IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-04 E-ISSN: 2347-2693

An Adaptive Replication Approach for Relocation Services in Data Intensive Grid Environment

P.Sunil Gavaskar

Department of Computer Science and Engineering,
Sri Venkateswara University, Tirupathi, Andhra Pradesh, India

www.ijcseonline.org

Received: Mar/21/2016 Revised: Mar/30/2016 Accepted: Apr/17/2016 Published: Apr/30/2016

Abstract— **In the Data Intensive grid environment, researchers always try to avoid failures by improving the data availability with mechanism called replication. In this paper, data availability which is shown works towards efficient way of applying Object replication and its object replicas availability prediction and thus users are able to predict its dynamic replication status that decides replicas management and its performance in data grid. The proposed way of Availability prediction gives status of data availability at nodes and thus it gives data reliability information to the scheduler, in such manner our proposed scheme helps to make better jobs execution decisions with minimum jobs execution time and band width by considering account of some of the objective functions involved while data access. It is an eminent method to deal with object replicas utilization that can be improved jobs execution performance, and proposed method is an auspicious improvement over performance of replicas utilization without failures in data grid environment.**

Keywords- Replication, Fault tolerance, Data grid, Dynamic nature of data grid, tree-based replica location service, restrictions on security issues in grid, Resource utilization in data grid.

I. INTRODUCTION

A data grid is an architecture or set of services that gives effective solutions to individual user or group of users, where ability to access, modify and to transfer large volume of data over geographically distributed networks which is required for research activities [1]. The trend in data grids is continuously growing with its size of the data that is managed by data grid, and it has already reached Peta bytes, as in the Atlas Project Data store [2]. In data grid there exists two basic data management services: that are services for data transfer, and services for replica management. Therefore the main service for data management is the Grid FTP protocol , and it is an extension of FTP that supplies efficient and secure data transfer and then provides access to large files in grid environment[3].

Data replication becomes a practical and effective method in order to achieve highly efficient access and dependable data access in grid environment. The data replication technique uses dynamic replication schemes. The dynamic kind of data replication is to enrich and to provide enrich method in order to automatically manage replicas. Then this is followed in the status changing of job execution, system parameters and user access patterns, proposed in [4,5].

The data in a data grid can be located at a single site or multiple sites. In such kind of arrangement each site has its administrative rules that give restrictions on Job security conditions of resources status and that helps to predict data availability status. Such kind of Pre-Measurable data grid is applicable to users where they try to access the data dynamically. In such fashion, multiple replicas that belong to data grid are to be distributed in entire grid sites and outside of it. Finally administrative sites in data grid and critical access constraints are framed on security of data placement, So the original data access between sites are performed based on access limits and it must be equally applied to all the replicas. The tree-based replica location scheme (TRLS) is to decide the replica locations service, which is proposed in [6]. The main intension of TRLS is to diminish the total communication cost and storage cost of the replication that is involved in data grid.

Main importance to use of the fault tolerance, replication techniques are to reduce delay in data access thereby the job execution cost also reduced. The bandwidth, job execution costs are not only depending on the computational resource assignment but also on the location of data replica files that are required by the jobs [7].Data replication has long been considered as an important technique to reduce data transfer rates that improves data access performance in data grids, it is an approach noticeably to reduce the network bandwidth consumption and the access latency [8].

II. RELATED WORK

The tree-based replica location scheme (TRLS) is to decide and reduce the cost of data access and thus objective performance of replication schemes are achieved with other resources that are involved in grid.

*Corresponding Author:
P.SunilGavaskar
e-mail: sunil079@ gmail.com, Tel.: +91-08572-258274

Job execution time is another very important parameter. Some replication strategies always target to minimize the job execution time with optimal replica placement. Such strategies can reduces delay while data access in gird thereby minimizes the access time and thus job execution time becomes reduced. Such methods will increase the throughput of the system.

The goal of the fault tolerance in grid is to maximize data availability with the help of replication strategies and that reduces the cost of other resources that are involved in grid. A well defined fault tolerance approach can improve the Quality of Service of overall grid [7].

A data replication process involves creating identical copies of a file and placing them onto multiple sites so that they can be accessed simultaneously from various locations. Three main decisions need to be made by a replication mechanism: which files to replicate, when to replicate, and where to replicate [9].

Another important factor is that the site has the ability to give accommodation for replicas; the jobs which are related to that site can be executed in speedup manner, otherwise: replicas stored in nearby sites can be accessed based on locality of reference methods. All the times popularity of replicas could be not same, so popularity becomes reduced such replicas are to be deleted in order to provide storage space and to enhance feasibility of replicas access [10].

Efficient Dynamic Replication Algorithm (EDRA) always prefers to choose best replica based on availability status, workload of the node and available bandwidth, and then computing capacity of the nodes.
Dynamic replica strategy, here replicas creation happens on the basis of number of file access and node load. The selection of replicas and replacement of replicas depends on creation time, number of file access and this kind of strategy provides low response time and rapid data download and good performance [12].

In earlier approaches, they did not consider the Dynamic nature of updatable data access cost of the grid and the assumption is that all the replicas are available throughout the process of replication. Hence proposed method provides dynamic nature of replicas and its status prediction that decides feasible sites for jobs execution.

III. PROPOSED WORK

A. *Proposed methodology*

In data grid environment file of replicas movement over networks needs certain performance regulation in turn to achieve finest rates of data movements. As part of that the proposed object replication methodology can be applicable while managing replicas in data grid environment and also it addresses some object replication issues that come at the time of data access. The replica used under each sites involves replicas management and is as follows.

The existing schemes like Lru may cause the valuable old file lost, Lfu may lead to the lost of file recently duplicated.
Lru: Scheme that always replicates, it deletes least recently created files
Lfu: Scheme which always replicates, it deletes least frequently accessed files.

Therefore Object replication scheme is proposed .A single file will generally contain many objects. This is necessary because the number of objects become so large. Moreover, the object persistency solutions used so far are only work efficiently if there are many objects per file and there exists limitations in accessing file of replicas. To map the high-level object view of our experiment to the lower level storage infrastructure of replicated files, we perform the following replicas placement and replacement.

B. *Replicas Placement and Replacement*

Similarly the proposed scheme reduces the disadvantages that are occurring in branch replication scheme. In branch replication root node replicas are divided into disjoint sets, so that leaf nod having these disjoints sets, in parallel applications we need to access entire or all the leaf nodes. Unlike branch replication scheme, our proposed strategy of object replication almost avoids accessing of unwanted/update less replicas (i.e. files that are supporting less access rates of object replicas) into computation, from lowest access rate replicas over its leaf nodes. The following representation shows that files occupy nodes based on the division of file $_m$ into all supporting nodes.

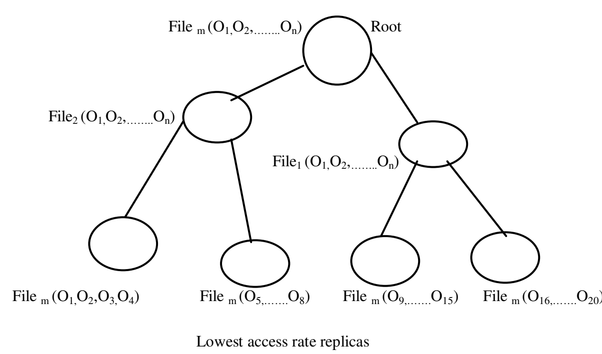

Figure1: Hierarchical representation of replicas model

By considering the above shown model, replicas are placed according to their access rates priority, In this model highest access rate replicas are placed in file$_1$ and the lowest access rate replicas compare to previous file is placed in File$_2$, File$_3$

and it is continued up to File $_m$ .The measures taken from the above model and arrangement of replicas are as follows:

In the proposed scheme, any one of the leaf nodes or non-leaf nodes file uses low access rate replicas and such categories of unwanted or less access rate object replicas are moved to one file and same is placed probably under leaf nodes. Hence we can avoid number of traversal for replicas thereby nodes which are having highest access rate object replicas file will be considered into account while at parallel application starts its execution. The lowest access rate replicas are probably applicable for deletion, that is based on locality of reference .Suppose when access rates and update are increased on lowest access rate replicas and same is replaced according to its feasibility. The feasibility indicates files supportability, number of request on particular replicas and other measures.

Replicas Replacement:

Lowest access rate replicas are removed, unlike the other replication methods in file based object replicas method, instead of removing the entire file, the particular file use to remove the lowest access rate object replicas those comes under min-threshold of access rate or that causes job failures .

In $File_{M+1}$, based on replacement methods that it work; we have that the status copy of replicas, therefore search rate is also reduced and search rate can be detailed in my previous papers.

C. Modeling Terms and Equation

Step1.The Different files, same replica and different requests are used for selection of replicas.

The Different files, same replica and different requests are used for selection of replicas.

The Number files replicas are used to maintain object replicas are 'M': where 'M' is the number of files used as replicas and which starts from F_1 to F_M. Therefore Object replicas are part of file replicas.

$File_1 O_1 . r_1 + File_2 O_1 . r_2 + File_3 O_1 . r_3 + \ldots\ldots\ldots\ldots File_m O_1 . r_i$ (1).

$$\sum_1^m File_m \sum_1^j O_{n\,rj}$$

Different request on the same replicas which are presented in different files, then there is presence of Period of replication (P_r) and Overhead of replication (O_r) due to the same replica modified at same time. Therefore, in later the proposed search rate method that gives solution towards updates requests which happens on the same replicas.

Step2. In a same file the different replicas with different requests are used for selection of replicas that shows the wait states.

$File_1 O_1 . r_1 + File_1 O_2 . r_2 + File_1 O_3 . r_3 + \ldots\ldots\ldots\ldots File_1 O_n . r_j$

$$\sum_1^{n,j} O_n \, r_j \, File_m \qquad (2)$$

Number of replicas present in entire model is considered as 'n': where 'n' is the number of replicas and which starts from O_1 to O_n. The Number of files is used to maintain an object replica, which ends with 'M': where 'M' is the number of files used and which starts from F_1 to F_M. Therefore Object replicas are also part of file of replicas.

The file $F_{m+1,}$ is used to maintain the value less replicas, which are obtained from F_1 to F_M. Therefore reduced access rate replicas and value less replicas which are obtained from F_1 to F_M are moved to F_{m+1}, as part of that some replicas uses group of F_{m+1}files or file, there is no restriction over the number of access rates elements to be used (i.e unlike model shown above, the file F_{m+1} can also uses additional access rate element when compared to other files that are involved in replication).

The replicas placed in F_{m+1} can be probably deleted due to its replicas lowest access rates or according to their access rates some times when 'such replicas are not required', and are moved to other locations as part of replicas replacement.

IV. ALGORITHM

The procedural arrangement of replicas and access rates are shown in the form of algorithm and its complexity measure.

Step1: Find the access rate of replicas and place it in 'A_r' (for all the access rate of replicas);

Step2: Counts the number of access rates that present in A_r (i.e count the access rates used in particular instance): we consider Number of replicas 'n', and it leads to elements in $(A_r) \in n$.

Step3: Predict the number of files required to maintain object replicas according to the access rates of replicas: Therefore Number of Files is 'M', $\{n \in M\}$

Number of files required to place replicas involved in replication (F_m)

=Number count of replicas access rate categories / 2

Step4: Arrange the access rates equally according to the feasibility such as highest access rate replicas to be occupied into files present at first, followed by lowest access rate replicas in next levels of files in a tree based structure of replicas model.

The proposed model have possibility of the {(A_r) ϵ (∀n)} ϵ (∀M)} categories of replications. The following model shows the framing of object replicas into files based on their access rates shown in A_r array. The following arrangement shows such kind of replication arrangement.

$File_1$(O1,O2,………..On)------> Highest access rate supporting replicas

$File_2$(O1,O2,………..On) ------> Replicas that supports Lesser access rate than previous file

$File_3$(O1,O2,………..On) ------> Replicas that supports Lesser access rate than previous file

$File_m$(O1,O2,………..On) ------>Lowest access rate supporting replicas

Step.5: To arrange the replicas based on their access rate we perform sorting of replicas access rate which is present in A_r.

5.1. The sorted array (A_r) of access rate can be divided according to strength of files.

Number count of access rates of replicas used in each file (N_{Ac}) = (Total element in access rates Ar) / (FM files)

5.2. In an Array A_r is having an equal access rate

5.2.1 For equal access rates ***If*** (Number count of replicas are same), If number count is also same: find such replicas and move to different files as part of replacement of replicas: /* The access rates and number counts of replicas are same then the replication concept can be violated and also it causes equal access rate replicas can be scattered at one node or in one place so that the disadvantage behind popularity based replication cannot be completely avoided */.

Else: equal access rates, but there is no number count of replicas are same then there is no need of replacement is required and leave it in same place /

5.3. Place all the access rates elements which are sorted and present in Ar array according to the access rate.

5.3.1. Highest access rate supporting elements at File1 and then access rate elements that supports lesser access rate than previous file is placed in file2; this is to be continued until last before file called $file_{m-1}$ and then followed by lowest access rate supporting elements at final stages 'F_m'.

Step.6: Divide the total files from fil_1 to $file_m$ according to access rate A_r and place replicas in each file according to their sorted access rate A_r.

6.1. Highest access rate supporting replicas are at File1 and then replicas that supports lesser access rate than previous file is present in file2, this is continued until last before file and then followed by lowest access rate supporting replicas at final stages.

Step.7: Movement of replicas when access rate gets down: (i.e. replicas are moved to M+1 file)

Movement of replicas present in all the files to lowest access rate supporting file based on its threshold value when the access rate gets down, and this happens irrespective of access rate elements used in step5 and step6.

$File_{m+1}$(O1,O2,………..On) ------> is a file with lowest access rate supporting replicas and not valuable replicas are moved from all the files into one file called $File_{m+1}$ .The $File_{m+1}$ can maintains the access rate count but there is no restriction over access rate elements present in $File_{m+1.}$

The $File_{m+1}$ is used to placement of replicas which are having not majority of usage (i.e. not valuable) in nature that are carried out based on their access rate count or when the access rate count of replicas goes down. Moreover all the files have specific min-threshold, therefore access rate of replicas downs when compare to specific threshold then the replicas supporting files from $file_1$ to $file_m$ are moved to $File_{m+1.}$ As result of that $File_{m+1}$ always supports the not valuable replicas and probably such kind of replicas comes under the deletion categories.

It requires **log (n)** divisions to split an array of size 'n' into single elements.

Therefore it require O(**log(n)**) Iterations and each iteration can takes O(n) times and then the run time complexity algorithm is O(n **log(n)**),Where 'n' is the number of elements present in the access rate count A_r array.

In proposed access rate based replicas deletion algorithm, the entire complexity of the algorithm is based on the counts of access rate sorted order and thus once the sorting complexity is reduced then we place and move the replicas according to the order of movement specified in step7 of our algorithm which is said above.

V. SCHEDULE: ACCESS RATES BASED JS PARAMETE

Based nodes capability and availability status; there is presence of replicas replacement, selection and thus there is presence of JS parameter for jobs to perform feasible job run.

Condition1: Initially job security to be predicted using the reliability/cost estimation of the replicas in all nodes, (i.e. when the job security with cost estimation is measured for the required resources are within the range of JS, that indicates reliability of nodes = {between 0 to n}), The JS is divided into 3 stages. According to its reliability status as stated earlier, the reliability of cost estimation ranges from '0' to 'n' and it can be changed dynamically between 0 to n, if a job run on job 'X' submitted with some initial cost estimation, that posses through with following condition.

While $JS^X \leq Nodes^Y_{(a,b,c,d).}$ **Then** there exists no failures in cost estimated nodes (i.e. with respect to distance and Communication cost all are acceptable)

Where Y indicates corresponding node with key entries involved in computing job 'X', for secure job run: The probability of job 'X' submitted on node y should be higher than '0',so there is presence of secure job run then the optimality of algorithm is trivially true.

For leaf nodes there exists availability /unavailability of replicas (due to lowest access rate replicas present at leaf nodes) and its feasible job run with different nodes

Replicas which are used at leaf nodes comes under the lowest access rate replicas because leaf node distance from roots or Frs servers that always causes delay and more bandwidth utilization while replication: Therefore based on the Access rate (Ar) of replicas, have to be moved to higher-level nodes (i.e nearer to the Frs).and we use to predict JS based on the Frs servers delay and accessing replicas ,such kind of JS status is maintained in Resource Broker(RB) and is useful while jobs migrated between nodes .Therefore distance becomes reduced, while performing summation of rooted sub tree cost at the children of 'n', then the communication cost 'Cc' is also reduced in order to access replicas.

Condition2: *Second stage of job security integrated with cost estimation*

$JS^X \geq Nodes^Y_{(a, b, c, d)}$

While job security (JS) is required, with respect to cost estimation and reliability becomes greater when compare to the $Nodes^Y_{(a, b, c, d)}$ at node y then there is chances to failure or lack of resources in the grid structure(G) as consequence we need to improve the availability of resources which is nearer to 'Y' based on the λ arrival rate of replicas and mean failure rate=1/λ with following estimation.

Which states that failures in the job run due to unavailability of data or lack of resources in grid (G) with replication communication cost in RCc >1.(ie. if existing resources not enough to complete Job due to above said condition becomes true), then the job migration is required from one node to another node, for safe, lowest communication cost with availability of replicas and then performs feasible job run. The following are set of migrations that are required and also used to predict the job migration:

The job Migration cost for nodes involved in 'G' (JMC) = (y1,y2…..Yn) for nodes involved in 'G' and applicable for job migration.
When Yϵ{y1,y2,y3} shows that the communication cost between the node y1 and node y2 with entries involved in computation $Nodes^Y_{(a, b, c, d)}$ (if job 'X' fails to attain replicas in node y1 then job migrated to y2 and so on until number of nodes in Cluster of grid or Grid (G)).

MJC(y1,y2,y3)= Data Replication Size of (a,b,c,d) with jobs submitted / Bandwidth of Y_1,Y_2,Y_3

Bandwidth= y1->(Communication Bandwidth $Y_1$and Y_2)->y2

For resource movement from one node to adjacent or another nodes in G is mandatory, in order to recover job failures that occur in a node and JS shows the job security with respect to such kind of resource status. The replication communication cost is the dependency factor for estimation of secure job run with replicas in a node or migration of job is continued between nodes based on JS parameter.
According to figure1and earlier algorithm the following are some steps relates to predict jobs security decisions in proposed scheme.

Step1: The `U' depends on the Policy based parameters $[JS \geq Nodes^Y_{(a, b, c, d)}] \rightarrow Y \in y1,y2,\ldots,y_n$
becomes true then there is occurrence of job migrated to adjacent nodes in order to attain proper and capable node requirements which perform successful job run. Otherwise constraint
$JS \leq Nodes^{Yn}_{(a, b, c, d)}$
is applied, if it becomes true then there exists secure run of jobs with U_E; Where $U_E \xrightarrow{}$ indicates execution on constraint, JS $\xrightarrow{}$ Demand on resources required based on status prediction where 'q' is task.

Step2: Applying 'λi' Arrival rate's' to know access rates of replicas and scheduled length of jobs and to know the status of job security while execution.

Step3: The jobs execution with threshold time out value T_n and is applied for all Jobs

The above specified model is useful where number node failures are high because of the multi nodes connectivity in heterogeneous environment. This is applicable where is faces the problem of mean time to failure of jobs, between nodes while jobs scheduling in grid. The proposed replicas utilization status oriented parameter entries in distributed structure provides high degree of redundancy because of availability data.

VI. IMPLEMENTATION

Replicas movement is based on the communication cost between replicas server and file of replicas, in connection with that replicas movement is also depends on number of replicas loaded on particular nodes and its access rate upgraded value. The replicas access frequencies are considered into account, therefore we have to access all of the replicas based on read cost and update cost of replicas with standard and the varying distance (i.e. due to access frequencies increases) on all of the access frequencies of different replicas/same replicas that present at different locations.

Lookup operations applied only on selected replicas whose access distance becomes less due to proposed optimality algorithm which is based on distance and JS parameters as result of that replica manager(RM) allows to access only on minimum distance supporting replicas with its suitable replica optimizer(RO)class. While measuring the replication overhead performance at the time of replication, the FBR method uses to perform lookups on replicas placement while replicas read and updates objective functions are considered into the account of modification and notification of replicas.

VII. RESULTS

The proposed approach compared with existing approaches that shows optimized performance. The mean job time is calculated with constraints on the time to execute the job submitted + waiting time / number of jobs completed) and is as follows.

Mean job Time of the File Based object Replicas Scheme =

$$\sum_{j=1}^{m}(JT_j+WT_j) / m$$

Where 'm' is the number of jobs processed by the grid, JT_j is the job time to execute the j^{th} job and WT_j is the job waiting time of the J^{th} job.

In FBR approach is considered with accessing Object replicas of File. The FBR strategy almost gives lowest jobs execution times, which means that FBR strategy can have capable of accessing the files within less time and that reduces the wait time as result of that data availability increases, In such manner according to loads changes in proposed technique proves that within less time, the loads get feasible while jobs execution.

According to the measures the storage capability of no replication (NR) is always less, because the NR strategy uses no replicas creation. Moreover due to object replicas movement our File based object replicas (FBR) strategy gives the moderate performance and we believe that our strategy gives the better performance when number of jobs gets increased.

We assume that network bandwidth between nodes that are connected to one master and is almost all same and has high when compare to two different sites. As part of this initial condition files present within sites are already have high bandwidth so that accessing times of files present within sites are not considered as major factor into the account of bandwidth measure.

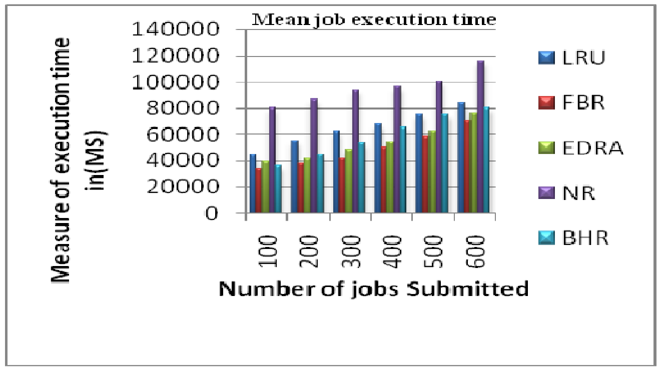

Figure2: Jobs execution time

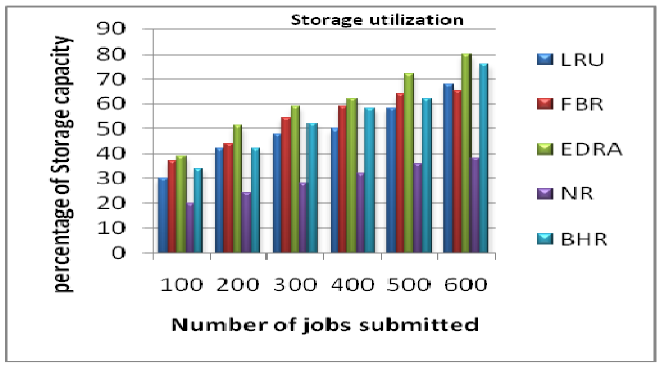

Figure3: Storage utilization

The band width is concern; due to above said factors NR uses maximum bandwidth because it requires more files to be transfer. The LRU give better results due to less files transfer but sometimes valuable file can be deleted, and the BHR, and EDRA also can gives much better results. In our proposed FBR scheme provides increased the local and centralized copies of replicas along with object replicas. The data objects are placed into files in the order to perform Replicas management based on access rate of object replicas as result of that less access is required thereby reduced files transfer rate and thus there is cause of increasing availability.

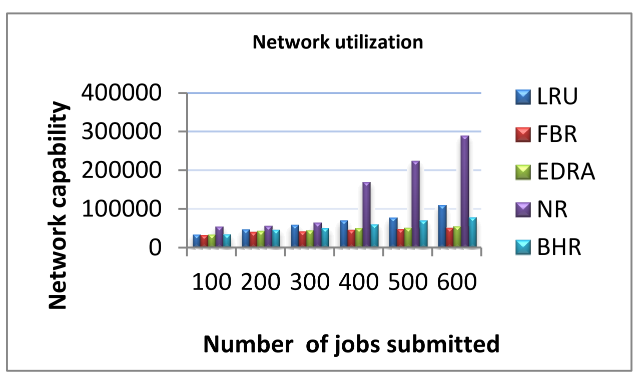

Figure4: Network utilization

VIII. CONCLUSION

In this paper the FBR (File based object replicas Algorithm) using priority based object replicas in data grids is applied to estimate optimized performance of replicas utilization in data grid environment. In connection with this approach, it also addresses an issue related to job security parameters (i.e. feasible job run) that decide jobs requirements status and job execution time based on the access rates and replication cost.

The proposed algorithms provides availability status of replicas and the replacement of replicas that takes the decision on jobs execution or jobs migrated between particular nodes within region and its sub-regions. The proposed algorithm reduces the resources utilization cost by replacing replicas in each region and its sub-regions.

Almost all the data grid becomes dynamic in nature due to these assumption considered into all the nodes status. The nodes status is available at all the time with status check at each stage, and thus proposed scheme provides appropriate service for data intensive grid environment.

ACKNOWLEDGMENT

We deeply thank god for his valuable help and generated such ideas in mind and lead me throughout my research

REFERENCES

[1] H. Lamehamedi, Z. Shentu, B. Szymanski, E. Deelman, Simulation of dynamic data replication strategies in data grids, in: *Proc. 12th Heterogeneous Computing Workshop, HCW2003,* Nice, France, April 2003, IEEE Computer Science Press, Los Alamitos, CA, 2003.

[2] D. Deatrich, S. Liu, C. Payne, R. Tafirout, R. Walker, A. Wong, M. Vetterli, Managing Petabyte-scale storage for the ATLAS Tier-1 centre at TRIUMF, in: *22nd International Symposium on High Performance Computing Systems and Applications, HPCS 2008,* 9-11 June 2008, pp. 167-171.

[3] J. Bresnahan, M. Link, G. Khanna, Z. Imani, R. Kettimuthu, I. Foster, Globus GridFTP: What's new in 2007, in: *Proceedings of the First International Conference on Networks for Grid Applications, GridNet 2007,* Lyon, France, 2007.

[4] R. Slota, D. Nikolow, L. Skital, J. Kitowski, Implementation of replication methods in the Grid environment, in: Advances in Grid Computing - EGC 2005,in: Lecture Notes in Computer Science, vol. 3470/2005, 2005, pp. 474-484.

[5] P. Liu, J. Wu, Optimal replica placement strategy for hierarchical Data Grid systems, in: Proceedings of the Sixth *IEEE International Symposium on Cluster Computing and the Grid, CCGRID 06*, 2006, pp. 417-420.

[6] C.D. Nam, C. Youn, S. Jeong, E. Shim, E. Lee, E. Park, "An efficient replication scheme for data grids", in: Proceedings 12th *IEEE International Conference on Networks*, ICON 2004, 2004, pp. 392-396.

[7] H. Bell, D.G. Cameron, L. Capozza, P. Millar, K. Stockinger, and F. Zini, "Evaluation of an economy-based file replication strategy for a data grid," *Proc. IEEE Int'l Symp. Cluster Computing and the Grid (CCGrid)*, pp. 661-668, 2003.

[8] R.S. Chang, J.S. Chang, and S.Y. Lin, "Job scheduling and data replication on data grids," *Future Generation Computer Systems*, Vol. 23, Issue 7, pp. 846-860, August 2007.

[9] K. Ranganathan and I. Foster, "Computation scheduling and data replication algorithms for data Grids", *Grid resource management: state of the art and future trends*, pp. 359-373, 2004.

[10] R.S. Chang and H.P. Chang, "A dynamic data replication strategy using access-weights in data grids," *Journal of Supercomputing*, Vol.45, Issue 3, pp. 277 – 295, 2008.

[11] K. Sashi and A. S. Thanamani, "Dynamic replication in a data grid using a modified BHR Region Based Algorithm," *Future Generation Computer Systems*, vol. 27, no. 2, pp. 202–210, 2011.

[12] N.Mansouri," A Threshold-based Dynamic Data Replication and Parallel Job Scheduling strategy to enhance Data Grid" In Proceedings of the Cluster Computing (2014) 17:957–977.

Authors Profile

Mr. P.SunilGavaskar pursed Bachelor of Engineering from University of Madras, Chennai in 2001 and Master of Engineering from Sathyabama University in year 2006. He is currently pursuing Ph.D in Department of Computer Science and Engineering, Sri Venkateswara University, Tirupath,AndhraPradesh since 2010. He is a life member of the ISTE since 2012,. He has published 10 research papers in reputed international journals including Thomson Reuters and conferences including IEEE. He has also reviewed a article that belongs to the Institution of Engineers Journal-B series,Springer publications. His main research work focuses on Distributed Systems,and Grid Computing. He has 8 years of teaching experience.

