# Load Balancing in Green Cloud Computation

Nada M. Al Sallami

***Abstract*-This paper discusses a proposed load balance technique based on Artificial Neural Network ANN. ANN predict the demand and then allocate resources according to demand. Thus, it always maintains the active servers according to current demand, which results in low energy consumption than the conservative approach of over-provisioning. Furthermore, high utilization of server results in more power consumption, server running at higher utilization can process more workload with similar power usage. Finally the existing load balancing techniques in cloud computing are discussed and compared based on various parameters.**

**Keywords: Green Cloud Computing, Load Balancing, Artificial Neural Networks**

## I. INTRODUCTION

Cloud computing can help business shift their focus to developing good business applications that will bring true business value [1]. Cloud computing can mainly provide four different service like: virtual server storage (Infrastructure as a service or IaaS) such as Amazon Web Services, software solution provider over the internet (Software as a Service or SaaS), software and product development tools (Platform as a Service or PaaS) such as Google Apps and Communication as a service or CaaS) [2][3]]. Clouds are deployed on physical infrastructure where Cloud middleware is implemented for delivering service to customers. Such an infrastructure and middleware differ in their services, administrative domain and access to users. Therefore, the Cloud deployments are classified mainly into three types: Public Cloud, Private Cloud and Hybrid Cloud. Due to the exponential growth of cloud computing, it has been widely adopted by the industry and there is a rapid expansion in data-centers. This expansion has caused the dramatic increase in energy use and its impact on the environment in terms of carbon footprints. The link between energy consumption and carbon emission has given rise to an energy management issue which is to improve energy-efficiency in cloud computing to achieve Green computing [4].This paper proposed a new algorithm to achieve Green computing in load balancing. This algorithm use artificial neural network to solve load balancing in cloud, its performance is discussed and compared with the existing load balancing techniques. Although cloud computing can be seen as a beneficial tool for businesses for several reasons [4][5], however, like all new technology it comes with both benefits and draw backs.

Al Sallami Nada Author is with Al Zaytoonah university of Jordan, Science and Information Technology college, e-mail: nada.alsalami@yahoo.com.

## II. GREEN COMPUTING

As High Performance Computing (HPC) is becoming popular in commercial and consumer IT applications, it needs the ability to gain rapid and scalable access to high end computing capabilities. This computing infrastructure is provided by cloud computing by making use of datacenters. It helps the HPC users in an on-demand and payable access to their applications and data, anywhere from a cloud [6]. Cloud computing data-centers have been enabled by high-speed computer networks that allow applications to run more efficiently on these remote, broadband computer networks, compared to local personal computers. These data-centers cost less for application hosting and operation than individual application software licenses running on clusters of on-site computer clusters [11]. However, the explosion of cloud computing networks and the growing demand drastically increases the energy consumption of data-centers, which has become a critical issue and a major concern for both industry and society [8]. This increase in energy consumption not only increases energy cost but also increases carbon-emission [7]. According to reference [9], there are following four key factors that have enabled the Cloud computing to lower energy usage and carbon emissions from ICT.

A. **Dynamic Provisioning:** In traditional setting there are two reasons for over-provisioning: first, it is very difficult to predict the demand at a time and second, to guarantee availability of services and to maintain certain level of service quality to end users. Cloud providers monitor and predict the demand and thus allocate resources according to demand. Those applications that require less number of resources can be consolidated on the same server. Thus, datacenters always maintain the active servers according to current demand, which results in low energy consumption than the conservative approach of over-provisioning.
B. **Multi-tenancy:** The smaller fluctuation in demand results in better prediction and results in greater energy savings. Using multi-tenancy approach, Cloud computing infrastructure reduces overall energy usage and associated carbon emissions.
C. **Server Utilization:** High utilization of server results in more power consumption, server running at higher utilization can process more workload with similar power usage.
D. **Datacenter Efficiency:** The Cloud datacenters are quite different from traditional hosting facilities. A cloud datacenter could comprise of many hundreds or thousands of networked computers with their corresponding storage and networking subsystems, power distribution and conditioning equipment, and cooling infrastructures. A data center hosts computational power, storage and applications required to support an enterprise business. A data center is central to modern IT infrastructure, as all enterprise content is sourced from or passes through

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

it. There are two major and complementary methods [8] to build a green data center: first, utilize green elements in the design and building process of a data center. Second, Greenify the process of running and operating a data center in everyday usage.

Load balancing can be one such energy-saving solution in cloud computing environment. Thus load balancing is required to achieve Green computing in clouds which can be done with the help of the following factors [6]:

A. **Throughput**
B. **Overhead Associated**
C. **Fault Tolerance**
D. **Migration time**
E. **Response Time**
F. **Resource Utilization**
G. **Scalability**
H. **Performance**
I. **Energy Consumption**
J. **Carbon Emission**

## III. THE PROPOSED METHIOD

The goal of load balancing is to minimize the resource consumption which will further reduce energy consumption and carbon emission rate that is the dire need of cloud computing. This determines the need of new metrics, energy consumption and carbon emission for energy-efficient load balancing in cloud computing as described in the previous sections.

Artificial neural networks are used in the proposed algorithm because of its simplicity and efficiency to satisfy many metrics stated in the previous section, like throughput, fault tolerance, response time and resource utilization. Also it can work efficiently with noise and incomplete information. Back Propagation learning algorithm was used to train ANN such that it can distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed, while others are idle or doing little work. It helps in optimal utilization of resources and hence in enhancing the performance of the system.

The proposed ANN composed of three layers; the first layer is the input layer which represents the current workload for N nodes. The second layer is the hidden layer, while the third layer is the output layer which represents the balanced workload for N nodes. Each node in the input layer represents either the current server's workload or the current average workload of a cluster of servers and an integer number was assigned to it, as shown in figure 2. While the corresponding node in the output layer represents either server's workload or cluster's average workload after balancing respectively. Therefore the number of neurons in the input and output layers are equal. Load balancing process is done by training neural network on many different and representative examples of balanced and unbalanced cases. Obviously, a substantial reduction in energy consumption can be made by powering down servers when they are not in use. This case satisfies when the output is (0). Furthermore, negative values (-1) is associated to each input and output node when its workload is unknown as shown in figure 4, since incomplete data examples are also considered. Figure 5 shows the architecture of one network.

## IV. RESULT

To train ANN, the actual loads are applied at the input layer then the outputs are calculated at the output layer and compared with the desired (balanced) loads, errors are computed and weights are adjusted. The above process is repeated with large set of example until ANN is trained with accepted error rate. Once the network was trained within a tolerable error, the network is tested with different data set. Otherwise, ANN must be retrained with more examples and/ or change training parameters. Thus training is stopped when ANN is learned. Obviously if good examples are used then good learning is forced. The number of hidden layers and the number of neurons in each hidden layer are changed during the training phase so that good performance is derived, as shown in figures 4.

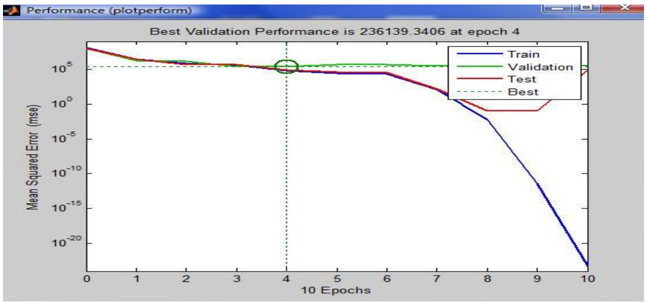


Figure 4: ANN performance in training, test and validation stages

The existing load balancing techniques have been compared in Table 1, this comparison is similar to that given in references [4][6] except that two additional matrices are added. Existing load balancing techniques that have been discussed worked in distributed, cloud, and large scale cloud system environment and mainly focus on reducing associated overhead, service response time and improving performance etc. but none of them have considered the energy consumption and carbon emission factors (i.e. matrices I and J). Therefore, there is a need to develop an energy-efficient load balancing technique that can improve the performance of cloud computing by balancing the workload across all the nodes in the cloud along with maximum resource utilization, in turn reducing energy consumption and carbon emission to an extent which will help to achieve Green computing. The proposed method utilizes green elements in the design and building process of a load balancing (i.e. ANN). ANN predict the demand and thus allocate resources according to demand. Thus, it always maintains the active servers according to current demand, which results in low energy consumption than the conservative approach of over-provisioning. High utilization of server results in more power consumption, server running at higher utilization can process more workload with similar power usage. Table 1 summarizes the existing techniques and the proposed technique of load balance in cloud.

## V. CONCLUSION

Load balancing in clouds is a mechanism that distributes the dynamic local workload evenly across all the nodes. The existing load balancing techniques in clouds, consider various parameters like performance, response time, scalability, throughput, resource utilization, fault tolerance, migration time and associated overhead. But, for an energy-efficient load balancing metrics like energy consumption

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and carbon emission should also be considered which will help to achieve Green computing.

ANN predict the demand and thus allocate resources according to demand. Thus, it always maintains the active servers according to current demand, which results in low energy consumption than the conservative approach of over-provisioning. Furthermore, high utilization of server results in more power consumption, server running at higher utilization can process more workload with similar power usage.

## REFERENCES


[1] Abdulaziz Aljabre, " Cloud Computing for Increased Business Value", *International Journal of Business and Social Science, Vol. 3 No. 1; January 2012*

[2] K. Dinesh, G. Poornima, K.Kiruthika, " Efficient Resources Allocation for Different Jobs in Cloud", *International Journal of Computer Applications (0975 – 8887) Volume 56– No.10, October 2012*

[3] Gary Garrison, Sanghyun Kim, and Robin L. Wakefield, " Success Factors for Deploying Cloud Computing", communications of the ACM | September 2012 | vol. 55 | no. 9, doi:10.1145/2330667.2330685.

[4] Nidhi Jain Kansal and Inderveer Chana, " Existing Load Balancing Techniques in Cloud Computing: A Systematic Re-View", Journal of Information Systems and Communication ISSN: 0976-8742, E-ISSN: 0976-8750, Volume 3, Issue 1, 2012, pp- 87-91. Available online at http://www.bioinfo.in/contents.php?id=45

[5] Bhatt, G. and Grover, "Types of information technology capabilities and their role in competitive advantage: An empirical study", Journal of Management Information Systems 22, 2 (2005), 253–277.

[6] Nidhi Jain Kansal1, Inderveer Chana2, "Cloud Load Balancing Techniques : A Step Towards Green Computing ",IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012 , ISSN (Online): 1694-0814, www.IJCSI.org.

[7] Al-Dahoud Ali and Mohamed A. Belal, (2007) "Multiple Ant Colonies Optimization for Load Balancing in DistributedSystems", *ICTA'07*, Hammamet, Tunisia.

[8] M. Mani B. Srivastava, IEEE, Anantha P. Chandrakasan, and F. and Robert W. Brodersen, IEEE, "Predictive System Shutdown and Other Architectural Techniques for Energy Efficient Programmable Computation," IEEE Transactions on VLSI ystems, vol. 4, p. 15, March 1996.

[9] Truong Vinh Truong Duy, Yukinori Sato and Yasushi Inoguchi,"Performance Evaluation of a Green Scheduling Algorithm for Energy Savings in Cloud Computing", IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 .

[10] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.

[11] A. M. Nakai, E. Madeira, and L. E. Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC), 2011, pages 156-165.

[12] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", An international Journal on Performance evaluation, In Press, Accepted Manuscript, Available online 3 August 2011.

[13] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and Jun-Yuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), 2011, pages 1-4.

[14] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, pages 89-96.

[15] A. Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), January 2010.

[16] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, "LBVS: A Load Balancing Strategy for Virtual Storage", International Conference on Service Sciences (ICSS), IEEE, 2010, pages 257-262.

[17] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[18] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, April 2010, pages 551-556.

[19] Z. Zhang, and X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240- 243.

[20] Al-Dahoud Ali, M.A. Belal and M. Belal Al-Zoubi, "**Load Balancing of Distributed Systems Based on Multiple Ant Colonies Optimization** ", American Journal of Applied Sciences 7 (3): 428-433, 2010 , ISSN 1546-9239

[21] S. Wang, K. Yan, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, September 2010, pages 108-113

[22] V. Nae, R. Prodan, and T. Fahringer, "Cost-Efficient Hosting and Load Balancing of Massively Multiplayer Online Games", Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (Grid), IEEE Computer Society, October 2010, pages 9-17.

[23] R. Stanojevic, and R. Shorten, "Load balancing vs. distributed rate limiting: a unifying framework for cloud control", Proceedings of IEEE ICC, Dresden, Germany, August 2009, pages 1-6,

[24] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea, August 2009, pages 170-175.

[25] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.

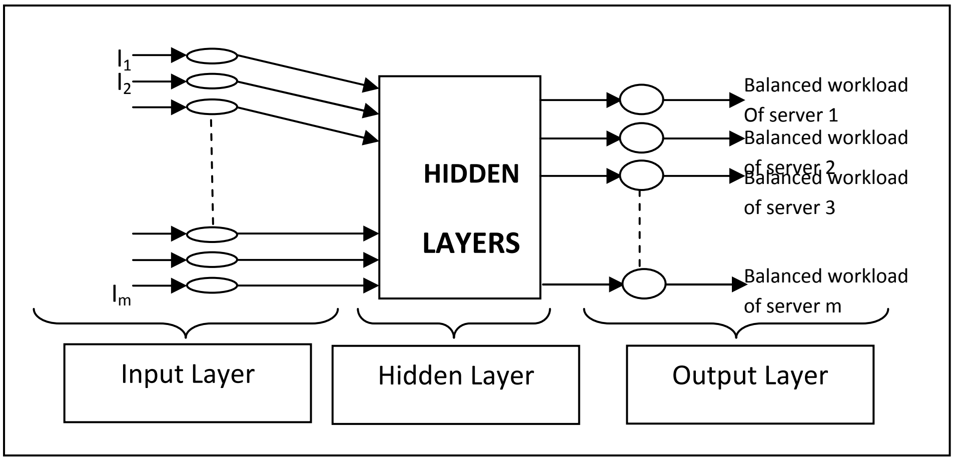


**Figure 1: ANN Architecture**

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | **0** | **0** | **5** | **0** | **0** | **0** | **0** | **0** | **0** | **5** | **0** | **0** | **0** | **0** | **6** | **0** | **0** | **0** | **4** | **0** | **0** | **0** | **3** | **0** | **0** | **0** | **0** | **0** | **8** |

**(a) Input-Vector**

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |

**(b) Output-Vector (equal value in each cell)**

**Figure 2: complete data Vectors (without turn off servers)**

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |

(a) Input-Vector

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

(b) Output-Vector (0 for unused servers)

**Figure 3: complete data Vectors (with turn off servers)**

Table 1:A comparison between the existing techniques and the proposed technique

| ***No.*** | ***Techniques*** | ***Description*** | ***Findings*** | **Satisfied Metrics** |
|---|---|---|---|---|
| 1 | Decentralized content aware [10] | 1. Uses a unique and special property(USP) of requests and computing nodes to help scheduler to decide the best node for processing the requests<br>2. Uses the content information to narrow down the search | 1. Improves the searching performance hence increasing overall performance<br>2. Reduces idle time of the nodes | F |
| 2 | LB for Internet distributed services [11] | 1. Uses a protocol to limit redirection rates to avoid remote servers overloading<br>2. Uses a middleware to support this protocol<br>3. Uses a heuristic to tolerate abrupt load changes | 1. Reduces service response times by redirecting requests to the closest servers without overloading them<br>2. Mean response time is 29% smaller than RR(Round Robin) and 31% smaller than SL(Smallest Latency) | B, F, & H |
| 3 | Join-Idle-Queue [12] | 1. First assigns idle processors to dispatchers for the availability of the idle processors at each dispatcher<br>2. Then assigns jobs to processors to reduce average queue length of jobs at each processor | 1. Effectively reduces the system load<br>2. Incurs no communication overhead at job arrivals<br>3. Does not increase actual response times | B, D, &F |
| 4 | Lock-free multiprocessing[13] | 1. Runs multiple load-balancing processes in one load balancer | 1. Improves overall performance of load balancer | F & G |
| 5 | Scheduling strategy | 1. Uses Genetic algorithm, historical data | 1. Solves the problems of load | B & F |

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

| | | | | |
|---|---|---|---|---|
| | on LB of VM resources [14] | and current state of system to achieve best load balancing and to reduce dynamic migration | imbalance and high migration cost | |
| 6 | Central LB policy for VMs [15] | 1. Uses global state information to make load balancing decisions | 1. Balances the load evenly to improve overall performance<br>2. Up to 20% improvement in performance<br>3. Does not consider fault tolerance | A, E, F & H |
| 7 | LBVS: LB strategy for Virtual Storage [16] | 1. Uses Fair-Share Replication strategy to achieve Replica Load balancing module which in turn controls the access load balancing<br>2. Uses writing balancing algorithm to control data writing load balancing | 1. Enhances flexibility and robustness<br>2. Provides large scale net data storage and storage as a service | C, E, G & H |
| 8 | Task Scheduling Based on LB [17] | 1. First maps tasks to virtual machines and then virtual machines to host resources | 1. Improves task response time<br>2. Improves resource utilization | E, F & H |
| 9 | Honeybee Foraging Behavior [18] | 1. Achieves global load balancing through local serve actions | 1. Performs well as system diversity increases<br>2. Does not increase throughput as system size increases | A, G &H |
| 10 | Biased Random Sampling [18] | 1. Achieves load balancing across all system nodes using random sampling of the system domain | 1. Performs better with high and similar population of resources<br>2. Degrades as population diversity increases | A, G & H |
| 11 | Active Clustering [18] | 1. Optimizes job assignment by connecting similar services by local re-wiring | 1. Performs better with high resources<br>2. Utilizes the increased system resources to increase throughput<br>3. Degrades as system diversity increases | A, G & H |
| 12 | ACCLB (Ant Colony and Complex Network Theory) [19][20] | 1. Uses small-world and scale-free characteristics of complex network to achieve better load balancing | 1. Overcomes heterogeneity<br>2. Adaptive to dynamic environments<br>3. Excellent in fault tolerance<br>4. Good scalability | C, F, G & H |
| 13 | Two-phase scheduling(OLB + LBMM) [21] | 1. Uses OLB (Opportunistic Load Balancing) to keep each node busy and uses LBMM(Load Balance Min-Min) to achieve the minimum execution time of each task | 1. Efficient utilization of resources<br>2. Enhances work efficiency | F & H |
| 14 | Event-driven [22] | 1. Uses complete capacity event as input, analyzes its components and generates the game session load balancing actions | 1. Capable of scaling up and down a game session on multiple resources according to the variable user load<br>2. Occasional QoS breaches as low as 0.66% | B, E, F, G & H |
| 15 | Carton(LB + DRL) [23] | 1. Uses Load Balancing to minimize the associated cost and uses Distributed Rate Limiting for fair allocation of resources | 1. Simple<br>2. Easy to implement<br>3.Very low computation and communication overhead | E, G & H |
| 16 | Compare and Balance [24] | 1. Based on sampling<br>2. Uses adaptive live migration of virtual machines | 1. Balances load amongst servers<br>2. Reaches equilibrium fast<br>3. Assures migration of VMs from high-cost physical hosts to low-cost host<br>4. Assumption of having enough memory with each physical host | B, E & H |
| 17 | VectorDot [25] | 1. Uses dot product to distinguish node based on the item requirement | 1. Handles hierarchical and multidimensional resource constraints<br>2. Removes overloads on server, switch and storage | A & H |
| 18 | The proposed ANN techniq | 1. Use ANN to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. Bp supervised neural networks are used | 1. Optimal Balances load amongst servers.<br>.2. Adaptive to dynamic environments<br>3. Excellent in fault tolerance<br>4. Good scalability<br>5. unused server are turned off | C, F, G, H, I & J |

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)