

# 전자상거래를 위한 동적 스케줄링 응용서버 방식의 성능 평가

장철수<sup>\*</sup> 이경호 김성훈 함호상

한국전자통신연구원 시스템통합연구부 전자거래연구팀

## Performance Evaluation of a Dynamic Scheduling Application Server for Electronic Commerce

*ChoulSoo Jang KyeongHo Lee SungHoon Kim HoSang Ham*  
*Electronic Commerce Team,*  
*System Integration Department, ETRI*

### 요 약

월드 와이드 웹(World Wide Web)의 급속한 성장은 인터넷을 통한 전자상거래를 활성화시켰다. 전자상거래 서버는 데이터베이스 통로(Database Gateway) 프로그램을 통하여 데이터베이스에 접속하여 정보를 제공하거나 갱신하여 사용자와 많은 상호 작용을 할 수 있도록 설계된다. 빠른 응답 시간을 요구하는 전자상거래 서비스는 사용자와의 실시간적인 상호 작용을 충족시킬 수 있도록 고성능의 데이터베이스 통로를 요구하고 있다. 본 논문에서는 고성능의 데이터베이스 통로로 활용될 수 있는 새로운 방식의 동적 스케줄링 응용 서버(Dynamic Scheduling Application Server) 방식을 소개하고 이에 대한 성능 평가 결과에 대해 설명한다.

### 1. 서 론

인터넷을 기반으로 하는 전자상거래 서비스에서는 데이터베이스 통로(Database Gateway)를 통하여 데이터베이스 등을 접속하여 상품 정보와 가격 정보 등 필요한 정보를 얻어와서 사용자에게 다양한 정보를 제공하며, 또한 사용자로부터 구매 정보 등을 입력 받아 데이터베이스에 고객의 장바구니 정보와 구매 정보 등을 저장하거나 갱신하는 등 사용자와 많은 상호 작용을 한다. 인터넷을 기반으로 하여 사용자와 상호 작용하기 위해서는 웹 서버와 응용 프로그램이 연동을 할 수 있도록 하여야 한다. 연동 방법 중에서 일반적으로 가장 쉽고 단순한 방법인 CGI(Common Gateway Interface)라는 표준화된 방식을 사용하고 있다. 빠른 응답 시간을 요구하는 전자상거래 서비스는 사용자의 서비스 만족도를 충족시킬 수 있도록 고성능의 데이터베이스 통로를 요구하고 있다. 그러나, CGI 방식을 이용한 데이터베이스 통로는 시스템 자원을 많이 사용하여 효율을 저하시키며, 데이터베이스에 초기 접속하는 동안 많은 시간을 소모하여 전체적인 수행속도가 늦어 고객의 서비스 요구에 대해 즉각적인 응답을 할 수 없게 되어 대용량의 전자상거래에서는 적당하지 못하다. 이러한 CGI 방식의 수행속도 문제를 개선하기 위해서 전자상거래 서비스에서는 웹 서버 제작자가 제공하는 API(Application Interface) 방식을 사용하거나, 응용 프로그램을 데몬(daemon)화 시키는 응용서버(Application Server) 방식 등을 사용하여 데이터베이스 통로 프로그램을 웹 서버와 연동시킨다.

그러나, API 방식은 웹 서버와 밀접하게 접촉되어 있어,

응용프로그램의 오류로 인하여 웹 서버 자체가 멈추는 경우가 발생할 수 있는 단점을 갖고 있다. 그리고, 응용서버 방식은 미리 고객의 서비스 요구 수를 예측하여 적절히 필요할 수 만큼 데몬화 시킬 프로그램을 초기에 프로세스로 생성하여 놓는다. 그러나, 고객의 서비스 요구 수는 미리 예측하기가 상당히 어렵기 때문에 대기 시간의 증가나, 시스템 자원의 낭비가 있을 수 있다.

본 논문에서는, 이러한 응용 서버 방식의 단점을 해결할 수 있도록, 고객의 서비스 요구 수가 증가하면 더 많은 프로세스가 동적으로 생성되어 고객의 서비스 요구를 처리하게 하고, 고객의 서비스 요구 수가 감소하는 경우에는 시스템 자원의 낭비를 막을 수 있도록 여분의 프로세스가 소멸할 수 있는 동적 스케줄링 응용 서버(Dynamic Scheduling Application Server) 방식을 제안한다. 또한 제안된 방식과 기존의 정적 스케줄링하는 응용 서버 방식과의 성능 비교 결과를 설명한다.

이 논문의 구성은 다음과 같다. 2장에서는 CGI와 응용서버 방식에 대해 알아보고, 3장에서는 제안된 동적 스케줄링 응용 서버 방식의 구조에 대해 알아본다. 그리고 4장에서는 제안된 구조의 성능 평가 결과를 살펴본 후, 마지막에 5장에서 결론과 향후 연구 과제를 알아본다.

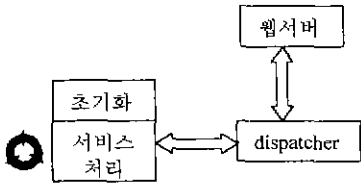
### 2. CGI 방식과 응용서버 방식

가장 일반적으로 사용되는 CGI 방식의 데이터베이스 통로 프로그램은 고객의 요구가 있을 때마다 웹 서버에 의해

호출되어 시스템 디스크로부터 응용 프로그램이 주 메모리(main memory)로 적재(load)되어 프로세스가 생성되고, 시간이 많이 소모되는 데이터베이스 초기화 과정을 처리한 후, 고객의 서비스 요구에 대한 처리를 단 한번 수행하고 종료된다. 그러나 시간이 많이 소모되는 데이터베이스 초기화 과정이 단 한번의 데이터베이스 서비스 처리를 한 후 소멸되어, 새로운 서비스 요구가 있을 때마다 다시 초기화 과정을 수행하여야 하므로 수행 속도가 매우 느리다.

웹 서버에 연동된 응용 프로그램의 성능을 향상시키는 방법 중 가장 널리 사용되는 방식은 fastCGI와 같은 응용 서버 방식이다. 응용서버 방식에서는, 응용 프로그램이 데이터베이스에 초기에 접속되는 것과 같이 수행시간이 많이 요구되는 초기화 과정을 응용 프로그램 시작시에 한번만 수행하게 하고, 나머지 서비스 처리를 하는 부분은 계속 데몬(daemon)으로 남아 고객의 서비스 요구가 있을 시 즉시 처리하게 한다.

웹 서버와 응용 프로그램의 사이는 <그림 1>과 같이 웹 서버가 디스패처(dispatcher)와 같은 작은 소켓 프로그램을 호출하여 응용서버의 포트(port)와 연결할 수 있으며, 또한 웹 서버의 제작자가 제공하는 API를 이용해 fastCGI와 같이 웹 서버를 수정하여 응용 서버의 포트와 연결할 수 있도록 할 수 있다.



<그림 1> 응용 서버 방식의 구조

응용서버 방식에서는 미리 고객의 서비스 요구 수를 예측하여 적절히 필요한 수만큼 데몬화 시킬 프로그램을 초기에 프로세스로 생성하여 놓는다. 그러나, 고객의 서비스 요구 수는 미리 예측하기가 상당히 어려우며, 특히, 고객의 서비스 요구 수가 갑자기 급증하는 경우에는 고객이 요구를 보낸 후 요구처리를 대기하는 시간이 늘어나게 된다. 또한, 원활한 서비스 요구 처리를 위해 충분히 많은 수 만큼 프로세스를 생성하여 놓는다면, 고객이 서비스를 요구하지 않는 한가한 시간에도 프로세스들이 주 메모리에 적재되어 있어야 하는 등 시스템 자원을 많이 낭비하게 된다.

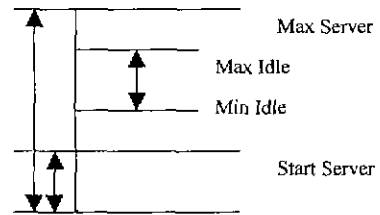
### 3. 동적 스케줄링 서버 방식의 구조

본 논문에서는, 이러한 응용 서버 방식의 단점을 해결할 수 있도록, 고객의 서비스 요구 수가 증가하면 더 많은 프로세스가 동적으로 생성되어 고객의 서비스 요구를 처리하게 하고, 고객의 서비스 요구 수가 감소하는 경우에는 시스템 자원의 낭비를 막을 수 있도록 여분의 프로세스가 소멸되는 동적 스케줄링 응용 서버(Dynamic Scheduling Application Server) 방식을 제안한다.

동적 스케줄링 응용 서버 방식에서는 프로세스 수행 초기에 데몬 관리자(Daemon Manager)가 <그림 2>와 같이 지정된

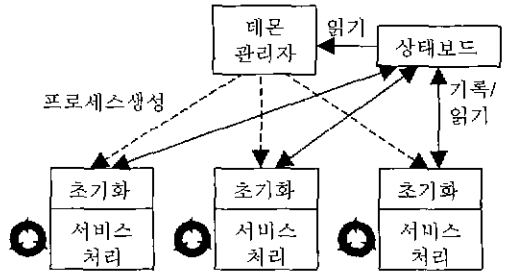
수(Start Server Number) 만큼 프로세스를 생성시키고, 생성된 각 프로세스는 데이터베이스에 접속하는 초기화 과정을 마친 후 서비스 처리를 위한 데몬으로 메모리에 상주하고 있어 고객의 서비스 요구가 있을 때에 데몬이 즉시 서비스 요구를 처리한다.

데몬 관리자는 전체 프로세스의 상태를 주기적으로 점검하여, 고객의 요구 수가 증가하여 한가한 상태(idle state)의 프로세스 수가 지정된 수(Min Idle Number) 보다 모자라는 경우에 추가로 프로세스를 생성한다. 또한 각각의 데몬 프로세스는 주기적으로 전체 프로세스의 상태를 점검하여 고객의 요구 수가 감소한 경우 한가한 상태에 있는 프로세스가 지정된 수(Max Idle Number)보다 큰 것이 확인된 경우에는 스스로 소멸하여 불필요한 시스템 자원의 낭비를 막는다.



<그림 2> 동적 스케줄링 응용 서버 방식의 파라미터

프로세스 상태를 주기적으로 점검하기 위해서 동적 스케줄링 응용 서버 방식에서는 <그림 3>과 같이 공유 메모리를 이용한 상태 보드(Status Board)를 관리한다. 각 프로세스는 자신의 현재 상태를 상태 보드에 기록하고 데몬 관리자는 기록된 상태 보드를 주기적으로 상태 보드를 점검한다.



<그림 3> 동적 스케줄링 응용 서버 방식의 구조

서비스 처리를 하는 각 프로세스는 디스패처나, 웹서버의 API를 이용하여 연결하는 소켓 클라이언트에게 서비스를 제공하기 위해 소켓 서버로 동작된다.

### 4. 동적 스케줄링 서버 방식의 성능 평가

동적 스케줄링 응용 서버 방식의 성능을 평가하기 위하여 웹 클라이언트는 웹벤치(WebBench)라는 셰어웨어(shareware) 프로그램을 Windows NT에서 사용하였다. 웹벤치를 이용하여 50개의 스레드(thread)가 서버에게 서비스 요구를 1분동안 동시에 계속 보내는 방법을 이용하였다. 또한

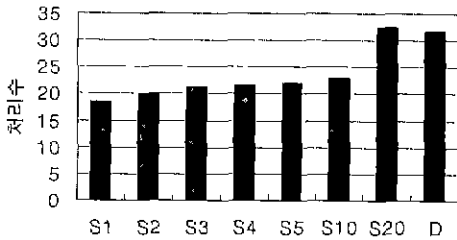
일반 CGI 방식과 응용 서버 방식은 성능 차이가 워낙 커서 일반 CGI 방식의 성능 평가는 제외시켰고, 일반 정적 스케줄링 응용 서버 방식과 동적 스케줄링 응용 서버의 성능 평가를 하여 결과를 비교하였으며, 성능 평가 환경은 다음과 같다.

- 서버측
  - Hardware : Hewlett Packard Visualize C160 (128M)
  - Operating System : HPUX 10.20
  - 웹서버 : Apache 1.3
- 클라이언트측
  - Hardware : Intel Pentium II (333 MHz, 128M)
  - Operating System : Windows NT 4.0
  - 웹클라이언트 : WebBench 2.0

성능 평가에 사용된 동적 스케줄링 응용 서버 방식의 파라미터는 다음과 같다

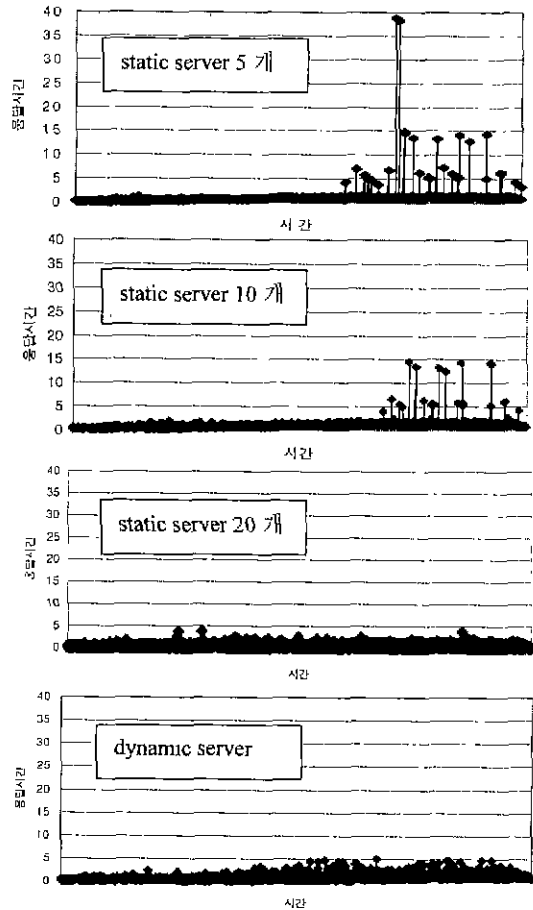
- Start Server Number : 5
- Min Idle : 3
- Max Idle : 8
- Max Server Number : 20

성능 평가 결과는 <그림 4>와 같이 정적 응용 서버 방식에서는 프로세스 수가 5개인 경우에 22개, 10개인 경우 23개, 20개인 경우 32개의 서비스처리를 하고 있어 프로세스 수가 늘어날수록 단위시간당 보다 많은 서비스 처리를 할 수 있음을 알 수 있다 반면에 동적 스케줄링 응용 서버 방식에서는 서비스 요구의 증감에 따라 프로세스 수가 조절되면서 약 31개의 서비스 처리를 하고 있다. 동적 스케줄링 응용서버 방식이 20개의 정적 응용 서버보다 요구 처리 수가 적은 것은 동적 스케줄링 응용서버 방식이 데몬 관리자에 의해 프로세스를 관리하는 추가의 오버헤드에 의한 것으로써 그 차가 극히 작다.



<그림 4> 초당 처리된 요구 수

평균 요구 처리 수 만큼 중요한 것이 평균 응답 시간이다. 각각의 평균 응답 시간은 <그림 5>와 같다. 결과에서 보는 바와 같이 정적 응용 서버의 경우에 각각 5개, 10개의 프로세스인 경우에는 10초 이상의 응답속도가 많이 발생하고 있다. 특히 5개의 프로세스인 경우에 40초에 가까운 응답시간이 나타나기도 하였다 이는 고객이 요구처리를 기다릴 수 없는 상태이다. 20개의 정적 응용 서버 프로세스인 경우에는 적절한 응답시간의 결과를 얻었으며, 동적 스케줄링 응용 서버 방식에서도 적절한 응답시간의 결과를 얻었다.



<그림 5> 응용 서버 방식의 응답시간

### 5. 결론 및 향후 연구과제

이와 같은 평가 결과를 분석하여 보면, 새로 제안된 동적 스케줄링 응용서버 방식은 고객의 요구 수에 적절히 대응하여 프로세스 수를 조절해 시스템 자원을 효율적으로 활용하면서도 충분히 좋은 성능을 낼 수 있음을 알 수 있었다

현재는 응용서버에 연결되는 방식이 디스패처를 이용하여 하고 있으나 향후에는 fastCGI와 같이 웹서버를 수정하여 응용서버에 바로 연결하게 하여 성능을 보다 향상시키고자 한다.

### 참고문헌

- [1] 김평철, "UniWeb - 웹을 위한 UniSQL/X 데이터베이스 톨로", 데이터베이스 저널, 제 3 권, 1996, pp.65-84
- [2] 장철수의 3인, "Apache 서버의 확장 기술을 이용한 서버측 캐쉬 모듈의 설계", 한국정보과학회 추계학술대회, 1997
- [3] 조현성의 4인, "전자상거래를 위한 고성능 Web 서버에 관한 연구", 한국 CALS/EC 학회 학술대회, 1997