International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4 | ISSN : 2456-3307

Conducive Tracking, Monitoring, and Managing of Cloud Resources

E. Soumya[1], V. Santhosh Kumar[2], T. Vineela[2], M. Aishwarya[2]
[1]Associate Professor, Department of Computer Science and Engineering, St. Martins Engineering College, Hyderabad, Telangana, India
[2]B.Tech,Department of Computer Science and Engineering, St. Martins Engineering College, Hyderabad, Telangana, India

ABSTRACT

The development of Infrastructure as a Service structure brings new openings, which additionally goes with new difficulties in auto scaling, resource allocation, and security. A major test supporting these issues is the consistent tracking, monitoring and managing of cloud resources in the framework. In this paper, we propose ATOM, a capable and fruitful system to consequently track, screen, and oversee assets utilized as a part of an Infrastructure as a Service (IaaS) structure that is by and large used as a piece of cloud foundation. We utilize novel tracking technique to persistently track essential framework utilization measurements with low overhead,and build up a Principal Component Analysis (PCA) based way to deal with constantly monitor and consequently discover abnormalities in view of the approximated tracking results. We show how to dynamically set the tracking threshold based on the detection results, and further, how to adjust tracking algorithm to ensure its optimality under dynamic workloads. We exhibit the extensibility of ATOM through virtual machine (VM) clustering. In conclusion, when potential anomalies are recognized, we utilize introspection tools to perform memory forensics on VMs guided by analyzed results from tracking and monitoring to find malevolent behavior inside a VM. We assess the performance of our framework in an open source IaaS framework.

Keywprds : Infrastructure As A Service, Cloud, Tracking, Monitoring, Anomaly Detection, Virtual Machine Introspection

I. INTRODUCTION

The Infrastructure as a Service (IaaS) framework is a popular model in realizing cloud computing services. In this model, a cloud provider manages and outsources her computing resources through an IaaS system. For example, Amazon offers cloud service with its Elastic Compute Cloud (EC2) platform which is an IaaS system. While IaaS is an attractive model, since it enables cloud providers to outsource their computing resources and cloud users to cut their cost on a pay-per-use basis, it has raised new challenges in auto scaling, resource allocation, and security. For example, auto scaling in the IaaS framework is the process to automatically add and remove computing resources based upon the actual resource usage. Cloud users want to pay for more resources only when they need them, and to make the best use of their (paid) resources by evenly distributing their workloads. Auto scaling and load balancing, two critical services provided by Amazon hb Service (AWS) and other IaaS platforms, are designed to address these issues. A critical module in achieving auto-scaling and load balancing is the ability to monitor resource usage from many virtual machines (VMs) running on top of EC2. In Amazon cloud, resource usage information needs to be collected and reported back to a cloud controller, not

only for the cloud controller to make various administrative decisions, but also for cloud users to query. Security is another paramount issue while using an IaaS system. For example, it was reported in late July 2014, adversaries attacked Amazon cloud by installing distributed denial-ofservice DDoS) bots on user VMs by exploiting a vulnerability in Elasticsearch . Resource usage data could provide critical insights to address security concerns.

Thus, a cloud provider needs to constantly monitor resource usage and utilize these statistics not only for resource allocation, but also for anomaly detection in the system.

Until now, the best practices for mitigating DDoS and other attacks in AWS include using CloudWatch to create simple threshold alarms on monitored metrics and alert users for potential attacks. Scope of ATOM, an efficient and effective framework to automatically track, monitor, and orchestrate resource usage in an Infrastructure as a Service (IaaS) system that is widely used in cloud infrastructure. We utilize novel following technique to constantly track imperative framework use measurements with low overhead,and build up a Principal Component Analysis (PCA) based way to deal with consistently screen and consequently discover peculiarities in view of the approximated following outcomes. We show how to dynamically set the tracking threshold based on the detection results, and further, how to adjust tracking algorithm to ensure its optimality under dynamic workloads. We exhibit the extensibility of ATOM through virtual machine (VM) bunching. Finally, when potential abnormalities are distinguished, we utilize contemplation instruments to perform memory legal sciences on VMs guided by broke down outcomes from following and checking to recognize malignant conduct inside a VM. We assess the execution of our structure in an open source IaaS framework.

II. LITERATURE SURVEY

To the best of our knowledge, none of existing IaaS platforms is able to provide continuous tracking, monitoring, and orchestration of system resource usage. Furthermore, none of them is able to do intelligent, automated monitoring for a large number of VMs and carry out orchestration inside a VM.

Cloud data monitoring. Most existing IaaS systems follow the general, hierarchical architecture as shown in figure 2. Inside these systems, there are imperative needs for the controller to continuously collect resource usage data and monitor system health. AWS and Eucalyptus, use CloudWatch service to monitor VMs and other components in some fixed intervals, e.g., every minute. This provides cloud users a system-wide visibility into resource utilization, and allows users to set some simple threshold based alarms to monitor and ensure system health. OpenStack is developing a project called Ceilometer, to collect resources utilization measurements. However, these approaches only provide a discrete, sampled view of the system. Several emerging startup companies such as DATADOG and librato could monitor in a more fine-grained granularity, provided the required softwares are installed. However, this inevitably introduces more network overhead to the cloud, which becomes worse when the monitored infrastructure scales up. On the contrary, ATOM significantly reduces the network overhead by utilizing the optimal online tracking algorithm, while providing just about the same amount of information. Furthermore, all these cloud monitoring services offer very limited capability in monitoring and ensuring system health. Astrolabe is a monitoring service for distributed resources, to perform user-definedaggregation. It is intended as a “summarizing mechanism”.

Like Astrolabe, SDIMS is another framework that totals data about substantial scale arranged frameworks with better versatility, adaptability, and managerial confinement. Ganglia is a general-purpose scalable distributed monitoring system for high performance computing systems which also has a

hierarchical design to monitor and aggregate all the nodes and has been used in many clusters.
The allowed error budgets. It suites systems like SDIMS well. InfoEye is a model-based information management system for large-scale service overlay networks through a set of monitoring sensors deployed on different overlay nodes with reduced overhead achieved by ad-hoc conditions filters. InfoTrack is an observing framework that is like ATOM's following module, in that it tries to limit nonstop checking cost with most data accuracy saved, by utilizing fleeting and spatial correlation of monitored attributes, while ATOM utilizes an optimal online tracking algorithm that is proved to achieve the best saving in network cost without any prior knowledge on the data. MELA is a monitoring framework for cloud service which collects different dimensions of data tailored for analyzing cloud elasticity purpose (e.g. scale up and scale down).

III. OVERVIEW OF THE SYSTEMS

ATOM is an end-to-end framework that could be easily plugged into an IaaS system, to provide automated tracking, orchestration, and monitoring of resource usage for a potentially large number of VMs running on an IaaS cloud, in an online fashion. ATOM introduces an online tracking module that runs at NC and continuously tracks various performance metrics and resource usage values of all VMs. The CLC is denoted as the tracker, and the NCs are denoted as the observers. ATOM then uses an automated monitoring module that continuously monitors the resource usage data reported by the online More specifically, UBL can be plugged/integrated into ATOM's monitoring component as an alternative anomaly detection method to be more effective in capturing different types of anomaly. Note that PCA-based approach has the advantage of enabling us to analyze the theoretical bounds, when there are bounded tracking errors present in the continuously tracked measurements returned by the tracking component. UBL is more an empirical method which may perform really well on some instances, but it remains as an open problem to theoretically study its performance especially with approximate measurements when being used together with ATOM's tracking module. PCAbased approach also allows us to adjust the tracking threshold automatically in an online fashion by only adjusting the false alarm rate tracking module. The goal is to detect anomaly by mining the resource usage data. Lastly, SOM requires an explicit training stage and needs to be trained by normal data, while PCA identifies what is normal automatically and is able to adapt to the dynamic change from the underlying data.

The Infrastructure as a Service (IaaS) framework is a popular model in realizing cloud computing services. In this model, a cloud provider manages and outsources her computing resources through an IaaS system. For example, Amazon offers cloud service with its Elastic Compute Cloud (EC2) platform, which is an IaaS system. While IaaS is an attractive model, since it enables cloud providers to outsource their computing resources and cloud users to cut their cost on a pay-per-use basis, it has raised new challenges in auto scaling, resource allocation, and security. For example, auto scaling in the IaaS framework is the process to automatically add and remove computing resources based upon the actual resource usage. Cloud users want to pay for more resources only when they need them, and to make the best use of their (paid) resources by evenly distributing their workloads. Auto scaling and load balancing, two critical services provided by Amazon Web Service (AWS) and other IaaS platforms

They are designed to address these issues. A critical module in achieving auto-scaling and load balancing is the ability to monitor resource usage from many virtual machines (VMs) running on top of EC2. In Amazon cloud, resource usage information needs to be collected and reported back to a cloud controller, not only for the cloud controller to make various administrative decisions, but also for cloudusers to

query. Security is another paramount issue while using an IaaS system. For example, it was reported in late July 2014, adversaries attacked Amazon cloud by installing distributed denial-ofservice (DDoS) bots on user VMs by exploiting a vulnerability in Elasticsearch Resource usage data could provide critical insights to address security concerns. Thus, a cloud provider needs to constantly monitor resource usage and utilize these statistics not only for resource allocation, but also for anomaly detection in the system. Until now, the best practices for mitigating DDoS and other attacks in AWS include using CloudWatch to create simple threshold alarms on monitored metrics and alert users for potential attacks. In our work we show how to detect the anomalies automatically while saving users the trouble on setting magic threshold values. These perceptions outline that a principal challenge supporting a few essential issues in an IaaS framework is the constant following and checking of asset utilization in the framework. Furthermore, several applications (e.g., security) also need intelligent and automated orchestration of system resources, by going beyond passive tracking and monitoring, and introducing auto-detection of abnormal behavior in the system, and active introspection and correction once anomaly has been identified and confirmed. This motivates us to design and implement ATOM, an efficient and effective framework to automaticallytrack, orchestrate, and monitor resource usage in an IaaS system.

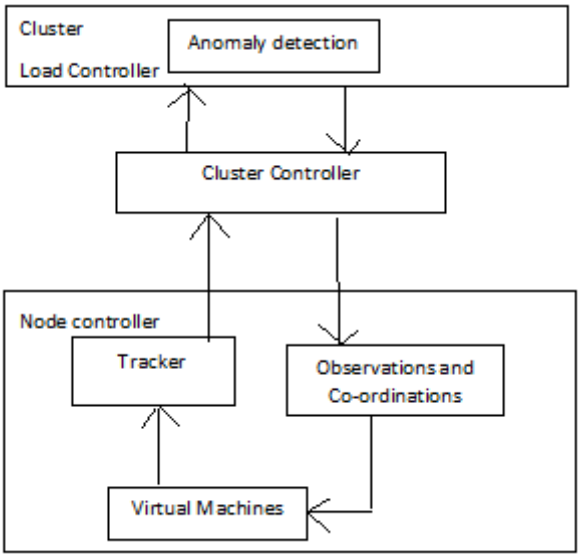

Figure 1. Architecture

IV. METHODOLOGY

In the admin module, admin has to login with valid username and password. After login successful he can do some operations like username, role, mail-id, mobile no, gender, address, profile image), add new places with valuable data, and view user tweets and ratings(place name, user name, tweets and ratings, dates, images etc.,), and view all user history(like username, visited place, visited date, place photo).

In the user module, there are n numbers of users are present. User should register before doing some. After registration successful he can login by using valid user name and password. Login successful he will do some operations like view login user profile details, search city and view historical places in that city, and user can give tweet and ratings, view previous visited user history, and user add trips, view all previous users added trip details

Inaccurate input data are the most common causes of errors in data processing. These perceptions outline that a principal challenge supporting a few essential issues in an IaaS framework is the constant following and checking of asset utilization in the framework. It consists of developing specification and procedure for data preparation.

The main objectives of input design are:

1. 1. Controlling amount of input: Due to so many reasons, design should control the quantity of data for input. Reducing the data requirement can lower cost by reducing labour expenses. By reducing input requirement, the analyst can speed the entire process from data capture to providing results to the users.
2. Avoiding delay: A handling delay coming about because of information arrangement or information section administrator is called bottleneck. Keeping away from bottleneck ought to dependably be one target of the examiner while planning yield.
3. Avoiding errors in data: The rate at which errors occurs depends on the quantity of data, i.e. smaller the amount of data to input the fewer the opportunities for errors.
4. Keeping the process simple: Simplicity works and is accepted by the users. Complexity should be avoided when there are simple alternatives.

Output Design:

The term output necessarily implies to information on printed or displayed by an information system. Following are the activities that are carried out in output design stage.

- Identification of specific output required to meet the information requirements.
- Selecting of methods for processing outputs.
- Designing of reports, formats or other documents that acts as a carrier of information.

Output Design Activities

The output design of an information system must meet the following objectives:

1. The output design should provide information about the past, present or future events. The operational control level outputs provide operations of the past and present events. On the other hand, strategic planning level provides information of the future events.
2. The output design should indicate the important events, opportunities and problems.
3. The output design should be designed keeping in mind that an action must be triggered in response to some event. A set of rule is pre-designed for such trigger.
4. The output design should produce some action to the transaction. For e.g. when the telephone bill is generated, a receipt is printed.

V. RESULT AND DISCUSSION

We actualized ATOM utilizing Eucalyptus as the fundamental IaaS framework. The virtual machine hypervisor running on every NC is the default KVM hypervisor. Each VM has an m1medium type on Eucalyptus. ATOM tracks 7 metrics from each VM instance: CPUUtilization, NetworkIn, NetworkOut, DiskReadOps, DiskWriteOps, DiskReadBytes, DiskWriteBytes. All experiments are executed on a linux machine with an 8-core Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz computer.

Below graph report shows how the online tracking component works. It shows both values sent by standard CloudWatch (without tracking) and values sent by modified CloudWatch with ATOM tracking, with a time interval of 1000 seconds for the NetworkOut metric b This clearly illustrates that at each time instance, with online tracking, the current (exact) value is not sent if it is within D threshold of the last sent value; and at each time point, the last value sent to CLC is always within D of the newest value observed on NC. The values sent by the tracking method closely approximate those exact values, with much smaller overhead.

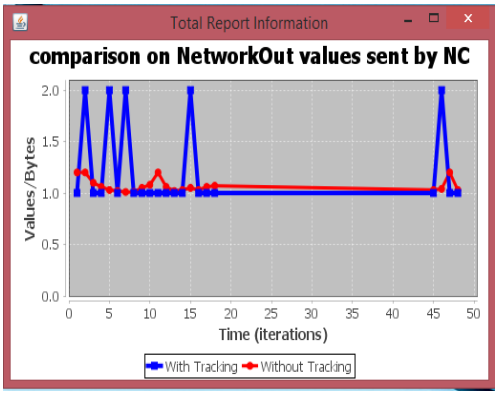

Figure 2. Comparison on NetworOut Values

Below graphs evaluate the scalability of ATOM, we evaluated the key performance metrics of ATOM with an increasing number of VMs. In each configuration, we perform online monitoring using the adapted PCA-based anomaly detection using a sliding window of size 100 (time instances), combined with either online tracking or no tracking (i.e., send everything). We report the average for the false alarm rate, the average PCA running time, and the total number of messages sent from NC to the CLC, per sliding window. The results are shown in graph. Bigger number of VMs prompts higher correspondence cost in ATOM. In any case, the following part of ATOM turns out to be more viable with more VMs, as appeared in graph. This is because there are more opportunities for communication savings when there is a higher probability of temporal locality on one of the many VMs' performance metrics. The computation cost in ATOM is linear to the number of VMs, as shown in virtual machine graph, which is as expected. Nevertheless, the overall computation overhead of ATOM is still fairly small (in just a few milliseconds per sliding window). The deliberate false caution rate really diminishes at first with more VMs. But when the number of VMs keeps increasing, the measured false alarm rate will eventually start to increase, as indicated in final report graph. Initially, when presented with more data, the PCA-based approach becomes more effective in "learning" the normal subspace, hence results in a reduced false alarm rate. But as number of VMs continues to increase, the dimensionality of the data matrix becomes larger, eventually making it less effective to detect abnormal subspace after dimensionality reduction. Nevertheless, ATOM remains very effective in all cases; the false alarmrates are smaller than 1% in nearly all test cases.

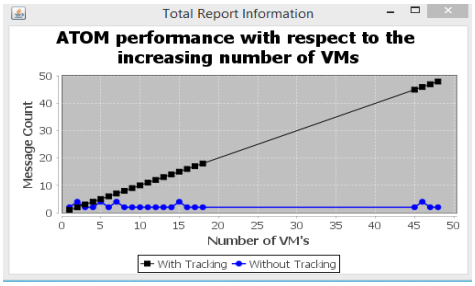

Figure 3. Performance with respect to increasing VM's

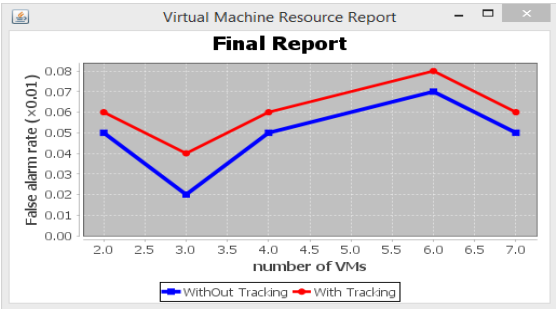

Figure 4. Final Report

VI. CONCLUSION

We present the ATOM framework that can be easily integrated into a standard IaaS system to provide automated, continuous tracking, monitoring, and managing of system resource usagein nearly real-time. ATOM is extremely useful for anomaly detection, auto scaling, and dynamic resource allocation and load balancing in IaaS systems.

ATOM is extremely useful for anomaly detection, auto scaling, and dynamic resource allocation and load balancing in IaaS systems. Interesting future work include extending ATOM for more sophisticated resource orchestration and incorporating the defense against even more complex attacks in ATOM.

VII. REFERENCES

[1]. Amazon.http://www.aws.amazon.com/.Accessed Nov.5, 2016.

[2]. http://www.itworld.com/security/428920/attackers-install-ddos-bots-amazon-cloud-exploiting-elasticsearch-weakness.Accessed Nov.5, 2016.
[3]. Amazon.AWS Best Practices for DDoS Resiliency.https://d0.awsstatic.com/whitepapers/ DDoS White Paper June2015.pdf.Accessed Nov.5, 2016.
[4]. Eucalyptus.http://www8.hp.com/us/en/cloud/helion-eucalyptus.html.Accessed Nov.5, 2016.
[5]. D.Nurmi, R.Wolski, C.Grzegorczyk, G.Obertelli, S.Soman, L.Yous-eff, and D.Zagorodnov, "The eucalyptus open-source cloud-computing system," in CCGRID, 2009.
[6]. W.Dawoud, I.Takouna, and C.Meinel, "Infrastructure as a service security: Challenges and solutions," in INFOS, 2010.
[7]. D.J.Dean, H.Nguyen, and X.Gu, "Ubl: Unsupervised behavior learning for predicting performance anomalies in virtualized cloud systems," in ICAC, 2012.