서버 클러스터에서의 인터넷 서비스를 위한 효율적인 연결 스케줄링 기법

최재웅 김성천
서강대학교 컴퓨터학과
{choijw, ksc}@arqlab1.sogang.ac.kr

Round Robin(RR) ONE-IP: Efficient Connection Scheduling Technique for Hosting Internet Services on a Cluster of Servers

Jae-Woong Choi Sung-Chun Kim
Dept. of Computer Science, Sogang University

요 약

웹을 사용하는 사용자들의 급속도로 증가하는 서비스 요청을 신속하고 저렴한 비용으로 처리하기 위한 대응책으로, LAN 환경의 웹 서버 클러스터 구조가 각광을 받고 있다. 높은 가용성 및 확장성을 보장하는 웹 서비스를 제공하기 위해 많은 부하의 서비스 요구를 여러 서버에게 효과적으로 나누어 처리할 수 있어야 하며, 따라서 서비스 요청 패킷을 고르게 분배할 수 있는 합리적인 스케줄링 기법을 필요로 한다. ONE-IP 스케줄링 기법은 이더넷의 브로드캐스트 메시지에 의해 스케줄링이 분산되도록 하는 전략을 사용함으로써, 클러스터에 유입되는 패킷의 집중화로 인해 발생할 수 있는 병목 현상(bottleneck)과 치명적인 오류(Single-point of Failure) 문제를 효과적으로 해결하였다. 그러나, 서비스를 요청하는 패킷의 발신지 주소만을 이용하는 단순한 패킷 스케줄링을 사용하기 때문에 클러스터를 구성하는 서버들 간의 부하 불균형을 가중시키며, 결과적으로 클러스터의 효율성을 저하시키는 문제점을 가지고 있다. 본 논문에서는 이러한 문제점을 해결하기 위하여 RR ONE-IP 기법을 제안하였다. 제안한 기법은 서버에 할당되는 부하간에 불균형이 발생하는 문제점을 해결하기 위해 TCP 연결 단위의 스케줄링 전략을 사용하였으며, 서버의 부하 정보를 이용하지 않는 RR 스케줄링 기법을 도입함으로써, ONE-IP 기법의 장점을 그대로 유지하면서 보다 나은 부하의 균등한 분배로 시스템의 처리 능력을 향상시키도록 하였다. 또한, 실험을 수행한 결과 제안한 기법이 기존의 기법에 비해 평균 3.84%의 시스템의 성능 향상을 보였으며, 과부하 발생율에서는 평균 23.5%의 감소를 가져왔음을 보였다.

1. 서론

급증하는 웹 서비스 요구에 대하여, 빠른 응답과 지속적인 과일 전송을 보장할 수 있는 고성능의 웹 서버에 대한 연구가 활발히 진행되고 있다. 하나의 서버에 대한 성능을 개선하는 방법으로는 많은 서비스 요청을 처리하는데 있어서 명확한 한계가 있으며, 다수의 서버 컴퓨터를 클러스터링 하여 서버의 부하를 분담시키는 웹 서버 클러스터 기술이 최근 효율적인 대안으로 각광을 받고 있다.

웹 서버 클러스터로 유입되는 모든 패킷에 대해 매번 어느 서버로 패킷이 전송될 것인가를 결정하는 작업을 패킷 스케줄링이라고 하는데, 이것은 클러스터를 구성하는 각 서버에게 사용자 요청이 고르게 분배되도록 하는 매우 중요한 과정이다. 스케줄링이 합리적으로 이루어지지 않는다면, 특정한 서버에서 과부하(overload) 혹은 저부하(underload) 상황이 발생하며 결국 전체 클러스터의 성능을 저하시키는 요인으로 작용한다.

Cisco사의 LocalDirector™, IBM의 NetworkDispatcher™, LinuxVirtualServer 프로젝트의 LVS-DirectRouting[1-3]과 같은 분산기 기반 기법들은 클러스터의 각 서버로부터 부하 정보를 수집, 분석하여 스케줄링을 수행하는 중앙 집중적인 스케줄링 장비를 사용한다. 이러한 기법들은 보다 많은 정보를 토대로 정확한 스케줄링을 수행하여 부하 분배에 있어서 유리하다는 장점이 있으나, 분산기에 패킷에 대한 정보 수집 및 처리 부하가 집중되어 병목 현상 및 단일점오류 문제가 발생할 가능성이 크다는 단점을 지닌다.

O. P. Damani와 P. E. Chung, Y. Huang[4]은, 분산기 기반 기법의 중앙 집중적인 처리 방식에서 야기되는 문제점을 해결하기 위해 LAN 기반 구조에서 이더넷(Ethernet) 브로드캐스트(broadcast) 메시지를 이용하는 ONE-IP 기법을 제안하였다.

이 기법에서, 클러스터에 유입되는 모든 패킷은 이더넷 브로드캐스트 메시지 형식으로 전환되어 모든 서버에게 보내지고, 각 서버는 이를 선택적으로 수신하여 서비스 여부를 스스로 판단하게 된다. ONE-IP 기법은 병목 현상이나 단일점오류 문제를 근본적으로 해결하고 있으나, 임의(random)함수에 근접한 성능을 보이는 해쉬(hash)함수에 패킷 스케줄링을 의존함으로써 높은 수준의 부하 분배 효과를 얻지 못하며 결과적으로 클러스터의 전체적인 성능을 저하시키게 되는 문제점을 드러낸다[5].

본 논문에서는 이와 같은 ONE-IP 기법에서의 문제점을 인식하고, 기존 기법의 단점을 두 가지 합리적인 스케줄링으로 보완하는 RR ONE-IP 기법을 제안하였다. 제안 기법은 해쉬 함수에 의한 스케줄링을 RR 스케줄링으로 대체하고, TCP 연결 단위까지 서버에 분배되도록 하는 전략을 사용함으로써 보다 높은 수준의 스케줄링 성능을 얻을 수 있다.

2. ONE-IP 기법

2.1 ONE-IP 기법의 구조

웹 서버 클러스터를 구성하는 서버들은 이더넷 프로토콜을 사용하는 LAN에 의해 상호간의 연결을 이루고 있으며, 수신하는 패킷을 처리하기 위한 여과(filtering) 프로그램을 각 서버가 동일하게 내장하고 있다. 패킷에 대한 스케줄링은 이러한 여과 프로그램에 의해 수행되기 때문에 스케줄링을 위한 별도의 장치를 필요로 하지 않는다. 또한, N개의 서버에는 0부터 (N-1)까지의 일련 번호(server_ID)가 부여되어 있으며, 이것은 부하 분배를 위한 패킷 스케줄링에 이용된다.

2.2 클러스터 주소에 의한 단일서버이미지 제공

클러스터를 구성하는 모든 서버는 하나의 공통된 IP 주소를 자신의 두 번째 IP 주소(secondary address)로 등록하여 공유한다.

이러한 주소를 클러스터 주소라고 하며, UNIX를 기반 운영체제로 사용하는 서버에서 사용할 수 있는 *ifconfig alias* 옵션을 통해 유지될 수 있다. 서비스를 요청하는 모든 사용자 패킷은 이 하나의 클러스터 주소를 목적지로 하며, 동시에 단 하나의 서버에 의해서만 이 주소에 의해 서비스 된다. 클러스터가 제공하는 서비스 이외의 모든 내부적인 통신은 별도로 각 서버의 첫번째 주소(primary address)에 의해 이루어질 수 있다. <그림 2.2>는 *ifconfig alias* 옵션을 이용한 각 서버 및 라우터의 주소 구성을 보여준다.

ONE-IP 기법에서는 하나의 IP 주소 또는 URL에 의해 여러 대의 서버로 구성된 클러스터를 운영하기 위해 *ifconfig alias* 옵션을 이용하고 있지만, 실제로 이 옵션은 하나의 서버로 여러 주소의 서비스를 제공하기 위한 해결책으로 사용된다. 다시 말해서, 하나의 서버에 있는 단일 네트워크 인터페이스(single network interface)를 통해 다중의 IP 주소나 URL을 사용할 수 있도록 해 준다는 것이다. 따라서 이런 여러 주소를 목적지로 하는 사용자의 모든 서비스 요청 패킷은 하나의 서버에 의해 수신될 수 있다.

2.3 해쉬 함수에 의한 패킷 스케줄링

ONE-IP 기법은 구조적으로 병목 현상이 발생할 가능성이 있는 장치를 두지 않고 있다. 대신 사용자의 발신지 IP 주소에 대한 해쉬 함수 값을 이용하여 정적인 스케줄링을 하고 있다. 이러한 경우 해쉬 함수의 특성에 따라, 동일한 TCP 연결에 포함된 모든 패킷은 하나의 서버에 매핑되어 처리될 수 있음을 보장할 수 있다. 또한 서버의 정보 등을 이용하지 않으므로 빠른 스케줄링이 가능하다는 이점이 있다.

아래의 식(1)은 각 서버에 내장되어 있는 여과 프로그램의 해쉬 함수에 의한 조건식을 나타낸다. 패킷의 발신지 주소가 (1)의 조건을 만족하는 서버 *k*는 서비스 요청을 받아들이게 된다. 여기에서 *k*, *CA*, 그리고 *N*은 각각 서버의 일련 번호, 사용자의 *IP* 주소(Cluster Address: 4bytes), 클러스터를 구성하는 서버 개수를 의미한다. 연산자 %는 modular연산을 뜻한다.

$$k = CA\%N \quad \text{.....(1)}$$

2.4 문제점

ONE-IP 기법은 스케줄링의 부하 균등화 성능 면에서 매우 심각한 문제점을 가지고 있다. 패킷의 발신지 IP 주소만을 이용하는 단순한 스케줄링을 사용하기 때문에 야기되는 문제점은 크게 두 가지이다. 먼저, 발신지의 위치를 예측할 수 없는 인터넷의 특성에 따라, 해쉬 함수에 의존하는 스케줄링 기법의 성능은 임의(random) 함수의 성능에 근접한다는 사실이다. 임의 함수를 이용한 스케줄링의 성능은, 같은 정적 스케줄링 기법인 RR이나 LC에 비해 현저히 낮은 수준임을 알 수 있다.

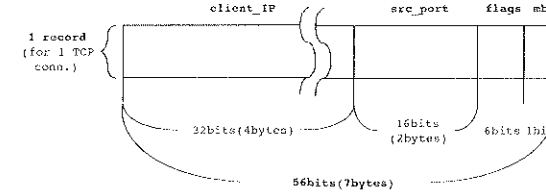
동일한 사용자로부터 요청되는 다수의 패킷이 독립적인 TCP 연결이나 사용자 세션(session) 단위에 관계없이 모두 하나의 서버에서 처리될 수 밖에 없다는 점을 두 번째 문제점으로 지적할 수 있다. 웹 트레픽을 사용자 세션에 관련하여 연구한 논문에 따르면[6], 한 사용자에 의한 하나의 사용자 세션에서 2 ~ 10회의 TCP 연결 요청이 발생한다. 이러한 TCP 연결 단위는 서로 다른 서버에서 독립적으로 처리될 수 있다.

3. RR ONE-IP 기법

3.1 RR 스케줄링

RR ONE-IP 기법은 기존 연구인 ONE-IP 기법의 구조를 그대로 따른다. 이것은 ONE-IP 기법이 가지고 있는 병목 현상 제거, 빠른 스케줄링 및 용이한 클러스터 구성 등의 장점을 그대로 유지하기 위함이다. 클러스터는 이더넷 프로토콜을 이용하는 LAN 환경에서 구성하며, 서브네트워크 라우터를 통해 클러스터로 유입되는 서비스 요청 패킷은 이더넷 브로드캐스트 메시지로 변환되어 모든 서버에게 전송된다. 각 서버의 장치 드라이버(device driver)에는 스케줄링에 필요한 자료 구조 및 여과 프로그램이 내장되어 있어서 수신한 패킷을 처리(accept)하거나 또는 폐기(discard)한다.

제안 기법에서는 서버 간에 부하 정보를 교환하지 않으면서 순차적인 스케줄링이 가능하도록 하기 위해서 두 가지 자료구조와 변수를 이용하였다. 먼저, 카운터(counter)는 서버가 패킷을 수신할 때마다 1씩 증가하도록 하여, 그 값이 자신의 일련번호와 일치하는지의 여부에 따라 처리할 차례임을 알도록 하



<그림 3.1> 연결 테이블의 레코드 구성과 필드 크기

는데, 이러한 과정을 통해 RR 패턴으로 스케줄링이 될 수 있다. 또한 패킷의 IP 헤더 및 TCP 헤더 정보를 검사하여, 동일한 TCP 연결에 속하는 패킷은 카운터 값에 관계 없이 같은 서버에서 처리할 수 있도록 하였고, 이를 가능하게 하기 위해 각 서버는 연결 정보 테이블(connection table)을 유지하도록 하였다. <그림 3.1>은 연결 정보 테이블의 필드 구성 및 하나의 레코드 크기를 나타낸다.

3.2 TCP 연결 단위 스케줄링 전략

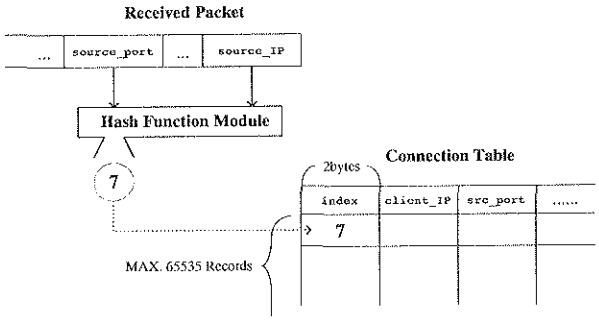
RR ONE-IP 기법은 기본적인 RR 스케줄링 외에, 독립적인 TCP 연결 단위 수준으로 패킷을 분배하는 전략을 사용한다. <그림 3.1>의 자료 구조에서 볼 수 있는 src_port, flags, 그리고 mb(mark_bit)등의 값은 이러한 TCP 연결 단위를 구분하기 위해 필요한 최소한의 정보임을 알 수 있다.

패킷을 TCP 연결 단위로 분배하는 전략은 부하의 균등한 분배의 측면에 있어서 기존의 ONE-IP 기법이 지니는 문제를 해결할 수 있다. 제안하는 전략에서는, 각각의 서버마다 스케줄링에 src_port 및 flags 정보를 이용하도록 함으로써, 같은 사용자에 의한 요청이라 할 지라도 다른 서비스 포트 번호로부터 요청되는 TCP 연결인 경우 서로 다른 서버에 나뉘어 처리될 수 있도록 하고 있다. 최근의 연구에 따르면, 일반적으로 한 사용자는 평균 2~10개의 서비스 세션을 사용하며 각 세션은 평균 2~16회의 TCP 연결 요청으로 이루어지며[6], 따라서 이와 같은 특성을 고려한 TCP 연결 단위 수준의 패킷 스케줄링은 부하의 균등화 면에서 보다 우수한 성능을 나타낼 수 있다.

서비스 요청 패킷이 웹 서버 클러스터에서 스케줄링 되는 과정은 다음과 같다. 먼저, 클러스터 주소를 목적지 IP 주소로 가지는 서비스 요청 패킷이 클러스터의 게이트웨이(gateway) 역할을 담당하는 서브네트워크 라우터에 도달한다. 라우터는 라우팅 테이블을 참조하여 서비스를 요구하는 패킷은 모두 클러스터가 구성된 LAN으로 연결된 포트로 전달되도록 한다. 이 때, ONE-IP 기법과 같은 방법으로 클러스터 주소의 ARP 캐쉬가 변경되는 문제점을 해결하기 위해 수정된 라우팅 테이블 구성을 사용한다. 즉, 클러스터 주소를 향하는 패킷은 고스트 IP 주소가 위치하는 서브네트워크로 전송되며, 이에 따라 패킷은 1-홉(hop)의 지연시간을 거쳐 다시 서브네트워크 라우터에 도달하여 고스트 IP에 해당하는 라우팅 테이블을 참조한다.

이후, 라우터는 패킷의 앞부분에 이더넷 프로토콜 헤더를 부착하는데 이 때 6bytes의 이더넷 목적지 주소는 브로드캐스트 메시지, 즉 FF... FF(6byte)가 된다. 헤더가 부착된 패킷은 ARP 엔트리 테이블의 정보에 의해 이더넷 LAN에 전송되며, 각 서버는 브로드캐스트 주소를 인식하여 일단 모두 패킷을 수신한다. 각 서버에 내장된 여과 프로그램은 수신한 패킷의 발신지 IP 주소, 발신지 TCP 포트 번호, TCP flags 값의 세가지 정보를 검사하여 패킷에 대해 응답을 할 것인지 혹은 거부할 것인지의 여부를 판단한다.

RR ONE-IP 기법은 알고리즘에서 나타나는 몇 가지 자료 구조를 유지하여야 하므로 패킷을 처리할 때마다 지연 시간이 누적되는 오버헤드가 발생할 수 있다. 이러한 문제에 대해서는, 연결 유지 테이블을 접근하는 데에 해쉬 함수를 사용함으로써



<그림 3.3> 연결 유지 테이블 접근

지연 시간을 최소화 하고 있다. 패킷이 어느 TCP 연결에 속하는지를 알기 위해 필요한 정보는 client_IP와 src_port 정보이다. 두 가지 정보를 합치면 하나의 레코드에서 6bytes를 차지하며 또한 이는 테이블에서 유일한 식별자(identifier)가 된다. 따라서 해쉬 함수에 두 정보를 입력하여 생성되는 결과값을 인덱스로 사용한다면, 최대 65535개의 TCP 연결 레코드를 포함하고 있는 테이블을 순차적으로 검색하지 않고도 곧바로 관련된 TCP 연결 레코드에 접근할 수 있다. 이를 위해 테이블에는 2bytes(16bits: 2^{16}-1 = 65535)의 크기를 가지는 index 필드가 추가된다. 이와 같은 과정을 <그림 3.3>에 나타냈다.

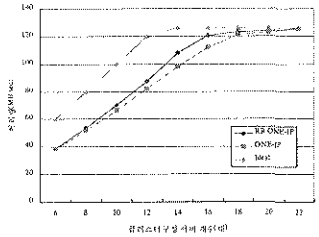
4. 실험 및 결과 분석

4.1 성능 평가 요소 및 모델링

성능 평가 기준은 주어진 시간 동안의 서비스 요청에 대한 데이터 전송 처리량(throughput)과 각 서버들의 과부하 발생율(probability of overload)의 함으로 나타내는 클러스터 과부하 발생율이다. 두 가지 요소는 ONE-IP 및 RR ONE-IP 기법의 패킷 스케줄링 효율성 및 서비스 처리 성능을 나타내는 기본적인 지표가 되며, 패킷 분배가 고르게 이루어질수록 그 결과로 높은 처리량과 낮은 과부하 발생율을 얻을 수 있다.

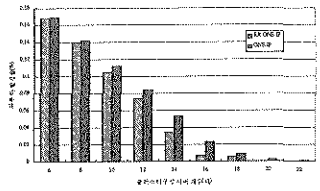
클러스터를 구성하는 서버와 서비스 요청 패킷을 발생시키는 트래픽 생성을 위해 SPECweb99™[7]의 데이터 전송 요청 파일의 혼합비율과 사용자 요청 당 TCP연결 요청 횟수에 관련된 연구[6]를 참고하였다. 연구 보고에 따르면, 한 사용자에 의해 3회 이상 동일한 발신지 IP 주소로부터의 요청이 발생하는 경우가 전체 트래픽의 75%를 차지하고 있음을 알 수 있다. 이러한 점은 발신지 IP 주소만을 이용한 해쉬 함수로 패킷을 스케줄링하는 기존의 ONE-IP 기법에 매우 불리한 특성이다. ANSI C Compiler를 이용하여 시뮬레이션 엔진을 구현하고, 일반적인 Pentium II-450MHz CPU의 WindowsNT W/S 환경에서 시뮬레이션을 수행하였다.

4.2 결과 및 분석



<그림 4.1> Heavy Traffic, 처리량

<그림 4.1>는 heavy traffic의 상황에서, 서버의 개수 변화에 따른 웹 서버 클러스터의 처리량 변화를 보여준다. 그림의 가장 위쪽에는 모든 서버에 이상적으로 패킷이 스케줄링되는 이론상으로 가능한 경우의 처리량을 표시하였다. 실험 결과에서, 14대의 서버를 이용하여 클러스터를 구성한 경우, RR ONE-IP 기법이 기존 ONE-IP 기법에 비해 최대 10.26%의 처리량 향상을 보이고 있으며 전체적으로는 평균 4.15%의 향상되었음을 알 수 있다. 이러한 결과는, 기존 기법의 임의 함수에 근접하는 스케줄링 방식에 비해 제안 기법의 순차적인 스케줄링 기법과 TCP 연결 단위 스케줄링 전략의 사용이 보다 효과적으로 패킷을 분배하고 있음을 보여주고 있다.



<그림 4.4> Heavy Traffic, $P_{overload}$

<그림 4.2>는 heavy traffic에서의 서버 개수 변화에 따른 클러스터의 과부하 발생율을 나타내고 있다. 그래프를 보면 구성 서버의 개수가 6대 혹은 8대와 같이 적은 경우에는 제안 기법과 기존 기법 모두 스케줄링에 의한 부하 분배 효과를 효과적으로 얻지 못하고 있으며, 이에 따라 높은 과부하 발생율을 나타내고 있다. 그러나 서버의 수가 적정 수준이 되는 14대의 경우, RR ONE-IP 기법이 ONE-IP 기법에 비해 절대량으로 0.019%의 발생율을 감소시켰다. 이는 상대적인 비교로 볼 때 35.6%의 발생율 감소율을 나타내며, 클러스터가 수신하는 패킷의 수에 대비하여 볼 때, 10,000번의 요청에서 1.9회의 패킷 손실 횟수를 줄이고 있는 것과 같다.

5. 결론

웹 서비스를 저비용으로 신속하게 처리하는 LAN 환경에서의 서버 클러스터는, 안정적이고 빠른 서비스 응답을 하기 위해 서버들에게 사용자가 요구하는 서비스 요청 패킷을 효과적으로 분배할 수 있는 스케줄링 기법을 필요로 한다.

본 논문에서는 기존 ONE-IP 기법이 가지는 여러 장점을 그대로 유지하면서 보다 효과적인 패킷 부하를 균등하게 분배하는 전략인 새로운 RR ONE-IP 기법을 제안하였다. 제안한 기법에서는 독립적인 TCP 연결 단위까지 복수의 서버에 나누어 처리하였으며, RR 스케줄링 기법을 도입하여 보다 균등한 분배 성능 향상을 유도하였다. 제안한 전략은 기존 기법에 비해 평균 3.84%의 클러스터 처리량의 성능 향상을 얻었으며, 서버의 평균 과부하 발생율을 28.5% 낮추고 있다.

RR ONE-IP 기법은 LAN으로 구성되는 클러스터 환경을 전제로 하며, 서비스 요청이 빈번한 상황일수록 효과적인 분배 성능을 얻을 수 있는 기법이다. 제안된 스케줄링 기법을 적용한 여과 부하을 각 서버의 장치 드라이버에 이식하고 클러스터를 구성하여 실제 실험을 수행한다면 보다 현실적인 네트워크 상황이 고려된 실험 결과를 얻을 수 있을 것으로 생각된다.

6. 참고문헌

[1] Cisco Systems Inc., Cisco LocalDirector Version 4.1 Documentation. http://www.cisco.com/univercd/cc/td/doc/product/iaabu/localdir/

[2] G.Hunt et al., Network Dispatcher: A Connection Router for Scalable Internet Services, *Proc. 7th Intl World Wide Web Conf.*, 1998.

[3] Wensong Zhang, Linux Virtual Servers for Scalable Network Services, *The 22nd Ottawa Linux Symposium*, July 19, 2000. http://www.linuxvirtualserver.org/

[4] V. Cardellini et al., Dynamic Load Balancing on Web-Server Systems, *IEEE Internet Computing*, Vol.3, No.3, May/June 1999, 28-39.

[5] O. P. Damani et al., ONE-IP: Techniques for Hosting a Service on a Cluster of Machines, *J. Computer Networks and ISDN Systems*, Vol.29, Sept. 1997, 1019-1027.

[6] M. Arlitt, Characterizing Web User Sessions, *Performance and Architecture of Web Servers(PAWS)*, June 17-18, 2000, Santa Clara, California, ACM SIGMETRICS 2000.

[7] SPECweb99 Design Document, *White Paper, Standard Performance Evaluation Corporation(SPEC)*, 2000. http://www.specbench.org/osg/web99/docs/whitepaper.html