"Dimitrie Cantemir" Christian University
Knowledge Horizons - Economics
Volume 6, No. 1, pp. 158–161
P-ISSN: 2069-0932, E-ISSN: 2066-1061

www.orizonturi.ucdc.ro

A NEW DYNAMIC DATA FRAGMENTATION AND REPLICATION MODEL IN DDBMSs. COST FUNCTIONS

Constantin BARON[1], Nicoleta Magdalena IACOB (CIOBANU)[2]

[1]*Faculty of Marketing, Bucharest, "Dimitrie Cantemir" Christian University, [1]E-mail: constantin_baron@yahoo.com*
[2]*Faculty of Finance, Banking and Accounting Bucharest, "Dimitrie Cantemir" Christian University, [2]E-mail: nicoleta.iacob_2007@yahoo.com*

Abstract *For globally expanding organizations, applications generate dynamic workflows with frequent changes in database access models (write, read) at different sites. In those situations a dynamic process to solve the requests on the site where were generated is recommended. The innovation consists in the possibility to integrate the three fundamental concepts specific to distributed databases: fragmentation, replication and fragments allocation in a model of an unbalanced dynamic system, completely decentralized and fully automated (the system continuously monitors the database and adjusts itself to the recent workload) which will permit remote read/writes to master replica and will offer high availability and performance boost.*

Key words:
Fragmentation, replication, allocation, dynamic system model

JEL Codes:

1. Introduction

A distributed database system consists of a collection of local databases, geographically located in different points (nodes of a network of computers) and logically related by functional relations so that they can be viewed globally as a single database [3].
Given the fact that the right to information [7], the volume and the diversity of data grow considerably year by year, the problem of efficient data management rise because the data must be available at any time and must be accurate. The best way to follow is to create well structured databases, which offer the possibility to store and process a big volume of information.

2. System model

Let's consider an unbalanced distributed database system (Figure 1) that is formed by a number of *n* sites, (S_1, S_2,...S_n) and a global table *T*. A table *T* can be entirely stored on a single site S_i, i=1,n or can be horizontally fragmented (F_1,F_2,…,F_m) on a number of *m* sites.
The dynamic characteristic of the model consist in the fact that the change of access models (read, write) must lead to the re-fragmentation and reallocation of fragments and creation or deletion of fragments replicas (the fragments replicas can change their rights) depending on the users data access histograms.
Database data access is continuously made. The old data is periodically removed and statistics will only include the recent visits. The statistics are stored using dynamic histograms. Whenever a tuple is accessed in one of the local replicas, the histogram is updated accordingly. Each site offers a set of histograms for each fragment that has a local replica.
Informations regarding fragmentation, the nodes where the copies are stored and the rights of the fragments (read/write) in nodes are realized by a common catalog service using a *distributed hash table*.
The proposed method for an unbalanced distributed database system has two major components:

- the detection of replicas access models and
- given those statistics, decisions on re-fragmentation and reallocation will be made.

These decisions are taken by algorithms utilizing cost function which estimate the difference in future communication costs between the change of a given replica and keeping that replica on the actual condition.
The proposed model is also efficient, a major concern being to obtain the best system response time with a minimum cost.

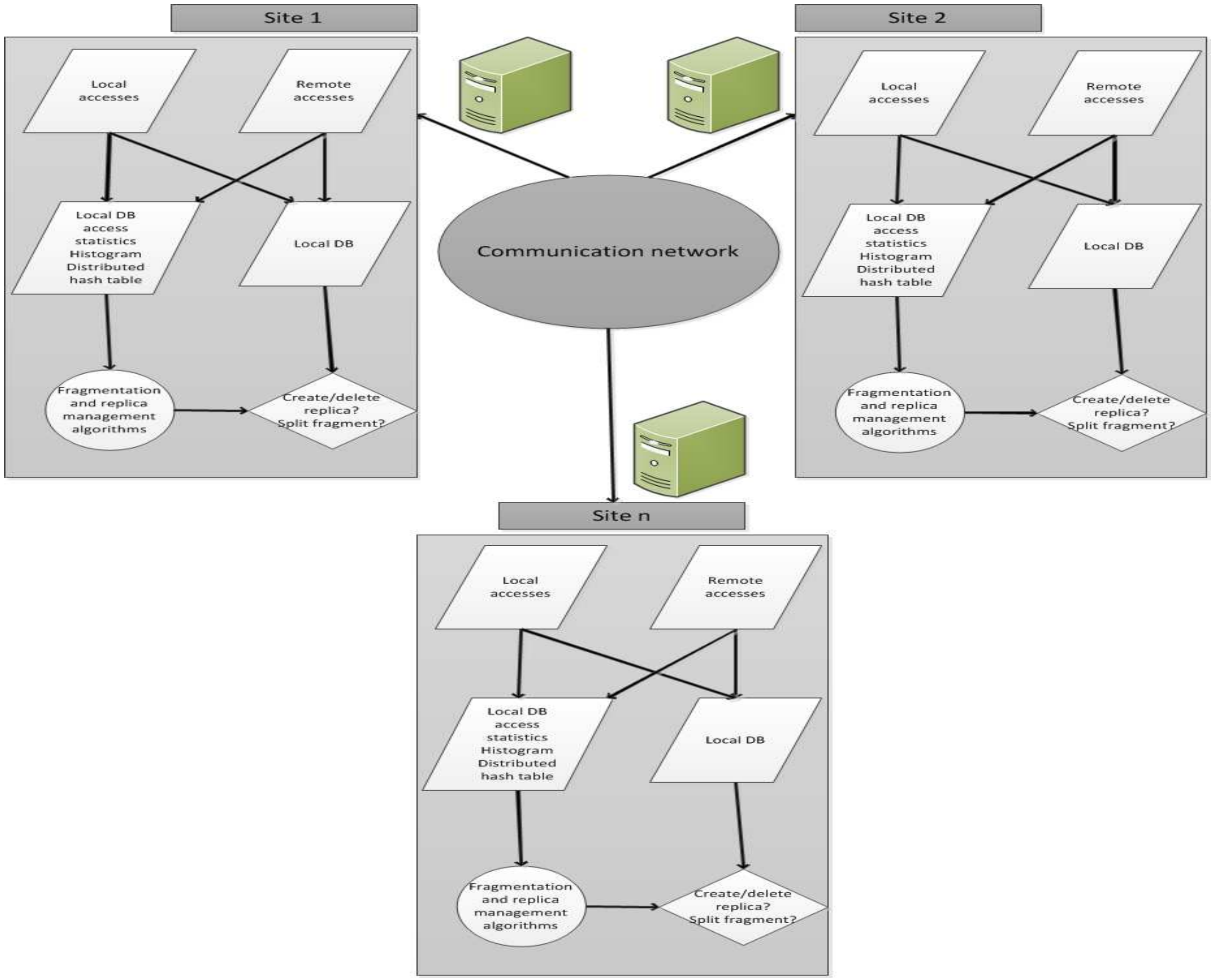

Figure 1. The distributed database from the proposed system model

2.1. Fragmentation and replication. Cost functions

The core of the algorithms is the cost functions. The functions estimate the communication cost difference (or utility) between taking a given action (create, delete, split) and keeping the status quo. The basic assumption is that future accesses will resemble recent history as recorded in the access statistics histograms [5].

The basic form of the costs functions are:

Util = Benefit – Cost (1)

Replica creation: The replication/creation of multiple copies of the same information [4] is justified (in economic terms) only if the fragment allocation cost R_i at the S_j site is lower than the remote access accomplished by the k applications from the S_j site, to the other copies of the fragment from the $S_{j'}$ ($j' \neq j$) stations.

The cost of the new R_i replica creation on the S_j site lies primarily in the fact that the new replica should be updated whenever writable replica is updated, and the second part of the cost consists from the effective transfer and from the replica storage on the new site:

$$C_{ij} = \sum_{k=1}^{p} \sum_{\substack{j'=1 \\ j' \neq j}}^{n} cu_{kij'} + cm_{ij} + \sum_{k=1}^{p} cal_{kij}$$

i=1,...,m; j=1,...,n; k=1,...,p (2)

p - represents the applications number (the number of fragment users),

m - number of fragments;

n - number of sites;

$cu_{kij'}$ - represents the costs of the update accesses of the R_i fragment to the S_j site, made by the k applications from the other $S_{j'}$ $(j' \neq j)$ sites;

cm_{ij} – represents the cost of storing the R_i fragment on the S_j site;

cal_{kij} - represents the local access costs to the R_i fragment on the S_j site, made by the k applications $(cal_{kij} = 0)$.

The benfit of crating a new R_i replica on the S_j site is that the remote reads will become local operations and thus, they don't have network communication cost $(cc_{kij'} = 0)$:

$$B_{ij} = \sum_{k=1}^{p} car_{kij'} + \sum_{k=1}^{p} cc_{kij'}$$

i=1,...,m; j=1,...,n; k=1,...,p (3)

$car_{kij'}$ - represents the remote access costs made by the k applications from the other S_j site to the other copies of the R_i fragment from the $S_{j'}$ $(j' \neq j)$ sites;

$cc_{kij'}$-represents the communication cost between sites.

Replica deletion: The removal of a replica is justified only if the remote access cost to fragment R_i from the S_j site is lower than local access cost, made by the k applications from the S_j site, to the other copies of the fragment from the $S_{j'}$ $(j' \neq j)$ stations.

The cost of a replica deletion R_i on the local site consist in the fact that accesses from the local site to replica R_i will become remote accesses and thus have a network communication cost:

$$CD_{ij} = \sum_{k=1}^{p} cal_{kij} + \sum_{k=1}^{p} cc_{kij'}$$

i=1,...,m; j=1,...,n; k=1,...,p (4)

The benefit of deleting a R_i replica on the S_i local site is that the update of master replica do not need to be transmitted to the local site:

$$BD_{ij} = \sum_{k=1}^{p} \sum_{\substack{j'=1 \\ j' \neq j}}^{n} cu_{kij'}$$

i=1,...,m; j=1,...,n; k=1,...,p (5)

The algorithm [2] must go through each site and should verify for each fragment if the deletion\creation of a new replica (Util > 0) is appropriate. In the site selection it will be taken into account the best response time (thus, it will be taken into account the sites load) with the greatest benefit [6]. The result is a utility value that estimates the lowering of communication costs by deleting\creating a new replica.

In the deletion\creation process of a new replica it will be taken into account the available space from the site on which the replica allocation is desired, and also the minimum\maximum number (data will have at least two write replicas) of replicas from the imposed system.

Splitting fragments and migrating master replicas: The re-fragmentation and data allocation algorithm on different nodes from database where are frequently accessed, have the role to minimize the network traffic by identifying parts of a table which must be extracted to form a new fragment and migrate them to a remote site by taking into consideration the number of fragments on a node and their dimensions [1].

The benefit of replica migration (write access replica) from the local site N to remote site N1 consist of the fact that the remote writes will become local operations. The cost will consist of writes at local sites and the cost of migration.

To view the percentage of occupied bandwidth, run the following command on Cisco routers:

```
show ip flow top-talkers
```

which generates the following results:

```
SrcIf     SrcIPaddress   DstIf      DstIPaddress   Pr SrcP DstP    Bytes
Gi0/2      10.1.9.200    Gi0/0.2    10.2.39.10    06 1F40 12E4    25M
Gi0/2      10.1.9.200    Gi0/0.2    10.2.3.25     06 1F40 0540    25M
Gi0/2      10.1.9.200    Gi0/0.2    10.2.2.30     06 1F40 05F2    24M
Gi0/2      10.1.9.200    Gi0/0.2    10.2.38.3     06 1F40 067D    24M
Gi0/2      10.1.9.200    Gi0/0.2    10.2.27.12    06 1F40 0C59    17M
Gi0/2      10.1.9.200    Gi0/0.2    10.2.9.58     06 1F40 05C9    17M
Gi0/2      10.1.9.200    Gi0/0.2    10.2.8.19     06 1F40 0656  2876K
```

To view the response time for a location, run the following command:

```
ping 10.2.2.1
```

which generates the following results:

```
Pinging 10.2.2.1 with 32 bytes of data:
Reply from 10.2.2.1: bytes=32 time=18ms TTL=253
Reply from 10.2.2.1: bytes=32 time=19ms TTL=253
Reply from 10.2.2.1: bytes=32 time=19ms TTL=253
Reply from 10.2.2.1: bytes=32 time=19ms TTL=253
```

3. Conclusions and further work

In conclusion, the proposed system model is not only innovative, but also offers good performance, is configurable and easy to administer. The model can also be applied in parallel databases, because each site takes decisions to partitionate, migrate and/or replicate fragments based on the available informations and the decisions are taken without statistics report or synchronization between sites.

Future researches focus on the development of the proposed system model in order to detect, based on query analysis, the models which appear recurrently.

References

[1] Ciobanu (Iacob) N.M. (2012), "Algorithm for Dynamic Partitioning and Reallocation of Fragments in a Distributed Database", ICVL, Virtual Learning - Virtual Reality Models and Methodologies Technologies and Software Solutions, University of Brașov, November 2 - 3, 2012, pp. 387-393, Brașov, International Journal of Computer Science Research and Application- IJCSRA, Vol. 3, Issue 1.

[2] Ciobanu (Iacob) N.M. (2012), "Proposed Algorithm for Dynamic Creation or Deletion of Replicas in a Distributed Database. Cost Analysis", PDEs and Stochastic Processes, The 5th Workshop Series on Mathematics, October 12-14, 2012, University of Piteşti.

[3] Ciobanu (Iacob) N.M.and Ciobanu (Defta) C.L. (2012), "Synchronous Partial Replication – Case Study: Implementing E-Learning Platform in an Academic Environment", 4th World Conference on Educational Sciences – WCES 2012, University of Barcelona, Spain, February 02nd – 06th 2012, Procedia-Social and Behavioral Science Journal, Volume 46, 2012, pp. 1522–1526.

[4] Ionita C. and Oprea P. (1998), "Proiectarea bazelor de date distribuite", Revista Informatica Economica, nr. 8/1998, pp. 51-57.

[5] Jon Olav Hauglid, Norvald H. Ryeng and Kjetil Norvag, DYFRAM: dynamic fragmentation and replica management in distributed database systems,Journal Distributed and Parallel Databases, Volume 28 Issue 2-3, December 2010, Pages 157-185.

[6] Lungu I. and Fodor A.G. (2006), "Optimizing Queries in Distributed Systems", Economic Informatics Journal, no. 1 (37)/2006, pp. 67-72.

[7] Rădulescu D.M. and Rădulescu V. - Educating the consumer about his right to a healthy environment, *Procedia-Social and Behavioral Sciences on ScienceDirect*, Volume 15/2011, 5 pp. 466-470.