I.J. Information Technology and Computer Science, **2017, 11, 23-34**
Published Online November 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2017.11.03

Forecasting Cloudlet Development on Mobile Computing Clouds

Rashid G. Alakbarov
Institute of Information Technology of ANAS, Baku, Azerbaijan
E-mail: rashid@iit.ab.az

Fahrad H. Pashaev and Oqtay R. Alakbarov
Institute of Control Systems of ANAS, Baku, Azerbaijan
Institute of Information Technology of ANAS, Baku, Azerbaijan
E-mail: pasha.farhad@gmail.com, oqtayalakbarov@yahoo.com

Received: 22 May 2017; Accepted: 10 August 2017; Published: 08 November 2017

Abstract—Article studies development dynamics of mobile cloud technologies. Advantages of this technology and problems occurring during its use are analyzed. At the same time, issues related to meeting the demand of computing and memory resources of mobile equipment using this technology are studied. For this purpose, main monitoring parameters, especially delays in network, CPU usage percentage, memory usage, completion time of intended essential operations etc. are analyzed. Considering significance of selected parameters in Cloudlet creation, Integral monitoring parameter was established. In the article analyze technologies were given as the time series of integral parameter. Thus article studies conditions necessitating development of cloudlets on mobile computing clouds and solves issues of forecasting location time of cloudlets near certain base stations.

Index Terms—Mobile computing clouds, mobile equipment, computing and memory resources, Cloudlet, computing clouds, mobile trade, mobile healthcare, cloud services, forecasting.

I. Introduction

Currently, intensive works are carried out world-wide to efficiently use computing and memory resources of data processing centers using Cloud Computing technology. Such systems with large computing and memory resources are developed based on computer networks with high speed connection channel. Cloud Computing technology allows using the computing and memory resources of data processing center's more efficiently. Using this technology, user information is stored, processed on computing clouds' servers and at the same time, results can be reviewed through browsers [1, 2]. Cloud computing services provide wide use of clustering and virtualization of computing and memory resources of processing centers. Hence, Cloud Computing system is an Internet service consisting of technical-software and allowing internet users to use remote computer resources (computing and memory resources, program and data etc.) through an appropriate web-interface.

Cloud computing technology allows users to acquire strong computing and memory resources and at the same time, user is not interested in location and settings of these resources. Using mentioned service, it is possible to provide efficient use of computing and memory resources at data centers, reduction of problem solution time and less loading of the network [3]. Cloud technologies optimally distribute the existing memory resources of the processing center among users, thus attracting more users to the system and obtaining the results faster. Recently, mobile users have started widely using the services of Cloud Computing technologies.

Rapid acceleration of world-wide use of mobile devices (laptop, tablet, smartphones etc.) and their connection to computing clouds via relevant telecommunication technologies (GPS, 3G, 4G, Wi-Fi etc.) lead to development of new technologies – Mobile Cloud Computing technologies [4]. The article studies issues related to more efficient use of computing and memory resources of mobile computing clouds widely used recently. It is known that, capabilities of any mobile device (computing and memory resources) are limited. But users use these devices to solve issues that require large computing and memory resources. Computing cloud technologies are widely used for this purpose. Hence, computing and memory resource deficiencies existing in users' phones can be solved using cloud technologies.

II. Architecture of Mobile Computing Cloud Technologies and Problems

Mobile computing clouds are the new platform formed on equipment and computing clouds allowing mobile users to solve complex issues and store large-volume data [5].

Technical capabilities (computing and memory resources) of mobile devices are limited. Cloud computing capabilities are widely used in order to eliminate this limitation. Mobile users solve any issues use computing cloud services. Recent reduction in the cost of cloud servers enables mobile users to use these services more widely [6]. Currently, many companies develop multiple applications (Google, Gmail, Maps and Navigation systems for Mobile, Voice Search, Mobil Me from Apple, Live Mesh from Microsoft) for mobile equipment users, which enables them to widely use mobile computing clouds [7]. According to the calculations of the Juniper Research centers' analysts, development speed on computing clouds increases at 88% due to software and applications.

In mobile computing clouds, processing and memorization of information in carried out outside of the mobile devices. Mobile devices connect to internet networks through base stations (GPS, 3G, 4G, and Wi-Fi etc.) and use necessary services in mobile computing clouds. Currently, users widely use three types of cloud services (IaaS, PaaS and SaaS) [8]. IaaS (Infrastructure as a Service) service allows users to use computing and memory resources (for example, Amazon Elastic Cloud Computing - EC2 and Amazon S3 - Simple Storage Service) of the cloud system. PaaS (Platform as a Service) service is a platform allowing users to use operation systems and special applications (Google App Engine, Microsoft Azure, Azache, My SQL etc.) located on virtual servers. SaaS (Software as a Service) service allows users to solve their issues using software located in the computing cloud servers (Google Apps, Google Docs, Autodesk) and applications. In the SaaS service, the user can obtain results by launching the software from computing cloud servers using internet service, without loading the resident portion of the necessary software to his/her own computer. Applications operate on servers provided by SaaS service provider and send the processed information to the user. Thus, user does not purchase the application and pays according to the use. Currently, millions of mobile users are widely using mobile applications (mobile commerce, mobile education, mobile health, mobile games etc. fields) using mobile cloud provider services [5, 9]. Developed mobile applications do not depend on the type of operation systems and device type of the mobile devices. For this reason, number of mobile device users that use cloud services increases rapidly every day. In this case, mobile devices with minimal computing and memory resources act as thin client terminal connected to Internet network.

According to the calculations of the Gartner company analysts, number of Internet users will reach 2.16 billion in 2016. This number will rise to 2.56 billion in 2018.

Currently, despite centralized clouds have high computing and memory resources, processed information cannot be delivered to the users on a high speed. Rapid increase of number of users on computing clouds loads the network which causes significant delays in delivery of processed information to the user. Cloud computing resources have to be located closer to the users in order to eliminate this deficiency.

Computing cloud service users use two modes: offline and online. Let's assume that virtual computing devices are acquired from the cloud for solution of issues that require vast computing power. He/she (One) sends the issues to the cloud and obtains the result after a certain period of time. In this mode, there is no direct connection between the user's computer and the cloud during the period necessary for the solution of the problem. But there are such issues, where user and cloud are connected until the problem solution process is complete. This, leads to loading of the network during solution of the mentioned problem. Mobile computing cloud technologies are used to eliminate the network load. Sometimes it is also referred to as decentralized cloud technologies in literature. Architecture of mobile computing clouds is shown (Fig. 1).

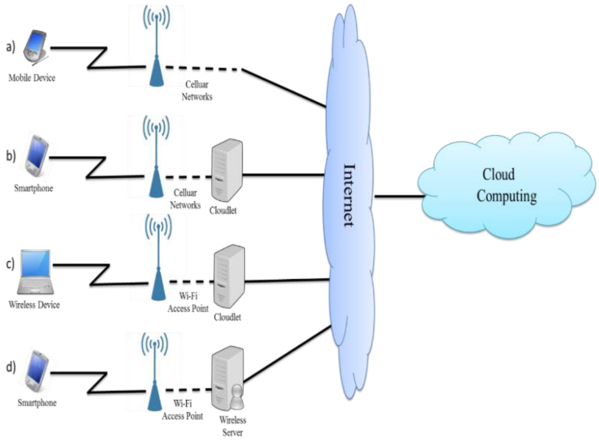

Fig.1. Architecture of Mobile computing clouds.

As seen on mobile computing cloud architecture, system consists of several components: mobile users (mobile device, smartphone etc.), mobile connection operators, wireless connection devices (Wi-Fi access point), internet provider (ISP), and computing cloud providers (Amazon IBM, Google, Microsoft etc.). as seen on the diagram, mobile users connect to Cloud Computing over internet using base stations (Mobile connection, Wi-Fi Access Point) such connections cause high load of the networks and delays in provision of processed information. At the same, these services are expensive. Cloudlets (small clouds) are developed near base stations (cellular networks, Wi-Fi Access Point) using Cloud Computing connection options b, c, and d by the users in order to eliminate indicated deficiencies [5, 10].

Thus, mobile servers (Cloudlets) beside base stations of mobile operators are developed for vaster use of cloud technologies by mobile equipment. Cloudlets (small computing clouds) are devices (server) located close to the users and provide faster delivery of necessary information to the users from the central servers. After the user completes his/her work, if necessary, information is re-loaded to the central servers. Cloud services necessary for the users can be implemented through Cloudlet servers [10-12]. Let's assume that, if a user needs to work with a software (SaaS service), he calls that software to the closest mobile service, works on it for a while and then, sends that software to the central server. This, reduces the use cost of the clouds, reduces the issue solution period and loads the network less.

Advantages of mobile computing clouds are listed below [10 - 12]:

- increasing battery life of mobile devices;
- increasing computing and memory resources of mobile devices;
- storage of user data on several back-up computers increases reliability, which in its turn reduces information loss risk on mobile devices;
- Dynamic distribution of resources. This provides timely provision of necessary resources to the user without prior ordering;
- increases network bandwidth;
- Service charge costs less;
- organizing the storage of user data on clouds enables to use this information from any place worldwide;
- increased scaling. System meets the computing and storage demands of the user for a short-term due to flexible resource capability;
- Security capabilities of the system. When mobile users use computer services they encounter problems related to storage and protection of information on clouds. Despite service providers protect the user information through different security software, at the same time they can observe that information. For this reason, confidentiality problem is very relevant. Security applications developed for mobile devices are used in order to solve these problems and prevent illegal interferences. These pieces of security software use a certain part of computing and memory resources of mobile devices. For this reason, it would be reasonable to transfer threat detection software to cloud servers from mobile devices. Cloud AV platform has been developed for this purpose; it provides multiple services detecting malware on clouds and mobile devices alike.
- Easy integration. Meets user demands by integrating services of different provides through clouds and Internet.

Scope of technical capabilities (number of virtual machines, storage space, network bandwidth etc.) of servers used for development of small clouds near base stations is among important issues. At the same time, provision of connection among developed small clouds is a relevant issue. At the same time, provision of connection among developed small clouds is a relevant issue.

On one hand users require high computing resources, and on the other, they wish to use small mobile devices. Currently, cloud technologies are used in order to eliminate the mentioned deficiency of the mobile devices (computing and memory resource deficiency).

On the other hand, use of the traditionally centralized cloud services can meet the demand for high computing and memory resources. But results or data are obtained with delay in this type of cloud services. For example, Google has developed an online translation service (Google Translate API), which enables users that speak different languages to speak to each other. Mobile devices used for this service (smartphones) do not translate independently and send the words and sentences to Google Cloud platform servers. Translation is carried out on the servers and results are sent to people talking to each other. We feel that the service is operating slowly in such type of services. While translation is carried out within seconds, its delivery to necessary destination over the network is delayed. Thus, it would be reasonable if the translation software was located on local servers (cloudlet) close to its users and rapid implementation of translation in real-time mode was enabled. At the same time, location of the software on closest cloud servers provides cheap, fast and quality service to the users using the SaaS service.

Users are guided by following criteria while using cloud technologies:

- expenses required for problem solution;
- problem solution duration;
- provision of security of user information;
- Fast and reliable delivery of data and the results to the users.

Indicated criteria depend on the geographical distance of the user from the cloud server, computing power of the virtual machine and network load level. For this reason, it is attempted to place mobile cloud technology servers near the users. It would be reasonable to create Cloudlets near each base station.

Article studies conditions necessitating development of mobile computing clouds (cloudlets) and solves issues of forecasting location time of Cloudlets near certain base stations.

III. Problem Statement

Conducted monitorings have demonstrated that, the frequency of use of newly launched games by users is high. Hence, it is purposeful to store such games in Cloudlets resources for a particular period. In this case, Cloudlets resources are spent on storing the extensions of this type of applications. Thus, the resources of Cloudlets mentioned are not sufficient to serve to new mobile users. For this reason, the establishment of network infrastructure of Cloudlets located in the vicinity of Wireless Metropolitan Area Networks (WMAN) is among the topical issues. It is more desirable to create cloudlets in the vicinity of all access points of the mentioned network. This is not economically feasible and rises the cost of established networks. On the other hand, it is possible that, cloudlets located in the vicinity of any access point are poorly used or not used at all. Hence, cloudlets are mostly located in the vicinity of locations with more mobile users (shopping malls, libraries, schools, universities, stadiums, stations, airports, etc.) [13].

As mentioned above, the number of mobile users and game users in particular utilizing cloud technologies is rapidly growing. It is to be noted that, the interests for intellectual games is related to the intellectual need of users and serves to improving their intellectual capacity [14]. The works dedicated to the creation of such games has been carried out extensively in recent times [15, 16]. Similar to software tools (translation, navigation software, etc.) utilized in online regime, games require high-speed interface as well. The solution of such problems requires the use of cloudlets. The majority of recently conducted studies has been oriented to the addressing of problems of cloudlets use. The sequence of cloudlet computing has been determined schematically. Important features of cloud computing have been identifed. These features include [17]:

- *The on-demand self-service;*
- *The broad network access;*
- *The resource pooling;*
- *The rapid elasticity;*
- *The measure service*

The life cycle of cloudlets has been analyzed and some necessary digital characteristics have been explored [18, 19]. While working with interactively used software tools in On_Line regime, the issues of constructing time diagrams of the implementation of requests have been addresses [20]. Some authors have reviewed the issues of service migration in mobile clouds and some advantages of this method have been indicated [21]. In order to carry out this process, a simple algorithm can be generated by considering the frequency of requests to software tools [22]. Each software tool stored in this way can be deleted from memory resources of a cloudled whenever there is a need for storing higher priority software tool.

It is known that establishment of Cloudlets near base stations (cellular networks, Wi-Fi access point) in the architecture of the mobile computing clouds depicted in Figure 2 is related to increasing demand of the mobile users for computing resources [23].

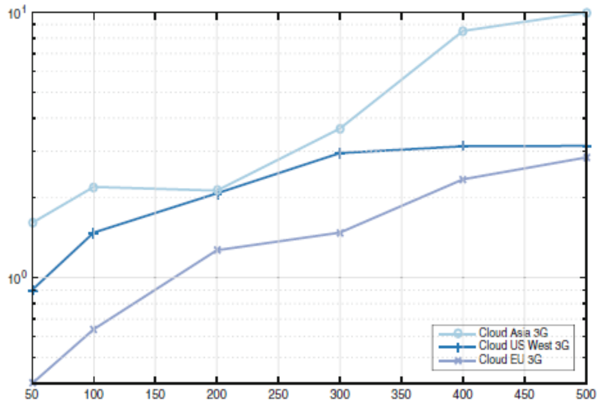

Fig.2. Comparative diagram of dependence of network delays of traffic volume on Amazon clouds

Amol C. Adamuthe, Vikram D.Salunkhe, Seema H. Patil and Gopakumaran T. Thampi have mentioned the rapid growth of the number of published articles as a result of studies carried out in Cloud Computing field and the rapid expansion of investments devoted to such research works, financial resources, the number of Cloud service providers and etc. during the period from 2006 to 2014. Here, it is also mentioned that, the govenments of US, UK, Japan and Australia promote the conduction and implementation of works in the field of Cloud Computing [24].

In order to determine Cloudlet establishment necessity, traffic and separate parameters in base stations must be monitored. Following are among monitoring parameters used for analysis:

- Network delays. Delays of network request responses, cloud software responses etc. must be measured. Passage of a certain mark by the delay renders the use of the system impossible. This is the main parameter for assessment of the used network in systems created using global clouds and implemented operations. For this reason, this parameter becomes an object of study as it changes time-wise and based on number of users. For example, studies conducted at the Telecooperation lab of Technische Universität Darmstadt demonstrate that the marks of this parameter are continuously growing. (Fig. 2). Here, dependence of network delays on traffic volume on Cloud Asia, Cloud US West and Cloud EU is demonstrated. İn this diagram, traffic volume is presented as Kb/sec on X axis. On Y axis, we can see that traffic delay time increases in multi-fold depending of traffic volume marks [25]. Traffic volume increases as the number of users increase and as the users use more complex software pieces;
- Using computing resources. Computing force of processors used during development of cloudlets must be higher than the computing force required by the users;
- Using memory resources. Memory resource requirements can limit Cloudlet development. Due to this requirement, memory size of the server located in the Cloudlet must be higher than the computing power required by the users.
- Completion time of considered necessary operation. This time is usually related to network delays between the base station and global cloud. Growth of network delays increases completion time of the operations;
- Operation completion energy. This can be defined as energy used by all equipment in unit time considering all operations of the users connected to the base station. Amount of energy spent in unit time can reach the limit threshold at a certain number and condition of mobile users. After reaching this threshold, energy spent on operations does not increase despite network delays.

- Operation price. Here, price for each operation and all operations conducted by the users connected to the base stations can be monitored. This parameter increases with network delays.

In order to provide monitoring and analysis results' adequacy, mentioned parameters (diversity and usage frequency of used application programs, reliability and security of communication channels, etc.) must be calculated and analyzed with equal time intervals. Article aims to solve issues such as creation of a final integral indicator based on parameters calculated at equal time intervals, smoothing the values on this indicator with wavelets transforms and forecasting future value of the integral indicator applying Neuron networks. Cloudlet development can be decided by analyzing approaching limit time.

IV. Forecasting Cloudlet Development Time

Let's assume that, during a certain T period M number of values are obtained by monitoring the values of J number of parameters. If we mark parameters as $P_1, P_2,, P_3, \dots,, P_J$ we will get this matrix.

$$\begin{pmatrix} P_{11} & \dots & P_{1M} \\ \cdot & \cdot & \cdot \\ P_{J1} & \dots & P_{JM} \end{pmatrix} \tag{1}$$

In some cases, normalizing parameter values and writing their values as untitled values with maximum level of 100 can be useful. Thus, value of each parameter is written as the percentage of its own maximum value. If we denote the corresponding maximum value of each j$^{\text{th}}$ parameter as P_{jmax}, the following can be written:

$$P_{jm} = P_{jm} * \frac{100}{P_{jmax}}$$

Thus, the values of (1) matrix will be normalized to 100 as maximum value.

The importance or weight of each parameter in Cloudlet development can be determined by experiments. Those weights can be written as $(w_1, w_2, \dots, w_J)$ vector, which satisfies the following condition:

$$\sum_{j=1}^{J} w_j = 1 \tag{2}$$

If we multiply (2) weight vector by (1) matrix by using the rule of multiplication of matrices, the values of integral parameters can be obtained [26]:

$$(I_1, I_2, \dots, I_M) = (w_1, w_2, \dots, w_J) * \begin{pmatrix} P_{11} & \dots & P_{1M} \\ \cdot & \cdot & \cdot \\ P_{J1} & \dots & P_{JM} \end{pmatrix} \tag{3}$$

Here, I_m is calculated as $I_m = \sum_{1=1}^{J} w_i * P_{mi}$.

The values of $(I_1, I_2, \dots, I_M)$ vector obtained for integral parameter can be desctibed as $I_1, I_2, \dots, I_M$ time series and used for predicting the time of Cloudlet development.

Wavelet conversions can be carried out in order to smooth obtained time series.

Theoretically the discrete wavelet transform is defined as:

$$d_{j,n} = \langle x(t), \varphi_{j,n}(t) \rangle = \int_{-\infty}^{+\infty} x(t)\, \varphi_{j,n}(t)\, dt \tag{4}$$

Where the mother wavelet function $\varphi_{j,n}(t)$ is defined as:

$$\varphi_{j,n}(t) = a_0^{-j/2} \varphi_{j,n}(a_0^{-j} t - nb_0) \tag{5}$$

The scaling a_0 and translation b_0 parameters are specifically chosen such that $\varphi_{j,n}(t)$ constitute orthonormal bases with $a_0 = 2$ and $b_0 = 1$ [27-29]. So we have:

$$\varphi_{j,n}(t) = 2^{-j/2} \varphi_{j,n}(2^{-j} t - n) \tag{6}$$

Using equation (1), the orthonormal wavelet transform is thus given by:

$$d_{j,n} = \langle x(t), \varphi_{j,n}(t) \rangle = 2^{-j/2} \int_{-\infty}^{+\infty} x(t)\, \varphi_{j,n}(2^{-j} t - n)\, dt \tag{7}$$

The inverse discrete wavelet transform is given by:

$$x(t) = \sum_{j=1}^{J} \sum_{n=1}^{N} d_{j,n} \varphi_{j,n}(t) \tag{8}$$

The following threshold can be used:

$$d'_{j,n} = \begin{cases} \lambda - d_{j,n} & |d_{j,n}| \geq \lambda,\ d_{j,n} \geq 0 \\ d_{j,n} - \lambda & otherwise \end{cases} \tag{9}$$

Where λ is defined as follows:

$$\lambda = \frac{\sum_{j=1}^{J} \sum_{n=1}^{N} d_{j,n}}{J \times N}$$

After inverse discrete wavelet transform we will have:

$$x'(t) = \sum_{j=1}^{J} \sum_{n=1}^{N} d'_{j,n} \varphi_{j,n}(t) \qquad (10)$$

Various algorithms can be applied for wavelet conversion of calculated discrete values of integral parameter.

If we use D4 wavelet created by Ingrid Daubechies for this purpose, those algorithms can be applied for N number of input values in form of N=2k in practice. Input values are processed via low pass and high pass filtering functions.

In some cases, low pass and high pass filters are called high-cut and low-cut filters, respectively. In most cases, high-cut or low pass filter contains more information on initial input values. In contrast, high pass filter shows the difference between input values and versions of values repaired from low-cut filters. If the mean value of low-cut filter is sufficiently low, it can be disregarded in repairing process and it would not deteriorate the quality of repairing process.

It is known that, low pass coefficients in D4 wavelet are specified as

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}},\ h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}},$$

$$h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}},\ h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}.$$

High pass coefficients in D4 wavelet are specified as $g_k = (-1)^k h_{2N+1-k}$. Hence:

$$g_0 = (-1)^0 h_{3-0} = h_3,$$

$$g_1 = (-1)^1 h_{3-1} = -h_2,$$

$$g_2 = (-1)^2 h_{3-2} = h_1,$$

$$g_3 = (-1)^3 h_{3-3} = -h_0.$$

It can be concluded that, the following can be specified:

$$g_0 = \frac{1-\sqrt{3}}{4\sqrt{2}},\ g_1 = -\frac{3-\sqrt{3}}{4\sqrt{2}},$$

$$g_2 = \frac{3+\sqrt{3}}{4\sqrt{2}},\ g_3 = -\frac{1+\sqrt{3}}{4\sqrt{2}}$$

The conversion can be enhanced by applying repeated conversions to low pass filters.

If input values are fj-lar (j=0-2m), the repair of values can be carried out with following simple algorithms:

$$a_{1,i} = h_0 f_{2i} + h_1 f_{2i+1} + h_2 f_{2i+2} + h_3 f_{2i+3};\ i = 0 \div \frac{N}{2} - 1 \qquad (11)$$

$$a_{2,i} = h_0 a_{1,2i} + h_1 a_{1,2i+1} + h_2 a_{1,2i+2} + h_3 a_{1,2i+3};\ i = 0 \div \frac{N}{2^2} - 1 \qquad (12)$$

$$a_{3,i} = h_0 a_{2,2i} + h_1 a_{2,2i+1} + h_2 a_{2,2i+2} + h_3 a_{2,2i+3};\ i = 0 \div \frac{N}{2^3} - 1 \qquad (13)$$

$$a_{k,i} = h_0 a_{k-1,2i} + h_1 a_{k-1,2i+1} + h_2 a_{k-1,2i+2} + h_3 a_{k-1,2i+3};\ i = 0 \div \frac{N}{2^k} - 1 \qquad (14)$$

At each level, high pass filter will be employed as following:

$$c_{k,i} = g_0 a_{k-1,2i} + g_1 a_{k-1,2i+1} + g_2 a_{k-1,2i+2} + g_3 a_{k-1,2i+3};\ i = 0 \div \frac{N}{2^k} - 1 \qquad (15)$$

Here, $a_{0,2i+j} = f_{2i+j};\ j = 0 \div 3$

After conversions, the values can be successively repaired with following algorithms:

$$a_{k-1,0} = h_2 a_{k,0} + g_2 c_{k,0} + h_0 a_{k,1} + g_0 c_{k,1} \qquad (16)$$

$$a_{k-1,1} = h_3 a_{k,0} + g_3 c_{k,0} + h_1 a_{k,1} + g_1 c_{k,1} \qquad (17)$$

$$a_{k-1,2i} = h_2 a_{k,i} + g_2 c_{k,i} + h_0 a_{k,i+1} + g_0 c_{k,i+1} \qquad (18)$$

$$a_{k-1,2i+1} = h_3 a_{k,i} + g_3 c_{k,i} + h_1 a_{k,i+1} + g_1 c_{k,i+1} \qquad (19)$$

The last step will be as $a_{0,2i+j} = f_{2i+j};\ j = 0 \div 1$ and the values are repaired. While the values are repaired from $a_{k,i}$ and $c_{k,i}$, the values of those below specific boundary level can be replaced with zero. In this case, initial values will be replaced with smoothed values. It is clear that, the maximum depth of conversions can be obtained when the number of $a_{k,i}$ and $c_{k,i}$ is equal to 4. Such that, if the number of input values is 128, maximum 4 conversions can be carried out. Hence, if we denote boundary values of $a_{k,i}$-s as λ_a and $c_{k,i}$-s as λ_c, we can determine those as:

$$a^1_{k,i} = \begin{cases} a_{k,i}, abs(a_{k,i}) > \lambda_a \\ 0, abs(a_{k,i}) <= \lambda_a \end{cases} \text{ and} \qquad (20)$$

$$c^{1}_{k,i} = \begin{cases} c_{k,i}, abs(c_{k,i}) > \lambda_c \\ 0, abs(c_{k,i}) <= \lambda_c \end{cases} \qquad (21)$$

and use $a^1_{k,i}$ and $c^1_{k,i}$ during the repair of values.

In order to employ (11)-(21) algorithms, those last values of initial signal must be selected, the number of which can be expressed as a sufficiently large number of power of 2.

If initial number of monitoring values are equal to M, the value of N=2^k must be found which is 2^k<M<2^{k+1}. Hence, if initial values are equal to $I_i, i = 1 \div M$ and the signals selected for conversion and smoothing is denoted as f_j, then $f_j = I_{M-N+j}$.

Gaps (lost values) in f_j must be detected and filled with random values for conversion. If number of gaps exceeds 1 percent of total number, the selection must be discarded and the algorithm functioning must be terminated with corresponding warning. The presence of lost values can be detected with following simple algorithm. It is known that, the values of Integral parameter are noted with own time. For each value of f_j, T_j time corresponds. Hence,

$\Delta t = \dfrac{T_N - T_1}{N - 1}$ must be determined.

If for each j, $\dfrac{T_{j+1} - T_j}{\Delta t}$ >=1.5, the number of missing values will correspond to the number of integral parts of $\dfrac{T_{j+1} - T_j}{\Delta t}$ parameter between jth and j+1th values.

So, the following common algorithm can be proposed for smoothing the values of Integral parameter:

a. Select N=2^k number of input data and generate f_j input values;
b. Specify the location and number of missing values;
c. Insert random values at first approach instead of low number of missing values;
d. Determine the values with N+1=2^k+1 and N+2=2^k+2 indices;
e. Determine D4 coefficients;
f. Carry out conversion with (11)-(15) formulas;
g. Determine λ_a and λ_c boundary values;
h. Calculate new repairing values with (20)-(21) formulas;
i. Repair noise characteristics with (16)-(19) formulas;
j. If the outcome is not satisfactory, define λ_a and λ_c boundary values once again and repeat the algorithm starting from hth step.

The analysis of the change of integral parameter generated as a result of monitoring of base stations with time enables to predict the approaching time of Cloudlet development.

In order to solve the prediction problem, following methods can be used:

- Autoregressive or moving average models such as ARMA, ARIMA, GARCH, etc.
- Exponential moving average method;
- Frequency distribution of time series elements and construction of distribution functions;
- Application of neural networks.

If a simple autoregressive model is applied, the following can be written for predicting the future value x^p_{t+1} of series:

$$x^p_{t+1} = \sum_{j=1}^{n} \omega_j x_{t-j+1} \qquad (22)$$

Here, x_{t-j+1} values are row vectors of

$$F = \begin{pmatrix} x_{t-1}, x_{t-2,}, x_{t-3}, \ldots, x_{t-n} \\ x_{t-2}, x_{t-3}, x_{t-4}, \ldots, x_{t-n-1} \\ \ldots \\ x_n, x_{n-1}, x_{n-2}, \ldots, x_1 \\ x_{n-1}, x_{n-2}, x_{n-3}, \ldots, x_0 \end{pmatrix} \qquad (23)$$

matrix.

ω_j - weight coefficients, satisfying $\omega_j \geq 0$ and $\sum_{j=1}^{n} \omega_j = 1$ conditions.

The quality of predicting is dependent on the values of ω_j weight coefficients. Hence, ω_j weight coefficients are found satisfying

$$Q_t(\omega, x^l) = \sum_{i=n}^{t} (x^p_i(\omega) - x_i)^2 = \left\| F\omega - X \right\|^2 \to \min$$

by using column vector:

$$X = \begin{pmatrix} x_t \\ x_{t-1} \\ \ldots \\ x_{n+1} \\ x_n \end{pmatrix} \qquad (24)$$

(22) formula provides the predicted values of series of integral parameters as a result of the application of the algorithm. This value shows the necessity of creating a cloudlet.

Exponential smoothing method is used in some cases.

$$x_{t+1}^{p} = x_{t}^{p} + \alpha(x_t - x_t^p) = \alpha x_t + (1-\alpha)x_t^p \qquad (25)$$

At first step,

$x_1^p = x_1$ is taken.

This method can be used for smoothing, as well as for predicting.

Here, α parameter takes values of $\alpha \in (0,1)$. As the values of α approaches to 1, the last values of generated time series become more important and their impact on the predicted value is stronger. As the value of α is lower, the impact power of previous values on selected series with time increases.

Let's generate a massive consisting of N number of last values of generated time series in order to construct the experimental distribution. Let's mark maximum and minimum values of massive elements as $x_{\max}$ and $x_{\min}$, respectively. Firstly, $[x_{\min}; x_{\max}]$ range must be divided into several equal parts. The number of parts must be taken as equal to K experimentally and marked as m_K. The width of each part is:

$$\Delta h = \frac{x_{\max} - x_{\min}}{m\_K} \qquad (26)$$

$$x_{\max} = m_{\min} + m\_K * \Delta h \quad \text{or}$$
$$x_N = x_1 + m\_K * \Delta h. \qquad (27)$$

Accordingly, in order to store the values of distribution function, $d\_m[i\_i]$ massive must be created given that, $i\_i \in [1, m\_K]$ and its elements must be equalized to zero in advance.

Hence, it is possible to find $m \in [1, m\_K]$ for any $n \in [1, N]$, which can be expressed as:

$$x_n \in [x_1 + (m-1) * \Delta h, x_1 + m * \Delta h] \qquad (28)$$

It is to be noted that, as $x_n \in [x_1, x_N]$, m value satisfying this conditions always exists.

When any m value is found satisfying the above condition for any $n \in [1, N]$, the exponential histogram values of integral parameter values are found as:

$$d\_m[m] = d\_m[m] + 1$$

For any $m \in [1, m\_K]$ values,

$$d\_m[m] = \frac{d\_m[m]}{\sum_{n=1}^{N} x_n} \qquad (29)$$

expression would produce the values of frequency distribution function. Should the graph of experimental frequency distribution function be similar to any graph of known distribution functions, parameters of known expression of frequency function can be determined from experimental values. The algorithm given here can be used for practical issues.

Following characteristics are satisfied here:

$$0 \le d\_m[m] \le 1 \text{ and}$$

$$\sum_{m=1}^{m\_K} d\_m[m] = 1.$$

Hence, $d\_m[m]$ value can be taken as a value of distribution frequency for m^th

$$[x_1 + (m-1) * \Delta h, x_1 + m * \Delta h] \text{ range.}$$

The value of distribution function can be determined as

$$F(m) = \sum_{i=1}^{m} d\_m[i]. \qquad (30)$$

In our case, it is an expression of the distribution function obtained in discrete form. Hence, the probability of predicted time series value to fall within $[x_1 + (m_1 - 1) * \Delta h, x_1 + m_2 * \Delta h]$ interval can be expressed as:

$$p[m_1, m_2] = \sum_{i=m_1}^{m_2} d\_m[i] \qquad (31)$$

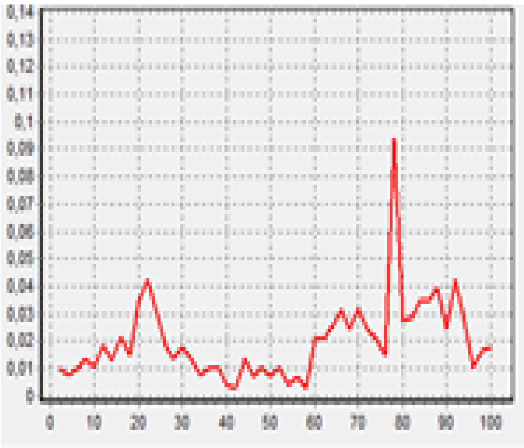

Fig.3. Frequency distribution function obtained during computation experiments

The graph of frequency distribution function constructed during the computation experiments conducted in the campus of Azerbaijan National Academy of Sciences is given in Fig. 3.

As seen from the figure, the predicted value can take values corresponding to the values of m variable above 60. Here, it is taken as $80 \le m \le 90$, which can be $x^p \in [x_1 + 80 * \Delta h, x_1 + 90 * \Delta h]$.

In order to calculate the values of experimental frequency distribution function, HELPI programming fragments can be used as below:

```
for  i_i:=1  to  100  do    d_m[i_i]:=0;
    max:=x[1];   min:=x[1];
  for  i_i:=1  to  n  do  begin
              if  max<x[i_i]  then  max:=x[i_i];
              if min > x[i_i] then min:=x[i_i];
                    end;
    d_h:=(max-min)/50;
   d_g:=max-min;
    if d_h=0 then d_h:=1;
for  i_i:=1  to  say  do  begin
        for  j_j:=1  to  m_K  do  begin
           if  (y1[i_i]-min>=(j_j-1)*d_h )  and  (y1[i_i]-
min<(j_j)*d_h )
           then    begin
            d_m[j_j]:=pay_m[j_j]+1;
           end;  end;    end;
  max_p:=d_m[1];
  sum_p:=0;
  for  i_i:=1  to  m_K  do  begin
 if  max_p<pay_m[i_i]  then  max_p:=pay_m[i_i];
                sum_p:=sum_p+ d_m[I_I];
                    end;
  for  i_i:=1  to  m_K  do  begin
 if  max_p<pay_m[i_i]  then  max_p:=pay_m[i_i];
 d_m[I_I]:= d_m[I_I]/  sum_p;
                    end;
```

The application of neural networks can produce more accurate and adequate results.

For this purpose, neural networks with one hidden layer can be employed (Fig. 4) [15, 16]. In this case, N1 number of inputs of neural network are smoothed and specified from repaired $I_1, I_2, \dots, I_M$ values. To study neural network,

$$\{I_{M-(N1+k-1)}, I_{M-(N1+k-1)+1}, \dots, I_{M-(N1+k-1)+(N1-1)}\}$$

can be defined as k number of input values and $I_{M-(k-1)}$ as output values corresponding to each input set.

Collections from input values set are provided to neural network inputs alternately and the process is carried out by correcting the network weights until the difference between obtained results and output values are satisfactory.

The common gradient in network is calculated as the sum of all gradients related to set generated for studying the network. Denote the weight for transmission from j^th neuron of prime layer to i^th neuron of subsequent layer as w_ij. In this case, the gradient is calculated as below:

$$\Delta\omega_{ij} = -\frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial \omega_{ij}}, \text{ here } net_i = \sum_{r=1}^{n} \omega_{ri} x_r \quad (32)$$

Here, first multiplier is the difference occurring in i^th neuron. Second multiplier can be expressed as below:

$$\frac{\partial net_i}{\partial \omega_{ij}} = \frac{\partial}{\partial \omega_{ij}} \sum_k \omega_{ik} y_k = y_j \quad (33)$$

We get from (15) and (16):

$$\Delta\omega_{ij} = \delta_i y_j .$$

That is, to calculate the desired gradient, the activity and differences in network nodes must be calculated. The activity can be calculated as following:

$$y_i = f_i\left(\sum_j \omega_{ij} y_j\right) \quad (34)$$

tanh can be used as activation function. Thus, neural activity will be expressed as $y(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. One of the advantages of this function is its derivation simplicity. $(y(x))' = 1 - (y(x))^2$.

The difference between output values and estimated outputs is calculated as

$$E = \frac{1}{2}\sum_{\circ}(t_{\circ} - y_{\circ})^2 .$$

In this case, the difference in network output is $\delta_{\circ} = t_{\circ} - y_{\circ}$.

The differences emerging in neurons in hidden layers can be estimated as $\delta_j = -\sum_{i \in P_j} \frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial y_j}\frac{\partial y_j}{\partial net_j}$. Here, first multiplier is the difference occurred in ith neuron. Second multiplier is the weight of transition from jth neuron to ith neuron:

$$\frac{\partial net_i}{\partial y_j} = \frac{\partial}{\partial y_j}\sum_k \omega_{ik} y_k = \omega_{ij}$$

Third multiplier is the derivation of activation function for j^th neuron:

$$\frac{\partial y_j}{\partial net_j} = \frac{\partial f_j(net_j)}{\partial net_j} = f_j'(net_j)$$

If we use the derivation of hyperbolic tangent, the following can be written for hidden neurons:

$$f_h'(net_h)=1-y_h^2$$

Considering all above-mentioned:

$$\delta_j = f_j'(net_j)\sum_i \delta_i \omega_{ij}$$

Hence, neural network calculates the weights again:

$$\omega_{ij} = \omega_{ij} + \Delta\omega_{ij}$$

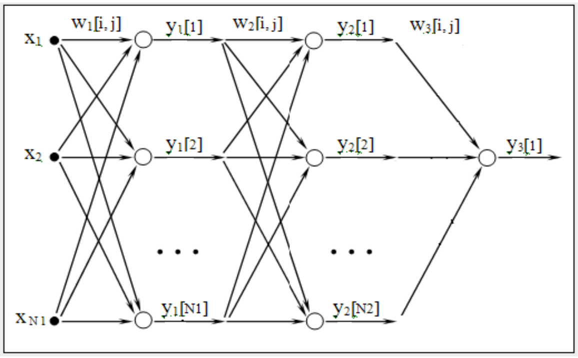

Fig.4. Neural network for predicting time series.

The process continues until output values are sufficiently close to designated values. After neuron is taught, (M+1)-th value of integral parameter (I_{M+1}) is calculated with formula (17) by assigning N1 number of values to network inputs from the end.

If estimated value is greater than specified boundary value (I_{lim}) of parameter, Cloudlet development time can be predicted:

$$I_{M+1} \geq I_{lim}$$

V. Conclusion

The article explores specific parameters necessary for Cloudlet development in Base stations, which establishes connections between mobile users and Mobile calculation clouds. Integral parameters were generated to exhibit the importance of Cloudlet development but using weight vector; the latter shows the importance of investigated parameters. The values of integral parameter are considered as time series and smoothed by applying wavelet conversions, and some missing values were repaired.

The article has demonstrated that various methods can be used for predicting future values of generated time series. Wavelet conversions and exponential smoothing methods among these methods are presented for the smoothing of time series values. Autoregressive method, exponential distribution frequency and the construction of its function, as well as the application of Neural network is presented for calculating the predicted values of series. It was proposed to use the generation of several educative collections from primary data for the process of studying neural network. When monitoring parameters are selected correctly, the results of applied methods must be similar to each other. In this case, a less time-consuming method can be chosen. However, the application of neural networks can produce more adequate and correct results. The results of other methods must be compared with results produced by neural network and corrected.

To predict the future value of this time series, three-layered neural network with hidden layer was implemented. The future value of parameter was calculated in neural network output. If the value exceeds the specified maximum value of parameter, it can predict the time of Cloudlet development.

It is to be noted that, any important parameter among the parameters monitored in base station can be used instead of integral parameter.

The tools and methods presented in the article can also be implemented for the solution of similar problems.

Acknowledgment

This work was supported by the Science Development Foundation under the President of the Republic of Azerbaijan – Grant № EİF-2014-9(24)-KETPL-14/02/1

References

[1] Marios D. Dikaiakos, George Pallis, Dimitrios Katsaros, PankajMehra, Athena Vakali.Cloud Computing - Distributed Internet Computing for IT and Scientific Research// IEEE internet computing, 2009. № 9, pp.10-13.

[2] Alguliyev R.M., Alekperov R.K. Cloud Computing: Modern State, Problems and Prospects// Telecommunications and Radio Engineering, 2013, vol.72, no.3, pp. 255-266.

[3] Rashid Alakbarov, Fahrad Pashayev, Mammad Hashimov. A Model of Computational Resources Distribution Among Data Center Users. International Journalof Advancementsin Computing Technology. Volume 7, Number 2, Mar ch 31, 2015, pp. 1-7

[4] Md. Tanvir Rahman, Md. Sifat Ar Salan, Taslima Ferdaus Shuva, Risala Tasin Khan,"Efficient Sensor-Cloud Communication using Data Classification and Compression", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.6, pp.9-17, 2017. DOI: 10.5815/ijitcs.2017.06.02

[5] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: Architecture, applications, and approaches. Wireless Communications and Mobile Computing. Volume 13, Issue 18. 25 December 2013. Pages 1587–1611

[6] Amandeep Verma, Sakshi Kaushal, "Cost Minimized PSO based Workflow Scheduling Plan for Cloud Computing", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.8, pp.37-43, 2015. DOI: 10.5815/ijitcs.2015.08.06

[7] Han Qi, Abdullah Gani. Research on Mobile Cloud Computing: Review, Trend and Perspectives. https://arxiv.org/ftp/arxiv/papers/1206/1206.1118.pdf.

[8] Tinankoria Diaby, Babak Bashari Rad,"Cloud Computing: A review of the Concepts and Deployment Models", International Journal of Information Technology and

Computer Science(IJITCS), Vol.9, No.6, pp.50-58, 2017. DOI: 10.5815/ijitcs.2017.06.07

[9] Mukesh Goyal, Sukhwinder Singh. Mobile Cloud Computing. International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 3 Issue 4, April-2014, pp: 517-521.

[10] Fangming Liu, Peng Shu, Hai Jin, Linjie Ding, Jie Yu, Di Niu, Bo Li. Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. IEEE Wireless Communications (Volume: 20, Issue: 3, June 2013) Page(s): 14 - 22

[11] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14–23, 2009.

[12] Yujin Li and Wenye Wang. The Unheralded Power of Cloudlet Computing in the Vicinity of Mobile Devices. Globecom 2013 - Wireless Networking Symposium Pages 4959-4964

[13] Ryzhov A.L., Tkhostov A., Pechnikova L. Psychological aspects of involvement in Massively Multiplayer On-line Game (MMOG). International Journal of Psychology. 2015. том 51, № S1, Page(s): 594-594

[14] Eric Lengyel. Foundations of Game Engine Development. Terathon Software LLC. 2016. 200 p.

[15] Joshua Glazer, Sanjoy Madhav. Multiplayer Game Programming. Architecting Networked Games. Pearson Edication Inc. 2015. 365 p.

[16] Debessay Fesehaye, Yunlong Gao, Klara Nahrstedt, Guijun Wang. Impact of Cloudlets on Interactive Mobile Cloud Applications. Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International, pp. 10.

[17] Tinankoria Diaby, Babak Bashari Rad,"Cloud Computing: A review of the Concepts and Deployment Models", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.6, pp.50-58, 2017. DOI: 10.5815/ijitcs.2017.06.07

[18] Debessay Fesehaye, Yunlong Gao, Klara Nahrstedt, Guijun Wang. Impact of Cloudlets on Interactive Mobile Cloud Applications. Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International, pp. 10.

[19] Yujin Li and Wenye Wang. Can Mobile Cloudlets Support Mobile Applications?. https://research.ece.ncsu.edu/netwis/papers/14LW-Info.pdf

[20] Young-Chul Shim. Effects of cloudlets on interactive applications in mobile cloud computing environments. International Journal Of Advanced Computer Technology, Volume 4, Number 1, 2015, pp.54-62.

[21] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, “Dynamic service migration in mobile edge-clouds,” in Proc. of IFIP Networking 2015, May 2015, pp. 9.

[22] R. Alakbarov, F.Pashayev, M. Hashimov. Development of the Method of Dynamic Distribution of Users’ Data in Storage Devices in Cloud Technology. // Advances in Information Sciences and Service Sciences (AISS), Vol. 8, No 1,2016. pp.16-21.

[23] Usman Shaukat, Ejaz Ahmed, Zahid Anwar, Feng Xia. Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges Journal of Network and Computer Applications 62 (2016) pp. 18–40

[24] Amol C. Adamuthe, Vikram D. Salunkhe, Seema H. Patil, Gopakumaran T. Thampi,"Cloud Computing – A market Perspective and Research Directions", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.10, pp.42-53, 2015. DOI: 10.5815/ijitcs.2015.10.06

[25] Christian Meurisch, Alexander Seeliger, Benedikt Schmidt, Immanuel Schweizer, Fabian Kaup, and Max Munlhauser. Upgrading Wireless Home Routers for Enabling Large-Scale Deployment of Cloudlets. https://www.tk.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_TK/Slides/CM/2015_MobiCASE_Presentation.pdf

[26] David Cherney, Tom Denton, Rohit Thomas and Andrew Waldron. Linear Algebra. First Edition. Davis California, 2013. 436 p.

[27] Daubechies I. Ten lectures on wavelets. Izhevsk: Scientific-Resarch Center “Regular and chaotic dynamics”, 2001, 464 pp.

[28] Daubechies, I. Orthonormal Bases of Compactly Supported Wavelets, Comm. Pure Applied Mathematics, vol. 41, 1988, pp. 909-996.

[29] S. G. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 2nd Edition, 1999.

[30] R. Rojas. (1996) Neural networks. Springer-Verlag, Berlin, pp. 479.

[31] Stanislav Osovsky. (2002) Neuron networks for information processing. Publisher: «Finance and statistics» 343 pp.

Authors’ Profiles

Rashid G. Alakbarov graduated from“ Automation and Computer Engineering” faculty of Azerbaijan Polytechnic University named after C.Ildirim. He received his PhD degree in 2006 from Supreme Attestation Commission under the President of the Republic of Azerbaijan. His primary research interests include various areas in cloud computing, data processing, computer networks, virtual computing, particularly in the area of distributed computing. He is head of department at the Institute of Information Technology as of 2002. Since 2010, he has been leading the development of "AzScienceNet" infrastructure. In 2011, he was appointed a deputy director of the institute by the decision of the Presidium of Azerbaijan National Academy of Sciences. He is the author of 64 scientific papers, including 5 inventions.

Fahrad H. Pashayev graduated from Azerbaijan State University. He received his PhD degree in 2011 and Doctor of Sciences in Engineering in 2017 from Supreme Attestation Commission under the President of the Republic of Azerbaijan. His primary research interests include various areas in mathematical modeling, signal processing, software development. He is the author of 121 scientific papers, including 5 inventions.

Oqtay R. Alakbarov is PhD student of Institute of Information Technology of Azerbaijan National Academy of Sciences. His primary research interests include various areas in cloud computing, mobile cloud computing, mobile technologies, particularly in the area of cloud technology applications. He is the author of 2 journal scientific papers and 3 proceedings.

How to cite this paper: Rashid G. Alakbarov, Fahrad H. Pashaev, Oqtay R. Alakbarov, "Forecasting Cloudlet Development on Mobile Computing Clouds", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.11, pp.23-34, 2017. DOI: 10.5815/ijitcs.2017.11.03