

# Cloud Resources Optimization for Air Pollution Monitoring Devices and Avoiding Post Pillar Problem

Parampreet Singh<sup>(✉)</sup> and Pankaj Deep Kaur

Computer Science Department, Guru Nanak Dev University,  
Regional Campus, Jalandhar, India  
param.pasricha@gmail.com, pankajdeepkaur@gmail.com

**Abstract.** Cloud Computing is 21<sup>st</sup> century's precious gem that is revolutionizing the computing world. Cloud computing is progressively transforming the world through its wide applicability in diverse fields. One such field is environment monitoring. Today cloud computing is being utilized for monitoring the air pollution levels in association with different sensory devices and aid the ecologists around the globe to derive subtle ways to lower down its impact factor. But the major problem with such noble application is the elasticity factor of resource provision in cloud for handling the gargantuan amount of data that is generated by sensors. This elasticity cause troublesome to the service provider as the need of resources are very erratic and spontaneous. In this paper we present an algorithmic technique that attempts to quash this problem and provide a way to optimally allocate and utilize the resources. The evaluated simulation results reveals a very positive side and suggest an increase in utilization factor by 25 %–40 %.

**Keywords:** Cloud computing · Cloud computing resource optimization · Air pollution monitoring

## 1 Introduction

The modern era of 21st century encompasses a digital backbone. Fresh principles are entering our lives each and every day by use of technology. Things which were once dreams for our forefathers are now existing in reality and serving the basic human needs. Cloud computing is one such aspect that is propagating contemporary human civilization to a next level by its unmatched services. The fundamental definition of cloud computing is “the usage of computing resources over the internet”. Today cloud computing has acquired a serious level of popularity as compared to existing techniques. The chief contributing aspect to this substantial growth is the “Elasticity” behavior catered by the cloud computing to the user. This elasticity permits the user to increase or decrease the demand of computing resources at any instant of time as per their requirements. This allows the users to lower down the superfluous costs that are generally taxed on them if physical or personnel resources were deployed from their end. Now only usage fee is charged upon them and nothing else. Moreover the maintenance costs also lowers down by shifting to cloud. Cloud computing has

completely revolutionized the scenario of business computing. But the continuous evolution of this gem has now turned its application towards diverse fields. One such aspect where cloud computing has spread its wings is “Air Pollution Monitoring”. Currently, a lot of research work is being carried out by different researchers who are leaving no stone unturned to find an antidote to this perilous mess. Number of pollution monitoring devices are deployed by various organizations to carry out the surveillance work. The major problem associated with the assigned devices is that they generate huge amount of data which requires some serious computation. Now cloud computing can be efficient solution to this due to its elastic nature. But what turns out to be boon for one, sometimes turns out to be a bane for the other. The resources that are allocated by the service provider to the devices are sometimes overused and sometimes under-used. The major influential factor that leads to this muddle is of improper management of resources. The provider has to bear huge losses for the underutilization part as unutilized resources are just eating up the power without providing any substantial benefits. The over utilization taxes the resources pretty brutally and leads to deterioration of the resources. Hence there is some serious need to address this issue and a dynamic approach is required which can find a blend to allocate appropriate amount of resources in such a manner that utilization factor the resources is always high and succor the provider with some good amount of benefits.

In this paper we have tried to adopt a completely different technique to increase the utility of the resources. This algorithmic technique named as “OCR-SAPM” (optimizing cloud resources to sate air pollution monitoring) endeavors to utilize the unused resources of peculiar host by adjusting the load on different hosts along with considering their capacities to handle the load. The future part is divided into 5 sections. Section 2 discusses the literature survey and analysis the work that has been carried out earlier by various authors to gain the best possible solution. Section 2.1 describes the problematic nature of the mentioned techniques. Section 2.2 discusses the major motivational force behind the work. Section 3 firstly describes the terminology required for the smooth understanding of the algorithm. This section further contains Sects. 3.1 and 3.2 which discusses the “Post and Pillar Problem” and OCR-SAPM algorithm respectively. The subsequent part the results and analysis part followed by conclusion respectively. At last all the references of different works are listed out without help of which this work wouldn’t have been possible.

## 2 Related Work

For a service provider, an effective management of resources is becoming a convoluted task as with the popularity of cloud computing. Thoughtful efforts are being made towards achieving the desired outcomes. The research under this is not fresh but had been there for quite a time. [2] Under his work provided an appropriate “MLE” method of estimating the resource requirement of the users under severe complexities of IAAS and E-Learning system. The practical implementations were carried out with the help of “Samsung School Solutions” to monitor the effectiveness of the algorithm. Few Additional Resources were always reserved in case of emergency requests. Nagpure in [3] employs a special load balancing server, whose sole purpose is to envisage the

upcoming demands. “Skewness” was used to determine the factors responsible for creation of chaos under resource optimization and how it can be dealt. [4] Provides a methodology to counter back and forth occurrence of resource demand using certain bound values. [1] Proposes a proficient technique named as “auto-scaling algorithm” to provide resource allocation for different workflow inputs. [5] Provides a technique to balance the load on the system by considering the response time associated with each task. The author first address the contemporary issues with other load balancing algorithms out there in the world. The algorithms predicts the response time for the incoming requests. In [6], author strives to publish a perfect difference between the static and dynamic techniques out there that helps to balance the load. It also caters a new elasticity policy to help the provider to cater needs in a better and appropriate manner. [7] Works for the multimedia cloud and helps the cloud industry by an algorithm for allocation of resources on “Near-Client-Datacenter” basis. The network analysis was taken into due consideration along with the allocation part in his work. The problem of catering sudden outburst in workload with efficient management and allocation of resources was handled in [8]. Here traffic burst was considered for different applications under different scenarios like using dataset of 1998 World Cup to find out the desired results. Saad and Babar in their taxonomy [9] tries to present the analysis of various categories under “Resource Management” which includes Energy-aware management, SLA-aware management and many more categories. A detailed comparative analysis was provided by the authors to process the fortes and flaws of different techniques. [10] Addresses a very critical, unmanageable problem of energy consumption of the data centers and tries to propose a holistic approach based on multi-criteria decision system. S. Manvi in his work [11] studies the various issues of one of the services of cloud computing namely IAAS and suggests various schemes out in the computing world that are being utilized to counterpoise the issue. An adaptive migration technique for various resources by using skewness and prediction methodology was provided by [12]. The next section explores a brief comparison between certain specified algorithms for specifying the problems that these algorithms are facing (Table 1).

**Table 1.** Brief comparative analysis between certain specified algorithms.

Algorithm name	Average based	Centrally monitored
Maximum Likelihood Estimation.	No	Yes
Skewness based Allocation System.	Yes	Yes
Auto-Scaling Algorithm.	No	Yes

## 2.1 Problem Description

Though the proposed works of all the above mentioned author is pretty impressive, but the major stage used for their deployment is either “Mean/Average” based or a

completely different “Central monitoring system” based whose purpose is to be a spectator and just monitor the other hosts for any adverse circumstances or for switching task if required. The problem with the latter is that deploying a complete system just for monitoring purpose is not valuable taking into effect that the contemporary systems are way more powerful and efficient. This “Patterned” or synchronous approach is not appropriate as the system can be utilized for some other useful computing and scientific work. Moreover, the problem with the former “Mean” method is that analysis are based on average of a parameter as compared to a bigger picture. As “Mean” can only lower the gap between the available and the optimal but cannot provide an optimal solution which eventually hampers the optimization of resources. Certain Algorithms also uses “Prediction” based methodology to assign resources which most of the times backfires. OCR-SAPM address these problems by switching from a “Patterned” approach to a “Non-Patterned” approach. Here there is no deployment of any central monitoring system, rather there is an integration of the existing systems.

## 2.2 Motivational Work

Whenever we talk about the advancements in the technology, one name is there that can't be left unnoticed. The name is “Google”. As we all are very well aware about the drastic effects that are being foreseen by the environmentalists regarding climate change due to snowballing pollution levels, Google has come upfront and took the initiative to put curtains to this mess. At the 11th Annual Clinton Global Annual Google declared his partnership with Aclima, a San-Francisco based sensor manufacturing organization, to utilize its street view cars to generate pollution maps [15] which would help the users using google maps to get instantaneous updates of the pollution levels of different areas. The Cars will be properly equipped with different sensors to monitor different pollutants like CO<sub>2</sub>, NO<sub>2</sub>, and Particulate Matter (PM) etc. The data collected by the sensory nodes is transferred to cloud systems, but before transmitting it, the values are amplified refined as per requirements to make a hassle free transfer. The cloud system analyzes the data and provide feedback as RED zones or GREEN zones according to severity of pollution level. Moreover, the data that has been collected is further made available to ENVIRONMENTALIST, public Government agencies etc. to use that data to cater certain useful measures that can help to put curtains to this ever growing hazardous problem. The following figure illustrates the appropriate stepwise process of the above mentioned scenario (Fig. 1).

Data generation by these sensory devices varies from moderate to gigantic level. This fitful generation makes data processing at the cloud side difficult in terms of resource allocation. Provider cannot simply predict the amount of resources that are required to be allocated. This unevenness of data depends upon the working area of the sensors. For instance, a crowded market place will generate an enormous amount of data after monitoring as compared to a residential area. Hence, our algorithm “OCR-SAPM” considers this chaos and delivers an efficient and effective algorithmic solution for the plight of provider. The next section will address the work that has been proposed under “OCR-SAPM”.

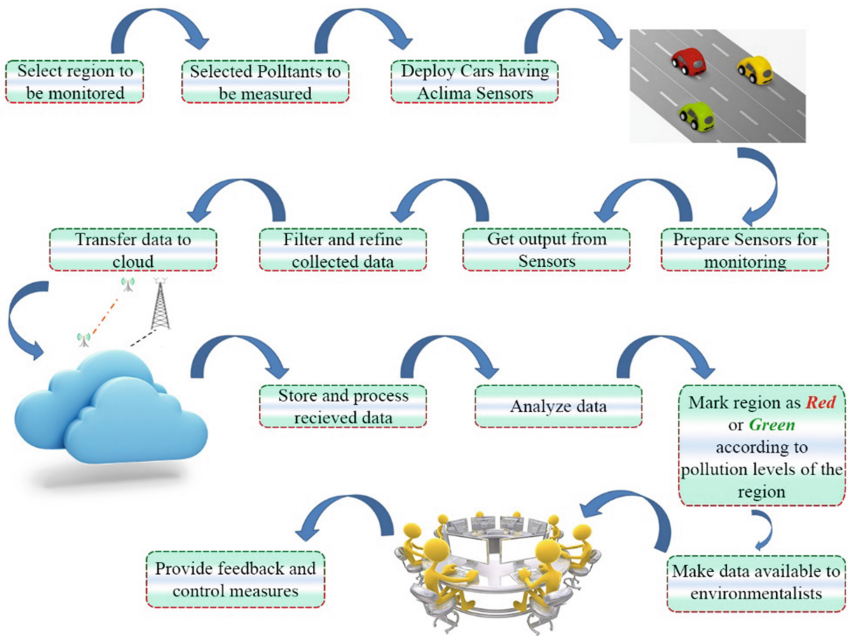


Fig. 1. An overview of general steps under specified Air pollution monitoring scenario.

### 3 Proposed Work

There is a definite set of terminology associated with OCR-SAPM that needs to be addressed before leaping into any subsequent part. The following table describes them all (Table 2).

Table 2. Terminology and Definition.

Terminology	Definition
Common_Tab_Map	This is a common mapping table that contains all the hosts that are available and number of resources they can serve.
CL_Data	Data transferred by monitoring device.
TOBO	The Top bound that is assigned to host
BOBO	The Bottom bound that is assigned to host.
HDL,HDLR_ID	Represents the host ID's that are available to aid the loaded host.
OL_Cond	Overloaded condition of host.
UL_Cond	Under loaded condition of host.
RAP	Resource availability proportion. i.e. amount in percentage of resources that are available at any instant.
MAX_RA	Maximum resource availability under a specific host.
CL_Data_Min	Data Item that entails minimum resources.
CL_Data_Max	Data Item that entails maximum resources.

### 3.1 Post and Pillar Effect

Before jumping into the pool of OCR-SAPM, there is a concept associated with BOBO and TOBO values that needs to be addressed. The concept is named as “Post and Pillar Effect”. The concept is named after back and forth flow that occurs sometimes due to allocation and deallocation of load i.e. on increasing the load, the overload condition occurs, whereas on decreasing the load, the under load condition occurs. This is known as “Post and Pillar effect”. To make sure that such situation never occurs, the values of BOBO and TOBO for each hosts are assigned using two methods which are discussed as under:

- Inert Allocation: Under this technique the value BOBO and TOBO are assigned in a static context. Here the TOBO is assigned higher to achieve a greater utilization factor. Now assigning a higher bound also increases the chances of underutilization. To counterpoise this problem we limit the value of BOBO as:

$$BOBO < \frac{TOBO}{2} \tag{1}$$

- Relative allocation: Under relative allocation the TOBO is settled first, but the value of BOBO is assigned by scrutinizing TOBO. Under this method, there is occurrence of frequent changes to BOBO value according to the load on the system. Under this BOBO is given in terms of TOBO as:

$$BOBO < \frac{(MAX\_RA - RAP) - (MAX\_RA - RAP')}{MAX\_RA - RAP} . TOBO \tag{2}$$

- Declaration: Decreasing load does not simultaneously generates under loading situation if BOBO and TOBO are assigned as stated above.
- Evidence: Initial total load before decrease in load can be given as:

$$(MAX\_RA - RAP) . BOBO \tag{3}$$

Now after the load is decreased, the new total load according to (3) can be given as,

$$\frac{(MAX\_RA - RAP)}{(MAX\_RA - RAP) - (MAX\_RA - RAP')} . BOBO$$

Were RAP' is the new proportion of available resources. To make sure that on increasing load, overload condition does not occurs, we must set

$$TOBO > \frac{(MAX\_RA - RAP)}{(MAX\_RA - RAP) - (MAX\_RA - RAP')} . BOBO$$

Or

$$BOBO < \frac{(MAX\_RA - RAP) - (MAX\_RA - RAP')}{MAX\_RA - RAP} . TOBO \quad (4)$$

- *Declaration:* Increasing load does not simultaneously generates over loading situation if BOBO and TOBO are assigned as stated above.
- *Evidence:* Initial total load before increase in load can be given as:

$$(MAX\_RA - RAP) . TOBO \quad (5)$$

Now after the load is increased, the new total load according to (5) can be given as,

$$\frac{(MAX\_RA - RAP)}{(MAX\_RA - RAP) + (MAX\_RA - RAP')} . TOBO$$

To make sure that on removing, under load condition does not occurs, we must set,

$$BOBO < \frac{(MAX\_RA - RAP)}{(MAX\_RA - RAP) + (MAX\_RA - RAP')} . TOBO \quad (6)$$

Now from (4) and (6),

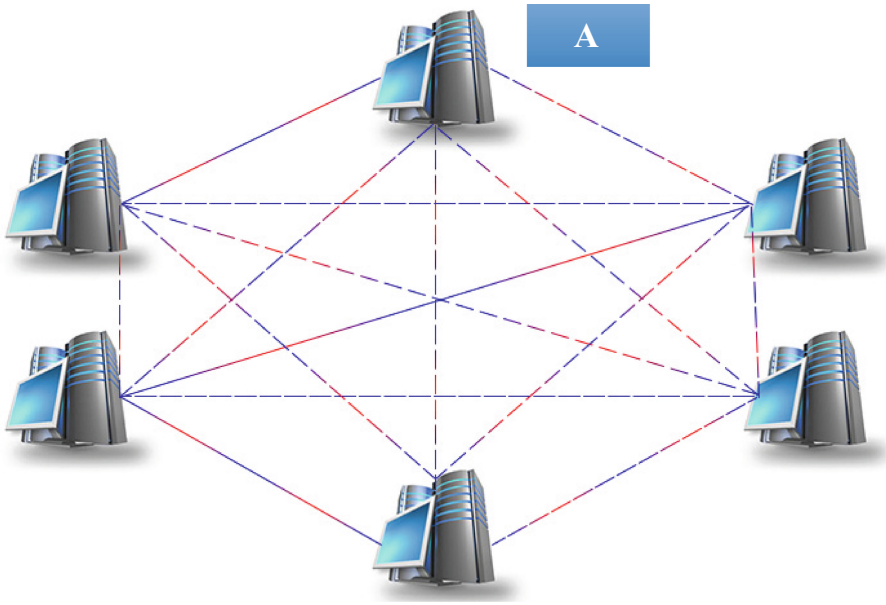
$$\frac{(MAX\_RA - RAP)}{(MAX\_RA - RAP) + (MAX\_RA - RAP')} . TOBO > \frac{(MAX\_RA - RAP) - (MAX\_RA - RAP')}{MAX\_RA - RAP} . TOBO > BOBO$$

Hence to make sure that post pillar problem never occurs, set

$$BOBO < \frac{(MAX\_RA - RAP) - (MAX\_RA - RAP')}{MAX\_RA - RAP} . TOBO$$

### 3.2 OCR-SAPM Algorithm

The modern industry computers have become so powerful that handling hundreds of users at a particular instance of time is just like a piece of cake. Appointing a scrutinizer over these powerful machines is just a wastage of computing power which can be labeled with some other scientific and beneficial work. In order to put an end to this approach, OCR-SAPM algorithm uses “entangled” approach as stated in diagram. Under this each computing node has a copy of Common\_Tab\_Map table which gets apprised by each and every other node once the resource are manipulated by incoming or outgoing of Data that requires resources for processing. As below in Fig. 3, as soon as the resources are consumed from node A, all other nodes are apprised by it from A and the Common\_Tab\_Map is refreshed by all (Fig. 2).



**Fig. 2.** Entangled Approach for Common\_Tab\_Map.

The Common\_Tab\_Map and RAP performs the core functionality of the algorithm. The successful working of algorithm depends upon these two parameters. The greater value of RAP signifies that there is substantial amount of underutilization is prevailing among hosts and vice-versa. Genesis of OCR-SAPM contains a module as Regular\_Optimize() which first gets introduced to the incoming data. The incoming CL\_data is simply assigned to Host [x] presuming it to be capable enough to provide service. Now, Common\_Tab\_Map is revised for the existing resources and is sorted out in an aspiring manner. Check\_HDL() now joins the party and looks for the ideal host that satisfies two conditions namely, Fulfill Data resource requirement and Improve resource utilization. These conditions are surveyed from Common\_Tab\_Map in such a manner that the specific host has ample RAP value (minimum) available to provide resource supply without fracturing its TOBO. Now if there is an availability, then the ID of that host is stored in HDL and that CL\_Data is assigned to it for processing. Common\_Tab\_Map is again restructured via sorting.

But, this simple thing can become problematic if there is not even a single host to provide service and the data arrived sticks to the existing host which also unfortunately becomes overloaded (Monitor\_OL()) and demands immediate offloading. If this kind of situation arrives then OCR-SAPM transforms itself from “intra host” where exchange was from one host to another of a machine to “inter host” where now it will be carried out between hosts under different machines. OL\_OCR\_SAPM() is asked for a solution. The OL\_OCR\_SAPM() first derives the TOBO and BOBO values and along with this, a list named as “CL\_Data\_List” is prepared for the current data that is under processing on that host. Now in order to fully optimize the resource usage, it is not appropriate to assign any ith CL\_Data. To achieve the best results, The CL\_Data that requires



minimum amount of resources is selected and is stored under, `CL_Data_Min`. To assign this under a suitable Host, `get_HDLR_ID_Common_Tab_Map_OL()` is summoned which scrutinizes the `Common_Tab_Map` for the host (other than current) that can successfully satisfy optimization conditions as well fulfil resource catering service for the `CL_Data`. The RAP again plays an important role here as selection of host is based on the minimal level i.e. the host having minimum RAP is preferred, provided it does not tax its resources upon selection. If under extremely unfavorable circumstances, a situation comes when there is no host that can cater the service, then we have to apply new nodes to lower the burden on current host.

Till now our central theme for optimization was restricted to overloading situation. But optimization of resources doesn't always associated with overloading. Sometimes the situation of under loading can also severely obstruct the optimization. Most of the contemporary algorithm doesn't take this situation into their radar for optimization. OCR-SAPM challenges this problem and provides an efficient solution. `MonitorUL()` raises an under load condition whenever the RAP becomes greater than BOBO i.e. the maximum resources are unwaged. `UL_OCR_SAPM()` is called which works likewise as `OL_OCR_SAPM()` but instead of selecting the `CL_Data_Min`, it chooses `CL_Data_Max` i.e. the supreme resource yearning is chosen. On the similar guidelines `get_HDLR_ID_Common_Tab_Map_UL()` is invoked and on accomplishment, the unrequired freed hosts are Signed-Off to provide resource provider ample and high yielding amount of benefits. The following section discusses the complete pseudo-code OCR-SAPM algorithm (Tables 3, 4, 5, 6, 7, 8, 9 and 10).

### 3.3 OCR-SAPM Algorithm

**Table 3.** Algorithm: `Regular_Optimize(Common_Tab_Map)`.

```

1: Host[x] ← CL_Data[x]; //Host[x] ∈ machine[x].
2: Refresh and Sort Common_Tab_Map;
3: HDL ← Check_HDL(Common_Tab_Map, Resources[CL_Data]);
4: if (HDL) {
    Host [HDL] ← CL_Data[x] and repeat 2: only;
}
5: Inspect if Overloaded/Under loaded;
6: If Overloaded, assign OL_Cond=true;
7: If Under loaded, assign UL_Cond=true;
8: if (OL_Cond==true) {
    Call OL_OCR_SAPM();
}
9: else if (UL_Cond==true) {
    call UL_OCR_SAPM();
}
10: More CL_Data, Goto 1;

```

**Table 4.** Algorithm: Check\_HDL (Common\_Tab\_Map, Resources).

```

1: for (count ← 0 to Size(Common_Tab_Map)) {
    If (TOBO[count] > RAP[count] ≥ Resources) {
        Return count;
    }
    Break; }

```

**Table 5.** Algorithm: OL\_OCR\_SAPM (Host [x], Common\_Tab\_Map).

```

//Host[x] signals an overloaded condition.
// CL_Data_List is initialized as an empty list.
1: TOBO[x] ← Host[x];
   BOBO[x] ← Host[x];
2: For Host[x] {
   CL_Data_List ← Host[x]; //Preparing list for data processing under a
   specific host.
}
//Step 3 & 4 fetches the CL_Data_Min
3: CL_Data_Min ← CL_Data_List[0];
4: For (count ← 0 to Size (CL_Data_List)) {
   If (CL_Data_List [count] < CL_Data_Min) {
       CL_Data_Min ← CL_Data_List [count];
   }
}
5: Sort (Common_Tab_Map);
6: for (j ← 0 to Size(Common_Tab_Map)) {
   HDLR_ID ← get_HDLR_ID_Common_Tab_Map_OL (Common_Tab_Map, Resources
   [CL_Data_Min]);
}
7: if (HDLR_ID) {
   Host[HDLR_ID] ← CL_Data_Min;
   Remove CL_Data_Min from Host[x];
   Refresh RAP[x] in Common_Tab_Map;
   Refresh RAP[HDLR_ID] in Common_Tab_Map;
} else {
   Assign_New_Host ← CL_Data_Min;
}
8: if (Host[x] == Overloaded) {
   Go to 2;
} else {
   Return;
}

```

**Table 6.** Algorithm: get\_HDLR\_ID\_Common\_Tab\_Map\_OL (Common\_Tab\_Map,Resources).

```

1: For (Every hosts[j] in Common_Tab_Map) {
  If (TOBO [j] >RAP [j] > =Resources){
    HOVL← j;//local variable for storage
    break;
  } else {
    continue;
  }
}
2: If (HOVL ==null) {
  HOVL← -1;
}
return HOVL;}

```

**Table 7.** Algorithm: MonitorUL(Common\_Tab\_Map).

```

1: If (RAP [x] > BOBO [x]) {
  Return "under loaded";
}

```

**Table 8.** Algorithm: MonitorOL(Common\_Tab\_Map).

```

1: If (RAP [x] <TOBO [x]) {
  Return "overloaded";
}

```

**Table 9.** Algorithm: UL\_OCR\_SAPM (Host [x], Common\_Tab\_Map).

```

//Host[x] signals an under load condition.
// CL_Data_List is initialized as empty list.
1: TOBO[x]←Host[x]; //Fetching TOBO for host[x].
BOBO[x]← Host[x]; //Fetching BOBO for host[x].
2: for Host[x] {
  CL_Data_List ← Host[x]; //Preparing list for data processing under a
  specific host.
}

3: CL_Data_Max← CL_Data_List [0];
4: For (count=0 to Size (CL_Data_List)) {
  If (CL_Data_List [count]> CL_Data_Max) {
    CL_Data_Max ← CL_Data_List [count];
  }
}
5: Sort (HashMap);
6: for (j←0 to Size(HashMap)) {
  HDLR_ID ←
  getHDLR_ID_Common_Tab_Map_UL(Common_Tab_Map,Resources[CL_Data_Max]);
}
7: if (HDLR_ID) {
  Host[HDLR_ID] ← CL_Data_Max;
  Remove CL_Data_Max from Host[x];
  Refresh RAP[x] in Common_Tab_Map;
  Refresh RAP[HDLR_ID] in Common_Tab_Map;
} else {
  Assign_New_Host ← CL_Data_Max;
}
8: if (Host[x]==Under loaded) {
  Go to 2:
} else if ((Size)CL_Data[Host[x]]==0){
  Host[x]← Sign-Off;
  Return;
}

```

**Table 10.** Algorithm: getHDLR\_ID\_Common\_Tab\_Map\_UL (Common\_Tab\_Map, Resources).

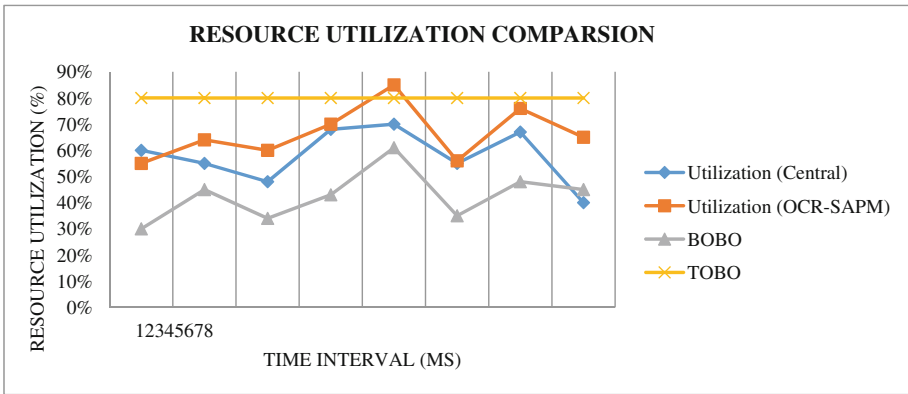
```

1: For (Every hosts[j] in Common_Tab_Map) {
  If (TOBO [j] >RAP [j] >BOBO [j] && Resources<=RAP[j]) {
    HUDL←j; //local variable for storage.
    break;
  } else {
    HUDL←-1;
  }
}
return HUDL;
}

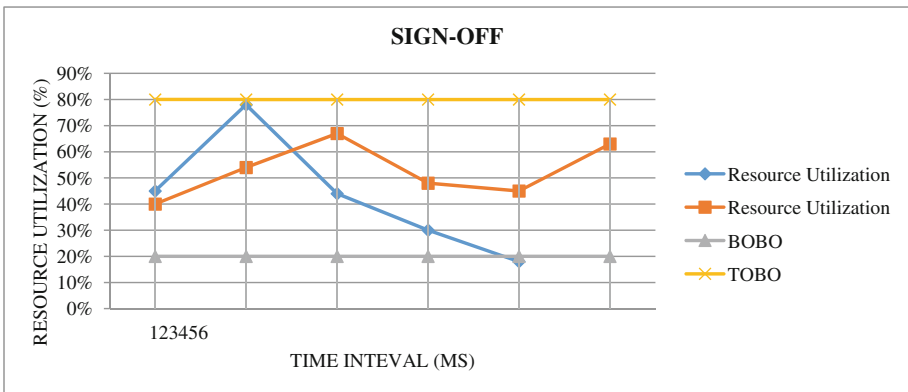
```

**Table 11.** Different values of different pollutants received from different areas. (In AQI).

PM 2.5	O <sub>3</sub>	NO <sub>2</sub>	SO <sub>2</sub>
184	8	12	4
168	16	4	2
185	10	2	1
164	4	3	3
205	7	12	1
107	4	17	1
159	2	20	2
177	4	19	4
145	5	17	2



**Fig. 3.** Resource utilization comparison. (Color figure online)



**Fig. 4.** Sign-Off. (Color figure online)

## 4 Results and Analysis

Under this section a detailed graphical analysis is presented containing the percentage increase in the optimization levels using OCR-SAPM algorithm. The Sample data from [13, 14] was used to calculating the results. Pollutants such as Particulate Matter (PM), Ozone (O<sub>3</sub>), Nitrogen Dioxide (NO<sub>2</sub>) and Sulfur Dioxide (SO<sub>2</sub>) were taken into consideration as shown in Table 11.

Simulation experiments were carried out using CloudSim 3.0.3. The inception load was assigned to different hosts on different machines using the random data generator function and further the run was augmented by the dataset for evaluation. Comparative analysis were performed using different number of task under different time intervals as described in Figs. 3 and 4. Preliminary analysis clearly reveals that there is significant resource utilization improvement if we switch to the given algorithm.

The simulation results clearly indicates the increase in utilization factor of the resources. Yellow and Grey lines indicate the TOBO and BOBO respectively Here TOBO is static, but BOBO keeps on adjusting as per the load. Blue and Orange line signifies the improvement factor discovered with the use of OCR-SAPM as compared to centrally appointed systems.

Figure 4 address the site of immediate load shrinkage which leads to increase in the RAP and as soon as RAP goes greater than BOBO, The load of that host is transferred to other host that can service its needs and also remain under TOBO and the current host is logged off.

## 5 Conclusion

Air pollution control is one of the utmost important agendas that needs some serious addressing. This paper proposes a dynamic resource optimization technique that caters its services to air pollution monitoring devices for efficient working. The Experimental analysis does reveal that there is significant improvement achieved in utilization part if OCR-SAPM is deployed. The algorithm benefits the resource providing authority with some serious reductions in cost factors by signing off certain unused resources and effectively utilizing the deployed ones. The future work can include to further optimize the algorithm to adapt network fluctuations and unavoidable circumstances related to optimization.

## References

1. Fakhfakh, F., Kacem H.H., Kacem A.H.: A Provisioning approach of cloud resources for dynamic workflows. In: 2015 IEEE 8th International Conference on Cloud Computing, New York City (2015)
2. Koch, F., Assuncao, D.M., Cardonha, C., Netto, M.A.S.: Optimizing resource costs of cloud computing for education. *Future Generation Computer Systems* **55**, 473–479 (2015). Elsevier

3. Nagpure M.B., Dahiwale P., Marbate P.: An efficient dynamic resource allocation strategy for VM environment in cloud. In: 2015 International Conference on Pervasive Computing (ICPC), Pune, pp. 1–5 (2015)
4. Khatua, S., Manna, M.M., Mukherjee, N.: Prediction-based instant resource provisioning for cloud applications. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC), London, pp. 597–602 (2014)
5. Sharma, A., Peddoju, S.K.: Response time based load balancing in cloud computing. In: 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, pp. 1287–1293 (2014)
6. Aslazandeh, S., Chaczko, Z., Chiu, C.: Cloud computing-the effect of generalized spring tensor algorithm on load balancing. In: 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), South Kuta, pp. 5–8 (2014)
7. Gong, W., Chen, Z., Yan, J., Qianjun, S.: An optimal VM resource allocation for near-client-datacenter for multimedia cloud. In: 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, pp. 249–254 (2014)
8. Zhang, Q., Chen, H., Shen, Y., Ma, S., Lu, H.: Optimization of virtual resource management for cloud applications to cope with traffic burst. *Future Gener. Comput. Syst.* **58**, 42–55 (2016)
9. Mustafa, S., Nazir, B., Hayat, A., Khan, A.R., Madani, S.A.: Resource management in cloud computing: taxonomy, prospects, and challenges. *Comput. Electr. Eng.* **47**, 186–203 (2015)
10. Arianyan, E., Taheri, H., Sharifian, S.: Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers. *Comput. Electr. Eng.* **47**, 222–240 (2015)
11. Manvi, S.S., Shyam, G.K.: Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey. *J. Netw. Comput. Appl.* **41**, 424–440 (2014)
12. Ashalatha R., Agarkhed J.: Dynamic load balancing methods for resource optimization in cloud computing environment. In: 2015 Annual IEEE India Conference (INDICON), New Delhi, pp. 1–6 (2015)
13. Air Quality Forecasting in Northern India (2014). <http://aqicn.org/faq/2016-02-28/air-quality-forecasting-in-northern-india/>
14. Air Pollution in India: Real-time Air Quality Index Visual Map. <http://aqicn.org/map/india/#@g/26.8439/77.5961/5z>
15. Schiffman R.: Air pollution – live online. In: *New Scientist*, vol. 228, pp. 20