
실시간 웹 서비스를 위한 피드백 부하 제어 기법

정석용*

동양미래대학 전산정보학부

Feedback Load Control Mechanism for Real-Time Web Services

Suk-Yong Jung*

School of Computing and Information, Dongyang Mirae University

요약 약 본 논문에서는 실시간 웹 서비스 시스템에서 과부하를 조절하는 기법을 제안한다. 기존의 많은 기법은 단순히 웹 연결 수를 조절하여 과부하를 방지한다. 이는 웹서버의 요청 처리를 위한 추가적인 웹 태스크의 생성을 조절하는 것만 가능하고, 이미 생성된 웹 태스크나 내부 디바이스의 최신 상태를 알려주기 위해서 주기적 동작하는 내부의 실시간 태스크 등에 대해서는 과부하 방지를 위한 아무런 조치를 취하고 있지 않다. 본 논문에서는 주기적 동작하는 내부의 실시간 태스크 등에 대해서도 동적으로 부하를 반영할 수 있는 제어를 설계한다. 설계된 제어를 적용한 피드백제어 시스템을 구성하여 안정된 부하 조절이 가능함을 확인하였다.

Abstract This paper proposes a mechanism for managing overload in real-time web service system. The many previous mechanisms manage overload with controlling web request. These mechanisms can control only new web request. They don't have any control existing tasks, especially periodic tasks. We design a controller that able to meet real-time performance with controlling even periodic tasks. A feedback control system is implemented applying the proposed mechanism. And we verified the stable operation of system.

• **Key Words** : Feedback Control System, Real-time Web Service

1. 서론

인터넷을 활용한 응용분야가 넓어지면서 웹 기술은 웹 서버를 탑재하고 내부의 태스크뿐만 아니라 웹을 통한 서비스 요청에 대해서도 실시간 스케줄링을 제공하는 실시간 내장형 시스템에도 응용되고 있다. 웹 카메라나 센서 데이터 퓨전(sensor data fusion), 타겟 트래킹(target tracking) 등과 같은 내장형 시스템에 웹 기술을 적용한 인터넷 디바이스들은 웹 브라우저를 통해 디바이스에 대한 여러 상태를 감시하고 제어하는 기능까지도

수행한다.

이와 같이 실시간 내장형 시스템은 우선 웹 서버를 내장하고 있어야 하고, 운영자가 센서나 카메라와 같은 내부 디바이스의 상태 변화를 즉각 알 수 있도록 주기적으로 최신의 디바이스 내부 정보를 추출하여 운영자에게 보내주는 시스템 내부의 실시간 태스크와 운영자가 웹을 통해 요청하는 디바이스 제어 명령을 처리하기 위한 웹 태스크 등을 두어 시스템 내부의 실시간 태스크뿐만 아니라 웹으로 요청되는 실시간 웹 서비스 요청에 대해서

본 논문은 동양미래대학 2009 산업체현장연구과제로 수행되었음.

*교신저자 : 정석용(syjung@dongyang.ac.kr)

접수일 2010년 12월 08일 수정일 2010년 12월 25일 게재확정일 2010년 12월 28일

도 실시간 스케줄이 제공되어야 한다. 그러나 웹서버를 내장한 실시간 시스템은 인터넷의 개방성으로 인해 웹 요청의 소용량을 예측할 수 없다. 따라서 웹서버로 서비스 요청이 폭주하는 경우 웹 서버는 서비스 처리를 위해 많은 태스크를 생성하여 자원을 소비하게 되기 때문에 시스템의 실시간 속성을 만족시키지 못하는 결과가 초래된다.

지금까지 피드백 구조를 활용한 실시간 데이터베이스 서비스 제공에 대한 연구와 실시간 웹 서비스를 제공하기 위해서 실시간 운영체계에 웹 서버를 탑재하고, 웹 서버를 통한 서비스 요청에 대해서는 웹 서버가 별도의 실시간 스케줄을 제공하는 연구가 이루어졌다[1, 2, 6]. 그러나 시스템 내부의 실시간 태스크에 대해서는 커널 수준의 실시간 스케줄이 제공되고, 웹 태스크에 대해서는 웹 서버 수준의 실시간 스케줄이 제공되는 이원화된 스케줄링 구조로 인해 우선순위 역전현상이 발생하는 등의 지연현상으로 실시간 속성을 충족시켜주지 못하고 있다 [1].

한편 이원화된 스케줄링 구조를 통합하여 실시간 속성을 충족시키는 통합 스케줄링 구조가 제안[2, 3]되었으나 웹서버로 요청이 폭주하는 경우는 커널로 접수되는 연결 요청을 처리하는데 많은 자원이 소비되고, 웹 요청을 처리하기 위해 많은 태스크들이 생성되기 때문에 초래되는 시스템의 과부하로 인해 이 역시 실시간 속성을 충족시켜주지 못하고 있다[1].

웹서버로의 서비스 요청이 폭주하는 경우에도 과부하를 방지하기 위해 시스템의 부하 상태에 따라 피드백 제어를 통해 웹 연결 수를 동적으로 제어하는 시스템에 대한 연구도 이루어졌다. 과부하 상태가 예측되는 경우는 시스템의 목표 성능을 유지하기 위하여 연결 요청단계에서 초기에 연결 요청 패킷(SYN 패킷)을 폐기함으로써 웹 태스크의 추가 생성을 막아 과부하를 방지하고 있으며, 제어를 통해 시스템의 부하 상태에 따라 SYN 패킷의 접수 비율을 조절하고 있다[4].

그러나 웹 연결 수를 조절하여 과부하를 방지하는 방법으로는 웹서버의 요청 처리를 위한 추가적인 웹 태스크의 생성을 조절하는 것만 가능하고, 이미 생성된 웹 태스크나 내부 디바이스의 최신 상태 알려주기 위해 주기적 동작하는 내부의 실시간 태스크 등에 대해서는 과부하 방지를 위한 아무런 조치를 취하고 있지 않다. 따라서 시스템의 부하가 아주 높은 상태에서는 긴급한 웹 요청에 대해서는 부하 제어기에 의해 접수가 거부되면서도

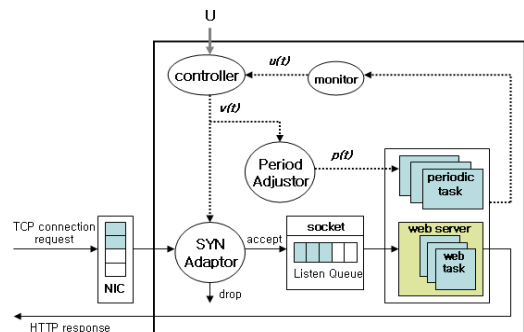
주기적으로 디바이스의 상태 등을 알려주는 태스크에 대해서는 실행을 보장하게 되어 웹 태스크와 내부 태스크 간에 불균형 상태가 초래되는 문제가 발생한다.

이에 본 연구에서는 웹 태스크의 경우는 신규 웹 연결의 수를 조절하고, 기존 시스템 내부의 주기적 태스크의 경우는 주기를 조절함으로써 과부하를 사전에 방지하면서 웹 태스크와 내부 태스크 간에 균형적인 실행을 보장할 수 있는 시스템을 제안한다. 이를 위해 웹 태스크의 경우는 실시간 시스템의 부하를 측정하고 그에 따라 SYN 패킷의 접수 비율을 동적으로 조절하고, 내부 주기 태스크의 경우도 시스템 부하 정도에 따라 실행 주기를 탄력적으로 조절할 수 있도록 한다.

2. 통합 피드백 부하 제어 시스템 구조

본 연구에서는 피드백 제어 시스템의 제어 목표를 CPU 효율(CPU Utilization)로 둔다. 샘플링 창 W 에 대한 샘플링 주기 $((k-1)W, kW)$ 동안 측정된다. 그런데 CPU 효율의 경우 여러 가정된 부하 상황에서 태스크로 인해 유발되는 전체 효율의 합이 일정 임계치를 넘지 않을 경우 모든 실시간 태스크가 마감 시간을 모두 만족한다는 것을 보여주고 있다. 본 연구에서는 CPU 효율을 제어 대상으로 두고 제어함으로써 시스템의 실시간 성능을 보장 받을 수 있도록 한다.

따라서 매 주기 관측된 CPU 효율과 목표 CPU 효율을 바탕으로 웹 서버로 투입되는 웹 요청 접수율과 주기 태스크의 주기를 결정한 후, 시스템의 부하 조절을 위한 피드백 제어 시스템의 구조는 [그림 1]과 같다.



[그림 1] 피드백 제어 시스템 구조

피드백 제어 루프는 관측기(monitor), 제어기

(controller), SYN 패킷 접수기(SYN Adaptor), 주기 보정기(Period Adjustor)로 구성한다. 우선 관측기는 주기별로 시스템의 CPU 효율을 측정하여 제어기로 알려준다. 제어기는 관측기로부터 샘플링 주기별로 통보되는 CPU 효율과 초기 설계된 CPU 효율 목표값(U)과의 차이 $E(k) = U - u(t)$ 을 계산하고, 이를 기초로 SYN 패킷 접수율과 주기 태스크의 실행 주기를 결정할 새로운 조작 변수 값 $v(t)$ 을 결정한다.

SYN 패킷 접수기는 제어기가 결정한 조작변수 값 $v(t)$ 에 따라 웹 요청을 선별적으로 허용한다. SYN 패킷 접수율에 따라 SYN 패킷 접수기는 일부 SYN 패킷을 폐기한다. SYN 패킷의 폐기는 SYN+ACK 패킷을 기다리는 클라이언트의 타임아웃을 유발하고, 서비스 요청을 지연시키는 효과가 있어 CPU 효율이 조절된다. 주기 보정도 제어기가 결정한 조작변수 값 $v(t)$ 에 따라 측정 CPU 효율이 목표치를 넘는 과부하 상황의 경우는 주기 태스크의 주기를 늘리고, CPU 효율이 목표치를 밑도는 경우는 주기를 줄인다. 주기의 변경은 태스크의 실행 빈도를 변경함으로써 CPU 효율이 조절되는 효과가 있다. 예를 들어 목표 CPU 효율이 80%이고 현재 샘플링 주기에서 측정된 CPU 효율 값이 87%인 경우 오차 $E(k)$ 는 7%가 발생하고, 제어 시스템은 과부하 상황을 줄이기 위한 방향으로 작동한다.

3. 통합 피드백 부하 제어 시스템 설계

제어기는 관측기로부터 샘플링 주기별로 통보되는 CPU 효율과 초기 설계된 CPU 효율 목표값(U)과의 차이 $E(k) = U - u(t)$ 를 기초로 SYN 패킷 접수율과 주기 태스크의 실행 주기를 결정할 효율 조절율과 이에 따라 시스템에서 관측된 시스템 효율과의 관계를 단일입력 단일출력(SISO: Single Input, Single Output) 시스템으로 모델링한다.

실시간 웹 서버 시스템의 CPU 효율이 과거의 패킷접수 비율에 의존하는 경우 입력 $x(t)$ 와 $y(t)$ 의 차분 방정식은 식 (1) 과 같이 표현된다.

$$y(t) = \sum_{i=0}^n a_i y(t-i) + \sum_{j=0}^m b_j x(t-j) \quad (1)$$

용이한 분석을 위해 식 (1)을 시공간에서 주파수 공간

으로 변환하는 z 변환 ($Y(z) \sum_{t=0}^{\infty} y(t)z^{-t}$)을 적용하면 식 (2)와 같다.

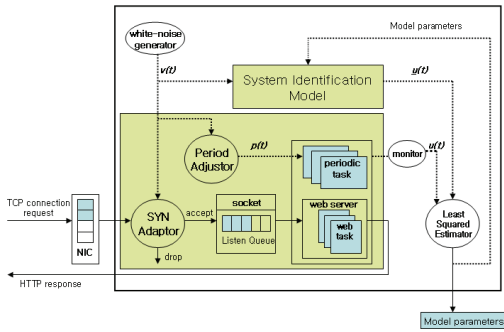
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{j=0}^m b_j z^{m-j}}{z^n - \left(\sum_{i=1}^n a_i z^{n-i}\right)} \quad (2)$$

식 (2)로부터 시스템 z 평면에서의 전달 함수 $G(z)$ 을 구한다. 시각 t 에서 CPU 효율을 $u(t)$ 라고 하고, 효율 조절율을 $v(t)$ 라고 할 때 2차 시스템의 전달함수 $G(z)$ 는 식 (3)과 같다.

$$G(z) = \frac{U(z)}{V(z)} = \frac{b_0 z + b_1}{z^2 - a_1 z - a_2} \quad (3)$$

따라서 식 (3)에서 파라미터들(a_i, b_j)은 최소제곱 추정법을 사용하여 추정한다. 최소제곱 추정법을 사용하여 파라미터를 추정하기 위한 구조는 [그림 2]와 같다. 웹서버 모델링 시스템과 실제 웹서버 시스템 모의 실험기에 대해 white-noise 생성기는 무작위로 패킷 접수율을 결정하여 통보한다. 웹서버 시스템의 모델 파라미터는 최초 임의의 값으로 설정한다. white-noise 생성기로부터 패킷 접수율을 토대로 시스템 모델을 예측한 시스템 효율과 실제 웹서버 시스템의 모의 실험기를 통한 시스템 효율을 비교하여 최소제곱 추정법에 따라 모델링 시스템의 모델 파라미터를 재설정하고, 모델링 시스템으로부터 예측 효율과 웹서버 시스템 모의실험기로부터 실제 시스템 효율의 차이가 수렴할 때 까지 이 과정을 반복한다.

실험을 통해 재귀 최소제곱 추정기법을 적용하여 시스템 모델의 파라미터 a_i, b_j 의 값을 추정하였다. 모델 파라미터 추정을 위해 모의실험에 사용할 웹 태스크의 도착율은 평균 λ 를 갖는 포아송 분포를 따른다고 가정한다. 시스템내 상존하는 평균 태스크의 수가 k 인 경우에 시스템의 평균효율이 1(100%)이 되도록 태스크의 실행 시간을 $1/k$ 로 정하였다. 따라서 시스템의 평균 부하는 $\rho = \lambda/k$ 이다. 각 태스크 집합은 30분 동안 도착한 태스크로 구성된다. white noise 발생기에 의해 무작위로 발생하는 패킷 접수율 값에 대해 모의실험을 통해 1초 단위의 주기별로 시스템 효율을 측정하였다.



[그림 2] 모델 파라미터 추정 구조

식 (3)에 대하여 기본 제어를 설계하기 위해 루트 로커스 제어 이론[5]을 적용하였다. z-도메인에서 PI 제어기의 전달함수는 식 (4)으로 표현할 수 있고, 폐 루프 서버(closed loop server)의 전달함수로 변환하면 식 (5)로 표현된다.

$$D(z) = \frac{g(z-r)}{z-1} \quad (4)$$

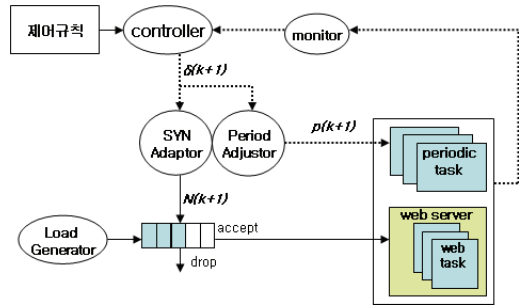
$$G_c(z) = \frac{D(z)G(z)}{1+D(z)G(z)} \quad (5)$$

제어 이론에 따르면 시스템의 성능은 전달함수 극점에 영향을 받는다. 제어함수 g와 제어 제로 r 값을 성능 규격을 만족시킬 수 있도록 MATLAB의 루트 로커스 도구를 사용하여 조정하였다. 분석결과 제어기 파라미터 g, r이 각각 0.1676, 0.8로 결정되었고 극점은 $p_0 = -0.5427$, $p_1 = 0.9900 \pm 0.0078i$ 에 배치되었다.

4. 성능

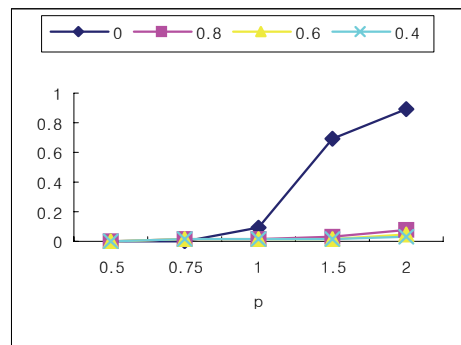
여러 부하상황에서 웹서버 시스템이 목표효율 U를 달성할 수 있도록 제어를 설계하여 피드백 제어 성능을 실험한다. 모의실험을 위하여 (그림 3)의 구조를 갖는 실시간 웹서버 시스템 모의 실험기를 제작하였다. 모의 실험기는 부하발생기(Load Generator), 실행기(Executor), 관측기(Monitor), 제어기(Controllor), SYN 패킷접수기(SYN Adaptor), 주기조정기(Period Adjustor)의 6개의 컴포넌트로 구성되었다.

모의실험에서는 웹 태스크의 도착율이 평균 λ 의 포아송 분포를 따른다고 가정하였으며, 시스템내 상존하는 평균 태스크의 수가 k인 경우 시스템의 평균효율이 1이 되도록 평균 태스크의 실행 시간을 $1/k$ 로 정하였다. 따라서 시스템의 평균부하는 $\rho = \lambda/k$ 이다. 제어기는 PI 제어기를 적용하여 제어를 구현하였다. PI 제어기 전달함수의 파라미터 g, r은 근계법을 사용하여 결정한 0.16764, 0.8을 사용하였다.



[그림 3] 모의실험기의 구조

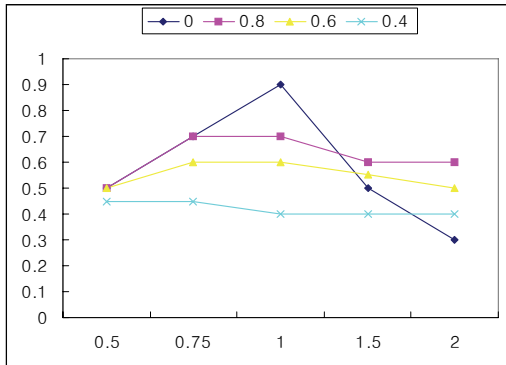
과부하로 발생하는 실시간 특성을 실험하기 위해 [그림 4]와 같이 서로 다른 부하 환경 ρ (0.5, 0.75, 1, 1.5, 2)에서 다양한 목표효율 U (0.8, 0.6, 0.4)에 대하여 제어를 수행하는 경우와 제어를 수행하지 않은 경우에 대해 마감시간 초과 비율과 처리량을 각각 측정하였다.



[그림 4] Deadline Miss Ratio

측정결과 제어를 수행하지 않은 경우 마감시간 초과 비율이 부하가 증가함에 따라 비례하여 증가함을 알 수 있다. 그러나 제어를 수행하는 경우는 마감시간 초과 현상이 거의 일어나지 않음을 확인할 수 있다. 이것은 과부

하상황에서 웹요청을 초기에 적절히 제어함으로써 시스템의 실시간 속성을 제어하고 있음을 알 수 있다.



[그림 5] Throughput

또한 [그림 5]와 같이 실행이 완료된 태스크의 비율을 측정한 처리량을 살펴보면, 제어를 수행하지 않은 경우 웹요청 증가에 따라 처리량이 증가하지만 과부하 상황에서는 처리량이 급격히 감소하는 것을 알 수 있다. 그러나 제어를 수행하는 경우는 과부하 상황에서도 처리량의 급격한 감소없이 유지되는 것을 확인하였다.

5. 결론

웹 연결 수를 조절하여 과부하를 방지하는 방법으로는 웹서버의 요청 처리를 위한 추가적인 웹 태스크의 생성을 조절하는 것만 가능하고, 이미 생성된 웹 태스크나 내부 디바이스의 최신 상태 알려주기 위해 주기적 동작하는 내부의 실시간 태스크 등에 대해서는 과부하 방지를 위한 아무런 조치를 취하고 있지 않다. 본 연구에서는 웹 태스크의 경우는 신규 웹 연결의 수를 조절하고, 기존 시스템 내부의 주기적 태스크의 경우는 주기를 조절함으로써 과부하를 사전에 방지하면서 웹 태스크와 내부 태스크 간에 균형적인 실행을 보장할 수 있는 시스템을 제안하였다. 제안된 기법은 실시간 웹서버 시스템을 시스템 식별기법으로 모델링하고, 근계적법으로 피드백 제어를 설계하였다. 실험 결과 제어를 통해 시스템 과부하 상황에서도 마감시간 초과현상이나 처리량의 급격한 감소없이 실시간 속성을 제어하고 있음을 확인하였다.

참고 문헌

- [1] S. Jung, B. Kang, K. Choi and K. Chung, "Integrated Scheduling for Reducing the Delays by Priority Inversion in Real-Time Web Service," IEICE transactions on communications, Vol.E86-B, No.7, pp.2143-2153, July 2003.
- [2] J.M. Almeida, M. Dabu, A. Manikutty, and P. Cao, "Providing differentiated quality-of-service in web hosting services," 1998 SIGMETRICS Workshop on Internet Server Performance, Madison, Wisconsin, June 1998.
- [3] R. Iyer, V. Tewari, and K. Kant, "Overload Control Mechanisms for Web Servers," Proc. Workshop on Performance and QoS of Next Generation Networks, Nagoya, Japan, pp.225-244, November, 2000.
- [4] 강봉직, 정석용, 김영일, 최경희, 정기현, 유혜영, "피드백 제어 이론을 이용한 실시간 웹서버 시스템의 부하제어 기법," 정보처리학회논문지A 제 10-A권 제6호, 2003년 12월
- [5] N.S. Nise, Control System Engineering, John Wiley & Song, Inc, 2000.
- [6] K. D. Kang, J. Oh, and S. H. Son, "Chronos: Feedback Control of a Real Database System Performance", In Proceedings of the 28th IEEE Real-Time Systems Symposium, Tucson, Arizona, USA, December, 2007.

저자 소개

정 석 용(Suk-Yong Jung)

[중신회원]



· 2010년 1월 현재 : 동양미래대학교수

<관심분야> : 정보통신, 실시간시스템