

다중 프로세스와 스레드 모델의 웹 서버 성능 비교 분석

정기훈^o 염미령 노삼혁

홍익대학교 컴퓨터공학과, 전자계산학과, 정보컴퓨터공학부

{khchong, miryeong}@cs.hongik.ac.kr, samhnoh@hongik.ac.kr

Performance Measurement and Analysis of the Web Server Using Multi-Process and Multi-Thread

Kihun Chong^o Mi-ryeong Yeom Sam H. Noh

Dept. of Computer Engineering, Computer Science, Information and Computer Engineering, Hongik Univ.

요 약

다중 스레드 모델은 다중 프로세스 모델에 비하여 확장성이 좋은 것으로 알려졌다. 이에 따라서 초당 수백내지 수천개의 컨넥션들을 처리해야 하는 웹 서버에 다중 스레드 모델을 채택해야 하는 것을 긍정적으로 생각해 볼 수 있다. 하지만 다중 프로세스 모델을 적용한 웹 서버와 두 종류의 혼합형 다중 프로세스/스레드 모델을 적용시킨 웹 서버의 성능을 평가하였다. 실험은 Pentium-III 750의 Linux 2.4.5에서 진행하였으며, 실험 결과 다중 스레드 모델이 다중 프로세스 모델에 비하여 항상 확장성이 있다고 할 수 없으며, 다중 프로세스 모델이 다중 스레드 모델보다 더 좋은 안정성을 보이는 것으로 나타났다.

1. 서론

스레드(thread)는 프로세스(process)의 단점을 보완하고, 장점을 살리기 위해 나온 개념이다[8]. 단일 프로세스로 구성된 프로그램보다 다중 프로세스로 구성된 프로그램에서는 프로세스를 새로 생성하거나(fork) 없애는(kill) 일이 빈번하게 이루어지는 경우가 많은데, 특히 프로세스간의 문맥교환(context switch)이 자주 일어난다. 이러한 환경에서는 프로세스를 생성하고, 없애고, 문맥교환 하는 오버헤드가 크다. 즉, 각 프로세스마다 동일한 프로그램 실행 코드와 프로세스 고유의 데이터를 갖게 되는데 동일한 프로그램 실행 코드를 같은 프로그램의 모든 프로세스가 갖고 있기 때문에 프로세스를 생성하거나 없애는 경우에 오버헤드가 크다. 게다가 프로그램 수행 중에 빈번히 일어나는 문맥교환은 이러한 오버헤드를 가중시키게 된다. 따라서 이러한 문제를 해결하기 위하여 같은 프로세스 내에서 중복되는 실행 코드와 데이터 등을 공유할 수 있도록 스레드를 만들었다.

스레드는 경량 프로세스(lightweight process, LWP)라고도 불리며, 프로그램 내의 실행 코드, 데이터뿐 만 아니라 파일 기술자, 시그널등의 운영체제의 자원도 공유를 하기 때문에, 같은 일을 여러 번 수행하는 경우에 다중 프로세스방식에 비해 생성, 소멸, 문맥교환 등의 오버헤드가 훨씬 적게 들게 된다. 이러한 특성으로 인하여 점점 다중 스레드 방식의 프로그램이 증가하고 있다. 특히 웹 서버의 경우 여러 가지 프로세스 또는 스레드 모델이 존재하며, 각 모델의 장단점도 각기 다르다.

일반적으로 웹 서버는 다중 프로세스 모델을 많이 사용하는데 최근 들어 다중 스레드 모델이 나타나기 시작했다. 따라서 본 논문에서는 웹 서버의 구동 방식, 특히 다중 프로세스 모델(DPP)과 두 종류의 혼합형 다중 프로세스/스레드 모델(HMPT)에 대한 성능을 평가하고 비교하도록 하겠다. 실험은 위의 세 가지 모델을 모두 적용할 수 있는 Apache 웹 서버를 웹 서버 벤치 마크인 WebStone을 이용하여 진행하였다. Netcraft[9]에 따르면 최근 30일간 가장 많은 요청이 있었던 50개의 사이트의 운영 체제를 조사한 결과 2002년 3월 25일 현재 Linux가 가장 많은 사용을 하고 있는 것으로 나타났다. 따라서 성능 평가는 Linux에서 하였으며 이런 이유로 인하여 프로세스와 스레드는 Linux상에서의 프로세스와 스레드로 국한되었다. 성능 평가 결과 각 모델마다 장단점이 있었으며, 안정성에 있어서는 DPP 모델이, 효율성에 있어서는 HMPT 모델이 더 좋은 것으로 나타났다.

논문의 구성은 다음과 같다. 먼저 2절에서는 관련 연구에 대하여 논한다. 3절에서는 현재의 웹서버의 구동 방식에 대하여 알아보고 4절에서는 이에 대한 실험 및 성능 평가를 하고 그 결과에 대하여 논한다. 마지막으로 5절에서 결론 및 향후 과제에 대하여 언급한다.

2. 관련 연구

모든 웹 서버가 다중 프로세스 또는 다중 스레드 모델로만 구성된 것은 아니다. 단일 프로세스 또는 스레드가 모든 클라

이벤트의 요청뿐 만 아니라 I/O 처리 및 응답까지 이벤트에 맞추어 처리하는 방식인 이벤트 처리(event-driven) 또한 많은 웹 서버가 사용하는 모델이다. 이러한 모델을 사용하는 웹 서버로는 Flash 웹서버, IIS(Internet Information Services), Zeus 등이 있다[2][3][4].

또 다른 모델로 커널영역에서 동작하는 방식이 있는데, 보통 커널 스레드를 이용하여 동작하며 정적(static)문서의 요청만 처리하고 동적(dynamic)문서의 처리는 사용자 영역(user-level)의 웹 서버로 넘겨준다. 이러한 모델을 사용하는 웹 서버로는 kHTTPd, TUX 등이 있다[5][6].

실제로 사용자가 많이 사용하는 웹 서버는 Apache 웹 서버이다[1][9]. Apache 웹 서버는 초기 버전에서는 다중 프로세스 모델을 사용하였으나 새로이 개발중인 버전에서는 다중 프로세스와 다중 스레드 모델을 섞어서 사용하고 있다. 따라서 본 논문에서는 다중 프로세스와 스레드 모델에 대한 성능을 비교하기 위하여 Apache 웹 서버를 사용하였다.

3. 웹 서버의 구동 방식

이 절에서는 현재 가장 많이 사용하고 있는 웹 서버의 구동 방식, 특히 다중 프로세스 모델과 다중 스레드 모델에 대하여 알아보도록 하겠다.

3.1 DPP(Dynamic Process Pool)

DPP(Dynamic Process Pool)모델은 프로세스 풀 안에 여러 프로세스가 대기하고 있다가 요청이 들어오면 대기중인 프로세스가 맡아서 처리하는 방식이다. 일반적으로 MP(Multi Process) 모델이라고도 불리며 가장 보편적인 다중 프로세스 모델이다. 보통 요청을 기다리는 프로세스가 따로 있어서 요청이 들어오게 되면 요청을 처리하는 프로세스는 현재의 프로세스 풀을 확인하여 한가한 프로세스에게 해당 요청을 넘겨주게 된다. 그러면 쉬고 있던 프로세스는 해당 요청을 넘겨 받아 요청에 대한 처리를 하게 된다. 요청에 대한 처리를 마치고 요청한 클라이언트에게 응답을 하게 되면 다시 프로세스는 프로세스 풀에서 대기하게 된다.

확장성(scalability)의 향상을 위하여 이러한 프로세스 풀에 존재하는 프로세스의 수를 조절하기도 한다. 요청이 많지 않을 때에는 대기하는 프로세스 수를 적게 하다가 요청이 많아지면 대기하는 프로세스의 수를 늘리게 된다.

3.2 HMPT(Hybrid Multi-process/thread)

HMPT(Hybrid Multi-process/thread)모델은 기존의 DPP모델과는 달리 다중 프로세스와 다중 스레드 방식을 섞은 모델이다. 다중 프로세스 방식의 단점과 다중 스레드 방식의 장점을 결합하기 위하여 만들어진 모델이며, 기본적인 처리 방식은 프로세스는 스레드를 관리하고 스레드는 요청에 필요한 대부분의 작업을 하게 된다. 보통 정적 문서에 대한 처리는 스레드가

맡으며, 동적 문서에 대한 처리는 프로세스가 맡는다. 또한 전체 프로세스를 관리하기 위한 프로세스가 존재하기도 한다. 프로세스의 수를 고정시키느냐 스레드 수를 고정시키느냐에 따라서 다음의 두 가지 모델로 나뉜다.

3.2.1 DPP-STP (Dynamic Process Pool and Static Thread Pool)

DPP-STP(Dynamic Process Pool and Static Thread Pool)모델은 스레드의 수를 고정하고 프로세스의 수를 변하게 하여 처리량을 조절하는 방식을 사용한다. 프로세스를 생성하게 되면 그에 따라 많은 스레드가 생성되기 때문에 프로세스 생성에 대한 오버헤드가 크다. 그렇기 때문에 미리 프로세스와 스레드를 생성하게 되는데 각 프로세스는 자기가 관리하는 스레드를 생성하거나 소멸시킴으로써 항상 일정한 수의 스레드를 유지하게 된다.

3.2.2 SPP-DTP (Static Process Pool and Dynamic Thread Pool)

SPP-DTP(Static Process Pool and Dynamic Thread Pool)모델은 프로세스의 수를 고정하고 각 프로세스당 스레드의 수를 변하게 하여 처리량을 조절하는 방식을 사용한다. DPP-STP와 같이 처리량을 높이기 위한 스레드의 증가를 위하여 프로세스를 새로 생성하지 않기 때문에 처리량을 높이기 위한 스레드의 생성에 대한 부담은 DPP-STP보다 적다. 하지만 갑작스런 요청의 폭주 등에는 DPP-STP보다는 대응이 늦다.

4. 실험 및 성능 평가

본 논문에서는 단일 프로세스 또는 스레드 환경을 제외한 다중 프로세스와 다중 스레드 모델에 대해서 성능 평가를 하였다. 그 이유는 프로세스 모델과 스레드 모델의 장단점을 조사하기 위해서였으며, 이번 절에서는 실험 환경 및 성능 평가에 대한 결과를 차례로 논한다.

4.1 실험 환경

실험 환경은 Linux 환경으로 구축하였다. 서버 시스템은 750MHz P-III/128MB/Linux 2.4.5를 사용하였으며, 클라이언트 시스템은 1GHz P-III/256MB/Linux 2.4.5를 사용하였다. 클라이언트 쪽의 시스템이 서버 쪽 시스템보다 더 성능이 좋은 이유는 웹 서버에 많은 부하를 주기 위해서이다. 즉, 웹 서버의 시스템을 클라이언트 시스템보다 성능이 좋아서 클라이언트가 보내는 모든 요청을 거뜬히 처리할 수 있다면 성능 평가를 하는 의미가 없어지기 때문이다. 따라서 클라이언트 시스템은 웹 서버에 성능 평가를 위한 충분한 부하를 주어야 하며, 따라서 클라이언트의 시스템이 웹 서버의 시스템보다 성능이 더 좋아야 한다.

웹 서버는 Apache 웹 서버 2.0.16 베타 버전을 사용하였다[1] 버전 2.0에서는 버전 1.3과는 달리 DPP뿐 만 아니라 HMPT도 지원하기 때문에 HMPT 모델의 성능 평가를 하는데 매우 유

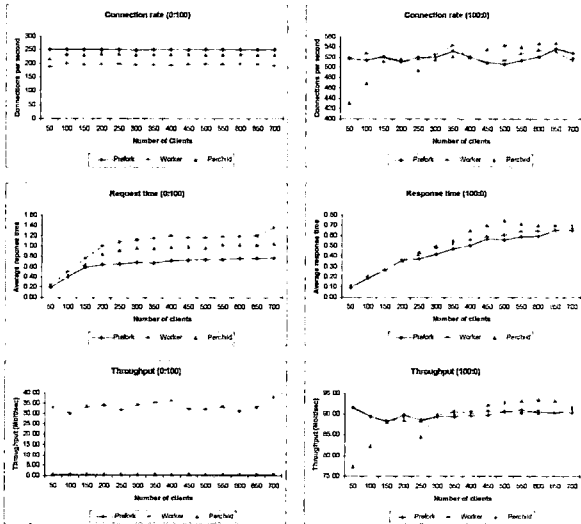


그림 1. 실험 결과. 위에서부터 순서대로 컨넥션 비율, 응답 시간, 처리율 순이며, 왼쪽은 정적 문서, 오른쪽은 동적 문서의 실험 결과이다.

용하다. 게다가 전 세계적으로 가장 많이 사용하는 웹 서버이며[9], 공개 프로젝트 방식으로 개발이 진행되고 있다.

웹 서버의 벤치마크는 WebStone 2.5를 사용하였다[7]. 결과는 각 집단마다 클라이언트의 수를 50에서 700까지 증가시키면서 세 번씩 실험하여 그 평균을 구하였다. Apache에서는 HMPT 모델은 Prefork, DPP-STP 모델은 Worker, SPP-DTP 모델은 Perchild라는 이름으로 지원을 하며, 설정 파일에서의 MaxClients 값을 80으로 맞추었다.

4.2 실험 결과

각 모델당 클라이언트 수에 따른 컨넥션 비율(connection rate), 응답시간(response time), 처리율(throughput)을 측정하였으며 그 결과는 그림 1에 나타내었다. 동적 문서의 경우 prefork 방식이 성능이 가장 우수했으며, 정적 문서의 경우에는 비슷비슷하지만 worker 모델이 성능이 가장 우수했다. 이것은 스레드를 사용함으로써 얻는 이득은 정적 문서를 처리할 때였으며, 동적 문서를 처리하는데 있어서는 처리율에서 보는 것과 같이 스레드가 오버헤드가 될 수 있다는 것을 보여준다. 동적 문서 처리의 경우, 스레드는 요청을 받고, 해당 동적 문서 처리 프로세스에게 그 결과를 받아서 클라이언트에게 응답하는 일만 하기 때문에 스레드의 역할이 상대적으로 적다. 따라서 많은 스레드가 오히려 오버헤드로 작용하게 된 것이다. 이와는 반대로 정적 문서 처리의 경우, 적은 수의 클라이언트에서는 prefork 모델이, 350이상의 많은 수의 클라이언트에서는 perchild 모델이 성능이 좋게 나타났다. 이것은 스레드 모델이 정적 문서 요청에 대해서 확장성이 좋다는 것을 보여 준다. 결론적으로 다중 프로세스 모델은 동적 문서 요청에 대하여 다중 스레드 모델보다 좋으며, 다중 스레드 모델은 다중 프로세스 모델

에 비하여 정적 문서 요청을 처리하는 능력이 좋다. 특히 정적 문서 처리에 있어서 저 부하일 때는 DPP-STP 모델인 prefork가 좋은 성능을 보였으며, 고 부하일 때는 SPP-DTP 모델인 perchild가 더 좋은 성능을 보였다.

5. 결론

본 논문에서는 다중 프로세스와 두 종류의 혼합형 다중 프로세스/스레드의 성능에 대한 비교를 다중 프로세스 또는 다중 스레드의 대표적인 프로그램인 Apache 웹 서버를 웹 서버 벤치마크인 WebStone을 이용하여 분석하였다. 실험 환경의 운영 체제는 웹 서버 운영에 가장 많이 사용되는 Linux를 사용하였으며, 그렇기 때문에 프로세스와 스레드는 Linux 환경의 프로세스와 스레드가 사용되었다. 혼합형 다중 프로세스/스레드 방식은 다중 프로세스 방식에 비해 확장성을 갖는다. 그러나, 다중 프로세스와 다중 스레드가 혼합된 방식으로 동작되는 HMPT 모델은 다중 스레드가 갖는 많은 장점을 수용하지만 다중 프로세스 보다 항상 좋은 확장성을 갖는 것은 아니었다. 정적 문서만을 서비스 할 때는 HMPT의 성능이 좋았지만, 동적 문서를 서비스 해줄 때는 DPP 모델의 성능이 우수했다. 따라서 다중 프로세스 다중 스레드 모델이 항상 다중 프로세스 모델보다 나은 성능을 보이는 것이 아니므로 웹 서버 관리자는 웹 서버가 수행해야 할 데이터의 성격에 맞는 웹 서버 모델을 선택해야 할 것이다.

참고 문헌

- [1] The Apache Group, Apache http Server Project, <http://www.apache.org>.
- [2] Vivek S. Pai, Peter Druschel, and Willy Zwaenpoel, "Flash: An Efficient and Portable Web Server," *In Proceedings of the USENIX 1999 Annual Technical Conference, 1999*
- [3] Zeus Technology Ltd., Zeus Web Server, <http://www.zeu.s.com>.
- [4] Microsoft Corporation, Internet Information Services Features, <http://www.microsoft.com/windows2000/server/evaluation/features/web.asp>.
- [5] Moshe Bar, "kHTTPd, a Kernel-Based Web Server," *Linux J.*, 2000
- [6] "Answers from Planet TUX: Ingo Molnar Responds," <http://slashdot.org/articles/00/07/20/1440204.shtml>.
- [7] Mindcraft, Inc., "WebStone—the Benchmark for Web Servers," <http://www.mindcraft.com/webstone>.
- [8] Arbaham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts, sixth ed.," John Wiley, 2001
- [9] Netcraft Web Server Survey, ["http://www.netcraft.com/survey"](http://www.netcraft.com/survey)