Dynamic Task Scheduling Method in Cloud Computing Environment Using Optimized Neural Network

Noha Hamdy, Amal Elsayed Aboutabl, Nahla ElHaggar, and Mostafa-Sami M. Mostafa

***Abstract*—Cloud computing is a framework for enabling access to distributed computing resources. These resources may be extra storage, network bandwidth, memory space or processing power elements. The cloud user and service provider perceive the service in the cloud from different perspectives. The cloud user focuses on minimizing response time and cost of the service while the provider focuses on efficient utilization, of cloud resources, service reliability and minimization of maintenance costs. To satisfy both points of view, efficient methods to optimize cloudlets scheduling have to be provided. This paper proposes a scheduling method for cloud computing environment based on artificial neural networks (ANN) optimized with firefly algorithm to pic k out the most convenient scheduling algorithm.**

***Index Terms*—Cloud computing, artificial neural network, task scheduling, firefly algorithm, cloudlet scheduling controller.**

I. Introduction

A cloud is a type of parallel and distributed system [1]. It consists of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more computing resources. These resources are presented to the customer as a service. There are three categories of services according to National Institute of Standards and Technology (NIST) [2] standard:

- Infrastructure as a Service (IaaS).
- Platform as a Service (PaaS).
- Software as a Service (SaaS).

Virtualization is the golden key in cloud computing, it guarantees maximizing the effectiveness and performance of the shared resources. Cloud resources are shared by multiple users and dynamically re-allocated on demand. Virtualization software allows a single physical device to be accessed as one or more virtual devices. Each one is called a virtual machine (VM). Virtualization provides elasticity to serve customers requests to speed up IT operations, and reduces cost by increasing infrastructure utilization [3].

The incoming requests from cloud customers to service provider are called cloudlets. Every cloudlet has an assigned instruction length by Million Instructions (MI).

Scheduling is a planning process to assign particular tasks for processing. Cloud environment has two levels of scheduling as shown in Fig. 1, the first one is scheduling of incoming cloudlets to available VMs and the second one is virtual machine scheduling [4].

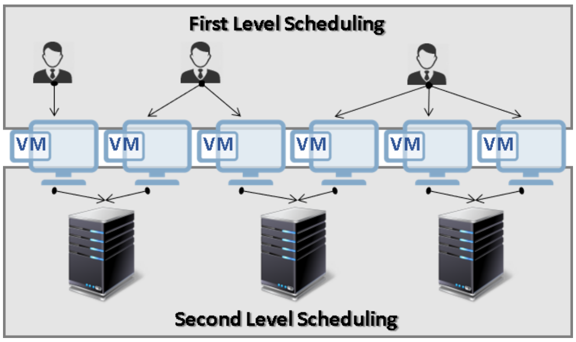

Fig. 1. Scheduling in cloud computing.

Cloudlets scheduling deals with the problem of deciding which of the cloudlets in the arriving queue will be allocated to the VM and what is the applied mechanism to achieve this mission. The job waits in a queue until it is scheduled and executed. The time spent in the queue, or wait time, depends on several factors:

- Job priority.
- Load on the VM.
- Availability of requested resources.

Turnaround time is the total time elapsed between job submission and return of the complete output to the user. It includes the waiting time as well as the jobs actual execution time. Response time is the time elapsed between job submission and the response the user gets from the system. Consequently, response time is an important performance metric for cloud customers, who look forward to get minimal response time [5]. The purpose of the first level of cloudlet scheduling is to balance the performance of different virtual machines.

The second level of cloudlet scheduling is virtual machine scheduling which is responsible for provisioning the physical nodes resources among different virtual machines. The number of tasks that run on a virtual machine varies from time to time. Therefore, the resources utilization varies over time. In such a case, some of the virtual machines suffer from lack of memory and computing capacity.

Efficient virtual machines scheduling mechanisms can improve resources utilization by migration process. Migration refers to the process of moving a running virtual machine or application between different physical devices without disconnecting the client or application. Both cloudlet scheduling and virtual machine scheduling are the

Manuscript received September 25, 2017; revised December 7, 2017.

The authors are with the Computer Science Department, Faculty of Computers and Information, Helwan University, Cairo, Egypt (e-mail: noha7amdy@yahoo.com).

doi: 10.18178/ijfcc.2017.6.4.507

core that affects the performance of the cloud.

The paper is organized as follows: Section II presents related work. Section III illustrates the firefly algorithm. Section IV illustrates the artificial neural networks methodology. Section V presents the proposed model. Section VI shows the performance analysis of the proposed approach and finally, Section VII concludes the paper.

II. Related Work

Nature Inspired Algorithms have gained attention to solve different optimization problems. Such algorithms include ant colony, bee colony, firefly, particle swarm optimization, bat algorithm and artificial neural network [6].

Esa and Yousif [7] use the firefly algorithm to minimize the time to complete tasks in a cloud-computing environment. The method based on jobs and resources data such as length of job and speed of resources. The scheduling function firstly creates a set of jobs and resources to generate the population by assigning the jobs to resources randomly and evaluating the population using a fitness value that represents the execution time of jobs. Secondly, the function uses iterations to regenerate populations based on firefly attitude to produce the best job schedule that gives the minimum execution time of jobs. Guo *et al*. [8] apply a Particle Swarm Optimization (PSO) algorithm; assign each solution to a particle in the swarm. Linan Zhu1 *et al*. [9] use an ant colony algorithm, by first checking the priority of incoming tasks then applying ant colony algorithm based on shortest execution time. Daoud *et al*. [10] present a two-steps algorithm called Hybrid Heuristic-Genetic Scheduling (H2GS) for tasks scheduling in heterogeneous distributed computing systems based on Genetic Algorithm (GA).The first phase of the algorithm is based on Longest Dynamic Critical Path (LDCP) list for scheduling with high quality. The second phase receives from the first stage output as the initial population of a GA. The function of the second phase leads to a shorter schedule. The efficiency of H2GS is compared with two scheduling methods of Dynamic Level Scheduling (DLS) and Heterogeneous Earliest Finish Time (HEFT). Results show that this algorithm outperforms the mentioned methods.

Heuristic Search Algorithms are another class of algorithms that have been used to solve the resource allocation problems. This category of algorithms follows a predefined procedure, such as FCFS, Round Robin, Shortest Job first, Min-Min algorithm and Max-Min algorithm. Kokilavani *et al*. [11] present the Minimum Execution Time (MET) algorithm based on the Min-Min algorithm. It allocates each task to the resource with minimum expected execution time, without considering the resource availability at that time. The idea of this approach is to assign each task to its best resource. This method results in a severe load imbalance across the resources. Maniyar *et al.* [12] apply the Round Robin (RR) [5] algorithm; one of the most common mechanisms in task scheduling. It is based on assigning an equal time slice (quantum) to each cloudlet in order and in a circular queue without priority. Tasks that are unable to complete during its turn, will go back to the queue waiting for another turn. RR scheduler generally uses time-sharing. The drawback of this algorithm is that if the virtual machine is heavily loaded it will take longer time to complete execution of all tasks. The main benefit is that no task has to wait for another one to complete as in First Come First Served (FCFS) [13] and others.

III. Firefly Algorithm

The firefly algorithm (FFA) presented by Xin-She Yang [14] simulates firefly behavior. It belongs to nature-inspired meta-heuristic algorithms that simulate animals' behavior to reach food such as ant colony, bee colony, particle swarm optimization, etc... It relies on the flashing of each firefly to define the intensity of light for other ones. The main purpose of their flashing is to attract a mate. There are three rules to govern the algorithm, and create a modeled firefly behavior.

- Fireflies are unisex, so they are attracted to each other based on the intensity of each firefly.
- Attractiveness is proportional to brightness; a less bright firefly will move towards a brighter firefly.
- The brightness of a firefly is related to the value of the objective function being maximized.

When comparing the brightness of any two fireflies, the locations of the fireflies must be considered. When the distance increases between two fireflies, then both attractiveness and brightness will decrease dramatically. In addition, if a firefly does not find a firefly in its surroundings, it will travel in a random direction.

The intensity of light of each firefly is the main property of firefly algorithm, which is responsible for attracting the mating fireflies. Suppose there exists a swarm of n fireflies and the solution for the ($i^{th)}$) firefly is (I_i), where f (I_i) denotes its fitness value. Now, the brightness (β) of a firefly is chosen to reflect its current position (I) of its fitness value f (I).

$$\beta_i = f\,(I_i), \qquad \text{where } 1 \leq i \leq n \tag{1}$$

All fireflies have their attractiveness value β, which indicates how it can attract other fireflies in the swarm. The attractiveness β will diverge with its distance factor d_{ij} at the locations X_i and X_j, between the two corresponding fireflies i and j is given by the comparison.

$$d_{ij} = |X_i - X_j| \tag{2}$$

The attractiveness function β of the firefly is computed as:

$$\beta = \beta_0\, e^{-\gamma\, \mathrm{r2}} \tag{3}$$

where β_0 is attractiveness at r=0 and γ is coefficient of light absorption.

The movement of the less bright firefly toward the brighter firefly is computed by

$$X_i = X_i + \beta_0\, e^{-\gamma \mathrm{r2ij}}\, (X_j - X_i) + \alpha\ (\mathrm{r\ and} - 1/2) \tag{4}$$

where α is the randomization parameter and rand is a randomly selected number in the interval [0, 1].

IV. Artificial Neural Network

Artificial Neural Network (ANN) consists of an interconnected group of nodes (artificial neurons). It is based on the idea of the big network of neurons in the human brain. It has internal adjustable parameters called weights that refer to the strength of a connection between two nodes (neurons). This network can learn an arbitrary vector mapping from the space of input to the space of output by modification of weights [15].

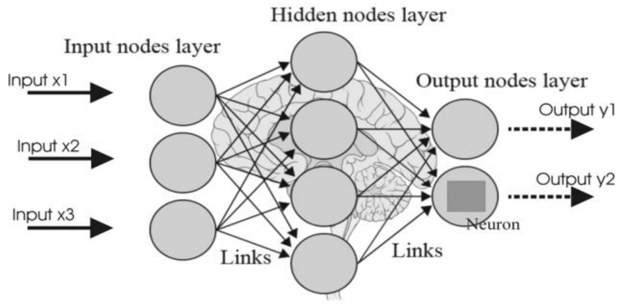

Fig. 1. ANN structure [20].

An ANN is composed of three layers as shown in Fig. 2, the input, hidden and output layers. The output of each neuron is a function of the weighted sum of the inputs weights. The output of the ith neuron can be described by:

$$y_i = f_i(\sum_{j=1}^{n} w_{ij} x_j + \theta_i) \tag{5}$$

Majority of ANNs use sigmoid functions as activation function, because it is smooth, continuous, and gradually increasing (derivative is always positive). It has the shape of an s-shaped curve and is represented by Eq. 6 [16].

$$F(x) = 1/ (1 + e^{-x}) \tag{6}$$

There are different architectures of ANN as feed forward, feedback and recurrent neural network. ANN is based on learning from experience (by example). There are two types of learning in neural networks, supervised learning and unsupervised learning.

In unsupervised learning, only inputs are supplied. In supervised learning, a neural network is supplied with inputs and the desired outputs. In each epoch the weights are modified to reduce the difference between the actual and desired outputs in such a way that the Mean Square Error (MSE) between the desired output and the actual output is reduced. This process requires that the neural network computes MSE in each epoch. This means that it must calculate how the error changes at each weight and check if it is increased or decreased slightly [15]-[17].

$$\text{MSE} = (\text{target} – \text{output})^{2} \tag{7}$$

The back propagation algorithm is the most widely used method for determining the MSE. Due to reach an optimized ANN we have to reach the optimal weights and minimum MSE in a few epochs, therefore we propose using the firefly algorithm to compute the weights of ANN and generate “Firefly-Artificial Neural Network” FF-ANN.

Firefly Artificial Neural Network

Artificial neural network weights are main factors affect the ANN output and MSE therefore, as a number of epochs to reach the minimum MSE decreased the better ANN is reached.

The firefly algorithm is selected because of its random scenarios nature. The algorithm is in a position to search a collection of solutions through randomisation. In meta-heuristic algorithms, the solutions are searched through trial and error rules and the probabilities of obtaining the best output in feasible time is higher. Meta-heuristic searching methodology depends on two concepts exploitation and exploration. Exploitation is searching for best solutions among the neighbours to extract the data, while exploration is the method of searching the varied solutions within the search area. The firefly algorithm is a meta-heuristic algorithm, therefore its search method overcomes some drawbacks of other local search algorithms and randomisation is performed at some higher level, i.e. the algorithm produces the simplest work solution from the amount of freshly generated solutions within the search area. Due to the randomisation nature, the algorithm is not stuck at local minima [6].

Firefly artificial neural network is a series of back propagation algorithm combined with firefly algorithm to optimize the feed forward neural network. The following steps identify how to compute the FF-ANN weights:

- The weights for ANN is computed using the light of the fireflies’ intensity.
- Each firefly is considered as a set of weights of ANN at an instance of time. The population of fireflies is generated randomly, initial brightness of the fireflies is initialized to 0.4 and absorption coefficient is set to 0.6.
- Then the fitness of each firefly is calculated by using firefly equations Eq. (1) which is used as intensity of the fireflies.
- Apply each firefly weights and calculate MSE Eq. (7). In each epoch.
- Calculate the attractiveness Eq. (3) and the less bright moves towards the brighter one Eq. (4).

V. Proposed Model

The aim of the presented model is minimizing the response time of a service and maximizing resources utilization. Therefore, we build a “**Cloudlet Scheduling Controller**” (CSC) component based on FF-ANN.

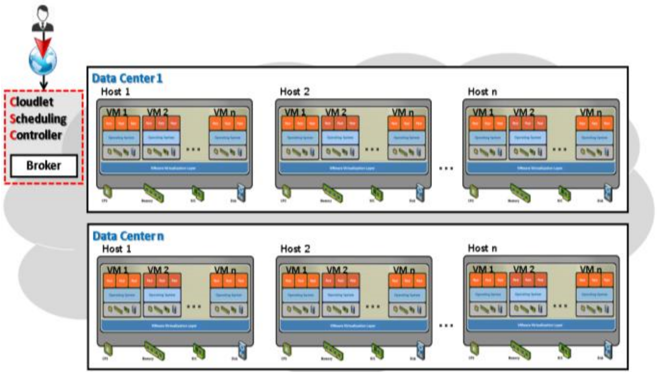

Fig. 3. Proposed cloud architecture with CSC.

As shown in Fig. 3, Cloud Resources Controller –CSC– is a newly added component to the cloud-operating environment; it is an intermediate controller in cloudlet scheduling process. The CSC is responsible for two functions:

- The first function is predicting the execution time of each incoming cloudlet from different customers of the cloud. Moreover, it observes resources utilization variations through processing.
- The second function is picking out the most convenient task-scheduling algorithm then applying it to the cloud according to the first phase output.

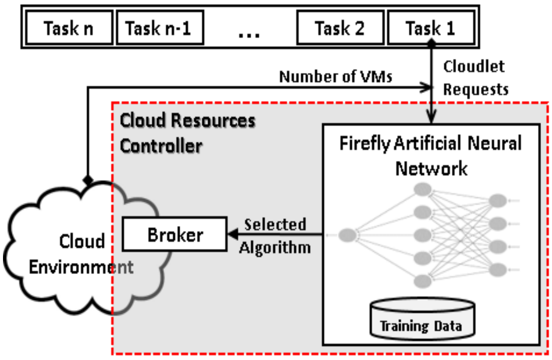

Fig. 4. CSC architecture.

The CSC fetches the incoming tasks from cloud customers in their arriving order. In addition, CSC gets the number of available resources in the cloud as shown in Fig. 4. Both as the number of incoming cloudlets and the number of available resources are the inputs to the FF-ANN.

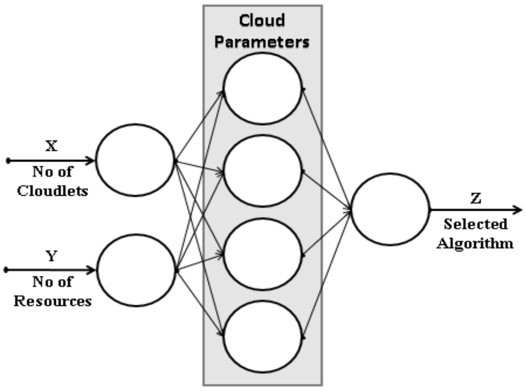

Fig. 5. ANN architecture in CSC.

The used neural network is shown in Fig. 5 and is composed of three layers; input layer, one hidden layer and output layer. The input layer has two inputs; the number of cloudlets and the number of available resources denoted as ‘x’ and ‘y’ respectively. The hidden layer simulates cloud parameters and contains four neurons according to thumb rule to identify the number of hidden layer and hidden nodes [17]. The output layer contains one neuron denoted as ‘z’ that represents the selected algorithm. Each neuron in the hidden layer performs the summing up of the weighted inputs and applies log-sigmoid function (Eq. 6), then computes the MSE using (Eq. 7) and adjust weights in the ANN. The FF-ANN has trained with previous experience a dataset size 250 records, inputs (number of cloudlets and number of available resources) and number of classes is four.

VI. Experimental Results and Analysis

This section presents simulation of the proposed model and performance analysis of the output of this work. We use CloudSim simulator version 3 that simulates cloud computing infrastructures and services [18]. CloudSim runs on Windows 8, Eclipse version is Neon3, the JDK version is jdk1.6.0

We have simulated two data centers with one host in each. The number of incoming cloudlets and available resources is variable over time. The total execution time of the proposed model is compared with three methods in cloudlet scheduling; time-shared method [13], [19] that simulates Round Robin (RR) scheduling, space-shared method that simulates First Come First Served (FCFS) [19] and priority algorithm that executes the incoming cloudlets with priority regardless of the arrival time, where the priority factor here is cost. These algorithms are selected because they are commonly applied in the cloud environment.

In Table I, every row represents a case where the number of available resources and the number of incoming cloudlets is varied. In each case, the turnaround time (in ms) is shown for each method and for the proposed model.

TABLE I: Turnaround Time Comparison

# of Resources	# of Cloudlets	Turnaround time (ms)			
		Priority	Space Shared	Time Shared	Proposed Model
10	**25**	52.87	41.84	69.7	46.84
30	**50**	172.94	121.75	129.21	100.21
60	**80**	316.73	285.43	300.74	238.74
120	**210**	327.12	300.39	410.31	370.39
150	**350**	758.61	801.92	958.51	580.61
200	**500**	1061.3	913.62	1124.6	866.25

The proposed model selects different task scheduling algorithm in every turn based on the available resources and amount of incoming cloudlets, which affects the elapsed turnaround time. Fig. 6 shows a statistical representation of differences in the turnaround time.

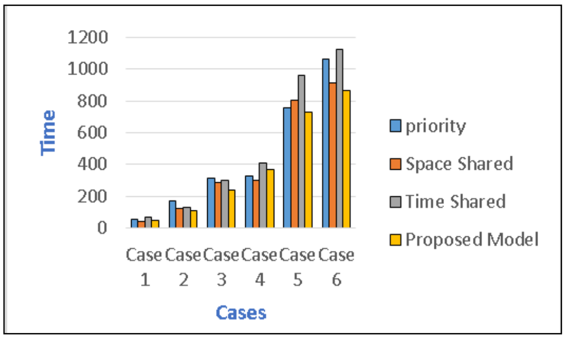

Fig. 6. Turnaround time comparison.

As shown in Fig. 6, the proposed model achieves the minimum execution time in cases 2, 3, 5 and 6. Results show that this model shows its benefits in heavy load clouds as cases 5 and 6.

VII. Conclusion

We designed and implemented a component to allow dynamic selection of the task-scheduling algorithm in cloud computing environment. This component applies optimized neural network with firefly algorithm. The neural network is trained with previously experimented dataset. The goal is achieving fast convergence rate and to minimize the training error. Experiments show that CSC can specify the appropriate task-scheduling algorithm in different circumstances.

REFERENCES

[1] A. S. Tanenbaum and M. V. Steen, "Distributed coordination based system," *Distributed Systems Principles and Paradigms*, 2nd ed., New Jersey: Prentice Hall, 2006.

[2] T. Grance and P. Mell, "The NIST definition of cloud computing," Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, September 2011.

[3] T. Erl, Z. Mahmood, and R. Puttini, "Understanding cloud computing," *Cloud Computing Concepts, Technology and Architecture*, 2nd ed., United States of America: Prentice Hall 2013.

[4] Y. Chawla and M. Bhonsle, "A Study on scheduling methods in cloud computing," *International Journal of Emerging Trends and Technology in Computer Science*, vol. 1, no. 3, 2012.

[5] A. Silberschatz, P. B. Galvin, and G. Gagne, "Process scheduling," *Operating System Concepts Essentials*, 8th ed., USA: John Wiley: 2009.

[6] X. S. Yang, "Introduction," *Nature Inspired Metaheuristic Algorithms*, 2nd ed., United Kingdom: Luniver Press, 2010.

[7] D. I. Esa and A. Yousif, "Scheduling jobs on cloud computing using firefly algorithm," *International Journal of Grid and Distributed Computing*, vol. 9, no. 7, pp.149-158, 2016.

[8] L. Z. Guo, S. G. Zhao, S. G. Shen, and C. Y. Jiang, "Task scheduling optimization in cloud computing based on heuristic algorithm," *Journal of Networks*, vol. 7, no. 3, March 2012.

[9] L. N. Zhu, Q. S. Li, and L. N. He, "Study on cloud computing resource scheduling strategy based on the ant colony optimization algorithm," *International Journal of Computer Science Issues*, vol. 9, no. 5, September 2012.

[10] M. I. Daoud and N. Kharma, "A hybrid heuristic–genetic algorithm for task scheduling in heterogeneous processor networks," *Journal of Parallel and Distributed Computing*, vol. 71, pp. 1518-1531, 2011.

[11] T. Kokilavani, and A. D. I. George, "Load balanced min-min algorithm for static meta-task scheduling in grid computing," *International Journal of Computer Applications*, vol. 20, 2011.

[12] B. Prasad, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. Fifth International Joint Conference on INC*, 2009.

[13] B. Maniyar and B. Kanani, "Review on round robin algorithm for task scheduling in cloud computing," *Journal of Emerging Technologies and Innovative Research*, vol. 2, no. 3, 2015.

[14] X. S. Yang, "Firefly algorithms for multimodal optimization," Springer-Verlag Berlin Heidelberg pp. 169-178, 2009.

[15] D. Kriesel, "Components of artificial neural networks," *A Brief Introduction to Neural Networks*, 2005.

[16] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, "Neuron model and network architectures," Neural Network Design, 2nd ed., USA: Martin Hagan, 2014.

[17] K. Gurney, "Real and artificial neurons," *An introduction to Neural Networks*, 1st ed., London: UCL Press limited, 1997.

[18] R. Kumar and G. Sahoo, "Cloud computing simulation using cloudsim," *International Journal of Engineering Trends and Technolog*, 2014.

[19] Himani and H. S. Sidhu, "Comparative analysis of scheduling algorithms of cloudsim in cloud computing," *International Journal of Computer Applications*, vol. 97, no. 16, July 2014.

[20] K. M. Tadiou. (2014). The future of human evolution. [Online]. Available: http://futurehumanevolution.com/artificial-intelligence-future-human-evolution/artificial-neural-networks

Noha Hamdy is currently a Ph.D. student at the Faculty of Computers and Information, Helwan University. She works as an assistant lecturer at modern academy in Maadi. She received her B.Sc. and M.Sc. in computer science from Faculty of computers and information, Helwan University.

Amal Elsayed Aboutabl is currently an Associate Professor at the Computer Science Department, Faculty of Computers and Information, Helwan University, Cairo, Egypt. She received her B.Sc. in Computer Science from the American University in Cairo and both of her M.Sc. and Ph.D. in Computer Science from Cairo University. She worked for IBM and ICL in Egypt for seven years. She was also a Fulbright Scholar at the Department of Computer Science, University of Virginia, USA. Her current research interests include parallel computing and software engineering.

Nahla Elhaggar is currently an assistant professor at Information Technology Department, Faculty of computers and information, Helwan university. Cairo, Egypt. She received both B.Sc. And M.Sc.in Mathematics and Computer Science from Ain Shams University, Cairo, Egypt and Ph.D. in Scientific Computing from Cairo University, Egypt. She worked for National Research Institute of Astronomy and Geophysics (NRIAG), Helwan, Cairo, Egyptfor20 years. Her current research interests include natural language processing, image processing and information security.

Mostafa-Sami M. Mostafa is currently a Professor of computer science, Faculty of Computers and Information, Helwan University, Cairo, Egypt. He worked as an Ex-Dean of faculty of Computers and Information Technology, MUST, Cairo. He worked also as an Ex-Dean of student affairs and Ex-Head of Computer Science Department, faculty of Computers and Information, Helwan University, Cairo, Egypt. He is a Computer Engineer graduated 1967, MTC, Cairo, Egypt. He received his MSC 1977 and his PhD 1980 from University of Paul Sabatier, Toulouse, France. His research activities are in Software Engineering and Computer Networking. He is awarded supervising more than 80 Masters of Sc. and 18 PhDs in system modeling and design, software testing, middleware system development, real-time systems, computer graphics and animation, virtual reality, network security, wireless sensor networks and biomedical engineering.