Resource Provisioning of Multi-tier Web Applications in Cloud Computing

Tejinderjit Singh
Research Scholar, RIET Phagwara, Punjab India.

Parminder Singh
Assistant Professor, RIET Phagwara, Punjab, India.

Abstract – Web applications these days are mostly multi-tier for the sake of software reusability and flexibility. However it is very difficult to estimate the behavior of workloads in these tiers due to the fact that resource demand at each tier is distinct. So it becomes very confronting to allocate resources at each tier. In order to fulfill SLA (Service Level Agreement) requirements it becomes necessary to provision resources at each tier separately. Recently various approaches on control theory and dynamic approaches have been proposed for the provisioning of resources in cloud. In this paper we identify those challenges and conduct a review on those control theory and dynamic approaches for allocation of resources in multi-tier applications. On the basis of advantages and disadvantages we develop a blueprint for research directions to enhance the effectiveness of resource provisioning in multi-tier applications.

Index Terms – Web service, multi-tier architecture, resource provisioning, web applications.

1. INTRODUCTION

As there is tremendous advancement in web technologies so internet is moving towards versatile web services [11, 12, 13]. The best example these days could be online trading. Also we know cloud computing is all about availability of resources on demand in scalable and dynamic fashion [14]. In this process services are provided on lease to the users by the service providers from large scale data centers. Currently most of the web applications are currently deployed as multi-tier applications consisting of basically presentation layer, application layer, data tier. However various web sites such as amazon, yahoo etc. do not deliver static web pages but they deliver user customized web pages. Web pages generated for each user are not created by a single application but hundreds of applications running simultaneously [1].

As all the tiers have different set of requirements of computing resources so it becomes a challenge to monitor both Quality of service (QOS) and resource cost. To overcome this difficulty generally various QOS requirements are used (i.e. throughput, delay, latency) and are generally stated in SLA. When defined QOS satisfies SLA then service providers get revenue otherwise they have to pay penalty to the users. So ultimately our objective is to provide appropriate resources to the users at each tier so as to reduce the overall cost. According to Jiang et.al. to provide optimal resource provisioning in cloud service providers often impose as a SLA (Service Level Agreement) which defines various parameters that application must offer i.e. Maximum average response time, Latency etc.[2]

Now capacity planning is a classical method to estimate the amount of resource usage. It is determined by maximum request rate to a system in a given target period. We can estimate using either periodic data or historic data. However maximum request rate occurrence is very rare. Figure 1 shows the demonstration of 24 hours of working of system with CPU utilization and disk utilization. We can clearly conclude that CPU usage for most of the time is less than 50% and disk utilization is below 20% for about 70% of the time.

Fig. 1. Average CPU utilization and peak disk utilization in a production SAP application server for a 24-hour period [5].

However with emerging virtualization technology it becomes feasible to allocate the resources according to the fluctuations in resource demand. Particularly cloud paradigm is concerned with “Pay as you go”. For example amazon charges $0.1 per hour per virtual instance.

A general scenario of 3 tier application is shown in figure 2. Although single tier is easier to implement than multi-tier and has simple architecture but it lacks functionality. Moreover most modern web sites these days use multi-tier architecture. In multi-tier architecture each tier has certain functionality.

The given figure basically three tiers are used namely presentation tier, application tier and data tier and these tiers
are implemented as Web server, application server and data server respectively.

![3-tiered web application architecture](image)

**Fig. 2. 3-tiered web application architecture.**

Although there are tremendous advantages of cloud computing but it comes with many challenges which are discussed below.

1. First challenge is that each class of resource has different impact on QOS. So it becomes very challenging to predict the allocation of resources to users.
2. Secondly in multi-tier applications each tier has substantially different set of requirements. So it becomes much more difficulty to cope with this problem dynamically.
3. No performance guarantees are given by the service providers with regard to application level performance.

We will analyze whether current approaches are able to cope up with above mentioned challenges. Recently various algorithms have been proposed which can be divided either in the category of control theory or dynamic. Let’s have a look over these techniques.

**2. RESOURCE PROVISIONING APPROACHES**

Although various techniques are available for the provisioning of resources we will take a look over them one by one. Some techniques have disadvantages as well as advantages but optimal resource provisioning technique is not yet available. It is very difficult to understand the overall good understanding of the behavior of whole system [3]. Basically we divide various approaches on the basis of control theory and dynamic approaches.

**Control theory**- This theory has been used for adaptive resource management for different classes of applications. Feedback control methods have been used to regulate operating parameters of the system. It assumes a closed loop controller and linear time-invariant system. Closed loop controller is based on an open loop system transfer function that satisfies sensitivity and stability constraint. Control theory helps in meeting QoS objectives and operating constraints that can be expressed as a cost of function or stand-alone web server, database server etc. Structure of an optimal controller is shown in figure 3.

**Dynamic**- Dynamic techniques are employed where the demand for workloads is either unpredictable or there is large amount of fluctuations. In dynamic techniques VM’s concept comes into picture. We vary assigned VM’s according to the fluctuations in workloads. Providers assign more resources when demanded and removes excessive resources when not in use.

![Structure of an optimal controller](image)

**Fig. 3. Structure of an optimal controller**

**3. CONTROL THEORY APPROACHES**

SLA-driven resource provisioning of end to end performance instead of dealing with individual tier separately. In this approach run time caches are used which further enhances the efficiency and performance. With the use of caches similar requests can be processed faster and efficiently but keeping in mind the consistency problems. This depends on cache hit ratio. Various advantages of this approach is that it deals with end to end performance. Moreover SLA of service can be maintained with minimum number of servers. However some disadvantages also persist with this approach that is it is very difficult to maintain consistency of caches. If data in caches it too old then it needs to be updated in order to maintain consistency. However many algorithms are available to tackle with this problem [1]. SLA (API) is applied only to the front end service and rest of the services are automated. Using this approach various services can be provisioned whether an application or data service. Each service may consist of number of instances with replicated code and various caches in order to improve performance. In this paper they are creating acyclic graph in which services are adjusted. Each service computes its “What if analysis” according to number of available machines i.e. predicting future performance and these performance promises are reported to their parent services. Each service consists of aggregated performance values of their child nodes. Responsibility is totally of the intermediate service for provisioning of resources locally. One advantage of this approach is that it is a decentralized approach where each service is responsible autonomously for its own provisioning. Limitation also persists that they are assuming that the availability of the resources is infinite. So this technique is only subjected to data centers and clouds [2]. Statistical machine learning approach firstly offline training is provided to machines and tested under diverse workloads offline. When the system is used in real-time workloads regression analysis is used to model the internet service behavior pattern whenever the service is violated then system uses predefined metrics and act accordingly.

Technique are that once the system has cleared training process it is fully automated. However some cons also exist 1. Training
should be efficient. If training is not up to the mark then overall system will be affected. 2. Caches or queuing model is not deployed so this approach is not suitable for heavy workloads as it will compromise with the efficiency [3]. Proposing admission control and scheduling mechanisms to tackle with the dynamic workload nature of cloud computing. Process of scheduling used is,

1. Admission control: Admission control decides whether a VM can be allocated to the request to serve and if accepted then it signs SLA with the user.
2. VM manager: VM manager initiates a VM and allocate it to the request then job scheduler schedules applications on this VM.
3. Job scheduler: Job scheduler assigns jobs to initiated VM’s.
4. SLA manager: SLA monitors each accepted application. SLA enforcement and rescheduling algorithm is used. It performs following functions. 1. Enforce SLA. 2. Schedule jobs from batch job. 3. Consolidation [7].

The historical data is used and on the basis of that data they allocate the resources i.e. 1. Predictive. 2. Reactive. In predictive they are using historical data and on the basis of that data they allocate the resources but with the use of reactive technique we can recomputed the demand of resource at finer times. Advantage of this approach is that this model is simple and efficient for small centers. However there are also some disadvantages that queuing model is not used and this approach is not suitable for large centers where multiple requests arrive in a small time span [8]. Machine learning approach in which are deployed in front of the applications. These caches use a genetic algorithm that identifies the request that can be most benefited from caches and dynamically resizes itself so as to accommodate all requests. In architecture of V-cache there are:

1. Workload analyzer – It performs following functions. □ Clustering of requests according to their size, types And processing costs. □ it also maintains statistics of completed requests.
2. Policy generator – It performs following functions. □ Identifies the request that benefits the most from the caches. It takes cluster of requests as input and request redirection map to request redirectors as output. □ It also determines the size of cache to accommodate those requests.
3. Request predictor – It performs following functions. □ It determines whether to send or forward the request to cache tier or to forward the request to web tier on the basis of URL and host sent by policy generator.

We can extend work further for heterogeneous applications and integrating admission control for overload control and performance guarantees [12].

4. DYNAMIC APPROACHES

Applying Kalman filters to dynamic resource provisioning of virtualized server applications have proposed two controllers based on Kalman filters which adapt according to changing workloads. Moreover we can make this mechanism to adapt more quickly or slowly to workload changes and also parameters for these controllers can be computed online. In this method controllers are used to allocate the resources to the CPU. In each CPU there is manager that periodically sends CPU usage to controller and controller provides resources accordingly. In this approach they are using two controllers. 1. Basic controller. 2. Process noise covariance controller. Although in previous works only basic controllers were used but with the addition of PNCC further improvement is achieved by considering resource coupling tiers. Moreover PNCC adaptive is used to take non-stationary noises under consideration. Although kalman uses an efficient approach but this technique is suitable with small servers only maintaining controller with each CPU and linking those controllers is a difficult task to achieve and makes system complex [4]. Discrete particle swarm optimization algorithm (DSPO) are creating t+p matrix and each row representing task allocation and column representing allocated tasks in a processor. With the use of this method we are able to achieve the speed of convergence and ability to obtain faster and feasible allocation. However speed of convergence and obtaining faster and feasible allocation are the advantages of this approach. But cons are that dynamic workloads are not taken into the consideration. Moreover this approach is not suitable for large systems [5]. To remove this problem another technique for dynamic resource provisioning under cluster based virtualized multi-tier applications.

It consists of resource pool where the physical machines are present in data center along with virtualized applications running on them. Advantage of using virtualized machines is that the demand of the resources increases or decreases with respect to time so in order to cope with this problem virtualized environment is used. It consists of resource pool and self-management community. In self-management community basically four functions are performed. 1. Monitor. 2. Analyzer. 3. Resource scheduler. 4. Virtualized application executor. Resource computational pool sends delegate to self-management community and SMC generates the response accordingly. Under virtualized multi-tier application queuing model we are having analytic performance models. In this model they are using ÖDS (On Demand Scheduler) for the first tier only to schedule the request. Once the request is scheduled it only needs to be processed and passed to successor tiers. In open queuing model first step is to estimate the capacity of
VM’s at each tier in accordance to the rate of request they can handle.

Process the request and satisfy the requirements. Advantage of using this approach is increased efficiency and flexibility for cloud data centers. To pursue further research we can use load prediction method technique [6]. Moreover predictive and reactive techniques a flexible queuing model is also incorporated for determining how many resources are needed to each tier. Two basic things which we need to consider while provisioning resources are, 1. How much to provision and 2. When to provision. In this paper the required capacity at each tier is evaluated using the queuing model thus allocating the desired capacities at various tiers all at once. Various advantages of this approach are efficient and flexible model. Also queuing model along with predictive and reactive techniques further enhances the capability of model [9]. Rather than adding or removing whole virtual machine instances because latter bears more cost and overhead. They are using LS algorithm. According to it if value of t0>tu (LSU) then we need to scale up resources and if the value of t0<tl (LSD) then we need to scale down resources. Now tL= Lower bound and tu = Upper bound. To use this approach effectively we need to check the performance periodically. Various advantages of this approach are firstly this approach is simple and efficient and secondly it adapts cost effective scaling. We can extend this approach by scheduling the resources between application and also understanding tradeoffs between reservation cost and risk of high running cost [10]. Calculate performance profile of each individual instance and then these parameters help us to balance the load more efficiently and more accurately rather than allocating the load on the basis of parameters calculated from overall tier. They controlled workload by applying a load balancer in front of provisioned instance. Then load balancer calculates weighted workload distribution. Now whenever a new instance is added in any tier load balancer calculates its performance profile in order to balance the workload intensities. In this paper heterogeneous instances are taken into account. So it accommodates various advantages firstly heterogeneous nature is considered and secondly better allotment of instances [11].

5. CONCLUSION AND FUTURE WORK

In this review paper we presented two approaches on the basis of which we can divide all the techniques. Our review concludes that although large number of techniques are available for provisioning of resources in multi-tier applications yet approaches which take heterogeneity of applications are limited and lot of areas are needed to be explored. Moreover controllers can be tuned to react more quickly to workload changes.

REFERENCES