Volume 6, Issue 4, April 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

P2P Indexed Based Dynamic Road Network

N. Pavithra*
M.Phil Scholar, Department of Computer Science,
Shrimati Indira Gandhi College, Trichy,
Tamilnadu, India

R.Chitra
Asst. Prof, Department of Computer Science,
Shrimati Indira Gandhi College, Trichy,
Tamilnadu, India

Abstract— P2P applications emerged as a powerful paradigm for sharing information across the internet. They cause much of web traffic. The unorganized P2P frameworks typically use a restricted flooding technique for searching. Observations have appeared that a few of peers offer most of information in a record sharing system. In request to increment the victory rate of blind look furthermore, load balancing, replication strategies are utilized in these systems. This paper presents an improved information replication technique for unadulterated unorganized P2P record sharing frameworks such as Gnutella, based on ubiquity of files. Proposed technique employments both of Document furthermore, record replication. Reproduction results appear that this technique has the best execution than some others, although it have a bit more cost.

Keywords— P2P, Information Replication, Unstructured, Record Popularity.

I. INTRODUCTION

P2P frameworks were classified by in to three different categories. Some P2P systems, such as Napster, are centralized in that they have a focal directory server to which clients can submit queries. Other P2P frameworks are decentralized furthermore, have no focal server; the hosts form an ad hoc framework furthermore, send their questions to other peers. Of these decentralized designs, some are organized in that they have close coupling between the P2P framework topology furthermore, the area of data. Other decentralized P2P systems, such as Gnutella are unorganized with no coupling between topology furthermore, information location. In these systems, peers use the restricted flooding framework (to look the asset of the framework furthermore, directly exchange this resource. This model do not lead to the paralysis of the entire framework since of the failure of some hubs in the network, but the flooding look framework has poor efficiency, furthermore, will bring a parcel of power exponential growth of the demAlso, message number amid the process of the asset search.

Information replication consists of maintaining numerous copies of data, called "replicas" on separate computers. Replication is broadly utilized in appropriated information management with read-intensive workloads for a number of reasons such as:

1. P2P framework is a dynamic self-organized network, in which peer can freely join or leave, so there will lost a parcel of critical information when some critical hubs fail. This feature is not great to the expansion of the P2P application, so replication is utilized to enhance the reliability of the network.
2. The unorganized P2P frameworks typically use a blind flooding technique for searching. Where a question initiator sends demands to all or a haphazardly picked subset of framework neighbors furthermore, these demands are forwarded up to some predetermined depth or when a preset time-to-live (TTL) expires. Implementation of flooding is straightforward but it does not ensure finding the asked information item. In request to increment the victory rate of blind look furthermore, information availability, replication strategies are needed in these systems.
3. In a P2P system, if there is a substantial number of download demands furthermore, not so numerous peers are sharing the desired file(s), some peers will become overloaded with download demands that could be satisfied by other peers which host imitations of the asked file(s). By imitating the prevalent records at more than one peer, access demands have a choice among numerous replicas; load balance can be improved furthermore, results in higher throughput furthermore, shorter reaction times.

The rest of the paper is organized as follows: Segment 2 presents the related work. In segment 3 we propose an improved replication strategy, in segment 4 we present a reproduction model, Segment 5 presents the reproduction results furthermore, comparison of our proposed methodology with other strategies, and furthermore, finally in segment 6 we conclude the paper.

II. RELATED WORK

Replication has been much studied in organized peer-to peer systems. For instance, Beehive furthermore, EpiChord are approaches custom-made for DHT routing mechanisms or proposed systems in. But most of their approaches are not applicable to unorganized shared systems.

A broadly recognized analysis of replication in unorganized P2P systems is the work of Cohen furthermore, Shenker. They investigate the question which replication methodology minimizes the expected number of peers that have to be sampled until the asked information items are found (or question size). The main constraint in this setting is the limit capacities of peers. This approach just tries to minimize maximum question size by imitating pointers, instead of adjusting load or reducing normal download time by imitating files.

Sozio et al. extend the approach of by explicitly addressing the question where to place replicas. They propose a appropriated replication calculation that was custom-made to realistic look protocols furthermore, accomplished nearoptimal imitation placements with a proven execution guarantee.

Another interesting result is the work of Tsoumakos et al.. Their approach couples lookup indices along with an aging framework in request to identify, in genuine time, question intensive areas inside the unorganized overlay. Peers then individually choose on the time furthermore, extent of replication, based on local workload computation. APRE adaptively expands or contracts the imitation set of an object in request to improve the sharing process furthermore, achieve a low load appropriation among the providers. The scheme proves particularly useful in the event of flash crowds, managing to adapt quickly to sudden surges in load process.

In authors proposed a decentralized model for dynamic creation of imitations in a decentralized P2P network. The aim was to ensure a determined degree of information availability.

Paper proposes an appropriated replication framework for decentralized unorganized P2P systems which consider content popularity. A Document of the prevalent content will be allocated on the hub midway between the requester furthermore, the provider. The hub counts the number of references to each Document that it has, when the number surpasses a certain threshold; an imitation of the content is put instead of Document at this moment. This approach is very straightforward but it is more efficient to decrease framework activity furthermore, increment look victory ratio than decrease download time.

Paper presents an arrangement for the problem of load adjusting in shared record sharing systems, based on the automated replication of records into “good” peers. Among the possible candidates to host an imitation the best ones Agreeing to their bandwidth furthermore, accessibility is chosen. This approach considers appropriate factors to pick imitation host.

Requester replication furthermore, way replication have been utilized by Gnutella furthermore, Freenet systems. In requester replication strategy, when a look is successful, the object is imitated furthermore, stored at the requester hub only. In way replication strategy, when a look succeeds, the object is stored at all hubs along the way from requestor hub to the supplier node. In reality Gnutella implements passive replication, a record is only imitated at hubs requesting the record whereas Freenet allow proactive replications, where the object may be imitated at a hub though the hub has not asked the object.

III. OUR REPLICATION METHOD

In this work we focus on unorganized P2P systems like Gnutella. Information shared here accepted to be perused only. In recent P2P record sharing frameworks the information is moreover perused only. In this way the clients offer records such as music, films, books, etc, which won't change in future.

The proposed arrangement for replication is based on ubiquity of files. Peers measure the record ubiquity by counting the number of requests. When a peer gets a demand so, for a record that it is capable to give it (or its index), it increases ubiquity of that file. In this work we attempt to decrease load of prevalent records supplier hubs by imitating a duplicate of them in other nodes, furthermore, attempt to increment fruitful look rate of center prevalent records by imitating their Document in other nodes. So we consider two limits T1 furthermore, T2 for ubiquity ($T1<T2$).

The replication framework is divided into four critical subjects: which record should be replicated, when to perform replication, where the new imitations should be put furthermore, how imitations are destroyed (imitation substitution policy).

3.1 What should be Replicated?

Allocating more imitations of more prevalent substance brings change of look victory as a whole since prevalent substance are more often searched. It must be moreover a benefit to decrease framework activity since the number of jumps to reach a sought content becomes little.

The size of a Document (Addresse of requester furthermore, provider) is much littler than that file; accordingly the cost of keeping a Document is much littler than the one of copying furthermore, storing the file.

3.2 When should be Replicated?

When a peer gets a demAlso, for a record that is capable to give it (or its index), will increment ubiquity of that file, then it checks, if ubiquity of that record surpasses some limit in a period, the peer will choose to reproduce the record (or its index).

3.3 Where should be Replicated?

Our technique for replication is to simply reproduce a record (or its index) along the way that a question has followed from requester to provider. The center hub located between requester furthermore, supplier of a record is the best place to reproduce its index, since the whole of jumps from Document will be minimized. Fig. 1 appears the reason why the midway hub is the best to place a replica. The whole of the bounce numbers will be minimized if a imitation is put on the hub midway the look path. This is more effective than probable allocation.

In an ideal world, each user shares or downloads the files, stays associated to the P2P framework for a long time, in practice these frameworks have numerous problems like having a substantial number of low bandwidth or low accessibility peers. Low bandwidth peers, such as modem users, are not capable to act as servers furthermore, are not capable to let other peers download their records. Low accessibility peers are peers that stay associated to the P2P framework only for short periods at a time. These peers typically interface to the framework to download a few records furthermore, then disconnect. Measurements appear that low bandwidth peers, stay associated to the P2P framework for a shorter time. Therefor among the candidate hubs (the hubs between supplier furthermore, requester), the best for hosting imitation is picked based on bandwidth.

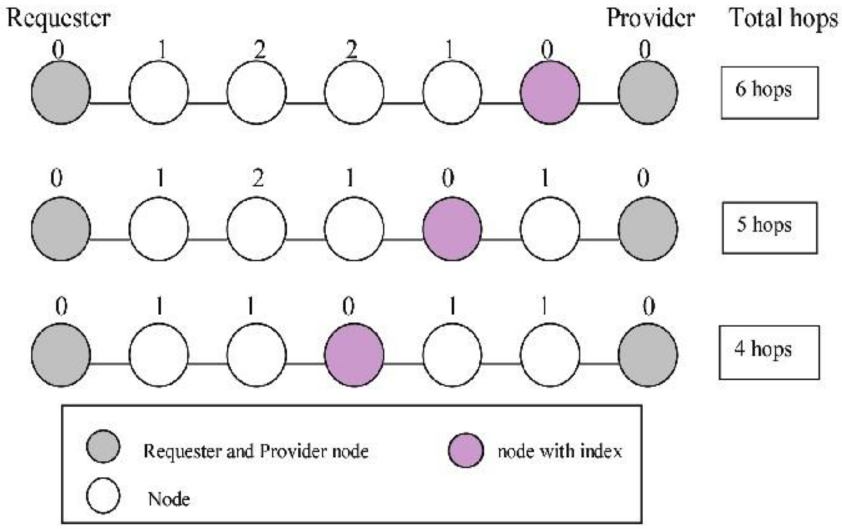

Fig. 1. Document replication on look path

When a hub gets a demAlso, that it is capable to give it, investigates the demands history for the asked file, if the number of demands surpasses T2 then a duplicate of record will be imitated in another node, but if the number of demands is more than T1 furthermore, less than T2 an Document of asked record will be imitated in another node.

3.4 Imitation Substitution Arrangement

Limit of peers is restricted to keep imitated records or records so a imitation substitution arrangement is essential to decide which stored imitations should be reput with the new imitation in the shortage of limit spase at the selected node. We use Least Recently Utilized (LRU) to delete the imitations or indexes. In reality we use temporary locality utilized in Gnutella network.

The complete replication calculation is given in Fig. 2.

IV. REPRODUCTION SETUP

To simulate our proposed replication technique we utilized C# language programming with SQL database. Our simulator is a Query-Cycle simulator. A reproduction process proceeds in a question cycle. In each question cycle, a peer i in the framework may be actively issuing a query. Upon issuing a query, a peer waits for incoming responses, selects a download source among those hubs that respond furthermore, starts downloading the file. The question cycle finishes when all peers who have issued questions download a satisfactory response. Statistics may be collected at each peer, such as ubiquity of asked file.

4.1 Framework Model

We accept a unadulterated Shared model like Gnutella furthermore, ignore physical connectivity. In a decentralized unorganized framework like Gnutella, each hub typically connects to 3 other nodes. For straightforwardness like we moreover use a network n×n structure to model framework topology since the normal of network associations (neighbors of each node) is between 3 furthermore, 4. We accept that hubs furthermore, their associations are fixed furthermore, will never change amid simulation. We consider a network with 50×50 dimensions that it contains 2500 nodes.

4.2 Content furthermore, Question Appropriation

For straightforwardness like, we moreover consider all records the same size (4MB). In the first of reproduction we have 6000 files, disseminated among hubs randomly. Like we consider five levels for record popularity. It has been watched in P2P record sharing frameworks that questions follow a zipf appropriation. Agreeing to this distribution, records with more ubiquity are more searched. Record ubiquity levels furthermore, their dem also, rate appeared in Table 1 which follows zipf distribution. Agreeing to these levels limits are as follows T1=3 furthermore, T2=5, Limits for algorithms in, are moreover 3 furthermore, 5 respectively.

```
if each peer P receives query q for item i from R and TTL >= 1 then{
   if P contains i then  {
      R Accesses i from P;
    count of accesses for i in P increase 1;
     if count of accesses for i >= T2 then {
       s1= the peer is between R and P that it has the best bandwidth;
       if  size of i <= s1 available storage size then {
                 replicate a copy of i in s1;
            exit;
       }//end if
      else {
         delete some files from s1's replica storage using LRU;
         replicate a copy of i in s1;
         exit;
         }//end else
     }// end if
     else if  T1<= count of accesses for i <T2 then{
         s2= the peer is halfway between R and P;
         if s2 has enough index storage then{
               replicate an index of i in s2;
             exit;
         }//end if
       else {
            delete an index from index storage of s2 using LRU;
           replicate an index of i in s2;
           exit;
       }//end else
     }// end else if
 }//end if
else if an index of i is available in P then {
     count of accesses for i in P increase 1;
     if access count of index >= T2 then{
          replicate a copy of i from P, instead index;
      }//end if
 }//end elseif
 else{
    forward the q to all neighbors except the node which sent the query;
    TTL = TTL – 1;
 }//end else
}//end if
```

Fig.2 Replication Algorithm

Table 1 Ubiquity Level Furthermore, Demalso, Rate Of Files

Popularity	Request rate (%)	Number of Files
(Low) 1	5	3500
2	10	1300
(Middle) 3	20	700
4	30	350
(High) 5	40	150

4.3 Bandwidth

We have a straightforward understanding of a peer’s bandwidth in our simulations: Bandwidth at a peer is consumed only while uploading or downloading files. Bandwidth is assigned to peers based on measurements in. These observations appear that 8% of Gnutella clients interface with dial up (64 kbps or less), 60% of clients interface by broad-bfurthermore, associations (cable, DSL, T1, furthermore, T3) furthermore, 30% moreover have high bandwidth associations (3Mbps furthermore, more). Fig. 3 appears the empirical CDF utilized in the recreations to model the bottleneck bandwidth of the peers. The picked CDF reflects genuine life bottleneck bandwidths watched in Gnutella.

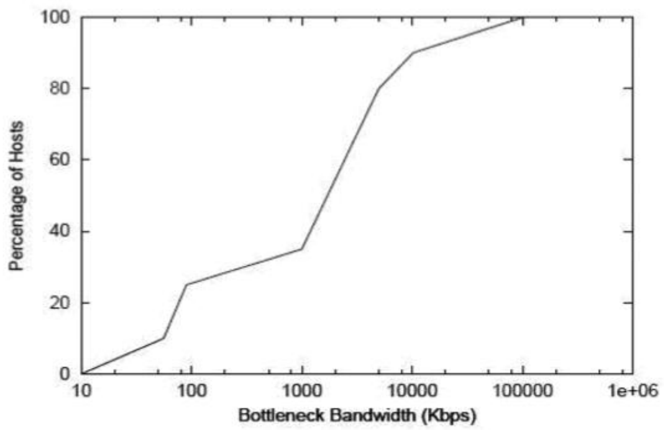

Fig. 3. CDF of bottleneck bandwidths for Gnutella peers

We accept that each peer has a restricted bandwidth limit to download furthermore, transfer files. Download furthermore, transfer limit is accepted to be 1. That means there is only one download or transfer operation that can be executed at a time. Table 2 presents the values of the other parameters utilized in the simulations.

Table 2 Reproduction Parameters

Simulation Parameter	Value
TTL	5
Maximum count of replicas that each node can holds (Replica Storage)	100
Maximum count of indexs that each node can holds (Index Storage)	1000
Number of queries in every experiment	100,000

V. REPRODUCTION RESULTS

To assess our proposed replication strategy, we pick these execution metrics: normal download time, normal of jumps for look furthermore, victory rate of queries. We compare our methodology with requester methodology furthermore, proposed systems in , since we believe that our methodology enhances them. We moreover present the costs for all of these strategies. Fig. 4 appears normal download time for mentioned strategies. Normal download time is calculated at each question cycle. It's clear that our methodology has a better execution over other strategies.

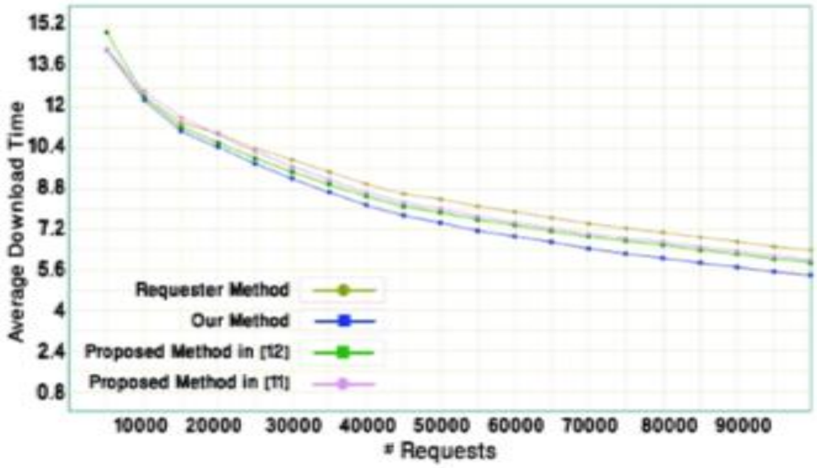

Fig. 4. Normal Download Time

Fig. 5 appears fruitful searches. If a asked information exist furthermore, it can be found then that look is fruitful otherwise unsuccessful.

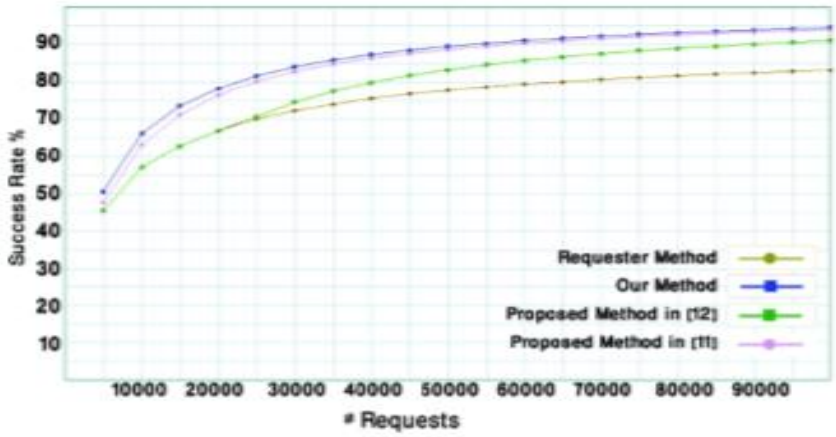

Fig. 5. Victory Rate of Search

Fig. 6 appears normal of look hops. When a hub re- quests a file, if the hub itself has the record the number of jumps is equal with zero, if record is found in another hub then the number is equal with whole of visited hubs between re- quester furthermore, supplier plus 1furthermore, if the look is unfruitful the number of jumps will equal TTL. As the Fig. 5 furthermore, Fig. 6 show, our methodology works better than others furthermore, methodology works better than since methodology focus on change of look productivity by indexing replication but methodology tries to improve normal download times by imitating the prevalent records in high bandwidth nodes.

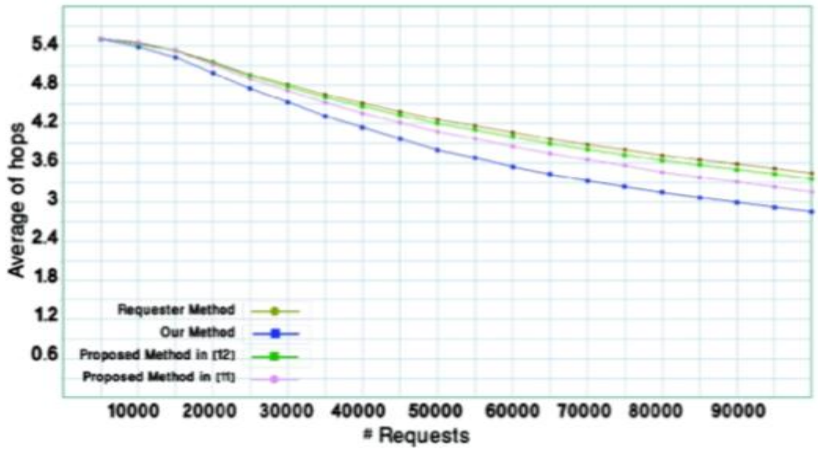

Fig. 6. Normal of Hops

Fig. 7 appears normal number of imitated files. We pre- sent this chart to assess the cost of replication for our strategy. It's clear the more the number of imitated records the more the cost of storage. As the Fig. 7 shows, our methodology has worst cost but difference of maximum point of our strategy's bend furthermore, green bend is a little value, accordingly cost of our methodology is not so much more than, strategies.

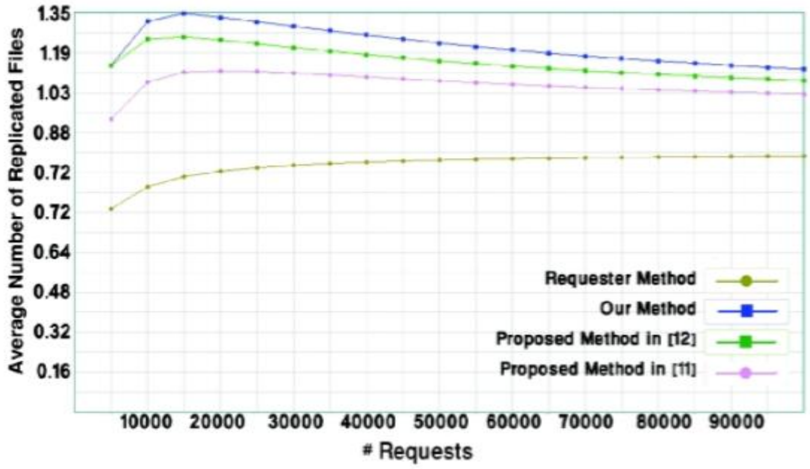

Fig. 7. Methodology

VI. CONCLUSION FURTHERMORE, FUTURE WORKS

In this paper, we proposed a pro-active novel methodology for information replication in unadulterated decentralized unorganized P2P systems. Our approach can be utilized in unorganized P2P systems such as Gnutella.

This technique is decentralized furthermore, combines information replication with Document replication furthermore, employments two limits for popularity.

Through the reproduction results, this methodology has a better execution in terms of victory rate, normal download time furthermore, normal number of look bounce analyzed with other strategies. Although, the proposed methodology might use more limit for the replication of index, the overhead is considered to be little analyzed to the benefits accomplished from its performance.

We tested our methodology with flooding search. There are some other strategies for look in P2P frameworks such as random walker furthermore, expanding ring so Future works are as follows. First, respect of ubiquity of record furthermore, its number of imitations must be investigated. Second, other P2P look strategies must be examined as well as flooding. Third, we are going to expect, proposed methodology to use in organized P2P systems.

REFERENCES

[1] Qaisar Rasool; Jianzhong Li; George S. Oreku; Shuo Zhang; Donghua Yang, "A load balancing replica placement strategy in Data Grid", Digital Information Management, 2008. ICDIM 2008. Third International Conference on, Year: 2008, Pages: 751 – 756.

[2] Anna Saro Vijendran; S. Thavamani, "Least Recently Used Replica Replacement Technique in Distributed Computing Network", Intelligent Computing Applications (ICICA), 2014 International Conference on, Year: 2014, Pages: 104 – 108.

[3] Wen-xiang Li; Zhao-jun Du; Zhi-chao Sheng; Yan-li Zhu; Tao Hu, "A Proximity-Aware Technique for Distributing Replicas in DHT-Based P2PNetworks", Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on, Year: 2009, Pages: 1 – 4.

[4] Szu-chi Wang; Hong-zu Chou; David S. L. Wei; Sy-yen Kuo, "On the fundamental performance limits of peer-to-peer data replication in wireless ad hoc networks", IEEE Journal on Selected Areas in Communications Year: 2007, Volume: 25, Issue: 1, Pages: 211 – 221.

[5] C. P. de Laborda; C. Popfinger; S. Conrad, "Dynamic intra- and inter-enterprise collaboration using an enhanced multi database architecture", 16th International Workshop on Database and Expert Systems Applications (DEXA'05)Year: 2005, Pages: 626 – 631.

[6] Jemal H. Abawajy; Mustafa Mat Deris, "Data Replication Approach with Consistency Guarantee for Data Grid", IEEE Transactions on Computers, Year: 2014, Volume: 63, Issue: 12, Pages: 2975 – 2987.

[7] Fatos Xhafa, "Data Replication and Synchronization in P2P Collaborative Systems", 2012 IEEE 26th International Conference on Advanced Information Networking and Applications, Year: 2012, Pages: 7 – 7.

[8] Dharma Nukarapu; Bin Tang; Liqiang Wang; Shiyong Lu, "Data Replication in Data Intensive Scientific Applications with Performance Guarantee" IEEE Transactions on Parallel and Distributed Systems, Year: 2011, Volume: 22, Issue: 8, Pages: 1299 – 1306.

[9] Xuejun Zhuo; Qinghua Li; Wei Gao; Guohong Cao; Yiqi Dai, "Contact duration aware data replication in Delay Tolerant Networks", 2011 19th IEEE International Conference on Network Protocols, Year: 2011, Pages: 236 – 245.

[10] Jilun Miao; Jiayu Zhou; Yongxiang Zhang, "The CE-SE Unstructure Technique Applied to Flow Simulation in Mountain River", Computational and Information Sciences (ICCIS), 2011 International Conference on, Year: 2011, Pages: 897 – 900.

