Saving Battery of Mobile Station & Response Time by Server with Compression

Rohit Kulkarni, Raghvendra Singh, Piyush Mathur

Abstract-There are some mobile applications which receive the information from application servers by user generated queries. Processing the request on the mobile devices drain the mobile battery. On the other hand, processing user-queries at application servers causes increased response time because of the communication latency during transmission of the large size query. In this thesis work, to minimize battery drain as well as response time query processing on one mid network node (Relay Node) had done. Leasing processing power form mid network node may decrease battery usage on the mobile devices and response times, so that is totally depend on service provider how much it has to lease? The trade of processed data with response time, memory required & energy required is studied. The dynamic programming approach for the optimality to distribute the amount of query processing load on relay node is also used. Here I extended our work with the compression & encryption. LZ4-HC compression technique is used to minimize the size of data so that its processing is automatically decreased thereby it's obvious that there is further more save of battery. At mobile station compression is done. We do feature extraction at relay node as a part of query processing. Encryption is also applied to the extracted features for security purpose at relay node. On the other hand, at application server feature decryption has done with training & classification which are application level functions.

IndexTerms- AES, Artificial Neural Network(ANN), Feature, Extraction, LZ4-HC.

I. INTRODUCTION

In Mobile Augmented Reality systems, processing is done either completely at the Mobile Station, rapidly draining its limited battery resource, or completely at the Application Server, causes to large communication delays. Running the MAR applications on mobile device, in this case depression analyzer application, is also battery intensive. If we run this application on application server, it will increase bandwidth demand over the network with several users using the same application competing for the spectrum. The first hop wireless link between mobile device & base station is a limited bandwidth. Moreover, the transmission latency on the uplink will be higher as larger query data is transmitted through the network. As mobile devices become more sophisticated with higher resolution image and video capabilities, the query data continues to grow. Mid-network processing will reduce the message size.

Manuscript Received on December 2014.

Rohit Kulkarni, Department of Computer Engineering, University of Mumbai, G. M. Vedak Institute of Technology, Tala, Dist. - Raigad, Maharashtra, India.

Raghvendra Singh, Department of Computer Engineering, University of Mumbai, G. M. Vedak Institute of Technology, Tala, Dist. - Raigad, Maharashtra, India.

Piyush Mathur, Department of Computer Engineering, Rajasthan Technical University, Sobhasaria Engineering College, Sikar, Rajasthan, India.

The motivation of Network-Supported Mobile Computing with Optimal Uplink Query Processing using Compression is to improve the Quality of Service of clients which are using mobile applications that are often battery and memory exhaustive. As the request message traverses network hop, some processing to be performed at this mid-network node. This will minimize the power drain at the Mobile Station because of the amount of processing required to be executed on the mobile device. Additionally, the large communication delays may be reduced as processing can reduce the message size. This thesis work removes some of the processing load off the Mobile Station while reducing the size of the request message, and in turn, reducing the communication delays.

Architecture:

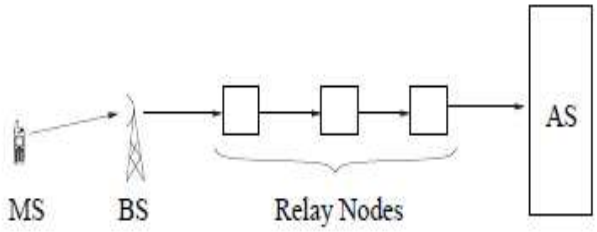

Figure 1. System Model: Mobile Stations (MS) transmit data to the application server (AS) via the Base Station (BS) & relay nodes. The requested data transmitted back to the mobile device. Link may be wired or wireless

In contrast, there are applications where the data necessary to complete a request is too large to store at the mobile device. In Mobile Augmented Reality applications, it is not beneficial to store even part of the large database required at MS. In the applications, the focus is on the request must be transmitted uplink to an Application Server in order to be fully satisfied. There is focus on the uplink scheduling of how much processing to perform at each node in order to minimize latency, battery usage, and leasing costs. In current systems, all of this processing is either done at the MS or the AS. The initial request message can be a extremely large image file and sending it over multiple congested links to the AS will result in large delays. If the request gets processed earlier to transmission, the information required to be transmitted may be smaller, considerably minimizing the communication delay. We consider how to utilize network supported computing to ease the processing load on the MS thereby reducing its battery consumption and extending its operational lifetime. Mid-network processing will reduce the message size. Using the dynamic programming framework, there is load distribution among mobile station, mid network node & application server. The objective is to optimimize the processing on relay node and communication latency. If

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

compression for data is used, obviously the communication latency further gets decreased. There is a use of depression analyzer application which is installed on android based mobile. This application recognizes the mood of speaker whether that person is happy or sad depending on the artificial neural network & feature extraction techniques. LZ4-HC technique for data compression is used here to further reduce latency, memory consequently battery drain of mobile device. Feature encryption is used for secure transmission of data using AES 512-bit key. Feature extraction techniques like F0, FFT, LPC, Cepstral, random features etc. are used. MARF framework is used to implement these techniques.

II. LITERATURE SURVEY

2.1 Mobile Computing

A technology that permits data transmission, by a computer, without connecting to a fixed physical link .The term "Mobile computing" is the use of computing devices, which interact with a central information system, while away from the normal, permanent workplace. Mobile computing technology facilitates the mobile worker to make, access, process, store and communicate information without need to stick up to a single location. By expanding the reach of an organization's permanent information system, mobile computing facilitate interaction with organizational staff. In mobile computing platform, information between processing units flows through wireless channels.

2.2 Feature Extraction & Its techniques

Feature extraction is the process of getting the valuable information of the signal while removing redundant and unwanted information or we can say this process involves analysis of speech signal. However, in practice, while removing the unwanted information, we may also lose some useful information in the process. Feature extraction may also involve converting the signal into a form, appropriate for the models used for classification. The aim is to find a set of properties of an sound that have acoustic link in the speech signal, that is, parameters that can be anticipated through processing of the signal waveform. Such parameters are called features. Next step after the preprocessing for removing noise of the speech signal in the signal modeling is feature extraction. Feature extraction typically includes the process of converting the signal to a digital form (i.e. signal conditioning), measuring some important characters of the signal like energy or frequency response. Many feature extraction techniques are available, some are include:

- Linear predictive analysis (LPC)
- Fast Fourier Transform (FFT)
- Cepstral Feature
- F0

2.3 Artificial Neural Network (ANN)

An Artificial Neural Network is used as recognition method. Architecture of ANN used in this system is a multilayer perception neural network. The network has 576 input neurons, which receive input of LPC cepstral coefficient. Number of hidden layer changes from 1 to 4 layers and number of neurons each layer varies from 5 to 30 neurons. The output of the network is code of recognized word. Figure 2 shows architecture of ANN that used in this system. Speech signals which are to be recognized, the more statistical computation must be done. In contrast, as long as the training of ANN is ended, the speech recognition on ANN is relatively fast. This is because that the dimension of ANN is fixed after it had been trained and is regardless to the amount of the signal to be recognized. Consequently, the advantage of the method ANN for speech recognition is that it can obtain a rapid recognition speed. In addition to the benefit in recognition rate, ANN has another benefit on fault-tolerant capacity, which is a specific characteristic of ANN with respect to other methods. The back-propagation algorithm is used for the training of ANN. Such a system is then called back-propagation neural network. The multiple layer in the model of multiple-layer perceptron means that it contains multiple-layered neurons, in which the way to transmit signals between every two neurons is just like the that in single layer.

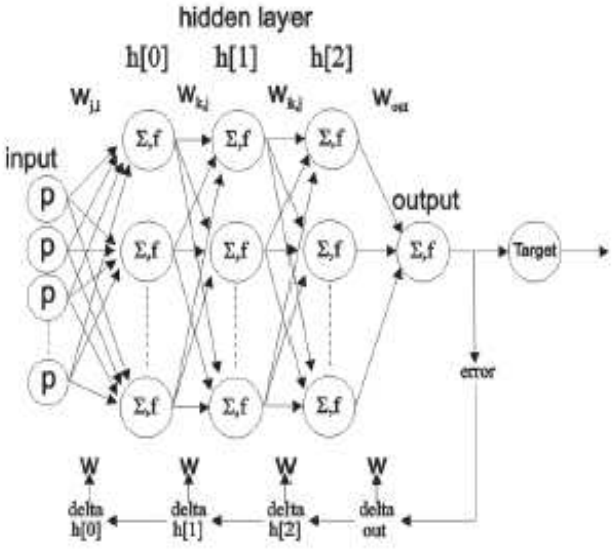

Figure 2. Artificial Neural Networks

2.4 AES with 512 bits key

It offers more security than DES. AES has three variable key lengths but block length is fixed to 128 bits. The three key sizes of AES are 128, 192 and 256 bits. AES 512 bit can be used when higher level of security throughput is needed without increasing overall design area as compared to the original 128 bit AES algorithm. New algorithm consist of the structure which is similar to original AES algorithm but having slight difference is that the plaintext size and key size using input of 512 bit instead of 128 bit. The AES algorithm consist of four main operations are performed during each round: byte substitution, shifting rows, mixing columns and finally adding the round key. AES 128 bit key is considered safe compared to other existing symmetric cipher algorithm. It is extensively used in many application where the security is very important the new AES algorithm provides even more security and double throughput. More security comes from using larger key size, and more throughput comes from using four times larger block size that the block size used in the original AES. The only disadvantage of AES 512 is the need for more design area. The proposed AES 512 algorithm has four main different byte based transformation. The first transformation is the byte substitution which substitutes the value of 512 bit and this is achieved via using parallel s-boxes. The second

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

transformation is shifting rows that shift the rows of the output from previous step by an offset equal to the row numbered. The third transformation is mixing column, where each column of the output from previous step is multiplied by different value. The last transformation in the round is adding round key to the result of this round. The top level architecture of the AES 512 bits the plaintext and key size are 512 bits each (organized in bytes). The AES 512 algorithm processes the data in 10 rounds the resulting cipher text is also 512 bits.

2.5 LZ4-HC

LZ4 is a very quick lossless compression algorithm, providing compression speed at 400 MB/s per core, scalable with multi-cores CPU. In computer science and information theory, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression shrinks bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it. The process of reducing the size of a data file is referred to as data compression. Compression is useful because it helps to reduce resource usage like data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing impose computational or other costs through decompression. Data compression is subject to a space–time complexity. Audio data compression has the potential to reduce the transmission bandwidth and storage requirements of audio data. In both lossy and lossless compression, information redundancy is reduced, using methods like coding, pattern recognition, and linear prediction to reduce the amount of information used to characterize the uncompressed data. LZ4 is a lossless data compression algorithm that is paying attention on compression and decompression speed. The LZ4 algorithm represents the data as a series of sequences. Each sequence starts with a one byte token that is broken into two 4 bit fields. The first field indicates the number of literal bytes that are to be copied to the output. The second field indicates the number of bytes to copy from the already decoded output buffer (with 0 representing the minimum match length of 4 bytes). A value of 15 in either of the bit fields indicates that the length is larger and there is an additional byte of data that is to be added to the length. A value of 255 in these extra bytes represents that yet another byte to be added. Hence arbitrary lengths are represented by a series of extra bytes containing the value 255. After the string of literals comes the token and any extra bytes needed to indicate string length. This is followed by an offset that indicates how far back in the output buffer to begin copying. The extra bytes (if any) of the match-length come at the end of the sequence. Compression can be done in a stream or in blocks. Higher compression ratios can be obtained by investing more effort in finding the best matches. This results in both a smaller file and a faster decompression.

III. METHODOLOGY USED

Training & Testing modules have done. In training all the voice samples, which are in the form of files, are trained to test the one particular voice sample. In testing we have categorized the voice sample in the “Depressed Person (0)” & “Not Depressed Person (1)”. The details of voice sample format are specified next. Let us see these means one by one in detail:

3.1 MARF

MARF means Modular Audio Recognition Framework. It contains a set of algorithms for Sound, Speech, and Natural Language Processing set in an consistent framework to make simple the addition of new algorithms for preprocessing, feature extraction, classification, parsing, etc. executed in Java. MARF is also a research platform to build up various performance metrics of the implemented algorithms.

3.2 Voice Sample Format

A voice sample format used in my implementation is specified as follows:

- Audio Format: PCM signed (WAV)
- Sample Rate: 8000 Hz
- Audio Sample Size: 16 bit
- Channels: 1 (mono)
- Duration: from about 7 to 20 seconds

All training and testing samples are recorded through an external sound recording program (MS Sound Recorder) with the help of a standard microphone. Each sample is saved as a WAV file with the above mentioned properties and stored in the correct folders where they will be uploaded from within the main application. The PCM audio format (Pulse Code Modulation) means the digital encoding of the audio sample contained in the file and is the format used for WAV files. In a PCM encoding, an analog signal is characterized as a sequence of amplitude values. The range of the amplitude values is given by the audio sample size which represents the number of bits that a PCM value consists of. For implementation, the audio sample size is 16-bit which means that a PCM value can vary from 0 to 65536 is selected. Since here is the use of PCM-signed format, this gives an amplitude range between −32768 and 32768. Means the amplitude values of each recorded sample can vary within this range. Also, this sample size offers a larger range and hence gives better accuracy in representing an audio signal, then using a sample size of 8-bit which is limited to a range of (−128, 128). Therefore, a 16-bit audio sample size is used in order to give the best possible results. The sampling rate means to the number of amplitude values taken per second during audio digitization. According to the Nyquist theorem, this rate should be at least twice the maximum rate (frequency) of the analog signal that is going to digitize. Otherwise, the signal couldn’t be correctly converted in digitized form. Here sampling rate is 8 kHz, this means that actual analog frequency of each sample is limited to 4 kHz. The number of channels refers to the output of the sound (1 for mono and 2 for stereo – left and right speakers). Here a single channel format to avoid complexity during the sample loading process is considered.

3.3 Feature Extraction

3.3.1 Hamming Window

In many DSP techniques, it is needed to consider a smaller piece of the whole speech sample rather than trying to

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

process the whole sample at once. The technique of dividing a sample into smaller pieces is called "windowing". The simplest type of window to use is the "rectangle", which is simply an unaltered cut from the larger sample. Unfortunately, rectangular windows can causes errors, because near the edges of the window there will be significantly a sudden drop from a high amplitude to nothing. A better way to window the sample is to slowly fade out toward the edges, by multiplying the points in the window by a "window function". If there is consecutive windows side by side, with the edges faded out, it will alter our analysis as the sample will get modified by the window function. To avoid this, it is needed to overlap the windows hence all points in the sample will be considered equally. To avoid all distortion, the overlapped window functions should add up to a constant. This is exactly what the Hamming window does. It is defined as:

$x = 0.54 - 0.46 \cdot \cos (2\pi n / l-1)$

Where x is the new sample amplitude, n is the index into the window, and l is the total length of the window.

3.3.2 FFT Feature Extraction

The frequency-domain view of a window of a time-domain sample gives us the frequency characteristics of that window. In feature identification, the frequency characteristics of a voice could be considered as a list of "features" for that voice. If there is combination of all windows of a vocal sample by taking the average between them, average frequency characteristics of the sample will be attained. Then, if average the frequency characteristics for samples from the same speaker is computed, finding the center of the cluster for the speaker's samples is possible. Once all speakers have their cluster centers recorded in the training set, the speaker of an input sample should be recognizable by comparing its frequency analysis with each cluster center by some classification technique. Since we are dealing with speech, greater accuracy should be attainable by comparing corresponding phonemes with each other. That is, "th" in "the" should bear larger resemblance to "th" in "this" than will "the" and "this" when compared as a whole. The only disadvantage of the FFT is the window used as input. Using a normal rectangular window can result in glitches in the frequency analysis as a sudden cutoff of a high frequency may distort the results. Hence it is required to apply a Hamming window to the input sample, and to overlap the windows by half. Since the Hamming window adds up to a constant when overlapped, no distortion is introduced. When comparing phonemes, a window size of about 2 or 3 ms is appropriate, but when comparing whole words, a window size of about 20 ms may be more to useful. A larger window size makes a higher resolution in the frequency analysis.

3.3.3 Training

Training in a Feed-Forward Neural Network is done through the an algorithm called Back-Propagation Learning. It is based on the error of the final result of the network. The error the propagated backward throughout the network, based on the amount the neuron contributed to the error. It is defined as follows:

$w_{i,j} \leftarrow \beta w_{i,j} + \alpha \cdot a_i \cdot \Delta_i$

where $\Delta_i = Err_i \cdot df / dx (in_i)$ for neuron i in the output layer.

&

$\Delta_i = df / dt (in_i) \cdot \sum_{j=0}^{n} (\Delta j)$ for neurons in other layers. The parameters α and β are used to avoid local minima in the training optimization process.

They weight the combination of the old weight with the addition of the new change. Usual values for these are decided experimentally. The Back-Propagation training method is applied in conjunction with epoch training. Given a set of training input vectors Tr, the Back-Propagation training is done on each run. However, the new weight vectors for each neuron, "vector" w', are stored and not used. After all the inputs in Tr have been trained, the new weights are committed and a set of test input vectors Te, are run, and a mean error is calculated. This mean error determines whether to continue epoch training or not.

3.3.4 Usage as a Classifier

As a classifier, a Neural Network is used to map feature vectors to speaker identifiers. The neurons in the input layer correspond to each feature in the feature vector. The output of the network is the binary interpretation of the output layer. Therefore the Neural Network has an input layer of size m, where m is the size of all feature vectors and the output layer has size (log2(n)), where n is the maximum speaker identifier. A network of this structure is trained with the set of input vectors corresponding to the set of training samples for each speaker. The network is epoch trained to optimize the results. This fully trained network is then used for classification in the recognition process.

3.3.5 Random Classification

This may feel strange, but there is a random classifier in MARF. This is more or less testing module just to quickly test the PR pipeline (Pattern Recognition). It picks an ID in the pseudo-random manner from the list of trained IDs of subjects to classification. It also serves as a bottom-line of performance (i.e. recognition rate) for all the other, slightly more sophisticated classification methods meaning performance of the aforementioned methods must be better than that of the Random; otherwise, there is a problem.

IV. IMPLEMENTATION

On client side, the android application namely depression analyzer is installed. We consider this is the kind of MAR application which is battery exhaustive. With this application we proceed further. By entering IP address & port number the connection with server (which is in same network with client) gets established (Fig. 3). After getting connection we start application. First by putting the log in id of the registered user (user is already registered on server database) & its password for authentication (Fig. 4). After authentication "Authenticated User" message will appear in welcome window. User profile "My Profile" is there. With the help of that one can view details of the user by clicking on this icon, it will take you to the details. User can update/delete the account (Fig. 5).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Figure 3. Snapshot while entering IP address & port number

Figure 4. Snapshot while log-in ID & Password of a user

Figure 5. Snapshot of Welcome screen for user

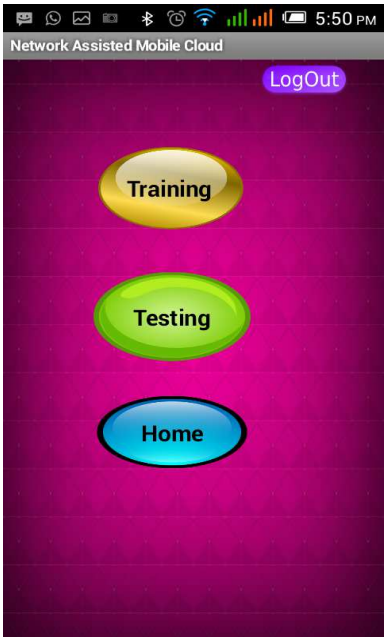

Figure 6. Snapshot showing Training & Testing

Under recognition there are three modules namely training & testing. In training, some common features of .wav audio recorded files by user are extracted. Means whether that person is in happy (not depressed) or sad (depressed) mood is recognized. This is done using artificial neural network (ANN) & feature extraction techniques. These .wav files are first uploaded on server after that training is performed. Using upload & delete module we can upload the .wav file on the server respectively. We can upload a file in sad/happy emotion type on the server. The selected emotion type of these files is then gets trained (Figure 6). In testing, there are two types file testing & mike testing. In file testing, the files on the server are classified in sad/happy mood through training. In mike testing, user records audio file in his voice. This .wav file gets transferred to application server via relay node & saved in database at server. Using feature extraction techniques & neural network these .wav are classified in sad/happy mood with the help of trained samples. When everything has done user may log out the application. At server database, we maintain information about user, recoded files by user, uploaded files by user & emotion type i.e sad/happy as classify (classifiers 0 is for sad & 1 is for happy are used) (Figure 7.)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

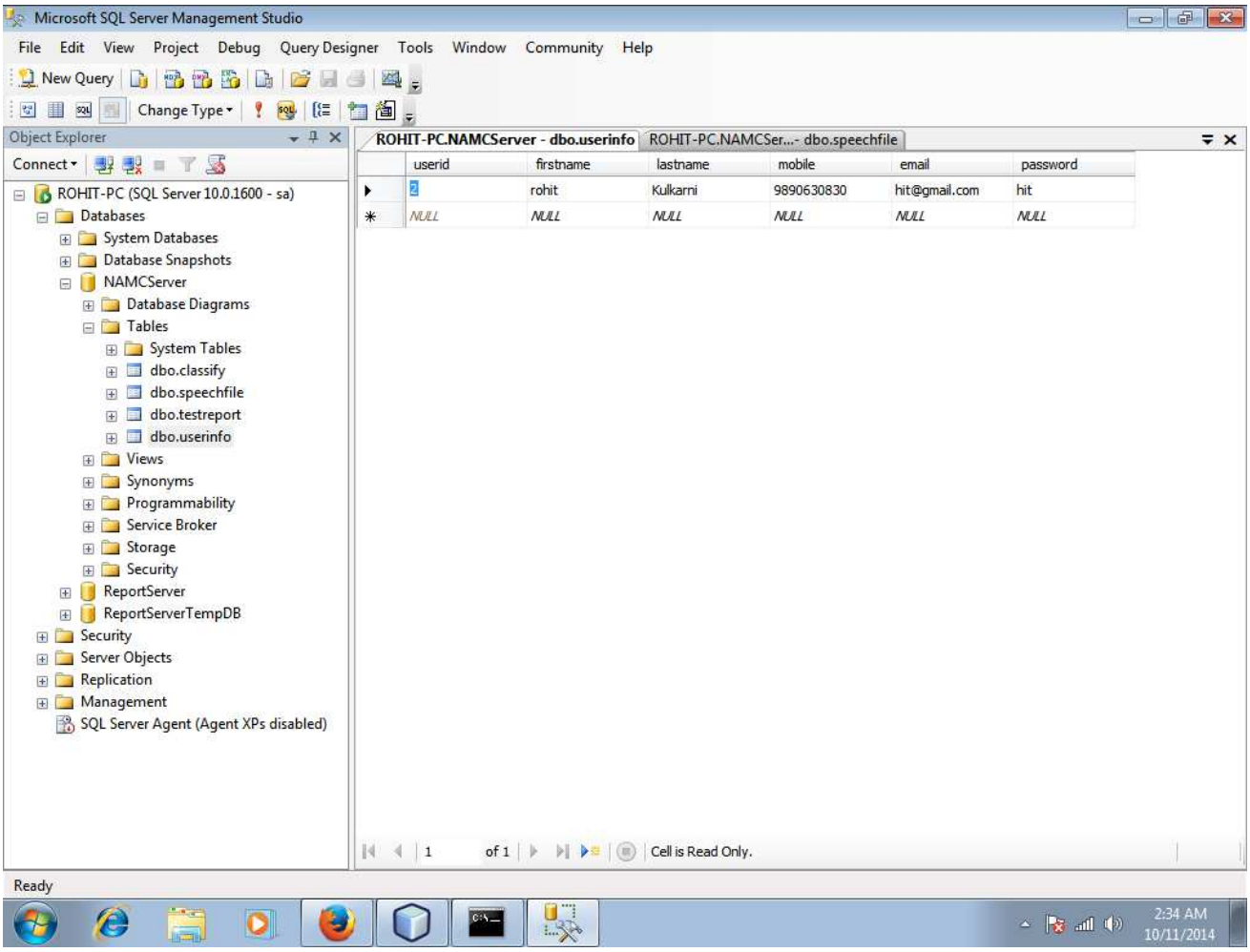

Figure 7. User data at Server

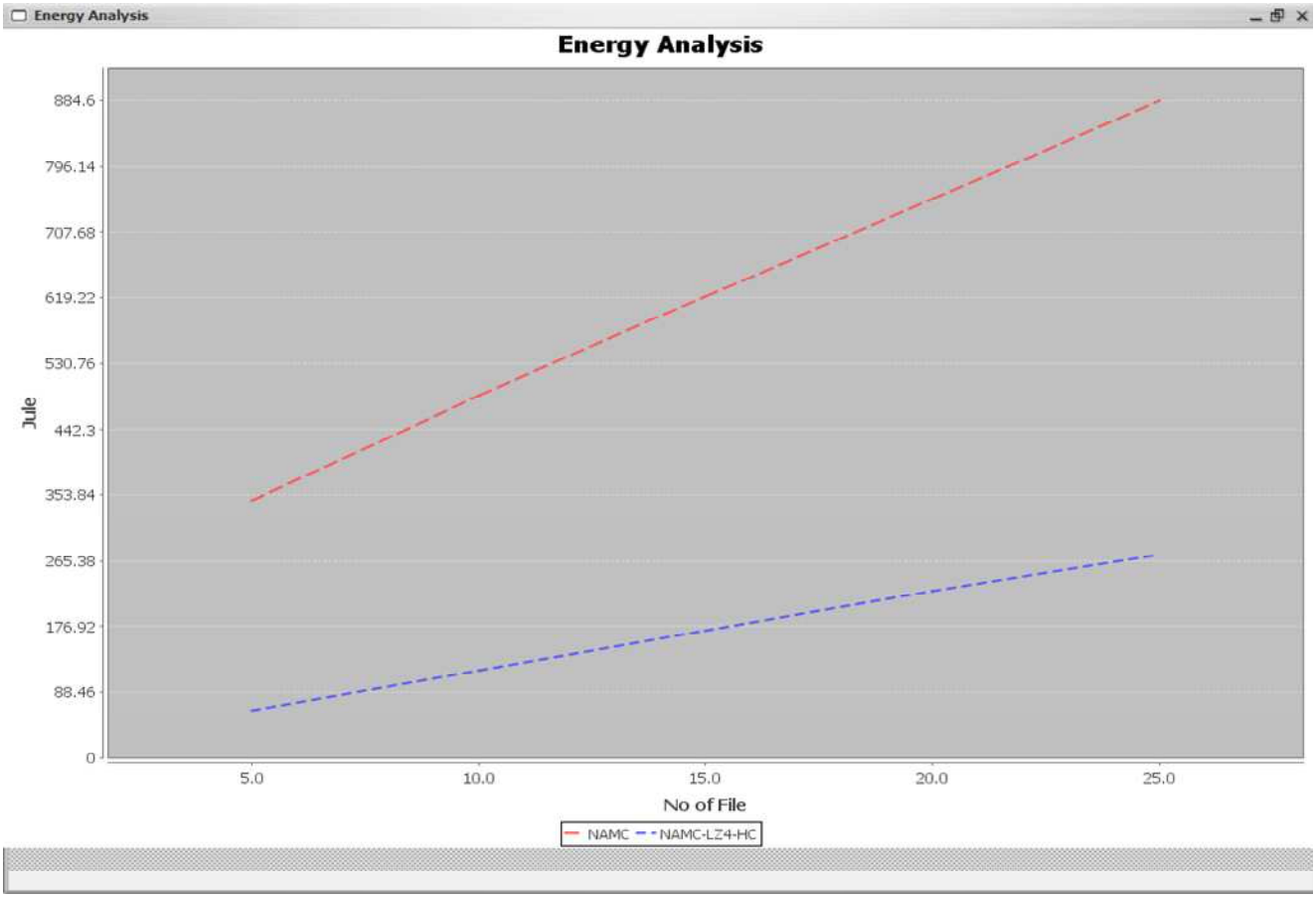

Figure 8. Time Analysis:A relation between data processed & memory by compression

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

V. CONCLUSION

In this thesis work, we have implemented much of the query processing at mid-network node to reduce the burden on MS as well as communication latency. The popularity of mobile applications is steadily increasing. Many of these applications require significant computation power, especially in the case of multimedia applications. As the demand, as well as the sophistication and required computation power, for these types of applications increases, battery and communication bandwidth limitations may prevent the use of many of these applications. Network-supported Mobile Computing by optimal uplink query processing with compression can help to ease the processing burden off the Mobile Station without increasing the service latency.For high security purpose we used AES with 512 bits keys for encryption. LZ4 HC algorithm is used to compress data so that its processing get further decreased. It will affect the factors like battery, communication latency etc. If one may want to add another algorithm for encryption & decryption he can proceed with it.

VI. ACKNOWLEDGEMENT

We are grateful to Prof. Mr. Piyush Mathur, Computer Engineering Department, Sobhasaria Engineering College, Sikar, Rajasthan, for his valuable guidance in the completion of this paper.

REFERENCES

[1] Network Assisted Mobile Computing with Optimal Uplink Query Processing by Carri W. Chan, *Member, IEEE,* Nicholas Bambos, *Member, IEEE,* and Jatinder Singh, *Member, IEEE,*
[2] Modular Audio Recognition Framework v.0.3.0.6 (0.3.0 final) and its Applications by the The MARF Research and Development Group.
[3] Dynamic Programming and Optimal Control Volume I by Dimitri P. Bertsekas.
[4] AES Algorithm Using 512 Bit Key Implementation for Secure Communication by Rahul Jeurkar & Shrikrishna Chopade.
[5] Review of Feature Extraction Techniques in Automatic Speech Recognition by Shanthi Therese S. ,Chelpa Lingam.
[6] The process of Feature Extraction in Automatic Speech Recognition System for Computer Machine Interaction with Humans: A Review Bhupinder Singh, Rupinder Kaur, Nidhi Devgun, Ramandeep Kaur.

Sites Referred:

- www nptel.com
- http://social.msdn.microsoft.com
- www.google.com
- www.wikipedia.org

Rohit Kulkarni, Assistant Professor, GMVIT, Raigad Dist., Maharashtra, India & M. Tech perusing from RTU (Kota).

Raghvendra Singh, Assistant Professor, GMVIT, Raigad Dist., Maharashtra, India & M. Tech perusing from RTU (Kota).

Piyush Mathur, Associate Professor, Computer Engineering Dept., Sobhasaria Engineering College, Sikar, Rajasthan, India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.