International Journal of Grid and Distributed Computing
Vol. 9, No. 3 (2016), pp.33-40
http://dx.doi.org/10.14257/ijgdc.2016.9.3.05

Efficient and Parallel Data Processing and Resource Allocation in the Cloud by using Nephele's Data Processing Framework

V.Saranya[1*], S.Ramya[2], R.G. Suresh Kumar[3] and T.Nalini[4]

[1,2] Master of Technology, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering & Technology, Pondicherry, India.
[3]Research Scholar, Vels University, Chennai, India.
[4]Professor, Department of Computer Science and Engineering, Bharath University, Chennai, India.
[1]saranya.vadivelu28@gmail.com, [2]ramyasivakavi.s1@gmail.com, [3]aargeek@gmail.com, [4]drnalinichidambaram@gmail.com

Abstract

Cloud computing is a technology in which the Cloud Service Providers (CSP) provide many virtual servers to the users to store their information in the cloud. The faults occurring on the assignment and dismission of the virtual machines, the processing cost in the allocation of resources must also be considered. The parallel processing of the information on the virtual machines must be done effectively and in an efficient manner. A variety of systems were developed to facilitate Many Task Computing (MTC). These systems aim to hide the issues of parallelism and fault tolerant and they are used in many applications. In this paper, we introduced Nephele, a data processing framework to exploit dynamic resource provisioning offered by IaaS clouds. The performance evaluation of the virtual machines has been evaluated and the allocation and de-allocation of job tasks to the specific virtual machines has also been considered. A performance comparison with the well known data processing framework hadoop has been done. Thus this paper tells about the effective and efficient manner of processing the data by parallel processing and allocating the correct resources for the desired task. It also helps to reduce the cost of resource utilization by exploiting the dynamic resource utilization.

Keywords*: CSP, Allocation, De-allocation, Many Task Computing, Nephele, Dynamic resource utilization, Hadoop, Parallel processing*

1. Introduction

Cloud computing is a computing environment that provides services over the internet. The cloud allows individuals and businesses to share and use hardware and software managed by third party users. The cloud services include online file storage, webmail, online business applications, social networking sites. The cloud computing provides its users an opportunity to access the resources available in the cloud from anywhere if there is a presence of network connection. It provides a shared pool of resources like data storage space, computer processing power, specialized corporate and user applications.

Cloud computing provides services to their users on subscription-based or pay-per-use. It provides its users not only to access the documents but also to edit and modify the documents stored the cloud. The elasticity characteristic of cloud provides its users a wide range of applications and services. These services and applications are spread throughout

V.Saranya is the corresponding author.

the enterprise and it is extended from single corporate to large enterprise. The applications and services offered by cloud are available to variety of users and enterprises.

Cloud provides the services over the internet by hiding the hardware it uses from the users. The cloud users connect to the cloud from their personal computer or any accessible device. The users are unaware of where their data are stored and how their data are stored. The data may be stored on various number of remote servers and data may be splitted into various parts before storage. This is usually done for security purpose. The cloud users are not known about this method followed by the cloud. Microsoft office communication online, Windows azure, Google app engine, Amazon web services, Go-grid are some applications that uses cloud computing [1].

Figure 1.Overview of Cloud Computing

Cloud computing has various deployment models. The major deployment models of cloud are Private cloud, Public cloud, Hybrid cloud and Community cloud. Private cloud is organized solely for a single organization either maintained internally or externally by a third party and it is hosted either internally or externally. For undertaking a private cloud major requirements includes virtualization the business environment and also requires the organization to re-evaluate the decisions about existing resources. Private cloud is more expensive since the operating and bandwidth costs are maintained by the organization, but at the same time private cloud is more secure. Public cloud is where in which the storage, resources, software and hardware are available to the general public via internet. Public cloud is maintained by a service provider and it is provided to users on a pay-per-use basis. Usually public cloud is suited for small or medium scale industries in which users get the hardware and other resources they want on a pay-per-use basis. Amazon, Microsoft, Google are major public cloud providers. Hybrid cloud is a combination of other deployment models. It derives the benefits of other deployment models. By utilizing the hybrid cloud, companies and individuals are able to obtain the degree of fault tolerance combined with immediate usability without dependent on internet connection. Hybrid cloud requires both on-premise and off-site server based cloud infrastructure. Community cloud is organized for a particular community say some government organization. The non-government organizations are not allowed to access the community cloud. The community cloud may be located on in-house or in the public cloud based on the need of the organization.

Cloud architecture consists of multiple cloud components that communicate with each other based on a mechanism that is loosely coupled such as messaging queue. Elasticity of cloud helps in the provision of using either a loose or tight coupling [2] depending upon the need of the users. Cloud storage comes into picture when companies and enterprises need to store a large amount of data ranging to terabytes and when the local storage does not provide sufficient space to store such a huge amount of data. Cloud storage provides the users to store their data on remote servers as backup and retrieve them when needed. Cloud allows users to upload and download the files in a feasible manner. The data is available for any authorised users accessing via a web browser from any electronic gadget. The service can automatically backup and synchronize files on any device without any extra effort put on the user.

2. Related Works

In the paper "Maximum Likelihood Network Topology from Edge based Unicast Measurements" by Mark Coates *et al* [3], a scheme requiring no clock synchronization has been introduced. This methodology has the capability to find two switching elements. A maximum probability criterion for topology recognition has been introduced. A novel Markov Chain Monte Carlo (MCMC) procedure for fast evaluation of the most likely topologies has been introduced.

The paper "Map Reduce: Simplified Data Processing on Large Cluster" by Jeffrey Dean *et al*.[4] proposes a mapreduce programming model for computing and producing large data sets. A map function is specified that executes a key/value pair and generates set of intermediate key/value pairs and reduce function that groups all intermediate values associated with the intermediate key. Many real time applications are based upon this model. Programs written in this model are automatically parallelized and computed.

In the paper "SCOPE: Easy and Efficient Parallel Processing of Massive Data Set" by Ronnie Chaiken *et al*. [5] a new programming model to handle massive data sets has been introduced. A new annotative and extensile scripting language, SCOPE (Structured Computations Optimized for Parallel Execution) has been introduced to handle these huge data sets. The language is designed to use with no clear parallelism, while designed to execute effectively on large clusters. It inherits several characteristics from SQL. Data is modelled as rows and columns. SCOPE allows nesting of expressions and also allows series of step execution.

The paper "Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing" by Dominic Battre *et al*. [6] a parallel data processor centered around a programming model called Parallelization Contracts (PACTs) and parallel execution engine Nephele has been introduced. PACT model is a map/reduce model extending it further to functions of second order as well as with Output contracts that gives guarantee about the behaviour of a function. Methods to transform PACT program into data flow for Nephele, which deals with conveyance, coexistence and fault tolerance has been proposed.

The paper "Adhoc Data Processing in the Cloud" by Dionysios Logothetis *et al*. [7] introduces a model that coordinates the power and simplicity of the mapreduce. The increased mapreduce operation avoids re-processing of data. The physical data flow of operators across the wide area is managed by stream processor. Deployment of continuous mapreduce job over a distributed engine has been introduced. Ad-hoc data processing is a powerful abstraction for mining terabytes of data. The systems give the programmers a procedural interface to process unstructured data. Today, such systems manage parallel processing tasks across thousands of machines in a single data center.

3. Existing System Model

The growing companies have huge amount of data to be stored because of which they move to the cloud. Cloud provides many virtual machines for storing the information. But the cloud providers must know the assignment and dismission of resources for the data. The right virtual machine has to be allocated on storage and de-allocated after processing.

MapReduce and hadoop model has been introduced. In this the task is instantiated to different virtual machines and revoked after completion of the process. The processing framework then takes care of dispersing the program among the available nodes and executes each exponent of the program on the appropriate fragment of data.

The MapReduce and Hadoop model has various disadvantages. It includes: More expensive to organise the model, it is more complex and the construction of the database is increased. To overcome these disadvantages and to provide parallel processing of information Nephele framework has been introduced [8]. Nephele has its own benefits over the Hadoop and MapReduce in terms of performance and cost. Even the multiplicity in organisation of the model has been reduced.

4. MTC and Nephele Model

MTC and Nephele framework helps in parallel data processing and dynamic assignment and dismission [9] of virtual machines. These models take the benefits of mapreduce. MapReduce is used to run data analysis on large set of data which is stored across large set of shared-nothing commodity. Once the program has been fitted into the required map and reduce pattern, the execution framework takes the responsibility of splitting the job into sub tasks, distributing and executing them. Thus a single map reduce job will contain different map reduce programs.

MTC helps processing of multiple tasks at the same time to reduce the time taken to complete a particular job. Thus parallel processing is achieved in MTC model. Nephele framework is used to exploit the dynamic assignment and dismission of virtual machines. This framework is used for the correct allocation of resources to the data and de-allocation of resource after the execution is completed.

There are many advantages of MTC and Nephele model. They are: Implementation of parallelism, allocation of dynamic resource, less expensive, more effective and faster than other models, designed to run data analysis job on large amount of data.

5. System Architecture

At first user must start a VM and then submit Nephele compute job. The started VM runs the so called Job Manager (JM). The Job Manager receives the client's jobs, schedules them, and coordinates their execution. Job Manager communicates with the interconnection the cloud operator provides to control the instantiation of VMs. The interface is called the Cloud Controller [10]. By means of the Cloud Controller the Job Manager can assign and dismiss VMs according to the current job execution phase.

Instance type is used to differentiate between VMs with different characteristics of hardware. *E.g.*, the type "m1.small" denotes VMs with one CPU core, one GB of RAM, and a 128 GB disk while the instance type "c1.xlarge" refers to machines with 8 CPU cores, 18 GB RAM, and a 512 GB disk. Each instance runs a (TM) Task Manager.

The Task Manager obtains the tasks from the Job Manager performs and executes the tasks and informs the JM about the completion of the task or occurrence of any errors. Unless a job is submitted to the Job Manager, the set of instances (and hence the set of Task Managers) is empty. Upon reception of job, the Job Manager decides depending on the particular task, how many and what type of instances the job should be evaluated, and when the respective instances must be assigned and dismissed to ensure a continuous and cost-efficient processing.

The allocated instances boot up a VM image and this image starts a Task Manager and that Task Manager must register with the Job Manager. When the Task Managers have registered and contacted with the Job Manager, it prompts the execution of the scheduled job. Initially before registration and starting up of the tasks the VM images are blank and do not contain any jobs. The persistent storage must contain the job's input and also collects the job's output. The storage must be accessible by both the Job Manager and Task Managers whether accessed from virtual or private network.

Figure 2. shows the system architecture. The client from public network invokes the cloud. The job manager accepts the request and passes it to the task managers. The overall cloud environment is controlled by the cloud controller. The response is passed to client via job manager.

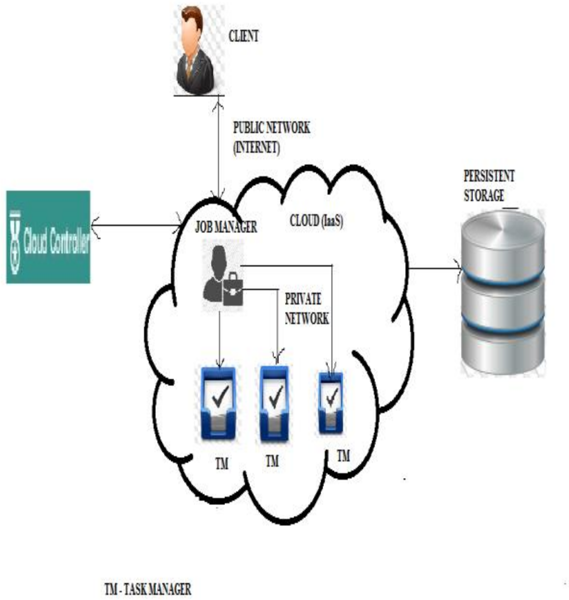

Figure 2.System Architecture

6. Execution Graph

Nephele's Job Manager transforms the valid job graph into Execution Graph. A primary data structure used for planning and observing the implementation of a Nephele job is called an execution graph. All the data needed to plan and implement the received job on cloud is contained in execution graph. It explicitly models parallelization and the mapping of tasks to instances.

Depending on the degree of comments provided by the user, Nephele have different levels in construction of Execution Graph [11]. Eg. Task 1 is split into two parallel subtasks connected to the Output 1 via file channels and are all planned to operate on the same illustration.

The execution graph thus converts valid job graph. It is used to plan and monitor all the execution and processing of the job based on task parallelization.

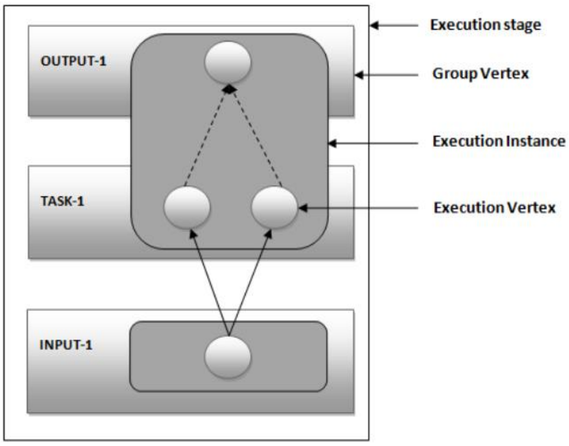

Figure 3.Execution Graph

7. Construction of ASR

The ASR (Anonymizing Spatial Region) construction is one that abides by the user's privacy requirements. Particularly, specified an anonymity degree K by user u, the ASR must satisfy two properties: (i) It includes u and at least another K * 1 user, and (ii) The LS (Location Set) knew the exact locations of all users in the system.

- An edge ordering anonymization approach for users has been proposed, which ensures K-anonymity under the strict reciprocity requirement.
- The crucial concept of border nodes, which is an important indicator of the size of CS and of the query processing cost at the LS has also been identified.
- Various edge orderings has been considered, and qualitatively assess their query performance based on border nodes [12].
- Efficient query processing mechanisms has been designed that takes advantage of the existing network database infrastructure, and ensure inclusiveness and minimality of CS. Furthermore, they apply to various network storage schemes.
- A batch execution technique has been devised for anonymous queries that significantly minimize the overhead of the LS by computation sharing.

8. Conclusion

In this paper we discussed the challenges and opportunities for proficient parallel data processing in cloud environments and also presented Nephele, the first data processing that takes advantage of the dynamic resource provisioning offered by IaaS. The basic architecture of Nephele has been presented and compared with the data processing framework hadoop. The performance evaluation gives the ability to assign virtual machine to specific task and also assign and dismiss virtual machines in course of job execution, which can improve the overall resource utilization and also reduce the processing cost.

9. Future Enhancement

We plan to address many open research issues in future. In particular we aim at improving the Nephele's ability to adapt to resource overload or underutilization automatically during execution of job. Our current profiling builds a basis for this, but still the system requires reasonable amount of user annotations.

We also focus on developing the mechanics to showcase the state of individual node at varying time interval. We also aim at developing a module that defends against any intrusion. We planned for secure transmission of information from source to destination that are forwarded by active nodes.

Acknowledgements

We are grateful and thankful to the CARD research system of our college.

References

[1] L. Ramakrishnan, C. Koelbel, Y.-S. Kee, R. Wolski, D. Nurmi, D. Gannon, G. Obertelli, A. YarKhan, A. Mandal, T. M. Huang, K. Thyagaraja, and D. Zagorodnov, "VGrADS: Enabling e-Science Workflows on Grids and Clouds with Fault Tolerance" , In Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC'09, pages 47:1-47:2, New York, NY, USA, 2009, ACM.

[2] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, "Maximum Likelihood Network Topology Identification from Edge-Based Unicast Measurements", Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 11-20, 2002.

[3] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", In Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.

[4] 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. Very Large Database Endowment, vol. 1, no. 2, pp. 1265-1276, 2008.

[5] D. Battr´e, S. Ewen, F. Hueske, O. Kao, V. Markl, and D. Warneke, "Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing", Proc. ACM Symp.Cloud Computing (SoCC'10), pp. 119-130, 2010.

[6] Dionysios Logothetis, Kenneth Yocum, "Ad-Hoc Data Processing in the Cloud", Proceedings of the VLDB Endowment, Vol 1 Issue 2, pp. 1472-1475, August 2008.

[7] D. Warneke and O. Kao, "Nephele Efficient Parallel Data Processing in the Cloud", Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers, Article No.8, 2009.

[8] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks", Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, pp. 59-72, 2007.

[9] 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 Proceedings of the 1st ACM Symposium on Cloud Computing, pp. 3-14, 2010.

[10] R. Russell, "Virtio: Towards a De-Facto Standard for Virtual I/O Devices", At ACM SIGOPS Operating Systems Review '08, Vol. 42, no. 5, pp. 95-103, 2008.

11.2. Journal Article

[1] 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 IEEE Congress on Services, pp. 199-206, 2007.

[2] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke, "Condor- G: A Computation Management Agent for Multi-Institutional Grids", In Cluster Computing 5, pp. 237-246, 2002.

Authors

V.Saranya, She pursued B.tech in Computer Science & Engineering and graduated from Rajiv Gandhi College of Engineering and Technology in 2012. Pursued MBA in HRM and graduated from Pondicherry University in 2014. Currently pursuing M.tech in Computer Science & Engineering at Rajiv Gandhi College of Engineering & Technology. Her areas of interest includes Cloud computing.

S.Ramya, She pursued B.tech in Computer Science & Engineering and graduated from Rajiv Gandhi College of Engineering and Technology in 2014. Currently pursuing M.tech in Computer Science & Engineering at Rajiv Gandhi College of Engineering & Technology. Her areas of interest includes Cloud computing.

Suresh Kumar RG, He pursued MCA at Annamalai University in 2001. Pursued M.Tech(Information Technology) from AAD University in 2005. Currently pursuing PhD in Computer Science & Engineering at Vels University. He is a Sun Certified Professional and His main research areas include Client Server computing, Cloud computing and Utility Computing.