European Journal of Computer Science and Information Technology

Vol.2, No.2, pp.17-23, June 2013

Published by European Centre for Research Training and Development UK (www.ea-journals.org)

MONITORING DATA IN DISTRIBUTED COMPUTING SYSTEM

Ummatullah Parveen

IT Skills Department, King Saud University
Riyadh, Saudi Arabia

ABSTRACT: *At present distributed computing is one of the most exploited computing platforms with the emerging techniques like cloud computing. Therefore it is becoming essential to understand all features of the distributed computing; by setting up the hardware to applications of several software's on the distributed systems possibly the most significant factor is "monitoring service". Distributed Computing System (DCS) aims at attaining higher execution speed than the one obtainable with uni-processor system by exploiting the collaboration of multiple computing nodes interconnected in some fashion. Present paper focused on the importance of DCS and its major role in monitoring data, earlier DCS models which has been developed to monitor data along with its advantages and limitations. Finally paper focuses on cost optimization using DCS in general. Monitoring is an important tool for program visualisation, debugging, testing, and development. Thus there is a need to develop the generic monitoring service to support all aspects of management in a distributed system.*

KEYWORDS*: monitoring; distributed computing; Grid computing; publish/subscribe systems*

INTRODUCTION

The progress in computer science have caused many institutions and companies to collect enormous amount of data, from the projects pertaining to Bioinformatics, High Energy Physics, Astronomy, Climate Modelling and Earth Observation and many other [Nabrzyski et al., 2004; Milojicic et al., 2002]. The increasing concern among organizations, their management and even to its end users is the level of compliance of systems with the regulating policies of IT in handling critical data. Monitoring and supervising the compliance of the systems is crucial in IT due to the processes involved (collection, process, and data sharing) are associated in most of the current infrastructures such as finance, banking, entertainment, social networks and communication. There has been increase in legislative laws such as Sarbanes-Oxley Act (SOX) (2002), US Health Insurance Portability and Accountability Act (HIPAA) [1999], and EU Directive 95/46/EC [1995], in order to mandate the compliance. This paper is structured as follows. Section I of this paper discusses the significance of Distributed Computing System (DCS) and its role in monitoring data. Section II would review the previous models of DCS that have been developed to monitor data followed by its advantages and limitations. Section III would focus on cost optimization using DCS in general, and finally the paper would conclude with future recommendations and conclusion.

DISTRIBUTED COMPUTING SYSTEM

Distributed computing refers to computing that include multiple loosely coupled processors working together to solve an overall problem [Keren & Barak, 2003]. Distributed computing provides a natural method to solve complex data and computation intensive problems that arise in power systems analysis and control. Distributed Computing System (DCS) aims at attaining higher execution speed than the one with uniprocessor system by exploiting the collaboration of multiple computing nodes interconnected in a framework. The best possible speed up will be obviously obtained if the several partitions of the given computational task can run independently in parallel. In DCS a single large problem is fragmented into multiple small problems, processes them and combines all the solutions into one solution for the problem. The purpose of the DCS is to coordinate the use of shared resources and also provide communication services to users. In DCS various set of processors handles multiple tasks and execution of all the requested tasks and modules [Khan and Govil, 2013].

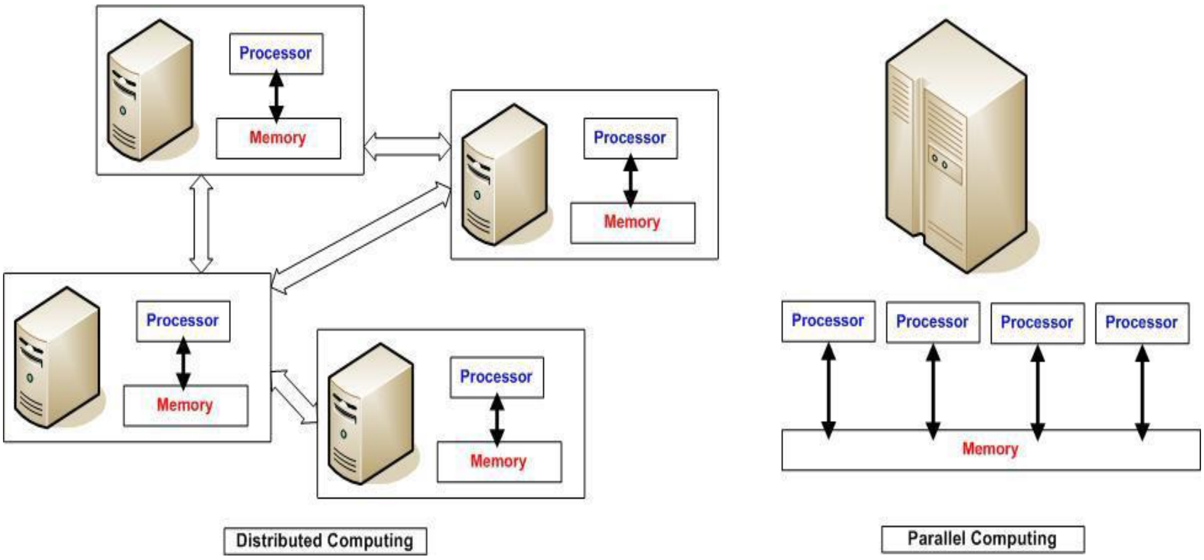

Fig: Difference between Distributed and Parallel Computing

In earlier days, when CPU power and/or memory were limited the main driving force for Distributed System (DS) was to develop the system with larger computing power and as well as significant amount of (shared) memory to tackle much computer-intensive task (mainly scientific and engineering problems). This also led to cluster and supercomputer architecture. With the advent of hardware technologies, not only the CPU computing (Moore's law still driving the increase rate for CPU power), but also other hardware like memory, GPU computing and cluster computing, is attaining pet scale computing with a modest size cluster. Dong et al. [2009] designed taxonomy for the classification of scheduling algorithms in distributed systems. In addition, Dong et al., [2009] has broadly categorized scheduling algorithms as (a) optimal vs. suboptimal (b) static vs. dynamic (c) local vs. global, (d) application centric vs. resource centric, and (e) distributed vs. centralized [Hussain et al., 2013].

Moreover, in the last decade, internet and additional technologies such as mobile computing, has started new concepts which includes real time computing distributed system, geographically distributed computing, cloud computing etc. and this has attracted both academia (to construct large scale scientific computing facilities such as TeraGrid [2010] as well as industry [Wei & Brian Blake, 2010; Microsoft Azure, 2014; Amazon cloud service, 2014]. When compared to centralized monitoring, a real time computer monitoring system is challenging than monitoring sequential computing system due to: (1) Multiple asynchronous processes: Asynchronous parallel process cannot be duplicated and has unpredictable behaviour patterns arising from competitive conditions in real-time distributed computing system (such as a local network), and conceiving the execution sequence of the system's various processes in unattainable. (2) Crucial time constraints: The accuracy of the real-time DCS is dependent on its behaviour pattern in reference with the time period. Inherently in certain domains like those of a chemical plant or weapon system, the real-time DCS are much similar to execution of the real world and the accuracy depends on the output of the processors involved, inclusive of the time constraints imposed by the processes in the real world. However any alterations observed in the monitoring activity is unacceptable in real time distributed environment. (3) Suspension in important communication: The geographically widespread nodes of real-time DCS results in a considerable lag in communication leading to improper synchronization thereby providing incorrect location and timing across the globe. The real-time DCS execution is inhibited by time inaccuracies, unreliable system behaviour and inconclusiveness which possess challenges in monitoring the execution making it nearly impossible for states across the globe. The two elements, global state and timing which are representative of the system behaviour during execution cannot be monitored. In addition, the entire current monitoring systems are formulated to share the resources in computing system [Garcia-Molina et al., 1984; Haban, 1987; Joyce et al., 1987; LeBlanc & Robbins, 1985; Plattner & Nievergelt, 1981; Plattner, 1984; Snodgrass, 1988; Tokuda et al., 1989]. The following section would analyse some of the models and approaches used for monitoring and cost optimization models in DCS.

III. RELATED WORKS

Over the past decades, many approaches to distributed computing systems have been developed. These include Socket Programming, Remote Procedure Calls (RPC), DCOM, object-oriented DCE, CORBA, Java RMI, and Message-Oriented Middleware (MOM) (Van Steen & Tanenbaum, 2002) and in addition a variety of monitoring and discovery services exists, ZenOSS, INCA, VMware vCloud, MonALISA, XCat.

So far, there have been only few published results which have been studied quantitatively for the performance and monitoring of present distributed systems. The potential and inhibitions of Globus Toolkit's Monitoring and Discovery Service was studied by Aloisio et al. [n.d] limiting their observations by testing only on GIIS, while Laszewski, Dinda and Plale (2002) explored more on the positive and negative aspects of structuring a Grid Information Service on relational (flat table) and hierarchical representations . The comparison in these representations mainly focused on the elements in the concepts and services. ; A synthetic workload was standardized by Plale [2003] in his study, which was formulated utilizing a predetermined fundamental operators such as updates, queries etc., against the theoretical data obtained from two dissimilar databases (MySQL, and Xindice) and thereafter evaluate the outcome. However, there are no information services which employ either of the two approaches. Ganglia is a scalable distributed monitoring system for high performance computing systems such as grids and clusters [Garcia-Molina et al., 1984; Haban, 1987; Joyce et al., 1987; LeBlanc & Robbins, 1985; Plattner & Nievergelt, 1981; Plattner, 1984; Snodgrass, 1988; Tokuda et al., 1989] is. The experiences obtained through real world deployments on systems of varying configurations, scales, and target application domains with more than five hundred clusters across the globe were applied to develop, design, implement and evaluate Ganglia by Massie, Chun and Culler (2004). Similarly, other studies conducted on the monitoring of cultures are Supermom [Sottile & Minnich, 2002] which is a hierarchical cluster monitoring system that employs a statically constructed hierarchy of intricate connections for the collection and compiling of cluster data obtained through client kernel modules executed on individual nodes.. Yet another statistically constructed hierarchy which is of the relational database to obtain index and gather cluster data was developed [Anderson & Patterson, 1997] as CARD. Another server/client cluster monitoring system is the PARAMON which employs the server to send a determined amount of node database, and the clients obtain the information from the servers and interpret the information extracted [Buyya, 2000]. The popular client server (http://www.b4.com) used in commercial aspects is the Big Brother for data distribution in heterogeneous systems and also to monitor them.

Wang et al. [2010] proposed a unified monitoring framework for distributed environment with heterogonous monitoring system using a case study based on Enterprise Service Bus (ESB). The framework minimised the costs of software development, maintenance and increased the flexibility, availability, scalability and robustness. Schovancova [2012] reviewed several distributed computing system using ATLAS grid resources (CPU resource, storage systems, network lines). The monitoring tools converged in terms of data communication and data visualization provided by a backend in the lightweight data-interchange format of Javascript Object Notation (JSON) which is based on the JavaScript library jQuery and plug ins. Visualization frameworks are the xbrowse framework, and the hBrowse framework, whereas the framework for plotting libraries are Highcharts (historical views dashboard), flot (Panda monitor), graphtools, and google charts (PD2P monitoring). Renesse, Birman & Vogels [n.d] had created ASTROLABLE to solve wide variety of scalable problems. The protocol used a peer-to-peer protocol based on the SQL Query language for aggregation.

Zhang, Freschl, & Schopf [n.d] illustrated the scalability and performance of three monitoring and information services: (1) the Globus Monitoring and Discovery Service (MDS) (Czajkowski, 2001); (2) Monitoring & Discovery System (MDS) (n.d), the Relational-Grid Monitoring Architecture (R-GMA) DataGrid [2002] used in the European DataGrid [2003], Hawkeye [2003], and (3) part of the Condor Project [Litzkow, 1988]. Each of these systems was used in production or near-production grid test beds. The findings showed that all the three services offered good scalability and established higher benefits for caching or pre-fetching the data. In addition the systems were also required to have primary components at well-connected sites because of high load observed in every system. Additionally, in order to monitor the data, several algorithms have been proposed. For instance, the study proposed a solution based on the intractability of monitoring an arbitrary linear-time temporal logic formula on partially ordered logs and also to monitor the usage of data in concurrent distributed systems using a case study. Authors in this study illustrated the feasibility and advantages of monitoring the usage of sensitive data.

Moreover few studies have examined the pricing issues in distributive cloud computing system. Resource-consumption based pricing is particularly sensitive to the manner in which the system is designed, optimized, configured, monitored, and measured. For instance, in the study conducted by Wang et al. [2010], it was suggested that decoupling users from cloud providers with a pricing scheme would act as a bridge in such scenarios. Cloud computing has basically changed the landscape of system design and optimization using Amazon EC2 cloud service. The studies by Napper & Bientinesi, (2009) and Walker, (2008) compared Amazon EC2 with a private cloud for high performance computing. The availability, cost, and performance of Amazon services were studied with simple operations [Garfinkel, 2007; Palankar, 2008]. Enhanced Mosix (E-Mosix) a tailored version of Mosix project (13), gluster, Tycoon (20), faucets (116), cluster on demand [Hussain, 2013] (22), Libra (26), open SSI (25), REXEC (30), PVM (28), GNQS (31) have been used to test the cost attributes with the help of either single, multiple or parallel task.

CONCLUSION

Monitoring is an important tool for program visualisation, debugging, testing, and development. It is also very important for day to day management of large systems. There is a necessity to establish an underlying general monitoring service to carry out all aspects of management in a distributed system. Amidst many researches which has been conducted and various tools are developed for monitoring the systems, still they are not practical for monitoring real-time distributed computing systems because of monitor's invasive nature. The hardware and OS support could be cost prohibitive to end-users. Though, programming level is a big burden to end-users. Grid and cloud computing appears promising among the three HPC categories and a lot of research has been performed in each category. The focus of future distributed computing systems is to minimise the operational cost of data centres and enhance the resilience in adaptability, failure, and graceful recovery.

REFERENCES

[1] J. Nabrzyski, J. Schopf, J. Wˆeglarz, "*Grid Resource Management: State of the Art and Future Trends*", Kluwer Academic Publishers, 2004.
[2] D. Milojicic, V. Kalogeraki, R. Lukose, Nagaraja, K., et al, Peer to peer computing, HP laboratories Palo Alto, *Technical Report HPL-2002-57*, 2002.
[3] The Health Insurance Portability and Accountability Act (HIPAA), "104th Congress'', *Public Law*, pp. 104-191, 1996.
[4] Sarbanes-Oxley Act, "107th Congress", *Public Law*, pp. 107-204, 2002.
[5] Directive 95/46/EC of the European Parliament and of the Council, "On the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data", *Official Journal L 281*, 1995.
[6] A. Keren, A. Barak, "opportunity cost algorithms for reduction of I/O and inter process communication overhead in a computing cluster", *IEEE transaction on parallel and distributed systems*, Vol. 14, No. 1), pp. 39-50, 2003.
[7] N.F. Khan, K. Govil, "Cost Optimization Technique of Task Allocation in Heterogeneous Distributed Computing System", *Int. J. Advanced Networking and Applications*, Vol. 5, No. 3, pp. 1913-1916, 2013.
[8] B. Dong, Q. Zheng, J. Yang, H. Li, M. Qiao, M, An e-learning ecosystem based on cloud computing infrastructure, In: *9th IEEE International Conference on Advanced Learning Technologies*, pp. 125–127, 2009.

[9] H. Hussain, S.R. Malik, R. Hameedb, S.U. Khanb, G. Bickler et al. "A survey on resource allocation in high performance distributed computing systems", *Parallel Computing*, Vol. 39, pp. 709–736, 2013.
[10] TeraGrid, TeraGrid Archives, *https://www.xsede.org/tg-archives, 2010.*
[11] Y. Wei, M. Brian Blake, "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities", IEEE *Internet Computing*, Vol. 14, No. 6, pp. 72-75, 2010. .
[12] Microsoft Azure, The cloud for modern business, *http://azure.microsoft.com/en-us/, 2014.*
[13] Amazon cloud service. (2014) *Amazon cloud service*, Available at: http://aws.amazon.com/ec2/
[14] H. Garcia-Molina, F. Germano, W. H. Kohler, "Debugging a Distributed Computing System", *IEEE Trans. Software Engineering*, Vol. 2, pp. 210-219, 1984.
[15] Haban, D. DTM - A Method for Testing Distributed Systems, *Proc. 6th Symp. Re- liability in Distributed Software and Data- base Systems*, CS Press, Los Alamitos, Calif, Order No. 737, pp. 66-73, 1987.
[16] J. Joyce, G. Lomow, K. Slind, B. Unger, "Monitoring Distributed Systems", *ACM Trans. Computer Systems*, Vol. 5, No. 2, pp. 121-150, 1987.
[17] R.J. LeBlanc, A.D. Robbins, Event- Driven Monitoring of Distributed Programs, *Proc. 5th Int'l Conf Distributed Computer Systems*, pp. 515-522, 1985.
[18] B. Plattner, J. Nievergelt, "Monitoring Program Execution: A Survey", *IEEE Trans, on Computers*, pp. 76-93, 1981.
[19] B. Plattner, "Real-Time Execution Monitoring", *IEEE Trans. Software Engi- neering*, Vol. 6, pp. 756-764, 1984.
[20]R. Snodgrass, "A Relational Approach to Monitoring Complex Systems", *ACM Trans. Computer Systems*, Vol. 6, No. 2, pp. 157-196, 1988.
[21] H. Tokuda, M. Kotera, C. W. Mercer, "A Real-Time Monitor for a Distributed Real- Time Operating System", *ACM SlCPlan Notices*, Vol. 24, No.1, pp. 68- 77, 1989.
[22] M. Van Steen, A. Tanenbaum, Distributed *Systems: Principles and Paradigms. Englewood Cliffs*, NJ: Prentice Hall, 2002.
[23] G. Aloisio, M. Cafaro, I. Epicoco, S. Fiore, S. "Analysis of the Globus Toolkit Grid Information Service",*http://www.gridlab.org/Resources/Deliverables/D10.1.pdf*, n.d.
[24] B. Plale, P. Dinda, G. Laszewski, Key Concepts and Services of a Grid Information Service, *ISCA 15th International Parallel and Distributed Computing Systems* (PDCS), 2002.
[25] B. Plale, Whitepaper on Synthetic Workload for Grid Information Services/Registries, Data *Workshop* 2003 held in conjunction with Globus World 2003, San Diego, 2003.
[26] M.L. Massie, B.N. Chun, D.E. "Culler, "The ganglia distributed monitoring system: Design, implementation, and experience", *Parallel Computing*, vol. 30, no. 5-6, pp. 817-840, 2004.
[27] M. Sottile, R. Minnich, Supermon: a high-speed cluster monitoring system, In: *Proceedings of Cluster*, 2002.
[28] E. Anderson, D. Patterson, Extensible, scalable monitoring for clusters of computers, In: *Proceedings of the 11th Systems Administration Conference*, 1997.
[29] R. Buyya, 'Parmon: a portable and scalable monitoring system for clusters', *Software—Practice and Experience*, Vol. 30, No. 7, pp. 723–739, 2000.
[30]X. Wang, H. Wang, Y. Wang, 'A Unified Monitoring Framework for Distributed Environment ', *Intelligent Information Management*, Vol. 2, pp. 398-405, 2010.
[31] J. Schovancova, "ATLAS Distributed Computing Monitoring tools after full 2 years of LHC data taking", *Journal of Physics: Conference Series*, pp. 396, 2012.

[32] R.V. Renesse, K.P. Birman, W. Vogels, W. *Astrolabe: A Robust and Scalable Technology For Distributed System Monitoring, Management, and Data Mining*, Department of Computer Science, Cornell University, Ithaca, n.d.
[33] A. Zhang, H. Freschl, P. Schopf, “A Performance Study of Monitoring and Information Services for Distributed Systems”, *http://arxiv.org/ftp/cs/papers/0304/0304015.pdf*, n.d.
[34] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, Grid Information Services for Distributed Resource Sharing, In: *Proc. 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, 2001.
[35] Monitoring & Discovery System (MDS), “GT Information Services: Monitoring & Discovery System (MDS)”, *http://toolkit.globus.org/toolkit/mds/*, 2001.
[36] DataGrid, DataGrid Information and Monitoring Services Architecture: Design, Requirements and Evaluation Criteria, Technical *Report*, 2002.
[37] DataGrid, “DataGrid Tutorial”, *http://eu-datagrid.web.cern.ch/eu-datagrid/Tutorial/tutorial.htm*, 2013.
[38] Hawkeye, “Hawkeye”, *http://www.cs.wisc.edu/condor/hawkeye*, 2013.
[39] M. Litzkow, M. Livny, M. Mutka, M. Condor - A Hunter of Idle Workstations, In: *Proceedings of the 8th International Conference of Distributed Computing Systems*, pp. 104-111, 1988.
[40] J. Napper, P. Bientinesi, Can cloud computing reach the top500? In Un Conventional high performance computing *workshop*, 2009.
[41] E. Walker, “Benchmarking Amazon EC2 for high-performance scientific computing”, *The USENIX Magazine*, Vol. 33, No. 5, 2008.
[42] S.L. Garfinkel, An evaluation of Amazon’s grid computing services: EC2, S3 and SQS. Technical *Report* TR-08-07, Harvard Univ, 2007.
[43] M.R. Palankar, A. Iamnitchi, M. Ripeanu, S. Garfinkel, “Amazon S3 for science grids: a viable solution?” In *DADC*, 2008.