Abstract No: 901 *IT, Mathematics and Statistics*

A SYSTEMATIC APPROACH FOR THE UTILISATION OF EXCESS HARD DISK MEMORY OF PERSONAL COMPUTERS

R. Jarachanthan* **and K. Sarveswaran**

Department of Computer Science, University of Jaffna, Sri Lanka
**ratminjar@gmail.com*

Access to the multimedia content has been rapidly increasing with the popularisation of the Internet. Users therefore need a more Hard Disk Memory (HDM) to store such content. Organisations and educational institutions like universities are buying cloud space and new hardware such as Network Access Storage to store electronic content. On the other hand, HDM of Personal Computers (PC) in such organisations is underutilised. However, no existing software or approaches available to utilise such unused space of PCs. Key challenges of this problem involve dynamic nature of the location of PCs, heterogeneity in configurations and security. For instance, PCs in an organisation can be removed or added newly. Also the configurations of PCs may not be the same in a location. Further, the administrator of a PC has the privilege to access anything stored in it. There are several approaches and software available to backup data within a computer or between computers such as File Transfer Protocol (FTP), Backup4u, rsync. Further, effort has been made to utilising memory on homogeneous server computers called Google File System (GFS). During the feasibility study of this research, it was identified that only 19% of HDM are being utilised at the University of Jaffna among 14 Laboratories; this means more than 52,000GB HDM are unused.

A systematic approach has been proposed in this research that would provide a way for the utilisation of the excess HDM of the PCs in an organisation with due consideration to dynamicity, security, heterogeneity and availability. A system also has been developed based on the proposed approach. It has two software components, namely client software and server software. Both have been developed using Java so that it can be installed irrespective of the type of operating system. The server software should be installed in a computer (server) through that files can be uploaded and downloaded. The client software should be installed in PCs (clients) where the uploaded files will be stored. When a file is uploaded to the server it will be compressed, if the file size can be reduced, so that transmission time between server and clients also can be reduced. Then, the compressed or uncompressed file will be chunked into 64MB pieces. Next, each of these chunks will be sent to three clients (replicas). Not all the chunks of a file will be stored entirely on a single computer. A log of uploaded file will be maintained in the server and if a user wishes to retrieve a file, then the chunks of the relevant file will be retrieved from the clients. Then the chunks will be merged and given to the user. Hash function and encryption are used to ensure the security. The hash value of all the chunks will be stored at the server before they are transferred to clients. And then when a chunk is retrieved from a client, the hash value is cross-checked to make sure no tampering has happened while being transferred or stored in clients. Further, chunks in clients will not be much more meaningful. However, in order to make data more secure, all the chunks are encrypted using Advanced Encryption Standard (AES). The study of encryption algorithms showed that the AES, a symmetric key algorithm, is more suitable for this purpose. In addition, using the hash functions and replica the Byzantine failures also can be handled. The pilot run of the developed system in a laboratory was successful. During the pilot run, an average of 63Mbps transfer speed was noted when storing and retrieving a file. Therefore, the proposed approach is viable and can be implemented in organisations.