AUTO SCALABLE BIG DATA AS-A-SERVICE IN THE CLOUD: AN EMPIRICAL STUDY

Naseer Ahmad Shinwari[1], Dr. Neeta Sharma[2]
M.Tech Pursuing[1], Noida International University, Greater Noida, (U.P.)
Assistant Professor[2], Dept of CSE, Noida International University, Greater Noida, (U.P.)

Abstract: big data is what most industries are working with for growing their businesses and making timely decisions to win the competitive data oriented market. Huge size and complexity of data have made people move their data to the cloud since cloud providers claim the automatic scalability of resources for applications to prevent demand unmet and resource wastage. Most cloud service providers do provide scalability but fail to meet SLA level QoS, especially in case of big data processing. Because their approach to scalability is based on prediction systems and threshold based which fails for sudden load changes and the time they take for provisioning new resources is degrading the performance. In this paper we have proposed a model for auto scalable big data as-a-service services of the cloud by combining vertical and horizontal scalability to prevent resource provisioning latency and be able to handle sudden changes in load to meet SLA level QoS.

Keywords: **auto-scalability, unpredictable load balancing, big data as-a-service, cloud scalability.**

I. INTRODUCTION:

Efficient processing of big data is essential for crucial and critical decision making in current era where every decision is data oriented. But it is quite challenging due to the huge volume, high velocity, varying variety, and uncertain value of big data. This is where traditional systems and approaches of data handling and processing fail and new technologies like Hadoop are used for dealing with big data tasks, which in turn requires much bigger computers and computational resources to properly avail big data sets and gain competitive advantages. This is where cloud computing comes into picture to provide computational resources on demand, as and when needed forming Big data as-a-service and Hadoop as-a-service. Big data and tools for handling and processing big data are taken to the cloud because of its elastic and scalable nature since Big data's varying features state that the volume, velocity and variety of data cannot be trusted and may change quite rapidly which will sure require placing more servers, hiring more staff and dealing with more burden which eventually leads to higher costs in case of in-house computation. Why do all these if we can avail all the computational resources from the cloud as utility and pay for only what we have used.

Big data as-a-service services of the cloud should be elastic and have the ability to scale as per user requirements and application demand to satisfy Service Level Agreement (SLA) and meet Quality of Service (QoS). These are the only reasons alongside the pay-as-you-go (you pay only for what you use) model of the cloud why people move their big data to the cloud for effective and limitless computing.

Thanks to the virtualization technology which made it easy for cloud service providers to provision/de-provision Virtual Machines (VMs) and/or change a virtual machine's memory capacity and/or CPU core numbers through software configuration. Most current solutions for scalability in the cloud benefit from this virtualization feature (scalability in the cloud is mostly achieved through altering virtual machine's capacity and capability). For deciding when and how to scale a system there are multiple scaling techniques as: (1) reinforcement learning, (2) static and threshold-based policies, (3) control theory, (4) queuing theory and, (5) time series analysis. Among which some are threshold based while others are based on forecasting load and provisioning resources based on predictions made using historical data and behavior of the application. But in case of big data these measure fail, because the nature of big data states high velocity and uncertain load. Even if the predictions are made accurate, provisioning new virtual machines take tens of minutes which leads to the fear of demand unmet and/or resource wastage (see section II) which leads to either degraded performance which in turn leads to customer dissatisfaction and/or unnecessary costs incurred, none of which is bearable.

To overcome these issues, we propose an auto scalable model which fits well for big data as-a-service services in the cloud in specific and may as well have fruitful results for other domains in the cloud which require automatic scalability on demand.

II. RESOURCE WASTAGE & UNMET DEMAND:

Over-provisioning and under-provisioning are the cases which lead to resource wastage and unmet demand. If we always keep our resources for the peak level demand then there will be cases that lead to resource wastage, else if we reserve resources for the normal case (average usage) then times will come when the demand is high but resources are not available as per demand which results in unmet demand.

In both cases both parties (customer and service provider) are losing something, in resource wastage (over-provisioning), service provider is wasting resources reserved by customer but not utilized for any purpose while customer is paying for resources he/she is not using. In the other hand, in demand unmet (under-provisioning) customer's application require more resources but are not available at the moment resulting in poor performance and thus unhappy customer which turns for service provider to lose customers.

In an auto scalable cloud environment, resources are provisioned based on the application demand which prevents unmet demand and resource wastage, as depicted in figure 01 the left hand side show a case where the scalability in controlled manually which leads to unmet demand and resource wastage, while in the right hand side show the case where there is an auto scalable system which solves both the problems, customers pays for only what he/she is using.

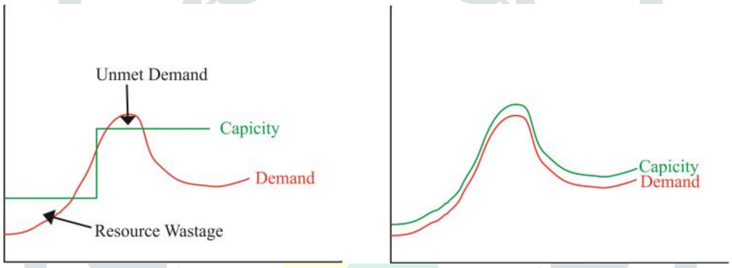

Figure 1 unmet demand & resource wastage in presence and absence of auto scalability

III. RELATED WORK:

Che-Lun Hung et. al. [1] proposed a cloud auto-scaling architecture and an auto-scaling algorithm which contains three main components: front-end load balancer, virtual cluster monitor system and auto-provisioning system with an auto-scaling algorithm.
Mehran N. A. H. Khan et. al. [2] conducted an overview of 'the auto scaling operations' on various commercial cloud service providers with the purpose of identifying common features and entities from these operations. Their observation of the study states that these operations mainly adopt the rule-based approach by using a set of metrics collected at the infrastructure and platform level and thereby they presented a model that consist of parameters whose values calibrate the auto-scaling workflow in time series data collected for these metrics. Their model explores time series data and workload prediction technique to capture the dynamics of the auto-scaling workflow.

Yang Xia et. al. [3] proposed a scalable framework enabling cloud powered execution for scientific workflows. The framework supports automatic cluster provisioning and scaling. They also proposed and algorithm to translate workflows so that operations within a flow are also parallelized.

Olubisi Runsewe et. al. [4] proposed a layered multi-dimensional hidden markov Model (LMD-HMM) for facilitating the management of resource auto-scaling for big data streaming applications in the cloud. Their architecture includes a data ingestion layer, a processing layer and a storage layer. To support resource scaling, present within the architecture is the resource controller that is made up of a resource monitor, a predictor and a resource allocator.

Anshul Gandhi et. al. [5] presented the design and implementation of a model-driven autoscaling solution for Hadoop clusters for the concern of unforeseen events such as node failures and resource contention. For dynamically adding or removing nodes from Hadoop cluster, they created a customized VM image preloaded with Hadoop. A new slave can dynamically be added by booting a new VM using the customized image, and then starting the TaskTracker and DataNode services on it. And for removing a slave node, they update the exclude list on the Master and dynamically refresh the node configuration.

Zhenlong Li et. al. [6] proposed an auto-scaling framework to automatically scale cloud computing resources for Hadoop cluster based on the dynamic processing workload, with the aim of improving the efficiency and performance of big geospatial data processing. Their framework contains following components (1) Cloud computing platform, (2) CoveringHDFS enabled Hadoop cluster, (3) Auto-scaler, and (4) Cluster monitor.

Nilabja Roy et. al. [7] made three contributions to overcome the general lack of effective techniques for workload forecasting and optimal resources allocation, their work mainly describes a look-ahead resource allocation algorithm based on model predictive techniques which allocates or de-allocates machines to the application based upon optimizing the utility of the application over a limited prediction horizon.

Yazhou Hu et. al. [8] proposed a prediction framework for virtual machines provisioning which includes three main modules: monitor, filter and predictor, with the aim of predicting upcoming workload in order to overcome virtual machine provision latency. Moreover for processing raw data they proposed the Kalman filter method and as based predictor they presented five different prediction models, namely: moving average (MA), auto regression (AR), auto regression integrated moving average (ARIMA), neural networks (NN) and support vector machine (SVM).

Dang Tran et. al. [9] worked on a proactive auto scaling model for cloud computing which consists of two major components, namely: prediction and scaling decision. For the prediction module they used fuzzy approach to process multivariate monitoring resources, genetic algorithm, back-propagation, and neural network with the purpose of efficient and precise forecasting; and for the decision module they proposed a formula to calculate SLA violations, then the SLA-aware data is sent back to system in order to integrate with predicted values to adapt their auto scaling model.

IV. PROPOSED MODEL:

As mentioned in section I and II we have multiple auto-scaling methods and techniques which are either reactive and/or proactive and are mostly based on forecasting load to provision resources accordingly. These methods mostly proceed with scaling in/out (horizontal scaling) which takes tens of minutes to provision and bring a VM into operation, while some also worked on scaling up/down (vertical scaling) which is not the ultimate solution for many applications specially in big data domain, since a VM may grow in size but there is always a peak level beyond which VMs cannot get bigger and there is no choice left but to scale out which again takes tens of minutes and hence degrading application performance.

Even if someone cop up the VM provision latency in case of scaling in/out which seems to be an ultimate solution for auto scaling, this fails because of the unpredictable load which is pretty common in most fields and domains specially when we are talking about big data. High traffic of data (load) never sends any signal prior to happening and thus cannot be predicted.

Considering these issues, we proposed a model for automatic reactive and proactive scaling in the cloud environment by combining scaling in/out (horizontal scaling) and scaling down/up (vertical scaling) approaches to best fit for each scenario, (a) either predictable load changes or unpredictable load happing in real time, (b) or scaling the infrastructure for a sudden fix for a short term load or scaling the infrastructure for a permanent fix for scenarios where the load remains high.

Figure 2 demonstrates a bigger picture of proposed model where the Virtual Machine cluster contains all the running VMs, VM monitor module is responsible for monitoring load on each virtual machine and evaluating performance of the entire VM cluster to decide action based on load and performance, if there is a need for scaling, the Auto Scalar decides to either scale up/down and/or scale in/out.

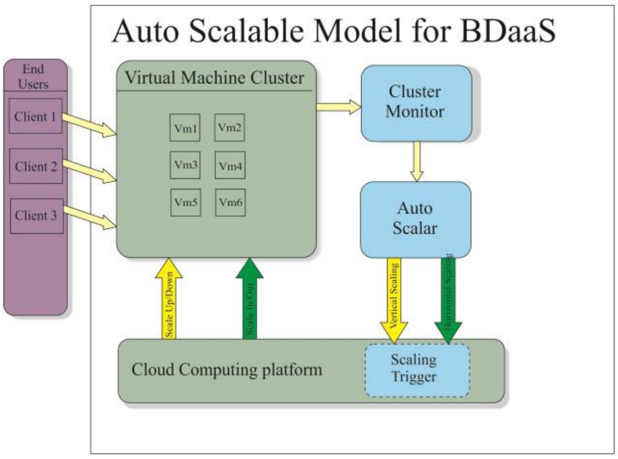

Figure 2 proposed model (bigger picture)

Figure 3 demonstrates a detailed picture of the proposed model, where the VM Cluster module contains two types of nodes (Master and Slave); (1) Master node has the list of all slave nods and thus manages and distributes work among them, (2) slave nodes can have two variations (Core Slave and Compute Slave); (2.a) Core slaves are those nodes which can both store data and perform computation on it, (2.b) compute nodes are the slaves which serves the purpose of computation only and cannot store data. The concept of separating core and compute slaves is introduced by [6] which have fruitful results for big data processing since there are times when we only need storage capacity of some nodes rather than its computational power. This concepts plays a vital role is scaling in where we have to remove some VMs when we are not optimally utilizing its computational power or may be the storage capacity of some nodes to safely move its data and shutdown the node. The second module is Cluster monitor which monitors the cluster to check if VM cluster needs to be expanded or shrunk down and tells the Auto Scalar module to act accordingly, while the second part of Cluster Monitor is Resource predictor which tries to predict the amount of resources required for meeting SAL's QoS. Based on the call for scaling the Auto Scalar module may only decide from any of the two options available, either Sudden Fix which triggers the Vertical Resource Scalar or Long Term fix which triggers the Horizontal Resource Scalar.

The advantage with this method is that load changes occur so instantly that we may not accurately predict it and provision new VMs as well takes time; we can first apply Sudden Fix by triggering the Vertical Resource Scalar since it takes few seconds as per [10], while the client's request is serving and at the same time if load continues to increase or even remain constant we can use to the time to provision new VM(s).

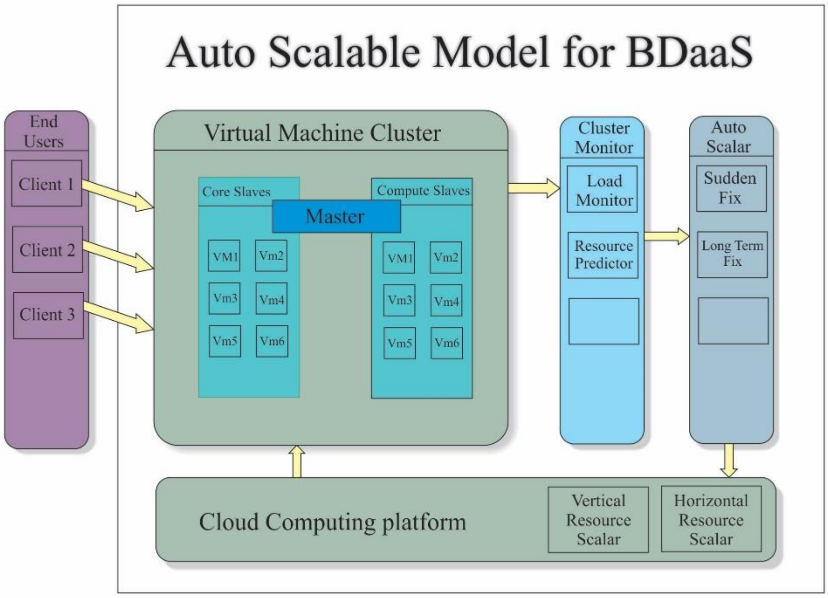

Figure 3 proposed model (detailed)

V. Conclusion:

In the study we had conducted in [11] our findings stated some problems with current auto scalability approaches for big data as-a-service. Continuing our study, here we proposed a model for automatic reactive and proactive scalability in the cloud for processing big data. Our model suggests best solution for unpredictable load changes and guarantees decreased auto scalability latency by combining scaling in/out and scaling up/down approaches. Our proposed model will be an ultimate solution not only for big data domain but for auto scalability of cloud resources. In our future work we will implement our solution in a cloud environment and test it on big data sets.

References:

[1] Che-Lun Hung et. al. “Auto-scaling model for cloud computing system”, international journal of hybrid information technology Vol. 5, No. 2, April, 2012.

[2] Mehran N. A. H. Khan et. al. “Modeling the auto-scaling operation in cloud with time series data”, 2015 IEEE 34th symposium on reliable distributed systems workshop, DOI 10.1109/SRDSW.2015.20.

[3] Yang Xia et. al. “A scalable framework for cloud powered workflow execution”, Globcom 2013 workshop – cloud computing systems, networks and applications.

[4] Olubisi Runsewe et. al. “Cloud resource scaling for big data streaming applications using a layered multi-dimensional hidden markov model”, 2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing, DOI 10.1109/CCGRID.2017.147.

[5] Anshul Gandhi et. al. “Autoscaling for hadoop clusters”.

[6] Zhenlong Li et. al. “Automatic scaling Hadoop in the cloud for efficient process of big geospatial data”, (2016) International journal of Geo-Information, DOI 10.3390/ijgi5100173.

[7] Nilabja Roy et. al. “Effective autoscaling in the cloud using predictive models for workload forecasting”, 2011 IEEE 4th international conference on cloud computing. DOI 10.1109/CLOUD.2011.42.

[8] Yazhou Hu et. al. “Autoscaling prediction models for cloud resource provisioning”. 2016 2nd IEEE international conference on computer and communications.

[9] Dang Tran et. al. “A proactive cloud scaling model based on fuzzy time series and SLA awareness”, international conference on computational sciences, ICCS 2017.

[10] Turowski, M., & Lenk, A. (2015). Vertical scaling capability of OpenStack. In Service-Oriented Computing-ICSOC 2014 Workshops (pp. 351-362). Springer, Cham.

[11] Naseer Ahmad Shinwari, & Neeta Sharma. (2019). Auto Scalable Big Data as-a-service In The Cloud: A Literature Review- IJRAR (volume 6, issue 1) (pp. 20-24).