International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Cloud Monitoring System: Basics, Phases and Challenges

Mahantesh N. Birje, Chetan Bulla

ABSTRACT--- Due to many economical benefits cloud computing has quickly become popular and is widely used for delivering services over the Internet. As the number of cloud users increases day by day, data centers are continuing to grow in terms of hardware resources, virtual resources and traffic volume; thus making cloud operation and management more intricate. To manage complex infrastructure of cloud, an efficient and effective Cloud Monitoring System (CMS) is needed to improve the overall performance of cloud. Cloud monitoring is the process of reviewing, controlling and managing the operational workflow and processes within a cloud-based IT asset or infrastructure. This paper deals with detailed study of CMSs based on their architecture, monitoring phases, properties and functions. It describes various phases of cloud monitoring activities and presents the comparative study of the state-of-the-art works of theses phases. Usage of agents for monitoring of cloud activities is also described. Finally, various challenges/issues and possible future directions of cloud monitoring is discussed. This paper helps researchers, engineers and scientists to know the state-of-the-art works for monitoring cloud activities.

Index Terms: Cloud computing, Cloud monitoring, Multi agent system, Monitoring phases, Challenges.

I. INTRODUCTION

Cloud computing has quickly become popular as it is widely used for delivering services over the Internet. This is due to various economical benefits like fast setup, easy to adopt, lesser effort, reduced price, on demand usage, improvement of energy efficiency, optimization of hardware and software resource utilization, performance isolation, flexibility, elasticity and on-demand service schema [4]. To provide cloud services, the cloud service provider has to do lots of work in the background like advanced virtualization techniques, robust and dynamic scheduling approaches, advanced security measures and disaster recovery mechanisms are implemented and operated in cloud computing systems. The cloud computing involves various activities [1] such as resource planning and management, data center management, SLA management and billing, cloud troubleshooting, performance management, fault management, configuration management, security management and data management [8]. Monitoring and Managing of these activities in complex, and heterogeneous cloud environment is crucial. Also, as the number of cloud users increases day by day, data centers continue to grow in terms of hardware resources, virtual resources and traffic volume, thus making cloud operation and management more and more complex. Hence there is a need of an efficient and effective Cloud Monitoring System (CMS). Cloud monitoring is the process of reviewing, controlling and managing the operational workflow and processes within a cloud infrastructure. It is the use of manual or automated IT monitoring and management techniques to ensure that a cloud infrastructure or platform performs optimally [2]. The cloud monitoring is a key tool for controlling and managing cloud infrastructure by collecting information from different probes, aggregating related information, filtering the unrelated/unwanted data and finally analyze or evaluate the performance of the cloud. It also takes control actions in the interest of performance improvement in cloud.

CMS [3] helps to manage the performance of cloud especially when consumer adapted critical services or scientific applications. For instance, if a consumer may host application at multiple clouds to ensure high availability, then switching between multiple clouds becomes easy with help of CMS. Effective use of cloud computing platforms, management of their high complexity and ensuring appropriate levels of Quality of Services (QoS) [79] require specialized CMS. Further, CMS supplies information like workload, QoS parameter, Key Performance Indicator and status of resource usage to both consumers and providers. This helps in maintaining transparency between provider and consumer regarding billing [5].

1.1 General view of a Cloud Monitoring System

Cloud computing has three different service layers that are offered as services. These are Infrastructure as a Service, Platform as a Service and Software as a Service. CMS monitors various elements at each layer as mentioned below [4]:

1) Infrastructure layer: The CMS monitors all physical and virtual components of cloud such as CPU, memory, disk, virtual resources, network traffic etc.
2) Platform layer: The CMS monitors various services and platform (operating system) related metrics like response time, application startup time, number of process, number of VM , number of threads and resources per application etc.

Manuscript published on 30 September 2019
* Correspondence Author
Mahantesh N. Birje*, Center for Post Graduate Studies, Visvesvaraya Technological University, Belagavi, Karnataka, India. (E-mail mnbirje@yahoo.com)
Chetan Bulla, Dept. of CSE, KLE College of Engineering & Technology, Chikodi, Karnataka, India. (E-mail chetan.bulla@klecet.edu.in)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

3) Application layer: The CMS monitors different metrics which provide status of service performance and other service specific information such as average CPU utilization, query latency, CPU usage and memory usage etc.

The Figure 1 shows CMS based on three layers of a cloud. An agent is a small software program that is installed on each level of cloud to be monitored. The agent's primary functions are to collect data on user activity from different probes, and to transmit the data to a central server for processing, analysis and storage. A probe is an action taken or an object used for the purpose of learning something about the state of the network or hardware component. Monitoring is used to measure the usage of all resources based on different metrics with different granularity, according to the type of service and price model adapted. CMS should filter collected data to remove unwanted, noisy, unrelated data and aggregate them for analysis purpose. The analyzed data will be helpful for both cloud user and service provider. The cloud user and the provider have different perspective of cloud monitoring [6]. The cloud user's perspective is to focus on the QoS, economical and availability of cloud services offered by the CSP. And, the cloud provider's perspective is to focus on the effective resource utilization and the performance enhancement of cloud system.

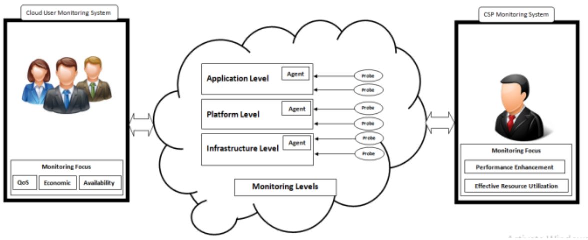

Figure 1: General view of a Cloud Monitoring System

The rest of the paper is organized as follows: (2) Taxonomy of CMS (3) State-of-the-art works on cloud monitoring, (4) Challenges/ Issues and Future directions and (5) Conclusion remarks.

II. BASICS OF CLOUD MONITORING SYSTEM

This section describes basics of CMS considering different aspects of a cloud such as monitoring architecture, monitoring phases, cloud properties and functions. The Figure 2 shows a typical CMS.

2.1 Architectures of CMS

There exist two types of CMSs based on architecture: Centralized and Decentralized architecture. The centralized architecture consists of a single monitoring server that collects metrics from different nodes and stores the data in centralized storage for further processing. The centralized architecture is low in cost but leads to two problems: it suffers from single point of failure and lack of scalability [6] [67]. To avoid these problems decentralized architecture is used, where monitoring task is distributed across various nodes of the cloud. The monitoring architecture can be developed either by using agent based or agentless system.

2.2 Monitoring Phases of CMS

The monitoring activity of CMS is divided into five phases: i) Data collection ii) Data filtering iii) Data aggregation iv) Data analysis and v) Alert and reporting. Figure 3 shows these phases of cloud monitoring activity. This section briefly defines these phases. Comparative study of state of the art works on these phases is given in Section 3.1.1.

2.2.1. Data Collection

The monitoring system must collects the different types of information or metrics like processing time, processing speed from CPU probe, memory utilization, latency from memory probe, energy consumption, energy utilization from energy probe, bandwidth, delay, latency etc from different probes. Figure 4 shows the taxonomy of data collection considering main aspects of data collection such as architectures, strategies, update techniques and categories.

The data can be collected by using centralized or decentralized architecture. The centralized architecture uses data collection trees where a monitor server sits at the root of the tree and is supported by lower level servers which propagate state from monitoring hosts. Failure of a monitoring server will interrupt data collection process from its sub-tree. The centralized data collection system is suffering from various challenges like single point of failure, performance degradation, replication and fault tolerance [30]. To overcome these challenges, the monitoring system must adopt decentralized system. The decentralized system makes use of peer-to-peer concept. The system uses either agent based data collection or agent less data collection. In agent based data collection, the agents are installed at different component of cloud to collect and send the data to central server. In the agentless CMS, data collection is easy and cheaper in cost, as it is not essential to install software agent [85]. The agentless services, talk directly to the underlying cloud platform through the service provider's API to collect the data.

Strategy: To collect the data from cloud five strategies are identified in literature: i) Push ii) Pull iii) Hybrid push, iv) Hybrid push pull and v) Adaptive push. The push method pushes the information from end-components of cloud to central server. In the pull method, central server query end-component of cloud to send the required information [9]. The Hybrid push method pushes the data or information to the central node based on fixed time interval or based on event which occurs in the cloud [2]. In the hybrid push-pull method an entity pull the data from external entities and pushes the same data to central node [10]. The adaptive push collects the data from different probe and stores them in window [11]. At the next step, it pushes the data to central server only if updated data is found in window.

Update techniques: The data in the cloud is updated very frequently. So it is essential to consider updated data for analysis. Four update techniques are identified in the existing work. First, the data which is updated at regular interval is called periodic update [12]. Second, the data is collected when some event occurred in the cloud is called event based update [13].

Third, the new data will be updated only when there is a change in previously collected data is called content based update [14]. Fourth, the window based system store the data in window till next data arrives, and then compare window data with newly arrived data. If there is change in the data then only the data is sent to central server.

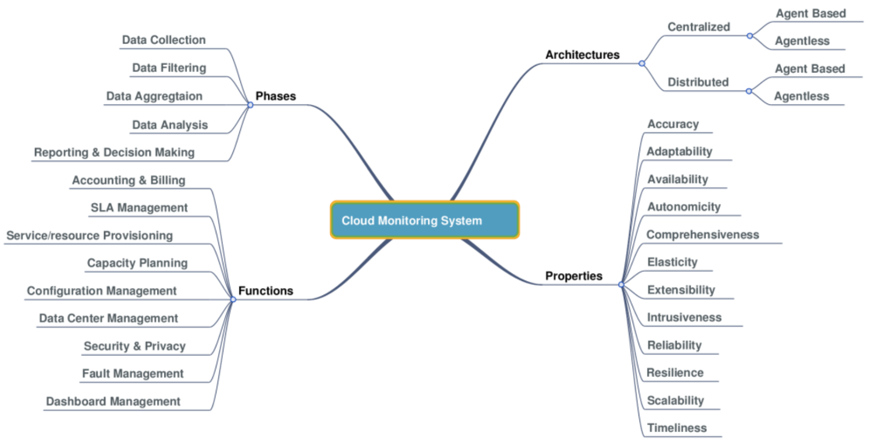

Figure 2: CMS-architectures, phases, functions and properties

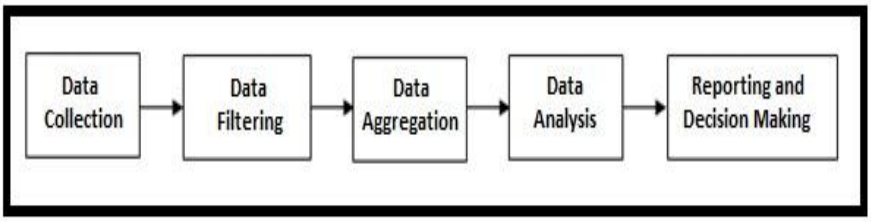

Figure 3: Five phases of cloud monitoring activity

Figure 4: Taxonomy of Data Collection Techniques

2.2.2. Data Filtering

The collected information may contain high percentage redundant, invalid, conflict and irrelevant data. It is essentials to filter these data. A filtering algorithm could be more efficient to deliver more relevant information or data. Data cleaning is very important activity to remove unwanted data from cloud system. Filtering may reduce the impact of monitoring data transfer over the network load and increase the performance of cloud. Figure 5 shows the taxonomy of data filtering. The filtering techniques can be applied over data, resource status, and computational status.

There are four types of filtering methods are found in existing work. These are time based filtering, window based filtering, content based filtering and threshold based filtering. The time based filtering attempts to make filtering data frames by time [13]. In other words, it filters by collecting data frames at particular time period. The window based filter collects the amount of data as per the window size and apply filtering technique for the data that is available in window [38]. The Content based filtering assign the rank and threshold values to collected data. The highest rank within threshold values are only accepted as useful data [14]. The threshold based filtering set some threshold values based on certain criteria, then the data within the threshold values are accepted [15]. The most popular filtering methods are Data preprocessing, Data deduplication, Data Compression and Dimension reduction [16]. The data compression method keeps original data in the compressed form. The data deduplication method removes the redundant data in the cloud. Data

Retrieval Number C6857098319/2019©BEIESP
DOI: 10.35940/ijrte.C6857.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

preprocessing is the important phase of big data processing, and it filters the unrelated data at large-scale infrastructures. Dimensionality reduction is the process of reducing the random variables by obtaining a set of principal variables. The CMS collects two types of data where the filtering can be applied, these are: resource statistics and data. The data filtering can be implemented either by agent based system or agentless system.

2.2.3. Data Aggregation

The data aggregation is a process in which information is collected and articulated in a summary form for statistical analysis. The data aggregation reduces the network traffic and secure private data [17]. The data aggregation can be done in the regular interval depending upon the requirement of application. The data aggregation can be implemented in cloud monitoring by adopting data mining techniques like clustering and classifications.

Figure 5: Data Filtering Techniques

2.2.4. Data Analysis

Once data has been aggregated it must be processed and analyzed in order to extract the useful information. Data analysis is a process of inspecting, transforming, and modeling data to discover useful information that will help in decision making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of domains like business, science, and technology [36]. The data analysis is used to improve the performance by identifying present resource status, predicting future status and detecting critical conditions & abnormal conditions.

2.2.5. Data Reporting and Decision making

Using data analysis, a complete report of cloud status will be generated. The report will be in graphical representation and descriptive format that tells about what is status of cloud at particular point in time. This analysis report will be used to take the control action to improve the performance of cloud or resolve the issue. If particular resource is in critical situation like overloaded or memory leak etc. then certain notification will be sent to the admin in the form of email or raising alarm.

2.3. Properties of CMS

In order to perform the smooth operation, the CMS must have several properties [8]. In Figure 1, the properties of cloud in taxonomy of CMS aspects are reported.

1. Accuracy: The CMS should provide accurate measures that are as close as possible to real values.
2. Adoptability: The CMS must support dynamic nature of cloud as requirements, technology and resources (type, size, structure) of cloud are changing dynamically.
3. Autonomic: The CMS can able to manage its resources automatically without human interactions.
4. Availability: The CMS should be available all the time for all users of cloud by adapting fault tolerance techniques.
5. Comprehensiveness: The CMS must support heterogeneous resource (physical and virtual), variety of data and multiple tenants.
6. Elasticity: The size of cloud increases and decreases dynamically based on its usage. The monitoring system should support scaling up and scaling down of cloud resources.
7. Extensibility: The functionality of CMS should be extending as per new requirements of user.
8. Intrusive: The CMS should do significant modifications as the changes in the requirements in terms of security.
9. Scalability: Scalability means cloud computing should support even if the data, parameters and measures increases with changes in the requirements.
10. Timelineness: The CMS should respond within time limit. It should not take more time to send requested data.
11. Resilient: A monitoring system must be persistent to the dynamically changing cloud requirements.
12. Reliable: A monitoring system perform required task at any point of time under stated conditions.
13. Portable: Cloud environment must able work with heterogeneous platforms and services.
14. Multi tenancy: The CMS should maintain concurrency in serving different data or information to many users at a particular point in time.
15. Customizability: CMS should maintain customizability for all the operations of cloud to cope up with requirements changes.

2.4. Functions of CMS

The cloud computing involves many activities for which monitoring is an important task. Functions of these activities are:

1. Accounting and Billing: Accurate accounting and billing information is very important for both customer and service provider [14]. The functions of accounting and billing are:
   a. Collecting accurate IT resource usage data.
   b. Measure the quantity and nature of runtime IT resource usage activity. Based on the measurement and SLA, Bill has to be generated [77].
   c. Protect Account and Billing data against forgery and false modification.

2. SLA Management: A Service Level Agreement (SLA) is a contract between a cloud service provider and a customer that specifies service offering [33] [63]. SLA management functions are:
   a. Measure QoS parameters and resources usage.
   b. Store and Analysis resource data collected in monitoring system.
   c. SLA parameter assessment and verify SLA violations.

3. Data Center Management: A data center is a centralized repository, for the storage, management, data distribution and information organized around a particular part of knowledge [27] [57]. The CMS must support these data center functions:
   a. Keep track of desired hardware and software metrics.
   b. Keep track of resource utilization and accounting.
   c. Data analysis that helps in resource provisioning and troubleshooting.
4. Service / Resource Provisioning: The important property of cloud monitoring is elasticity where the resources allocated in dynamic nature of cloud [24]. The service or resource provisioning involves the optimal allocation of resources in order to match the workload requirement [54]. There are two types of resources provisioning: static and dynamic resource provisioning. The static resource provisioning creates and allocates fixed size VMs. The dynamic provisioning adjusts VM dynamically based on required size at runtime. The main functions of service/Resource Provisioning are: resource allocation permission based on Access Control list and Keep track load on each resource with time stamp.
5. Capacity planning: The capacity planning involves quantifying resources capacity and determines estimated workload [8]. It ensures resource availability to meet the capacity demand necessary for new customer requirements.
   a. Measure capacity usage and determine resource wastage.
   b. Predicting resource availability in near future.
   c. Determine the status of all physical and virtual resources.
6. Configuration Management: Configuration is arrangement of different set of parameter (QoS) and values that determines the behavior of resources and applications. In multi-tenant environment, customers have different requirements. Based on customer requirement cloud services are configured and provided to customer. If customer requirement changes at run time then cloud services are re-configured. Configuration management system verifies specified configuration and possible changes in cloud computing environment.
   a. Verifying the present configuration.
   b. Configurations drift identification.
   c. Reconfigure cloud services based present configuration and new configuration.
   d. Auditing the performance effect after re-configuration.
7. Security and Privacy management: The security is very important for consumer and service provider, as consumer store his or her private data and producer store accounting and billing data[23] [28]. The capability of detecting security breaches and attacks is essential for cloud and monitoring system can help in this regard by following functions
   a. Identifying malicious process that consumes more system resources.
   b. Enabling strict auditing and trust policies.
8. Fault Management: To provide resilient and reliable services over the cloud, fault management should be very strong [2][30]. Fault management involves identifying faults, find the root cause of the fault and troubleshoot the fault by collecting various related information. The functions of fault management are:
   a. Detection of failures and predicting of failures.
   b. Handle the failure by replacing alternative components.
   c. Find and troubleshot the fault.
9. Dashboard management: A dashboard is a real-time user interface showing a graphical presentation of the current status (snapshot), resource utilization and key performance indicators. It updates the information either periodically or event based.
10. Alerts and reporting: The user gets intimation regarding status of cloud resources. The intimation can be send by either email or sms service. Most of the alerts are generated through threshold analysis, where a particular component and its usage crosse

III. STATE-OF-THE-ART WORKS ON CLOUD MONITORING & RESULTS

This section presents state-of-the-art works on cloud monitoring activity. Cloud monitoring activity is mainly discussed in two approaches 1) Based on various phases of monitoring activity and 2) Based on usage of agents for monitoring.

3.1. Based on various phases of Monitoring activity

The monitoring activity consists of five phases: i) Data collection ii) Data filtering iii) Data aggregation iv) Data analysis and v) Alert and reporting.

These phases are briefly defined in the section 2.2.1. The comparative study of these phases is given in following sub sections. Note that we have not found much work on Alert and reporting phase; so our comparative study is limited to first four phases.

3.1.1 Comparative study of Data Collection methods:

Data collection is used to gather and measure the information from a variety of sources to get a complete and accurate data of interest. The CMS can collect variety of data like log messages, security events, application data, network flows, performance measures etc. The comparative study is carried out by considering different aspects such as architecture, communication strategy, data updation techniques and agent based system adoption. Table 1 presents the summary of this comparative study.

Architecture based Data Collection: data collection happens either through Centralized or Distributed architecture of CMS. These architectures are briefly defined in section 2.2.1. The Cloud security monitoring system [22] has adopted centralized data collection architecture to monitor the security aspects of cloud computing.

The resource monitoring framework [35] has used transportation window technique where collected data is analyzed to avoid duplicate data transfer from hosts to server. The above architectures are economical but leads to single point of failure. To overcome this problem, distributed data collection architecture [12] is used in cloud monitoring systems, where data collection task is distributed across the various nodes of the cloud. The most of existing data collection works have adopted distributed architecture to collect the data.

Communication Strategy based Data collection: The communication between cloud manager and cloud entities may follow push, pull, hybrid push, hybrid push pull, and adaptive push strategies [61]. These strategies are defined in section 2.2.1. The recent CMSs dynamically collect the status or information of various resources. The data or information is pushed from various resources to the centralized monitoring system and convert incoming monitoring data to a unified format, producing simplified meaningful monitoring data [32]. The main drawbacks of push method are: 1. collecting unrelated/unwanted data, 2. Consume more transmission bandwidth. To avoid these problems, pull method is used where central collector node pulls the required data by executing queries [33]. The main problem with pull method is that the central collector node should have knowledge to execute queries for the required data. The hybrid push method pushes the data to the central storage based on two strategies 1) periodic push and 2) event based push [2]. The periodic push consumes more communication bandwidth as it pushes the data in regular time interval even when cloud contains no or lesser operations. Alternatively, the event based push strategy pushes the data only when the change in cloud is detected for particular entity. The hybrid push-pull method [34] combines the basic push and pull model to resolve the above said disadvantages but cost of implementation is more. The adaptive push uses client/server architecture that collects the data from different probe and stores them in to transportation window [35, 11]. At the next step, it pushes the data to central server only if updated data found in transportation window. The mechanism gives better data coherency between hosts and monitoring server. It has been observed that event based hybrid push as it consumes less computational and communication cost.

Data updation techniques based Data collection: As the data in the cloud is updated very frequently, it is essential to consider updated data for performance analysis. The CMS may adopt one or more data updation techniques, these are threshold based, window based, periodic based and event based.

JCatascopia [78] has employed a pub/sub mechanism which is similar to basic pull model, but here monitor manager explicitly mention the required data to monitoring agent and the monitor agent send the same data to the monitoring manager. This technique minimizes network communication overhead but monitor manger may receive duplicate data when status of cloud unchanged. To resolve this issue, the window-based event-driven push strategy [61] pushes resource status information only if the change in resource status information is larger than a threshold.

To increase the performance, resource state monitoring [36] has adopted Markov Chain Model to predict the future resource state. The proposed model updates the information, when the difference between predicted values and actual value exceed Error Tolerance Degree. These updates are used to tune the prediction model.

Agent based Data collection: The agent based CMS has lot of advantages over agentless CMS. Most of the works in the literature used agent based data collection architecture. JCatascopia [38] is an agent based multi-level cloud monitoring system used to collect heterogeneous metrics of different granularity across multiple levels of the cloud. The proposed model follows an agent based approach where the agents calculate performance metrics from collected data and transfer them to storage server. JCatascopia achieve portability as agents are capable of running on any physical machine or VM instances. The JTangCMS [31] is the agent based cloud monitoring system where agents are used in data dissemination framework to collect the runtime information from various entities from different levels of the cloud periodically. This runtime information will be encapsulated into groups and delivered efficiently and timely to the server by Data Distribution Service. Rest of the state-of-the-art architectures has used agent based data collection method.

Table 1: Comparative study of Data collection methods in existing CMS.

References	Architecture		Communication Strategy					Data Updation Technique				Agent based
	Centralized	Distributed	Push	Pull	Hybrid push	Hybrid push-pull	Adaptive Push	Windows Based	Periodic	Event based		

[2]		✓	✓	✓	✓					✓	✓
[3]		✓		✓							✓
[9]		✓	✓	✓	✓						
[10]		✓				✓			✓	✓	✓
[12]		✓							✓		✓
[13]		✓							✓		✓
[22]	✓						✓			✓	✓
[31]		✓		✓					✓		✓
[32]		✓	✓								
[33]		✓		✓						✓	✓
[34]		✓				✓					✓
[35]	✓						✓	✓	✓		
[36]		✓	✓		✓					✓	
[38]		✓			✓					✓	✓
[58]		✓	✓	✓							✓
[61]		✓	✓		✓			✓			
[62]		✓			✓					✓	✓
[80]		✓				✓				✓	

The traditional push based data collection method collect and updates the data periodically. This leads to unnecessary pushing the same and increases the communication and computational cost. To minimize the these communicational and computational cost, update the data when there is no change in data that is last updated. Wu-Chun Chung and Ruay-Shiung Chang [80] have proposed GRIR (Grid Resource Information Retrieval), which is considered a new algorithm to improve Push model in grid monitoring system. Three different algorithms were proposed, such as the OSM (Offset- Sensitive Mechanism) protocol, the TSM (Time-Sensitive Mechanism) protocol, and the hybrid ACTC (Announcing with Change and Time Consideration) protocol. This hybrid protocol dynamically adjust update time interval and consider for quick update when the change is larger than a dynamic threshold.

Han Fang-Fang, Peng Jun-Jie, Zhang Wu, and et al [81] have proposed a periodically and Event-driven Push (PEP) monitoring algorithm. The algorithm proposed by combining the advantages of the push and event-driven mechanism and simplifies the communication cost between the consumer and the producer. The proposed model never misses the important updates which would be happened during the push interval. This algorithm light weight and provide more adequate and information.

H. Huang and L. Wang [82] have presented a hybrid resource monitoring algorithm for Cloud computing called “P&P”. This algorithm takes the advantages push and pull Models for resource monitoring in the Cloud Computing Environment. The proposed model can intelligently switch between Push and Pull models and minimizes the number of updating according to the requirements of the users. This algorithm reduces the updating rate coherence in accordance with the users’ requirements.

3.1.2 Comparative study of Data Filtering methods:

The filtering methods are basically used to optimize the performance by reducing unwanted data from the cloud [38]. The reduced and relevant data are more useful than collecting raw, duplicate, inconsistent, and unwanted (noisy) data. It saves processing power, increases the speed of execution and minimizes the network bandwidth consumption [16]. Table 2 shows the comparative study of existing work with respect to data filtering. The comparative study includes filtering techniques adopted, type of data to be filtered and agent based or agentless filtering. The techniques to implement the filtering mechanism are window, time, content and threshold based (discussed in 2.2.2). There are two possible types of data that needs to filter: data and resource status.

Window based data filtering: The resource monitoring system [62] has adopted an auto window filter algorithm to reduce the network traffic between node to cluster and cluster to manager in cloud computing. The window size is flexible in nature and resize automatically as the size of data grows or shrinks. The monitoring node will accept the data only if it is within the window range and threshold value. The threshold values are static but if system is upgraded or downgraded then it is difficult to get accurate values. To resolve this issue, JCatascopia [38] has adopted window based adoptive filtering algorithm where window range depends on the percentage of values previously filtered. The algorithm calculates filter_value_percentage. If calculated value is lesser than target_filter_percentage then size of window increases otherwise decreases. The value within this window range will be considered for further processing. The improved version of JCatascopia [78] has adopted metric filtering mechanism to reduce the communication overhead that occurs during metric distribution and storage. This mechanism introduced filters at probe level to allow the users to attach filters to metrics. At runtime, the filter mechanism checks collected metrics are in the appropriate range. The metrics within the range [prevValue−F, prevValue+F] will be discarded in place rather than being distributed through the network.

Time based data filtering: DARGOS [13] employed data filtering techniques aimed to reduce monitoring overhead by eliminating unnecessary monitoring data distributions. The first mechanism is a time-based filter used by Node Supervised Agents (NSA) to reduce the number of updates accepted within an assured period of time from a given resource. For example, an application requires only one measurement per minute can limit the reception rate. The second mechanism is a value-based filter that permits NSAs to be notified of updates only if the resource usage status has changed since the last update. This strategy avoids unnecessary traffic in scenarios where an approximate estimation of resource usage is more than satisfactory.

Content based data filtering: The Relief algorithm [39] is a feature weight-based algorithm, which works on relevance evaluation of each feature given in a training data set in which samples are labeled. The main task is to compute a ranking score for every feature indicating how well present feature differs neighboring samples. The low ranking data will be discarded as author claims that it is unrelated data in the cloud. The algorithm is not scalable as it consumes more computational time to calculate ranking for large data.

Threshold based data filtering:

The threshold based filter [15] has operates periodically with samples gathering from various agents in the cloud. These samples are stored in a local database and send on demand to the manager.

The proposed model transfers the monitoring data that are within specified threshold range and these threshold range can be refined or adjusted aiming to reduce the network load caused by large amount of monitoring data.

3.1.3. Comparative study on Data Aggregation methods

The aggregation is a method to join the similar type of data into a group of information. The filtered data can be aggregated based on related data to improve the effectiveness and performance of cloud [73]. Data aggregation can be done either periodically, event based or when data changes in cloud. The Cloud monitoring architecture [15], has adopted data aggregation to improve the cloud performance, where the main function of aggregation is organizing data of the same type originating from several agents that are spread in a cloud computing. Multi-clouds based application monitoring architecture [40] has used aggregation method to reduce the communication overhead between multiple clouds. The collected data is aggregated to provide a comprehensive view of applications and send to the central manager. To reduce the number of collected messages, network traffic in each virtual machine, the data is aggregated into 5-tuple flows (e.g.

Table 2: comparative study of data filtering techniques in existing work.

	Data Filtering Techniques				Data Types		
References	Window Based	Time Based	Content Based	Threshold Based	Data	Resources Stats	Agent Based
[62]	✓			✓	✓		
[13]		✓	✓			✓	✓
[55]	✓				✓		
[39]			✓		✓		
[14]			✓		✓	✓	✓
[15]				✓		✓	✓
[38]	✓				✓	✓	✓
[2]	✓				✓		✓

Table 3: Data analysis technique in existing work

Ref.	Model	Cloud architecture	Analysis algorithms	Application	Agent Based	Analysis method
[43]	Varanus	Distributed	Load analysis-Gossip Protocol	Load analysis	Yes	Underutilized VMs are selected for analysis function. The analysis method is not defined.
[41]	D-CEP4CMA	Hybrid	Query Based Analysis	Behavior analysis	No	Calculate z-score to detect outliers. If z-score reaches some threshold value then its outlier.
[39]	Taguchi	Hybrid	Relief algorithm	Performance analysis	No	Compute a ranking score for every feature indicating how well current feature differs neighboring samples.
[44]	Trace analyzer K-means Clustering	Distributed	Holt-Winters Triple Exponential Smoothing, threshold based filtering.	Performance analysis	No	K-means clustering is used to classify data to identify related data. This intern helps to analysis data.
[45]	SAMF	Distributed	Principle component analysis	anomaly detection	Yes	Calculates the correlations between different metrics to evaluate the importance of metrics, and then use PCA to compute and quantify the anomaly degree,
[46]	MAS-CM	Distributed	Statistical analysis with ANN and GA.	Load analysis	Yes	For all tasks the mean and standard deviation of the workload are calculated. Then, if the workload cross threshold value, the tasks is rejected.

Retrieval Number C6857098319/2019©BEIESP
DOI: 10.35940/ijrte.C6857.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 4: Differences between agent based and agent less monitoring systems

Module	Agent Based Monitoring	Agentless Monitoring
Analysis	In-Depth Analysis	No depth Analysis
Deployment	Harder to deploy	Easier to deploy
Cost	Expansive	Cheaper
Network Overhead	Lesser	More
Troubleshoot	Easier	Difficult
Communication Model	Push/pull/Hybrid	Push/Pull
Data collection	Continues	Event based
Security	High level security	Lower level
Automotive	More automotive	Lesser automotive
Capacity planning	Good for CP	Not Good for CP
Bandwidth consumption	Less-bandwidth consumption	More-bandwidth consumption

The JCatascopia [38] has used aggregation method which adopts pub/sub model to collect the metrics from various agents. Monitoring server processes the collected metrics forming a unified and composite metrics upon user request, aggregate them and group the metrics from various instances together. This mechanism reduces the network related communication overhead as the monitoring server avoids the constantly poll agents for new metrics.

The JCatascopia [78] has adopted aggregation function that uses user-defined policies like AVG, MEAN to aggregate the metrics. JCatascopia employed two policies for aggregation: Time-based policy and Volume-based policy. The time-based policy distributes collected metrics every X seconds and volume-based policy distributes metrics if message size exceeds X KB. Multiple aggregation policies can be applied together with policies being configured through the agent configuration file. But there is tradeoff between efficiency and accuracy in the proposed architecture.

3.1.4 Comparative study on Data Analysis methods:

Data analysis is a process of inspecting, transforming, and modeling data to discover useful information that will help in decision making. The data analysis will help to improve the performance of cloud [41]. To analyze data, it is essential to obtain statistical analysis of the relationships between the gathered performance parameters on the different layers of the cloud [42]. The data analysis is used for anomaly detection, outlier detection, abnormal behavior and performance analysis etc. The comparative study is based on various application of analysis like load analysis, behaviors analysis, performance analysis and anomaly detection. Table 4 presents the comparative study of data analysis used in existing work where it is categorized based on Model, Cloud architecture, Analysis application, Agent based and Analysis method.

3.2.1 Comparative study

This sub-section analyzes existing agent based CMS with respect to cloud monitoring phases and properties. These are defined in the section 2.2 and 2.3. The CMS is effective only if it supports more number of monitoring properties [8]. The comparative study is based on various cloud monitoring elements like resources monitoring, performance monitoring, application monitoring and SLA monitoring. Table 5 shows the comparative study of existing agent based cloud monitoring with respect to monitoring phase and properties.

Based on resource monitoring:

JCatascopia is a fully-automated, multi-layer and interoperable monitoring framework [2] that supports maximum properties of CMS, (accuracy, adoptability, elasticity, scalability and portability) and adopted the data collection, filtering and aggregation phases. In the proposed framework, monitoring servers exchange periodically gossip messages that contain the network location and status of agents assigned to each peer. When a monitoring server is down, the respective seed nodes will rebalance the topology by requesting from the monitoring agents assigned to the faulty server to reconnect to other monitoring server(s) based on the monitoring agent placement policy. Varanus [43], a highly decentralized monitoring system used to monitor the performance of large scale cloud systems with a reduced need for dedicated monitoring infrastructure. The monitoring agent participates in a gossip based overlay network. Using this overlay network, monitoring agents broadcast the status to all other agents. The proposed model supports adoptability, comphrensiveness, scalability and multi-tenancy. It does not provide a full, comprehensive monitoring suite.

Load Analysis: The Cloud monitoring system [46] has adopted agent based artificial intelligence techniques to monitor the cloud. It uses Artificial Neural Networks (ANNs) and Genetic Algorithms (GAs) combined with the statistical analysis of the tasks workload to take the allocation decisions. For all tasks the mean and standard deviation of the workload are calculated. The workload of the particular task is below or above the certain level, the tasks are returned to the user. The result shows that agents can take action in non-deterministic time. An agent based multi-tier cloud monitoring architecture [43] has adopted the strategy to balance the VM load by collecting and analyzing the various parameters of large scale cloud services. The proposed method allocates analysis process to least loaded VM to utilize the resources effectively. If all the VM are heavily loaded then dedicated VM is provided. The author has not defined analysis method.

Behavior analysis: D-CEP4CMA [41] has proposed a cloud monitoring and analysis model based on complex event processing.

The analysis part is used for outlier detection with help of z-score calculation. The outlier detection in the proposed architecture has used robust statistics and it consists of two phases: training period and z-score calculations. During first phase, the outlier detector has trained the data by computing the values of the median and the mean absolute deviation of its inputs. The second phase is used to compute the z-score of the incoming monitored data (xi). The z-score of xi is the difference between xi and the median, if the absolute value of the z-score exceeds some threshold value then it is outlier otherwise it is accepted value.

Anomaly detection: A self adoptive monitoring system [45] has adopted the framework to automate the monitoring task by adjusting monitoring metrics, periods and frequency. The important monitoring metrics are selected by applying correlation analysis between different metrics and these metrics will give running status of cloud resources. Next, it characterizes the status using principle component analysis [42] to estimate the anomaly degree and predict possibility of faults. The proposed approach can effectively improve the accuracy and timeliness of anomaly detection and reduce the monitoring overhead.

Performance analysis: The Performance Analysis Model (PAM) [39] is used to measure and evaluates performance metrics in big data application. The proposed method works on three service concepts: 1) service not avail 2) service done correctly and 3) Service done incorrectly. The PAM consists of two main steps 1) identify degree of relationship 2) mapping performance concepts. The degree of relationship between performance concept and sub concept is identified by collecting various performance measures from MapReduce log files and mapping them with performance concepts defined in the monitoring framework by means of the formulae defined in the ISO 25023. The proposed model can be optimized and used in CMS to increase the effectiveness of performance analysis. The trace analyzer [44] is used for analyzing cloud monitoring data by deploying the REpresentational State Transfer (REST) API. It includes three modules: search package, analysis package and report package. The search package queries the time series data and classify them using k-means algorithm. The analysis package has applied Kernel Density Estimation on classified pattern to determine the workload pattern and detect anomalies by applying threshold based filtering.

The monitoring framework [28] has adopted agent based system that allows the user to check the status of his or her heterogeneous resources. The agents are used to collect and measure the performance metrics. The proposed framework has offered high elasticity, scalability and extensibility by the providing a high level customization of the performance indexes and metrics. The proposed work considers limited parameters and may degrade the performance for current cloud scenario. JTangCMS [31] is an efficient and robust data dissemination framework used to transfer the huge amount of runtime information reliably with high throughput and low latency. The agents are installed on each virtual machine and physical machine to receives, stores, and process all this runtime information received from the other agents. An effective and intelligent cloud action platform has also developed to support decision-making for cloud management based on Complex Event Processing (CEP). The proposed CMS supports accuracy, adoptability elasticity and scalability. The rest of the resource monitoring systems support very few properties of cloud and needs to optimize these work to gain high effective utilization.

Based on performance monitoring: Self-adaptive performance monitoring framework [45] contains five autonomic elements: monitor, analyzer, planner, executor and knowledge base. It will automatically fine-tune the monitoring period to improve the accuracy and timeliness of anomaly detection and reduces monitoring overhead. The proposed monitoring method supports adoptability and availability properties. The performance of the proposed work can be improved by applying filtering and aggregation on collected data. The new monitoring framework has adopted Multi-Agent System [46] where agents can perform resource scheduling, task execution, and performance analysis. The proposed model mainly focused on scheduling process optimization, effective resource utilization and prevents the unauthorized tasks injection and modification. Agents are deployed in each part of the cloud system and these are capable of learning cloud environment and taking self-decisions using AI. The proposed model supports accuracy, elasticity, autonomic, intrusiveness and portability.

Based on application monitoring: The monitoring architecture [51] relies on a modular structure that is composed of three parts: 1. Infrastructure definition module, 2. Setup and management module and 3. Data processing module. The first two parts are used to setup the infrastructure and collect the data from various agents that are scattered throughout the cloud. Third part is data analysis part used to configure and the triggering of events based on some conditions. The Complex event processing used to check SLA fulfillment and distributed attack detection.

Two adaptive algorithms, the Check Period Relaxation (CPR) algorithm and the Modified-CPR algorithms have been deployed to efficiently manage the communication between the agents that monitor the web service. JCatascopia [38] is an automatic resource provisioning for Cloud applications using metric Subscription Rule Language, which consists of two operations: (i) aggregate and group low-level metrics originating from single instances, and (ii) generate high-level metrics dynamically at runtime from low-level metrics. The JCatascopia is capable of supporting a fully automated cloud resource provisioning system with proven interoperability, scalability and low runtime footprint.

Based on SLA monitoring: The main idea behind autonomic resource management system [53] is to reduce SLA violation rate for delivery of cloud services. The proposed model reserves the resources that are used when SLA violation rate is more than threshold value. Further, different QoS parameters such as execution time, cost, latency, reliability and availability etc. are used to analyze the impact of QoS parameters on SLA violation rate are also considered.

Based on SLA, SLA manager prepares SLA document which contains information about SLA violation rate and high priority workloads that needs to execute immediately.

Table 5: comparative study of existing agent based CMS

		Monitoring Phases				Properties Supports																
Reference	Monitoring elements	Data Collection	Data Filtering	Data Aggregation	Data Analysis	Accuracy	Adoptability	autonomic	Availability	Comphrensiveness	Elasticity	Extensibility	Intrusiveness	Scalability	Timeliness	Resilient	Reliable	Portable	Multi tenancy	**Customizability**
[43]		✓		✓	✓		✓			✓				✓					✓	
[27]										✓				✓			✓			
[12]		✓	✓			✓											✓		✓	
[28]											✓	✓		✓						✓
[14]		✓		✓					✓									✓		
[47]	Resource	✓					✓							✓			✓			
[31]		✓			✓	✓	✓				✓			✓						
[2]		✓	✓	✓		✓	✓				✓			✓				✓		
[48]		✓			✓						✓			✓			✓	✓		
[49]			✓											✓	✓					
[39]		✓	✓		✓							✓		✓				✓		
[50]	Performance					✓	✓											✓		
[45]		✓			✓		✓		✓											
[46]					✓	✓		✓	✓				✓				✓	✓		
[32]		✓		✓								✓		✓						
[38]	Application	✓	✓								✓			✓			✓			
[51]						✓	✓											✓		
[53]	SLA							✓									✓	✓		

IV. CHALLENGES / ISSUES AND FUTURE DIRECTIONS

Cloud computing has achieved successes and growth during last decade. The cloud monitoring is still facing several challenges/issues due to the complex nature of cloud infrastructure such as architecture complexity, computational and network workload, volume of monitoring parameters, and dynamic computing environment. In this section, cloud monitoring solutions have been surveyed to understand the technical aspects of existing cloud monitoring system and also investigate the challenges/issue. Table 6 show challenges/issue with possible future directions. The main issues/ challenges are : Performance issue, Maintenance challenges, agent cost, Energy aware, Adoptability, Interoperability, Scalability, Robustness, Charging and metering Issues and SLA Issues.

V. CONCLUSION

Cloud monitoring system supervises and manages the operational work-flow and processes within cloud computing environment to ensure its performance capacity and capabilities. In this paper, the basic concepts of CMS such as monitoring architecture, monitoring phases, cloud properties, and monitoring functions are described. Further, a comparative study of state-of-art cloud monitoring works based on various phases of cloud monitoring activity and usage of agent for monitoring is also given. The challenges/issue and possible future directions are highlighted to carry out further research in the field of cloud monitoring.

Table 6: Challenges/issues and future directions

Sl. No	Challenges /Issues	Description	Possible Future Directions
1	Performance	**Data Filtering**: Huge data processing in cloud involves many challenges relevant to inefficiency, parallel memory bottlenecks, and deadlocks. So to avoid these problem it is essential to filter unrelated, noisy and invalid data [6][69].	• Adopting big data filtering approaches to reduce unrelated/unwanted data. • Development of new data reduction algorithms to suite complex cloud structure.

			• Policy based polling frequency needs to introduce in CMS.
		Data Analysis: There is less work available w.r.t to analysis part of monitoring process. These analysis methods are traditional and may not suit for present or future cloud.	Multi-resolution analysis allows the application to analyze the data in stages, starting with a larger scope of analysis with coarse-grained filters of spatial and temporal dimensions.
2	Maintenance	**Root Cause Analysis**: It is very difficult to find the root cause of problem in cloud system. The In depth study of cloud is essential to find the root cause of problem.	An effective root cause analysis method has to be developed to find the root cause of the problem within short period of time.
		Proactive Monitoring: The existing CMS are reactive in nature [69]. So it is difficult to optimize the performance in critical conditions.	The CMS needs to adopt prediction techniques that alert users and cloud manager before it crashes. The machine learning techniques are the best solution for predicting status of cloud.
3	Agent cost	**Light Weight Agent:** The agent is heavy weight process, so it consumes more computational cost [6].	It is essential to optimize the agent so that they consume less computational cost.
4	Energy aware	**Energy Efficient:** It is essential minimize energy consumption in CMS. Existing CMSs not concentrate on reducing energy consumption in cloud [6] [56].	The high resource usage rate in cloud consumes more energy. The monitoring data can be utilized to develop new policies that reduce energy consumption.
5	Adoptability	**Configuration:** The requirements of customer are changing rapidly in cloud usage. To fulfill customer requirements, cloud has to dynamically reconfigure the resources.	To improve the adoptability, it is essentials to develop self-configurable CMS. The agent based CMS with AI is the best solution.
6	Interoperability	The difficulties in monitoring hybrid cloud performance are : lack of visibility to predict resource utilization, difficulty in supporting dynamic environment and add overhead in using multiple tools to manage	The standard interoperable tools or protocols have to be developed to resolve the interoperability issues.
7	Scalability	Collecting the more number of performance metrics for large number resources are difficult to manage. Most of existing work tried to improve the scalability. But still it is essential to optimize for highly responsive application are hosted in cloud [68, 6].	The CMS must capable to handle large volume of data by adopting data mining techniques with machine learning algorithms.
8	Robustness	The launch time of application and reconfiguration take more time. [6].	The robust CMS has to be developed to reconfigure the resources within short period of time and reduce the launch time of application.
9	Charging and metering Issues	There are some issues in implementing usage based charging policy. For example, a new set of metric data collection for new fine grained billing policy will be required. This will leads collecting same metrics multiple times for different purpose [6].	A special care must be taken while collecting the fine-grained metrics to avoid to multiple copies.
10	SLA Issues	Now days a user-side SLA management has increased due to false penalty from service providers. It remains a challenge for researchers to provide a user-satisfactory SLA management policy at cloud user-end [6].	The CMS must adopt new policies that correctly identify the requirements and give appropriate service on priority basis to both user and service provider.

REFERENCES

1. Ahmed Mohammed Fahad, Abdulghani Ali Ahmed, "The Importance of Monitoring Cloud Computing An Intensive Review", 10 TENCON confeence, IEEE 2017.
2. Mingwei Lin, Zhiqiang Yao, Tianqiang Huang, "A hybrid push protocol for resource monitoring in cloud computing platforms" , 2007–2011,Optik 127, Elsevier, 2016, 2007–2011.
3. Xuan Liu, FengXu, "Cloud Service Monitoring System Based on SLA", ISDBAES, IEEE, 2013.
4. Jesus Montesa, Alberto Sánchez, Bunjamin Memishic, Maria S. Perez, Gabriel Antoniu, "GMonE: A complete approach to cloud monitoring", Vol 29, Issue 8, Future Generation Computer Systems, Elsevier, 2013, 2016.
5. R. H. Goudar, Manisha T. Tapale, Mahantesh N. Birje, "Price negotiation for cloud resource provisioning", International Conference On Smart Technologies For Smart Nation (SmartTechCon), IEEE, 2017.
6. Hassan Jamil Syed, Abdullah Gani, Raja Wasim Ahmad, Muhammad Khurram Khan, "Cloud Monitoring: A Review, Taxonomy, and Open Research Issues", Journal of Network and Computer Applications, Elsevier, 2017, 11-26.
7. Khalid Alhamazani, Rajiv Ranjan, Karan Mitra, Fethi Rabhi, Prem Prakash Jayaraman, Samee Ullah Khan, Adnene Guabtni, Vasudha Bhatnagar, "An overview of the commercial cloud monitoring tools research dimensions, design issues, and state-of-the-art", Volume 97, Issue 4, Journal of Computing, Springer, 2015, 357–377.
8. Giuseppe Aceto, Alessio Botta, Walter de Donato, Antonio Pescape, "Cloud monitoring: A survey", Journal of Computer Networks, Elsevier, 2013.
9. J. A. Perez-Espinoza, Victor J. Sosa-Sosa, J. L. Gonzalez, "Distribution and load balancing strategies in private cloud monitoring", 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), IEEE, 2015.
10. Mauro Andreolini, Michele Colajanni, Marcello Pietri, Stefania Tosi, "Adaptive, scalable and reliable monitoring of big data on clouds" Journal of Parallel Distributed. Computing, Science Direct,79–80, 2015.
11. Gokce Gorbil, David Garcia Perez, Eduardo Huedo Cuesta, "Principles of Pervasive Cloud Monitoring', Information Sciences and Systems, 117-124, Elsevier, 2014,
12. Mauro Andreolini, Marcello Pietri, Stefania Tosi, Andrea Balboni,"Monitoring Large Cloud-Based Systems", 4th International Conference on Cloud Computing and Services Science, ACM, 341-351, 2014.
13. Javier Povedano-Molina, Jose M. Lopez-Vega, Juan M. Lopez-Soler, Antonio Corradi, Luca Foschini,"DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds", Future Generation Computer Systems, Elsevier, 2041-2056, 2013.
14. Kwang Mong Sim, "Agent-based Approaches for Intelligent Intercloud Resource Allocation", , 1-14, vol-1, Iss-99, IEEE Transactions on Cloud Computing, IEEE, 2016.
15. Guilherme da Cunha Rodrigues, Gl´ederson Lessa dos Santos, Vin´ıcius Tavares Guimaraes, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco, " An architecture to evaluate Scalability, Adaptability and Accuracy in CMSs", ICOIN, 46-51, IEEE, 2014.
16. Muhammad Habibur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, Samee U. Khan, "Big Data Reduction Methods: A Survey", Data Science Engineering, , 265-284, Springer, 2016.
17. Yi Wei and M. Brian Blake,"An Agent-based Services Framework with Adaptive Monitoring in Cloud Environments", IEEE, 2013.
18. Mahantesh N. Birje, Praveen S. Challagidad, R.H. Goudar, Manisha T. Tapale, "Cloud computing review: concepts, technology, challenges and security", Journal of Cloud Computing, Vol. 6, No. 1, Inderscience, 2017.
19. Arun Kumbi, Pavankumar Naik, Kirthishree C. Katti, Kiran Kotin, "A Survey Paper on Internet of Things Based Healthcare System", Vol5, Iss5, pp 1-4 Journal of Internet of Things and Cloud Computing, Science PG, 2017.
20. Di Stefano, G. Morana, D. Zito, "Scalable and Configurable Monitoring System for Cloud Environments", Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 2013.
21. Chuliang Weng, Qian Liu, Kenli Li, and Deqing Zou, " CloudMon: Monitoring Virtual Machines in Clouds". IEEE Transaction on computers, , pp.3787-3793, Vol. 65, No. 12, IEEE, 2016.
22. M. Kozlovszky, L.Kovács, M. Törőcsik, G.Windisch, S.Ács, D.Prém, Gy. Eigner, P.I. Sas, T.,Schubert, V. Póserné, "Cloud security monitoring and vulnerability management",17th International Conference on Intelligent Engineering Systems, IEEE, 2013.
23. Alina Oprea, Ata Turk, Cristina Nita-Rotaru,Orran Krieger, "MOSAIC: A Platform for Monitoring and Security Analytics in Public Clouds", , pp69-70, Cybersecurity Development, IEEE,2016.
24. Muhammad Usman, Aris Cahyadi Risdianto, JongWon Kim, "Resource Monitoring and Visualization for OF@TEIN SDN-enabled Multi-site Cloud", International Conference on Information Networking (ICOIN), IEEE,2016.
25. Long Li , Buyang Cao , and Yuanyuan Liu, "A Study on CEP-based System Status Monitoring in Cloud Computing Systems", vol1, ICIII, IEEE, 2013.
26. Bei Guan, Yanjun Wu, Liping Ding, Yongji Wang, "CIVSched: Communication-aware Inter-VM Scheduling in Virtual Machine Monitor based on the Process", ISCCGC, IEEE/ACM, 2013.
27. A.Meera, S.Swamynathan, "Agent based Resource Monitoring system in IaaS Cloud Environment",International Conference on Computational Intelligence: Modeling Techniques and Applications-CIMTA, Elsevier, 2013.
28. Rocco Aversa, Luca Tasquier, Salvatore Venticinque, "Agents based Monitoring of Heterogeneous Cloud Infrastructures", 10th International Conference on Ubiquitous Intelligence & Computing, IEEE, 2013.
29. Guilherme Da Cunha Rodrigues, Rodrigo N. Calheiros, "Monitoring of Cloud Computing Environments: Concepts, Solutions, Trends, and Future Directions", Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp378-383ACM, 2016.
30. Mohit Kumar Gokhroo, Mahesh Chandra Govil, Emmanuel S. Pilli, "Detecting and Mitigating Faults in Cloud Computing Environment", 3rd IEEE International Conference on Computational Intelligence and Communication Technology , CICT, IEEE, 2017.
31. Xingjian Lu, Jianwei Yin, Neal N. Xiong, Shuiguang Deng , Gaoqi He, Huiqun Yu, "JTangCMS: An efficient monitoring system for cloud platforms", Information Sciences, pp402–423Elsevier, 2016.
32. Michael Smit, Bradley Simmons, Marin Litoiu, "Distributed, application-level monitoring for heterogeneous clouds using stream processing", Future Generation Computer Systems, Elsevier, 2013.
33. S. Tharunya, M. Divya, K. L. Shunmuganathan, "A multi-agent based query processing system using RETSINA with intelligent agents in cloud environment",International Conference on Computing Technologies and Intelligent Data Engineering, IEEE, 2016.
34. Beniamino Di Martino, Salvatore Venticinque, Dimosthenis Kyriazis, Spyridon V. Gogouvitis, "A Comparison of Two Different Approaches to Cloud Monitoring Inter-cooperative Collective Intelligence: Techniques and Applications", Springer, 2014.
35. Kai Lin, Wiqin Tong, Xiaodong Liu, Liping Zhang, "A self-adaptive mechanism for resource monitoring in cloud computing", ICSSC, IEEE, 2013.
36. Khalid Alhamazani, Rajiv Ranjan, Prem Prakash Jayaraman, Karan Mitra, Meisong Wang, Zhiqiang (George) Huang, Lizhe Wang, Fethi Rabhi, "Real-time QoS monitoring for Cloud-based Big Data Analytics Applications in Mobile Environments", 15th International Conference on Mobile Data Management, IEEE,2015.
37. AwsNaser, Mohamed FadliZolkipli, Shahid Anwar, MunaSulieman Al-Hawawreh, "Present Status and Challenges in Cloud Monitoring Framework: A Survey", European Intelligence and Security Informatics Conference (EISIC), IEEE, 2016.
38. Demetris Trihinas, George Pallis, Marios D. Dikaiakos", JCatascopia: Monitoring Elastically Adaptive Applications in the Cloud", International Symposium on Cluster, , 226-236, Cloud and Grid Computing, IEEE/ACM, 2014,
39. Luis Eduardo Bautista Villalpando, Alain April and Alain Abran,"Performance analysis model for big data applications in cloud computing", Journal of cloud computing, Springer, 2014.
40. Do Le Quoc, Lenar Yazdanov, Christof Fetzer, "DoLen: User-side multi-cloud application", International Conference on Future Internet of Things and Cloud, pp76-81,IEEE, 2014.
41. AfefMdhaffar, Riadh Ben Halima and Mohamed Jmaiel, Bernd Freisleben, "D-CEP4CMA: a dynamic architecture for cloud performance monitoring and analysis via complex event processing", Int. J. Big Data Intelligence, pp89–10 Inderscience Enterprises Ltd, 2014.

42. Min Du, Student Member, IEEE and Feifei Li, "ATOM: Efficient Tracking, Monitoring, and Orchestration of Cloud Resources", IEEE Transaction on Parallel and Distributed Systems, VOL. 28, NO. 8, pp2172-2189,IEEE, 2017.
43. Jonathan Stuart Ward and Adam Barker," In Situ Monitoring for Large Scale Cloud Systems", International Conference on Cloud Computing Technology and Science, IEEE, 2013.
44. Samneet Singh, YanLiu, "A Cloud Service Architecture for Analyzing Big Monitoring Data", pp55–70, Tsinghua Science and Technology, IEEE, 2016.
45. Tao Wanga, Jiwei Xu, Wenbo Zhang, Zeyu Gu, Hua Zhong a, "Self-adaptive cloud monitoring with online anomaly detection", Journal of Future Generation Computer Systems, pp89-101, Elsevier, 2018.
46. Daniel Grzonkaa, Agnieszka Jak´obik, Joanna Ko lodziej, Sabri Pllana, "Using Multi-Agent System and artificial Intelligence for Monitoring and Improving the Cloud Performance and Security", Future Generation Computer Systems, pp1106-1117 Elsevier, 2018.
47. Rocco Aversa, Luca Tasquier, "Design of an Agent Based Monitoring Framework for Federated clouds", 30th International Conference on Advanced Information Networking and Applications workshops, pp115-121IEEE,2016.
48. Jieyu Lin, Byungchul Park, Qi Zhang,Hadi Bannazadeh, Alberto Leon-Garcia, "Scalable Monitoring Analytics Architecture in Software-Defined Infrastructure", Tridentcom-15, Vancouver, Canada, 2015.
49. Simin Cai, Barbara Gallina, Dag Nystrom, Cristina Seceleanu, Alf Larsson, "Design of CMSs via DAGGTAX: a Case Study",8th International Conference on Ambient Systems, Networks and Technologies, Procedia Computer Science, Elsevier, 2017.
50. Adil Maarouf, Mahmoud El Hamlaoui, Abderrahim Marzouk, Abdelkrim Haqiq, "Combining Multi-agent systems and MDE approach for Monitoring SLA violations in the Cloud Computing", International Conference on Cloud Technologies and Applications (CloudTech), IEEE, 2015.
51. Rocco Aversa, Nicola Panza, Luca Tasquier, "An Agent-based Platform for Cloud Applications Performance Monitoring", 9th International Conference on Complex, Intelligent, and Software Intensive Systems, , 535-541, IEEE, 2015.
52. Pierfrancesco Bellini, Ivan Brunoa, Daniele Cenni, Paolo Nesi, "Managing cloud via Smart Cloud Engine and Knowledge Base", Journal of Future Generation Computer Systems, pp142-154Elsevier, 2018.
53. Sukhpal Singh, Inderveer Chana and Rajkumar Buyya, "STAR: SLA-aware Autonomic Management of Cloud Resources: IEEE Transaction on Cloud Computing, pp1-14,IEEE, 2017.
54. Gary A. Mc Gilvary, Josep Rius, Inigo Goiri, Francesc Solsona, Adam Barker, Malcolm Atkinson, "C2MS: Dynamic Monitoring and Management of Cloud Infrastructures", ICCCTS, pp290 – 297 IEEE, 2013.
55. Chrysostomos Zeginis, Kyriakos Kritikos, Panagiotis Garefalakis, Konstantina Konsolaki, Kostas Magoutis, and Dimitris Plexousakis, "Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications", ESOCC, , pp188–195, Springer 2013.
56. Gregory Katsaros, Josep Subirats, J. OriolFitó, Jordi Guitart, Pierre Gilet, Daniel Espling, "A service framework for energy-aware monitoring and VM management in Clouds", Future Generation Computer Systems, Elsevier, 2013.
57. Bin Wanga, Zhengwei Qi, Ruhui Ma, Haibing Guan, Athanasios V. Vasilakos, "A survey on data center networking for cloud computing", Computer Networks, pp 528-547Elsevier, 2015.
58. J.A. Perez-Espinoza, Victor J. Sosa-Sosa, J.L.Gonzalez, Edgar Tello-Leal, "A Distributed Architecture for Monitoring Private Clouds", 26th International Workshop on Database and Expert Systems Applications, pp186-171, IEEE, 2015.
59. Jieyu Lin, Rajsimman Ravichandiran, Hadi Bannazadeh, Alberto Leon-Garcia, "Monitoring and Measurement in Software-Defined Infrastructure", International Symposium on Integrated Network Management (IM), 742-745 IFIP/IEEE, 2015.
60. Guilherme Da Cunha Rodrigues, Rodrigo N. Calheiros, Vinicius Tavares Guimaraes, Glederson Lessa dos Santos, Márcio Barbosa de Carvalho, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco, Rajkumar Buyya, "Monitoring of Cloud Computing Environments: Concepts, Solutions", Trends, and Future Directions. SAC 2016, ACM, 2016.
61. Xiaobo Ji, Fan Zeng, Mingwei Lin, "Data transmission strategies for resource monitoring in cloud computing platforms", Optik - International Journal for Light and Electron Optics, Volume 127, Issue 16, pp6726-6734 Elsevier, 2016.
62. Rongheng Lin, Yao Zhao, Budan Wu, Hua Zou, "An Auto Window Filter Algorithm for Resource Monitoring in Cloud",6th ICCC, IEEE, 2013.
63. Erkuden Rios, Wissam Mallouli, Massimiliano Rak, Valentina Casola, Antonio M. Ortiz,"SLA-driven Monitoring of Multi-Cloud Application Components using the MUSA framework",36th International Conference on Distributed Computing Systems Workshops, IEEE, 2016.
64. Priscila Cedillo, Javier Gonzalez-Huerta, Silvia Abrahao, Emilio Insfran,"A Monitoring Infrastructure for the Quality Assessment of Cloud Services", ISD2015 Proceedings, Springer, 2016.
65. Md Sadek Ferdous, Andrea Margheri, Federica Paci, Mu Yang, Vladimiro Sassone, "Decentralised Runtime Monitoring for Access Control Systems in Cloud Federations", 37th International Conference on Distributed Computing Systems, pp2682-2683, IEEE, 2017.
66. Yujian Zhu, Junming Ma, Bo An, Donggang Cao, "Monitoring and Billing of A Lightweight Cloud System Based on Linux Container",37th International Conference on Distributed Computing Systems Workshops, pp325-329, IEEE, 2017.
67. Praveen Kumar, Priyavrat Singh, Sarthak Chopra, Jagmeet Singh Sarna, Krishna Rawat, "Inspection of Cloud Computing Monitoring Tools", IEEE, 2017.
68. Jonathan Stuart Ward and Adam Barker, "Observing the clouds: a survey and taxonomy of cloud monitoring", open access, Journal of cloud computing, Springer, 2014.
69. Maitreya Natu, Ratan K. Ghosh, Rudrapatna K. Shyamsundar, Rajiv Ranjan, "Holistic Performance Monitoring of Hybrid Clouds", Complexities and Future Directions. vol.3, issue 1, IEEE Cloud Computing ,2016.
70. Jonathan Stuart Ward and Adam Barker, "Cloud cover: monitoring large-scale clouds with Varanus", Journal of cloud computing, Springer, 2015.
71. Waheed Aslam Ghumman, Alexander Schill, "Distributed Monitoring of Cloud SLAs Using S3LACC", Symposium on Service-Oriented System Engineering, pp114-119, IEEE, 2017.
72. Morgan Brattstrom, Patricia Morreale, "Scalable Agentless Cloud Network Monitoring", 4th International Conference on Cyber Security and Cloud Computing, pp171-176, IEEE, 2017.
73. Juan Gutierrez-Aguado, Jose M. Alcaraz Calero, Wladimiro Diaz Villanueva, "IaaSMon: Monitoring Architecture for Public Cloud Computing Data Centers", Springer, 2016.
74. Saeed Zareian,Marios Fokaefs, Hamzeh, Xi Zhang, "A Big Data Framework for Cloud Monitoring", 2nd International Workshop on BIG Data Software Engineering, BIGDSE'16, Austin, TX, USA, 2016.
75. Dmitrii A. Zubok, Tatiana V. Kharchenko, Aleksandr V. Maiatin, Maksim V. Khegai, "A Multi-Agent Approach to the Monitoring of Cloud Computing System with Dynamically Changing Configuration",Proceedings of the 18th conference of Fruct Associations, IEEE, 2017.
76. Jose M. Alcaraz Calero a, Juan Gutiérrez Aguado, "Comparative analysis of architectures for monitoring cloud computing infrastructures", Journal of Future Generation Computer Systems, pp16-30 Elsevier, 2015.
77. Hassan Mahmood Khan , Gaik-Yee Chan, Fang-Fang Chua, "An Adaptive Monitoring Framework for Ensuring Accountability and Quality of Services in Cloud Computing", IEEE, 2016.
78. Demetris Trihinas, George Pallis, Marios D. Dikaiakos, "Monitoring Elastically Adaptive Multi-Cloud Services", Issue: 99, pp1-14,IEEE Transactions on Cloud Computing, IEEE, 2017.
79. Khalid Alhamazani, Rajiv Ranjan, Prem Prakash Jayaraman, Karan Mitra, Meisong Wang, Zhiqiang (George) Huang, Lizhe Wang, Fethi Rabhi, "Real-time QoS monitoring for Cloud-based Big Data Analytics Applications in Mobile Environments", pp337-341, IEEE,2015.
80. W. Chung, R. Chang , "A New Mechanism For Resource Monitoring in Grid Computing," Future Generation Computer Systems - FGCS 25, PP 1-7, Elsevier, 2009.
81. H. Fang-fang, P. Jun-jie, Z. Wu, L. Qing, L. Jian-dun, J. Qin-long, Y. Qin (Oct 2011), "Virtual Resource Monitoring in Cloud Computing," Journal of Shanghai University, 15, PP 381-385, springer, 2011.
82. H. Huang, L. Wang (Jul 2010), "P&P: A Combined Push-Pull Model for Resource Monitoring in Cloud Computing Environment," 3rd IEEE International conference on Cloud Computing. Miami, FL, pp260 – 267, IEEE, 2010.

83. Dr.Mahantesh N Birje Chetan Bulla,"Cloud Monitoring System: A Review", vol 1, issue 1, IJESM,2019.
84. Dr.Mahantesh N Birje, Chetan Bulla, "Commercial and Open Source Cloud Monitoring Tools: A Review", International Conference on Emerging Trends in Engineering, Springer, 2019.
85. MN Birje, SS Manvi, "Wigrimma: A wireless grid monitoring model using agents", 9 (4), 549-572, Journal of Grid Computing, Springer, 2011.