

Multi Scheduling Reactive Resource Sharing for Dynamic Dataflow in Cloud Environment

L. Gokila, V. Poongodi and Dr.K. Thangadurai

Abstract--- *In recent years cloud parallel data processing has emerged to be one of the killer applications for Infrastructure-as-a-Service (IaaS) clouds. Major Cloud computing companies have started to integrate frameworks for parallel data processing in their product portfolio making it easy for customers to access these services and to deploy their programs. The growing computing demand from industry and academia has lead to excessive power consumption which not only impacting long term sustainability of Cloud like infrastructures in terms of energy cost but also from environmental perspective. The problem can be addressed by replacing with more energy efficient infrastructures, but the process of switching to new infrastructure is not only costly but also time consuming. Cloud being consist of several extended centers under different administrative domain, make problem more difficult. Thus, for reduction in energy consumption, the proposed work address the challenge by effectively distributing compute-intensive parallel applications on cloud. This proposes a Meta scheduling algorithm which exploits the heterogeneous nature of Cloud to achieve reduction in energy consumption.*

Keywords--- *Cloud Service, Scheduling, Distributed Service, Dataflows.*

I. INTRODUCTION

CLOUD have an ability to utilize scalable, distributed computing environments within the confines of the Internet, a practice known as cloud computing. In this new world of computing, users are universally required to accept the underlying premise of trust. Within the cloud computing world, the virtual environment lets users' access computing power that exceeds that contained within their own physical worlds. Cloud computing is the process of providing computer facilities via internet. And it's provided us better and efficient way to access information in timely manner and also increases storage of capacity for user in.

Cloud Computing (or cloud for short) is a compelling technology. In clouds, clients can dynamically allocate their resources on-demand without sophisticated deployment and management of resources. Key enabling technologies for clouds include the Map Reduce programming paradigm

distributed file systems virtualization and so forth. These techniques emphasize scalability, so clouds can be large in scale, and comprising entities can arbitrarily fail and join while maintaining system reliability. Distributed file systems are key building blocks for cloud computing applications based on the Meta information programming paradigm. In such file systems, nodes simultaneously serve computing and storage functions; a file is partitioned into a number of chunks allocated in distinct nodes so that server tasks can be performed in parallel over the nodes. For example, consider a word count application that counts the number of distinct words and the frequency of each unique word in a large file. In such an application, a cloud partitions the file into a large number of disjointed and fixed-size pieces (or file chunks) and assigns them to different cloud storage nodes. Each storage node (or node for short) then calculates the frequency of each unique word by scanning and parsing its local file chunks. In such a distributed file system, the load of a node is typically proportional to the number of file chunks the node possesses. Because the files in a cloud can be arbitrarily created, deleted, and appended, and nodes can be upgraded, replaced and added in the file system the file chunks are not distributed as uniformly as possible among the nodes. Load balance among storage nodes is a critical function in clouds. In a load-balanced cloud, the resources can be well utilized and provisioned, maximizing the performance of Map Reduce-based applications. State-of-the-art distributed file systems (e.g., Google GFS and Hadoop HDFS) in clouds rely on central nodes to manage the metadata information of the file systems and to balance the loads of storage nodes based on that metadata. The centralized approach simplifies the design and implementation of a distributed file system.

A. Effective Meta Scheduling Dynamic Computing Environment

Cloud computing is an internet based computing where virtual shared servers provide software, infrastructure, platform, and other resources and hosting them to customers on a pay-as-you-use basis. The reason is that whatever technologies this propose have known from many authors are not fully worthy and not enough to give integrity to the cloud data for its level best in cloud itself. Because they have not given full security and integrity to get satisfied to user with cloud data from CSP and it is not so efficient by using a few technologies such as Flexible distributed scheme and Reed Solomon Technique to save and give more integrity from modification, deletion, and append of data from cloud server. So to give yet sophisticated and efficient integrity to the cloud data so that all applications, software and highly valuable things cannot be affected and modified according to someone attempt from un-trusted cloud, This propose have to process

L. Gokila, M.Phil., Research Scholar, Department of Computer Science, Thanthai Hans Roever College, Perambalur. E-mail: gokil2016123@gmail.com

V. Poongodi, Assistant Professor, Department of Computer Application, Thanthai Hans Roever College, Perambalur.

Dr.K. Thangadurai, Head, PG & Research Dept. of Computer Science, Government Arts College, Karur, Tamil Nadu, India.

DOI : 10.9756/BIJDM.8307

and provide more security methods apart from previous methods with the help of CSP who maintains our data from cloud servers from their IP address domain. But since we do not have a copy of outsourced data and data from its local site, CSP can behave unfaithfully regarding the status of our outsourced data. And since we do not have any physical data possession this propose have to fully depend on CSP for cloud data access without any separate login for our self access in data management.

It is one of the main drawbacks that existing systems have in its module in its security level. So This propose have some restriction in this regard so that CSP only can take more access to manage our data with his IP address domain. But here This propose need to maintain data integrity for our cloud data in efficient manner for our self usage whenever there is a need for that and must have some strong security measurements for our data stored to protect from internal and external attack including byzantine as well as malicious attack. And in this case, there are some different measures such as file distribution algorithm with dynamic concept, error location correctness and data verification using distributed protocol for its integrity. But all these come from existing system with a few limitations without studying data integrity in efficient manner using cloud data weight or value. Here as proposed one, this propose have done storage measurement and space comparison algorithm with time management for making efficient way for data integrity. To our credit, the proposed work have done a few new implementation and they are categorized as,

1. New implementation using the technique of Storage measurement and space comparison algorithm.
2. Cloud storage database management with automatic time specification with the help of CSP or automatically in users audit when user updates the cloud data.
3. Efficient calculation of storage space before and after the data update.
4. Server failure recovery access point in data storage and Dynamic operations in data update in each server block. Data needs a dynamic computing infrastructure. The basis for the dynamic infrastructure is a standardized, scalable, and secure physical infrastructure. There should be a level of redundancy to ensure high level of availability. Next, it must be virtualized.

II. RELATED WORK

Laney, “ data management: Controlling data volume, velocity, (2001) variety Distributed Computing Environment (DCE), developed at IBM Transarc research laboratory, provides a user with the ability to store and access knowledge at remote sites, just like the techniques used with Network classification system (NFS). Structurally, DCE DFS could be a assortment of many file systems that square measure mounted onto one virtual classification system area with a single namespace. The top user has direct access to all or any files during this distributed file system without knowing wherever the physical files reside. Putting file systems onto

different servers so as to supply the optimum service for the top users, moreover as optimize the use of accessible resources, is load equalization of DFS servers.

J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,(2008) Cloud Computing is a technology, which connects so many nodes together for allocating resources dynamically. Different types of technologies are used in clouds such as Map Reduce programming paradigm, distributed file systems, virtualization. These kinds of techniques are scalable which can add or delete new nodes or systems making it reliable. In a large cloud we can add thousands of nodes together. The main aim is to allocate files to these nodes without making heavy load to any of the nodes, for that files are partitioned into different modules.

M. Gatti, P. Cavalin, S. B. Neto, C. Pinhanez, C. dos Santos, D. Gribel, and A. P. Appel, “Large-scale multi-agent-based modeling and simulation of microblogging-based online social network Access to files(2014). Load balancing is the main issue in large scale distributed computing. It is the process of distributing tasks to all nodes involved in cloud computing. Efficient allocation of resources to every computing task helps to achieve resource utilization ratio and high user satisfaction. Minimizing resource consumption, avoiding bottlenecks, implementing fail-over, enabling scalability, reducing network inconsistencies and solving network traffic are the main goals of load computing. Whole cloud gets fail while analyzing existing system clouds performance bottleneck due to failure of central node. Functional difficulties and technical difficulties are caused because of those failures.

M. Terho, and J. Vlasenko, “Failure prediction based on log files using random indexing and support vector machines (2013) Cloud computing is a relatively new way of referring to the use of shared computing resources, and it is an alternative to having local servers handle applications. Cloud computing groups together large numbers of computer servers and other resources and typically offers their combined capacity on an on-demand, pay-per-cycle basis without sophisticated deployment and management of resources.

Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, “Cloud-based software platform for big data analytics in smart grids (2013) The end users of a cloud computing network usually have no idea where the servers are physically located, they just spin up their application and start working. This flexibility is the key advantage to cloud computing, and what distinguishes it from other forms of grid or utility computing and software as a service (SaaS). The ability to launch new instances of an application with minimal labor and expense allows application providers to scale up and down rapidly, recover from a failure, bring up development or test instances, and roll out new versions to the customer base.

L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, “S4: Distributed stream computing platform (2010) The machine’s load is thus determined by summing over several virtual nodes’, creating a tight con cent ration of load near the

average. Based upon consistent hashing which requires virtual copies to be operated for every node. The node will occasionally check its inactive virtual nodes, and may migrate to one of them if the distribution of load in the system has changed. Since only one virtual node is active, the real node need not pay the original Chord protocol's multiplicative increase in space and band width costs. Our solution to this problem therefore allows nodes to move to arbitrary addresses; with this freedom we show that we can load balance an arbitrary distribution of items, without expending much cost in maintaining the load balance

Storm: Distributed and fault-tolerant realtime computation [Online]. Available: <http://storm.apache.org/>, 2014 Cloud computing could be a comparatively new approach of pertaining to the employment of shared computing resources, associated it's a completely different to having native servers handle applications. Cloud computing teams along giant numbers of pc servers associated alternative resources and usually offer their combined capability on an on-demand, pay-per-cycle basis while not refined readying and management of resources.

B. Satzger, W. Hummer, P. Leitner, and S. Dustdar, "Esc: Towards an elastic stream computing platform for the cloud (2011)The tip users of a cloud computing network sometimes don't have any plan wherever the servers are physically situated; they solely spin up their application and begin operating. In an exceedingly giant cloud we are able to add thousands of nodes along. The most aim is to allot files to those nodes while not creating significant load to any of the nodes, for that files are divided into completely different modules. Another objective is to scale back the network inconsistencies and network traffic as a result of the unbalancing of masses.

A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production cloud services (2010)Cloud computing refers to applications and services offered over the Internet. These services are offered from data centers all over the world, which collectively are referred to as the "cloud." This metaphor represents the intangible, yet universal nature of the Internet. The idea of the "cloud" simplifies the many network connections and computer systems involved in online services. In fact, many network diagrams use the image of a cloud to represent the Internet. This symbolizes the Internet's broad reach, while simplifying its complexity. Any user with an Internet connection can access the cloud and the services it provides. Since these services are often connected, users can share information between multiple systems and with other users.

I. Moreno, P. Garraghan, P. Townend, and J. Xu, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud (2014) The main goal of our project is to decrease the overloads of the main cloud and increase the performance of the cloud. In recent years ad-hoc parallel data processing has emerged to be one of the killer applications for Infrastructure-as-a-Service (IaaS) clouds. Major Cloud computing companies have started to integrate frameworks for parallel data processing in their product portfolio, making it easy for customers to access these services and to deploy their

programs. However, the processing frameworks which are currently used have been designed for static, homogeneous cluster setups and disregard the particular nature of a cloud. The main objective of our project is to decrease the overloads of the main cloud and increase the performance of the cloud by segregating all the packet of the cloud by cloud storage, packet manager and task manager. and perform the different task using different resources as the infrastructure needed.

Many-Task Computing for Cloud and Supercomputers

Many-task computing aims to bridge the gap between two computing paradigms, high throughput computing and high performance computing. Many task computing differs from high throughput computing in the emphasis of using large number of computing resources over short periods of time to accomplish many computational tasks (i.e. including both dependent and independent tasks), where primary metrics are measured in seconds (e.g. FLOPS, tasks/sec, MB/s I/O rates), as opposed to operations (e.g. packet) per month. Many task computing denotes high-performance computations comprising multiple distinct activities, coupled via file system operations. Tasks may be small or large, uniprocessor or multiprocessor, compute intensive or data-intensive. The set of tasks may be static or dynamic, homogeneous or heterogeneous, loosely coupled or tightly coupled.

A. Problem Definition

A growing number of companies have to process huge amounts of data in a cost-efficient manner. Classic representatives for these companies are operators of Internet search engines. The vast amount of data they have to deal with every day has made traditional database solutions prohibitively expensive. The need for low latency analysis over high-velocity data streams motivates the need for distributed continuous dataflow systems. Contemporary stream processing systems use simple techniques to scale on elastic cloud resources to handle variable data rates. However, application QoS is also impacted by variability in resource performance exhibited by clouds and hence necessitates autonomic methods of provisioning elastic resources to support such applications on cloud infrastructure. The prior concept of dynamic dataflows which utilize alternate tasks as additional control over the dataflow's cost and QoS. Further, we formalize an optimization problem to represent deployment and runtime resource provisioning that allows us to balance the application's QoS, value, and the resource cost. The greedy heuristics, centralized and shared, based on the variable-sized bin packing algorithm and compare against a Genetic Algorithm (GA) based heuristic that gives a near-optimal solution. A large-scale simulation study, using the linear road benchmark and VM performance traces from the AWS public cloud, shows that while GA-based heuristic provides a better quality schedule, the greedy heuristics are more practical, and can intelligently utilize cloud elasticity to mitigate the effect of variability, both in input data rates and cloud resource performance, to meet the QoS of fast data applications. Instead, these companies have popularized an architectural paradigm based on a large number of commodity servers. Problems like processing crawled documents or regenerating a web index are split into several independent

subtasks, distributed among the available nodes, and computed in parallel.

- The cloud’s virtualized nature needs higher resource utility to enable promising new use cases for efficient parallel data processing.
- However, it also imposes new challenges compared to classic cluster setups.
- The major challenge this propose see is the cloud’s opaqueness with prospect to exploiting data locality.

III. PROPOSED SOLUTION

We develop a practical cloud resource allocation algorithm. It is capable of both overload avoidance and green computing. It is applicable to scenarios with multi-dimensional resource constraints. We give a theoretical proof that the number of servers in where d is the number of bottleneck resources..Especially, with only one bottleneck resource, the number of servers in use is bounded by times of the optimal value. We also prove that the number of VM migrations is bounded as well. We conduct extensive simulation and experiments. The results demonstrate good performance compared with the existing work.

The load of applications encapsulated in the VMs may change dynamically. The resource demands of applications can be highly variable due to the time of the day effect, sudden surge in user requests, and other reasons. Moreover, some applications such as on-demand data processing and virtual desktop have random load. It is hard to forecast their resource demands. Although server consolidation has the effect of absorbing load fluctuation, the sum of the peak resource demand of VMs sharing a physical machine (PM) can be much higher than the mean. Thus, it is often inefficient to over-provision resources based on the peak demand because doing so will leave the resources under-utilized most of the time. Live migration allows a VM to be moved from one server to another without interrupting the application running inside. This can be used to adjust the VM layout for load balancing or energy saving purpose. For example, when the resource

utilization of a server becomes too high, some VMs running on it can be migrated away to reduce its load. On the contrary, when the average server utilization in the system becomes too low, the VMs can be aggregated to a fraction of the servers .

Meta scheduling is designed to run data analysis packet on a large amount of data, which is expected to be stored across a large set of share-nothing commodity servers. Once a user has fit his program into the required map and reduce pattern, the execution framework takes care of splitting the packet into subtasks, distributing and executing them. A single Map Reduce packet always consists of a distinct map and reduce program. , we focus on the first sub problem the problem of host overload detection. Detecting when a host becomes overloaded directly influences the QoS, because if the resource capacity is completely utilized, it is highly likely that the applications are experiencing resource shortage and performance degradation. What makes the problem of host overload detection complex is the necessity to optimize the time-averaged behavior of the system, while handling a variety of heterogeneous workloads placed on a single host. To address this problem, most of the current approaches to dynamic VM consolidation apply either heuristic-based techniques, such as static utilization thresholds decision-making based on statistical analysis of historical data or simply periodic adaptation of the VM allocation. The limitations of these approaches are that they lead to suboptimal results and do not allow the administrator to explicitly set a QoS goal. In other words, the performance in regard to the QoS delivered by the system can only be adjusted indirectly by tuning parameters of the applied host overload detection algorithm. In contrast, our approach enables the system administrator to explicitly specify a QoS goal in terms of a workload independent QoS metric. The underlying analytical model allows a derivation of an optimal randomized control policy for any known stationary workload and a given state configuration. It is important to note that the model proposed in this study is based on Markov chains requiring a few fundamental modeling assumptions.

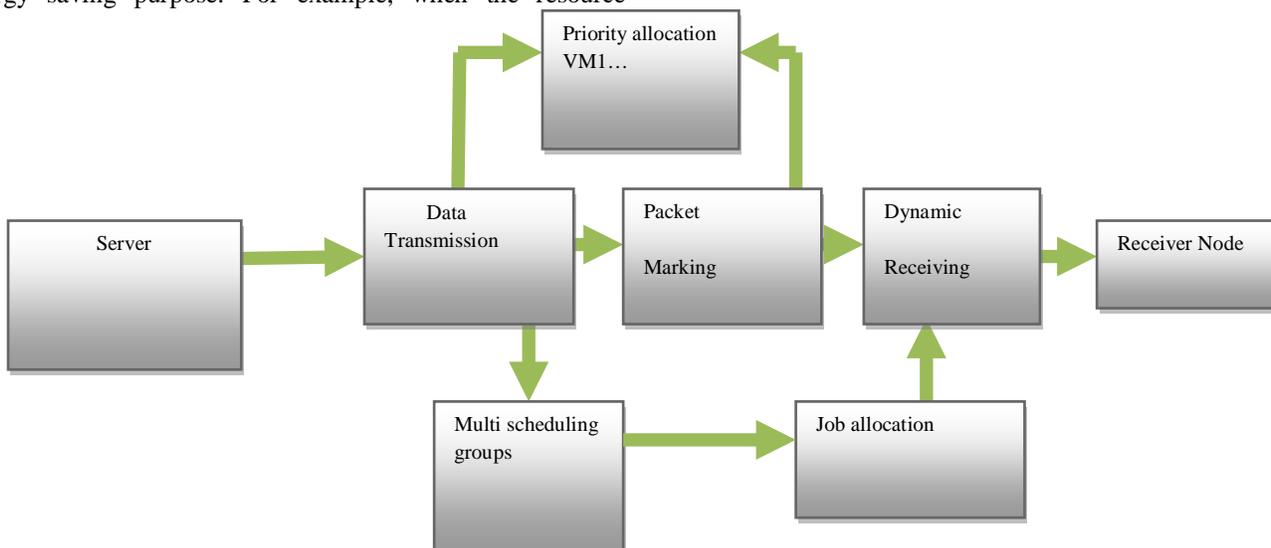


Figure 3.1: Architecture Diagram for Meta Flow Model

Meta Scheduling Algorithm

Best-Fit VM with VM Clustering Algorithm (Newly Proposed)

Input: Migrating VM ClusterList and Available Host List Partition

Output : Migration Map

Sort hosts in Available Host List by descending order of current CPU utilization

1. For each host in Available Host List do
2. If Migrating VM meta info List is not empty then
3. Find optimal combination of VM multi scheduler in Migrating VMC meta dataList for that host using dynamic programming
4. For each VMCluster in optimal combination of VMClusters do
Add (VMCluster, host) pair to MigrationMap
Delete VMCluster from MigratingVMList
Else
Break
5. If any optimal combination of VM meta does not fit any host then
Create new host
Add (VM, host) pair to Migration Map
Delete VM meta info from Migrating VM meta List

Figure 3.1 show the workload must satisfy the Markov property, which implies memory less state transitions and an exponential distribution of state transition delays. These assumptions must be taken into account in an assessment of the applicability of the proposed model to a particular system.

A. Resource Allocation Packing Data

The load of applications encapsulated in the VMs may change dynamically. The resource demands of applications can be highly variable due to the time of the day effect, sudden surge in user requests, and other reasons. Resource to allocate from the server. Moreover, some applications such as on-demand data processing and virtual desktop have random load. It is hard to forecast their resource demands. Although server consolidation has the effect of absorbing load fluctuation, the sum of the peak resource demand of VMs sharing a physical machine (PM) can be much higher than the mean. Thus, it is often inefficient to over-provision resources based on the peak demand because doing so will leave the resources under-utilized most of the time.

B. Remote Method Workload Analyzing

For provisioning to be meaningful, the query workload has to exhibit some predictability. We assume the availability of a representative workload, which can be obtained either manually (e.g., by the application provider or the administrators) or automatically (through DBMS profilers). We require that the representative workload consists of a set of query classes along with the expected distribution of queries across the different classes. In either case, the elasticity of cloud resources allow quick re-provisioning, which ameliorates the negative impact of inaccurate workload predictions.

C. Scheduling the Server to Response

The approach presented periodically performs an offline bin packing algorithm to calculate a new VM layout, where hot spots are eliminated and the number of servers in use is minimized.

Scheduling response are then migrated accordingly. one server to another without interrupting the application running inside. This can be used to adjust the VM layout for load balancing or energy saving purpose. The new layout, however, may be quite different from the existing one, which may incur a large number of VM migrations. Wang et al. adopts an online bin packing algorithm which incurs fewer migrations. The focus of that work is on the network bandwidth only and does not consider multi-dimensional resource constraints

D. VM Live Migration of Data

We present an approach that uses live migration to allocate data center resources dynamically based on application demands and supports green computing by optimizing the number of servers used. We make the following contributions. We develop a practical bin packing resource allocation algorithm. It is capable of both overload avoidance and green computing. It is applicable to scenarios with multi-dimensional resource constraints. We give a theoretical proof that the number of servers, where d is the number of bottleneck resources. Especially, with only one bottleneck resource, the number of servers in use is bounded by the optimal value. We also prove that the number of VM migrations is bounded as well. We conduct extensive simulation and experiments. The results demonstrate good performance compared with the existing work.

E. Remote Flow Data Response

Cloud computing refers to delivery of computer resources from a remote place based on user needs. Network connections are necessary to access information and utilize resources. According to multi cloud computing can be defined as, a style of computing, where massively scalable IT-enabled capabilities are delivered „as a service“ to external customers using Internet technologies. Switching remote data to increase the challenges on how to transfer and where to store and compute data are the issues caused by large distributed file systems in cloud computing. Cloud Computing Technologies such as Map Reduce paradigm, virtualization and Distributed File Systems are used to achieve scalability and reliability in clouds.

IV. RESULT AND DISCUSSION

Construct analytical models to quantify the network performance of scientific workflows using cloud-based computing resources, and formulate a task scheduling problem to minimize the workflow end-to-end delay under a user-specified financial constraint. We rigorously prove that the proposed problem is not only resource complete but also non-approximable. We design a heuristic solution to this problem, and illustrate its performance superiority over existing methods through extensive simulations and real-life workflow experiments based on proof-of-concept implementation and deployment in a local cloud tested.

Performance of the proposed algorithms is evaluated in this section. The experiments were performed on a 2.80 GHz Intel core 2 duo Processor with 3.5 GB memory. The operating system is Microsoft Windows 7. The simulated that are implemented in java language. Both real and server client vm are used in the experiments. File process were generated from the data generator with responding time management. Real world Server extended, process are obtained from virtual Repository scheduling is optimized with server is obtained from shared resource is acquired from scheduling.

VM migration techniques have been investigated for non real-time applications. utilize the live migration technology to move the VMs away from the lightly loaded physical servers and thus the empty servers can be switched to low-power mode. It create a dynamic server consolidation algorithm with migration control and avoid unnecessary migrations to reduce

the number of powered on servers and migration cost. To find that virtual machines do not usually use all their resources, and they create an algorithm which also considers the migration cost according to the records of migration history for saving energy

The optimized strategy divides the execution of the workflow into multiple stages; and at each stage, bandwidth and resources are allocated independently. And also in every stage, the minimization algorithm is executed to allocate the optimum number of resources for the services involved in the current stage. An algorithm is required to determine the number of stages. Directed execution graphs (DAG) are made up of the workflow of services. The DAG is divided into the execution stages. A special virtual infrastructure executes the execution stages. In every execution stage, the infrastructure is reconfigured for deployment for services involved in the stage.

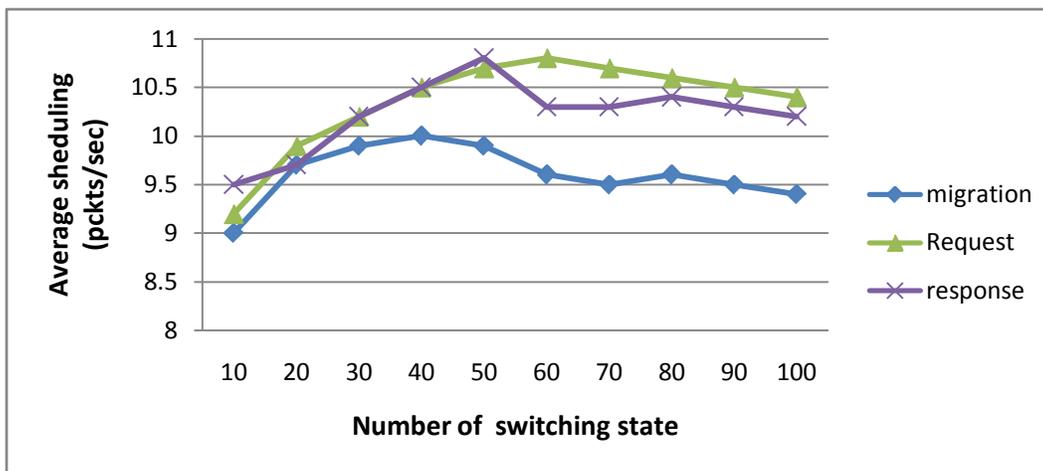


Figure 4.1: Scheduling Request/Response State

Figure 4.1 shows the resources are allocated based on the number of invocations required for each service. At present, many research studies on balanced machine resources scheduling are based upon dynamic migration of machines.

Thus, using dynamic migration all of the above systems can achieve system load balance; but frequent dynamic migration would employ a large number of resources that might lead to degrading the entire system performance.

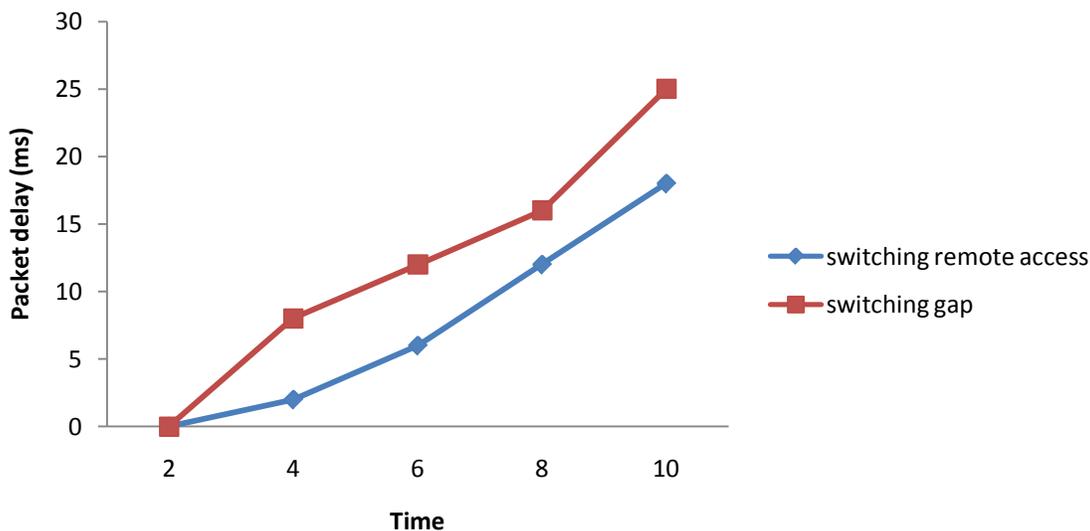


Figure 4.2: Time Complexity in Data Switching VM

Figure 4.2 shows the server switching in every data inclusion or any other activity by unauthorized person or any internal and external attack coming into the CSP address domain, one server access point or restore point is given to the

cloud server when the client is doing some delete, modification, and append operations in his will. This is a new technique that This propose have introduced in our system design.

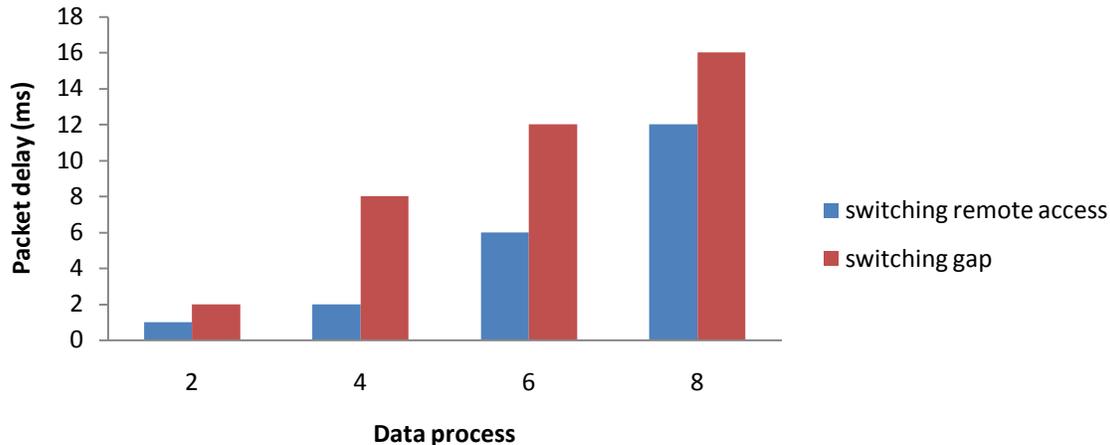


Figure 4.3: Performance of Data Process

Figure 4.3 shows the process is preceded by fixing one access point to server performance when this propose finally finish our data exchange in cloud server. It is a major drawback to our cloud server to maintain our individual or group data. In the days to come, it can be rectified with the help of CSP.

V. CONCLUSION

To conclude multi scheduling ensures minimum end-to-end data transmission for the highest priority data while exhibiting acceptable fairness towards lowest-priority data. Experimental results show that the proposed multi scheduling scheme has better performance than the existing and Multilevel Queue Scheduler in terms of the average task waiting time and end to-end delay. As enhancements to the proposed multi scheme, we envision assigning task priority based on data's deadline instead of the shortest task processing time. To reduce processing overhead and save bandwidth, we could also consider removing tasks with expired deadlines from the medium. Furthermore, if a real-time task holds the resources for a longer period of time, other tasks need to wait for an undefined period time, causing the occurrence of a deadlock. This deadlock situation degrades the performance of task scheduling schemes in terms of end to end delay. Hence, we would deal with the circular wait and preemptive conditions to prevent deadlock from occurring. We would also validate the simulation result using a real test-bed.

REFERENCES

- [1] D. Laney, "3d data management: Controlling data volume, velocity, and variety", META Group Research Note, Vol. 6, 2001.
- [2] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters", ACM Commun., Vol. 51, No. 1, Pp. 107–113, 2008.
- [3] M. Gatti, P. Cavalin, S.B. Neto, C. Pinhanez, C. Dos Santos, D. Gribel, and A.P. Appel, "Large-scale multi-agent-based modeling and simulation of microblogging-based online social network", Proc. 14th Int. Workshop Multi-Agent-Based Simul., Pp. 17–33, 2014.

- [4] I. Fronza, A. Sillitti, G. Succi, M. Terho and J. Vlasenko, "Failure prediction based on log files using random indexing and support vector machines", J. Syst. Softw., Vol. 86, No. 1, Pp. 2–11, 2013.
- [5] A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos and C. Moran, "IBM infosphere streams for scalable, real-time, intelligent transportation services", Proc. Int. Conf. Manage. Data, Pp. 1093–1104, 2010.
- [6] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids", Comput. Sci. Eng., Vol. 15, No. 4, Pp. 38–47, 2013.
- [7] L. Neumeyer, B. Robbins, A. Nair and A. Kesari, "S4: Distributed stream computing platform", Proc. IEEE Int. Conf. Data Min. Workshops, Pp. 170–177, 2010.
- [8] Storm: Distributed and fault-tolerant realtime computation [Online]. Available: <http://storm.apache.org/>, 2014.
- [9] M. Zaharia, T. Das, H. Li, S. Shenker and I. Stoica, "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters", Proc. USENIX Conf. Hot Topics Cloud Comput., 2012.
- [10] B. Satzger, W. Hummer, P. Leitner and S. Dustdar, "Esc: Towards an elastic stream computing platform for the cloud", Proc. IEEE Int. Conf. Cloud Comput., Pp. 348–355, 2011.
- [11] V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente and P. Valduriez, "Streamcloud: An elastic and scalable data streaming system", Trans. Parallel Distrib. Syst., Vol. 23, No. 12, Pp. 2351–2365, 2012.
- [12] A. Iosup, N. Yigitbasi and D. Epema, "On the performance variability of production cloud services", Proc. IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., Pp. 104–113, 2011.
- [13] A. Li, X. Yang, S. Kandula and M. Zhang, "Cloudcmp: Comparing public cloud providers", Proc. ACM SIGCOMM Conf. Internet Meas., Pp. 1–14, 2010.
- [14] I. Moreno, P. Garraghan, P. Townend and J. Xu, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud", IEEE Trans. Cloud Comput., Vol. 2, No. 2, Pp. 208–221, 2014.
- [15] A. Kumbhare, Y. Simmhan and V. K. Prasanna, "Exploiting application dynamism and cloud elasticity for continuous dataflows", Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., 2013.