A Matching-theoretic Framework for Consolidation of Flexible Cloud-native Central Units in 5G-RAN

Debashisha Mishra, Himank Gupta, Mehul Sharma, Bheemarjuna Reddy Tamma, and Antony Franklin A
Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, India
Email: [cs15mtech01003, cs16mtech01001, cs17mtech11020, tbr, antony.franklin]@iith.ac.in

***Abstract*—In this study, we investigate the flexible many-to-one mapping of central units (CUs) to compute servers in 5G Radio Access Network (5G-RAN) using two-sided matching theory considering spatio-temporal tidal traffic patterns. Initially, we use a well known bin-packing heuristic known as First Fit Decreasing (FFD) to obtain sub-optimal CU to compute server mapping. To address the traffic heterogeneity, we formulate a novel strategy of dynamic reassignment (known as Machine Admission Game or MAG) among a set of CUs and compute server in each mapping interval, using analytical approaches of coalitional game theory and college admission game. To solve this, we devise a modified version of the classical Deferred Acceptance Algorithm (DAA) satisfying the resource constraints of compute servers. We assess the benefit of the proposed matching theory framework with baseline FFD in terms of compute resource multiplexing gain (i.e., number of active servers in the CU pool) and the number of relocations incurred in the dynamic reassignment of candidate CUs. We observe that the proposed MAG framework though consumes nearly 6.8% and 4.1% of more servers than baseline FFD, it reduces the number of relocations by 33.9% on weekdays and 25.7% on weekends as compared to FFD.**

I. Introduction

In a traditional mobile network deployment, the RF, amplifier, and signal processing components are linked as a fixed allocation of $1:1$ (one-to-one) mapping between signal processing functions and hardware in a close proximity at the cell site. Techniques to separate signal processing functions from antenna and power amplifier are being studied in the literature [1] [2]. Next Generation Radio Access Network (NG-RAN, a.k.a. 5G-RAN) is a competitive cellular network architecture which aims to provide significant opportunistic gains related to infrastructure wide cost elements (CAPEX), maintenance, energy, and operational cost savings (OPEX) [3]. Existing cellular base station functionalities are segregated into cheaper and smaller footprint remote radio units (DUs) present at cell sites and compute-intensive baseband functionalities known as Central Units (CUs) that are moved to CU pool (commonly referred as BBU pool in C-RAN literature) where cloud computing and virtualization mechanisms are used as the key enabling technologies. With the flexibility in the segregation and deployment of RAN functions between CU and DU, NG-RAN envisioned in 5G architecture provides a substantial advantage to cellular operators as well as to mobile subscribers [4]. CU is accepted as a generic term after 3GPP Rel-14 technical specification which supports the concept of flexible centralization of base station functionality. With Network Functions Virtualization (NFV) being chosen as a potential choice to implement CU, NG-RAN enables dynamic coordination among cells which are distributed across geographical locations.

A. Background & Motivation

The end-users within the coverage of a typical DU have different traffic demand pattern following the diurnal tendency of human beings. In general, the traffic demands at DU are highly uncorrelated over time, which means that all cell sites do not experience peak traffic hours at the same time [5]. Therefore, the computational resource requirements of CU in terms of CPU cores, memory, I/O, network bandwidth, and disk space vary dynamically w.r.t. traffic conditions [6]. Henceforth, we refer this dynamic CU resource requirement as the compute load or simply load. Fig. 1 shows the NG-RAN architecture considered in our work in compliant with the reference architecture of ETSI NFV. Each CU serves a designated DU and connects to the core network (5G-Core). From an implementation perspective, CU can be realized using abstraction mechanisms such as containers (*i.e.,* Docker/LXC) or hypervisors (*i.e.,*VM) on a shared infrastructure layer in the data center [7]. Typically, multiple virtualized CU instances are deployed on a single compute server because of the isolation flexibility of cloud platform, thereby creating a many-to-one deployment relationship between CU and compute

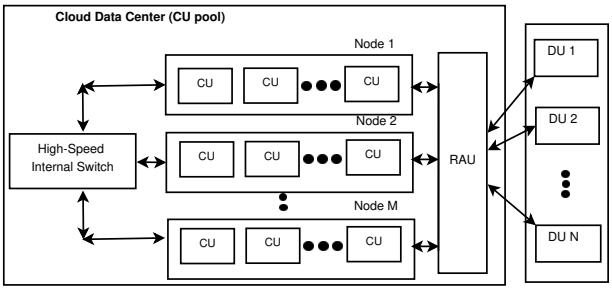

Fig. 1: Programmable & Virtualized Computing Center for CUs in NG-RAN.

server. This many-to-one deployment of CUs to compute server is known as "consolidation" of CUs. An allocation matrix A_t can potentially represent the consolidation at time t where each row corresponds to CU and each column represents the servers. An entry $A_t(i, j)$ is 1 if CU i is hosted on server j, else 0. When traffic demand rises or falls, the allocation matrix can be re-formed accordingly. Assuming a scenario without virtualized CUs, we might need to allocate one compute server for one CU *i.e.,* $1:1$ case which underutilizes the computational resources of the data center. This inefficient $1:1$ allocation of CU to compute server increases the CAPEX and OPEX of the mobile network operator.

Most of the time in a day, the sum of all individual offered compute load at CUs from each of the corresponding DU will be less than the sum of their peak loads at any given time instant. For example, the allocation matrix at time t may not be a good allocation at time $(t + 1)$. Hence, the compute loads of different CUs hosted on the same compute server may sometimes overload beyond server capacity, thereby degrading the Quality of Service (QoS) for end-users. By intelligently relocating CU instances across available compute servers in the CU pool, suitable mitigation plan can be devised in response to server overload situation to minimize the service disruption. This phenomenon is known as "relocation" of candidate CUs. In a given interval, known as the periodicity of CU clustering or epoch, the consolidation and relocation of CUs (which results in the re-formation of allocation matrix) significantly lower the net computational load on a cloud platform, saving energy by switching-off idle compute servers. Relocation of virtual resources (*i.e.,* VMs or containers) incurs additional computational resource overhead because it involves iteratively writing all the memory copy operations and transfers over a communication network from one server to another server. Therefore, the softwarized and virtualized CU instance relocations must be triggered in a controlled manner with consideration of additional overhead cost to model the performance accurately.

In this work, our research goal is to adaptively consolidate a set of CU compute loads to compute servers in the CU pool so as to improve the overall resource utilization of the cloud platform using two-sided matching theory. The proposed solution to this optimal allocation problem is a dynamic CU-compute server mapping framework which automatically adapts to traffic heterogeneity from DUs. The proposed solution minimizes the service disruption by carefully planning the relocations decisions for CUs, thereby improving the QoS for the end-users.

B. Contributions

We present three main contributions in developing the matching-theory framework for efficient and scalable CU-compute server mapping for 5G-RAN. They are summarized as follows.

1) By using analytical approaches from two-sided matching theory, we propose a light-weight and scalable Machine Admission Game (MAG framework), a novel way of reassigning multiple CU compute loads among available server resources. This striking analogy is a variant of popular College Admission Game model [8] discussed in game theory literature. In this game formulation, the set of players *i.e.,* CU compute loads imitate jobs of various sizes and the second set of players *i.e.,* compute servers imitate the machines that process jobs.
2) We derive a preferential classification of players, a rank-ordered list, which enables players of one set to express priorities among all the possible matches in the other set. This ranking is decided based on the individual player's choice of getting matched to its counterpart maximizing a known, often conflicting objective function.
3) We devise a modified version of classical Deferred Acceptance Algorithm (DAA) [8] for solving the proposed MAG framework and assess its performance with respect to cloud resource multiplexing gain and additional overhead incurred in dynamic CU relocation.

The rest of the paper is organized as follows. We highlight the related works in Section II. In Section III, we present the system model. In Section IV, we present the proposed two-sided matching theory framework. Experimental setup and performance results are presented in Section V. Finally, concluding remarks are given in Section VI.

II. Related Work

Both consolidation and relocation mechanisms are extremely vital for efficient resource planning in 5G RAN. Although RAN resource management (C-RAN) in a data center is still in its infancy, few of the existing resource management works are based only on the "bin packing" approach or stochastic modeling to computational resource consolidation process. In [9], the authors presented a multi-dimensional Markov model to evaluate the statistical multiplexing gain (denotes the extent to which the resources can be shared across multiple parties) of Virtual Base Station (VBS) pools considering the user session level traffic dynamics. Although this model considers the delay-tolerant traffic and expressions for blocking probability, the performance w.r.t. spatio-temporal traffic fluctuations are not factored in the gain calculation. In [10], the authors proposed a bin packing formulation to the BBU to VM packing on an iterative approach to minimize the total number of active BBUs. However, they did not consider the relevance of BBU relocations in accordance with tidal traffic variation. The authors in [11] proposed a bin packing solution to consolidate BBUs, which minimizes the energy

consumption without factoring the BBU (VM) migration scenario as described before. On similar notion, authors in [12] highlights a dynamic DU reassignment algorithm (synonymous with the concept of CU migration) which minimizes the total number of active servers in the cloud platform by considering the spatio-temporal traffic variation, but without factoring relocation overhead. In this current work, these limitations are overcome by factoring the relocation overhead in addition to CU consolidation.

III. System Model

Let $\mathcal{D} = \{d_1, d_2, \ldots, D\}$ be the finite set of D DUs, $\mathcal{C} = \{c_1, c_2, \ldots, C\}$ be the finite set of C CUs, and $\mathcal{S} = \{s_1, s_2, \ldots, S\}$ as the finite set of S compute servers in the data center, respectively. Since each DU traffic is served by its corresponding CU, we have $D = C$. Let $\mathcal{U} = \{u_1, u_2, \ldots, U\}$ be the set of U users served by D DUs. As shown in Fig. 2, the deployment of DUs is assumed as per Matern hard core point process type II (MHCPP II) and the deployment of UEs as per Poisson Point Process (PPP) [13]. Suppose, spatial distribution of DUs and UEs be Φ_{DU} and Φ_{UE}, respectively, where $\{\Phi_{UE}, \Phi_{DU}\} \in \mathbb{R}^2$. The coverage regions of DUs are plotted and depicted by voronoi tessellation.

IV. Matching Theory Framework

Two-sided matching theory is a Nobel prize-winning framework, originally studied in economics, but can also be applied to several engineering disciplines, especially in solving wireless resource allocation problems [14]. In general, basic framework of resource allocation problem involves resources and users. Depending upon the context, the resources can be viewed from different abstractions such as compute resource, power, time-frequency chunks, eNodeBs, etc. Users can be smartphones, wireless stations, etc. A matching is a mapping between users and resources given the individual player's preferences, often having conflicting objectives. As per player's quota limit, the matching game can be :-

- One-to-One (Stable Marriage Model)
- Many-to-One (College Admission Game Model)
- Many-to-Many (Consultant Firm Matching Model)

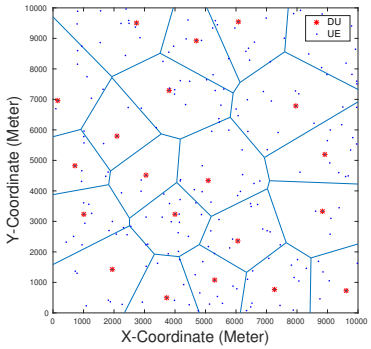

Fig. 2: Spatial Distribution of UEs and DUs as per MHCPP II.

In the following subsection, we briefly introduce Many-to-One College Admission Game and the viable solution framework to solve this class of matching game.

A. College Admission Game

The College Admissions Game consists of two disjoint sets of players: a set of students M and a set of colleges N. Each student $m \in M$ has a strict, preference relation over the set N. Similarly, each college $n \in N$ has a strict preference relation over M. Every college $m \in M$ has a finite quota of maximum allowable students it can take. Let us denote this quota as $q_c \in \mathbb{Z}^+$. The solution to this game is essentially a many-to-one stable assignment from students to colleges.

1) Game Model & Deferred Acceptance Algorithm (DAA): Gale and Shapley [8] first studied this game where they proposed a Deferred Acceptance Algorithm (DAA) to find stable matching between colleges and students. This algorithm is popularly known as the Gale-Shapley (GS) algorithm and the overall procedure is pictorially summarized in Fig. 3. In student proposing version of this algorithm, students first propose to their favorite colleges. If the college has not fulfilled its quota and has at least one vacant seat to admit the student, it accepts the student's proposal. If no more seat is available for this student and student is least preferred among all the students that are currently matched to this college, then college rejects the proposal. In case, the student deems fit than other matched student to the same college, the college offers admission to the student applicant and unmatch from the previous acceptance, which is least preferred. However, this model cannot be applied directly to the context of CU-compute server mapping due to the challenges discussed as follows.

2) Challenges for DAA in CU-compute server Mapping: In many-to-one College Admission game model, each college is bounded by some fixed quota up to which it can accept students. The players in student set are matched to exactly one college and a college can admit multiple students, not violating its quota limit. Although DAA is proved to be an efficient and stable algorithm to solve this type of game model, it cannot be applied directly to our CU-compute server assignment scenario in NG-RAN since we cannot define the quota for servers in terms of the number of compute loads to be processed. Also, compute loads are heterogeneous that vary as a function of space and time. Each server has

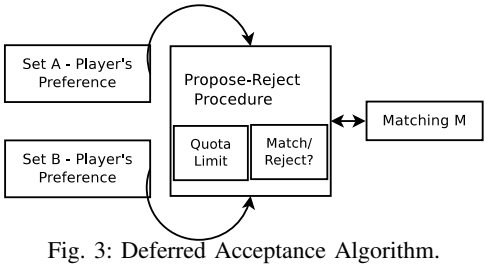

Fig. 3: Deferred Acceptance Algorithm.

a fixed capacity in terms of compute resources. Hence, it can accommodate multiple compute loads until their total sum remains lower than the rated server capacity. Moreover, traffic heterogeneity from DUs makes this matching problem very challenging to solve.

B. Machine Admission Game (MAG)

To address the special case of CU-compute server assignment problem in NG-RAN, we propose a novel strategy of Machine Admission Game (MAG) model based on a two-sided many-to-one matching framework. The players are divided into two distinct sets. They are:

- Load set consisting of heterogeneous compute loads from Candidate CUs (i.e., *RelocationCandidates*)
- Set of compute servers of the CU pool (i.e., non-overloaded servers)

1) Rank Ordering via Preference Relation: A preference relation $\succ$ is a strict, complete binary relation over the set of all players where players of one set rank another player of other set satisfying individual incentives of being matched. In the game formulation of MAG, each compute server derives a strict ranking over all the acceptable CUs whose compute load is less than the host server capacity. Similarly, each CU will prefer a server whose available compute resources are sufficient enough to serve all its users covered by the corresponding DU. Let $r(s)$ be the available virtual compute resources of server s and $s(c)$ be the size of the compute load at CU c. Additional relevant notations are summarized in Table I. For a server $s \in \mathcal{S}$, we define the preference relation $\succ_s$ of compute server s over set of CU compute loads such that for any two CU loads $c_1, c_2 \in \mathcal{C}$,

$$c_1 \succ_s c_2 \Leftrightarrow s(c_1) \geq s(c_2) \tag{1}$$

Similarly, for a CU compute load $c \in \mathcal{C}$, we define the preference relation $\succ_c$ of CU load c over set of compute servers such that for any two servers $s_1, s_2 \in \mathcal{S}$,

$$s_1 \succ_c s_2 \Leftrightarrow r(s_1) \geq r(s_2) \tag{2}$$

Eqn. 1 ensures that server always prefers CU loads that are the best fit to the available compute resources. Eqn. 2 ensures that CU loads prefer the servers whose available compute resources are high enough following a worst fit assumption from compute load set. To summarize, CU prefers a server based on the worst fit scheme of available compute resources and a server prefers a CU based on the best fit scheme of available compute resources. The proposed MAG, by using a modified version of DAA, efficiently matches CUs to servers, satisfying the player's conflicting objectives as mentioned above.

TABLE I. Notations for Machine Admission Game.

Symbol	Definition
$\mathcal{S}$	Set of all compute servers
$\mathcal{C}$	Set of all CU compute loads
$s(c)$	Size of CU load c, $c \in \mathcal{C}$
$l(s)$	Load on server s, $s \in \mathcal{S}$
$peak(s)$	Peak capacity of server s
$pref(c)$	Preference list of CU load c
$pref(s)$	Preference list of server s

Algorithm 1 : Modified DAA for MAG

INPUT: Previous allocation matrix A_{t-1}.
OUTPUT: Best possible allocation matrix A_t at time t.

```
For each CU load c, c ∈ C, Status(c) ← "unmatched"
procedure MACHINEADMISSIONGAME
    while ∃c ∈ C and status of load c is "unmatched" do
        ptr ← First element of list pref(c)
        while ptr ≤ length of pref(c) do
            s' ← Most Preferred server of load c at ptr
            if s' can serve c then
                Match load c to server s'
                Status(c) ← "matched"
                Break loop & proceed to next free load
            else
                Get candidate loads c' ∈ C s.t. c ≻_s' c'
                if s' can serve c by rejecting loads then
                    Assign load c to server s'
                    Status(c) ← "matched"
                    ∀c' ∈ C, Status(c') ← "unmatched"
                    Break loop & proceed to next free load
                else
                    Reject load c
                    Go to next preferred server in ptr
                end if
            end if
        end while
    end while
    Return all the matched CU-compute server pairs, A_t
end procedure
```

2) Modified DAA for MAG: The proposed algorithm starts with two distinct sets of players: a set of CU loads $\mathcal{C}$ and a set of servers $\mathcal{S}$. Initially, all the players of both the sets are initialized with unmatched (free) status. We consider $|\mathcal{C}| = |\mathcal{S}|$, i.e., we have ensured to allocate $|\mathcal{S}|$ servers in the worst case which corresponds to the fully-loaded cell sites in case of traditional cellular network, but in the realistic case, the count of servers is much less than $|\mathcal{S}|$ due to spatio-temporal traffic inhomogeneity of mobile subscribers. Each CU load $c \in \mathcal{C}$ starts by proposing to its most preferred server $s \in \mathcal{S}$ based on the rank-order computed by the CU over all the elements of set $\mathcal{S}$. If server s has enough available computational resource to meet the requirement of CU load c, then s accepts the proposal from c. If server s does not possess sufficient resource and unable to hold the current proposal offer, two cases may arise.

- **Case 1:** If all the loads in the current matching list of server s are more preferable than CU load c, then s rejects the proposal from CU load c.
- **Case 2:** If server s has less preferable compute loads than CU load c in its matching list, then two subcases may arise.

TABLE II. Time Complexity Analysis.

Test Input	FFD	MAG
Best Case	$O(\|C\|log\|C\|)$	$O(\|C\|)$
Average Case	$O(\|C\|^2)$	$O(\|CS\|)$
Worst Case	$O(\|C\|^2)$	$O(\|CS\|)$

- **Sub case 2.1:** By removing all or some least preferred loads than c from its current matching list (candidate loads for rejection), if server s acquires sufficient resources to serve j, then unmatch those candidate loads from server s and match load c to server s.
- **Sub case 2.2:** By removing candidate loads, if server s still does not have enough resources to serve CU load c, then it rejects c.

The above propose-reject sequence is repeated for all the unmatched CU loads until they are matched to compute servers. Note that some of the servers may have an empty matching list as there are no outstanding loads to be processed by the algorithm. Those compute servers are declared as idle/inactive and are potential candidates to be switched off to save energy and compute resources.

3) Time Complexity Analysis: Assuming $|C|$, the number of CU compute loads and $|S|$, the number of server hosts ($|S| \ll |C|$), the time complexity analysis for FFD and MAG schemes are summarized in Table II. FFD requires all CU loads to be sorted in non-increasing fashion, thus taking $O(|C|log|C|)$ time in the best case. However, MAG does not perform any CU reassignment in the best case if there are no overloaded clusters, but it has to scan all the CU loads which take $O(|C|)$. Every CU load may demand one server with $O(|C|^2)$ number of comparisons in the worst case in FFD. In MAG, each CU load has to propose at most $|S|$ servers in the preference list requiring time $O(|CS|)$. In FFD average case, considering $\frac{|S|}{2}$ clusters are overloaded, the number of comparisons are at most $O(|C|^2)$. In MAG, the average case time complexity is $O(|CS|)$ as each CU load may need to propose at least $\frac{|S|}{2}$ servers.

V. Performance Results

Considering the downlink transmission processing in a typical NG-RAN environment, we generate spatio-temporal varying DU workloads by a modeling function highlighted in [15]. In this model, a Gaussian Mixture Model (GMM) is used to model the behavior of spatio-temporal traffic variation and cell-specific DU loads at a given instant of time which is exponentially distributed in the spatial domain. Based on this, we obtained two extensive datasets for synthetic DU workloads for both weekday and weekend traffic profiles spanning 24 hours. User traffic generated during a day is divided into three segments. They are "Low Load" from 12AM to 8AM, "Medium Load" from 8AM to 12 Noon and 8PM to 12AM, "High Load" from 12 Noon to 8PM. With a sampling interval of 6 minutes, we generated 10 samples per hour and a total of 240 samples for 24 hours.

TABLE III. Simulation Parameters

Parameter	Value
Number of DUs	[100 1000]
Sampling Interval	6 Minutes
Total Traffic Capture Duration	24 Hours
Total Generated Samples	240
Traffic profiles	Weekday & Weekend
Network Region	Urban
DU workload Range	Normalized in [0,1]
Peak DU Load	1 (100%)
Spatial load distribution	Exponential
Time-varying rate parameter	Gaussian Mixture Model

To capture the geographical randomness and large-scale spatial deployment scenario, we ran the simulations from 200 DUs to 1000 DUs. The system-level simulations with modified DAA are performed in Intel x86, eight-core, Ubuntu Linux 64-bit distribution with 2.4 GHz processor and each independent trial is accompanied with 15 random seeds. We have evaluated the performance of the proposed MAG framework w.r.t (a) resource multiplexing gain, (b) relocation overhead. Table III briefly summarizes the simulation parameters.

Note that, First Fit decreasing (FFD) is a well-known bin-packing heuristic algorithm most commonly used in data center environments to map VMs to servers. In [11] and [16], the authors have also used this packing algorithm to implement the CU consolidation process. Authors in [12] highlights an extended version of FFD (known as DRA), that performs the consolidation and relocation of CUs. Hence, in this work, we have utilized FFD and DRA as the baseline schemes to showcase various trade-off features and efficacy of our proposed MAG framework.

A. Resource Multiplexing Gain

We assess the performance of MAG framework in terms of compute resource multiplexing gain *i.e.,* the metric denoting the number of active servers in the cloud platform. MAG framework with modified DAA is compared with baseline FFD and DRA schemes. Figs. 4 and 5 show the required number of active servers in a time scale of 24 hours of the day from 00:00 hours up to 23:59 hours for weekday and weekend traffic profiles, respectively. With respect to one-to-one CU to server mapping scheme, the server savings in MAG is nearly 86%. The savings offered by MAG in closely aligned with savings obtained by the FFD scheme but overestimates by 6.8% and 4.1% on weekday and weekend, respectively.

B. Relocation Overhead

Although the number of active servers required in MAG and DRA is slightly higher than that in FFD, the relocation overhead in FFD is higher compared to MAG. We evaluated the relocation overhead for all the mapping schemes in Figs. 6 and 7 for weekday and weekend, respectively. For a weekday traffic profile, MAG incurs 33.9% fewer relocations than FFD and

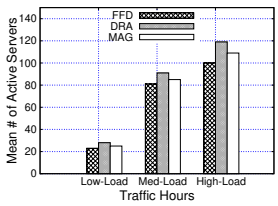

Fig. 4: Mean Number of Active Servers for 1000 DUs for a Weekday Traffic.

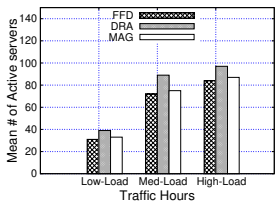

Fig. 5: Mean Number of Active Servers for 1000 DUs for a Weekend Traffic.

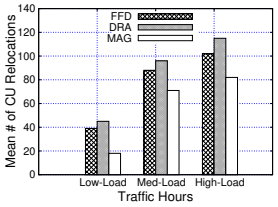

Fig. 6: Mean Number of CU Relocations for a Weekday Traffic.

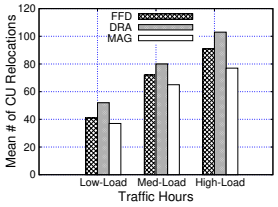

Fig. 7: Mean Number of CU Relocations for a Weekend Traffic.

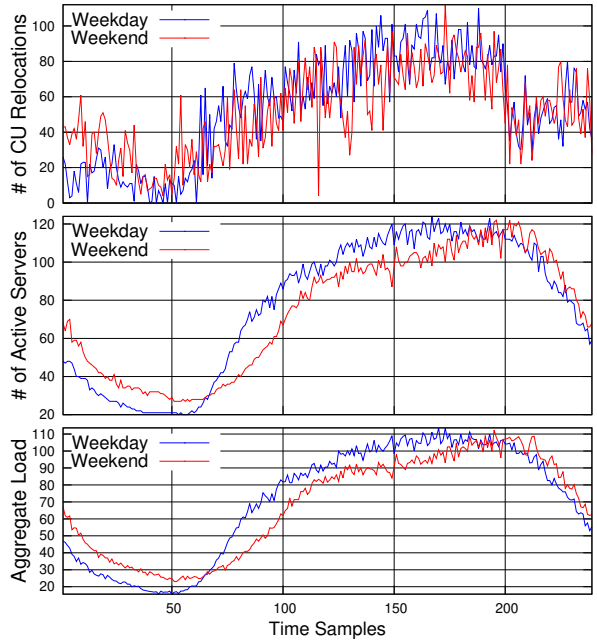

Fig. 8: Variation in Load, # of Active Servers, and # of CU Relocations during a Day in MAG for 1000 DUs.

44% fewer relocations than DRA. For a weekend traffic profile, MAG incurs 25.7% fewer relocations than FFD and 36.33% fewer relocations than DRA.

VI. Conclusions and Future Work

We have presented a two-sided matching theory framework known as Machine Admission Game (MAG) for dynamic many-to-one mapping between CUs and compute servers in NG-RAN. Compared to the existing algorithms of bin packing and DRA, MAG excels in green computing aspects (i.e., minimizing number of active servers) of 5G-RAN as well as reduces the total number of CU relocations to ensure better QoS for the end-users. In future, we aim to incorporate a workload predictor to MAG framework and provisioning of on-demand reservation of servers in order to study and analyze the compute resource usage.

Acknowledgment

This work was supported by the project "Energy Efficiency in Converged Cloud Radio Next Generation Access Network", Intel India.

References

[1] G. Faraci, A. Lombardo, and G. Schembra, "A Building Block to Model an SDN/NFV network," in *IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2017.

[2] E. J. Kitindi, S. Fu, Y. Jia, A. Kabir, and Y. Wang, "Wireless Network Virtualization With SDN and C-RAN for 5G Networks: Requirements, Opportunities, and Challenges," *IEEE Access*, vol. 5, pp. 19099–19115, 2017.

[3] M. Simsek, D. Zhang, D. Öhmann, M. Matthé, and G. Fettweis, "On the Flexibility and Autonomy of 5G Wireless Networks," *IEEE Access*, vol. 5, pp. 22823–22835, 2017.

[4] P. Marsch, I. D. Silva, O. Bulakci, M. Tesanovic, S. E. E. Ayoubi, T. Rosowski, A. Kaloxylos, and M. Boldi, "5G Radio Access Network Architecture: Design Guidelines and Key Considerations," *IEEE Communications Magazine*, vol. 54, pp. 24–32, November 2016.

[5] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding Traffic Dynamics in Cellular Data Networks," in *IEEE INFOCOM*, pp. 882–890, 2011.

[6] D. Mishra, H. Gupta, B. R. Tamma, and A. A. Franklin, "KORA: A Framework for Dynamic Consolidation & Relocation of Control Units in Virtualized 5G RAN," in *IEEE International Conference on Communications (ICC)*, pp. 1–7, 2018.

[7] S. S. Kumar *et al.*, "FLEXCRAN: Cloud Radio Access Network Prototype using OpenAirInterface," in *IEEE COMSNETS Demos & Exhibits,*, pp. 421–422, Jan 2017.

[8] D. Gale and L. S. Shapley, "College Admissions and the Stability of Marriage," *The American Mathematical Monthly*, vol. 69, no. 1, 1962.

[9] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Statistical Multiplexing Gain Analysis of Heterogeneous Virtual Base Station Pools in Cloud Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5681–5694, 2016.

[10] N. Yu, Z. Song, H. Du, H. Huang, and X. Jia, "Multi-Resource Allocation in Cloud Radio Access Networks," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.

[11] K. Wang, W. Zhou, and S. Mao, "On Joint BBU/RRH Resource Allocation in Heterogeneous Cloud-RANs," *IEEE Internet of Things Journal*, vol. 4, pp. 749–759, June 2017.

[12] D. Mishra, P. Amogh, A. Ramamurthy, A. A. Franklin, and B. R. Tamma, "Load-aware Dynamic RRH Assignment in Cloud Radio Access Networks," in *Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2016.

[13] H. ElSawy *et al.*, "Modeling and Analysis of Cellular Networks Using Stochastic Geometry: A Tutorial," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 167–203, Firstquarter 2017.

[14] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching Theory for Future Wireless Networks: Fundamentals and Applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, 2015.

[15] E. Nan, X. Chu, W. Guo, and J. Zhang, "User Data Traffic Analysis for 3G Cellular Networks," in *CHINACOM*, pp. 468–472, IEEE, 2013.

[16] T. Sigwele, A. S. Alam, P. Pillai, and Y. F. Hu, "Evaluating Energy-Efficient Cloud Radio Access Networks for 5G," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pp. 362–367, Dec 2015.