*International Journal of Research in Engineering and Science (IJRES)*
*ISSN (Online): 2320-9364, ISSN (Print): 2320-9356*
*www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 552-565*

# Performance Aware Intelligent Resource Management in Cloud

J Arravinth , D Manjula

*J Arravinth is with the Department of Computer Science and Engineering, College of Engineering Guindy, Anna University Chennai, Tamil Nadu, India*

*D Manjula is with the Department of Computer Science and Engineering, College of Engineering Guindy, Anna University Chennai, Tamil Nadu, India*

***ABSTRACT***

***T**he dynamic nature and on demand service of cloud computing has made the resource management challenging for Cloud Service Providers (CSP) and introduces new levels of complexity to Quality of Service (QoS). The static analysis gives the raw information of issues influencing the performance. Machine learning algorithm helps to find the general trend, patterns, and characteristics to find the parametric equation of functionality of the dynamic system. With the result, an inference could be driven from which performance of the system could be analyzed and predicted. We propose Performance Aware Intelligent Resource Management Architecture (PAIRMA) that reads the value of the performance diminishing parameters. We use Principal Component Analysis (PCA) to eliminate performance non-influencing parameter. We also propose Bayes included load dependent server algorithm to analyze and evaluate the performance of the parameters than analyzing the tradi- tional threshold based method which gives a high error percentage. We found four performance diminishing parameters and trained using four different machine learn- ing models, end of the training incremental model was built using Support Vector Machine (SVM) and evaluated with Amazon Web Services (AWS), Azure. We also evaluated the error percentage against real time logs of the cloud using the Mean Absolute Error (MAE) technique.*

***Keywords: Cloud, Cloud Computing, Machine Learning, Predictive Models, Principal Component Analysis, Bayes method.***

Date of Submission: 29-05-2022 Date of acceptance: 10-06-2022

## I. INTRODUCTION

With the fast development in Information Technology (IT), Cloud computing has risen as the substitute for traditional distributed computing for offering different types of services to customers at any time, on demand and on a pay per use basis [1]. This enables customers to access a set of consumable computing resources such as storage, networks, servers, and applications. Leading enterprises and IT companies such as AWS, Microsoft Azure, Google Anthos, IBM cloud, Red Hat, Verizon cloud, also called cloud service providers, offer above stated cloud services to customers on demand, and on a pay as use basis [2]. Theprimary objective of the cloud is to utilize the distributed resources efficiently in order to achieve high performance and at an effective cost. It frees customers from traditional computing and offers high end computing. Cloud computing supports auto scaling in or out virtual resources without manual support, dynamically.

Cloud computing is segregated in two ways based on service or location. Service based computing is categorized into threetypes, fundamentally, that is Platform as a Service (PaaS),

F Software as a Service (SaaS) and Infrastructure as a Service ( (SaaS) [3].

Networking, Servers were offered by IaaS. Computer hardware and Operating System (OS) is offered by PaaS. In addition, Database as a Service (DaaS), Expert as a Service (EaaS), Storage as a Service (SaaS), Network as a Service (NaaS), Securityas a Service (SECaaS), Communication as a Service (CaaS) [3] are also offered. Location based cloud is segregated into three types. They are public, private and hybrid or community cloud. Public cloud services can be accessed by anyone but this cloud suffers from various security issues. Private cloud services are exclusively available to groups in an organization. It is highly secured. Hybrid cloud is a mixture of both public and private cloud. It is available particularly for a specific community. The main aim of the Cloud Service Provider (CSP) is to find better ways to allocate the resources efficiently while maintaining theprovision of the service level agreements (SLA) as per the expectation. Minimal Quality of Service (QoS) between CSP and Customer is maintained by the contract, SLA [4]. Due to the characteristics of the cloud, dynamic, heterogeneous workloads, thestatic way of resource planning and solution will not work. Therefore, it needs additional support in planning and managementof the resources. The resource management in the cloud comprises all the techniques and procedures that adjust the resource configuration with respect to the customers and applications in the cloud. In this article, our objective is to identify the major challenges in cloud and solution to the same. We present the various anomaly parameters to enhance the performance. Initially, data was collected from the public cloud such as AWS and Azure. It was analyzed and found that the parameters that are diminishing the performance of the cloud using Bayes included the load dependent server. Then, with the help of ARIMA, machine learning algorithm, future resources demand by a specific customer is predicted and resources were reserved and allocated to improve the performance of the cloud. The performance of the algorithm is evaluated using real time logs of the cloud.

## II. LITERATURE SURVEY

Large scale cloud applications are distributed and deployed in cloud clusters [5]. Static resource provisioning causes inefficiency and unsynchronized functions in the clusters. To overcome this, workload prediction, component placement and system consolidation and application elasticity are addressed. Workload Analysis and Modeling: Uses PCA to find critical metrics [6, 7]. Using Big Data Bench, workload traces were retrieved and k-means clustering was applied and classified the crucial and independent metrics which does not influence the performance of the cloud [8-10]. This method is limited only to the publicly available dataset and its evaluation has a huge impact on workload characterization. Component placement and system consolidation: It is modeling the resource allocation as graph problem and solved using traditional linear programming techniques [8]. Adding this method to offloading improves the application performance. Still this method suffers from expensive cost and delays. Its evaluation with respect to genetic algorithms, neural networks and heuristics gives high cost delays, which is not highly recommended for the CSP. Elasticity and Remediation: Load balancing is addressed as optimization problems. The objective of this method is to minimize the energy consumption and delays [9].

Most of the studies provide the workload predictions and auto scaling. But, all these proposed methods suggest generic methods. The first two methods were suffering from high cost and delays. But workload characterization and PCA is considered in our proposed model for performance aware intelligent resource allocation proposed dynamic load balancing algorithms, provisioning and de provisioning of resources. The aim of load balancing is to achieve provisioning and de provisioning of resources, continuous support even in case of failure of any of the service components, and reduce the energy consumption [10,31]. To achieve this, some of the efficient algorithms were proposed by the researchers. They are, Meta heuristics algorithm, Over-loading in cloud environment, a deadline-constrained scheduling algorithm for aperiodic tasks in an inter-cloud environment, is the Swift - Balancing workload on cloud storage system [11]. All these algorithms are focusing only on load balancing which is one of the parameter influencing the performance of the cloud. There are various parameters associated with performance. It is not addressed in any of the algorithms. Implementing algorithm in resource manager only for the load balancing will have huge costs associated with the cloud service provider.

The scalability addresses volume based scaling of cloud software services and provide a practical measure and features of CSP [12]. This is significant to help successful estimation and testing the versatility of the cloud service provider. Data Intensive applications and skew data problems are solved by Balanced Data Cluster Practitioner (BDCP). BDCP reduces tasks by sampling and providing feedback to the current processing task. This algorithm performance is evaluated with Hadoop-Hash and Range algorithms [13]. This BDCP algorithm is concentrating only on the skewing data problem which is also part of the performance of the cloud. It does not completely address the performance issue. We have proposed an approach to model the pattern and decentralized cloud data center using pareto distribution to estimate the resource requirement with consideration to limitation and failure of the cloud [14, 32]. This problem is solved using genetic algorithms since the complexity of the problem is high. With the feature of dynamic cloud, dynamic pricing problem was GA4SRP used to solve the same. Its correctness and complexity is measured and analyzed to find out the various performance affecting factors.

As opposed to these works, our work has a more integrated perspective of the PAIRMA in the cloud. It covers both application based workload analysis and anomaly detection techniques and management of resource especially, auto-scaling methods as the main resource management for cloud-hosted applications. We have specifically covered the works that integrate both performance analysis and corresponding resource auto scaling techniques, implementing performance monitoring, performing data analysis and modeling machine learning algorithm to predict the resource demand and allocate in advance, and also to free up the resources that are not efficiently used.

## III. ARCHITECTURE

### 3.1 ALGORITHM

```
 ALGORITHM: BAYES INCLUDED LOAD DEPENDENT SERVER
FOR $I$ = 1 TO $M$ DO
$Q_i$ = 0 // Fixed capacity and delay cloud center
   $P_i$ (0) = 0 // General load dependent cloud centers
  FOR $n$ = 1 TO $N$ DO
      FOR $i$ = 1 TO $M$ DO

$R_i = \begin{cases} S_i\,(1+Q_i) \text{ //Fixed Capacity} \\ R_i = S_i \text{ // Delayed Cloud Centers} \\ \sum_{j=1}^{n} P_i(j\lambda I)\frac{j}{\mu(j)} \text{ // Load dependent server} \end{cases}$

$\sum_{i=1}^{M} R_i V_i$
$N=\frac{N}{Z+R}$

FOR $I$ =1 TO $M$ DO
$Q_i = X*V_i + R_i$ // Fixed or density
FOR j = n TO 1 DO
$P_i(j/i)=\frac{X}{\mu(j)}P(i/j-1)P_i(i)$
$P_i(0)=1-\sum_{j=1}^{n} P_i(j)$
                END FOR
        END FOR
   END FOR
END FOR
FOR $I$=1 TO M DO
$X_I = X*V_I$
END FOR
FOR $I$=1 TO N DO
$U_I = X*S_I*V_I$ // Fixed capacity or delay center
$U_I = 1- P_I(0)$    // Load dependent center
```

A load dependent service center service rate varies with the load, that is, the number of jobs in the center 5. For example, a VM with $m$ disk drives sharing a single request is represented whose service rate is represented by $\mu(n)$ with $n$ requests,

$$\mu(n) = \begin{cases} \frac{n}{s}, & n=1,2,3...m-1 \\ \frac{m}{s}, & n=m,m+1, \ldots, \infty \end{cases}$$

Where S - Service time per request when there is only one request in the VM. This is like M/M/$m$ traditional queuing model. As the name implies, mean value analysis gives the, mean performance of the system. Mean value analysis is applying to cloud with both fixed and varying number of VMs. Like closed queuing network, a fixed number of VMs with N jobs or request, the response time is state below,

$R_i = S_i\,(1 + Q_i)$, where $Q_i$ is mean length of the
Response queue of $i^{th}$ system.

$R_i = S_i$ , Delayed Cloud Center

$$R_i = \sum_{j=1}^{n} P_i(j\lambda l)\frac{j}{\mu(j)}$$ Load dependent server

Given the individual response time of the VMs, the overall mean response time is calculated using,

$$R = \sum_{i=1}^{M} R_i V_i$$

Where, $R_i$ is the individual response time of the $i^{th}$ VM, $V_i$ is the number of request made to VM.

The following equation computes the probability of the $j^{th}$ job in $i^{th}$ VM using Bayes theorem and also measures it's through- put

$$P_j(j/i) = \frac{X}{\mu_i(j)} P(i/j-1) P_i(i)$$

Using the above, future jobs and respective throughput can be measured so that resource manager can be dynamically reserved to serve the future request with less amount of time and to enhance the performance of the cloud.

The overall throughput of the CSP can be measured using,

$$X_i = X * V_i$$

The overall cloud service utilization is measured using

$U_i = X * S_i * V_i$ ; for fixed number of VMs

$U_{iB} = 1\lambda P_i\ (0)/m$ ; for load dependent server

### 3.2 Flow Diagram

Performance of the cloud is influenced by various parameters. Using respective performance related APIs, values were read and analyzed using Bayes included load dependent serer algorithm [15, 16]. ARIMA model is trained and implemented with incremental learning to predict the values from which resource manager reserves resource in advance which efficiently improves the performance and overall throughput of the cloud services also is increased.

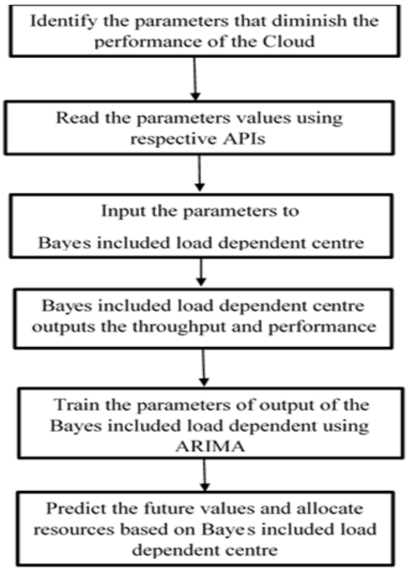

Identify the parameters that diminish the performance of the Cloud
Read the parameters values using respective APIs
Input the parameters to Bayes included load dependent centre
Bayes included load dependent centre outputs the throughput and performance
Train the parameters of output of the Bayes included load dependent using ARIMA
Predict the future values and allocate resources based on Bayes included load dependent centre

**Figure 1**

### 3.3 Source of Anomaly

### 3.3.1 Performance Degradation in Cloud

In a highly reliable environment, consistent, stable and low latency performance is expected. There is a wide scope of causes distinguished for these issues, from changes in the approaching remaining task at hand to failing the hardware or buggy software that can influence the exhibition of the application or VMs [17]. In this manner, the beginning time of the performance ought to be known so that an appropriate and auspicious restorative activity can be started. Checking sensors that follow the presentation of every segment produce huge measure of information, which incorporate concealed examples and indications of the strength of the framework [26, 29]. Already, we had to depend on the human administrators to skim information and find dubious conduct [18]. In any case, considering the size of the information created from many machines situated in various locations, the manual methodology is not, at this point doable. Consequently, researchers have begun to exploit propelled information examination techniques and more remarkable and practical registering equipment to computerize and quicken the procedure. These outcomes have better-quality information on the execution.

### 3.3.2 Corrective Measures

To diminish the performance problems of the system, the Resource Manager should start a restorative measure in the form of load balancing, resource provisioning, migrations etc. [18, 27]. Present cloud service providers, such as Amazon or Microsoft Azure, offer migrations and threshold-based scalability that supports the dynamic support of the extending VMs. There are additionally customized approaches for cloud, for example, on- the-fly changes in the resource setup of one VM, which is offered by some CSPs. The following factors were found to improve the performance of the cloud.

*Technical Limitations*:

Cloud model supports dynamic VMs to be added, and deleted for enhanced performance and on demand service [19, 29]. However, hypervisor and kernel support is needed for the same but this is not supported by the famous CSP Amazon, Azure.

*Business prerequisite*:

There are many resources offered by CSP with different prices based on the service they provide. There are many rules associated for vertical scaling of VMs [20, 30] Thus, some of the performance degradations may be acceptable by some of the owner.

*Service Level Agreements*:

SLA understandings are contracts among clients and CSPs that recognize the normal QoS got by clients [21, 27]. These are generally founded on the yields of the framework detectable by clients, for example, the accessibility of the administration or the deferrals accordingly. Having explicit necessities for the yield of the framework may restrain accessible decisions of the RM.

### 3.3.3 Architecture

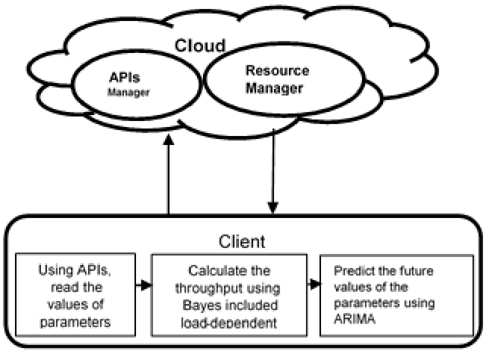

Cloud
APIs
Manager
Resource
Manager
Client
Using APIs, read the values of parameters
Calculate the throughput using Bayes included load-dependent
Predict the future values of the parameters using ARIMA

**Figure 2**

The proposed architecture is generic which is compatible with all cloud service provider. All the famous CSPs are providing APIs for reading the value of performance parameters [22, 34]. Initially, performance related parameters values were read using APIs. The read values are input to the Bayes included load-dependent server. This algorithm estimates the performance of the cloud based on input parameters. It calculates the cloud response time, system response time and throughput of the cloud in each iteration. Iteration is repeated till the number of observations are available. In every iteration queue length and expected probabilities are also calculated. While calculating the probability, our model is implemented with Bayes theorem which estimates the expected probability of the same type of request in future which helps to reserve the resource in future unlike constantreservation [33, 35]. At end of the outermost nested for loop in the algorithm, the overall throughput and

CSP utilization was measured. Using this, current performance of the cloud service can be estimated. The same parameter values are input then to the ARIMA model which learns and predicts the values of the performance parameters. By considering parameters, resources canbe reserved so that throughput and utilization of the cloud can be improved and hence the performance of the cloud is enhanced.Our model, PAIRMA uses the Bayes model unlike the traditional conditional probability, which does not predicts the value and outputs only the approximate value, whereas ours models predicts and gives the accurate value. Also, we have estimated the throughput, cloud service utilization to estimate the overall performance. The existing approaches discuss CPU load, playsinfluences performance but with respect to our algorithm and CSP jitter plays one of the major role in performance of cloud CPU usage is likely to be considered as minor parameter since it is dependent on response time.

## IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

*Parameters*:

AWS Global Accelerator (AGA) network performance such as round-trip time, network latency, jitters over a four- week period from 38 global vantage factors and compares it to the default Internet-extensive AWS connectivity from the equal vantage factors [23, 28]. Parameters are classified into two categories, viz, Constant Parameters and Variable Parameters

*Constant Parameters*:

Parameters whose value cannot be changed from the cloud setup till tear comes under this category. Some of such parameters are VM Image size, Memory, Data center architecture, load balancing algorithms etc., [24]. These parameters are not considered for performance evaluation since they do not impact performance of the system, they are retained as constants and is dependenton the service provider.

*Variable Parameters*:

Parameters, whose value are often changed, are dependent on users, and independent from service provider comes under thiscategory[25]. For experimentation purpose, we have taken CPU load, Latency, Response time and Jitter.

We have taken two public clouds AWS and Microsoft Azure for performance evaluation. The values for these parameters were driven from APIs. Four-weeks of data were aggregated using mean. Using standard deviation means of the parameters were cross validated to identify the outlier and adjusted to summarize the data for every day. Using the following specified APIs, we can read the parameters value as shown in Table 1 .

**Table 1:** API's to read SCP parameter values.

| CSP/Parameters | AWS | Azure |
|---|---|---|
| CPU<br>Load | awscloudwatch<br>list-metrics-namespace<br>AWS/EC2-metric-name<br>CPU Utilization | com.microsoft<br>azure.managa<br>ment.monitor |
| Latency | awscloudwatch<br>list-metrics-namespace<br>AWS/EC2-metric-name<br>Latency | com.microsoft<br>applications<br>insights.extensibility |
| Response Time | awscloudwatch<br>list-metrics-namespace<br>AWS/EC2-metric-name<br>Response Time | com.microsoft<br>azure.magane<br>ment.monitor |
| Jitter | awscloudwatch<br>list-metrics-namespace<br>AWS/EC2-metric-name<br>Jitter | com.microsoft<br>azure.sdk.iot.d<br>evice.transport |

Using the above APIs of both CSP, the parameters values were read for 28 days as shown in Table 2 and 3, AWS and Azurecloud respectively. The same were input for Algorithm 3.1 for the performance analysis, trained using linear regression, SVM Polynomial, SVM RBF and ARIMA. The predicted values by ARIMA are shown in Figure 3 since it gives very low MAE. The mean service time per response for AWS Cloud is 0.26 seconds. All the requests were gone through the flow equivalent centerwhich is modeled with a load dependent service center. It is visited by every request once. Its service rate is represented by $\mu(n)$.

AWS average service rate is $\mu(aws)$=0.34 response/second

Azure average service rate is $\mu(\mathrm{ma})$=0.39 response/second

Iteration 1:

Initialize $Q_B(0) = 0$ and $P(0|)) = 1$

Response Time:

$R_B(1) = S_{\mathbf{B}}[1+Q_B(0)] = 0.26$

$R_{FEC}(1) = P(0|0)\ 1/\mu(1) = 3.13$

Throughput:
$X(1) = N/R(1)$
$R(1) = R_B(1)\ V_B + R_{FEC}\ (1)\ V_{FEC}$
$= 0.26 \times 2 + 3.13 = 3.65$
$=1/3.65 = 0.27$
Queue length and probabilities:
$Q_B = X(1)\ R_B(1)\ V_B = 0.27 \times 0.26 \times 20.14$

$P(1|1)= X(n)/\mu(n)\ P(0|0)= 0.37/0.32 \times 1 = 0.66$
$P(0|1)= 1\text{-}P(1|1) = 1\text{-}0.66 = 0.34$
Repeating the iteration for 28 times for both clouds, we will be getting the values as stated in table 2 and 3.

AWS throughput was initially 1.08 jobs/second and in final iteration it would become 0.87 jobs/second. It should not be the case. By applying PCA on the dataset, response time, latency and jitter would be the main parameters that decide the CPU utilization. Unit variance scaling is applied to rows; SVD with imputation is used to calculate principal components. X and Y axes show principal component 1 and principal component 2 that explain 69.3% and 30.7% of the total variance, respectively. N = 3 data points. Therefore, CPU load is ignored and rest of the three parameters were used to improve the performance of the cloud. By the results of the PCA, response time, latency and jitter has been prioritized in order. Training the model using linear regression to predict the future values of the same parameters, we will be getting the below values for

**Table 2: 2**8 day's observation of parameters in AWS

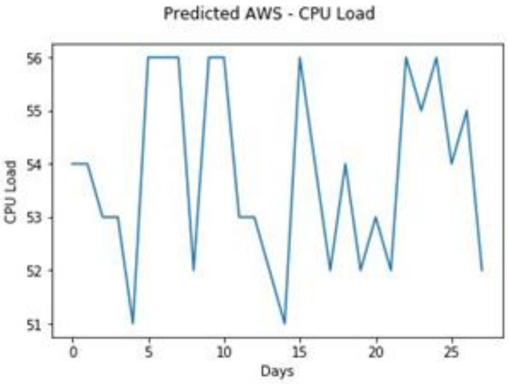
| CSP/ Parameters/ Date | CPU Load | Latency | Response Time | Jitter |
|---|---|---|---|---|
| 1-11-2019 | 51 | 110 | 568 | 209 |
| 2-11-2019 | 51 | 380 | 502 | 273 |
| 3-11-2019 | 54 | 232 | 309 | 262 |
| 4-11-2019 | 56 | 353 | 330 | 164 |
| 5-11-2019 | 52 | 334 | 262 | 189 |
| 6-11-2019 | 56 | 406 | 258 | 274 |
| 7-11-2019 | 54 | 273 | 291 | 187 |
| 8-11-2019 | 51 | 179 | 448 | 257 |
| 9-11-2019 | 55 | 380 | 369 | 213 |
| 10-11-2019 | 54 | 243 | 429 | 297 |
| 11-11-2019 | 54 | 412 | 546 | 226 |
| 12-11-2019 | 56 | 309 | 383 | 178 |
| 13-11-2019 | 52 | 280 | 422 | 189 |
| 14-11-2019 | 56 | 192 | 224 | 250 |
| 15-11-2019 | 53 | 186 | 408 | 158 |
| 16-11-2019 | 52 | 230 | 258 | 282 |
| 17-11-2019 | 56 | 241 | 289 | 190 |
| 18-11-2019 | 53 | 303 | 486 | 296 |
| 19-11-2019 | 52 | 334 | 368 | 208 |
| 20-11-2019 | 54 | 216 | 237 | 289 |
| 21-11-2019 | 55 | 332 | 202 | 216 |
| 22-11-2019 | 55 | 245 | 347 | 155 |
| 23-11-2019 | 51 | 167 | 233 | 157 |
| 24-11-2019 | 53 | 339 | 534 | 248 |
| 25-11-2019 | 51 | 308 | 338 | 200 |
| 26-11-2019 | 54 | 310 | 414 | 224 |
| 27-11-2019 | 55 | 429 | 449 | 231 |
| 28-11-2019 | 51 | 243 | 517 | 161 |

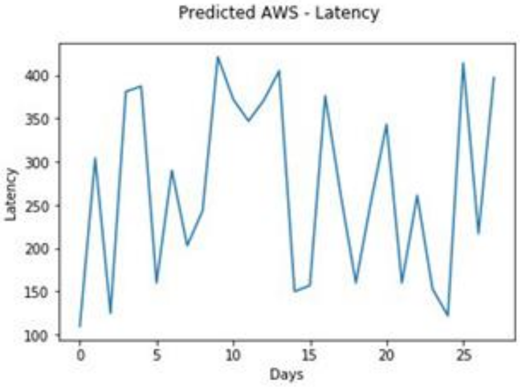
AWS:
Score of the training model is S = 0.976454
Coefficient of the classifiers $C_i$ = 1.08931242, -0.04424296,
-0.02088607
Intercept of the model is 186.21191028653243
**Azure:**
Score of the training model is S = 0.965636
Coefficient of the classifiers were $C_i$ = 0.29353546, 0.04406765, 0.00645871I
Intercept of the model is 263.8751297528675

**Table 3:** **2**8 day's observation of parameters in Azure cloud

| CSP/<br>Parameters/<br>Date | CPU Load | Latency | Response Time | Jitter |
|---|---|---|---|---|
| 1-11-2019 | 37 | 437 | 376 | 352 |
| 2-11-2019 | 49 | 483 | 432 | 229 |
| 3-11-2019 | 38 | 368 | 308 | 387 |
| 4-11-2019 | 39 | 572 | 605 | 388 |
| 5-11-2019 | 36 | 423 | 460 | 352 |
| 6-11-2019 | 44 | 574 | 376 | 330 |
| 7-11-2019 | 54 | 406 | 397 | 350 |
| 8-11-2019 | 36 | 250 | 514 | 305 |
| 9-11-2019 | 31 | 593 | 590 | 274 |
| 10-11-2019 | 45 | 515 | 546 | 203 |
| 11-11-2019 | 51 | 344 | 692 | 311 |
| 12-11-2019 | 38 | 487 | 373 | 273 |
| 13-11-2019 | 45 | 449 | 629 | 240 |
| 14-11-2019 | 54 | 356 | 598 | 349 |
| 15-11-2019 | 39 | 354 | 372 | 252 |
| 16-11-2019 | 40 | 200 | 350 | 358 |
| 17-11-2019 | 42 | 373 | 448 | 372 |
| 18-11-2019 | 32 | 380 | 674 | 342 |
| 19-11-2019 | 47 | 543 | 490 | 331 |
| 20-11-2019 | 43 | 480 | 605 | 281 |
| 21-11-2019 | 34 | 499 | 455 | 320 |
| 22-11-2019 | 40 | 587 | 317 | 265 |
| 23-11-2019 | 48 | 275 | 583 | 343 |
| 24-11-2019 | 40 | 443 | 602 | 340 |
| 25-11-2019 | 33 | 540 | 490 | 294 |
| 26-11-2019 | 31 | 209 | 647 | 335 |
| 27-11-2019 | 31 | 249 | 644 | 270 |
| 28-11-2019 | 44 | 365 | 417 | 217 |

The graphs in figure (3) and (4) show the future predicted values using linear regression. All the parameters future values are predictedbased on the training and Bayes included load dependent server algorithm. The above results were evaluated against real time logs of the future of 28 days.





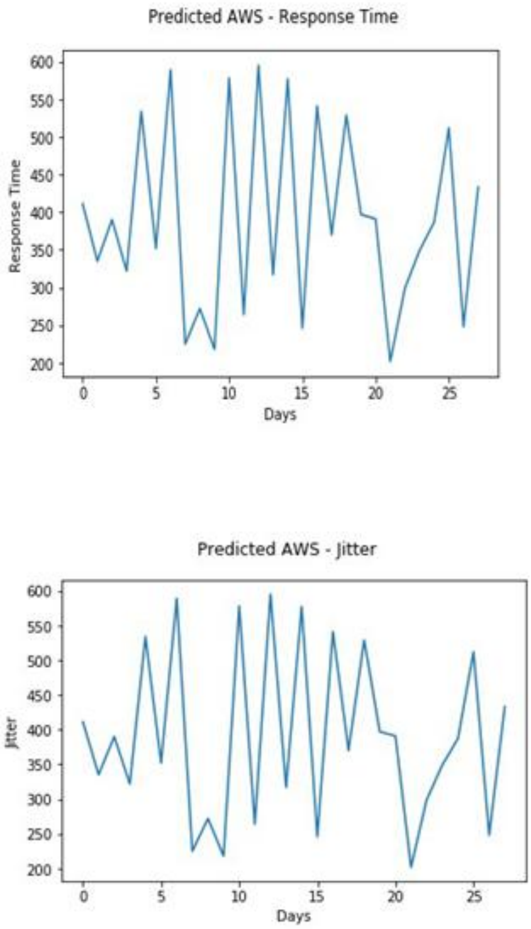
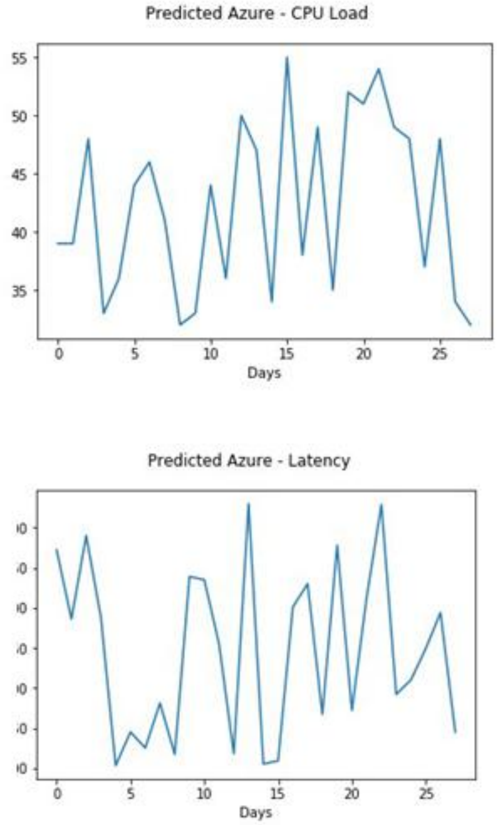
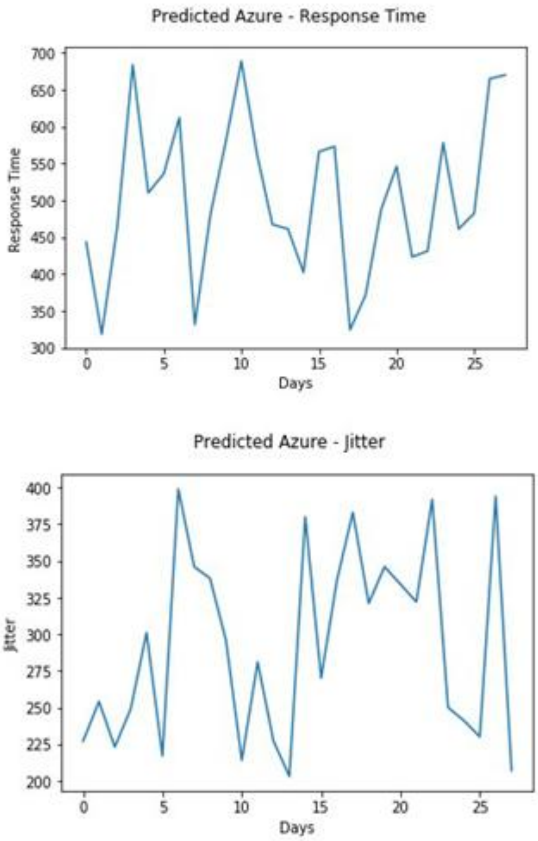


**Figure 3:** Various plots for Predicted AWS.





Predicted Azure - Response Time

Predicted Azure - Jitter

**Figure 4:** Various plots for Predicted Jitter.

## V. RESULTS AND DISCUSSION

In this section, we present the parameters and results of the ARIMA model for predicting the CPU Load, Latency, Response time, Jitter based on the one month of historical observations using APIs (input dataset). Then using these predictions and the Bayes included load dependent server algorithms performance has been evaluated. To prove the accuracy of the algorithm and proposed model, we used test dataset that allows us to compare and predict the future data. This allows us to compare our model results with real VM load and estimated loads and optimal resource utilization. We compare our proposed model with basic different machine learning algorithms such as Linear Regression, based on last value. The input data chosen to train the ARIMA model and our results were evaluated using real time logs of the Vms.

*Performance Evaluation*:

We have applied different machine learning techniques with AWS and Azure cloud and finally we have chosen the ARIMA model for our proposed algorithm since it is giving more accurate results than all other models. We have evaluated CPU load, Latency, Response time and Jitter against real time logs with all the algorithms. We also measure the error percentage using Mean Absolute Error (MAE) technique and shown in Table 4 and Table 5 AWS and Azure respectively.

$$MAE=\frac{1}{n}\sum_{j=1}^{n}(A_i - P_i)/A_i$$

where, $A_i$- Actual value from the log and
$P_i$ – Predicted value by the model

AWS:

The table 4 shows the MAE for the AWS cloud. By considering the table, we can infer that Linear Regression produces the high error than rest of the model. So, we can say that Linear Regression is less useful. Both the SVM Polynomial and SVM RBF model shows very less error percentage but higher than ARIMA model. Finally, ARIMA is the best model for performance evaluation of the cloud. The graphs in Figure 5 shows the exact replica of the same.

**Table 4:** MAE of ML Models on AWS cloud

| Mean Absolute Error % | CPU Load | Latency | Response Time | Jitter |
|---|---|---|---|---|
| Line Reg SVM | 2.1222 | 2.3124 | 2.2122 | 2.7856 |
| Polynomial | 1.1231 | 1.3243 | 1.9766 | 1.1234 |
| SVM RBF | 1.1233 | 1.3211 | 1.2311 | 1.4198 |
| ARIMA | 0.5876 | 0.3756 | 0.2322 | 0.1999 |

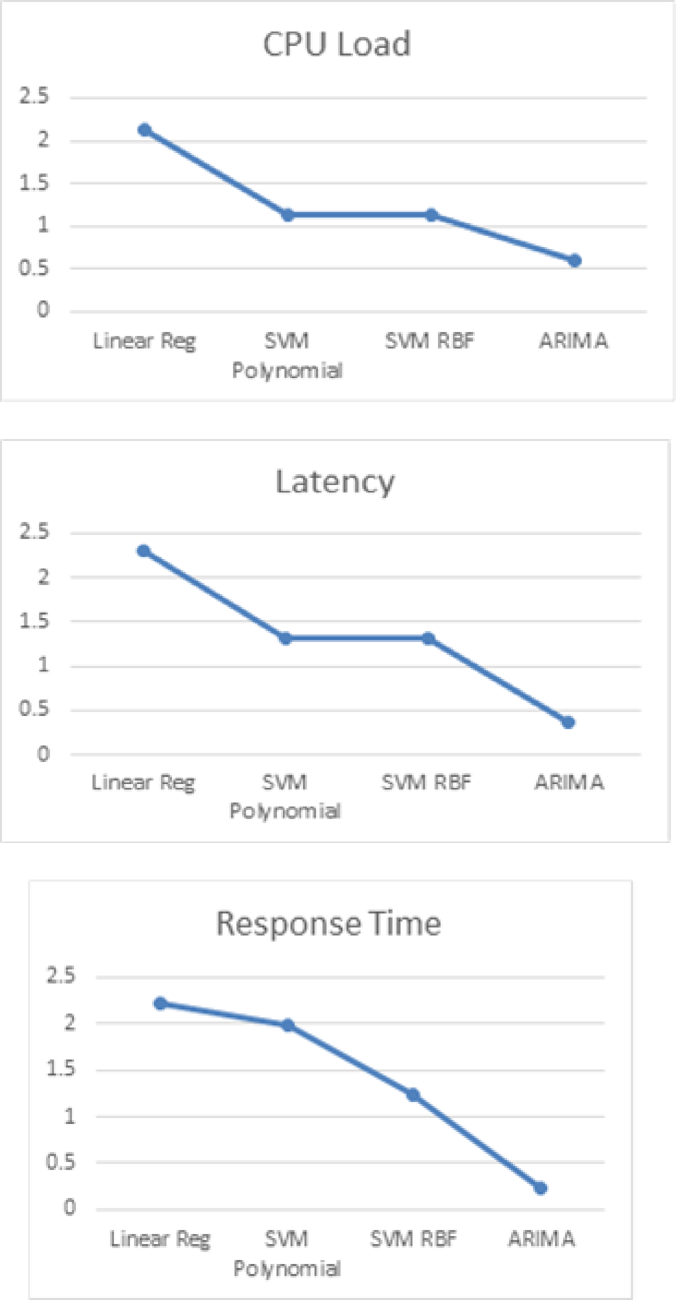


**Figure 5:** Various plots for AWS Predicted.

Azure:

We have also evaluated with Azure cloud. As AWS, linear regression produces high error than all other models. Both SVM polynomial and SVM RBF are less likely to produce error. But as we found in AWS, the ARIMA model works better than all their models. Figure 6 shows MAE of ML models on Azure Cloud.

**Table 5:** MAE of ML Models on Azure cloud

| Mean Absolute Error % | CPU Load | Latency | Response Time | Jitter |
|---|---|---|---|---|
| Line Reg SVM | 2.7222 | 2.4124 | 2.6122 | 2.7856 |
| Polynomial | 1.9231 | 1.4243 | 1.9766 | 1.3234 |
| SVM RBF | 1.4233 | 1.5211 | 1.3311 | 1.6198 |
| ARIMA | 0.5276 | 0.3556 | 0.2522 | 0.2299 |

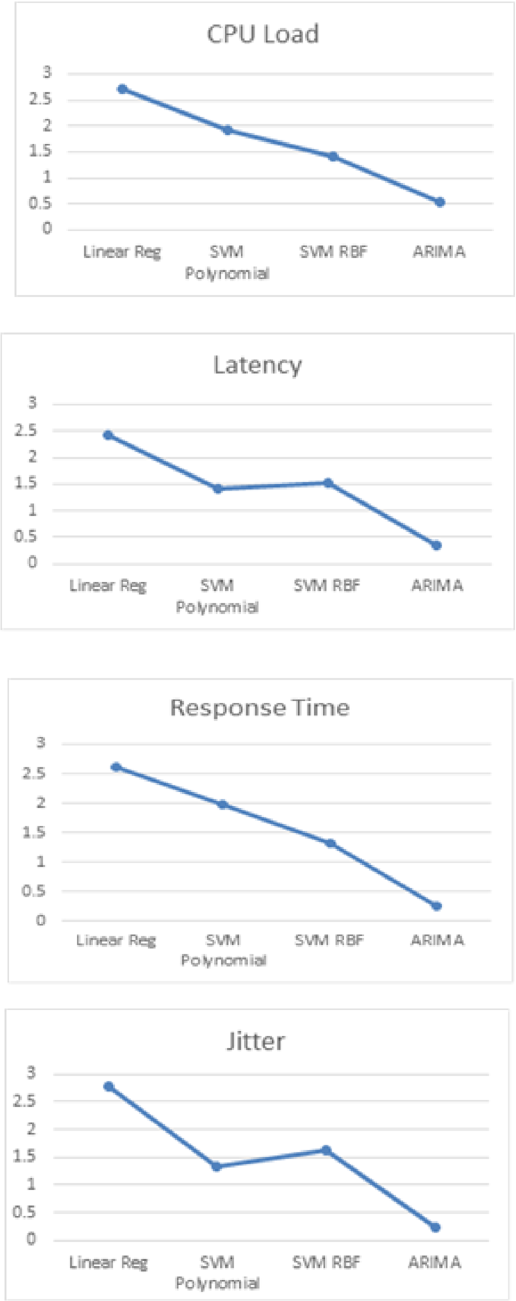


**FIGURE 6:** Various plots for Azure Predicted.

Overall performance of our PAIRMA architecture is shown below in Table 6 by consolidating both AWS and Azure together.The performance of our algorithm gives the best result with both cloud. Once again finding the MAE percentage also gives good results. As we can see from the table, even linear regression and SVM Polynomial Algorithm give good results but that areonly for certain parameters. Linear regression gives more error percentage than all other algorithms. Comparatively, ARIMA models give very less error percentage and hence used for implementation of the proposed model PAIRMA.

**Table 6:** MAE of ML Models on AWS and Azure cloud

| Mean Absolute Error % | CPU Load | Latency | Response Time | Jitter |
|---|---|---|---|---|
| Line Reg SVM | 0.30 | 0.05 | 0.20 | 0.00 |
| Polynomial | 0.40 | 0.05 | 0.00 | 0.10 |
| SVM RBF | 0.15 | 0.10 | 0.05 | 0.10 |
| ARIMA | 0.03 | 0.01 | 0.01 | 0.02 |

## VI. CONCLUSION

With the development of cloud computing and the power of machine learning algorithms, new doors have opened for the improvement of performance-aware resource management techniques. The analysis of two different clouds, gives meaningful insight about the performance. In this article, we investigate different parameters that are associated with the performance of the cloud. To distinguish the significant limitations and contemplation’s in the determination of the best strategy for resource management in the cloud there is a need for more complicated and determination strategies to deal with the dynamism of the cloud. We have proposed the taxonomy of the performance diminishing parameters as a source of knowledge for automated resource allocations and presented the survey. We conclude that our PAIRMA architecture is the most suitable method for automated dynamic resource allocation.

## VII. FUTURE WORK

Our proposed Bayes included load dependent server algorithm involves more arithmetical calculations. Since all the data werenumerical, deep learning techniques can be applied instead of machine learning. The time complexity of the algorithm can be reduced using randomization or approximation algorithm. The overall performance of the proposed model relies on the ARIMAmodel. In future, it can be replaced with other machine learning algorithms.

## REFERENCES

[1]. Sara Kardani Moghaddam, Rajkumar Buyya, and Kotagiri Ramamohanarao, Performance-Aware Management of Cloud Resources: A Taxonomy and Future Directions. ACM Computing Surveys, Vol. 52, No. 4, Article 84. Publication date: August 2019.

[2]. Pawan Kumar, Rakesh Kumar, Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey. ACMComputing Surveys, Vol. 51, No. 6, Article 84. Publication date: February 2019.

[3]. Samuel Ajila and Akindele Bankole. 2016. Using machine learning algorithms for cloud client prediction models in a webVM resource provisioning environment. Transactions on Machine Learning and Artificial Intelligence 4, 1, 28.

[4]. M. H. Ghahramani, M. Zhou and C. T. Hon, "Toward cloud computing QoS architecture: analysis of cloud systems and cloud services," in IEEE/CAA Journal of Automatica Sinica, vol. 4, no. 1, pp. 6-18, Jan. 2017.

[5]. Google Cloud 2020. Retrieved on July 8, 2020 https://cloud.google.com/solutions/using-clusters-for-large- scale-technical-computing.

[6]. Raphael Andreas Hauser , Armin Eftekhari , Heinrich F Matzinger, "PCA by Determinant Optimisation has no Spurious Local Optima", KDD ’18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data MiningJuly 2018, pp 1504–1511.

[7]. O. Anisfeld, E. Biton, R. Milshtein, M. Shifrin and O. Gurewitz, "Scaling of Cloud Resources-Principal Component Analysisand Random Forest Approach," 2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE), Eilat, Israel, 2018, pp. 1-5.

[8]. Shalmoli Gupta , Ravi Kumar , Kefu Lu, Benjamin Moseley, Sergei Vassilvitskii, "Local search methods for k- means withoutliers", Proceedings of the VLDB Endowment, Volume 10, Issue 7 March 2017, pp 757–768.

[9]. D. Ardagna, M. Ciavotta, R. Lancellotti, and M. Guerriero. 2018. A hierarchical receding horizon algorithm for QoSdrivenControl of multi-IaaS applications. IEEE Transactions on Cloud Computing, 1–1.

[10]. Seungcheol Ko , Seongsoo Park, Hwansoo Han, "Design analysis for real-time video transcoding on cloud systems", SAC ’13: Proceedings of the 28th Annual ACM Symposium on Applied ComputingMarch 2013, pp 1610–1615.

[11]. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25, 6, 599–616.

[12]. Xavier Dutreilh, Aurélien Moreau, Jacques Malenfant, Nicolas Rivierre, and Isis Truck. 2010. IEEE Computer Society, Washington, DC, 410– 417.

[13]. Ibrahim Adel Ibrahim and Mostafa Bassiouni "Improvement of job completion time in data-intensive cloud computing applications", Journal of Cloud Computing: Advances, Systems and Applications, Article Number: 8 (2020).

[14]. Amro Al-Said Ahmad and Peter Andras, "Scalability analysis comparisons of cloudbased software services", Journal of Cloud Computing: Advances, Systems and Applications, Article number: 10 (2019).

[15]. Omer Y. Adam; Young Choon Lee; Albert Y. Zomaya, "Constructing Performance-Predictable Clusters with Performance- Varying Resources of Clouds", IEEE Transactions on Computers, Volume: 65, Issue: 9, Pp 2709 – 2724.

[16]. Philipp Leitner and Jürgen Cito, "Patterns in the Chaos—A Study of Performance Variation and Predictability in Public IaaS Clouds", ACM Transactions on Internet Technology, Volume 16, and Issue 3August 2016, Article No.: 15, pp 1–23.

[17]. Juan J. Merelo, "Cloudy Distributed Evolutionary Algorithms", GECCO '16 Companion: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, July 2016, pp 819–821.

[18]. Yasuhiko Kanemasa; Shuji Suzuki; Atsushi Kubota;Junichi Higuchi, "Single-View Performance Monitoring of On-Line Applications Running on a Cloud", 2017 IEEE 10th International Conference on Cloud Computing (CLOUD).

[19]. Tessema Mengistu ; Abdulrahman Alahmadi ; Abdullah Albuali ; Yousef Alsenani ; Dunren Che, "A" No Data Center" Solution to Cloud Computing", 2 017 IEEE 10th International Conference on Cloud Computing (CLOUD).

[20]. Nikita Konovalov; Nikolay Kazantsev, "Shaping Decision- Making on Cloud Services Application in Business Processes", 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud).

[21]. Jonathan Lejeune, Julien Sopena, Pierre Sens "Service Level Agreement for Distributed Mutual Exclusion in Cloud Computing" CCGRID '12, May 2012 pp. 180–187.

[22]. Amazon AWS APIs.Retrieved on Nov 8, 2019 https://docs.aws.amazon.com/userguides.

[23]. AWS Global Accelerator (AGA).Retrieved on Nov 15, 2019 https://aws.amazon.com/global-accelerator/.

[24]. Abdel-Rahman Al-Ghuwairi, Mohammad Khalaf, Zahar Salah, Ayoub Alsarhan "Dynamic changes of QoS parameters in cloud computing service level agreement", International Journal of Business Information Systems, Volume 32, Issue 1.

[25]. Mamata Rath, "Resource provision and QoS support with added security for client side applications in cloud computing", International Journal of Information Technology volume 11, pp 357–364(2019).

[26]. Johannes Grohmann, Patrick K Nicholson, Jesus Omana Iglesias, Samuel Kounev, D. Lugones "Monitorless: Predict- ing Performance Degradation in Cloud Applications with Machine Learning" Middleware '19: Proceedings of the 20th International Middleware ConferenceDecember 2019 Pages 149–162.

[27]. Usman Wazir, Fiaz Gul Khan, Sajid Shah, "Service Level Agreement in Cloud Computing: A Survey", International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 6, June 2016.

[28]. Keith R. Jackson ; Lavanya Ramakrishnan ; Krishna Muriki ; Shane Canon ; Shreyas Cholia ; John Shalf ; "Perfor- mance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud", 2010 IEEE Second International Conference on Cloud Computing Technology and Science.

[29]. Young Choon Lee and Albert Y. Zomaya "Energy efficient utilization of resources in cloud computing systems", The Journal of Supercomputing vol. 60, pp 268–280(2012).

[30]. Victor Chang; Gary Wills; David De Roure "A Review of Cloud Business Models and Sustainability", 2010 IEEE 3rd International Conference on Cloud Computing.

[31]. Martin Randles ; David Lamb ; A. Taleb-Bendiab "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[32]. Hota A., Mohapatra S., Mohanty S. (2019) Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review. In: Behera H., Nayak J., Naik B., Abraham A. (eds) Computational Intelligence in Data Mining. Advances in Intelligent Systems and Computing, vol 711. Springer, Singapore.

[33]. Swe Swe Aung; Thinn Thu Naing "Naïve Bayes Classifier Based Traffic Prediction System on Cloud Infrastructure", 2015 6th International Conference on Intelligent Systems, Modelling and Simulation.

[34]. Reginaldo Ré, Rômulo Manciola Meloca, Douglas Nassif Roma Junior, Gabriel Costa Silva, Marcelo Alexandre da Cruz Ismael, "An empirical study for evaluating the performance of multi-cloud APIs", Future Generation Computer Systems Volume 79, Part 2, February 2018, pp 726-738.

[35]. Jiao-Hong Qiang, Ding-Wan Ning, Tian-Jun Feng, and Li- Wei Ping "Dynamic Cloud Resource Reservation Model Based on Trust", J Inf Process Syst, Vol.14, No.2, pp.377-395, April 2018.