Gossip Protocols Applicability in Grid & Cloud Computing

Praveen Kumar[1] Kumarivibha[2]
[1, 2]M. Tech
[1, 2]Computer Science and Engineering Department (School of Science and Engineering Department)
[1, 2]Galgotias University, Greater Noida, Uttar Pradesh, India

Abstract---"Grid" computing has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and in some cases, high-performance orientation. In this article, we define this new field. First, we review the "Grid problem," which we define as flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources. In Grid computing, resource scheduling & management have always an issue. To achieve high query satisfaction rate, randomness, less overhead, load balancing and many others is priority to increase performance. In this scenario Gossip protocol may provide the solution to the resource scheduling problem. Gossip protocol is of three type Epidemic gossip protocol, pairwise gossip protocol and adaptive dissemination protocol. Epidemic Gossip protocol disseminates information to multiple neighbors while pairwise gossip protocol disseminates information to limited subset. Pairwise protocol works best for uniform load distribution. In Adaptive dissemination configuration have been made by getting results from both epidemic and pairwise gossip protocol. There are three performance criteria according to which this protocol has been tested Query satisfaction rate, packet overhead & requester distance.

I. Introduction

Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly. It is analogous to the power grid. One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale cluster computing based systems. The size of a grid may vary from small a network of computer workstations within a corporation to large collaborations across many companies and networks. In Grid computing, resource scheduling & management have always an issue. To achieve high query satisfaction rate, randomness, less overhead, load balancing and many others is priority to increase performance. In this scenario Gossip protocol may provide the solution to the resource scheduling problem. Gossip protocol is of three type Epidemic gossip protocol, pairwise gossip protocol and adaptive dissemination protocol. [Figure1] Epidemic Gossip protocol disseminates information to multiple neighbors while pairwise gossip protocol disseminates information to limited subset. Pairwise protocol works best for uniform load distribution. In Adaptive dissemination configuration has been made by getting results from both epidemic and pairwise gossip protocol. There are three performance criteria according to which this protocol has been tested Query satisfaction rate, packet overhead & requester distance.

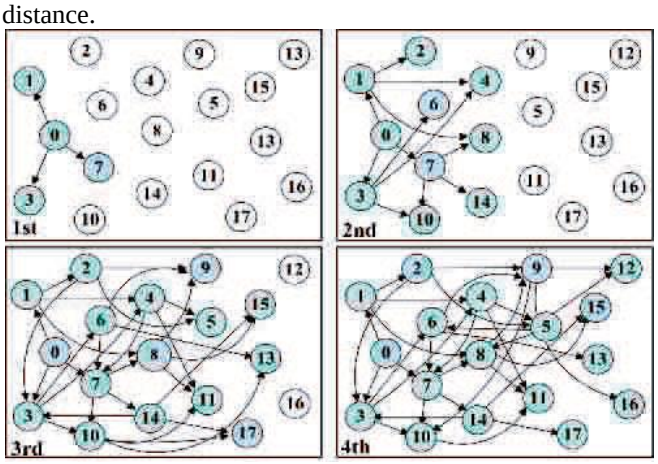

Fig. 1: Process Of communication in gossip protocols

II. Grid Computing and Cloud Computing

Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly. It is analogous to the power grid.

Some of the definitions on grid computing are given below:

Definition: Buyyaet. al. defined grid as 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 users quality-of-service requirements.

Definition: Ian Foster defined grid as a system that coordinates resources which are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service.

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and system software in the data centers that provide those services.

Definition: According to Buyyaet. al. a cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreement. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction . Cloud computing provides basically three kinds of service:

Software as a Service (SaaS)is a kind of services where in many users can make use of the software hosted by the service provider and pay only for time it's being used. It will be better than buying the hardware and software as

keeps off the burden of updating the software to the latest version, licensing and is of course more economical. Some example service providers are Salesforce, Customer Relationships Management (CRM) system and Google Apps.

Platform as a Service (PaaS) provides a high-level integrated environment to design, build, test, deploy and update online custom applications. Some example service providers are Google's App Engine, Microsoft Azure, RightScale and SalesForce .

Infrastructure as a Service (IaaS) refers to the services provided to the users to use processing power, storage, network and other computing resources, to run any software including operating systems and applications. Some of the IaaS providers are AWS, Eucalyptus, Open Stack, GoGrid and Flexiscale.

III. CHALLENGES IN GRID COMPUTING AND CLOUD COMPUTING

(1) Dynamicity: Resources in grid are owned and managed by more than one organization which may enter and leave the grid at any time causing burden on the grid.
(2) Administration: To form a unified resource pool, a heavy system administration burden is raised along with other maintenance work to coordinate local administration policies with global ones.
(3) Debugging and profiling: Parallel and remote debugging has always been a problem for developing HPC programs and is an issue in cloud computing also.
(4) Development: Problems are concerned with ways of writing software to run on grid-computing platforms, which includes decomposing and distributing to processing elements, and then assembling solutions.
(5) Dynamic scalability: The compute nodes are scaled up and down dynamically by the application according to the response time of the user's queries. The scheduling delays involved are real concern which leads to the need of effective and dynamic load management system.
(6) Accounting: Finding ways to support different accounting infrastructure, economic model and application models that can cope well with tasks that communicate frequently and are interdependent.
(7) Querying and access: Scalable provenance querying and secure access of provenance information is open problems for both grid and cloud environment.
(8) Standardization: As every organization has their own APIs and protocols used which makes the user data or vendor lock-in. Thus integration and interoperability of all the services and application is a challenge.
(9) Heterogeneity: Finding ways to create a wide area data intensive programming and scheduling framework in heterogeneous set of resources.
(10) Programming: The low-coupling between nodes and the distributed nature of processing make the programming of applications over grids more complex.
(11) Multi-tenancy: When the number of applications running on the same compute node increases, it will reduce the amount of bandwidth allocated to each application which may lead to performance degradation.
(12) Reliability and fault-tolerance: Tools for testing the application against fault tolerance and compute failures are required which help in developing a reliable system.
(13) Security and Privacy: The user has no idea where data is stored and who will use it as there are more hackers than developers.
(14) Power: Though cloud computing offers many type of services finally to meet the needs of users, enormous amount of power is consumed. An autonomic energy aware resource management is very much required

IV. DIFFERENT GOSSIP PROTOCOL

Epidemic gossip protocol provider disseminates information to multiple neighbors & they forward to their neighbors and so on. Coverage Characteristics are non-uniform, concentrated. Pairwise gossip protocol exchanges information about limited subsets of other resources. It provides uniform coverage & longer dissemination. This protocol doesn't do well in information dissemination for the purpose of scheduling requests onto resources, when resource & request are not uniformly distributed. It doesn't require a structured overlay, it makes it robust. By altering their default behavior, to make it better suited to grid resource scheduling. By Moving protocol away from uniform complete coverage (usual goal) to achieve higher query satisfaction rate. Result shows this protocol isn't well suited to grid resource scheduling with default parameters especially in case of non-uniform distribution. Adaptive Information Dissemination gossip protocol enhanced version Of Epidemic gossip protocol Based on dynamic grid Resource states. Every node has their local score ranging 1 to 100. Score determines how well information disseminates. Each neighborhood sets the neighborhood score as the arithmetic mean of neighborhood node scores.

V. GOSSIP MODELS

Gossip algorithms are distributed algorithms that "mimic the way information spreads when people gossip about some information with each other" [14]. Nodes are autonomous, and do not require any overhead for formation or maintenance of complex routes. They are simple and robust because they are not affected by the network conditions or nodes joining and leaving the network at random, and do not possess a single point of failure. Gossip algorithms have been proposed for many distributed applications, including database replication [4], data aggregation [5], failure detection [6], system monitoring [7],broadcast and multicast [8-10], ad hoc network routing [11], and video streaming [12] among others. These randomized algorithms are locally very simple and have substantially higher degree of fault tolerance, compared to deterministic algorithms, which makes them highly scalable at the same time. Gossip algorithms are based on pair wise random exchange of information where communication takes place in parallel rounds. In each round every node contacts another node at random and either receives or transmits a piece of information (or both). There are three basic variants of gossip algorithm -- Push, Pull, and Push&Pull:

(1) *Push:* in push algorithm each informed node randomly picks another node and sends its message.

(2) *Pull:* in pull algorithm each uninformed node randomly picks anther node and, if the selected node is informed, receives its message.
(3) *Push & pull:* in push & pull algorithm all the nodes pick another node at random and exchange their message if either node is informed.

VI. CONCLUSION

Gossip protocol is an effective and efficient protocol in distributed system environment. Gossip protocol is implementable according to the requirements with particular customization. Protocol expansions are available according to conditions of implementation. Due to their properties, the gossip algorithms can be implemented in the framework of different distributed control strategies. When implementing the gossip algorithms, the nodes in the network have no knowledge of the global network topology. They communicate on local level, with their neighbors only. Using the acquired information, they can execute simple tasks which at the end, will lead to reaching a global objective. While the centralized approach for message dissemination requires the central coordinating unit to have an overview of the network and pull or push information form the nodes, the gossip schemes use push and pull data exchange on local level, thus reducing the overall traffic in the network.

REFERENCES

[1] http://grid.cs.binghamton.edu Gossip-P2P07 Grid Resource Scheduling with Gossiping Protocols De˘ger Cenk Erdil and Michael J. Lewis Department of Computer Science, Binghamton University (SUNY) Binghamton, NY mlewis@cs.binghamton.edu
[2] An Adaptive Algorithm for Information Dissemination in Self-Organizing Grids Deger Cenk Erdil, Michael J. Lewis and Nael B. Abu-Ghazaleh Department of Computer Science State University of NY (SUNY) at Binghamton nael@cs.binghamton.edu
[3] Cluster, Grid and Cloud Computing: A Detailed Comparison the 6th International Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. SuperStar Virgo, Singapore
[4] D. Agrawal, A. El Abbadi, and R. C. Steinke, "Epidemic algorithms in replicated databases (extended abstract)," in Proceedings of the 16th Symposium on Principles of Database Systems (PODS), 1997, pp. 161–172.
[5] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In 44th Annual IEEE Symposium on Foundations of Computer Science, Cambridge, MA, USA, 2003.IEEE Computer Society.
[6] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-style failure detection service. In Proc. of IFIP Int-l Conference on Distributed Systems Platforms and Open Distributed Processing, pages 55-70, 1998.
[7] Van Renesse, R., Birman, K.P., Vogels, W.: Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. ACM Transactions on Computer Systems 21(2) (2003)164–206.
[8] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in Proceedings of the 24th International Conference on computer Communications (INFOCOM), vol. 4, 2005.
[9] L. Ding and P. Manoj. On randomized broadcasting and gossiping in radio networks. In Proceedings of the 8th Annual International Conference on Computing and Combinatorics, pages 340–349, 2002.
[10] P. Eugster, R. Guerraoui, S. Handurukande, A-M Kermarrec, and P. Kouznetsov. Light weight probabilistic broadcast. ACM Trans. On Computer Systems, 21(4), 2003.
[11] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. IEEE/ACM Transactions on Networks (TON), vol. 14, no. 3, pp. 479– 491, 2006.
[12] Frey, D., Guerraoui, R., Kermarrec, A.-M., Monod, M., Qu´ema, V.: Stretching Gossip with Live Streaming. In: Proc. of DSN, 2009.
[13] R. Bakhshi, L. Cloth, W. Fokkink and B. Haverkort. Mean-field analysis for the evaluation of gossip protocols. In Proc. of 6th International Conference on Quantitative Evaluation of Systems (QEST), Pages 247-256, IEEE Computer Society, 2009.
[14] Satish Kumar Verma, Techniques for Improving Predictability and Message Efficiency of Gossip Protocols, Doctor of philosophy, National University of Singapore, 2009.

