

다양한 운영체제에서의 다중 프로세스와 스레드 모델의 웹 서버 성능 비교[†]

도인환⁰ 김경중 정규현 정기훈 노삼혁
홍익대학교 컴퓨터공학과, 정보컴퓨터공학부

indhoh@cs.hongik.ac.kr⁰, skkkj@hanmail.net, skyjgh2000@hotmail.com,
khchong@cs.hongik.ac.kr, samhnoh@hongik.ac.kr

Performance Comparisons of Multi-Process and Multi-Thread based Web Servers on Multiple Operating System Platforms

Inhwan Doh⁰ KyoungJoong Kim GyuHern Jeong Kihun Chong Sam H. Noh
Dept. of Computer Engineering, Computer Science, Information and Computer Engineering, Hongik Univ.

요 약

많은 사용자 영역의 프로그램들이 확장성과 성능 향상을 목적으로 다중 프로세스 모델에서 다중 스레드 모델로 바뀌고 있다. 웹 서버도 폭발적으로 증가하는 사용자 요청을 수용할 수 있도록 확장성과 성능 향상을 위해서 다중 스레드 모델을 적용하였다. 본 논문에서는 다중 프로세스 모델을 적용한 웹 서버와 다중 프로세스/다중 스레드 모델을 적용한 웹 서버에 대한 성능을 여러 운영체제 환경에서 평가하였다. 실험 결과 전체적으로 다중 프로세스 모델이 다중 스레드 모델보다 더 좋은 성능과 안정성을 보였다.

1. 서론

사용자 단계에서 같은 프로세스를 여러 개 생성하고 소멸시키는 다중 프로세스 프로그램에서는 각각의 프로세스들이 독립적으로 가상 주소 공간과 동일한 실행코드, 그리고 데이터 등을 가지고 있다. 그러므로 프로세스의 생성과 소멸, 그리고 특히, 프로세스간의 문맥교환이 이루어질 때 많은 오버헤드를 가지게 된다. 이러한 문제를 해결하기 위해서 스레드는 하나의 프로세스 내에서 실행코드부분, 데이터 부분, 그리고 파일과 시그널을 포함하는 운영체제 자원들을 공유할 수 있도록 하였다. 그러므로 한 프로세스 내에 같은 일을 수행하는 다중 스레드를 사용하게 되면 다중 프로세스에 비해 생성, 소멸, 문맥교환 그리고 실행중인 다른 스레드들 간의 통신에 있어서 오버헤드가 줄어들게 된다[1]. 이러한 스레드의 장점으로 인하여 점점 다중 스레드 방식의 프로그램들이 증가하고 있다.

웹 서버도 성능 향상과 확장성을 제공하기 위해서 다중 프로세스 방식에서 다중 스레드 방식으로 변화하고 있다. 따라서 본 논문에서는 다중 프로세스 모델과 다중 스레드 모델의 성능을 비교하기 위하여 두 가지 모델을 적용한 Apache 웹 서버를 이용하여 성능 측정을 하였다. 그리고 운영체제마다 프로세스와 스레드가 상이하게 구현되어 있으므로 다양한 운영체제상에서 프로세스와 스레드 모델의 성능을 평가하였다.

Windows 2000, FreeBSD, NetBSD, Linux에서 성능 평가를 실시한 결과 운영체제 별로 그리고 프로세스/스레드 모델 별로 많은 유용한 결과를 얻을 수가 있었으며 애초의 기대와는 다르게 대체적으로 프로세스 모델의 성능이 좋은 것으로 나타났다.

논문의 구성은 다음과 같다. 먼저 2절에서는 관련 연구와 Apache 웹 서버의 다중 프로세스/스레드 모델에 대해서 언급한다. 3절에서는 운영체제 별로 프로세스와 스레드 관리를 어떻게 하고 있는지에 대해서 살펴보고, 4절에서는 실험 및 성능 비교가 이어진다. 마지막으로 5절에서는 결론 및 향후 과제에 대하여 논하도록 한다.

2. 관련 연구와 Apache 웹 서버

다중 프로세스와 스레드 모델의 웹 서버 성능을 비교 분석하려는 노력은 이전에도 있었다. Linux 운영체제에서의 성능 분석 결과 다중 프로세스와 다중 스레드가 혼합된 방식의 모델은 정적인 문서만을 서비스할 때 좋은 성능을 보였고 동적 문서를 서비스할 때에는 다중 프로세스 모델이 더 좋은 성능을 보였다는 연구 결과가 있다[2]. 본 논문에서는 다중 프로세스 모델과 다중 스레드 모델로 구현된 웹 서버의 성능 평가를 Linux 운영체제로 한정하지 않고, 다양한 운영체제 환경에서 프로세스와 스레드의 성능을 비교하였다.

Apache 웹 서버는 그 버전에 따라서 다중 프로세스 방식과 혼합형 다중 프로세스/스레드 방식이 있다[3]. Apache 1.3은 다

[†] 이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구 되었음 (KRF-2002-041-D00426).

중 프로세스 모델이며 Prefork 모델을 사용하고 있다. Prefork 모델은 클라이언트로부터 들어온 요청을 처리하기 위한 프로세스를 미리 생성해 놓음으로써 그 성능을 향상시킬 수 있다 [4]. Apache 2.0은 다중 스레드 모델을 지원하는 다중 프로세싱 모듈(MPM)을 가지고 있다. 이 모듈은 MPM-Prefork, MPM-Worker, MPM-Perchild로 구성되어 있다. 이들 세가지 모듈에 대해서 간단히 살펴보면 MPM-Prefork은 Apache 1.3의 Prefork모델과 정확하게 같으며 MPM-Worker은 프로세스의 수가 동적이고 스레드의 수가 고정되어 있는 혼합형 다중 프로세스/스레드 방식을 취하고 있으며 MPM-Perchild는 프로세스의 수가 고정되어 있고 스레드의 수가 가변적인 방식으로 동작한다[5].

본 논문에서는 다중 프로세스 모델로 Apache 1.3을, 다중 프로세스/스레드 모델로 Apache 2.0의 MPM-Worker를 사용하였다.

3. 운영체제 별 프로세스/스레드

기본적인 프로세스와 스레드에 대한 개념은 운영체제마다 크게 다르지 않지만 그 세부적인 구현에 있어서는 다소 차이가 있다. 이 절에서는 운영체제에 따라서 프로세스와 스레드가 가지는 특징에 대해서 살펴보고 그 세부적인 구현에 대해서도 알아보겠다.

3.1 Linux

Linux에서 프로세스는 task_struct라는 구조체로써 표현된다 [1]. 프로세스를 생성하면 이 구조체가 커널에서 생성되고 선형 벡터로써 관리된다. 프로세스가 가지고 있는 속성으로는 가상 메모리공간과 문맥을 구성하는 레지스터와 스택 정보들을 비롯하여 그 외에도 다양한 구성요소들을 가지고 있다. 새로운 프로세스는 현재의 프로세스의 속성들을 새로운 구조체에 복사함으로써 생성된다. 이때 새로운 프로세스는 가상 메모리 공간과 파일들 그리고 시그널 핸들러와 같은 자원들을 현재의 프로세스와 공유하도록 할 수가 있다. 두 프로세스가 같은 가상 메모리 공간을 공유할 때 이들을 같은 프로세스내의 스레드로서 동작한다. 이와 같이 Linux에는 스레드를 위한 자료구조는 존재하지 않기 때문에 사실상 스레드와 프로세스가 구분되어 있지 않다.

3.2 Windows 2000

Windows 2000에서의 프로세스와 스레드는 Linux와는 달리 각각의 오브젝트 구조체를 가지고 있다. 프로세스와 스레드의 생성시에는 프로세스와 스레드 오브젝트 클래스를 사용하여 각각의 프로세스와 스레드 오브젝트를 생성한다. 프로세스와 스레드 오브젝트 클래스는 각각 속성들과 메소드들을 포함하고 있으며 스레드의 속성은 프로세스와 공통적인 것들도 존재한다. 기본적으로 다중 프로세스 환경을 지원하고, 다중 스레드 환경도 지원하고 있다[6]. 프로세스당 적어도 하나의 실행 스레드(primary thread)를 생성하여 일을 수행한다. 그리고 선점

적인 멀티태스킹(preemptive multitasking)을 지원하며 그 단위로써 스레드를 사용한다.

3.3 FreeBSD/ NetBSD (4.4BSD)

4.4BSD는 기본적으로 멀티프로그래밍을 지원하도록 구현되어 있다[7]. 스레드 개념을 적용하기 위해서 4.4BSD는 개발자들에 의해서 다시 구현되었고 프로세스 구조체를 수정함으로써 프로세스가 저장하고 있던 많은 내용들을 공유할 수 있도록 하였다. 프로세스 구조체 역시 그 크기가 예전의 프로세스 구조체에 비해 4분의 1이나 줄어들게 되었다. 다른 운영체제의 스레드와는 달리 4.4BSD의 스레드(variable-weight process)는 공유되는 자원들의 집합을 선택할 수 있도록 설계되었다. 하지만 내부적으로 커널 스레드는 제공하지 않고 있으며 따라서 스레드와 프로세스가 구분되지 않는다.

4. 실험 및 성능 평가

본 논문에서는 다중 프로세스 모델과 다중 스레드 모델에 대하여 다양한 운영체제상에서 성능 평가를 실시하였다. 하나의 운영체제에서만 성능평가를 할 경우 그 운영체제의 프로세스/스레드 구현에 다소 영향을 받을 수 있고 그 결과를 일반화시키기에는 무리가 따르기 때문에 다양한 운영체제상에서 다중 프로세스 모델과 다중 스레드 모델의 성능을 비교하였다.

4.1 실험 환경

실험이 진행된 운영체제는 Linux, FreeBSD, NetBSD, Windows 2000이다. 서버 시스템은 펜티엄3 - 750MHz, 128M Ram의 사양으로 구성하였고, 클라이언트는 펜티엄3 - 1GHz, 256M Ram의 사양을 가진 시스템으로 구성하였다. 만약 서버의 성능이 충분히 좋다면 클라이언트가 보내는 요청들을 지체 없이 처리할 수 있을 것이기 때문에 다중 프로세스 모델이든 다중 스레드 모델이든 그 성능상의 차이는 미미하게 된다. 따라서 서버 시스템의 성능보다 클라이언트 시스템의 성능이 더 좋아야 한다.

웹 서버는 다중 프로세스 모델과 혼합형 다중 프로세스/스레드 모델을 둘 다 지원하는 Apache 웹 서버를 사용하였다. Apache 1.3 버전을 다중 프로세스 모델로, Apache 2.0 버전을 다중 프로세스/스레드 모델로 적용할 수가 있다. 혼합형 다중 프로세스/스레드 모델은 MPM-Worker모듈을 사용하여 구성하였다. 예외적인 것은 Windows 2000용으로 배포된 Apache 1.3 웹 서버는 다른 운영체제의 그것과는 달리 다중 스레드만으로 동작한다.

웹 서버의 벤치마크는 WebStone 2.5를 사용하였다[8]. 작업부하(workload)는 동적 문서와 정적 문서의 비율을 달리하였다. 전체 부하에 대해 동적인 문서의 비율을 100, 70, 30, 0%로 하여 각각 클래스 1, 4, 8, 11로 나누어 실험하였다. 결과는 Apache 1.3과 MPM-worker모델에 대해 각 클래스마다 클라이언트의 수를 50에서 700까지 증가시키면서 세 번씩 실험하여 그

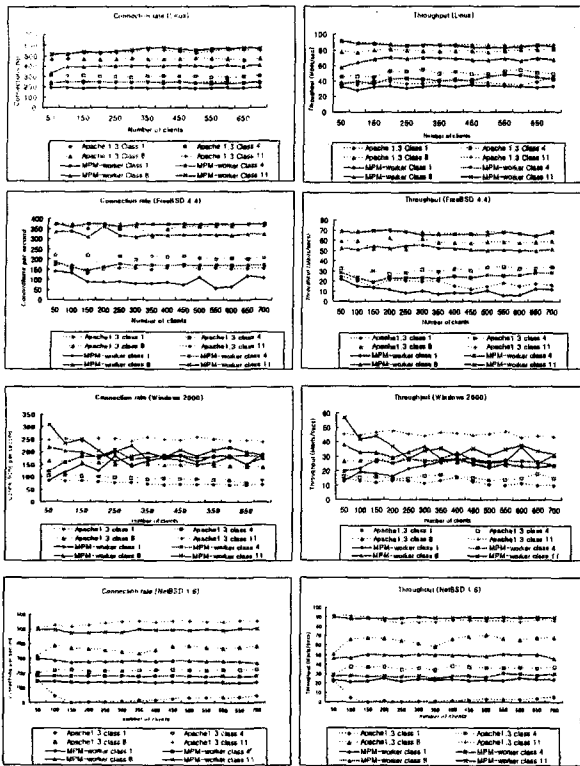


그림 1. 실험 결과. 위에서부터 차례로 Linux, FreeBSD, Windows 2000, NetBSD. 왼쪽은 초당 커넥션 수, 오른쪽은 처리율이다.

평균을 구하였다.

4.2 실험 결과

Linux, FreeBSD, Windows 2000, NetBSD 상에서 작업부하 클래스와 클라이언트의 수를 변화시켜가면서 Apache 1.3과 Apache 2.0(MPM-worker)의 초당 커넥션 수와 처리율을 측정하였다. 그림 1은 그 결과를 그래프로 나타낸 것이다. Windows 2000을 제외한 다른 운영체제에서는 다소 차이가 있지만 클라이언트의 수가 증가함에도 불구하고 커넥션 수와 처리율은 크게 변하지 않는다. 이것은 적은 양의 클라이언트의 요청에 이미 서버가 포화상태에 도달한 것이다. 하지만 고부하의 요청에도 불구하고 일정량의 서비스를 안정적으로 해주는 모습을 볼 수 있다. Windows 2000에서는 Apache 1.3도 다중 스레드 모델로 구성되어 있으나 안정적인 반면 MPM-worker 모델은 상당히 불안정한 모습을 보이고 있다. 운영체제에 따른 성능의 차이는 대체적으로 Linux, NetBSD, FreeBSD, Windows 2000으로 그 순위가 정해진다. 다중 프로세스 모델과 다중 스레드 모델의 성능 차이는 가대했던 것과는 반대로 대체적으로 다중 프로세스 모델의 성능이 더 좋은 것으로 나타났다. Linux와 FreeBSD에서는 정적 문서만을 요청하는 클래스를 제외한 모든 작업 부하 클래스에 대해서 다중 프로세스 모델의 성능이 우수하게 나타났

다. 그리고 NetBSD에서 다중 프로세스 모델은 동적 문서만으로 구성된 요청에 대해서 상당히 낮은 수행 능력을 보이고 있지만 다른 클래스에 대해서는 역시 다중 프로세스 모델이 비슷하거나 더 좋은 성능을 나타낸다. 결론적으로 실험상의 모든 운영체제에서 다중 프로세스 모델은 안정되게 동작하였으며 다중 스레드 모델은 Windows 2000에서 다소 불안정한 모습을 보였다. 또한 작업부하에 정적 문서에 대한 요청이 많이 포함되어 있을수록 다중 스레드 모델의 성능은 좋아지며 그렇지 않고 일반적인 작업부하의 경우에는 다중 프로세스 모델의 성능이 다중 스레드 모델의 성능보다 우수하게 나타났다.

5. 결론

본 논문에서는 다양한 운영체제 상에서 다중 프로세스 모델과 혼합형 다중 프로세스/스레드 모델의 성능을 평가하고 분석하였다. 실험은 두 가지 모델을 적용하기가 쉬운 Apache 웹 서버를 웹 벤치마크 프로그램인 WebStone을 사용해서 진행하였으며 운영체제로는 Linux, FreeBSD, NetBSD, Windows 2000이 사용되었다. 일반적인 작업부하에서 다중 프로세스 모델의 성능이 다중 스레드 모델의 성능보다 우수하며 또한 안정적으로 동작한다. 다중 스레드 방식은 정적 문서의 요청으로만 구성된 작업부하에서는 다중 프로세스 방식보다 더 좋은 성능을 발휘하였으나 그 외의 작업부하에서는 그렇지 못하였다. 따라서 다중 스레드 모델의 성능을 저하시키는 요인을 찾고 개선하는 연구가 필요할 것이며 그러한 연구는 웹 서버의 확장성과 성능을 크게 높여줄 것으로 기대된다.

참고 문헌

- [1] William Stallings, "Operating Systems Internals and Design Principles, 4th edition," Prentice Hall, 2001.
- [2] Kihun Chong, Miryeong Yeom, and Sam H. Noh, "An Assessment of the Apache Web Server 2.0 Performance on Linux," In Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems(SPECTS 2002), 2002.
- [3] The Apache Group, Apache http Server Project, <http://www.apache.org>
- [4] <http://www.ncsa.uiuc.edu/InformationServers/Performance/v1.4/report.html>
- [5] <http://httpd.apache.org/docs-2.0/mpm.html>
- [6] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/multiple_threads.asp
- [7] M. K. McKusick, K. Bostic, M. J. Karels, J. S. Quarterman, "The Design and Implementation of the 4.4BSD Operating System," Addison Wesley, 1996
- [8] Mindcraft, Inc., "WebStone—the Benchmark for Web Servers," <http://www.netcraft.com/survey>