

北京邮电大学

本科毕业设计（论文）



题目：边缘计算架构下物联网服务质量评价与优化方法研究

姓 名 李松远

学 院 计算机

专 业 计算机科学与技术

指导教师 黄霁崴

2018 年 6 月

北京邮电大学

本科毕业设计（论文）诚信声明

本人声明所呈交的毕业设计（论文），题目《边缘计算架构下物联网服务质量评价与优化方法研究》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

边缘计算架构下物联网服务质量评价与优化方法研究

摘要

物联网将现实世界中的各类末端设备或设施相互连接起来，最终形成大规模设备节点的网络拓扑。通过嵌入功能或通信相关的软硬件模块，在既有互联网技术架下，物联网中的设备节点实现自主的信息交互和功能控制。近年来，物联网的应用领域逐步渗透到智慧城市、工业控制、智能电网、资源测量等行业，参与节点的数量迅速扩张，网络拓扑规模增速巨大。然而，物联网节点的大幅扩张使得系统面临着资源紧张、网络拥挤等一系列服务质量相关的挑战。因此，边缘计算架构应运而生。作为新兴的计算范式，它在云计算架构的基础上，根据物联网的特点作出计算模式的优化，应用程序、数据、计算服务的控制权由云中央计算节点迁移至接近物联网设备的边缘计算节点，极大地缓解了物联网系统所遇到的挑战。

本文针对边缘计算架构下的物联网服务，围绕服务建模、量化服务质量指标、服务调度优化，由性能、可靠性、能源效率三个服务质量指标出发，研究服务质量的评价和优化。本文的工作主要包括以下两个方面：

(1) 面向物联网中可修复的服务组件，基于广义随机Petri网模型，提出一种可靠性感知的性能评价方法。从平均失效前时间和平均修复时间两个角度，量化分析表达服务的可靠性，并以此为基础，预测物联网服务在边缘计算系统中的性能水平。通过仿真实验，评价模型的有效性最后得到了验证。

(2) 在边缘计算架构的背景下，运用随机模型，对物联网服务在边缘计算系统中的服务过程给出了模型分析方法，并提出了性能、能源效率的量化分析表达。综合权衡能源效率、性能的影响，基于马尔可夫决策过程，提出了资源管理、任务调度的能效-性能综合优化算法。结合序优化技术，缩减优化问题的搜索空间，减少调度的执行时间，使调度算法在大规模系统中更加实际可用。基于物联网数据集，调度算法的有效性和高效性得到实验验证。

关键词 物联网 边缘计算 服务质量 多属性 动态优化

QoS Evaluation and Optimization for IoT Services in Edge Computing Architecture

ABSTRACT

Internet of Things (IoT) enables the interconnection of various terminal devices and facilities in the real world, where a large scaled network of device nodes participates in. On the basis of Internet technical framework, autonomous information exchange and self-directed function control are implemented with software or hardware modules. In recent years, applications range from smart city, industrial control, smart grid to resource measurement, which is accompanied by the extension of device nodes and the expansion of network scale. Under the circumstances, a series of challenges related to Quality of Service (QoS) have emerged, such as, resource shortage and network congestion. Therefore, a burgeoning computing paradigm namely edge computing comes into being. Edge computing is an emerging computing paradigm which modifies and optimizes the computational pattern based on the traditional cloud computing paradigm. To be particular, control over applications, data and computing services is migrated to the edge of network which can be accessed to by IoT devices.

To address these issues, research work is conducted on the IoT services under edge computing architecture from the perspective of QoS in three aspects which are performance, reliability and energy efficiency. We study the theoretical modeling of IoT services, quantitative analysis of QoS metrics and service scheduling for multi-attribute QoS optimization. The dissertation is summarized as follows.

(1) To study the repairable service components in the IoT environment, a predictive approach of reliability-aware performance evaluation based on Generalized Stochastic Petri Net (GSPN) is put forward. Quantitative analysis of service reliability, including mean time to failure (MTTF) as well as mean time to repair (MTTR) is carried on, based on which performance prediction results for IoT services under edge computing paradigm is worked out. Finally, effectiveness of the evaluation model is validated based on experiments.

(2) In the context of edge computing paradigm, the procedure of IoT services in edge computing system is dynamically analyzed with stochastic models. Mathematical formulations of performance and energy efficiency is carried out. Balancing the tradeoff between performance and energy efficiency, the task scheduling and resource management are formulated by Markov Decision Process (MDP). In order to make our algorithm more practical, Ordinal Optimization (OO) techniques are applied to the MDP algorithms, which narrows the search of MDP and reduces the execution time of scheduling. Based on the IoT dataset, the effectiveness and efficiency of scheduling algorithm get verified experimentally.

KEY WORDS Internet of Things Edge Computing Quality of Service Multi-attribute Dynamic Optimization

目 录

第一章 引言	1
1.1 研究背景	1
1.1.1 物联网	1
1.1.2 边缘计算	1
1.2 研究内容与意义	2
1.2.1 多属性服务质量指标的建模分析	2
1.2.2 任务调度与资源管理	3
1.3 研究难点与挑战	3
1.3.1 故障发生/修复给性能建模带来的挑战	3
1.3.2 多指标服务质量的折衷权衡	3
1.3.3 大规模系统特性给优化算法设计带来的挑战	4
1.4 论文组织	4
第二章 相关研究综述	5
2.1 服务质量的评价方法	5
2.1.1 基于测量的评价方法	5
2.1.2 基于预测的评价方法	5
2.1.3 基于模型的评价方法	5
2.2 服务质量的多目标优化方法	6
2.2.1 多目标聚集	6
2.2.2 ϵ -约束	6
2.2.3 帕累托最优	7
2.3 讨论与总结	7
第三章 可靠性感知的性能评价模型与分析	8
3.1 本章引论	8
3.2 基于广义随机 Petri 模型的可靠性感知建模	8
3.2.1 机群模型	8
3.2.2 通信模型	9

3.2.3 系统模型.....	10
3.3 实验验证	11
3.3.1 数据集简介.....	11
3.3.2 实验结果.....	12
3.4 本章小结	12
第四章 性能与能效优化的任务调度与资源管理.....	14
4.1 本章引论	14
4.2 系统模型	14
4.2.1 终端设备.....	14
4.2.2 服务机群.....	15
4.2.3 系统架构模型.....	15
4.3 模型分析与服务质量评价	17
4.3.1 性能指标.....	17
4.3.2 能效指标.....	17
4.4 性能与能效优化	18
4.4.1 回报模型.....	18
4.4.2 基于马尔可夫决策的优化框架.....	19
4.4.3 搜索空间压缩的调度算法.....	20
4.5 实验验证	23
4.6 本章小结	25
第五章 总结与展望.....	26
5.1 论文总结	26
5.2 研究展望	26
参考文献.....	28
致 谢.....	31
在学期间参加课题的研究成果.....	32
外 文 资 料.....	33
外 文 译 文.....	39

第一章 引言

1.1 研究背景

1.1.1 物联网

计算的未来发展方向将远远超出桌面计算(Desktop Computing)的范畴^[1]。当前,物联网(Internet of Things, IoT)技术的应用领域广泛。它作为一种关于目标数据的测量收集工具,为人类决策的行为提供了重要辅助。物联网已经几乎囊括了各类物理设备,如智能移动电话、汽车、传感器以及其他嵌入式计算设备。成千上万的传感器和移动设备不间断地产生原生物联网数据,并透过复杂的网络拓扑进行重要信息交互以实现端到端的通信以及智能架构的监控和控制^[2]。

诸如智能交通^[3]、智慧城市^[4]、智能电网^[5]等十分流行的应用,都是基于物联网技术进行上层开发。由此可见,物联网对人类日常生活的影响的深远程度可见一斑,并且将持续深化加强。在商贸经济领域中,物联网技术也扮演着重要角色。中国国务院于2015年印发的《中国制造2025》白皮书^[6]将物联网技术作为多项重要支撑技术之一,对中国商贸的发展有促进作用。全球互联的物理设备规模已经远远超出世界人口总和。2012年,全球互联设备数量增长至90亿之多^[1],并预测在2020年将达到750亿台设备的互联规模^[4]。在不久的将来,物联网设备将成为重要且不可忽视的海量数据来源之一^[7]。

1.1.2 边缘计算

由于物联网中互联物理设备数量的持续增长,以云数据中心为计算核心的传统计算模式不再能满足物联网应用的服务质量(Quality of Service, QoS)要求^[8]。海量的原生数据从物联网终端设备中生成,数据的云端上传过程将占用大量的网络带宽,并且海量数据的集中处理会给云端带来沉重的计算负载,最终造成资源拥挤、增大响应延迟。在很多物联网的应用场景中,数据处理和交互需要保证实时性,而实现即时交互处理则需要加强物联网中边缘部分的计算处理能力。传统的云计算架构,偏重于云数据中心计算能力的加强和完善,而疏于云数据中心以外的控制和维护^[9]。因此,传统的云计算架构不能解决物联网边缘处的海量传感数据爆炸问题,从而无法适应有实时性要求的物联网应用。

因此,一种新兴计算架构“边缘计算”被提出,它解决了物联网和本地计算的需求、缓解了终端设备附近资源拥挤的加剧状况。与传统云计算相比,边缘计算将部分数据处理和存储任务分配给物联网中接近终端设备的逻辑边缘。这样就将原来云端的计算负载拆分给多个边缘计算节点,实现云端负载的降低、响应延迟的缩短^[10]。同时,伴随着5G技术的即将商用,边缘计算将更好地解决物联网中的难题^[11,12]。5G技术的关键挑战是无线接入网络技术(Radio Access Network, RAN)^[13]。在移动边缘计算环境中,实时的RAN数据被提供。由于实时RAN信息提供了基于上下文感知的服务,所以服务提供商能够

利用实时的 RAN 信息来提升用户的体验质量(Quality of Experience, QoE)^[14]。

目前的物联网系统通常基于边缘计算架构搭建部署,引入位于物联网终端设备与云之间的中间边缘层,用来处理部分工作负载和本地化服务^[15]。中间边缘层由地理分布式的服务器机群组成,相当于轻量级的云服务器。除此之外,由于高性能的云服务器和轻型的边缘服务器通常以层次化的结构被组织成系统整体,所以边缘计算架构下的物联网

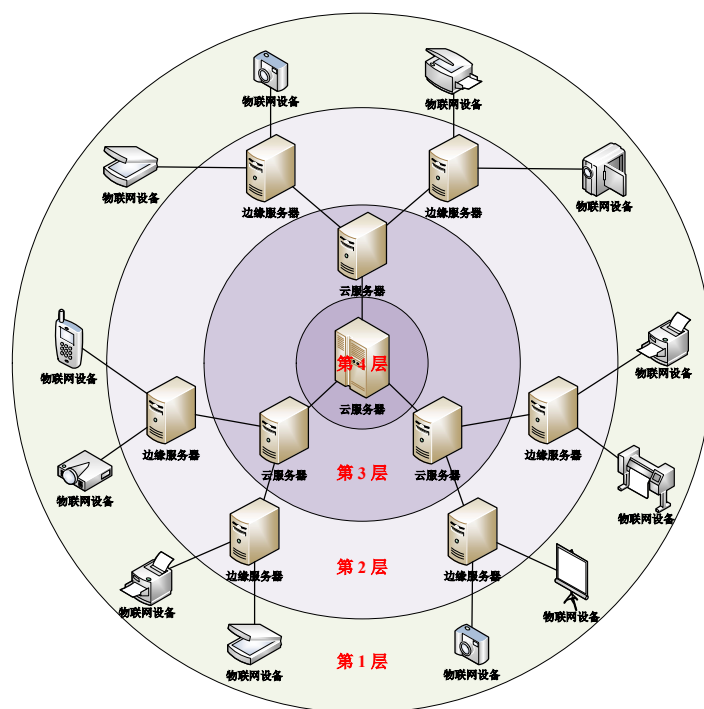


图 1-1 物联网系统的层次化结构示意图

系统可以认为是一种结构层次化的系统,如图 1-1 所示。

根据边缘计算架构的特点,将每一类设备/服务器分配为单独的一层,并将各层从 1 到 L 编号。更具体地说,将负责收集原始数据的物联网设备分类置为第 1 层。边缘服务器直接连接第 1 层的物联网终端设备,并处理它们收集的原始数据,该层作为第 2 层。云服务器具有更强大的配置权限,用于处理边缘服务器向上层提交的任务,标记为第 3 层;以此类推。部分设备连接可能跳过层次结构中的一层或多层。例如,在一些云端主导的应用中,物联网设备可以直接与集中式云服务器通信^[16,17],因此在图 1-1 中的 4 中可能存在从物联网设备到云服务器的连接线。

1.2 研究内容与意义

1.2.1 多属性服务质量指标的建模分析

随着对服务质量需求的日益提高,单一服务质量属性已经不能满足服务质量优化的需要。相反地,亟需建立多属性的服务质量指标评价体系,不仅能够全面客观地进行系统评价,并且更为铺垫后续服务质量的优化工作。关于多属性服务质量的评价,本文采用模型的方法定量分析。

首先，建立可靠性感知的性能评价模型。针对物联网中可修复的服务组件，从平均失效前时间和平均修复时间两个角度，给出服务可靠性的量化表达，并以此为基础，预测物联网服务在边缘计算系统中的性能水平。

其次，建立物联网系统的服务模型，量化分析多属性的服务质量。根据边缘计算架构的层次化特征，建立分层的排队服务网络。在模型的基础上，量化表达服务的性能、能源效率，为后续的多属性服务质量优化提供量化依据。

1.2.2 任务调度与资源管理

任务调度、资源管理是服务质量优化的关键问题。任务调度是指当服务请求或任务到达时，如何将它们调度到合适的执行单元，以满足用户或服务提供商的需求。而资源管理是监控、维护系统资源的使用情况，对系统资源提供管理。任务调度与资源管理的过程中，涉及到服务质量的优化，其中服务质量往往是多属性的。例如，对于用户，通常关注响应时间、价格等指标；对于服务提供商，资源的利用率和开销等直接关乎其切身利益。

本文提出了基于性能与能源效率优化的任务调度和资源管理方法。在物联网系统的排队服务模型中，量化分析了服务性能、能源效率。在模型的基础上，采用马尔可夫决策过程对系统任务调度和资源管理进行建模，给出了性能与能源效率的联合优化决策算法。基于任务调度和资源管理的模型框架，引入序优化理论，压缩优化问题的决策空间，使提出的调度决策算法适用于大规模物联网系统。

1.3 研究难点与挑战

边缘计算架构下的物联网系统中，服务故障、服务质量的多指标属性、系统规模庞大给服务质量评价及优化的研究工作带来一系列的难点与挑战，具体表现为如下几点：

1.3.1 故障发生/修复给性能建模带来的挑战

在物联网系统中，各类设备接入网络连接，并参与交互通信。参与物联网的设备种类繁多，每类设备的容错性、可靠性表现良莠不齐，系统的可靠性更是难以预测估计。物联网服务发生故障之后，涉及的故障检测、错误修复、服务重启等一系列过程占用服务的响应时间。而传统的性能建模方法忽略故障的发生，不适用于故障发生/修复的情况。由此可见，考虑服务故障/修复的情况下，如何准确地对性能进行建模是研究工作的难点和重点。

1.3.2 多指标服务质量的折衷权衡

服务质量牵涉到多个指标维度，比如响应时间、吞吐率、能源效率、安全性、可靠性、经济指标等。在对服务质量的多指标优化过程中，部分指标的优化方向之间存在相互矛盾的情况。例如，性能与能源效率的联合优化过程中，如果性能提升，则导致能耗

降低。然而，性能优化的通用措施是提高硬件资源的投入，使系统以较高速度、频率状态运行，从而能耗上升。面对如上情况，性能与能源效率之间需要折衷与权衡，以最大程度地符合性能与能源效率的需求。同理，对于其他优化互斥的服务质量指标之间，也存在折衷与权衡的情况。单目标优化方法不能应用于多指标服务质量优化。因此，只能采用多目标优化方法，给多指标服务质量的综合优化带来挑战。

1.3.3 大规模系统特性给优化算法设计带来的挑战

随着物联网设备接入数量的剧增趋势，系统规模不断迅速增大。在大规模分布式系统上，操作任务调度和资源管理，以往基于状态的任务调度和资源管理方法面临决策空间爆炸(State Explosion)和维数灾难(Curse of Dimensionality)的瓶颈，调度执行效率低下，复杂度高，在大规模系统上难以适用。亟需提出新的方法来解决系统规模增大带来的研究挑战。

1.4 论文组织

本文内容分为5章，其余的4章组织如下：

第2章介绍相关研究现状。介绍了服务质量的评价方法、服务质量的多目标优化方法，并在章节末尾作了小结。

第3章研究边缘计算架构下物联网服务可靠性感知的性能评价。采用基于模型的评价方法，动态描述物联网系统的服务过程，给出服务的平均失效前时间、平均修复时间的量化表达，预测系统的性能指标。

第4章研究边缘计算架构下能效与性能优化的任务调度和资源管理。运用随机模型，分别描述边缘计算系统中终端设备、边缘/云服务器机群，整个系统的任务调度和资源管理过程。基于马尔可夫决策过程，提出能效与性能优化算法。借助序优化的基本思想，剪枝动作空间，解决了马尔科夫决策的空间爆炸问题，降低调度执行时间。

第5章总结全文的研究工作，同时对未来工作提供了研究方向。

第二章 相关研究综述

2.1 服务质量的评价方法

物联网服务的性能评价手段,大致分为三大类——测量方法、预测方法、模型方法。

2.1.1 基于测量的评价方法

在对物联网系统服务质量的评价中最直接的方式便是测量的方法。在基于测量的服务质量评估方法中,一般需要设计特定的硬件设备或计算机程序,安装或部署到实际运行系统或仿真器中,来最终测得服务质量指标。所有通过测量收集的服务质量数据都是真实的,因此基于测量的方法是最准确和直观的服务质量评估方法。

已经有不少研究工作专门用于物联网环境中的服务质量测量。为评测 OSGi 物联网框架, Stusek 等使用软硬件结合的测量方法进行了大量的实验工作^[18]。Chen 和 Kunz 设计并实现了硬件测量平台,并应用网络仿真器来分析多种物联网协议的服务质量^[19]。根据用户对云和多媒体服务质量评估的反馈结果, Wang 等深入研究了关于服务质量测量的数据可信度问题,并提高了 50% 的服务质量评估准确性^[20]。

2.1.2 基于预测的评价方法

随着物联网服务规模的迅速增长,对地理分布式服务进行服务质量测量并非易事。研究难点在于,在大规模系统上设计出对所有服务作服务质量进行测量评价的有效办法。因此,提出了基于预测的服务质量评估方法。通过调用服务的先前用户的历史性能数据,预测当前用户在特定服务上的服务质量^[21]。由于基于预测的方法允许数据丢失,因此该方法在性能评估中显示出强大的优势,特别是在大规模系统中。

在物联网的应用场景里,服务质量预测已经有了一些开创性的工作。Luo 等提出了一个数据驱动的评估方案,用于预测基于 Web 服务的大规模物联网系统中缺失的 QoS 指标值^[22]。具体研究方法为,采用核机器学习算法(Kernel Machine Learning Algorithm),分析已知的 QoS 数据,挖掘相应 QoS 数据之间存在最高相似度的隐藏关系,据此预测未知 QoS 指标值。此外, Tang 等人考虑了底层网络对服务质量的影响,并结合矩阵分解技术与网络映射技术,最终设计出 Web 服务的网络感知 QoS 预测方法^[23]。

2.1.3 基于模型的评价方法

虽然基于预测的评价方法在大规模系统中取得不错的应用效果,但实际系统中部分服务质量指标是在设计和实现系统的过程中已经被强制规定的。在该情景下,就无法通过预测的方法进行服务质量的评估。因此,需要根据系统结构和动态服务特性,设计出系统的理论模型,而不是基于历史测量数据。

为了应对这一挑战,一些研究人员提出了基于模型的服务质量评估方法。基本思想

是建立数学模型来描述系统或服务的动态过程,并定量分析模型参数对服务质量的影响。该方法的优点在于,可以节省在大规模系统中部署评价模块所产生的代价和开销。Matos 等应用马尔可夫链(Markov Chain),分析了 Web 服务的性能和可靠性,并且在模型的基础上设计了用于 QoS 感知服务组合的优化算法^[24]。Li 等采用 M/M/n 排队模型来预测物联网节点的服务时间,并设计了面向物联网服务的调度模型^[25]。Zhang 等提出了用于物联网异构网络性能评价的概率模型,并在此基础上深入研究了 QoS 的优化和配置^[26]。

2.2 服务质量的多目标优化方法

2.2.1 多目标聚集

多目标聚集(Multi-Objective Aggregation)是多目标优化问题中广泛采用的一种模型。通过对多个优化目标的加权聚合,实现所有优化目标的归一化。具体地说,忽略不同优化目标的单位和范围差异,通常同时给每个优化目标设置相应的权重,然后用一个效用函数(Utility Function)来代表所有的目标。而原多目标优化问题的最优解被定义为具有最优的效用函数值所对应的解。由此可见,原优化问题被转换为单目标优化问题。

多目标聚集模型在服务质量的多目标优化中被广泛使用。Sun 等提出了基于服务质量的服务聚合平台,通过选择合适的 OpenAPI 聚合,运用多目标聚集模型实现响应时间、可用性、价格等七个指标的综合优化^[27]。Zeng 等提出中间件平台,最大化价格、响应时间、可用性等五个 QoS 指标的加权效用函数^[28]。Qi 等提出基于服务质量的服务聚合平台,选择合适的 OpenAPI 聚合以实现功能,考虑了响应时间、可用性、价格等七个指标的综合优化^[29]。

2.2.2 ϵ -约束

ϵ -约束(ϵ -constraint)的基本思想是从 k 个优化的目标中选择一个目标作为最终的优化目标;而针对其他 $k-1$ 个目标,通过设置相应的约束,将其转换为约束条件。该模型的最优解是满足约束条件下,目标函数最优的解。选择哪个指标作为最终的优化目标,常常取决于用户或者服务提供商的喜好、不同指标间的优先级。

Tordsson 等探讨了多个服务供应商间的虚拟机部署问题,考虑了开销和响应时间的 QoS 指标,并采用 ϵ -约束模型,将最小化开销作为优化目标,而把响应时间作为约束条件^[30]。Bossche 等研究了云服务提供商的任务调度和外包问题,综合考虑了开销和服务质量多维指标,采用 ϵ -约束模型,将优化服务质量设置为约束条件^[31]。文献^[32]研究了弹性的服务供应与部署,考虑了收益和服务质量多维指标,最大化收益,同时将服务质量作为约束条件,并将服务等级协议考虑进优化问题。

2.2.3 帕累托最优

优胜关系是帕累托优化模型(Pareto Optimization Model)中的重要概念,对于任意两个可行解 x 和 x' ,若解 x 的所有优化目标解都不比解 x' 差,且解 x 有至少一个目标函数严格优于解 x' ,则称 x 解优胜(dominate)于解 x' 。一般而言,多个目标之间存在权衡和折衷的关系,难以同时达到最优。因此,在帕累托优化模型中,往往存在多于一个最优解,这些解组成的集合叫做帕累托集合。帕累托优化模型的目标是求解帕累托集合。

He 等研究了服务器速率的最优化工作,优化目标包括可靠性与开销,采用帕累托优化模型刻画优化问题,定义解的优胜关系和帕累托最优解^[33]。Xu 等研究了虚拟化数据中心中虚拟机的部署问题,考虑了系统利用率和能耗等多维指标,采用帕累托优化模型,优化目标包括最大化利用率、最小化能耗^[34]。Wagner 等研究了不同候选工作流下的服务组合问题,考虑了响应时间和开销多维指标,目标是最小化响应时间、最小化开销,并将帕累托最优解定义为最优解^[35]。

2.3 讨论与总结

为了保证物联网服务中端到端的服务质量,首先亟需提供一个准确的物联网服务/系统性能评价方法,以此作为服务等级协议(Service Level Agreement, SLA)^[36]的基本判定准则和服务质量优化的重要参考依据。基于模型的性能评价方法可以在搭建具体系统之前的系统设计阶段进行实施,所以相比基于测量、及预测的方法更加经济,尤其在大规模服务计算系统中更显优势。

虽然目前不少学者对服务质量的多目标优化进行了相关的研究,但是仍旧有补充的空间。不同服务质量的指标间相互影响,多目标优化中存在折衷与权衡的关系。伴随系统规模增大,优化决策空间呈现爆炸增长的趋势。这些问题和挑战都给服务质量的多目标优化研究提供了完善的空间。

第三章 可靠性感知的性能评价模型与分析

3.1 本章引论

伴随着越来越丰富的物联网服务应用在各个现实领域中，对物联网服务的性能评价方面的研究势在必行。由于对自适应传感和数据处理分析的实时性需求愈发强烈，所以一种称作“边缘计算”的新兴计算架构因应物联网的技术需求而诞生。它将部分物联网服务从中心处理节点迁移至物联网的逻辑边缘，进而保障了位于数据源处的即时数据处理和知识生成^[37]。综合云计算和移动计算技术，边缘计算在各类物联网服务或应用中广为流行^[38]。

为了提升服务/系统的可靠性和容错性，服务往往被部署在虚拟机或虚拟化的云服务器上。当错误或故障发生时，虚拟机应用迁移和修复技术来保证虚拟机无严重中断的快速重启。由于故障探测、错误修复、系统重启占用服务的响应时间，所以考虑故障发生/修复过程给性能评价带来挑战。基于 GSPN 模型，本章节将提出一种物联网服务的可靠性感知性能评价方法。

3.2 基于广义随机 Petri 模型的可靠性感知建模

本节将介绍物联网服务/系统的基本模型。物联网服务的动态行为通过广义随机 Petri 模型来刻画，并提供了服务可靠性、性能的定量分析方法。

3.2.1 机群模型

物联网服务代表服务提供者和服务使用者之间的交互或活动，以实现特定的业务目标或解决方案目标。在物联网系统中，不同种类的服务提供不同的功能。服务的动态特性体现在以下三个基本部分。首先，任务请求根据需要到达服务节点完成某些任务其次，由于服务节点上的资源不是无限的，所以这些任务请求有时必须在队列中等待，直到服务可用。否则，到达的任务请求会立即进入服务。第三，任务请求得到服务并最终离开系统。

传统 Petri 网模型通常被用于信息处理系统的描述，被广泛应用到很多实际计算机系统。广义随机 Petri 网模型(Generalized Stochastic Petri Net, GSPN)，不仅继承了原始 Petri 网模型的特性，而且被赋予随机控制的属性。GSPN 模型的定义如下：

$$GPSN = (P, T, F, W, \Pi, M_0, \lambda)$$

具体为，

- P : 库所(Place)的有限集合；
- $T = T_t \cup T_i$: 变迁(Transition)的有限集合。其中，延时变迁(Timed Transition)是 T_t ，而即时变迁(Immediate Transition)是 T_i ；
- $F \subseteq (P \times T) \cup (T \times P)$: 有向弧(Arc)的有限集合，由库所指向变迁或变迁指向库所；

- $W: F \rightarrow \mathbb{R}^+$ 弧的权重(Weight)函数;
- $\Pi: T \rightarrow [0,1]$ 变迁的优先级(Priority)函数,表示变迁的概率;
- M_0 : GSPN 网的初始状态;
- $\lambda: T_i \rightarrow \mathbb{R}^+$ 延时变迁的触发率(Firing Rate)集合。

利用 GSPN 模型技术来表述物联网系统的动态服务过程。其中, 变迁表示服务的基本活动和过程, 而被分配到库所的令牌(Token)代表系统中正在等待或被服务完成的任务请求。通过库所和变迁之间的有向弧, 将任务请求和各种类型的服务联系了起来。同时, 有向弧的权重 W 表示每次任务分派时允许调度的最大任务请求数。

在真实物联网系统的服务过程中, 任务请求通常在机群中被服务处理。为了描述机群的工作过程, 一个基于 GSPN 网的数学模型被建立, 如图 3-1 所示。

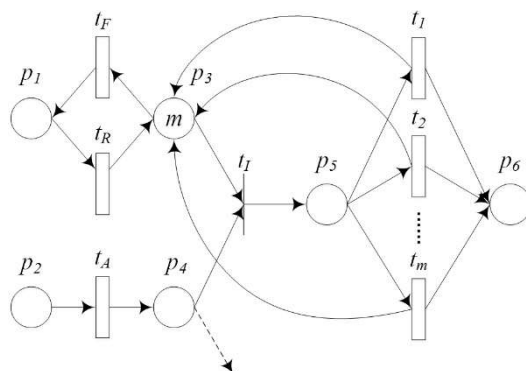


图 3-1 机群模型

图 3-1 中的循环 $p_3 \rightarrow t_F \rightarrow p_1 \rightarrow t_R \rightarrow p_3$ 象征机群中机器发生故障、修复的过程。不失一般性地, 设定机群中部署有 m 台服务器, 而且它们的服务过程分别由 t_1, t_2, \dots, t_m 表示。每台服务器的平均故障率是 λ_F , 而平均修复率是 λ_R 。延时变迁 t_A 描述任务的到达过程, 并且任务到达率设定为 λ_A 。停留在 p_3 的令牌表示空闲且能正常接受任务请求的服务器, 而 p_5 中的令牌用来代表已经接受任务请求、处于工作忙状态的服务器。 p_4 处引向机群以外的变迁(图 3-1 中虚线)表示任务上载至云端更高性能机群, 上载云端的迁移概率是 $1 - \pi_I$ 。对于云端机群, π_I 设置为 1。

根据 Little 公式(Little's Law), 机群平均任务处理时间由式(3-1)计算得到。其中, q_4 、 q_5 分别是 p_4 、 p_5 处的滞留令牌个数, $q_4 \cdot \pi_I + q_5$ 表征集群中任务队长。

$$RS = \frac{q_4 \cdot \pi_I + q_5}{\lambda_A \cdot \pi_I} \quad \text{式 (3-1)}$$

3.2.2 通信模型

在边缘计算架构下, 任务请求的初始化工作在边缘层完成, 而任务被选择在边缘层或云层被服务处理。如果任务请求被调度至云机群, 从边缘层到云层的跨层传输需要经过广域网(Wide Area Network, WAN)。通信模型如图 3-2 所示。通信网络带宽是有限的, 因而假定网管(Gateway)处缓存的最大任务请求实例数是 Q 。由于网络故障对整体通信性

能的影响，所以使用延时变迁 t_F 、 t_R 来表示网络故障的发生和修复过程。在图 3-2 中，变迁 t 象征边缘层、云层之间的数据传输过程；同时，库所 p_4 代表云端机群的任务到达。机群内服务器间通常以高速局域网(Local Area Network, LAN)实现通信连接，所以它们之间的通信延时在模型分析中忽略不计。

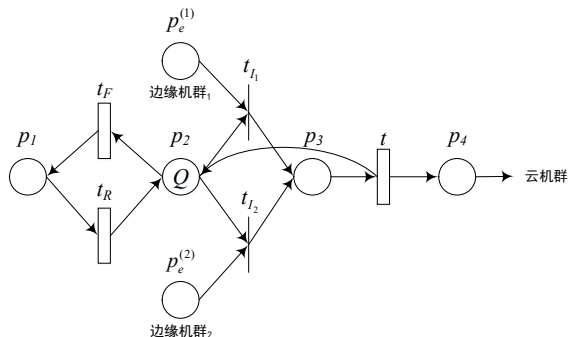


图 3-2 通信模型

基于如上的模型描述和分析，平均通信延时可以由式(3-2)表示，其中 i 是边缘机群的索引， q 是相应库所内的令牌数量。

$$TR = \frac{q_e^{(i)} \cdot (1 - \pi_l^{(i)}) + q_3}{\lambda_A^{(i)} \cdot (1 - \pi_l^{(i)})} \quad \text{式 (3-2)}$$

3.2.3 系统模型

根据边缘计算的架构特征，任务请求通常由物联网终端设备发起，并由上层处理节点服务。一些任务请求非常简单，可以在边缘层机群上完成，而其他任务请求则需要通过网络上传至云机群处理。

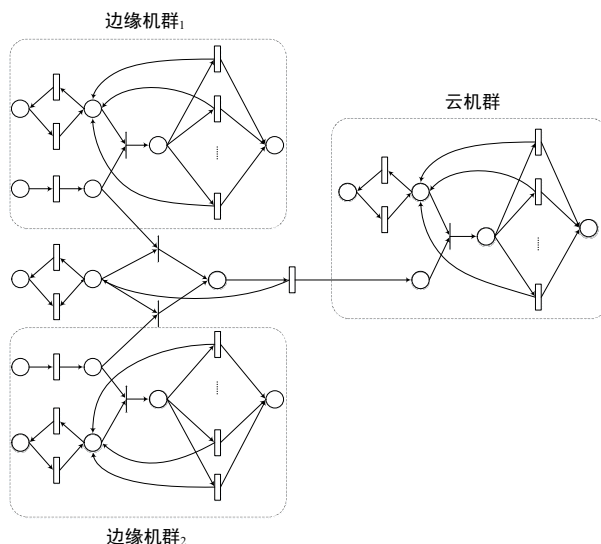


图 3-3 系统模型

基于 GSPN 网的机群模型、通信模型是系统模型的基本构件。在以上模型的基础上，边缘计算架构下的物联网系统模型被提出，如图 3-3 所示。系统模型由两个主要部分组

成，分别是边缘层和云层。任务请求从边缘层进入系统，其中，部分任务请求在边缘层被服务完成，剩下未处理的请求被上载至云层、被云机群服务。任务请求在云层被处理的比例由跨层传输概率定义。

假定任务请求在边缘层与云层间跨层传输概率为 $(1-\pi)$ ，那么系统跨层传输延时 TR 可以用式(3-2)计算得到。此外，边缘机群内服务平均响应时间 $RS^{(e)}$ 和云机群内服务平均响应时间 $RS^{(c)}$ 通过式(3-1)计算。因此，关于云机群内被服务请求的系统总平均响应时间由式(3-3)表示。因此，物联网系统的总平均响应时间由式(3-1)表示。

$$RS^{(c)} = ST^{(c)} + TR \quad \text{式 (3-3)}$$

$$RS^{(sys)} = \pi \cdot RS^{(e)} + (1-\pi) \cdot RS^{(c)} \quad \text{式 (3-1)}$$

3.3 实验验证

3.3.1 数据集简介

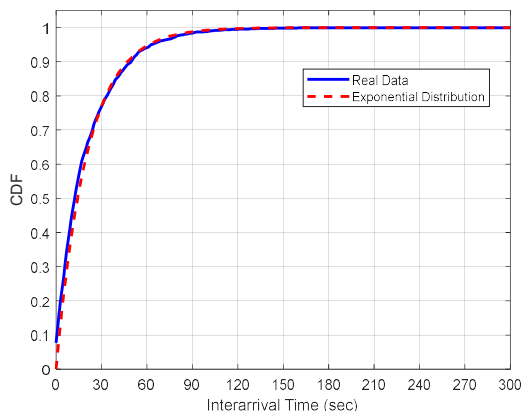


图 3-4 任务到达间隔时间的累积分布曲线

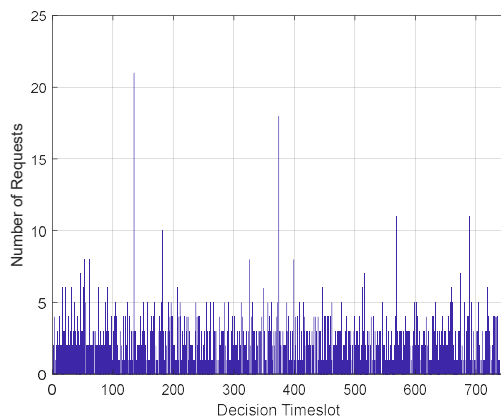


图 3-5 T-Drive 工作负载轨迹

T-Drive 数据集是由微软研究院发布的车联网数据集，数据由 GPS 系统收集提供^[39, 40]。该数据集中的数据由 GPS 车载记录仪或者车上具备 GPS 功能的移动电话采集。GPS 服务是指从地理分布式的 GPS 传感器收集大量 GPS 数据来计算被服务对象的位置。它属于典型的物联网应用，在实际中已被广泛应用。T-Drive 数据集包含了在 2008 年 1 周内 10357 辆北京出租车的 GPS 轨迹。共有近 1500 万条数据，车辆轨迹总距离达 900 万公里。除了详细的 GPS 信息（经度和纬度）之外，每条数据记录中还记录了时间戳。时间戳记录的时间是提交给系统的数据的确切时间，在实验验证中被用来作为系统任务到达的详细信息。任务请求的到达分布是建模和分析的关键因素。随机选择 10 辆出租车上传的时间戳数据，并作为系统任务请求到达时间。聚合分析任务到达信息，得到任务到达间隔时间的累积分布曲线，如图 3-4 所示。虽然不是一个典型的数学分布，但根据指数分布的拟合结果，任务到达可以近似地认为指数分布。图 3-5 是 T-Drive 工作负载随时间线的变化图。

各类故障率和修复时间从大规模系统故障日志中获取，系统日志则由 LANL 国家实

验室提供^[41]。该故障日志记录了来自 23 个高性能计算系统的 23739 条故障信息，时间跨度从 1996 年直至 2005 年。LANL 中这些系统发生的故障在日志中被分为六个类别，包括硬件故障(Hardware Failure)、软件故障(Software Failure)、设备故障(Facilities Failure)、网络故障(Network Failure)、人为失误(Human Error)和未知错误(Undetermined)。在本文可靠性感知的系统模型中，软件故障、硬件故障被应用到机群模型部分，而网络故障、修复被应用于通信模型的参数设定。

3.3.2 实验结果

基于来自真实物联网系统的开源数据集，本文搭建了物联网服务的模拟仿真环境。仿真环境是基于 GSPN 仿真软件“PIPE”搭建的^[42,43]。系统配置两个边缘机群、一个云机群，每个机群由 10 个并行服务器组成且共享任务请求的缓冲队列。实验假设延时变迁按照指数分布触发。

首先，通过调节机群内机器的 MTTF 和 MTTR，观察可靠性对系统的影响。如图 3-6 所示，机群资源利用率随平均失效前时间的增加而下降；平均修复时间的下降导致了资源利用率的降低。资源利用率的下降意味着更高的资源冗余度，也说明系统可靠性的加强。在相同任务到达的条件下，资源利用率下降有利于性能的改善。因此，减少故障率和加快故障修复过程有助于改善系统的可靠性和性能。

接着，使用平均响应时间来评价系统性能。图 3-7 展示了系统平均响应时间随机群服务率的变化情况。由图 3-7 可见，平均响应时间随着服务的增加而减小。同时，性能分析过程引入了软件故障、硬件故障。由于软件故障可及时修复，硬件故障一般需要更多的时间来修复。因此，从软件故障的实验结果看，平均响应时间比硬件故障下的平均响应时间小。由此得出如下结论，硬件故障比软件故障更难处理，硬件故障的发生往往对系统而言有灾难性的影响。

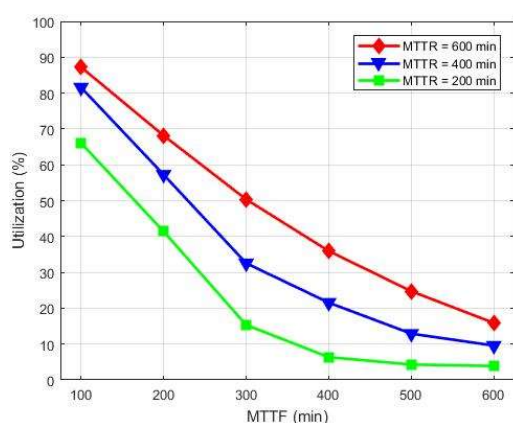


图 3-6 机群资源利用率随 MTTF 的变化情况

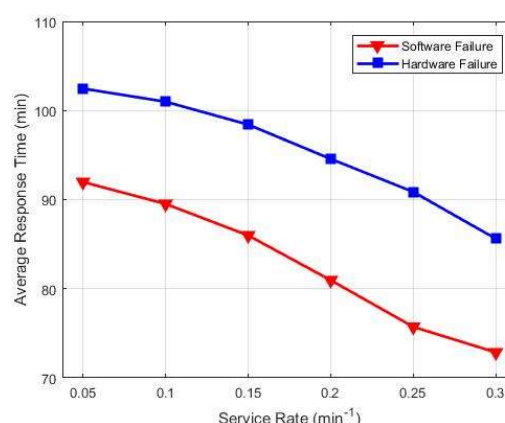


图 3-7 系统响应时间随处理机服务速率的变化情况

3.4 本章小结

在边缘计算架构下，本章提出了一种针对物联网服务的可靠性感知性能评价方法。

提出 GSPN 网模型, 用来描述物联网服务/系统的动态服务过程, 其中充分考虑了故障发生、错误修复的情况。基于模型, 机群子系统、通信子系统以及整个物联网系统的性能量化表达式被分别给出。最后, 通过实验, 评价模型的有效性得到验证。

第四章 性能与能效优化的任务调度与资源管理

4.1 本章引论

边缘计算架构将云中心的计算能力分散至各处边缘节点，并以分布式的方法管理各个边缘节点，实现任务的负载均衡，降低了服务的响应时间。除了边缘计算带来的性能提升外，能源效率也是一个被广泛讨论的话题。一方面，在分布式系统中，根据任务负载的变化情况，准确管理计算资源，已经被认可为一种提升能源效率的有效办法^[44]。另一方面，通过将计算密集型任务由计算能力偏弱的设备（如物联网终端，边缘服务器）上载至高性能云服务器，计算产生的能耗会降低^[45]。然而，任务上载期间，数据发送、传输、接收也会产生额外的能源消耗。由此可见，如何科学地在边缘层、云层之间做任务调度以实现能源高效是一个亟需探讨的科研议题。

在本章中，首先，给出边缘计算架构下物联网服务的随机服务模型，量化分析性能、能源效率指标。然后，在此基础上，设计性能与能效优化的任务调度与资源管理算法。

4.2 系统模型

4.2.1 终端设备

数据融合技术已经在基于传感器的边缘计算应用中被广泛采用^[46]。它整合来自多个不同传感器的多片数据，来为被测量目标提供更加准确、一致性的描述。为了实现数据融合，感测器中枢(Senor Hub)被部署在边缘计算系统前端。除了同步整合多元传感器数据，当任务空闲时，感测器中枢终止运行，节省能源^[47]。

从感测器中枢的内部结构看，首先，到达的任务和数据被内部缓存。直到等待处理的缓存数据达到某一阈值时，数据才会被感测器中枢内的微控制单元(Microcontroller Unit, MCU)集中处理^[48]。该批次数据被集中处理完成后，感测器中枢则再次积累任务和数，等候集中处理。如上过程以此类推，传感数据在感测器中枢内被批量处理。因此，感测器中枢处理传感数据的过程可以用具备批服务属性的 $M/M^{[b]}/1$ 排队模型分析描述，如图 4-1 所示。

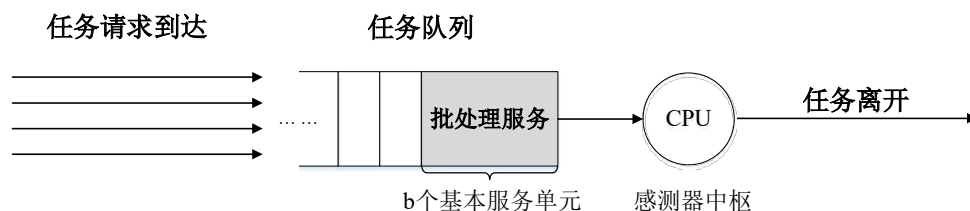


图 4-1 感测器中枢的批处理模型

已经有研究工作指出，会话层以上的任务到达可以近似表达为泊松分布^[49]。每个任务请求的服务时间假设服从指数分布。本文用 λ_i 表示单个任务到达速率； $\hat{\mu}_i$ 表示批服务速率。以批任务为单元的任务流可以近似认为是泊松到达的过程^[50]，并且批任务

的到达率可以用式(4-1)数学表示, 其中 b_i 为批处理粒度 (每次批处理的任务数)。感测器中枢的 MCU 占用率可以用式(4-2)计算得到。

$$\hat{\lambda}_i = \frac{\lambda_i}{b_i} \quad \text{式 (4-1)}$$

$$\rho_i = \frac{\hat{\lambda}_i}{\mu_i} \quad \text{式 (4-2)}$$

4.2.2 服务机群

边缘计算系统由计算处理能力相异的边缘服务器机群、云服务器机群组成, 并以良好的层次结构 (云核心服务器—地区分布边缘处理集群—物联网终端) 组合而成。边缘/云机群内的处理机之间通过局域网(Local Area Network, LAN)连接通信, 服务请求在它们之间调度。

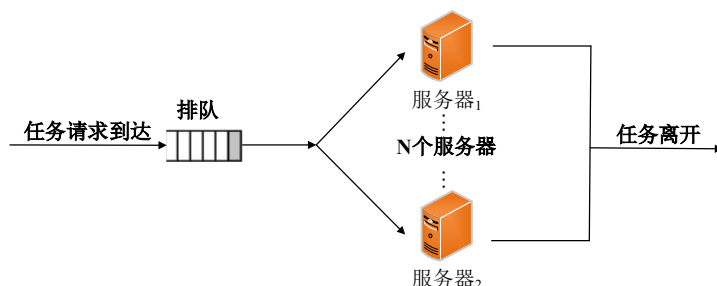


图 4-2 边缘/云服务器机群的并行计算模型

运用 $M/M/n$ 排队模型, 准确描述处理机间的计算任务分配和调度过程, 如图 4-2 所示。设定服务机群 j 有 N_j 个物理服务器组成, 其中 n_j 表示当前通电启动的服务器数 ($1 \leq n_j \leq N_j$)。服务机群 j 的任务到达率和服务率分别用 λ_j 、 μ_j 表示, 机群的利用率根据式(4-3)计算得出。

$$\rho_j = \frac{\lambda_j}{\mu_j} \quad \text{式 (4-3)}$$

4.2.3 系统架构模型

图 4-3 刻画了边缘计算系统的层次化结构。作为原始数据的来源和系统入口, 最外层是感测器中枢以及传感器。多个传感器的数据处理任务在最外层的感测器中枢处被批处理、整合为单个服务请求, 该单个服务请求提供了更精确的内容描述。下面一层则是边缘层(Edge Layer), 该层中边缘服务器机群与所属的感测器中枢和传感器发生通信连接, 并负责基本计算任务的处理。中央云服务器机群位于边缘计算系统的核心位置。

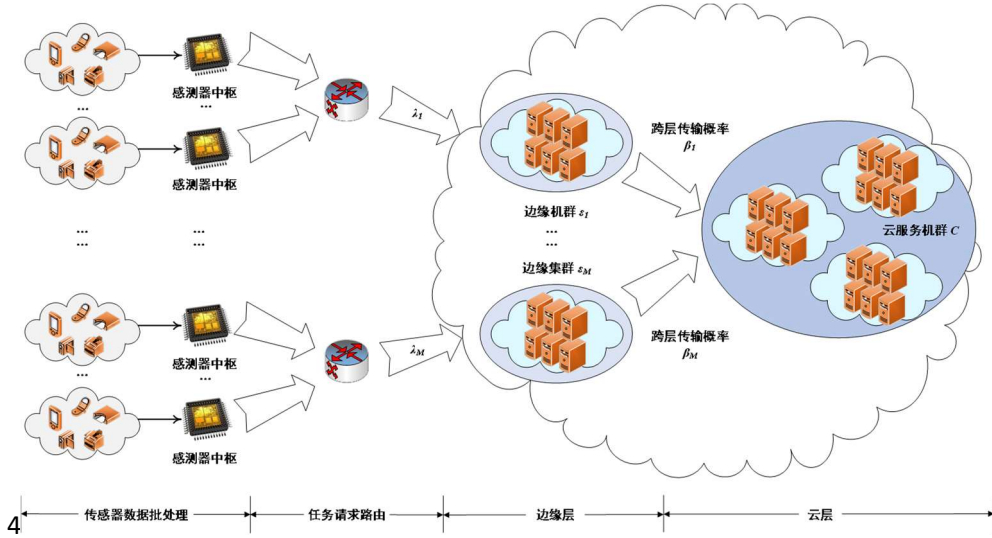


图 4-3 边缘计算系统架构概览

边缘计算系统中第 j 个边缘服务机群用 ε_j ($1 \leq j \leq M$) 表示，机群 ε_j 最多 N_j 个服务器同时运行， n_j 用来表示机群 ε_j 当前运行的服务器数 ($1 \leq n_j \leq N_j$)。 $H_i^{(\varepsilon_j)}$ ($1 \leq i \leq I_j$) 表示所属于边缘机群 ε_j 的第 i 个感测器中樞。中央云服务器机群用符号 c 标记，共有 N_c 个云服务器部署在机群内， n_c 表示中央云服务器机群 c 中正在运行的服务器数。

$\lambda_i^{(\varepsilon_j)}$ 、 $\hat{\mu}_i^{(\varepsilon_j)}$ 分别表示感测器中樞 $H_i^{(\varepsilon_j)}$ ($1 \leq i \leq I_j$) 任务请求的到达率、批处理速率。批任务的到达率用式(4-4)数学表示， $b_i^{(\varepsilon_j)}$ 是批处理粒度。边缘机群中单个处理机服务率是 μ_j ；云机群中处理机服务率表示为 μ_c 。

$$\hat{\lambda}_i^{(\varepsilon_j)} = \frac{\lambda_i^{(\varepsilon_j)}}{b_i^{(\varepsilon_j)}} \quad \text{式 (4-4)}$$

任务在边缘层和云层之间支配调度，用 $\beta_j \in (0,1)$ 表示边缘服务机群 ε_j 和云服务器机群 c 之间的任务跨层概率，而 D_j 表示边缘服务机群 ε_j 和云服务器机群 c 之间的传输距离。根据 Burke 定理(Burke Theorem)， $M/M^{[b]}/1$ 或 $M/M/n$ 的离开过程也是泊松的^[50]，并且任务的离开率等于到达率。因此，边缘机群、云机群的任务到达率可以分别用式(4-5)、式(4-6)表示。

$$\lambda_j = (1 - \beta_j) \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)} \quad \text{式 (4-5)}$$

$$\lambda_c = \beta_j \cdot \sum_{i=1}^{I_j} \hat{\lambda}_i^{(\varepsilon_j)} \quad \text{式 (4-6)}$$

4.3 模型分析与服务质量评价

4.3.1 性能指标

感测器中枢的批处理过程用 $M/M^{[b]}/1$ 排队模型描述。根据 Little 公式，感测器中枢批任务的平均响应时间用式(4-7)数学表示。其中， $E[q_i]$ 是平均任务队长，由式(4-8)得到。

$$T_i^{RS} = \frac{E[q_i]}{\hat{\lambda}_i} = \frac{b_i \cdot E[q_i]}{\lambda_i} \quad \text{式 (4-7)}$$

$$E[q_i] = \frac{\rho_i}{1 - \rho_i} \quad \text{式 (4-8)}$$

边缘/云服务机群的任务调度和资源管理过程由 $M/M/n$ 排队模型分析描述。同样地，根据 Little 公式，边缘/云服务 u 机群的任务平均响应时间用式(4-9)计算所得。其中， $E[q_i]$ 表示平均任务队长，由式(4-10)得到。

$$T_j^{RS} = \frac{E[q_j]}{\lambda_j} \quad \text{式 (4-9)}$$

$$E[q_j] = n_j \rho_j + \frac{\rho_j (n_j \rho_j)^{n_j}}{n_j! (1 - \rho_j)^2} \left[\sum_{k=0}^{n_j-1} \frac{(n_j \rho_j)^k}{k!} + \frac{(n_j \rho_j)^{n_j}}{n_j! (1 - \rho_j)} \right]^{-1} \quad \text{式 (4-10)}$$

在边缘计算系统中，感测器中枢和边缘/云服务机群的平均计算任务耗时分别用 $T_i^{(\varepsilon_j)}$ 、 T_j 或 T_c 表示。从边缘服务机群 ε_j 到云服务机群 c 的跨层数据传输时间用式(4-11)表示，其中， t 是单元传输距离的传输耗时。

$$T_j^{TR} = D_j \cdot t \quad \text{式 (4-11)}$$

4.3.2 能效指标

已经有研究指出^[51]，计算设备的能耗由两部分组成，分别为静态能耗、动态能耗。前者由漏电流产生耗电量，独立于 CPU 时钟频率和资源占用情况；而后者和设备内电路状态有关，主要和 CPU 时钟频率、占用率、I/O 状态有关。当机器处于休眠空闲状态时，相对于工作状态时产生的能耗，这时的能耗微小至忽略不计（e.g. 休眠耗电 0.5W，工作耗电 100.0W）。

在边缘计算架构下的物联网系统中，感测器中枢的耗电功率用式(4-12)表示。其中， σ_i 表示感测器中枢的休眠/工作状态（0 表示休眠，1 表示工作）， $p_i^{(static)}$ 、 $p_i^{(dyn)}$ 分别代表感测器中枢的静态功率、动态功率。

$$p_i = \sigma_i \left(p_i^{(static)} + \rho_i \cdot p_i^{(dyn)} \right) \quad \text{式 (4-12)}$$

同理，边缘/云服务机群的耗电功率用式(4-13)数学表示。其中， σ_k 表示边缘/云服务

机群中第 k 个处理机的休眠/工作状态 (0 表示休眠, 1 表示工作), $p_k^{(static)}$ 、 $p_k^{(dyn)}$ 分别代表相应的静态功率、动态功率。

$$p_j = \sum_{k=1}^{N_j} \left(\sigma_k \left(p_k^{(static)} + \rho_k \cdot p_k^{(dyn)} \right) \right) \quad \text{式 (4-13)}$$

在边缘计算系统中, 感测器中枢和边缘/云服务器机群的耗电功率分别用 $P_i^{(\epsilon_j)}$ 、 P_j 或 P_c 表示。通信网络中的数据传输主要分为两个部分^[51], 一是与通信信道选择、MAC 层控制相关的固定能耗开销, 二是由传输数据大小决定的能耗开销。所以, 从边缘服务机群 ϵ_j 到云服务器机群 c 的跨层数据传输产生的能耗功率可以由式(4-14)计算得到。其中, $p^{(dyn)}$ 、 $p^{(static)}$ 是常数。

$$P_j^{TR} = p^{(dyn)} \cdot \lambda_j \cdot \beta_j + p^{(static)} \quad \text{式 (4-14)}$$

4.4 性能与能效优化

4.4.1 回报模型

为了提出任务和资源管理的综合调度算法, 首先提出一个基于边缘计算系统的回报模型。由于感测器中枢的工作状态与其负责的特定任务属性有关、并且通常以一种稳定的周期工作状态运行, 所以感测器中枢的能耗和性能可以视为常量。鉴于以上论述, 本文的调度算法主要着眼于边缘层、云层之间的任务分配和资源管理。调度算法的优化目标之一是最小化硬件资源投入所致的能源消耗。如 4.3.2 节所述, 能源消耗分为两部分, 分别为边缘/云服务器机群计算任务产生的能耗开销、跨层数据传输的能耗支出。因此, 边缘计算系统的整体能耗支出可以被式(4-15)定义。

$$P^{sys} = \sum_{j=1}^M \left(P_j + P_j^{TR} \right) + P_c \quad \text{式 (4-15)}$$

另一个优化目标是满足响应时间的服务等级协议 SLA, 用 T_{SLA} 表示。在边缘层, 本文认为回报值与瞬时响应时间紧密相关。而在云层, 由于云通常部署有充分足够的计算资源, 并且面对动态变化的任务负载, 云的性能表现也相对稳定, 所以本文使用平均响应时间用来估计云层的响应时间。考虑到系统响应时间由机群计算任务、跨层传输任务两部分组成, 任务在系统中的响应时间可以用式(4-16)定义, 其中, q_j 是边缘集群 ϵ_j 的当前任务队列长度。对于响应时间超过 T_{SLA} 的情况, 将在回报模型中滤除。在本文中, 每个任务的 T_{SLA} 被假设为同构的。

$$T^{sys} = \sum_{j=1}^M \lambda_j \cdot \left(\frac{q_j \cdot (1 - \beta_j)}{n_j \cdot \mu_j} + \beta_j \cdot (T_j^{TR} + T_c) \right) \Bigg/ \sum_{j=1}^M \lambda_j \quad \text{式 (4-16)}$$

综上所述, 回报模型在能耗与响应时间约束的综合基础上建立。在边缘层、云层间调度, 跨层的任务传输概率、机群中工作机器数是决策变量。在回报函数中, 响应时间

和能耗支出被正则化表示，如式(4-17)所示。

$$R = \frac{T_{SLA} - T^{sys}}{T_{SLA}} \bigg/ \frac{P^{sys}}{P_{max}} \quad \text{式 (4-17)}$$

4.4.2 基于马尔可夫决策的优化框架

基于马尔可夫决策过程(Markov Decision Process, MDP)，本文提出基本的调度优化框，由以下五个元素详细定义：

(1) 决策时刻

决策发生在时间槽末尾，每个决策时间槽占用 τ 个时间单元。决策时刻序列表示为 $t = \tau, 2\tau, 3\tau, 4\tau, \dots$

(2) 状态空间

如 4.4.1 小节所述，假定云服务器具备充分的计算处理能力、性能稳定，因此决策动作在边缘层操作。状态 $S(n)$ 由 M 元组（见式(4-18)）定义，表征各边缘服务机群的负载情况。

$$q = (q_1, \dots, q_j, \dots, q_M) \quad \text{式 (4-18)}$$

(3) 决策空间

能源效率是通过两方面的措施来使之提升的，一方面是在边缘层和云层之间分配任务负载，另一方面是配置各机群内开/关机器数量。用跨层任务传输概率 β 表示两层间的负载支配划分情况，而 n 描述各机群内开/关机器数量。因此，决策变量 a_n 由 $\beta \times n$ 数学表示，而所有候选决策变量构成决策空间 $A(n)$ 。

$$\beta = (\beta_1, \dots, \beta_j, \dots, \beta_M) \quad \text{式 (4-19)}$$

$$n = (n_1, \dots, n_j, \dots, n_M, n_c) \quad \text{式 (4-20)}$$

$$A(n) = \{a_n \mid a_n \in n \times \beta\} \quad \text{式 (4-21)}$$

(4) 状态转移函数

虽然根据排队模型直接构建连续时间马尔可夫链(Continuous-Time Markov Chain, CTMC)具有精确的优势，但是对连续时间排队模型的分析 and 求解过程往往十分繁乱琐碎，从而不得不借助一些近似技术求解。因此，离散时间马尔可夫链(Discrete-Time Markov Chain, DTMC)被嵌入 CTMC。假设时间线被平均划分为若干时间长度为 τ 的时间槽，并且多个状态迁移不能同时触发。嵌入的 DTMC 具有如式(4-22)~式(4-24)的状态迁移函数。其中， $p_{q_j \rightarrow q_{j+1}}$ 、 $p_{q_j \rightarrow q_{j-1}}$ 代表一步状态转移概率， p_{loop} 是状态维持的概率表达式。

$$p_{q_j \rightarrow q_{j+1}} = \frac{\lambda_j \left(1 - e^{-\sum_{k=1}^M [(\lambda_k + q_k \mu_k) \tau]} \right) (1 - \beta_j)}{\sum_{k=1}^M (\lambda_k + q_k \mu_k)} \quad \text{式 (4-22)}$$

$$p_{q_j \rightarrow q_{j-1}} = \frac{q_j \mu_j \left(1 - e^{-\sum_{k=1}^M [(\lambda_k + q_k \mu_k) \cdot \tau]} \right)}{\sum_{k=1}^M (\lambda_k + q_k \mu_k)} \quad \text{式 (4-23)}$$

$$p_{loop} = e^{-\sum_{k=1}^M [(\lambda_k + q_k \mu_k) \cdot \tau]} + \frac{\lambda_j \beta_j \left(1 - e^{-\sum_{k=1}^M [(\lambda_k + q_k \mu_k) \cdot \tau]} \right)}{\sum_{k=1}^M (\lambda_k + q_k \mu_k)} \quad \text{式 (4-24)}$$

(5) 回报和值函数

每个决策时刻都对应一个回报函数 $r(S(n), a_n) = r_n$ 。回报函数的具体数学表达已经在 4.4.1 小节中详细定义（见式(4-17)）。用 π 表示决策时刻的决策策略，由决策空间 A 定义。决策过程旨在找出最优的决策策略 π^* ，使得总回报期望值最大。优化目标（值函数）由式(4-25)定义。其中 $\gamma \in (0, 1)$ 是折扣因子，表示未来回报值对总回报值的影响。

$$V^{\pi^*} = \max E^{\pi} \left\{ \lim_{N \rightarrow \infty} \sum_{n=1}^N \gamma^{n-1} \cdot r(S(n), a_n) \right\} \quad \text{式 (4-25)}$$

根据贝尔曼方程(Bellman Equations)的定义，优化目标（值函数）也能用式(4-26)表达。

$$V_n(S(n)) = \max_{a_n \in A(n)} \left\{ r(S(n), a_n) + \gamma \sum_{s' \in S} p(s' | S(n), a_n) V_{n+1}(s') \right\} \quad \text{式 (4-26)}$$

4.4.3 搜索空间压缩的调度算法

虽然采用传统的值迭代(Value Iteration)或策略迭代(Strategy Iteration)算法^[53]能够精确求解马尔可夫决策优化问题，但是在大规模系统中应用迭代算法会花费高昂的计算代价。然而，在许多实际场景中，得到“足够好(Good Enough)”的解已经满足应用要求。适度降低解的最优性要求，通常会使得所付出的求解代价陡然下降。与执行传统算法所花费的巨大计算开销相比，这是一条解决问题与挑战的途径。

引入序优化(Ordinal Optimization)的基本思想^[54]，本文提出一种搜索空间压缩的任务与资源调度算法。为了大幅降低算法的搜索空间和执行时间，通常引入一个粗粒度而高效的评价模型(Coarse Evaluation Model)，用来估计决策空间中各个候选调度方案的回报值。这里的决策空间由 4.4.2 小节中 MDP 的决策空间 A_n 定义。基于序效曲线(Ordered Performance Curve, OPC)和误差等级，从决策空间中挑选出 s 个调度方案以保证最低包括 k 个“足够好”的调度方案。粗粒度评价模型、挑选调度方案的方法将在如下部分详细介绍。

(1) 粗粒度评价模型

不同于细粒度评价模型（本文指 MDP 值函数 V^{π^*} ），粗粒度评价模型忽略了未来决策时刻的回报收益，仅着眼于即时回报。换言之，在每个决策时刻，采用回报函数（见式(4-17)）来估量候选调度方案的质量（回报值）。若回报值小于 0，粗粒度评价模型的估计值 r_e 置为 0。

(2) 拣选“足够好”的候选调度方案

设定从决策空间中拣选出 s 个的调度方案来保证其中“足够好”的调度方案数量不低于 k 。拣选完毕后， s 个“足够好”的调度方案被作为 MDP 的决策空间 A_n ，执行值/策略迭代算法、选出“足够好”调度方案中最优的调度方案。根据序优化理论， s 的取值和优化问题本身的性质（序效曲线和误差等级）有关。

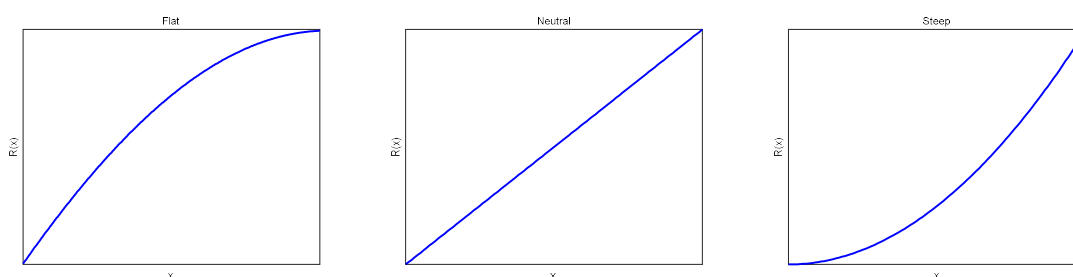


图 4-4 序效曲线的种类（平坦/温和/陡峭）

序效曲线用来描述所有调度方案的 r_e 分布状况，从而表征调度方案的整体质量水平（“足够好”的调度方案是否易于求解）。本文应用粗粒度模型的评价方法 r_e 来定义序效曲线的类型。假定决策空间 A_n 包含 \bar{N} 个候选调度方案，并且决策空间 A_n 中每个候选调度方案都通过粗粒度模型进行预评价。所有候选调度方案的预评价值以非降序的方式排列，用式(4-27)表示。非降序的序列通过式(4-28)在(0,1)范围内以正则化的形式表达。

序效曲线通常分为三种类型，分别是平坦类(Flat)、温和类(Neutral)、陡峭类(Steep)，如图 4-4 所示。平坦类的序效曲线表明获取“足够好”的调度方案难度较大。相反地，如果序效曲线属于陡峭类型，则说明决策空间中存在较多接近最优的调度方案，从而获得“足够好”的调度方案较为容易，变量 s 的取值可以偏小。

$$\text{非降序的序列: } r_e^{[1]}, r_e^{[2]}, \dots, r_e^{[\bar{N}]} \quad \text{式 (4-27)}$$

$$r(x_i) = \frac{r_e^{[i]} - r_e^{[1]}}{r_e^{[\bar{N}]} - r_e^{[1]}}, \quad x_i = \frac{i-1}{\bar{N}-1} \quad \text{对于 } i=1, \dots, \bar{N} \quad \text{式 (4-28)}$$

误差等级是决定变量 s 取值的另一重要因素。它表征估计值 r_e 和精确值 V^{π^*} 之间的分散程度。在 MDP 优化求解过程中，最优策略的选择主要取决于回报值之间的偏序关系，而回报值的具体数值对策略选取结果的影响较小。因此，本文中的误差等级是根据回报值的偏序误差(Partial Disorder)定义的。

决策空间中所有调度方案回报的估计值用长度为 \bar{N} 的向量 R_e 表示，而由向量 R 表示根据贝尔曼方程（见式(4-26)）计算得到的精确回报值向量， R_e 、 R 的具体数学表达如

式(4-29)、式(4-30)所示。每个调度方案 k 的误差等级 $diff_k$ 由 R_e 、 R 间的向量夹角测量得到，并以正则化方法表示，如式(4-31)所示。总误差等级 $error$ 由决策空间中 \bar{N} 个调度方案的误差等级间最大值定义，由式(4-31)表示。如果 $error < 0.5$ ，那么表示误差处于较低水平，变量 s 取较小值； $0.5 \leq error < 1.0$ 则表明误差处于良好中等水平； $error \geq 1.0$ 表示误差较大，变量 s 需要取较大值。

$$R_e = (r_e^{(1)}, r_e^{(2)}, \dots, r_e^{(\bar{N})}) \quad \text{式 (4-29)}$$

$$R = (r^{(1)}, r^{(2)}, \dots, r^{(\bar{N})}) \quad \text{式 (4-30)}$$

$$diff_k = 1 - \cos \langle R_e^{(k)}, R^{(k)} \rangle = 1 - \frac{R_e^{(k)\top} \cdot R^{(k)}}{\|R_e^{(k)\top}\| \times \|R^{(k)}\|} \quad \text{式 (4-31)}$$

$$error = \max_{1 \leq k \leq \bar{N}} \left\{ 1 - \frac{R_e^{(k)\top} \cdot R^{(k)}}{\|R_e^{(k)\top}\| \times \|R^{(k)}\|} \right\} \quad \text{式 (4-32)}$$

完成对序效曲线、误差等级的定义之后，再根据序优化技术，挑选出 s 个调度方案以保证最低包括 k 个“足够好”的调度方案。调正等级(Alignment Level) k 和“足够好”调度方案的数量 g 通常被自由定义，而 s 个调度方案对 k 个“足够好”的调度方案的命中概率(Alignment Probability) P_A 基于经验设定为 0.95。根据式(4-33)~式(4-36)，计算出调度方案的所需挑选数量 s ，其中参数 Z_0, ρ, γ 已经在文献^[55]中规定。定义式(4-33)~式(4-35)，将本文参数设定映射到文献^[55]的参数上来消除因决策空间规模差异所致的参数不一致性。

$$k' = \max \left\{ 1, \left\lceil \frac{10000 \cdot k}{\bar{N}} \right\rceil \right\} \quad \text{式 (4-33)}$$

$$g' = \max \left\{ 1, \left\lceil \frac{10000 \cdot g}{\bar{N}} \right\rceil \right\} \quad \text{式 (4-34)}$$

$$s'(k', g') = e^{Z_0} (k')^\rho (g')^\gamma + \eta \quad \text{式 (4-35)}$$

$$s = \left\lceil \frac{\bar{N} \cdot s'}{1000} \right\rceil \quad \text{式 (4-36)}$$

综上，本文提出边缘计算系统的任务与资源调度算法 (Task Scheduling and Resource Management of Edge Computing System, TRECS)，算法伪码详见算法 1。

算法 1 边缘计算系统的任务与资源调度算法 (TRECS)

输入: 由候选调度方案组成的决策空间 A_n , “足够好”调度方案的数量 g , 命中等级 k

输出: 决策调度方案 a_n

1: **for** 候选调度方案 $i \in A_n$ **do**

$$2: \quad r_e^{(i)} \leftarrow \frac{T_{SLA} - T_{(i)}^{sys}}{T_{SLA}} \bigg/ \frac{P_{(i)}^{sys}}{P_{max}};$$

3: **end for**

4: 对序列 $r_e^{(i)}$ ($\forall i = 1 \dots \bar{N}$) 非降序排列, 得到 $r_e^{[1]}, r_e^{[2]}, \dots, r_e^{[\bar{N}]}$;

5: **for** $i \in \{1 \dots \bar{N}\}$ **do**

$$6: \quad x_i \leftarrow \frac{i-1}{\bar{N}-1};$$

$$7: \quad r(x_i) \leftarrow \frac{r_e^{[i]} - r_e^{[1]}}{r_e^{[\bar{N}]} - r_e^{[1]}};$$

8: **end for**

9: 根据 x_i 、 $r(x_i)$, 评估序效曲线的类型

$$10: \quad error \leftarrow \max_{1 \leq k \leq \bar{N}} \left\{ 1 - \frac{R_e^{(k)T} \cdot R^{(k)}}{\|R_e^{(k)T}\| \times \|R^{(k)}\|} \right\};$$

$$11: \quad k' \leftarrow \max \left\{ 1, \left\lceil \frac{10000 \cdot k}{\bar{N}} \right\rceil \right\};$$

$$12: \quad g' \leftarrow \max \left\{ 1, \left\lceil \frac{10000 \cdot g}{\bar{N}} \right\rceil \right\};$$

$$13: \quad s' \leftarrow e^{Z_0} (k')^p (g')^y + \eta;$$

$$14: \quad s \leftarrow \left\lfloor \frac{\bar{N} \cdot s'}{1000} \right\rfloor;$$

15: 对 s 个拣选方案执行 MDP 值迭代或策略迭代算法, 求解决策调度方案 a_n ;

16: **return** a_n ;

4.5 实验验证

为了验证 TRECS 算法的有效性和高效性, 本文仿真实现一个边缘计算系统。边缘层部署有 2 个服务机群, 服务机群之下连接着若干感测器中枢, 而云层部署有一个云服务器机群, 并作为 2 个边缘服务机群在云层上的共同相联机群。任务负载根据 T-Drive 数据集^[39,40]随着时间戳的推进而生成。T-Drive 数据集的详细介绍见 3.3.1 小节。据分析, T-Drive 任务达到可以近似认为是泊松流 (详见图 3-4)。

图 4-5 展示了序效曲线。根据序效曲线的分类, 图中曲线属于 Steep 型, 表明易于从全体候选调度方案中拣选出“足够好”的调度方案, 仅需选出较少的调度方案便能保证最“足

够好”调度方案的最低数量。实验还估计了误差等级 $error$ ，根据式(4-32)得到的归一化误差

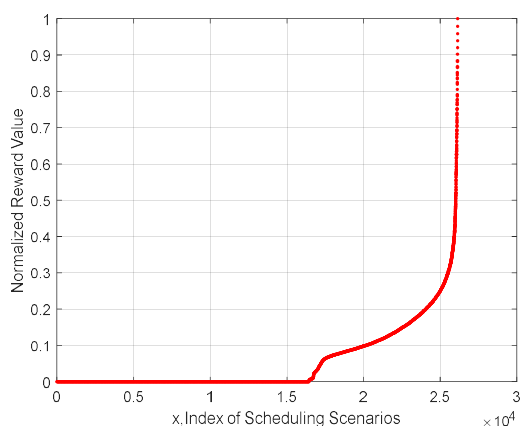


图 4-5 序效曲线

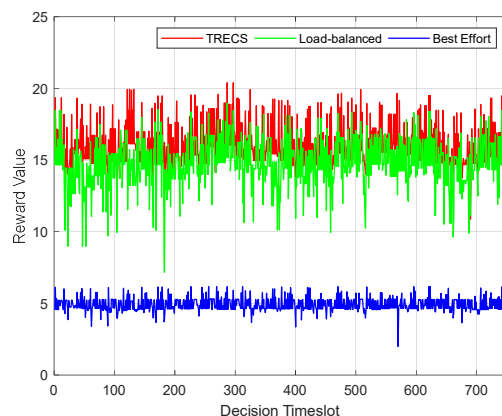


图 4-6 不同调度算法下的回报值比较

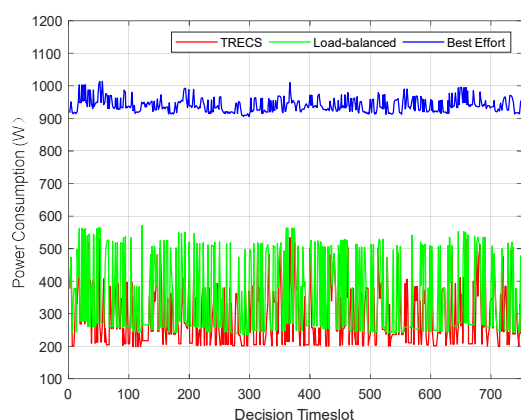


图 4-7 不同调度算法下的耗电情况比较

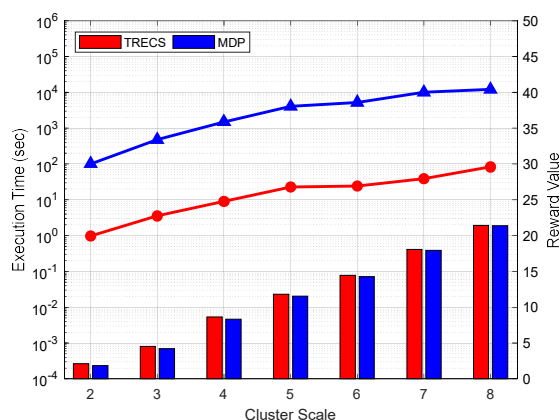


图 4-8 调度算法执行时间和回报值比较

表 4-1 不同调度算法下的平均响应时间比较

T_{SLA} (秒)	平均响应时间 (秒)		
	TRECS 算法	L-B 算法	B-E 算法
8.00	7.67	7.16	7.59

为 0.5789，对应于良好中等水平。序效曲线与误差等级都表明 TRECS 算法是有效性和高效性。

通过比较其他典型算法，评价 TRECS 算法的性能。两种典型调度算法被选用作对比分析。(1) **负载均衡算法(Load-balanced Algorithm, 简称 L-B 算法)**: 基于各机群的计算能力，服务被分配到边缘或云服务机群，而资源管理算法与 TRECS 算法一致；(2) **尽力而为算法(Best Effort Algorithm, 简称 B-E 算法)**: 只要机群任务队列长度不为空，全体机群服务器都保持工作状态、随时准备接受到达的服务请求，而任务调度算法与 TRECS 算法保持一致的策略。

图 4-6 比较了不同算法得到的回报值。由图可见，TRECS 算法获得了三种调度算法之间的最高回报值。因为 B-E 算法能源消耗最大、但三种调度算法之间性能(响应时间)最差，所以相对于 B-E 算法而言，L-B 算法具有更高的回报值。对能耗和响应时间的比较将在下面详细介绍。关于能源支出，图 4-7 比较了三种算法的能耗情况。TRECS 算法

是最节能的。由于 B-E 算法的资源管理策略是使所有机器运行工作，因此产生最大能耗。至于响应时间，三种算法均满足 SLA 的要求，三者之间仅略有差异。表 4-1 列出了三种调度算法下的平均响应时间，其中，TRECS 算法的响应时间最长。虽然 TRECS 牺牲了部分性能，却实现了最节能的效果。

此外，在回报值和算法执行时间两方面，将 TRECS 算法和传统 MDP 迭代算法作对照。图 4-8 比较了两种方法下的调度执行时间和奖励值。随着机群规模的扩大，两种方法的执行时间的增长速率优于指数增长。然而，两种方法之间的执行时间存在显著数量级的差异。其中，TRECS 算法的执行时间保持在在 100 秒以内，而传统的 MDP 迭代算法的执行求解时间远远超过 TRECS 算法。但是，两者的回报值差异非常小，故而 TRECS 算法可以认为是对传统 MDP 迭代算法的有效近似。

4.6 本章小结

本章研究了边缘计算架构下物联网系统中性能与能效效率综合优化的问题。首先，提出了一种基于模型评价的能效感知性能评价方法，给出性能、能源效率的量化表达。然后，提出综合资源管理和任务调度的 TRECS 算法，权衡了性能优化和能效优化之间的关系。为了使 TRECS 算法适用于大规模分布式边缘计算系统，引入序优化技术，大幅提高了算法的执行效率。最后，TRECS 算法的有效性和效率得到实验验证。

第五章 总结与展望

5.1 论文总结

随着人类社会步入万物互联时代，物联网技术的发展潜力巨大。与此同时，物联网系统规模不断扩大，服务数目迅速增长。因此，在比较不同服务/系统、有效设计新系统、改进已有系统等工作中，服务质量的评价和优化已经成为学术界和工业界共同关注的研究议题，尤其在大规模系统中更受重视。

边缘计算架构具有层次化结构，有助于大规模分布式系统实现负载均衡，宏观优化系统的体系结构。因此，为了更好地执行服务任务、优化服务质量，物联网系统通常基于边缘计算架构搭建。鉴于目前服务质量的多目标需求，本文探讨了多属性服务质量的评价及优化技术，主要创新性工作如下：

(1) 从可靠性、性能、能源效率出发，研究了边缘计算架构下物联网服务的服务质量评价方法。基于 GSPN 网模型，给出可靠性感知的性能评价模型，可靠性指标、性能指标得到定量分析。运用随机服务模型，对边缘计算架构下的物联网服务建模，提供服务性能、能源效率的量化分析。

(2) 基于边缘计算架构的层次化特征，设计了任务调度、资源管理的基本框架。运用马尔可夫决策过程，描述任务调度、资源管理的决策过程。任务请求在边缘层、云层间被动态支配调度，资源的工作状态伴随着任务负载变化而更新。期间，联合优化了服务质量的性能、能源效率指标。

(3) 针对马尔可夫决策过程中决策空间爆炸的问题，引入序优化技术，提高调度优化算法的执行效率。在不影响最终优化效果的前提下，适度降低决策调度方案的最优性要求，缩小了优化问题的决策空间，降低了算法的执行时间，最终使调度算法适用于大规模物联网系统。

(4) 在开源数据集的基础上进行仿真实验，验证了物联网服务质量评价模型和优化方法的有效性。

5.2 研究展望

本文围绕边缘计算架构下物联网服务质量评价与优化方法取得了一些成果，但是仍旧存在问题值得进一步完善和研究。具体从以下几个方面着手进一步开展研究：

(1) 加强关于更多种服务质量指标的评价及优化的研究。除了性能、可靠性、能源效率以外，服务质量指标还包括经济指标、安全性指标等。在未来的工作中，加强研究多属性指标之间的相互关系，提出综合的评价和优化方法。

(2) 本文的工作中，任务请求到达时间间隔、设备的故障频率和修复时间都假设服从特定的统计分布。在进一步的研究工作中，将去除上述假设，使物联网服务质量的评价及优化适用于更加普适的场景。

(3) 在具体的物联网应用场景中, 研究服务质量评价及优化。在具体应用背景下研究, 有利于更详细地评价服务质量, 提出更加务实的优化方法和建议, 对理论研究成果在现实系统中的验证有帮助。

参考文献

- [1] J. Gubbi, R. Buyya, S. Marusic, et al. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 29(7). 2013, 9: 1645–1660.
- [2] W. Yu, F. Liang, X. He, et al. A survey on the edge computing for the internet of things. *IEEE Access*. . 2018, 6: 6900-6919.
- [3] J. Lin, W. Yu, X. Yang, et al. A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. *IEEE Transactions on Vehicular Technology*. 66(3). 2017: 2551–2566.
- [4] N. Mohamed, J. Al-Jaroodi, I. Jawhar, et al. Smart City Ware: A service-oriented middleware for cloud and fog enabled smart city services. *IEEE Access*. 2017, 5: 17576–17588.
- [5] J.Lin, W.Yu, and X.Yang. Towards multistep electricity prices in smart grid electricity markets. *IEEE Transactions on Parallel and Distributed Systems*. 27(1). 2016: 286–302.
- [6] 国务院关于印发《中国制造 2025》的通知. [Online]. Available: http://www.gov.cn/zhengce/content/2015-05/19/content_9784.htm
- [7] J. A. Stankovic. Research directions for the Internet of Things. *IEEE Internet of Things Journal*. 1(1). 2014, 2: 3–9.
- [8] 施巍松, 孙辉, 曹杰等. 边缘计算: 万物互联时代新型计算模型.《计算机研究与发展》.54(5). 2017, 5: 907-924.
- [9] P. G. Lopez, A. Montesor, D. Epema, et al. Edge-centric computing: Vision and challenges. In *Proceedings of ACM SIGCOMM Computer Communication Review*. 2015: 37-42.
- [10] W. Shi, J. Cao, Q. Zhang, et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*. 3(5). 2016: 637-646.
- [11] W. Yu, H. Xu, H. Zhang, D. Griffith, et al. Ultra-dense networks: Survey of state of the art and future directions. In *Procings of 25th International Conference on Computer Communications and Networks (ICCCN)*. 2016: 1–10.
- [12] M. Agiwal, A. Roy, and N. Saxena. Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*. 18(3). 2016: 1617–1655.
- [13] P. Demestichas, A. Georgakopoulos, D. Karvounas, et al. 5G on the horizon: Key challenges for the radioaccess network. *IEEE Vehicular Technology Magazine*. 8(3). 2013, 9: 47–53.
- [14] A. Ahmedand, E. Ahmed. A survey on mobile edge computing. In *Proceedings of 10th International Conference on Intelligent System Control (ISCO)*. 2016: 1–8.
- [15] R. Deng, R. Lu, C. Lai, et al. Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption. *IEEE Internet of Things Journal*, 3 (6), 2017: 1171-1181.
- [16] R. D. Lauro, F. Lucarelli and R. Montella. SaaS - sensing instrument as a service using cloud computing to turn physical instrument into ubiquitous service. In *Proceedings of 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications (ISPA)*. 2012: 861–862.
- [17] Y. Jararweh, A. Doulat, O. AlQudah, et al. The future of mobile cloud computing: integrating cloudlets and mobile edge computing. In *Proceedings of the 23rd International Conference on Telecommunications (ICT)*. 2016: 1–5
- [18] M. Stusek, J. Hosek, D. Kovac, et al. Performance analysis of the OSGi-based IoT frameworks on restricted devices as enablers for connected home. In *Proceedings of the 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2015: 178–183.
- [19] Y. Chen and Kunz, T. Performance evaluation of IoT protocols under a constrained wireless access network. In *Proceedings of the 2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*. 2016: 1–7.
- [20] S. Wang, L. Sun, Q. Sun, et al. Reputation measurement of cloud services based on unstable feedback ratings. *International Journal of Web and Grid Services*. 11(4). 2015: 362–376.

- [21] S. Wang, Q. Sun and F. Yang. Towards web service selection based on QoS estimation. *International Journal of Web and Grid Services*. 6(4). 2010: 424–443.
- [22] X. Luo, J. Liu, D. Zhang, et al. A large-scale web QoS prediction scheme for the industrial Internet of Things based on a kernel machine learning algorithm. *Computer Networks*, 101. 2016: 81–89.
- [23] M. Tang, Z. Zheng, G. Kang, et al. Collaborative web service quality prediction via exploiting matrix factorization and network map. *IEEE Transactions on Network and Service Management*. 13(1). 2016: 126–137.
- [24] R. S. Matos, P. R. Maciel, and R. M. Silva. QoS-driven optimisation of composite web services: an approach based on GRASP and analytical models. *International Journal of Web and Grid Services*. 9(3). 2013: 304–321.
- [25] L. Li, S. Li, and S. Zhao. QoS-aware scheduling of services-oriented Internet of Things. *IEEE Transactions on Industrial Informatics*. 10(2). 2014: 1497–1505.
- [26] R. Zhang, M. Wang, X. Shen, et al. Probabilistic analysis on QoS provisioning for Internet of Things in LTE-A heterogeneous networks with partial spectrum usage. *IEEE Internet of Things Journal*. 3(3). 2016: 354–365.
- [27] L. Sun, S. Chen S, Q. Liang, et al. A QoS-based self-adaptive framework for OpenAPI. In *Proceedings of 2011 Seventh International Conference on Computational Intelligence and Security*. 2011: 204–208.
- [28] L. Zeng, B. Benatallah, A.H.H. Ngu, et al. QoS-aware middleware for web services composition. *IEEE Transactions on Software Engineering*. 30(5). 2004: 311–327.
- [29] L. Qi, Y. Tang, W. Dou, et al. Combining local optimization and enumeration for QoS-aware web service composition. In *Proceedings of 2010 IEEE International Conference on Web Services*. 2010: 34–41.
- [30] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, et al. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Generation Computer Systems*. 28(2). 2012: 358–367.
- [31] R. V. Bossche, K. Vanmechelen, and J. Broeckhove. Cost-optimal scheduling in hybrid IaaS clouds for deadline constrained workloads. In *Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing*. 2010: 228–235.
- [32] D. Breitgand, A. Maraschini, J. Tordsson. Policy-driven service placement optimization in federated clouds. *IBM Research Report*. 2011.
- [33] He, Wen P, Wu J, et al. Multi-objective service monitoring rate optimization using memetic algorithm. *Journal of Software*. 7(5). 2012: 990–997.
- [34] J. Xu, and J. A. B. Fortes. Multi-objective virtual machine placement in virtualized data center environments. In *Proceedings of 2010 ACM/IEEE International Conference on Green Computing and Communications (GreenCom)*. 2010: 179–188.
- [35] F. Wagner, A. Klein, B. Klopfer, et al. Multi-objective service composition with time- and input-dependent QoS. In *Proceedings of 2012 IEEE 19th International Conference on Web Services (ICWS)*. 2012: 234–241.
- [36] J. Yan, R. Kowalczyk, J. Lin, et al. Autonomous service level agreement negotiation for service composition provision. *Future Generation Computer Systems*. 23(6). 2007: 748–759.
- [37] J. Holler, V. Tsiatsis, C. Mulligan, et al. From machine-to-machine to the Internet of Things: Introduction to a new age of intelligence. *Academic Press*. 2014.
- [38] L.-J. Zhang, J. Zhang, and H. Cai. *Services Computing*. Springer Berlin Heidelberg. 2007.
- [39] J. Yuan, Y. Zheng, C. Zhang, et al. T-drive: Driving directions based on taxi trajectories. In *Proceedings of 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*. 2010: 99–108.
- [40] J. Yuan, Y. Zheng, X. Xie, et al. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2011: 316–324.

- [41] Los Alamos National Laboratory. All systems failure/interrupt data 1996-2005. [Online]. Available: <http://institute.lanl.gov/data/fdata/>, 2011.
- [42] N. J. Dingle, W. J. Knottenbelt, and T. Suto. Pipe2: A tool for the performance evaluation of generalised stochastic petri nets. *SIGMETRICS Performance Evaluation Review*. 36(4). 2009: 34–39.
- [43] P. Bonet, C. Llado, R. Puijaner, et al. Pipe v2.5: A petri net tool for performance modelling. In *Proceedings of the 23rd Latin American Conference on Informatics (CLEI)*, 2007: 1-12.
- [44] S. Liu, G. Quan, and S. Ren. Online real-time service allocation and scheduling for distributed data centers. In *Proceedings of IEEE International Conference on Services Computing*. 2011: 528–535.
- [45] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, et al. Fog of Everything: Energy-efficient networked computing architectures, research challenges, and a case study. *IEEE Access*. 2017, 5: 9882-9910.
- [46] X. Yang, R.-P. Yang, and D.-D. Wu. A resilient data fusion algorithm in wireless sensor networks. In *Proceedings of International Conference on Information System and Artificial Intelligence*. 2016: 192-195.
- [47] M. Hayashikoshi, H. Kawai, H. Ueki, et al. Normally-off MCU architecture and power management method for low-power sensor network. In *Proceedings of International SoC Design Conference*. 2015: 151-152.
- [48] T.-K. Chien, L.-Y. Chiou, S.-S. Sheu, et al. Low-power MCU with embedded ReRAM buffers as sensor hub for IoT applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. 6(2).2016: 247-257.
- [49] E. Chlebus and J. Brazier. Nonstationary Poisson modeling of web browsing session arrivals. *Information Processing Letters*. 102(5). 2007:187-190.
- [50] B. Gunter, G. Stefan, D. M. Hermann, et al. *Queueing networks and Markov chains - Modeling and performance evaluation with computer science applications*. Hoboken. John Wiley. 2005.
- [51] A. Beloglazov, R. Buyya, Y. Lee, et al. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*. 82(2). 2011: 47–111.
- [52] S. Lindsey, K. Sivalingam, and C. S. Raghavendra. *Handbook on wireless networks and mobile computing*. Hoboken. John Wiley. 2001.
- [53] C. G. Cassandras, S. Lafortune. *Introduction to discrete event systems*. Springer. 2008
- [54] T. W. Edward Lau and Y. C. Ho. Universal alignment probabilities and subset selection for ordinal optimization. *Journal of Optimization Theory and Applications*. 93(3). 1997: 455-489.
- [55] Y.-C. Ho, Q.-C. Zhao, and Q.-S. Jia. *Ordinal Optimization: Soft optimization for hard problems*. Berlin, Germany. Springer. 2008.

致 谢

衷心感谢我的指导老师黄霁崑讲师在科研工作中给予的悉心帮助和指导。从2016年起,我便在网络与交换技术国家重点实验室,跟随黄老师参与科研项目,至今已经两年。参与项目期间,黄老师引领我步入科学研究的神圣殿堂,他严格的治学操守、敏锐的洞察能力、严谨的思维逻辑都在不经意间深深地影响着我,使我终生受益。每当我在科研中遇到瓶颈时,黄老师总会耐心引导、为我解惑,展现出他广博的视野和学识,令我由衷的钦佩。在完成毕业设计的过程中,黄老师更是给予我悉心的帮助。无论是为人的方面,还是为学的方面,黄老师都是我学习的标杆和榜样。

感谢网络技术研究院的王雅文副教授、网络空间安全学院的石瑞生副教授,在我学习和科研道路中为我指点迷津,您们传授的科研通用方法对我日后的科研道路起到了巨大的帮助和指导作用,受益匪浅。

始终感谢我的家人对我的支持和鼓励,您们正直善良的高尚品德一直是我学习的楷模,您们的爱是我不断向前的动力源泉。

最后,向评审本文的专家、学者们致以诚挚谢意,感谢您们在百忙之中评阅我的毕业论文,并期待提供宝贵的建议。

在学期期间参加课题的研究成果

- [1] **Songyuan Li**, and Jiwei Huang. "GSPN-Based Reliability-Aware Performance Evaluation of IoT Services", *2017 IEEE 14th International Conference on Services Computing (SCC 2017)*, pp. 483-486, June 27-30, 2017, Honolulu, Hawaii, USA. (CCF 推荐 C 类会议, EI 收录, 检索号: 20174404327275)
- [2] **Songyuan Li**, and Jiwei Huang. "Energy Efficient Resource Management and Task Scheduling for IoT Services in Edge Computing Paradigm", *15th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2017)*, pp. 846-851, December 12-15, 2017, Guangzhou, China. (CCF 推荐 C 类会议, EI 收录, DOI: 10.1109/ISPA/IUCC.2017.00129)
- [3] Jiwei Huang, **Songyuan Li**, Ying Chen, and Junliang Chen. "Performance Modeling and Analysis for IoT Services", *International Journal of Web and Grid Services (IJWGS)*, vol. 14, no. 2, pp. 146-169, 2018. (2016 年影响因子: 1.105; SCI 收录, 检索号: 000429483400002; EI 收录, 检索号: 20181504996106; 第一作者为本人指导老师)

外文资料

Edge-centric Computing: Vision and Challenges

Pedro Garcia Lopez
Universitat Rovira i Virgili

Alberto Montresor
University of Trento

Dick Epema
Delft University of Technology

Anwitaman Datta
Nanyang Technological University

Teruo Higashino
Osaka University

Adriana Iamnitchi
University of South Florida

Marinho Barcellos
Universidade do Vale do Rio dos Sinos

Pascal Felber
University of Neuchatel

Etienne Riviere
University of Neuchatel

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

In many aspects of human activity, there has been a continuous struggle between the forces of centralization and decentralization. Computing exhibits the same phenomenon; we have gone from mainframes to PCs and local networks in the past, and over the last decade we have seen a centralization and consolidation of services and applications in data centers and clouds. We position that a new shift is necessary. Technological advances such as powerful dedicated connection boxes deployed in most homes, high capacity mobile end-user devices and powerful wireless networks, along with growing user concerns about trust, privacy, and autonomy requires taking the control of computing applications, data, and services away from some central nodes (the “core”) to the other logical extreme (the “edge”) of the Internet. We also position that this development can help blurring the boundary between man and machine, and embrace social computing in which humans are part of the computation and decision making loop, resulting in a human-centered system design. We refer to this vision of human-centered edge-device based computing as *Edge-centric Computing*. We elaborate in this position paper on this vision and present the research challenges associated with its implementation.

1. INTRODUCTION

In many areas of human society, there is a recurrent struggle between the forces of centralization and the forces of decentralization. In federal states, power may shift back and forth between the federal government and the constituent states. Energy generation was first concentrated in large power plants but is now moving to decentralized power grids.

In computing, we have witnessed similar shifts between centralized and decentralized control. In the 1980s a wave of decentralization led to a shift away from centralized mainframes to PCs and local networks, which culminated in fully decentralized systems using peer-to-peer and autonomous computing approaches.

Recent years have seen a proliferation of powerful computing devices at the user-facing end of the Internet. High capacity mobile devices, always-on and dedicated Internet connection boxes and home routers, or high-bandwidth pervasive wireless networks are prominent examples. We also faced simultaneously an important wave of centralization. The control, data and intelligence of computing systems

moved back to the cloud, dematerialized but nonetheless centralized computing systems.

Clearly, cloud computing with the enormous capacities of its dedicated data centers and the use of simple centralized architectures creates effective economies of scale. However, we believe that when pushed to such a logical extreme, full centralization brings more harm than good in several ways. The first fundamental problem is the loss of privacy by releasing personal and social data to centralized services such as e-commerce sites, rating services, search engines, social networks, and location services. A second fundamental problem is the complete delegation of the applications and systems control from the users to the cloud, which requires unilateral trust from clients to the clouds and prevents establishing finer grain trust between users. Third, there is the missed opportunity of exploiting the enormous amount of computational, communication, and storage power of modern personal devices. Finally, centralization hampers novel human-centered designs that would allow blurring the boundaries between man and machine and emerge novel applications.

We position in this paper that the advent of clouds should not be the final paradigm shift, and that a new decentralization wave is necessary. We advocate for *Edge-centric Computing* as a novel paradigm that will push the frontier of computing applications, data, and services away from centralized nodes to the periphery of the network. We position that this paradigm will retain core advantages of using clouds as a support infrastructure but will put back the control and trust decisions to the edges and allow for novel, human-centered computing applications.

We consider a node-oriented view of the Internet consisting of data centers and clouds at the *core* as illustrated in Figure 1. Surrounding this core are smaller web servers and content distribution networks as the next layer, which is in turn followed by the “edge” consisting of individual human-controlled devices such as desktop PCs, tablets, smart phones, and nano data centers (stable computing devices such as routers or media centers). The next layer of IP-enabled sensors and embedded processors is ignored in the context of this paper, as we focus on human-operated devices. Note that this view of the Internet stands in contrast to a network-oriented view in which the network itself is regarded as the core, and all computing devices and systems small and large are considered to be edge devices.

Edge-centric Computing encompasses the following ele-

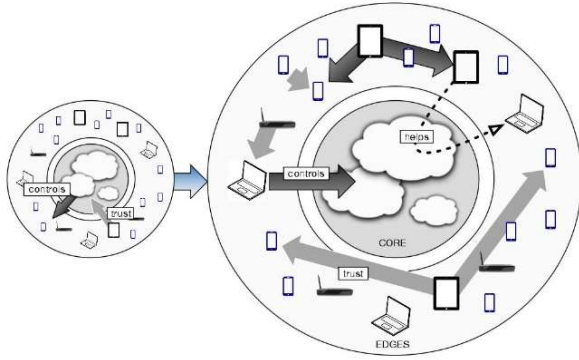


Figure 1: Centralized cloud model (left) versus Edge-centric Computing (right).

ments:

- *Proximity is in the edge*: This is the old but still valid argument of peer-to-peer (P2P) systems and content distribution networks (CDNs). It is more efficient to communicate and distribute information between close-by nodes than to use far-away centralized intermediaries. Here, “close-by” can be understood both in a physical and a logical sense.
- *Intelligence is in the edge*: As miniaturization still continues and computing capacity still increases, edge sensors and devices become more powerful. This opens the way to autonomous decision-making in the edge such as novel distributed crowdsensing applications, but also human-controlled actuators or agents reacting to the incoming information flows.
- *Trust is in the edge*: Personal and social sensitive data is clearly located in the edge. The control of trust relation and the management of sensitive information flows in a secure and private way must therefore also belong to the edges.
- *Control is in the edge*: The management of the application and the coordination also comes from the edge machines that can assign or delegate computation, synchronization or storage to other nodes or to the core selectively.
- *Humans are in the edge*: Human-centered designs should put humans in the control loop, so that users can retake control of their information. This should lead to the design of novel crowdsourced and socially informed architectures where users control the links of their networks. Finally, it also opens opportunities for novel and innovative forms of human-centered applications.

We do not see Edge-centric Computing as only implying purely decentralized or P2P systems. An Edge-centric Computing architecture may consist of a federation of edge-centric distributed services deployed across data centers and nano data centers, and accessible from edge devices. Furthermore, following the decentralized nature of Internet services such as e-mail, hybrid edge services may be deployed by different vendors and be able to talk to each other. We foresee interesting scenarios where Edge-centric Computing services may be the natural decentralized evolution of a variety of Personal and Social communication and storage services.

2. RELATED FIELDS

Content Delivery Networks: The term Edge Computing was coined around 2002 and it was mainly associated with the deployment of applications over CDNs, when some large companies announced deals to distribute software through CDN edge servers. The main objective of this approach was to benefit from the proximity and resources of CDN edge servers to achieve massive scalability. In this early flavor of Edge Computing, the “edge” was restricted to CDN servers distributed around the world. This architectural model was studied and extended by several researchers, notably for deploying and replicating applications in CDNs [8].

Our vision of Edge-centric Computing goes far beyond this initial approach linked to CDNs. In our view, the edge is not restricted to CDN nodes but it can also include the myriad of user devices and sensors that are at the periphery of the network. Furthermore, we consider additional aspects beyond just proximity, by also taking into account trust, intelligence, and humans.

P2P: P2P computing is not only a field closely related to edge computing, it is also its main precursor. The term P2P was first introduced around 2000 with the appearance of popular file-sharing systems such as Napster and Kazaa. Since then, it has grown to be an important subfield of distributed systems, where decentralization, extreme scalability, tolerance to high levels of churn, and protection against malicious behavior have been major topics of research. Among the main achievements of the field one can mention *distributed hash tables* that later evolved in the more general paradigm of distributed key-value store in cloud computing; *generalized gossip protocols* that have been successfully used for complex tasks beyond simple information diffusion, e.g., data aggregation and topology management; or *multimedia streaming*, in the form of video on-demand, live TV, person-to-person communication, etc.

Unfortunately, the P2P term has always been tainted by its use for illegal file sharing and the wide media coverage of the associated prosecution and lawsuits. As a consequence, a number of commercial technologies that are actually based on the P2P paradigm do not acknowledge it (e.g., Akamai’s NetSession interface).

The edge-centric computing paradigm originates from P2P but expands to new avenues. It avoids the naive pursuit of the “decentralization myth” that considers decentralization as a cure-all. Instead, it extends the concept of peer to all the devices at the edge of the Internet, and blends P2P computing with the cloud.

Decentralized Cloud Architectures: Cloud computing is a naturally centralized paradigm, with storage and processing resources hosted within large data centers. Nevertheless, there have been many efforts in recent years to combine P2P and Cloud computing architectures. On the one hand, Cloud services can strengthen P2P systems by providing them with stable resources when necessary, e.g., when facing high churn or sub-critical peer populations. On the other hand, P2P can reduce the operating costs of Cloud services by contributing additional resources, and they can enhance them by providing geographical diversity and proximity to customers.

Along these lines, various peer-assisted [10] services have recently emerged, combining peer and cloud resources in hybrid architectures. For example, researchers have shown that a hybrid architecture where resources at the peers (band-

width, storage) are complemented with temporary usage of Cloud storage services can perform comparably to traditional client-server architectures but at a fraction of the costs [10].

Another interesting line of research is the use of relatively stable peer resources to build nano data centers [5], micro clouds, community clouds, or edge clouds [9]. For example, in [5] all the home appliances are controlled and managed centrally by the telecommunication provider. In contrast, our vision of edge-centric computing systems is user-centric and the control comes from the edges towards the core, not the other way around.

Fog Computing: Fog Computing is a recent research field that has substantial overlap with Edge-centric Computing. As defined by CISCO [2], “*Fog Computing is a paradigm that extends Cloud computing and services to the edge of the network.*” Proximity to end-users, dense geographical distribution, and support for mobility are the main distinguishing characteristics of Fog Computing.

Fog Services [1] may be hosted by the network, or even in end devices such as set-top-boxes or access points. The major benefit is the combination of proximity with intelligence in the edge to obtain real-time or predictable latency for a number of applications. Fog Computing is thus well positioned for real time data processing and analytics.

Finally, in the same line that Fog Computing, [7] propose an open application model based on swarmlets to bridge the gap between cyber-physical systems (sensors, actuators) and the Cloud benefiting from proximity and intelligence in the edge. Again, our vision of edge-centric computing is more focused on human-driven applications controlled from the edges of the network.

3. RESEARCH CHALLENGES

3.1 Human-driven distributed systems

The defining aspect of Edge-centric Computing is the key role of humans. Human-centered designs should put users in the control loop, so that they can retake control of their information. The massive proliferation of personal computing devices is opening new human-centered designs that blur the boundaries between man and machine.

Employing powerful capability of mobile devices such as smartphones has become a promising approach for large-scale environmental and human-behavioral sensing. Several techniques for mobile phone sensing [6] and opportunistic sensing [3, 4] have been proposed.

This should lead to the design of novel socially-informed architectures where users control the information provided or aggregated in a secure way. There is an important research challenge in designing novel safe methods for including humans in the data-analysis loop through means such as crowdsensing.

Users acting as sensors may create enormous flows of useful information in the context of the Internet of Things. Humans then become an important source of training data for learning algorithms, data analytics and visualization tools.

Classical centralized architectures to such crowd-sensing and crowdsourcing information may entail strong privacy risks. An important challenge is thus to design secure and sensitivity-aware edge big data analytics systems respecting users privacy. There are strong ethical issues related to centrally monitoring edge users. Edge-centric Computing can provide the platform to get the services without paying

the costs for aggregated personal information.

Finally, the analysis of human activity and their interactions with physical and digital artifacts will also be extremely useful for closing the control loop of adaptive distributed systems. This may open a new research playground for distributed systems that adapt to user behaviors in different contexts.

3.2 Edge Architectures and Middleware

An important difference with P2P approaches is that these new architectures may rely on novel edge-centric distributed services deployed in data centers. Novel standard distributed services must be created for rendezvous, communication, computation, content distribution and storage for edge nodes. These services should enable the distribution of applications across datacenters and edge devices, while ensuring end-user control and privacy. Novel programming abstractions and middleware for Edge-centric Computing applications and services will be required as well.

Edge-centric Computing goes beyond the hybrid cloud model where one part is trusted and the public one is not. Edge-centric Computing is based on a decentralized model that interconnects heterogeneous cloud resources controlled by a variety of entities. Novel combinations of overlay technologies with cloud resources may open new research possibilities.

Another important difference is that the inherent nature of churn and transient availability of P2P may be overcome by the reliance on stable resources for edge applications. This will naturally allow for novel design alternatives that have not been previously addressed in the P2P community.

Finally, an important challenge for edge architectures will be to find the correct tradeoffs between mobile terminals and cloud servers. Minimizing computation and battery exhaustion in mobile terminals while ensuring privacy and security will represent novel and interesting research challenges.

3.3 Security and Privacy

Edge-centric Computing goes beyond previous attempts on using E2E (End-to-End) encryption and user-centric privacy systems that try to protect users information in the cloud. Edge-centric architectures will challenge researchers in new ways. Beyond encryption to protect private information, more secure proxies will be needed for rendezvous, communication, and access control using different techniques like re-encryption or attribute-based encryption among others. Furthermore, novel secure middleware for privacy-aware information sharing must be created to boost edge-centric systems.

Many existing works on cloud security such as encrypted data stores, queries over encrypted data, homomorphic systems could contribute to the creation of novel edge-centric services. An important difference with traditional cloud security research is that Edge-centric Computing may assume the existence of trusted, or partially trusted, stable resources performing some communication, persistence, queries, and even computation for applications deployed and controlled in the edges. Edge-centric computing may also consider the coexistence of trusted nodes with malicious ones in distributed edge-based overlays. This will again require secure routing, redundant routing, trust topologies and previous P2P research applied to this novel setting.

Finally, another key difference is that Edge-centric Com-

puting prevents the concentration of information as compared to centralized computing. Previous cloud security research on fragmentation of information combined with encryption may converge with decentralized overlay technologies to ensure appropriate data protection for sensitive data. Furthermore, secure cloud queries and computation over fragmented data and indexes in overlay networks may create entirely new models respecting the privacy of sensitive information.

3.4 Scalability

Scalability is a recurring research challenge both in peer-to-peer and cloud computing settings. The design of architectures that scale to millions of users must take into account issues like fault-tolerance, churn, elasticity and many others. In P2P, churn and dynamism complicate the feasibility of these architectures and their overall service availability. In cloud computing, scaling and elasticity are recurrent topics, and even major cloud providers may be overcome by massive denial of service attacks.

Edge-centric Computing, however, changes completely the scalability challenges presented before. Churn is not such a limiting factor anymore, thanks to the use of stable cloud resources. A major challenge is the correct tradeoff between computing and communication responsibilities between edge devices, trusted servers and untrusted services.

Given that the control is in the edges, scaling problems are still very relevant. Building massive overlays combining mobile devices with limited batteries with stable cloud resources require special attention for communication protocols among nodes. Furthermore, cloud edge services must also be efficient and take into account the heterogeneous nodes they must be serving.

Another research challenge is the combination of scalability with security in massive overlays. Edge architectures requiring security will impose non-negligible overheads due to encryption, that must be dealt with to provide scalability.

4. SCENARIOS

4.1 Personal Spaces in the Edge

Our digital life is now scattered among a myriad of devices and applications in the Cloud. We have files in Dropbox, our email in Gmail, selected photos in Instagram, our work contacts in LinkedIn, and our social network in Facebook. And the rest of our information, such as work data and personal data (photos, videos, finance data, and health data), is spread on hard disks and a variety of user devices.

In the next years, Personal Information Spaces will emerge to unify the multiple flows of our entire digital life. All our personal information will be stored in the Cloud, and we will have mechanisms to let third-party applications access part of our data repositories. In this context, Edge-centric Computing can offer:

Trust and Control: A strong challenge of future Personal Information Spaces is privacy and user-control of their own information. In the next years, Edge-centric Computing will enable a novel generation of user-centric Personal Spaces where users will be able to decide which parts of their information silos can be accessed by third-party applications, but also by third-party users. This requires the design of novel architectures offering controlled privacy-aware data sharing and advanced access control mechanisms.

User information may be stored in Cloud providers, but

with encryption and privacy guarantees that will ensure that the Cloud provider cannot access users data without permission. Furthermore, secure queries should enable users to look for data in their Personal Information Spaces without the cloud provider being able to infer information about them (blind servers).

Another key aspect is trust in other users or entities that may establish different kinds of social links. This is essential for sharing information with others and for collaborative interactions between participants. These connections can be permanent or spontaneous. An example of permanent connection is members of a family sharing their photos, videos, songs, books, apps and other purchased digital content. An example of spontaneous collaboration is for example the transient overlapping of two Personal Spaces to share some information at some time.

Humans: Human-centered Personal computing is here to stay since we are surrounded by connected devices. This is in line with existing research efforts in Pervasive Computing, invisible computing, ubiquitous systems and augmented reality interfaces.

Edge-centric Computing architectures will produce distributed systems that adapt to user behaviors depending on their location or context. It will also handle the interaction with other humans through their available connected devices. Every human will carry multiple mobile devices (phones and wristwatches) and sensors (such as bands and implanted devices). These devices may obtain information from their owner (health, sensory), from other close-by devices in their location, from other close-by devices from other users, and from remote links through the Internet.

Furthermore, users may participate in secure distributed crowdsourcing platforms where they provide part of their selected personal data to external analytic systems. Imagine a user letting a third company access their energy usage at home to optimize her bill. The design of such infrastructures will pose serious challenges to distributed systems and security researchers.

Proximity and Intelligence: Our Personal Space must adapt to our current location: at home, at work, in the car, walking in the street, in a mall, in an airport, etc. An important aspect of the design of these edge distributed architectures is that they will be decentralized, and that the different information flows will belong to heterogeneous services and entities. Interaction, synchronization and content distribution that benefit from proximity will play important roles in the design of such systems.

Edge-centric Computing may also become a key facilitator for the deployment of personal agents and multi-agent systems in a variety of scenarios. Agents may receive flows of information from external entities and even react to these flows. Edge-centric Computing platforms may provide the needed communication, discovery and trust platforms for the deployment of agents.

There are a lot of research challenges involved in the paradigm shift towards more edge-centric autonomous agents. Whereas current centralized models limit the possibilities of agents, placing trust in the edges may facilitate the necessary peer interactions among agents.

4.2 Social Spaces in the Edge

Most current online social networks (OSNs) such as Facebook and LinkedIn impose a centralized model with a datas-

tore owned by the company that maintains all their data and that is accessed by the users. Of course, users are aware of the business model behind such OSNs, based on advertising or paid premium services.

Like in many previous works we argue that this centralized model is a serious danger to the privacy of users. But decentralized OSNs such as Diaspora do not have enough traction since they imply costly installations for the users. Semi-decentralized or federated alternatives such as Quitter also imply trust in the federated server which in fact follows a centralized data store model for their own users.

We argue that Edge-centric Computing hybrid architectures may be an adequate solution for OSNs for the following reasons:

Trust and Control: Privacy in these novel architectures may be achieved combining end-to-end security with semi-trusted data center support for Edge-centric Computing. Secure and sensitive information such as friend lists, online social profiles, log of computer-mediated social interactions should be carefully protected and controlled by the users.

On the one hand, users will not be forced to install and administer complex server software, which will reduce the barrier for entering the network. Data center support for Edge-centric Computing will offer the necessary secure infrastructure for social interactions. Such technologies should be open and standardized such as open Internet protocols in order to reach traction as OSN communication means between users.

On the other hand, the combination of cloud security and P2P technologies may create novel systems where even compromised servers may not imply a leakage of users sensitive data. A lot of research challenges emerge here to make feasible this kind of networks. Since any mobile device or even server may be compromised by attacks, such system will in the end have to reach a trade-off between affordable security and users interaction. What privacy guarantees can be given assuming some inevitable leakages to edge nodes?

Humans: Social networks are at the heart of human-driven distributed systems where connections are established between human (and their associated devices). When the underlying connection architecture reflects those human connections, many research challenges may arise in distributed systems. For example, previous P2P research on secure routing, reputation and trust may be applied to this new setting where the edge topology is driven by human interactions.

Another critical issue is social networks as valuable sensors of human activity. When information is not centralized, the access and aggregation of social information may be extremely useful for a number of applications.

There are important research challenges to create open platforms that permit third-party applications access to selected information in their social networks in a privacy-sensitive way. What protocols need to be in place for social apps to work? How to protect data from such applications reporting it to a third party? What new social activity would be enabled by edge-based OSNs?

Another important challenge is human collaboration (Computer Supported Cooperative Work) thanks to Edge-centric Computing platforms. Social Networks may evolve to provide human participation in heterogeneous groups. For example, new edge platforms may facilitate citizen participation and the reinforcement of social links in local communities. Novel distributed services may be designed for this kind of services

addressing the participation of mobile devices and server resources from adhoc or permanent collaborative groups.

Proximity and Intelligence: Location or physical proximity may be also relevant for the interactions in close-by social spaces such as companies, universities, neighborhoods or even bars and pubs among others. In this case, direct connections using Bluetooth, Wifi Direct, and short range technologies may be key to establish close-by communications between social spaces. This involves a combination of direct connections between mobile devices and connections between server edge resources. Proximity in these cases should be key to provide the correct tradeoffs that minimize the computational and battery costs of mobile devices involved in these communications.

When the information of these massive social networks is not controlled by a single centralized entity a wealth of information is then accessible to authorized third parties. Mining the deep social web creates interesting opportunities to intelligent agents, crawlers, and authorized applications. Matching and searching applications such as Dating, Work, Reputation, or even sales can then be controlled by the own users.

Previous work on recommender systems could now be applied to this massive edge social networks. But novel intelligent agents and assistants may benefit from this source of knowledge to extract useful information for groups or users. In this field we can also consider data sensitivity agents that may help users to simplify the protection and exposure of their own information in these networks.

4.3 Public Spaces in the Edge

Public Spaces are the more challenging and complicated scenarios for the next generation of distributed systems. They can include Smart Cities, Smart Grids, Smart Transportation Systems, IoT (Internet of Things), or IoE (Internet of Everything).

The public space is also the confluence of a myriad of Personal and Social Spaces that interact in public locations such as streets, roads, buildings or stadiums. For this reason, many of the aforementioned challenges in the two previous sections may partially overlap with the challenges of the public space.

Another important reason is efficiency and real-time interactions. Support for mobility and proximity implies that fast responses to users or devices (cars, M2M) are much more efficient if they do not require the intervention of a central party. The IoE implies a variety of heterogeneous mobile and fixed computing devices interacting with each other in different ways. This clearly precludes centralized designs and favors Edge-centric Computing hybrid architectures.

Trust and Control: Every end-user participates in the public space through its own mobile devices and sensors. A mobile user in the public space may switch between different service providers and contexts that may compromise its security and privacy.

When users (and their devices) are exposed to a huge variety of different interactions with service providers and sensors, novel technologies are required to preserve their security and privacy across domains. In particular, novel edge distributed technologies should help end-users to perform threat analysis and to protect (or be aware) accordingly to close-by risky interactions.

Again, edge technologies should seamlessly integrate with

their own cloud-based security/privacy schemes. As the user moves in the public space, she may generate flows of information that may compromise her own privacy. The interactions of the user that requires access to Cloud technologies should guarantee the confidentiality and security of users content and sensitive information.

Finally, Edge-centric Computing trust mechanisms should make end-users active participants of the public space. Instead of the passive citizen as a sensor of the centralized smart city, Edge-centric Computing can promote the active participation of users in their local communities.

Open research issues include reputation systems in the public space, trusted interaction between users and sensors, or anonymous participation mechanisms, among others.

Humans: Humans are the most important factor to take into account in these novel distributed systems. Human behavior is of paramount importance as a valuable information for adaptive distributed systems.

For example, one goal of the Smart City may be to optimize energy usage, but another key goal is to improve the quality of life of its citizens. Edge distributed systems may even use personal information to provide personalized advice to some citizens. For example, if one is allergic to some specific plants, she could receive different path recommendations that avoid risky zones.

Here the information flows are bidirectional between humans and platforms. On the one hand, users generate their own flows of information that may share with the IoE environment. They can make public some personal information, or they can even capture and contribute information with their own devices (user's sensors).

On the other hand, the public space and their different service providers (advertising, entertainment, social, public institutions, sensors) may also generate information flows that may be of interest for humans and their devices. In this case, the user's intelligent agent may receive these information flows and react to them according to their user's interests.

Proximity and Intelligence: In the public space, proximity is very relevant both for analyzing close-by information and for storing local information. Like in Fog Computing [1], one of the key characteristics of Edge-centric Computing is its proximity to end-users and its support for mobility.

With the progress of M2M and IoT, the amount of data generated from Giga-ordered sensors in urban areas might become Exabyte order. In such huge data, it is difficult to store all of the data in remote cloud servers with reasonable costs. In general, neighboring geospatial data might have strong correlation not present in distant geospatial data.

Thus, it might be suitable for storing neighboring geospatial data in local edge servers (controlled by community networks, users or institutions) and providing local dependent services using those data. Since such geospatial data are welled out continuously everywhere, all of such big data cannot be stored in the cloud. Thus, we need to study about (i) what information processing are needed for treating such huge data, (ii) what analytic mechanisms are useful for those geospatial data, (iii) how and when we can discard sensing data welled out continuously and (iv) how to protect user's data obtained from user edge sensors. The solutions for such questions can really provide future safe and smart urban life to people.

5. CONCLUSIONS

Edge-centric Computing is a novel paradigm that moves the locus of control of Cloud Computing applications and services to the edges of the network. An edge may be a mobile device, a wearable device but also a nano-data center or a user-controlled device. While the fundamental reason is privacy, since Edge-centric Computing allows users to retake control of their information, leveraging user's resources and even reducing response times make edge-centric computing appealing to novel personal and social online services.

The distinguishing characteristics that we find in the edges of the network are: (i) Humans: indistinguishable from their devices in many cases, (ii) Trust: based on edge encryption under user's control, (iii) Control: coordination from trusted edges (iv) Intelligence: leveraging the resources of edge devices and (v) Proximity: edge location and support for mobility.

Edge-centric Computing is the natural confluence of peer-to-peer and cloud computing to create hybrid architectures that combine stable resources with mobile terminals. It overcomes the limitations of P2P models (churn, availability) while providing security and privacy to hybrid Edge services.

6. REFERENCES

- [1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *Proc. of the 2nd Workshop on Mobile Cloud Computing (MCC)*, pages 13–16. ACM, 2012.
- [2] Cisco Research Center Requests for Proposals (RFPs). Fog computing, ecosystem, architecture and applications. http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp13078.html.
- [3] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.
- [4] T. Higuchi, H. Yamaguchi, and T. Higashino. Mobile devices as an infrastructure: A survey of opportunistic sensing technology. *Journal of Information Processing*, 23(2):94–104, 2015.
- [5] I. P. Kurniawan, H. Febiansyah, and J. B. Kwon. Cost-effective content delivery networks using clouds and nano data centers. In *Ubiquitous Information Technologies and Applications*, pages 417–424. Springer, 2014.
- [6] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *Communications Magazine*, 48(9):140–150, 2010.
- [7] E. Lee et al. The swarm at the edge of the cloud. *Journal of Design & Test*, 31:8–20, 2014.
- [8] M. Rabinovich, Z. Xiao, and A. Aggarwal. Computing on the edge: A platform for replicating internet applications. In *Web content caching and distribution*, pages 57–77. Springer, 2004.
- [9] M. Ryden, K. Oh, A. Chandra, and J. Weissman. Nebula: Distributed edge cloud for data intensive computing. In *Proc. of the 2nd Int. Conf. on Cloud Engineering (IC2E)*, pages 57–66. IEEE, 2014.
- [10] L. Toka, M. Dell'Amico, and P. Michiardi. Online data backup: A peer-assisted approach. In *Proc. of the 10th Int. Conf. on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE, 2010.

外文译文

摘要

在人类生活的诸多方面，长期存在着关于集权与分权的争议和讨论。计算科学领域表现出同样的现象：从初始大型机到 PC 机和本地局域网络，并且过去的几十年里，我们也经历了数据、服务、应用在数据中心和云中合并集中管理的时代。但是，我们主张一些新的转变是必要的。伴随着用户对可信赖性、隐私、自动化的需求日益增加，诸如功能丰富的专属智能家居盒子、具备优秀性能的移动终端用户设备、强大的无线网络等技术进步，都需要将对计算应用程序、数据、服务的控制权，从中心节点（核心）迁移到互联网的逻辑边缘。我们同时认为，这种发展趋势有助于人与机器的界限融合，促进了人类参与计算决策的社交计算、人本系统的设计。我们将以人为中心、基于边缘设备的计算模式称为“以边缘为中心的计算”。在本文中，我们详尽地阐释了它的研究挑战和实现。

1. 引言

在人类社会的许多领域，关于集权和分权之间经历了反复的讨论。在联邦制国家，权力在联邦政府和所属州省之间争夺、此消彼长。原本由大型发电厂负责的发电工作正在转移到分散的电网。

在计算机领域，我们在集中式控制和分布式控制之间目睹了类似的变化。20 世纪 80 年代，分布式控制的浪潮触发了由当时占主体位置的中央控制框架到个人电脑和本地网络的转变，当 P2P 技术和自动计算方法的应用，分布式系统达到了发展高峰。

近年来，在面向用户的互联网端，越来越多的功能强大的计算设备出现。高容量的移动设备，实时在线和专用的互联网连接盒和家庭路由器，或者无处不在的高带宽无线网络就是突出的例子。我们同时也面临着一个重要的集权浪潮。计算系统的控制，数据和智能重新回到了云端，集中但对用户透明的计算系统。

显然，云计算凭借其专用数据中心的巨大能力以及使用简单的集中式架构创造了有效的经济规模。但是，我们认为，当被集权达到一个极端时，完全集权在几个方面带来的弊大于利。第一个基本问题是通过将个人和社交数据发布到诸如电子商务网站，评级服务，搜索引擎，社交网络和位置服务等集中式服务会引发隐私问题。第二个根本问题是用户的应用程序完全委托给云端系统控制，这需要客户端对云端的单方面信任。第三，错失了利用现代个人设备进行信息处理、通信和数据存储任务的机会。最后，集中化阻碍了新兴的人本计算的发展，它融合了人与机器之间的界限并能产生新的应用类型。

我们在本文中的立场是，云的出现不应该是最终的计算范式，而且新的分权浪潮是必要的。我们主张以边缘为中心的计算作为一种新的计算范例，将计算应用程序，数据和服务的前沿从集中式节点推向网络外围。我们认为这种模式将保留使用云作为支持基础架构的核心优势，但会将控制和信任决策放回边缘，并允许创新的、以人为中心的计算应用程序。

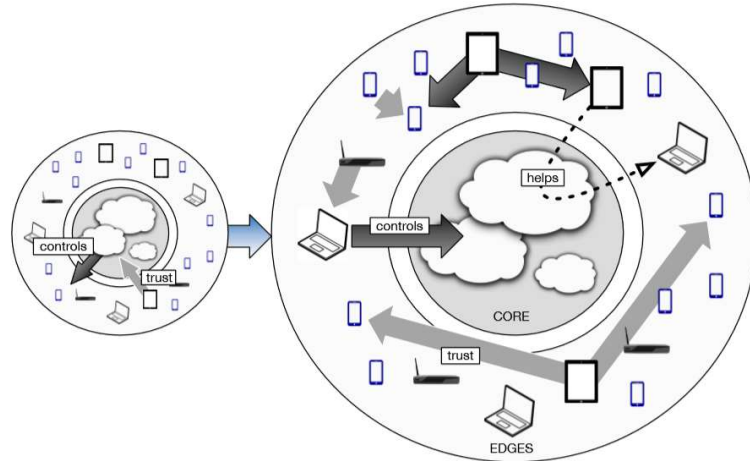


图-1. 集中式云模型（左）与边缘中心计算（右）

我们考虑一个面向节点的互联网视图，其中包括数据中心和云核心，如图-1 所示。围绕这个核心的是较小的网络服务器和内容分发网络作为下一层；接下来是“边缘”，由台式 PC、平板电脑、智能手机和纳米数据中心等个人人控设备组成（性能稳定的计算设备，如路由器或媒体中心）。在本文中，下一层基于 IP 地址标记的传感器和嵌入式处理器，我们忽略这些设备，因为我们专注于人工操作的设备。请注意，面向互联网的观点与面向网络的观点形成对比，在网络观点中网络本身被视为核心，所有计算设备和系统大小都被视为边缘设备。

以边缘为中心的计算包含以下要素：

- **临近即边缘：**这是关于点对点（P2P）系统和内容分发网络（CDN）的观点，虽然陈旧但仍然有效。在靠近节点之间传递和分配信息比使用远程集中式节点更有效。在这里，“临近”可以从物理和逻辑意义上理解。
- **智能即边缘：**由于小型化仍在继续，计算能力仍在增加，边缘传感器和设备变得更加强大。这为边缘自主决策开辟了道路，例如新型分布式众智程序，还有人为控制的执行器或接受信息流的 Agent 代理。
- **信任即边缘：**个人和社交敏感数据显然位于边缘。信任关系的控制和敏感信息的管理以安全和私密的方式流动，因此也必须属于边缘。
- **控制即边缘：**应用程序的管理和协调也来自边缘机器，可以将计算，同步或存储分配或委托给其他节点或选择性地分配给核心。
- **人类即边缘：**以人为本的设计应该将人类置于控制回路中，以使用户可以重新控制他们的信息。这应该引发用户控制其网络链接的新型众包和社交消息结构的设计浪潮。最后，这也为以人为本的应用创新提供了机会。

我们没有看到以边缘为中心的计算仅仅意味着纯粹的分布式或点对点系统。以边缘为中心的计算体系结构可能由跨数据中心和纳米数据中心部署的边缘中心分布式服务组成，可从边缘设备访问。此外，随着互联网服务（如电子邮件）的分布式，混合边缘服务可能会由不同的供应商部署，并且能够相互交流。我们预见到有趣的场景，其中以边缘为中心的计算服务可能是各种个人和社交通信和存储服务的自然分散演变。

2. 相关领域

内容交付网络:“边缘计算”这个术语是在 2002 年前后创建的,当时一些大公司宣布通过 CDN 边缘服务器分发交易软件,这主要与 CDN 上的应用部署有关。这种方法的主要目标是从 CDN 边缘服务器的接近度和资源中获益,以实现巨大的可扩展性。在边缘计算的这种早期风格中,“边缘”仅限于遍布全球的 CDN 服务器。这个架构模型由几位研究人员研究和扩展,特别是在 CDN 中部署和拷贝应用^[8]。我们的以边缘为中心的计算远远超出了与 CDN 相关的初始方法。我们认为,边缘并不局限于 CDN 节点,它还可以包括位于网络外围的无数用户设备和传感器。此外,考虑到可信度、情报和人类,我们考虑除了边缘之外的其他方面。

P2P: P2P 计算不仅是一个与边缘计算密切相关的领域,它也是边缘计算的主要前身。术语 P2P 是在 2000 年左右首次引入的,其中流行的 P2P 文件共享系统如 Napster 和 Kazaa。从那时起,它已经发展成为分布式系统的重要子领域,在这些 P2P 系统中,分布式、极高的可扩展性,容忍高水平的客户流失以及防止恶意行为成为研究的主要议题。在该领域的主要成就之一中,需要提到分布式哈希表,它是在云计算中分布式键值存储的更一般范例的演变;一般化的 Gossip 数据传输协议,已经成功地用于复杂任务,而不仅仅是简单的信息处理,例如数据聚合和拓扑管理,视频点播、直播电视、人对人通信等形式多媒体流传输。

不幸的是, P2P 术语一直因其用于非法文件共享而受到污染,并且涉及相关起诉的媒体广泛报道和诉讼。因此,许多实际上基于 P2P 范例的商业技术不被承认(例如, Akamai 的 NetSession 接口)。

以边缘为中心的计算模式源于 P2P,但扩展到新的途径。它避免了对“分权化神话”的天真追求,认为分权化是一种万能药。相反,它将对等概念扩展到互联网边缘的所有设备,并将 P2P 计算与云混合在一起。

去中心化的云架构:云计算是一种中心计算的范例,存储和处理资源托管在大型数据中心内。尽管如此,近年来, P2P 和云计算架构相结合已经有很多工作要做。一方面,云服务可以通过在必要时向他们提供稳定的资源来加强 P2P 系统,例如,当面临高流失或亚临界对等人群时。另一方面, P2P 可以通过提供额外资源来降低云服务的运营成本,并且可以通过提供地理多样性和客户接近程度来加强它们。

根据这些方针,最近出现了各种同行协助的服务,将混合体系结构中的对等和云资源结合起来。例如,研究人员已经表明,混合体系结构中,对等体(带宽,存储)的资源与临时使用的云存储服务相辅相成,可以与传统的客户机-服务器体系结构相媲美,但成本只有其中的一小部分^[10]。

另一个有趣的研究是使用相对稳定的对等资源来构建纳米数据中心^[5],微云、社区云或边缘云^[9]。例如,在文献^[5]中,所有家用电器都由电信提供商集中控制和管理。相比之下,我们的以边缘为中心的计算系统的愿景是以用户为中心的,控制来自边缘而不是其他方式。

雾计算:雾计算是一个最近的研究领域,与边缘中心计算有重大的重叠。正如 CISCO

定义的那样，“雾计算是一种将云计算和服务扩展到网络边缘的范例。”接近终端用户、密集的地理分布和对移动性的支持是雾计算的主要特征。

雾服务^[1]可以由网络托管，甚至可以在诸如机顶盒或接入点的终端设备中托管。主要好处是边缘智能与接近度相结合，以获得大量应用的实时或可预测延迟。雾计算因此适用于实时数据处理和分析。

最后，雾计算^[7]提出了一个基于 swarmlets 的开放式应用模型，以弥合网络物理系统（传感器，执行器）与 Cloud 之间的差距，从边缘的接近度和智能度。再一次，我们的以边缘为中心的计算更侧重于从网络边缘控制的人为驱动应用。

3. 研究挑战

3.1 人类驱动的分布式系统

在以边缘为中心的计算中，人类具有关键角色。以人为本的设计应该将用户置于控制环路中，以便他们可以控制他们的信息。个人计算设备的大规模扩散正在打开新的以人为本的设计，模糊了人与机器之间的界限。

利用搭载高性能计算核心的移动设备已成为大规模环境和人类行为感知的有效方法。已经提出了几种手机感应技术^[6]和机会主义的感知技术^[3,4]。

这应该导致产生新的社交消息体系结构，用户可以通过安全的方式控制、提供或汇总信息。在设计新的安全方法方面存在重要的研究挑战，这些方法通过群体检测等手段将人包括在数据分析循环中。

作为传感器的用户可能在物联网的背景下创造大量有用的信息。然后，人类将成为学习算法，数据分析和可视化工具的重要训练数据源。

经典的集中式体系结构对这种人群感知和众包信息可能带来很大的隐私风险。因此，一个重要的挑战是设计尊重用户隐私的边缘大数据分析系统。与集中监控边缘用户相关的道德问题很严重。以边缘为中心的计算可以提供无需为汇总的个人信息成本支付，即可获得平台的服务。

最后，人类活动的分析与交互对于关闭自适应分布式系统的控制环路也是非常有用的。这可能会为分布式系统开辟一个新的研究场所，以适应不同情况下的用户行为。

3.2 边缘架构和中间件

P2P 方法的一个重要差异是这些新架构可能依赖于在数据中心部署新颖的以边缘为中心的分布式服务。新的标准分布式服务必须在边缘节点部署，执行通信、计算、内容分发和存储创建等任务。这些服务应能够跨数据中心和分布式边缘设备应用程序，同时确保最终用户的隐私控制。开发面向边缘中心的计算应用程序和新型编程服务模型抽象和中间件也将是必需的。

以边缘为中心的计算超越了混合云模型，其中一部分是可信的，而另一部分则不是。以边缘为中心的计算基于分布式模型，该模型将由各种实体控制的异构云资源相互连接起来。覆盖技术与云资源的新型组合可能会开启新的研究可能性。

另一个重要的区别是 P2P 的流失性和瞬态可用性的固有特性可以通过依赖稳定的边

缘应用资源来克服。在 P2P 社区中，以前没有被提到该新设计方案。

最后，边缘架构面临的一个重要挑战将是发现移动终端与云服务器之间的正常交易行为。尽量减少移动终端的计算和电池耗尽，同时确保隐私和安全将代表新颖和有趣的研究挑战。

3.3 安全和隐私

以边缘为中心的计算超越了以前尝试使用 E2E（端到端）加密和以用户为中心的保护系统的尝试来保护用户云中的信息。以边缘为中心的架构将以新的方式挑战研究人员。除了用于保护私人信息的加密之外，使用不同的技术（如重新加密或基于属性的加密等）还可以为会合，通信和访问控制提供更安全的代理。此外，必须创建用于隐私感知信息共享的新型安全中间件，以促进以边缘为中心的系统。

许多云安全方面的工作，例如加密数据存储，对加密数据的查询，同态系统可能有助于创建新颖的以边缘为中心的服务。传统云安全研究的一个重要差异是，以边缘为中心的计算可能会假设存在可信的或部分可信的稳定资源，这些资源对边缘部署和控制的应用程序执行一些通信，持久性，查询甚至计算。以边为中心的计算也可以考虑在分布式的基于边的覆盖中可信节点与恶意节点共存。这将再次需要安全路由，冗余路由，信任拓扑和先前的 P2P 研究应用于这种新颖的设置。

最后，另一个关键的区别是，与集中式计算相比，以边为中心的计算可以防止信息集中。之前关于信息碎片化与加密技术的云安全研究可能与分散式覆盖技术相结合，以确保为敏感数据提供适当的数据保护。此外，针对覆盖网络中的碎片数据和索引的安全云查询和计算可以创建关于敏感信息的隐私的全新模型。

3.4 扩展性

可扩展性是 P2P 和云计算环境中反复出现的研究挑战。扩展到数百万用户的体系结构设计必须考虑到容错、流失、弹性和其他问题。在 P2P 中，动态流失使这些体系结构及其整体服务可用性的可行性变得复杂。在云计算中，扩展和弹性是经常性的话题，甚至主要的云提供商也可能被大规模的拒绝服务攻击造成崩溃。

但是，以边缘为中心的计算完全改变了之前呈现的可扩展性挑战。由于使用稳定的云资源，客户流失不再是一个限制因素。主要挑战是边缘设备、可信服务器和不可信服务之间的计算和通信责任之间的正确交易。

鉴于控制处于边缘，扩展性问题仍然非常重要。构建将移动设备与有限电池和稳定云资源结合的大规模覆盖需要特别关注节点之间的通信协议。此外，云/边缘服务也必须是高效的，并考虑到它们中必要的异构服务节点。

另一个研究挑战是大规模覆盖的可扩展性与安全性的结合。安全性的边缘体系结构由于加密而产生不可忽略的开销，必须对其进行处理以提供可扩展性。

4. 案例

4.1 边缘位置的私有空间

我们的数字生活现在分布在众多云设备和应用程序中。我们有 Dropbox 的文件，Gmail 中的电子邮件，Instagram 中的精选照片，LinkedIn 中的工作联系人以及我们的 Facebook 社交网络。我们的其他信息，如工作数据和个人数据（照片，视频，财务数据和健康数据）分布在硬盘和各种用户设备上。

未来几年，个人信息空间将出现，以统一我们整个数字生活的多个流程。我们所有的个人信息都将存储在云端，我们将有机制让第三方应用程序访问我们数据存储库的一部分。在这种情况下，以边缘为中心的计算可以提供：

信任和控制：未来个人信息空间的强大挑战是隐私和用户控制他们自己的信息。在未来的几年中，以边缘为中心的计算将启用新一代以用户为中心的个人空间，用户将能够决定他们的信息孤岛中哪些部分可以被第三方应用程序访问，但也可以由第三方用户访问。这需要设计新颖的架构，提供受控的隐私感知数据共享和先进的访问控制机制。

用户信息可能存储在云提供商中，但是提供加密和隐私保证，确保云提供商在未经许可的情况下无法访问用户数据。此外，安全查询应使用户能够在其个人信息空间中查找数据，而无需云提供商能够推断有关它们的信息（盲服务器）。

另一个关键方面是对可能建立不同类型社交链接的其他用户或实体的信任关系。这对于与他人分享信息以及参与者之间的协作互动至关重要。这些连接可以是永久或临时的。永久连接的例子是分享照片，视频，歌曲，书籍，应用和其他购买的数字内容的家庭成员。临时协作的一个例子是例如两个个人空间的暂时重叠以在某个时间分享一些信息。

人类：以人类个体为本的计算会得到发展，因为我们被连接的设备所包围。这将与普适计算，无处不在的系统和增强现实界面中的现有研究成果保持一致。

以边缘为中心的计算体系结构将会是分布式系统，根据其位置或上下文来适应用户行为。它还将通过可用的连接设备处理与其他人的交互。每个人都会携带多个移动设备（手机和手表）和传感器（例如乐队和植入设备）。这些设备可以从其所有者（健康设备，传感器），其他近距离设备，以及通过互联网的远程链接获取信息。

此外，用户可以参与安全的分布式众包平台，在这些平台上将部分选定的个人数据提供给外部分析系统。想象一下，一位用户让第三家公司在家中访问他们的能源使用情况，以优化她的账单。这种基础设施的设计将对分布式系统和安全研究人员构成严重挑战。

接近度和智能：我们的个人空间必须适应我们当前的位置：在家中，在工作中，在汽车里，在街上行走，在商场里，在机场等等。这些边缘分布式架构设计的一个重要方面是它们将分散，而且不同的信息流将属于不同的服务和实体。相互影响的互动，同步和内容分发将在此类系统的设计中发挥重要作用。

以边缘为中心的计算也可能成为在各种场景中部署个人代理和代理系统的关键促进因素。代理商可以从外部实体接收流动信息，甚至可以对这些流动做出反应。以边缘为中心的计算平台可以为代理部署提供所需的通信，发现和信任平台。

在向以边缘为中心的自治代理模式转变中涉及许多研究挑战。鉴于目前的集中模式

限制了代理商的可能性，将信任放在边缘可能会促进代理商之间必要的同行交流。

4.2 边缘位置的社交空间

大多数当前的在线社交网络（OSN），例如 Facebook 和 LinkedIn，都强加了一个集中式模型，该模型包含公司拥有的数据存储，该数据存储维护其所有数据并由用户访问。当然，用户基于广告或付费高级服务意识到这些 OSN 背后的商业模式。

和以前的许多作品一样，我们认为这种集中式模式对用户的隐私是一种严重的危险。但是分布式的 OSN 没有足够的吸引力，因为它们意味着昂贵的用户安装。像 Quitter 这样的半分布式或联合式的替代方案也意味着对联合服务器的信任，而联合服务器实际上遵循针对其用户的集中式数据存储模型。

我们认为，以边缘为中心的计算混合体系结构可能是 OSN 的一个适当的解决方案，原因如下：

信任和控制：这些新体系结构中的隐私可以通过端到端安全与对边缘中心计算的数据中心支持。安全和敏感的信息，如朋友列表，在线社交资料，计算机调解的社交互动记录应受到用户的仔细保护和控制。

一方面，用户不会被迫安装和管理复杂的服务器软件，这将减少进入网络的障碍。数据中心对以边缘为中心的计算的支持将为社交互动提供必要的安全基础设施。这些技术应该是开放和标准化的，例如开放的互联网协议，以便如同用户之间的 OSN 通信手段一样获得牵引力。

另一方面，云安全和 P2P 技术的结合可能会创建新的系统，即使是受损的服务器也不一定意味着用户敏感数据的泄露。这里出现了很多研究挑战来使这种网络变得可行。由于任何移动设备甚至服务器都可能受到攻击的危害，因此这种系统最终必须在易于安全和用户交互之间进行交易。假设一些不可避免的边缘节点泄漏，可以给出什么隐私保证？

人类：社交网络是人类（及其相关设备）之间建立连接的以人类为驱动的分布式系统的核心。当底层连接体系结构反映这些人际连接时，分布式系统中可能会出现许多研究难题。例如，先前关于安全路由，信誉和信任的 P2P 研究可能会应用于边界拓扑由人为交互驱动的新设置。

另一个关键问题是社交网络作为人类活动的宝贵传感器。当信息不集中时，社交信息的访问和聚合对于许多应用程序可能非常有用。在创建开放平台方面存在重要的研究挑战，这些平台允许第三方应用程序以隐私敏感的方式访问社交网络中的选定信息。社交应用需要使用哪些协议才能工作？如何保护这些应用程序的数据向第三方报告？基于边缘的 OSN 将启用哪些新的社交活动？

由于以边缘为中心的计算平台，另一个重要挑战是人工协作（计算机支持的协作工作）。社交网络可能演变为提供人类参与异构群体。例如，新的边缘平台可能会促进公民参与和加强当地社区的社交联系。新型分布式服务可以设计用于解决来自特殊或永久性协作组的移动设备和服务器资源的参与问题。

接近度和智能：位置或物理接近度也可能与附近社交空间（如公司，大学，社区甚

至酒吧和酒吧等)之间的互动有关。在这种情况下,使用蓝牙, Wi-Fi Direct 和短距离技术的直接连接可能是建立社交空间之间的近距离通信的关键。这涉及移动设备之间的直接连接和服务器边缘资源之间的连接的组合。在这些情况下,接近度应该是提供正确的贸易关键的关键,以最大限度地减少这些通信中涉及的移动设备的计算和电池成本。

当这些大规模社交网络的信息不受单一中央实体控制时,授权第三方便可获取大量信息。挖掘深层社交网络为智能代理,爬虫和授权应用程序创造了有趣的机会。匹配和搜索应用程序,如约会,工作,声誉,甚至销售,然后可以由自己的用户来控制。

以前在推荐系统上的工作现在可以应用到这个庞大的边缘社交网络。但是新颖的智能代理和助手可以从这种知识来源中获益,从而为团体或用户提供有用的信息。在这个领域,我们也可以考虑数据敏感代理,它可以帮助用户简化他们自己在这些网络中的信息的保护和暴露。

4.3 边缘位置的公有空间

公共空间是下一代分布式系统更具挑战性和复杂性的场景。它们可以包括智能城市,智能电网,智能交通系统,物联网(IoT)或IoE(万物互联)。

公共空间也受个人和街道,道路,建筑物或体育场等公共场所相互作用的影响。出于这个原因,前面两节中提到的许多挑战可能与公共空间的挑战部分重叠。

另一个重要原因是效率和实时交互。对移动性和接近度的支持意味着如果用户或设备(汽车, M2M)不需要中央参与者的干预,则快速响应就更有效率。IoE意味着各种不同的移动和固定计算设备以不同的方式彼此交互。这清楚地排除了集中式设计,并有利于以边缘为中心的混合架构。

信任和控制:每个终端用户都通过自己的移动设备和传感器参与公共空间。公共空间中的移动用户可能会在不同的服务提供商和可能危及其安全性和隐私的环境之间切换。

当用户(及其设备)面临与服务提供商和传感器的各种不同交互时,需要采用新技术来保持跨域的安全性和隐私性。特别是,新颖的边缘分布式技术应该能够帮助最终用户执行威胁分析,并相应地保护(或意识到)对近距离风险交互的影响。

同样,边缘技术应该与他们自己的基于云的安全/隐私方案无缝集成。当用户在公共场所移动时,她可能会产生流动的信息,这可能会危及自己的隐私。需要访问云技术的用户的交互应保证用户内容和敏感信息的保密性和安全性。

最后,以边为中心的混合信任机制应使最终用户成为公共空间的积极参与者。以边缘为中心的混合计算可以促进用户积极参与当地社区,而不是被动公民作为集中式智能城市的传感器。

开放研究问题包括公共空间中的声誉系统,用户与传感器之间的可信交互,或匿名参与机制等等。

人类:人类是这些新型分布式系统中最重要考虑因素。人类行为作为适应性分布式系统的宝贵信息极为重要。

例如,智能城市的一个目标可能是优化能源使用,但另一个关键目标是提高其公民

的生活质量。边缘分布式系统甚至可以使用个人信息为一些公民提供个性化建议。例如，如果某人对某些特定植物过敏，她可能会收到不同的路径建议，以避免危险区域。

这里的信息流在人与平台之间是双向的。一方面，用户可以生成自己的流信息，这些信息可能与 IoE 环境共享。他们可以公开一些个人信息，或者甚至可以使用自己的设备（用户传感器）捕获和提供信息。

另一方面，公共空间及其不同服务提供商（广告，娱乐，社交，公共机构，传感器）也可能产生对人类及其设备可能感兴趣的信息流。在这种情况下，用户的智能代理可能会收到这些信息流，并根据用户的兴趣对它们做出反应。

接近度和智能：在公共空间中，接近度与分析接近信息和存储本地信息非常相关。与雾计算^[1]一样，以边缘为中心计算的关键特征之一是它接近终端用户并支持移动性。

随着 M2M 和物联网的发展，城市地区千兆级传感器产生的数据量可能会达到 Exabyte 订单。在如此庞大的数据中，以合理的成本将所有数据存储远程云服务器中是非常困难的。一般来说，相邻的地理空间数据可能在遥远的地理空间数据中不存在很强的相关性。

因此，它可能适用于将邻近的地理空间数据存储在本地的边缘服务器（由社区网络，用户或机构控制）中并使用这些数据提供本地相关服务。由于这些地理空间数据在各地都不断涌现，所有这些数据都无法存储在云中。因此，我们需要研究（1）需要哪些信息处理来处理这些庞大的数据（2）哪些分析机制对这些地理空间数据有用（3）我们如何以及何时可以丢弃感知数据（4）如何保护从用户边缘传感器获得的用户数据。 这些问题的解决方案可以真正为人们提供未来安全和智能的城市生活。

5. 结论

以边缘为中心的计算是一种新颖的计算范例，将云计算应用程序和服务的控制过程转移到网络边缘。边缘可以是移动设备、可穿戴设备，也可以是纳米数据中心或用户控制设备。边缘计算解决了隐私问题，因为以边缘为中心的计算功能允许用户重新控制其信息，并且充分利用用户资源甚至能够缩短响应时间，因此，以边缘为中心的计算吸引了新颖的个人和社交在线服务。我们在网络边缘发现的显著特征是：（1）人类：在很多情况下与设备不可区分；（2）信任：基于用户控制下的边缘加密；（3）控制：来自可信边缘的协同合作；（4）智能：利用边缘设备的资源；（5）接近度：边缘位置对移动性的支持。以边缘为中心的计算是受 P2P 和云计算的启发，以创建混合架构，将稳定的资源与移动终端相结合。它克服了 P2P 模式（流失，可用性）的局限性，同时为混合边缘服务提供安全性和隐私性。

原文索引

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In Proc. of the 2nd Workshop on Mobile Cloud Computing (MCC), pages 13–16. ACM, 2012.

[2] Cisco Research Center Requests for Proposals (RFPs). Fog computing, ecosystem, architecture and applications. http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp13078.html.

[3] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. IEEE

Communications Magazine, 49(11):32—39, 2011.

[4] T. Higuchi, H. Yamaguchi, and T. Higashino. Mobile devices as an infrastructure: A survey of opportunistic sensing technology. *Journal of Information Processing*, 23(2):94—104, 2015.

[5] I. P. Kurniawan, H. Febiansyah, and J. B. Kwon. Cost-effective content delivery networks using clouds and nano data centers. In *Ubiquitous Information Technologies and Applications*, pages 417–424. Springer, 2014.

[6] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *Communications Magazine*, 48(9):140–150, 2010.

[7] E. Lee et al. The swarm at the edge of the cloud. *Journal of Design & Test*, 31:8–20, 2014.

[8] M. Rabinovich, Z. Xiao, and A. Aggarwal. Computing on the edge: A platform for replicating internet applications. In *Web content caching and distribution*, pages 57–77. Springer, 2004.

[9] M. Ryden, K. Oh, A. Chandra, and J. Weissman. Nebula: Distributed edge cloud for data intensive computing. In *Proc. of the 2nd Int. Conf. on Cloud Engineering (IC2E)*, pages 57–66. IEEE, 2014.

[10] L. Toka, M. Dell’Amico, and P. Michiardi. Online data backup: A peer-assisted approach. In *Proc. of the 10th Int. Conf. on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE, 2010.