RESEARCH INVENTY: International Journal of Engineering and Science
ISBN: 2319-6483, ISSN: 2278-4721, Vol. 1, Issue 10
(December 2012), PP 11-15
www.researchinventy.com

Operating Stoop for Efficient Parallel Data Processing In Cloud

Krishna Jyothi K [1], Dr.R.V.Krishnaiah[2]
[1]*Department of CSE, DRK College of engineering & Technology, Ranga Reddy, Andhra Pradesh, India*
[2]*Principal*
Department of CSE, DRK Institute of Science & Technology, Ranga Reddy, Andhra Pradesh, India

Abstract: *Major Cloud computing companies have started to integrate frameworks for parallel data processing in their product portfolio, making it easy for customers to access these services and to deploy their programs. However the processing frameworks which are currently used have been designed for static, homogeneous cluster setups and disregard the particular nature of a cloud. Consequently, the allocated computer resources may be inadequate for big parts of the submitted job and unnecessarily may increase processing time and cost. We discuss here the opportunities and challenges for efficient parallel data processing in clouds and present our research project Nephele. Particular tasks of a processing job can be assigned to different types of virtual machines which are automatically instantiated and terminated during the job execution. Based on this new framework, we perform extended evaluations of Map Reduce-inspired processing jobs on an IaaS cloud system and compare the results to the popular data processing framework Hadoop.*

Index Terms—*Many-Task Computing, High-Throughput Computing, Loosely Coupled Applications, Cloud Computing*

I. Introduction

Today a growing number of companies have to process huge amounts of data in a cost-efficient manner. Classic representatives for these companies are operators of Internet search engines, like Google, Yahoo, or Microsoft. The vast amount of data they have to deal with every day has made traditional database solutions prohibitively expensive. Instead, these companies have popularized an architectural paradigm based on a large number of commodity servers. Problems like processing crawled documents or regenerating a web index are split into several independent subtasks, distributed among the available nodes, and computed in parallel. In order to simplify the development of distributed applications on top of such architectures, many of these companies have also built customized data processing frameworks. Examples are Google's Map Reduce, Microsoft's Dryad, or Yahoo!'s Map-Reduce Merge. They can be classified by terms like high-throughput computing (HTC) or many-task computing (MTC), depending on the amount of data and the number of tasks involved in the computation. Although these systems differ in design, their programming models share similar objectives, namely hiding the hassle of parallel programming, fault tolerance, and execution optimizations from the developer. Developers can typically continue to write sequential programs. The processing framework takes care of distributing the program among the available nodes and executes each instance of the program on the appropriate fragment of data. For companies that only have to process large amounts of data occasionally running their own data center is obviously not an option. Instead, Cloud computing has emerged as a promising approach to rent a large IT infrastructure on a short-term pay-per-usage basis. Operators of so-called IaaS clouds, like Amazon EC2, let their customers allocate, access, and control a set of virtual machines (VMs) which run inside their data centers and only charge them for the period of time the machines are allocated. The Virtual Machines are typically offered in different types, each type with its own characteristics (number of CPU cores, amount of main memory and cost).

The main goal is to decrease the overloads of the main cloud and increase the performance of the cloud. In recent years ad-hoc parallel data processing has emerged to be one of the killer applications for Infrastructure-as-a-Service (IaaS) clouds. Major Cloud computing companies have started to integrate frameworks for parallel data processing in their product portfolio, making it easy for customers to access these services and to deploy their programs. However, the processing frameworks which are currently used have been designed for static, homogeneous cluster setups and disregard the particular nature of a cloud.

II. Related Work

Today a growing number of companies have to process huge amounts of data in a coefficient manner. Classic representatives for these companies are operators of Internet search engines, like Google, Yahoo Microsoft. The vast amount of data they have to deal with every day made the traditional database solutions prohibitively expensive.

Companies only have to process large amounts of data occasionally running their own data center is obviously not an option. Instead, Cloud computing has emerged as a promising approach to rent a large IT infrastructure on a short-term pay-per-usage basis. Operators of so called IaaS clouds, like Amazon EC2 , let their customers allocate, access, and control a set of virtual machines (VMs) which run inside their data centers and only charge them for the period of time the machines are allocated.

III. Architecture

The execution of tasks which a Nephele job consists of is carried out by a set of instances. Each instance runs a so-called Task Manager (TM). A Task Manager receives one or more tasks from the Job Manager at a time, executes them, and after that informs the Job Manager about their completion or possible errors. Unless a job is submitted to the Job Manager, we expect the set of instances (and hence the set of Task Managers) to be empty. Upon job reception the Job Manager then decides, depending on the job and particular tasks, how many and what type of instances the job should be executed on, and when the respective instances must be allocated/deallocated to ensure a continuous but cost-efficient processing.

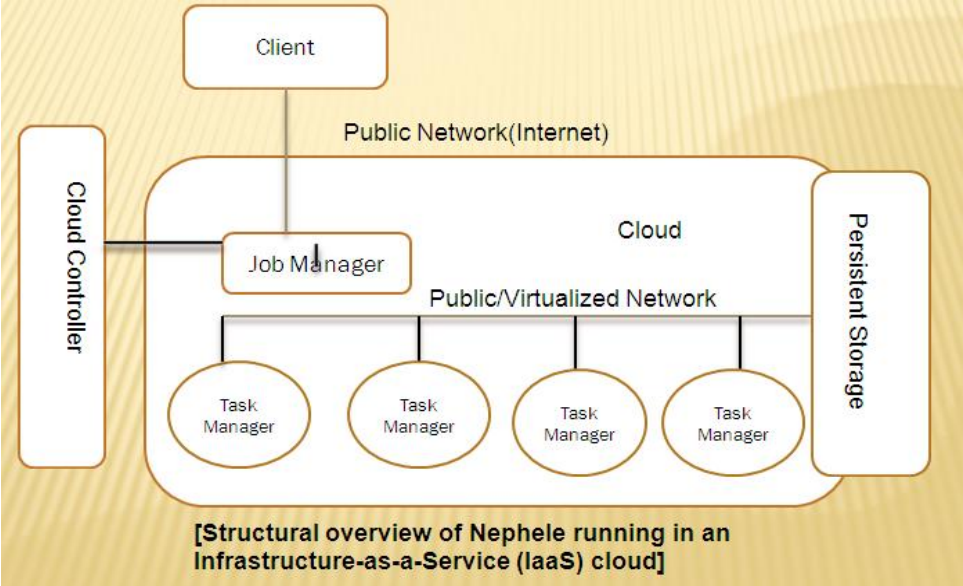

Fig1.1 Nephele Architecture for cloud

Before submitting a Nephele compute job, a user must start a Virtual Machine in the cloud which runs the so called Job Manager (JM).The Job Manager receives the client's jobs, and is responsible for scheduling them, and coordinates their execution. It is capable of communicating with the interface. The cloud operator provides to control the instantiation of Virtual Machines. We call this interface the Cloud Controller. By means of the Cloud Controller the Job Manager can allocate or deallocate Virtual Machines according to the current job execution phase. These comply with common Cloud computing terminology and refer to these Virtual Machines as instances. The term instance type will be used to differentiate between Virtual Machines with different hardware characteristics.

The actual execution of tasks which a Nephele job consists of is carried out by a set of instances. Each instance runs a so called Task Manager (TM). A Task Manager receives one or more tasks from the Job Manager at a time, executes them, and after that informs the Job Manager about their completion or possible errors. Upon job reception the Job Manager then decides, depending on the job's particular tasks, how many and what type of instances the job should be executed on, and when the respective instances must be allocated/deallocated to ensure a continuous but cost-efficient processing.

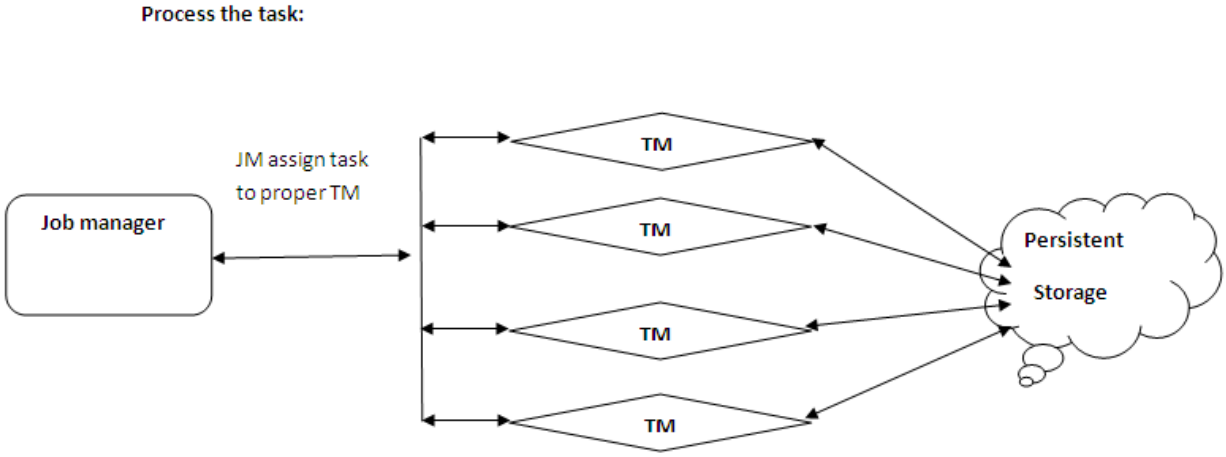

Fig 1.2 Segregating of Tasks

The newly allocated instances boot up with a previously compiled Virtual Machine image. The image is configured to automatically start a Task Manager and register it with the Job Manager. Once all the necessary Task Manager have successfully contacted the Job Manager, it triggers the execution of the scheduled job.Nephele are expressed as a directed acyclic graph (DAG).Each vertex in the graph represents a task of the overall processing job, the graph's edges define the communication flow between these tasks.Due to the dynamic nature of cloud more than one user can access the resource at a time. Also more users can store their documents in a cloud resource hence the security is important problem. The hackers can hack the password if it is in textual format so an alternative approach is used here to overcome

IV. Results

We create a user page using graphical user interface, which will be the media to Connect user with the cloud and through which client is able to give request to the cloud and cloud server can send the response to the client, through this we can establish the communication between client and cloud. In this page user is able to know about the overview of the whole application and get better knowledge about the whole application.

Before client creation we check the user credentals by login page, we receive the username and password by the user and we will check in the database, that the user has the credential or not to give request to the cloud.If the client wants to register we can add new user through user registration by taking all the important details like user's name, gender, username, password, address, email id, phone no from the user

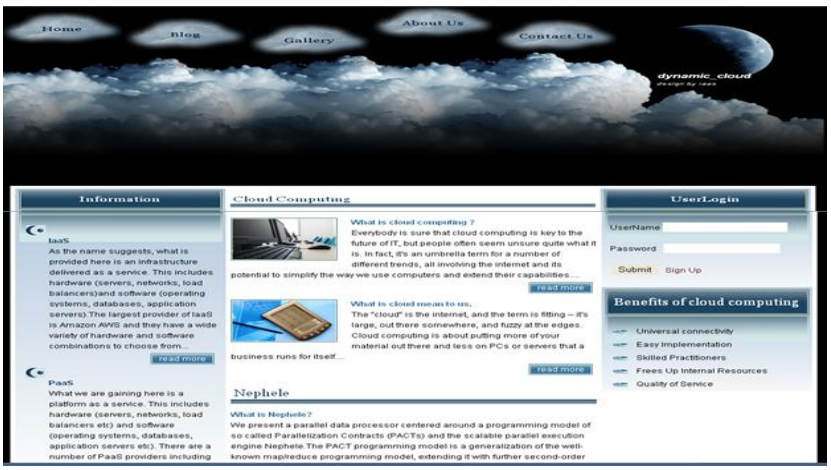

Fig:1.3 Home page of cloud

We create a home page where all the resources would be displayed, the resources our cloud can provide and what are the services cloud can give, from here , user can choose the resources and services they need for their application. This page will be connected to the job manager.

Fig 1.4 Resource Page of cloud

we create a job manager (JM) who will be the interface between the cloud and resources (task manager) and who will schedule and allocate the job of the client to the proper resources. we create the task manager who will be under the Job Manager and will process the task that is allocated by the Job manager, after processing the task it would be sent to the client. User can get details about the resources available in cloud and they can choose resource as their requirement

To receive the sql type of request, user can put any type of sql request here by clicking on the execute button.The request is sent to the appropriate sql server on the cloud and the result will be posted back to the client machine

Fig: 1.5 SQL Page of cloud.

we process the client task by appropriate task manager assigned by the Job manager.Every task manager has the capacity of handling any type of job by taking the help of the persistent storage, as all the resources which are needed for processing a job are present in the persistent storage.

To receive the sql type of request, user can put any type of sql request here by clicking on the execute button.The request is sent to the appropriate sql server on the cloud and the result will be posted back to the client machine

V. Conclusion

In this paper, we have discussed the challenges and opportunities for efficient parallel data processing in cloud environments and presented Nephele, the first data processing framework to exploit the dynamic resource provisioning offered by today's IaaS clouds.In this we described Nephele's basic architecture and presented a performance comparison to the well-established data processing framework Hadoop. The performance evaluation gives a first impression on how the ability to assign specific virtual machine types to specific tasks of a processing job, as well as the possibility to automatically allocate/deallocate virtual machines in the course of a job execution, can help to improve the overall resource utilization and, consequently, reduce the processing cost. With a framework like Nephele at hand, there are a variety of open research issues, which we plan to address for future work. In particular, we are interested in improving Nephele's ability to adapt to resource overload or underutilization during the job execution automatically. Our current profiling approach builds a valuable basis for this, however, at the moment the system still requires a reasonable amount of user annotations.In general, we think our work represents an important contribution to the growing field of Cloud computing services and points out exciting new opportunities in the field of parallel data processing.

References

[1] Amazon Web Services LLC, "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon.com/ec2/, 2009.
[2] Amazon Web Services LLC, "Amazon Elastic MapReduce," http://aws.amazon.com/elasticmapreduce/, 2009.
[3] Amazon Web Services LLC, "Amazon Simple Storage Service, "http://aws.amazon.com/s3/, 2009.
[4] D. Battre´, 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.
[5] 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.

AUTHORS

	Krishna Jyothi K student of DRK College of Emgineering and technology, Hyderabad, AP, INDIA. She has received MCA Degree in computer science and , M.Tech Degree in computer science and engineering. Her main research interest includes Data mining, Networking
	Dr.R.V.Krishnaiah is working as Principal at DRK INSTITUTE OF SCINCE & TECHNOLOGY, Hyderabad, AP, INDIA. He has received M.Tech Degree EIE and CSE. His main research interest includes Data Mining, Software Engineering.