Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Construction of Virtual Organization with Security and Fault Management Mechanism

Ms. P. Priyanka, IInd ME., (CSE)
Mr. C. Eyigee M.E., M.B.A., Assistant Professor, Dept. of CSE,
Mr. T.P. Jayakumar. M.E., (Ph.D)., AP/HOD/CSE
Maharaja Engineering College for Women, Perundurai, Tamilnadu, India

Abstract - A computational grid is a hardware and software infrastructure. Grid provides dependable, consistent and inexpensive access to high-end computational facilities. Globally distributed data and resources are scheduled to the tasks. Computational grids and Data grids are the 2 major types of the Grid environment. Processor and network bandwidth resource are shared under computational grids. Data Grid provides a scalable infrastructure for managing and storing data files. Scientific applications produce huge volume of data files. Mutual and economic resource sharing schemes are available in the grid environment.

GSPs are permanently connected with the Virtual Organizations under the static VO model. Dynamic VOs are formed to execute a given task and once the task is completed they are dismantled. The life cycle of a VO can be divided into four phases. They are identification, formation, operation and dissolution. The possible partners and the VO's objectives are identified in the initiation phase. The terms, goal and the duration of collaboration are finalized in the formation phase. The operation phase is involved to solve the specific task. In the dissolution phase the VO is dissolved after the completion of the tasks. The Merge-and-Split VO Formation Mechanism (MSVOF) allows the GSPs to make their own decisions to participate in VOs. The MSVOF algorithm constructs the VO with cost and deadline considerations. The selfish split rule is used to find the optimal VO in the VO structure. Branch-and-bound method is used for mapping of the tasks to each of the VOs that minimizes the cost of execution.

The MSVOF mechanism is enhanced to support trust and security for the GSPs. The system provides the security for resource information and data values. Fault tolerant resource scheduling model is integrated with the VO scheme. The VO construction process upgraded with communicational aspects.

I. Introduction

Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed "autonomous" resources dynamically at runtime depending on their availability, capability, performance, cost and user's quality-of-service requirements Grid applications are special class of distributed applications that have high computing and resource requirements, and are often collaborative in nature. Grid Computing can be denned as applying resources from many computers in a network at the same time to a single problem; usually a problem that requires a large number of processing cycles or access to large amounts of data.

At its core, Grid Computing enables devices, regardless of their operating characteristics to be virtually shared, managed and accessed across an enterprise, industry or workgroup. This virtualization of resources places all of the necessary access, data and processing power at the getups of those who need to rapidly solve complex business problems, conduct compute-intensive research and data analysis, and engage in real-time. The grid computing enables the virtualization of distributed computing resources such as processing, network bandwidth and storage capacity to create a single system image, granting users and applications seamless access to vast IT capabilities. Just as an Internet user views a united instance of content via the Web, a Grid user essentially sees a Single, large virtual computer.

@IJMTER-2015, All rights Reserved

II. Related Work

There are several studies on how to measure reputation in different domains. These studies model different problems using graphs in which the weight associated with each edge represents the value of trust. One approach makes use of trust propagation to find the reputation [1]. If two nodes are not adjacent, each node can evaluate the value of trust to another one using an existing path between them based on the trust transitivity property. That is, if A trusts B and B trusts C, then A trusts C to some extent. Hang *et al.* [1] defined three operators, aggregation, concatenation, and selection, in order to improve the accuracy of trust propagation. They also considered the selection of the path with the highest propagated trust value in cases in which multiple paths exist between two participants. A social trust inference algorithm was designed by Kuter and Golbeck [2] to estimate the confidence in the trust. This algorithm is based on probabilistic reasoning where the confidence is calculated based on the preference similarity. Confidence and trust values are propagated over the entire network. Another approach proposed by Agrawal *et al.* [3] is based on the use of the network flow to find the reputation. Many reputation systems are designed based on graph centrality measures [4]. These studies focused on the centrality of nodes within a graph. The centrality of a node determines the reputation of the node among all nodes. Various centrality metrics were defined such as degree centrality, between's centrality, closeness centrality, and eigenvector centrality [8].

Trust is one major concern when establishing sharing relationships among the GSPs in a grid system. Azzedin and Maheswaran [9] proposed a trust model for grid systems that considers the trust between GSPs and users. Their model assumed that the trust and the reputation decay with time. The amount of trust between two participants is a weighted sum of the direct trust and reputation. They used trust in three heuristic mapping algorithms: minimum completion time, Min-min, and Sufferage. The simulation results showed improvement in the overall quality of the schedules in terms of utilization and average completion time. The assumption of decaying trust and reputation with time limits the applications of this method in grids. This method converges to a state in which the formation of new VOs is not possible. GSPs form VOs and as a result would tend to just trust the members of their respective VOs. Lin and Huai [5] proposed a method in which a GSP decides to allocate resources based on the combination of the trust value and the bidding price of a requester. Their proposed method, QGrid, is based on Qlearning techniques that balance the relative importance of trust and price. QGrid is a distributed method for computing the reputation.

The combination of reputation and global trust was used to build a grid reputation management framework called GridEigenTrust. Reputation in GridEigenTrust is determined using the eigentrust algorithm proposed by Kamvar *et al.* , while the global trust is computed using the method proposed by Azzedin and Maheswaran. In GridEigenTrust, a GSP selects trusted resources and GSPs to satisfy the requirements of the application based on a hierarchical process. Each organization has a set of entities: resources, GSPs and users. A VO is a set of organizations or some parts of organizations. A hierarchy consists of entities, organizations, and VOs. A reputation is assigned to each entity. The reputation of the organization is computed based on the reputation of the entities that are part of the organization. Finally, the reputation of a VO is computed based on the reputations of its component organizations. The VO formation problem was not considered by the authors. To the best of our knowledge, our paper is the first to take into account the global trust of the GSPs in the VO formation process.

III. Construction of Virtual Organization in Grids

Grid computing systems enable efficient collaboration among researchers and provide essential support for conducting cutting-edge science and engineering research. These systems are composed of geographically distributed resources owned by autonomous organizations. Resource management in such open distributed environments is a very complex problem, which if solved leads to efficient utilization of resources and faster execution of applications. The existent grid resource management systems do not explicitly address the formation and management of virtual organizations (VO). We argue that incentives are the main driving forces in the formation of VOs in grids and thus, it is imperative to take them into account when designing VO formation mechanisms. To provide better performance and increase the efficiency, it is essential to develop mechanisms for

@IJMTER-2015, All rights Reserved

VO formation that take into account the behavior of the participants and provide incentives to contribute resources.

A VO is "a collection of geographically distributed functionally and/or culturally diverse entities that are linked by electronic forms of communication and rely on lateral, dynamic relationships for coordination". While this definition covers a wide range of VOs, this paper focuses on more specific types of VOs, those that emerge in the existing grids [9]. The VOs we are focusing on are alliances among various organizations that collaborate and pool their resources to execute large-scale applications. More specifically, we will consider VOs that form dynamically and are short lived, that is, they are formed to execute a given task and once the task is completed they are dismantled.

The life cycle of a VO can be divided into four phases: identification, formation, operation and dissolution. During the initiation phase, the possible partners and the VO's objectives are identified. In the formation phase, the potential partners negotiate the exact terms, the goal and the duration of collaboration. Once the VO is formed, it enters the operation phase in which the members of the VO collaborate in solving a specific task. Once the VO completes the task, it dissolves. This paper focuses on designing mechanisms for the second phase, the formation of VOs. We model the VO formation as a coalitional game, where GSPs decide to form VOs in such a way that each GSP maximizes its own profit, the difference between revenue and costs. The VO formation phase can be further divided into three subphases: coalitional structure generation, solving the task allocation problem of each coalition and dividing the value among the members of the coalition. In the first subphase, the set of GSPs is partitioned into disjoint VOs. In the second subphase, the task assignment that maximizes the profit of the participating GSPs in each VO is determined. In the third subphase, the profit obtained by the VO is divided among the members of the VO. A GSP will choose to participate in a VO if its profit is not negative. The VOs provide the composite resource needed to execute applications [10]. A VO is traditionally conceived for the sharing of resources, but it can also represent a business model. In this work, a VO is a coalition of GSPs who desire to maximize their individual profits and are largely indifferent about the global welfare. We design a VO formation mechanism based on concepts from the coalitional game theory. The model that we consider consists of a set of GSPs and a grid user that submits a program and a specification consisting of a deadline and payment. A subset of GSPs will form a VO to execute the program before its deadline. The objective of each GSP is to form a VO that yields the highest individual profit.

We address the problem of VO formation in grids by designing a mechanism that allows the GSPs to make their own decisions to participate in VOs. In this mechanism, coalitions of GSPs decide to merge and split to form a VO that maximizes the individual payoffs of the participating GSPs. The mechanism produces a stable VO structure, that is, none of the GSPs has incentives to merge to another VO or split from a VO to form another VO. We propose the use of a selfish split rule to find the optimal VO in the VO structure. The mechanism determines the mapping of the tasks to each of the VOs that minimizes the cost of execution by using a branch-and-bound method. As a result, in each step of the mechanism the mapping provides the maximum individual payoffs for the participating GSPs that make the decisions to further merge and split. We analyze the properties of our proposed VO formation mechanism and perform extensive simulation experiments using real-workload traces from the Parallel Workloads Archive [7]. The results show that the proposed mechanism determines a stable VO that maximizes the individual payoffs of the participating GSPs.

IV. Problem Statement

GSPs are permanently connected with the Virtual Organizations under the static VO model. Dynamic VOs are formed to execute a given task and once the task is completed they are dismantled. The life cycle of a VO can be divided into four phases. They are identification, formation, operation and dissolution. The possible partners and the VO's objectives are identified in the initiation phase. The terms, goal and the duration of collaboration are finalized in the formation phase. The operation phase is involved to solve the specific task. In the dissolution phase the VO is dissolved after the completion of the tasks. The Merge-and-Split VO Formation Mechanism (MSVOF) allows the GSPs to make their own decisions to participate in VOs. The MSVOF algorithm constructs the VO with cost and deadline considerations. The selfish split rule is used to find the optimal VO in

@IJMTER-2015, All rights Reserved

the VO structure. Branch-and-bound method is used for mapping of the tasks to each of the VOs that minimizes the cost of execution. The following problems are identified from the existing system.

- Trust relationships between GSPs are not considered
- Resource failures are not handled
- Security mechanism is not provided
- Communication load factors are not considered

V. Merge-and-Split VO Formation (MSVOF) Scheme

The proposed MSVOF scheme is given in Algorithm 1. The scheme is executed by a trusted party that also facilitates the communication among VOs/GSPs. The design of the mechanism assumes that the players report their true execution speeds and costs to the trusted party executing the mechanism. This is needed to guarantee the stability of the VO formed by the scheme. In practice, the mechanism will require the verification of these parameters as part of each GSP's agreement to participate in the scheme. Achieving truthfulness in this context without such verification procedures is a difficult task that needs significant further research.

Algorithm 1. MSVOF Mechanism

1. $CS = \{\{G_1\},\ldots..,\{G_m\}\}$
2. Map program T on each $S_i \in CS$
3. **repeat**
4. stop $\leftarrow$ true
5. **for** all $S_i, S_j \in CS$ $i \neq j$ **do**
6. visited $[S_i][S_j] \leftarrow$ False
7. **end for**
8. {Merge process starts:}
9. **repeat**
10. flag $\leftarrow$ True
11. Randomly select $S_i, S_j \in CS$ for which Visited$[S_i][S_j]$=False $i \neq j$
12. Visited $[S_i][S_j]$ = True
13. B&B-MIN-COST-ASSIGN $(S_i \cup S_j)$ {Map program T on $S_i \cup S_j$ }
14. **if** $S_i \cup S_j \blacktriangleright_m \{S_i, S_j\}$ **then**
15. $S_i \leftarrow S_j \cup S_j$ {merge S_i and S_j }
16. $S_j \leftarrow \emptyset$ { S_j is removed from CS}
17. **for all** $S_k \in CS$, $k \neq I$ **do**
18. visited $[S_i][S_k] \leftarrow$ False
19. **end for**
20. **end if**
21. **for** all $S_i, S_j \in CS$, $i \neq j$ **do**
22. **if** not visited $[S_i][S_k]$ **then**
23. flag $\leftarrow$ False
24. **end if**
25. **end for**
26. until (|CS| = 1) or (flag = True)
27. {Split process starts:}
28. **for** all $S_i \in CS$ where $|S_i| >$ **do**
29. **for all** partitions $\{S_i, S_j\}$ of S_i, where $c = S_j \cup S_k$, $S_j \cap S_k = \emptyset$ do
30. B&B-MIN-COST-ASSIGN (S_j) {Map program T on S_j }
31. B&B-MIN-COST-ASSIGN(S_k)

{Map program T on S_k}
32. **if** {S_j , S_k} ►$_s$ S_i **then**
33. $S_i \leftarrow S_j$ {that is CS = CS \ S_i }
34. CS = CS ⋃ S_k
35. stop ← False
36. Break (one split occurs; no need to check other splits)
37. **end if**
38. **end for**
39. **end for**
40. **Until** stop = True
41. Find k = argmax$_{Si \epsilon CS}$ {v (S_i) / | S_i |
42. Map and execute program T on VO S_k

MSVOF uses a branch-and-bound method to solve the MIN-COST-ASSIGN problem for the formed VOs, and thus, to obtain the mapping of the tasks to GSPs. We will denote by B&B-MIN-COST-ASSIGN (S_i) the function that implements the branch-and-bound method for solving the MINCOST- ASSIGN problem for a VO S_i. Branch-and-bound are methods for solving global optimization problems. They are commonly used for solving integer programs by implicit enumeration in which linear programming relaxations provide the bounds. These methods are based on the observation that a systematic enumeration of integer solutions has a tree structure. The main idea in branch-and-bound is to avoid growing the whole tree as much as possible by permanently discarding nodes, or any of its descendants, which will never be either feasible or optimal. A branch-and-bound method requires two procedures. The first one is a branching procedure that returns two or more smaller sets of the problem. The result of this step is a tree structure whose nodes are the subsets of the problem. The second one is a bounding procedure that computes upper and lower bounds for the objective function within a given subset of the problem by solving the relaxed linear program. Updating bounds for active nodes in a tree enables the method to prune some nonpromising branches. We use the branch-and-bound method but any other mapping algorithms such as those solving variants of the general assignment problem (GAP) can also be used by the VOs to find the minimum cost mapping of the tasks on the GSPs.

The coalition structure CS obtained by the merge process is then subject to splits. In the split process, all coalitions that have more than one member are subject to splitting (lines 27-40). The mechanism tries to split Si that has more than one member into two disjoint coalitions S_j and S_k , where $\mathbf{S_j} \bigcup \mathbf{S_k} = \mathbf{S_i}$,. The for loop in line 29 iterates over all possible partitions in the colexicographical order introduced by Knuth [6]. The problem of finding all partitions of a set is modeled as the problem of partitioning of an integer. Since the split checks the partitions of a set into two subsets, we use the partitioning of an integer into two parts, i.e., two positive integers, where the sum of those is equal to the integer. For example, to partition a set of four GSPs, we partition 15 into two integers. The binary representations of those integers indicate which GSPs are selected in a subset. To increase the speed of the splitting process, we check the subsets with the largest number of GSPs of these partitions first and if they are not feasible there is no reason to check their subsets. This way we reduce the number of partitions that have to be checked and implicitly the execution time of the mechanism. B&B-MINCOST- ASSIGN is called twice to find an optimal allocation on S_j and an optimal allocation on S_k for application T. Since the split is a selfish decision, the splitting occurs even if only one of the members of coalition S_j or S_k can improve its individual value. As a result, the coalition with the higher individual payoff is the decision maker for the split.

VI. Security and Fault Management Mechanism for VOs

The MSVOF mechanism is enhanced to support trust and security for the GSPs. The system provides the security for resource information and data values. Fault tolerant resource scheduling model is integrated with the

VO scheme. The VO construction process upgraded with communicational aspects. The virtual organization is constructed to solve the specific task. Dynamic VO formation mechanism is used in the system. VO communications are carried out with security and cost factors. The system is divided into six major modules.

The Grid Service Provider (GSP) module is designed to manage resource providers. Consumer module is designed to handle the task submission process. The virtual organization is constructed under the VO construction module. Security features are integrated with the VO under the Secured VO construction process. Task execution process is carried out under the operations management process. Fault tolerance process is used to handle resource failures.

6.1. Grid Service Provider

The Grid Service Provider (GSP) provides resources to the grid users. Computational resources are provided to perform task executions. Storage space and data resources are also shared by the GSPs. Resources are provided with cost and deadline factors.

@IJMTER-2015, All rights Reserved

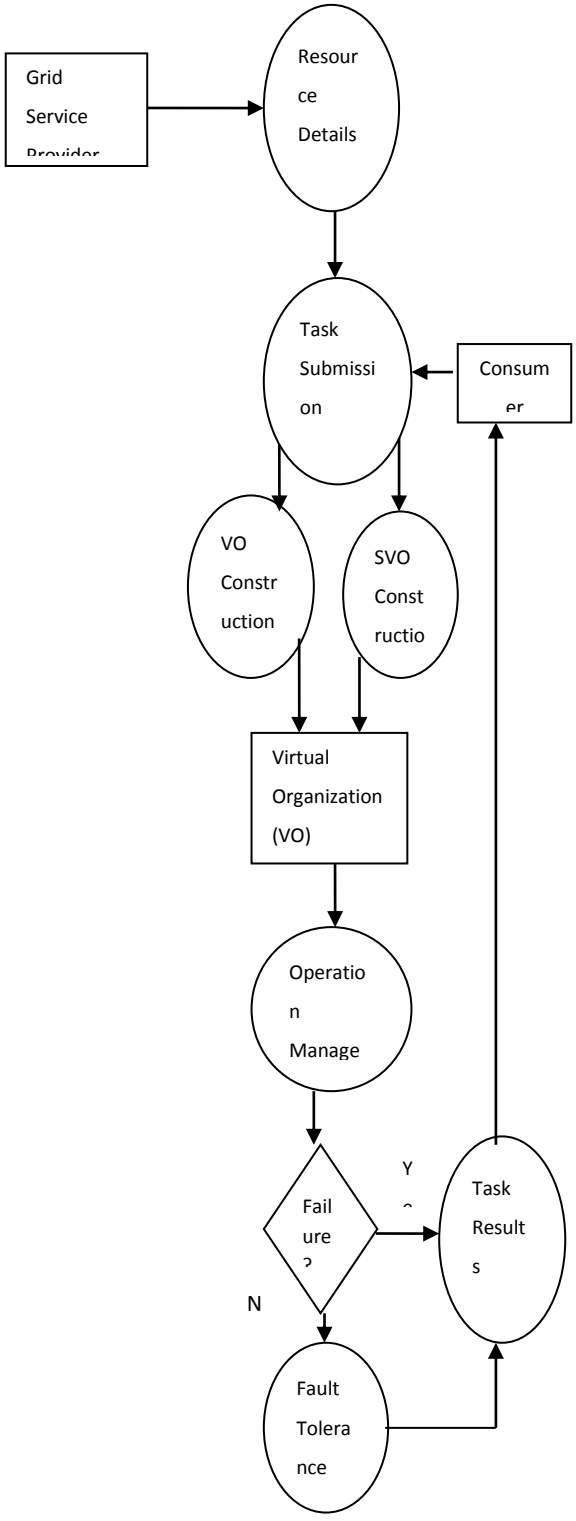

Fig. No: 6.1. Security and Fault Management Mechanism for VOs

6.2. Consumers

The consumers are grid resource users. Task submission operations are initiated by the consumers. Resource capacity, count and duration details are submitted by the consumers. Resource utilization cost payment transactions are managed by the consumers.

6.3. VO Construction

Virtual Organization (VO) is constructed with a set of Grid Service Providers (GSPs). Merge and Split Virtual Organization Formation (MSVOF) mechanism is used in the VO construction process. VO initialization phase identifies the suitable GSPs for the construction process. Rules and duration factors are used in the formation process.

6.4. Secured VO Construction

The Virtual Organization is constructed with security and trust features. Secured Merge and Split Virtual Organization Formation (SMSVOF) scheme is used in the system. Key management process is applied to handle the key distribution between the GSPs in the VO. Resource and data values are secured in the security phase.

6.5. Operations Management

Task execution process is handled in the operations management. Resources are assigned from the GSPs in the VO. Communication cost is estimated with reference to the data transmission process. Virtual Organization is dissolved after the completion of the tasks.

6.6. Fault Tolerance Process

Fault tolerance mechanism is used to manage resource failures. Resources are reallocated at the time of resource failures. Resources are allocated from the GSPs from the same VO. GSPs can be connected to the VO to provide additional resource requirements.

7. Conclusion

Grid computing environment provides resources to the globally distributed systems. Virtual Organization (VO) is a collection of geographically distributed resources linked by electronic communication for coordination. Merge-and-Split VO Formation Mechanism (MSVOF) is used to build virtual organization for a specific task. The MSVOF mechanism is enhanced with trust and security features. The system achieves high revenue for the GSPs. Resource failure management model is integrated with the grid service provider. Grid service providers are improved with efficient security mechanism. Communication overhead is reduced in the message and data transmission process.

REFERENCES

[1] C. Hang, Y. Wang and M. Singh, "Operators for propagating trust and their evaluation in social networks," in *Proc. of the 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 2009.

[2] U. Kuter and J. Golbeck, "Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models," in *Proc. Of the National Conference on Artificial Intelligence*, vol. 22, no. 2, 2007, p. 1377.

[3] D. Agrawal, H. Chivers and J. McDermid, "A proposal for trust management in coalition environments," IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 2008.

[4] K. Avrachenkov, D. Nemirovsky and K. Pham, "A survey on distributed approaches to graph based reputation measures," in *Proc. of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, 2007.

[5] L. Lin and J. Huai, "QGrid: an adaptive trust aware resource management framework," *IEEE Systems Journal*, vol. 3, no. 1, pp. 78–90, 2009.

[6] D. Knuth, The Art of Computer Programming, Vol. 4, Fascicle 3: Generating All Combinations and Partitions. Addison-Wesley Professional, 2005.

[7] Parallel Workloads Archive, http://www.cs.h-uji.ac.il/labs/ parallel/workload/, 2013.

[8] S. Borgatti and M. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466–484, 2006.

[9] Juan Li1, Samee Ullah Khan1, and Nasir Ghani, "Semantics-Based Resource Discovery In Large-Scale Grids", Wiley Publication, 2013

[10] Lena Mashayekhy and Daniel Grosu, "A Merge-and-Split Mechanism for Dynamic Virtual Organization Formation in Grids" IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 3, March 2014