International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Cutting-Edge Data Mining Approach for Dynamic Data Replication That also Involves the Preventative Deletion of Data Centres That are Not Compatible with One Other

Bassam Talib Sabri

Submitted: 14/09/2022 **Accepted:** 26/12/2022

Abstract

At this time, large cloud-based applications have produced an increase in the number of demands for data center storage. Replicating data offers an effective method for managing data files in an expansive Cloud environment, which ultimately results in increased data dependability and availability. In this research work, we made a proposal for a data replication approach that we referred to as the Hybrid Repetition Stratagem. This strategy implemented the facsimile assignment, assortment, then auxiliary processes. The Hybrid Repetition Stratagem system is designed to replicate data files in the cloud and includes three primary stages. it chooses optimal location (the location that is the greatest dominant and has the highest number of accesses) for storing the new copy in order to shorten the amount of time required to retrieve it. In the second phase, HRSs takes into consideration a number of different parameters in order to determine which replica node will provide the best experience for users. These parameters include micro chip procedure competence, net broadcast ability, Input and output competence of CDs, weight, then net dormancy. In the tertiary step, the choice to replace something is taken so that the system may have a faster reaction time. A uncertain implication scheme by 3 inputs constraints allows HRS to determine the significance of valued replicas based on their characteristics (amount of admissions, price, then the previous period the model was accessed time). Using the Cloud Sim toolkit package, the newly designed replication policy is modeled and tested. The data are replicated among the cloud nodes in an acceptable manner via the technique that we have developed, which is very simple to put into action in an actual setting. The results of the experiments demonstrate that HRSs may considerably improve the availability, performance, and load balancing of applications that need a large amount of data. In addition, there is no need to increase any extra overhead costs since it is still viable.

Keywords: Raincloud compute, Replicating, Cloud-Sim, Fuzzing systems

1. Introduction

Because of the exceptional qualities of cloud storage, cloud computing has become more desirable for use in scientific research as well as in commercial settings [1-3]. In the past, investigations into the context of grid computing focused on similarly capable solutions with less resources. Due to the high degree of resemblance between these ideas, they were employed in place of one another. Foster et al. provided a survey presentation in which they compared the features of clouds and grids to one another [4]. The comparison between cloud computing and grid computing is condensed into Table 1. As a result, cloud computing is a cutting-edge technology that is built around the concept of an on-demand service. The Software as a Service model may be shown in Figure 1. (SaaS),Cloud computing primarily consisted of Stage as a Facility (PaaSs) and Organization as a Amenity (IaaSs). IaaSs, i.e, responsible for presentation of virtualization resources, e.g., communication on demand, storage consumption, situated in lower layer of cloud computing.

Department of Business Information Technology /University of Information Technology and Communications
Baghdad, Iraq
bassam.ali@uoitc.edu.iq

Table 1

Grid VS. Cloud

Grid	Cloud
Collaborative sharing of resources	Using of facility (eradicates the feature)
Low	High
Low (grid certificate service)	Highs (virtualization)
Low	High
Restrictions due to hardware	Unlimited
Normal	High
Yes	Yes
Low level command	High level services (SaaS0
Not a commodity	Vital

At higher level of IaaS, situated PaaS layer, i.e., as software delivery model [5], to enhance the programmability of cloud. The impotency of cloud is dramatically increased and as a consequence employment of software, hardware and services in infrastructure to enhance the performance are inevitable. It was necessary to note that, quantification of performance in cloud is challenging for researchers. Cloud computing requires enormous computation and storage capacity, which may be supplied by data centers equipped with high-performance distributed techniques and technologies. The requirement of large-scale applications that are hosted in the cloud, on the one hand, and the necessity of gigantic data centers, on the other, have both enhanced the vital role of reducing the cost of storage. The use of data replication as a strategy for ensuring the dependability of data is an effective method in modern distributed storage systems. Amazon S3, Google File System [6], and Hadoop Distributed File System [7] are typically the major data storage that employed replication algorithm with the capability of creation of 3 copies from data at once to improve data reliability. [6] Hadoop Distributed File System [7] is another major data storage.

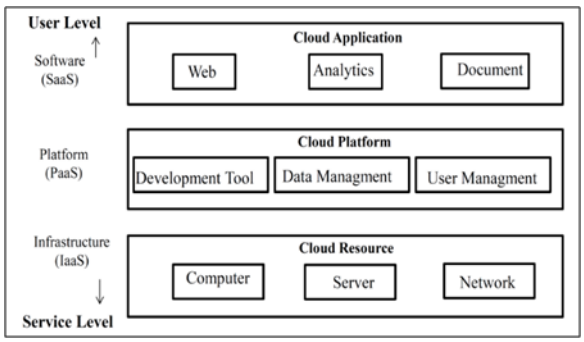

Fig. 1. Schematically representation of cloud computing services [5].

Data replication strategies are able to prepare large-scale parallel read and query data by distributing various data replicas within various nodes of the cloud. As a result, these strategies can reduce the amount of time a user must wait for data, increase the amount of data that is available, and reduce the amount of bandwidth that is consumed. A replication structure for cloud environments with a mix of different data centers is shown in Figure 2. By various configurations, cloud provides that information middles are dissimilar in footings of replication price, free storing for storing replicas, repetition algorithms, presentation, amount of multitudes, number of CPUs in each host, and bandwidth usage etc. In order to achieve the advantageous of replication for client, connection of one or more data center are inevitable. Replica placement, replica selection, and replica replacement are the three phases of data replication. By analyzing the net procedure and operator demand, facsimile placement pinpoints the optimal destination for a duplicate of a data file. The replica selection process determines which replica hosts an individual's data file during the execution of a task. in the cloud environment. The methodology that employed to determine the provider data center is named replica selection strategy and has a significant impact not only on the contentment of the consumer but also the cloud service provider. Because users are able to obtain ideal experiences, such as minimal latency, the least amount of packet loss, or high available bandwidth, by selecting the appropriate replicas, users may achieve optimal experiences. Choices from among appropriate replicas enables the user to minimize the latency, least packet or maximize available bandwidth. Unique features of replica selection, i.e., load balancing between several replicas and lowering of cost, outstands its key role in cloud environment. In real cloud storage environment, there are various network bandwidths, CPU speed and disks, maximizing the session's number of a data node that is able to answer to the requests. Each data node has its own workload intensity and capacity, consequently has its unique blocking probability [8].

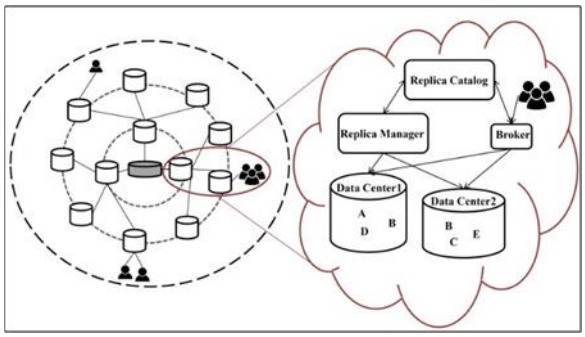

Fig.2. Structure for replicating data across many, distinct data centers in the cloud.

As the dispersed world evolves, the HRSs approach may shift to accommodate it. Furthermore, we evaluate the performance of many cloud-based simulation tools and ultimately choose on Cloud toolkit. The simulation results showed that HRSs is effective for a heterogeneous cloud environment from a variety of angles, including storage utilization, hit ratio, number of

communications, and balancing non preemptive independent processes[9].

2. Associated Research

All distributed systems rely heavily on replication. To the best of our knowledge[10], only a small number of studies have tried to provide dynamic replication methods for cloud-based systems[11], whereas many more exist for grid-based settings. Static replication [12-13] and dynamic replication [14-18] are two of the most common types of replication utilized today. The two approaches have inspired a wide variety of replication procedures and schemes, each with its own claimed performance advantages. Data files in a static replication strategy have their number of copies predetermined at the start of the strategy. The location of copies is similarly predetermined in this sort of technique, and these replicas are managed by hand. It doesn't modify itself in response to the dynamic nature of the system and the changing habits of its users. In contrast, dynamic replication systems produce, store, and update copies of data files on the fly to account for changing needs and user behaviors. Since users' access habits in cloud computing environments may change over time, the replication algorithms must be able to dynamically adjust to these shifts in order to maintain high availability and improve speed. When it comes to the cloud, dynamic replication techniques are often preferred over static ones. Data replication strategies in cloud computing systems are few in the published literature. Recent studies on these subjects are discussed below. The replication technique [19] takes into account the cloud system's inherent hierarchical nature. Using the idea that data that has been accessed more recently would be more desirable in the near future, they employed temporal locality [20, 21] as a criterion for selecting the most popular data file. In order to figure out how many copies of a system are required, researchers looked at both system uptime and the likelihood of system breakdown. When there is a great demand in one area, a copy is set up close by. In a separate paper [22], the authors combine this tactic with the checkpoint approach to propose their DAFT (Dynamic Adaptive Fault-tolerance) solution. However, DAFT's primary flaw is that it does not properly distribute workloads among available system resources. Using storage load and historical replica selection data [23] suggested a PGSA (Plant Growth Simulation Algorithm) based data replica selection strategy for the cloud. Their strategy took plant phototropism into account by using morphactin concentration as a cutoff for tailoring data copies to individual requests. A appropriate data copy is selected by regulation and comparison of its characteristic to morphactin concentration [24]. CloudSim's simulation was used to assess PGSA in terms of replica usage and mean access time. According to the findings, PGSA has acceptable response times. Regrettably, PGSA did not look at data center fault tolerance. EDR is a cost-based replica selection algorithm created by [25]. (Energy-Aware Distributed Running system). Two distributed optimization algorithms were used to create this decentralized system. As a first step, they simulated the money spent on energy by cloud data centers. The team then formalized the replica selection issue as a convex optimization problem, which optimized the bandwidth and latency of each data center to decrease the energy cost of each data center. Analysis revealed that data-intensive applications like online video streaming and distributed file sharing saved 12 percent on electricity costs thanks to EDR [26] provide two distinct techniques for the replication process, High Quality of Service First Replication (HQFR) and Minimize Cost Maximum Flow, that take into account QoS needs as a criteria (MCMF). The former, a greedy strategy, is carried out by copying the program from higher to lower priority while taking QoS needs into account. In contrast, the latter used MCMF to determine which option was superior. MATLAB is used to validate HQFR and MCMF, and the results are compared to those obtained using random Hadoop techniques. The key flaws of the suggested technique are the overemphasis on modeling costs the disregard for economic units. In a cloud setting, [27] developed a MORM (Multi-objective Optimized Replication Management) technique. This method relies on the artificial immune algorithm [28]. MORM considers a variety of factors, including file availability, service time, load fluctuation, energy use, and delay value, while establishing a correlation between replica number and performance indicators. The five aforementioned goals are then used to establish the optimal number of copies for each data file and the optimal placement of replicas among nodes. They used an enhanced version of the CloudSim simulator and the MATLAB toolbox to assess MORM. The testing findings showed that the proposed method improved large-scale cloud storage clusters' file availability, load balancing, service time, latency, and energy consumption. The primary drawbacks of the MORM technique are that it pays little attention to fault tolerance and replacement difficulties. [29] developed a technique, taking into account transmission cost, assessment information of history, system load, and users' QoS choice as criteria. Does the right file already exist on the local node? is an issue our approach addresses. If so, then it can be put to immediate use. In such case, it's important to double-check the evaluation data and replica features. QoS preference may make an educated guess about the amount of availability, timeline, and dependability. This is necessary for determining the degree to which modern demanding settings resemble

their historical counterparts. At last, the copy with the most trustworthy ecosystem resemblance is chosen. According to the simulation findings, the suggested technique may adaptively raise data availability because of that replication by modifying the dependability show how to implement a workload-aware data replication technique (SWORD) to cut down on cloud resources. SWORD uses partitioning methods to decrease the average number of computers involved in the processing of a request or transaction. Specifically, they optimized analytical and transactional workloads by modifying query span utilization. The authors also introduced a data-placing strategy by making analogies to various graph-theoretic ideas. The usage of fine-grained quorums was investigated with the aim of reducing query spans and increasing throughputs. Their design uses granular quorums to accommodate a wide variety of workloads, a need for cloud computing. The viability of the framework is validated by simulation using two distinct types of workloads as case studies. The key problem of this technique is that it can only replicate the average duration of queries inside the system. Here, we use a number of variables to provide a direct comparison of different approaches to replication.

3. Method for Keeping Copies of Data is Presented

3.1. Duplicate location

Finding the most suitable location for the installation of replicas is one of the significant challenges presented by the rapid expansion of cloud storage[30]. We store replicas in a manner that takes into account the centrality as well as the amount of replica accesses. According to the idea of time-based locality, situations that newly viewed files are more likely to be accessed again, the number of replicas that have been accessed plays a significant impact in the selection of where to store replicas [31]. In addition, the centrality of a node in a graph is a factor that may be used to assess the relative significance of a location within the system. In order to cut down on retrieval time, our technique takes the centrality into consideration. There are many distinct metrics that may be used to measure centrality, including proximity importance, grade importance, amid importance, and oddness importance [32]. In the process of duplicate location, we just take into account the proximity measure. If a site consumes the lowermost worth for the total the detachments that it is from all of the other locations in a network, then that site is considered to be the closest site in the network. The total of a site's distances to all of the other locations should be as low as possible for it to be considered centrally located. The following is an example of how the It is possible to determine the proximity centrality value for site v. [32]:

$$Centrality(v) = \frac{N-1}{\sum_{a \neq v} d(v,a)} \quad (1)$$

The number of nodes in the system is denoted by N, while the distance between nodes v and an is denoted by d(v,a). The greatest place to keep a duplicate seems to be in the sequel (finest site has the uppermost worth of Value). Equation () is used to determine merit value (2).

$$Merit = w_1 \times Numberof Access + w_2 \times Centrality \ (2)$$

Where W1.0 and W2.0 are the percentage masses that correspond to the deuce primary limitations that have been discussed so far. We are aware that the scales of the aforementioned factors are not the same. Because of this, it is essential to convert the values of the parameters onto a scale ranging from 1 to 10 before incorporating them into the equation (2). We are operating on the assumption that the normalization scale is constant. Let us suppose that the value of Inc. is equal to (maxing–mining)/11. Before, the price of the issues that fall amid the minimum and the minimum plus the increment should be normalized to 1, the value of the factors that fall between the minimum and the minimum plus the increment should be normalized to 2, and so on.

3.2. Picking a copy to keep

In large-scale cloud storage systems, the data nodes have a variety of characteristics, including different kinds of disks, network bandwidth, CPU speed, and so on. If there is a duplicate of the data accessible in more than one location, then selecting the most appropriate replica provider offers a significant number of advantages. As a result, we will discuss a imitation assortment technique that employs a united total occupation that takes into account the capacity of the VM, the load on the VM, and the performance of the network.

3.2.1. VM Storage Capacity (C)

In the suggested approach for replica selection, the letter C stands for the capabilities of the virtual machine. We are able to calculate C by taking into account two characteristics, which are the process capacity of the CPU and the I/O capability of the disks, which are indicated by and, respectively. In addition, the value of n represents the total number of processors. In order to facilitate the computation of the C value, we first normalize the performance parameter of the virtual machine to the range [0, 1]. In the continuation, the equation for finding C is as follows:

$$C = (n \times \alpha) + \beta \ (3)$$

3.2.2. It's a VM pile (L)

In most contexts, "load" refers to the entire amount of time that a virtual machine is tasked with doing "tasks" [33]. As a result, the load of a virtual machine can be determined by dividing the number of tasks that are now waiting in the service queue of the virtual machine by the service rate (Sr) for the virtual machine at the current moment. In the continuation, the load (L) may be determined using the formulas below:

$$L = \frac{Nt}{Sr} \quad (4)$$

3.2.3. Capacity and Throughput of a Network (N)

The following formula is used to calculate the the throughput of the network between a and b in terms of Mbits/s and milliseconds of delay

$$N(a,b) = \frac{Bandwidth(a,b)}{Network\_Latency(a,b)} \quad (5)$$

Finally, TotalCost function is given by:

$$TotalCost = W_4 \times \frac{1}{C} + W_4 \times L + W_5 \times \frac{1}{N} \quad (6)$$

In this case, the weights w3.0, w4.0, and w5.0 are suitable. simple to maximize the rate of weightiness; one theory that might be used is that the greater the value of the weight, the more general the factor[33]. One such line of reasoning is the predilection of the operator or the inspiration that is agreed to a certain component rather than the others [34]. The more recent concept has been used in the work that we have done. Finally, if the requested folder fixes not occur at the home-grown place, the HRSs method will build a list of potential candidates for replica providers, and then choose one based on the EQ scores.. that has the lowest value for TotalCost among those candidates (6).

3.3. Equivalent Substitute Replica

When there is insufficient storage space available in the location that would be ideal for storing new copies, it is necessary to remove either single or additional of current copies. The HRSs receipts into consideration trio characteristics, including the cost, the number of times the replica has been visited, and the most recent time it was accessed. The likelihood of accessing the file again is determined by the total number of times it has been accessed as well as the most recent time a duplicate was requested. It should go without saying that deleting a file that has a high cost value will not be a beneficial use of your time. This is due to the fact that if a position demands that folder in the forthcoming, we will need to recompence a large price for copying it over, which is not an economically viable option. When calculating the cost of replication, we use the following equation:

$$Cost = \frac{Size}{Bandwidth(x,y)} + PropagationDelayTime(x,y) \quad (7)$$

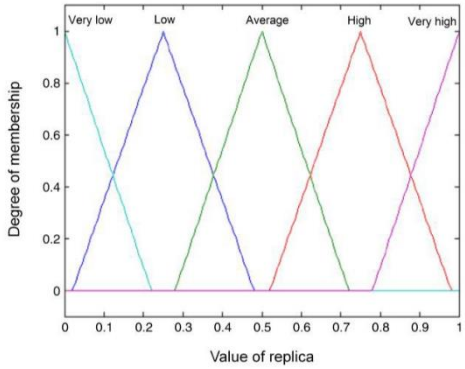

Fig. 3. Fuzzy system to determine the value of replica.

Where Size represents the extent of the facsimile, Band width (x, y) represents the band width amid the breadwinner location x and the supplicant location y, and Propagation Delay Time (x, y) represents the amount of historical needed to broadcast the needed replica from the breadwinner site x to the requester site y. We are aware that a location that is physically nearby may facilitate a faster transfer of replica. The fuzzy inference system is something that HRSs takes into consideration. This system has three input factors (the number of accesses, the cost, and the amount of time that has passed since the replica was last accessed), however it only has one productivity parameter (i.e, Cost). The ambiguous inference system is shown in Figure 3, and it is used to give a value to each copy. The ambiguous set theory has the ability to demonstrate actual-world facts in the presence of ambiguity. It does this by providing the fuzzy set with a membership grade. Within the fuzzy system, several degrees of membership, ranging from 0 to 1, are assigned. The components of the universe are mapped into a range from 0 to 1 using a membership function. Using a fuzzy function, HRS determines the value of each and every file that may be accessed from the optimal location. After that, it arranges the list such that Value increases from lowest to highest. It continues to choose candidate files from the list until there is sufficient space available. Matlab's Fuzzy Logic Toolbox serves as the foundation for our implementation of the fuzzy function that we have suggested. Matlab provides convenient tools that may be used to develop and alter fuzzy inference systems. The membership function graphs of the factors are shown in Figure 4. The value of "Number of accesses" may vary anywhere from 0 to 55, and the simulation reveals that the maximum number of times a file can be accessed is 60. The value of "Replication cost" may be anywhere between 0 and 25, while the value of "Last access time interval" can be anywhere from 0 to12 *105. As a consequence of this, the value of the output parameter, also known as the

replica's value, might be anywhere from 0 to 1. The facsimile that has the least cost is an excellent entrant for removal from the database. In addition, 22 rubrics have been developed for the ambiguous system that is being suggested. Table 3 provides an explanation of some of the suggested ambiguous

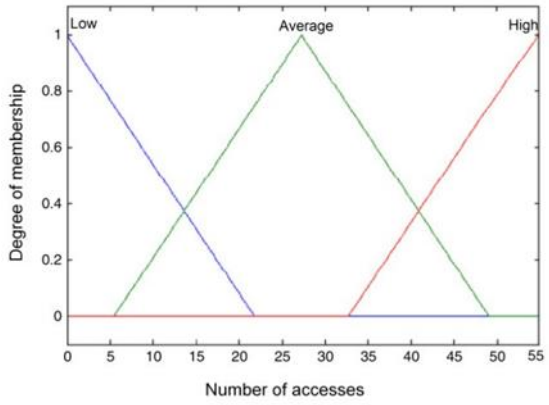

Fig.4 An explanation of the effort and production of the fuzzing implication system .

system rubrics. Flow chart of HRSs algorithms is exposed in Fig.5.

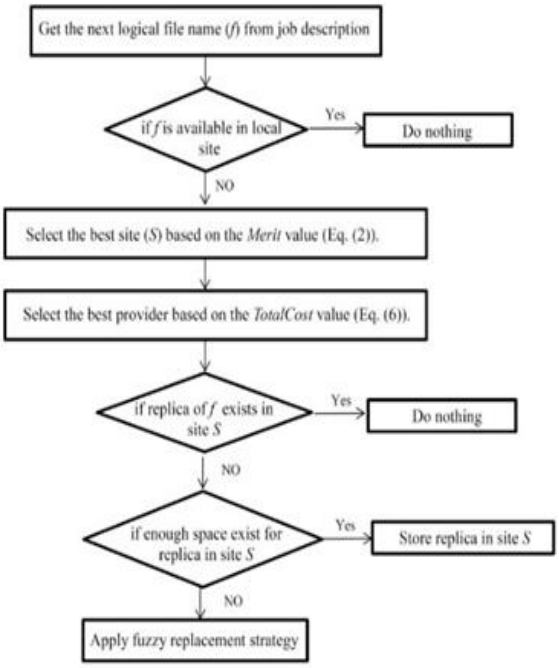

Fig. 5. Flowchart of HRSs.

4. Parameters of a Cloud Modeling Tool

In a virtual environment, several concerns are addressed, including issues such as virtual machine load balancing, cloud federation, data security, energy consumption, and scalability.These issues often serve as the primary focus of several cloud computing research projects, but with differing degrees of relevance. Cloud computing makes a variety of software and hardware capabilities available on an unprecedented scale.Dynamic provisioning or deprovisioning must manage the alternations in demand. Considering all these issues, we cannot directly apply the cloud computing system. We cannot do powerful evaluation for different approaches in real cloud with considering various QoS requirements of users such as speed, security and cost. [35] investigated the fast access security in Ubuntu clouds. Therefore a suitable simulator for experimental purposes is necessary. There are a number of simulators out there that attempt to depict the dynamics and development of such infrastructure, and here are a few examples: CloudSim [36], CDOSim [37], MDCSim [38] and iCanCloud [37]. Table 4 shows the widely used tools for modeling cloud systems, and displays whether or not they are utilized for assessing the performance or quality of service in a cloud environment[38].

4.1. Cloudsim's Internal Architecture

CloudSim is characterized by unique qualities that set it apart from other toolkits already on the market[39], such as SimJava and GridSim. [40]. Firstly, CloudSim performs in large-scale cloud infrastructure with data nodes, service brokers, virtual machines, and scheduling. In addition, CloudSim presents availability of virtualization engine and selection for allotment of both space and time on a shared basis[41]. A CloudSim structure with several layers is shown in figure 6. The functionality of SimJava and GridSim serves as the basis for the foundation of CloudSim. The most important features, aspects such as event queuing, the creation of system components, and communication between components are examples.and the administration of the clock, are included within these levels. The CloudSim layer presents an approach to the simulation of the data center that is predicated on virtualization. In addition to this, it is responsible for the execution of the fundamental components, include servers, guests, hosts, and software. When it comes to hosts, features of apps, users, the number of virtual machines, and so on, the User code layer is where everything is laid down in detail.

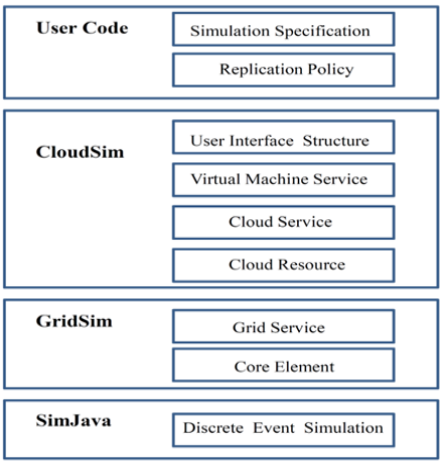

Fig. 6. CloudSim architecture .

4.2. CloudSim work style

Figure 7 indicates the work style of CloudSim platform. Generally, tasks of various users have a good deal of freedom to do their own thing. Assume m users, and label them User1, User2,..... In the same way that T1, T2,... are each a separate task, the userm tasks that make up the userm Tn, where n is the number of virtual machines, with VM1, VM2, etc. Datacenter1, Datacenter2,..., for VMn and p data centers. In-house Computer System: The CIS (Cloud data Service) assigns requests of user to suitable cloud providers. Resource discovery and information interaction between CIS and Data-CenterBroker are the core of simulated operations. DatacenterBroker: DatacenterBroker class mediates between consumers and service providers based on QoS requirements of users. In addition, it propagates service tasks across clouds by considering different parameters. Users can implement their proposed scheduling strategies in DataCenterBroker process. VmScheduler: Host component implements the abstract class of VmScheduler. The spacing-sharing and timing-sharing approaches for dispensation control allocation to virtual machines are provided by it. Therefore, users can override the components of this class to propose specific processor sharing methods.

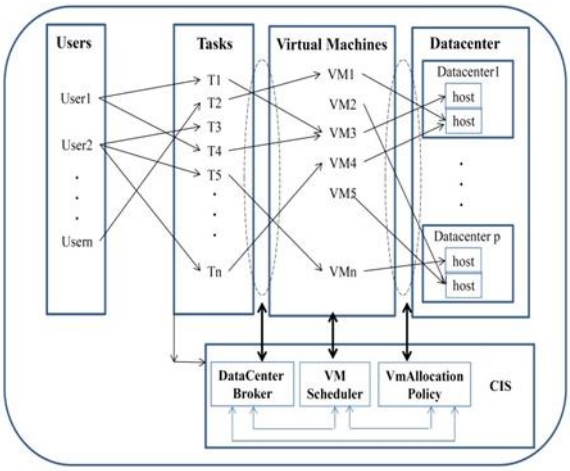

Fig. 7. CloudSim work style.

Resource Provisioning Policy is an abstract class that demonstrates the process of assigning hosts to virtual machines in a data center in a way that fulfills the storage and availability need for a VM deployment. Vm Allocation Policy can be found in the Vm Allocation Policy package.

5. Regulatory Context for Deployment

For the purpose of testing replication methods, we have enhanced many classes of the CloudSim toolkit. The parameters' settings for the cloud simulator are shown in Table 5. The simulation's parameters were determined on the basis of the previously conducted research [42] in order to provide a realistic depiction of a typical cloud environment. In CloudSim, we created a cloud system with 20 data centers and between 100 and 1,000 workloads. After completing one assignment, the next one is delegated using the Poisson distribution, and the necessary range for the number of files is between 1 and 5. The amount of computational labor involved in the challenge ranges from 100 Million Instructions to 5 Billion Instructions. We began the simulation by dispersing, at random, the main copies of all of the data files over a number of distinct locations.

6. An Assessment of Performance

As performance measures for the analytical assessment, we utilized the average response time, the mean latency, the hit ratio, the bandwidth consumption, the number of connections, the storage use, and the load variation. 6.1. The Typical Amount of Response Time If the time that elapses between the completion of a job and its return of results is referred to as the "response time," then the following equation may be used to get the average response time:

$$AverageResponseTime = \frac{\sum_{j=1}^{m}\sum_{k=1}^{m_j}(ts_{jk}(rt)-ts_{jk}(st))}{\sum_{j=1}^{m} m_j} \quad (8)$$

where the reporting time and the returning time of the outcome of task k for user j are denoted by the variables ts jk (st) and ts (rt), correspondingly, and the variable mj indicates the total amount of responsibilities for worker j.

Based on the results of the study shown in According to Figure 8, which illustrates a contrast of the regular reaction durations of sixing distinct active repetition processes, it would seem that EDR and HRSs had the longest and lowest average response times, respectively. While EDR's lower inflexibility to share the burden of replica access as well as its weakness to alter dynamic abilities both contribute considerably to an increase in the average response time, EDR's lower inflexibility to distribute the load is the primary factor. In addition, when compared with SWORD and HQFR, MORM has the potential to cut the average reaction time by up to 24 and 32 percent respectively. Since MORM takes into consideration a number of different characteristics, such as the We need to look at things like mean file unavailability, mean service time, load fluctuation, energy usage, and mean access latency to see how they're affected by changes in replica shape and number, and how that, in turn, affects performance. of each of them, As shown in Figure 8, the PGSA technique performs better than the EDR algorithm by a margin of up to 5% when the number of tasks is 1100. This is due to the fact that PGSA increases the number of copies in order to find a suitable duplicate. Under these circumstances, the

amount of stress on replica access was efficiently reduced and balanced.

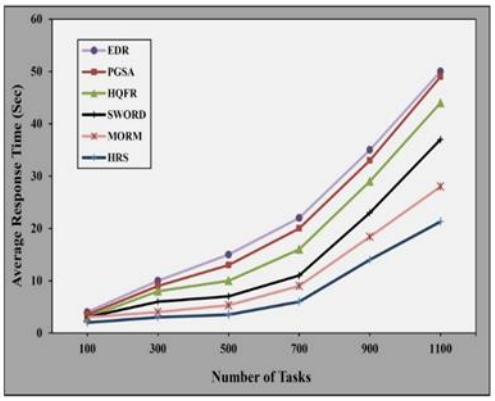

Fig. 8. Average response time by varying number of tasks.

6.1. Mean Latency

As shown in mean latency changes by different replication algorithms (Fig. 9), HQFR behavior is more desirable than EDR due to neglecting from latency as optimization target. Moreover, MORM is able to decrease the mean latency respect to HQFR and EDR equal to 14% and 27%, respectively. Since When optimizing, MORM takes into account the mean access latency[43]. The results of the studies reveal that the EDR technique is effective only when the connection bandwidth is the sole criterion for the network. However, in the cloud, this is not a practical scenario. When comparing latency with bandwidth, latency is the superior parameter to employ in the decision process. By looking at the data, we can see that HRS has a 15percentage points lower mean latency than MORM. This occurs because HRS prioritizes sites based on their suitability, taking into account factors including network speed, processing speed, and available memory[44].

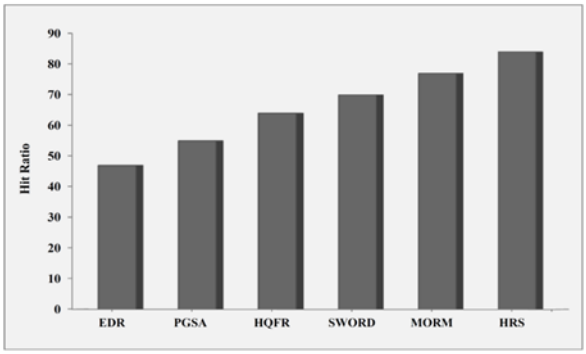

Fig. 9. different replication algorithms

6.2. Hits Percentage

The hit ratio is calculated as the total number of local file accesses divided by the total number to the number of accesses (i.e., local file accesses, total number of replications and total number of remote file accesses). Exhibit 10 presents an explanation of the hit ratio across various replication strategies when the number of tasks is set to 1000. As can be seen rather plainly from the data shown in Figure 10, HRSs has the greatest valuing of hit ratio when compared to the other replication methods[45]. The total number of local accesses has been enhanced as part of the HRS method by storing replicas in the right locations according to the amount of folder accesses and the criticality influence, while simultaneously circumventing duplication that isn't essential. As a consequence of this, the overall number of replications and remote accesses has been reduced, and as a direct result, the hit ratio has risen.

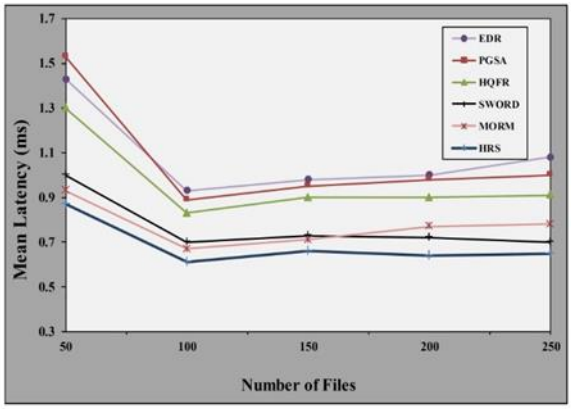

Fig. 10. Hit ratio for each of the six different replication procedures.

6.3. Use of Available Bandwidth

Figure 11 illustrates the overall bandwidth use of generally six different replication algorithms while using a variety of file sizes. It has been shown that HRS is capable of increasing performance, particularly in big number sizes. Because it takes into account centrality and the amount of replica accesses in file placement, the HRS approach performs better than the EDR, PGSA, HQFR, SWORD, and MORM algorithms. This is due to the fact that the HRSs strategy places more emphasis on centrality. The majority of the time, the required files may be found at the neighborhood site. Therefore, the technique that was suggested may be used in a distributed environment, which will result in enhanced data access performance as well as consumption of bandwidth[46]. Figure 11 illustrates that in comparison to EDR, the bandwidth usage of PGSA is 24% lower. This difference can be seen when comparing the two algorithms. This is due to the fact that the PGSA replica selection stage takes into consideration the state of the network, the load on each node, and the data history. Based on the concept of PGSA, it selects the proper data copy to reply with to the individual who has requested the data.

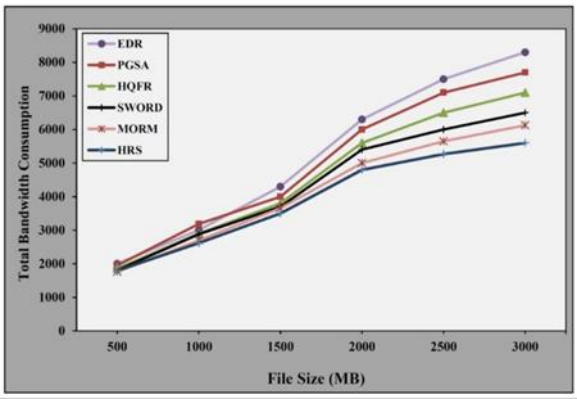

Fig. 11. Total bandwidth consumption for six replication strategies.

6.4. Amount of Contacts

The amount of communications for HRSs will be compared with the number of communications for other replication techniques in the future assessment that is scheduled. It is very necessary to cut down on the total number of communications in order to cut down on the information admission dormancy and avoid the mobbing of the bandwidth. When the arcs given in Figure 12 are compared, it is shown that HRSs performs better than MORM by 9% and that it performs better than SWORD by 30%. Based on the principles of temporal and geographical locality, HRSs chooses to store replicas at the location that is both optimal in terms of the number of access points it provides and also the location that is geographically most central. As a consequence of this, it has the potential to reduce the overall amount of communications between data centers.

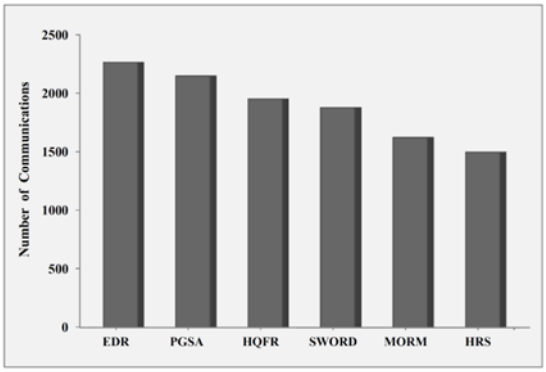

Fig. 12. Sum of all messages for each of the six different replication procedures.

6.5. Making Use of Storing Space

We are aware that the many alternative approaches that have been offered for replicating data may result in varied storage utilization, which in turn may influence the storage-capacity planning. The amount of space used by replicas according to the replication scheme may be stated as follows in Equation (9):

$$StorageUsage = \frac{Filled\_Space\_Available}{Space} \quad (9)$$

Monitoring the utilization of storage resources may give useful information since storage is clearly one of the most important components of a cloud system. This can be useful in the process of Putting forth a practical approach to replication while considering two crucial factors: (1) the possibility that storage space is expensive and thus should be used as little as possible; and (2) the possibility that storage space is expensive but should be used as much as possible.

The calculated amount of storage space used by the active algorithms. Instead of keeping replicas in a number of different locations, MORM allows for them to be put in the optimal position so that the amount of space needed for storage may be decreased. When compared to HQFR, the use of storage space is reduced by 17% thanks to SWORD's superior performance. This is in reference to the consumption of resources. This is due to the fact that SWORD establishes a file both of which minimize the average query span and the amount of resources that are used. While the EDR approach occupied more than half of the total storage space available in the cloud environment, the HRS strategy utilized up over half of the storage capacity that was used by EDR. The reason for this is that HRS will only duplicate the most important files depending on the value that is assigned to each file. When there is not enough room in a site's storage area, fuzzy rules choose replicas that have a low value to replace[47].

6.6. Variation in Demands

Usually, The load variance, also known as the standard deviation of data nodes in the cloud storage, is used to illustrate how load balancing works inside the network.. As shown in Figure. 13, HRSs yields a 30% at least smaller load variance than SWORD and a 34% at least than HQFR. Accordingly, HRS has the best load balancing respect to the other investigated strategies. This could be connected to HRS's skill in choosing, from among a huge number of replicas, the best one to use based on the load of the site and the I/O capacity of the disks.

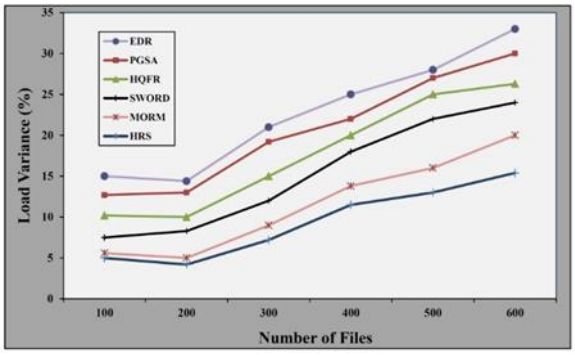

Fig. 13. Variation in load dependent on the varied numbers of files being processed.

7. Conclusion

The cloud computing system is becoming an increasingly popular topic of discussion as a unique approach to the administration of data. The method of data replication has seen widespread application in recent years as a means of enhancing data retrieval in cloud storage. In order to shorten the amount of time it takes for applications to respond, to provide high availability, and to ensure that the system load is evenly distributed, a novel replication approach known as HRSs has been developed. When it comes to the replica placement process, the locations are selected based on the centrality as well as the number of replica access points. This helps to cut response time by more than 25 percent. After that, it combines the capabilities of the network with those of the load process (for example, the capacity of the CPU to process data and the I/O capability of the disks) in order to choose replicas. However, because of the limitations imposed by the available storage capacity, a replica replacement technique is required in order to ensure the effectiveness of the dynamic replica management. The fuzzy inference system is used to determine the value of the replicas, which is then used to determine which file needs to be replaced and why. Using the CloudSim toolkit package, the newly developed replication mechanism is modeled and tested. HRSs not only ensures that data is accessible, but it also speeds up data access, and it maintains the reliability of the whole storage system. Based on the findings of the experiment, one can draw the conclusion that HRSs is capable of achieving a significant improvement in performance over earlier similar works. This improvement can be quantified in terms of the hit ratio, load variation, number of communications, bandwidth consumption, and storage utilization in addition to the average response time and effective network usage. In the future, one of our goals is to expand this form of load balancing to workflows that include jobs that are reliant on one another. We want to employ a model that accounts for energy usage in order to cut down on the amount of wasted resources and strike a balance between the various quality of service needs.

8. References

[1] Liu Q, Wang G, Liu X, Peng T, Wu J. Achieving reliable and secure services in cloud computing environments. Computers & Electrical Engineering, 2017; 59: 153-164.

[2] Jakóbik A, Grzonk D, Palmieri F. Non-deterministic security driven meta scheduler for distributed cloud organizations. Simulation Modelling Practice and Theory, 2017; 76: 67-81.

[3] Mishra S.K, Puthal D, Sahoo B, Jena S.K, Obaidat M.S. An adaptive task allocation technique for green cloud computing. The Journal of Supercomputing, 2017; 1-16.

[4] Foster I, Zhao Y, Raicu I, Lu S. Cloud computing and grid computing 360- degree compared. In: Grid Computing Environments Workshop, GCE'08, 2008; 1-10.

[5] Rajkumar B, Rajiv R, Calheiros R.N. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities. High Performance Computing & Simulation, 2009; 1-11.

[6] Ghemawat S, Gobioff H, Leung S. The Google file system. In: ACM Symposium on Operating Systems Principles, 2003; 29-43.

[7] Borthakur D. The Hadoop distributed file system: Architecture and design. Available: http://hadoop.apache.org/common/docs/r0.18.3/ hdfs_design.html, 2007.

[8] Feng D, Qin L. Adaptive object placement in object-based storage systems with minimal blocking probability. In: Proceeding of the 20th international conference on Advanced Information Networking and Applications, 2006.

[9] López-Pires F, Barán B. Many-objective virtual machine placement. Journal of Grid Computing, 2017; 15 (2): 161-176.

[10] Tao M, Ota O, Dong M. Dependency-aware dependable scheduling workflow applications with active replica placement in the cloud. IEEE Transactions on Cloud Computing, 2017; 99.

[11] Mansouri N, Kuchaki Rafsanjani M, Javidi M.M. DPRS: A dynamic popularity aware replication strategy with parallel download scheme in cloud environments. Simulation Modelling and Theory, 2017; 77: 177–196.

[12] Rahman R.M, Barker K, Alhajj R. Replica placement design with static optimality and dynamic maintainability. In: Sixth IEEE International Symposium on Cluster Computing and the Grid, 2006; 434-437.

[13] Shvachko K, Kuang H, Radia S, Chansler R. The Hadoop distributed file system. In: IEEE 26th Symposium on Mass Storage Systems and Technologies, 2010; 1-10.

[14] Mansouri N, Dastghaibyfard G.H. A dynamic replica management strategy in data grid. Journal of Network and Computer Applications, 2012; 35: 1297-1303.

[15] Ibrahim I.A, Dai W, Bassiouni M. intelligent data placement mechanism for replicas distribution in cloudstorage systems. In: IEEE International Conference on Smart Cloud (SmartCloud), 2016; 134-139.

[16] Mansouri N, Dastghaibyfard G.H, Mansouri E. Combination of data replication and scheduling algorithm for improving data availability in data grids. Journal of Network and Computer Applications, 2013; 36: 711-722.

[17] Mansouri N, Dastghaibyfard G.H. Enhanced dynamic hierarchical replication and weighted scheduling strategy in data grid. Journal of Parallel and Distributed Computing, 2013; 73: 534-543.

[18] Mansouri N. Adaptive data replication strategy in cloud computing for performance improvement. Frontiers of Computer Science, 2016; 1-11.

[19] Sun D.W, Chang G.R, Gao S, Jin L.Z, Wang X.W. Modeling a dynamic data replication strategy to increase system availability in cloud computing environments. Journal of Computer Science and Technology, 2012; 27: 256-272.

[20] Chang R.S, Chang H.P. A dynamic data replication strategy using access-weights in data grids. Journal of Supercomputing, 2008; 45(3): 277-295.

[21] Kim Y.H, Jung M.J, Lee C.H. Energy-aware real-time task scheduling exploiting temporal locality. IEICE Transactions on Information and Systems, 2010; 93(5): 1147-1153.

[22] Sun D.W, Chang G.R, Miao C, Jin L.Z, Wang X.W. Analyzing modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments. Journal of Supercomputing, 2013; 66: 193-228.

[23] Zhang B, Wang X, Huang M. A PGSA based data replica selection scheme for accessing cloud storage system. Advanced Computer Architecture, 2014; 451: 140-151.

[24] Ding X, You J. Plant growth simulation algorithm. Shanghai People's Publishing House, 2011; 1-59. [25] Li B, Song S.L, Bezakova I, Cameron K.W. EDR: An energy-aware runtime load distribution system for data-intensive applications in the cloud. In: IEEE International Conference on Cluster Computing, 2013.

[25] Lin J.W, Chen C.H, Chang J.M. QoS-aware data replication for data-intensive applications in cloud computing systems. IEEE Transactions on Cloud Computing, 2013; 1: 101-115.

[26] Long S.Q, Zhao Y.L, Chen W. MORM: A multi-objective optimized replication management strategy for cloud storage cluster. Journal of Systems Architecture, 2014; 60: 234-244.

[27] Luo Y, Li R, Tian F. Application of artificial immune algorithm to function optimization. In: Fifth World Congress on Intelligent Control and Automation, 2004; 3: 2248-2252.

[28] Lou C, Zheng M, Liu X, Li X. Replica selection strategy based on individual QoS sensitivity constraints in cloud environment. Pervasive Computing and the Networked World, 2014; 8351: 393-399.

[29] Kumar K.A, Quamar A, Deshpande A, Khuller S. SWORD: workload-aware data placement and replica selection for cloud data management systems. The VLDB Journal, 2014; 23: 845-870.

[30] Saleh A, Javidan R, Fatehikhaje M.T. A four-phase data replication algorithm for data grid. Journal of Advanced Computer Science & Technology, 2015; 4.

[31] Newman M. Networks: An introduction, Oxford University Press, 2009.

[32] Korat C, Gohel P. A novel honey bee inspired algorithm for dynamic load balancing in cloud Environment. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2015; 4.

[33] Dasgupta K, Kumar Mondal J, Dutta P. Optimized video steganography using genetic algorithm. In: International Conference on Computational Intelligence: Modeling, Techniques and Applications, 2013; 10: 131-137.

[34] Chang B, Tsai H, Huang C.F, Lin Z.Y, Chen C.M. Fast access security on cloud computing: Ubuntu enterprise server and cloud with face and fingerprint identification. In: Proceedings of the 2nd International Congress on Computer Applications and Computational Science, 2012; 451-457.

[35] Calheiros R.N, Ranjan R, Beloglazov A, De Rose C.A.F, Buyya R. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 2011; 41: 23-50.

[36] Fittkau F, Frey S, Hasselbring W. Cloud user-centric enhancements of the simulator CloudSim to improve cloud deployment option analysis. In: Proceedings of the 1st European conference on Service-Oriented and Cloud Computing, 2012.

[37] Lim S, Sharma B, Nam G, Kim E, Das C. Mdcsim: a multi-tier data center simulation, platform. In: Proceedings of IEEE International Conference on Cluster Computing and Workshops, 2009.

[38] Nunez A, Vazquez-Poletti J.L, Caminero A.C, Castane G.G, Carretero J, Llorente I.M. iCanCloud: a flexible and scalable cloud infrastructure simulator. Journal of Grid Computing, 2012; 10 (1): 185-209.

[39] Jararweh Y, Alshara Z, Jarrah M, Kharbutli M, Alsaleh M. Teachcloud: a cloud computing educational toolkit. In: Proceedings of the 1st International IBM Cloud Academy Conference, 2012.

[40] Garg S, Buyya R. Networkcloudsim: modelling parallel applications in cloud simulations. In: Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing, 2011; 105-113.

[41] Kliazovich D, Bouvry P, Khan S.U. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. The Journal of Supercomputing, 2012; 62(3): 1263-1283.

[42] Barroso L.A, Clidaras J, Holzle U. The datacenter as a computer: an introduction to the design of warehouse-scale machines. 2nd ed. Morgan and Claypool Publishers, 2013.

[43] Howell F, Mcnab R. SimJava: A discrete event simulation library for java. In: Proceedings of the first International Conference on Web-Based Modeling and Simulation, 1998.

[44] Khlebus, S.F., Hasoun, R.K., Sabri, B.T., " A modification of the Cayley-purser algorithm" International Journal of Nonlinear Analysis and Applications, 2022, 13(1), pp. 707–716.

[45] Razzaq Abdul Hussein, R., Hamza, Z.F., Sabri, B.T." Forecasting the number of COVID-19

infections in Iraq using the ARIMA model" Journal of Applied Science and Engineering (Taiwan)this link is disabled, 2021, 24(5), pp. 729–734.

[46] Bassam Talib Sabri, Noaman Ahmed Yaseen AL-Falahi, Isam Adil Salman, " Option for optimal extraction to indicate recognition of gestures using the self-improvement of the micro genetic algorithm" Journal of international journal of nonlinear analysis and application, vol. 12, no. 2, pp. 2295-2302, 2021.

[47] Bassam S Ali, Osman N Ucan, " Lossy Hyperspectral Image Compression Based on Intraband Prediction and Inter-band Fractal," in 2018 Proceedings of the Fourth International Conference on Engineering & MIS 2018, turkey, Istanbul, 2018/6/19.