Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 11; April-June, 2015 pp. 103-106

http://www.krishisanskriti.org/acsit.html

Implementation of Distributed Application in Virtualized Cloud Environment using Aneka

Ram Mohan Rao Kovvur[1] and M. Swamy Das[2]

[1]Dept. of CSE Vasavi College of Engg., Hyderabad, India
[2]Dept. of CSE CBIT, Hyderabad, India
E-mail: [1]krmrao@staff.vce.ac.in, [2]msdas@cbit.ac.in

Abstract—*Cloud computing is a class of parallel and distributed system comprising of a collection of virtualized computers or computers are delivered on demand to the consumers over Internet based on SLA between the service provider and customers. Virtualization is one of the key technologies with Cloud computing utilized to create virtual resources. Aneka is an implementation of Platform as a Service approach on the Cloud Environment employed to design and implement distributed applications. However, Aneka does not support virtualization to create virtual machines. Thus in this paper we have created virtual machines using VMware Esx$_i$ and implemented distributed application on virtual machines with Aneka. We study the impact of number virtual machines and size of the file. From our experiments it is clear that as number of virtual machines increases, computation efficiency improves gradually.*

Keywords*: Cloud, Virtualization, Distributed, Service;*

1. INTRODUCTION

Cloud computing is a utility based computing (pay-as-you-go model) to deliver the Infrastructure, Platform and Software as services on demand over the internet based on SLA (Service Level Agreements) among the service providers and customers meeting the QoS (Quality-of-Service) constraints.[1]

Virtualization is a software layer in between the bare hardware and the operating system. Basically, this software layer does is to separate the resources of the bare hardware between all the guest operating systems on the virtual machines [1, 2]. In computing, virtualization is broadly refers to the abstraction of computer resources such as disk, memory, file etc. **Resource virtualization**, the virtualization of system resources such as storage, memory and network resources. **Virtual memory** allows contiguous addressing of physically memory and non-contiguous memory of disk storage**. Storage Virtualization**, the process of completely abstracting logical storage from physical storage. **Network virtualization**, creation of a virtualized network addressing space within or across network subnets. **Multi-computer (cluster) and grid computing** is a collection of homogenous or heterogeneous computers in local area network or wide area network respectively. [2, 14, 15]

A hypervisor or virtual machine monitor (VMM) is a piece of computer software, on the hardware that creates and runs virtual machines. A computer on which a hypervisor is running one or more virtual machines is defined as a host machine. Each virtual machine is called a guest machine. The hypervisor presents the guest OSs with a virtual operating platform and manages their execution. There are well-known virtualization software's on Windows and Linux host Operating System. The various virtualization software's on Windows Operating as host OS such as VMware Workstation (any guest OS), Virtual Box (any guest OS), Hyper-V (any guest OS). Whereas the various virtualization software's on Linux as the host OS such as VMware Workstation, Microsoft Virtual PC, VMLite Workstation, Xen. [16]

The Hypervisors basically classified into two categories namely Type 1 i.e., Bare metal installation and Type 2 i.e., Hosted installation. A Type 1 hypervisor is a client hypervisor that interacts directly with hardware that is being virtualized. It is absolutely autonomous from the operating system, where as Type 2 hypervisor, and boots prior to the operating system. A Type 2 hypervisor is a class of hypervisor that sits on top of an operating system. In contrast to Type 1, a Type 2 hypervisor demands heavily on the operating system. [16]

In our study, we used Type 1 Hypervisor i.e., Bare Metal installation VMware Virtualization namely VMware ESX$_i$ for creation of virtual machines for our experiments.

Aneka is a Cloud Application Development Platform (CAP) for developing and executing compute and data intensive applications. As a platform it provides users with both runtime environments for executing applications developed using any of the three supported programming models, and a set of APIs and tools that allow you to design new applications or run existing legacy code. [9, 10]

Aneka allows various types of applications such as scientific computing, social computing to be executed on the Grid/Cloud Infrastructure using Globus/ Aneka respectively. To sustain such flexibility it offers different abstractions through which it is achievable to implement multithreaded distributed

applications. These abstractions are mapped to different execution models. Presently, Aneka frame work supports three different models namely [12]

- Task Model
- Thread Model
- Map Reduce Model

In this paper, we have implemented the Thread Model for developing multithreaded distributed applications on remotely executable threads with Aneka. The Thread Model allows developers to rapidly virtualizes multi-threaded applications with Aneka. It introduces the concept of *AnekaThread* that represents a thread that is executed on a remote computing machine in the Aneka network.

Christian et al. Presented "Deadline-Driven provisioning of resources for scientific applications in hybrid Clouds with Aneka" for providing quality of service execution of scientific applications in hybrid clouds resources acquired from various sources.[4] Rodrigo et al. proposed "The Aneka platform and QOS-driven resource provisioning for elastic applications on hybrid Clouds" for extending scalable applications on the Hybrid Aneka cloud by integrating desktop Grids and Clouds.[5] Sunil et al. presented "Resource Management for Infrastructure as a Service in Cloud Computing" focuses on various resource management techniques such as resource provisioning, resource allocation, resource mapping and resource adaptation for IaaS in Cloud Computing.[6] Giuseppe et al. proposed "Cloud Monitoring: A Survey" discusses motivation for cloud monitoring, issues arising in cloud monitoring and describes services for cloud monitoring.[7] In this study, we implemented distributed application on a virtualized cloud environment using Thread Model of Aneka.

The rest of the paper is organized as follows. In section 2, we outline the Cloud model used in this work. Section 3 describes the proposed algorithm for Thread Model of Aneka cloud. Our experimental results are presented in section 4. Finally we conclude in section 5.

2. CLOUD MODEL

The In our study, we consider the Cloud model as shown in Fig.1 The cloud environment consists of Physical machines which are interconnected through LAN/ WAN. At systems level/site, there is a Cloud Resource consisting of multiple physical machines of dissimilar processing capability and just above systems level there is Virtualization layer with a set of virtual machines are created as per request of the end user. At resource allocator level, consists core middleware with various modules such as execution &Monitoring module, SLA, Metering, pricing and Dispatcher. At cloud user level consists of user level middleware comprises of various modules such as concurrent & distributed programming, scripting, workflows and libraries. Cloud application has distributed application such as scientific and social computing to be scheduled on the multiple machines/virtual machines by the cloud resource allocator/ scheduler. [2, 3]

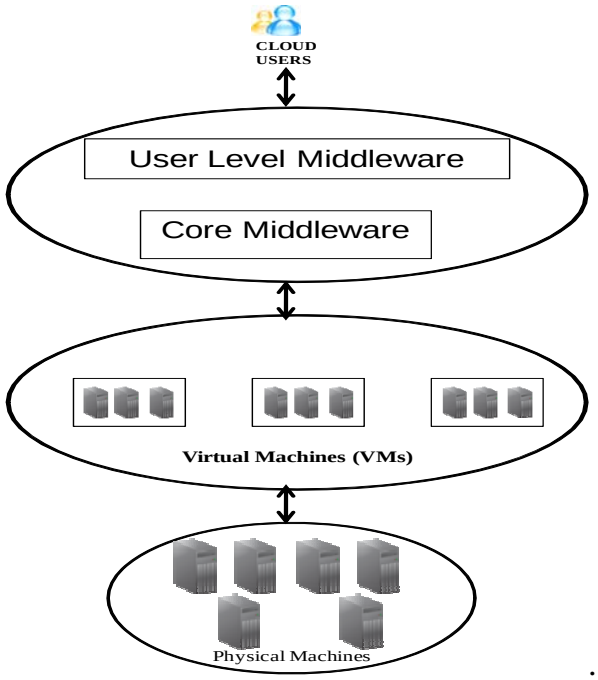

Fig. 2 Cloud Model

3. PROPOSED ALGORITHM FOR THREAD MODEL OF ANEKA

The wordcount class uses the Thread Model for performing the distributed execution of Text file (*.txt) for searching of given searched word and combing the results at the master Node. This is responsible of managing the interaction with Aneka and controlling the execution of the distributed application by managing the AnekaThread. [12] The pseudo code of main method is depicted in Fig. 2.

```
static void Main(string[] args)
{
// the initialisation of Aneka Thread Manager;
app = new AnekaApplication
   <AnekaThread,ThreadManager>(conf);

// Reading list of text File with extension *.txt;


hw = new WordCountClass(word, s.ToString());

// sending WordCount method as an argument to one of the
   node

th[i] = new AnekaThread(hw.WordCount, app);


// where i = Number of threads;
// these threads are assigned to freely available nodes
}
}
```

Fig. 2: Pseudo code of main method of WordCount class

The thread method wordcount reads lines from the input text file and further reads characters from the line and splits it into words. After splitting words, first it compares length of each word with the searching word. In case length is equal, then compare each character of splited word with searching word. If word matches, then increment wordcount variable. The pseudo code of method wordcount is depicted in the figure Fig. 3.

```
public void Wordount() // wordcount method
{
   Step 1: Reading lines from the text

 while (( Read line) != null)
 {
 Step 2: Reading characters from the each line
 for (i = 0; i < line.Length; i++)
 Reading characters from the line;
 Step 3: collecting words from the each line

       for ( i = 0; i < convert_array.Length; i++)
        {
 if (convert_array[i] == ' ' || convert_array[i] == '\0')
 {
 string s = new string(word);
 int f, j;
 Step 4: Comparing each word length with searching word

 if (count_charcter == a.Length)
 {
 Step 4.1: Comparing each character of splitted word with
searching word
 for (j = 0; j < Length of searching word; j++)
 {
 if (a[j] == s[j])
 incrementing length of spilted word;
 }

 if (length of spilted word == length of Searching word)
 count++; // counting number of words
 }
 count_charcter = 0;
 //initializing character count for next word
 }
 else
 {
 word[count_charcter] = convert_array[i];
 count_charcter++;
 } } }}}
```

Fig. 3: Pseudo code of method wordcount

4. EXPERIMENTAL RESULTS AND ANALYSIS

The In this section, we present the results of implementation distributed application on virtualized cloud environment using Aneka. We used the VMware Esx_i for creation of virtual machines and created cloud environment on these virtual machines using Aneka for our experiments. Aneka frame work is used for processing huge data in parallel on multiple computers running in a cluster/grid, functioning as if they were single large computer.

The Thread Model allows to virtualizes multi-threaded applications with Aneka. It starts the execution of the distributed application by creating AnekaThread instances. It uses *AnekaThread* to execute on a remote computing node on the Aneka network. [11, 12]

4.1 Experimental Setup and Cloud Environment

We have setup a cloud environment system on 4 physical nodes by creating 2 virtual machines in each, which are Pentium- i5 based systems with processor clock speed of 3.0 GHz, 4GB Memory and Windows 7.0 professional operating System running on them. We used Aneka to setup cloud environment on 8 virtual machines. We conducted experiments by varying (i) the file size and (ii) the number of virtual Machines.

Distributed application used for the experiments is *WordCount* example using *thread* programming model of Aneka. It counts the number of each word which appears in a large number of documents. The disk space needed to store all the text documents cannot be accomplished by single machine. As a result, these text documents distribute over a multiple machines. The *WordCount* application uses the *threads* operations in order to count the occurrences of one word into a document and to sum all the occurrences of the same words computed from different files.

Experiment 1 *(Varying number of virtual machines):* In this experiment our objective is to study the impact of varying number of virtual machines on cloud Environment performance. We considered the 10 No. Text files with each of 25 MB of size are distributed over multiple virtual machines through Aneka Master. From Fig.4, we observe that computational performance of the Cloud Environment is improved as the number of virtual machines increased.

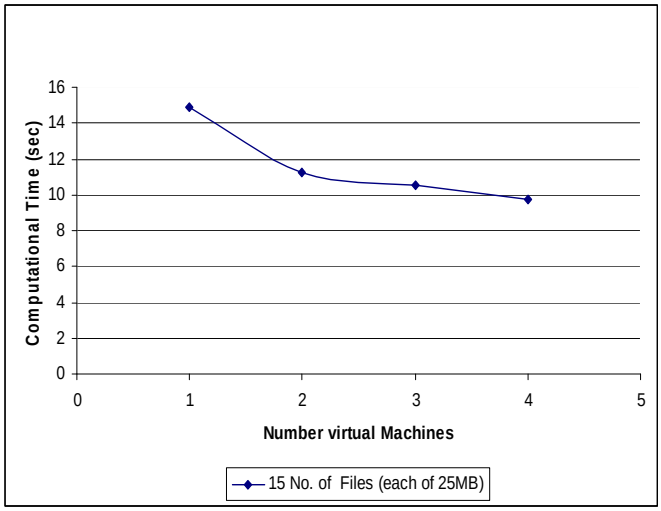

Fig. 4: Cloud Performance by varying Number of Virtual Machines

Experiment 2 (Varying number of text Files): In this experiment our objective is to study the impact of file size on cloud performance with varying number of text files. Here we considered file size of 25 MB with varying No. of files as (i) 15 No. (ii) 20 No. and (iii) 30 No. and executed on the varying number of virtual Machines. From the Fig. 5, it is clear that the cloud Environment performance is enhanced by increasing the number of number of virtual machines for all the above considered number files.

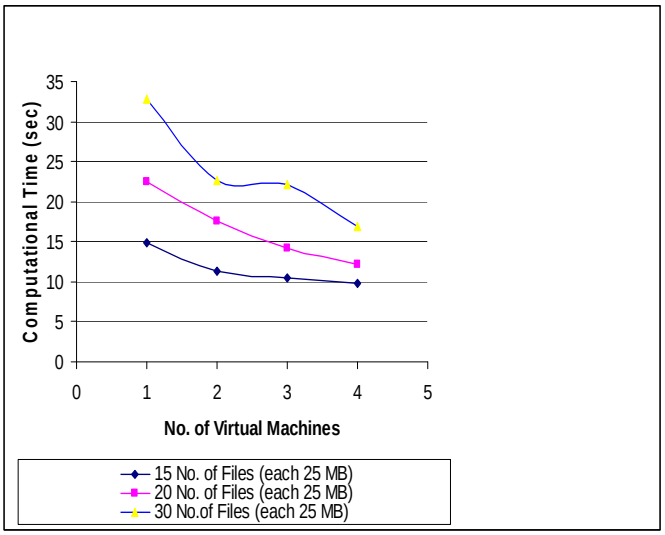

Fig. 5: Cloud Performance by Varying Number of Text Files

5. CONCLUSIONS

In this paper, we employed Platform as a Service approach of Aneka Frame work on the virtualized environment of Cloud and implemented Distributed application. Further the Aneka Frame Work uses Thread Model to implement to distribute threads on multiple virtual Machines available in cloud environment. From our analysis, the performance of virtualized cloud environment improves with increased number of virtual Machines and also shows computational superiority of cloud-based method theoretically as compared to traditional method i.e on a single Machine. In future, we would like to use Map Reduce/ Task Model of Aneka Frame work to implement Distributed Application on cloud Environment.

REFERENCES

[1] RajKumar Buyya, James Broberg, Andrzej M.Goscinki, "CloudComputing: principles and paradigms",ISBN:p78-0-470-88799-8, Feb 2011

[2] Yongwei Wu, Kai Hwang, and Rajkumar Buyya, Kai Hwang, Geoffrey Fox, and Jack Dongarra ISBN: 978-0-12-385880-1, Morgan Kaufmann, San Francisco, USA, October 2011.

[3] Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, Mastering Cloud Computing, Tata McGraw Hill, ISBN-13: 978-1-25-902995-0, New Delhi, India, Feb 2013.

[4] Christian Vecchiola, Rodrigo N. Calheiros, Dileban Karunamoorthy, Rajkumar Buyya, "Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka" Future Generation Computer Systems, Volume 28, Issue 1, January 2012, Pages 58- 65.

[5] Rodrigo N. Calheiros, Christian Vecchiola, Dileban Karunamoorthy, Rajkumar Buyya, "The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds" , Future Generation Computer Systems, Volume 28, Issue 6, June 2012, Pages 861-870.

[6] Sunilkumar S. Manvi, Gopal Krishna Shyam, "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey", *Journal of Network and Computer Applications, Volume 41, May 2014, Pages 424-440.*

[7] Giuseppe Aceto, Alessio Botta, Walter de Donato, Antonio Pescapè, "Cloud Monitoring: A survey", *Computer Networks, Volume 57, Issue 9, 19 June 2013, Pages 2093-2115*

[8] J. Octavio Gutierrez-Garcia, Kwang Mong Sim, "Agent-based Cloud bag-of-tasks execution", *Journal of Systems and Software, Volume 104, June 2015, Pages 17-31*

[9] http://www.manjrasoft.com/products.html

[10] http://www.manjrasoft.com/flyers/Brochure%20-%20Aneka.pdf

[11] http://www.manjrasoft.com/download/2.0/AnekaInstallationGuide.pdf

[12] http://www.manjrasoft.com/download/3.0/ThreadModel.pdf

[13] R. Buyya, C.S. Yeo, S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities" , in: Proc. 10th IEEE Int. Conference on High Performance Computing and Communications, HPCC 2008, Dalian, China, Sept. 2008.

[14] Kovvur Ram Mohan Rao, S Ramachandram, Kadappa VijayaKumar and A Govardhan , A Reliable Distributed Grid Scheduler for Independent Tasks, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[15] Ram Mohan Rao Kovvur, S. Ramachandram, Vijayakumar Kadappa, A. Govardhan, A Reliable Distributed Grid Scheduler for Mixed Tasks, PDCTA 2011, CCIS 203, pp. 213 -233, 2011.

[16] Gaurav Kumar And Amit Doegar. "Article :Research Areas and simulation in Cloud Computing ", Open Source For You, pp 91-93, Feb 2015