

雾计算的概念、相关研究与应用

贾维嘉, 周小杰

(上海交通大学电子信息与电气工程学院, 上海 200240)

摘 要: 首先, 系统地分析和总结了雾计算的研究现状, 重点介绍了雾计算出现的背景及其相对于云计算的优势, 对雾计算及其他相似的计算模式进行比较, 指出了雾计算相比于传统计算模式的优点。然后, 总结了雾计算的体系结构与各层功能。同时, 对于雾计算在网络管理和资源调度这 2 个方面的研究问题展开讨论, 总结了前人提出的解决方法并指出了现有方法的不足。最后, 对于雾计算的一些相关应用进行了阐述, 并以智能驾驶、工业物联网这 2 个示范应用为例指出了当前雾计算在实际应用上仍需攻关的重要课题。

关键词: 雾计算; 时延; 地理分布; 移动性

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018086

Concepts, issues, and applications of fog computing

JIA Weijia, ZHOU Xiaojie

School of Electronic, Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

Abstract: Firstly, the previous work of fog computing was systematically analyzed and summarized. The background of fog computing and the comparison with cloud computing were introduced. Besides, based on the comparison with other computing style, the advantages and characteristics of fog computing were explained. In addition, the architecture of fog computing was described. Moreover, network management and resource scheduling of fog computing were discussed, where the related previous work were summarized and analyzed. At last, the applications of fog computing were described. Taking the intelligent driving and industrial Internet of things applications as examples, the key research issues of fog computing were proposed.

Key words: fog computing, delay, geo-distributed, mobility

1 引言

随着人们对计算和存储资源需求的不断提升, 个人计算机在很大程度上已经不能满足人们的需求。对此, 云计算带来的按需付费 (pay as you go) 的模式使人们可以用一种相对低廉的价格获取所需的计算和存储资源^[1]。然而云计算需要将数据发

往密集的云中心进行处理, 在实际应用场景上不可以提供移动性的支持, 这对车联网、无线接入网等移动场景下的网络来说是一种限制^[2]。与此同时, 由数据的长距离传输所带来的不可接受的时延也使云计算不适用于时延敏感的应用^[2]。更进一步地, 长距离的传输也使数据被攻击的可能性提升, 从而影响数据的安全性^[3]。

收稿日期: 2017-11-08; 修回日期: 2018-03-29

通信作者: 贾维嘉, jiawj@sjtu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61532013); 国家重点基础研究发展计划 (“973” 计划) 基金资助项目 (No.2015CB352401); 上海市科研计划基金资助项目 (No.15JC1402400); 上海交通大学 985 基金资助项目 (No.WF220103001)

Foundation Items: The National Natural Science Foundation of China (No.61532013), The National Basic Research Program of China (973 Program) (No.2015CB352401), Shanghai Scientific Innovation Act of STCSM (No.15JC1402400), 985 Foundation Project of Shanghai Jiaotong University (No.WF220103001)

对此,需要在本地部署带有计算和存储功能的设备以缩短终端用户到计算和存储资源的距离。因此,雾计算的概念应运而生。通过在远程的云和设备之间引入中间雾层,以部署本地化的计算和存储资源^[2]。雾层不仅可以自己向用户直接提供服务,还可以利用云层更为强大的计算和存储能力协同进行服务^[4]。相较于传统的云计算模型,这种方法无疑更加适合车联网^[5]、无线接入网^[6]等移动场景和时延敏感的应用^[7]。对此,CISCO[®]提出了雾计算的设想^[8],HP[®]实验室在其之上更进一步地提出了雾计算的概念^[9]。不难看出,这样的想法与过往的如移动计算(mobile computing)^[10]、移动云计算(mobile cloud computing)^[11]、边缘计算(edge computing)^[12]等计算模式存在着一定的相似性。因此,本文对从云到雾的过程进行相应阐述,并将雾计算与这些相似的计算模式进行了比较。

雾计算作为一个新兴的研究热点,其各部分的功能受到了广泛的研究。其关键主要在于网络管理与雾层资源调度上。在网络管理上需要完成终端设备发现、任务数据转发的功能;而在资源调度上,考虑的是如何安排任务取得尽可能高的收益。对于这2个关键问题的阐述也是本文的重点所在。

由于雾计算相较于云计算适用于移动场景和时延敏感的应用,因此,以智能交通为代表的需要移动性支持的应用和以大数据无线传感网络为代表的对于时延敏感,又需要相对较大处理能力的应用适合通过雾计算的方式进行处理^[13]。对此,前人提出了不少雾计算在这些应用上的应用实例和实现平台。

2 雾计算

2.1 雾计算的由来

雾计算的概念最早由 CISCO[®]提出,其在云和终端设备之间引入中间雾层部署运算、存储等设备。中间雾层主要面向一个较小的区域提供计算、存储、网络传输服务^[2]。对此,CISCO[®]提出了 IOx 框架,使用户可以在上面开发部署应用^[8]。在此基础上,HP[®]实验室对于雾计算给出了更为明确的定义:“雾计算是由大量异构普遍存在的无线(有时为无线自组织)分布式设备通过网络相互协作在无第三方干预的情况下共同完成计算和存储任务的场景。这些任务可能是关于网络等基本功能的,也可能是关于具体应用的。用户可以通过有偿租用这

些分布式设备来获取这样的服务^[9]”。这个定义中不仅包含了 CISCO[®]先前所提出的雾计算的计算、存储、网络传输服务这3个功能,还包括了雾计算相较于云计算的分布式,通过无线网络进行协作、自组织等特点。

由此可见,雾计算的概念由云计算引申而来。这2种计算模式间的主要差别在于与终端用户的接近程度、能否支持地理分布与能否支持边缘的数据分析和挖掘这3个方面^[14]。此外,众多学者还在服务质量、访问控制、时延功耗等方面对于雾计算和云计算进行了比较^[6~7,9,15~18]。表1对雾计算和云计算在通信访问方式、控制管理、服务质量、采用设备以及目标用户这5个方面的差别进行了总结。

根据比较的结果不难发现,雾计算这一模式相当于把云拉到地上,在本地部署带有计算和存储功能的设备,以缩短终端用户到计算存储资源的距离。这样部署的本地化设备虽然可以降低时延、提供移动性的支持,但也受限于自身的计算和存储能力。雾计算并不是云计算的替代品,而是通过这些本地化的设备不仅可以自己向用户直接提供服务,还可以利用云层更为强大的计算和存储能力协同进行服务^[4]。除云计算外,移动计算、移动云计算、边缘计算等概念与雾计算有一定的相似之处。

2.2 雾计算的相似概念

2.2.1 移动计算与移动云计算

移动云计算的前身是移动计算。移动计算顾名思义就是在手机、便携式电脑等移动设备上进行计算存储的模式^[10]。但这种计算模式下任务主要在移动终端设备中进行计算和存储,会遇到资源稀缺性、电源不足和连接不稳定等问题,不足以应付图像处理、自然语言处理等计算任务^[19],因而提出移动云计算,将计算任务由移动端发送到云端,再由云端将结果传回移动端。相较于传统的云计算,其对移动性提供了支持^[20],但没有改变云计算需要将计算任务传送到云端所带来的时延和对于网络所带来的巨大负载^[21]。与此同时,设备上传任务到云端所带来的发送功耗和资费成本有时也是不可接受的^[10]。对此,雾计算在本地化引入计算存储设备,通过在本地形成小的数据中心^[22],可以在不失去移动性支持、保证一定程度的处理能力下,很好地解决上面提到的时延、功耗和资费等问题^[23]。

表 1 雾计算和云计算的比较

| 方面 | 具体指标 | 云计算 | 雾计算 |
|--------|--------------------|---------------|------------------|
| 通信访问方式 | 节点的访问方式 | 通过互联网（主要是有线网） | 通过本地网络设施（主要是无线网） |
| | 服务的访问方式 | 通过核心网 | 通过边缘设备 |
| | 终端用户到节点距离 | 多跳 | 一跳 |
| | 对于移动前传和集中式的基带单元的负担 | 重 | 轻 |
| | 缓存与无线信号处理 | 集中式 | 兼具集中式与分布式 |
| | 无线资源管理 | 集中式 | 兼具集中式与分布式 |
| 控制管理 | 控制方式 | 完全控制 | 部分控制 |
| | 软件虚拟的基础设施 | 数据中心服务器 | 边缘设备 |
| | 地理分布 | 否 | 是 |
| | 控制端数目 | 少 | 多 |
| 服务质量 | 时延 | 较高（分钟级~月级） | 较低（毫秒级~秒级） |
| | 时延抖动 | 高 | 很低 |
| | 可用性 | 99.99% | 不定 |
| | 数据传输过程受攻击可能性 | 可能性高 | 可能性很低 |
| | 安全策略 | 难以定义 | 易被定义 |
| | 移动性支持 | 有限支持 | 支持程度高 |
| | 服务来源 | 全球 | 本地 |
| | 使用成本 | 高 | 低 |
| | 网络要求 | 高 | 低 |
| | 对实时应用的支持 | 差 | 好 |
| 采用设备 | 任务的传输功耗 | 大 | 小 |
| | 终端用户数 | 数十万或数百万 | 数十万 |
| | 节点数目 | 少 | 多 |
| | 硬件设备平均成本/美元 | 1 500~3 000 | 50~200 |
| | 设备部署位置 | 远端数据中心 | 靠近网络边缘 |
| | 硬件 | 存储容量大、计算能力强 | 存储容量与计算能力有限 |
| | 设备部署环境 | 带制冷设备的大型仓库 | 小型仓库或室外 |
| | 服务器拥有者与管理者 | 大公司（如 Google） | 小公司或个人 |
| 目标用户 | 部署速度（代价） | 慢 | 快 |
| | 内容产生者 | 主要是人 | 主要是传感器设备 |
| | 内容丰富程度 | 丰富 | 单一 |
| | 目标用户 | 互联网用户 | 移动用户 |

2.2.2 边缘计算

边缘计算的初衷也是由于大部分的任务在网络的边缘产生，将其传输到远端的云数据中心时延过大，因此考虑在数据的边缘外加计算和存储设备，接受任务并与云数据中心进行协同处理^[12]。因而其与雾计算有很大的相似性^[15]。其中，边缘计算中的边缘是相对于用户层来说的概念，而雾计算的雾是相对于云层的概念。边缘计算更加关注用户，

而雾计算更加关注雾层服务提供商、雾层中各设备的拥有者与管理者^[24]。

2.2.3 分布式云数据中心与迷你数据中心

顾名思义，分布式云数据中心（distributed cloud data center）、迷你数据中心（mini datacenter）就是将服务器分布在各地，建立小的数据中心^[25]。相较于将用于计算和存储的服务器分布在世界各地，雾计算可以缓解传统云计算将服务器集中部署带来的传输距

离大、容易遭受分布式拒绝服务攻击 (DDoS, distributed denial of service) 等问题。与此同时, 也更能因地制宜地提高能源使用效率。但这样的数据中心和雾计算在部署的密度上通常有着不小的差距。分布式云数据中心之间的距离一般很大, 往往相距 10^6 m 以上, 仍然不能有效降低时延^[26], 而雾计算的服务器之间相距很小, 能将时延控制在可控范围内。

2.2.4 微云

微云 (cloudlet) 和雾计算也有很大的相似性, 它也是在靠近用户的位置分布式地部署计算设备, 为用户时延敏感、带宽有限的应用提供服务。其一开始基于移动计算所提出, 因此, 同样具备对移动性的支持^[27]。但它与雾计算主要有 2 个不同之处: 1) 微云主要针对的是计算任务, 而雾计算除计算外还可通过智能网关 (smart gateway)、路由器等设备支持存储、路由等服务^[16]; 2) 微云基于移动计算所提出, 主要针对移动设备, 但是雾计算除了移动设备外, 对于固定的设备也适用^[15]。

2.2.5 内容分发网络

内容分发网络 (CDN, content delivery network) 通过分布式的缓存服务器 (cache server) 进行信息存储, 以减少用户对于信息获取的时间。相较于雾计算, 内容分发网络的通用性不够强 (一般只缓存特定的信息), 而且计算能力也比较弱^[16]。

综上所述, 从这些比较中不难发现, 雾计算在降低时延、移动性支持和信息安全上相比于传统的计算模式有着不小的优势。雾计算相当于在移动云计算的基础上在靠近设备终端的位置上引入了由密集的服务器构成的中间雾层。

3 雾计算的系统架构

3.1 雾计算的系统架构介绍

相较于云计算终端用户层、网络层和云层的 3 层架构, 雾计算的系统在引入中间雾层后架构可以分成 5 层, 分别是终端用户层 (end user layer)、

接入网络层 (access network layer)、雾层 (fog layer)、核心网络层 (core network layer) 和云层 (cloud layer), 如图 1 所示。不难看出, 离底层越近, 分布的区域越大, 且终端用户数据传输到该层的时延越小^[14]。表 2 给出了这 5 层的主要设备和主要功能。

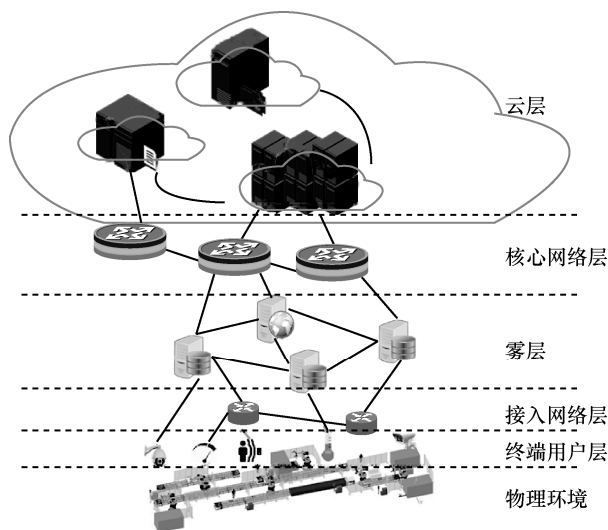


图 1 雾计算的系统架构

3.1.1 终端用户层

终端用户层主要由用户的手机、便携式电脑等终端设备构成^[14], 而且随着传感器网络技术的发展, 传感器节点也将在这一层中发挥重要作用^[28]。这些设备既有可能是安放在某处的固定设备, 如放在交通灯上、道路两旁的传感器^[29], 也有可能是移动的终端, 如用户的手机、便携式电脑^[14]。在这一层中, 这些设备将起到内容的生产者 (content producer) 和内容的消费者 (content consumer) 的作用。任务将在这一层中产生, 处理后的结果也将返回到这一层。此外, 终端设备还需要发现并指定对应任务转发的雾节点^[23]。

3.1.2 接入网络层

当终端用户层的用户生成内容后, 会由接入网络

表 2

雾计算各层主要设备与功能

| 层次 | 主要设备 | 主要功能 |
|-------|------------------------|--------------------------|
| 终端用户层 | 用户的手机、便携式电脑等终端设备与传感器节点 | 内容的生产者和内容的消费者 |
| 接入网络层 | 无线网络设备为主, 有线网络设备为辅 | 按照预定的规则将终端用户的任务发送到对应的雾节点 |
| 雾层 | 雾边缘节点、微雾、雾服务器 | 提供一定程度的计算、存储和通信 |
| 核心网络层 | 核心网络设备 | 将超出雾层计算或存储能力的任务发送到云数据中心 |
| 云层 | 云数据中心服务器 | 数据备份、大计算量任务处理 |

的网络设备将这些信息按照预定的规则发送到对应的雾节点上。由于底层设备既包含有线设备,又包含无线设备,因此在这一层中通信的网络既包含有线局域网,又包含 Wi-Fi、5G 等无线接入网络^[6]。

3.1.3 雾层

这是雾计算的核心。在雾层中部署有贴近用户的、高密度的计算和存储设备^[30],极大地降低了传统云计算的时延,同时也能给予用户移动性的支持。这些设备被称为雾节点(fog node)。可以将雾节点按照其部署位置和功能分成3类:雾边缘节点(fog edge node)、微雾(foglet)和雾服务器(fog server)^[17]。雾边缘节点一般由智能网关、边界路由器等构成^[31],是距离终端用户层最近的雾节点,提供一定程度的计算、存储和通信功能。对于一些简单的任务,可以在上面直接进行处理并返回结果,但对于较为复杂的任务,这些节点的处理能力往往不能满足要求,只能起到数据预处理、预分析等作用,然后将处理后的结果发往雾服务器进行进一步的处理。而雾服务器相比于雾边缘节点具有较强的计算能力和较大的存储空间,能够应对较多的请求。其可直接与远端的云数据中心通过网络进行连接,因而对于超出雾服务器处理能力的请求将会被发往云数据中心。对于相对复杂的雾层结构来说,在雾边缘节点和雾服务器之间可能还有其他的雾节点,这些节点既不直接与底层的终端设备相连,也不直接通过网络与远端云数据中心相连,而是在雾边缘节点和雾服务器之间起着预处理、路由等中间件(middleware)的作用,这些节点被称为微雾,通过 SDN (software defined network) 可以实现流量的智能转发,避免拥塞。对此,文献[32]设计了基于 SDN 的雾节点通信方法,并用 MQTT 作为其通信协议。与此同时,对于一些相对简单的雾层结构,可能存在有些节点既直接与底层的终端设备相连,也直接通过网络与远端云数据中心相连,那么这些节点将同时起到雾服务器和雾边缘节点的功能。

对于雾节点来说,其需要提供存储计算等服务,其内部可以分成底层硬件和上层服务2层^[14],这2层通过 API 相连。底层硬件又可以细分为硬件与虚拟层(hardware & virtualization layer)和硬件资源分配与监控层(hardware allocation & monitor layer)。在硬件与虚拟层中包括内存、CPU 等实体和虚拟的硬件设备;而在硬件资源分配与监控层中

则是对实体和虚拟的资源进行监控,如进行系统监控、网络管理、负载均衡等。上层服务同样可以分成服务层(service layer)和应用层(application layer)2层,服务层可以直接通过 API 与底层硬件进行交互,提供加密验证、计价等服务;而应用层基于服务层,内有用户定义的应用程序。

3.1.4 核心网络层与云层

对于超出雾层计算或存储能力的任务,会被雾服务器通过 IP 核心网络发往云数据中心。相比于接入网络,这里一般是多跳的有线网。同样地,SDN 技术也将在网络资源管理、流量高效转发上发挥巨大的作用^[16]。

而云层主要由远端的云数据中心服务器构成,这些服务器往往具备雾层服务器更强的计算能力和更大的存储能力,因此,其一般起到数据备份、大计算量任务处理的作用。与雾层相类似,云端的服务器也是相连的,装载有任务的虚拟机可在云服务器间来回迁移,以增大执行效率^[23]。

3.2 雾计算系统架构的设计与实现

在具体实现上,CISCO®提出了 IOx 框架,使用户可以在上面开发部署应用^[8]。而对于一些小型的基于雾的应用,文献[33]用树莓派实现了一个小型的通用雾计算平台。此外,针对一些类似于智慧城市、智能交通等的特定的应用场景,现有工作也设计了一些面向专门应用场景的雾计算平台^[34]。此类平台将在第5节详细叙述。而对于虚拟仿真平台方面,文献[35]提出了一个名为 iFogSim 的仿真平台。该仿真平台通过 Java 语言进行实现。在该仿真平台上可以对终端设备、云计算中心与网络链路进行配置,并根据自己的需要实现不同场景下的雾计算资源调度与管理算法,并可通过时延、功耗、网络资源占用、控制成本等指标对设计的算法进行性能的衡量。虽然这给调度算法研究者带来了福音,但是其功能仍比较有限,仿真效果与实际效果仍存在比较大的差别。

4 雾计算的研究问题

相比于云计算这样的计算模式,雾计算中引入了中间雾层。这一方面使网络变得更为复杂,需要更加高效地进行管理;另一方面也使部分的任务可以直接在雾层进行处理从而提升效率。前者保障的是传输上的高效性,后者保证的是任务在计算和存储上的高效性。

4.1 雾计算中的网络管理

雾计算中引入了中间雾层,因此,网络结构显得更为复杂,变成了终端节点到雾层的接入网络、雾层设备间的局域网络与雾层到云层之间的广域核心网络这 3 层。其中,终端节点到雾层的接入网络主要为 Wi-Fi、蓝牙、5G 等无线网络;而雾层设备间的局域网络与雾层到云层之间的广域核心网络则主要为有线网络。

4.1.1 雾计算的无线接入网络

雾计算的接入网络与移动云计算、分布式数据中心的接入网络相同,均通过无线信道进行传输。相比于云计算主要通过有线网络的模式更能提供移动性的支持,雾计算相比于移动云计算与分布式数据中心也存在其特点。在雾计算中终端节点只需要将任务上传至最近的雾节点。这样的直接好处是节省了任务传输的时间和终端节点的功耗并降低了数据在传输过程中受到威胁的可能性。而间接上由于一个雾节点对应的传感器数目的减少,也使无线网络的机构变得更为简洁,感知的内容变得更为有序。

对此,文献[16]提出了一种基于面向应用和地理位置的底层网络协议。在这个方案中,某个雾节点只服务一个指定的区域,同时提供有限的服务,一方面保证了这个区域的网络信号强度,另一方面一个较小区域内的服务的特性和需求往往是可预测的,从而保证了网络服务质量。文献[36]同样利用这种地域相似性的特点,考虑在内容分发的情境下通过点到点、寻找最近的雾节点和本地分布式协同这 3 种模式进行数据分发。其中,点到点模式中用户将从一定区域内支持点到点的设备中获取数据;最近的雾节点模式则是从最近的雾节点中进行数据获取,对于不支持点到点或一定范围内没有其他拥有该内容的点到点的设备的情况下只能用此方法;而本地分布式协同模式则是向区域内所有有此内容的雾节点进行请求^[6]。其通过对接入网络的建模分析,提出了一种自适应模式选择机制,并通过蒙特卡洛模拟加以验证。

这种内容的有序除了带来上述的好处外,还为内容的抽象带来了可能。由于其可以在一定程度上预测终端节点的需求,因此,可以采用雾节点主动向终端节点发送预测值的方法,使终端节点只需要传输与预测值不一致的参数,来进一步降低其功耗。

4.1.2 雾计算的有线网络

在雾计算中,雾节点既可以依靠自己的计算能力处理任务,也可以将任务迁移到更为合适的其他雾节点或云数据中心。由于雾节点均是预先部署的,因此,雾节点的数据转发行为可以很方便地进行集中控制。这为高效的数据转发提供了可能,非常适合采用 SDN 这种进行网络集中管理的技术^[16]。通过分离逻辑控制层与物理转发层,对网络进行集中管理,相比于传统的类似于 OSPF 的网络协议能够提升数据分组的投递率 (delivery ratio),减轻拥塞,降低时延^[37]。

然而 SDN 技术虽然有不少优势,但也同时具备一些挑战。首先是控制器的部署和协同问题,即对一个现有网络需要多少控制器,这些控制器部署在什么位置的问题。对此,文献[38]对于现存的多种拓扑结构进行了研究,并根据时延得出了部署的方案。但这种方案对于雾计算来说仍然存在问题,原因在于雾计算中节点的种类繁多,所执行的功能各异,可能存在不同种类的控制,而这些控制器如何进行协同是一个问题。其次是雾计算中对于时延、移动性等有比较高的要求,而传统的 SDN 的控制开销比较大,如何对 SDN 协议进行设计优化也是亟待研究的问题。此外,将有线网络与无线接入网络进行整合,通过 SDN 进行控制也是非常具有前景的一项研究。文献[6]对于 SDN 在接入网络上进行部署的可行性进行了讨论,其认为 SDN 可以扩展到接入网络中,更高效地管理接入网络。但这种扩展目前仍不成熟,原因在于 SDN 需要进行集中控制而底层设备是分布的。并且 SDN 要求设备周期性地向控制器汇报状态,这种控制开销也是不可接受的。

由于需要大量部署雾节点,因此,为节约成本,每个雾节点的资源相对有限,容易受到恶意流量的攻击。这使雾计算中的访问控制与入侵检测显得尤为关键。

在访问控制上,需要防止对任何资源进行未授权的访问。由于雾计算中大多为时延敏感的应用,因此,雾计算中的访问控制的关键在于提升效率。对此,文献[39]设计了一个基于策略的访问控制方法,对于不同的资源类型的要求,在雾节点上进行协同访问控制,以提升单节点的处理效率。

而在入侵检测中,需要及时发现内部攻击、泛洪攻击等攻击行为。对此,文献[40]对于云计算中的入侵检测进行了总结。但传统的应用于云计算的检

测方法不一定都能应用到雾计算上,因为雾计算相比于云计算的运算能力有限,且在雾计算中计算效率很关键。为此,文献[41,42]对于雾计算中内部攻击的形式进行了讨论。文献[43]设计了一套对智能电网中异常流量进行高效识别和处理的算法。

4.2 雾计算中的资源调度

在雾计算中,任务将被装载在虚拟机(Virtual Machine)或容器(container)中运行^[14]。这些虚拟机或容器可以在物理机之间进行迁移,以更高效地利用物理机的资源。虚拟机或容器迁移的位置依赖于调度算法的设计。在调度算法的设计上,相比于云计算的复杂之处,一方面,雾计算对时间的敏感程度很高,因此,用户到雾节点的时间、任务在不同雾节点中迁移的时间、雾节点将任务转发到云数据中心的时间都可能被纳入考量。另一方面,雾节点除了计算节点外可能还设置了存储相关的节点,这些节点的放置方式也会影响到相关的服务质量。因此,在进行调度算法的设计上需要考虑如下因素。

1) 存储容量(storage capacity)

存储是雾层的一大功能,雾层相比于云层的存储优势在于用户可以直接从雾层中获取所需数据,从而降低了数据获取的时延。与此同时,对于计算任务,其所需的源数据也可能分散在各雾节点上。对于存储则希望数据能尽量靠近需求,但也同时希望使用尽量少的存储空间^[37]。对此,最为经典的评价指标是数据获取时间。数据获取时间可以直接反映数据的局部性,但一味追求局部性可能导致一份数据同时存放在很多不必要的地方,而这可以通过限制数据备份数目进行控制。因此,这2个指标一般会同时出现作为存储情况的衡量。

2) 时延(delay)

除存储外,雾层中雾节点更多地承担着计算的功能,而时延是对于计算能力的一个重要的衡量指标。雾计算相比于云计算,计算任务可以在雾节点上进行而不必将任务上传到云数据中心,从而有效地降低了时延。对此最为准确的定义方式是往返时延(round-trip time),即从任务开始在终端设备进行上传到终端设备收到任务返回结果所用的时间。这样的定义在同样的任务进行横向比较下直观且方便,但这种方法没有考虑到任务计算量大小与任务所需传输的数据量大小的差异。即使采用单位时间执行的指令数目,也可能因为用户具体实现的原

因使人们并不能保证该程序以一个恒定的CPU利用率进行处理,从而使这种衡量方法在衡量计算时延上存在问题^[44]。对于计算时延来说,采用如违反服务协议(SLA violation)^[45]这样的对于资源短缺程度的衡量方法将更为合理^[46]。在违反服务协议中,对于短缺程度的定义为

$$SLA\ violation = \frac{\sum_{i=1}^m \int (r_i(t) - a_i(t)) dt}{\sum_{i=1}^m \int r_i(t) dt} \quad (1)$$

其中, m 为任务的数目, $r_i(t)$ 为第 i 项任务在 t 时间所请求的资源量, $a_i(t)$ 为第 i 项任务在 t 时间所实际分配到的资源量。由式(1)可知,资源占用率的程度越高,时延程度越大。这两者在一定条件下存在着线性关系。对此,证明如下。

假设每个任务相互独立,所请求的资源量为 X_i ,均值为 μ_i ,所分配的资源量为 $X_i + \varepsilon_i$,那么在违反服务协议下资源短缺程度的期望值为

$$\begin{aligned} E[SLA] &= \sum_{i=1}^m E \left[\int \left(1 - \frac{a_i(t)}{r_i(t)} \right) dt \right] \\ &= \sum_{i=1}^m \int E \left[1 - \frac{a_i(t)}{r_i(t)} \right] dt \\ &= \sum_{i=1}^m \int \left(1 - \frac{E[X_i] + E[\varepsilon_i]}{E[X_i]} \right) dt \\ &= - \sum_{i=1}^m \int \frac{E[\varepsilon_i]}{\mu_i} dt \end{aligned} \quad (2)$$

由于资源量不足,而产生时延的指令数的期望值为

$$\sum_{i=1}^m \int E[X_i - (X_i + \varepsilon_i)] dt = - \sum_{i=1}^m \mu_i \int \frac{E[\varepsilon_i]}{\mu_i} dt \quad (3)$$

显然,资源短缺程度的期望值与时延的指令数的期望值呈正比,而由于时延的指令数又与时延的时间呈正比,因此,可得出资源短缺程度的期望值与时延的时间呈正比。

不难发现,这样的衡量方法无法反映网络中的差分服务的时延代价,也无法反映非线性的时延代价。因此,对于时延代价的定义还有进一步研究的空间。

3) 功耗(power consumption)

这也是对于计算能力的一个重要的衡量指标。为提升资源使用效率,任务一般加载在虚拟机中执行,同一设备上的虚拟机将共享该设备的资源。图2展示了一些服务器的功耗参数。

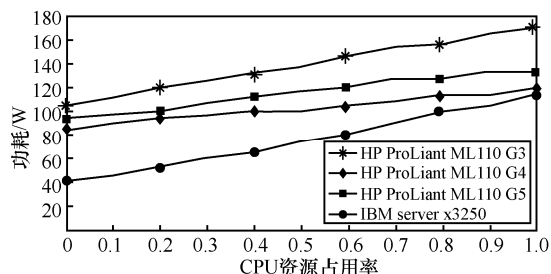


图2 服务器功耗参数

从图2中不难看出,对于一台设备来说,其功耗与CPU资源的占用率大致遵循线性关系。因此,可通过如下的定义进行功耗的建模

$$p = \int \sum_{i=1}^m \left[p_{\text{idle}_i} + (p_{\text{max}_i} - p_{\text{idle}_i}) \frac{a_i(t)}{c_i} \right] dt \quad (4)$$

其中, m 为服务器总数, p_{idle_i} 、 p_{max_i} 分别为第 i 台服务器的空闲时功耗与满载时功耗, $a_i(t)$ 、 c_i 分别为该台服务器占用的资源量与资源总量。从图2中还可以看出, CPU 占用率为 0 时的功耗远大于休眠时的功耗。因此,人们寄希望于将尽量多的虚拟机打包到尽量少的设备上面,但打包的程度越大,资源短缺的风险越大,从而时延的风险也随之增大。因此,一般会结合功耗和时延对计算能力进行统一的考量。此外,还有一些类似的指标也可以起到相同的效果,例如,与资源消耗量呈正比的二氧化碳排放量。

4) 效用 (utility)

雾和云一样具有其商业价值。虽然当前雾计算的部署大多具有特定的应用场景,且以私用雾为主。但随着雾计算的普及,将任务上传至私有的雾节点不一定能满足自身的需求。可能未来人们可以像租用云服务器一样租用雾节点,例如,在车联网中利用周围车辆与道路两旁的智能传感器所提供的算力进行计算,而在这里面就产生了计价的问题。但由于现在还没有一个像云按需服务这样成熟的商业模式,因此,这项指标主要针对具体的情景,但相信其未来会是一个研究的重点。

5) 网络资源占用与迁移时间 (network utilization/migration time)

为了达到更高的收益,可以在必要时对虚拟机进行迁移。由于网络资源有限,虚拟机在网络上进行迁移存在代价,而这个代价一般通过网络资源占用和迁移时间来反映。其中,网络资源占用直接衡量带宽的资源,而迁移时间不仅间接反映了带宽的情况,与此同时还反映了在网络线路传

输过程中的时延。但由于雾节点之间距离较小,这种时延实际上可以忽略不计,因此,这2种衡量方法均可采用。

雾计算中的调度策略如表3^[13,47~55]所示,现有很多研究从上面这些因素出发对于调度策略进行了研究(表3中√表示该项研究中考虑了这项指标,括号内为对该指标的描述,×表示该项研究中没考虑这项指标)。不可忽视的是,雾计算相比于云计算调度的难度更大,原因在于雾计算相较于云计算各项指标的波动更为明显,如何应付这样的明显波动是调度算法在实际应用上的一个主要问题。对此,可以通过雾节点只服务于有限范围的有限应用,利用在应用特征上的局部相似性,通过人工智能的方法加以分析和预测^[23],其或多或少存在对任务情况的假设(例如,假设任务的到来服从速率为 λ 的泊松分布^[50]或服从均值为 μ 、协方差矩阵为 Σ 的高斯混合分布^[56])与参数的人为设定的传统算法的表现不佳。对此,可认为不需预先假设分布并可根据任务的实际情况进行在线学习 (online learning) 的强化学习 (reinforcement learning) 方法是一种很好的解决方案。在该方法中会对历史的分流行为进行打分,并不断引导将任务按照分高的策略进行分流。

与此同时,由于雾计算中需要考虑的因素较多、状态复杂,不少学者采用深度学习进行问题的降维。文献[57]将深度强化学习用到网络资源调度中。文献[58]提出了在云数据中心中一种权衡整体资源分配和电源管理问题的新颖层次框架,采用深度强化学习 (DRL, deep reinforcement learning) 技术,结合自动编码器 (auto-encoding)、LSTM 等神经网络,可以有效地在功耗和处理时延间取得最佳权衡。此外,还有一个不足之处在于,之前的算法几乎都是基于独立任务的优化。而在实际情况下,任务之间是存在着依赖关系的。这些依赖关系会对任务执行的顺序和任务分配的位置有着直接的影响。对此,文献[59]提出了基于任务依赖关系的云数据中心的任务调度方法。文献[55]在雾数据中心的调度中也考虑到了依赖关系,但是只考虑了时间因素,没能考虑功耗、网络质量等因素。但是如果依靠深度学习进行在线学习,从而达到自适应将其消耗计算资源,这在雾计算中的适用性仍存在着疑问。当前高效低耗的自适应调度算法仍是一个重要的研究课题。

表 3 雾计算中的调度策略

| 文献来源 | 存储容量 | 时延 | 功耗 | 效用 | 网络资源占用与迁移时间 |
|--------|-------------|---------------------|-------------|------------|-----------------|
| 文献[13] | × | √（计算时延、传输时延） | √（功耗） | × | √（给定带宽下的网络资源占用） |
| 文献[47] | × | √（往返时延） | × | × | √（给定带宽下的网络资源占用） |
| 文献[48] | × | × | √（平均二氧化碳排放） | √（平均传输数据量） | × |
| 文献[49] | × | √（计算时延、传输时延） | √（功耗） | √（基础设施成本） | √（迁移时间） |
| 文献[50] | √（服务器的存储容量） | √（计算时延、I/O 时延、传输时延） | × | × | × |
| 文献[51] | × | × | × | √（平均传输数据量） | √（给定带宽下的网络资源占用） |
| 文献[52] | × | √（往返时延） | √（功耗） | × | × |
| 文献[53] | × | √（往返时延） | × | × | √（给定通信功率） |
| 文献[54] | × | √（资源短缺程度） | × | √（平均传输数据量） | × |
| 文献[55] | × | √（往返时延） | × | × | × |

5 雾计算的应用

由于雾计算相比于云计算具有贴近用户、能够提供地理分布与移动性支持的特点，因此其适于 4 种应用类型：1)以大型网络游戏、超高清视频的加载为代表的具有大数据量但要求较低时延的应用；2)以无线传感网络为代表的基于地理分布的应用；3)以车联网、可穿戴设备为代表的需要快速反馈的移动应用；4)以智能电网、智能交通指挥系统为代表的大规模分布式决策系统^[13]。

5.1 雾计算的应用场景

5.1.1 具有大数据量但要求较低时延的应用

对于大型网络游戏、超高清视频这样的应用来说，雾计算可以像内容分发网络一样，将大数据缓存在靠近用户的位置，从而减少了从远端数据中心获取数据的时延，也节省了网络的带宽。与此同时，雾节点也可以进行一定程度的数据处理，从而降低了终端设备的计算任务，提升了应用的性能^[37]。

5.1.2 基于地理分布的应用

雾节点可以部署在靠近传感器的位置以减少从终端设备到雾节点的距离^[4]。这样做可以有效降低传感器的传输距离，从而节约了传感器的能耗，保证了数据的质量，降低了数据被窥探的风险。与此同时，雾节点可以协助终端传感设备进行计算，提升了数据的计算效率。对此，在 2015 年 11 月，ARM、思科、戴尔、英特尔等公司在北美建立了国际雾计算产学研联盟（OpenFog 联盟）。而在

2017 年 4 月，国际雾计算产学研联盟（OpenFog 联盟）大中华区在上海成立。该联盟意在针对物联网制订出严格的功能和架构要求以及详尽的应用程序接口和性能标准，以指导实施可互操作的设计，加速雾计算的商用化^[60]。

5.1.3 快速反馈的移动应用

相比于云计算，雾计算通过安装在移动物体上的传感器与该区域内固定装载的传感器进行协同感知，然后及时交由对应的雾节点进行处理，并将得到的结果立即反馈给用户，可以有效降低时延。对此，文献[5]将雾计算引入车联网，并发掘了车辆本身作为雾节点的巨大潜力，其认为北上广深等大城市汽车的保有量大，在任意的区域内都有足够多的汽车能够提供足够强大的算力，从而用户的任务可以直接转发到邻近的汽车上进行处理，而不必都传输到远端的云数据中心。这种时延的降低对于未来的无人驾驶大有裨益。文献[61]设计了一款基于雾计算的健康监测系统。该系统的测量结果可以直接在本地的智能网关进行处理，从而可以对人体的突发病症及时报警。

5.1.4 大规模分布式决策系统

在大规模分布式决策系统中，雾计算可以改善传统的基于云计算在位置感知、移动性支持和时延上的问题，形成快速有效的物联信息决策系统，以达到更有效、更智能的决策的目的。其中，典型的应用有智能交通、智能电网和智能建筑。在智能交通中，通过在车辆上与道路两旁部署的传感器，可

以及及时发现驾驶员的危险驾驶行为,及时做出预警并进行合适的处罚^[62]。同时也可以利用车速、道路状况等传感得到的信息,自动调节交通灯的颜色,使无车辆时自动关闭交通灯,还可根据实际情况(行人、救护车)调控交通灯的颜色,并及时向驾驶员发出警告以减速^[29]。而在智能电网中,通过在网络的边缘部署能量平衡的应用,可以智能调度风能、水能等能源的使用方式,从而提升能源的使用效率^[63]。在智能建筑中,将建筑中的传感器进行联网,通过智能网关对传感得到的数据进行处理,并根据处理后的数据和预先指定的规则进行控制。

5.2 雾计算应用的相关研究

5.2.1 智能驾驶示范应用

智能驾驶示范应用如图 3 所示,在传统的驾驶模式中,车辆利用 GPS 传感器从卫星处获取位置信息,并将该信息发往导航软件的云数据中心。数据中心在收集数据后经过计算得到导航信息并将信息通过网络发至车辆。受限于网络的时延和安全性,这样得到的信息较为粗放,只能呈现出车辆经过哪些道路的大致的行驶路径,而不能对车辆在实时路况下的加速、减速、避让等行为做出判断,从而无法实现无人驾驶。而智能驾驶相比于传统驾驶模式,可以通过摄像头、超声等传感设备获取道路的实际情况,实现车辆的无人驾驶。在这些场景中需要解决的关键问题在于对收集到的信息进行快速传输、处理与验证。

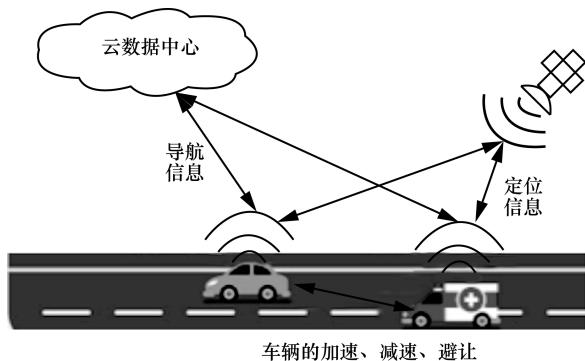


图 3 智能驾驶示范应用

在雾计算的具体应用中,时延的瓶颈极可能出现在网络传输与任务处理上。对于网络传输来说,据 4.1 节所述,在雾计算中主要涉及终端用户到雾节点、雾节点之间与雾节点到云数据中心这 3 个部分的网络。由于雾节点需要负责时延敏感的任务的处理,因此,终端用户到雾节点的无线接入网络的

传输质量尤为关键。在智能驾驶中,车辆的导航信息对于时延的需求不大,因此,可以交由远端云数据中心进行处理。但要在行驶过程中实现对其他车辆与行人的避让对于时延的需求很高,稍有时延极有可能车毁人亡,因此,必须在近地的雾节点中进行数据的收集与处理^[64]。但当前无线网络的传输速度与信号质量仍不能达到这样的需求。未来当 5G 技术成熟并商用后,利用 5G 比 4G 快数 10 倍的传输速度,可以使雾计算在智能驾驶等对于时延极其敏感的应用上成为可能。因此,结合 5G 技术的雾计算将会是非常重要的研究方向。

而在任务处理上,4.2 节对现有的资源调配算法进行总结,从而使雾计算服务提供商可以更高效、更有效益地完成。由于车辆本身的计算能力可能有限,因此,任务有可能迁移至车外的雾节点中进行执行。当前不论是虚拟机的迁移还是容器的迁移都需要移动大量的信息,而这很可能带来不可接受的时延。且当前对于雾计算的虚拟化技术研究仍处于空白阶段。此外,由于雾计算需要应付实际的网络环境,而该环境变化较为剧烈且存在一定的时空性。对此不宜预先假设分布,而需要采用自适应的算法加以解决。当前自适应的算法主要通过强化学习算法进行实现,但该方法要达到自适应的效果极其消耗计算资源,因此,其在雾计算中的适用性仍存在着疑问。此外,由于在智能驾驶中车辆的位置随时间会发生较大的变化,而这种变化也可能带来额外的传输时延。因此,如何采用一个高效低耗的自适应调度算法来选择合适的雾节点进行迁移,使信息的消费者能够尽快地拿到数据是其中一个亟待解决的问题。

由于在智能驾驶中雾节点主要部署在开放的场景中,因此如果有不法分子进行信息的窃取,甚至是信息的篡改很可能造成灾难性影响^[65]。例如,如果后车已经做出了准备超车的决定,但前车没有及时收到该信息或者收到了错误的信息而没能及时避让就很可能酿成交通事故。这关键在于保障消息的完整性和可用性。虽然当前已有文献对于雾计算中可能出现的攻击行为^[4,65]进行分析,但当前所提出的解决方案在具体应用上的时延和顽健性仍未得到测试。因此在雾计算中高效的加密和验证算法也将是研究的重点。

5.2.2 工业互联网示范应用

当前工业生产主要基于云计算进行。但在实

践中发现由于云服务器的时延过大,很多精细化的操作(例如微型元件的切割)无法按时完成。此外,企业在底层所收集到的数据量不断上升也使企业的骨干网备受考验。工业互联网示范应用如图4所示,为了解决此类问题,企业可以在终端设备和云数据中心之间部署雾层计算和存储设备,以处理时延敏感的应用与大数据特征的提取。这除了对时延和安全性有要求外,还对平台的统一性和云雾间协同提出了更高的要求^[66]。

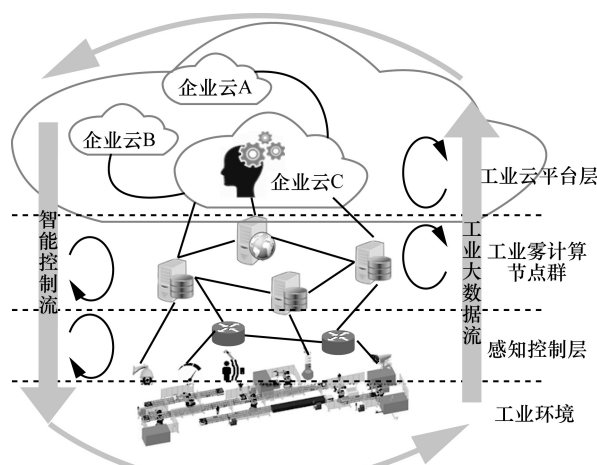


图4 工业互联网示范应用

首先,由于工业生产环境涉及的底层元件较为复杂,为使这些元件的数据能够很好地收集并进行控制,需要制作统一的控制平台对云雾端架构进行有效的覆盖和连接^[14]。同时,实现软件定义雾节点群组,对云平台、异构网络和海量终端设备进行承上启下的有效覆盖和连接,形成具备云端融合能力的标准API接口及规范。但在当前对于统一接口的研究仍处于起步阶段。与此同时,对于控制器部署数量以及位置的问题研究仍不充分,无法将网络、存储和计算等资源进行统一管理。而这些都是未来亟待解决的问题。

其次,工业生产环境中的雾节点除了要应对时延敏感的应用外,还需要对提交到云数据中心进行处理的数据进行初步的特征提取,以降低企业骨干网的负荷。这就涉及云雾端融合的关键技术。在该技术中需要研究雾层要将什么样的任务提交到云数据中心,以什么状态提交到云数据中心,提交到云数据中心的哪个服务器,通过怎样的路由进行提交等核心问题。

6 结束语

雾计算作为云计算引申出的新型计算模式,其相比于传统的计算模式具有低时延、能提供地理分布与移动性的支持等特点。本文对雾计算的概念、架构、研究问题与应用进行了综述,并相信雾计算未来在基于地理分布、移动性等应用和大规模分布式决策系统上有着巨大的发展潜力。而在理论研究方面,未来的重要研究点仍将主要集中在雾层中,尤其是高效的网络通信协议、系统安全算法、雾层虚拟化技术、云雾端融合技术等方向。与此同时,在雾计算的应用中,既要保证算法的效率又要保证算法的顽健性,而这也是雾计算应用上急需攻关的重要课题。

参考文献:

- [1] ARMBRUST M, FOX A, GRIFFITH R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4):50-58.
- [2] BONOMI F, MILITO R, ZHU J, et al. Fog computing and its role in the internet of things[C]//Edition of the Mcc Workshop on Mobile Cloud Computing. 2012:13-16.
- [3] HASHIZUME K, ROSADO D G, FERNÁNDEZ-MEDINA E, et al. An analysis of security issues for cloud computing[J]. Journal of Internet Services and Applications, 2013, 4(1):5.
- [4] STOJIMENOVIC I, WEN S. The fog computing paradigm: scenarios and security issues[C]// Federated Conference on Computer Science and Information Systems. 2014:1-8.
- [5] HOU X, LI Y, CHEN M, et al. Vehicular fog computing: a viewpoint of vehicles as the infrastructures[J]. IEEE Transactions on Vehicular Technology, 2016, 65(6):3860-3873.
- [6] PENG M, YAN S, ZHANG K, et al. Fog-computing-based radio access networks: issues and challenges[J]. IEEE Network, 2015, 30(4):46-53.
- [7] LU R, ZHU H, LIU X, et al. Toward efficient and privacy-preserving computing in big data era[J]. IEEE Network, 2014, 28(4):46-50.
- [8] BONOMI F. Connected vehicles, the internet of things, and fog computing[C]//The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET). 2011: 13-15.
- [9] VAQUERO L M, RODERO-MERINO L. Finding your way in the fog[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(5):27-32.
- [10] FERNANDO N, SENG W L, RAHAYU W. Mobile cloud computing: a survey[J]. Future Generation Computer Systems, 2013, 29(1):84-106.
- [11] DINH H T, LEE C, NIYATO D, et al. A survey of mobile cloud computing: architecture, applications, and approaches[J]. Wireless Communications & Mobile Computing, 2013, 13(18):1587-1611.
- [12] DAVIS A, PARIKH J, WEIHL W E. Edge computing: extending enterprise applications to the edge of the internet[C]// International Conference on World Wide Web-Alternate Track Papers & Posters. 2004:180-187.

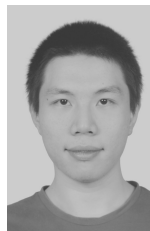
- [13] DENG R, LU R, LAI C, et al. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption[J]. *IEEE Internet of Things Journal*, 2017, 3(6):1171-1181.
- [14] YI S, HAO Z, QIN Z, et al. Fog computing: platform and applications[C]// *Hot Topics in Web Systems and Technologies*. 2015:73-78.
- [15] FIRDHOUS M, GHAZALI O, HASSAN S, et al. Fog computing: will it be the future of cloud computing?[C]// *International Conference on Informatics & Applications*. 2014: 1-8.
- [16] LUAN T H, GAO L, LI Z, et al. Fog computing: focusing on mobile users at the edge[J]. *arXiv Preprint*, arXiv: 1502.01815, 2015.
- [17] LI J, JIN J, YUAN D, et al. EHOPES: data-centered fog platform for smart living[C]// *Telecommunication Networks and Applications Conference*. 2015:308-313.
- [18] HAJIBABA M, GORGIN S. A review on modern distributed computing paradigms: cloud computing, jungle computing and fog computing[J]. *Journal of Computing & Information Technology*, 2014, 22(2):69.
- [19] SATYANARAYANAN M. Fundamental challenges in mobile computing[J]. *ACM Symposium on Principles of Distributed Computing*, 1996:1-7.
- [20] LI W, ZHAO Y, LU S, et al. Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing[J]. *IEEE Communications Magazine*, 2015, 53(3):89-97.
- [21] TULI A, HASTEER N, SHARMA M, et al. Exploring challenges in mobile cloud computing: an overview[C]// *Confluence 2013: the Next Generation Information Technology Summit*. 2013:496-501.
- [22] NISHIO T, SHINKUMA R, TAKAHASHI T, et al. Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud[C]// *International Workshop on Mobile Cloud Computing & NETWORKING*. 2013:19-26.
- [23] BITTENCOURT L F, LOPES M M, PETRI I, et al. Towards virtual machine migration in fog computing[C]// *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. 2015:1-8.
- [24] SHI W, CAO J, ZHANG Q, et al. Edge computing: vision and challenges[J]. *IEEE Internet of Things Journal*, 2016, 3(5):637-646.
- [25] CHURCH K, GREENBERG A, HAMILTON J. On delivering embarrassingly distributed cloud services[C]// *Hotnets*. 2008: 55-60.
- [26] GU C, LIU C, ZHANG J, et al. Green scheduling for cloud data centers using renewable resources[J]. *Proceedings-IEEE INFOCOM*, 2015, 2015: 354-359.
- [27] SATYANARAYANAN M, BAHL P, CACERES R, et al. The case for VM-based cloudlets in mobile computing[J]. *IEEE Pervasive Computing*, 2009, 8(4):14-23.
- [28] YANNUZZI M, MILITO R, SERRAL-GRACIA R, et al. Key ingredients in an IoT recipe: fog computing, cloud computing, and more fog computing[C]// *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*. 2014:325-329.
- [29] LUAN T H, CAI L X, CHEN J, et al. VTube: towards the media rich city life with autonomous vehicular content distribution[C]// *Sensor, Mesh and Ad Hoc Communications and Networks*. 2011:359-367.
- [30] GARCIA L P, MONTRESOR A, EPEMA D, et al. Edge-centric computing: vision and challenges[J]. *ACM Sigcomm Computer Communication Review*, 2015, 45(5):37-42.
- [31] JUTILA M. An adaptive edge router enabling Internet of things[J]. *IEEE Internet of Things Journal*, 2016, 3(6):1061-1069.
- [32] XU Y, MAHENDRAN V, RADHAKRISHNAN S. Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery[C]// *International Conference on Communication Systems and Networks*. 2016:1-6.
- [33] KRISHNAN Y N, BHAGWAT C N, UTPAT A P. Fog computing—network based cloud computing[C]// *International Conference on Electronics and Communication Systems*. 2015:250-251.
- [34] BRUNEO D, DISTEFANO S, LONGO F, et al. Stack4Things as a fog computing platform for smart city applications[C]// *IEEE INFOCOM 2016-IEEE Conference on Computer Communications Workshops*. 2016:848-853.
- [35] GUPTA H, VAHIDDASTJERDI A, GHOSH S K, et al. iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of things, edge and fog computing environments[J]. *Software: Practice and Experience*, 2017, 47(9): 1275-1296.
- [36] YAN S, PENG M, WANG W. User access mode selection in fog computing based radio access networks[J]. *arXiv preprint*, arXiv: 1602.00766, 2016.
- [37] YI S, LI C, LI Q. A survey of fog computing: concepts, applications and issues[C]// *The Workshop on Mobile Big Data*. 2015:37-42.
- [38] HELLER B, SHERWOOD R, MCKEOWN N. The controller placement problem[C]// *Workshop on Hot Topics in Software Defined Networks*. 2012:7-12.
- [39] DSOUZA C, AHN G J, TAGUINOD M. Policy-driven security management for fog computing: preliminary framework and a case study[C]// *IEEE International Conference on Information Reuse and Integration*. 2015:16-23.
- [40] MODI C, PATEL D, BORISANIYA B, et al. Review: a survey of intrusion detection techniques in cloud[J]. *Journal of Network & Computer Applications*, 2013, 36(1):42-57.
- [41] KULKARNI S, SAHA S, HOCKENBURY R. Preserving privacy in sensor-fog networks[C]// *Internet Technology and Secured Transactions*. 2015:96-99.
- [42] STOLFO S J, SALEM M B, KEROMYTIS A D. Fog computing: mitigating insider data theft attacks in the cloud[C]// *IEEE Symposium on Security and Privacy Workshops*. 2012:125-128.
- [43] VALENZUELA J, WANG J, BISSINGER N. Real-time intrusion detection in power system operations[J]. *IEEE Transactions on Power Systems*, 2013, 28(2):1052-1062.
- [44] STONE M. Cross-validatory choice and assessment of statistical predictions[M]// *Introduction to chaos*: Institute of Physics Pub. 1999:111-147.
- [45] FARAHNAKIAN F, LILJEBERG P, PLOSIJA J. Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning[C]// *Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. 2014:500-507.
- [46] HAN Z, TAN H, CHEN G, et al. Dynamic virtual machine management via approximate Markov decision process[C]// *IEEE Conference on Computer Communications*. 2016: 1-9.
- [47] BEATE OTTENWÄLDER, KOLDEHOFE B, ROTHERMEL K, et al. MigCEP: operator migration for mobility driven distributed complex event processing[C]// *ACM International Conference on Distributed Event-Based Systems*. 2013:183-194.
- [48] DO C T, TRAN N H, PHAM C, et al. A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing[C]// *International Conference on Information NET-*

- WORKING. 2015:324-329.
- [49] ZHANG H, XIAO Y, BU S, et al. Fog computing in multi-tier data center networks: a hierarchical game approach[C]// IEEE International Conference on Communications. 2016:1-6.
- [50] ZENG D, GU L, GUO S, et al. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system[J]. IEEE Transactions on Computers, 2016, 65(12): 3702-3712.
- [51] HUANG C Y, XU K. Reliable realtime streaming in vehicular cloud-fog computing networks[C]//IEEE/CIC International Conference on Communications in China. 2016:1-6.
- [52] ZHANG H, ZHANG Q, DU X. Toward vehicle-assisted cloud computing for smartphones[J]. IEEE Transactions on Vehicular Technology, 2015, 64(12):5610-5618.
- [53] OUEIS J, STRINATI E C, BARBAROSSA S. The fog balancing: load distribution for small cell cloud computing[C]// Vehicular Technology Conference. 2015:1-6.
- [54] HONG H J, TSAI P H, HSU C H. Dynamic module deployment in a fog computing platform[C]// Network Operations and Management Symposium. 2016:1-6.
- [55] PHAM X Q, HUH E N. Towards task scheduling in a cloud-fog computing system[C]//Network Operations and Management Symposium (APNOMS). 2016: 1-4.
- [56] YANG J, ZHANG S, WU X, et al. Online learning-based server provisioning for electricity cost reduction in data center[J]. IEEE Transactions on Control Systems Technology, 2016, PP(99):1-8.
- [57] MAO H, ALIZADEH M, MENACHE I, et al. Resource management with deep reinforcement learning[C]// ACM Workshop on Hot Topics in Networks. 2016:50-56.
- [58] LIU N, LI Z, XU J, et al. A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017: 372-382.
- [59] ZHU X, CHEN H, YANG L T, et al. Energy-aware rolling-horizon scheduling for real-time tasks in virtualized cloud data centers[C]// High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC). 2013: 1119-1126.
- [60] OpenFog Consortium Architecture Working Group. OpenFog reference architecture for fog computing[J]. OPFRA001, 2017, 20817: 162.
- [61] CAO Y, HOU P, BROWN D, et al. Distributed analytics and edge intelligence: pervasive health monitoring at the era of fog computing[C]// The Workshop on Mobile Big Data. 2015:43-48.
- [62] ROY S, BOSE R, SARDDAR D. A fog-based dss model for driving rule violation monitoring framework on the Internet of things[J]. International Journal of Advanced Science & Technology, 2015, 82:23-32.
- [63] FADLULLAH Z M, KATO N. On optimally reducing power loss in micro-grids with power storage devices[M]// Evolution of Smart Grids. Springer International Publishing. 2015: 1361-1370.
- [64] KOPETZ H, POLEDNA S. In-vehicle real-time fog computing[C]// IEEE/IFIP International Conference on Dependable Systems and Networks Workshop. 2016:162-167.
- [65] KHAN S, PARKINSON S, QIN Y. Fog computing security: a review of current applications and security solutions[J]. Journal of Cloud Computing, 2017, 6(1):19.
- [66] LOM M, PRIBYL O, SVITEK M. Industry 4.0 as a part of smart cities[C]//Smart Cities Symposium Prague. 2016:1-6.

[作者简介]



贾维嘉 (1957-), 男, 河北承德人, 博士, 上海交通大学致远讲席教授, 主要研究方向为人机物融合、多播、选播、路由、移动多媒体通信、分布式系统等。



周小杰 (1994-), 男, 广东梅州人, 上海交通大学硕士生, 主要研究方向为数据中心、雾计算、资源调度等。