

边缘计算与雾计算

5.1 雾计算基本概念

雾计算最早由思科公司提出,也可称为薄雾计算、雾化或微云。尽管这些不同术语之间存在一些细微的差异,但在更高的层次上,这些术语可以视为同义词。雾计算是一种分布式计算环境,它扩展了云计算范例,旨在将数据的智能、处理和存储推向更靠近网络边缘的位置,从而更迅速地提供与计算机相关的服务,并更靠近构成物联网部分的互连智能事物。因此,雾计算可定义为一种水平的系统级体系结构,将计算、存储、控制和网络功能分布在整个“云到事物”连续体上,使其更靠近用户。雾计算的产生打破了传统云计算的许多局限性,通过减小网络服务延迟,增加上下文感知,从而节省网络带宽,增强操作的效率,在提高服务质量的同时改善用户体验并加快实时处理及在数据生成位置附近存储敏感数据的速度。雾计算还为移动性和移动通信的使用提供了更好的支持。以响应时间为例,在云计算中,响应时间以分钟为单位,通常为数天,而数据存储时间更长,通常是永久性的,并且位置覆盖范围广且通常是全球范围;在雾计算中,响应时间通常以毫秒为单位,数据

存储时间段是瞬态的,位置覆盖范围是本地,并且具有更高的访问频率。目前,雾计算已成功部署在许多需要雾本地化和云全球化的商业和工业应用场景中,包括智能电网、智能家居和城市、联网汽车、自动驾驶汽车和火车、交通信号灯系统、医疗保健管理和许多其他的网络物理系统。

如图 5.1 所示,雾计算的特点如下。

(1) 低延迟: 由于雾节点与本地终端设备距离较近,因此可以在更快的时间内进行响应和分析。

(2) 多租户: 受控环境中的多租户(由于高度虚拟化的分布式平台)。

(3) 异构终端用户支持: 丰富多样的最终用户支持(由于边缘设备靠近计算节点)。

(4) 支持移动性: 因为雾计算应用程序与移动设备之间的通信更加直接。

(5) 实时交互: 与批处理相反,例如在基于云的应用程序中。

(6) 情境感知: 环境中的设备和节点对环境有相关认知和了解。

(7) 地理广泛分布: 由于雾环境在地理上分布,因此它在交付高质量流服务中起着积极的作用。

(8) 无线访问网络: 更适合需要时间分布分析和通信的无线传感设备。

(9) 支持雾节点异构性: 雾节点采用不同的外形尺寸,并部署在各种分布式环境中。

(10) 无缝互操作性和联盟性: 可以使来自不同供应商的设备和跨不同域的设备之间的通信更好。

(11) 实时分析: 由于源附近数据的摄取和处理很容易实现。

(12) 支持工业应用: 通过实时处理和分析,支持各种工业应用。

雾计算架构的核心特征是,它可以更迅速地提供计算和数据分析服务,并且更靠近生成此类数据的物理设备,即在网络边缘,从而绕开了更广阔的互联网。目前物联网技术的飞速发展,使人工智能、虚拟现实、触觉互联网和 5G 应用的数字创新正在改变我们的工作、通勤、购物和娱乐方式。这些新兴的移动终端及应用数据预

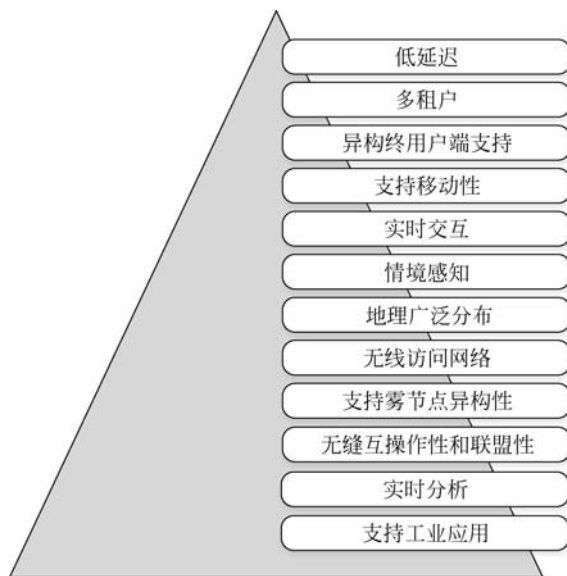


图 5.1 雾计算特点

计将从 2016 年的每年 89 艾字节增长到 2025 年的每年 175 泽字节。当前的“仅云”体系结构无法跟上整个网络中数据的数量和速度,从而降低了可以从这些投资中创造和获取的价值。雾计算提供了云到事物连续体中缺少的链接。雾架构选择性地将计算、存储、通信、控制和决策移至生成数据的网络边缘,以解决当前基础架构的局限性,以实现关键任务,以及数据密集型用例。雾计算是对传统的基于云的计算模型的扩展,该架构的实现可以驻留在网络拓扑的多个层中,但是,应通过雾的这些扩展来保留云的所有好处,包括容器化、虚拟化、编排、可管理性和效率。在很多情况下,雾计算可与云一起使用。

5.2 雾计算参考架构

本节将讨论雾计算范式的概念、原理及相关的范式和技术,介绍云和雾架构之

间的区别,并简要讨论 OpenFog 参考架构。希望本节为本书其余部分介绍的各种与雾有关的主题奠定一个基础。

5.2.1 OpenFog 架构

OpenFog 联盟由 ARM、思科、戴尔、英特尔、微软公司和普林斯顿大学于 2015 年 11 月成立。该联盟通过吸纳雾计算领域领先的技术参与者,试图定义一个开放、可互操作的雾计算架构,其工作重点是为高效、可靠的网络 and 智能终端创建框架,并结合基于开放标准技术的云、端点和服务之间的可识别、安全和隐私建立友好的信息流。根据 OpenFog 联盟的定义,雾计算是一种水平的系统级体系结构,可将计算、存储、控制和网络功能分布在整个“云到事物”连续体上,使其更靠近用户。雾节点是雾体系结构的基本元素,它可以是任何提供雾架构的计算、网络、存储和加速元素的设备,例如交换机、路由器、工业控制器及智能物联网节点等。因此,OpenFog 架构下的雾计算被认为是一个可在终端设备和传统的云计算数据中心之间提供计算、存储和网络服务的高度虚拟化的平台。

2017 年 2 月,OpenFog 联盟发布了 OpenFog 参考架构(Reference Architecture, RA),旨在支持物联网、5G、人工智能等数据密集型需求而形成的通用技术框架。OpenFog 参考架构的关键是该架构所基于的 8 个核心技术原则,也称为 8 个支柱,分别是安全性、可扩展性、开放性、自主性、RAS、敏捷性、层次性架构和可编程性,其中 RAS 是指可靠性(Reliability)、可用性(Availability)、可维护性(Serviceability),如图 5.2 所示。

OpenFog 参考架构包括多个水平层次、透视面和垂直视角。水平层次包括硬件平台基础设施层、协议抽象层、传感器 & 执行器 & 控制器、节点管理与软件背板层、应用支持层、应用服务层等。垂直视角包括性能、安全、管理、数据分析与控制、IT 业务与跨雾应用,如图 5.3 所示。

安全性	可扩展性	开放性	自主性	RAS	敏捷性	层次性架构	可编程性
<ul style="list-style-type: none"> • 可信 • 证书 • 隐私 	<ul style="list-style-type: none"> • 性能 • 容量 • 可靠性 • 安全 • 硬件 • 软件 	<ul style="list-style-type: none"> • 可组合 • 互操作 • 通信 • 位置透明 	<ul style="list-style-type: none"> • 探索 • 编排 • 管理 • 操作 • 节约成本 	<ul style="list-style-type: none"> • 可靠性 • 可用性 • 可维护性 	<ul style="list-style-type: none"> • 决策制定 • 数据分析 • 数据管理 	<ul style="list-style-type: none"> • 向上支持 • 各层自治 	<ul style="list-style-type: none"> • 硬件可编程 • 虚拟化 • 多用户 • 应用流动性

图 5.2 OpenFog 参考架构的八大支柱



图 5.3 OpenFog 参考架构

OpenFog 架构下的雾计算模型分为 5 个层次,从下到上依次是端点层、网关层、访问层、核心网络层、互联网层/云服务资源层,具体架构如图 5.4 所示。在 OpenFog 架构下的雾计算模型雾节点可以链接形成网格,将资源和数据源与驻留在南北边缘设备(云到传感器)、东西边缘设备(功能到功能或点对点)的层次结构结合在一起,以提供负载平衡、弹性、容错、数据共享和最小化云通信,从而获得最大效率。

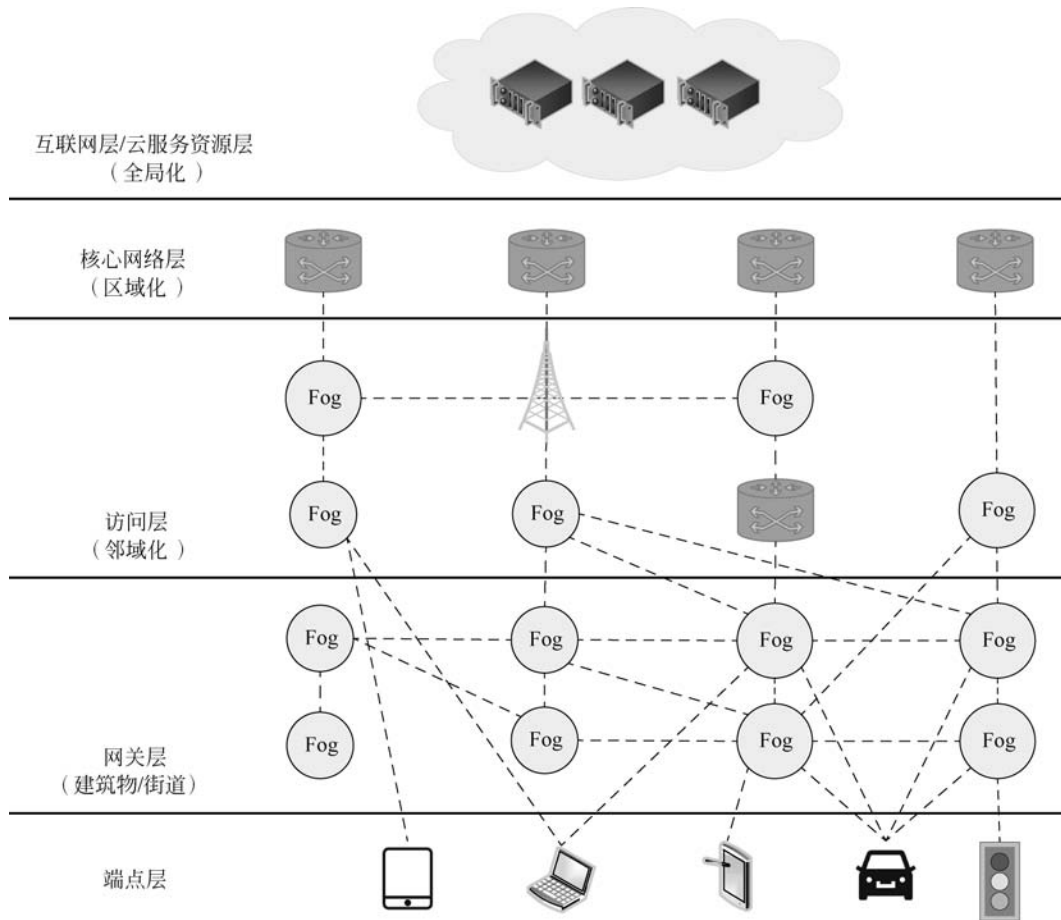


图 5.4 OpenFog 雾计算模型

OpenFog 架构定义了雾和云之间的接口,以及雾和雾之间的接口,优点如下。

- (1) 认知: 以客户端为中心目标,具有自主性。
- (2) 效率: 在最终用户设备上,动态合并本地资源。
- (3) 敏捷: 快速创新和基于通用构架快速拓展。
- (4) 延时: 实时处理和物理信息系统控制。

5.2.2 OpenFog 架构实例

OpenFog 描述了一种通用雾平台,该平台旨在适用于任何垂直市场或应用程序。此架构可应用于许多不同的市场,包括但不限于交通、农业、智能城市、智能建筑、医疗保健、酒店、金融服务等,为需要实时决策、低延迟、增强的安全性且受网络限制的物联网应用创造业务价值。本节将重点介绍用于智能汽车和交通控制的雾计算,并说明 OpenFog RA 如何满足要求。图 5.5 概述了 OpenFog RA 的智能高速公路应用。

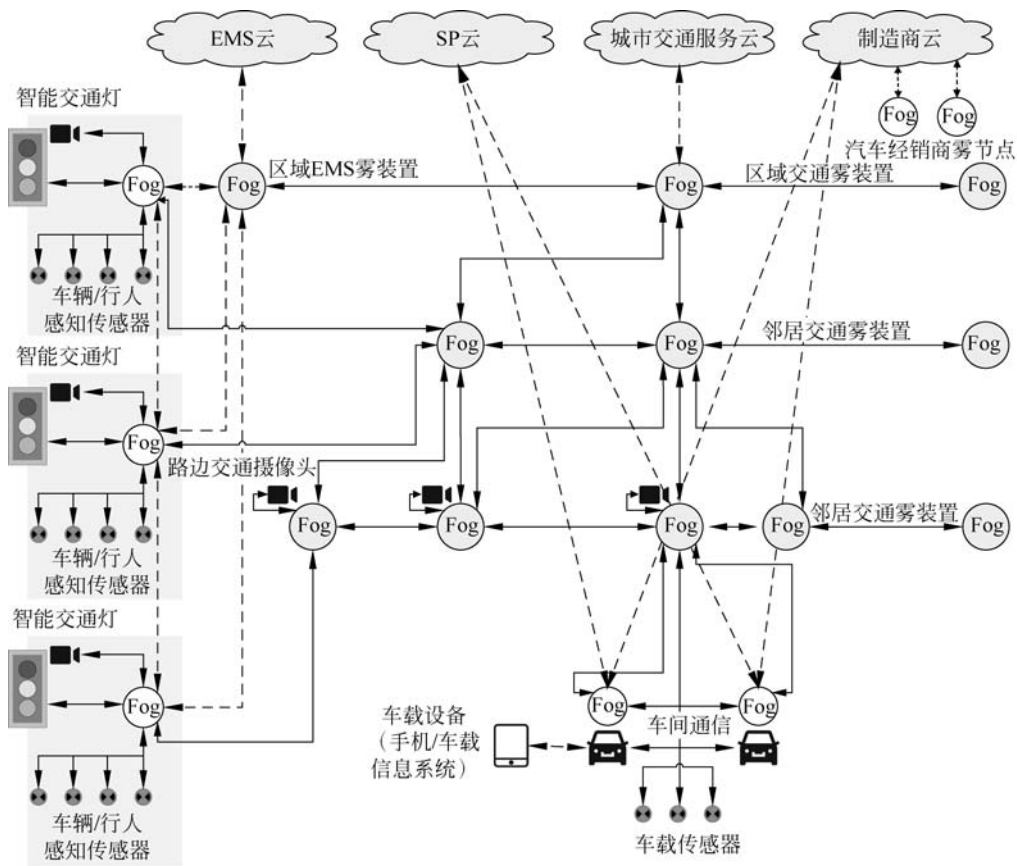


图 5.5 OpenFog 交通：智能汽车和交通控制系统

5.3 边缘计算与雾计算的差异

雾计算旨在将数据的智能、处理和存储推向更靠近网络边缘的位置,从而更迅速地提供与计算机相关的服务,并更靠近构成物联网部分的互连智能事物。雾计算与边缘计算从概念上很相似,因此,有时也会被称为边缘计算,这两个术语经常互换使用。虽然边缘计算与雾计算在数据处理方面都将其转移到了产生数据的源头,通过任务的下沉减少云端的压力,从而降低延迟,但两者在结构、数据处理、数据通信等方面仍存在差异。从时间角度,边缘计算概念在出现时间上早于雾计算,指代云和设备的边界雾计算,因为和云相比位置上更接近设备,所以表示为雾;从体系结构角度,雾可以与云一起使用,而边缘则是通过排除云来定义的。雾是分层的,边缘倾向于限制为少量的层。除计算外,雾还解决网络、存储、控制和加速问题。

1. 体系结构方面的差异

尽管雾计算和边缘计算都指在网络边缘,即更靠近数据源的位置提供智能处理及存储功能,但两者在网络架构中的位置存在差异。雾计算处于网络架构的局域网级别,因此,处理数据通过雾节点或 IoT 网关;而边缘计算是将智能处理和存储能力直接置于智能设备中,例如可编程自动化控制器(Programmable Automation Controllers,PAC)。

边缘计算通常是指服务实例化的位置,而雾计算则表示在终端用户端控制下靠近设备和系统上的通信、计算、存储资源和服务等的分布。

2. 数据处理方面的差异

在雾计算中,只有一个集中式设备,负责处理来自网络中不同端点的数据;而在边缘计算架构中,物联网数据由连接的设备直接收集和分析,各自设备独立地对数据的存储位置进行判断,因此每个网络节点都参与处理。

3. 数据通信方面的差异

在雾计算中,数据生成设备与云环境之间的数据通信需要许多步骤:首先将通信定向到可编程控制器的输入/输出点运行控制系统程序,以执行 PAC 自动化;然后将数据发送到协议网关,该协议网关将数据转换为易于理解的数据格式,例如 HTTP;最后将数据传输到本地网络的雾节点,该节点执行所需的分析处理工作,之后将结果组织回传到云端进行存储。

边缘计算中的通信较雾计算更为简单。数据生成设备物理连接到 PAC,以实现板载自动化及数据的并行处理和分析,因此,在边缘计算中,PAC 用于决定哪些数据存储在本地或发送到云端。这种通信方式,除了可减少故障的可能性外,还节省了时间并简化了通信,降低了体系结构的复杂性。针对目前的市场环境,服务提供商和数据处理公司更倾向于使用雾计算,而拥有骨干网和无线网络的中间件公司则更倾向于使用边缘计算架构。

边缘计算与雾计算之间的优势比较如表 5.1 所示,主要包括位置及延迟、计算能力、性能开销三方面的对比分析。

表 5.1 边缘计算与雾计算优势比较

边缘计算	雾计算
所有节点参与以降低延迟	位置感知、低延迟、QoS 保障
实时本地分析	具有一定的数据处理能力且更贴近用户
降低开销并且避免网络拥塞,提高性能	分布式数据与云服务的集成

5.4 本章小结

本章首先从基本概念出发介绍了什么是雾计算,并在此基础上介绍了典型的基础架构 OpenFog 及该架构下的应用实例,最后分别从体系结构、数据处理及数据通信三方面对边缘计算与雾计算之间存在的差异进行了分析比较。