

분산 상호 배제 카운트 알고리즘을 이용한 클라이언트 사용자 구분 시스템 개발

장 승 주[†]

요 약

본 논문에서는 분산 상호 배제 알고리즘을 이용한 배타적인 카운트 값의 유효성으로 분산 시스템 서버를 이용하는 사용자들을 구분한다. 본 논문의 개발 내용은 분산 시스템 환경으로 많이 이용되고 있는 인터넷 시스템에 이용가능하다. 특히 요즘 많이 사용되고 있는 전자 상거래 시스템 등에 적용가능하다. 사용자들이 전자 상거래 시스템 등에 회원으로 등록을 해야만 사용할 수 있는 기존의 환경에서 벗어나도록 함으로써 기존 전자 상거래 시스템 사용 형태의 불편함을 해소시켜 준다. 분산 상호 배제 알고리즘을 이용한 카운트 값의 유효성은 서버를 이용하는 각 사용자들의 구분자로 사용된다. 이러한 부분들은 실제 분산 시스템 환경에서 설계 및 구현되었다. 또한 실제 인터넷 환경에서 설계 및 구현된 내용을 실험하였다. 실험 결과 각 사용자에게 대한 배타적인 카운트 값이 할당됨을 확인할 수 있었다. 본 논문에서 개발한 배타적인 카운트 값을 이용한 분산 시스템을 구축할 경우 모든 분산 시스템 서버에 사용자들이 자신의 정보를 등록해야 하는 불편함을 해소할 수 있다. 따라서 인터넷 환경에 보다 많은 사람들이 편리하게 접근할 수 있는 여건을 제공할 수 있다.

Design and Implementation of Distributed Mutual Exclusion Lock Counter Algorithm

Seung-Ju Jang[†]

ABSTRACT

In this paper, we propose new concepts that the distributed counter value with the distributed EC system identify each user who uses the distributed system. The web user should register his/her own user ID in the cyber shopping mall system. Instead of registration, this paper proposes the proprietary mechanism that is distributing counter. The counter assigns the distinguished number to each client. The distributed lock algorithm is used for mutual assignment of the counter to each client. The proposed algorithm is the best solution in the distributed environment system such as cyber shopping mall. If a user should register his/her own ID in every EC system, he/she may not try to use these uncomfortable systems. The mutual counter is used to identify each client. All of these features are designed and implemented on Windows NT web server. Also these features were experimented with 5 clients for 300 times. According to the experiments, clients have their own mutual counter value. The proposed algorithm will be more efficient in internet application environment. Moreover, it will improve the number of internet users.

1. 서 론

인터넷의 발달은 산업혁명 이상의 신 산업혁명이라

고 할 정도로 인류의 삶의 질 향상에 기여하고 있다. 인터넷의 발달로 우리가 접하게 되는 많은 일상 생활의 부분들이 웹이라는 부분에 통합 흡수되고 있다. 이러한 환경으로 인터넷 상거래, 디지털 도서관(digital library), 디지털 신문, 디지털 방송 등과 같은 많은 부

[†] 정 회 원 동의대학교 컴퓨터공학과 교수
논문집수, 1999년 9월 15일, 심사완료, 2000년 3월 24일

분들이 해당된다. 이러한 인터넷 환경은 분산 시스템 환경이 가능하기 때문이다. 인터넷의 발달은 분산 시스템 기술의 발달 때문에 가능했다고 할 수 있다.

이러한 많은 웹 사이트 응용 환경들은 대부분 공통점을 가지고 있다. 단지 사용자가 이용하는 응용 환경이 서로 다르다는 것 외에 기능 및 기술적인 차이가 별로 없다. 이러한 인터넷 웹 환경의 급속한 확산은 보다 편리한 환경을 요구하게 되었고 사용자들은 완벽한 해법을 요구하고 있다[4]

일반적인 인터넷 웹 페이지 작성 도구로 많이 사용하고 있는 것이 HTML(Hyper Text Markup Language) 형식이다 그러나 HTML은 제공되는 기능의 한계와 기존 기능과의 접목등에 있어서 많은 문제점을 안고 있다. 따라서 여러 가지 대체 기능들이 나타나게 되었다. 이러한 대체 기능으로 Dynamic HTML, 기존의 프로그램 환경과 접목할 수 있도록 하는 CGI(Common Gateway Interface), 데이터 베이스와 접목될 수 있도록 하는 ASP(Active Server Page), ADO(ActiveX Data Object) 등이 있다[9, 10].

분산 시스템의 응용 형태로 널리 사용되고 있는 것이 전자 상거래 시스템이다. 전자 상거래 시스템이 활성화되면서 많은 사이트들이 우후죽순처럼 생겨나고 있다. 그러나 이러한 전자 상거래 시스템들은 한결같은 특징들을 가지고 있다. 사용자들이 반드시 회원으로 등록을 해야 한다는 것이다 그러나 사용자들에게는 전자 상거래 시스템을 이용한 물품 구매를 하면서 한 사이트에 모든 원하는 물품을 찾기란 쉽지 않다. 따라서 여러 전자 상거래 서버에 접속해야 하는데 모든 전자 상거래 서버들에 대해서 사용자들이 회원으로 등록을 해야 한다는 것은 너무나 불편하다 이런 측면에서 분산 시스템 응용 형태로 많이 사용되고 있는 전자상거래 시스템들이 대부분 회원제 형태로 운용되고 있다는 것은 전자 상거래를 활성화 시키는데 걸림돌이 되고 있다.

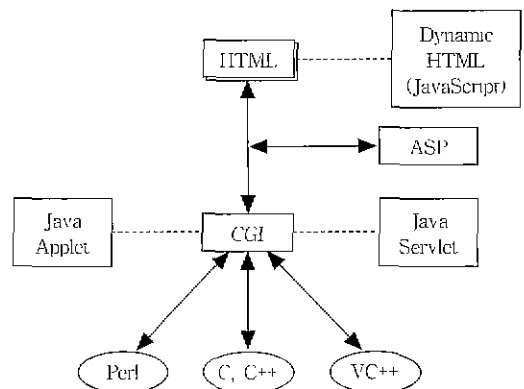
본 논문은 분산 시스템 응용 형태[5]로 많이 이용되고 있는 전자 상거래 서버 사용에 따른 이와 같은 문제를 해결하기 위한 방안과 구현 내용에 대해서 제안한다. 본 논문은 기존의 전자 상거래와 같이 최상위 인터페이스로 HTML을 사용한다[6, 7]. 그리고 데이터 베이스와 연동을 위하여 ASP를 사용한다[1, 13, 14, 15]. 사용자가 회원제 형태로 등록이 되지 않더라도 회원과 동일한 권한을 부여하도록 하는 기능을 갖고 있다. 이

러한 기능을 위하여 사용자는 주민등록번호 등의 필수 기본 정보를 입력해야 한다. 이 정보를 이용하여 회원과 같은 효력을 가지고 물품 구매가 가능하도록 하였다 그리고 이 정보와 연동해서 사용할 수 있는 카운트 기능을 이용하여 사용자들을 구분하는 구분자로 사용한다. 카운트 정보를 각 사용자마다 구분되는 정보로 할당하기 위하여 상호 배타적으로 이 값을 관리해야 한다 상호 배타적으로 카운트 값을 관리하기 위하여 분산 상호 배제 알고리즘[12, 13]을 이용한다 또한 실제 이러한 전자 상거래 서버 구축을 통하여 분산 상호 배제 알고리즘을 이용해서 할당된 카운트 값이 사용자를 구별할 수 있는 구분자로 이용 가능한지에 대한 실험을 했다

본 논문의 구성은 2장에서 관련 연구를 언급하고 3장에서는 카운트를 이용한 사용자 구분, 4장에서는 타당성 검토, 5장은 실험 및 평가, 6장 결론의 순으로 언급한다

2. 관련연구

일반적인 웹 페이지의 구성은 HTML 인터페이스 구조와 이 구조를 이용한 CGI 인터페이스로 구성되어 있다[14] 최근 들어서 ASP를 이용한 웹 페이지 구성이 늘어나고 있다. 전자 상거래 시스템들은 대부분 이와 같은 구조를 갖는다 전자 상거래 시스템을 구축할 수 있는 방법은 다음과 같은 구조에 의하여 가능하다



(그림 1) 전자 상거래 시스템 구축 방법

전자 상거래 시스템의 기술적인 특징은 여러 가지로

요약할 수 있다. 일반적으로 가장 문제가 되는 부분이 사용자가 쉽게 사용할 수 있는 환경을 구축하는 문제와 사용자의 개인 신상 정보에 대한 유출 방지에 대한 대책이다. 전자 상거래 보안 문제는 보안 프로토콜을 이용하여 해결하는 방법들이 제시되고 있다[2] 또한 최근에는 Ipv6(Internet Protocol Version 6)프로토콜에 통합시키는 것에 대한 논의가 있다[9]. 사용자가 쉽게 사용할 수 있는 전자 상거래 시스템 환경의 제공은 중요하다. 그런데 현재 사용되고 있는 전자 상거래 시스템의 80% 이상이 회원제나 회원등록 등의 절차를 거쳐야 만이 상거래 시스템을 사용할 수 있도록 되어 있다 이러한 폐쇄 시스템 형태로 나아가는 것은 사용자로 하여금 전자 상거래 시스템 사용에 대한 거부감을 불러 일으킨다는 것이다. 이러한 메카니즘에 대한 문제점은 기존의 상거래 시스템과 전자 상거래 시스템을 비교하면 쉽게 도출 가능하다[2,3].

우리가 기존 상거래에서 신분을 드러내놓고 상거래를 하는 경우는 없다. 마찬가지로 전자 상거래도 기존의 상거래와 똑같은 환경에다 편리성과 시간을 절약할 수 있다는 장점을 최대한 보장해 주어야 한다. 기존의 상거래와 같이 신분 노출을 극소화하면서 자유롭게 상거래를 할 수 있는 환경 구성이 중요하다. 그리고 회원제는 사용자들이 원할 경우 등록 절차를 거쳐 상거래 시스템을 이용하도록 하고 회원제 등록을 원하지 않을 경우는 회원 ID가 없이 임시 구분자 정보를 이용한 상거래가 이루어 질 수 있도록 한다. 현재 전자 상거래 시스템은 대부분이 개인의 신상 정보를 등록하고 이 회원 정보를 이용하여 상거래가 이루어지도록 되어 있다. 이러한 환경은 전자 상거래 시스템 사용을 제약하는 요인 중 하나이다. 그리고 이와 같이 회원제로 전자 상거래 시스템을 유지할 경우 사용자는 특정한 시스템에 한정하여 전자 상거래를 해야 하는 불편함이 있고 모든 전자 상거래 시스템에 대한 회원 정보를 관리하는 것이 개인적으로 한계가 있다

본 논문은 이러한 문제점을 해결하기 위한 방안으로 회원으로 등록되지 않은 상거래 사용자들도 편리하게 상거래 시스템을 이용할 수 있도록 하기 위한 임시 구분자 정보를 이용한다. 그리고 이러한 임시 구분자 정보는 각 사용자를 구분할 수 있는 중요한 정보로 이용된다.

3. 분산 상호 배제 카운트를 이용한 사용자 구분

3.1 시스템 구조

전자상거래를 운영하기 위해서 고려해야 하는 가장 중요한 항목으로는 보안과 사용자의 편리성이다. 이 두 가지 기능은 전자 상거래 시스템 구성에서 없어서는 안되는 중요한 기능이다 보안 기능은 사용자들의 개인 신상 정보를 이용하여 상거래를 하는 개인 사용자 신상 정보가 유출되지 않고 안전하게 관리될 수 있도록 보장하는 것이다 사용자의 편리성은 전자상거래를 하는데 있어서 불편함이 없이 쉽게 물품 구매를 할 수 있는 인터페이스를 제공하는 것이다

전자상거래는 고객이 인터넷을 통하여 물품을 구매한다. 인터넷을 통한 물품 구매는 물건을 직접 만져 보지는 못하지만 사진과 제공되는 설명을 읽어 보고 원하는 상품일 경우 신용카드 번호와 같은 지불 정보를 입력함으로써 소비자의 구매행위는 끝나게 된다 이 과정에서 소비자가 입력한 신용카드와 같은 지불정보는 인터넷을 통하여 판매자의 서버에 저장된다. 소비자가 입력한 정보는 인터넷을 통하여 판매자에게 전달되는 과정에서 악의 있는 해커에 의해 소비자의 지불정보가 누출되고 누출된 정보를 악용하여 소비자와 판매자 양쪽에 피해를 주게 된다. 이러한 문제점을 해결하기 위해서 사용자로부터 판매자에게 지불정보가 전달되는 과정에서 SSL(Secure Socket Layer)보안 프로토콜을 사용하여 지불정보가 해커에 의해 누출되어도 정보자체의 해독이 불가능하게 함으로써 소비자와 판매자의 피해를 막을 수 있다. 그리고 지불정보 이외의 내용에 대한 보안 처리를 하지 않음으로서 불필요한 처리를 하는데 따른 소모 시간을 단축하도록 하였다

전자 상거래 시스템에 대한 사용자의 편리한 접근이란 측면은 여러 가지로 고려할 수 있지만 현재 운용되고 있는 전자 상거래의 문제점 측면에서 살펴보면 전자 상거래 운용 시스템의 대부분이 사용자들에게 회원으로 등록할 것을 강요하고 있다는 것이다. 운용자 측면에서 사용자들의 회원제 등록은 편리한 사용자 관리와 데이터 베이스의 구축이란 측면에서 좋은 장점을 가지고 있지만 전자 상거래를 이용하는 사용자 입장에서 보면 모든 전자 상거래 시스템을 사용하려면 모든 시스템에 사용자 등록을 해야 하는 불편함이 있다. 현재 전자 상거래 시스템이 이와 같이 회원제로 운용할

수 밖에 없는 이유는 HTML이나 CGI등 전자 상거래 시스템의 구축 환경이 회원제로 등록하여 만드는 것이 쉽게 되어 있고 전자 상거래시스템을 구축하는 패키지들이 회원제로 전자 상거래 구매 고객들의 정보를 관리하도록 만들어져 있기 때문이다.

본 논문은 이와 같은 문제점을 해결하기 위하여 사용자들이 웹 페이지에 접속하게 되면 회원이 아닌 사용자라도 고유의 임시 ID를 발급하여 각 사용자들을 구분할 수 있도록 하고 구매 등의 자유로운 활동이 가능하도록 하는 방법을 제시 및 구현하였다. 발급된 고유의 ID를 이용하여 회원과 동일한 권한을 부여한다 각 사용자 마다 고유의 ID는 상호 배타적으로 할당되어도록 한다. 사용자 마다 고유의 ID는 본 논문에서는 분산 상호 배제 카운트라고 이름 붙인다.

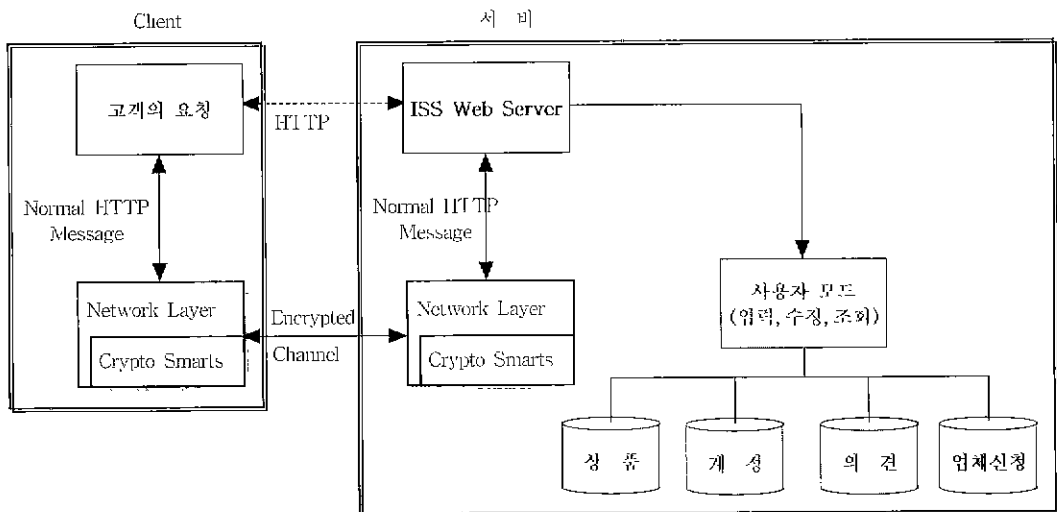
본 논문에서 제안하는 분산 공유 카운트를 이용한 전자 상거래 서버는 고객이 회원으로 등록하지 않더라도 "손님" 권한으로 물품 구매하는 기능을 갖도록 한다. 손님 권한으로 물품을 구매하기 위해서 서버에 접속한 손님에 대하여 개인의 ID를 부여해야 한다. 손님의 ID를 부여하기 위해서 고객이 서버에 접속시 유일한 구분자로 사용되는 카운트 값을 손님의 고유 ID로 할당한다. 이 카운트 값은 "손님"들 사이에 동일한 값을 갖지 않도록 상호배타적으로 할당한다. 각 사용자들에 대해서 임시로 할당된 구분자 ID로 물품 구매가 발생할 경우 각 사용자 마다의 장바구니를 구분하

는 역할을 한다. 이때 다른 주민 등록 번호 같은 정보는 기본 정보로 구분자 ID와 함께 저장된다. 임시 테이블에 지정된 손님의 구매 정보는 물품 구매를 완료 시 실제 주문 테이블로 저장되며 저장된 정보는 전자 상거래 운영자에 의해 확인될 수 있다.

다음은 고객이 인터넷을 통하여 전자 상거래 서버에 접속 후 동작형태는 (그림 2)와 같이 나타낼 수 있다.

고객은 전자 상거래 서버에 접속하여 구매 물품 정보 및 의견 등을 입력한다. 이 과정에서는 일반 HTTP 프로토콜을 사용하지만 물품 구매를 하기 위해 물품에 대한 지분 정보를 입력한 경우 SSL보안 프로토콜을 사용하여 보안 과정을 거친다.

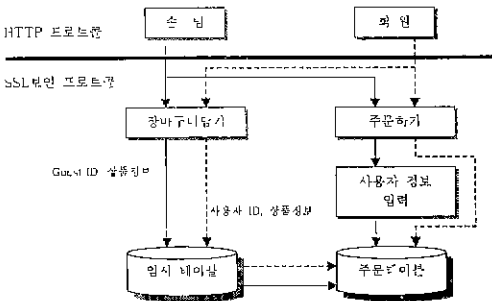
일반적으로 고객은 웹 서버에 어떤 상품, 어떤 서비스가 있는지 궁금해 한다. 이러한 고객의 궁금증을 해결하기 위해 물품 검색 기능이 있다. 물품 검색 기능은 웹 서버에 있는 모든 데이터를 데이터 베이스화 함으로써 상품코드, 상품명 또는 부분적인 상품명, 종류, 물품의 특징으로 검색을 할 수 있다. 이러한 검색 기능은 고객의 사용에 편리성을 제공한다. 그리고 고객은 자신이 입력한 물품 구매에 대한 정보를 얻고자 한다. 이러한 고객의 요청을 만족시키기 위해서 자료 검색 기능을 구현한다. 자료 검색은 회원일 경우 ID를 통한 검색과 비회원일 경우는 앞에서 설명한 고유의 카운트 값을 이용하여 수행이 이루어진다.



(그림 2) 전자 상거래 서버의 동작 과정

3.2 제안된 시스템의 동작 과정

고객이 처음 전자 상거래 서버에 접속하면 자동으로 할당되는 카운터 값을 이용하여 각 개인에 대한 고유 ID 값으로 사용한다. 물론 고객이 회원일 경우는 로그인 과정을 통하여 실제 ID를 이용하여 모든 전자 상거래 시스템을 이용할 수 있다. 그러나 전자 상거래 서버에 처음 접속한 손님일 경우 사용자 등록 과정(회원 등록)이 없이 "손님"으로 물품 구매가 가능하도록 하였다. "손님"으로 물품을 구매하기 위한 과정을 살펴보면 다음 (그림 3)과 같다.



(그림 3) "손님"으로 접속한 후 동작 과정

"손님"으로 전자 상거래 서버에 접속하게 되면 서버에서 자동으로 사용자 ID와 같은 권한에 해당하는 카운트를 할당하게 된다. "손님"이 물품을 구매하기 위해 물품에 대한 정보와 설명을 읽고 자신이 원하는 상품을 선택하고 자신이 할당 받은 고유한 ID 값을 구분자로 하여 장바구니 담기 기능 등을 이용하게 된다. 사용자를 구분하는 구분자로 사용되는 카운터 값은 손님이 장바구니 담기 기능을 이용하여 물건을 구매할 때 임시로 마련된 테이블에 선택된 물품정보와 함께 저장된다. 장바구니 담기를 마친 경우 주문하기를 누르면 사용자 신상 정보를 기본적인 정보로 입력하게 되어있다. 사용자의 기본 정보로 입력된 정보 중 주문등록 번호가 구분자와 연동되어 임시 테이블의 상품 정보와 함께 주문 데이터베이스에 저장된다. 주문 데이터베이스에 저장된 정보는 고객에 의해 수정, 삭제가 가능하다.

3.3 시스템 구축

"손님"을 이용하여 물품 주문을 하기 위해 가장 중요한 개념은 각 사용자 마다 고유의 카운터 값을 할당

하고 할당된 카운터 값을 이용하여 어떻게 사용자들을 구분하느냐 하는 것이다. 카운터 값을 발생시키기 위해서 global.asa 파일을 이용한다. "손님"이 물품 구매를 하는 경우에 카운터 값을 이용하여 장바구니 담기 등과 같은 사용자를 구분할 필요가 있는 부분에 대해서 임시 테이블에 저장된 사용자의 고유 카운터 값을 이용하게 된다.

3.3.1 카운터 값을 이용한 사용자 구분

Global.asa 파일은 사용자가 이벤트 스크립터를 지정하고 세션 또는 응용 프로그램 영역을 갖는 개체를 신택할 수 있는 파일이다. 사용자에게 표시되는 목차 파일이 아니라 이벤트 정보와 응용 프로그램이 전역적으로 사용하는 개체를 저장한다. 이 파일은 응용 프로그램의 root 위치에 저장되며, 하나의 응용 프로그램은 하나의 Global.asa 파일을 가진다. Global.asa 파일은 다음과 같은 항목을 포함한다.

- ✓ Application 이벤트
- ✓ Session 이벤트

Global.asa 파일은 스크립트로 작성되며 다수의 이벤트 스크립트나 개체 스크립트가 동일한 스크립트 언어로 사용되면 하나의 단일 <SCRIPT> 태그 집합 내부에 묶을 수 있다. 단일 <SCRIPT> 태그를 갖지 않는 세션 또는 응용 프로그램 영역을 갖지 않는 개체를 정의하면 서버는 오류를 반환한다. 서버는 응용 프로그램이나 세션 이벤트가 사용하지 않는 태그로 묶인 스크립트를 무시하고 파일에 들어있는 모든 HTML을 무시한다.

사용자 요청이 발생하면 응용 프로그램과 세션을 만들고 Application_OnStart 이벤트와 Session_OnStart 이벤트를 호출한다. 그리고 현재 사용자 요청을 모두 처리하고 난 뒤에 서버는 모든 활성화된 세션을 삭제하고 삭제된 세션에 대하여 Session_OnEnd 이벤트를 호출하여 응용 프로그램을 닫고 Application_OnEnd 이벤트를 호출한다.

사용자가 전자 상거래 서버에 처음 접속하였을 때 자동적으로 고유의 카운터가 할당된다. 카운터 값을 증가시킨 경우는 분산 상호 배제 매커니즘을 이용한다. 분산 상호 배제 매커니즘은 일반적인 록 알고리즘과 같은 기능을 한다. 카운터에 대한 처리는 global.asa의

세션 영역 변수에 사용자의 카운티 값을 할당하였다. 세션 영역 변수는 페이지를 이동할 때 변경되지 않는 특성을 이용하여 손님의 물품 구매 기능을 구현할 수 있다. 다음의 프로그램 코드 (그림 4), (그림 5)는 카운트에 대한 사용자 상호 배제를 보장하기 위한 프로그램 코드 부분을 보여주고 있다

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
    Application("PageSize") = 10

    Visit_Count_F = Server.MapPath("/") & "visitors.txt"
    'visitors.txt은 Count가 지정되어 있는 파일명
    (현재 즉, global.asax와 동일한 디렉토리에 visitors.txt가 있는 것이다)

    Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
    'FileSystemObject는 TextStream을 리턴한다 FileObject 개체 생성
    Set Out= FileObject.OpenTextFile(Visit_Count_F, 1, FALSE, FALSE)
    'Visit_Count_F 변수에 지정되어 있는 TEXT-파일을 OPEN한다.
    Application.Lock
    Application("visitors") = Out.ReadLine
    'OPEN된 파일에서 첫줄을 읽어 올 수 있는 응용 프로그램 변수에 저장한다

    Application("Visit_Count_F") = Visit_Count_F
    Application.Unlock
    'TEXT의 경로가 지정한 Visit_Count_F을 어플리케이션 변수에 저장한다
END SUB
</SCRIPT>
```

(그림 4) Application 시작을 위한 프로그램

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
    ' chatting 시작
    Session("IsMember") = "NO"
    Session("AccessError") = 0
    Session("UserName") = ""

    ' counting 끝
    Session.Timeout = 60 * 24
    '세션의 Timeout 시간을 설정할 때는 분으로 하루를 설정
    Session("ConnectionString") = "dsn=space;uid=sa;pwd="
    Application.Lock
    Application("visitors") = Application("visitors") + 1
    t_visitors = Application("visitors")
    Session("VisitorID") = t_visitors
    If t_visitors MOD 15 = 0 Then
        SET FileObject = Server.CreateObject("Scripting.FileSystemObject")
        Set Out= FileObject.CreateTextFile
        (Application("Visit_Count_F"), TRUE, Overwrite)
        Out.WriteLine t_visitors
    End If
    Application.Unlock
END SUB
</SCRIPT>
```

(그림 5) Session 시작을 위한 프로그램

(그림 4)의 Visit_Count_F=server.mappath("/") & "vis-

itors.txt" 문장에서 Server객체는 스크립터 자체가 서버에서 실행되는 것을 말한다. 그중 Mappath는 서버의 가상경로를 물리적 경로로 바꾸어 주는 역할을 하여 카운티 값이 저장되어 있는 "visitors.txt"의 경로를 설정한다.

Set FileObject = server.CreateObject("Scripting.FileSystemObject") 문장은 파일을 읽고 저장하기 위한 FileObject객체를 생성한다 여기서 FileSystemObject는 TextStream을 반환한다.

Set OUT = FileObject.OpenTextFile(Visit_Count_F, 1,FALSE,FALSE) 문장에서 OpenTextFile은 FileObject 객체에 TextStream을 돌려준다. Application.Lock 문장은 클라이언트들이 어플리케이션 변수를 수정하지 못하도록 하는 분산 상호 배제 록 기능을 한다. 이 이후의 부분은 임계 영역(critical section)이라고 한다. Application("visitors")=Out.ReadLine 문장은 Open된 파일에서 한 줄을 읽어서 어플리케이션 변수 visitors에 저장한다 Application("Visit_Count_F") - Visit_Count_F 문장은 text파일의 경로가 지정된 Visit_Count_F을 어플리케이션 변수에 저장

(그림 5)에서 Session("IsMember") = "NO", Session("AccessError") = 0, Session("UserName") = "" 문장은 chatting에 사용될 세션변수를 초기화하는 역할을 한다. Session.Timeout = 60*24 문장은 활성화된 세션의 timeout속성을 지정, 분으로 하루를 설정한다 session("ConnectionString") = "dsn=space;uid=sa;pwd="문장은 ConnectionString 세션변수에는 연결에 필요한 서버의 이름, uid, password를 설정한다 Application.Lock 문장은 클라이언트들이 어플리케이션 변수를 수정하지 못하도록 분산 상호 배제 록 기능을 한다. Application("visitors") = Application("visitors") + 1 문장은 어플리케이션의 visitor변수의 값, 즉 카운티 값을 1만큼 증가 시킨다 t_visitors = Application("visitors") 문장은 임시 변수 t_visitors에 어플리케이션 변수 visitors의 값을 할당한다. Session("VisitorID") = t_visitors 문장은 세션변수 VisitorID에 t_visitors값을 할당한다.

4. 타당성 검토

본 장에서는 분산 상호 배제 카운트 알고리즘에 대한 타당성을 검토한다 분산 상호 배제 카운트 알고리즘에 대한 타당성은 여러 가지 측면에서 검토가 가능하다. 여러 가지 검토 가능한 측면은 데이터의 일치성

을 보장할 수 있느냐는 문제와 교착 상태(deadlock)가 발생하지 않느냐 하는 문제이다. 본 논문에서 제안하는 알고리즘의 타당성 검증은 자료의 일치성을 중심으로 한다. 교착 상태가 발생하지 않느냐에 대한 검증은 실험을 통해서 직접적인 검증 과정을 거친다.

본 논문에서 제시하는 분산 상호 배제 카운트 알고리즘이 제대로 동작이 되는지에 대한 검증은 다음과 같은 well-formed의 집합을 이용하여 자료 일치성이 보장됨을 증명할 수 있다.

논리 4.1. $\{T_1, T_2, \dots, T_k\}$ 가 two-phase transactions이고 well-formed의 집합인 경우, 어떠한 합법적인 schedule Q_0 도 일치성을 보장한다.

증명. 이것의 증명은 Q_0 가 serial schedule과 같다는 것을 보이면 된다. UNLOCK(T)를 Q_0 내의 T에 대한 최초의 unlock 행위라고 정의한다. 일반성을 잃지 않으면서 우리는 다음과 같은 추론을 세울 수 있다. UNLOCK(T_i) < UNLOCK(T_{i+1}) for all $0 \leq i \leq k$. Q_i ($i=1, \dots, k-1$)인 partial schedule을 정의하면 Q_i 는 T_j ($j < i$) action을 포함하지 않고 Q_{i-1} 은 $T_i Q_i$ 이다. 따라서 Q_0 는 serialize하다 증명을 단순화하기 위하여 $i=1$ 인 경우에 대해서 증명한다. Q_0 에 대해서 2개의 이웃하는 action을 생각하면 $(T_i, A_i < O_i >, V_i)$ ($T_1, A_1 < O_1 >, V_1$) with $T_i \neq T_1$. 우리는 이러한 2개의 action이 동일한 schedule Q_0' 와 상호 변환 가능하다 여러가지 고려 가능한 상황은

- (1) A_1 또는 A_i 는 commit_object action이라고 가정한다. 왜냐하면 이 action은 locking status나 objects의 값에 영향을 미치지 않기 때문에 Q_0' 는 legal이고 $DEP(Q) = DEP(Q_0')$ [$DEP(Q)$ 의 의미는 스케줄러 Q내의 트랜잭션 T_i 와 T_j 의 쌍에 대해서 T_i 가 쓰여졌던 값에 직접 의존하거나 또는 T_i 가 앞서 읽히거나 쓰여진 값을 덮어쓰기 위한 최초의 트랜잭션 T_i 이다.] 이다
- (2) $O_i \neq O_1$ 이라고 가정한다. A_1 과 A_i 는 충돌이 발생하지 않는다. 왜냐하면 이들은 서로 다른 object에 접근할 수 없다. 결과적으로 Q_0' 는 legal이고 $DEP(Q) = DEP(Q_0')$ 이다.
- (3) $O_i = O_1$ 이라고 가정한다. A_i 나 A_1 어떤 것도 write_object action이나 lock_X action이 될 수 없다. 왜냐하면 이것의 의미는 T_1 과 T_i 가 O_1 에 대해서 동

시에 배타적으로 접근한다는 것을 의미한다. 우리는 A_i 와 A_1 이 lock_S, unlock, read_object actions 중 하나가 되어야 한다. 이것은 2가지 subcase를 생각할 수 있다.

- (a) $A_1 \neq \text{unlock}$ 이라고 가정한다. A_i 와 A_1 은 schedule legality나 DEP action에 영향을 받지 않고 교환 가능하다.
- (b) $A_i = \text{unlock}$ 이라고 가정한다. $DEP(Q_0) = DEP(Q_0')$ 왜냐하면 DEP는 lock action을 갖지 않기 때문이다. Q_0' 의 legality를 확립하기 위하여 A_1 은 lock_S가 될 수 없다. 왜냐하면 이것은 UNLOCK(T_1) < UNLOCK(T_i)라는 가정에 위배된다. 만약 A_1 이 unlock 또는 read_object action일 경우 legality는 명확해진다.

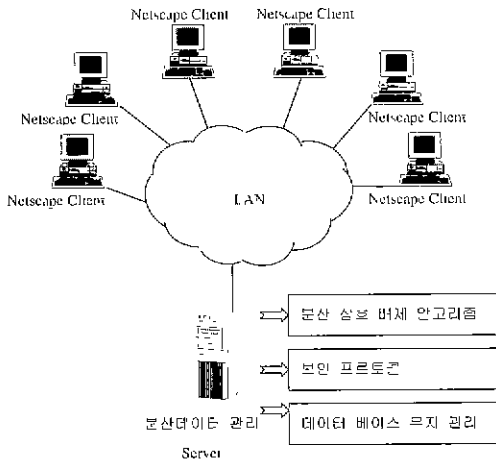
따라서 T_1 의 모든 action을 $T_1 Q_1$ 의 순서를 형성할 수 있다. $T_1 Q_1$ 은 Q_0 와 같다. 이러한 과정을 T_2, T_3, \dots, T_n 에 대해서 계속 정리하면 Q_0 는 serial schedule $T_1 \dots T_n$ 임을 알 수 있다

5. 실험 및 평가

위의 3절에서 제시한 알고리즘을 실제 분산 시스템 형태로 구축을 하였다. 분산 시스템으로 구축된 서버는 Windows NT 운영체제를 사용하고 전자 상거래 서버로 IIS(Internet Information Server)를 사용하였다. 본 논문에서 제시한 분산 시스템의 구조기 제대로 동작하는지에 대한 평가를 위하여 실험을 실시하였다. 본 논문에서 제시한 알고리즘에 대한 실험은 여러 측면으로 실험이 가능하다. 그러나 가장 중요한 측면은 분산 상호 배제 알고리즘을 이용한 사용자 구분자인 카운트 값이 중복없이 할당이 되느냐 하는 문제이다.

실험 방법은 "손님"으로 접속을 하였을 경우 각 사용자를 구분하는 카운트 할당이 배타적으로 정확히 이루어지는지에 대해서 이루어졌다. 이러한 기능 시험을 위하여 5대의 클라이언트 사용자가 구축된 웹 서버에 "손님"으로 동시 접속하였다(그림 6). 처음 몇 번의 실험은 한대의 클라이언트에서 이루어졌다. 정상적인 동작을 확인하고 나서 5대의 클라이언트를 구축한 후 웹 서버 동시 접속 환경을 만들었다. 5대의 클라이언트 접속이 끝나고 나서 "visitors.txt" 파일 내의 값을 확인하는 작업을 통해서 실험이 정상적으로 종료되었는

지를 확인한다. 실험 결과는 다음의 <표 1>과 같다



(그림 6) 실험 환경

<표 1> 클라이언트 접근을 이용한 실험

클라이언트 실험횟수	클라이언트 1	클라이언트 2	클라이언트 3	클라이언트 4	클라이언트 5
1	3	5	1	4	2
2	6	8	10	7	9
3	15	11	13	12	14
4	19	16	20	18	17
5	23	24	22	25	21
6	30	27	29	26	28
7	31	33	35	32	34
8	40	36	39	38	37
9	42	45	44	41	43
10	49	48	46	47	50
11-100	정상	정상	정상	정상	정상
101-200	정상	정상	정상	정상	정상
201-300	정상	정상	정상	정상	정상

위 실험은 각 클라이언트 시스템을 5대 구축하고 구축된 클라이언트 시스템에서 사용자 구분 정보인 카운트 값에 대한 상호 배제가 정확히 보장되는지를 실험하였다. 실험 횟수는 300번까지 하였다. 300번의 실험에서 한번이라도 카운트 값에 이상이 발생하면, 즉 중복 할당되는 값이 생길 경우는 본 논문에서 제시한 카운트를 이용한 사용자 구분 방법을 전자 상거래 서버 시스템에 적용할 수 없다는 의미이다 따라서 카운트 값에 대한 정확한 계신이 이루어지는가에 대한 실험을 실제 환경과 같이 구축된 상황에서 이루어지도록 하였다.

위 실험을 통해서 얻을 수 있는 결론은 본 논문에서 제시한 방법에 대한 사용자들 간에 정확하게 동기화가 이루어지고 있음을 알 수 있다. 만약 클라이언트 사용자들 간에 정확한 동기화가 이루어지지 않으면 분산 시스템을 사용하는 사용자들에 대한 데이터 접근에 대한 혼란을 초래할 것이다. 본 실험은 이러한 기능이 정확히 이루어지는지에 대한 검증은 위한 것이다. 따라서 본 실험을 통해서 분산 시스템 서버에 구축된 카운트를 이용한 사용자 구분 방법은 사용자가 회원으로 등록을 하지 않고 각 사용자들을 구분할 수 있는 대체 기능으로 사용 가능하다는 것을 보여주고 있다.

6. 결 론

본 논문은 분산 시스템 구조를 이용한 전자 상거래 시스템 구축에서 사용자에게 편리한 기능을 제공하기 위한 방안의 하나로 회원으로 등록하는 빈거로유과 같은 불편함을 해소하기 위한 방안으로 회원이 아닌 경우는 각 사용자를 구분하는 구분자 ID(카운트)를 이용한 메카니즘을 제시하고 실제 시스템 구축을 통해서 본 논문에서 제시한 방법의 타당성을 입증하였다. 일반적으로 전자 상거래 시스템들은 회원 등록을 통하여 이용하도록 되어 있다. 본 논문은 이러한 문제점을 해소하기 위하여 회원이 아닌 사용자들을 구분할 수 있는 구분자를 할당하여 이 값으로 각 개인들을 구분하도록 하였다.

이 카운터 값은 배타적인 상호 배제를 이용하여 모든 사용자들이 서로 다른 값을 갖도록 한다. 그리고 이 카운터 값과 사용자의 주민등록번호를 연계함으로써 카운터에 해당하는 인적 정보와의 연동이 가능하나 각 사용자들이 전자 상거래 시스템에 “손님”으로 접속하게되면 배타적인 접속이 가능하도록 카운트를 할당하게 된다. 이 카운트는 사용자들을 구분하는 역할을 하게 된다. 이 카운트를 배타적으로 할당하기 위하여 분산 상호 배제 알고리즘을 이용하였다. 락킹(locking) 알고리즘의 기본적인 기능은 ASP에서 제공하는 것을 사용한다.

본 논문에서 제시한 기본 분산 락킹 알고리즘을 사용한 카운트 값의 할당으로 전자 상거래 서버를 이용하는 사용자들을 구분할 수 있는지에 대한 검증을 위하여 실제 시스템을 구축했다. 실제 구축된 시스템은 Windows NT 서버를 이용하여 구축했다. 그리고 각

사용자들에 대해서 배타적인 카운트 할당이 가능한지에 대한 검증은 위하여 실험을 하였다. 실험은 1대의 서버와 5대의 클라이언트로 구성되었다. 실험 횟수는 300번의 접속으로 이루어졌다. 300번의 전자 상거래 서버 접속을 통해서 각 실험 때 마다 배타적인 카운트 할당이 가능한지를 실험하였다. 실험 결과 매 실험 마다 배타적인 카운트 할당이 가능함을 확인하였다. 따라서 본 논문에서 제시하는 개념이 분산 시스템 환경의 전자 상거래 시스템에 쉽게 응용이 가능하고 편리한 인터넷 환경을 구축할 수 있는 개념으로 사용할 수 있다. 향후 본 논문에서 제안한 내용을 분산 시스템 환경인 인터넷 시스템에 적용을 통해서 문제점들을 찾아내고 이러한 문제점들을 개선하고자 한다.

참 고 문 헌

[1] 박석, 양지혜, "데이터베이스 보안 모델", Korea Institute of Information Security & Cryptology, 1996.
 [2] 하경주, 윤계우, 강창구, 정승주, "보안로큰을 이용한 웹 보안 시스템 개발". Vol.6, No.3, 정보처리학회 논문지, 1999 3.
 [3] 송주영, "Internet과 초고속정보통신망", 정보과학회지, 제14권, 제11호, 1996 11
 [4] 김홍근, 최영철, "전자상거래 경쟁력 강화와 정보보호 기술 개발 전략", 한국정보보호센터, 1998.
 [5] 김상하, "클라이언트/서버 최신 기술 동향", Vol.4, No.6, 정보처리학회지, 1997. 11.
 [6] 정영아, 김용근, 한승조, "웹상에서 정보시각화 및 줌 브라우저의 구현", 한국정보처리학회 논문지, Vol.6, No 4, 1999 4.
 [7] 이정매, "웹을 기반으로 한 원격 제어시스템 환경 설계 및 구현". 한국정보처리학회 논문지, Vol.6, No 2, 1999 2.
 [8] 이준석, "웹 브라우저와 CGI 프로그램 사이의 보안 통신을 지원하는 시스템 설계 및 구현", 한국정보처리학회 논문지, Vol.6, No 3, 1999 3
 [9] 조인준, 정희경, 김동규, "인터넷 보안 매카니즘에

대한 연구", 통신정보보호학회 논문지, 1998. 6.
 [10] 이경천, "전자상거래 소프트웨어 에이전트", 한국정보처리학회 논문지, Vol.6, No 1, 1999. 1.
 [11] Peter D Hipson, 윈도우 NT 서버, 사이비출판사, 1997
 [12] Scot Hilber, Daniel Meziak Inside Active Server Pages, 영진출판사, 1998
 [13] Maekawa, Oldehoeft. Operating Systems(Advanced Concepts), Benjamin/Cummings Publishing Co., 1987.
 [14] Netscape, Inc., "SSL 3.0 Specification." <http://home.netscape.com/neg/ssl3/3-SPEC.HTM>, 1997.
 [15] Java Servlet API, <http://jserv.javasoft.com/products/java-server/>
 [16] 'A Framework for Global Electronic Commerce.' Dec 1996. <http://www.intf.mst.gov.eleccomm/global-comm.htm>.
 [17] Mihani, A. and Marcugini, S., COOPBOT : Distributed Cooperating Agents for Electronic Markets, Proc. Of the First International Conference on Electronic Commerce, 1998.
 [18] A Framework for Electronic Commerce in NII, NII, 1996.
 [19] Jang Seung-Ju, Kim Gil-Yong, "Spin-Block Mutual Exclusion Algorithm in the Large Scalable Server," International Journal of PDCN, 1998.

장 승 주



e-mail : sjjang@hyomin.dongueui.ac.kr
 1985년 부산대학교 계산통계학과 졸업(계산학)
 1991년 부산대학교 계산통계학과 (계산학 이학석사)
 1996년 부산대학교 컴퓨터공학과 (공학박사)

1987년~1996년 한국전자통신연구원 시스템S/W 연구실
 1996년~현재 동의대학교 컴퓨터공학과 조교수
 관심 분야 : 운영체제, 보안, 분산처리 시스템