

Secure Data Dynamics in Tandem with Dynamic Resource Allocation

Bindu Madhavi^{#1}, G.Kalpana^{*2} Dr.R.V.Krishnaiah^{*3}

^{#1}*M.Tech, CSE, DRKIST, Hyderabad, Andhra Pradesh, India*

[#]*Associate Professor, Department of CSE, DRKIST, Hyderabad, Andhra Pradesh, India*

[#]*Principal, Department of CSE, DRKIST, Hyderabad, Andhra Pradesh, India*

Abstract--In Cloud computing both security with perfect data dynamics and optimal resource allocation are essential. For best realization of cloud computing parallel and reliable data processing is required. There are many providers of cloud services such as Oracle, Microsoft, IBM, and Google. The existing systems used for cloud computing are homogenous in nature. The resource allocation and execution of jobs parallelly has some limitations. The security is also concerns as the cloud servers are treated as un-trusted by the cloud users. In this paper parallel processing, dynamic resource allocation challenges are addressed. We built a prototype application to demonstrate the proof of concept and the empirical results are encouraging.

Index Terms –Parallel processing, cloud computing, Map Reduce, many-task computing.

I. INTRODUCTION

Many organizations in the real world are into processing of large volumes of data. This has to be done in a cost effective fashion. Such organizations include Microsoft, Yahoo, and Google etc. They deal with increased volumes of data every day. For this purpose storing and retrieving data using conventional databases is very expensive [1]. To overcome the problem, many companies have started using commodity servers in a big way. When number of such servers is being used for processing huge amount of data, the processing work is divided into multiple tasks and assigned dynamically to every server involved. This facilitates the system to work faster as the available nodes share the job of processing such voluminous data in a short span of time. This is possible due to parallel processing of data. To enhance the performance further these companies have developed customized frameworks that take care of parallel data processing in an efficient fashion. There are many such frameworks existing in the real world. They are created by industry giants such as Microsoft, Yahoo, and Google etc. For instance the framework developed by Google is known as MapReduce [2]. Microsoft

developed a framework known as Dryad [3] while Yahoo developed a framework known as Map-Reduce-Merge [4]. These products are varying capabilities for achieving many task computing or high throughput computing in terms of amount of data to be processed and also the number of tasks involved in the processing [5]. These applications are having different architectural designs. However, they share a common approach in fulfilling objectives such as fault tolerance, parallel programming, optimized executions, and hiding the unnecessary things. These products are basically meant for parallel processing. Developers can write the programs sequentially. When they are given to these products, they are distributed among multiple nodes and the work is done parallel.

It is not recommended to establish costly data centers for parallel processing of data when a company involves in processing large volumes of data occasionally. For such organizations, the recommended solution is the usage of cloud computing. Cloud computing has emerged as a technology that enables individuals and organizations to gain access to state-of-the-art servers, data centers, network infrastructure without the need for capital investment in pay per use fashion. Many companies are providing cloud services. They include IBM, Oracle, Microsoft, Google etc. These companies are cloud service providers who provide different kind of cloud services. The services are categorized into IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a service). This paper deals with opportunities and challenges of parallel data processing in IaaS clouds. The cloud products available are EC2 from Amazon [6], Azure from Microsoft etc. The cloud computing technology is based on the concept of virtualization. The virtualisation technology makes the cloud computing a reality as it can reduce the cost of maintaining clouds substantially. The virtualization technology involves creation and destroying of VMs of different

types based on the need. The data processing frameworks such as Map Reduce framework from Google of late depend on the virtualization being used in IaaS cloud [7] [8]. Recently Amazon also started using the open source product Hadoop for its IaaS cloud [9]. However, the current data processing frameworks are still using static approach for resource allocation. This causes problems in clouds as the resources may be exhausted when huge number of parallel jobs is being executed. This is the problem to be explored and resolved as it reduces the overall performance of the cloud for parallel data processing.

This paper focuses on discussing various challenges involved in parallel data processing and the possibilities as well with respect to the framework proposed in this paper. This is the framework known for its dynamic resource allocation capabilities for the first time in the world. The work in this paper is influenced by [10].

II. RELATED WORK

Recently there was lot of research went on parallel data processing and its implications and possibilities. Many systems came into existence for processing MTC applications where parallel processing of data is essential. All such systems have a common goal such as fault tolerance and parallel execution of tasks and they are being used in different fields. Open source version of MapReduce i.e. Hadoop [7] or the MapReduce were designed to run jobs in parallel in cost effective manner using commodity servers. For simplicity an example framework is MapReduce. Once job is given to it, it automatically takes care of dividing the given job into tasks and spreading them across the available servers. There are two programs involved namely Map and Reduce for specific functionality. There are many other programs that coordinate with the jobs of MapReduce nature [11], [12]. As a matter of fact, the MapReduce framework was developed for huge clusters of static in nature. It can deal with ad hoc failure of nodes, allocation of resources to machines of homogenous kind. It lacks efficient dynamic resource allocation capabilities.

In [13] a framework by name Pegasus framework was designed for grid based systems to work on complex MTC works of scientific in nature. The working of this framework is somewhat similar to the proposed one. The workflows created for Pegasus are abstract in nature until a mapping is established between the resources available at runtime and the tasks to be processed in parallel. The proposed framework focuses on reducing the cost while

allocating VM instances to process parallel jobs using its stage concept. It is not the case with Pegasus as it deals with abnormal things at runtime. The execution engine of Pegasus is DAGMan and Condor-G [14]. Another framework by name Swift [15] was introduced in [15] to deal with issues pertaining to management that are result of running multiple tasks in parallel. The components used by the authors are Globus [16], Falkon [17] and CoG Karajan [18]. The approach followed in this paper resembles Dryad [3] which also runs jobs pertaining to DAG. The product Dryad assumes a set of worker nodes which are homogenous in nature. The notion of cost of processing is not implemented by this. In [19] a new approach is presented to handle peak-load issues in cloud computing products such as EC2 in terms of allocating resources on-demand. In [20] also discussion is made on the resource allocations in grid environment for workflows related to scientific work. Both were focusing on data-intensive and batch-driven workflows which are similar to that of proposed framework. Recently in [21] an operating system was presented which can allocate VMs on demand. It is meant for clouds and multicore.

III. OPPURTUNITIES AND CHALLENGES

The existing data processing frameworks like MapReduce and Dryad were developed for working in cluster environments. They can't be directly used in cloud environments. Moreover they are built on certain assumptions. This section discusses the opportunities and challenges when those assumptions are got rid of. The assumptions are that the worker nodes are static in nature and homogenous. In cloud environment, the resources are to be allocated to worker nodes on demand. The new possibilities with cloud computing include scheduling data processing jobs, allocating resources dynamically, optimum utilization of VM instances with proper management. It is also essential to determine which task has to be executed in which VM instance. Keeping these in mind the proposed framework is designed which is as shown in fig. 1. The challenges in parallel processing include:

- ✓ When compared with static cluster environment, the cloud environment has challenges such as its opaqueness with respect to data locality.
- ✓ Large amounts of data transfer due to the non exposure of customer information.
- ✓ Network topologies which were subjected to research are not clear whether they can be applied to cloud environment.

IV. ARCHITECTURAL DESIGN OF PROPOSED FRAMEWORK

This section described the architectural design of a new framework that takes care of parallel processing of jobs with efficient resource allocation. It takes ideas from existing frameworks and improves them further.

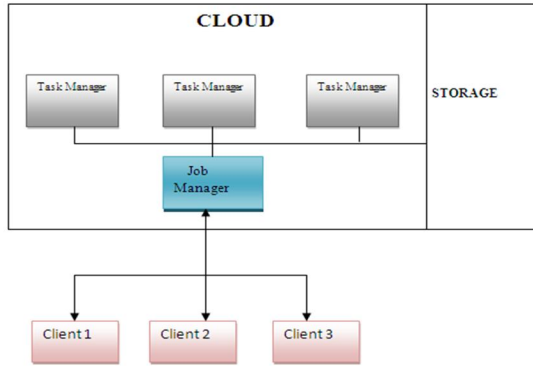


Fig. 1 –Proposed framework that runs in IaaS cloud

As can be seen in fig. 1, the proposed framework runs in IaaS cloud for parallel data processing. The users of this can send jobs by running a VM (Virtual Machine) which is also known as job manager. Then the job manager with the help of the controller provided by cloud is responsible to create VMs as and when required. It takes job from the clients and distributes them to different task managers. The task managers are nothing but the worker nodes that actually process the tasks. It does mean that the job manager divides the given job into number of tasks and assigns them to task managers at runtime. It is also having capabilities to create and terminate virtual machines as required. The IaaS cloud has common storage which can be shared by all task managers.

Job Scheduling and Execution

When job manager receive job from client, it gets transformed into execution graph which is a data structure which has adequate information related to scheduling and execution of jobs. The proposed framework allows splitting the job into tasks and subtasks and also takes care of creating and terminating VM instances besides managing them for efficiency. In order to overcome the problem of availability of VMs, the framework allows the

dividing the execution graph into many execution stages. VM instances are associated with stages so that it is easy to manage those VM instances with stages. These stages act as checkpoints that ensure the complete execution of tasks without causing problems.

V. EXPERIMENTS AND RESULTS

Three experiments are conducted on local IaaS cloud of commodity servers. Each server has two CPUs of 2.66 GHz speed. Main memory is of 32 GB. For virtual machine deployment, Eucalyptus [22] is used. For virtual IO access virtio [23] is used. The first experiment is done with MapReduce and Hadoop. The second experiment is done using MapReduce and Proposed framework while the third experiment is done using DAG and Proposed Framework. As part of first experiment, three programs are written for aggregate task. The first job is responsible to reads the given data and sorts it in ascending order and sends them to Hadoop file system. The second and third jobs operate on the sorted data for data aggregation. In the second experiment also the three maps reduce programs are reused by writing wrapper classes. This facilitated to run unmodified Hadoop MapReduce programs with proposed framework. This experiment is meant for observing the efficient dynamic resource allocation. The third experiment we deviated from MapReduce processing pattern. Here the sort/aggregate problem is implemented as DAG to test the ability of the proposed framework for managing worker nodes of heterogeneous in nature. It does mean that this experiment is made to see how the resources are dynamically allocated efficiently for heterogeneous compute nodes. The results of first experiment are shown in fig. 2.

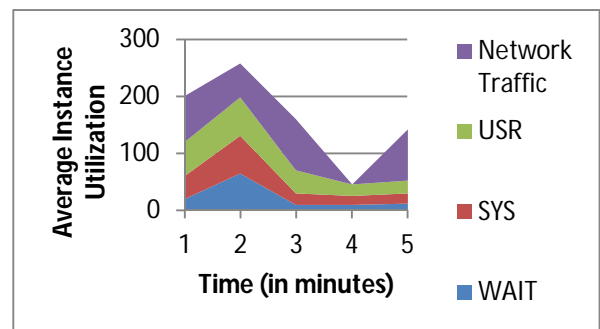


Fig. 2 – Results of first experiment

As can be seen in fig. 2, the performance results of the first experiment are presented. It shows average network utilization and also the average network

traffic among instances. Comparatively fair resource utilization is reflected in the graph.

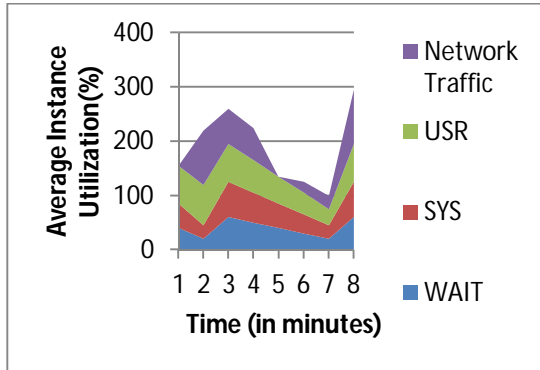


Fig. 3 –Results of experiment 2

As can be seen in fig. 3, the performance results of the second experiment are presented. It shows average network utilization and also the average network traffic among instances for executing the same MapReduce programs on top of the proposed framework.

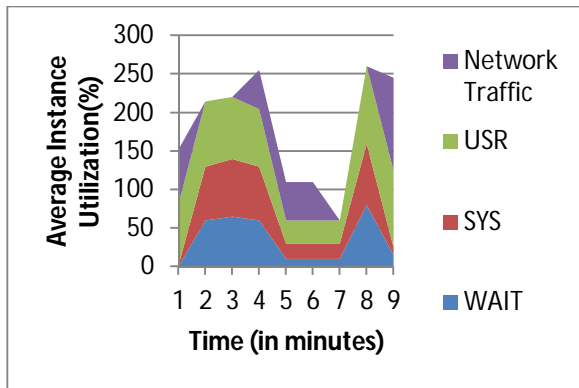


Fig. 4 – Results of experiment 3

As can be seen in fig. 4, the performance results of the second experiment are presented. It shows average network utilization and also the average network traffic among instances for executing DAG on top of the proposed framework.

VI. CONCLUSION

This paper has presented the possible opportunities and challenges with respect to parallel data processing in IaaS clouds using the environment proposed in this paper. The main focus was on the dynamic resource allocation for parallel data processing. We presented the architecture of a new framework and even compared with the Hadoop which is a well known data processing framework.

The evaluation provides insights into various aspects such as assigning VMs to tasks, how parallel jobs are processed. It also discusses about the dynamic resource allocation to reduce processing cost and improve optimal utilization of resources.

REFERENCES

- [1] R. Chaiken, B. Jenkins, P.-A. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets. Proc. VLDB Endow., 1(2):1265–1276, 2008.
- [2] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation, pages
- [3] M. Isard, M. Budy, Y. Yu, A. Birrell, and D. Fetterly. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks. In EuroSys '07: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, pages 59–72, New York, NY, USA, 2007. ACM.
- [4] H. chih Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker. Map-Reduce-Merge: Simplified Relational Data Processing on Large Clusters. In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pages 1029–1040, New York, NY, USA, 2007. ACM.
- [5] I. Raicu, I. Foster, and Y. Zhao. Many-Task Computing for Grids and Supercomputers. In Many-Task Computing on Grids and Supercomputers, 2008. MTAGS 2008. Workshop on, pages 1–11, Nov. 2008.
- [6] Amazon Web Services LLC. Amazon Elastic Compute Cloud (Amazon EC2). <http://aws.amazon.com/ec2/>, 2009.
- [7] The Apache Software Foundation. Welcome to Hadoop! <http://hadoop.apache.org/>, 2009.
- [8] T. White. Hadoop: The Definitive Guide. O'Reilly Media, 2009.
- [9] Amazon Web Services LLC. Amazon Elastic MapReduce. <http://aws.amazon.com/elasticmapreduce/>, 2009.
- [10] D. Warneke and O. Kao. Nephelē: Efficient Parallel Data Processing in the Cloud. In MTAGS '09: Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers, pages 1–10, New York, NY, USA, 2009. ACM.
- [11] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the Data: Parallel Analysis with Sawzall. Sci. Program., 13(4):277–298, 2005.
- [12] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig Latin: A Not-So-Foreign Language for Data Processing. In SIGMOD '08: Proceedings of the 2008 ACM

SIGMOD international conference on Management of data, pages 1099–1110, New York, NY, USA, 2008. ACM.

[13] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems. *Sci. Program.*, 13(3):219–237, 2005.

[14] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke. Condor-G: A Computation Management Agent for Multi-Institutional Grids. *Cluster Computing*, 5(3):237–246, 2002.

[15] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, V. Nefedova, I. Raicu, T. Stef-Praun, and M. Wilde. Swift: Fast, Reliable, Loosely Coupled Parallel Computation. In *Services, 2007 IEEE Congress on*, pages 199–206, July 2007.

[16] I. Foster and C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *Intl. Journal of Supercomputer Applications*, 11(2):115–128, 1997.

[17] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, and M. Wilde. Falcon: a Fast and Light-weight task execution framework. In *SC '07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, pages 1–12, New York, NY, USA, 2007. ACM.

[18] G. von Laszewski, M. Hategan, and D. Kodeboyina. *Workflows for e-Science Scientific Workflows for Grids*. Springer, 2007.

[19] T. Dornemann, E. Juhnke, and B. Freisleben. On-Demand Resource Provisioning for BPEL Workflows Using Amazon's Elastic Compute Cloud. In *CCGRID '09: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 140–147, Washington, DC, USA, 2009. IEEE Computer Society.

[20] L. Ramakrishnan, C. Koelbel, Y.-S. Kee, R. Wolski, D. Nurmi, D. Gannon, G. Obertelli, A. Yar Khan, A. Mandal, T. M. Huang, K. Thyagaraja, and D. Zagorodnov. VGrADS: Enabling e-Science Workflows on Grids and Clouds with Fault Tolerance. In *SC'09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–12, New York, NY, USA, 2009. ACM.

[21] D. Wentzlaff, C. G. III, N. Beckmann, K. Modzelewski, A. Belay, L. Youseff, J. Miller, and A. Agarwal. An Operating System for Multicore and Clouds: Mechanisms and Implementation. In *SoCC'10: Proceedings of the ACM Symposium on Cloud Computing 2010*, pages 3–14, New York, NY, USA, 2010. ACM.

[22] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems. Technical report, University of California, Santa Barbara, 2008.

[23] R. Russell. virtio: Towards a De-Facto Standard for Virtual I/O Devices. *SIGOPS Oper. Syst. Rev.*, 42(5):95–103, 2008.

AUTHORS



Bindu Madhavi is student of DRK Institute of Science and Technology, Hyderabad, AP, INDIA. She has received B.Tech in Biotechnology, M.Tech Degree in computer science and engineering. Her main research interest includes data mining and Cloud computing.



Kalpana Gudikandula is working as Associate Professor at DRK Institute of Science & Technology, Ranga Reddy, and Andhra Pradesh, India. She has received M.Tech Degree in Computer Science. His Main Interest includes Cloud Computing, Software Engineering.



Dr. R.V. Krishnaiah (Ph.D) is working as Principal at DRK INSTITUTE OF SCIENCE & TECHNOLOGY, Hyderabad, AP, INDIA. He has received M.Tech Degree EIE and CSE. His main research interest includes Data Mining, Software Engineering.