USING DYNAMIC WORKLOADS FOR EFFICIENT SERVER PROVISIONING APPROACH WITH DECISION MAKING

Ms.S.Anna Suganthi[1], Mr.K.Karnavel[2].
[1] PG Scholar, Dept of CSE, Anand Institute of Higher Technology,Kazhipattur Chennai, India. [2](M.E.(Ph.D).,Asst. Professor, Dept of CSE, Anand Institute of Higher Technology, Kazhipattur, Chennai, India.
First:anasuganthi@gmail.com

Abstract-- **Dynamic virtual server provisioning is critical to quality-of-service assurance for multitier Internet applications. In this paper, we address the important challenging problems. First, we propose an efficient server provisioning approach on multitier clusters based on an end-to-end resource allocation optimization model. It is to minimize the number of virtual servers allocated to the system while the average end-to-end response time guarantee is satisfied. Second, we design a model-independent fuzzy controller for bounding an important performance metric, the 90th-percentile response time of requests flowing through the multitier architecture.**

Third, to compensate for the latency due to the dynamic addition of virtual servers, we design a self-tuning component that adaptively adjusts the output scaling factor of the fuzzy controller according to the transient behaviour of the end-to-end response time. Fourth, to crash the application server and store the database in to main server after some time revoke the application server with database, the provisioning approach is able to significantly reduce the number of virtual servers allocated for the performance guarantee compared to an existing representative approach.

Keywords—**Multi-tier Clustering, Utilization Control, Response Time, Crashing.**

1 INTRODUCTION

A computer network, or simply a network, is a collection of computers and other hardware components interconnected by communication channels that allow sharing of resources and information. Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network. Simply, more than one computer interconnected through a communication medium for information interchange is called a computer network. Applications used to access the cloud. The back end provides the applications, computers, servers, and data storage that creates the cloud of services. Efficient Server Provisioning with Control for End-to-End Response Time Guarantee on Multitier Clusters: Networking (ie) the process between the source and destination is accessible by network only,- Distributed sites like Amazon.com and Brokerage Sites are typically need Internet application employs Multi-tier Cluster architecture. And each tier of the applications has the separate functionality.

Dynamic resource provisioning is critical to quality-of service assurance for Internet applications. The problem was well studied in the context of single-tier servers. It is however, nontrivial or even infeasible to extend mechanisms designed for single tier architecture to a virtualized multitier architecture. The challenges include the inter tier interaction, concurrency limits, and cross-tier Dependencies. For example, adding servers to one tier does not necessarily increase the effective system performance due to cross-tier dependencies. End-to-end response time is the major performance metric of multitier Internet applications. It is the response time of a request that flows through a multitier system.

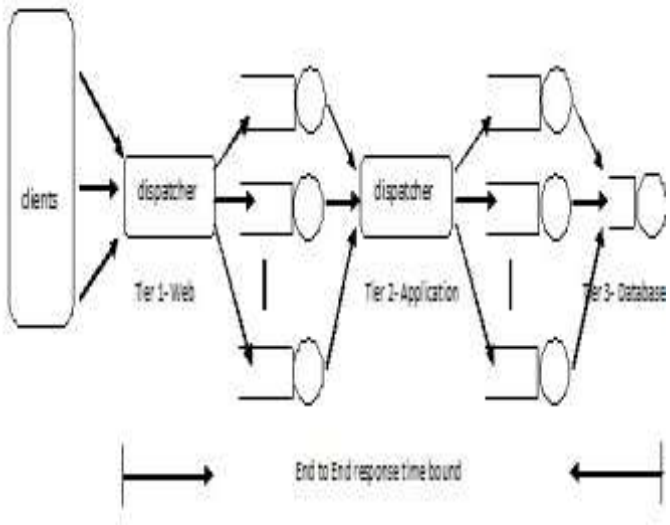

Figure 1.1 multitier server cluster architecture and end-to-end response time

2 RELATED WORK

Resource management for quality-of-service provisioning in multitier Internet applications is a very important and active research topic. Queuing model-based approaches was proposed for network server performance control [1]. Recently, there has been a few studies on the modelling and analysis of multitier servers with queuing foundations [3] For instance, described a performance model for differentiated services of multitier applications [3]. A tier-to-tier management architecture was designed for delay control with a M=M=1 queuing model. Per-tier concurrency limits and cross-tier interactions were addressed in the model. The work in proposed an analytic model for session-based multitier applications using a network of queues. The mean value analysis algorithm for queuing networks was used to measure the mean response time. The queuing model-dependent approaches, however, are not effective in providing percentile-based end-to-end response time guarantee. Resource allocation optimization has been applied for single-tier Internet server performance improvement. For example, the work in studied an optimization for allocating servers in the application tier that increase a server provider's profits. An optimization problem is constructed in the context of a set of application servers modelled as M=G=1 processor sharing queuing systems. That single-tier provisioning method does not consider the end-to-end response time constraint. Recently, they designed an important dynamic provisioning technique on virtualized multitier server clusters [6]. It sets the per-tier average response time targets to be certain percentages of an end-to-end response time bound. Based on a queuing model, per-tier server provisioning is executed at once for the per-tier response time guarantees. The work provides important insights on dynamic virtual server provisioning for multitier clusters. There is, however, no guidance nor optimization about the decomposition of end-to-end response time to per-tier response time targets. Furthermore, it relies on a queuing model with application profiling for the 90th-percentile response time guarantee. It translates a 90th-percentile response time-based service level agreement into a new service level agreement based on the average response time. The application profiling process is executed offline to find the 95thpercentile of the distribution of a workload before the server replication and allocation. It uses the mean of that distribution for the new service level agreement based on a queuing model. The profiling process itself could be time consuming and complex due to the dynamic nature of Internet workloads. In our work, this service level agreement translation is not required since we use a model independent fuzzy controller to guarantee the 90th-percentile response time without assuming an accurate workload model. The fuzzy controller is designed based on heuristic knowledge through trial and error. However, it is done only once before the deployment. Importantly, the rules formed for the fuzzy controller provide a generalized control policy irrespective of the distribution of dynamic workloads. Our work provides resource allocation efficiency merit by the integration of the optimization model and the fuzzy controller and crashes the particular application server and revoke the server after crash. As the studies in [7], our work assumes using virtual machines for dynamic server allocation in a multitier cluster. Feedback control has been used in real-time systems for long time. Designed a utilization control algorithm (EUCON) for distributed real-time systems in which each task is comprised of a chain of subtasks distributed on multiple processors [5]. It is based on a model predictive control approach that models utilization control on a distributed platform as a multivariable constrained optimization problem. Wang et al. extended it to a decentralized algorithm, called DEUCON [8]. In contrast to the centralized control schemes, DEUCON features a novel decentralized control structure that requires only localized coordination among neighbour processors. Feedback control has also been used for service differentiation and performance guarantee on Internet servers linear control techniques were applied to control the resource allocation in single-tier Web servers [2]. However, the performance of the linear feedback control is often limited. Recent work applied adaptive control for performance guarantee [6]. For instance, a multitier e-commerce application was modelled as one M=G=1 server [6]. A proportional integral (PI) controller-based admission control proxy was developed to provide the end-to-end response time guarantee. However, using the average response time as the performance metric is unable to represent the shape of a response time curve. For the end-to-end response time guarantee, the inherent process delay needs to be considered and addressed. Fuzzy control was applied for performance differentiation and guarantee in computer networks and systems. In [9], fuzzy control was used to determine an optimal number of concurrent child processes to improve the aggregated server performance. Fuzzy controller was used for provisioning guarantee of user-perceived response time of a web page. It demonstrated that due to the model independence, the approach significantly outperforms linear PI controllers. The work was done on a single server. We use fuzzy control for dynamic server provisioning with end-to-end response time guarantee in multitier server architecture, together with a resource

allocation optimization model. We also consider the use of non-uniform membership function and self-tuning capability for fine granularity control of the system performance.

3 EFFICIENT SERVER PROVISIONING APPROACH

We propose an optimization-based server provisioning [6] scheme that minimizes the total number of virtual servers allocated to a multitier cluster while the end-to-end response time guarantee is satisfied. The basic idea is to divide the provisioning process into a sequence of intervals. In each interval, based on the measured resource utilization, end-to-end response time, and the predicted workload, the servers are allocated to the tiers at once. We model the workload of each virtual server at each tier by one M=G=1 queuing system.

We consider the use of homogeneous virtual servers. First we assume that requests at a virtual server are processed with the FCFS principle. Then, we consider the principle of processor sharing for concurrent request processing at a virtual server. Both FCFS and processor sharing disciplines are explored with the optimization model and the fuzzy control integration.

4 SYSTEM ARCHITECTURE

Datacentre running an environment often contains a large number of machines that are connected by a high-speed network First thing in this project is design for the user going to get the fair and statistical guarantee response from the server, the client who need the information about the such esteem profile, and make communication for the esteem. Generally server allows one particular mean, possible to allow the client by means of his/her valid, because the server maintain database of valid clients. If user needs to search the information in server he should face the login otherwise he should register their profile to server. Then the server allows the user to process the internet application like the web server and the application server. The application server always contains the information about the specific esteem.

The web server always interlinked with the number of application servers and does the end-to-end response to the client. In the project the client get the response from the server by reasonable time. If any application server have get crash at the time, the particular application server database will be stored in to the main server it means web server, After revoking the crash from application server that will handle the database. By using rule based system main server will request to particular application server and the main server will send and receive the request in the form of fuzzification and defuzzification manner it means numeric inputs in to fuzzy values and vice versa.

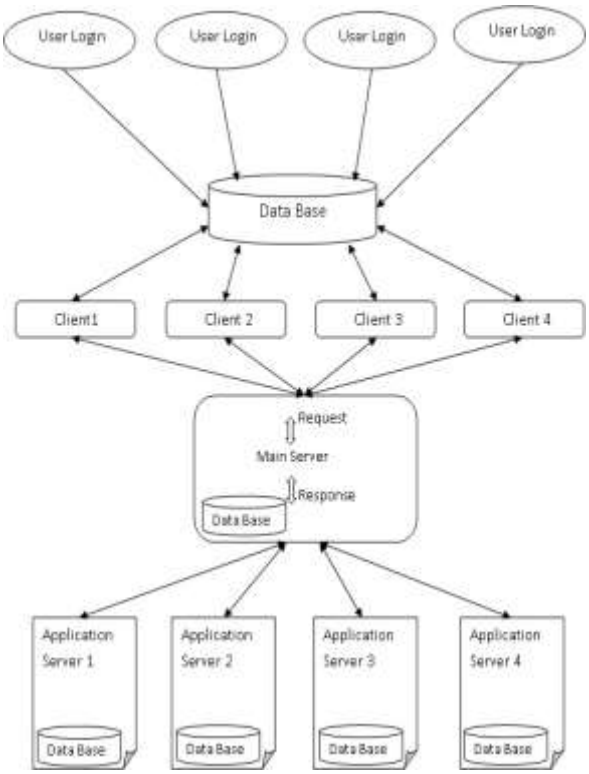

Figure 4.1 System architecture

MULTI-CLIENT CLUSTERS

In this Module, we focus on the problem of a multi-Client heterogeneous cluster acceptable response times (i.e., a response time that is faster than the pre-specified response time) in the presence of Server Provisioning. Response time is an important part of the system's Quality-of-Service (QoS), and is also the main concern of the client. We argue that for a heterogeneous cluster to handle client-server applications, a Server that enables dynamic sharing of heterogeneous resources is necessary. The n number of client hit server using that their own configuration the server will response the specific client by using their configuration.

IMPLEMENTING APPLICATION SERVER

we design three application server in this paper, each application server have the separate data base and the server having the functionality of add some components and retrieve component and remove such things for information of particulars. Application server, a server dedicated to running certain software applications.

SERVER PROVISIONING OVER WEB-CLUSTERING

The web server starts means the Server initially loaded to sub server information. If any application server has any modification means the web server will monitor the modification. The clustering technique is to cluster the application server into the web server.

ADAPT AND HANDLING –END-TO-END RESPONSE

The application servers are clustered with Web server. The web server plays the vital role in it. The web server get all clients request and analysis the request using the fuzzy Rule base mechanism, and find the information related to the request, and send the request to specific application server but the value are changed object into fuzzy values using the Fuzzy inference mechanism like fuzzification and defuzzification, application server process the request using the database and return response to the web server. Here deffuzzification process is done that is fuzzy values in to numeric values. After the response goes the Particular client with the End-to-End Response time [4] guaranteed. In this paper three application servers is running at same time. Server response the client by using the application server features. If any application server is such failed for some time but not long time, at the time of situation the web server handle the process, because server takes the backup of the failed application server. If application server is revoked for the crash, it will continue it execution

5. SELF INDEPENDENT FUZZY CONTROLLER

The server provisioning [6] scheme based on the optimization, however, is model dependent. We enhance it with a self-tuning fuzzy controller that is model independent. The fuzzy controller determines the number of servers to be allocated to each tier at once without relying on an accurate performance model. It is to guarantee the average end-to end response time on a multitier system, to bind the 90thpercentile end-to-end response time, and to integrate the fuzzy controller with the optimization model. To provide fine granularity control on the response time and efficient resource utilization, we consider the use of both uniform and non-uniform membership functions in the fuzzy controller. The self-tuning capability is to compensate for the process delay due to the addition of a server to a tier. It is achieved with a scaling-factor controller ***Rule base***- The rule base is the core component. It contains a set of rules based on which fuzzy control [9] decisions made.

Fuzzification-The fuzzification interface converts numeric values of controller inputs into equivalent fuzzy values. It determines the certainties of fuzzy values based on input membership functions

Inference-The inference component applies predefined rules according to the fuzzified inputs and generates fuzzy Conclusions

Deffuzzification-The deffuzzification interface combines fuzzy conclusions and converts them to a single output, i.e., the resource allocation adjustment in a numeric value

6 PERFORMANCE EVALUATION

We evaluate the server provisioning approach based on the optimization model alone, the model-independent fuzzy control [9] system, and the integrated approach in a three-tier server cluster simulation model. Here, FCFS scheduling discipline is assumed for each virtual server, for performance evaluation with the processor sharing discipline. We assume that the database tier can be replicated on demand. We use a synthetic session-based work load generator derived from a customer behaviour model. It allows us to perform sensitivity analysis in a flexible way. A session generator produces head requests that initiate sessions. The subsequent requests of a session are generated according to the customer behaviour model. The think time was generated by an exponential distribution with a mean of 5 seconds. We use bounded Pareto distributions that are representatives for modelling the service time distribution in Internet applications. During the simulation, the end-to-end response time was measured periodically with a sampling interval of 1 minute. Each result reported is an average of 100 runs.

7 CONCLUSION

Able to significantly reduce the number of virtual servers allocated for the end-to-end response time guarantee of multitier Internet applications, suitable for long term process. Response time is satisfied by the client. This type of implementation is suitable for the heavy load internet application like railways. In the future work, we will implement the approach in a prototype data center and we will discuss the nontrivial heterogeneous server configuration in virtualized multitier systems.

8 REFERENCE

[01] L. Sha, X. Liu, Y. Lu, and T. Abdelzaher, "Queuing Model Based Network Server Performance Control," Proc. IEEE Real-Time Systems Symp. (RTSS), 2002.

[02] J. Chen, G. Soundararajan, and C. Amza, "Autonomic Provisioning of Backend Databases in Dynamic Content Web Servers," Proc. IEEE Int'l Conf. Autonomic Computing (ICAC), 2006.

[3] Y. Diao, J.L. Hellerstein, S. Parekh, H. Shaihk, and M. Surendra, "Controlling Quality of Service in
Multi-Tier Web Applications," Proc. IEEE 26th Int'l Conf. Distributed Computing Systems (ICDCS), 2006.

[4] X. Liu, L. Sha, and Y. Diao, "Online Response Time Optimization of Apache Web Server," Proc. Int'l Workshop Quality of Service (IWQoS), 2003.

[5] C. Lu, X. Wang, and X. Koutsoukos, "Feedback Utilization Control in Distributed Real-Time Systems with End-To-End Tasks," IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 6, pp. 550-561, June 2005.

[6]B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood, "Agile Dynamic Provisioning of MultiTier Internet Applications," ACM Trans.
Autonomous and Adaptive Systems, vol. 3, no. 1, pp. 1-39, 2008.

[7] P. Padala, K.-Y. Hou, K.G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant,
"Automated Control of Multiple Virtualized Resources," Proc. European Conf. Computer Systems (EuroSys), 2009.

[8] X. Wang, C. Lu, and X. Koutsoukos, "DEUCON: Decentralized End-To-End Utilization Control for
Distributed Real-Time Systems," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 7, pp. 996-1009, July 2007.

[9] Q. Zhang and Y.A. Phillis, "Fuzzy Control of Arrivals to Tandem Queues with Two Stations," IEEE Trans. Fuzzy Systems, vol. 7, no. 3, pp. 361367, June 1999.

