World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:4, 2010

Metadata Update Mechanism Improvements in Data Grid

S. Farokhzad, M. Reza Salehnamadi

***Abstract*—**Grid environments include aggregation of geographical distributed resources. Grid is put forward in three types of computational, data and storage. This paper presents a research on data grid. Data grid is used for covering and securing accessibility to data from among many heterogeneous sources. Users are not worry on the place where data is located in it, provided that, they should get access to the data. Metadata is used for getting access to data in data grid. Presently, application metadata catalogue and SRB middle-ware package are used in data grids for management of metadata. At this paper, possibility of updating, streamlining and searching is provided simultaneously and rapidly through classified table of preserving metadata and conversion of each table to numerous tables. Meanwhile, with regard to the specific application, the most appropriate and best division is set and determined. Concurrency of implementation of some of requests and execution of pipeline is adaptability as a result of this technique.

***Keywords*—**Grids, data grid, metadata, update.

I. Introduction

APPLICATION programs is employed in various fields like high-energy physics, molecular modeling, geological sciences, create large databases, simulations and tests with high scale. Analysis of database and distributing them among researchers, being located within the framework of giant geographical data, requires sources with high capacity like supercomputers, networks with high bandwidth and mass storage systems. A formulated trend is required for taking advantage of technologies of comptrollers, storing, network and web structuring altogether for facilitation of cooperation, and collaboration affair and scientific research being sensitive to data. This activity requires new patterns in internet-related computations with the aim of publishing address like multi-domain application, assisting, coordinating owners of resources and also eradicating borders of system [7]. The other significant services in data grid includes: storage system, metadata management system. Metadata management service is a core service which is placed at nucleus part of data grid; support other services high grid level like replica files. [8]

There are many data in grid which are distributed equally. These data are used and/or produced in processing which is made by grid. Mechanisms should exist, aimed at processing and accessing these data transparently. "Metadata" is regarded as one of these mechanisms.

"Metadata is a data, describing content, form, shape or specifications of a data record or a data source. Metadata can be applied in descriptor of completely structured resources or non-structured data such as textual documents. Metadata also can be used for descriptor of electronic resources, digital data (including digital images) and printing documents like books, magazines, periodicals and reports. In addition, Meta Data can be placed inside a data resource (such as web resources) and/or kept in a database separately." [3].

Various types of metadata include:

1- Application Metadata: Content of data, which are displayed by file, has been comprised of descriptor of obtained data and other data. The data pertains to scientific application programs through application of tools like NetCDF, HDF and XML.

2- Replica Metadata: Data copies, selection policies of accessing and storing place.

3- System Configuration Metadata: Data on grid, for example, connections and capacity of storing and applicable policies [5]. At this paper, application metadata is discussed.

II. Related Works

Metadata Management System (MMS) should safeguard integrity of its metadata while facing with unanticipated fall of system like power cut (outage) or breakdown of Operating System (OS). Therefore, this issue should be considered in updating metadata, aimed at preserving compatibility of metadata.

Synchronous writings for sequence of updating metadata is regarded as one of updating methods of metadata in many file systems like VMS, DOS most of UNIX models. Consequently, updating metadata deals with speed of disk rather speed of memory or processor. Destruction of performance can be considered as noticeable upon ignoring the way of implementation. Hence, reduction of integrity, security and availability will be obtained.

Hiring NVRam technologies as a nonstop feed supply source (emergency power) for whole system and/or a separate RAM flash device is regarded as one of effective ways for meeting demands to maintain compatible status of disc. According to this method, only updating NVRam should be kept compatible. Updating can be introduced inside disc at any order and at any time. Efficacy of this method is superior to concurrent writings, for, it consumes less time. There exists this worrisome that operating system (OS) may face breakdown and consequently, some contents of drawback

S. Farokhzad is with the Islamic AZAD University of IRAN, Tehran South Branch, (Phone: 00989127248414-00989121005323, P.O. Box: 16535-334, e-mail: S_farokhzad@yahoo.com).

M.Reza Salehnamadi is with the Islamic AZAD University of IRAN, Tehran South Branch, e-mail: m_saleh@azad.ac.ir)

NVRam (Non-Volatile Random Access Memory) cache may be eradicated. But, this issue can be followed up strictly with effort and endeavor. Formulation of NVRam costs is regarded as one of major problems at this method. Of course, this approach is appropriate only for the systems which are equipped with it. Moreover, system will tolerate the files which support on NVRam devices as additional overload for transfer of data among them. Eventually, file system, after a system breakdown, revises both NVRam and contents of disc [10].

Although sequence of updating preserves integrity of a file system, another method is this that a complex of updating, related to each other, should be turned into a group, operated as an atomic procedure [4].

Using delay in writings for all updating is the other method. At this method, there exists a cache which keeps rewritings code and also imposes some operation on disc. Following up dependencies is vivid between buffers, but this method will create only a marginal reduction in number of congruous writings and consequently, non-improvement is happened, for, this system should prevent from establishment of spiral dependencies.

Using soft updating for following up and executing metadata depends on carrying out writings for all its operation with delay. Consequently, boosting safety, security and integrity will be obtained [2].

Updating soft is a technique of modulation for low-cost sequence from small-size updating to delayed cache writings blocks. In the same direction, soft updating mechanism follows up dependencies between cached updating. Metadata and execution of these dependencies are copied by sequence of updating as metadata unclean data, details of which are written later at non-volatile memory. In soft updating, data is kept per each unique updating of metadata which determines the very dependant updating [2].

With updating soft, cost of preserving integrity will reach to minimum rate. For work trends, which emerge metadata repeatedly, this method is regarded as most appropriate approach, as compared with common concurrent writing method, which improves productivity. Eventually, integrity and security is ameliorated as compared with a number of previous modulations. In addition to this status, disc is kept in such a way that to be mantled safely after each system breakdown and used immediately without any additional operation like checking compatibility. It should be noted that this advantage will reduce system recovery time in less than a second.

In metadata management systems, which take advantage of database for storing metadata, using SRB (Storage Resource Broker) middleware is one of accessing and updating methods of metadata. SRB is an intermediate based on client server. SRB is a transparent virtual middleware which is used for sharing data between inharmonious and distributed data resources that have different management approaches. Through SRB along with MCAT, regarded as catalogue of metadata, connection is established with storage of metadata and following getting access to metadata is made. [9]

III. UPDATING METADATA: PROPOSED METHOD

Dynamic behavior of grid, which is created due to change of accessibility of resources, unanticipated updating of data and effect of sites' local policies are of the factors which cause implementation of capabilities of grid face problem. such complicate distributed environments cannot be controlled and reorganized as hierarchical, for, each service will be a bottleneck central manual for performance of system and is considered as the sole break point.

Presently, data grids use application metadata catalogue and SRB middleware for management of metadata, details of which are observed at Fig. 1.

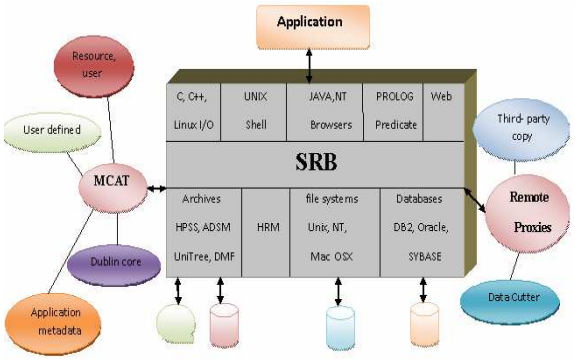

Fig. 1 Outline of the SRB and MCAT [9]

SRB is a harmonious and distributed file system, regarded as an intermediate for getting access to database and can be beneficial and fruitful in activities like digital libraries, web, data grid and advanced archive system. The application metadata, which is allocated to the metadata related to references and users in data grid, is stored in database within the framework of Dublin Core Standard. Logical approach to replica physical and chemical specifications and access control is managed by advanced SRB middleware package system. A number of 15 elements with simple level and also 7 elements with specific level have been envisioned in Dublin Core standard.

At the present paper, more concentration has been made on elements of common or simple level and changes were adopted with the aim of amelioration and improvement of updating mechanism and getting access to data metadata on database as well as way of storing application metadata within the framework of Dublin Core standard at connecting database. Due to its predetermined elements, Dublin Core standard can be stored at connection database and within the framework of table and also took advantage of connection database like simplicity of comprehension and usage, etc. However, the main objective is as follows: a number of 15 elements of common level of Dublin Core, instead of placing in a table with 15 fields, should be put at two tables, each comprised of nine fields. Of course, this fractioning or segmentation is carried out in data grid thanks to the application of this metadata. It is oblivious that a little agglomeration will be appeared at this method.

The simple level elements have been comprised of:

Title, author, subject description, publisher, cooperator, date, type, format, identifier, language, relation, source, coverage and rights.

The way of classified is according to the following two tables:

1. Title, date, identifier, author, description, publisher, cooperator, coverage and rights.
2. Date, title, ID, subject, type, format, language, source, relation, etc.

Of course, this vertical dividing can be changed thanks to specific applications.

Paying due attention to this issue seems necessary in a way that main owner of a shared resources source at data grid can change its application metadata.

The proposed idea depends on this supposition that when a search or navigation is made at data grid, for example, for finding a paper, getting access to title field, date and identifier, which is actually the same main address of its extraction, it is troubleshooting and significant. For this reason, for this reason, in column classification, these elements have envisioned commonly between two tables derived from Dublin Core standard. If needed, element of subject can be shared along with title and identifier instead of date element. In that case, row classified should be made on subject. At the second stage, they were classified based on date of tables.

At the first stage, with classified elements into two groups or two tables, it is intended to improve speed of searching and updating and/or easing possibility of fulfilling two updating simultaneously or envisioning concurrent reading and writing. Of course, this issue is significant and vital exception in such a way that possibility of concurrent updating of identifier in a table and reading the same row at other table is impossible. In addition, two concurrent updating is impossible on identifier. Of course, except a slight amount of aggregation, emerged as a result of repetition of some fields at table, the presented method is followed with a meager complexity. For example, controlling of cases like the above mentioned case It is expected that speed of searching and ordinary updating will increase. Ordinary updating is meant non-similar updating with regard to the mentioned exception case. Since major owner of resource can change main location of accessibility to it (not replicating ways), the mentioned exception case and/or similar cases may rarely happen.

At the next stage, rows are classified based on date. Namely, two tables are turned into four tables. (For example, resources produced up by 2000 and after it). Refer to Fig. 2.

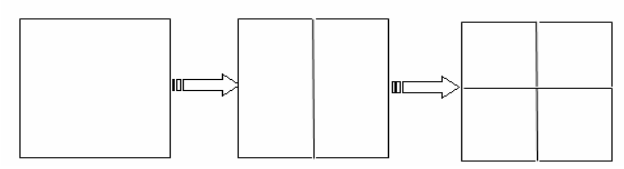

Fig. 2 How divided the main table

At this case, it is expected that searching speed will increase, aimed at taking advantage of minute grading. Under such circumstance, pipelining some requests will be created. Of course, metadata catalogue is slightly faced with change and complexity. All these classifications and pertinent considerations can be described within framework of a small metadata entitled "Meta Metadata" for management of metadata, aimed that analyzing input request for referring to main metadata at small meta and then guided to related part. When a request is received, it is studied in small Meta as follows: In which part of table does fields and specifications of its desired record place? Through an application program, the small Meta and analysis of requests can be inserted in metadata catalogue and/or a program is observing on request prior to responding bank with the aim of carrying out elementary analysis for finding related table. The said issue will cause that consecutive classified refrains reduction of response time as row and changes reduction trend of responding time, inclined towards gradual increase.(due to analysis consumption time and finding related table) According to classified and segmentation of tables, data volume will turn smaller at the time of updating.

In row point of view, tables can be divided. Of course, it should be studied fruitfulness of these classifications and segmentations.

IV. SIMULATION AND EVALUATION

Since proposed method was imposing changes on bank of preserving metadata, it is referred to SQL SERVER 2008 with the aim of studying and observing results obtained from this method in a nearly actual environment and finally, a databank was created with the help of this software. In the beginning, at first stage of tests, related metadata, pertinent to one million papers, are inserted inside a table and then, response time to various search requests and updating was measured. In the same direction, an average time was gained for requests related to search. Moreover, an average time was obtained for updating requests as well.

After that, table is divided into two smaller tables vertically, aimed at taking advantage of two tables (9+1) instead of keeping data in one table (15 + 1) field (15 fields Dublin Core common level and one identifier for safeguarding tables' relation. As mentioned in above, basic and fundamental fields, like title, date (it can be replaced with subject. Linear classified can also be carried out as subject basis), identifier (main availability place) have been repeated at two tables. Now, a series of requests related to search and updating was made at two tables and following, execution time was registered. Then, the two obtained tables are divided into two groups based on date of papers (year). Consequently, four tables were envisioned. Searching and updating requests were carried out again on these four tables and response time was registered as well. Linear classified is changed and then, linear divisions should be divided into four groups instead of two groups. Finally, eight tables were envisioned. Once again, response time to a number of searching and updating request was registered.

This operation was made for a number of 16 tables. The result obtained in this regard has been shown at diagram No. 3.

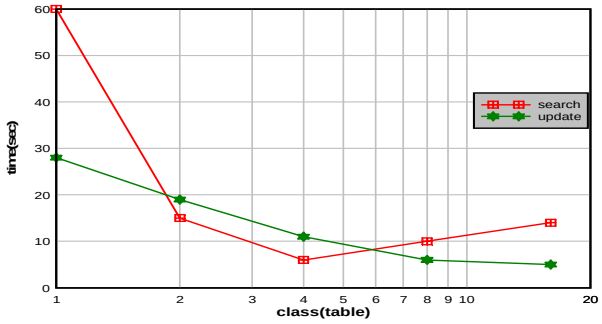

Fig.3 Average search and update times in different classes for 1000000 records

As you can observe in above diagram, response time to searching and updating requests is decreased through linear and column classification. Of course, this diagram is not completely linear form. With increase of classification of a greater extent, remarkable time is spent for searching related table. This issue will increase response time gradually.

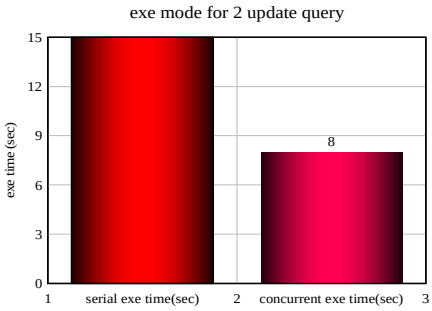

Fig.4 compared to serial execution times and concurrent tow update request

At two tabular modes, which only column divisions had been made and possibility of responding two updating requests had been provided simultaneously on metadata related to one source, the below results were obtained. These results are shown at Table No. 4 & 5 vividly.

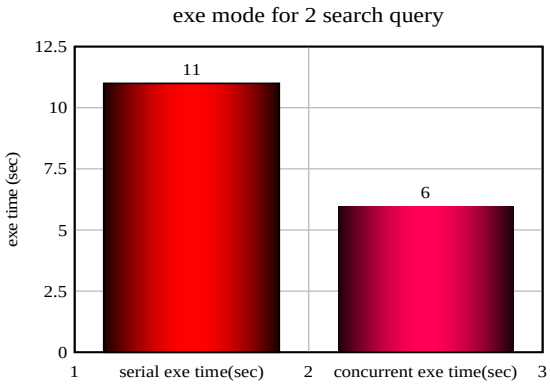

Fig.5 compared to serial execution times and concurrent tow search request

At Diagram No. 4, response time to two updating requests has been mentioned as serial and harmonious forms. The Diagram No. 5 shows response time at these two modes with regard to searching requests.

With due observance to the results obtained in this regard, productivity, efficacy and speed will be increased upon appropriate classification and grouping on metadata. Even, implementation of pipeline of requests can be attained through toleration of aggregation and slight complexity. Customarily, foreign database are used in data grid for keeping metadata. Consequently, aggregation can be tolerated and obtained high responding speed. In the same direction, complexity is hidden in view of users at whole data grid. Fruitful and effective classified should be taken into consideration for accessibility of high productivity and speed.

At the second step, the number of initial records at main table is changed into 1200000 records for studying effect of degree of records in increase of speed and productivity by the above mentioned classifications. The carried out stages were repeated at the first step. The results, obtained in this regard, are shown at Fig. 6.

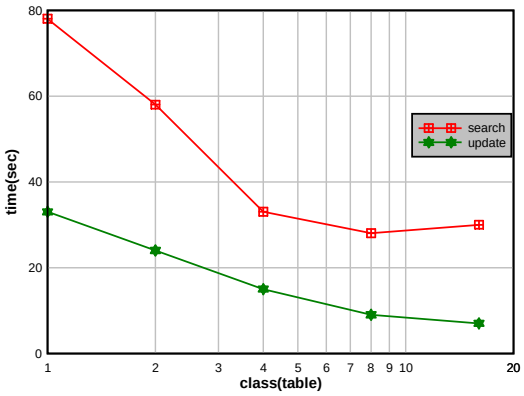

Fig.6 Average search and update times in different classes for 1200000 records

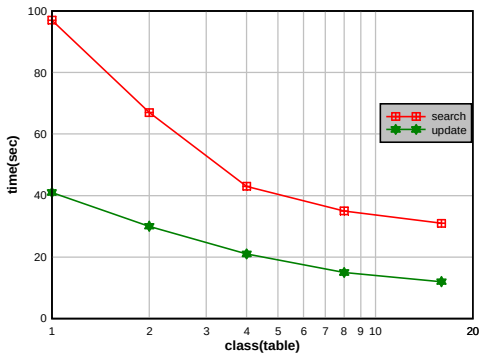

Fig.7 Searching and Updating Times Average in Various Classifications for 1500000 Records

The same step was made for 1500000 records. The average registered times have been shown at Fig. 7.

As it is observed at the above mentioned diagrams, effect of classified is boosted upon increase of number of initial records. Meanwhile, we will reach time increase trend lately. In other words, upon increase of records, capacity of classified is boosted as well.

V.CONCLUSIONS

Responding speed to searching and updating can be increased through proposed method i.e. linear and column classified and instead of keeping metadata related to data grid in a relational table as well as using many tables. With due observance to specific application, we should be careful to find and act the best classification. At this method, due to existence of many tables, we can take advantage of concurrent execution of some requests and even implementation of pipeline as well. Of course, with regard to simultaneousness of requests, exceptions should be determined due to application. May be, it can be claimed that if consistency control is made with locking operation, probability of facing deadlock, due to smallness of grading, is reduced through the said method.

REFERENCES

[1] A. Rajasekar, M. Wan, R. Moore, " MySRB & SRB components of a Data Grid", 2002.
[2] G. Ganger, Y.PATT, “Metadata update performance in file systems”. In Proceedings of the 1st USENIX Symposium on Operating Systems Design and Implementation (OSDI ’94, Monterey, CA, Nov.), USENIX Assoc., Berkeley, CA, 49–60. 1994.
[3] D. Haynes, “Metadata for information management and retrieval. London: Facet Publishing. 2004.
[4] R. HAGMANN, Re-implementing the cedar file system Using logging and group commit. In Proceedings of the Eleventh ACM Symposium on Operating systems Principles (Austin, TX, Nov. 8-11, 1987), L. Belady, Ed. ACM Press, New York, NY, 155–162. 1987.
[5] K. Kaczmarski, Piotr Habela, K. Subieta, "Metadata in A Data Grid Construction", Proc. Of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004.
[6] M. Aldinucci, M. Danelutto, G. Giaccherini, M. Torquati, M. Venneschi ," wards a Distributed Scalable Data Service for the Grid ",John von Neumann Institute for Computing, J¨ulich, NIC Series, Vol. 33, pp. 73-80, 2006.
[7] O.A. Rahmeh, P. Johnson, “Towards scalable and Reliable Grid Networks”, IEEE 2008.
[8] S. Venuyopal, Rajkumar Buyya, kotagiri Ramamohanarao, A Taxonomy of Data Grids for Distributed Management and processing” ,ACM computing surveys, Vol.38, March 2006.
[9] V. Silva, " Build a data grid with the Storage Resource Broker", IBM, 2007.
[10] M. WU, W. ZWAENEPOEL, “A non-volatile, main Memory storage system. In Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS VI, San Jose, CA, Oct. 4–7 ACM Press, NY, 86–97. 1994.
[11] Won Kim, “On Metadata Management Technology Status and Issues”, in Journal of Object Technology, vol. 4, no. 2, March-April 2005, pp. 41-47.