Control Systems application in Java based Enterprise and Cloud Environments – A Survey

Ravi Kumar Gullapalli
Hewlett-Packard
Bangalore, India

Dr. Chelliah Muthusamy
Yahoo
Bangalore, India

Dr.A.Vinaya Babu
JNTUH
Hyderabad, India

Abstract— **The classical feedback control systems has been a successful theory in many engineering applications like electrical power, process, and manufacturing industries. For more than a decade there is active research in exploring feedback control systems applications in computing and some of the results are applied to the commercial software products. There are good number of research review papers on this subject exist, giving high level overview, explaining specific applications like load balancing or CPU utilization power management in data centers. We observe that majority of the control system applications are in Web and Application Server environments. We attempt to discuss on how control systems is applied to Web and Application(JEE) Servers that are deployed in Enterprise and cloud environments. Our paper presents this review with a specific emphasis on Java based Web, Application and Enterprise Server Bus environments. We conclude with the future reserach in applying control systems to Enterprise and Cloud environments.**

Keywords- Control Systems; Java, Web Servers; Application Servers; Web Services; Enterprise Service Bus.

I. INTRODUCTION

Control Systems theory has been successfully applied to several engineering applications such as electrical power, process engineering, manufacturing plants. More than a decade there is an active research in investigating the applicability of control theory in different areas of the computing systems and data networks. It provides a systematic approach to achieve service level objectives by designing appropriate feedback control loops [1]. The control theory has been explored in various areas of computing such as web servers, multi-media systems, proxy caches, database servers, multi-tier web sites and real-time systems, power management of data centers [2]. The advantages of control theory are the self-managing or self-correction capabilities that make any physical or computing system to manage itself [3] [4]. This is achieved without continuous monitoring thereby avoiding huge manual intervention. Typically the software developments teams' focus on the self-managing mechanisms is very limited and is handled in an adhoc manner based on the customer needs or the defects that come post release of the software [4]. The other major advantage is the automation that is provided by the controllers. They come with a mathematical model that justifies their stability, convergence and accuracy [4]. With the advent of the internet and changes that are occurring to businesses and their service delivery modes, there is a big necessity for the computing platforms and systems to operate in self-healing and self-managing manner. Control theory has become an important choice to provide these kinds of capabilities in the computing environments.

Most of the software application layers including enterprise and cloud applications use Web and Application Servers as platform to host the service delivery components [5]. Enterprise Service Bus is heavily used in Service Integration both in enterprise and cloud environments [6]. It is highly desirable that high performance and scalability [7] to be provided by these software applications. This triggered the researchers in investigating the applicability of control systems in Web, Application Servers with some interesting results.

We attempt to study control systems application in web and application servers and provide a useful review for either researchers or practitioners of this subject. This paper is a first step towards investigating how the control systems theory is applied to the problem areas in Enterprise and Cloud environments that employ Web Servers, Application Servers and Enterprise Service Bus. We have not discussed the experimental results achieved in the various research papers but the conclusion is highlighted as required. The figures are referred from the original papers and the references are appropriately cited. Our study is based mainly on Java based Web and Application servers as they are one of the successful platforms that host applications for service delivery either enterprise or cloud environments.

The paper is organized in the following manner. Firstly, we discuss the background of why control system is suitable for solving some of the challenges in computing environments followed by a brief overview of control system concepts. In the next set of sections we discuss how different type of controllers are being applied in Java based Web, Application Servers, and Enterprise Service Bus and Cloud environments. We conclude with our observations, possible future research areas and our next steps in this research.

II. BACKGROUND

Feedback Control Systems has been in practical application in many engineering disciplines [8]. The inherent characteristics due to the feedback phenomenon bring significant self-managing capabilities hence it is a successful theory. It brings strong mathematical concepts to model both linear and non-linear systems [9], and determine the stability and controllability of the system using techniques such as Nyquist [10], Routh-Horwitz Criterion [11]. The Control theory provides pre-defined controllers such as P, PI, and PID

Controllers. The rich theory has motivated the researchers to start experiments in using control systems in computing. The advantage of having some basic controllers is put to use in improving the performance of computing systems. In the case of distributed systems control systems is mainly used to improve performance of web or application servers by improving their caching or tuning other important parameters such as cpu, memory utilization, no of users. The study shows the feedback control system efficient in building self-managing middleware systems such as CORBA where a control service (FC-ORB) was implemented to improve the CPU utilization [12]. It is interesting to observe discrete control systems achieving IT workflow automation, easing the IT management [13]. In the case of computer networking the controllers are put to use for flow and congestion control [14][15]. There are additional applications in Database and DatawareHouse servers to improve the memory management [16]. It is interesting to note that the majority of investigation of control systems in computing environments is in distributed computing system environments [17][18][19][20] and computer networking [21][22][23][24]. We observe that feedback control mechanisms though have been well explored; still there is scope to extend the research in making it as a first class building block of the distributed environments. Additionally the recent trends in the Enterprise and cloud environments [25] demand more self-managing capabilities, which motivated us to conduct a study on this subject.

III. CONTROL THEORY PRIMITIVES

In this section we discuss types of controllers, control strategies and modeling of control systems theory briefly. The Figure 1 shows a typical feedback control system where the controller is used to tune the system to be controlled in order to achieve the desired output. The system will have external disusturbance also called noise that affect the system behaviour.

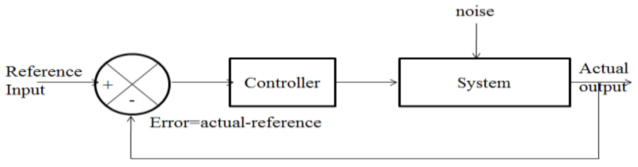

Figure 1. A typical Feedback Control System

A. *Controllers* [25]

- Proportional (P): The controller output is proportional to the error signal.
- Proportional-Integral (PI): The controller output is dependent upon the proportional and integral values to the error signal.
- Proportional-Integral-Derivative (PID): The controller output is dependent upon three separate constant parameters, proportional derivative and integral to the error signal.

B. *Control Strategies* [26]

- Hierarchical: In this type of control strategy, the governing control software is arranged in a tree. If the tree links are implemented by computer network then it is a Networked Control System.
- Robust: Robust control works within pre-defined boundaries of a data set. It is suitable in the presence of bounded modeling errors.
- Adaptive: Adaptive control involves in modifying the control law when the system parameters to be controlled are time-varying or uncertain in nature. It is widely used in self-tuning of the system to be controlled.
- Stochastic: Stochastic control is used when there is uncertainty in the data. It designs optimal control that performs the desired control task with minimum average cost despite presence of the noises.
- Optimal: Optimal control deals with finding control law of a system to achieve a certain optimality criterion are achieved. It is a set of differential equations to minimize the cost functional.
- Intelligent: The controllers that use AI computing techniques like neural network, Bayesian probability, fuzzy logic, machine learning, and genetic algorithms come under Intelligent Controllers

C. *Modeling Computing Systems*

The following are the various modeling techniques used for representing the systems on which the control techniques are applied.

- Queuing Theory: It is a mathematical model of queues. It helps in calculation of performance measures of the system to be controlled.[27]
- Model Predictive Control(MPC): MPCs [28] rely on dynamic models of the process, most often linear empirical models obtained by system identification
- Petri Nets: Petri Nets [29] are popular mathematical model languages to represent Distributed Systems. It is a bipartite graph. The nodes represent transitions and places. The directed arcs describe which places are pre-and/or post conditions for which transitions.

D. *Types of Control Systems* [30]

- Linear Systems are mathematical models that have the properties of superposition and homogeneity.
- Non-Linear systems do not satisfy the properties of superposition and homogeneity.

IV. WEB SERVERS

In this section we discuss the application of control systems in Web environments such as web servers and web services.

A. *Web Caching*

With the advent of internet the importance of Web Servers is very significant and it is obvious that they run with high performance always. There are many ways to improve the web server’s performance, Web caching being an important measure. In Web Servers it is desirable to self-manage the web cache in high traffic internet environments. Web proxy Cache provides a key differentiation in the information access on web servers and performance differentiated architecture in proxy caches is proposed in [31]. The content is classified into multiple classes to cache in the web proxy. It proves the average response time of client decreases using a Hit Rate Control Loop implemented over Squid proxy cache. A feedback mechanism is used along with space allocation heuristic to adjust the cache space based on the actual (Ri) and desired(R) relative hit ratios. The Figure 2 below illustrates the controller designed to minimize the error. The figure is redrawn from the original paper to better clarity.

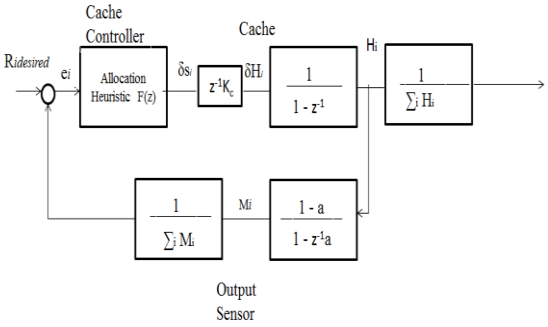

Figure 2. Web Caching Feedback loop (in z-transform)

But this method has the drawback of fixed parameters in the model that are used to adjust the cache size. Additionally, empirical and synthetic workloads are used for experimentation. The uncertainty of the parameter estimate is not incorporated in the controller design proposed in [31].

An improvement is proposed in [32] as shown in Figure 3 which is an adaptive control technique used. The controller design with on-line recursive parameter estimation for QoS guarantees in distributed environments is implemented. Additionally the parameter estimate is incorporated in the controller design. The empirical and synthetic workloads are used are improved for effective convergence of online estimates of the parameters in [32] for the stochastic workloads are considered.

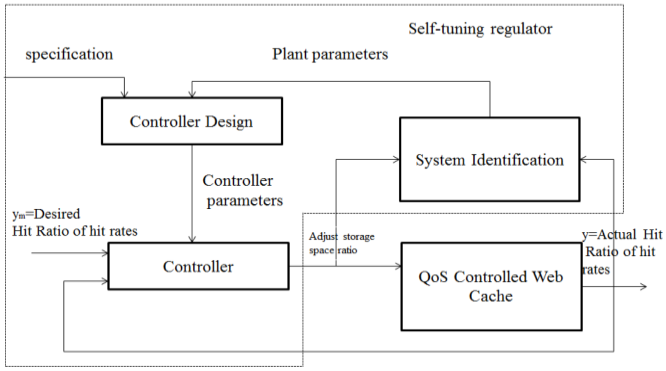

Figure 3. Adaptive QoS Web Cache control system

To summarize, in the adaptive cache control in [32], the QoS of the web cache is managed by including the sensitivity analysis which has the following factors: excitation signal, workload uncertainty and a priori knowledge of the system.

$$u(k) = \sum_{j=1}^{i} a_{i-j}(k)u(k-j) + \sum_{j=1}^{i} b_{i-j}(k)[y_m(k-j) - y(k-j)] \quad (1)$$

$a_{i-j}(k)$ and $b_{i-j}(k)$ are controller parameters. At the end of each sampling time the controller is fed with output y reference y_m and input u. The controller order is represented by l. The output should track asymptotically the reference if the estimates are accurate. But the conventional adaptive control has a dilemma between asymptotically good control and asymptotically good parameter estimate. Additionally for uncertain and dynamic work load large scale distributed environments, it is difficult to build good a priori knowledge. This constraint reduces the prediction accuracy and increases time for prediction to converge. This uncertainty is handled by implementing a dual control framework as shown in Figure 4. Which is a redrawn based on the original paper for clarity purposes.

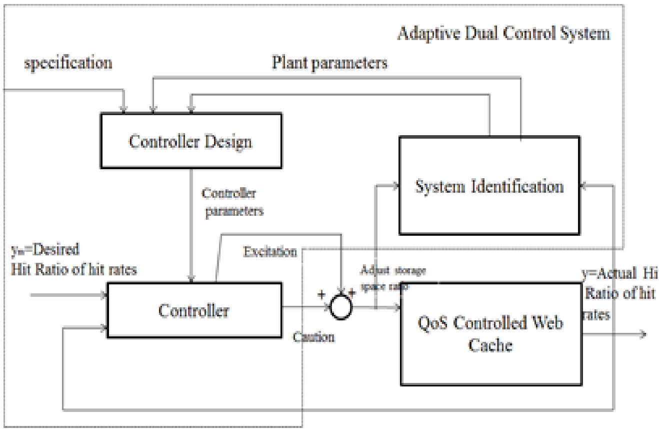

Figure 4. Adaptive QoS Web Cache dual Control

It incorporates uncertainty in the control strategy with the control signal. The system modeled is a discrete time-varying as in Equation 1. The Equation 2 below represents the hit rate at $k + 1$ interval.

$$y(k+1) = -a_1(k)\,y(k) + ..-a_n(k)\,y(k-n+1) + = -b_1(k)\,u(k) + ..-b_m(k)\,u(k-m+1) \quad (2)$$

where

$y(k)$ - actual ratio of hit rate
$u(k)$ - control signal for adjusting storage space ratio
k - discrete time index

$a_i(k)$ and $b_j(k)$ for $i = 1..n$ and $j = 1..m$ are unknown time-varying system parameters. The uncertainty is modeled using an additional stochastic parameter drift in the Equation shown below:

$$\mathbf{p}(k+1) = \mathbf{p}(k) + \varepsilon(k) \quad (3)$$

where

ε(k) - is the white noise drift vector.

p(k) is estimated using standard technique for online system identification.

In order to derive control law the following cost functions (4) and (5) have to be minimized. These equations will help in meeting the goals of dual adaptive controller to control the system output and to accelerate the estimation for future control improvement.

$$J^a{}_k = -\mathrm{E}\{\beta^2[y_n(k+1) - y(k+1)]^2 \mid \mathfrak{I}_k\} \qquad (4)$$

Where
$\mathfrak{I}_k$ = represents the set of inputs and outputs at time k
β^2 = coefficient to simplify algebraic manipulations
$y(k+1)$ = output

The equation (4) is used to minimize the deviation from system output $y(k+1)$from nominal output $y_\mathrm{n}(k+1)$

$$J^a{}_k = -\mathrm{E}\ \{\ [\ y(k+1)] + \sum_{i=1}^{m} c_i\, y(k - \mathrm{i} + 1) - \mathrm{p}^\mathrm{T}\mathrm{m}(k)]^2 \mid \mathfrak{I}_k\} \quad (5)$$

c_i = desired pole values
p = predictive error value

The equation (5) is used to accelerate the parameter estimation. We infer that the dual controller framework optimizes the tradeoff between the control goal and the uncertainty prediction.

The Figure 5 shows how adaptive control law is applied to improve the web cache hit ratio discussed in [33]. Two adaptive controls are designed: a deterministic control to deal with the parameter uncertainities and a stochastic design to compenstate system noises. The cache is classified a multiple classes of content in proxy cache. In their empirical study it is observed that the ratio of average hit ratio is closely related to the ratio of cache storage assigned to the classes. The experimental results prove stochastic adaptive controller performs better than a deterministic controller.

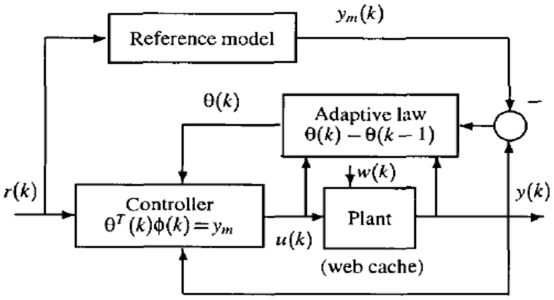

Figure 5. Direct adaptive control system for a Web Cache

B. Web Server Performance

In this section we discuss on improvement of the performance of Web Servers. In [34] a general control theoretic model is developed and validated on a general single server queue. In [35] the authors have modeled the system using linear MIMO of Apache Web Server to design feedback controllers, analyze pure pole placement and LQR [36] based techniques which doesn't have the imaginary part in the pole placement. This was one of the initial works on control theory on computing. The focus is on regulating CPU and Memory utilizations identifying tuning parameters for controller design as shown in Figure 6. The tuning parameters are "KeepAliveTimeOut" (KA) and "MaxClients" (MC). Also the interest is to modify these parameters dynamically. An empirical approach is used to model the server based on the statistical models of the data which is a linear time invariant model (ARX) in nature. The proposed PI controller in [35] aims at managing between speed of response and overreaction to noise through PI controller. In their pole placement technique it is observed that large control gains result in excessive control reaction to the stochastic of the system and large changes in KA and MC leading more oscillatory closed loop response. Their experiments proved that LQR controller has smaller gains leading to a less oscillatory behavior. The equation (6) represents the LQR low gain controller.

$$\mathrm{J} = \sum_{k=1}^{\infty} \quad [\ e_k\ v_k]^\mathrm{T}.Q. \begin{bmatrix} ek \\ vk \end{bmatrix} + u^T{}_k R.u_k \qquad (6)$$

As this is the initial works in a MIMO model, a limited Tuning or control parameters are considered. Usually the linear models are proper for good operating regions.

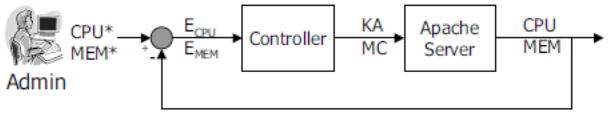

Figure 6. Feedback control of Apache Server for CPU and MKemory Utilizations

But the practical environments exhibit non-linear behavior. A non-linear control theory is discussed in [37] to design admission control mechanisms of a server system which is modeled as a GI/G/1-system. The related work in [38] uses PI controller in admission control of servers using linear control theory. Though PID controllers are used in ATM flow control discussed in [39] they are explained using linear deterministic models and in [40] it is discussed such models cannot be used in queuing systems. So a non-linear control is considered and a PI and RST controllers are used in [41] where the server requests are modeled as queuing system. The admission control has a gate, a controller and a monitor measuring average server utilization represented by ρ(kh). The control time is divided into 'k' intervals and length of each interval is 'h' seconds. The Figure 7shows the discussed control theoretic model and the system under study is shown in Figure 7. This figure is redrawn from the original picture.

The arriving requests are only admitted if there is an available token. New tokens are generated at the rate of $u(kh)$.

The server utilization during interval kh, ρ(kh) can be estimated as

$$\rho(kh) = \min(\ (u(kh) + x(kh)) / (\sigma_{max(kh)}),\ 1\ ) \qquad (7)$$

where

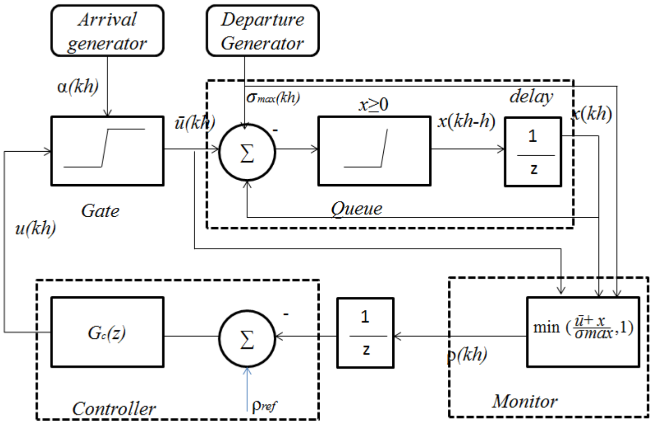

Figure 7. A control theoritic model of a GI/G/1-system with admission control

$u(kh)$ = desired admittance rate for interval kh

$x(kh)$ = represents the state of the queue

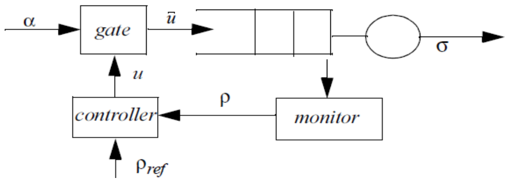

Figure 8. Investigated System

The PI-Controller is represented as shown in the equation (8)

$$u(kh) = K_e(kh) + \sum_{i=0}^{k-1} (K/T_i)\, e(ih)^T \qquad (8)$$

The gain K and T_i integral time are set to make the controlled system behave as desired and need to be determined with respect to stability and robust ness. This control is applied on D/D/1-system, M/M/1-system and M/H_2/1-system and the variations are calculated. Also, the controller was implemented in the discrete-event simulation program and the results are found to be similar to the Simulink model.

Another controller designed is the RST Controller represented as

$$R(q)\, u(kh) = T(q)\, \rho_{ref}(kh) - S(q)\, \rho(kh) \qquad (9)$$

$R(q)$, $T(q)$ and $S(q)$ are expressed in forward shift operator q. They are the controller polynomials and the results between using PI-Controller and RST-Controller; it is observed that the settling time is shorter for the latter. With PI-Controller the bursty stochastic processes are difficult to control and it becomes important to consider the non-linearity and stochastic of the systems to be controlled and model the systems in that manner. The objective is to keep the server utilization as close to reference value and the settling time should be short. The stability of the server node is analyzed when a PI-controller is applied based on linear queue model, and compared with the admissible control parameters derived from nonlinear analysis. In order to study stability Tsypkin criterion [42] or the Jury-Lee criterion [43], which are the discrete counter parts of the popov criterion, continuous systems are used.

SLA management of Web Servers is managed using "Queue Length Model Based Feedback Control" under dynamic traffic that provides delay regulation reducing residual errors, which is discussed in [44] as shown in Figure 9.

The queue length is used for controller design related to the delay of a request. The queue length is predicted at each control invocation and updates the request rate estimate. Based on the new queue length based feed forward predictor adjusts the estimated service rate based on request rate estimation and server queue built up. The controller calculates the service rate adjustment based on difference between delay reference and measured delay in each control interval.

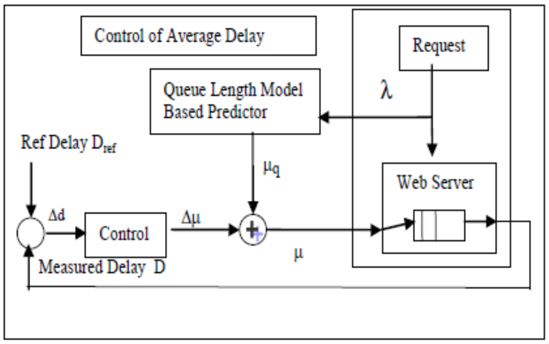

Figure 9. Queue Length Model Based Feedback Control Architecture

The classic Queuing based feedback control is observed to be ineffective with bursty traffic [45] and a Pareto On/Off distribution model [46] is used to model such web traffic. It is proved that a PI controller applying Queue Length Model based Feedback has better improvement. Queue length is the predicted unlike the load metrics compared to other theories.

We observe PI controller being commonly used to improve the web caching and Web server performance, also adaptive dual controllers designed to improve the performance in web server environments.

V. Web Services

In this section we discuss how the control system concepts are applied in Web Services environment.

A. Service Execution Engine QoS

We have seen the application dual feedback control loops in Web caching [31]. A similar approach can be seen on a Service Execution Engine to provide response time Guarantee even when the workload varies significantly, and increasing the throughput of the engine in WebJetFlow [47] as shown in Figure 10, which is redrawn from the original paper. Most of the workflow engines executing the composite service workflows never focused on the qualities of the composite services in running environment [48]. The solution proposed in WebJetFlow tries to address this concern by classifying the services and assign them to the process executor according to their class using a dual feedback control. The service invocations are assigned the threads according to the class. For each such class there is a queue maintained. Once again a PI – controller is used to control the input which is shown below

$$u_k(t) = u_k(t - 1) + g.(e(t) - r.e(t - 1)) \qquad (10)$$

where

$e(t)$ is the controller input,

u_k is the controller output,

g and r are design parameters.

The controlled process executor is modeled as a difference equation as

$$v(t) = a_j(t - 1) + b_j(t - j) \qquad (11)$$

The root-locus technique is used to place the closed-loop poles for a given response time control by setting g and r.

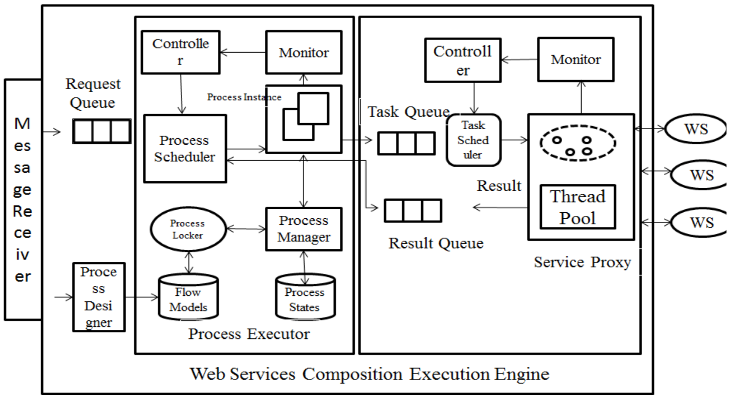

Figure 10. Web Services Composition Execution Engine Framework

B. Adaptive Control in Web Services

ControlWare is developed in [49] which is a middleware package to control the performance and QoS of the Web Servers along with the scheduling and queuing theory. It considers the classical problems of the server performance: rate, delay and ratio control problems.

The delay and ratio pose non-linearities in the system and feed forward control models to predict the system behavior when the inputs are changed. Another important improvement suggested is to retrofit the controller in a non-intrusive approach. The ControlWare provides a means of plug-and-play sensors, actuators, and controllers into performance loops.

The following Figure 11 shows the computing model and Figure 12 shows the control oriented representation of the ControlWare.

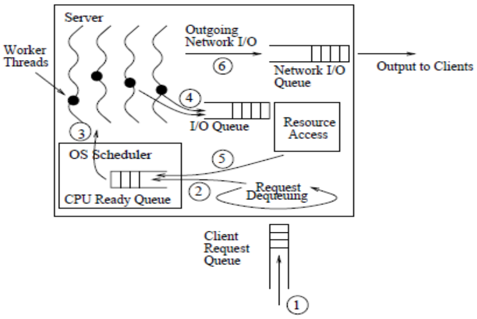

Figure 11. Computing model

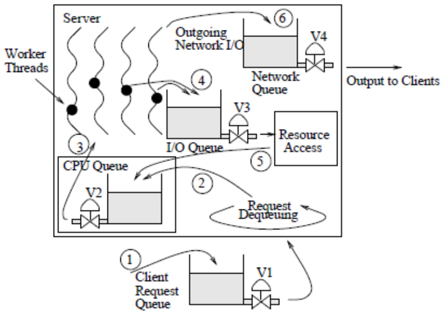

Figure 12. Control Oriented representation

The server is modeled using the ARMA difference equation [50]. The equation can be derived from a state space representation of the server model

$$\mathbf{x}(k) = A\,\mathbf{x}(k-1) + b\,u(k) \qquad (12)$$

$$y(k) = C\,\mathbf{x}(k) \qquad (13)$$

where

$\mathbf{x}(k)$ – state vector,

A,b and C represent system model

The prediction part of the feedback system on controlling the web server performance takes the non-linear behavior of the control loops into account.

VI. Application Servers

The Application Servers provide infrastructure for development and deployment of 3-tier applications. These servers are popular in hosting many mission critical applications and it is important that Application Servers exhibit high performance always. The Application Servers that we refer in this review are Java Enterprise Edition (JEE) [51] based implementations. The previous versions of JEE servers are also called as J2EE Servers. The following discuss the control theory application in different aspects of the Application Servers.

A. Data Flow Control in J2EE Servers

The flow control in J2EE servers is discussed in [52] using control theory and dynamic probabilistic scheduling. The web tier load is regulated to prevent the overload on the database tier. The classical control theory principle is supplemented with the workload classification and dynamic queue scheduling to maintain the fairness in the requests, handle larger variance in resource demands. The feedback control is used to regulate the load that enters at the web tier, based on the load at database tier. This is similar to congestion control in networking with variations like support continuous processing, throughput is lower. A simple Integral Controller is used and the control law for *k*th interval is

$$u(k) = u(k - 1) + K_i\, e(k) \qquad (14)$$

$$e(k) = y(k) – r(k)$$

$u(k)$= Number of allowed HTTP request per time period.

$y(k)$ = measured output

$r(k)$ = Maximum database connection per time (reference value)

$e(k)$ = Control error

The database usage is profiled offline and the requests are classified using k-means algorithm as large and small requests. The average usage of the database of each class is passed to the feedback control to estimate the usage for the next period. In order to support small requests Dynamic Probabilistic scheduling (DPS) is introduced by increasing their priority than large requests but providing a fair priority to the large requests. The Figure 13 shows the overall software architecture.

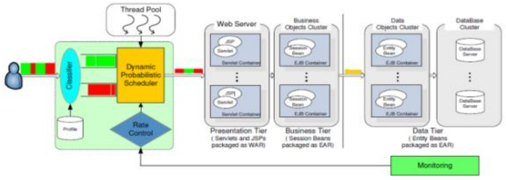

Figure 13. Admission Control Subsystem

The authors used simple Integral Controller to regulate the load on the database tier. But the database usage is classified offline. The controller works like a knob to control incoming requests at the web tier itself. They claimed to improve the accuracy, but this is a simple technique.

B. Repair Management in J2EE Servers

The automatic repair control of J2EE servers using feedback control is introduced in [53]. It is called as JADE in which an autonomous repair management system for J2EE clusters is constructed based on FRACTAL model. The feedback control is used by connecting the managed system and the repair management functions. There are multiple features in [53] but we explain only the Repair Management component control loop that uses the feedback control. The Manager component contains a policy component that decides on the node allocation to deploy the application components during repair management. Nodes are referred to as physical computer abstraction. The block diagram is shown in Figure 14.

The sensors monitor the managed system like state, resource usage, and failures. The actuators perform life cycle and configuration actions. The Transport component binds sensors and actuators. The Manager component implements the control loop for analysis and decision of the repair management when there is a failure either in the node or a component in the node.

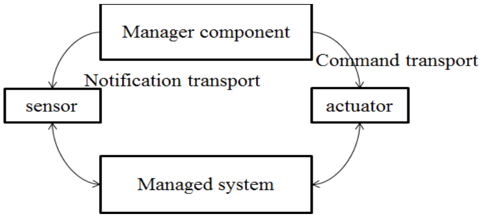

Figure 14. Management Control Loop

Though it is interesting to see the fault and repair management using feedback control, the policies considered are simple and the feedback controller types are not put to use and the modeling and analysis provided by the control theory is not applied completely.

C. Improving J2EE Server Performance

In this section we will discuss on how the adaptive control is put to use in the J2EE Servers. In [54] the authors implemented feedback control for adaptive self-configuration capability to improve the performance of the J2EE Servers. The data is collected by the Data Collector which is further processed which in turn is stored in the Database. The predictor will predict the future performance data to pro-actively determine the time-varying future loads using the database. The Comparator compares the difference between the predicted performance data and the SLA data. The Decision Maker will take an appropriate decision on the tuning strategy from the knowledge based which will be applied back to the J2EE Server, based on SLA, predicted values. An experiment is conducted on JBoss Server where it is observed that the EJB Container performance is depending upon the thread pool size (MaxPoolSize). The effect of MaxPoolSize response time is verified

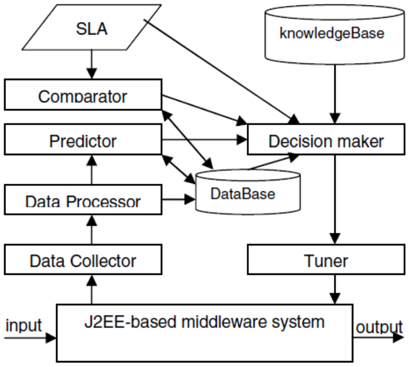

Figure 15. Adaptive performance configuration architecture

An architectural approach is proposed to handle variable workload in [54] called as Adaptive Performance Configuration architecture. But the improvement observed in [55] is to feed measures into the model and adaptively self-configure the system as shown in Figure 15. A knowledge based framework is proposed in [55] that has monitoring and knowledge based configuration mechanisms which is an improvement over [56][57] where linear or feed forward controllers are used. Though there are adaptive controllers proposed in [58] but are limited rule of thumb. The following Figure 16 shows how the adaptive performance is achieved using qualitative knowledge. The knowledge is represented in the form of fuzzy logic. This controller is based on fuzzy control as shown in the Figure below:

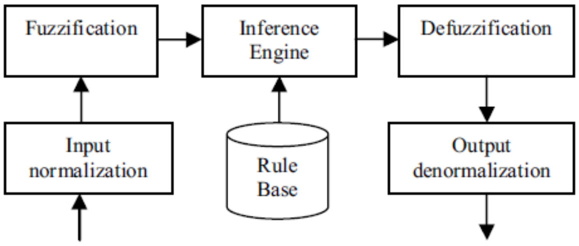

Figure 16. Structure of Fuzzy control

The fuzzy rules are defined for the “Max Pool Size” in JBoss Application Server. The authors of [55] claim that these rules can be proved by Lyapunov method that converge to the optimal MaxPoolSize at steady state. For ease of computation Triangular Membership functions are chosen and Centre of Gravity (COG) is used for defuzzification. The fuzzy control algorithm is used to compute the Next-Change-In-MaxPoolSize for given the values of Change-In-MaxPoolSize and Change-In-Response Time.

We observe that for adaptive mechanisms Fuzzy control is proved to be providing useful results over the conventional controllers.

An automatic tuning system proposed in [55] where controlling of QoS of modern E-Business site is studied. The performance data is monitored regularly and is fine-tuned by a controller if the QoS is not in the acceptable range. The server is tuned to the appropriate configuration to meet the expected QoS. An offline testing is done on the Application Server to identify the best configurations. JBoss Server is chosen to experiment the solution as a target system performing configuration tuning through a Controller, which is shown in Figure 17. This figure is redrawn from the original paper for clarity. There is a monitor component that collects the performance data.

The following are the configuration parameters chosen for performing the tuning Server Throughput, Server Side Response Time (RT). The Controller maintains the Agreed QoS and compares with the Actual QoS periodically. If the evaluation is above fixed threshold then the appropriate server configuration is applied.

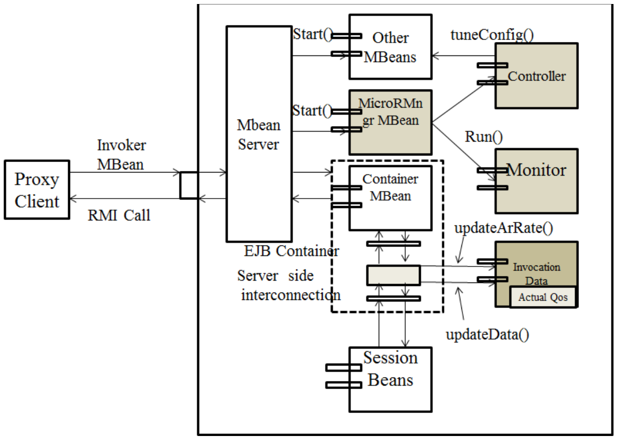

Figure 17. Automated Tuning System

The feedback control system though applied no conventional feedback controller is used. The optimum configurations is decided based on an offline test that is definitely an incomplete exercise. It may not be able to depict the real time load conditions.

Though there are different controllers available, PI controller is being heavily experimented in computing. But the results are showing a trend towards implementing the adaptive and intelligent controls that bring more advantages in improving the performance of Application and Web Servers.

VII. Enterprise Service Bus

Enterprise Service Bus (ESB) has become an important choice for Service Integration in huge enterprise applications. ESB implementations are run as standalone applications, but preferred to run as an integral part of the standard Application (JEE) Servers. A feedback control algorithm is discussed in [59] to realize load balancing routing and fail-over in ESBs that integrate IT services and Telecom services. A Hybrid Services Execution Environment (HSEE) is proposed which is a distributed architecture using ESB framework and routing that has to be available and manage load changes. There are two controllers, local and distributed which work together to maintain load balancing across the ESB nodes. The Figure 18 shows the load balancing using distributed feedback control. The detailed analysis and results are not discussed in our review.

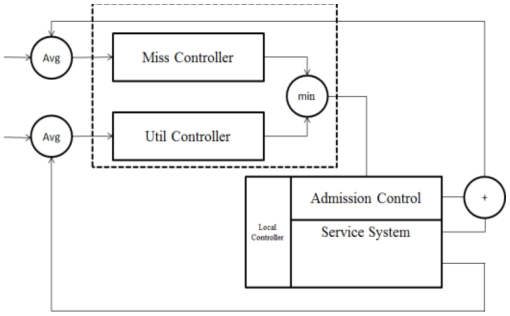

Figure 18. Load Balancing based on Feedback Control

VIII. Cloud Environments and Services

Cloud computing is the most discussed infrastructure platform for many of the service providers to host their application services due to cost based advantages. The service providers avoid investing and maintaining the infrastructure to host their services. In such scenario it is important for the service providers to ensure the QoS and SLA are maintained by them for their end customers. There are variety of cloud service delivery environment developed using JEE platform, besides other choices like OSGi. The control theory is investigated for maintaining the SLA of such cloud services in [60]. Besides the existing challenges and solutions discussed so far, there are additional important problems that need to be addressed in Cloud computing systems. In [61] the authors identified different challenges in providing automated control in cloud environments. The first challenge is to decouple the control into a cloud controller in the cloud infrastructure provider side and the application control to be moved out to the guest. These controllers have independent policies. This requires API from cloud hosting to support the controller's message exchange. The next one is the level of granularity of control. This depends upon how the resource providers choose to export the access to hypervisor level actuators. The previous implementations of the feedback controllers have fine grained access to the sensors and actuators on single virtualized nodes. For horizontally scalable clusters when the granularity is coarse relative to allocated resource, it is required to dampen the control loop at smaller sizes. The API that the Cloud resource provider exposes to the guests need to consider various constraints like how much the internal control to be exposed, how to integrate the guest control policy, how to design effective controller for 3-tier interactions, expose sensors that are suitable for stable control, expose suitable actuators for the controller policy to configure adapt request routing or other programmable network elements. In [61] Proportional Thresholding is proposed to meet these challenges. The solution is to modify P-controller by using a target range, which decreases as the accumulated actuator value increases. To eliminate steady state errors, Integral control is used as a policy defined by the equation (15).

$$u_{k+1} = u_k + K_i(v_{ref} - v_k) \qquad (15)$$

where

u_{k+1} - is the new actuator value

u_k - is the current actuator value,

K_i - is the Integral gain parameter.

v_{ref} *and* v_k are target and current sensor measurements respectively

This solution is experimented with Apache web server was front end, ORCA is used as underlying architecture, Tomcat cluster in the back end. If we observe carefully Tomcat is JEE container and it becomes important to consider the control systems for JEE servers hosting services in cloud environments. In [61] the infrastructure balancing using feedback control is discussed. The challenges when JEE based applications run on such an infrastructure providing cloud based services, the design of controllers for such problems are not discussed in detail. It will be important to investigate the applications of control theory on cloud services that are built using JEE environment.

In [60] the focus is again on the infrastructure layer and discusses briefly the various control theory applications but the objective is on resource allocation mainly. It is evident that large scale resource management problems could be effectively handled by adopting advanced control theory concepts like supervisory, cascaded, hybrid and optimal control.

An autonomic Service Delivery Platform (SDP) is proposed in [62] that host cloud services. We see an opportunity in applying the feedback control theory in building adaptive SDPs.

IX. Conclusions and Future Research

The feedback control theory has inherent advantages as discussed in [2] such as Quantitative I/O models, Dynamics and Transients, Correlation between multiple metrics, control algorithms, stability analysis, non-linear time-varying behavior. These kinds of support elements provide a methodical design and implementation choices of controllers suitable for the computing environments. Though control theory is being applied to various areas in Computing like Networks, Database systems, IT infrastructures, we observed that it is extensively investigated in Distributed Computing Systems starting at elementary components like web caching, Web Services to the Application Servers (e.g., JEE Servers), . We have also noticed that control theory is being investigated in the latest distributed systems like cloud computing and environment that poses a different set of challenges to the controllers like modeling the systems and providing high accuracy in prediction during feedback cycles. We have seen approaches of online modeling of the system to be adapted, applying different controllers starting PI controllers to hybrid controllers to certain extent. But it is essential to mimic the behavior of human in adapting the dynamic varying conditions in the network which can be achieved by combining the dynamic modeling of the systems with hybrid controllers and more innovative controllers, based on the computing system condition. More importantly it is observed that the JEE Servers are not just containers but also moving towards building Service Delivery platforms in SaaS environments, deployed as hosting components for service integration. Though there are very preliminary attempts to apply feedback control, it is essential to investigate it further and propose improved modeling and control designs. The controllers are built as plug and play components in computing environments, but not as obvious as any other plug-n-play architectures. Also the study of applying feedback controls in dynamic module component software like OSGi is not explored. We observe the JEE and OSGi are converging and it may be a good area to explore the feedback control applications in such containers hybrid containers.

Another important relevant work we observe is in modeling. The current modeling of the system for feedback is differential equations based, which brings difficulty in design and implementation of the controller software.

Spring containers have been more popularly in use by many of the distributed software applications at enterprise level. There is an attempt to apply the control theory in spring frameworks [63], but as the spring frameworks applicability is increasing it is obvious that they would face challenges similar to the ones faced by JEE and other Application Servers. It is important to explore the applicability of control theory in spring servers.

Based on this survey we observe that building hybrid intelligent controllers would make feedback controllers for computing, as regular building blocks. Though there are investigations in progress in designing hybrid controllers, most of them are limited to the web server environments [64][65], utilization of the resource management [66][67][68] or server admission control [69].

Data Mining [70] being a confluence of various Machine Intelligence based concepts. It has rich algorithms to handle and process the data efficiently for high accurate prediction capabilities, pattern recognition mechanisms. We are investigating such abilities and how effectively they can be used in designing intelligent hybrid controllers.Improved and hybrid feedback control systems on next generation distributed environments like cloud computing, services and associated technologies would aid in building more robust enterprise and cloud applications and services.

X. FUTURE WORK

We are investigating the applicability of control systems in different areas of the Distributed Systems with a specific emphasis on Java based Enterprise Servers. As a first step towards this we have designed a feedback controller to improve the statement caching mechanism in the JDBC drivers [71]. Our investigations are progressing in identifying optimal controllers for different components in distributed systems.

We are investigating the possibility of bringing the feedback control system modeling into the UML modeling and provide a generic API for integrating feedback controllers with computing systems. This enables the feedback control systems as first class elements of the UML.

Further, we are exploring control system application in Service Orchestration Engines and SDPs that are based on JEE using ESB and OSGi in their environments.Our goal is to design optimal hybrid controllers that are simple and well suited for the latest Java based environments.

REFERENCES

[1] Tarek Abdelzaher, Yixin Diao, Joseph L Hellerstein, Chenyang Lu, and Xiaoyun Zhu., “Introduction to Control Theory and Its Applications to Computing Systems”, International Conference on Measurement and Modeling of Computer Systems SIGMETRICS’08

[2] What Does Control Theory Bring to Systems Research? Xiaoyun Zhu, Mustafa Uysal, Zhikui, Wang , Sharad Singhal, Arif MerchantPradeep Padala, Kang Shin, ACM SIGOPS Operating Systems Review, Volume 43 Issue 1, January 2009

[3] Chris Toft et al, www.hpl.hp.com/techreports/2004/HPL-2004-49.pdf Self Managed Systems - A Control Theory Perspective HPL, 2004

[4] Joseph L. Hellerstein, Yixin Diao, Sujay Parekh, and Dawn Tilbury Feedback Control of Computing Systems, John Wiley 2004

[5] Web and Application Servers for services delivery, http://en.wikipedia.org/wiki/Service_delivery_platform

[6] ESB Service Integration http://en.wikipedia.org/wiki/Enterprise_service_bus

[7] Yixin Diao, Joseph L. Hellerstein, and Sujay Parekh “Control of Large Scale Computing Systems”, ACM SIGBED Review, Vol3, Issue 6, 2006

[8] I.J.Nagrath, M.Gopal,”Control Systems Engineering,”, New Age International Publishers, 2007

[9] Zoran Gajic Lecture notes on Nyquist criterion, “www.ece.rutgers.edu/~gajic/psfiles/nyquist.pdf”

[10] Kunio Takaya,Lecture Notes on Routh Hurwitz criterion, “http://www.engr.usask.ca/classes/EE/481/takaya_notes/ee481-p8-routh_hurwitz.pdf”, 2009

[11] Xiaorui Wang et al, “FC-ORB: A Robust Distributed Real-time Embedded Middleware with End-to-End Utilization Control”, ACM Journal of Systems and Software, Vol 80, Issue 7, 2007

[12] Yin Wang, Terence Kelly, Stéphane Lafortune, “Discrete Control for Safe Execution of IT Automation Workflows” , ACM European Conference on Computer Systems, 2007

[13] Seungwan Ryu, Chulhyoe Cho,”PI-PD-controller for robust and adaptive queue management for supporting TCP congestion control”, 132 - 139 18-22 April 2004

[14] Ohsaki, H. Murata, M. Ushio, T. Miyahara, H, “A control theoretical approach to a window-based flow control mechanism with explicit congestion notification”, Pages 2715-2720, vol.3 IEEE Conference on Decision and Control, 1999

[15] Kyoung-Don Kang and Jisu Oh, Sang H. Son,” Chronos: Feedback Control of a Real Database System Performance”, Real Time Systems Symposium, Pg No.267-276, IEEE 2007

[16] C. Lu, X.Wang, and X. Koutsoukos “Feedback utilization control in distributed real-time systems with end-to-end tasks” IEEE Trans. Parallel Distrib. Syst., 16(6):550–561, June 2005

[17] Baochun Li, Klara Nahrstedt,” Impact of Control Theory on QoS Adaptation in Distributed Middleware Systems “,American Control conference, 2987-2991, vol 4, 2001

[18] R. Zhang, C. Lu, T. F. Abdelzaher, and J. A. Stankovic. “ControlWare: A Middleware Architecture for Feedback Control of Software Performance” International Conference on Distributed Computing Systems (ICDCS), IEEE Vienna, Austria, July 2002.

[19] S. Abdelwahed, N. Kandasamy and S. Neema, "A Control-Based Framework for Self-Managing Distributed Computing Systems",Workshop on Self-Managed Systems (WOSS'04), Newport Beach,CA USA, 2004.

[20] S. Mascolo, “Classical Control Theory for Congestion Avoidance in High-Speed Internet”, 38th IEEE Conference on Decision and Control, pp. 2709-2714, 1999

[21] Lui Sha, Xue Liu, “Queueing Model Based Network Server Performance Control “, Real Time Systems Symposium, 2002

[22] Jose Yepez, Pau Marti, Josep M.Fuertes, “Control Loop Performance Analysis over Networked Control Systems“. IECON02

[23] Phillip M. Dickens ,Vinod Kannan, “Application-Level Congestion Control Mechanisms for Large Scale Data Transfers Across Computational Grids“, The International Conference on High Performance Distributed Computing and Applications 2004

[24] Brandic, I., “Towards Self-Manageable Cloud Services”, Computer Software and Applications Conference, pages 128-133, 2009. COMPSAC '09.

[25] Zoran Vukic, Ognjen Kuljaca, “Lecture Notes on PID Controllers, http://arri.uta.edu/acs/jyotirmay/EE4343/Labs_Projects/pidcontrollers.pdf,

[26] Control Strategies: http://en.wikipedia.org/wiki/Control_theory

[27] Sheldon M. Ross, “Introduction to Probability Models”, Academic Press, 2009

[28] Model Predictive Control (MPCs): http://en.wikipedia.org/wiki/Model_predictive_control

[29] Gianfranco Balbo ,”Introduction to Stochastic Petri Nets” : http://www.mendeley.com/research/stochastic-petri-nets-an-

introduction-to-the-theory/, Lectures on Formal Methods and Performance Analysis 2001

[30] Lecture Notes, “http://www.me.berkeley.edu/ME237/2_general_properties.pdf”

[31] Ying Lu, Avneesh Saxena and Tarek E Abdelzaher Differentiated Caching Services; A Control-Theoretical Approach, IEEE International Conference on Distributed Sysytems, 2001

[32] Keqiang Wu, David J. Lilja, Haowei Bai “The Applicability of Adaptive Control Theory to QoS Design: Limitations and Solutions”, IEEE Parallel and Distributed Processing Symposium, 2005

[33] Ying Lu, Tarek Abdelzaher and Gang Tao, “Direct Adaptive Control of A Web Cache System”, Proceedings of the American Control Conference, Denver, Colorado, 2003

[34] A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson “Design and evaluation of load control in web server systems”, IEEE American Control Conference, 2004.

[35] N. Gandhi and D. M. Tilbury, Y. Diao, J. Hellerstein, and S. Parekh “MIMO Control of an Apache Web Server, Modeling and Controller Design”, IEEE American Control Conference, 2002

[36] LQR, http://en.wikipedia.org/wiki/Linear-quadratic_regulator

[37] Xue Liu, Jin Heo, Lui Sha, “Modeling 3-Tiered Web Applications”, Modeling, Analysis, and Simulation of Computer and Telecommunication Systems,. 13th IEEE International Symposium, 307 – 310, 2005

[38] C. Lu, T.F. Abdelzaher, J.A. Stankovic and S.H. Son, “A feedback control approach for guaranteeing relative delays in web servers” Proc. of the 7th IEEE Real-Time Technology and Applications Symposium, pp 51-62, 2001

[39] A. Kolarov and G. Ramamurthy, “A control-theoretic approach to the design of an explicit rate controller for ABR service”, IEEE/ACM Transactions on Networking, Vol. 7, No. 5, , pp 741-753, Oct. 1999

[40] S. Stidham Jr., “Optimal control of admission to a queueing system”, IEEE Transactions on Automatic Control, Vol.30, No.8, pp 705-713, Aug 1985

[41] M. Kihl, A. Robertsson, and B. Wittenmark, “Performance modelling and control of server systems using non-linear control theory.” Berlin, Germany: 18th International Teletraffic Congress, Sept. 2003.

[42] Michael Larsen, Petar V. Kokotović, “A brief look at the Tsypkin criterion: from analysis to design”, International Journal of Adaptive Control and Signal Processing, Vol 15, Issue 2, Pages 121-128, Mar 2001

[43] Jury-Lee criterion: http://en.wikipedia.org/wiki/Jury_stability_criterion

[44] Xue Liu, Rong Zheng, Jin Heo, Qixin Wang, Lui Sha, “Timing Performance Control in Web Server Systems”, ACM ICAS-ICNS’05

[45] L. Sha, X. Liu, Y. Lu, T. Abdelzaher, “Queueing Model Based Network Server Performance Control”, IEEE Real-Time Systems Symposium, Phoenix, Texas, December, 2002

[46] Pareto distribution: http://en.wikipedia.org/wiki/Pareto_distribution

[47] Chunming Gao, Weian Chen, Huowang Chen, “A Feedback Control Framework of Service Composition Execution for Response Time Guarantee”, Chunming Gao, Weian Chen, Huowang Chen, IEEE ICWS, 2007

[48] Ivona Brandic et. al.. “QoS Support for Time-Critical Grid Workflow Applications”, e-science, pages 108-115, First International Conference on e-Science and GridComputing (e-Science'05), 2005.

[49] Tarek Abdelzaher. Yina Lu, Ronahua Zhana, Dan Henriksson, “Practical Application of Control Theory to Web Services”, American Control Conference, 2004

[50] ARMA: http://en.wikipedia.org/wiki/Autoregressive_moving_average_model

[51] JEE: http://www.oracle.com/technetwork/java/javaee/tech/index.html

[52] Wei Xu, Zhangxi Tan, Armando Fox, David Patterson, “Regulating Workload in J2EE Application Servers”, http://www.controlofsystems.org/febid2006/files/16225_Wei.pdf

[53] Sara Bouchenak, Fabienne Boyer, Daniel Hagimont, Sacha Krakowiak et al., “Architecture-Based Autonomous Repair Management: An Application to J2EE Clusters”, ICAC’05

[54] Yan Zhang, Wei Qu, Anna Liu, “Adaptive Self-Configuration Architecture for J2EE-based Middleware”, Vol 9, HICSS’06

[55] Giovanna Ferrari, Santosh Shrivastava,Paul Ezhilchelvan, “An Approach to Adaptive Performance Tuning of Application Servers”, IEEE International Workshop on QoS in Application Servers, 2004

[56] N. Gandhi, J. L. Hellerstein, S. Parekh, and D. M. Tilbury, “Managing the Performance of Lotus Notes: A control TheoreticApproach”, in Proceedings of 27th International ComputerMeasurement Group Conference, 2001

[57] L. W. Russell, S. P. Morgan, and E. G. Chron, “Clockwork A new movement in autonomic systems”, IBM SYSTEMSJOURNAL, VOL 42, NO 1, pp77-84, 2003

[58] M. Raghavachari, D. Reimer, and R. D. Johnson, “The Deployer’s problem: Configuring Application Servers for Performance and Reliability”, In Proceedings of the 25th International Conference on Software Engineering (ICSE’ 03), 2003

[59] Yang Zhang, “Dependable ESB Routing in Hybrid Service Execution Environment”, AISS : Advances in Information Sciences and Service Sciences, Vol. 2, No. 1, pp. 83 ~ 93, 2010

[60] Harold C. Lim, Shivnath Babu, Jeffrey S. Chase, Sujay S. Parekh, “Automated Control in Cloud Computing: Challenges and Opportunities Challenges and Opportunities”, ACDC’09

[61] Christos A. Yfoulis and Anastasios Gounaris, “Honoring SLAs on cloud computing services: a control perspective”, Proceeding of the 2nd workshop on Bio-inspired algorithms for distributed systems,Pages:29-38, 2010

[62] Robert D. Callaway, Michael Devetsikiotis, Yannis Viniotis, Adolfo Rodriguez, “An Autonomic Service Delivery Platform for Service-Oriented Network Environments”, vol. 3 no. 2, pp. 104-115, April-June 2010

[63] Dr. Wolfgang Winter , “Applying control theory concepts in software applications”, http://www.theserverside.com/feature/Applying-control-theory-concepts-in-software-applications

[64] Yaya wei, Chuang Lin, Thiemo Voigt, Fengyuan Ren, "Fuzzy Control for Guaranteeing Absolute Delays in Web Servers, QoS in Wireless Networks" International Conference

[65] Jianbin Wei ,Cheng-Zhong Xu, “eQoS: Provisioning of Client-Perceived End-to-End QoS Guarantees in Web Servers”, IEEE Trans on Computers, 2006

[66] Mehmet H. Suzer ,Kyoung-Don Kang, “Adaptive Fuzzy Control for Utilization Management”, ISORC , Pages 383-390, 2008

[67] Palden Lama , Xiaobo Zhou , “Autonomic Provisioning with Self-Adaptive Neural Fuzzy Control for End-to-end Delay Guarantee”, IEEE International Symposium on Modeling, Analysis and Simulation of Computer Telecommunication Systems, 2010

[68] P. Lama , X. Zhou, “Efficient server provisioning for end-to-end delay guarantee on multi-tier clusters. In Proc. IEEE Int’l Workshop on Quality of Service (IWQoS), 2009

[69] Jiang Ying, , Meng Dan, “Enforcing Admission Control, Using Admission-Time-Ratio and PI Controller,” Journal of Computer Research and Developmen, 2007

[70] Jiawei Han, Micheline Kamber, , “Data Mining – Concepts and Techniques”, Morgan Kaufmann Publishers, 2006

[71] Ravi Kumar Gullapalli, Dr.Chelliah Muthusamy, Dr.A.Vinaya Babu, Raj.N.Marndi “A Feedback Control Solution in improving Database Driver Caching”, IJEST,Vol 3, No 7, Jul 2011

AUTHORS PROFILE

Ravi Kumar Gullapalli is working as aTechnical Expert in Hewlett-Packard., Bangalore, India. He obtained his M.Tech in Computer Science from Birla Institute of Technology, Mesra,India. He is currently pursuing Ph.D from JNTU Hyderabad,AP, India.

Dr.Chelliah Muthusamy is Academic Relations Head at Yahoo, Bangalore,. He obtained his Ph.D from Georgia Tech and M.Sc(Engg) in Computer Science from Indian Institute of Science(IISc), Bangalore India

Dr.A.Vinaya Babu is a Professor of Computer Science working as Director, Admissions at JNTU, Hyderabad, AP, India. He obtained his Ph.D and M.Tech in Computer Science from JNTU, Hyderabad.