ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Efficient Approach for Fault Tolerant Grid Computing in Distributed Data Mining

A.Srinivasa Rao*
Associate Professor,
Department of Information Technology, Sir C.R.Reddy College of Engineering, Eluru,
Andhra Pradesh, India
asr.unguturu@gmail.com

Dr.Ch.Divakar
Principal,
Visaka Institute of Engineering and Technology,
Visakhapatnam,
Andhra Pradesh, India

Dr.A.Govardhan
Professor,
College of Engineering, JNTUH, Hyderabad,
Andhra Pradesh, India

***Abstract*—** Fault tolerance is a key problem in Grid computing in which various kinds of devices are employed In this paper we project an efficient approach to handle failures in DDM services and requirements for fault tolerance in the Grid. This strategy permits user to obtain failure recovery whenever and wherever a crash can happen on a grid node in the network. The implemented Tool has been estimated on a real Grid setting to predict its effectiveness and performance.

Keywords-DDM, Grid, Fault Tolerance, failure handling, failure recovery, crash

I. INTRODUCTION

Grid applications are in general distributed, heterogeneous multi-task applications where the component tasks integrated into the Grid applications could be implemented by arbitrary applications. Each task has its own failure semantics; that is, failure definition and failure handling strategies are specific to the task. For Example, a simulation task requires a certain amount of disk space to save temporary results. If enough disk space is not available, the simulation task will fail due to the lack of disk space. The Grid environment [1] refers to the Internet-connected computing environment in which computing and data resources are geographically dispersed in different domains. There are various kinds of computing resources such as a single PC, workstations, clusters and supercomputers

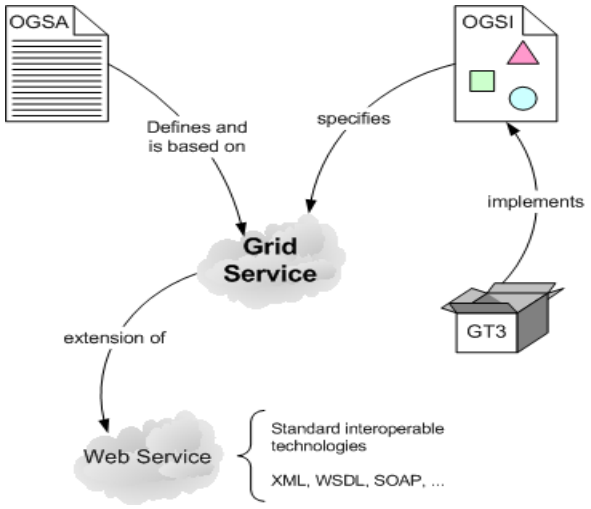

Figure 1: Grid Servicing

The emergence of Grid technology and the ability to provide secure, reliable and scaleable high bandwidth access to distributed data sources across various administrative domains is set to play an important role in the area of data mining. This paper will present the nature of this industry with typical business examples to explain the types of errors encountered in real world data and the fuzzy approach to data matching.

The Grid based Distributed Data Mining architecture [5] presented here is based on the Open Grid Services Architecture model [7] derived from the Open Grid Services Infrastructure specification defined by the OGSI Working Group within the GGF [8]. Open Grid Services Architecture is a service oriented architecture composed of a set of interfaces and their corresponding behaviors to facilitate distributed resource sharing and accessing in heterogeneous dynamic environments [9].

II. LITERATURE SURVEY

In the Literature there has been many systems are mentioned which proposes an interaction protocol and a specification language to meet the unique constraints of delivering data mining e-services. While current data mining query and specification languages focus on description of the data mining process, the proposed specification language captures preferences of clients, capabilities of service providers and specifies mutual access to data and computational resources. This facilitates processes such as selection of service providers and automated interaction to perform a DDM task following the selection. A mapping of the specification language to XML

is developed to support the implementation. The interaction protocol and the specification language have been used to implement a prototype system to demonstrate support for service provider selection through matching, ranking and negotiation as shown in the following figures.

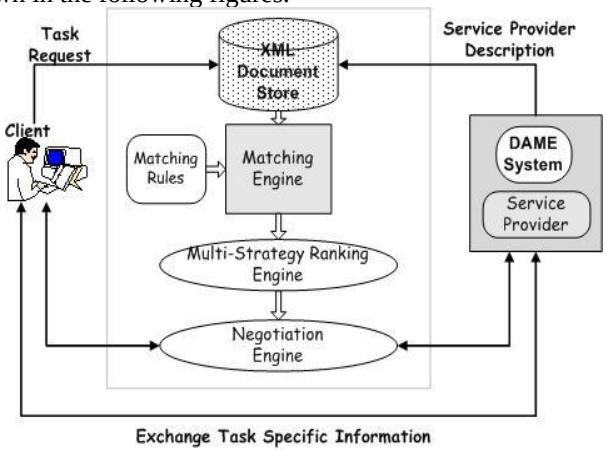

Figure 2: Task Execution

Security for distributed processing is affected by the need to transmit information from one site to another over a potentially nonsecure medium (such as the Internet). This article doesn't cover security other than to note the issues involved and some of the techniques available. One approach to the distributed security-management problem, where many interacting parties may or may not be directly known to one another, is to use the federated network model [2],[3],[4].

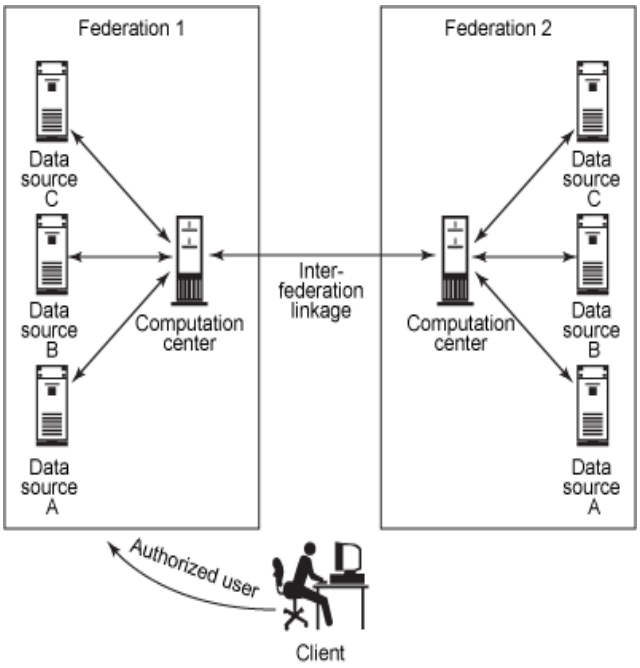

Figure 3: Inter Federation Process

III. PROPOSED WORK

This Part presents the extension of the Mechanism Cited in the Part II with the aim of making it fault tolerant. This paper focus on presenting the fault tolerant Global Miner.

a) Model of Distributed Data Mining

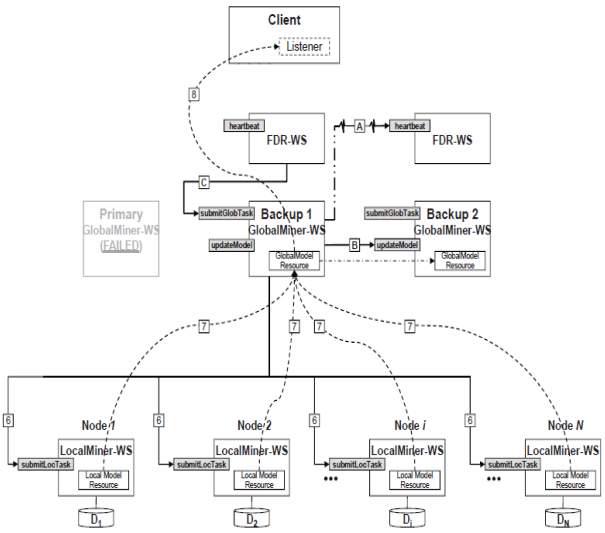

Figure 4: Fault Tolerant Architecture

A. Flow chart:

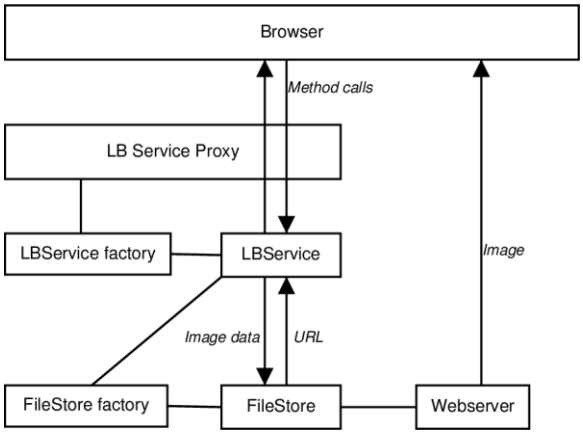

Fig 5: Data Flow Mechanism

The Fault tolerance on the Global Miner has been designed and implementing the primary backup Mechanism. Three main points satisfy the choice of this mechanism rather than the active replication.

First, the primary-backup Strategy, which is simpler since the client has to communicate with only one service (i.e primary) and not with a whole group .Second practically it requires less CPU resources working, because at any instant only one service is running. Third, some algorithms exploiting this strategy assume that random operations (e.g., initialization) are performed on the *Global Miner*, means that an active-replication strategy [6] , that strictly requires the operations on the replica to be deterministic [10], can not be applied in this case. On the other side, a drawback of the primary-backup is that, in case of failure, the client receives the response with delay. Though since this mechanism is oriented to long-running tasks, the cut-over time should result a very short time-overhead for the overall time requested by the complete task execution.

The proposed fault-tolerant framework supposes the presence of a set of *Global Miner* replica, which is the *primary* at any time. The others are named *backups*.

The primary-backup strategy [10] contemplates the following general steps:

a. The client sends the invocation to the *Primary Global Miner*
b. The *primary* receives the invocation and requests for local computations, immediately after such computation results are returned and the *Global Model* has been re-computed, the *primary* sends a model-update message to the *backups*.
c. If the primary crashes in the above point, a new primary is selected among the replicas, and it becomes the new primary of the system.
d. Once the primary has received a reply of the state update from all backups, the response is sent to the client.

The Three stages in any implementation of a primary-backup mechanism are as follows:

i. ***Check pointing:*** The primary time to time sends the change in the *Global Model* (its state) to the backups; it generally consists of storing a snapshot of the current application state. The consistency has to be guaranteed among the backup states. The primary can continue its work (or reply to the client) only when it is known that the backups have applied the state change.

ii. ***Failure detection***: Failure of the primary node can be observed by a periodic message is sent to the backup; if no messages are sent for a given time, then then there is no failure on the primary node.

iii. ***Recovery phase***: Originally, one of the service instances is designated as a primary and others as backups. After a failure of the primary, the backups accept to restart the execution from the last checkpoint state. Hereafter, all future requests are directed to and processed by it.

Let us explain the architecture and approaches implemented for the proposed fault-tolerant frame work. It contemplates the presence of r *Global Miner*, whose one is the *Primary Global Miner* and the others r − 1 are the *Backup Global Miner.* Moreover, the architecture contemplates also r − 1 *Failure Detection and Recovery* each one associated to a backup service

iv. **Task Execution:** Figure 4 shows architecture with the *Primary Global Miner* and Two *Backup Global Miners.* After initialization process the client invokes the *submit Global Task* method of the *Primary Global Miner*, which observe proper executing the task .As seen in the Section II, the global node asks for some local elaborations to the local nodes and waits for their responses. During such a time, the *Primary Global Miner* periodically sends a heartbeat message to the Failure Detection Recovery system, to communicate that it is alive and correctly running. After all the local computations are completed the local models are dispatched to the global location, the integration of all these local models is performed and stored in the *Global Model Resource.* After completion of this process the new *Global Model* is sent to the backups by a synchronous method: The *Primary Global Miner* can send the final model to the client by notification. During Exe3cution the Primary Global Miner sends Heart Beat Message to fault detection and recovery system.

IV. CONCLUSION

Proposed Grid technology presents a framework that aims to provide access to heterogeneous resources in a secure, reliable and scalable manner across various administrative boundaries. The data mining sector is an ideal candidate to exploit the benefits of such a framework. However before widespread adoption happens within this sector a number of fundamental areas will need to be addressed: We use differential privacy to limit the potential information exposure about individual records during the data mining process

V. REFERENCES

[1]. S. AlSairafi, F. S. Emmanouil, M. Ghanem, N. Giannadakis,Guo, D. Kalaitzopoulos, M. Osmond, A. Rowe, J. Syed and . Wendel. The Design of Discovery Net: Towards Open Grid services for Knowledge Discovery. International Journal of high Performance Computing Applications, vol. 17, no. 3, pp. 97-315, 2003.

[2]. B. Schroeder and G. A. Gibson. ”A large-scale study of ailures in high-performance computing systems”. In Proc.of the Int.Conference on Dependable Systems and Networks, 2006.

[3]. D. Talia, P. Trunfio and O. Verta. ”Weka4WS: A WSRFEnabled eka Toolkit for Distributed Data Mining on Grids”.In Proc. 9th European Conference on Principles and Practice f Knowledge Discovery in Databases, 2005.

[4]. X. Zhang, D. Zagorodnov, M. Hiltunen, K. Marzullo and R. D. Schlichting. ”Fault-tolerant Grid Services Using Primary Backup: Feasibility and Performance”. In Proc. of 2004 IEEE International Conference on Cluster Computing, 2004, pp.105-114.

[5]. E. Cesario and D. Talia. ”Distributed Data Mining Models as Services on the Grid”. In Proc. of 10th International Workshop on High Performance Data Mining (HPDM 2008), in conjunction with ICDM’08, IEEE, 2008, pp. 409-495.

[6]. R. Guerraoui and A. Schiper. ”Fault-Tolerance by Replication in Distributed Systems”. In Proc. of Conference on Reliable Software Technologies, 1996, pp. 38-57.

[7]. OGSA http://www.globus.org/ogsa/

[8]. http://www.gridforum.org/ogsi-wg/

[9]. S. Burbeck, “The Tao of e-Business Services, ”IBM,Corporation(2000);http://www.4.ibm.com/software/developer/library/ws-tao/index.html.

[10]. R. Guerraoui and A. Schiper. ”Fault-Tolerance by Replication in Distributed Systems”. In Proc. of Conference on Reliable Software Technologies, 1996, pp. 38-57.