将会更加敏捷、灵活和强大。本章节将会探讨无服务器边缘计算网络的生态建设，包括整个能力的开发者、使用者和运营者。

4.1 开源生态发展

无服务器边缘计算网络开源生态主要由边缘计算和无服务器计算的相关开源项目构成，本小节对主流的边缘计算开源项目和无服务器计算开源项目进行介绍。未来这些项目直接可能会相互渗透、融合，从而形成无服务器边缘计算网络整个生态的重要组成部分。

4.1.1 边缘计算开源生态

(1) EdgeGallery

EdgeGallery 是由华为、紫金山实验室、中国信息通信研究院、中国移动、中国联通、腾讯、九州云、安恒信息等八家创始成员发起的一个 MEC 边缘计算开源项目。目的是打造一个符合 5G 边缘“联接+计算”特点的边缘计算公共平台，实现网络能力（尤其是 5G 网络）开放的标准化和 MEC 应用开发、测试、迁移和运行等生命周期流程

的通用化。目前 EdgeGallery 整个边缘计算平台是基于 Kubernetes 打造，在规划中会引入无服务器计算的相关特性，提升在监控、伸缩性方面的能力。

EdgeGallery: 打造MEC统一生态, 加速MEC商用节奏



(2) KubeEdge

KubeEdge 是一个开源系统, 用于将容器化应用程序编排功能扩展到 Edge 的主机, 是一个面向边云协同的云原生边缘计算框架。KubeEdge 在 Kubernetes 原生的容器编排调度能力之上实现了边云之间的应用协同、资源协同、数据协同和设备协同等能力, 完整打通了边缘计算中云、边、设备协同的场景。KubeEdge 架构上分为云、边、端三个层次。云端负责应用和配置的校验、下发, 边缘侧负责运行边缘应用和管理接入的设备, 设备端运行各种边缘设备。KubeEdge 完整打通了边缘计算中云、边、设备协同的场景。

KubeEdge 的功能特点如下: 1) 支持复杂的边云网络环境: 双向多路复用边云消息通道提供应用层可靠增量同步机制, 支持高时延、

低质量的网络环境；2) 应用/数据边缘自治：支持边缘离线自治及边缘数据处理 workflow；3) 边云一体资源调度和流量协同：支持边云节点混合管理、应用流量统一调度；4) 支持海量边缘设备管理：资源占用业界同类最小；提供可插拔设备管理框架，支持自定义插件扩展；5) 开放生态：100%兼容 Kubernetes 原生能力；支持 MQTT、Modbus、Bluetooth、Wifi、ZigBee 等业界主流设备通信协议。

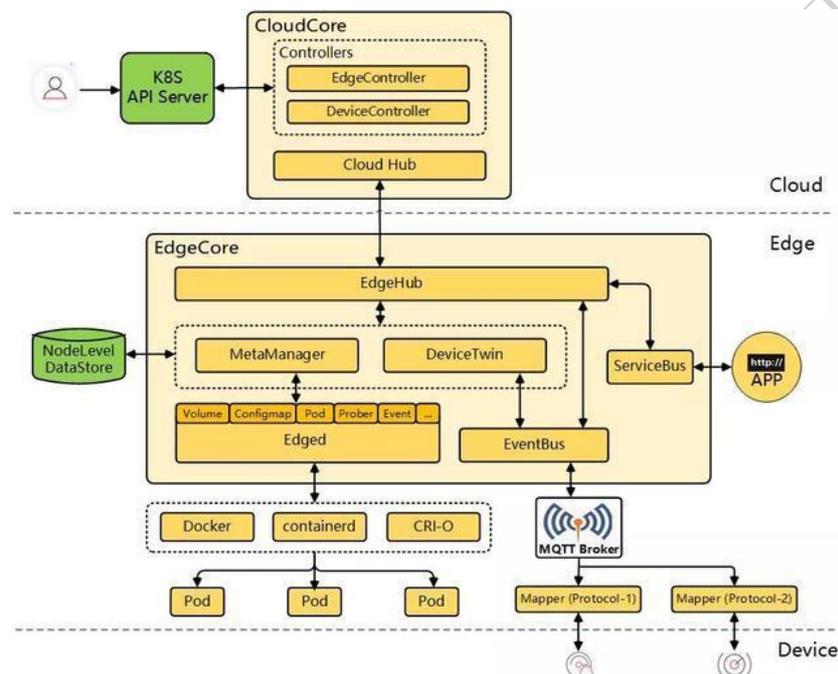


图 4-2 KubeEdge 介绍

(3) K3S

K3S 是专为在资源有限的环境中运行 Kubernetes 的研发和运维人员设计，通过消除安装 Kubernetes 的复杂性和学习成本，K3S 极大地简化了边缘部署的复杂度。K3S 与架构无关，并且占用空间极小。K3S 的易用性使组织可以从堆栈中获得更高的价值，将集群部署至数百甚至数千个地点，并快速启动这些集群。目前 K3S 可以使用 Istio 支持快速部署无服务器容器，自由的进行容器的扩缩容，未来随着

K3S 的应用越来越广，无服务器边缘计算的特性会得到进一步的发展。

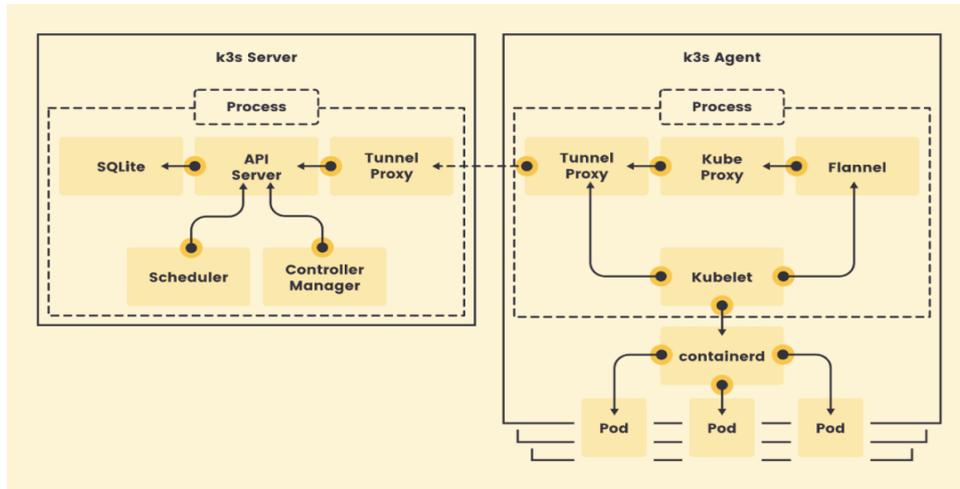


图 4-3 K3S 架构

(4) OpenYurt

OpenYurt 是阿里云开源的云原生边缘计算解决方案。已经应用于 CDN、音视频直播、物联网、物流、工业大脑、城市大脑等实际应用场景中，并服务于阿里云 LinkEdge、盒马、优酷、视频云等多个业务或项目中。目前开源的能力包括了边缘自治能力和原生 K8S 集群一键式转换为边缘集群。OpenYurt 已经具备了边缘自治、高效运维通道、边缘单元化管理、边缘流量拓扑管理，安全容器、边缘 Serverless/FaaS、异构资源支持等能力。

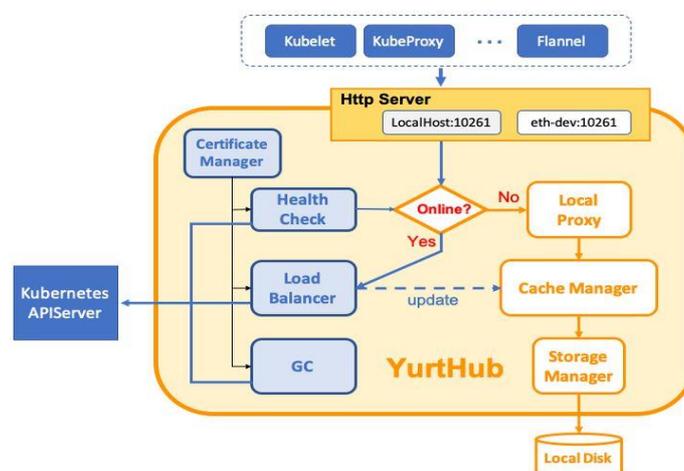


图 4-4 OpenYurt 架构

4.1.2 无服务器计算开源生态

Kubernetes 的蓬勃发展由催生了一系列以它为基础的 Serverless 框架，目前开源的 Serverless 框架大多以 Kubernetes 为基础，本小节主要介绍 Knative、OpenFaaS、OpenWhisk 和 Kubeless。

(1) Knative

2018 年 7 月，Google 发布了 Knative 无服务器开源平台[5]。Knative 是 Google 开源的基于 Kubernetes 和 Istio 的 Serverless 开源实现，目标是为了提供更高层次的抽象，让开发者无需关注基础设施（虚拟机或者容器，网络配置，容量规划），而专注于业务代码即可，旨在标准化 Serverless。只需使用几个 YAML 文件就可以轻松地开始使用 Knative 了。这也意味着，在本地或者托管云服务上，任何可以运行 Kubernetes 的地方都可以运行 Knative 和业务的代码。目前参与的公司主要是 Google、Pivotal、IBM、Red Hat，Knative 是为了解决容器为核心的 Serverless 应用的构建、部署和运行的问题。

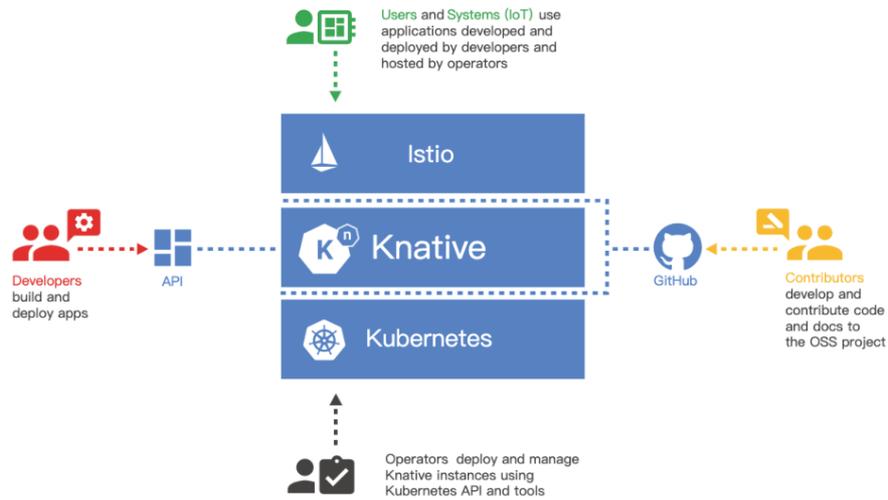


图 4-5 Knative 层级关系

Knative 采用 Go 语言编写，支持 C#, Go, Java, Node.js, PHP, Python, Ruby, and Rust 语言。包含 Serving 和 Eventing 两大组件，他们都通过 Kubernetes custom resource definitions (CRDs)来配置与运行。

Serving: 基于 Kubernetes 和 Istio 平台部署，运行无状态的应用和函数。Serving 通过使用 Kubernetes CRDs 来控制以下特性：Revision（修订版本）、Configuration（配置）、Route（路由）和 Service（服务），来实现应用的路由、升级策略、自动扩缩容等功能。

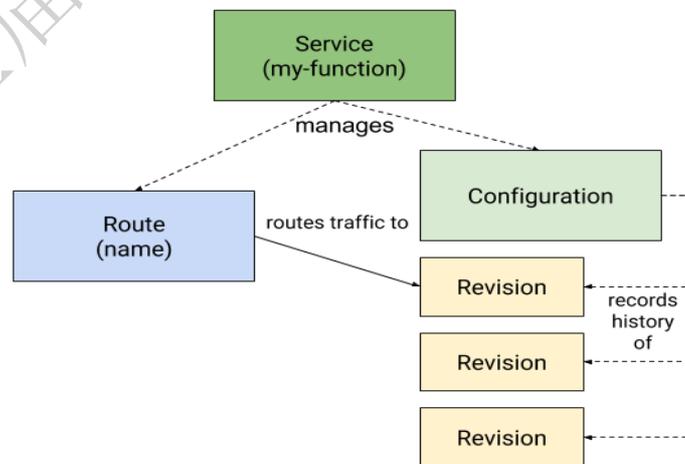


图 4-6 Knative 架构

Eventing: 事件系统，引入 Source（源）、Channel（通道）和 Subscription（订阅）概念，无需特定代码来选择消息代理，自动完成事件的绑定和触发。

(2) OpenFaaS

OpenFaaS 无服务器功能框架，通过将功能打包，无需重复的样板化编码，简化操作流程[6]。具有以下优点：

- 提供易用的 UI 界面，一键点击安装即可轻松使用。
- 支持 Linux 或 Windows 平台编写函数功能，并打包 Docker 镜像格式。
- 轻量可移植，可以通过容器方式，运行在现有硬件或者云环境中。
- 支持命令行工具，可以通过 YAML 制作模板和定义函数。
- 可以根据实际需求，灵活自动扩缩。

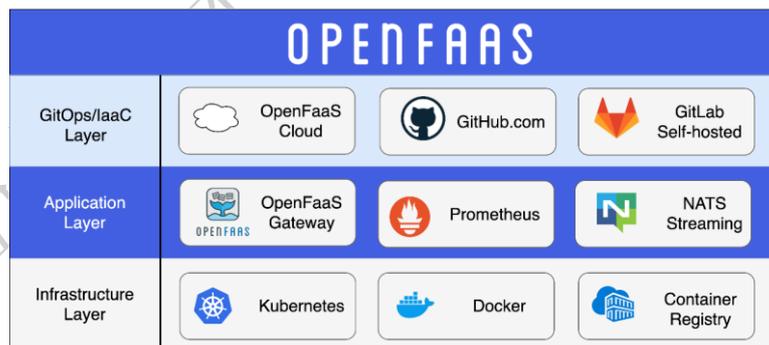


图 4-7 OpenFaaS 运行栈

运行栈：

- OpenFaaS Cloud 层位于最上层，来分发 GitOps 操作。
- NAT 层，提供异步执行和查询功能。

- Prometheus 提供平台指标收集，结合 AlertManager，实现服务的自动扩缩功能。
- 容器仓库，可以维护一些不可变的组件，可以通过 API 快速部署。

在 OpenFaas 整个工作流程中用户通过 CLI 或者 UI 界面，向 OpenFaas 网关发送 RESTAPI 请求，操作平台功能。平台上所有的服务和函数都通过默认路由对外暴露。Prometheus 搜集平台指标为自动扩缩功能提供参考条件，Faas-netes 为平台提供编排支持，当然 Docker Swarm, Hashicorp Nomad, AWS Fargate/ECS, and AWS Lambda 这些编排工具也是支持的，需要内置到 faas-provider SDK 中来使用。

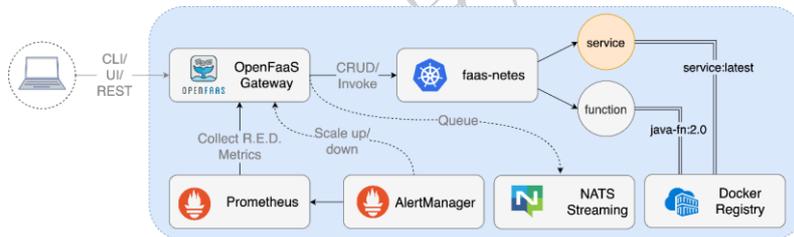


图 4-8 OpenFaas 工作流程

(3) OpenWhisk

OpenWhisk 是一款分布式的 Serverless 开源平台，最早来源于 IBM 的 Serverless 平台，目前由 Apache 基金会进行孵化和管理[7]。OpenWhisk 是一个功能完备的 FaaS 平台，包含事件驱动及函数执行时等核心组件，可以运行在不同的基础架构上，如物理机、虚拟机、容器平台、PaaS、公有云和私有云等。OpenWhisk 拥有大量的代码库、高质量的特性和众多的贡献者，但是这个平台上庞杂的工具

(CouchDB、Kafka、Nginx、Redis 和 Zookeeper) 也给开发人员带来了挑战。此外，这个平台在安全性方面还有待完善。

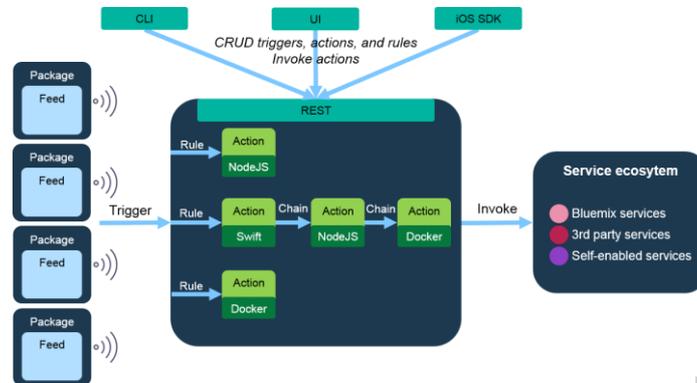


图 4-9 OpenWhisk 全局架构

在 OpenWhisk 的工作流程中包括以下组件：

- **Nginx:** Nginx 是一个 Http 反向代理服务器，作为整个系统的入口，主要用于接受外部请求，转发给内部 Controller 处理。
- **Controller:** Controller 为 Scala 语言实现的 REST API, Openwhisk 的一个核心组件，用于 Openwhisk 内部对象的 CURD（增删改查）操作和 Action 调用。
- **CouchDB:** CouchDB 一个开源的面向文档的数据库管理系统，在该平台中被用于认证和鉴权，通过检索 CouchDB 里的数据来查看用户是否存在以及是否有调用对应 Action 的权限。此外，CouchDB 还会加载相应的 Action 及运行参数，处理结果也会存入该数据库中。

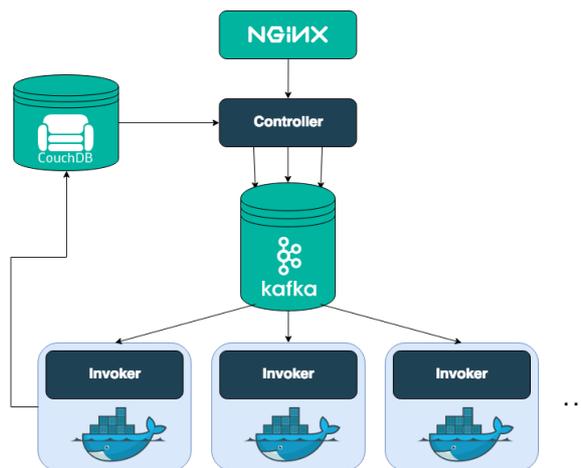


图 4-10 OpenWhisk 工作流程

- **Load Balancer:** Load Balancer 作为 Controller 的一部分，通过不间断的健康检查，可以获得全局的 Invokers 的状态，从而得知哪些 Invokers 可以被用于处理动作请求。
- **Kafka:** Kafka 为 Openwhisk 提供了消息队列功能，可以防止系统奔溃时，调用丢失，还能起到削峰填谷的功能，防止系统过载。从 Controller 发出来的所有 Action 调用，都会转化成对应的消息，在 Kafka 里缓存，等待被执行。
- **Invoker:** Invoker 是 Openwhisk 最核心的部分，是各类 Action 的最终执行者。通过 Docker，Invoker 可以独立安全的执行动作，每次执行动作都会创建 Docker 容器，相关操作代码会被注入其中，在执行得到结果后，容器会被销毁回收资源。

(4) Kubeless

Kubeless 是运行在 Kubernetes 平台之上的 FaaS。Kubeless 官方强调其是 Kubernetes 原生 (Kubernetes native) 的 Serverless 实现。Kubeless 在设计之初就引用了许多 Kubernetes 原生的组件，如 Service、

Ingress、HPA (Horizontal Pod AutoScaler) 等。此外，相比于其他开源 Serverless，可以很方便的从商业平台中（如 AWS Lambda, Azure Functions, and Google Cloud Functions）克隆相应的 Serverless 解决方案。由于 Kubeless 的功能特性是建立在 Kubernetes 上的，所以对于熟悉 Kubernetes 的人来说非常容易部署 Kubeless，其主要实现是将用户编写的函数在 Kubernetes 中转变为 CRD (Custom Resource Definition, 自定义资源)，并以容器的方式运行在集群中。

Kubeless 主要特点：

- 支持主流的 Python, Node.js, Ruby, PHP, Golang, .NET, Ballerina 和自定义的运行环境；
- 兼容 AWS Lambda CLI 命令；
- 使用 Kafka 消息队列作为事件触发器，同时支持 http 事件；
- 采用 Prometheus 监控函数调用和延迟数据；
- Serverless 框架插件化；

主要组件：

Runtimes: 运行时环境，即 Serverless 函数被触发运行的环境，默认支持几乎所有的主流函数运行时，分为稳定与孵化版本，支持 Python、Nodejs、Ruby、Go、Java、C#、Ballerina 以及用户自定义的 Runtime，每个函数运行时都会以镜像的方式封装在容器镜像中，通过在 Kubeless 配置中引用这些镜像来使用，可以通过 Docker CLI 查看源代码。

Triggers: Triggers 表示函数的事件源，当事件发生时，Kubeless

确保最多调用一次函数，Triggers 可以与单个或多个功能相关联，具体取决于事件源类型。Triggers 与函数的生命周期解耦，可以进行如下操作：

- Create: 使用事件源和相关功能的详细信息创建 Triggers;
- Update: 更新 Triggers 元数据;
- Delete: 删除 Triggers 及为其配置的任何资源;
- List: 显示 Triggers 列表;
- Function: Functions 表示要执行的代码, 即为函数, 在 Kubeless 中函数包含有关其运行时的依赖、构建等元数据, 具有独立的生命周期, 并支持以下方法:
- Deploy: Kubeless 将函数部署为 Pod 的形式运行在 Kubernetes 集群中, 此步骤会涉及构建函数镜像等操作;
- Execute: 执行函数, 不通过任何事件源调用;
- Update: 修改函数元数据;
- Delete: 在 Kubernetes 集群中删除函数的所有相关资源;
- List: 显示函数列表;
- Logs: 函数实例在 Kubernetes 中生成及运行的日志。

4.2 产业发展

4.2.1 边缘计算的产业发展

2015 年 8 月, ETSI 第一次提出了 MEC 的验证框架, 经过近 5

年的演进，相关标准体系也逐渐清晰。ETSI 定义 MEC 意义在于将边缘计算从 IOE 的视角扩展到 ICT 的视角。ETSI MEC ISG 标准委员会的董事 Alex Reznik 给了一个宽泛的定义：“任何不是传统数据中心的都可以成为某人的边缘节点。”ETSI MEC 定义是“为应用开发者和内容提供商提供在（运营商）的网络边缘侧的云计算能力和 IT 服务，这一环境的特点是极低的延时和极大的带宽，支持针对应用侧无线网络的实时访问。”目前边缘计算几个主要参与者和产品的介绍如下：

（1）微软

微软在边缘计算领域拥有 300 项专利，其中许多专注于内容流媒体。最近还推出了 Azure IoT Edge 服务，由容器模块，边缘运行时和基于云的管理界面组成。

Azure IoT Edge 将云分析和自定义业务逻辑移到设备，通过将业务逻辑打包到标准容器中，横向扩展 IoT 解决方案，然后将这些容器部署到任何设备，并从云中监视所有这些设备。如果希望尽快响应突发事件，可以在边缘运行异常情况检测工作负荷。如果想要降低带宽成本并避免传输数 TB 的原始数据，可以在本地清理和聚合数据，然后只将处理结果发送到云进行分析。

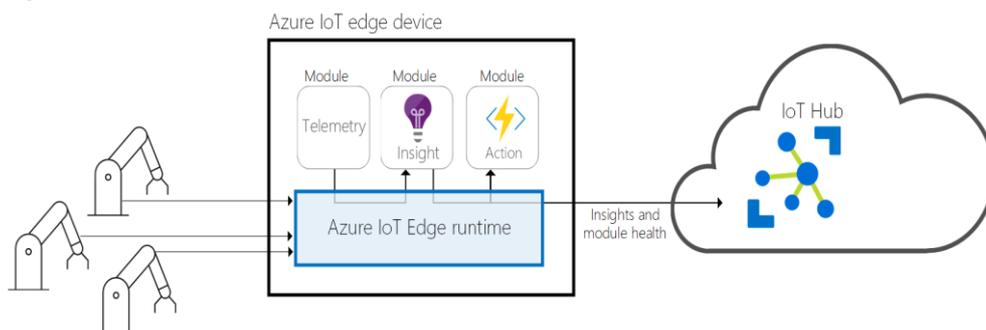


图 4-11 Azure IoT Edge 工作原理

(2) 中国联通

联通旗下的沃云推出了基于云网边一体化的边缘云产品，该产品能与 5G 技术高度融合，还可以基于中国联通遍布全国的计算资源和网络资源，充分发挥算力、网络下沉优势，将边缘节点广泛部署在靠近用户业务终端的位置，并通过边缘云管理平台实现云边组合统一调度，提高边缘能力协同。

同时，中国联通在算力网络的研究中，面向边缘计算节点和边缘网络，正在开展可提供 serverless 能力的算力网络平台研究，以实现无服务器关键技术在算力网络领域相关研究的部署验证，研究在 Serverless 框架下的服务化架构演进，探索边缘算力的在 serverless 框架下的服务抽象，以及研究服务管理功能、边缘网络控制功能、边缘资源编排管理功能等关键能力，基于资源的维度，统一协同管理分布式的边缘计算节点资源，实现转算存资源的感知、分配、弹性横向扩展；基于服务维度，统一进行网络和算力的编排和调度，实现边缘算力节点集群的算网一体化

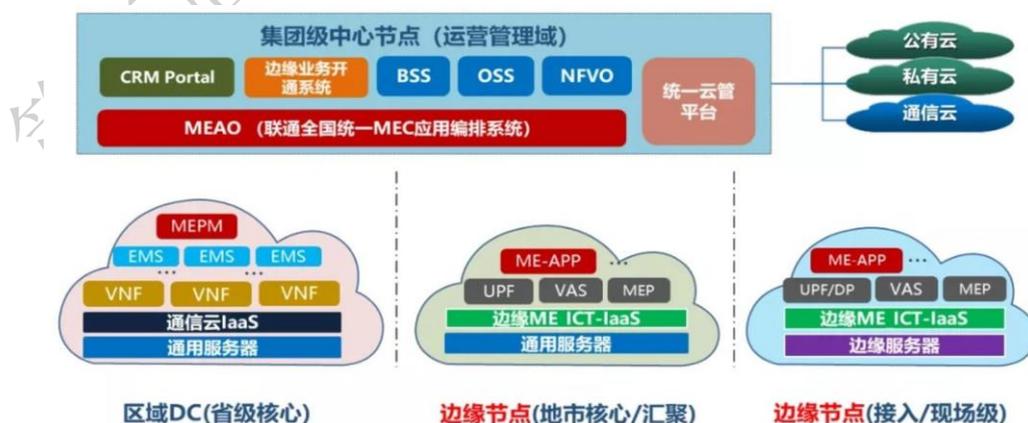


图 4-12 中国联通边缘计算架构

(3) 中国移动

中国移动结合自身网络优势及自主研发能力，设计并研发了 OpenSigma 边缘计算通用平台，依托 5G 网络的建设，为客户的实时性和带宽密集型业务提供更好的支持。OpenSigma 包括边缘计算业务运营平台 OpenSigma-ECM 和边缘计算 PaaS 平台 OpenSigma-ECP，具备端到端的边缘计算资源的运营和全生命周期管理能力，能够实现网边云的有效协同。边缘计算通用平台已在工业、交通、文娱、智慧城市、医疗等核心应用场景试用，对业务时延及数据安全等诉求都有显著改善。

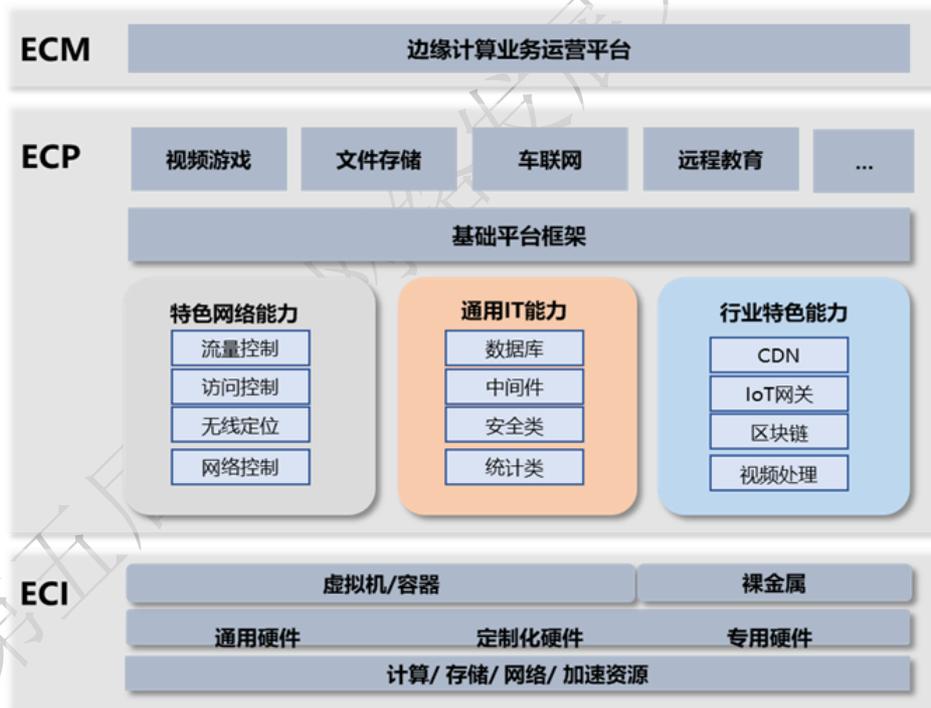


图 4-13 中国移动边缘计算能力产品架构

(4) 阿里巴巴

阿里云已经完成国内 30 多个省份 300+ 边缘计算节点的全域覆盖，接下来还将围绕边缘芯片/设备、边缘计算平台(操作系统)、城市边缘

中间件、城市边缘应用及服务这四个边缘技术栈进行布局。

目前阿里云已经与首汽约车开启了基于 5G 边缘计算的网约车移动业务合作试点项目，首汽约车把在车路协同过程中对路面交通状况的感知、传输、处理、响应等的整个通信交互过程将迁移至阿里云边缘节点（ENS）进行处理，通信响应速度降至毫秒级，有助于首汽约车更好地实现智能驾驶辅助、智能安全管理等，满足车路协同场景下从感知分析到反馈决策的实时化需求。

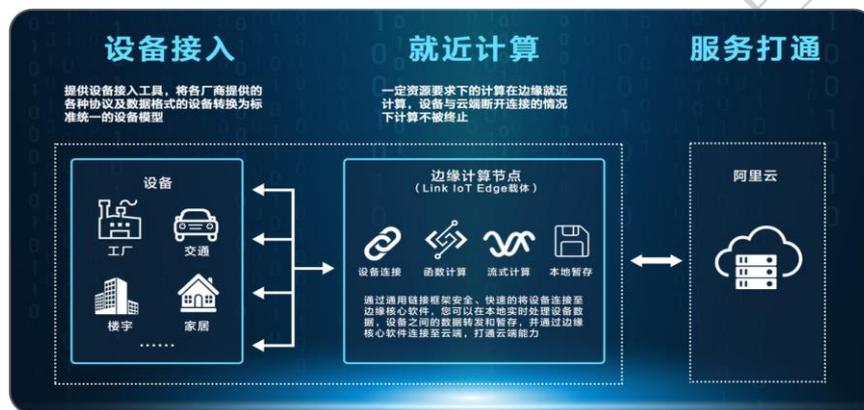


图 4-14 阿里云边缘计算场景

4.2.2 无服务器计算的产业发展

人类的进化每一次都伴随着生产效率的提升。同理，在整个 IT 计算的发展里程，也是逐步提高生产效率的里程，具体演进图如下所示：

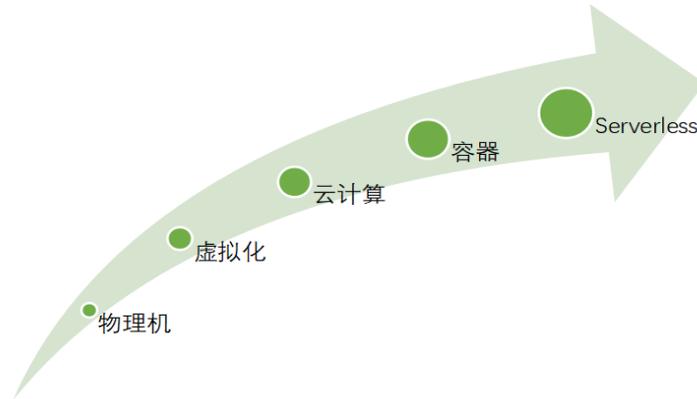


图 4-15 IT 计算演进

从大型物理机到通过虚拟化技术把物理机虚拟成单个的 VM 资源，从虚拟化集群到把集群搬到云计算上只做简单运维，再到把每一个 VM 按照运行空间最小化切分成更细的 Docker 容器，再从 Docker 容器变成不用管理任何运行环境的 Serverless 服务，即仅仅需要编写核心代码即可。技术变革都是把资源切分得更细致，让运行效率变得更高，让硬件软件维护变得更加简单。IT 技术架构的演进主要有以下几个特点：

- 硬件资源使用颗粒度变小；
- 资源利用率越来越高；
- 运维工作逐步减少。

Serverless 概念首次出现在 2012 年，由云基础设施服务提供商 Iron.io 的副总裁 Ken Fromm 在《Why the future of software and apps is serverless》一文中阐述，在 2014 年亚马逊发布了 AWS Lambda 之后，Serverless 开始变得流行了起来，基于这种架构，可以构建出很多应用场景，适用于各行各业，特别是轻计算、高弹性、无状态等场景。

随着 Serverless 概念的深入人心，各大云计算厂商纷纷推出了各自的 Serverless 产品，其中比较有代表性的有 AWS Lambda、Azure Function、Google Cloud Functions、阿里云函数计算等。此外，CNCF 在 2016 年就创立了 Serverless Working Group 工作组，致力于将云原生与 Serverless 技术结合，完善云原生生态。

Serverless 架构主要有以下特点：

- 实现了细粒度的计算资源分配；
- 不需要预先分配资源；
- 具备真正意义上的高度扩容和弹性；
- 按需使用，按需计费。

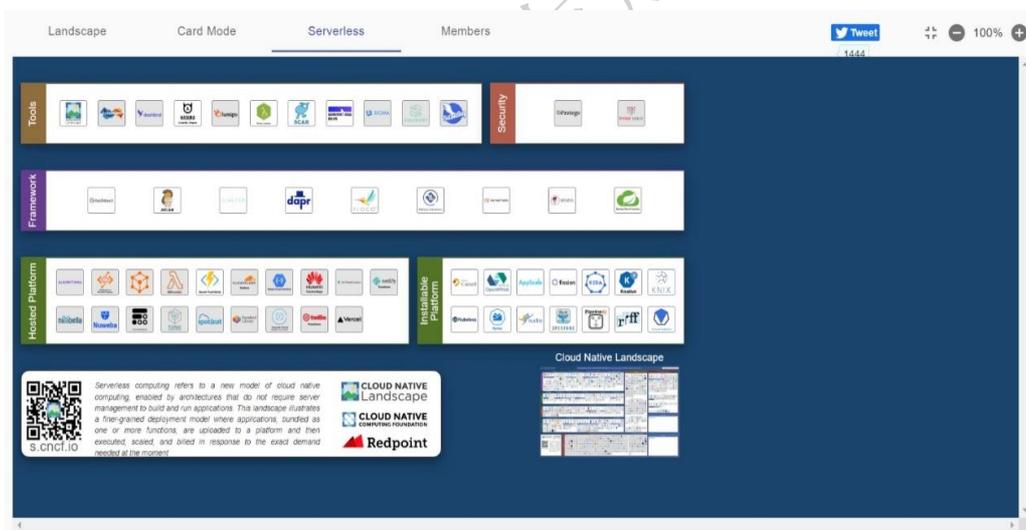


图 4-16 Serverless 生态

(1) 阿里

阿里云函数计算是事件驱动的全托管计算服务。通过函数计算，用户无需管理服务器等基础设施，只需编写代码并上传。函数计算会智能地准备好计算资源，以弹性、可靠的方式运行用户的代码，并提

供日志查询、性能监控、报警等功能，支持丰富的触发器类型和分类语言，并为开发者提供各类便捷的开发工具，广泛应用于 Web 应用、实时数据处理、AI 推理、视频转码等场景。

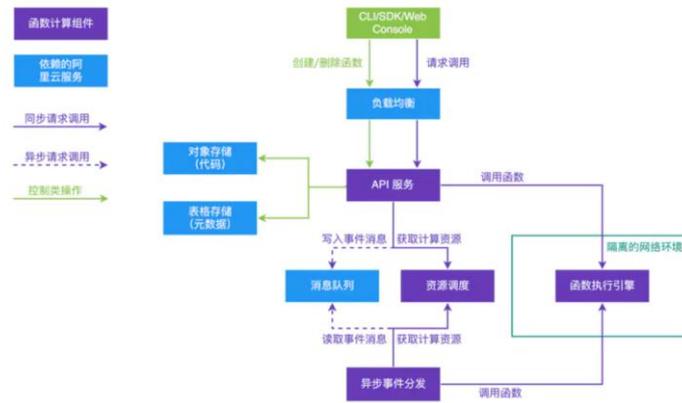


图 4-17 阿里云函数计算架构

(2) 谷歌

2018 年，谷歌在旧金山举行的 Google Cloud Next 会议上发布了 Google Cloud Functions。Google Cloud Functions 是 Google 的无服务器计算解决方案，可用于开发由事件驱动的应用，用户无需预配置、管理或升级服务器，Cloud Functions 可以根据负载自动扩缩，并集成了监控、日志记录和调试功能，基于最小权限原则的角色和函数级别的内置安全性，适用于混合云和多云方案的关键网络能力。应用场景丰富，包括无服务器应用程序后端，实时数据处理和智能应用程序，如虚拟助手，聊天机器人和情绪分析等。

(3) 华为

云容器实例（Cloud Container Instance）提供基于 Kubernetes 的 Serverless 容器服务，兼容 K8s 和 Docker 原生接口。采用业界领先的

Serverless Container 架构，用户无需感知的集群和服务器，即可便捷的部署容器应用，支持微内核粒度的自定义容器规格，秒级创建与计费，大幅降低用户的 IT 投入成本。服务支持 Kata Container 安全容器技术，可以提供虚拟机级别的安全隔离能力，兼容各类容器接口，降低业务云化门槛。

Serverless AI容器：为人工智能插上腾飞的翅膀

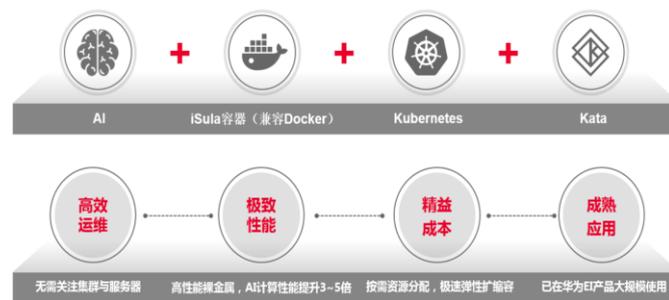


图 4-18 华为容器引擎

(4) 腾讯

云函数 (Serverless Cloud Function, SCF) 是腾讯云为企业和开发者们提供的无服务器执行环境，帮助用户在无需购买和管理服务器的情况下运行代码。使用者只需使用平台支持的语言编写核心代码并设置代码运行的条件，即可在腾讯云基础设施上弹性、安全地运行代码。SCF 是实时文件处理和数据处理等场景下理想的计算平台。支持各种应用场景，如文件处理及通知、数据 ETL 处理、移动与 Web 应用、AI 推理预测、小程序、消息转存、业务转流等。

(5) 微软

Azure Functions 是一项按需提供的云服务，可提供运行应用程序

所需的所有不断更新的基础结构和资源。你只需专注于对你最重要的代码，Azure Functions 处理其余代码。可使用 Azure Functions 来生成 Web API、响应数据库更改、处理 IoT 流和管理消息队列等。此外，业界非常流行的代码开发 IDE VSCode 支持快速部署 Azure Functions，提升研发人员的开发效率。

(6) 亚马逊

Amazon Lambda 是 Amazon 在 2014 年推出的无服务计算服务，可以自动运行代码，用户无需管理和维护基础设施，可以更加专注自己的业务。通过运行代码，可以响应各类事件，自动扩展应用功能，并可以根据工作负载，精确控制扩展。该服务按需使用付费，无需担心过度预支基础设施资源的额外费用。

借助 AWS Lambda，还可以通过为函数选择合适的内存大小来优化代码执行时间，任意规模都可以获得一致的超高性能体验。因其上手简单，迅速成为了 Amazon 的一项明星服务。



图 4-19 Amazon Lambda 服务流程

目前，Amazon Lambda 已经应用于许多领域，包括数据处理、实时文件处理、实时流数据处理、机器学习、Web 应用、后端/IoT 后端/移动后端等等。

4.2.3 无服务器边缘计算网络的产业发展

(1) 亚马逊

亚马逊作为全球最大的云计算公司，在 AmazonCloudFront 中提供 Lambda@Edge 无服务器边缘计算的功能，它的口号就是在靠近用户的地方运行代码。在宣传中，Kelkoo 作为法国的一家电子商务网站，为 22 个国家和地区的消费者服务，每天处理上亿条请求。为了降低处理图片的时延，他们使用 Lambda@Edge 来进行 URL 修改，在边缘节点中可以识别 URL，调整其中图片的尺寸和模式。

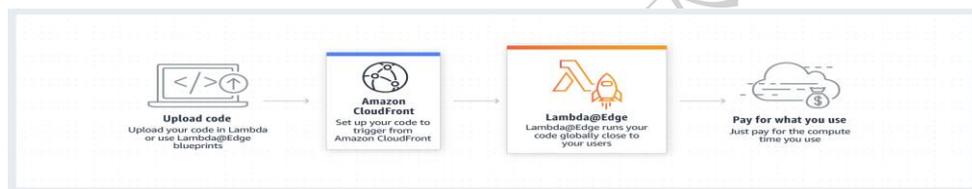


图 4-20 亚马逊 Lambda@Edge 工作原理

(2) Akamai

Akamai 是领先的内容交付网络 (CDN) 提供商,他的网络承载着全球 15%-30%的互联网 Web 流量。Akamai 推出了 EdgeWorkers 边缘无服务计算，使开发人员可以在全球部署的超过 25 万台边缘服务器中创建和部署微服务。当开发团队在边缘激活代码时，他们会将数据和处理逻辑推送到更靠近用户的位置。

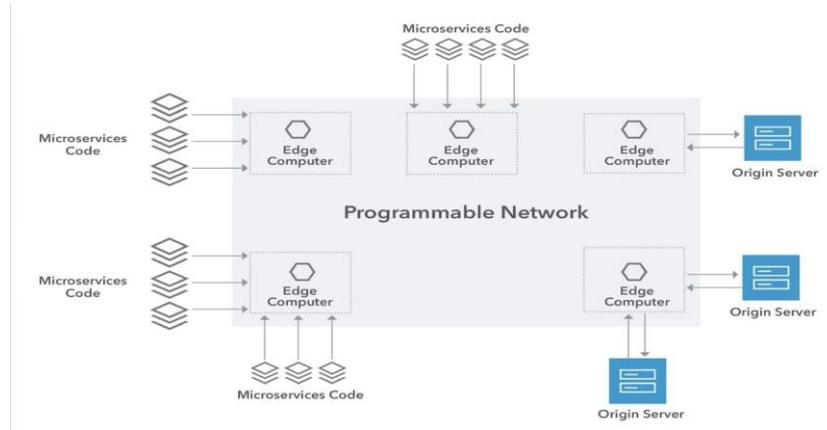


图 4-21 EdgeWorkers 架构

(3) Fastly

美国知名云计算服务供应商 Fastly 也推出了无服务器边缘运算服务 Compute@Edge，该服务提供一种灵活且可扩展的方法，让用户在网络边缘构建无服务器应用程序。在 Fastly 的无服务器边缘计算网络方案 Compute @Edge 中，允许用户把复杂的逻辑移动到边缘，可以为任何应用程序或后端服务部署和运行复杂的逻辑，同时 Compute @Edge 环境能以 35.4 微秒的速度启动，是市场上其他相似产品的 100 倍，从而为用户提供的安全、高性能和可扩展的无服务器计算方法。

(4) Cloudflare

Cloudflare 是一家美国的 CDN 加速服务独角兽企业。在他们的解决方案中，支持无服务器应用程序部署到网络边缘，从而大大减少延迟。他们的 Cloudflare Workers 是一个 FaaS 解决方案，它使用了 Cloudflare 的全球网络（由 194 台边缘服务器组成），可以在边缘缓存轻量化静态 HTML 页面，同时根据用户位置、设备类型或当时时间，整合动态内容。目前 Cloudflare Workers 已经被各行各业约

25,000,000 个互联网应用所使用,帮助他们节省了开发和服务器资源。

(5) 腾讯

腾讯云云函数目前支持 FaaS 功能,实现触发式运行;真正事件来临时,用户函数才会运行,用户代码运行时才有云函数代码的数据运算和费用计算。腾讯云函数目前在做的探索,从两方面出发。一是物联网方向,物联网主要是和设备打交道,实现设备上的边缘计算;从云函数本身的特点来讲,它属于触发型运算,真正数据产生之后才会拉起运算。云函数交由平台托管的调度,可以把云函数调度到用户设备上,二把云函数调度到 CDN 的节点上去,虽然 CDN 可以认为是云的一部分,但 CDN 本身已经很靠近用户,CDN 节点实际上已经在云的边缘。

4.3 标准化进展

国内外各个标准化组织都展开了边缘计算的标准化研究,或者在 5G、工业互联网、物联网等方向的研究中,加入了边缘计算的部分。

ETSI 组织定义了 MEC 的商业框架,包含软件架构、应用场景和 API 等。2014 年 ETSI 最初联合了 24 家公司,共同发起了 MEC 工作组,到 2017 年,工作组以 MEC 为主要场景,组织发布了 MEC 框架与参考架构、接口体系与功能模块、平台互通与部署要求等方面的标准,已经初步完成了 MEC 标准体系第一阶段工作。之后 ETSI 开始将移动边缘计算的概念扩展到多接入边缘计算,覆盖了包

括 3GPP 网络、本地网络以及外部网络的多种接入方式。目前 ETSI MEC 工作组增加到了 80 多个成员单位，同步发布了 12 个 PoC 应用，包含了物联网、AR/VR、远程医疗等场景。

2017 年 7 月，中国通信标准化协会（China Communications Standards Association，CCSA）成立工业互联网特设任务组（ST8），用于推进工业互联网标准体系建设，统筹规划和协调工业互联网国内外标准化工作。工作组于 2018 年开始，陆续发布了 10 余项标准项目，初步构建了服务于工业互联网的边缘计算标准体系。

ITU-T 国际电信联盟在 2019 年也立项了分布式云高层需求、分布式云功能架构和边缘云管理需求相关标准，中国移动、中国电信及国外运营商在其中也考虑了对 Serverless 的边缘侧服务拆解需求。

2020 年 5 月，中国信息通信研究院、中国电信、中国联通、华为、联想、浪潮、中兴、新华三、中国科学院沈阳自动化研究所等企业和科研机构在工业互联网产业联盟，共同发起“边缘计算标准件计划”，并召开线上启动会。这个计划针对边缘计算在实际部署应用过程中存在的产业碎片化、供给侧研发方向分散、需求侧建设选型困难、设备及平台标准缺失、可信开放测试机制不完善等突出问题，通过构建边缘计算标准设备（包括边缘控制器、边缘网关、边缘云）及边缘计算开放平台的技术要求及测试规范标准体系，探索边缘计算标准设备与开放平台的标准符合度评测，推动供给侧与需

求侧的精准对接，力争成为设备厂商产品研制及工业企业采购选型的风向标。

第五届未来网络发展大会白皮书

五、总结与展望

自动驾驶、全息通信、AR/VR 等计算密集型、时延敏感型业务正在快速发展，当前的网络面临着前所未有的技术挑战。无服务器边缘计算网络通过互联分布式的边缘计算节点，敏捷地分发计算任务，保证高效地利用分布式的计算资源，同时通过利用 Serverless 技术，提升边缘计算网络系统的运营效率，降低运维成本，促进网络新型业务的快速研发和迭代。无服务器边缘计算网络已成为未来网络的重要发展趋势，探索无服务器边缘计算网络对于全方位提升网络与服务能力具有重要意义，期待与业界一起研究探讨无服务器边缘计算网络架构、关键技术和产业应用。

第五届未来网络发展大会

附录 A：术语与缩略语

英文缩写	英文全拼	中文释义
AI	Artificial Intelligence	人工智能
AR	Augmented Reality	增强现实
CNCF	Cloud Native Computing Foundation	云原生计算基金会
CCSA	China Communications Standards Association	中国通信标准化协会
ETSI	European Telecommunications Standards Institute	欧洲电信标准协会
FlexE	Flexible Ethernet	灵活以太网
FaaS	Function-as-a-Service	函数即服务
IoT	The Internet of Things	物联网
IoV	Internet of Vehicles	车联网
MEC	Multi-Access Edge Computing	多接入边缘计算
SDN	Software Defined Network	软件定义网络
SCF	Serverless Cloud Function	云函数
TSN	Time-Sensitive Networking	时间敏感网络
VR	Virtual Reality	虚拟现实

参考文献

- [1] ETSI GS MEC 003-2020, Multi-access Edge Computing (MEC); Framework and Reference Architecture Disclaimer (V2.2.1)[S].
- [2] EdgeGallery, <https://www.edgegallery.org>.
- [3] Jonas E, Schleier-Smith J, Sreekanti V, et al. Cloud Programming Simplified: A Berkeley View on Serverless Computing[J]. 2019.
- [4] 黄韬,汪硕,黄玉栋等, 确定性网络研究综述[J]. 通信学报, 2019, v.40, No.386(06):164-180.
- [5] Knative, <https://github.com/knative/>.
- [6] OpenFaaS, <https://github.com/openfaas/faas>.
- [7] OpenWhisk, <https://github.com/apache/openwhisk>.
- [8] 中国联通研究院, 算力网络架构与技术体系白皮书, 2020.10.

第五届未来网络发展大会白皮书