Game Theoretic Approach for Real-time Task Scheduling in Cloud Computing Environment

Manoj Kumar Patra, Sampa Sahoo, Bibhudatta Sahoo, Ashok Kumar Turuk
Department of Computer Science & Engineering
National Institute of Technology
Rourkela, India
{manojpatra.in, sampaa2004, bibhudatta.sahoo, akturuk}@gmail.com

***Abstract*—Cloud computing is one of the emerging technology in the field of distributed computing and is designed as per the requirement and demand of the user. It uses the virtualization technique to create multiple virtual machines that are the basis of computation in the cloud. One of the major issues in cloud computing is to efficiently schedule the tasks and completing their execution before the deadline to maximize the utilization of the processor, maximize the throughput and reduce the waiting time of the task. In this paper, first, we propose a system model and proposed a game-theoretic framework for scheduling real-time tasks in the cloud computing environment to reduce the total completion time and total waiting time. In our game model task act as a player, the virtual machine act as a strategy and the payoff of the player is represented by completion time and waiting time. We have performed our experiments using the non-cooperative and cooperative game model. Our experimental results show that the total execution time and total waiting time are less in the cooperative game model than a non-cooperative game model.**

***Index Terms*—Cloud Computing, Game Theory, Cooperative, Non-cooperative, Task Scheduling.**

I. Introduction

Virtualization is the most fundamental technique used in cloud computing, which allows creating multiple virtual machines(VMs) in a single physical machine. In other words, cloud computing integrates the traditional high-performance computing distributed with the virtualization technique. Further, the ease of availability and computing capability of the cloud environment makes it useful for the deployment of various applications. Among these, some applications need a response within a given timing constraint or deadline, and the task generated by the applications are called real-time tasks.

Task scheduling is an efficient method to improve system performance, but it becomes cumbersome for the real-time task due to the timing limit. Task scheduling is the process of executing tasks(requests) with limited available resources(virtual machines) such that improvement in certain system performance can be achieved [1]. The absence of an efficient scheduling algorithm may lead to inefficient resource utilization, longer waiting times, high deadline miss, etc. A lot of work has been done in task scheduling field of cloud computing, but the development of new applications and technology makes it an open research challenge.

Even though there is an illusion that the cloud has unlimited resources, sometimes several tasks have to compete for limited resources. From the cloud service provider point of view, whether task competes for resources or resources compete for the task, the scheduling must improve the profit. Whereas, from the user point of view, the scheduling must be done in such a way that users' requirement is optimized with less overhead. The decision made by a cloud broker or scheduler to generate (task,VM) pair must be efficient and appropriate to achieve the above-mentioned goal. In this regard, we can make use of game theory, a branch of applied mathematics. Game theory is the study of the interaction between independent and competing players or decision-makers, that helps to obtain optimal decision [5]. Based on the interaction, the game model can be non-cooperative or cooperative. In this paper, we have designed a scheduling framework for real-time task considering both non-cooperative and cooperative game model. We have used total completion time and total waiting time as performance metrics.

The main objective of this article is to put forward a Game Theoretic approach for scheduling real-time tasks in the cloud computing environment. This paper is assembled as follows: we have presented a short introduction to cloud computing and task scheduling in Section-I, in Section-II, a quick review of works done in the field of task scheduling is discussed. In section-III, we have presented our system model that includes the VM model, task model, and scheduling model. In section-IV, a game-theoretic framework for real-time task scheduling is presented which includes a description of the scheduling mechanism using a cooperative game and non-cooperative game. In Section-V, the experimental result is discussed and finally, Section-VI summarizes the work done in this paper and draws the conclusion and future works.

II. Related Works

A lot of researchers have been studying the problem of scheduling tasks in the cloud environment in recent years. This study specifically focused on task scheduling based on the game theory model. In [2] the author has designed a cooperative game model to address task scheduling problems and to achieve energy efficiency in the cloud environment. Researchers in [3] developed a bottom-up Game-theoretic Task Allocation (BGTA) framework for the allocation of social sensing tasks to non-cooperative edge computing nodes. They aim to ensure QoS requirement of applications while optimizing the payoffs to the edge node. Zhang et al. [4] used a game-theoretic mathematical model to map a task to an optimal machine in cloud manufacturing. As reliability is an important feature of the cloud system in [5], a non-cooperative game model is designed for reliability-based task scheduling in the cloud system. A framework is proposed in [6] using the game theory concept for real-system, that helps to identify a worst-case scenario in the real-systems.

The author in [7] proposed an enhanced HEFT algorithm to achieve load balancing across VMs while minimizing makespan under certain budget constraints. The thermal imbalance in the data center impacts the efficiency of the cooling mechanism, which in turn increases energy consumption. In this context, Akbar et al. [8] proposed a cooperative game based thermal aware resource allocation to enhance the thermal balance.

In [9], researchers proposed a two-phase scheduling algorithm to reduce the execution cost of work-flow applications. The author in [10], presented a task-oriented resource allocation method, where the task is ranked by the pairwise comparison matrix and allocated the computing resources based on this rank. A learning automata concept is used in [11] to schedule real-time task in a heterogeneous cloud environment. Authors in [12], presented a comparison of the fairness of task scheduling approaches in high-performance computing environments. Su et al. in [13], proposed a Stackelberg game based resource allocation scheme to allocate mobile social user in the media cloud while obtaining maximum revenue.

Game theory is also being used in resource adjustment in cloud. Yan Gao et al. in [14], presented a game theory method for adjusting cloud resources. They have used fireworks algorithm to adjust the performance of SOA based cloud application that gives the method for utility calculation. Soamar Homsi et al. in [15], proposed a two person zero-sum game for static virtual machine allocation. They have modeled the game by considering attacker and provider as two player of the game. The optimal solution is being obtained by using mathematical programming approach and them, identify the strategy using proposed game.

The above study shows that a lot of work is done in designing a game theoretic based task scheduling framework. But, very few works concentrate on real-time task scheduling. In this regard, we proposed a scheduling framework for a real-time task using game theory concept.

III. System Model

In this section, we will discuss the system model of our proposed approach for real-time task scheduling. It is quite obvious that the architecture of a cloud computing system mainly includes three things; virtual machine, scheduler, and task. The scheduler act as an intermediary between tasks and virtual machine. User represents the source of task generator and Virtual Machine is responsible for executing users' request(task).

A. Virtual Machine Model

The cloud environment is having a set of virtual machines $V = \{v_1, v_2, ..., v_m\}$ which is responsible for providing computing infrastructure for user tasks. Each virtual machine has its computing capability i.e. speed of execution, sp_j, $j = \{1, 2, ..., m\}$. The computing capability of the virtual machine is measured using the widely-used metric, Million Instructions Per Second (MIPS).

B. Task Model

The user generates the task and forwards it to the scheduler. Let $T = \{t_1, t_2, .., t_n\}$ is the set of independent tasks generated by different users. Each task t_i, $i = \{1, 2, .., n\}$ in T is modeled by following parameters; $t_i = \{a_i, s_i, d_i\}$, where a_i, s_i, and d_i are the arrival time, task size (in terms of Million Instruction(MI)), and deadline of the i^{th} task respectively. The time at which a task becomes ready for the execution is termed as the arrival time of the task. Time at which a task finishes its execution is termed as completion time. The expected execution time (et_{ij}) of the task t_i on v_j is computed as:

$$et_{\text{ij}} = \frac{s_i}{sp_j} \tag{1}$$

Let, scheduling time (st_{ij}) is the time at which v_j is assigned task t_i. Then, the completion time (c_{ij}) of t_i on v_j is computed as:

$$c_{\text{ij}} = st_{\text{ij}} + et_{\text{ij}} \tag{2}$$

The waiting time of the task t_i is formulated as:

$$w_{\text{ij}} = st_{\text{ij}} - a_i \tag{3}$$

C. Scheduling Model

The primary job of the scheduler is to receive the tasks coming from a set of users and then map them to a set of virtual machines. Here, we developed a game-theoretic scheduling framework to generate a mapping between tasks and VM. A two-player game is played to select an appropriate VM for a task. First, the tasks are sorted based on their deadline. Since there are many tasks and a two-player game model is used, we considered k number of scheduler. Each scheduler is responsible for playing two tasks at a time and at most $k = \frac{n}{2}$ number of schedulers is possible. The proposed system model is shown in Fig-1.

IV. Game Theoretic Framework

In this section, the game formulation for real-time task scheduling is discussed. Game theory attempts to look at the relationship between participants in a particular model and predict their optimal decision. A game is an interaction among multiple players in which each players' payoff is affected by the decision made by other players. A game involves some players, a predefined set of strategies for each player and a payoff that describes the outcome of each player in terms of the amount they win or lose. Various components of a game model are:

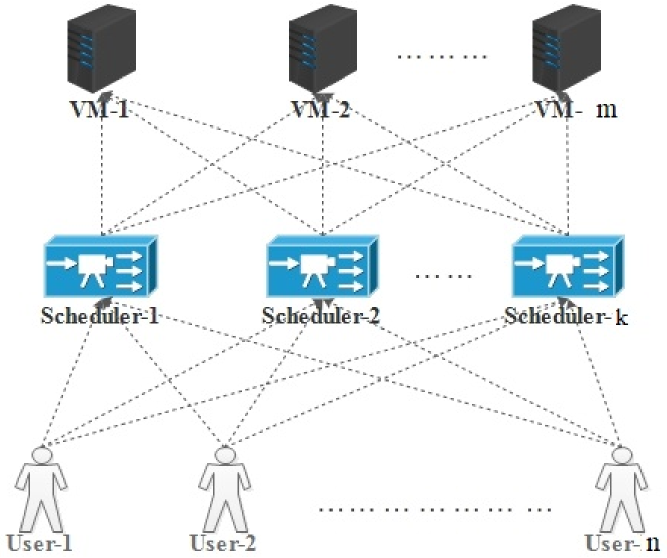

Fig. 1. System Model

t2 t1	V_1	V_2	V_3
V_1	(4, 3)	(1, 2)	(3, 2)
V_2	(5, 2)	(2, 7)	**(8, 4)**
V_3	(5, 9)	(3, 1)	(4, 6)

Fig. 2. Payoff Matrix

t2 t1	V_1	V_2	V_3
V_1	1	-1	1
V_2	3	-5	4
V_3	-4	2	-2

Fig. 3. Normalized Payoff Matrix: M

- Player: A strategic decision maker within the context of the game.
- Strategy: It defines the actions of players in a given circumstance.
- Payoff: It quantifies the benefits or loss of a player based on a particular outcome of the game.

Game theory has two main branches:

- Non-cooperative game: In the non-cooperative game, the players competitively interact with each other where there will be some gainers and some losers. The players compete with each other and choose the course of actions that benefits them the most, no matter what other player decides to do.
- Cooperative game: In a cooperative game, every player has agreed to work together towards a common goal.

Here, we consider both a two-player non-cooperative and cooperative game model for real-time task scheduling. Different attributes of the proposed game model are discussed below.

- Player: Each task is taken as a player i.e. $t_i = p_i$.
  The set of player is represented by
  $$P = \{p_i | 1 \leq i \leq n\}$$
- Strategies: All virtual machines act as strategies. All players will try to find the best strategy (v_j) to maximize their payoff.
- Payoff: The variable payoff can be either completion time (c_{ij}) or waiting time (w_{ij}) that t_i is going to acquire while selecting a strategy v_j.
- Payoff Matrix: The payoff matrix is a table in which each row represents the strategies of player-1 and each column represents the strategies of player-2. In proposed approach, t_1 is represented by player-1 and t_2 is represented by player-2. Strategy is represented by v_j , $j = 1, 2, .., m$. Each cell in the table contains two numbers, the first one represents the payoff of t_1 and the second number represents the payoff of t_2. An example of a payoff matrix with $m = 3$ is shown in Fig. 2. If t_1 selects v_2 and t_2 selects v_3, then the payoff for t_1 is 8 and payoff for t_2 is 4.

The procedure to find the optimum strategies for two players in the non-cooperative and cooperative game are described as follows.

1) Non-cooperative game: Let us consider this payoff matrix shown in Fig. 2, is for payoff as completion time. The payoff matrix is created by calculating the completion time of t_1 and t_2 with every virtual machine. The payoff matrix is converted into a normalized payoff matrix M by subtracting the payoff of t_2 from the payoff of t_1 as shown in Fig. 3. If the value in the normalized payoff matrix is positive, it is a loss to t_1 and gains to t_2. If the value in the normalized payoff matrix is negative, it is gain to t_1 and loss to t_2. As we already stated earlier, in our game model payoff could be either completion time or waiting time. Hence, the minimum is the payoff value maximizes the gain. Task t_1 follows the minimax principle, i.e., it will try to minimize the maximum loss. So, first, we will find the maximum element of each row and then select the minimum one as vm_1. The mathematical expression to find the minimum among the maximum value of each strategy of t_1 is

$$vm_1 = \min_{\forall_i} \left(\max_{\forall_j} M[i][j] \right) \tag{4}$$

In Fig. 4, Row-Max column contains the maximum values of each rows and the minimum one is 1 of v_1, so vm_1 is 1.

Task t_2 follows the maximin principle i.e. it will try to maximize the minimum gain. So, first, we will find the minimum element of each column and then select the maximum one

t2 t1	V_1	V_2	V_3	Row-Max
V_1	1	-1	***1***	**1**
V_2	3	-5	4	4
V_3	-4	2	-2	2
Col-Min	-4	-5	**-2**	

Fig. 4. VM Selection

as vm_2. The mathematical expression to find the maximum among the minimum value of each strategy of t_2 is

$$vm_2 = \max_{\forall_j}\left(\min_{\forall_i} M[i][j]\right) \tag{5}$$

In Fig. 4, the Col-Min row contains the minimum values of each column and the maximum one is -2 of v_3, so vm_2 is -2. Then, we find the intersection point of vm_1 and vm_2 i.e. in Fig. 4, 1 in the italic bold is the intersection point. So, the corresponding virtual machines, v_1 for t_1 and v_3 for t_2 will be selected. Same can be applied for payoff as waiting time w_{ij}.

2) Cooperative game: Let us consider the payoff matrix shown in Fig. 5 is for payoff as completion time. In the cooperative game, first, the payoff matrix is converted into a normalized payoff matrix by subtracting the payoff of t_2 from the payoff of t_1 and taking the mode of that value.

The above payoff matrix in Fig. 5, can be converted into a normalized payoff matrix as shown in Fig. 6.

In the cooperative game, two players are cooperating with each other in the sense that, they will not only think about their benefit but also the other player's benefit. Large value in the normalized payoff matrix indicates a large difference between the gain of two players and small value in the payoff matrix indicates a small difference between the gain of two players. So, we are selecting the strategies(virtual machines) corresponding to the smallest value in the normalized payoff matrix except the diagonal elements. In Fig. 6, 1 in the italic bold is the smallest value. So, the corresponding virtual machines, v_3 for t_1 and v_1 for t_2 will be selected. The same can be applied for payoff as waiting time w_{ij}.

V. Results

To demonstrate and prove that the proposed approach has an optimization result, we have performed our experiment by taking different numbers of tasks to range between 300 to 800, which is generated using Poisson distribution. Here, the results of taking 500 tasks for both non-cooperative and cooperative game model is discussed as the trend is all most similar for any number of tasks. The detailed simulation setting and parameters are as follows;

- The computing capacity of each virtual machine is set in the range between [3000-6000] MIPS.
- Tasks are generated by exponentially distributing inter-arrival time with the Poisson distribution.
- The deadline of the task is set as: $dl_i = a_i + baseD$, where $baseD$ is in uniform distribution $U(5, 10)$.
- The size of task is set in the range between 6000 to 10000 MI.

t2 t1	V_1	V_2	V_3
V_1	(5, 3)	(1, 4)	(6, 2)
V_2	(5, 1)	(2, 5)	(1, 4)
V_3	(5, 6)	(2, 4)	(4, 7)

Fig. 5. Payoff Matrix

t2 t1	V_1	V_2	V_3
V_1	2	3	4
V_2	4	3	3
V_3	***1***	2	3

Fig. 6. Normalized Payoff Matrix

The simulation outcomes show that the cooperative game is performing better because in non-cooperative game tasks are selfish i.e. they only think about their benefits. One small task may wait for a bigger task for a long time leading to higher overall waiting time as well as completion time. But, in the case of the cooperative game both the tasks will try to minimize their total completion time and waiting time.

A. Completion Time as Payoff

Here, we show a group of experimental results to perceive the performance of the cooperative and non-cooperative game taking completion time as the payoff. The number of virtual machines count is set to 10, 20, 30, 40 and 50. It can be realized from Fig. 7 and Fig. 8 that, the total completion time and total waiting time is less in a cooperative game model than a non-cooperative game model.

B. Waiting Time as Payoff

Here, we show a group of experimental results to perceive the performance of the cooperative and non-cooperative game taking waiting time as the payoff. The number of virtual machines count is set to 10, 20, 30, 40 and 50. It can be realized from Fig. 9 and Fig. 10 that, the total completion time and total waiting time is less in a co-operative game model than a non-cooperative game model.

VI. Conclusion And Future Work

As task scheduling is the core issue of cloud computing, it needs to handle properly. In this paper, our main focus was on the following points. In this paper, we propose a

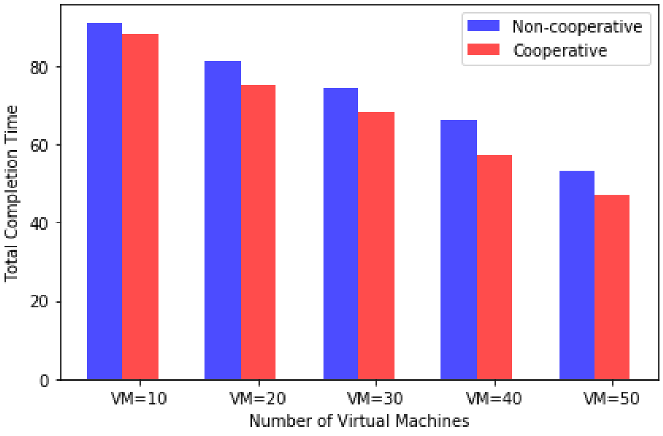

Fig. 7. Total Completion Time

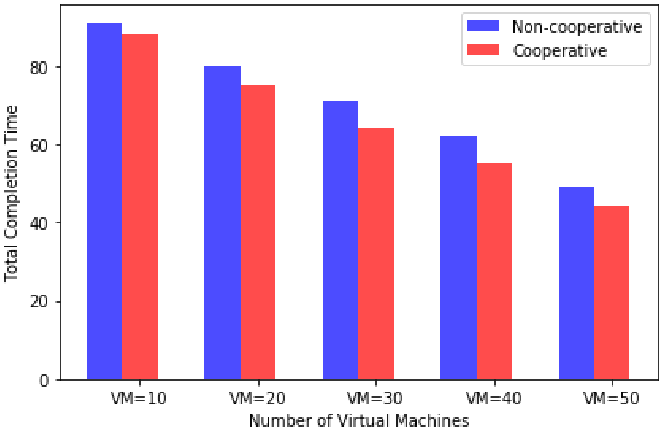

Fig. 9. Total Completion Time

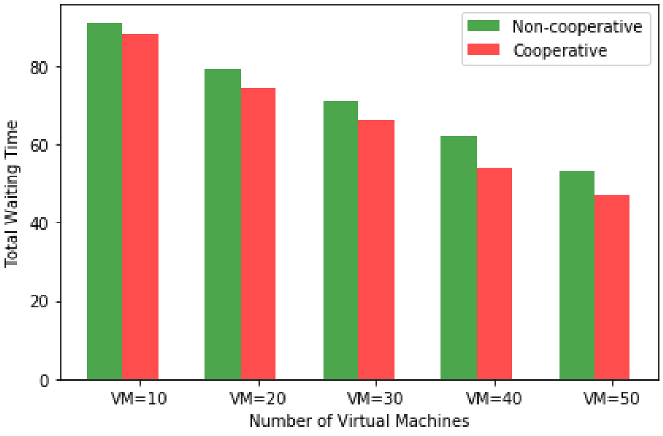

Fig. 8. Total Waiting Time

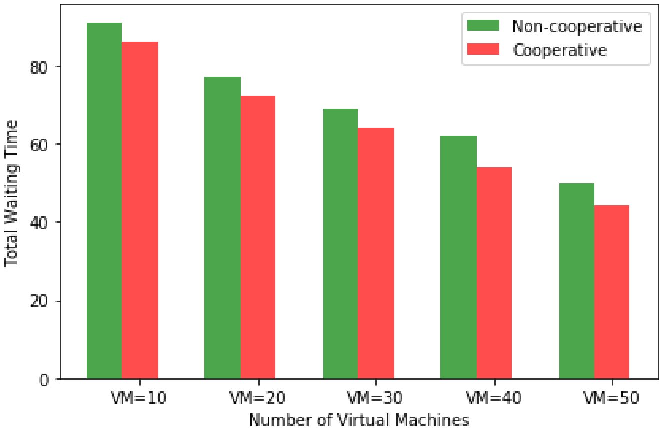

Fig. 10. Total Waiting Time

game-theoretic approach for real-time task scheduling in a cloud computing environment with a non-co-operative and co-operative game. We discuss how game theory can be applied in task scheduling and a detailed analysis has been done for scheduling real-time tasks cloud computing environment. We have considered all user tasks as the players of the game and the virtual machines are regarded as game strategies. We have compared the experimental results of the non-cooperative game and cooperative game. Taking payoff as completion time and waiting time, the experimental results show that the cooperative game model for task scheduling performs better than a non-cooperative game model. Further, total completion time and total waiting time in a cooperative game model is less than the non-cooperative game model in both the scenario (payoff as completion time and payoff as waiting time). The game theoretic approach can also be used for load balancing in IoT based application model as described in [16].

In the future, we will consider several other evaluations matric like reliability, energy consumption, etc. to enhance the proposed game-theoretic approach.

REFERENCES

[1] Arunarani, A. R., D. Manjula, and Vijayan Sugumaran. ”Task scheduling techniques in cloud computing: A literature survey.” Future Generation Computer Systems 91 (2019): 407-415.

[2] Yang, Jiachen, et al. ”A task scheduling algorithm considering game theory designed for energy management in cloud computing.” Future Generation Computer Systems (2017).

[3] Zhang, Daniel, et al. ”A real-time and non-cooperative task allocation framework for social sensing applications in edge computing systems.” 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE, 2018.

[4] Zhang, Yingfeng, et al. ”Game Theory Based Real-Time Shop Floor Scheduling Strategy and Method for Cloud Manufacturing.” International Journal of Intelligent Systems 32.4 (2017): 437-463.

[5] Li, Kai, Yong Wang, and Meilin Liu. ”A non-cooperative game model for reliability-based task scheduling in cloud computing.” arXiv preprint arXiv:1403.5012 (2014).

[6] Roungas, Bill, Femke Bekius, and Sebastiaan Meijer. ”The Game Between Game Theory and Gaming Simulations: Design Choices.” Simulation & Gaming 50.2 (2019): 180-201.

[7] Samadi, Yassir, Mostapha Zbakh, and Claude Tadonki. ”E-HEFT: enhancement heterogeneous earliest finish time algorithm for task scheduling based on load balancing in cloud computing.” 2018 International Conference on High Performance Computing & Simulation (HPCS). IEEE, 2018.

[8] Akbar, Saeed, et al. ”A Game-based Thermal-aware Resource Allocation Strategy for Data Centers.” IEEE Transactions on Cloud Computing (2019).

[9] Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. ”Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds.” Future Generation Computer Systems 29.1 (2013): 158-169.

[10] Ergu, Daji, et al. ”The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment.” The Journal of Supercomputing 64.3 (2013): 835-848.

[11] Sahoo, Sampa, Bibhudatta Sahoo, and Ashok Kumar Turuk. ”A Learning Automata-based Scheduling for Deadline Sensitive Task in The Cloud.” IEEE Transactions on Services Computing (2019).

[12] Sedighi, Art, Yuefan Deng, and Peng Zhang. ”Fariness of task scheduling in high performance computing environments.” Scalable Computing: Practice and Experience 15.3 (2014): 271-288.

[13] Su, Zhou, et al. ”Game theoretic resource allocation in media cloud with mobile social users.” IEEE Transactions on Multimedia 18.8 (2016): 1650-1660.

[14] Gao, Yan, Bin Zhang, Mengzhu Lu, and Anxiang Ma. ”Game Based SBCA Cloud Resource Adjustment Method.” In 2019 21st International Conference on Advanced Communication Technology (ICACT), pp. 47-51. IEEE, 2019.

[15] Homsi, Soamar, Gang Quan, Wujie Wen, Gustavo A. Chapparo-Baquero, and Laurent Njilla. ”Game Theoretic-Based Approaches for Cybersecurity-Aware Virtual Machine Placement in Public Cloud Clusters.” In 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 272-281. 2019.

[16] Patra, Manoj Kumar. ”An architecture model for smart city using Cognitive Internet of Things (CIoT).” 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2017.