# 클러스터 기반 웹서버 시스템에서의 사용자 감지 서비스 질 보장을 위한 입장 제어

임상석*, 이창규, 임승호, 박규호
한국과학기술원 전자전산학과
{*sslim@core, shlim@core, kpark@ee}.kaist.ac.kr, clee75@intizen.com

# User-perceived QoS Control in a Cluster-based Web Server System

Sang-Seok Lim, Chang-Kyu Lee, Seung-Ho Lim, Kyu-Ho Park
Dept. of EE, EECS KAIST

**요 약**

본 논문은 급속한 인터넷의 성장에 따라 사용자의 서비스 질을 보장하기 위한 Q-PID라는 제어기를 새롭게 제안하다. 이러한 제어기는 사용자 요청에 대한 응답시간을 보장하기 위하여 모든 웹 요청에 대해서 입장 허용 가능한지를 조사하고 이에 근거하여 입장 허용된 요청은 서버 지연 응답시간을 보장해 줄 수 있도록 한다. 이러한 Q-PID 제어기를 사용함으로서 복잡한 사용자의 요구를 충족시킬 수 있다. 이러한 Q-PID를 설계하고 구현하였으며 실험결과와 함께 제어성에 대해서 토의하였다.

## 1. 서 론

인터넷 서버는 최종사용자에게 다양한 서비스를 제공한다. 이런 서버의 사용 예로서 정적이거나 동적인 문서의 검색, 정보 검색을 위한 on-line 데이터베이스 검색, E-commerce, 그리고 검색 엔진 등이 있다. 이러한 서버들은 하나의 워크스테이션이나 LAN을 통해서 서로 연결된 워크스테이션의 클러스터 형태가 존재한다.[1] 인터넷 사용자의 폭발적인 증가로 인해 이렇게 구성된 인터넷 서버시스템은 두 가지 중대한 도전에 직면하게 되었다. 첫 번째로 이러한 서버는 사용자의 요청을 처리할 수 있는 충분한 자원을 확보해야 한다. 자원이란 CPU, 디스크, 네트워크 그리고 물리적인 메모리를 의미한다. 두 번째로 서비스의 질적인 보장을 통해 사용자의 요구사항을 만족시켜 주어야 한다. 사용자의 요구사항의 예로서는 사용자 요청에 대한 예측 가능한 응답시간이라든가 E-commerce의 트랜잭션 처리의 안전성, 검색엔진 등에 의해서 생성된 결과의 정확성을 보장 등이 있다. 마이크로프로세서, 메모리, 입출력 시스템에서의 괄목할 만한 성능향상과 이를 이용한 상용 워크스테이션의 클러스터의 배치를 통해서 첫 번째 도전에 대해, 싸고 확장성이 뛰어난 방식으로 대처할 수 있게 되었다. 사용자 요구사항이 날로 복잡해짐에 따라 두 번째 도전이 지속적으로 중요해지고 있다. 예를 들자면 사용자는 요청을 철회하기 전에 예측 가능한 응답시간을 제공받기를 원하고 이러한 것은 사용자 입장에서의 서비스의 질이라고 정의 할 수 있다.

클러스터기반 웹서버 시스템 상에서 동작하는 서비스 질 제어기를 설계하고 구현하였다. 기본적으로 이러한 제어기는 기본적으로 존재하는 서버의 구조에 매우 의존적이다. 그러므로 제어기가 동작하는 서버의 구조에 대해서 좀더 자세하게 설명하겠다

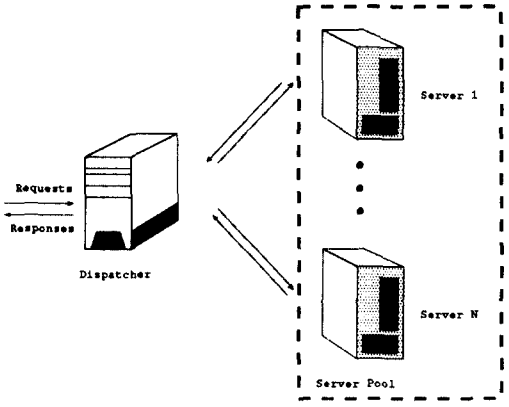


그림 1 클러스터 기반 웹 서버 시스템의 구조

그림1에서 보여진 것과 같이 웹 요청 분배기(dispatcher)는 네트워크 입력단에 위치하여 입력 들어오는 연결에 대해서 서버 쪽 프록시로서 동작한다. 웹 요청 분배기는 클라이언트와 접속을 맺고 클라이언트로부터 요청을 받게 된다. 이 요청은 파싱(parsing)되고 실제 서비스를 해줄 서버를 서버 풀(server pool)에서 선

택한다. 마지막으로 서버로부터 되돌아온 응답이 웹 요청 분배기를 통하여 클라이언트로 보내진다. 이러한 구조에서 성능향상을 위해서 TCP handoff[3]나 TCP splicing[4] 기술을 채택할 수 있으나 본 논문에서는 이를 고려하지 않았다. 결론적으로 모든 서버 pool로부터의 응답은 웹 서버 요청 분배기를 거쳐서 클라이언트로 전송되어지게 된다.

위와 같은 서버구조상에서, 본 논문에서는 두 번째의 도전에 초점을 맞추었다 : 사용자 인식 서비스 질 (User-perceived QoS). 사용자 인식 서비스질 을 아래와 같이 정의하였다.

**정의 1:** ***사용자 인식 서비스질은 인터넷 지연과 서버지연의 합으로 구성된다. 인터넷 지연은 요청이나 응답이 인터넷을 통해 전달되는 시간이고 서버 지연이란 서버에서의 응답 처리시간과 웹 요청 분배기와 서버 사이의 요청 및 응답 송수신 지연시간이다.***

인터넷 지연을 제어하는 것은 라우팅 경로에 위치하는 라우터의 성능과 배치에 의해서 결정하게 되므로 서버 입장에서 제어가 불가능하다. 그러므로 본 논문에서는 서버 지연을 예측하고 제어하는데 초점을 맞추기로 한다. 이러한 서버 지연은 두 가지 중요한 요소가 있다. 첫 번째 요소로는 CPU 와 입출력 시간에 의한 서버 처리 시간과 두 번째 요소로는 웹 요청 분배기와 서비스 서버 사이의 요청 및 응답 전달 시간이다. 실제 본 논문에서 구현한 시스템에서 서버 지연은 아래와 같은 수식으로 정의하여 측정하였다

서버지연 = 끝시간 - 시작시간

시작시간은 웹 요청 분배기가 클라이언트로부터 요청을 받은 시간이고 끝시간은 웹 요청 분배기가 서버로부터의 응답을 클라이언트로 보내는 시간이다. 사용자는 서버 처리시간과 요청-응답 전달 시간을 구분할 수 없으므로 서버지연을 위와 같은 방식으로 표본화하여 측정하였다. 비록 인터넷 지연은 제어를 할 수 없지만 이러한 서버 지연은 추정되고 관리되어질 수가 있다. 이러한 제어를 가능하게 하기 위해 본 논문에서는 입장 제어 기술을 기반으로 하는 Q-PID 제어기를 제안하고 구현하였다. 자세한 구현 사항은 다음 장에 계속 설명되어지겠다.

## 2. 서비스 질 제어기 설계 및 구현 : Q-PID 제어

그림 2는 Q-PID 제어기의 전체 구조 및 개념적인 제어법을 보여주고 있다

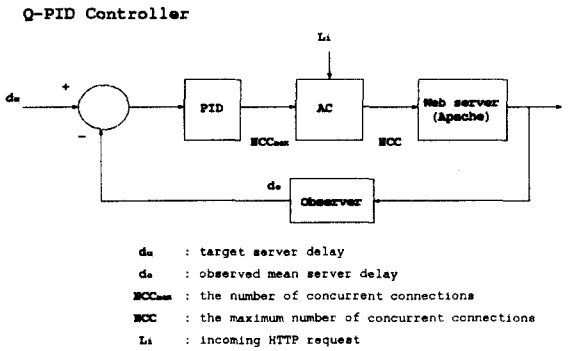


그림 2 Q-PID 제어기의 구조

Q-PID 제어기는 PID 제어기, 입장(adimission)제어기, 웹서버 지연 측정기로 구성되어있다. 웹 서버 지연 측정기는 평균 서버지연 시간을 매 시간 간격마다 측정한다. PID 제어기는 이렇게 측정된 서버지연을 입력으로 하여 제어에 이용한다. PID 제어기는 이 제어기와 웹서버간의 최고 동시 접속자수를 출력으로서 조절하게된다. 입장 제어기는 새로이 입력되어지는 HTTP 요청에 대해서 현재의 최고 동시접속자수를 기반으로 입장 허용 할 것인지 불허할 것인지를 결정하게 된다. 이러한 방식으로 Q-PID는 사용자에게 특정시간 내에 서비스를 제공할 수 있도록 전체 시스템을 통제하게 된다.

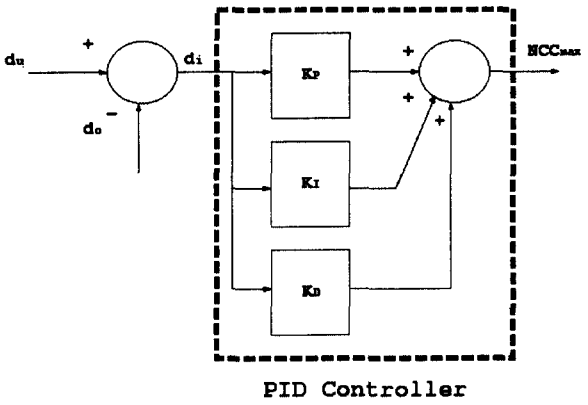


그림 3 PID 제어기

실제 PID 제어기를 상세하게 설명하겠다. 그림 3에서 제어기의 입력으로는 목표 서버 지연(du)과 관찰된 서버 지연시간(do)의 차이(di)가 되겠다. 이 제어기는 위의 차이를 보상해주기 위해서 변화하는 di값을 미분, 적분, 비례와 같은 연산의 조합을 이용하게된다. 비례상수(Kp)는 비례적인 제어를 하게 된다. 이러한 제어는 출력을 계산

하여 단순히 입력에 대해서 Kp에 비례하게 반영하는 제어 방식이다. 미세한 조정을 위해서는 이러한 비례 제어 방식 외에 입력에 대한 미분과 적분에 의한 방식을 적용해야 한다.[2] 그러므로 우리는 적분 상수(Ki)와 미분상수 (Kd)를 제어기에 포함시켰다. 이러한 제어기의 전달함수는 아래와 같다.

이다.

미분과 적분 연산을 구현하기 위하여 매 시간 마다 타이머 함수를 두어 위의 식의 값을 계산하도록 하였다.

는 PID 제어기에 의해서 결정되고 AC 제어기의 입력으로 사용된다. du를 달성하고 유지하거나 di를 0에 가깝게 유지하기 위해서 그 제어기는 를 넘어서는 과도하게 입력되어지는 HTTP를 버리게 된다. 그러한 요청을 한 클라이언트는 대신해서 서버 바쁨(server busy) 페이지를 전송 받게 된다.

### 3. 실험 결과

본 장에서는 Q-PID 제어기에 대한 실험결과 보여주겠다. 실험을 위해 사용된 클러스터는 450MHz CPU에 64 MB의 메모리를 갖추고 Linux를 사용하였다. 4대의 PC가 사용되었고 1대는 웹 서버 요청분배기, 나머지 3대는 서버 pool에 배치되었다. 웹 요청은 Apache를 이용해서 생성되었다. 그림 1,2,3,4는 각각 웹 요청 분배기로의 동시 접속수를 증가시켰을 때의 제어기의 제어성을 보여주기 위한 그래프이다.

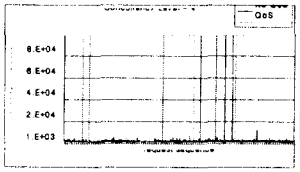

그림 4 동시성 4

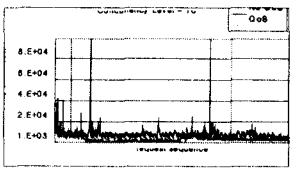

그림 5 동시성 16

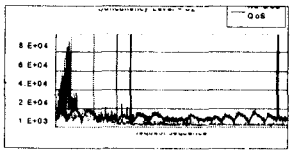

그림 6 동시성 32

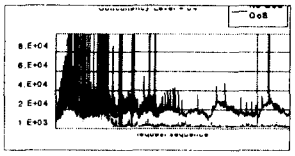

그림 7 동시성 64

그림을 통해 알 수 있듯이 du가 500msec일 때의 제어기의 제어성능을 보면 서버의 부하가 없을 때는 QoS,와 No QoS가 거의 비슷한 양상을 보여주지만 동시성이 32-64가 되어 부하가 급격히 늘어남에 따라 QoS는 거의 500msec내의 응답시간을 유지하지만 No QoS의 경우에는 거의 1000msec에 이르는 것을 알 수가 있었다.

이러한 제어기를 통하여 웹서버의 응답시간을 제어가능함을 보여주는 실험이었다.

### 4. 결론

본 논문에서는 Q-PID라는 사용자에 대한 서비스 질을 보장하기 위한 제어기를 제안하고 구현하였다. Q-PID는 기본적으로 서버지연시간을 보장하기 위하여 새로이 입력되어지는 HTTP 요청을 받아들이거나 버리게 된다. 이를 통하여 입장 허용된 요청들은 서버지연시간이 목표치에 도달하도록 제어하였다.

실험결과를 통하여 살펴봤듯이 서버 지연시간을 미세하게 조정하는 PID 제어기를 통하여 제어할 수 있었다. 이러한 제어기를 기반으로 기하급수적으로 늘어나는 웹 요청에 대한 사용자 감지 서비스 질을 보장할 수 있는 기본 방식을 제안하였다.

이러한 제어기는 현재에는 정적인 페이지에 대해서 동작이 확인되고 실험되었다. 그러므로 동적인 문서에 대해서도 계속적으로 확장하는 작업을 진행중에 있다.

### 5. 참고문헌

[1] Trevor Schoeder, Steve Goddard, and Byrav Rammaurthy, " Scalable Web Server Clustering Technologies", IEEE Network May/June 2000.

[2] "Automatic Control Systems" Prentice Hall

[3] Mohit Aron, Darren Sanders, Peter Druchel and Willy Zwaenepoel, " Scalable Content-aware Request Distribution in Cluster-based Network Servers", in Proceeding of the 2000 Annual Usenix Technical conference, June, 2000

[4] David Maltz and Pravin Bhagwat," TCP Splicing for Application Layer Proxy Performance", IBM Research Report, RC 21139 , March, 1998