ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A Study on Data Grid Middlewares

Raafiya Gulmeher*[1], Mohammed Abdul Waheed[2]
*[1]Research Scholar JJT University,jhunjhunu, Rajasthan, India
[2]Associate Professor, VTU Regional office, Gulbarga, Karnataka, India
raafiyagulmeher@yahoo.com

Abstract

Grid computing is one of the fastest emerging technologies within the high performance computing environment.Grid deployments that require access and processing of data are called data grids. They are optimized for data oriented operation. In a data grid environment, data replication is an effective way to improve data accessibility. Here data partitioning and dynamic replication in data grid are considered. In which security and access performance of a system are efficient. There are several important requirements for data grids, including information survivability, security, and access performance. More specifically, the investigation is the problem of optimal allocation of sensitive data objects that are partitioned by using secret sharing scheme or erasure coding scheme and replicated. DATA PARTITIONING is known as the single data can be divided into multiple objects. REPLICATION is known as process of sharing information (i.e.) storing same data in multiple systems. Replication techniques are frequently used to improve data availability. Single point failure does not affect this system, where the data will be secured.

General Terms: Performance, Reliability, Security

Keywords: Data grid, Replication, Data partitioning, Replica, Distributed computing.

Introduction

Data grid is a distributed computing architecture that integrates a large number of data and computing resources into a single virtual data management system. It enables the sharing and coordinated use of data from various resources and provides various services to fit the needs of high-performance distributed and data-intensive computing. Many data grid applications are being developed or proposed, such as DOD's Global Information Grid (GIG) for both business and military domains, NASA's Information Power Grid GMESS Health-Grid for medical services, data grids for Federal Disaster Relief, etc. These data grid applications are designed to support global collaborations that may involve large amount of information, intensive computation, real time, or non real time communication. Success of these projects can help to achieve significant advances in business, medical treatment, disaster relief, research, and military and can result in dramatic benefits to the society.

There are several important requirements for data grids, including information survivability, security, and access performance. For example, consider a first responder team responding to a fire in a building with explosive chemicals. The data grid that hosts building safety information, such as the building layout and locations of dangerous chemicals and hazard containment devices, can help draw relatively safe and effective rescue plans. Delayed accesses to these data can endanger the responders as well as increase the risk to the victims or cause severe damages to the property.

At the same time, the information such as location of hazardous chemicals is highly sensitive and, if falls in the hands of terrorists, could cause severe consequences. Thus, confidentiality of the critical information should be carefully protected. The above example indicates the importance of data grids and their availability, reliability, accuracy, and responsiveness. Replication is frequently used to achieve access efficiency, availability, and information survivability. The underlying infrastructure for data grids can generally be classified into two types cluster based and peer-to-peer Systems.

In pure peer-to-peer storage systems, there is no dedicated node for grid applications (in some systems, some servers are dedicated). Replication can bring data objects to the peers that are close to the accessing clients and, hence, improve access efficiency. Having multiple replicas directly implies higher information survivability. In cluster-based systems, dedicated servers are clustered together to offer storage and services. However, the number of clusters is generally limited and, thus, they may be far from most clients. To improve both access performance and availability, it is necessary to replicate data and place them close to the clients, such as peer-to-peer data caching. As can be seen, replication is an effective technique for all types of data grids. Existing research works on replication in data grids investigate replica access protocols resource management and discovery techniques replica location and discovery algorithms and replica placement issues.

Replication of keys can increase its access efficiency as well as avoiding the single-point failure problem and reducing the risk of denial of service attacks, but would increase the risk of having some compromised key servers. If one of the key servers is compromised, all the critical data are essentially compromised. Beside key management issues, information leakage is another problem with the replica encryption approach. Generally, a key is used to access many data objects. When a client leaves the system or its privilege for some accesses is revoked, those data objects have to be re encrypted using a new key and the new key has to be distributed to other clients. If one of the data storage servers is compromised, the storage server could retain a copy of the data encrypted using the old key. Thus, the content of long-lived data may leak over time. Therefore, additional security mechanisms are needed for sensitive data protection. In this paper, we consider combining data partitioning and replication to support secure, survivable, and high performance storage systems.

Existing System

The intrusion tolerance concept and data partitioning techniques can be used to achieve data survivability as well as security. The most commonly used schemes for data partitioning include secret sharing and erasure coding. Both schemes partition data into shares and distribute them to different processors to achieve availability and integrity. Secret sharing schemes assure confidentiality even if some shares (less than a threshold) are compromised. In erasure coding, data shares can be encrypted and the encryption key can be secret shared and distributed with the data shares to assure confidentiality. However, changing the number of shares in a data partitioning scheme is generally costly. When it is necessary to add additional shares close to a group of clients to reduce the communication cost and access latency, it is easier to add share replicas. Thus, it is most effective to combine the data partitioning and replication techniques for high-performance secure storage design.

Limitation of Existing System

- Low security of data if it is stored in grids.
- We need to remember the security key for associated file.
- GUI not so user friendly.
- Existing system does not provide absolute protection for stored data.

Proposed System

We consider data partitioning (both secret sharing and erasure coding) and dynamic replication in data grids, in which security and data access performance are critical issues. More specifically, we investigate the problem of optimal allocation of sensitive data objects that are partitioned by using secret sharing scheme or erasure coding scheme and/or replicated.

Replication techniques are frequently used to improve data availability and reduce client response time and communication cost. One major advantage of replication is performance improvement, which is achieved by moving data objects close to clients. In full replication all servers keep a complete set of the data objects.

In this paper, we propose a concept of system where there would be more secured environment for the privacy of the data stored in grids. Before making this concept we would keep dual layered security in mind, in which one layer would be the general login authentication for the user of the system to access any information. In second layer we can add cryptography feature to the file fragments that should be saved in all other systems over the LAN.

As advancement to the present cryptographic scenario where user encrypts or decrypts the data through the keys, we can generate a case where user need not provide the key but automatically it would be generated and saved in the database. As the person uploads any file the key would be generated which is associated with a respective file and at the time of viewing the file it would check with the login id and password of the user who tries to access the information, if correct then it would find the stored key and decrypt it automatically.

GRID Structure

The grid Architecture involves these following concepts, Data grid, middleware, data fragmentation, data replication.

A. **What is Data Grid?**

A Data Grid is an architecture or set of services that enable individuals or groups of users the ability to access, modify and transfer extremely large amounts of geographically distributed data for research purposes. Data grids make this possible through a host of middleware applications and services that pull together data and resources from multiple administrative domains and then present it to users upon request. The data in a data grid can be located at a single site or multiple sites where each site can be its own administrative domain governed by a set of security restrictions as to who may access the data. Likewise, multiple replicas of the data may be distributed throughout the grid outside their original administrative domain and the security restrictions placed on the original data for who may access it must be equally applied to the replicas. Specifically developed data grid middleware is what handles the integration between users and the data they request by controlling access while making it available as efficiently as possible.

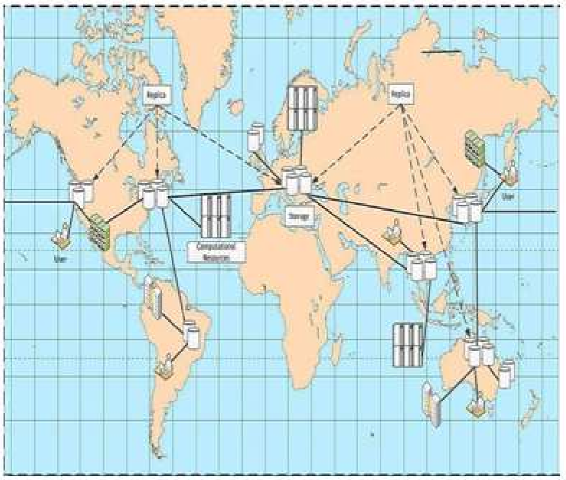

Fig 1.Datagrid

A. **Middleware:**

Middleware provides all the services and applications necessary for efficient management of datasets and files within the data grid while providing users quick access to the datasets and files Data access services work hand in hand with the data transfer service to provide security, access controls and management of any data transfers within the data grid. Security services provide mechanisms for authentication of users to ensure they are properly identified. Common forms of security for authentication can include the use of passwords. Authorization services are the mechanisms that control what the user is able to access after being identified through authentication. The user is able to access after being identified through authentication. The backbone of the grid computing systems is considered to the Grid Middleware. Because the communication across the entire network is not possible without the use of middleware. The purpose of grid middleware is to integrate the heterogeneous resources, efficiently assigning the resources to jobs, monitoring, managing and to provide the secure data access. In the past, while facing the grid infrastructure, many of the existing middleware shows the various limitations. To overcome these limitations, new middleware are proposed. We introduced here some of the most widely used middleware to meet the needs of complex applications.

1. *MPI Based Middleware:* For the development of parallel applications, MPI (Message Passing Interface) is introduced as the main communication library. For grid, new approach MPICH-G2 is introduced that is the implementation of MPI. With the use of MPICH-G2, across the network of computers users can run Message Passing Interface programs with the help of the same command. It also uses the Globus toolkit services. As a result, it provides the better results than its predecessor. Its applications can be seen in world-wide like,
to run and to distribute applications and conventional Message Passing Interface programs across computers located at different sites.

2. *Java RMI Based middleware:* To develop the large scale distributed applications, Java RMI (Remote Method Invocation) based middleware is introduced. For object replication, a new approach is introduced in java that is based on complier. The compiler is helpful for code generation for RMI and to check the consistency. This approach results in better java objects replication and also performance is improved for parallel object based programs.

3. *NetSolve:* In distributed environment, to solve the computational scientific problems, this client/server application is designed. This middleware provides the facility for searching the available resources, selecting the best one resource, and after solving a problem an answer is returned to the user . With the Use of TCP/IP sockets, the communication between NetSolve Agents, Clients and Servers is done. For searching and selecting the best one resource NetSolve agents are responsible.

4. *Globus based middleware:* One of the most widely used middleware is Globus. The most popular Globus toolkit is also provided by Globus. For computational grid, it provides high level services. Various services provides by Globus includes GRAM

ISSN: 2277-9655
Impact Factor: 1.852

(Globus Resource Allocation Manager) that is responsible for resource allocation, monitoring and management services. GSI (Grid Security Infrastructure) that provides single-sign-on authentication and authorization facilities.

5. gLite :For grid computing, gLite is another middleware. One of the unique features provided by gLite is that according to the requirements of users, they can implement the services without using the system as whole. As we know in medical sector, huge amount of data is produced, processed and manipulated. To run the medical image processing applications over the grid infrastructure, Medical Data Management approach is introduced. In this approach, advanced gLite data management services are used by medical data management system to manage and store all the data in secured fashion .

6. Legion By the University of Virginia, Legion middleware project is introduced. For grid applications, Legion is object based meta system software. Legion provides "vault "mechanism for persistent storage. The limitation of Legion is that it does not provide any mechanism to solve the issues like: Data load and Replica management [1].

7. Condor and Condor-G: High throughput computing environment is provides by Condor. To perform the computational tasks, Condor harnesses the capacity of idle workstations. Without modifying the applications, it provides the facility to schedule and monitor the applications. The limitation of Condor is that it does not support the parallel applications [8]. The combination of Condor and Globus results in Condor-G software system. For grid applications, the purpose of Condor-G is to provide the proper job management services.

8 . UNICORE: UNICORE is the Uniform Interface to Computing Resources. It provides the uniform Graphical User Interface (GUI) and security architecture where distributed resources can be accessed in secure fashion .It allows that without the user intervention, data movement function can be performed in well manner [9

9. NIMROD AND NIMROD-G: An interface is provided by Nimrod, where jobs can be independently submitted to a resource management system. A software system Nimrod-g is a combination of Nimrod and Globus. For the scheduling and management of computational resources that are geographically distributed world -wide Nimrod-G is introduced. The result of its implementation shows that how it is scheduling the tasks with in time and cost constraints in a well manner [11].

B. Data fragmentation

Data fragmentation is a process of division or mapping database where the database is broken down into number of parts then stored in the site or units of different computers in a data network, allowing for decision-making to data that has been divided. Data that has broken down is still possible to be combined again with the intention to complete the data collection. When doing fragmentation, data must meet several conditions for the fragment is correct, below is fragmentation principle.

1. *Completeness:* a unit of data that is still in the main part of the relationship, then the data must be in one fragment. When there is a relation, the distribution of the data must be an integral part of the relationship.

2. *Reconstruction*: an original relation can be reused or combined return of a fragment. When it has broken down, data is still possible to be combined again with no change in the structure of data.

3. *Disjointness:* data within the fragment should not be included in the other fragments in order to avoid redundancy of data, except for primary key attributes of vertical fragmentation.

C. Data replication

Replication is known as process of sharing information. (i.e.) storing same data in multiple systems.

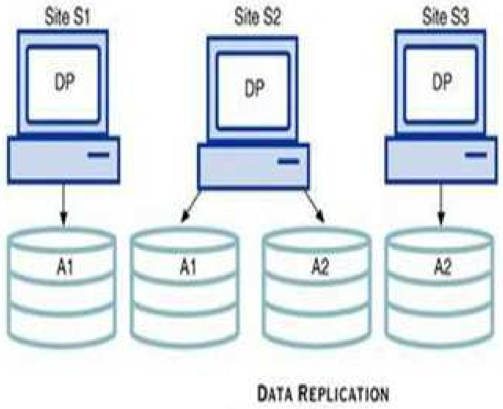

Fig 2.Data Replication

Suppose database A is divided into two fragments, A1 and A2. Within a replicated distributed database, the scenario depicted in the following Figure is possible: fragment A1 is stored at sites S1 and S2, while fragment A2 is stored at sites S2 and S3.

Replication techniques are frequently used to improve data availability reduce client response time and communication cost. Single point failure does not affect this system, Where the data will be secured. One major advantage of replication is performance improvement, which is achieved by moving data objects close to clients. In full replication all servers keep a complete set of the data objects. Three replication scenarios exist: a database can be fully replicated, partially replicated, or un replicated.

1. *A fully replicated database:* stores multiple copies of each database fragment at multiple sites. In this case, all database fragments are replicated. A fully replicated database can be impractical due to the amount of overhead it imposes on the system.

2. A partially replicated database: stores multiple copies of some database fragments at multiple sites. Most DDBMSs are able to handle the partially replicated database well.

3. An un replicated database: stores each database fragment at a single site. Therefore, there are no duplicate database fragments. Several factors influence the decision to use data replication.

4. Database size: The amount of data replicated will have an impact on the storage requirements and also on the data transmission costs. Replicating large amounts of data requires a window of time and higher network bandwidth that could affect other applications.

5. Usage frequency: The frequency of data usage determines how frequently the data needs to be updated. Frequently used data needs to be updated more often, for example, than large data sets that are used only every quarter.

6. Costs: including those for performance, software overhead, and management associated with synchronizing transactions and their components vs. fault-tolerance benefits that are associated with replicated data.

Advantages

- Data will be secured .
- It enables the sharing and coordinated use of data from various resources and provides various services to fit the needs of high-performance distributed and data-intensive computing.
- Replication techniques are frequently used to improve data availability and reduce client response time and communication .
- Single point failure does not affect this system .

Conclusion

We have combined data partitioning schemes (secret sharing scheme or erasure coding scheme) with dynamic replication to achieve data survivability, security, and access performance in data grids. The replicas of the partitioned data need to be properly allocated to achieve the actual performance gains. We have developed algorithms to allocate correlated data shares in large-scale peer-to-peer data grids. Data grid is a distributed computing architecture that integrates a large number of data and computing resources into a single virtual data management system. It enables the sharing and coordinated use of data from various resources and provides various services to fit the needs of high-performance distributed and data-intensive computing. Moreover, it may be desirable to consider multiple factors for the allocation of secret shares and their replicas. Replicating data shares improves access performance but degrades security. Having more share replicas may increase the chance of shares being compromised. Thus, it is desirable to determine the placement solutions based on multiple objectives, including performance, availability, and security.

Future Enhancement

Now we applied only in Data Grid security. In future we can apply at any sort of business application to produce absolute development and with security enhancement.

References

[1] *Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, C. Kesselman, P. Kunszt, M. Ripeanu, B. Schwartzkopf, H. Stockinger, and B. Tierney, "Giggle: A Framework for Constructing Scalable Replica Location Services," Proc. ACM/IEEE Conf. Supercomputing (SC), 2002.*

[2] *Baker, Mark, Rajkumar Buyya, and Domenico Laforenza. "Grids and Grid technologies for wide-area distributed computing." Software: Practice and Experience 32, no. 15 (2002): 1437-1466.*

[3] *Foster, Ian, Carl Kesselman, and Steven Tuecke. "The anatomy of the grid: Enabling scalable virtual organizations." International journal of high performance computing applications 15, no. 3 (2001): 200-222.*

[4] *I. Foster and A. Lamnitche, "On Death, Taxes, and Convergence of Peer-to-Peer*

ISSN: 2277-9655
Impact Factor: 1.852

and Grid Computing," Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS), 2003.

[5] J. Gray, P. Helland, P. O'Neil, and D. Shasha, "The Dangers of Replication and a Solution," Proc. ACM SIGMOD, 1996.

[6] K. Ranganathan and I. Foster, "Identifying Dynamic Replication Strategies for a High Performance Data Grid," Proc. Second Int'l Workshop Grid Computing, 2001.

[7] Karonis, Nicholas T., Brian Toonen, and Ian Foster. "MPICH-G2: a Grid-enabled implementation of the Message Passing Interface." Journal of Parallel and Distributed Computing 63, no. 5 (2003): 551-563.

[8] Krauter, Klaus, Rajkumar Buyya, and Muthucumaru Maheswaran. "A taxonomy and survey of grid resource management systems for distributed computing." Software: Practice and Experience 32, no. 2 (2002): 135-164.

[9] L. Xiao, I. Yen, Y. Zhang, and F. Bastani, "Evaluating Dependable Distributed Storage Systems," Proc. Int'l Conf. Parallel and Distributed Processing Techniques and Applications (PDPTA), 2007.

[10]M. Baker, R. Buyya, and D. Laforenza, "Grids and Grid Technology for Wide-Area Distributed Computing," Software- Practice and Experience, 2002.

[11]Maassen, J., Kielmann, T., & Bal, H. E. (2001). Parallel application experience with replicated method invocation. Concurrency and Computation: Practice and Experience, 13(8‐9), 681-712.

[12]Montagnat, Johan, Ákos Frohner, Daniel Jouvenot, Christophe Pera, Peter Kunszt, Birger Koblitz, Nuno Santos et al. "A secure grid medical data manager interfaced to the glite middleware." Journal of Grid Computing 6, no. 1 (2008): 45-59.

[13]Y. Deswarte, L. Blain, and J.C. Fabre, "Intrusion Tolerance in Distributed Computing Systems," Proc. IEEE Symp. Research in Security and Privacy, 1991.

[14]Global Information Grid, Wikipedia.