TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.7, July 2014, pp. 5559 ~ 5566
DOI: 10.11591/telkomnika.v12i7.4643

Intuitionistic Fuzzy Set Based Task Scheduling with QoS Preference Awareness

Wang Juan
School of Network Engineering, Chengdu University of Information Technology
Chengdu, China
email: wangjuan@cuit.edu.cn

Abstract

Existing task scheduling algorithms in cloud computing have lack the ability to be aware of users' quality of service (QoS) preference. In order to address this problem, "intuitionistic fuzzy analysis" is introduced to determine users' QoS preference. In addition, the "optimal sequence decision method" helps experts use their professional knowledge to decide the weights of QoS classes. Using these methods, we propose the "Intuitionistic Fuzzy Set Based Task Scheduling with QoS Preference Awareness: IFS-QoS PA" algorithm. By considering both user and expert experience, the method can determine users' QoS preference and reflect the characteristic of cloud storage system. The simulation results show that this method offers acceptable user satisfaction rate, has low computation complexity and more suitable for large scale task scheduling as compare to particle swarm optimization based ones.

Keywords: *quality of service, preference awareness, intuitionistic fuzzy set, task scheduling, cloud storage*

1. Introduction

With the development of the Internet of Things (IoT), the Internet extends to various terminals and creates explosive growth data. For the huge amounts data, not only are the hardware and softwares investment beyond imagination, but also the lack of professional data maintenance administrator make it is impossible for individuals to build data center themselves. In order to deal with the huge amounts of data, “cloud storage”has been suggested as a solution for these problems. The idea of cloud storage is that the professional cloud storage servicer build the cloud storage system include hardware and software and provide the “storage service” to individual users according to their need. This way avoids the repeat investment in both hardware and software and save the costly maintenance charge.

Cloud storage system deals with huge amounts data and tasks. The global throughput improvement, resource optimization and profit maximization are the ultimate objectives of the system [1-14]. Task scheduling algorithms are desiged to address these problems and play an important role. There are already many task scheduling algorithms in cloud computing area, but few ones in cloud storage system. These schemes aim to get higher system throughput, namely shorter makespan(the time difference between the start and finish of a sequence of jobs or tasks), such as the Min-Min and Max-Min algorithm [1] which are enumeration method and enumerate all the possible solutions and select one of it as the optimal solution. When number of instances is large, these methods cost unaccepted time and space which make them are not feasible for scheduling, then the heuristic algorithm is suggested to find reasonably solutions, such as ant colony based scheduling algorithmm [2], genetic algorithm (GA) based scheduling algorithms [3, 4], simulated Annealing (SA)based scheduling algorithms [5], particle swarm optimization(PSO) [6]. Another aim of scheduling algorithms is load balancing (make tasks are dispatching to resource nodes averagely), include weighted mean time(WMT) algorithm [7] and some of heuristic algorithms [7-10].

In recent years, the quality of service (QoS) has received increasing attention which is used to quantitatively measure aspects of the network service, such as error rates, bandwidth, throughput, transmission delay, availability, jitter, etc. Many QoS guided task-scheduling methods have been proposed, such as QoS Guided Min-Min heuristic [11], based on the Min-Min algorithm, which considers network bandwidth as the QoS parameter. The heuristic method

Received October 7, 2013; Revised January 7, 2014; Accepted February 2, 2014

which integrate the QoS Min-Min heuristic and the WMT heuristic (QWMTM) [12] can not only guarantee QoS but also guarantee load balance. Few multi-QoS guarantee algorithms [13-15] use multiple workflows or multiple components to separate deal with single QoS.

These methods are simply drawn from cloud computing and lack of QoS preference awareness (PA) ability, which is important for users. Furthermore, most of these algorithms take the system throughput improvement as the goal and ignore the user's requirements and result in low user satisfaction rate. Finally, the QoS factors are too professional to users to understand. So the weights of QoS factors are usually decided by researchers or technicians and do not reflect the user's real demand.

In order to address above problems, we transfer the technical QoS factors into user understandable QoS classes, and introduce the "intuitionistic fuzzy analysis (IFS)" into task scheduling area to help users describe the importance of QoS classes. By this way every QoS class get its weight from users' intuition and reflect users' requirement. The proposed task scheduling is judgeding by the "intuitionistic fuzzy comprehensive evaluation value (IFCEV)". The highest IFCEV node will dispatch to task and update the IFCEV for next scheduling. Simulation results show that compared with existing algorithms, the proposed algorithm can satisfy user's QoS preference, and has low complexity and high execution efficiency.

The remainder of this manuscript is organized as follows. Section 2 describes the details of our method that introduces intuitionistic fuzzy analysis into task scheduling in cloud storage to satisfy QoS preference. The simulations and comparisons analysis are presented in Section 3. Finally, Section 4 presents a short conclusion and future works.

2. IFS Analysis Based Task Scheduling with QoS Preference Awareness (IFS-QoS PA)

The reasons why existing scheduling algorithms lack support for QoS PA and do not let users themselves to decide the priority level maybe the following:

(1) There are contradictions among QoS factors, such as transmission speed and transmission quality. In order to ensure the transmission quality, the timeout detection and retransmission mechanism are need which cost more time, namely make the transmission speed slowly down. And the cost is also has contradiction with speed and quality. Lower cost and higher speed are need by user. However system cannot usually satisfy both of them at the same time.

(2) User lack of the knowledge of technical QoS factors that make users cannot decide the important level of them.

In order to overcome the above problems, we introduce IFS into task scheduling to help users describe the importance of QoS classes.

In this section, the related definitions are given firstly, and then the task scheduling based on IFS is described in detail.

2.1. Intuitionistic Fuzzy Set Related Definitions

IFS [16, 17] is extended from fuzzy set theory, which uses the degree of membership and degree of non-membership to describe the uncertain information, and for this advantage it is used in many fields already.

Definition 1. IFS: Let a set X fixed, then $A=\{(x,\mu_A(x),v_A(x))|x\in X\}$is a intuitionistic fuzzy set, where $\mu_A(x)$ is the degree of membership(the percent of x belong to A) and the $v_A(x)$ is the degree of non membership(the percent of x not belong to A) of the element $x\in X$ to $A\in E$,respectively. The functions μ_A and v_A should satisfy condition: $0\le\mu_A(x)+v_A(x)\le1$, $x\in X,\mu_A:X\rightarrow[0,1],v_A: X\rightarrow[0,1]$。

Definition 2. Indeterminacy Degree (π): For any IFS A, Let $\pi_A=1-\mu_A(x)-v_A(x)$, $x\in X$ denotes the degree of indeterminacy (Intuitionistic Index) which means the uncertainty level of $x\in A$, where $0\le\pi_A(x)\le1,\forall\ x\in X$。

Definition 3. Intuitionistic Fuzzy Number (*IFN*): The triple $[\mu(x),v(x),\pi_A]$ is defined as the intuitionistic fuzzy number(*IFN*). Then the set of *IFN*s is also defined as the *IFS*, denote as $A=\{[\mu(x),v(x),\pi_A]|x\in X\}$。

Definition 4. Intuitionistic Fuzzy Weight (*IFW*):

Let the weight coefficient *w* represents the relative importance level for one QoS attribute to all QoS attributes. This represents the attribute's impact on comprehensive evaluation when other QoS attributes are fixed. Usually, user's QoS requirements preference is described by language, such as important, not important, and ordinary. Here we use language description to get the weight by Table 1 definition.

In Table 1, the unknown variable π represents indeterminacy degree of user to the QoS attribute. And different user may give the different indeterminacy degree. We use percentage to describe the indeterminacy degree π, $\pi \in$(0~100%).

Table 1. The Language Description of QoS Requirements Preference [16]

Language description	Intuitionistic fuzzy numbers
Very important (5)	[0.9,0.1, π]
Important (4)	[0.7,0.3, π]
Ordinary (3)	[0.5,0.5, π]
Not important (2)	[0.3,0.7, π]
Not important at all (1)	[0.1,0.9, π]

Then the *IFW* for QoS attribute is defined as:

$$w_j=\mu(q_j)-v(q_j)\times\pi \quad (3)$$

Where $w_j \in [0,1], \sum w_j=1$, $j=1,2,\ldots,$ n. So conversion process is need to make weight w_j fall in [0, 1]. Conversion formula is:

$$w_j=w_j/(w_1+w_2+\ldots+w_n) \quad (j=1,2,\ldots,n) \quad (4)$$

Definition 5. Intuitionistic Fuzzy Comprehensive Evaluation Value (*IFCEV*):

Assume $x_1,x_2,\ldots,x_n$ is a group QoS attributes(classes) of candidate nodes, and $w_1,w_2,\ldots,w_n$ ($w_j \in [0,1]$, $\sum w_j=1$, $j=1,2,\ldots,n$) are *IFW* for these candidate nodes. Then the weighted mean sum is defined as the *IFCEV* as:

$$IFCEV=x_1w_1+x_2w_2+\ldots+x_nw_n=\sum x_jw_j \quad (5)$$

IFCEV is a comprehensive evaluation for candidate node to certain QoS class task. The candidate node with bigger *IFCEV* can satisfy user QoS requirement well than the lower *IFCEV* one.

2.2. The TQC Transfer to IQC by Optimal Sequence Method (OSM)

As mentioned above, the users lack of the knowledge to decide the important level of technical QoS factor. This is why researchers decided to stop letting users themselves to decide QoS factor weight. We want to find a balance between professional experience and user intuitive experience by classifying these technical QoS factors (TQF) into user understandable intuitive QoS classes (IQC), and implement IFS among IQC.

According to the existing research and experience [1-14], there are three main classes that users take care of. They are cost, time and quality. By this way, users can only weight the classes by their intuitive experience instead of face to many technical QoS factors. Let user to describe the importance of QoS factor that they totally do not understand is extremely unreasonable.

In these three main IQC, there are many technical QoS factors. We transfer technical QoS factors into intuitive class by using OSM as follows:

Assume there are n factors (c_i) in IQC and these factors have been normalized as section 3 described. Let w_i be the weight of the factor(c_i), satisfy $w_i \in [0,1]$, $\sum_i w_i = 1$. The weight of class c_i is defined as:

$$W_{c_i} = w_1 index_i^1 + w_2 index_i^2 + ... + w_n index_i^n \quad (6)$$

The w_i is decided by OSM to reduce subjective judgment.

Firstly, determine scale is described by 5 levels, with higher levels indicating the higher importance. Then compare factors couple by couple, if one factor's importance level is set to 5, then the another one's importance level is 0; if one is 3, then another is 2. By this way, the judgement matrix is built which is a $n \times n$ square matrix,n is the number of factors. And the value w_{ij} (row i, column j) is the importance of factor i compare to j, such as w_{ij}=3, then opposite w_{ji}=5-3=2 which indicate the importance of factor j compare to i, $i \neq j$. The sum of rows $\sum_i w_i$ indicate the importance of factor i in all factors. Take the sum of all rows and columns $\sum_j \sum_i w_{ij}$ as the denominator which indicate the importance level of factor i in all factors, and the $\sum_i w_i$ as numerator, then the quotient is the weight of the factor i:

$$W_i = (\sum_i w_i) / (\sum_j \sum_i w_{ij}) \quad (7)$$

The hierarchical weights balance professional and user intuitive experiences by considering both of them as shown in Figure 1.

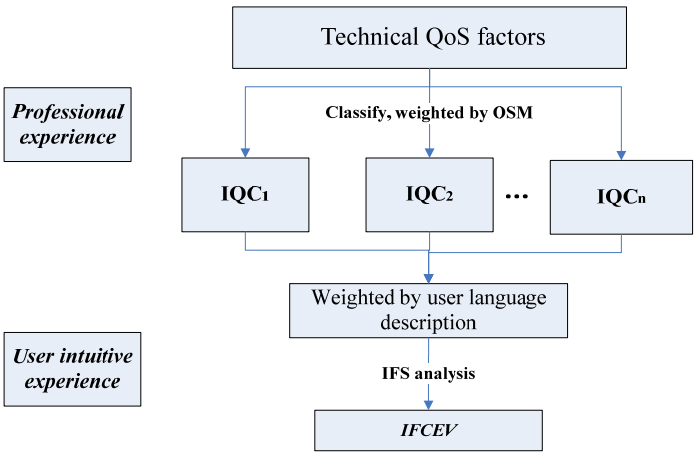

Figure 1. Balancing Professional and User Intuitive Experiences using Hierarchical Weighted IFS Analysis

2.3. IFS Based Task Scheduling with QoS Preference Awareness (IFS-QoS PA)

Based on the above definitions and methods, we proposed the IFS-QoS PA algorithm as shown in Figure 2.

Tasks= $\{t_1, t_2, \ldots, t_n\}$ is the task matrix which waiting for dispatching in unit time; Every task t_i is a task vector that has task properties, $t_i=[q_1, q_2, \ldots, q_n]$;

Links is the resource nodes matrix. if $i \neq j$, then the entry L_{ij} is a link vector between node i and node j that have link properties;if $i=j$, then the L_{ij} actually indicate the properties of node i;

Schedule Vector= $[v_1, v_2, \ldots, v_n]$ is the task scheduling vector, namely a scheduling solution. In cloud storage system, v_i represent the data of i-th task is offer by the v_i number

node. Then the length of *V* is the sum of tasks which waiting for dispatching in unit time. There is an example: one task scheduling vector is [4, 1, 3, 1, 2, 6, 5]. The length of vector is 7, so the task number is 7 that means in unit time there are 7 tasks needed to schedule. The value of sequence 1 is 4 which mean task 1's data is offered by node 4. Similarly, node 1 offers data to task 2 and 4; node 3 offers data to task 3; node 2 offers data to task 5; node 6 offers data to task 6; node 5 offers data to task 7.

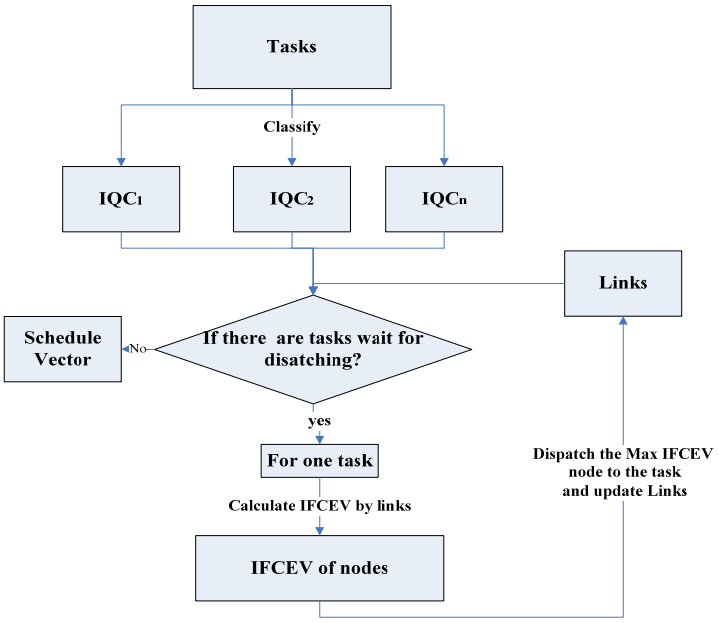

Figure 2. The Flow Chart of the IFS-QoS PA Algorithm

3. Simulation and Analysis

We developed a Cloud Storage Simulation System(CS^3) using Matlab7.0. This system includes three main modules: task scheduling module, update and the evaluation module.

The row of task matrix is the task vector which contains task size and task' QoS requirements, denoted as task (T_{size},q_1,q_2,…,q_n).The order of QoS factor as the Section 2 defined. If task does not require certain QoS factor, the according value set to be NULL. The nodes matrix contains the information of nodes, such as the node QoS similar to task vector and the nodes relation describe weather two nodes are connected or not, denoted as node vector ($node_i$-$node_j$, q_1,q_2,…,q_n),where if $i=j$, the q describe the $node_i$' QoS factor, and if $i\neq j$, the q describe the connect QoS factor between nodei and nodej, namely the links.

The system takes the task and node matrixes as the input. Then the "task scheduling module" dispatches the tasks. The update module updates the QoS factor value of the nodes matrix after one scheduling scheme is applied. Finally, the evaluation module evaluates the scheme effect by user satisfaction rate.

If a task is dispatched to a node, then we compare the task and node vectors, when all the QoS factors of the task are satisfied, the scheduling scheme is considered to satisfy the task. User satisfaction rate is defined as:

$$USR(\%) = \frac{satisfied\ task\ number}{totall\ task\ number}\% \tag{8}$$

In existing algorithm, only a few heuristic algorithms, such as PSO and GA, can offer multi-QoS constraint ability by redefined the fitness function [13, 14]. Their fitness functions include many QoS factors with different weights. The USR comparison between PSO based algorithm and our IFS based algorithm are shown in Figure 3.

We can see the USR of PSO based algorithm is obviously lower than the USR of our IFS-QoS PA algorithm. The reason is the weights used in PSO's fitness function are fixed, but the user's PA is changing. If the weights of PSO's fitness function are changing with user'sPA, does the PSO based algorithm offer satisfactory efficiency and USR?

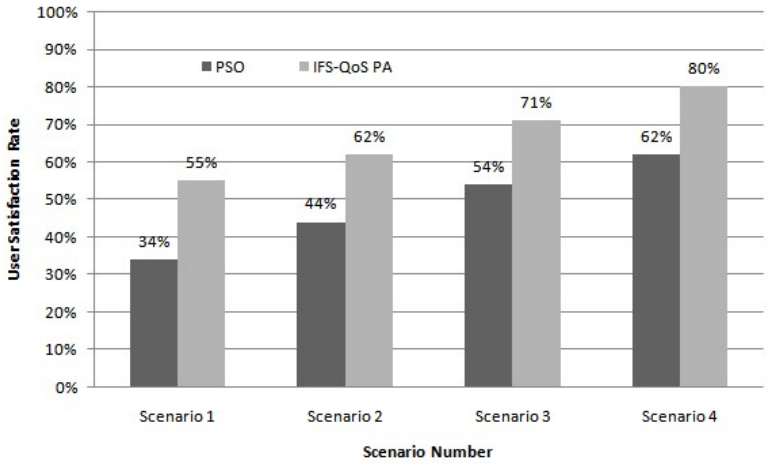

Figure 3. The USR Comparsion between the PSO and IFS-QoS PA Algorithms

We use IQC (include their weights) to redefined the fitness function of PSO to make it reflect the user's PA. And high level tasks are dispatched firstly, then the lower ones.The level of tasks is defined as the highest level 1 is the TQ tasks, level 2 is Q tasks, level 3 is T tasks and finally the lowest level 4 is the C tasks. This improved by intuitionistic fuzzy analysis PSO is called IFS-PSO, and its solution space is limited by existing matrix [6]. The USR comparison of the improved IFS-PSO and IFS-QoS PA is shown as Figure 4. In order to remove unexpected interference, we repeat the simulation 10 times and get their mean values. The simulation results show the USR of these two algorithms are almost the same. Sometimes the USR of the IFS-PSO is lower than IFS-QoS PA is because the PSO fall into the local optimal.

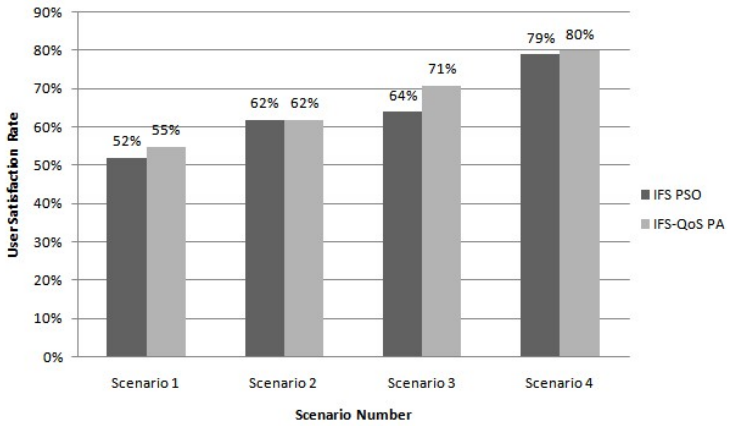

Figure 4. Comparison of the Improved IFS-PSO and IFS-QoS PA

However, the IFS PSO cost obviously more time than the IFS-QoS PA which means the PSO has lower efficiency. As shown in the Figure 5, when the task scale increase, the execution time of IFS-PSO is increase sharply. The execution time of IFS-QoS PA is linear growth and more suitable for large scale task.

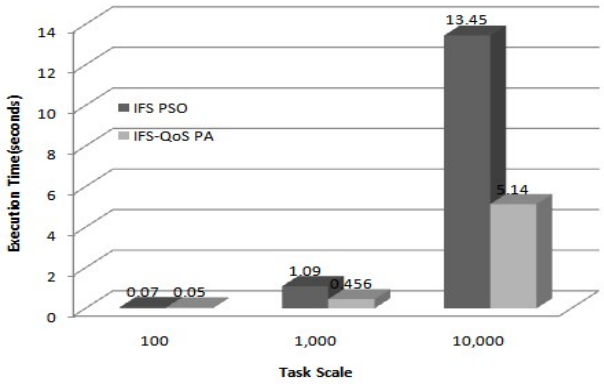

Figure 5. The Execution Time of Multi-QoS IFS and IFS-PSO

4. Conclusion

In this paper, we studied scheduling algorithms for cloud systems. The existing algorithms lack the ability to support multiple QoS factors and PA. IFS is introdued to hlep user desribe their QoS requirement, and the "IFS-QoS PA"scheduling algorithm is proposed. In addition, the OSM is used to help experts use their professional knowledge to decide the weights of QoS classes. By considering both user and expert experience, the method can determine users' PA and reflect the characteristic of cloud storage system. The simulations show this method not only offers multiple QoS constraint and satisfies user's QoS PA, but also has low computation complexity and is suitable for large scale tasks scheduling.

During the simulation, we found when the resource nodes are fixed, the task distribution is influence the USR. The relationship between task distribution and USR provides a guide to choose a fit task distribution for the system when the resource nodes cannot be increased immediately. However, the rules between the task distribution and the USR need more research in the future.

Acknowledgement

This work was supported in part by grants from:

1. The Application Basic Research Project of Sichuan Province (2013JY0064);
2. The Scientific Research Foundation of CUIT (Chengdu University of Information Technology), No.KYTZ201121);
3. The Research Fund Young of Middle-Aged Academic Leaders in CUIT(J201107);
4. The Sichuan Province Science and Technology Support Plan(No.2011GZ0195);
5. The Scientific Research Project of Education Department in Sichuan Province (No.10ZB093).

References

[1] Sandeep Tayal. Tasks Scheduling optimization for the Cloud Computing Systems. *(IJAEST) International Journal Of Advanced Engineering Sciences And Technologies*. 2011; 5(2): 111-115.
[2] Xiangqian Song, Lin Gao, Jieping Wang. Job scheduling based on ant colony optimization in cloud computing. *IEEE*. 2011.
[3] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie, Jicheng Hu. Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing. *IEEE*. 2009.
[4] Yujia Ge, Guiyi Wei. GA-Based Task Scheduler for the Cloud Computing Systems. *IEEE*. 2010.
[5] Gan Guo-ning, Huang Ting-lei, Gao Shuai. Genetic simulated annealing algorithm for task scheduling based on cloud computing environment. *IEEE*. 2010.
[6] Wang juan, Li Fei, Zhang luqiao. Task Schedule Algorithm in Cloud Storage System using PSO with Limited Solution Domain. *Application Research of Computers*. 2013; 30(1): 127-129, 154 (In Chinese).
[7] Jinquan Z, Lina N, Changjun J. *A Heuristic Scheduling Strategy for Independent Tasks on Grid*. Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA '05). 2005.

[8] Xin Lu, Zilong Gu. A load-adapative cloud resource scheduling model based on ant colony algorithm. *IEEE.* 2011.
[9] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization. *IEEE.* 2011.
[10] Zhang Bo, Gao Ji, Ai Jieqing. Cloud Loading Balance algorithm. *IEEE.* 2011.
[11] Xiao-Shan He, Xian-He Sun, QoS Guided Min-Min Heuristic for Grid Task Scheduling. *Jouranal of Computer Science & Technology*. 2003; (5): 442-451.
[12] Sameer Singh Chauhan, RC Joshi. QoS Guided Heuristic Algorithms for Grid Task Scheduling. *International Journal of Computer Applications*. 2010; 2(9).
[13] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi. A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing. *IEEE*. 2009.
[14] Frincu ME, Craciun C. Multi-objective Meta-heuristics for Scheduling Applications with High Availability Requirements and Cost Constraints in Multi-Cloud Environments. *IEEE.* 2012.
[15] Hong Sun, Shi-ping Chen, Chen Jin, Kai Guo. Research and Simulation of Task Scheduling Algorithm in Cloud Computing. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 20(11): 6664-6672
[16] ATANASSOV KT. Intuitionistic fuzzy sets. Fuzzy Sets and Systems. 1986; 20(1): 87-96.
[17] Lazim Abdullah, Hui Mei Ling.Intervalvalued Intuitionistic Fuzzy Weighted Entropy in Evaluation of Service Quality. *International Journal of Informatics and Communication Technology (IJ-ICT)*. 2013; 2(1): 17-24.