# An Accounting Solution for the Open-Source Eucalyptus Cloud Computing Framework

Rodolfo N. Duldulao, Jr.
Institute of Computer Science
University of the Philippines Los Baños
rnduldulaojr@gmail.com

Chryss Ann A. Belaguin
Institute of Computer Science
University of the Philippines Los Baños
caabelaguin@gmail.com

Regelyn T. Bañacia
Institute of Computer Science
University of the Philippines Los Baños
regelyn.banacia12@gmail.com

Joseph Anthony C. Hermocilla
Institute of Computer Science
University of the Philippines Los Baños
jachermocilla@uplb.edu.ph

## ABSTRACT

This research presents an accounting facility which is currently lacking from the open-source Eucalyptus cloud computing framework. The main contribution of this work is a command line tool for billing and web interface for data viewing. Utilization cost was computed based on the extracted information from virtual machine instances in the cloud. The cost rate applied was based on the pricing scheme of Amazon Elastic Compute Cloud(Amazon EC2).

## Keywords

Cloud computing, Eucalyptus, Accounting, Private cloud, Amazon EC2

## 1. INTRODUCTION

Cloud computing is defined as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". [12] It is considered an IT progression from client-server computing, wherein hardware and human resource expenses are highly needed for the deployment of an IT infrastructure.

There are three service models of cloud computing: Infrastructure-as-a-Service (IaaS), Software-as-a-Service (SaaS), and Platform-as-a-Service (PaaS). IaaS is a service where the user can transfer existing programs and data into the cloud in the form of a cloud server. In IaaS, the user has control over the cloud server's contents and some configuration. SaaS is a service wherein computer applications residing on a cloud infrastructure are accessed through a thin client interface such as a web browser. In SaaS, the user does not actually develop the applications being used. PaaS is a service where the user obtains access to a platform residing on the cloud infrastructure which can be used to develop and deliver applications. [12]

IaaS can be deployed as a private cloud. A private cloud consists of internal data centers of a business or other organization that are not made available to the public. [1] However, even if the cloud is not publicly available, it can be optionally commercialized within the organization using a utility business model.

Private clouds can be built by anyone. In building a private cloud environment, various tools are available. Some of these tools are OpenNebula, Nimbus, AbiCloud, vSphere and Eucalyptus. OpenNebula is an open-source framework designed to manage virtualization in distributed environment. Nimbus offers self-configuring virtual cluster support. AbiCloud is another open-source framework which supports virtual infrastructure management tools such as VirtualBox, Xen, KVM and VMware. vSphere is a commercial software for building a private cloud. Eucalyptus, which stands for "Elastic Utility Computing Architecture Linking Your Programs To Useful Systems", is a web services-based implementation of cloud computing which is available in open-source and enterprise versions.

Among the tools mentioned, Eucalyptus is one of the better frameworks because it is implemented using commonly-available Linux tools and basic web service technology, making it easy to install and maintain. Moreover, it provides an interface that is compatible with the Amazon Elastic Cloud Compute (Amazon EC2), a popular IaaS developed by Amazon which allows clients to use web service interfaces to launch virtual server instances in different operating systems. [3] It also has an open-source version which is free of charge and open to the community for improvement.

## 2. PROBLEM STATEMENT

Cloud providers usually set cost rates for their services. However, in the open-source version of Eucalyptus, the abil-

ity to audit client usage is still unavailable. Thus, an accounting system will be added to compute the cost of each instance in the cloud. The price will be computed based on the extracted information on instances and usage. Extracted information on instances and usage includes the *instance ID*, *launch time*, *type*, *owner*, and *total running time*. By knowing these pieces of information, a cost formula can be applied to generate the overall expense of each client.

Administrators can view the clients' bills through the console and web interface while the clients can view their service cost through the web interface only.

Through this system, cloud administrators running an open-source version of Eucalyptus who want to commercialize their cloud can monitor the cost of service availed by each user. Then, the clients can acquire the accounting information they need through the system's web interface.

Moreover, the Eucalyptus team is open for contributions to improve the Eucalyptus framework. [4] Thus, the accounting system, when submitted to the team, can be incorporated in the next release of Eucalyptus.

# 3. METHODOLOGY

## 3.1 Functional Requirements

The facility should enable the administrators to audit and view the cloud consumption and the users to view their consumption of instances and resources and how much it costs them. The administrator should be able to audit the accounts via the terminal through the command-line tool *euca-audit-instances*. The administrator will also be able to view the accounts via the terminal upon running *euca-audit-instances*. The administrator and the clients will also be provided with a web user interface to view their accountabilities using the web browser. The administrator can view all of the clients' accounts while the clients can only view theirs.

## 3.2 System Requirements

- At least two (2) units of machine (either a desktop or a server)
- Linux-based Operating System (CentOS 5.5 Final)
- Java OpenJDK 1.6.0
- Eucalyptus 2.0 packages
- Hypervisor (KVM/Xen/VMWare) - virtual machine monitor
- Euca2ools 1.3.1
- Euca2ools dependencies: Python 2.5 (dev), Boto 1.9b, M2Crypto 0.20.1
- virt-top 1.16

## 3.3 Private Cloud Setup

The setup was a single cluster installation using two machine units. One unit served as the *front-end* machine where all of the Eucalyptus components were installed, except the node controller (which is the server of the instances). The second unit served as the *node* machine where the node controller was installed. Both *front-end* and the *node* machines were connected to a Local Area Network (LAN). For the framework to function, a Linux-based system was installed on the machine. In this study, the researchers used CentOS 5.5 Final as the operating system.

Java OpenJDK 1.6.0, Eucalyptus 2.0 (Cloud Controller, Cluster Controller and Storage Controller packages), and Euca2ools 1.3.1, as well as its dependencies, were installed on the *front-end* machine. Xen hypervisor and Eucalyptus 2.0 (node controller package) were installed on the *node* machine.

## 3.4 Accounting System

### 3.4.1 Data Extraction

Necessary data such as *instance owner*, *instance ID*, *instance type*, and *launch time* of instance was extracted from each of the created instances. The data was collected through a shell script which parses the standard output of *euca-describe-instances*, a command-line tool for viewing instance data, as admin. However, the instance data was not persistent in the database since Eucalyptus uses an in-memory HyperSQL database (HSQLDB). Thus, when an instance is terminated, all its data will be erased from the database. To prevent this, a Comma-Separated Value (CSV) file is maintained to store all the instances data.

To get the total running time of an instance, a tool called *virt-top*, a top-like utility for showing stats of virtualized domains, was used. [11] It has a built-in functionality for writing virtualization statistics into a CSV file. However, the CSV file is overwritten once *virt-top* is executed again. Again to prevent this, a shell script was created to maintain another CSV file containing both previous and current data.

The scripts were added to *crontab*, a program which "allows scheduling of system-wide tasks" [2], so that they would act as polling daemons.

### 3.4.2 Cost Computations

A command-line tool written in Python, *euca-audit-instances*, was added to Euca2ools to audit the extracted instance data. Upon executing the command, a cost formula is applied on the extracted data and the cost of each instance is computed. The cost of an instance is calculated as $Cost_{user} = Price \times t$, where $t$ is the total running time of the task in hours and *Price* is the price per virtual machine hour. [13] This formula was based on the pricing scheme of Amazon.

The instance rate applied was based on Amazon EC2's pricing. Table 1 shows the pricing scheme.

| Instance type | Price per virtual machine hour |
|---|---|
| m1.small | $0.085 |
| c1.medium | $0.17 |
| m1.large | $0.34 |
| m1.xlarge | $0.50 |
| c1.xlarge | $0.68 |

**Table 1. Amazon EC2's pricing scheme.**

## 3.5 Terminal Interface

To view the accounts of the users through terminal or console, *euca-audit-instances* can also be used. Upon running the tool, a summary report containing details on all users' accountabilities and details on previous and current instances is printed on the terminal.

## 3.6 Web Interface

A *Reports* tab was added in the Web Interface of Eucalyptus (*https://<frontend-ip-address>:8443*) where the users can view their instance counts, durations, and overall cost. Under this tab, the administrator can view all the clients' information while the clients can only view theirs. Google Web Toolkit (GWT) Web UI was used for the interface and file reading in Java was used to gather the data.

## 3.7 Unit Testing

Functions in *euca-audit-instances* were tested via Python's unit testing system, *unittest*. Each function should give known results with known inputs.

# 4. RESULTS AND DISCUSSION

## 4.1 Data Extraction

The script successfully parsed the output of *euca-describe-instances* and updated the CSV file. The total running time of each instances was also computed accurately and was integrated in the file. Previous instances' data was also preserved. Figure 1 shows the contents of the file after a successful run.

```
i-3B4106A9,admin,m1.small,0:11:47,2011-03-08T09:11:21.356Z
i-43190839,chryss,m1.small,0:07:05,2011-03-08T09:16:12.479Z
i-3900070B,regelyn,m1.small,0:14:10,2011-03-08T09:29:57.828Z
i-34BE0774,admin,m1.small,0:02:36,2011-03-08T10:04:25.405Z
i-354706EF,admin,m1.small,0:09:59,2011-03-09T04:34:23.083Z
i-5E380ABE,regelyn,c1.medium,0:02:03,2011-03-09T04:50:24.987Z
i-45D70863,chryss,m1.small,1:45:47,2011-03-09T05:01:41.473Z
i-44570741,regelyn,m1.small,1:46:06,2011-03-09T05:02:30.265Z
i-28960604,admin,m1.small,0:41:59,2011-03-09T06:56:23.86Z
i-4DBE0992,admin,c1.medium,0:38:20,2011-03-10T02:38:50.928Z
i-4037082F,admin,c1.medium,1:49:13,2011-03-10T08:15:58.77Z
~
~
~
~
~
~
~
~
~
~
~
"log/idata.csv" [noeol] 11L, 657C                    1,1           All
```

**Figure 1: Instance data CSV file.**

## 4.2 Cost Computation

Executing the newly added command-line tool *euca-audit-instances* successfully computes the accountabilities of all users. Figure 2 shows the contents of the file where details on the accountabilities were stored.

## 4.3 Terminal Interface

Upon executing *euca-audit-instances*, the terminal displayed all the instances' data in detail, as well as a summary report containing the instance counts, duration and overall cost per user. Figure 3 shows the sample output.

## 4.4 Web Interface

A *Reports* tab was added in the Eucalyptus Web UI. Necessary data were gathered from the CSV file where account

```
admin,4,1.1058,2,2.4733,0,0,0,0,0,0.5145
chryss,2,1.8812,0,0,0,0,0,0,0,0.1599
regelyn,2,2.0044,1,0.0342,0,0,0,0,0,0.1762
~
~
~
~
~
~
~
~
~
~
~
~
<alyptus/cloud.d/report.csv" [dos] 3L, 130C 1,1           All
```

**Figure 2: Account reports CSV file.**

```
[root@admindesktop .euca]# euca-audit-instances
List of all instances (sorted by launch date):
i-3B4106A9,admin,m1.small,0:11:47,2011-03-08T09:11:21.356Z
i-43190839,chryss,m1.small,0:07:05,2011-03-08T09:16:12.479Z
i-3900070B,regelyn,m1.small,0:14:10,2011-03-08T09:29:57.828Z
i-34BE0774,admin,m1.small,0:02:36,2011-03-08T10:04:25.405Z
i-354706EF,admin,m1.small,0:09:59,2011-03-09T04:34:23.083Z
i-5E380ABE,regelyn,c1.medium,0:02:03,2011-03-09T04:50:24.987Z
i-45D70863,chryss,m1.small,1:45:47,2011-03-09T05:01:41.473Z
i-44570741,regelyn,m1.small,1:46:06,2011-03-09T05:02:30.265Z
i-28960604,admin,m1.small,0:41:59,2011-03-09T06:56:23.86Z
i-4DBE0992,admin,c1.medium,0:38:20,2011-03-10T02:38:50.928Z
i-4037082F,admin,c1.medium,1:50:04,2011-03-10T08:15:58.77Z
--------------------------------------------------------------------------------------------------------------
SUMMARY
--------------------------------------------------------------------------------------------------------------
User    | m1.small | Hours | c1.medium | Hours | m1.large | Hours | m1.xlarge | Hours | c1.xlarge | Hours | Total Price
 admin  |    4     | 1.1058 |    2     | 2.4733 |    0     |   0   |    0      |   0   |    0      |   0   |  0.5145
 chryss |    2     | 1.8812 |    0     |   0    |    0     |   0   |    0      |   0   |    0      |   0   |  0.1599
regelyn |    2     | 2.0044 |    1     | 0.0342 |    0     |   0   |    0      |   0   |    0      |   0   |  0.1762
[root@admindesktop .euca]#
```

**Figure 3: *euca-audit-instances* terminal output.**

reports were maintained (see Figure 2). These data are displayed on the browser. When the logged in user is the administrator, the interface shows all the accounts and bills of all users. On the other hand, when the logged in user is a client, he/she can only view his/her own account. Figure 4 and 5 show the administrator's and user's *Reports* tab.

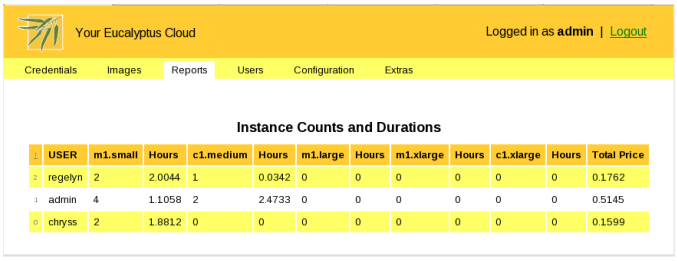


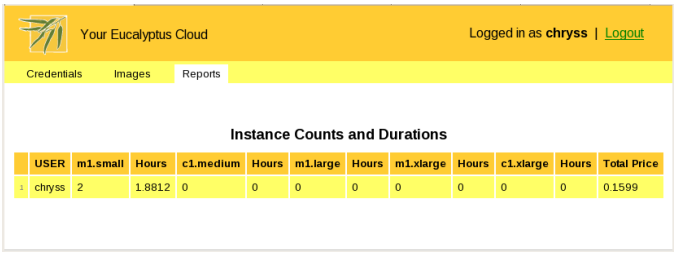

**Figure 4: Admin *Reports* tab.**

## 4.5 Unit Testing

Functions in *euca-audit-instances* were successfully tested. Figure 6 shows the output of the unit test.

# 5. CONCLUSION

The accounting facility presented worked succesfully with the open-source Eucalyptus 2.0, which by default, lacks the functionality. This research can help the private cloud adminstrators to utilize their clouds for commercialization without purchasing the enterprise version of Eucalyptus.




| | USER | m1.small | Hours | c1.medium | Hours | m1.large | Hours | m1.xlarge | Hours | c1.xlarge | Hours | Total Price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | chryss | 2 | 1.8812 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.1599 |

**Figure 5: User *Reports* tab.**

```
[root@admindesktop bin]# python test.py -v
check_instance_type should give known result with known input ... ok
compute_hours_dec should give known result with known input ... ok
get_total_running_time should give known result with known input ... ok
get_user_list should give known result with known input ... ok
populate_summary_list should give known result with known input ... ok
get_total_running_time should return 0 with empty string as input ... ok

----------------------------------------------------------------------
Ran 6 tests in 0.011s

OK
[root@admindesktop bin]#
```

**Figure 6: Unit test result.**

The accounting system, especially the process on harvesting instance data, heavily relies on file reading. Thus, the instance data logs are unsecure. To improve the data security, the researchers recommend to create a separate database for handling information. Moreover, network traffic data was not considered on the computation of costs. Thus, future researchers can add such data, which may be extracted through integrating Nagios and Ganglia in the system.

Improvements on the web interface may include a functionality for downloading a PDF containing the accountabilities, preferrably with more comprehensive accounting details such as graphs, statistics, etc. Moreover, future researchers may include a functionality wherein users can filter the displayed accounting details based on a specific range of date. For example, the users may only view his/her accountabilities from March 10, 2011 to April 10, 2011.

## 6. RELATED WORK

Very limited literature is available on cloud billing since cloud computing is a relatively new concept and its current services are still emerging as of today. Moreover, open-source frameworks like Eucalyptus only have the basic user management capabilities on its features while some enterprise versions have a billing capacity.

In 2005, Llorente and Montero developed OpenNebula, an open-source toolkit for building private, public and hybrid clouds. One of its features is user management, wherein it provides functionality for authentication framework, multiple cloud user and administration roles, quota management and secure multitenancy. [5]

In 2009, Keahey et al developed Nimbus, an open-source toolkit for transforming clusters into an IaaS. Nimbus has a feature called Per-client Usage Tracking wherein it makes possible to track the deployment time on a per-client basis. It also has another feature called Per-user Storage Quota wherein enforcement of per-user storage limits is possible through the VM image repository manager of Nimbus, Cumulus. [6]

In 2009, the Abiquo team developed AbiCloud, an open-source infrastracture software which can be used to create and manage private and public clouds. AbiCloud also has a user management feature which makes it possible to manage organizations, users, sessions and to implement basic hard and soft limits. [8]

In 2009, the VMware team developed vSphere, an enterprise software which is a platform for virtualization. One of its services is the vNetwork, which enables the administration and management of networking in virtual environments. [7]

In 2007, Wolski et al developed Eucalyptus, an open-source software infrastructure used to implement a private cloud from an information technology (IT) infrastructure. [10] In 2009, Wolski et al expanded its horizons by releasing the Eucalyptus Enterprise Edition (Eucalyptus EE), an extended version of the open-source Eucalyptus. Eucalyptus EE now supports quota management and accounting, complete with chargeback and billing platforms. [9]

Of the above mentioned tools, it is evident that the open-source tools support the basic user management needs of the client, but fail to provide an accounting feature, which is highly necessary for cloud commercialization. On the other hand, the commercially-available Eucalyptus EE already supports accounting, but it is not free and not open source.

## 7. ACKNOWLEDGMENTS

The authors would like thank the Institute of Computer Science for the equipment used in this research and the reviewers of this paper.

## 8. REFERENCES

[1] M. Armbrust et al. A View of Cloud Computing. *Communications of the ACM*, 53(4), Apr. 2010.

[2] CronHowTo - Community Ubuntu Documentation. `http://help.ubuntu.com/community/CronHowto`.

[3] Amazon Elastic Compute Cloud (Amazon EC2). `http://aws.amazon.com/ec2`.

[4] Developer's Corner | Eucalyptus Community. `http://open.eucalyptus.com/participate/contribute`.

[5] OpenNebula: The Open-Source Toolkit for Cloud Computing. `http://www.opennebula.org/documentation:features`.

[6] Documentation - Nimbus. `http://www.nimbusproject.org/docs/2.5/features.html`.

[7] VMware vSphere Features: Private Cloud Computing for Mid-size and Enterprise Businesses. `http://www.vmware.com/products/vsphere/mid-size-and-enterprise-business/features.html`.

[8] Abiquo 1.0 - Open Source Cloud Computing Management. `http://www.abicloud.org/display/abiCloud/Home`.

[9] Eucalyptus | Your environment. Our industry leading cloud computing software.

http://www.eucalyptus.com/products/eee.

[10] Eucalyptus Community. http://open.eucalyptus.com/.

[11] Virt-top. http://people.redhat.com/~rjones/virt-top/index.html.

[12] P. Mell and T. Grance. The NIST definition of cloud computing. http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc, 2009.

[13] H. Wang et al. Distributed Systems Meet Economics: Pricing in the Cloud. In *HotCloud'10: 2nd USENIX Workshop on Hot Topics in Cloud Computing*, 2010.