ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 3, Special Issue 3, April 2014*

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE ’14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University, Paiyanoor-603 104, Tamil Nadu, India

Executing Diverse Workloads in Clouds with Virtual Infrastructure Gateway

J. Jayaprabha, J.Selvakumar

M.Tech (CSE), Department of CSE, SRM University, Kattankulathur, Tamilnadu, India

Asst. Professor(S.G), Department of ECE, SRM University, Kattankulathur,Tamilnadu, India

ABSTRACT: Infrastructure-as-a-Service clouds offer entire virtual infrastructures for distributed processing while concealing all physical underlying machinery. Current cloud interface abstractions restrict users from providing information regarding usage patterns of their requested virtual machines (VMs). In this paper, we propose Nefeli, a virtual infrastructure gateway that lifts this restriction. Through Nefeli, cloud consumers provide deployment hints on the possible mapping of VMs to physical nodes. Such hints include the collocation and anticollocation of VMs, the existence of potential performance bottlenecks, the presence of underlying hardware features (e.g., high availability), the proximity of certain VMs to data repositories, or any other information that would contribute in a more effective placement of VMs to physical hosting nodes. The set of consumer-provided hints is augmented with high level placement policies specified by the cloud administration. Placement policies and hints form a constraint satisfaction problem that when solved, yields the final VM-to-host placement. As workloads executed by the cloud may change over time, VM-to-host mappings must follow suit. Using our prototype, we examine overheads involved and show significant improvements in terms of time needed to execute scientific and real application workloads. We also demonstrate how power-aware policies may reduce the energy consumption of the physical installation. Finally, we compare Nefeli’s placement choices with those attained by the opensource cloud middleware, OpenNebula.

Keywords : OpenNebula, Virtual machine, Cloud Computing, Nefeli design, IaaS Cloud.

1. INTRODUCTION

1.1. General Description

In this paper, we present the design, implementation, and evaluation of a cloud gateway, Nefeli. Nefeli performs intelligent placement of VMs onto physical nodes by exploiting user-provided deployment hints. Hints realize placement preferences based on knowledge only the cloud consumer has regarding the intended usage of the requested VMs [1]. By modeling workloads as patterns of data flows, computations, control/synchronization points, and necessary network connections, users can identify favorable VM layouts. These layouts translate to deployment hints. Such hints articulate (i) resource consumption patterns among VMs; (ii) VMs that may become a performance bottleneck; and (iii) portions of the requested virtual infrastructure that can be assisted by the existence of special hardware support. For instance, the fact that two VMs in a virtual infrastructure will hold mirrors of a database is only known to the cloud consumer. This information should be communicated to the cloud as a deployment hint so that the respective VMs will not be deployed on the same host. We refer to VM layout patterns as task-flows to distinguish them from the traditional workflow concept. Specifically, task-flows illustrate “ideal” deployments of VMs described by the cloud consumers using deployment hints. Nefeli exploits these hints so as to (re)deploy VMs in the cloud and achieve efficient task-flow execution.

1.2 Objective of the Cloud Design

The main contribution of our approach is that we present a complete solution in extracting and exploiting the knowledge cloud consumers posses regarding the operational aspects of their virtual infrastructures [3]. Our approach is compatible with the cloud abstractions that dictate users are kept agnostic of the physical infrastructure properties at all times. Furthermore, our approach is able to adapt to dynamic environments where both task-flows and user preferences change over time. Nefeli produces suitable VM to physical node mappings in response to signals coming from the infrastructures (both physical and virtual) or any other external notification mechanism. The produced mappings are applied through appropriate In this paper consistently display significant performance improvements when compared to the aforementioned policies. In video transcoding, Nefeli achieves 17 percent reduced processing times compared to the VM placement decided by Open Nebula. In scientific task flows and for a variety of simulated clouds, Nefeli demonstrates significantly higher throughput rates compared to other VM placement policies. Noteworthy savings in terms of power consumption are reported as well. We also present the performance overheads involved in the operation of Nefeli as the cloud infrastructure scales out.

1.3 Existing Cloud System Design

This proposes an approach to enforce the information flow policies at Infrastructure-as-a-Service (IaaS) layer in a cloud computing environment. Especially, this adopt a single task flow execution. We implement a proof-of-concept prototype system based on Eucalyptus open source packages to show the feasibility of our approach. This system facilitates the cloud management modules to resolve the conflict-of interest issues for service providers in clouds. Even though it enables us to dynamically provide servers with the ability to address a wide range of needs, this paradigm brings forth many new challenges for the data security and access control as users outsource their sensitive data to clouds, which are beyond the same trusted domain as data owners. A fundamental problem is the existence of insecure information flows due to the fact that a service provider can access multiple virtual machines in clouds. Sensitive information may be leaked to unauthorized customers and such critical information flows could raise *conflict-of-interest* issues in cloud computing.

2. PROPOSED CLOUDS WITH VIRTUAL INFRASTRUCTURE SYSTEM

The key benefit in using an IaaS-cloud is that it shields users and/or applications from all administrative tasks and resource sharing policies of the underlying machinery. Moreover, the decoupling of physical resources from system software offers enhanced server utilization through collocation of VMs and effective options for node recovery in light of failure(s). However, sharing physical resources may yield peak performance rates that are below expectation due to VM contention on particular physical nodes. Placement policies and hints form a constraint satisfaction problem that when solved, yields the final VM-to-host placement. As workloads executed by the cloud may change over time, VM-to-host mappings must follow suit. To this end, Nefeli captures such events, changes VM deployment, helps avoid bottlenecks, and ultimately, improves the quality of the rendered services. Using our prototype, it examine overheads involved and show significant improvements in terms of time needed to execute scientific and real application workloads. This also demonstrate how power-aware policies may reduce the energy consumption of the physical installation.

In this section, we present the design, implementation, and evaluation of a cloud gateway, Nefeli. Hints realize placement preferences based on knowledge only the cloud consumer has regarding the intended usage of the requested VMs. By modeling workloads as patterns of data flows, computations, control/synchronization points, and necessary network connections, users can identify favorable VM layouts.

2.1 MAIN MODULES IN PROPOSED SYSTEM DESIGN:

The proposed system design consists of the following modules as Cloud Consumer Registration, Cloud Consumer task, Cloud Administration task, Virtual Machine Scheduling [8], [11] and Nefeli Interaction Model. The detailed description is as follows

2.1.1 Cloud Consumer Registration

This is the first module of our paper. The important role for the cloud user is to move login window to cloud user window. This module has created for the security purpose. In this login page we have to enter login user id and password. It will check username and password is match or not (valid user id and valid password). If we enter any invalid username or password

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 3, Special Issue 3, April 2014

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University, Paiyanoor-603 104, Tamil Nadu, India

we can't enter into login window to user window it will shows error message[6].

2.1.2 Cloud Consumer task

This is the second module of our paper in this Constraints express user (cloud consumer).Each constraint is realized as a utility function F that evaluates a single deployment profile [7]. F takes as input a deployment profile and returns the degree of constraint satisfaction in the range of (0, 1).

2.1.3 Cloud Administration task

Administrative constraints are also realized as utility functions. These constraints serve a dual purpose as they can introduce high-level policies and assist in administration tasks. For instance, Empty Node relieves a hosting node of VMs. Reduce Dist enforces the high-level policy of clustering VMs of the same user on hosts that may not be far apart in terms of network hops; this is done to limit major traffic to be routed over long-haul physical networks.

2.1.4 Virtual Machine Scheduling

In this module we refer to VM layout patterns as task-flows to distinguish them from the traditional workflow concept. Specifically, task-flows illustrate "ideal" deployments of VMs described by the cloud consumers using deployment hints. Nefeli exploits these hints so as to (re)deploy VMs in the cloud and achieve efficient task-flow execution.

2.1.5 Nefeli's Interaction Model

This is the final module of the paper here we specified Nefeli, it will may interact with the physical infrastructure through a cloud middleware. However, the cloud middleware may not provide all the functionality required by Nefeli. For instance, OpenNebula [2] does not expose all host-related information it gathers.

2.1.6 Module Diagram

The Fig. 1-5 represents the various Modules utilized in the proposed cloud design for IaaS architecture

(i) Cloud Consumer Registration.

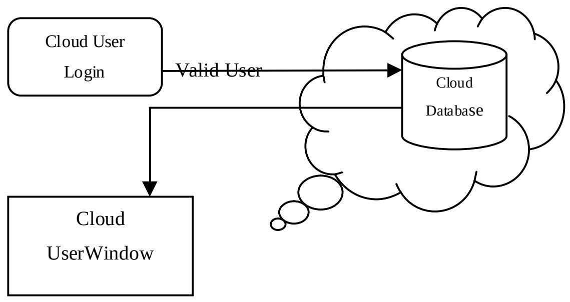


Fig. 1 Cloud Consumer Registration Block

(ii) Cloud Consumer Task.

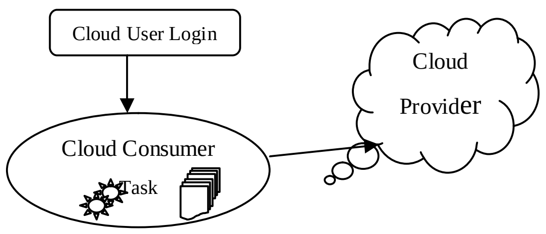


Fig. 2. Block Diagram of Cloud Consumer Task

(iii) Cloud Administration Task.

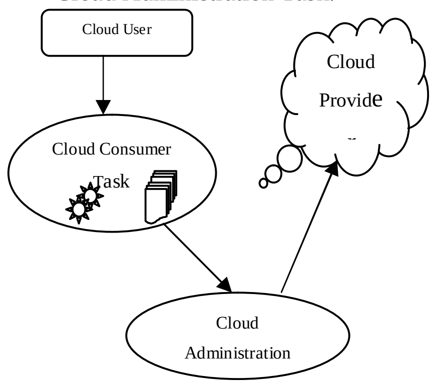


Fig. 3. Block Representation of Cloud Administration Task

(iv) Virtual Machine Scheduling

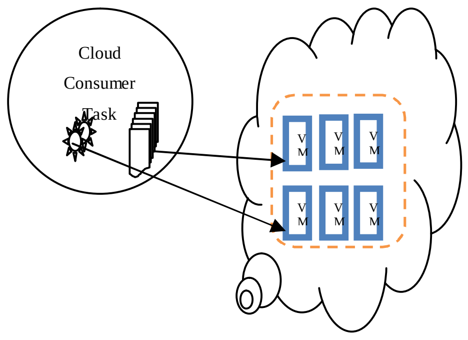


Fig. 4. VM Scheduling

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 3, Special Issue 3, April 2014

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

(v) Nefeli's Interaction Model.

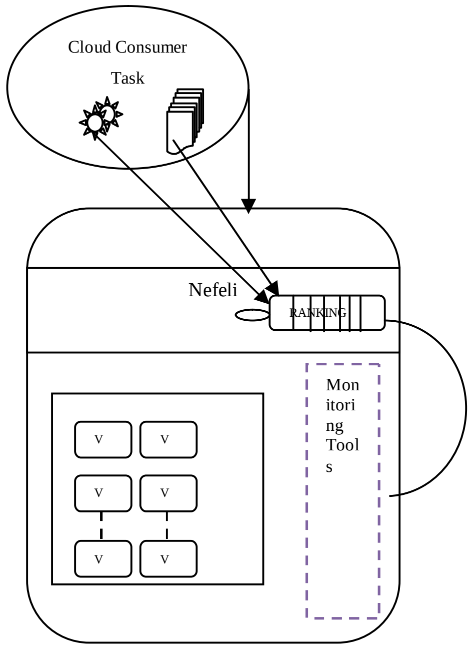


Fig. 5 Basic Nefeli's Interaction Model

2.1.6 Input Expected Output :

➢ Cloud Consumer Registration.
Input : Cloud User Login name and Password
Output : If Valid cloud user Open the user window otherwise error page.
➢ Cloud Consumer Task
Input : After login valid cloud user selecting constrains.
Output : Send to Cloud Data base Successfully.
➢ Cloud Administration Task
Input : Cloud Administration constrains choosing
Output : Admin take the cloud user constraints.
➢ Virtual Machine Scheduling [4]
Input : Based on cloud user constraints VM monitoring
Output : Scheduling cloud user constraints
➢ Nefeli's Interaction Model.
Input : Cloud user constraints send to calculate rank.
Output : Allocate the task to VM.

The software used for the modeling the design for the above input output required in the front end is JSP, J2EE (SERVLETS), Back End tool used is MY SQL5.5, Web Server is APACHE TOMCAT SERVER 6.0, BROWSER used is INTERNET EXPLORER, IDE used is NETBEANS 6.9/ECLIPSE [5].

3. DESIGN ENGINEERING FOR IaaS CLOUDS

Design Engineering deals with the various UML (Unified Modeling language) diagrams for the implementation of paper. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

3.1 Class Diagram of Proposed Cloud Design

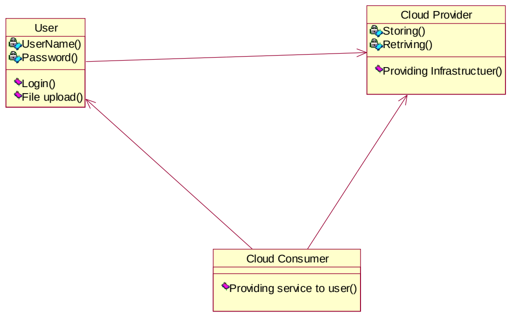


Fig. 6 Basic Class Diagram Representation

The class diagram in Fig. 6 is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. In class diagram took the cloud user, cloud provider and consumer. In cloud user we took the user login and user constraints. Now cloud provider contained service details like storing the user files and allocating task to VM. Finally cloud consumer may getting the service from the cloud.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 3, Special Issue 3, April 2014

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University, Paiyanoor-603 104, Tamil Nadu, India

3.1.2 Object Diagram of the Proposed Cloud System

Object diagram shown in Fig. 7 represents the flow of objects how the process is running. In the above digram tells about the flow of objects betwwen the classes.The main object of this diagram is cloud user login his window and send the his constraints to cloud. Cloud user constraints took by the cloud provider it having the nefeli based on this allocating the task to VM. Finally cloud user files stord to properly.

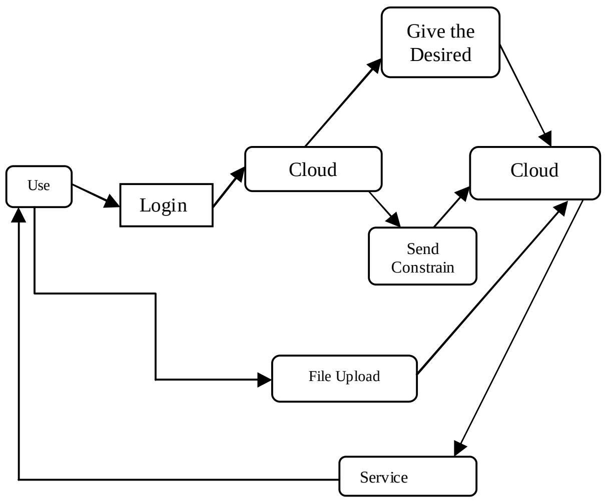


Fig . 7 Object Case Diagram of Proposed System

4.SYSTEM ARCHITECTURE

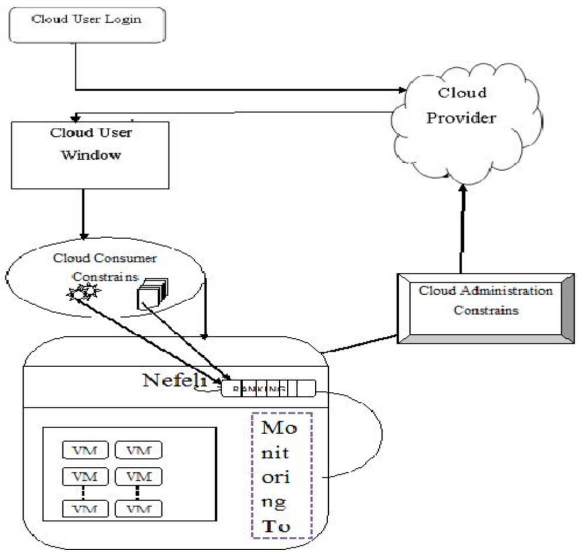


Fig. 8 Basic System Architecture

The systems architect establishes the basic structure of the system, this we know about that when a node is viewing or downloading a multimedia segment from the server or another peer how sharing it paper here we specified Nefeli, it will may interact with the physical infrastructure through a cloud middleware. However, the cloud middleware may not provide all the functionality required by Nefeli. For instance, OpenNebula does not expose all host-related information it gathers. In such cases, we have to realize any missing functionality and incorporate it in the "cloud middleware connector" component (denoted as "extra functionalities"). Nefeli plays a major role in helping attain user-favorable VM deployments. The user remains unaware of the cloud internals as any piece of his information arriving at Nefeli (the cloud gateway) strictly refers to the type of the workload(s) the virtual infrastructure is to serve. Nefeli has the role of an IaaS-cloud gateway.

4.2 PROPOSED ALGORITHM FOR HIGHER PERFORMANCE

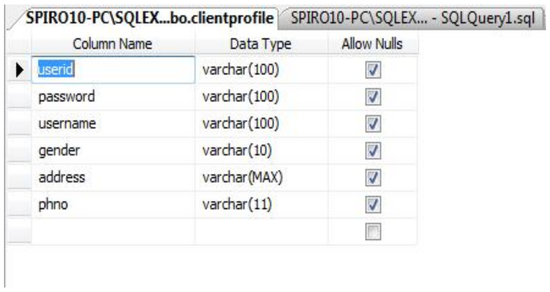
The basic algorithmic technique used for implementation for the proposed design architecture is the workflow tailored BOSS algorithm [11] and its associated RankOrder() function as shown below in Fig. 10.

Performance Algorithm: Workflow-Tail BOSS algorithm

```
Input : The workflow application DAG([n],[e])
Output: The Schedule of DAG on commercial multi-Cloud[m]
begin
    tasks = [n];
    ranks = Ø;
    Task _temp= exit
while tasks ≠ θ do
            Add (ranks, RankOrder(Task_temp));
            Task_temp = Tail(tasks);
            Tasks list to Task_temp
            Delete (tasks,Task_temp);
end
Tasks =[n];
Sort (tasks,ranks);
While tasks ≠ θ do
            Auctioneer =Head(tasks);
            w = BOSS (auctioneer);
            Submit (auctioneer,w);
            Delete (tasks,auctioneer);
end
Pay();
end
```

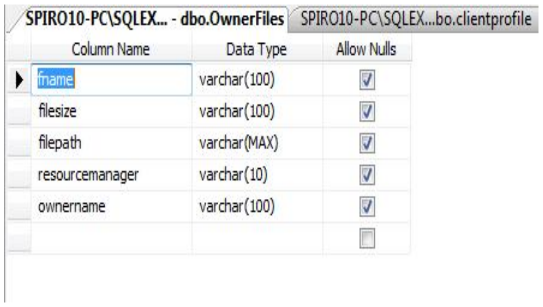
4.2.1 Database System Design

Table 1:Client Profile


SPIRO10-PC\SQLEX...bo.clientprofile | SPIRO10-PC\SQLEX... - SQLQuery1.sql

Column Name	Data Type	Allow Nulls
userid	varchar(100)	✓
password	varchar(100)	✓
username	varchar(100)	✓
gender	varchar(10)	✓
address	varchar(MAX)	✓
phno	varchar(11)	✓
		☐

Table .2 – Owner Database File


SPIRO10-PC\SQLEX... - dbo.OwnerFiles | SPIRO10-PC\SQLEX...bo.clientprofile

Column Name	Data Type	Allow Nulls
fname	varchar(100)	✓
filesize	varchar(100)	✓
filepath	varchar(MAX)	✓
resourcemanager	varchar(10)	✓
ownername	varchar(100)	✓
		☐

Fig. 10 Design Entry using sofware

The above design explains each and every diagram of table designs. Table 1 is used for login and registration. Table 2 contains username and password of admin .

4.1.2 Implementation of the Proposed Cloud System Design

In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard [9] [10]. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages.

5.RESULTS AND DISCUSSIONS

The Fig. 11-15 depicts the actual snapshots of the proposed Cloud Design for VM Deployment in the IaaS architecture. Snapshot is nothing but every moment of the application while running. It gives the clear elaborated of application. It will be useful for the new user to understand for the future steps.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 3, Special Issue 3, April 2014

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University, Paiyanoor-603 104, Tamil Nadu, India

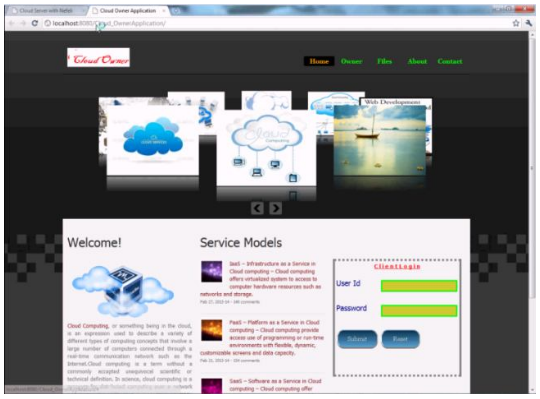


Fig.11 Snapshot specifies the Cloud Client Home Page

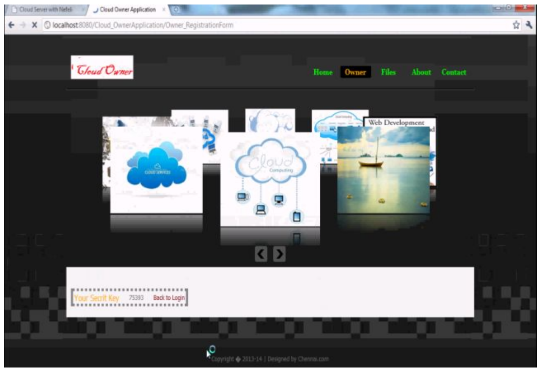


Fig. 12 Snapshot specifies Cloud Owner security key, after registration

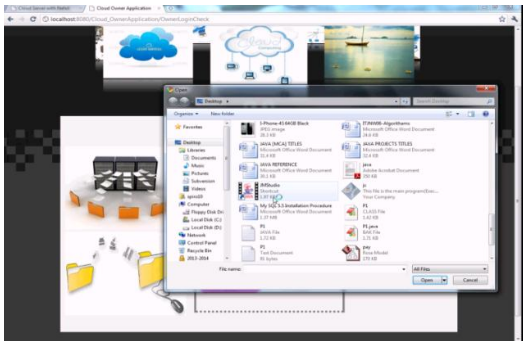


Fig. 13 Basic Snapshot specifies file uploading process.

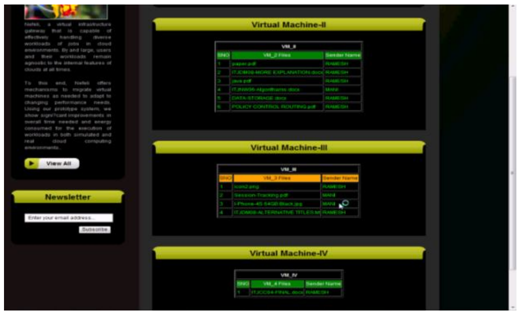


Fig. 14 Cloud Server contained file list in different Virtual Machine

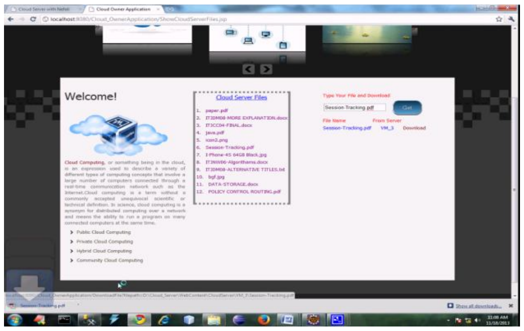


Fig. 15 Snapshot specifies cloud file downloading

6.CONCLUSION AND FUTURE WORK

In this paper we present Nefeli, a hint-based VM scheduler that serves as a gateway to IaaS- louds. Users are aware of the flow of tasks executed in their virtual infrastructures and the role each VM plays. This information is passed to the cloud provider, as hints, and helps drive the placement of VMs to hosts. Hints are also employed by the cloud administration to express its own deployment preferences. Nefeli combines consumer and administrative hints to handle peak performance, address performance bottlenecks, and effectively implement high-level cloud policies such as load balancing and energy savings. An event-based mechanism allows Nefeli to reschedule VMs to adjust to changes in the workloads served. Our approach is aligned with the separation of concerns IaaS-clouds introduce as the users remain unaware of the physical cloud structure and the properties of the VM hosting nodes. Our evaluation, using simulated and real private Iaas-cloud environments, shows significant gains for Nefeli both in terms of performance and power consumption.

In the future, we plan to: (i) investigate alternative constraint satisfaction approaches to address

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization — Vol. 3, Special Issue 3, April 2014

International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,

Paiyanoor-603 104, Tamil Nadu, India

scalability issues present in large infrastructures; (ii) offer deployment hints that will effectively handle the deployment of virtual infrastructures in the context of real large cloud installations; (iii) extend the support of Nefeli to other cloud middleware platforms by providing additional cloud middleware connectors.

REFERENCES

[1] M. Rosenblum and T. Garfinkel, "Virtual Machine Monitors: Current Technology and Future Trends," Computer, vol. 38, no. 5, pp. 39-47, May 2005.
[2] "OpenNebula," http://www.opennebula.org, May 2011.
[3] Amazon, "Elastic Cloud," http://aws.amazon.com/ec2/, 2009.
[4] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-Source Cloud-Computing System," Proc. IEEE/ACM Ninth Int'l Symp. Cluster Computing and the Grid (CCGRID), pp. 124-131, May 2009.
[5] "OpenStack," http://www.openstack.org/, Feb. 2011.
[6] H.N. Van, F.D. Tran, and J.-M. Menaud, "Autonomic Virtual Resource Management for Service Hosting Platforms," Proceedings of ICSE Workshop Software Eng. Challenges of Cloud Computing, pp. 1-8, 2009.
[7] X. Wang, D. Lan, G. Wang, X. Fang, M. Ye, Y. Chen, and Q.Q.B. Wang, "Appliance-Based Autonomic Provisioning Framework for Virtualized Outsourcing Data Center," Proc. Fourth Int'l Conf. Autonomic Computing, pp. 29, 2007.
[8] D. Jayasinghe, C. Pu, T. Eilam, M. Steinder, I. Whalley, and E. Snible, "Improving Performance and Availability of Services Hosted on IaaS Clouds with Structural Constraint-Aware Virtual Machine Placement," Proc. IEEE Int'l Conf. Services Computing (SCC), July 2011.
[9] G. Jung, K.R. Joshi, M.A. Hiltunen, R.D. Schlichting, and C. Pu, "Performance and Availability Aware Regeneration For Cloud Based Multitier Applications," Proc. IEEE/IFIP Int'l Conf. Dependable Systems and Networks (DSN), June 2010.
[10] P. deGrandis and G. Valetto, "Elicitation and Utilization of Application-level Utility Functions," Proc. Sixth Int'l Conf. Autonomic Computing, 2009.
[11] H.M.Fard, R.Prodan, and T.Fahringer, "A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments" , IEEE Transactions on Parallel & Distributed Systems, vol.24, no.6, June.2013, pp. 1203 - 1209