Towards Self-Optimization in Utility Computing using Fuzzy Logic Controller

Zhixiong Chen, Yi-Hua Zhu, and Michael Y. Yuan, Member, IEEE

Abstract—The main focus in the IT utility computing is services and service levels. In order to make successful services and to meet service level objectives while at the same time to stay in a competitive edge, service providers have to make intelligent utility allocation in a distributed on-demand virtual resources paradigm. In this paper we propose a fuzzy resource allocation approach in which a Fuzzy Logic Controller is used to dynamically adjust the number of resources involved in serving requests. Simulation shows that the controller is capable of capturing the complexity of utility systems such as nonlinear relations, uncertainty and dynamic nature. It performs extremely well in such an environment with unpredictable service request numbers, abrupt peak loads, and changeable service types.

Index Terms—fuzzy logic, fuzzy control, self-optimization, utility allocation, service computing

I. INTRODUCTION

For the last several years, we have seen the trend of decomposition of closely coupled complex software platforms and applications into small pieces of serviceable components, the trend of service location transparent, the trend of virtualization, the trend of open standards, and the trend of management de-centralization. Consequently, these trends lead to IT service utility computing [1] in which service providers like traditional utility companies have to study service system self-configuration and self-optimization, service capacity discovery and provisioning, service high availability with reasonable servicing time, complex service composition, adaptation and innovation, service level agreement, service maintenance schedule, etc. In this utility computing paradigm, services and service level objectives rather than individual hardware or software component are being monitored, configured, and optimized [2]. In order to meet service level objectives while to keep cost down, service providers need to manage resources effectively and efficiently.

In this paper we propose a fuzzy resource allocation approach where a fuzzy logic controller is used to dynamically adjust the number of resources needed in serving requests. The reason to use fuzzy logic in controlling resource allocation is due to the complexity, uncertainty and dynamic nature of heavily distributed on-demand virtual resources. Within such a complicated system, many new factors such as public network quality and availability, security, privacy, cost incentive, etc., will affect the final outcome of quality of service. Moreover, these factors usually are full of uncertainty, change dynamically and have nonlinear relations so that traditional control theory is unable to apply to. One direction to deal with this complexity is to use magic probabilistic inference like Bayesian learning [3], Bayesian network [4], Markov Decision [5], etc. An alternative direction proposed in this paper is to use fuzzy logical control, where system characters are modeled as linguistic variables with corresponding membership functions and appropriate controls are produced by a fuzzy inference engine based on intuitive or expert knowledge, usually in the form of IF-THEN rules. Thus, nonlinear system with great complexity and uncertainty can be effectively controlled based on fuzzy rules without dealing with complicated mathematical models [6, 7].

Our simulation shows utility systems with fuzzy logical control performs extremely well in an environment where service request numbers are unpredictable, peak loads are orders of magnitude larger than that in normal steady state, and service types are varied in terms of resource requirements.

The paper is organized as follows. In the next section, we describe our Fuzzy Logical Control system. In section III, we discuss applicable situations to which such Fuzzy Logical Control system could apply. Our focuses are mainly on the web server farm [8,9] and utility resource management in grid computing environment [10, 11]. The section followed is about our simulation settings and results. The last is devoted to discussion of further work.

II. THE FUZZY LOGIC CONTROL SYSTEM

Typically, a fuzzy logic control system (fuzzy controller) is a rule-based architecture whose main functional components involve a fuzzier (linguistic encoder), fuzzy rules, fuzzy inference engine, and defuzzifier (referred to as a decoder) [6]. Their relations can be depicted in Figure 1.
Our fuzzy controller has two input variables, \( N \) and \( U \), and one output variable \( \Delta S \). \( N \) is the number of requests in an observation window, \( U \), called usage ratio, the ratio of the occupied utility to the current available capacity, and \( \Delta S \) the change of utility capacity (See Figure 2).

Whenever each window expires, the fuzzy controller is invoked to generate a suitable change of the utility capacity according to the current values of \( N \) and \( U \). Both \( N \) and \( U \) are quantified through a series of fuzzy sets as Low (L), Medium (M), and High (H). As shown in Figure 3, their membership functions \( f(N) \) and \( f(U) \) are respectively chosen as trapezoidal and triangular ones, where \( N_L, N_M, N_H \) are critical points of \( f(N) \).

Besides, the fuzzy sets of the output variable \( \Delta S \) are defined as No Change (NC), Decrease Slightly (DS), Decrease Moderately (DM), Increase Slightly (IS), and Increase Moderately (IM). Their membership functions are selected as triangular ones (see Figure 4).

It should be stressed that this type of membership functions is easy for computation and is commonly used in practical applications of fuzzy control.

Intuitively, the utility capacity should be reduced when the usage ratio \( U \) goes low, and vice versa. The fuzzy rules included in Table 1 reflect this intuitive description of the control policy.

To derive the activation levels of the output fuzzy sets. The fuzzy inference engine adopts the Max-Min fuzzy inference method. All these weights are used to derive \( \Delta S \) during defuzzification.

### III. APPLICABLE SCENARIOS

We discuss mainly two application scenarios to which the fuzzy logical control could apply, that is, the IBM Océano project [8, 9] and web service resource allocation management in grid computing [10, 11]. In addition, there are many other similar management systems that the fuzzy control system can be used, for example, Sun’s Grid Utility Computing [12], HP’s SmartFrog project [13] and Duke University’s COD [14].

The IBM Océano project was intent to design and develop a pilot prototype of a scaleable, manageable infrastructure for a large scale computing utility power plant that enables multi-customer co-location hosting on a virtualized collection of hardware resources.

The observation is that incoming web requests are not always stable and predictable. There are spikes in that peak loads are orders of magnitude larger than that in a normal steady state.

In the original co-location hosting model similar to the dedicated hosting model, each customer has a dedicated infrastructure and some exclusive servers that are not shared with each other. To support such peak-load scale would require large investments in standby, non-shared resources that are in most of the time underutilized.

In the new co-location hosting service model, service...
providers are able to adjust resources such as bandwidth, server numbers and storage assigned to each customer quickly based on the dynamically fluctuating workload. Isolation between subscribers was achieved using physically separate servers and virtual local area networks in a reconfigurable switched-network infrastructure.

Another application scenario is in the area of grid computing [10, 11]. A utility system can be described by three layers: utility type definition, resource management logic and virtual resources. Resource management logic manages resources allocation such as disk logical partitions, CPU utilizations, or more complicated multi-tier services. It has functionalities like abstraction planner, resource provisioning, runtime monitoring and reconfiguration invocation if needed.

Because of the complexity, uncertainty and dynamic changes of these systems, Fuzzy Logical Control can be used as an engine to automate the adjustment of resources.

IV. EXPERIMENTAL STUDIES

We demonstrate in our simulation how the fuzzy logic control system can be used. We assume that a utility system has $S$ number of virtual resources and each resource has $p$ parallel service capacity. Basically, service requests can be classified as short informational service and transactional service. All service utility requirements (for instance, time duration) follow a Gamma distribution with the probability density function $f(t) = \frac{(\gamma \mu)^\gamma}{\Gamma(\gamma)} t^{\gamma-1} e^{-\gamma \mu t}$, whose mean is $1/\mu$ and the variance is $1/(\gamma \mu^2)$. $\mu$ is the average service utility requirement. We use $\mu_1$ and $\mu_2$ to represent informational and transactional requests, respectively. $\gamma$ is a parameter to be tuned to reflect different variations. The Gamma distribution is selected because it has a desirable property to fit an arbitrary distribution by setting appropriate parameters. We denote $u_o$ as the normal incoming service request rate while $u_s$ as peak load incoming rate. The ratio between these two should be in an integer scale. The number of incoming requests is uniformly distributed around its incoming rate, that is $[u_o - b_o, u_o + b_o]$ and $[u_s - b_s, u_s + b_s]$. The overall percentage of peak load is $\beta$.

A service will be dropped if its waiting time for service is over $D$. Therefore the mean response time $T_R$, average delay time per request $T_D$ and drops percentage over total served requests $L$ all depend on $\{S, p, t, \bar{\mu}, \gamma, u, b, D, \beta\}$, where $\rightarrow$ is used for a vector.

The fuzzy logical control observation period is $T$. Hence, we have fuzzy relation $\Delta S \leftarrow \{U, N, T\}$.

Our cost function is average percentage over virtual resources used. The goal is to have the overall cost down while to keep every metric under service objectives.

Here is one of our test sets:

$$\{S, p, \mu_1, \mu_2, \gamma, u_o, u_s, b_o, b_s, D, \beta\}$$

and

$$\{\text{IM, IS, DS, DM, NC}\} = \{3, 1, -1, -3, 0\}.$$

Figure 5 represents the incoming requests versus the time. The number is per 50 time unit that smoothes out the randomness in actual incoming requests per unit time. Figure 6 is the corresponding resource adjustment number.

In the simulation, we also compare the fuzzy control system with static number of resources: 100, 80 and 60. The number 100 is the maximum number of resources. Other resource numbers are measured using this scale. The average resource number from fuzzy control system is around 57. So we pick up number 60 and then 80 in the middle to analyze the results. The overall cost, the mean response time, the delay per request, and the drop rate are respectively given in Figure 7, 8, 9, and 10, where bar 1 is for the Fuzzy Logic control case, bar 2 for static 100 resource case, bar 3 for 80 and bar 4 for 60.

Although the average resource numbers, which is equivalent to the average cost because in our simulation we assume the unit cost for each resource is equal to 1, is around 57, the static resource number 60 case caused a drop rate of 0.05. In other words, every 1000 requests, approximately 50 of them will be lost. The delay per request is almost ten folds of the Fuzzy Logical case. For the case of 80 and 100, the overall results are almost the same, which strongly suggests that most resources are under-utilized. However, it
should be pointed out that our proposed fuzzy scheme makes
resources fully used.

We have made various tests under different situations. Basically, the conclusion is the same. That is, utility systems with fuzzy logical control perform well in terms of the resource cost while keep up service objectives.

V. DISCUSSION AND FURTHER WORK

In this paper, we discuss the possibility of applying Fuzzy Logical Control system to utility system. Our simulation shows that the fuzzy logical control engine is capable of capturing the complexity of utility systems such as nonlinear relations, uncertainty and dynamic nature and is able to perform well in situations with unpredictable service request numbers, abrupt peak loads, and changeable service types. We would like to test it from real computing environment. More complex design and experiments are needed.

In the future, we would like to apply the engine to more complicated utility situations like composite service requests where utility allocation is a continuous spatial process. And so is the fuzzy logical control. We also like to develop an algorithm to deal with service discovery and service section from an available service pool in which services are ranked due to historical performances.

REFERENCES
[8] The Océano project:
[12] Sun Grid Compute Utility,