다중 처리기에서 웹 서버의 구현을 통한 실험적 성능 평가

정진국 낭종호
서 강 대 학 교 컴 퓨 터 학 과
jjguk@plstar.sogang.ac.kr jhnang@ccs.sogang.ac.kr

Implementation and Experimental Analysis of Concurrent Web Servers on Multiprocessor

Jin Guk Jeong Jong Ho Nang
Dept. of Computer Science & Engineering, Sogang University

요 약

WWW의 급격한 발전은 고성능 웹 서버의 구축을 필요로 하게 하였다. 특히 프로세스의 오버헤드를 줄이기 위해 도입된 멀티 쓰레드 기법을 이용한 병행 웹 서버들이 많이 연구되었는데 본 논문에서는 이런 웹 서버들을 리눅스가 탑재되어 있는 다중 처리기상에서 구현하였으며, 다양한 환경 하에서 성능을 비교, 분석하였다. 실험을 통하여 Thread Pool 구조 웹 서버가 가장 좋은 성능을 보임을 알 수 있었고, 작업 기반 웹 서버와 요구 기반 웹 서버의 성능은 환경에 따라 차이가 있음을 알 수 있었다. 이와 같은 실험 결과는 다중 처리기를 이용한 고성능 웹 서버를 구축하는 데 있어서 이용될 수 있을 것이다.

1. 서론

WWW(이하 웹)은 불과 몇 년 사이에 폭발적인 발전을 보이고 있다. 이와 같은 급격한 웹의 발전은 몇몇 웹 서버에 사용자의 요구가 집중되는 현상을 보이게 하였다. 특히 서버에서 동적으로 수행되는 프로그램들 - CGI, Script, Java Servlet... - 을 사용하는 빈도가 많아지고 있으므로 사용자의 요구가 집중되는 현상은 몇몇 서버에 큰 과부하를 가져오게 할 것이다. 그러므로 사용자의 요구에 신속하게 응답할 수 있는 고성능 웹 서버의 구축이 필요하게 된다.

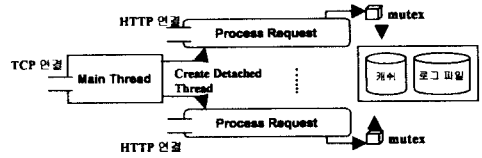
웹 서버에 관한 연구는 여러 방향으로 진행되어 왔었다. 하지만 기존의 연구들 대부분은 단순히 텍스트와 그림 파일과 같은 작은 파일들의 요구에 기반을 두고 있기 때문에 동적인 프로그램의 요구가 늘어나고 있는 최근의 웹 전송방식에 적용하는 데는 문제가 있고, 또한 단일 처리기상에서의 실험에 중점을 두고 있어 그 결과가 최근 많이 보급되어 있는 다중 처리기에서도 보장되는 지는 알 수 없다는 문제점이 있다.

본 논문에서는 최근 널리 보급되고 있는 다중 처리기상에서 세 가지 병행 웹 서버 구조 - 요구 기반 웹 서버(Request Based Web Server, 이하 RBW)[2,3], 작업 기반 웹 서버(Task Based Web Server, 이하 TBW)[6], Thread Pool 구조 웹 서버(Thread Pool Web Server, 이하 TPW)[4,5]구현하여 다양한 요구들과 환경에서 성능을 측정한 후 분석하였다. 본 논문의 실험 결과는 고성능 웹 서버를 구축하는 데 있어 참고 모델로 이용될 수 있을 것이다.

2. 병행 웹 서버의 구현

본 논문에서는 멀티쓰레드 기법의 세 가지 웹 서버 구조- RBW, TBW, TPW -를 이용하여 실험을 하였다. 이 장에서는 이 세 가지 웹 서버 구조 각각의 모델 특성과 구현 방법에 대하여 설명하도록 하겠다.

2.1 RBW의 구현

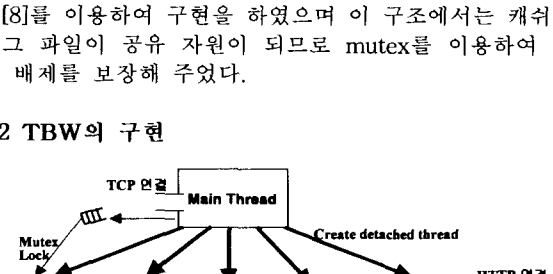


< 그림 1 > RBW의 구현

RBW 구조[2,3]라이언트의 요구에 대한 병렬성을 감안하여 고안한 구조이다. 지정된 포트를 통해 클라이언트의 TCP 연결 요청이 들어오게 되면 메인 쓰레드는 클라이언트의 요구를 처리하는 쓰레드를 생성하게 된다. 생성된 쓰레드는 클라이언트와의 HTTP 연결을 만들게 되고, 이 연결을 통하여 클라이언트의 요구를 받아들이게 된다. 그 후 요구를 처리하는 일련의 과정을 처리하게 된다. 이 병행 구조는 한꺼번에 많은 요구가 들어오는 경우나 동적 프로그램의 요구가 많은 경우 서버에서 동시에 수행되는 쓰레드의 개수가 많게 되므로 예상치 못한 현상이 발생할 수 있다.

기본적으로 Pthread 라이브러리[7]와 Socket 라이브러리[8]를 이용하여 구현을 하였으며 이 구조에서는 캐쉬와 로그 파일이 공유 자원이 되므로 mutex를 이용하여 상호 배제를 보장해 주었다.

2.2 TBW의 구현

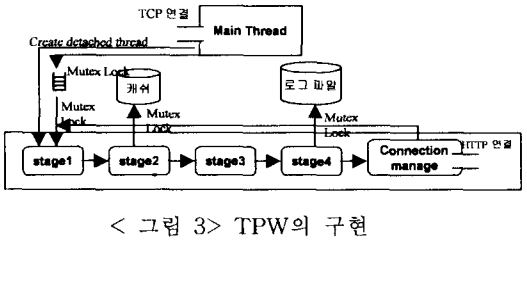


< 그림 2 > TBW의 구현

TBW[6]는 서버의 여러 수행 단계에 의해서 나올 수 있는 병렬성을 감안하여 고안한 구조이다. 이 구조에서는 서버의 각 단계별로 쓰레드가 할당이 된 후 각 단계에 할당된 쓰레드는 반 영구적으로 존재하여 자기가 맡은 단계를 책임지고 수행을 하게 된다. 이 구조에서는 각각의 요구를 단계사이에 전달을 할 때 큐를 이용하게 되는데, 단계사이의 처리 시간 차이가 클 경우 그 시간의 차이가 큐에서 누적이 되게 되어 큰 영향을 끼치게 된다. 또한 이 모델에서는 각각의 요구 처리 시간이 다른 요구 처리 시간에 끼치는 영향이 RBW보다는 큰 특징을 가지고 있다.

구현을 위해 이용한 라이브러리는 RBW와 같고, 이 구조에서는 단계 사이의 큐가 공유 자원이 되므로 상호 배제를 보장해주어야 한다.

2.3 TPW의 구현



< 그림 3> TPW의 구현

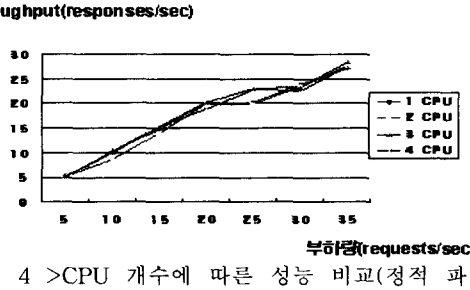
TPW[4,5]는 앞의 두 구조를 혼합한 형태이다. 이 구조는 서버가 요구를 처리하는 전 과정을 하나의 단계로 보게 된다. 큐는 요구를 받는 큐 하나만 존재하고 요구를 처리하는 쓰레드는 고정된 수의 쓰레드만 존재하게 된다. 각각의 쓰레드는 요구를 처리한 후에 수행을 종료하는 것이 아니라 요구를 받는 큐로 가서 다른 요구를 처리하게 된다. 즉, 이 구조에서의 쓰레드 또한 TBW와 마찬가지로 반 영구적인 쓰레드이다.

구현을 위해 사용한 라이브러리는 앞의 두 구조와 마찬가지이고, 이 구조에서는 큐, 로그 파일, 캐쉬가 공유 자원이 되어 상호 배제를 보장해주어야 한다.

3. 실험 및 분석

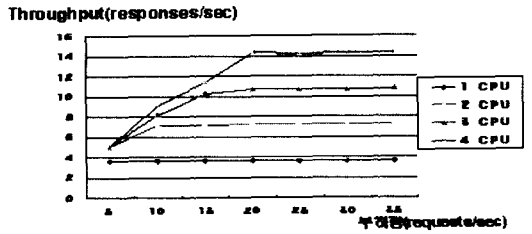
실험을 위하여 세 가지 웹 서버 모델을 리눅스를 탑재한 다중 처리기 상에서 구현을 하였다. 이용된 서버는 SMP(Symmetric Multiprocessing) 구조를 이용하고 4개의 CPU- Pentium II Xeon 450Mhz -와 256MB의 메모리를 가지고 있다. 네트웍으로는 10 Base-T LAN 환경을 이용하였다.

우선 각 웹 서버 구조의 특징을 알기 위하여 모든 요구에 대하여 서비스 시작 시간과 종료 시간에 대한 분석을 하였다. 그 결과 RBW는 시작 시간이 서로 비슷한 여러 요구들 중에 종료 시간이 다른 요구들에 비해 현저히 늦은 요구들이 있었다. TBW는 각 요구들의 처리 시간이 다른 요구들의 처리 시간에 영향을 주는 문제점이 있었고 TPW는 위의 두 구조의 문제점은 많이 보완이 되었지만 시작 시간 자체가 늦어지는 문제점이 있었다. RBW는 한번에 많은 쓰레드가 수행을 할 경우 그 쓰레드의 스케쥴링에 의하여 위와 같은 문제가 생기는 것이고, TBW는 단계 사이의 큐에서 요구들이 기다리게 되므로 서로 처리 시간에 영향을 주게 되는 것이다. 그리고 TPW는 요구를 받아들이는 큐에서 요구들이 쓰레드를 기다리게 되어 시작 시간 자체가 늦어지는 문제점이 생기게 되는 것이다.



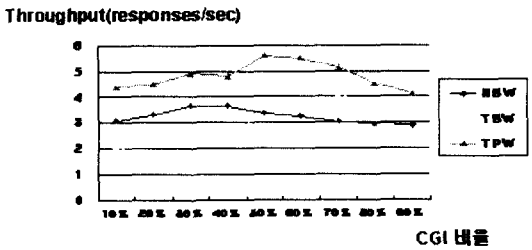
< 그림 4 >CPU 개수에 따른 성능 비교(정적 파일 - 115K - 요구, TBW)

다중 처리기 상에서 CPU 개수에 따른 성능을 비교 분석한 결과를 보면 <그림 4>와 같이 정적 파일의 요구에 대해서는 CPU 개수에 따른 성능 차가 거의 없음을 알 수 있었다. 이는 정적 파일 요구에 대해서는 웹 서버가 CPU를 거의 사용하지 않음을 알 수 있는 것이다.



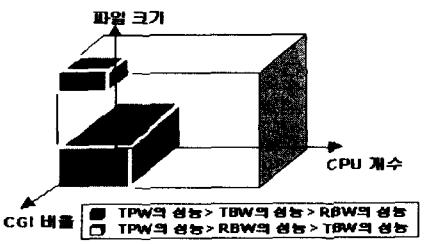
< 그림 5 >CPU 개수에 따른 성능 비교(CGI 요구, TPW)

이에 반해 동적 프로그램 요구에 대해서는 <그림 5>를 보면 알 수 있듯이 CPU 개수에 따른 성능 차가 확연히 드러났다. 앞에서도 언급했다시피 동적 프로그램 요구를 처리하는 경우에는 fork( ), exec( )와 같은 과정을 거친 후 프로세스를 생성하게 되는 데 이 과정에서 CPU가 많이 이용이 되기 때문에 위와 같은 결과가 나타나는 것이다.



<그림 6 >웹 서버의 성능 비교(1CPU,115K 정적 파일)

정적 파일 요구와 CGI와 같은 동적 프로그램에 대한 요구를 혼합하는 경우에는 CGI 비율, 정적 파일 크기, CPU 개수, 서버의 종류 등에 따라서 다른 결과를 나타냈다. 예를 들면 <그림 6>는 1개의 CPU를 사용하고, 115K의 정적 파일을 이용하였을 때의 그래프이다. 그래프를 보면 알 수 있듯이 TPW는 CGI 비율에 상관없이 가장 좋은 성능을 보이지만, RBW와 TBW는 조건에 따라 성능의 차이가 나고 있다. RBW는 하나의 CPU에서 수행이 되는 쓰레드의 개수가 많은 경우, TBW는 단계 사이의 수행 시간이 서로 차이가 많이 나는 경우 성능이 떨어지는 데 이러한 상황을 나타낼 수 있는 환경이 서로 다르기 때문에 위와 같은 결과를 나타내는 것이다.



< 그림 7 > 웹 서버의 성능 비교

<그림 7>은 여러 가지 조건에서 웹 서버의 성능을 비교한 그림이다. CPU 개수가 작고 요구하는 정적 파일의 크기가 작은 경우, 그리고 CPU 개수가 작고 요구하는 정적 파일 크기가 크고 CGI의 비율이 높은 경우에는 TPW, TBW, RBW의 순으로 전체 성능이 측정이 되고 나머지 경우에는 TPW, RBW, TBW 순으로 전체 성능이 나타났다. TPW의 경우에는 쓰레드의 개수가 조절이 되고 큐도 하나만 존재하기 때문에 큐에 대한 병목 현상이 조절이 되어 다른 웹 서버에 비해 좋은 성능을 보임을 알 수 있었다. 이에 반해 RBW와 TBW는 병목 현상을 조절하는 기능이 떨어지는 것을 볼 수 있었는데, RBW는 하나의 CPU에서 수행이 되는 쓰레드의 개수가 많은 경우 좋지 않은 성능을 보임을 알 수 있었고, TBW는 단계 사이에서 걸리는 수행 시간에 많은 차이가 보이는 경우 큐의 병목 현상으로 인하여 좋지 않은 성능을 보임을 알 수 있었다.

4. 결론

기존의 웹 서버에 대한 연구는 보통 하나의 웹 서버 구조, 정적 파일 요구 그리고 하나의 CPU를 가지고 실험을 한 경우가 대부분이다. 본 논문에서는 멀티 쓰레드 기법을 이용한 세 가지 웹 서버 구조 즉 RBW 구조, TBW 구조, TPW 구조를 리눅스를 탑재한 다중 처리기 상에서 구현을 하였고 다양한 요구 - 정적 파일 요구, 동적 프로그램 요구 - 와 환경 - CGI 비율, CPU 개수, 웹 서버의 구조, 정적 파일 크기 등 - 하에서 실험과 결과에 대한 분석을 하였다.

이러한 다양한 조건에서의 실험 결과는 다중 처리기 상에서 고성능 웹 서버를 구축하는 데 있어 참고 모델로 이용할 수 있을 것이다.

5. 참고 문헌

[1] N.J. Yeager and R.E. McGrath, *Web Server Technology*, Morgan Kaufman Publishers, 1996.

[2] P. Eriksson, *phttpd*, http://phttpd.signum.se/phttpd, 1999.

[3] J.C. Hu, S.M. Mungee, and D.C. Schmidt, *Principles for Developing and Measuring High-performance Web Servers over ATM*, Technical Report WUCS-97-09, Washington University, 1997.

[4] WWW Consortium, *Jigsaw HTTP Server*, http://www.w3.org/Jigsaw/, 1997.

[5] Netscape Inc., *Netscape Enterprise Server*, http://home.netscape.com/enterprise/v3.6/, 1999.

[6] 제영희, *병행 웹 서버의 설계 및 성능 평가*, 서강대학교 석사 학위 논문, 1997.

[7] B. Nichols, D. Buttlar and J.P. Farrell, *Pthreads Programming*, OReilly & Associates Inc., 1996.

[8] W.R. Stevens, *Unix Network Programming*, Prentice-Hall Press, 1991.

[9] Remy Card, Eric Dumas, Franck Mevel, *The Linux Kernel Book*, Wiley Press, 1998.