An Overload Protection Algorithm for the Webserver to Guarantee High Availability and Performance in Internet Environments

*이상문, 민병조, 김학배
*LPA, Inc., 연세대학교 전기전자공학과
moon@LPAtech.com bjmin@yonsei.ac.kr hbkim@yonsei.ac.kr

An Overload Protection Algorithm for the Webserver to Guarantee High Availability and Performance in Internet Environments

*Sangmoon Lee, Byungjo Min and Hagbae Kim,
*LPA, Inc., Yonsei University, Department of Electrical and Electronic Engineering

Abstract

We propose a traffic control algorithm to reduce the webserver workload. The overload of the webserver decreases by using the suggested feedback control. We also build the webserver model using Timed Petri Net (TPN), and the performance of the proposed algorithm is effectively evaluated.

I. Introduction

Generally, the web server in which client requests are concentrated is not sufficiently robust for unexpected traffic increments. Thus, certain systematical approaches for the workload control have been studied. In [1], the system built the general MIMO model. However, the system performance undesirably depends on S/W parameters. In [2], the performance of a webserver was described utilizing the classical feedback control theory. All contents should be pre-processed and store multiple copies differing in both of quality and size. In [3], the bandwidth feedback algorithm and the fair bandwidth allocation were presented. However, the approach should know about traffic situations at the destination. In this paper, we propose an overload protection algorithm that controls the webserver workload by adopting the feedback control theory. We focus on the relationship between the input and the output traffics of the webserver. The performance of the proposed algorithm is evaluated by building the TPN model which is tuned by experimental results.

II. System Architecture

We present the system architecture of the webserver for applying feedback control, depicted in Fig. 1. The architecture generically consists of three basic functional modules, a server process, a monitor, and a controller. We assume that the system is a single webserver with sufficient internal resources. The server process performs elementary webserver functions. The monitor and the controller are additional functional modules for feedback control. The monitor senses both of the request rate R and the throughput T, which are intrinsic feedback parameters. The controller uses PI (Proportional-Integral) control to compute the control input. The controller maintains the server workload under the desired level. To reduce the workload, the number of receiving requests are limited. If the workload increases over the predefined value, the requests are automatically rejected. Therefore, the controller effectively reduces the throughput.

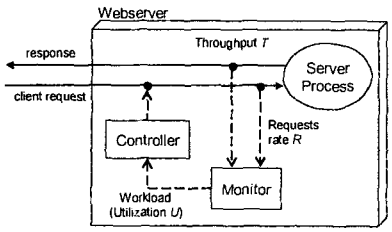

Fig 1. The system architecture of the webserver

III. Timed Petri Nets Model

To simulate the proposed algorithm, we design a TPN model [4] as shown in Fig. 2.

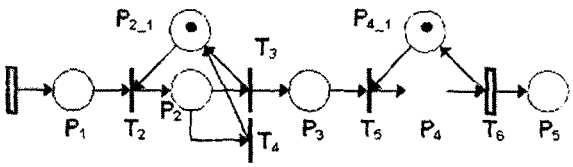

Fig 2. The TPN model of the webserver

The model is composed of eight places and six transitions, as presented in Table 1. To evaluate the performance of the server, we experiment with a benchmark program, WebBench 4.1 [5]. The experiment is performed by requesting