An Analysis of Nephele Framework in Cloud for Efficient parallel data processing

Asish Srivastava, Tinesh Kumar Goyal, Amitesh Kumar Gupta, Piyush Saxena

***Abstract*— The term virtualization refers to creating a virtual environment instead of an actual physical one. This enables a physical system to run different logical solutions on it by virtually creating an environment that meets the demand of the solution. By virtually creating several different operating systems on one physical workstation the administrator can create a cluster of computers that acts as if they were physical.**

Without virtualization situations would arise where machines would use only a percent-age of their maximum capacity. If the server would have virtualization active and enable more operating systems run on the physical hardware, the hardware would be used more effectively. This is why server and machine virtualization is of great benefit when creating a cloud environment because the cloud host can maximize effectively and distribute resources without having to buy a physical server each time a new instance is needed.

This paper is to improvise and optimize the scenario of parallel data processing in cloud. In order to avail utmost client satisfaction, the host server needs to be upgraded with the latest technology to fulfil all requirements. This will also make sure that the user gets its entire requirement fulfilled in optimal time. This will improve the overall resource utilization and, consequently, reduce the processing cost with the use of Nephele Framework.

***Index Terms*— Map reduce algorithm, Homogeneous cluster setup Computing Reference Architecture (CC RA), high throughput computing (HTC) or many-task computing (MTC),Data mining, Data warehousing, Parallel data processing.**

I. Introduction

By using a cloud service a company, organization or even a private person can outsource management, maintenance and administration of large clusters of servers but still keep the benefits. While using a public cloud provider is sufficient for most tasks; bandwidth, storage, data protection or pricing details might encourage companies to house a private cloud. The infrastructure to control and maintain the cloud can be proprietary like Microsoft Hyper-V Cloud, VMware vCloud and Citrix Open Cloud, but there are also a number of free and open-source solutions like Eucalyptus Cloud, Open Nebula and Cloud Stack.

The cloud can provide the processing power, but the actual framework to take benefit of these distributed instances does not inherently come with the machines. The Hadoop MapReduce claims to provide very high scalability and stability across a large cluster. It is meant to run on dedicated servers, but there is nothing that limits them from running on a virtual machine. This project is a study performed to provide familiarity with the cloud and its related technologies in general, focusing specially on the Eucalyptus cloud infrastructure. It shows a mean of setting up a private cloud, along with using the Hadoop MapReduce idiom/framework and Nephele framework on top of the cloud showing the benefits and requirements of running MapReduce on a Eucalyptus private cloud. As a proof of concept a simple MapReduce test is implemented and tested on the cloud to provide an analysis of the distributed computation of MapReduce.

The report will have a software study on the systems used in the project, followed by a description of the configuration, setup and usage of Eucalyptus, Hadoop and Nephele. Finally the result from the analysis will be presented along with a short conclusion. Today's processing frameworks typically assume the resources they manage consist of a static set of homogeneous compute nodes. Although designed to deal with individual nodes failures, they consider the number of available machines to be constant, especially when scheduling the processing job's execution. This new paradigm allows allocating compute resources dynamically and just for the time they are required in the processing workflow.

II. Existing system (Hadoop)

An In recent years a variety of systems to facilitate MTC has been developed. Although these systems typically share common goals (e.g. to hide issues of parallelism or fault tolerance), they aim at different fields of application. MapReduce (or the open source version Hadoop) is designed to run data analysis jobs on a large amount of data, which is expected to be stored across a large set of share-nothing commodity servers. MapReduce is highlighted by its simplicity: Once a user has fit his program into the required map and reduce pattern, the execution framework takes care of splitting the job into subtasks, distributing and executing them. A single MapReduce job always consists of a distinct map and reduce program. However, several systems have

Manuscript received April 20, 2014

Asish Srivastava, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-9555531144

Tinesh Kumar Goyal, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-9417170800.

Amitesh Kumar Gupta, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-7532836750.

Piyush Saxena, M.Tech (Computer Science and Engineering), Amity University, Amity School of Engineering and Technology, Noida, India, +91-9451427546.

been introduced to coordinate the execution of a sequence of MapReduce jobs.

MapReduce has been clearly designed for large static clusters. Although it can deal with sporadic node failures, the available compute resources are essentially considered to be a fixed set of homogeneous machines.

III. Proposed system (Nephele)

All Based on the challenges and opportunities we have designed Nephele, a new data processing framework for cloud environments that matches the dynamic and opaque nature of a cloud. With Nephele compute job, a user must start an instance inside the cloud which runs the so called Job Manager. The Job Manager receives the client's jobs, is in charge for scheduling them and coordinating their execution.

It has the power to allocate or de-allocate virtual machines according to the current job execution phase. The execution of tasks is done by a set of instances and runs a local component of the Nephele framework known as Task Manager. It also decided when respective instances must be allocated or reallocated in order to ensure a continuous but cost-efficient processing.

This paper serves as the definition of the Cloud Computing Reference Architecture (CC RA). A Reference Architecture (RA) provides a plan of a to-be-model with a well-defined range, needs it satisfies, and architectural decisions it realizes. An RA ensures steadiness and worth across improvement and release projects. The mission of the CC RA is defined as follows:

Definition of a single Cloud Computing Reference Architecture, enabling cloud-scale economics in delivering cloud services while optimizing resource and labour utilization and delivering a design blueprint for:

- Cloud services, which are offered to customers
- Private, public or hybrid cloud projects
- Workload-optimized systems
- Enabling the management of multiple cloud services (across I/P/S/BPaaS) based on the same, common management platform for enabling economies of scale.

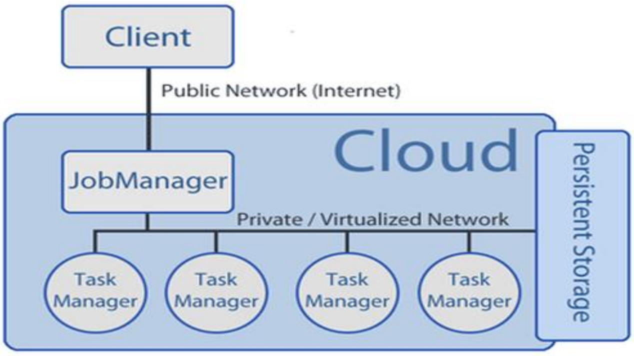


FIGURE I NEPHELE ARCHITECTURE

IV. Literature Survey

In order to simplify the development of distributed programs on top of such architectures, many of the companies have also built customized data processing frameworks. 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. The system programming models have similar objectives [3], namely hiding the hassle of parallel programming, fault tolerance, and execution optimizations from the developer.

For companies that only have to process large amounts of data occasionally need not own data centre, instead Cloud computing is used to rent a large IT infrastructure on a short-term pay-per-usage basis [2]. Ubuntu is the only Linux distribution to position itself as a true cloud OS [1] with three initial components of our cloud strategy already released. Two of these components are aimed at the infrastructure layer of the computing stack (commonly called IaaS or Infrastructure as a Service) while one component is aimed at the software layer.

The three components are known as:

- Ubuntu Server Edition on Amazon EC2 (IaaS)
- Ubuntu Enterprise Cloud powered by Eucalyptus (IaaS)
- Ubuntu One (SaaS)

Even though Ubuntu One is an important initiative for Canonical to deliver added functionality to its large user base, it should clearly be distinguished from the other two components as it focuses on consumer use of the cloud rather than the tools required to build cloud services. The approach enables Canonical to successfully leverage open source against many of the traditional software monopolies. Ubuntu provides a real and useable alternative to the operating system/productivity suite that once dominated the world.

Our cloud strategy follows this same mission; to select the best components from open source, to assemble and refine them, to encourage ecosystem development and to provide the best possible experience to our users whilst avoiding lock-in and the creation of monopolies in this new cloud industry. Since John McCarthy & Douglas Parkhill [2] in the 1960s, whilst many people, have predicted and described the development of the cloud computing industry, this transition from a product to a more service oriented and cloudy world has only begun in earnest during the last decade. This transition won't happen in one step and there exists many barriers to its current use such as governance, security of supply, lack of second sourcing options and trust.

However, the shift has started and the overwhelming benefits of cloud computing (i.e. elastic infrastructure, efficiency of resource utilization, reduction of capital expenditure, and focus on core activities) is likely to accelerate this change. What this means is that more and more of the applications [4] that we use today on our personal computers or servers will soon migrate to the cloud and self-service IT environments. While the benefits of public clouds are numerous, free software advocates frequently warn of the risks that the use of proprietary software and data formats implies for the long term survival of their data. In the cloud, these risks are heightened and new risks such as a lack of transparency in relationships appear.

V. Modules

Client Module: This Client module deals with the Client or the Customer whose needs are to be fulfilled. Functions such as file exchange, file access and database access are built on the client model. Users accessing services from their computer use client software to send a request to server. The client always requests to the server for executing a particular operation and send a response back to it accordingly. The client selects the file that it wants to download. After clicking the download button, it waits for the server to send a response back.

Server Module: The server module is a computing model that acts as a distributed program that divides tasks between the providers of a resource or service, called servers. The server is normally located remotely and is used to service requests from multiple clients. The Servers take the request from the client and try to fulfil these requests by providing the resources the clients need. The server is always responsible for maintaining resources and allocating them as required by the host clients. The server gives a function or service to one or many sub-units, even manages and controls data processing between server & client. In order to have efficient parallel data processing where many clients participate to execute certain tasks, the server needs to execute data processing faster and efficiently. The server is an entity that services the request made by the client and consists of two important criterions that are job manager and task manager. The resource allocation is used if the server needs to allocate resources dynamically at the same time when the file is getting downloaded. The data distribution type and the data processing will be parallel in nature.

Job Manager Module: In Nephele, the job manager is the central component for communicating with clients, creating schedules for incoming jobs, and supervising the execution of the jobs. A job manager may only exist once in the system and its address must be known to all clients. Task Managers can discover the job manager by means of an UDP broadcast and then advertise themselves as new workers for tasks. If a job graph is submitted from a client to the job manager, each task of the job will be sent to a task manager.

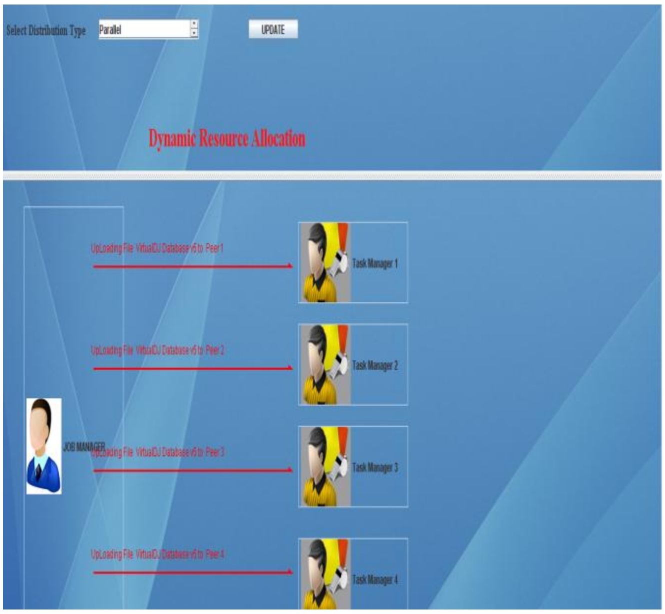
Task Manager Module: A task manager receives tasks from the job manager and executes them. After having executed them (or in case of an execution error) it reports the execution result back to the job manager. Task managers are able to automatically discover the job manager and receive its configuration when the job manager is running on the same local network. The task manager periodically sends heartbeats to the job manager to report that it is still running.

Instance Manager Module: In Nephele an instance manager maintains the set of available compute resources. It is responsible for allocating new computer resources and provisioning available compute resources to the Job Manager. Additionally it is keeping track of the availability of the utilized compute resources in order to report unexpected resource outages. For this purpose the instance manager receives heartbeats of each task manager. If a task manager has not sent a heartbeat in the given heartbeat interval, the host is assumed to be dead. The instance manager then removes the respective task manager from the set of compute resources and calls the scheduler to take appropriate actions.

VI. Conclusions

In this project we have come across various challenges and explored 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. We have 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.



Select Distribution Type: Parallel

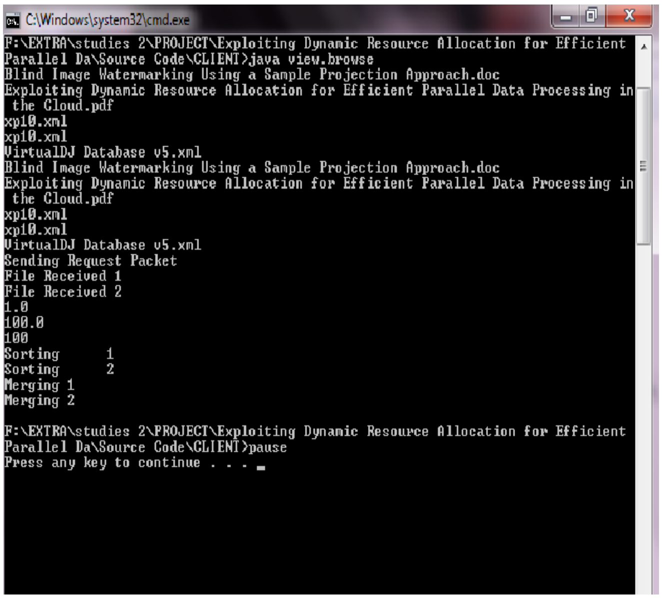
UpLoading File VirtualDJ Database v5 to Peer1

UpLoading File VirtualDJ Database v5 to Peer2

UpLoading File VirtualDJ Database v5 to Peer3

UpLoading File VirtualDJ Database v5 to Peer4

FIGURE III RESOURCE ALLOCATION IN ACTION



```
F:\EXTRA\studies 2\PROJECT\Exploiting Dynamic Resource Allocation for Efficient
Parallel Da\Source Code\CLIENT>java view.browse
Blind Image Watermarking Using a Sample Projection Approach.doc
Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in
 the Cloud.pdf
xp10.xml
xp10.xml
VirtualDJ Database v5.xml
Blind Image Watermarking Using a Sample Projection Approach.doc
Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in
 the Cloud.pdf
xp10.xml
xp10.xml
VirtualDJ Database v5.xml
Sending Request Packet
File Received 1
File Received 2
1.0
100.0
100
Sorting      1
Sorting      2
Merging 1
Merging 2

F:\EXTRA\studies 2\PROJECT\Exploiting Dynamic Resource Allocation for Efficient
Parallel Da\Source Code\CLIENT>pause
Press any key to continue . . .
```

FIGURE IIIII COMMAND INTERFACE

VII. FUTURE SCOPE

Eyeing for a better future in cloud, we have concluded from this project that parallel processing in cloud is an important criteria in today's web processing. Featuring better resource allocation for efficient parallel data processing and eliminating the disadvantages of Hadoop was the first priority of our project. Nephele allows eliminating homogeneous cluster setup in a cloud and most importantly it enriches the dynamic nature of the cloud.

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. Virtual machines (computing nodes) need not be fixed as in case of a homogeneous cluster setup. Nephele allows automatic instantiation and termination of virtual machines amplifying the dynamic nature of cloud.

REFERENCES

[1] Irfan Gul, M. Hussain. 2011. Distributed Cloud Intrusion Detection Model. International Journal of Advanced Science and Technology.

[2] Ms. Parag K. Shelke, Ms. Sneha Sontakke, Dr. A. D. Gawande. 2012. Intrusion Detection System for Cloud Computing. International Journal of Scientific & Technology Research Volume 1.

[3] Piyush Saxena, Satyajit Padhy, Praveen Kumar 2013.USE OF STORAGE AS SERVICE FOR ONLINE OPERATING SYSTEM IN CLOUD COMPUTING, International Conference on Telecommunications and Networks (TEL-NET 2013)

[4] Security Guidance for Critical Areas of Focus in Cloud Computing, April 2009. DOI

[5] Miranda Mowbray, Siani Pearson. 2009."A Client-Based Privacy Manager for Cloud Computing." COMSWARE '09: Proceedings of the Fourth International ICST Conference on Communication System software and middleware

[6] Jinpeng Wei, Xiaolan Zhang, Glenn Ammons, Vasanth Bala, Peng Ning. 2009."Managing security of virtual machine images in a cloud environment." CCSW '09: Proceedings of the 2009 ACM workshop on Cloud computing security.

[7] B.Meena, Krishnaveer Abhishek Challa. 2012."Cloud Computing Security Issues with Possible Solutions." in IJCST

[8] Steve Hanna. A security analysis of Cloud Computing. Cloud Computing Journal. DOI

[9] Vahid Ashktorab, Seyed Reza Taghizadeh, 2012. "Security Threats and Countermeasures in Cloud Computing" in IJAIEM.

[10] Frank Doelitzscher, Christoph Reich, Martin Knahl, Alexander Passfal and Nathan Clarke. 2012. An agent based business aware incident detection system for cloud environments. Journal of Cloud Computing: Advances, Systems and Applications

[11] Peter Mell, Timothy Grance. 2011. The NIST Definition of Cloud Computing (Draft). NIST

[12] Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari, Joaquim Celestino Junior. 2013. An intrusion detection and prevention system in cloud computing: A systematic review. Journal of Network and Computer Applications.

[13] Hassen Mohammed Alsafi, Wafaa Mustafa Abduallah and Al-Sakib khan Pathan. 2012. IPS: An Integrated Intrusion Handling Model for Cloud Computing Environment, International Journal of Computing and Information Technology (IJCIT)

[14] Deris Stiawan, Abdul Hanan Abdullah, Mohd. Yazid Idris. 2011. Characterizing Network Intrusion Prevention System. International Journal of Computer Applications.

[15] Dinesh Sequeira. 2002. Intrusion Prevention Systems-Security Silver Bullet. SANS Institute.

Asish Srivastava Pursuing Master of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India. Area of Interest: Cloud Computing, Virtualization, Computer Networks and Network Security.

Tinesh Kumar Goyal Pursuing Master of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India, Area of Interest: Cloud Computing, Computer Networks and Software Engineering.

Amitesh Kumar Gupta Pursuing Masters of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India, Area of Interest: Computer Networks, Network Security, Cloud Computing, Data Mining and Warehousing.

Piyush Saxena Pursuing Master of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India, Area of Interest: Cloud Computing, Data Mining and Warehousing and Soft Computing.