The 2006 International Arab Conference on Information Technology (ACIT'2006)

BESTCLUSTER: A DYNAMIC REPLICATION ALGORITHM IN DATA GRIDS

Lakhdar Loukil

Faculty of sciences, Computer Science Department, Oran University, BP 1524 El M'Naouer Oran Algeria
loukil_lakhdar@yahoo.fr

ABSTRACT

We propose in this paper a new dynamic replication algorithm called BestCluster. The BestCluster replication algorithm takes a replication decision according to the total number of requests initiated by a cluster of users rather than taking replication decision depending on requests initiated by a single user. The objective is to benefit a group of users rather that only one user. A cluster associated with a user node is composed by the user node and its neighbours. Two users' nodes in a network are neighbours if there exist a physical link between them. The implementation and evaluation of the BestCluster replication algorithm have been performed using Optorsim, a data grid simulator. The preliminary experimentations results have shown that BestCluster could be a good replication approach especially in wide area networks such as data grids.

Keywords: *Data grids, Content Distribution Networks (CDN), World Wide Web, Replication, Facility Location Problem, k-median Problem.*

1. INTRODUCTION

A grid is a distributed collection of computer and storage resources aggregated to serve the needs of some community or virtual organization (VO) [8]. Data grids are wide distributed environments where huge amount of data (in order of terabytes or even petabytes) are produced and stored to be accessed by users which are geographically distributed around the world. In such environments, user's jobs require access to large number of files. If the required files are locally available, jobs are processed without any communication delay. However, if required files are stored in sites where jobs are not processed, the necessary files must be fetched from other sites incurring generally a very long time. Files replication technique is then used to improve data access performance by placing objects (files, Web contents, services …) close to users.

2. RELATED WORKS

The goal of replica placement problem is to decide the location of object replicas in order to minimize client perceived performance given an existing infrastructure or to minimize the infrastructure cost given a system performance.

The replica placement problem was first studied in the context of file assignment problem [7] and many other fields like distributed databases, and data management [11]. Most recently, replication has also been a key technology in wide distributed systems such as the World Wide Web [2][20], Content Distributed Networks (CDNs) [18][19][14] and datagrids [15][13].

In Content Distribution Networks (CDNs), replication is used to push content from the origin server to geographically distributed servers which bring content to the edge of the network where the clients are attached.

Yank et al. [22] investigate the replication and placement problem of multimedia objects in CDNs. They implement and evaluate a set of replica placement algorithms such as *Greedy algorithm*, *Hot Spot algorithm*, and *Max Fanout Algorithm*. They prove by simulation tests that, contrary to the intuition, deploying as many replicas as possible is not always a good strategy. Qiu et al. [14] explore the problem of Web server replica placement in CDN environments. They implement a number of placement algorithms (*Tree-based*, *Greedy*, *Random*, *Hot Spot*, *Super-Optimal*) that use client latency and request rates as input to make informed placement decisions. They conclude that a greedy algorithm can provide Web server replicas placement with performance close to optimal and that it is insensitive to errors in estimates of the input parameters. Bartolini et al. [4] propose a dynamic replica placement in CDNs which takes into account the system dynamics as well as the costs of modifying the replica placement. They assume that the users' requests obey to a Markovian model and formulate the dynamic replica placement as a *Semi Markovian Decision Process (SMDP)*. The proposed linear

programming formulation associated with the SMDP model is too computationally intensive.

Most works on replication consider read requests only. Xu et al. [21] examine replica placement on transparent replication proxies for applications with both read and write operations. Given a number of potential replication sites, the authors try to find out the number of replicas of an object that should be created and the proxies on which to place the replicas in order to obtain the optimal performance.

In data grids, large quantity of data files is produced and data replication is used to reduce data access time. Park et al. [13] propose a dynamic replication strategy called BHR based on network-level locality. BHR replication strategy is motivated from the assumption that hierarchy of bandwidth appears in Internet and the try to replicate popular files as many as possible within a region where broad bandwidth is provided between sites. Rahman et al. [15] present and evaluate the performance of a set of dynamic replication strategies based on risk and utility index. Before placing a replica at a site, they propose to calculate an expected utility and a risk index for each site by considering current network load and user requests. A replication site is chosen by optimizing expected utility or risk indexes.

3. DATA PLACEMENT PROBLEM FORMULATION

Data placement problem can be modelled as one of the two well known problems in operational research: Facility Location Problem (FLP) or a k-median Problem [3][9][17]. The difference between these two problems is that in FLP, the number of potential sites is not fixed a priori, contrary to k-median problem where the number k of potential servers is fixed as input.

3.1 FACILITY LOCATION PROBLEM

The facility location problem (FLP) can be defined as follows. Given a network topology represented by an undirected graph $G=(V,E)$ and a subset $S=\{s_1,s_2,..,s_N\}\subseteq V$ of servers candidate to store a replica of an object. Each client $j\in C$ of the network is assigned to one site $s_i\in S$ incurring a cost $d_j c_{ij}$ where d_j denotes the number of demands by the client $j \in C$ for the object replica stored in server s_i, and c_{ij} denotes the minimum distance between client j and server s_i. The storage of an object replica in a server s_i incurs a storage cost $scost(s_i)$. The objective of the FLP is to find the number and location of replica servers which minimize the total cost (access costs plus storage costs).

The placement problem can then be formulated as the following integer linear program:

$$minimize \sum_{i,j} d_j c_{ij} x_{ij} + \sum_{s_i \in S} scost(s_i) y_i \quad (1)$$

under the constraints

$$\sum_{s_i \in S} x_{ij} = 1, \ \forall j \in C, \quad (2)$$

$$x_{ij} \le y_i, \ \forall i \in \{1,..,N\}, \forall j \in C, \quad (3)$$

$$x_{ij} \in \{0,1\}, \ \forall i \in \{1,..N\}, \forall j \in C, \quad (4)$$

$$y_i \in \{0,1\}, \ \forall i \in \{1,..N\}. \quad (5)$$

The constraints (2) ensure that each client $j\in C$ must be assigned to one server site $s_i\in S$. The constraints (3) ensure that whenever a client j is assigned to a server s_i, then the server s_i must contain a replica of the object.

If the limit of storage capacity of a site must be considered and if we associate to each server $s_i\in S$ a storage capacity called $cap(s_i)$, then we add to the constraints of the above ILP the following constraints:

$$\sum_k x_{ij}^k size(o_k) \le cap(s_i), \ \forall i \in \{1,..,N\}, \forall j \in C, \forall o_k \in O \quad (6)$$

If a limit in service capacity is imposed by the placement problem, that is if we consider that all server s_i must process no more than a limit U of client requests, the FLP is called *Capacitated* FLP, otherwise, it is called *Uncapacitated* FLP. For Capacitated FLP case, the constraints that express the service limit are as follows:

$$\sum_{j \in C} x_{ij} y_i \le U, \ \forall i \in \{1,..,N\} \quad (7)$$

3.2 MINIMUM K-MEDIAN PROBLEM

The difference between the FLP and the k-median problem is that in k-median problem, the upper bound $k\le N$ of candidate servers is fixed as an input and the goal is to select k servers among N that minimize the sum of the assignment cost [6]. An other difference is that the placement cost of an object in a server is not considered in k-median problem. Like for FLP, the k-median problem can be formulated in terms of an ILP as follows:

$$minimize \sum_{i,j} d_j c_{ij} x_{ij}$$

under the constraints

$$\begin{aligned}
&\sum_{s_i \in S} x_{ij} = 1, \ \forall j \in C, \\
&\sum_{i \in \{1,..N\}} y_i \le k, \ \forall i \in \{1,..,N\}, \\
&x_{ij} \in \{0,1\}, \ \forall i \in \{1,..N\}, \forall j \in C, \\
&y_i \in \{0,1\}, \ \forall i \in \{1,..N\}.
\end{aligned} \tag{8}$$

The facility location and k-median problems are proven to be NP-hard [3]. Several polynomial constant-factor approximation algorithms have been proposed in the literature [16][12][10][6] that give an approximation of the optimal solution.

4. REPLICA PLACEMENT ALGORITHMS

4.1 GREEDY ALGORITHM

Greedy algorithm [14] processes as follows. At first step each of the *N* potential sites is evaluated individually. For each site, we assume that client will get the object from it and we calculate the total cost. The site with the minimal cost value is chosen to get a replica. At the second step we select the second replica among the remaining replica sites such that when it is combined with the first replica, the total cost is minimal. This process will continue until all replicas are placed.

4.2 HOT SPOT ALGORITHM

Hot Spot algorithm [14] attempts to place replica near the clients generating the greatest load. It sorts the *N* potential sites according to the amount of traffic generated within their vicinity. It then places the replicas at the top *M* sites that generate the large amount of traffic.

5. BESTCLUSTER APPROACH

5.1 PRINCIPLE

The principle of the *BestCluster* approach is group grid nodes into clusters in order to identify the most *"active"* clusters in the grid. The placement of a data replica in the center of that cluster would, in our sense, benefit to members of that cluster. The approach is based upon the notion of *risk index* as defined in [15]. The choice of replica site depends not only on the number of requests of a single node but on the sum of requests of all the nodes of the cluster and the bandwidth of the links. A *cluster* associated with a grid node s_i contains the node s_i and its neighbours. Two nodes s_i and s_j are considered to be neighbours if there exists a direct physical link between s_i and s_j.

5.2 BESTCLUSTER ALGORITHM

The main steps of the BestCluster algorithm are given below.

```
1: Real ClusterRisk = 0
2: for (i=0 ; i < NbSites ; i++) do
3:    if Sites[i] not in ServersList then
4:       MinDistance = MinDistanceToServer(Sites[i])
5:       ClusterRequest = Request[i]
6:       for (j=0 ; j < NbSites ; j++) do
7:          if Sites[j]∈NeighboursOf(Sites[i]) then
8:       ClusterRequest = ClusterRequest + Request[j]
9:          end if
10:      end for
11:    if MinDistance × ClusterRequest ≥ ClusterRisk
          Then
12:       ClusterRisk = MinDistance × ClusterRequest
13:       BestSite = Sites[i]
14:      end if
15:    end if
16: end for
17: Return BestSite
```

Algorithm 1: BestCluster Replication Algorithm

5.3 IMPLEMENTATION

The implementation and evaluation of the BestCluster replication algorithm have been performed using **Optorsim** simulator [1]. **Optorsim** is a Data Grid simulator, written in Java, which was developed by the European DataGrid project [1]. The goal of **Optorsim** is to allow experimentation with and evaluation of replica optimization strategies. Using a Grid configuration and a replica optimiser strategy as input, **Optorsim** runs a number of Grid jobs on the simulated Grid.

Simulation tests have been realized using network topology of the EU Data Grid [5]. Each site, except CERN, contains a storage element SE and a compute element CE. The CERN site contains a storage element and no compute element. Routers are considered as particular sites with no SE nor CE. Table 1 gives of storage capacities of the SE's of the network. The weight on an edge represents the available bandwidth of the physical link.

Site	Bologna	Catania	CERN	Imperial College	Lyon
Site capacity (Gb)	30	30	10000	80	50

Site	Milano	NIKHLEF	NoduGrid	Padova	RAL	Torino
Site capacity (Gb)	30	70	63	50	50	50

Table 1: Site capacities of the European Data Grid

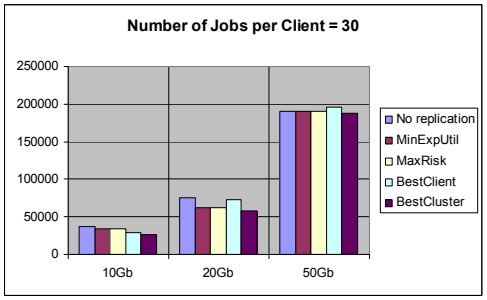
To compare the performance of BestCluster replica algorithm, we have implemented three replication algorithms proposed in [15]: *MinimizeExpectedUtility*, *Maximize MaxRisk* and *Best Client*.

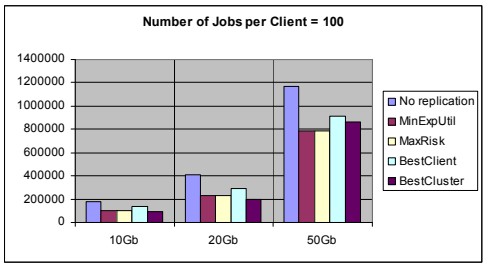
5.4 SIMULATION RESULTS

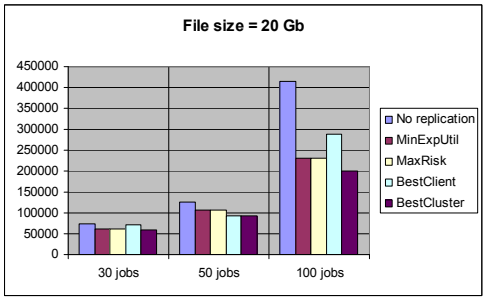
We assume that the master file *File1* is initially stored at SE's CERN site. All CE's sites process a job that refers to *File1*. To study the effect of *File1*'size on simulation time, we have taken different values for sizes of *File1*, 10Gb, 20Gb and 50Gb respectively. In addition, we have taken different values for number of jobs submitted to CE's. The values considered are 30, 50 and 100 jobs. The simulation results are synthesized in Table 2.

	No replication	MinExp Util	MaxRisk	Best Client	Best Cluster
Size: 10Gb NbRequests: 30	37372.78	34628.164	34628.164	29528.604	25961.734
Size: 20Gb NbRequests: 30	74789.15	61699.22	61699.22	72552.34	58384.26
Size: 50Gb NbRequests: 30	191103.36	191007.03	191007.03	195550.92	188262.03
Size: 10Gb NbRequests: 50	59342.777	45310.312	48921.67	53407.266	48350.723
Size: 20Gb NbRequests: 50	125283.56	107495.73	106260.72	93365.65	92635.9
Size: 50Gb NbRequests: 50	341022.28	304456.53	304456.53	308962.4	324234.2
Size: 10Gb NbRequests: 100	177489.17	98723.766	100118.97	138805.42	90036.125
Size: 20Gb NbRequests: 100	413496.88	230701.28	231223.06	289115.84	199500.00
Size: 50Gb NbRequests: 100	1166679.6	784827.2	784827.2	916719.5	865504.25

Table 2: Simulation time (in ms) in function of file size and number of requests







6. CONCLUSION AND FUTURE WORKS

In this paper, we have presented *BestCluster*, a dynamic replica placement algorithm. *BestCluster* algorithm consists in grouping users' nodes into clusters and placing an object replica in the most *active* cluster. The results obtained by simulation tests are promising and have significant benefits. We plan to proceed to more tests to confirm the results and implement this algorithm on a real grid.

REFERENCES

[1] http://cern.ch/edg-wp2/optimization/optorsim.html.

[2] Mahmood A., "Object Grouping and Replication Algorithms for World Wide Web," *Informatica,* vol. 29, pp. 347–356, 2005.

[3] Ivan D. Baev, Rajaraman R., "Approximation Algorithms for Data Placement in Arbitrary Networks," *in Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[4] Bartolini N., Presti F. L., Petrioli C., "Optimal Dynamic Replica Placement in Content Delivery Networks," *in ICON2003: Proceedings of the 11th IEEE International Conference on Networks*, pp. 125-130, 2003.

[5] Bell W., Cameron D. G., Capozza L., Stockinger K., Zini F., "Optorsim - A Grid Simulator for Studying Dynamic Data Replication Strategies," *International Journal of High Performance Computing Applications*, vol. 17, no. 4, pp. 403-416, 2003.

[6] Charikar M., Guha S., Tardos É., Shmoys D. B., "A Constant-factor Approximation Algorithm for the K-median Problem,"*in Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp. 1-10, 1999.

[7] Dowdy L. W., Foster D. V., "Comparative Models of the File Assignment Problem," *ACM Comput. Surv.*, vol. 14, no. 2, pp. 287-313, 1982.

[8] Foster I., Kesselman C., Tuecke S., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations,"*in International Journal of High Performance Computing Applications*, vol. 15, pp.200-222, 2001.

[9] Guha S., Munagala K., "Improved Algorithms for the Data Placement," *in proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms*, 2002.

[10] Levi R., Shmoys D. B., Swamy C., "LP-based Approximation Algorithms for Capacitated Facility Location," *in Proceedings of the 10th MPS Conference on Integer Programming and Combinatorial Optimization*, pp. 206-218, 2004.

[11] Maggs B. M., Meyer F., Vocking B., Westermann M., "Exploiting Locality for Data Management in Systems of Limited Bandwidth," *in IEEE Symposium on Foundations of Computer Science*, pp. 284-293, 1997.

[12] Mahdian M., Ye Y., Zhang J. "Improved Approximation Algorithms for Metric Facility Location Problems," *in APPROX 2002: 5th International Workshop on Approximation Algorithms for Combinatorial Optimization*. 2002.

[13] Park S., Kim J., Ko Y., Yoon W., "Dynamic Data Grid Replication Strategy Based on Internet Hierarchy," *in GCC (2)*, pp. 838-846, 2003.

[14] Qiu L., Padmanabhan V. N., Voelker G. M., "On the Placement of Web Server Replicas," *in Proceedings of INFOCOM*, pp. 1587-1596, 2001.

[15] Rahman R. M., Barker K., Alhajj R., "Replica placement in Data Grid: Considering Utility and Risk," *in ITCC'05: Proceedings of the International Conference on Information Technology: Coding and Computing*, pp. 354-359, 2005.

[16] Shmoys D B., Tardos É., Aardal K., "Approximation Algorithms for Facility Location Problems," *in Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp. 1-10,1999.

[17] Swamyy C., Kumarz A., "Primal-Dual Algorithms for Connected Facility Location Problems," *in Algorithmica*, vol. 40, no.4, pp.245-269, 2004.

[18] Tang M., Xu M., "QoS-Aware Replica Placement for Content Distribution," *in IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 10, pp. 921-932, 2005.

[19] Tang X., Xu J., "On Replica Placement for QoS-aware Content Distribution," *in INFOCOM*, 2004.

[20] Tenzakhti F., Day K., Ould-Khaoua M., "Replication algorithms for the World-Wide Web," *in Journal Syst. Archit.*, vol. 50, no. 10, pp. 591-605, 2004.

[21] Xu J., Li B., Lee D. L., "Optimal Replica Placement on Transparent Replication Proxies for Read/Write Data," *in IPCCC'02: Proceedings of the 21st IEEE International Performance, Computing, and Communications Conference*, pp. 103-110, 2002.

[22] Yang M., Fei Z., "A Model for Replica Placement in Content Distribution Networks for Multimedia Applications," *in ICC'03: Proceedings of the IEEE International Conference on Communications*, pp. 557-561, 2003.