Improvised Schema for Data Analysis in Big Data Cloud

Zainab Mohanad Issa

Phd computer science and engineering
Zainab.mohanad91@gmail.com
wasit , Iraq

ABSTRACT

The big data analysis using cloud computing is an ideal combination. The big data cloud greatly improves the performance of different applications and more helpful in problem-solving and decision making. The advancement of big data and cloud computing is attractive for data analysis. In big data for data processing and analysis, resource scheduling algorithms play a vital role. To meet the industry standards in data analysis, optimal usage of resources is very essential in the big data cloud. Moreover, the big data cloud has many challenges in optimal resource utilization. Many researchers have introduced a different kind of resource scheduling algorithms. But no algorithm can guarantee the performance and doesn't meet the user expectations. In existing system, there is no specification for monitor of VM status and does not aware the requirement of tasks. In this paper, we proposed improvised schema for better allocation and utilization of resources and improves the performance of data analysis in big data cloud. The proposed improvised schema considers different aspects such as analyzing tasks, scheduling and VM management. The proposed improvised schema improves the performance in terms of computation cost, execution time and optimal resource utilization. The proposed improvised schema achieved better results when compares with previous resource scheduling algorithms.

Keywords: Task Scheduling, Resource Allocation, Big Data, Cloud Computing

1. Introduction

The internet access is increases vastly and produce large amount data from day to day. The large amount of space is required to storage and maintain the data. The traditional storage system unable to maintain the data. For large storage of data and fast analytical results the current industry is moved to big data. Big data is emerging technology in the field of information technology[1]. Big data maintain the large analytical data and helpful to decision making.

Cloud computing is a novel computing paradigm subsequent to grid computing. The cloud computing have high amount of virtualized computing ability and storage space. So , it greatly maintain the computing resources through the network. The cloud computing is widely used in many real time applications. Cloud computing facilitate the services to the users in the form of pay-as-you-go model to get the computation and storage resources[2]. Cloud computing improves the services in data storage and maintains it will reduce users investment cost.

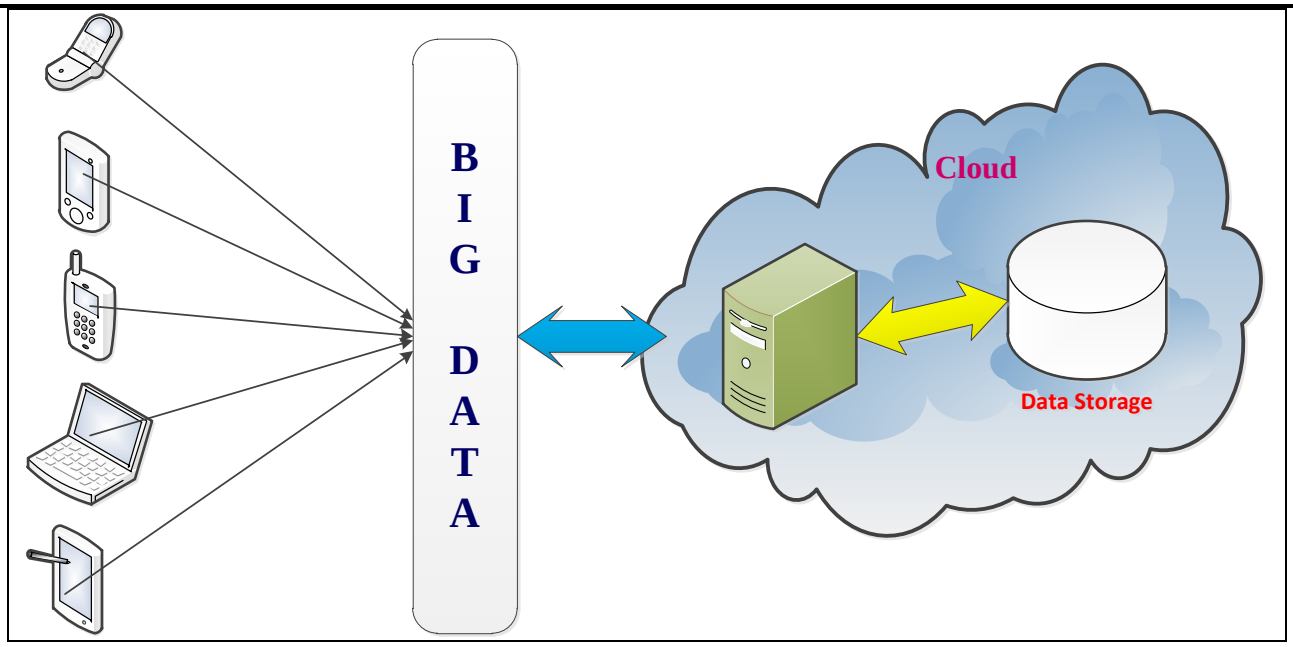

Fig 1. Mechanism of Big Data over Cloud Computing

In this paper combine the big data and cloud computing for storage and analysis of large amount of data. The combination of big data and cloud computing is an ideal technology in the field of information technology. The big data and cloud computing is greatly reduce the computation cost and time for large amount data maintenance and processing. The main important thing of big data using cloud computing is optimal utilization of resources. The optimal utilization of resources of big data and cloud computing is improves performance. The resource scheduling algorithms are suitable to utilize the resources in big data on cloud computing. More research applications are introduced for scheduling the resources in big data on the cloud environment. These research applications mainly focus on the tasks scheduling but not optimal utilization of resources. The novel scheduler algorithm should follows the optimal utilization of resources and have idea of task capacities and virtual memories.

In previous research, used task scheduling algorithm for resource sharing process. They also utilized the load balance and scheduling technique to minimize the task execution time and equally load the balance. This technique is very helpful to reduce the energy consumption and improve the throughput. It also helpful to make other resources not idle or unused[3].

The remainder of the paper is structured as follows. Section II provides review of literature on task scheduling and memory allocation to improve the performance of big data on cloud computing. Section III presents a improvised schema for data analysis in big data cloud. Section IV explore performance and comparison results while section V concludes the paper besides providing directions for future work.

2. Related Work

In this section, we discuss the previous resource scheduling algorithms and their improvements in big data on cloud computing. Priya Krishnan et al.[4] proposed a constrained genetic algorithm for the rebalancing of loads. In this rebalancing model, this algorithm consists of service sets and the group of hosts. The cloud provider gives the maximum number of hosts to achieve better results. The genetic algorithm rebalances the hosts optimally among the group of tasks. This algorithm does not perform the efficient host selection based on the task demands and has poor resource utilization.

Shubham Mittal et al[5] introduced task scheduling algorithm, it is improvised for the RASA, Max-Min and Min-Min algorithms in task scheduling. The load balancer is improved by using this task scheduling algorithm. But this

algorithm is only applicable for small and medium-sized data sets. The partitioning algorithms are introduced by Jianqiang Li[6]. This algorithm partition the tasks into small frames, these frames are executed efficiently and the problem arises when partitioning and execution of multi-frames. This algorithm arises the problem of data frames transfer and limited resources.

Pirtpal Singh [7] discussed the resource scheduling algorithms in terms of task capacity, execution time and other insights. Mostly the Round Rabin Algorithm and First Come First Serve algorithm are works based on the same technique. These algorithms interested in task scheduling and resource allocation only and they don't have awareness of task completion and status of machines.

Ruonan Lin et al [8] proposed a Pre-Allocation Ant Colony algorithm, as per this algorithm divide the tasks into smaller parts and pre-allocate the processor. The main objective is based on processor size the tasks are distributed and scheduled. There is no idea for the time taken to complete the tasks, sometimes it leads to incomplete the tasks. The processors are quit the job without knowing the status in given stipulated time. This algorithm also has a high computation cost.

Gunasekaran Manogarana et al. [9]. Sundararajan et al. [4] are motivated by the Map-Reduce algorithm for load balance and job scheduling of big data in the cloud environment. They are mainly focused on job schedule with equal load balance among all virtual memory. By the combine, the features al previous algorithms Saed Abed et al [3] enhanced task scheduling technique(TST) in big data cloud computing. However, they gain better performance in terms of latency, execution time and resource utilization and they do not awareness of historical capacities of tasks and allocated job schedulers. In this paper, our focus is on complete awareness of tasks capacities, task schedulers and optimal utilization of resources. The proposed resource management schema maximize performance.

3. Proposed Methodology

3.1 Problem Definition

The big data on cloud computing is tremendous changes in the performance of data analysis. The big data analytics is more helpful in the decision making. The data analysis using the big data over cloud computing has open problems in distribution of workloads among the different VMs. The nature of cloud computing and big data is leads to load balance and sub-optimal utilization of resources. In the computer network the data production is increased multiple times. Unaware of task conditions and sub optimal resource utilization is cause to degrade the performance in the big data over cloud computing. So the resource management and task distribution is needs the optimal utilization technique and task aware scheduler. This is the inspiration behind this research work to improve the performance of big data analysis over cloud computing.

3.2 Improvised Schema

The proposed improvised schema is focus on optimal resource utilization and task schedule. The ultimate goal of the proposed schema is to improve the performance of big data analysis and reduce the computation cost, execution time and maximize resource utilization. The improvised schema followed three different methods.

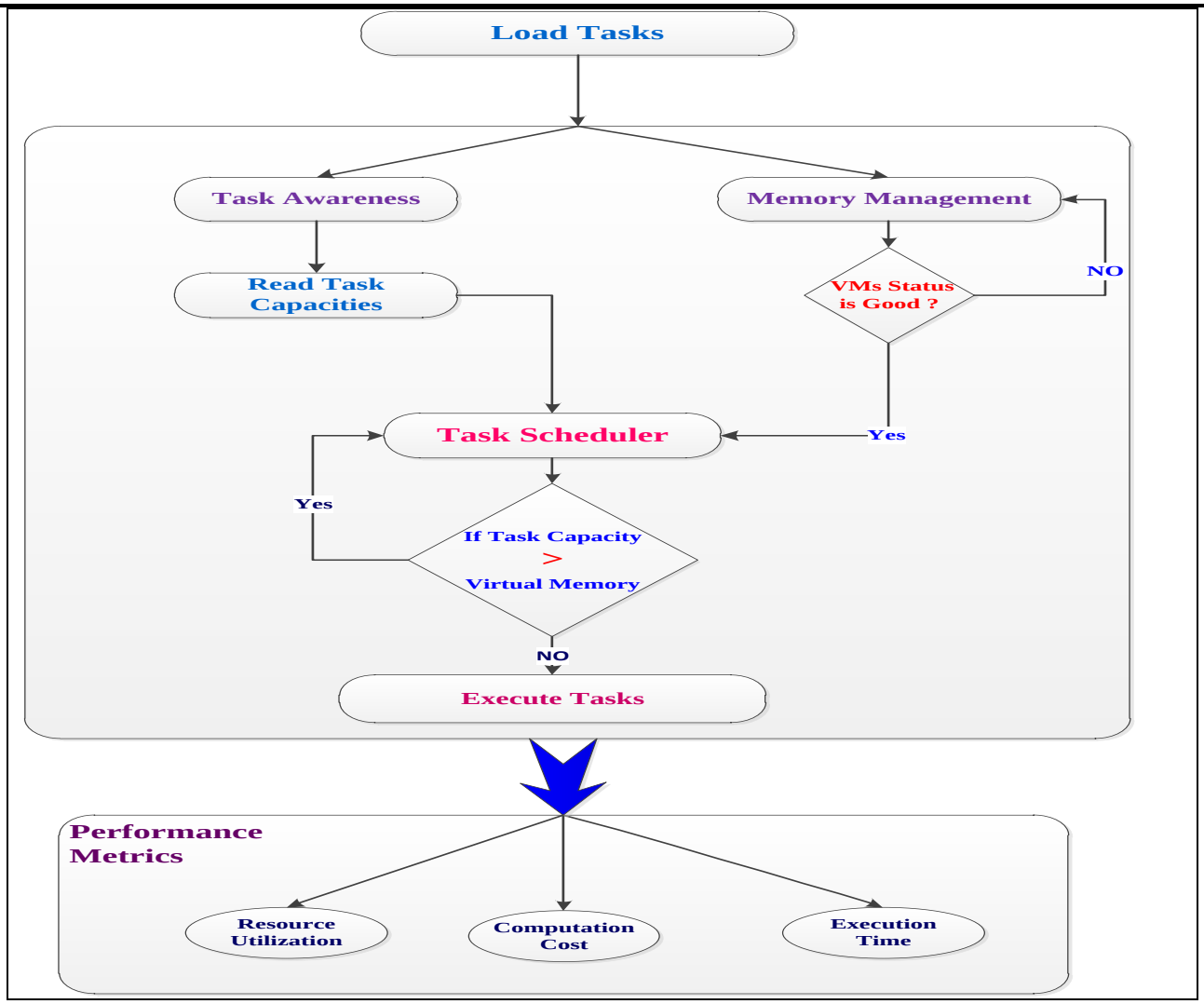

Fig 2 Architecture of Improvised Schema

Task Awareness

In this method, the schema knows the complete awareness of tasks. The task awareness means should know the capacity of the task, need of memory and compatibility of tasks. If schedule the tasks without know the task capacity its leads to failure of tasks.

Memory Management

The memory management will help to know the status of the VMs. Because some of the VMs is a failure in the middle of the task completion or some of the VM status is wait or sleep.

Task Scheduler

The task scheduler method depends on task awareness and memory management. Before scheduling the task, the schema must be the check task capacity and need of memory for the task. Also, need to verify the mode of VMs in system. In the process of task schedule, resource allocation plays an important role. In fig 2 illustrate the working model of the resource management schema

3.3 Algorithm

Algorithm Name : Improvised Schema Algorithm

Input : *Task Capacity ts, Virtual Memory Size vms*

Output : *Performance Maximization*

1. Load list of tasks
2. Apply Task Awareness
3. Read Task Capacities

```
4. Apply Memory Management
5. Check VM Status
6.          If (VM Status is good)
7.                  Go to Task Scheduler
8.          Else
9.                  Repeat Memory Management
10.         End If
11. Apply Task Scheduler
12.         If( ts > vms)
13.                 Repeat Task Scheduler
14.         Else
15.                 Execute Task
16.         End If
17. Calculate performance metrics
18. End
```

To overcome the resource sharing problem in big data analysis using cloud computing, proposed a improvised schema algorithm as is shown in algorithm 1. As per this algorithm, the schema followed different steps to process the big data.

4. Results

To obtain better performance results, the proposed schema implementation is done by using Cloud Simulation 3.0.3, Java 8 technology, Windows 8 Operating System and NetBeans IDE 8.2. The proposed improvised schema achieved better results in task scheduling and optimal utilization of resource using big data over cloud computing. In this section, analyze the results and compare with existing standard techniques.

4.1 Performance Metrics

The performance metrics in terms of resource utilization, execution time and computation cost are showed in this section. In this paper considered different performance metrics to evaluate the performance results. The definition of each performance metric is described here

Table 1. Performance Metrics

S NO	Metric	Description
1	Resource Utilization	The resource utilization performance metric is ratio between the busy time of resource and available time of resources.
2	Execution Time	The execution time performance metric is the difference between task end time and task start time.
3	Computation Cost	The computation cost performance metric is the cost of the complete data process.

The performance metrics presented in Table1 are used to evaluate proposed schema results. The results are observed in terms of these measures with mathematical analysis.

4.2 Comparison and Analysis

A. Resource Utilization

The ratio between the resource available time and busy time. The resource utilization of all tasks can be calculated as

$$\text{Resource Utilization} = \sum_{i=1}^{n}\left(\frac{BTi}{ATi}\right)*100 = 99.9\%$$

Here *i* denotes each task, BT represents the busy time for each task, AT represents the total available time. The resource available time and busy time can be measured as 114.39 and 114.29 respectively. So the percentage of resource utilization is 99.9%.

In comparison of average resource utilization, the proposed framework achieved 71.34%

high resource utilization when compare with the previous technique.

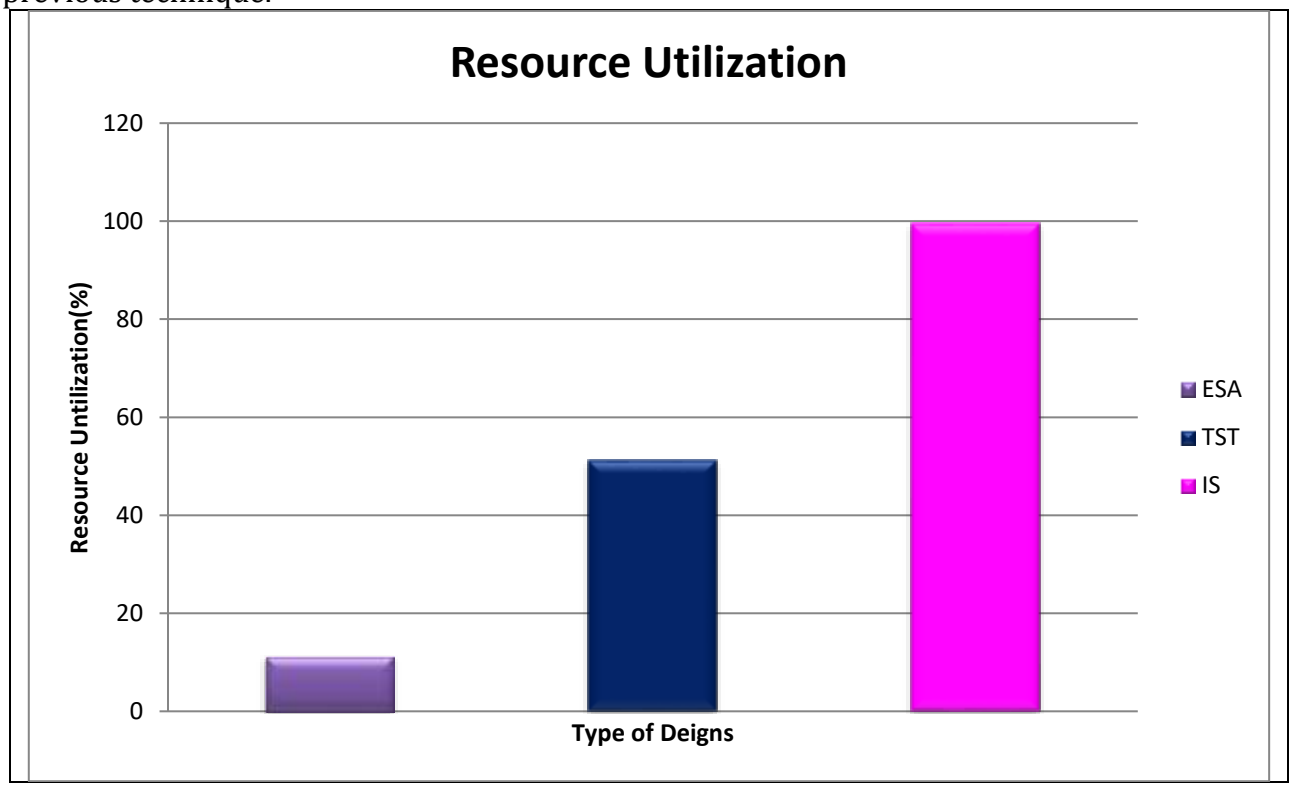

Fig 3. Performance of Resource Utilization

As shown in Fig 3. the performance of resource utilization of the proposed framework. The resources of big data on cloud computing are utilized high efficiently. The proposed framework utilized the resources optimally. The previous techniques such as Efficient Scheduling Algorithm(ESA) and Task Scheduling Technique(TST) are compared with the proposed Improvised Schema(IS). The IS achieved maximum utilization of resources. In fig 3. The type of design taken on the vertical axis and the percentage resource utilization is taken on horizontal axis

B. Execution Time

The time difference between the task end time and task start time represents the execution time

Average Execution Time = $\sum_{i=0}^{n}(TETi - TSTi)$

Here i denotes each task, TET represents task end time, TST represents the task start time

Table 2 Comparison Table for Execution Time

Task	Efficient Scheduling Algorithm		Task Scheduling Technique		Improvised Schema	
	VM No	Execution Time	VM No	Execution Time	VM No	Execution Time
0	0	200	0	127.08	0	114.29
1	0	200	1	127.08	1	114.29
2	0	200	2	127.08	2	114.29
3	0	200	3	127.08	3	114.29
4	1	400	4	148.15	4	114.29
5	1	400	5	148.15	0	114.29
6	1	400	6	148.15	1	114.29
7	1	400	7	148.15	2	114.29
8	2	800	8	177.7	3	114.29
9	2	800	9	177.77	4	114.29

In this case study the task start time for each task is 0.1 ms and task end time for each task is 114.39 ms. So the total execution time for each task is 114.29 ms. The average execution time is 114.29 ms.

When we compare the average execution time with previous techniques, our proposed framework reduce the average execution time to 78.47%.

$$\text{Average Execution Time} = \sum_{i=0}^{n}\left(\frac{Improvised\ Schema\ Execution\ Time}{TST\ Execution\ Time}\right) * 100 = 78.47\%$$

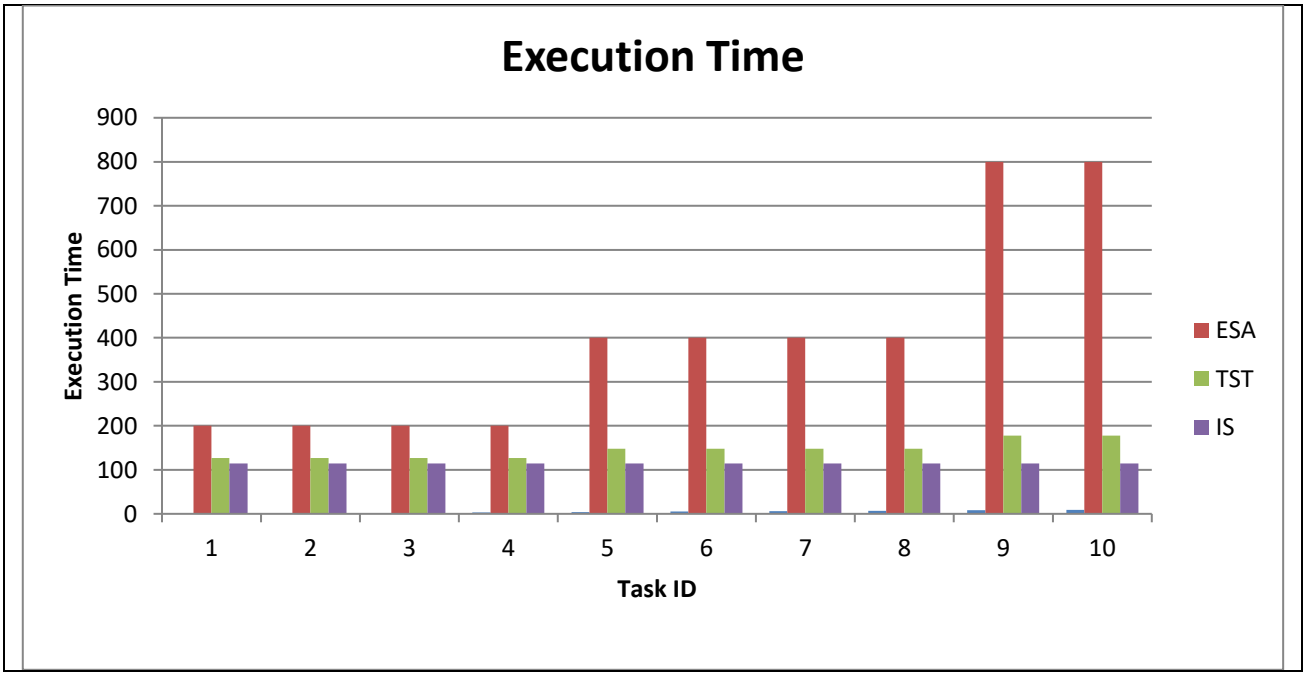

Fig 4. Performance of Execution Time

In Fig 4. presented that the performance of execution time of the proposed schema. The execution time of the proposed schema is very less. In the bar chart of execution time , x-axis denotes the present and previous schemas and y-axis denotes the execution time in milli-seconds. In result analysis the execution time of proposed schema is outperforms with the existing schemas.

C. Computation Cost

The computation cost can be calculated as

$$\text{Computation Cost} = \sum_{i=0}^{n}(CPU\ Cost + VM\ Cost + Storage\ Cost + BW\ Cost)$$

Here, the initial cost can be taken for CPU, VM , Storage and Bandwidth 3.0,0.05,0.1 and 0.1 respectively . So the total computation cost for our research simulation measured 104.81 sec

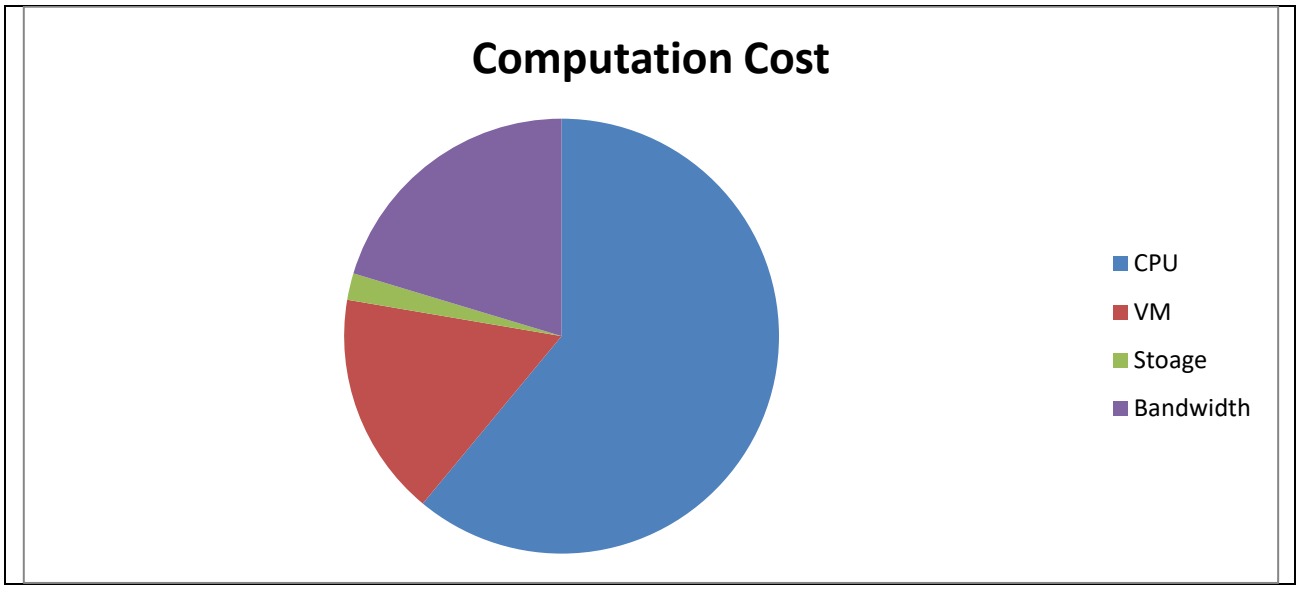

Fig 5. Performance of Computation Cost

As shown in fig 5 , the performance of computation cost is presented. In calculation of computation cost we accountability of CPU Cost, VM Cost, Storage Cost and Bandwidth cost. In Fig 5 can we observe the each cost of each value.

5. Conclusion And Future Work

In this research work, we mainly focused on improving the performance of data analysis of big data over cloud computing. The performance of big data affected by many factors. Those factors are data overload, the status of virtual memory and poor knowledge of resource utilization. In this paper proposed improvised schema. This schema takes accountability task characteristics and status of virtual memory. The implementation way of improvised schema improved the performance of big data analysis over cloud computing by optimal utilization of resources. The implementation of the proposed schema achieved better results. The proposed schema is highly effective in optimal resource utilization, reduce the computation cost and execution time.

However, the proposed schema in this research work achieved significant performance. This research is further extended in future to reach the expectations of present industry in big data cloud , we should enhance the implementation design.

References

1. Kezia Rani.B et al. (2015). Scheduling of Big Data Application Workflows in Cloud and Inter-Cloud Environments. IEEE. 1 (1), p2862-2864.
2. Wu.DaQin et al. (2018). Cloud Computing Task Scheduling Policy Based on Improved Particle Swarm Optimization. IEEE. 1 (1), p99-101.
3. Saed Abed et al . (2018). Enhancement of Task Scheduling Technique of Big Data Cloud Computing. IEEE. 1 (1), p1-6.
4. Priya Krishnan Sundararajan et al. (2015). A Constrained Genetic Algorithm for Rebalancing of Services in Cloud Data Centers. IEEE. 1 (1), p653-660.

5. Shubham Mittal et al. (2016). An Optimized Task Scheduling Algorithm in Cloud Computing. IEEE. 1 (1), p197-202.
6. Jianqiang Li. (2017). Computation partitioning for mobile cloud computing in a big data environment. IEEE. 1 (1), p1-10.
7. Pirtpal Singh et al. (2016). Cloud Computing using Various Task Scheduling Algorithms. International Journal of Computer Applications. 142 (7), p30-32.
8. Ruonan Lin et al. (2016). Task scheduling algorithm based on Pre-allocation strategy in cloud computing. IEEE. 1 (1), p227-232.
9. Gunasekaran Manogarana et al. (2016). MetaCloudDataStorage Architecture for Big Data Security in Cloud Computing. Elsevier. 8 (1), p128-133.