Efficient and Cost-Effective DDoS Defenses

MD Tafazul Arfa[1], B.Nithilesh[2], A. Pavan Kumar[3], Ms. K.Sangeeta[4]
[1,2,3] B.Tech. Student, Dept. of CSE Institute of Aeronautical EngineeringHyderabad, India
[4]Assistant professor, Dept. of CSE Institute of Aeronautical EngineeringHyderabad, India

Abstract: As cloud systems grow in popularity, they become more vulnerable to cyber-attacks. A distributed denial of service (DDoS) attack is one of the most notorious cyber-attacks. The attack aims to exhaust the system's resources, rendering it unresponsive to legitimate requests. Around the resource competition, DDoS defenseand attack essentially revolve. We have made decisions from a resource management and investment perspective. However, these defense strategies often assume unlimited resources to defend against attacks without considering financial costs. This coarse- grained approach can lead to resource overprovisioning and unnecessary expenses. To tackle this challenge, we perform a thorough analysis and propose a birth-death-based fine-grained resource management system that dynamically scales resources in/out and up/down. This system adaptively chooses the optimal resource leasing mode for cloud service customers, providing cost- effective defense against DDoS attacks. Extensive analyses and experiments based on empirical data validate the efficiency and effectiveness of our approach. Compared to existing methods, our proposal can reduce defense costs by an average of 53.58%, with potential savings of up to 93.75.

Keywords: resource management, cloud security, DDoS attacks

I. INTRODUCTION

The point of the paper is to create a strategy for minimizing the financial costs associated with defending against Distributed Denial of Service (DDoS) attacks in cloud environments. This is achieved through the implementation of a fine-grained resource management mechanism that can dynamically and adaptively allocate resources to cloud service customers (CSCs) in response to DDoS threats.

In the past decade, various applications have moved to the cloud, driven by its demonstrated advantages. Additionally, the cloud is increasingly embracing new paradigms [1]. More and more cyber attackers are naturally setting up shop in the cloud. The notorious DDoS attack is a major threat from all theattacks that cause enormous damage to industry [2], [3]. Thefundamental issue of DDoS defense and attack revolves aroundthe competition for resources between attackers and defenders, as has been demonstrated. Recently researchers [4], [5], [6] in the battle, the winner is the side that possesses more resources. In a cloud environment, the two primary entities are cloud service customers (CSCs) and cloud service providers (CSPs).

The CSPs control and maintain the cloud infrastructure, whereas DDoS attacks, typically executed through botnets with a few hundred or thousand active bots, frequently fail to impact CSPs significantly. However, individual CSCs, who rely on rented resources from CSPs to run their applications, are more susceptible to these attacks and lack the protection that CSPs inherently provide.

The growth has also made cloud platforms a prime target for DDoS attacks, which can overwhelm resources and disruptservices. Traditional resource management strategies often result in overprovisioning, leading to unnecessary financial burdens for cloud service customers (CSCs).
To address these challenges, this paper introduces a fine- grained resource management mechanism designed to optimize resource allocation during DDoS attacks. By employing a birth- death process model, the proposed approach dynamically adjusts the allocation of virtual machine (VM) instances, ensuring that CSCs can maintain service continuity while minimizing costs. The viability of this strategy is illustrated through recreations and observational information, appearing that it can give a more effective and cost-effective of DDoS defense in cloud environments

Due to the highly unpredictable and variable of cloud workloads, [9-10] it is essential for cloud service customers (CSCs) to dynamically and adaptively manage resource provisioning to ensure efficient utilization. By the CSP to CSCs the resource provision suggested unfortunately, is either not accurate or blurry. On the basis of performance and experience Most CSCs have to select cloud configurations [11]. To choose the optimal cloud configurations there are many efforts have been made.

Venkataraman et al. [12] highlighted the importance of predicting application performance across different resource configurations to find the optimal setup. Alipourfard et al. [11] developed CherryPick, a system using Bayesian optimization to identify optimal configurations for recurring big data tasks. These model-based methods rely on pre-trained performance models to guide cloud configuration but are limited by their accuracy, as they are typically trained on non-attack scenario data.

As a result, existing resource provisioning strategies [11], [12], [13] are inadequate for defending against DDoS attacks. Additionally, they often ignore the financial implications for service owners, particularly individual CSCs. Current strategies are typically coarse-grained, like adding virtual machines (VMs) to scale out, which can result in overprovisioning and unnecessary financial costs for CSCs. To address this problem, we introduce a method that lowers the financial impact on CSCs during DDoS defense by dynamically choosing the most efficient resource leasing option.

We introduce a fine-grained resource management mechanism that monitors services in real-time to ensure they perform as expected [15]. Specifically, the mechanism tracks the workloads of the CSC system, adaptively increasing resources during a DDoS attack and reducing them when the attack subsides, ensuring continuous service operation. The numerous existing projects focus on cloud-based DDoS defense [16], [17], [18], most, like the mechanism proposed by Yu et al. [8], rely on a coarse-grained approach. Their strategy of simply cloning the same VM(s) to mitigate attacks can lead to resource wastage, as it doesn't adaptively select different types of VMs for different scenarios.

We argue that the need for resource reinvestment varies depending on the situation, with factors like CPU and memory consumption both playing critical roles during a DDoS attack. Most existing works, including Yu [8], focus only on CPU resource consumption when modelling the system, which can be inefficient if memory is the limiting factor during an attack.

To address this, our approach considers both CPU and memory resources—key factors in most CSC systems and pricing models—when modelling the system. We create a birth-death process model to estimate memory use and processing time based on CPU usage, and then formulates an optimization problem to select the ideal VM type and number for cost-effective attack defense.

The key contributions of the paper are:

- It is crucial to take into account various resource types when safeguarding against DDoS attacks.
- We are creating a model of the birth-death process to gauge memory usage and compute processing duration.
- We are conducting thorough analyses and experiments that demonstrate the effectiveness of our approach and acceptable time overhead.

II. LITERATURE SURVEY

DDoS attacks pose significant threats to internet security by overwhelming a target's services, making them unavailable to legitimate users. These attacks can lead to service disruptions, resource conflicts, and severe financial and reputational damage. While various methods exist to mitigate DDoS attacks, organisations still find it challenging to fully counteract their impact. A highly effective approach combines bandwidth limiting and resource reservations, although these can increase response times for all users during an attack. To address this, we propose a two-tier defense system. The first line separates incoming requests based on the number of connections and directs them to the second line. Here, bandwidth is limited, and a load balancer directs requests to different containers based on the defense system's decisions. This method efficiently processes legitimate user requests, ensuring service availability for 98% of legitimate traffic, even during intense DDoS attacks. [1]

Edge computing (EC) and container technology enhance computing flexibility for real-time applications but face challenges from low-rate DDoS (LDDoS) attacks that are difficult to detect. Existing defense methods often increase resource consumption and service delays, reducing system efficiency. This paper proposes a solution using Moving Target Defense (MTD) and deep reinforcement learning (DRL) to mitigate LDDoS attacks efficiently. The approach combines lightweight MTD mechanisms and a deep Q-network (DQN) algorithm to balance effectiveness and resource usage. Simulations show significant improvements in security (31.7%) and service quality (26.95%) with minimal overhead, including the lowest response time per request and webpage load time compared to other strategies. [2]

This paper presents a distributed method for detecting DDoS flooding attacks at the traffic-flow level, designed for ISP core networks. The defense system uses distributed change- point detection (DCD) framework with change-aggregation trees (CAT) to identify early traffic anomalies across various network domains. This keeps victim systems from getting too damaged. The system involves attack-transit routers that cooperatively detect and report flooding alerts, which are then aggregated by CAT servers within each ISP domain. These servers collaborate to make final decisions, supported by a Secure Infrastructure Protocol (SIP) to resolve policy conflicts. Simulations on the DETER testbed demonstrate a 98% detection accuracy with only 1% false positives using four network domains, and the system scales effectively to protect up to 84 autonomous system domains, offering broad protection against DDoS attacks.[3]

DDoS attacks are a significant security challenge, especially when executed through distributed botnets. Early detection is crucial to protect users and network infrastructure. This paper introduces FireCol, a defense system built on intrusion prevention systems (IPSs) deployed at the ISP level. FireCol creates virtual protection rings around hosts, enabling collaboration between IPSs by sharing traffic information. Extensive simulations and evaluations with real datasets showcase minimal overhead, FireCol's effectiveness, and its capability for gradual deployment in real networks. [4]

In today's competitive IT landscape, cloud computing provides scalable, on-demand services but faces significant security challenges, particularly DDoS attacks that deplete resources and disrupt service availability. This paper proposes using fog computing, an extension of cloud computing, toanalyze and filter DDoS attack traffic at the network edge beforeit reaches the cloud, enhancing real-time decision-making and reducing data forwarded to the cloud. [5]

The rise in cloud computing has increased vulnerability to DDoS attacks. This paper presents a Multiple Layer Defense (MLD) scheme to detect and mitigate these attacks. MLD operates in two layers: the first sends alarms to cloud management when an attack begins, and the second detects anomalies indicating virtual machines (VM) are compromised. Testing MLD with various DDoS attack ratios shows it effectively reduces energy consumption and SLA violations while maintaining stability. [6]

DDoS attacks in cloud environments are becoming increasingly sophisticated, especially with low-rate DDoS attacks. With the rise of container technology and microservice architecture in cloud computing, which offers more lightweight virtualization and flexible scaling, there's a need to reassess how these new features can defend against DDoS attacks. This paper explores enhancing the resilience of container-based cloud environments against low-rate DDoS attacks by establishing a mathematical model to analyze the feasibility ofusing these new features. A mitigation strategy is proposed and validated through simulations and experiments. [7]

III. PROPOSED METHOD

Since their invention, cloud services have gained a lot of popularity because they provide heavy computation and storage spaces at a lower cost. This popularity opened up new ways for cyber attackers to introduce millions of BOTS to spread DDOS attacks, which make cloud servers busy and make genuine requests fail. To address this issue, cloud service providers (CSPs) allocate substantial resources to combat Distributed Denial of Service (DDOS) attacks. However, cloud customers may select resources based on their individual experience and requirements, which may not be optimal and may result in additional costs during DDOS attacks.

To address the aforementioned issue, the author of this paper presents the concept of birth and death, along with fine-grained cloud VM selection. This approach can assist customers in combating DDOS attacks while minimizing costs. In the proposed paper, the author calculates CPU and memory requirements based on request size, increasing BIRTH by 1. Once a request completes, the author decreases BIRTH by 1 and increases death by 1.

The author employs a fine-grained algorithm to monitor request requirements. If a request requires heavy resources, the algorithm allocates those resources. If not required, the algorithm releases memory and CPU data to freeup resources.

In the proposed paper, the author uses linear programmingto identify finer-grained resources, but you ask us to implement a GA algorithm with a dataset. The entire dataset will train the GA algorithm, which will then select an optimal virtual machine (VM) based on the requirements of the request. We will base the GA algorithm on population, mutation, andfitness. We will use the population to generate new random features, then apply mutation to calculate the fitness between the generated and actual data. If the fitness is high, we will select the output; if not, we will continue the process until we achieve an optimised output.

In our proposed work, we are also training the GA algorithm on a cloud VM dataset, and then this algorithm will be applied to test data to predict the optimal VM with less memory and CPU usage. Selecting a VM with less memory andCPU usage will automatically lower the cost.

The proposed method consists of the following key components:

1. Cloud Users Dataset Analysis: Analyse datasets of cloud users to understand behaviour, traffic patterns, and resource usage. This involves collecting data, performing behavioural analysis, recognising traffic patterns indicative of DDoS attacks, and examining resource utilisation trends.
2. DDoS Detection and Classification: Detect and classify DDoS attacks using anomaly detection algorithms to identify unusual traffic spikes, as well as machine learning classifiers to determine attack type and severity.
3. Resource Allocation Model: Create a dynamic resource allocation model based on real-time traffic patterns and attack classifications, including resource pooling, priority assignment, and automatic scaling.
4. Optimization Using Genetic Algorithms: Use genetic algorithms to improve the resource allocation model by starting with a population of strategies, testing them with a fitness function, and then using selection, crossover, and mutation to make solutions better over time for cheap DDoS defense.

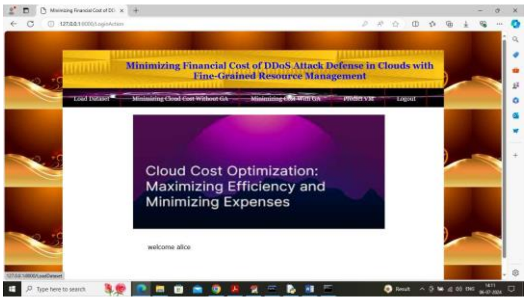
To train GA algorithm we are using below VM ResourcesDataset. In dataset we have information on CPU Usage, Memory Usage, Disk Usage and many more other features. Wehave evaluated GA VM selection performance in terms of accuracy, precision, recall and FSCORE. In given dataset we have 5 different types of VM's. We have compared GA accuracy with non-GA algorithm accuracy.

We designed modules to implement this project,

1) Sign up: user can sign up with the application
2) Login: user can login to application
3) Load Dataset: after login user can load and process dataset features such as shuffling, normalization etc.
4) Minimizing Cloud Cost without GA: in this module user can run V selection algorithm without GA and then calculate selection accuracy
5) Minimizing Cost with GA: in this module application will apply GA algorithm for optimal VM selection which can reduce cost
6) Predict VM: in this module user can enter REQUESTrequirements and then trained GA will predict optimalVM name.

IV. RESULT

To run the project, install Python 3.7.2, then install the packages listed in the requirements.txt file, and then install MySQL.


Minimizing Financial Cost of DDoS Attack Defense in Clouds with Fine-Grained Resource Management

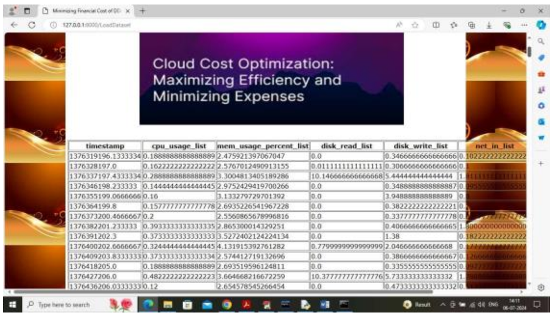
Cloud Cost Optimization: Maximizing Efficiency and Minimizing Expenses

welcome alice

Minimizing Financial Cost of DD
127.0.0.1:8000/LoginAction
Load Dataset
Minimizing Cloud Cost Without GA
Minimizing Cost With GA
Predict VM
Logout
Type here to search
Result

Fig 4.1. user can click on 'Load Dataset' link

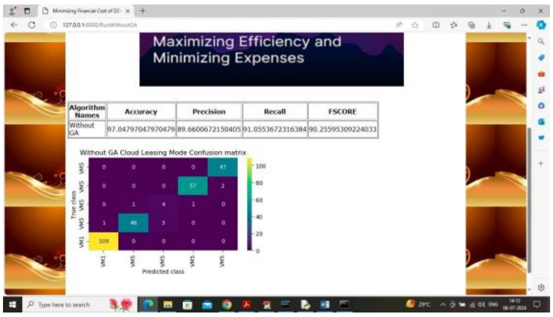
In the screen above, the user can click on the 'Load Dataset' linkto load and process the dataset, which will result in the page below,


Minimizing Financial Cost of DD
127.0.0.1:8000/LoadDataset
Cloud Cost Optimization:
Maximizing Efficiency and
Minimizing Expenses
Type here to search

timestamp	cpu_usage_list	mem_usage_percent_list	disk_read_list	disk_write_list	net_in_list
1376319196.1333334	0.1888888888888889	2.475921397067047	0.0	0.3466666666666666	0.1022222222222222
1376328197.0	0.1622222222222222	2.5767012490913155	0.0111111111111111	0.3066666666666666	0.1
1376337197.4333334	0.2888888888888889	3.3004813405189286	10.146666666666668	5.444444444444444	[illegible]
1376346198.233333	0.1444444444444445	2.9752429419700266	0.0	0.3488888888888887	[illegible]
1376355199.0666666	0.16	3.133279729701392	0.0	3.948888888888889	[illegible]
1376364199.8	0.1577777777777778	2.6935226541967228	0.0	0.3822222222222221	[illegible]
1376373200.4666667	0.2	2.5560865678996816	0.0	0.3377777777777778	[illegible]
1376382201.233333	0.3933333333333335	2.865300014329251	0.0	0.4066666666666665	[illegible]
1376391202.3	0.3733333333333333	3.5272402124224134	0.0	1.38	0.1822222222222222
1376400202.6666667	0.3244444444444445	4.131915392761282	0.7799999999999999	2.046666666666668	[illegible]
1376409203.8333333	0.3733333333333334	2.574412719132696	0.0	0.3866666666666667	[illegible]
1376418205.0	0.1888888888888889	2.693519596124811	0.0	0.3355555555555555	[illegible]
1376427206.0	0.4822222222222223	3.664668216672259	10.377777777777776	5.733333333333332	[illegible]
1376436206.0333333	0.12	2.654578545266454	0.0	0.4733333333333332	[illegible]

Fig 4.2. dataset loaded

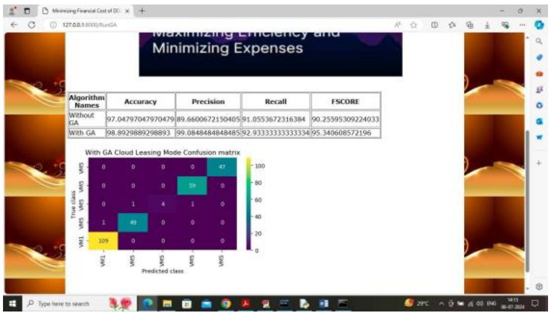
In above screen dataset loaded and now click on ‘Minimizing Cloud Cost without GA’ link to train algorithm without GA to predict optimal VM selection on TEST data and get below prediction accuracy


Minimizing Financial Cost of DD
127.0.0.1:8000/RunWithoutGA
Maximizing Efficiency and
Minimizing Expenses
Without GA Cloud Leasing Mode Confusion matrix
True class
Predicted class
VM1
VM5
100
80
60
40
20
0
Type here to search

Algorithm Names	Accuracy	Precision	Recall	FSCORE
Without GA	97.04797047970479	89.6600672150405	91.0553672316384	90.25595309224033

Fig 4.3. without GA algorithm got 97% accuracy on optimalVM selection

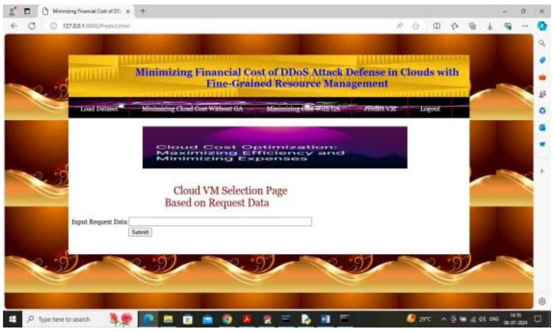
On the above screen, without the GA algorithm, I got 97% accuracy on optimal VM selection and could see other metrics like precision, recall, and FSCORE. Click on the 'Minimising Cost with GA' link to train the algorithm using GA optimisation, resulting in the output shown below,



Algorithm Names	Accuracy	Precision	Recall	FSCORE
Without GA	97.04797047970479	89.6600672150405	91.0553672316384	90.25595309224033
With GA	98.8929889298893	99.0848484848485	92.93333333333334	95.340608572196

Fig 4.4. GA algorithm got more than 98% accuracy

In above screen with GA algorithm got more than 98% accuracy and can see other metrics also. Now click on ‘PredictVM’ link to get below page



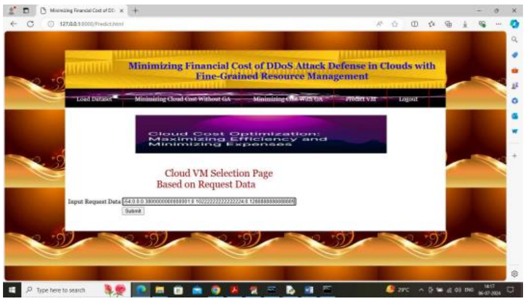
Minimizing Financial Cost of DDoS Attack Defense in Clouds with Fine-Grained Resource Management

Cloud VM Selection Page
Based on Request Data

Input Request Data

Fig 4.5. ‘Request Requirement’ data

In above screen paste ‘Request Requirement’ data which you can copy from ‘testData.csv’ file from code folder like below screen



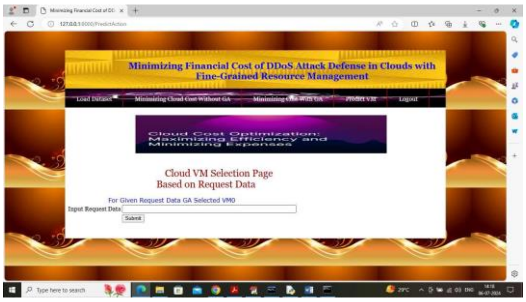
Minimizing Financial Cost of DDoS Attack Defense in Clouds with Fine-Grained Resource Management

Cloud VM Selection Page
Based on Request Data

Input Request Data 554,0.0,0.38000000000000001,0.10222222222222224,0.12888888888888

Fig 4.6. I pasted copied data

In above screen I pasted copied data and then press button to get below page like precision, recall, and FSCORE. Click on the 'Minimising Cost with GA' link to train the algorithm using GA optimisation, resulting in the output shown below,



Minimizing Financial Cost of DDoS Attack Defense in Clouds with Fine-Grained Resource Management

Cloud VM Selection Page
Based on Request Data

For Given Request Data GA Selected VM0

Input Request Data

Fig 4.7. blue colour text can see VM0 is the optimal selection

In above screen in blue colour text can see VM0 is the optimalselection for given request and similarly you can copy each lineand paste in application to get optimal VM selection

V. CONCLUSION

In this paper, we focus on reducing the financial burden for individual CSCs defending against DDoS attacks. We propose a fine-grained resource management system that dynamically selects the optimal resource leasing mode. Using a birth-and-death process model, we formulate the problem as a constrained optimization challenge and solve it using integer linear programming (ILP).Our approach is validated through detailed analysis and extensive simulations, showing its effectiveness and efficiency with real-world data.

REFERENCES

[1] Kumar, Anmol, and Mayank Agarwal. "Quick service during DDoS attacks in the container-based cloud environment." Journal of Network and Computer Applications 229 (2024): 103946.

[2] Zhou, Yuyang, Guang Cheng, Zhi Ouyang, and Zongyao Chen. "Resource-Efficient Low-Rate DDoS Mitigation With Moving Target Defense in Edge Clouds." IEEE Transactions on Network and Service Management (2024).

[3] Y. Chen, K. Hwang, and W .- S. Ku, "Collaborative detection of DDoS attacks over multiple network domains," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 12, pp. 1649-1662, 2007.

[4] J. Franc,ois, I. Aib, and R. Boutaba, "Firecol: a collaborative protection network for the detection of flooding DDoS attacks," IEEE/ACM Transactions on Networking, vol. 20, no. 6, pp. 1828-1841, 2012.

[5] Bhushan, Kriti. "DDoS attack defense framework for cloud using fog computing." In 2017 2nd IEEE international conference on recent trends in electronics, information & conmumication technology (RTEICT), pp. 534-538. IEEE, 2017.

[6] Khalil, Moataz H., Mohamed Azab, Ashraf Elsayed, Walaa Sheta,Mahmoud Gabr, and Adel S. Elmaghraby. "Maintaining cloud performanceunder DDOS attacks." IJCNC 11, no. 6 (2019): 1-22.

[7] Li, Zhi, Hai Jin, Deqing Zou, and Bin Yuan. "Exploring new opportunities to defeat low-rate DDoS attack in container-based cloud environment." IEEE Transactions on Parallel and Distributed Systems 31, no. 3 (2019): 695-706.

[8] T. Baker, E. Ugljanin, N. Faci, M. Sellami, Z. Maamar, and E. Kajan, "Everything as a resource: Foundations and illustration through internet-of-things," Computers in Industry, vol. 94, pp. 62-74, 2018.

[9] "Aliyun suffered from DDoS attack," http://tech huangiu.com/cloud/2014-12/5288347.html.

[10] "DDoSattackreport2017,"http://cybersecurityventures.com/ddos-attack-report-2017/.

[11] S. Yu, S. Guo, and I. Stojmenovic, "Can we beat legitimate cyberbehavior mimicking attacks from botnets?' in Proceedings of the 31st IEEE International Conference on Computer Communications, 2012, pp. 2851-2855.

[12] M. A. Fabian, R. J. Zarfoss, and M. A. Terzis, "My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging." in Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets, 2007.

[13] S. Yu, Y. Tian, S. Guo, and D. Wu, "Can we beat DDoS attacks inclouds?' IEEE Transactions on Parallel and Distributed Systems, vol. 25, no.9,pp. 2245-2254, 2014.

[14] M. N. Bennani and D. A. Menasce, "Resource allocation for autonomic' data centers using analytic performance models," in Proceedings ofthe Second International Conference on Autonomic Computing, 2005, pp. 229-240.

[15] K. Shen, H. Tang, T. Yang, andL. Chu, "Integrated resourcemanagement for cluster-based internet services," in Proceedings of the 5th Symposium on Operating System Design and Implementation, 2002.a.