

面向移动边缘计算基于强化学习的计算卸载算法*

杨戈^{1,2}, 张衡¹

(1. 北京师范大学珠海分校 智能多媒体技术重点实验室, 广东 珠海 519087;

2. 北京大学深圳研究生院 深圳物联网智能感知技术工程实验室, 广东 深圳 518055)

摘要: 针对移动边缘计算(Mobile Edge Computing, MEC)的计算卸载决策的问题, 基于强化学习方法提出了一个在多用户的 MEC 系统中的计算卸载决策算法(Offloading Decision algorithm based on Reinforcement Learning, ODRL)。ODRL 算法根据任务模型、计算模型以及信道状态对任务进行卸载决策, 采用强化学习方法求解最优计算卸载策略。仿真结果证明了所提出的 ODRL 算法与基线策略相比, 具有更低的系统总成本。

关键词: 移动边缘计算; 计算卸载; 强化学习; Q-learning

中图分类号: TN911.22

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.200920

中文引用格式: 杨戈, 张衡. 面向移动边缘计算基于强化学习的计算卸载算法[J]. 电子技术应用, 2021, 47(2): 36-40, 48.

英文引用格式: Yang Ge, Zhang Heng. Offloading decision algorithm based on reinforcement learning for mobile edge computing[J]. Application of Electronic Technique, 2021, 47(2): 36-40, 48.

Offloading decision algorithm based on reinforcement learning
for mobile edge computingYang Ge^{1,2}, Zhang Heng¹

(1. Key Laboratory of Intelligent Multimedia Technology, Beijing Normal University(Zhuhai Campus), Zhuhai 519087, China;

2. Engineering Lab on Intelligent Perception for Internet of Things(ELIP), Shenzhen Graduate School,

Peking University, Shenzhen 518055, China)

Abstract: For the problem of computing offloading decision in mobile edge computing, this paper proposes an offloading decision algorithm based on enhanced learning in multiuser MEC system. According to the task model, calculation model and channel state, ODRL algorithm makes the task unloading decision and uses reinforcement learning method to solve the optimal computing unloading strategy. Simulation results show that the ODRL algorithm proposed in this paper has lower total system cost compared with the baseline strategy.

Key words: mobile edge computing; computing offloading; reinforcement learning; Q-learning

0 引言

随着万物互联趋势不断加深, 网络边缘的智能终端设备不断增加^[1-10]。移动边缘计算是为提升移动设备服务质量而提出的一种具有前景的新范式^[11-20]。

(1) 以降低时延为优化目标

文献[3]将卸载决策归结为两种时间尺度下的随机优化问题, 并采用马尔科夫决策过程来处理这个问题。通过分析每个任务的平均时延和设备的平均能耗, 提出了一个能量约束的时延最小化问题, 并设计了一个有效的一维搜索算法找到了最优的任务卸载策略。

文献[4]提出了一种低复杂度的渐近最优在线算法,

该算法在每个时隙中通过封闭形式或二等分搜索获得最优解, 共同决定卸载决策、移动执行的 CPU 周期频率和计算卸载的发射功率。

文献[5]提出了一种基于云的分层车辆边缘计算(VEC)卸载框架, 在该框架中引入了附近的备用计算服务器来弥补 MEC 服务器的不足计算资源。基于此框架, 文献采用斯塔克尔伯格博弈方法设计了一种最佳的多级卸载方案, 该方案可以最大程度地利用车辆和计算服务器的效用。

(2) 以降低能耗为优化目标

文献[6]将卸载决策公式化为凸优化问题, 用于在边

* 基金项目: 广东高校省级重大科研项目(2018KTSCX288, 2019KZDXM015, 2020ZDZX3058); 广东省学科建设专项(2013WYXM0122); 北京师范大学珠海分校智能多媒体技术重点实验室(201762005); 北京师范大学珠海分校 2019 年校级“质量工程”课程思政项目(201932)

缘云计算能力无限和有限的两种情况以及在计算等待时间的约束下最小化加权和移动能量消耗。文献[7]对该方案做出了改进,相比前者降低了90%的能耗。

(3)以权衡能耗和时延为目标

文献[8]分析了单用户 MEC 系统的能耗延迟权衡问题,提出了一种基于 Lyapunov 优化的云卸载计划方案以及云执行输出的下载调度方案。在文献[9]中扩展到了多用户系统。

文献[10]研究了在多信道无线干扰环境下移动边缘云计算的多用户计算卸载问题。采用博弈论方法以分布式方式实现有效的计算卸载。

1 MEC

本文讨论一个由一个访问接入点(Access Point, AP)和多个移动设备组成的 MEC(Mobile Edge Computing)系统。无线 AP 可以是小型蜂窝基站,也可以是 Wi-Fi AP。除了用作核心网络的常规 AP 之外,它还安装了其他计算服务器作为 MEC 服务器。由于移动设备可能没有足够的计算能力或电池容量来完成计算密集型或对延迟敏感的任务,将部分工作分担给 AP 可以有效提高服务质量,降低移动设备能耗。

MEC 系统包括 2 个模型:任务模型、计算模型。

1.1 任务模型

本文假设计算任务按照泊松过程到达用户设备(User Equipment, UE),其速率为 λ_a 。将时间划分为单位时隙 t_0 , 获得一个参数为 $p_a = \lambda_a t_0$ 的伯努利过程^[14]。由此, UE 的任务到达时间是独立的随机变量,服从带参数 p_a 的几何分布。

到达的任务用二元组 (L_a, μ_a) 描述,其中 L_a 表示每个任务卸载到 AP 所需上传数据(例如输入数据和程序代码)的平均大小, μ_a 表示完成一项工作所需的平均 CPU 周期。任务到达后 UE 根据本地状态、MEC 服务器状态和信道状态进行卸载决策(即选择以 x 的概率卸载其任务)。决策结果表示为 $A_k \in \{0, 1\}$, 其中 $A_k=0$ 表示第 k 个 UE 选择本地执行任务, $A_k=1$ 表示第 k 个 UE 选择卸载任务到 AP 执行。

本文主要关注的是多用户共享的 MEC 服务器的计算卸载决策问题,简化了无线通信部分,考虑了用户在正交信道上工作的情况^[15]。因此用户彼此之间不会受到多用户干扰,这是目前 LTE(Long Term Evolution)等通信系统中常见的情况。

由于传输任务数据的信道状态会随时间变化,当任务到达并且信道太差而无法进行数据传输时,UE 应选择本地计算,而不是等待有利的信道条件。

设 h_k 为第 k 个 UE 到 AP 的小尺度信道增益, $k=1, 2, \dots, N$, 式(1)通过香农定理(Shannon theorem)^[16]计算出信道的可靠传输速率 R_k :

$$R_k = \log(1 + d_k^{-\alpha} |h_k|^2 P_t / \sigma^2), \quad k=1, 2, \dots, N \quad (1)$$

其中, d_k 表示用户 k 到 AP 的距离, α 表示路径损耗指数, P_t 表示传输功率, σ^2 表示在 AP 处接收到的噪声功率。将可达到的数据速率 R_k 与预期传输速率 β_k 进行比较,根据香农定理,当 $R_k \geq \beta_k$ 时 UE 才能成功地将其任务卸载到 AP。将 $\rho_k = d_k^{-\alpha} P_t / \sigma^2$ 设为信噪比(Signal-Noise Ratio, SNR),通过式(2)得出成功卸载的条件:

$$|h_k|^2 > (e^{\beta_k} - 1) \rho_k^{-1}, \quad k=1, 2, \dots, N \quad (2)$$

1.2 计算模型

根据卸载策略,已卸载到 AP 的任务将被推送到 AP 缓冲区中等待计算。否则,任务将被推送到本地缓冲区中等待本地计算。接下来将讨论本地计算模型与卸载计算模型。

1.2.1 本地计算模型

任务在本地计算时,UE 使用自己的计算能力来执行作业。到达且为执行的任务将排列在本地计算缓冲区中。本地计算单元的工作状态通过本地计算缓冲区中待执行的任务还需要几个时隙才能完成来描述,表示为:

$$C_i[t] \in \{0, 1, \dots, N-1\}, \quad t=1, \dots, N \quad (3)$$

f_m (cycles/s) 表示 UE 的计算能力,UE 本地的计算服务速率为 $\mu_m = f_m / \mu_a$ (jobs/s)。由此,对于每个任务,系统所花费的时间(包括计算执行时间和在任务队列中的等待时间)可以由式(4)得出:

$$D_k^1 = \frac{1}{\mu_m t_0 - p_a} t_0, \quad k=1, 2, \dots, N \quad (4)$$

代入 $p_a = \lambda_a t_0$ 化简,得到式(5):

$$D_k^1 = \frac{1}{\mu_m - \lambda_a}, \quad k=1, 2, \dots, N \quad (5)$$

本地计算的功耗由式(6)得出:

$$E_k^1 = k_m f_m^2 \mu_a, \quad k=1, 2, \dots, N \quad (6)$$

其中, $k_m f_m^2$ 表示每个 CPU 周期的功耗, k_m 是取决于设备芯片架构的能耗系数^[16]。

结合式(5)与式(6),可以按式(7)计算总加权成本:

$$Z_k^1 = c_k^D D_k^1 + c_k^E E_k^1, \quad k=1, 2, \dots, N \quad (7)$$

其中, c_k^D (1/Second) 为本地计算所花费时间的权重,且 $0 < c_k^D < 1$; c_k^E (1/Joule) 为本地计算能耗的权重,且 $0 < c_k^E < 1$ 。

1.2.2 卸载计算模型

当 UE 选择将其任务卸载到 AP,则整个计算卸载过程可以分为 3 步。

(1)UE 需要将任务数据(例如输入数据和程序代码)上传到 AP,该步骤所消耗的时间即传输时延。计算如下:

$$D_{k,1}^0 = \frac{L_a}{\beta_k}, \quad k=1, 2, \dots, N \quad (8)$$

每个终端设备以功率 P_t 传输数据,则卸载功耗为:

$$E_k^0 = P_t \frac{L_a}{\beta_k}, \quad k=1, 2, \dots, N \quad (9)$$

(2)卸载到达 AP 的任务需要一些时间才能在执行后

离开 AP。这些任务将被推送到 AP 缓冲区中等待计算。AP 计算单元的工作状态通过待执行的任务完成所需时间来描述,表示为:

$$C_o[t] \in \{0, 1, \dots, N-1\}, t=1, \dots, N \quad (10)$$

由排队论的叠加性知,到达 AP 的任务是来自多个

UE 的多个泊松过程的叠加^[11],则总到达率为 $\sum_{k=1}^N \lambda_a$ 。令 f_b (cycles/s) 表示 AP 的计算能力,卸载计算的服务速率为 $\mu_b=f_b/\mu_a$ (jobs/s)。则每个任务在 AP 上花费的时间(包括计算执行的处理时延和在 AP 缓冲区中等待时延)可以表示为:

$$D_{k,2}^o = \frac{1}{\mu_b - \sum_{k=1}^N \lambda_a}, k=1, 2, \dots, N \quad (11)$$

与许多现有的研究(如文献[8]、[12]、[13])类似,本文忽略了边缘计算的能耗,因为 AP 通常不存在能量不足的问题。

(3)任务在 AP 完成计算,AP 将计算结果返回 UE。类似于文献[11]–[14]等许多其他相关工作,本文忽略了 MEC 服务器将计算结果发送回 UE 的时间开销,假设在下行链路方向上保证了计算结果的顺利传输。因为对于许多应用程序(例如人脸识别)而言,计算结果的大小一般远小于计算输入数据的大小。则组合式(8)、式(9)、式(11),可以通过式(12)来计算任务卸载的总加权成本:

$$Z_k^o = c_k^D (D_{k,1}^o + D_{k,2}^o) + c_k^E E_k^o, k=1, 2, \dots, N \quad (12)$$

2 ODRL 算法

本文通过计算分析 MEC 系统的延迟及功耗计算系统总成本,将 MEC 系统的卸载决策看作一个以降低 MEC 系统加权总成本为优化目标的随机优化问题,并采用强化学习方法求解最优卸载策略。

强化学习中有状态(state)、动作(action)、奖赏(reward)这 3 个要素。智能体(Agent,指 MEC 系统)会根据当前状态来采取动作,并记录被反馈的奖赏,以便下次再到相同状态时能采取更优的动作。接下来将具体到上文提出的系统模型中介绍。

状态:系统状态由一个二元组 $\tau[t]=(C_i[t], C_o[t])$ 描述。其中, $C_i[t]$ 为本地计算单元的工作状态, $C_o[t]$ 为 AP 计算单元的工作状态。

动作:根据上述决策方案,UE 将选择以 x 的概率卸载其任务,设动作集为 $A=\{x_i, i=1, 2, \dots, N\}$,分别为 UE 的 n 种不同的卸载决策。

奖赏:本文通过最小化系统总成本 $Z = \sum_{k=1}^N (1-A_k) Z_k^1 + A_k Z_k^o$ 达到优化目标,强化学习的训练目标为奖赏最大化,二者成反比,所以本文将即时奖赏定义为 $-Z$ 。

本文采用基于贪婪策略的 Q-learning 方法求解优化

问题。Q-learning 是一个基于值的强化学习算法,其核心为 Q 表(Q-table)。Q-table 的行和列分别为状态集 S 的各个状态和动作集 A 的各个动作。Q-table 的值 $Q(s, a)$ 记录每个状态所对应的各个动作的效用值,即在某一时刻的 $s(s \in S)$ 状态下,采取动作 $a(a \in A)$ 能够获得奖赏的期望。

训练过程中环境会根据智能体的动作反馈相应的奖赏,所以算法的主要思想就是将状态集与动作集构建一张 Q 表来存储 Q 值,然后根据 Q 值来选取能够获得最大的奖赏的动作。

强化学习的训练目标是最大化其(未来)总奖赏。在算法训练的过程中,使用式(13)去更新 Q-table:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a)) \quad (13)$$

其中, s_t 表示当前状态, a_t 表示当前动作, s_{t+1} 表示下一状态; r_t 表示即时奖赏; α 表示学习速率,且 $0 < \alpha < 1$,其取值大小影响价值函数的收敛速度; γ 表示折扣因子,且 $0 < \gamma < 1$ 。

为避免算法陷入局部最优,本文采用 ϵ -greedy 策略权衡强化学习的开发与探索。即智能体在已知的 (s, a) 二元组分布之外,以概率 ϵ 选择探索其他未知的动作。

3 仿真结果

3.1 实验环境

本文采用 Python 编程实现仿真实验,表 1 给出了程序实现的具体环境。

表 1 实验环境

名称	配置
处理器	Intel® Core™ i7-4720HQ CPU@2.60 GHz(8 CPUs), ~2.6 GHz
内存	8 192 MB RAM
操作系统	Windows10 家庭中文版 64-bit(10.0, Build 17134)
开发环境	Python 3.7.7

ODRL 中的训练参数包括外循环次数 e 、折扣因子 γ 、学习速率 α 、贪婪策略 ϵ 。

一次外循环(episode)指智能体在环境里面根据某个策略执行一系列动作到结束的这一过程,算法通过多次外循环不断更新 Q 表得到一个最优的计算卸载策略。外循环次数 e 一般要足够大,但是如果过大,则会导致邻域搜索时间过长,从而影响算法的性能。经过实验(如图 1 所示)发现,100 次外循环后 Q 函数收敛速度放慢,奖赏提升趋势不大,因此可以设置 $e \geq 100$ 。

由于学习速率 $\alpha(0 < \alpha < 1)$ 影响 Q 函数的收敛速度,本文使用指数减缓(exponential decay)方法调整学习率,即学习率按训练轮数增长指数差值递减,如式(14)所示:

$$\alpha' = 0.95^e \cdot \alpha \quad (14)$$

其中, α' 表示下一轮外循环的学习速率, α 表示当前外循环的学习速率, e 为当前外循环次数。

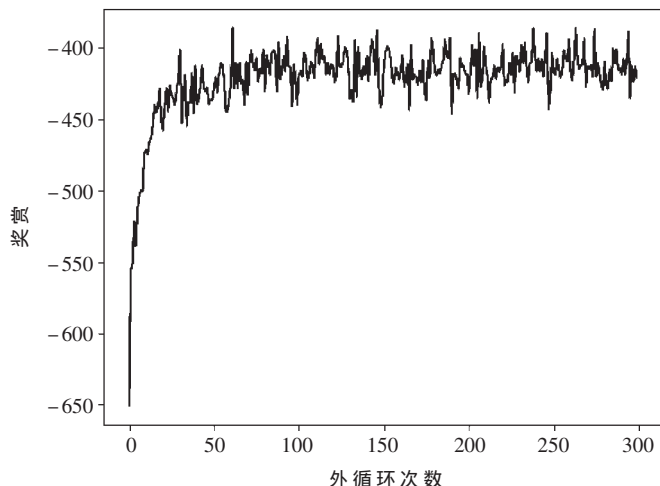


图1 奖励随外循环次数增加的变化

折扣因子 γ 的取值范围为 $0 \leq \gamma < 1$, $\gamma=0$ 表示重视即时奖励, γ 趋于 1 表示重视将来奖励。 γ 决定时间的远近对奖励的影响程度, 即牺牲当前收益来换取长远收益的程度。本文在实验中设置 $\gamma=0.5$ 。

贪婪策略 ε 表示每个状态探索新动作的概率, ε 较大时学习过程可以快速收敛, 但容易陷入局部最优, 通常取值为 0.1。

本文考虑由一个 AP 和多个移动设备组成的 MEC 系统。200 个 UE 均匀地随机分布在一个以 AP 为中心且半径为 $0 \leq d_k \leq 75$ (单位: m) 的环上。参考相关工作, 本文设置了各项环境参数, 如表 2 所示^[17-18]。其中, W 是取决于设备芯片架构的能耗系数, P_i 表示完成一项工作所需的平均 CPU 周期, σ^2 表示每个任务卸载到 AP 所需上传数据 (例如输入数据和程序代码) 的平均大小, α 表示任务按照泊松过程到达用户设备速率, f_B 表示本地计算所花费时间的权重, f_m 表示本地计算能耗的权重。

表2 环境参数

参数	值	参数	值
W/kHz	360	$k_m/(W \cdot s^3/\text{cycles}^3)$	10^{27}
P_i/mW	100	μ_a/cycles	2.5×10^{10}
σ^2/dBm	-40	L_a/KB	100
α	3.5	$\lambda_a/(\text{jobs/s})$	0.012
f_B/GHz	4	c_k^D	0.9
f_m/GHz	0.5	c_k^E	0.1

3.2 对比试验

为了验证本文所提出的 ODRL 在求解 MEC 计算卸载的卸载决策问题的有效性, 将所提出算法与两种基线策略进行了对比实验。

图 2 中给出了 100 次仿真实验中 3 种决策方法的系统总成本, Local execution 表示任务到达时所有 UE 都选择在本地执行任务, Edge execution 表示所有 UE 都选

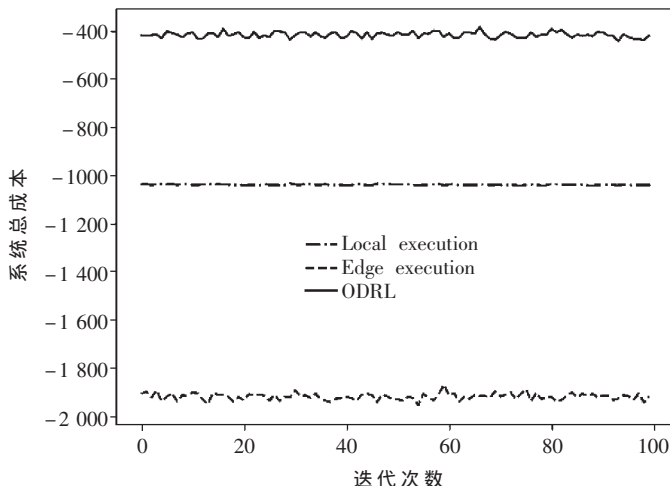


图2 3种决策方法在100次仿真实验中的系统总成本

择将到达的任务卸载到 MEC 节点执行。可以看出 ODRL 的平均表现优于两种对比策略, 使得系统总成本大幅度降低。

在图 3 中, 将 ODRL 与两种基线策略在 MEC 服务器性能不断提升的条件下各进行 100 次仿真实验, 得出不同 MEC 服务器性能下系统总成本的平均值。从实验结果中可看出, ODRL 的系统总能耗最低, 可以达到最好的效果。随着 MEC 服务器性能的提升, 被卸载任务的执行时间缩短, ODRL 与全卸载策略产生的平均系统总成本降低。当本地执行策略的总成本曲线并没有随着 MEC 服务器性能提升而变化。 $f_B > 5 \text{ GHz/s}$ 时系统总成本下降较慢, 且 ODRL 与全卸载策略产生的平均系统总成本趋近。则当 MEC 服务器性能过强时, 系统总成本主要受传输信道条件制约。

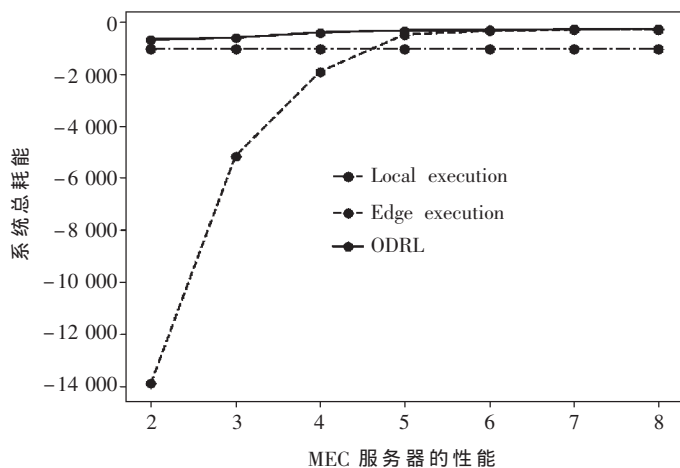


图3 系统总耗能随着 MEC 服务器性能的变化

4 结论

本文通过分析移动设备的时延和功耗, 提出了一种基于强化学习的计算卸载决策算法 ODRL。与两种基准策略相比, ODRL 在仿真实验中达到了最小系统总成

本,可以有效地提高服务质量。在未来研究中,会进一步考虑将算法拓展到多接入边缘计算的场景中。

参考文献

- [1] 施巍松,张星洲,王一帆,等.边缘计算:现状与展望[J].计算机研究与发展,2019,56(1):69-89.
- [2] 施巍松,孙辉,曹杰,等.边缘计算:万物互联时代新型计算模型[J].计算机研究与发展,2017,54(5):907-924.
- [3] MAO Y,ZHANG J,LETAIEF K B.Dynamic computation off loading for mobile-edge computing with energy harvesting devices[J].IEEE Journal on Selected Areas in Communications,2016,34(12):3590-3605.
- [4] ZHANG K,MAO Y,LENG S,et al.Optimal delay constrained off loading for vehicular edge computing networks[C].IEEE International Conference on Communications,2017:1-6.
- [5] KAMOUN M,LABIDI W,SARKISS M.Joint resource allocation and off loading strategies in cloud enabled cellular

networks[C].IEEE International Conference on Communications,2015:5529-5534.

- [6] ZHANG H,GUO J,YANG L,et al.Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC[C].2017 IEEE Conference on Computer Communications Workshops,2017:115-120.
- [7] JIANG Z,MAO S.Energy delay trade-off in cloud off loading for multi-core mobile devices[C].IEEE Globecom,2015:1-6.
- [8] MAO Y,ZHANG J,SONG S H,et al.Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems[J].IEEE Transactions on Wireless Communications,2017,16(9):5994-6009.
- [9] CHEN X,JIAO L,LI W,et al.Efficient multi-user computation offloading for mobile-edge cloud computing[J].IEEE/ACM Transactions on Networking,2016,24(5):2795-2808.

(下转第48页)

(上接第35页)

(3)施工现场监督管理的灵活性。管理人员可以不受时间、空间的限制,利用计算机、手机等终端设备实现施工过程中关键工序、关键工艺的远程监控,施工人员、设备、材料信息的透彻感知,以及安全防护和环境监督等。

(4)业务分析和决策更加及时准确,提升了集成管理和智慧决策水平。

①在物联网技术和人工智能的支持下,管理人员可获取大量、实时、准确的数据,及时得到工地对自身状况的全面反馈。针对风险和隐患信息,通过综合分析得出准确可靠的评价结果,提前预知施工现场潜在的风险,有针对性地制定应急预案,提高业务处理能力。

②云计算平台可完成数据的归集、整合、存储及融合,借助该平台可省去数据的前端预处理,降低后续的数据处理强度,在提高数据处理速度和系统可靠性的同时降低投资成本和维护成本。

③针对不同的业务需求,运用大数据技术从多角度对数据进行分析和挖掘,从而实现数据关联分析、指导施工、风险预测、决策优化等,加强对施工现场的把控能力,预测施工过程潜在风险,提前采取预知性的应对措施,提升智慧决策水平。

④在BIM模型中整合施工过程中的数据信息,工程各参与方可以基于BIM模型进行协同工作,实现工程目标(进度、成本、质量、安全)的统一管理和控制,实现更高效、更节能、更经济的智慧工地管理模式。

4 结论

本文基于现有石油管道工程工地管理需求,构建了基于物联网技术的智慧工地架构,该架构实现了数据的智能采集和处理、施工过程实时监测、施工过程可视化展示和各参与方高效协同作业。通过云计算技术归集、

整合、存储多类数据,大数据技术多角度分析、挖掘数据价值,BIM技术集成各类信息,协同多方人员工作,从而达到信息互通、数据共享、任务协同的目的,在多种新技术的加持下,智慧工地转变了现有工地的管理模式,提高了经济效益,推进现有工地逐步向科学化、智慧化迈进。

参考文献

- [1] 徐友全,贾美珊.物联网在智慧工地安全管控中的应用[J].建筑经济,2019,40(12):101-106.
- [2] 曾凝霜,刘琰,徐波.基于BIM的智慧工地管理体系框架研究[J].施工技术,2015,44(10):96-100.
- [3] 韩豫,孙昊,李宇宏,等.智慧工地系统架构与实现[J].科技进步与对策,2018,35(24):107-111.
- [4] 周小冬,刘民军,鲁万卿,等.基于物联网技术的智慧工地建设[J].施工技术,2018,47(S4):1008-1010.
- [5] 罗雪峰.中俄东线天然气管道工程试验段二期智能工地建设探索[J].石油工程建设,2018,44(S1):214-218.
- [6] 李英攀,史明亮,刘名强,等.基于Cloud-BIM和UWB的施工现场智能安全系统研究[J].中国安全生产科学技术,2018,14(9):151-157.
- [7] 闫文娟,王水璋.无人机倾斜摄影航测技术与BIM结合在智慧工地系统中的应用[J].电子测量与仪器学报,2019,33(10):59-65.
- [8] 范春辉.物联网短距离无线传输技术研究[J].无线互联科技,2017,10(19):23-24.
- [9] 陈海燕.一种分布式大数据管理系统的设计与实现[J].计算机科学,2014,41(S2):393-395.

(收稿日期:2020-04-30)

作者简介:

崔志诚(1992-),男,硕士,助理工程师,主要研究方向:智慧建造。

马胜(1974-),男,本科,工程师,主要研究方向:石油工程建设项目管理。

```

on record data:4096
In:0.00\ 00:27:52.68 [00:00:00.00] Out:73.8M [ ===|=== ]
Hd:3.6 Clip:46.1kon record data:4096
In:0.00\ 00:27:52.90 [00:00:00.00] Out:73.8M [ --|-- ]
Hd:3.6 Clip:46.1kon record data:4096
In:0.00\ 00:27:53.00 [00:00:00.00] Out:73.8M [ =| = ]
Hd:3.6 Clip:46.1kon record data:4096

On directive:{
  "directive":{
    "header":{
      "name":"RenderVoiceInputText",
      "namespace":"ai.dueros.device.interface.screen",
      "messageId":"1557145110_742bvu7hr",
      "dialogRequestId":"wp1557145105466"
    },
    "payload":{
      "text":"播放一段新闻",
      "type":"FINAL"
    }
  }
}
function
mplayer directive openFile
["http://xiaodu.baidu.com/speech/api/media?resource_id=3125cc41e5a3061cd51d12854adce12a&token=4e110745095a6a50011a0e..."]
mplayer directive play
On directive:{
  "directive":{
    "header":{
      "messageId":"YXVkw9fbmV3cysxNTU3MTQ1MTEwXzkxMjk5YzF4ag==",
      "name":"Play",
      "namespace":"ai.dueros.device.interface.audio_player",
      "dialogRequestId":"wp1557145105466"
    },
    "payload":{
      "audioItem":{
        "audioItemId": "9187860899188853997",
        "stream":{
          "expiryTime": "2030-01-01T00:00:00+08:00",
          "offsetInMilliseconds":0,
          "progressReport":{
            "progressReportIntervalInMilliseconds":15000
          },
          "streamFormat": "AUDIO_MPEG",
          "token": "eyJib3RfaWQ1OiJhdwRpb19uZXdzIiw1..",
          "url": "http://xiaodu.baidu.com/speech/api/media?resource_id=3125cc41e5a3061cd51d12854adce12a&token=4e110745..."
        }
      }
    }
  }
}

```

图9 语音终端运行结果

硬件的部署和软件的开发设计,基本实现了能够提供多项技能和生活服务功能的智能家居“魔镜”系统。

从创新和实际使用角度分析,该系统以树莓派为平

(上接第40页)

- [10] HASSIN R, HAVIV M. To queue or not to queue: equilibrium behavior in queueing systems[M]. Kluwer Academic Publishers, 2003.
- [11] LI L, REN J, YAN G, et al. An incentive-aware job offloading control framework for multi-access edge computing[J]. IEEE Transactions on Mobile Computing, 2015, 14(8): 2694-2707.
- [12] TSE D, VISWANATH P. Fundamentals of wireless communication[D]. Cambridge University Press, 2005.
- [13] MIETTINEN P, NURMINEN J. Energy efficiency of mobile clients in cloud computing[C]. HotCloud'10 Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, 2010: 1-7.
- [14] KWAK J, KIM Y, LEE J, et al. Dynamic resource and task allocation for energy minimization in mobile cloud systems[J]. IEEE Areas Communications, 2015, 33(12): 2510-2523.
- [15] SPATARU A, RICCI L, PETCU D, et al. Decentralized cloud scheduling via smart contracts: operational constraints and costs[C]. 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2019.

台,在系统运行、外设扩展和网络性能等方面都要比基于51单片的系统或以Arduino为平台的系统更具有优势,同时在运行系统上采用小巧灵动的Diet-Pi系统,不但具有丰富的软件支持,而且在使用和操作上更加简易和方便,是区别于更多人使用相对厚重的Raspbian系统的一次成功的实验和创新。

参考文献

- [1] 刘历.智能家居服务系统的设计与实现[D].南京:东南大学,2017.
- [2] SMITH S. Raspberry Pi assembly language programming[M]. Apress, 2019.
- [3] Ding Fei, Li Zhiliang, Ai Chengwan, et al. Design of an IoT-based efficient security scheme in home wireless system[M]. Springer International Publishing, 2019.
- [4] 李雪林.基于人机互动的语音识别技术综述[J].电子世界,2018(21):105.
- [5] 林奕欧,雷航,李晓瑜,等.自然语言处理中的深度学习:方法及应用[J].电子科技大学学报,2017,46(6):913-919.
- [6] 单振华,王舒憬,强杰.基于Websocket的智能家居远程控制系统的实现[J].电子技术应用,2017,43(10):124-127.
- [7] 陈永东.基于多类型NAT的TCP穿透技术研究[D].成都:四川师范大学,2016.

(收稿日期:2020-03-11)

作者简介:

潘志倩(1996-),女,硕士,主要研究方向:媒体大数据、舆情分析。

- [16] Song Jinyue. Smart contract-based computing resources trading in edge computing[C]. 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020.
- [17] Xu Chenhan, Wang Kun, Guo Mingyi. Intelligent resource management in blockchain-based cloud datacenters[J]. IEEE Cloud Computing, 2018, 4(6): 50-59.
- [18] Xu Jinlai. Zenith: utility-aware resource allocation for edge computing[C]. 2017 IEEE International Conference on Edge Computing (EDGE), 2017: 47-54.
- [19] Song Jinyue, Gu Tianbo, Ge Yunjie, et al. Smart contract-based computing resources trading in edge computing[C]. 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020.
- [20] Liu Tonglai. Smart contract-based long-term auction for mobile blockchain computation offloading[J]. IEEE Access, 2020, 8(99): 36029-36042.

(收稿日期:2020-09-21)

作者简介:

杨戈(1974-),通信作者,男,博士,副教授,主要研究方向:边缘计算, E-mail: yangge@pkusz.edu.cn。

张衡(1999-),女,本科,主要研究方向:边缘计算。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所