

# LOCATION TRANSPARENT SERVICE WITH IMPROVED AVAILABILITY [LTSIA] FOR GRID

A. Suthan<sup>1</sup> and D. Kesavaraja<sup>2</sup>

<sup>1</sup>Department of Master of Computer Applications, Dr. Sivanthi Aditanar College of Engineering, Tamil Nadu, India

E-mail: 83sutha@gmail.com

<sup>2</sup>Department of Computer Science and Engineering, Dr. Sivanthi Aditanar College of Engineering, Tamil Nadu, India

E-mail: dkesavaraja@gmail.com

## Abstract

*Grid Computing provide ample opportunities in many areas. A authorized user should be able to access any data that has been created by him, in the Grid, considering all the factor for security and other vital criteria such as reliability, availability, secure reading and writing of data. The proposed service helps the authorized user to create, modify, view the data that has been created by him or his group, irrespective of whether he is inside the grid environment or he is outside the grid. This service is maintained and moderated by a co-ordinator which takes care of factors of security, other vital criteria and properties of distributed and grid environment. In order to increase the availability of data LTSIA uses a replication algorithm that decides the number of replicas that has to be created and where to store the replicated data. The decisions are made from the information stored in the service table linked with co-ordinator. Elliptic curve crypto Algorithm is used to encrypt data while being transmitted outside the grid. The authorized user can use a unique id assigned to him to decrypt the data or shall use a digital certificate system as a tool to decrypt it.*

## Keywords:

*Availability, Cloud Computing, Grid Computing, Location Transparency, Replication, Security*

## 1. INTRODUCTION

The Location Transparent Service with Improved Availability does the work partly that of the Distributed systems[1]. Any distributed system has the property of providing location transparency in the network of systems, where the actual distributed system is implemented. The Location Transparent Service with Improved Availability actually provides the part work of location transparency only as the tool [6] [7]. Hence any distributed computing tool can be considered as the related work for this system. The important things to be discussed with regards to the distributed system and the Location Transparent Service with Improved Availability are Transparency, Location Transparency, Naming Transparency, Communication between nodes, File Accessing models, Replication Transparency [8][10].

Transparency is a main goal of a distributed system to make the various available systems in the network to look like the same as a single virtual system. Transparency is the concept of making the details of technical information unavailable to the user. The user does not know the actual working mechanism of the system [9][16]. The user is hidden from all these issues. According to the user all these happen only in the local system and there is no connection between the other systems [17].

There are plenty of Transparency issues identified by the international standards organization. Out of these the major transparency issues for a distributed system are Access

Transparency[11][12] Hiding details of Access from the user. Replication Transparency: Hiding details of replicated copies of file to the user [18]. Failure transparency: Hiding details about some system that has failed.

Hiding details about the migration of a process from one system to other. Performance Transparency: Hiding details about individual performance of the system. Scaling Transparency: This hides the details about the number of systems that has been added into the network. Concurrency Transparency: This deals with the concept of hiding information about concurrent process getting executed [19][20] [21] [22].

## 2. PROPOSED WORK

The Location Transparent Service with Improved Availability or the proposed work has been developed to solve some of the major issues that are considered critical in grid computing environment. The service mainly considers the problem of working with the data that a user created in a grid. It also takes into consideration the problem of availability when the originating node has been shut down or is not in a condition to boot up[13].

A user who creates some data in one of the system (originating node) from among the various systems that exists in a grid. The user saves the data in the originating node. The user later requires accessing the data in a different node being either inside or outside the grid. If this is the situation then the entire data and information the user was working with has to be copied in to the local node manually. If this kind of information was vital then it would be very critical and also consider a end user who remembers the file location and even the system in which he stored the file. But at that point of moment the system is being used by another user and he is not aware of how to get his file from that system to a remote system. This is a problem that most users encounter. Getting the file in a network may be a easiest job for a computer professional, but not that much easy for a end user[14][15]. The last set of problem is regarding the availability of the file. Consider a user has created a file and now the system in which he created the file has been shut down or is in a condition that it can't boot up. When this is the condition the user will be unable to retrieve his file until the system gets ready. This tool takes care about this factor and provides a solution. The LTSIA takes into consideration these three aspects and tries to find the solution for these problems.

The Location Transparent Service with Improved Availability acts as a solution for the above said problems by providing the facilities as follows

1. Location transparency to the user regarding the file information.
2. Retrieve any file that the user has registered into the system from any of the system in the network.
3. Create replicas of file that is distributed throughout the network of systems.

### 3. WORKING ARCHITECTURE

The necessity of LTSIA lies in its ability to distribute the files over the local network as well as over the cloud. The LTSIA avoids the general problems of locating the data files and other criticality in dealing with data. The Architecture of LTSIA describes the overall structure of the system.

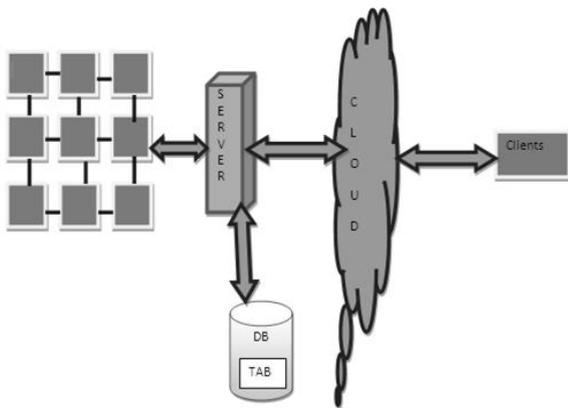


Fig.1. Working Architecture of LTSIA

The Architecture contains the following components.

1. Data Grid
2. Server
3. Cloud
4. Clients
5. Database

The Data grid is the part where the file is created and the data is going to be distributed. The server takes care of assigning the unique ID for the files that are created and also in deciding the replication of the file and the location of these replicas to be stored. These data are stored in the Database. The cloud describes the WAN through which the user will be contacting the local grid when he is in a different geographical area. The client system denotes the system through which the user will be communicating to the grid. All the communicating lines between the various components of the architecture are double sided since all data flow are double sided. The user searches for the file that he needs into the cloud irrespective of where he is, either outside the cloud or inside. The system takes care of all the other aspects like replicating the file, locating the file, allocating binary name for the files and all the other issues keeping it transparent from the user.

### 4. SYSTEM MODEL

The following Fig.2 shows the stages of data flow in the LTSIA.

<b>Service Boot up Process</b>
<b>Replication Algorithm</b>
<b>File Distribution Process</b>
<b>Indexing and Searching</b>
<b>External Cloud Data Service</b>
<b>Security using ECC Algorithm</b>

Fig.2. Flow of LTSIA

The System Model of LTSIA contains the following Processes to be part of the system.

1. Service Boot up Process
2. Replication Algorithm
3. File Distribution Process
4. Indexing and Searching
5. External Cloud Data Service
6. Security using ECC Algorithm

#### 4.1 SERVICE BOOT UP PROCESS

The Service boot up Process has the most important part in the effective utilisation of the service. It takes care of synchronising a new node that enters into the stream with the other existing nodes. Just imagine a situation where there are plenty of nodes that are at present working with each other along the service to implement the location and replication transparency. Now a new node enters into the scene, by some user booting up the machine. The user also initiates the service so that he will be able to retrieve some files that he needs from the grid. The system in which the service has been initiated may be either a system that has already participated in the grid or is the first time that it is participating. The service has to prepare itself to both the conditions as the existing nodes will have lot of information that has to be shared between the nodes of the grid.

Just imagine both the conditions of the node participating in the grid for the first time and that the service has already participated before a break and is now rejoining the grid. Let us first take into consideration that the node is participating in the grid for the first time. In this situation the possibility is that the service does not have any idea about other nodes and also about the list of data files that have been added into the grid. The node now sends a probe signal to all the nodes in the network stating its arrival into the grid. All the nodes that are already in the network will send back a reply to this node about their IP address and other information that will be necessary for the node to use in further communication. This reply message also will have a data that says about how long that sender node has been effectively participating in the network. This data can be called the Time Information Data (TID). This data is helpful in finding the node which will be the oldest among all the systems and which can be relied upon for any further queries. The TID is stored in the local node's database and also it starts a timer to create its own TID. The system then chooses the replier with the highest TID and starts its communication with it. This communication will share information about all the files that are present in the grid and other information that the LTSIA will

have to know for effective participation. Once it has obtained all the information and the service is ready it joins the group and is ready for further instructions. The other part of the initial stage is with a node that has participated in the network but has been disconnected and now is again ready to join. This stage is critical and also involves necessity to monitor the data consistency issues. The data consistency issues arrive here because this node will have some information along with it,

Which is the remains of the previous session of the node participating in the network. But there will be a lot of changes that the node has to make in its database before it once again enters into the network. These changes will be those that have occurred during the time when the service was shut down in the local node.

The service works in the same fashion and gets information from the system that has the highest TID, updates its database with all the information it provides, and gets ready for the next step.

## 4.2 REPLICATING THE FILE

This process takes the most important part in providing the availability for the system. The availability of the service gets increased with the file present at any one node when a user searches for it at the moment of request. In the LTSIA the service replicates the data file to select nodes within the grid and improves availability following the LALW Mechanism [12]. The decision of choosing the nodes where the replicated data files have to be stored is decided based on various factors like availability of the node (TID), Reliability etc. This service thereby trades availability to memory capacity and duplication. The replication of the file following the LALW Mechanism increases the complexity of maintaining data consistency, which is taken care by the TID.

## 4.3 FILE DISTRIBUTION PROCESS

In the File Distribution Process, the user registers any file that he created in the service. This registered file is assigned a binary name that is unique with all file names registered in various nodes of the grid. The file is stored in a database that is linked along with the central co-ordinator. This registration of the file is also intimated to all the systems that are at present in the grid, so that all the users can at the same time search for the file and use it if necessary from the moment of registration. The unique id given to the file helps in identifying the file if in case two users have used the same file name for a data file they have created in two different nodes of the grid.

## 4.4 INDEXING AND SEARCHING

When a user wants to access the file from the system, he selects the name of the file from the list of files available in the directory and asks for retrieval. The software tries to find the location of the file from the data that is present in its database and refers with the system that holds the file for usage. It also passes the address of the file to the retrieval Process for further work. The file location is found with the help of the binary name that has been created for every file. This file name ensures that the file is unique and avoids the various complexities in searching the file.

As soon as this Process gets the path of the file it requests for a copy of the file that is to be retrieved. The copy is brought to the system where the user is currently working and then the other copies that exist are locked for access, in order to avoid consistency problems that may occur in the future. The user has to lock the file before modifying it and has to register it to the system after modification. Once the file is locked the file gets inaccessible for any other user to use it during the process. Only if the file is added again it gets possible for other users to use the file. This mechanism is used in order to ensure data consistency.

## 4.5 EXTERNAL CLOUD DATA SERVICE

This property enhances the service by providing an opportunity for the user to access any data that he is permitted with to be accessed from anywhere irrespective of the geographical location of the user. Consider a situation where a user is located far away from the grid and he is in need of accessing some data that he has created. The External cloud data service takes care of this requirement. It requires that the user first logs in using his authentication. The user when requests for a file is provided with a self extracting file that contains the data file the user requested for. If the user is using a system that he can rely upon he can extract the file using a unique ID that is assigned for every user and is known only to him. This preventive action is used in order to prevent attacks where the authenticate login of a user has been compromised.

So even if an attacker knows the login details he will not be able to access any data. Similarly if the user is not confident of the system he uses and fears about key tapping and other issues he shall use hardware USB that is provided to him, which contains a digital certificate that will unlock the self extracting file.

## 4.6 SECURITY USING ECC ALGORITHM

The main attraction of ECC over RSA and DSA is that the best known algorithm for solving the underlying hard mathematical problem in ECC (the elliptic curve discrete logarithm problem (ECDLP) takes full exponential time. RSA and DSA take sub-exponential time [2][3]. This means that significantly smaller parameters can be used in ECC than in other systems such as RSA and DSA, but with equivalent levels of security. A typical example of the size in bits of the keys used in different public key systems, with a comparable level of security, is that a 160-bit ECC key is equivalent to RSA and DSA with a modulus of 1024 bits. The lack of a sub-exponential attack on ECC offers potential reductions in processing power and memory size [4][5]. These advantages are especially important in applications on constrained devices.

## 5. METHODOLOGY IMPLEMENTED

In any System the general methodologies that are implemented form a major part of the system being successful. Since the LTSIA has been based on the Distributed systems concepts a lot of methodologies that play a part of any distributed system are introduced into the system. Some of the methodologies that are implemented are Replicated Migrating blocks, Binary Naming etc. All these methodologies have been chosen since they satisfy the basic requirements of the system

that has been designed. These methodologies actually improve the overall functioning of the system and also help in providing efficiency for the user. These methodologies do provide basic functionalities of the system.

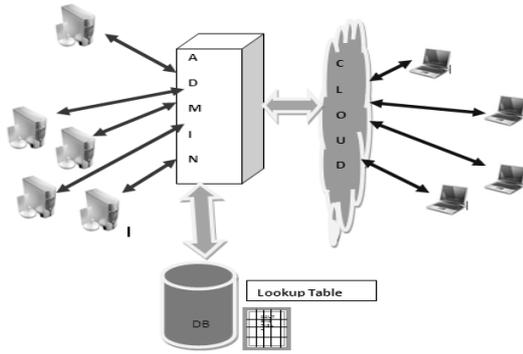


Fig.3. LTSIA Model

The above figure illustrates a detailed vision of the architecture given above in a distributed vision. All the components form a part of the architecture also, except the look up table. The lookup table is part of the Database which has all the information collected during the service start up process and other information appended into the database during processing.

There are various methodologies for implementing the aspects of the tool for various characteristics. Here are some of the methodologies implemented in the system.

### 5.1 REPLICATED MIGRATING BLOCKS

This strategy is a methodology for implementing the sequential consistency model for file sharing. According to this methodology the file migrates from one system to the other, and also is replicated. This strategy is used for replicating the file.

### 5.2 BINARY NAMING

This strategy is used for creating a secondary name for the file that is being registered with the tool. This secondary name is essential in order to avoid the complexity or duplication of names i.e Naming transparency. Also it is possible that the file when it is copied to the other system may face a name conflict with some file that has been already loaded into the system by some user who is working on it. The system thus identifies the two files from the different binary names that have been given to them. Spotting the files and retrieving the files are also very easy with this.

## 6. IMPLEMENTATION

This system has been simulated and tested using java swing in a lab environment. LTSIA was simulated with thirty two nodes scattering around with 7 different input functions in order to test the scalability and reliability of the system. The servers were protected with a security mechanism using the ECC algorithm. Availability of the data is ensured by properly replicating the file based upon the usability.

The below is a collection of various stages of the system, where a file is added, modified, downloaded etc.

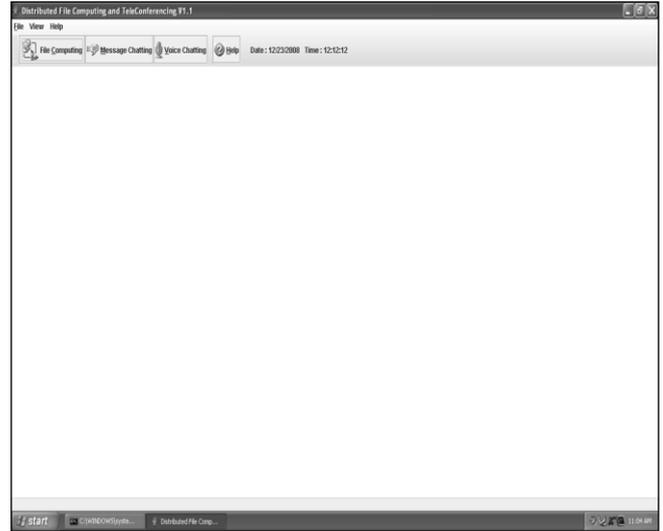


Fig.4. LTSIA System

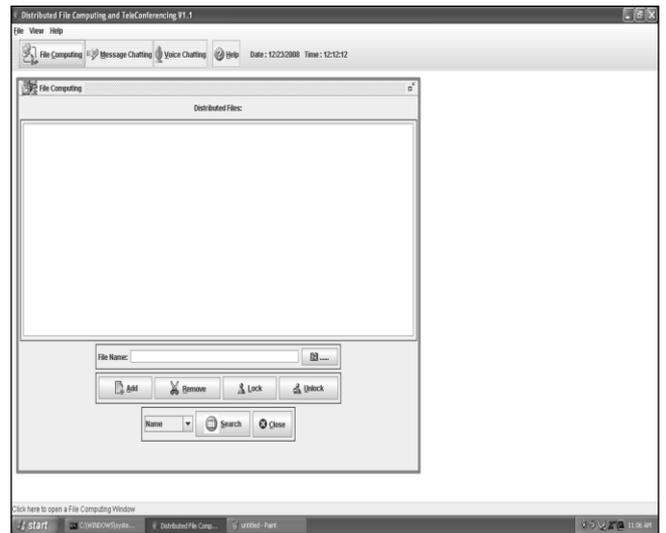


Fig.5. LTSIA System ready to add a new file into it

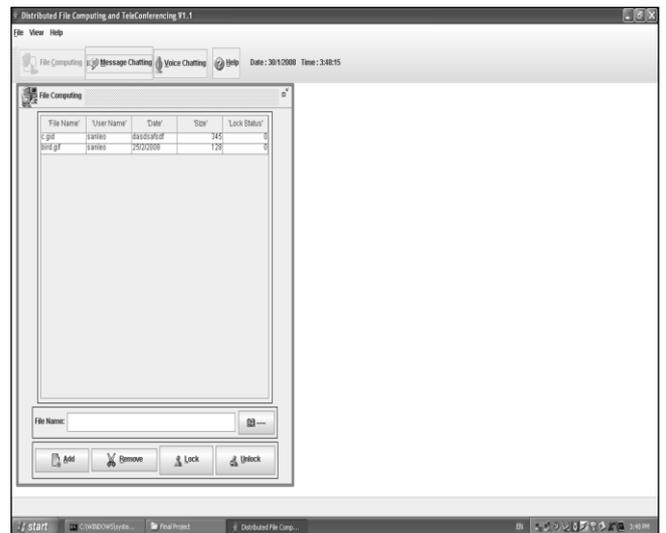


Fig.6. LTSIA Database showing List of Files added

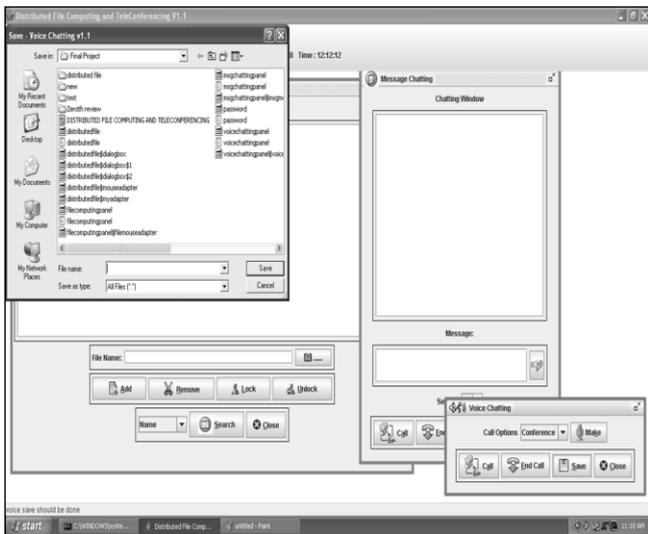


Fig.7. LTSIA Model for File distribution

### 7. PERFORMANCE ANALYSIS

The functioning of any system can be measured only with the effective output it gives to the system. The tool has been found to satisfy the basic considerations that were thought of before the system was designed. The tool has brought out Location Transparency as well as Replication Transparency to the user along with distributing the file for the user. The tool also tries to implement the concept of Security by providing user accounts for individual users.

The system on its aspect is secure. It provides security for the files in such a way as of to restrict its access to other users. This is achieved in the following way. It is possible for a user to join a group or to create a user account for the user. Any file that is added into one account can't be viewed by other users. The following table lists the availability of the data at different nodes.

Table.1. Processed Data Set

Nodes	Availability					AVG
	S-1	S-2	S-3	S-4	S-5	
File-1	84	95	99	87	83	89.6
File-2	98	97	81	85	75	87.2
File-3	90	78	71	96	91	85.2
File-4	100	91	71	94	97	90.6
File-5	98	71	86	72	73	80
File-6	76	89	95	77	75	82.4
File-7	87	75	100	76	99	87.4
File-8	85	76	95	94	99	89.8
File-9	100	86	72	73	77	81.6
File-10	82	95	86	87	75	85
File-11	85	80	69	100	80	82.8
File-12	91	88	100	78	98	91
File-13	69	72	78	69	87	75
File-14	81	100	97	85	90	90.6
File-15	97	98	73	91	96	91
Average Rate	88	86	84	84	86	

Table.1 lists the availability of every data file in a particular server the average column denotes the availability of a particular file in the grid, taking in to consideration the entire cluster of servers.

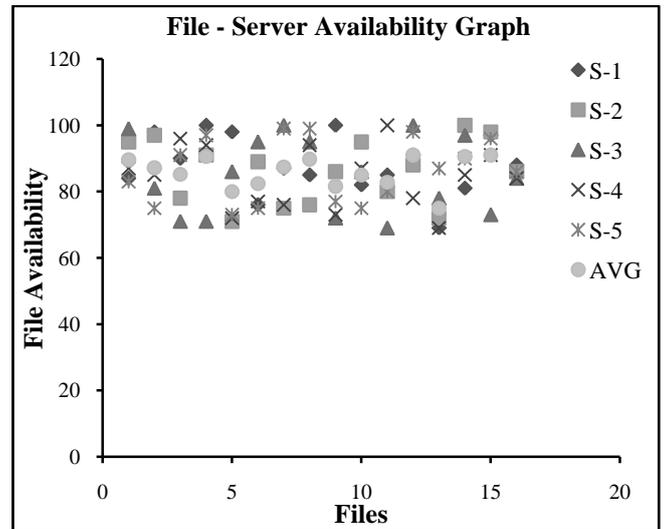


Fig.8. LTSIA File Availability graph per Server

Fig.8 shows a graphical representation of the availability of a particular file over the cluster rating against the availability in a server.

### 8. CONCLUSION

The Location transparency is a user friendly tool that solves the problem of using files in a group of systems connected through the network. The tool achieves this with the help of location transparency concepts of the distributed computing mechanism. This tool was the outcome of interest in the distributed computing environment and the eagerness to solve the problem of system availability and file usage that are experienced in our day to day life in environments where we come across networked systems. Some of the future enhancements that has been mentioned earlier, if implemented in the system can bring in a lot of improvement to the systems reliability and availability. The tool provides applicability in a lot of environments where it can be used to fulfil the requirements of the system.

### ACKNOWLEDGMENT

We would like to thank our management and co faculty members for helping and bearing with us during the period in which we were involved with this work. Also we would like to thank all the authors of the papers we have referenced to bring up our work, without which it would have been very difficult to complete our work

We are the heaviest burden to our parents in many cases yet their passion upon us gave us support to get this work done. We dedicate this research to our beloved parents.

## REFERENCES

- [1] Entezari-Maleki, Reza and Movaghar, Ali, "Availability modeling of grid computing environments using SANs Software", *19<sup>th</sup> International Conference on Telecommunications and Computer Networks (SoftCOM)*, pp.1-6, 2011.
- [2] Xinquan Jiao, Peijiao Ma, Yuguang Zhang, Yiran Wei and Yongfeng Ren , "ECC algorithm and the realization of the VHDL for high-capacity solid-state storage", *International Conference on Electronics and Optoelectronics (ICEOE)*, Vol. 3, pp. V3-239 - V3-242, 2011.
- [3] Wen shi Chen and Chunxiao Liu , "The Applied Research of ECC Encryption Algorithm in VPN Technology" *International Conference on Internet Technology and Applications (iTAP)*, pp.1-4, 2011.
- [4] Peng Gong, Feng-jiao Qiu and Meng Liu, "A new algorithm based on DES and ECC for CSCW", *International Conference on Computer Supported Cooperative Work in Design*, Vol. 1, pp. 481 - 486, 2004.
- [5] Fan Zhang and Zhijie Jerry Shi, "An Efficient Window-Based Countermeasure to Power Analysis of ECC Algorithms" *Fifth International Conference on Information Technology: New Generations, ITNG 2008*, pp. 120 – 126, 2008.
- [6] Andrew S. Tannenbaum, "*Distributed Computing*", Prentice Hall, 2005.
- [7] Pradeep K. Sinha, "*Distributed Operating Systems*" Prentice Hall, 1997.
- [8] Vijay K. Garg, "*Elements of Distributed Computing*" Prentice Hall 2000.
- [9] George F. Coulouris, Jean Dollimore and Tim Kindberg "*Distributed Systems Concepts and Design*" Prentice Hall.
- [10] Ruay-Shiung Chang, Hui-Ping Chang and Yun-Ting Wang, "A Dynamic Weighted Data Replication Strategy in Data Grids", *International Conference on Computer systems and Applications, IEEE/ACS*, pp. 414-421, 2008.
- [11] Kumar, R. and Ross, K.W, "Peer-Assisted File Distribution: The Minimum Distribution Time", *1<sup>st</sup> IEEE Workshop on Hot Topics in Web Systems and Technologies*, pp. 1-11, 2006.
- [12] Dalessandro, D. and Wyckoff, P, "Fast scalable file distribution over Infiniband", *Proceedings. 19th IEEE International on Parallel and Distributed Processing Symposium*, pp. 8, 2005.
- [13] Ma Lingjun, Pui-Sze Tsang and King-Shan Lui, "Improving file distribution performance by grouping in peer-to-peer networks", *IEEE Transactions on Network and Service Management*, Vol. 6 , No. 3 , pp. 149 - 162 , 2009.
- [14] Yi Xiao-lin and Yang Feng, "Research and Improvement of Embedded System ECC Algorithm", *1st International Conference on Information Science and Engineering (ICISE)*, pp. 1801 - 1804, 2009.
- [15] Ankolekar, P.P., Isaac, R. and Bredow, J.W. "Multibit Error-Correction Methods for Latency-Constrained Flash Memory Systems", *IEEE Transactions on Device and Materials Reliability*, Vol. 10, No. 1, pp. 33 - 39, 2010.
- [16] [www.wikipedia.org](http://www.wikipedia.org)
- [17] <http://distributedcomputing.info/>
- [18] [www.distributed Computing.com](http://www.distributed.com)
- [19] <http://philip.greenspun.com/seia/distributedcomputing>
- [20] <http://www.bacchae.co.uk/docs/dist.html>
- [21] <http://www.opengroup.org/dce/>
- [22] <http://www.dcia.info/>