An Effectual Failure Factor Augmented Aggregation Techniques for Computational Grid

Nini Elsa Shaji#1, Shamila Ebenezer A*2

#Post – Graduate Student, Department of Computer Science and Engineering,
Karunya University, India

*Assistant Professor, Department of Computer Science and Engineering,
Karunya University, India

***Abstract*— Information aggregation is a solution for reducing information being interchanged between Grid networks. Resource manager consider the scheduling decisions by using this aggregated information. Aggregated information is kept across each node and the detailed information is kept private, but the resources are available publicly for use. This paper gives an idea on aggregating resource information which includes the failure factor of resources along with other parameters like computational capacity, task queued in each resources and the time availability of the resources and its implementation details. Aggregating this information will help in scheduling the task to each of the resources in an efficient way, so that the task completion will not be hindered.**

***Keywords*— Grid Computing, Computational Grid, Information Aggregation, Failure Factor, Scheduling**

I. INTRODUCTION

Due to the different developments in science and engineering, problems in computation fields are becoming perplexed. In-order to solve these problems, we need to aggregate and use the computer resources garbled around the world. Hence we apply the conception of Grid computing. Grid computing is a way to combine the geographically spread and heterogeneous resources that belongs to unlike domains. The resource manager in Grid gets different user request and have to assign different task to resources. Some of the scheduling decisions include static and dynamic resource information such as storage and computation capacity, number of tasks queued in each resource etc.

Some characteristics of Grid like dynamicity, autonomy and heterogeneity makes it hard for resource scheduling in a Grid environment. Another ground to deal with is information that is being transferred to central monitor. When allocation of job comes, resources with good execution may get assigned to jobs, thus clogging the resource and cutting down the overall balance of the system. This may guide to diluted overall system performance. This paper provides implementation details of resource information aggregation with the help of parameters such as computational capacity, time availability of the resources, task queued in each resource and finally the failure factor of each resources. Also the paper contains the scheduling of task to the resources.

This paper is organized as follows. Section 2 includes a discussion on key concepts presented in this paper. Section 3 gives an idea on the related work or the literature survey that has been done in different networks. Section 4 presents the proposed algorithm that is being implemented. Section 5 gives the implementation details of the same. Section 6 gives the implementation results. Section 7 at the end gives the conclusion of this paper.

II. KEY CONCEPTS

A. Grid Computing

Grid computing essentially means the use of multiple resources to solve a single problem. At the same time this demands large number of computer cycles and vast amount of data. Application of grid computing is usually used in scientific and research projects were volunteer computers brought together to do research projects by extending client programs in the participating computers. Grid system has various features [1] like dynamicity, heterogeneity, large-scale, resource sharing etc. It is said to be dynamic because of the constant changes in resources and tasks. Again grid is an interconnected set of resources and tasks of wide variety, which makes it heterogeneous and grid system must cater to the requirements of users in large-scale, i.e. it is capable enough to deal with large number of resources and tasks.

B. Information Aggregation

Information aggregation [2] refers to summarization of resource information in a Grid network and this information is handed over to resource manager in order to induce scheduling decisions. As size of grid network raises, resource-related information size and dynamicity also springs up quickly, thus creating the aggregation and use of this massive amount of information turn into a challenge for resource management system. The computation and storage tasks are led non-locally with finer degree of granularity, so the flow of information between different systems throughout multiple domains will increase. Aggregation techniques are significant in order to dilute the amount of information being exchanged and frequency of these exchanges, at the same time maximizing its value to Grid resource manager.

Another significant need of information aggregation is confidentiality and interoperability. This means that, when more resources or domains of resources take part in Grid, it is suitable to keep sensitive and elaborated resource information private, while resources will be still available publically for use. By this way task scheduler will be able to efficiently and transparently use the resources. In all case, key to information aggregation is the degree to which the summarized data helps in carrying scheduling decision so as to make efficient use of resources.

Quality of information aggregation schemes [3] is assessed by its effect on efficiency of scheduler's decision and also by the reduction it brings in the total resource information that is being transferred to the central monitor.

C. Clustering

Clustering is the method of grouping computers, workstations, storage devices, interconnections etc in order to supply high availability of resources. It renders users with a highly available resource and appears to the users as a single large computer. These are spread for improved performance. Clusters can be categorized as hierarchical clusters [4], compute clusters, etc.

III. RELATED WORKS

This section contains various aggregation techniques in different networks. This can be used for a comparative study of how different information aggregation techniques work in different networks.

A. Tree Based Aggregation in Hierarchical Networks

In a sensor network, nodes are formed into tree [5]. Data aggregation happens at in-between nodes in the tree and brief information is communicated to root node of the tree. This type of aggregation is suited for applications that involve in-network data aggregation. One of the principal faces of tree based network is that, it has the potential to construct an energy efficient data aggregation tree. Data aggregation tree in sensor network commences with a sink which carries control message. Role of root node of aggregation tree is accepted by the sink. Control message comprises of five fields: ID, parent, power, status and hopcnt which shows the sensor ID, its parent, its residual power, the status and the number of hops from the sink respectively.

B. Distributed Aggregation Trees on a Structured P2P Networks

Distributed information aggregation has universal application on several distributed system, such as grid resource monitoring [6], P2P reputation aggregation [7], etc. The DAT [8] trees are constructed implicitly from native Chord routing paths. This has been done by leveraging Chord topology and routing mechanisms. DAT trees have to be balanced; with this intention a balanced routing algorithm on Chord is employed. This algorithm will dynamically choose the node's parent from its finger nodes by estimating its distance to the root. Nearly all aggregate functions are reducible, such as count, min, max, sum, average etc. There are two crucial properties for these reducible aggregation functions. Firstly, the function either give a single value from set of all values (example is min and max), or compute some property of all the values (example is count and sum). The output value will be very much smaller in size than the set of input values in both events. Secondly, aggregation functions are employed to a big set of inputs recursively. In order to figure out the above aggregation trouble, DAT trees are applied. In DAT, the aggregate function is enforced to each of the node on the values of its child node and the aggregated information is sent to the parent node. This is a recursive procedure and is in a bottom-up fashion.

C. Trust Vector Aggregation Algorithm in P2P Networks

Trust vector based scheme i.e. VectorTrust [9] for aggregation of distributed trust scores is issued with the coming forth internet-scale open message and resource sharing, social networks etc. Characteristics of VectorTrust are localized and distributed concurrent communication. By nature a VectorTrust enabled system is decentralized and does not trust on any centralized server or trust aggregation. VectorTrust system is constructed on a trust overlay network that is on lead of P2P networks. The trust overlay network is just like a directed graph. Vertices of the network match to peer in a system and the directed edge will be demonstrated from one vertex to another vertex, if and only if the first vertex is a client of the second vertex in a direct deal.

A real number ranging between 0 and 1 is assigned for each of the directed edge and this number represents how much one vertex trust other one. If the number is 0, then the vertex never believes the other one and if it is 1, the vertex trusts other vertex 100%. This real number is the trust rating and the link with trust rating is called Trust Vector. In the system with each direct transaction, a direct trust link is generated by the taking part peers. It also assigns a trust rating in order to constitute the quality of particular transaction. Each transaction in system can either adds a new directed edge in trust graph, or re-labels the value of an existing edge with its new trust rate.

D. Simple Aggregation Algorithm in High Performance Computing

In a practical grid system, which was framed of different domains, resource model is essential. One resolution is to use a common resource model between different domains. Since interoperable grid systems are compiled of numerous domains, there is a scalability issue with the quantity of resource information exchanged among brokers. In order to preserve data transferred, latency time and communication bandwidth interchange of resource information is exercised in an aggregated form. Trouble with aggregated data is the loss of points related to each resource description, but this summarized or aggregated information is plentiful for the choice of best broker to submit a job. Simple aggregation algorithm [10] aggregates the data as far as possible. Here algorithm looks for utmost compression for scalability.

Because of this, it misses more elaborated information. Input for the algorithm is a set of resources and its relationship that determines the computer and some fixed attributes. Let's take three fixed attributes for combining the information as an example. Let first be processor type for Computing System resource, and then second be operating system type for Operating System resource, and at the last let it be file system type for File System resource. End product of this aggregation algorithm will be a set of resources in aggregated figure and a set of relationships that shows original resource. Examples of this aggregated form is count of resources in same family, maximum or minimum values and the sum of all values i.e., the total. Depending on sort of resources, aggregated information differs. Once resource aggregation is performed, algorithm searches for correspondences with original form in order to create resource relationship.

E. Categorized Aggregation Algorithm using Grid Broker Selection

This algorithm [10] attempts to discover a good balance among resource data and scalability. Categorized aggregation algorithm regards dissimilar attributes and threshold values, in addition to input set of resources, relationships and fixed categories. From the name of algorithm itself, we can say that the resources are aggregated into categories and subcategories. There will be determined categories and attributes and thresholds are set as subcategory with in this category. These subcategories raise accuracy of the aggregated information. It is possible to use any number of attributes so that the elaborated information will be ample. Degree of details can be increased by defining more threshold values, thus it is possible to avoid loss of significant resource features and also it is possible to preserve benefits of aggregation. Algorithm computes category and subcategory of resources with the help of attributes, values and a set of thresholds that define the discrete categories.

F. Topology Aggregation Mechanisms for Delay Bandwidth Sensitive Networks

Primary aim of topology aggregation [11] is to simplify routing by summarizing and compressing information of lower levels and passing them to logical higher levels. Balance amongst accuracy and performance should be taken in to consideration because topology aggregation gives rise to distortion. Generally a network comprises of nodes and links, where nodes are originator and receiver of information while link is the one that serves as conveyer by connecting the nodes. Topology aggregation has two steps. First is to determine an appropriate staircase, which will help in symbolizing properties of different physical paths within the border nodes. Using this staircase and border nodes, each domain is translated to a full mesh topology. Second step is to transform the full mesh topology to a maximum weighted spanning tree and a star structure. For most of aggregation schemes, full mesh approach is regarded as foundation.

G. Directed Diffusion in Wireless Sensor Networks

Directed diffusion [12] is a popular data aggregation strategy in wireless sensor networks. It is data centric, which means all communication is for named data. All nodes in this network will be application aware. Advantage of directed diffusion is that it attains energy savings. Various elements are involved in directed diffusion such as interests, data messages, gradients and reinforcements. Interest message is a query which determines the demand of a user. It also contains description of a sensing task for gaining data that is held by a sensor network. These data is represented or named using attribute-value pairs. In a network, an interest is usually put into some nodes in network and these nodes are called as sink. For each task, sink will disseminate interest to its adjacent nodes. Every node keeps an interest cache, which has various areas like timestamp, duration, expiresAt etc. If interest is identical to each other, it is possible to have interest aggregation. When a node gets an interest, the node first sees whether that particular interest is already present in interest cache or not. If a match exists, data message is dropped silently. Otherwise, received message is contributed to data cache and data message is send out to the neighboring nodes.

H. Node and Link Aggregation in Hierarchical Networks

In hierarchical network, mainly two steps are involved in aggregation [13]. That is node aggregation within peer groups and succeeded by link aggregation between peer groups. Hierarchical structure is extremely suitable and efficient. In node aggregation one or more nodes in network are substituted with a single complex node and this is made transparent to nearest neighboring node. A node relating to the complex node is called ingress or egress nodes. Input / output relation back in an information network is qualified by QoS measures. In order to figure out complex node, nucleus node is determined. Ingress node to this nucleus is named as spoke. QoS measure is calculated for each spoke. Thus QoS value for each of the link is known. Then the best QoS measure from all potential paths between nuclei of peer groups. This value will be the aggregated one. In case, if QoS measure is additive, then the aggregated link will be the one that minimizes the sum of QoS values. Process for link aggregation will give the best link as aggregated logical link among two complex nodes. This aggregated logical link extends maximum condensation.

IV. PROPOSED SYSTEM

In existing system [14], several techniques such as single point aggregation scheme, intra – domain clustering aggregation scheme, reducing aggregated information using domination relations and border node – site pair scheme for resource information aggregation in grid networks has been proposed. In these techniques, each site is assigned a parameter or information vector that records the computation and storage capacity, resources time availability and number of tasks queued in each resource. Then this information vector of each site that belongs to a particular domain is aggregated into a smaller vector using appropriate aggregation technique. Parameter of each site that belongs to a particular domain is aggregated together and is passed to the domain monitor and from there it will be passed to central monitor. When a task is created, it will be allocated to any particular domain by central scheduler using this aggregated information. Domain scheduler that is present in the domain will allocate this task to any particular site with the help of original information present in domain monitor.

Main disadvantage is in the scheduling of a task to a particular resource. For example, consider a task which needs a computational capacity of 2000 MIPS is produced by a user. Scheduler will be checking for the information vector in each domain and the task will be allocated to a particular resource. If the resource's computational capacity to which the task has been allocated is less than that is needed by the task, and then the completion of that particular task will be failed completely. Again when a similar task is produced, there is a chance for the scheduler to allocate the new task to the same failed resource without knowing failure of the resource. Thus failure information of the resource is not considered for the allocation of task next time.

In this proposed system, failure factor or information of the resources is also considered. Thus whenever a resource is failed to complete a particular task, that information will be stored in a database. This is done for each of the resources. While aggregating the parameters of each resource to a vector, this information that is the failure factor of resources is also considered along with the other parameters like computational capacity, task queued in each resource and time availability of the resources. For this purpose, a new simulation environment is simulated using the gridsim resource failure class feature. This class is used by the users to check whether the resource is working or is totally failed. Gridsim resource failure class is an extension of GridResource class with added failure functionalities.

Whenever a resource is failed to complete the task, the number of machines in each resource is added to the database. Thus while information is being aggregated, failure factor that has been stored in database is retrieved and is aggregated with other parameters or characteristics. Aggregating the information vector using all these characteristics will help scheduler to schedule the task to appropriate resource that will complete the job. Thus we can have guarantee that resources will not fail from completing the job or task allocated to it.

V. IMPLEMENTATION OF PROPOSED SYSTEM

The whole project has been divided into different modules as follows:

A. *Module 1: Simulation Environment Creation*

A new simulation environment oriented for evaluation demands for aggregation schemes has been created. This simulation model tries to abstract features of any distributed computation environment. Through this environment several resources are created with four parameters (computational capacity, number of tasks queued in resources, time availability of resources and resource failure factor). Let's assume that there is a central scheduler for all Grids and a local scheduler for each domain.

First parameter, which is computational capacity, is static in nature as shown in Table I [15] and this is similar in nature to other static parameters like the number of CPU of a resource, storage capacity, its cost of use, etc. Computational capacity of each resource is assessed in MIPS and is distributed uniformly. Other two parameters are dynamic in nature. Their values are creating randomly, using random class in java. Value for parameter "number of tasks queued" ranges from 9 to 29 and for "availability of resources" from 49 to 349. Fourth parameter that is resource failure factor is updated from database. During the process of simulation, when GridSim resource failure class is used some resources are put in failed list and this list is updated in database. Thus the data that is stored in the database is considered as historical data for the failed resources. Regional GIS that is domains are created and resources are added or grouped together randomly.

This simulation is creating two users who in turn create tasks. These tasks are created with exponentially distributed inter-arrival times and their workload follows a uniform distribution within the range of 50000 to 100000 millions of instruction. They are created exponentially with varying characteristics and is then submitted to central scheduler for scheduling.

TABLE I
COMPUTATIONAL CAPACITY OF RESOURCES

Resource	**Computational Capacity (MIPS)**
Resource_0	1000
Resource_1	1100
Resource_2	1200
Resource_3	1300
Resource_4	1400
Resource_5	1500
Resource_6	2000
Resource_7	2200
Resource_8	2400
Resource_9	2600
Resource_10	3200
Resource_11	3600
Resource_12	3800
Resource_13	4000
Resource_14	4300

B. *Module 2: Aggregation Schemes*

1) Single point aggregation scheme with added parameter: A resource or a site consists of several numbers of machines and the machines contain several number of processing elements (PE). So capacity of each site will be the sum or the total of capacity of each processing elements that is present in each resource. While aggregation is done, information of each resource is added up and is given or passed to the central monitor. In single point aggregation scheme, the values of each corresponding parameter of the site in a domain is added together to get an aggregated value.

2) Intra – domain clustering aggregation scheme with added parameter: In intra-domain clustering algorithm, sites or resources are clustered randomly as intra-domain clusters based on the parameters that are almost similar to each other. Then corresponding values of these intra-domain clusters are aggregated and are passed to the central monitor. Here the aggregation process is similar to single point aggregation scheme, but the difference is that within the domain itself two or more clusters with one or more resources are made and aggregation is done on these clusters. For the resources to be grouped, first some resources are taken randomly. Then the distances between these taken resources and other resources will be calculated. From that distance the resource will be put to any randomly selected resource, if its distance is less than the other resources. A file named cluster.txt is created for writing the aggregated information and writes back to output screen.

3) Reducing aggregated information using domination relations with added parameter: A dominance relation is found out between the clusters that are created in a particular domain. Then with help of this dominance relation information is being aggregated. Dominance relation that has been used is greater than or less than relation. After applying this dominance relation, the cluster that has been selected will only be aggregated and passed to the central monitor for the purpose of scheduling task that is created by the user.

4) Border node – site pair scheme with added parameter: Here in border node-site pair scheme, first of all a border node is selected. Then any of the aggregation techniques mentioned above in this section is applied on this border node. This aggregated value is then passed to the central monitor. Thus in each GIS only one aggregated information is present.

C. Module 3: Task Scheduling

Task scheduling is done by the central monitor. When a task arrives, central scheduler selects a proper domain for execution of this new task. Using the exact aggregation information from domain matrices, central scheduler selects information vector that is having highest value. Thus from the central scheduler task will be passed to domain scheduler. Then domain monitor will select the appropriate resource for the task completion from original aggregated information that is present in the domain monitor.

VI. EXPERIMENTAL RESULTS

In this paper a simulation environment is created. Then two users have been created, who in turn creates the task that has to be allocated to the resources. Thirty resources have been created with computational capacity as shown in Table I. A resource or a site contains several numbers of machines and these machines contain several number of processing elements (PE). This project is done or is implemented in netbeans, using gridsim package.

Here all specified aggregation techniques have been implemented as explained above. After its implementation and evaluation, it is possible to tell that among all the four aggregation techniques that is specified in this paper. As the number of domains and the computational capacity of a resource increases, the aggregation that works out is the single point aggregation scheme. If number of domains is more and the computational capacity of the resources is less, then the best aggregation technique is border node – site pair scheme. In this simulation, in order to show that some resources are being failed to complete the task which is allocated to it, GridSim resource failure class is been used for implementation of these aggregation schemes.

VII. CONCLUSION

This paper shows the successful implementation of different aggregation techniques that involves different parameters of resources like resource failure factor, computational capacity, number of task queued in each resource and the time availability of the resources. These parameters are recorded in each site as in information vector.

Since existing system [14] is not having resource failure factor as one of the parameters to be aggregated, thus the proposed system with this parameter enables the central monitor to take efficient decisions in task scheduling. Aggregation also helps in achieving large information reduction. It is also discovered that uniformity of the site's and the task's characteristics importantly affects quality of the aggregated information. Again alters in dynamic characteristic of the resources will not always broadcast to the central monitoring system. This is because aggregated information vector sometimes take up these changes.

REFERENCES

[1] Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis and Eduardo G'omez-S'anchez, "Grid characteristics and uses: a grid definition", in: Proc. the First European Across Grids Conference, ACG'03, , pp. 291–298, 2004.

[2] Panagiotis Kokkinos and Emmanouel Varvarigos, "Data Consolidation and Information Aggregation in Grid Networks", Advances in Grid Computing, pp. 95-118, 2008.

[3] P. Kokkinos and E.A. Varvarigos, "Scheduling efficiency of resource information aggregation in grid networks", Future Generation Computer systems 28, pp. 9–23, 2012.

[4] Katherine A Heller and Zoubin Ghahramani, "Bayesian Hierarchical Clustering", Gatsby Computational Neuroscience, U. London, 2000.

[5] Ramesh Rajagopalan and Pramod K. Varshney, "Data aggregation techniques in sensor networks: A survey", Department of Electrical Engineering & Computer Science, Syracuse University.

[6] S. Czajkowski, K. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing", in Proc. of High Performance Distributed Computing Conference, 2001.

[7] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks", in Proc. of the 12th International conference on World Wide Web, pp. 640.651, 2003.

[8] M. Cai and K. Hwang, "Distributed aggregation algorithms with load-balancing for scalable grid resource monitoring", in: IEEE International Parallel and Distributed Processing Symposium, 2007.

[9] H. Zhao and X. Li, "VectorTrust: Trust Vector Aggregation Scheme for Trust Management in Peer-to-Peer Networks", The research

presented in this paper is supported in part by National Science Foundation (CNS-0709329), pp. 1-6.

[10] I. Rodero, F. Guim, J. Corbalan, L. Fong and S.M. Sadjadi, “Grid broker selection strategies using aggregated resource information”, Future Generation Computer Systems 26 (1), pp. 72–86, 2010.

[11] J. Zhang, Y. Han and L. Wang, “New Topology Aggregation Mechanisms for Delay bandwidth Sensitive Networks”, 1-4244-2424-5/08/$20.00 IEEE, pp. 737–742, 2008.

[12] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, “Directed diffusion for wireless sensor networking”, IEEE Transactions on Networking 11, pp. 2–16, 2003.

[13] P. Van Mieghem, “Topology information condensation in hierarchical networks”, The International Journal of Computer and Telecommunications 31 (20), pp. 2115–2137, 1999.

[14] P. Kokkinos and E. A. Varvarigos, “Scheduling Efficiency of Resource Information Aggregation in Grid Networks”, Future Generation Computer Systems, Vol. 28, pp. 9 – 23, 2012.

[15] Ruay-Shiung Chang, Chun-Fu Lin and Jen-Jom Chen, “Selecting the most Fitting Resource for Task Execution”, Future Generation Computer Systems, Vol. 27, pp. 227 – 231, 2011.