

理學碩士 學位論文

CAPI를 이용한 웹 기반의 Solver에 관한 연구

A study on the Web-based Solver using
C API programs

指導教授 金 宰 煥

2003年 2月

韓國海洋大學校 大學院

應 用 數 學 科

金 聖 和

목 차

Abstract	ii
I. 서론	1
1.1 연구배경 및 내용	1
1.2 연구범위	4
II. C API를 이용한 웹 사이트 솔루션 개발	5
2.1 CGI	5
2.2 API	7
2.3 CGI 와 API 비교	9
2.4 C API 개발 프로그램의 주요내용	10
2.4.1 개발환경	10
2.4.2 C API 개발 프로그램의 주요내용	11
III. 구현예제	22
3.1 소팅 프로그램	22
3.2 NRO 알고리즘의 솔루션	23
IV. 결론	25
참고문헌	26
부 록	27

ABSTRACT

In this thesis, we present C API programs which are capable of interface PHP programs to C-language programs. Using C API programs, we can develop web-based Solvers which provide conveniently solutions for the programs such as sorting programs, optimization programs, statistical analysis programs on the Web-site. And C API problems also provide users with effective data utilization on the different operating system, that is, 'Windows 2000' used for web server and 'Linux server' used for database server. Processing with C API, the flow of data on the different operating system become more economical and useful. Two examples, including a sorting program and a NRO(Network Reliability Optimizer) program, are given in this paper. The further study is constructing more user-friendly environment on the Web-site.

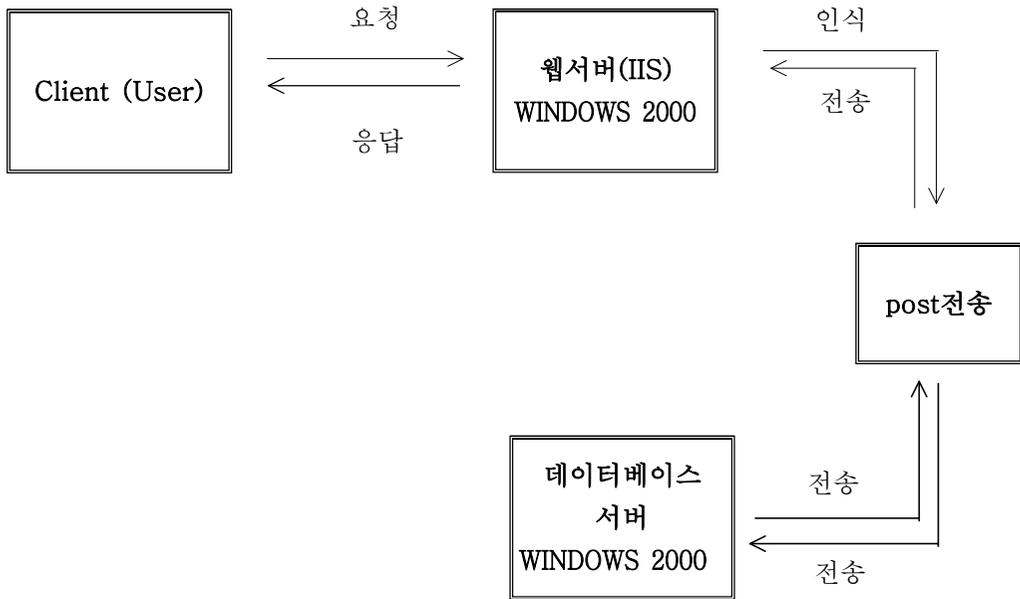
I. 서 론

1.1 연구배경 및 내용

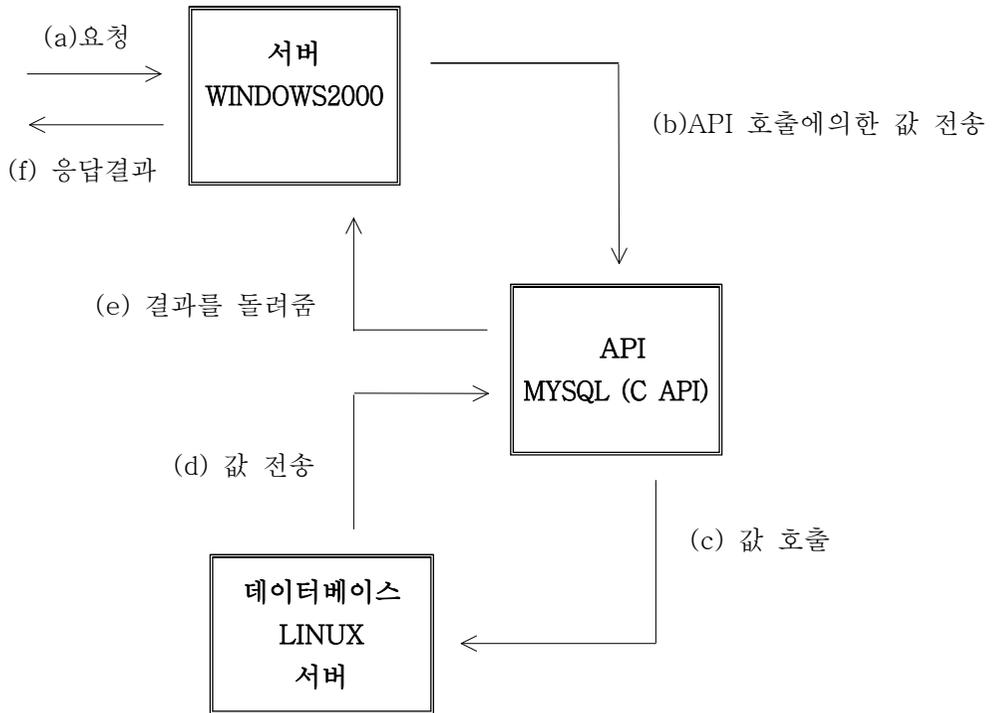
현재 인터넷의 사용량 증가에 따라 많은 웹 사이트가 만들어지고 있다. 하루에도 몇 만개의 사이트가 만들어져서 사용자에게 정보를 제공하고 있다. 그러나 현재 사용자에게 제공되는 거의 대부분의 사이트는 단순히 정보를 보여주는 것에 그치고 있다. 이제는 단순히 정보만을 제공하는 사이트보다는 정보의 가공에 의한 새로운 솔루션을 제공하는 사이트를 사용자들은 요구하고 있다. 서버의 형태도 지금까지는 한대의 컴퓨터에서 모든 과정을 처리하는 형태이나 지금은 여러 컴퓨터에 서버들의 기능을 분산시켜 처리하는 형태로 바뀌고 있다. 이러한 변화에 따라 사용자의 요구를 해결할 수 있는 사이트의 개발이 활발히 이루어지고 있으며 여러 컴퓨터의 사용에 따른 상호간의 자료의 이동을 적절하게 이용할 수 있는 방법에 대한 연구가 활발히 이루어지고 있다([8], [11], [16], [17]).

본 논문에서는 사용자에게 가공의 정보를 웹상에서 편리하게 제공할 수 있는 C API를 개발하였고 여러 대의 컴퓨터에 서버를 두어 사용하는 변화에 맞추고 나아가 여러 컴퓨터가 서로 다른 운영체제라 하더라도 같은 운영체제에서 사용하는 것과 같은 효과를 주는 방법도 개발하였다.

다음 <그림 1>은 현재 웹 사이트들의 흐름도이며 <그림 2>는 본 논문에서 개발한 흐름도이다. 현재 사이트는 대부분이 정보의 가공 없이 자료를 있는 그대로 사용자에게 제공하는 반면 본 논문에서 개발한 알고리즘은 사용자로부터 정보를 입력 받고 그 자료를 C를 이용하여 새로운 솔루션을 구하여 사용자에게 제공하는 형태이다. 또한 여러 컴퓨터를 사용하더라도 현재 웹 사이트에서 제공되는 서비스와 같은 효과를 주고 아무런 불편 없이 사용할 수 있도록 만들었으며 방대해진 자료의 이동으로 인한 자료의 손실이나 자료 손실로 인해 발생하는 웹 사이트의 오류 등을 줄이고자 한다. 그 방법을 본 논문에서는 다른 운영체제라도 자료의 이동이 가능하도록 개발하였다.



<그림 1> 현재의 웹 사이트 운영방식



<그림 2> API 개념도

<그림 2>는 본 논문에서 개발한 API 알고리즘의 흐름도이며 현재 프로그래밍의 흐름과 사용자의 요구와 지속적인 환경변화에 적극적으로 대처할 수 있는 방안으로 이 방법을 제시한다. 웹 서버인 WINDOWS 2000서버에서 사용자로부터 값을 입력받고 입력받은 값을 MYODBC를 이용하여 서로 다른 종류의 LINUX 서버에 있는 데이터베이스 서버 MYSQL에 값을 입력하고 LINUX 서버에서 C API 프로그램을 이용하여 데이터베이스에 있는 값을 호출하여 알고리즘에 맞는 계산을 하고 다시 MYSQL에 저장을 한다. WINDOWS 2000서버에서는 다시 MYODBC를 이용하여 LINUX 서버의 MYSQL 서버로부터 값을 받아 사용자에게 값을 전달하는 방법을 이용한다.

1.2 연구범위

본 논문에서는 현재 개인의 컴퓨터에서만 가능한 알고리즘(본 논문에서는 C알고리즘을 사용함)솔루션을 구하는 것을 웹 사이트를 통하여 편리하게 하게 제공할 수 있도록 그 환경을 개발한다. 서로 다른 환경(WINDOWS2000, LINUX)으로 구성된 서버환경에서 C API를 이용하여 자료의 입·출력이 가능하게 한다. 그리고 서로 다른 운영체제의 컴퓨터에 각각의 서버를 설치하여 운영하더라도 같은 운영체제의 컴퓨터에서 서버를 운영하는 것과 같은 효과를 줄 수 있는 환경을 개발한다. 현재의 웹 사이트에서 제공하는 서비스와 아무런 차이 없이 사용자에게 정보를 계속적으로 제공할 수 있도록 하며 사용자에게 가공의 정보를 웹상에서 편리하게 제공할 수 있는 환경을 개발한다.

II. C API를 이용한 웹 기반의 Solver

본 논문의 개발 프로그램인 C API를 설명하기 전에 현재 많이 사용되고 있는 일반 CGI나 PHP에 대하여 알아보고 그 문제점이 무엇인지 알아본다.

2.1 CGI (Common Gate Interface)

CGI는 HTML의 정보처리 능력의 한계를 보충해주는 역할을 한다. 외부 프로그램과 웹 서버(HTTP 서버)간의 연결역할을 하는 하나의 규약이다. 통상적으로 C/C++, PERL, UNIX Shell, Tcl/Tk를 많이 사용한다. 이 CGI를 구현하기 위해 웹 쪽에서는 FORM 태그를 통해서 사용자의 입력 값을 웹 서버(HTTP)로 보내고, 서버에서는 그 값을 CGI 프로그램에게 입력 값을 넘겨준다. CGI는 웹 서버의 기능을 확장해주며 웹 서버와 통신하기 위한 표준 규약을 제공한다. 서버 쪽에만 설치하므로 운영이 쉬우며 URL을 요청할 때 CGI를 따로 요청 할 필요가 없으며 동적인 서비스가 가능하다는 것이 CGI의 장점이다. 그러나 CGI는 시스템에 대한 부하를 많이 주게 된다. 모든 요청에 대해 개별적으로 요청에 응해야 함으로 시스템 자체의 많은 활용이 이루어진다. 그리고 URL을 통해 모든 것이 전달되므로 보안에 취약하며 앞뒤 서비스의 연결을 거의 유지하지 못한다. 앞의 요청이 있는 후 뒤에 전달하고 전달 뒤 다른 요청이 있으면 앞의 요청은 사라지며 더 이상 존재하지 않는다는 단점이 있다. 그러므로 상태의 저장이 자연스럽게 이루어지지 않으며 사용자간의 대화형 웹 페이지 구성이 어렵다. 클라이언트/서버 환경의 프로그램이 특정 통신규칙을 가지고 있음에 비해, CGI 프로그램은 어떤 통신규칙이 사용될지 사전에는 알 수 없다. 브라우저가 서버에 자료를 요구하는 헤더(MIME)를 분석해야만 답을 줄 수 있는 것이다. 이런 CGI는 많은 문제점을 가지고 있다. CGI는 서버의 CPU와 메모리의 퍼포먼스에 많은 영향을 주므로 서버의 CPU 사용에 대하여 모든 사용자가 공평하게 사용할 수 있도록 CGI의 남용을 자제한다. CGI 채팅의 경우 그 문제가 더 심각해지며 CGI

로 메일을 발송할 경우 sendmail을 사용하면 메일 발송을 프로그램 내부에서 작동시킬 수 있다. 이럴 경우 메일이 성공적으로 도착하지 않고 돌아올 경우 시스템 관리자에게로 발송되어 메일의 내용이 공개되어 개인의 프라이버시에 문제가 생긴다.

다음으로는 directory, file permission문제이다. CGI 파일은 mode 777로 된 디렉토리에 저장이 되고 0*W권한이 있는 파일이 생성되어 local에서 쉽게 웹 서버의 권한을 취할 수 있게 된다. 또한 잘못된 permission을 통해 해당 글이 저장되는 파일이 있다. 이것은 nobody를 제외한 유저들의 W권한을 없애는 것도 하나의 방법이고 해당 글이 저장되는 파일에 대한 permission을 검사해야한다. 또한 shell을 실행하는 함수의 취약점이 문제이다. C의 경우 system(), popen() 함수와 perl의 system(), open(), eval(), exec(), (Backticks) 등의 함수가 문제이다. file open또한 문제가 있다. open()이나 fopen() 같은 경우 FILE을 열 때 보통 많이 사용하는데 null byte, 역 패스나 w, l 또는 meta characters들을 제거해야 한다.

예외 처리에 있어 문제가 발생한다. 프로그램 작성시 예외처리를 하지 않으면 socket이나 file등 open 하는 함수를 사용할 때 오류가 발생하면서 해당 서버의 full path를 노출하게 된다. 프로그램 구조상 if등이나 제어문에서 조건을 판단할 시에는 변수의 조작 가능성을 살펴보는 것이 좋다. admin auth problem등 기타 인증관계의 문제가 존재하며 C, C++로 프로그램을 작성하는 경우 buffer overflow를 조심해야 한다[7].

2.2 API

응용프로그램은 컴퓨터 운영체제나 데이터베이스 등 다른 프로그램의 기능을 이용하기 위한 인터페이스이다. 실제로 운영체제 등의 기능과 그 기능을 사용하는 방법을 정의한 함수의 집합을 말한다. 응용 프로그램은 API를 사용하여 운영체제 등이 가지고 있는 다양한 기능을 이용할 수 있다. 초기에는 개인용 컴퓨터에서 응용프로그램이 하드웨어의 기능을 직접 조작하는 경우가 많았다. 그러나 파일관리나 정보의 화면 표시기능 등 모든 기능을 응용프로그램 내에 두면 프로그램 개발 효율이 떨어지고, 복수의 응용프로그램을 번갈아 사용하였을 때 문제가 발생한다. 따라서 많은 응용프로그램이 공통으로 이용할 수 있는 기능을 운영체제 등에 두는 것이 일반화 되었다. 응용프로그램 작성자가 프로그램에 함수를 이용할 수 있게 된다. 이 함수의 집합이 API이며, 종류가 다른 운영체제 사이에 API의 공통형식이 규정되면 다른 운영체제의 컴퓨터 사이에 응용프로그램의 이식성이 확보된다.

대표적인 서버 PHP를 살펴보자. PHP 만큼 쉽고 빠르게 웹 서버를 구축할 수 있는 언어도 드물다. 기존의 C나 Perl로 어렵게 CGI 프로그램을 하던 것들을 PHP는 너무도 쉽게 해결해준다. 쉬운 데이터베이스연결, 빠른 실행 속도, 멀티 플랫폼 지원 등의 장점을 가진 것이 PHP이다. 한마디로 말하면 서버에서 해석되는 스크립트언어이다. 이런 서버용 언어는 C나 Perl등으로 만들었던 CGI 프로그램과 마찬가지로 서버에서 해석되고 그 결과만을 HTML형태로 만들어서 클라이언트로 보내주기 때문에 웹 브라우저는 내부 소스 코드를 볼 수가 없다. 이것은 보안상으로 상당히 중요한 장점이라 할 수 있다. PHP에서는 데이터베이스연결이 쉽다. 실제 지원하는 데이터베이스는 Oracle, Sybase, mSQL, MySQL, Solid, ODBC, Postgre SQL, Adabas D, FilePro, Velocis, Infomix, dBase, Unix dbm등 거의 모든 데이터베이스가 포함되어 있다. 또한 PHP는 유닉스와 윈도우 환경에서 동작하는 대부분의 웹 서버를 지원한다. 유닉스 환경에서는 아파치 웹 서버에 모듈 형식으로 붙을 수 있고 CGI처럼 동작할 수도 있다.

PHP는 앞에서 설명한 바와 같이 서버에서 해석되는 서버스크립트 언어이

고 이 스크립트의 가장 큰 장점은 모든 운영체제(리눅스, 솔라리스, HP-UX, WINDOWS95/98, WINDOWSNT/2000)를 지원한다는 점과 손쉽게 코딩 할 수 있다는 것이다. 현재 PHP는 4.2.2버전 까지 나와 사용자들에게 무료로 배포되고 있다. 주목할만한 점은 해석 방식의 변화이다. PHP3의 컴파일 방식은 execute-while-parsing(한 줄씩 해석)인 인터프리터 방식 이었으나 PHP4는 C나 CGI프로그래밍과 같은 컴파일 방식으로 바뀐 것이다. 이것은 처리 속도의 향상을 가져 왔다. 또 한 더 많은 모듈을 지원하게 되었으며 웹 서버 인터페이스가 강화되었다. 웹 서버는 일반적으로 어떤 요청에 대해서 어떻게 응답한다는 것을 나타내주는 정보를 응답헤더에 실어 웹 브라우저 쪽으로 보낸다. PHP의 경우 웹 브라우저 → 웹 서버 → 웹 프로그램(PHP)에 대한 요청순서와 그 반대의 응답과정을 모두 거치고 웹 서버는 웹 프로그램이 발생하는 모든 헤더에 대해 분석해서 응답 헤더를 만들기 때문에 처리가 복잡해지고 속도가 지연될 수 있다. 이런 문제점을 해결하기 위해 NPH(Non-Parsed Header)기법을 사용한다. 이것은 어떤 웹 프로그램의 처리 결과를 웹 서버의 응답헤더를 생성하지 않고 웹 프로그램이 응답 헤더까지 생성해주는 기법이다. 그러나 이런 기법도 파일명에 'nph-'가 붙어야 되며 웹 서버가 생성하는 모든 응답헤더를 생성해서 보내야 되는 단점이 있다. 또한 upload에 취약점이 있다. php ph등의 file을 upload한 후 nobody 권한을 획득하고 bindshell을 통해서 시스템에 침투하고 apache site를 해킹한다. 이런 방법은 국내 크래커나 black hats가 많이 사용하는 기법이다. 파일확장자가 php, php3, ph등의 file이 upload되는 것을 방지하고 apache에서 또한 강력한 인증을 하는 것이 좋다. 또한 setup file(db, passwd, 기타 설정)의 노출이 문제가 된다. url을 통해 ph inc 파일에 접근할 수 있다. 이것은 apached의 http.conf 파일에 AddType application/x-httpd-php, php, php3, ph inc를 추가하여 문제를 해결할 수 있다. php역시 system(), open(), eval(), exec(), escapeshellcmd()등의 함수 사용에 의해 해킹의 위험이 있다 [1].

2.3 CGI와 API의 비교

	CGI	API
웹 서버와의 호환성	CGI를 지원하는 모든 웹 서버에서 사용가능.	특정 웹 서버에 종속.
수행속도	별도의 프로세스의 기동으로 실행속도 저하.	웹 서버와 동등한 위치에서 실행되어 속도가 빠름.
웹 서버와의 연동	환경변수와 표준입력만이 가능.	웹 서버의 실제 기능들을 대부분 이용.
처리기능	보통.	매우 높음.
시스템 요구사항	부하가 크므로 빠른 서버를 요구함.	일반적인 웹 서버의 기능이기만 하면 가능.
보안	기본 인증.	다양한 인증 가능.
개별 난이도	CGI 인터페이스 이해하면 충분함.	복잡한 API함수를 다루어야 하므로 복잡함.

<표 1> CGI와 API의 비교

2.4 C API 개발 프로그램의 주요 내용

2.4.1 개발환경

본 논문에서 사용된 개발환경은 다음과 같다.

▷ 웹 서버

- 운영체제 : WINDOWS2000
- 웹 서버 : IIS
- 언어 : PHP4
- 기반기술 : WINDOWS NETWORKING PROGRAMM,
PHP PROGRAM, MYODBC

▷ 데이터베이스 서버

- 운영체제 : LINUX
- 서버 : MYSQL
- 언어 : C
- 기반기술 : MYSQL, C API, LINUX SOCKET PROGRAMM

WINDOWS 2000 서버에는 웹 서버로는 IIS를 사용하였으며 언어는 PHP를 사용하였고 데이터베이스는 다른 운영체제인 LINUX서버의 MYSQL을 사용하였다.

현재 웹 사이트들의 규모가 방대해지고 사용자에게 더 많은 정보를 제공하는 대형 사이트가 개발되고 있는 상황이다. 기존의 웹 사이트는 한 컴퓨터에 웹 서버와 데이터베이스 서버가 같이 존재하므로 서버간의 연결이 간편하게 이루어 졌다. 그리고 자료의 이동 또한 간단한 스크립트에 의해 전달이 가능하였다. 그러나 현재의 상황은 한대의 컴퓨터에 모든 자료를 처리한다는 것이 거의 불가능한 상황에까지 이르렀다. 특히 데이터베이스 서버의 과포화 상태는 매우 심각한 상황에 이르러 데이터베이스 서버의 과부화로 인한 웹 서버의 다운 현상이 자주 나타나고 있다. 그래서 각 사이트에서는 이런 문제를 해결하기 위해 방안을 개발하고 있다. 가장 많이 이용하는 방법은 다른

컴퓨터로 데이터베이스를 옮겨 그것을 기존의 데이터베이스와 같이 연동하여 사용하는 방법이다. 자료의 양이 적은 경우는 옮기는 것이 쉽게 이루어지나 자료의 양이 많은 경우에는 자료를 옮기는 것이 쉽지만은 않다. 옮기는 작업이 힘들뿐만 아니라 옮기더라도 현재 기능을 100% 모두 다 발휘한다고 할 수 없으며 이전의 자료를 살리지 못하고 있는 경우가 많다.

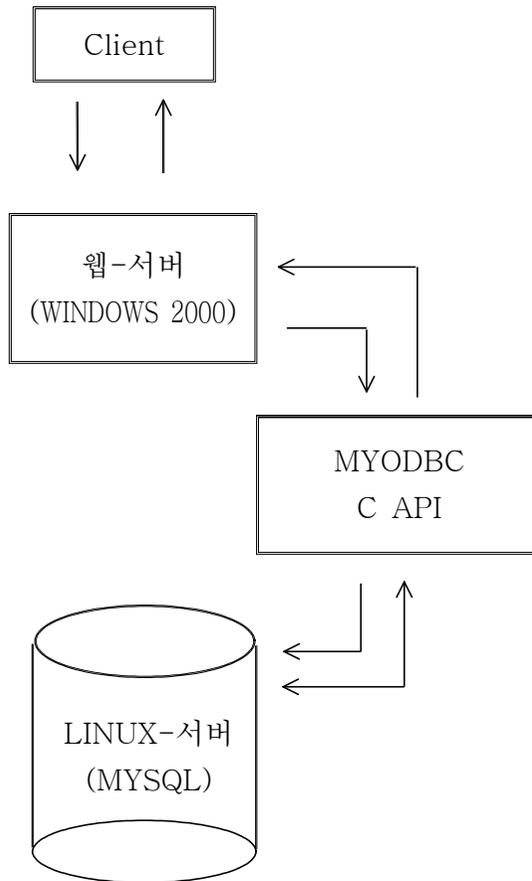
본 논문에서는 이러한 데이터베이스 서버의 이동으로 인한 자료의 손실이나 예전의 자료의 분실, 사이트의 접속불능 현상 등을 해결하고자 현재의 서버에서 다른 운영체제의 서버와 연결하여 지속적으로 웹 사이트를 운영하면서 사용자들에게 현재 상태와 같은 서비스를 제공할 수 있는 C API를 개발하고자 한다. 또한 다른 운영체제에 있는 서버라도 같은 운영체제의 서버로 옮겨 서비스를 제공하는 것과 같은 환경을 만드는 방법을 개발하고자 한다.

2.4.2 C API 개발 프로그램

본 논문에서는 현재 PHP와 C 언어가 상호교환이 되지 않는 단점을 극복하기 위해 C API를 이용하여 웹상에서 알고리즘의 솔루션을 사용자에게 직접 제공 할 수 있는 환경을 개발한다. 개발환경을 설명하기 위해 간단한 소팅 프로그램을 사용하였다. 일반 웹 서버에서는 이러한 소팅 알고리즘을 사용자에게 직접 제공하기 위해서는 웹 스크립트 언어로 직접 코딩을 해야 한다. 모든 알고리즘의 함수를 스크립트 언어로 표현하는 것은 한계가 있고 표현한다 하더라도 스크립트 언어의 전송방법이 해커들에 의해 노출되어있어 자료의 내용이 중요한 것일수록 자료의 보안문제는 여전히 남아있게 된다. 이러한 문제를 C API를 이용함으로써 알고리즘의 솔루션을 사용자에게 직접전달하고 스크립트 언어를 사용 할 때보다 더 안전한 방법으로 자료를 처리할 수 있다.

본 논문의 자료의 처리 순서는 다음과 같다. 클라이언트에서 사용자로부터 값을 입력받고 웹 서버(WINDOWS 2000)에서 그 값을 MYODBC를 이용하여 데이터베이스 서버(LINUX)의 MYSQL에 직접 연결하여 그 값을 입력한

다. 데이터베이스에 입력된 값을 C API를 이용하여 값을 읽어 계산하고 계산결과를 다시 MYSQL에 저장한다. MYSQL에 저장된 값을 MYODBC를 이용하여 다시 웹 서버(WINDOWS 2000)로 그 결과 값을 가지고와서 사용자에게 결과를 출력한다.



<그림 3> C API 알고리즘 개념도

<그림 3>은 본 논문의 C API 알고리즘 개념도이다. 스크립트 언어<부록 1>와 Automatic하게 사용자에게 제공하기위해 사용된 소켓 프로그램은 <부록2>에 다루었으며 여기서는 C API 알고리즘 부분을 설명한다.

```

#include <stdlib.h>
#include <string.h>
#include <mysql.h>
#include <errno.h>

int main(void)
{
    char *buffer;
    char one[5], two[5], three[5], four[5], five[5],
        six[5], seven[5], eight[5], nine[5], ten[5];
    MYSQL mysql;
    MYSQL_RES *res;
    MYSQL_ROW row;
    int fields;
    int cnt;
    int tmp;
    int i, j;
    int a[10];
    char *str[10];
    FILE *fp;
    char query[1000];
    char *querys;
    buffer = (char *)malloc(sizeof(char) *400);

```

<그림 2> 헤더파일선언과 변수선언

<그림 2>에서는 소팅알고리즘의 헤더 파일과 C API를 사용하기 위한 MYSQL헤더 파일을 선언하였으며 소팅알고리즘에 사용될 변수를 선언하였다. “**char *query; buffer =(char *)malloc(sizeof(char) *400);**”는 알고리즘의 결과를 MYSQL에 저장하기위해 알고리즘상의 가상공간을 미리 만드는 것이다.

```
mysql_init(&mysql);

if(!mysql_real_connect
    (&mysql,"203.230.253.70", "root", "7878", "test1",
     0, (char *)NULL, 0))

{
    printf("errno:%d\n%s\n",mysql_errno(&mysql),
           mysql_error(&mysql));
    exit(1);
}
```

<그림 5> 데이터베이스 서버에 연결

<그림 5>에서는 데이터베이스 서버에 연결하는 과정을 설명하고 있다. “**203.230.253.70**”, “**root**”, “**7878**”, “**test1**”, **0, (char *)NULL, 0)** LINUX서버가 있는 컴퓨터의 IP이며 사용자, 비밀번호, 데이터베이스이름, TCP/IP포트번호, 유닉스 소켓 파일경로, 클라이언트 기본값을 나타내고 있다. “**mysql_errno()**” 함수는 가장 최근에 수행된 MYSQL 관련 함수에 대한 에러 코드를 얻는다. 이 함수는 MSYQL에서 지원하는 함수이며 함수명은 MYSQL의 C API와 동일하다. 인자 mysql은 MYSQL 포인터이며 MYSQL 연결시 얻은 값이고 MYSQL 포인터가 아니면 에러 처리되며 이미 폐쇄된 MYSQL 포인터일 때에는 경고 메시지를 출력하고 result에 -1 의 값을 되돌린다. 에러 코드가 err_no에 돌려지며 그 값이 0 이면 에러가 없는

경우이다. err_no의 에러 코드는 MYSQL 데이터베이스에서 지원하며 에러 코드의 종류나 그 의미는 MYSQL에 쓰는 것과 같다. "Can't connect to local MYSQL server (2)" 에러는 MYSQL의 서버에 연결을 할 수 없다는 메시지로써 MYSQL 서버의 데몬이 실행 중이지 않을 때 나오는 메시지이다. mysql 디렉토리에서 safe_mysql_d 명령 등을 이용하여 데몬을 실행시켜 주면 해결 된다.

```
query="SELECT * FROM ok";
    if(mysql_real_query(&mysql,query,strlen(query)))
    {
    printf("error no:%d\n%s\n", mysql_errno (&mysql),
        mysql_error(&mysql));
        exit(0);
    }
    res=mysql_store_result(&mysql);
    fields=mysql_field_count(&mysql);
    while((row = mysql_fetch_row(res)))
    {
        for(cnt =0;cnt<fields;++cnt)
        {
            printf("%12s",row[cnt]);
            a[cnt]=atoi(row[cnt]);
        }
        printf("\n");
    }
    mysql_free_result(res);
    mysql_close(&mysql);
```

<그림 6> 데이터베이스 서버로부터 값 전송

<그림 6>에서는 <그림 5>에서 연결한 데이터베이스 서버로부터 사용자가 입력한 값을 불러오는 과정을 설명하고 있다. “mysql_error()” 함수는 가장 최근에 수행된 MYSQL 관련 함수에 대한 에러 메시지를 얻는다. 이 함수는 MSYQL에서 지원하는 함수이며 함수명은 MSYQL 의 C API와 동일하며 인자 mysql은 MYSQL 포인터이며 MYSQL 연결시 얻은 값이고 MYSQL 포인터일 때에는 경고 메시지를 출력하고 result에 -1 의 값을 되돌린다. 에러 메시지에 대한 문자열이 err_msg에 돌려지며 그 값이 Null string(“”)이면 에러가 발생하지 않은 것이다. err_msg의 에러 메시지는 MYSQL 데이터베이스에서 지원하며 메시지의 종류나 그 의미는 MYSQL 과 동일하다.

“mysql_store_result(), mysql_fetch_row()” mysql_store_result() 함수는 우리가 자주 사용하는 웹 스크립트인 PHP에서는 mysql_use_result() 함수와 비슷한 기능을 한다. 두 함수 모두 최근에 수행한 SQL쿼리에 대한 결과값을 가져오는 함수이다. mysql_store_result() 함수는 모든 레코드를 서버로부터 전송받아 클라이언트프로그램에 저장하게 되는 반면 mysql_use_result() 함수는 mysql_fetch_row() 함수를 이용하여 한 번 호출할 때마다 하나의 레코드를 서버에서 전송받게 된다. 따라서 mysql_store_result() 함수는 클라이언트에 많은 메모리를 사용하게 되는 반면 mysql_use_result() 함수는 레코드가 일단 서버에 저장된 후에 클라이언트로 전송되고 클라이언트는 하나의 레코드만을 가지고 있으므로 메모리도 mysql_store_result() 함수보다 적게 사용하게 된다. 또한 mysql_store_result() 함수는 클라이언트에서 복잡한 처리를 하게 되는데 mysql_use_result() 함수는 이런 일을 하지 않으므로 속도가 빠르다. mysql_use_result() 함수대신 mysql_store_result() 함수를 사용하는 이유는 mysql_use_result() 함수는 레코드를 서버에 저장하므로 서버가 메모리를 많이 사용할 수 있다. 또한 mysql_use_result() 함수를 사용하고 있는 테이블에는 READ LOCK이 걸려있는 상태이므로 다른 클라이언트 프로그램에서 해당 테이블에 UPDATE나 INSERT를 하려는 경우 다른 클라이언트 프로그램은 mysql_use_result() 함수가 모든 레코드를 사용할 때까지 기다려야 하므로 전반적으로 MYSQL의 속도가 느려진다. 따라서 클라이언

트에서 레코드를 이용해서 많은 계산을 하거나 사용자가 레코드 값을 천천히 보게 될 경우에는 `mysql_store_result()` 함수를 사용한다. 메모리 사용이나 속도 문제 이외에도 레코드에 접근할 수 있는 방법에서 차이가 난다. `mysql_use_result()` 함수는 하나의 레코드를 서버로부터 `mysql_fetch_row()` 함수를 이용해서 전송 받게 되므로 순차적인 방법으로만 레코드를 사용할 수 있다. 하지만 `mysql_store_result()` 함수는 레코드 전체를 서버로부터 전송 받아서 클라이언트에 보관하고 있으므로 `mysql_data_seek()` 함수를 이용해서 원하는 임의의 레코드로 바로 이동할 수 있다. 그리고 `mysql_use_result()` 함수를 사용하였는데 `mysql_fetch_row()` 함수를 이용하여 마지막 레코드까지 사용하지 않았다면 이 레코드는 삭제되지 않고 다음 쿼리를 수행하고 레코드를 가져올 때 이전의 레코드가 같이 전송되는 경우도 있다. 그러므로 여기서는 `mysql_use_result()` 함수보다는 `mysql_store_result()` 함수를 사용하였다. `mysql_store_result()` 함수에서 인자 `mysql`은 MYSQL포인터이며 MYSQL 연결시 얻는 값이고 MYSQL 포인터가 아니면 에러 처리되며 이미 폐쇄된 MYSQL 포인터일 때에는 경고 메시지를 출력하고 `result`에 -1의 값을 되돌린다. 그리고 임시 저장된 자료에 대한 MYSQL result 포인터가 `mysql_ptr`에 돌려진다. `mysql_store_result()` 함수의 별도의 예제는 <부록 5>에 다루었다. “**mysql_free_result**” 함수나 기타 결과를 저장하는 함수에 의하여 저장된 자료를 해제(메모리 해제)하는 역할을 한다. 이 함수는 MYSQL에서 지원하는 함수이며 함수명은 MYSQL의 C API와 동일하다. 인자 `mysql_ptr`은 MYSQL result 포인터이며 `mysql_store_result()` 함수나 기타 결과를 저장하는 함수에 의하여 얻은 값이고 MYSQL result 포인터가 아니면 에러 처리되며 이미 해제된 MYSQL result 포인터일 때에는 경고 메시지를 출력하고 `result`에 -1의 값을 되돌린다. 항상 정수 1이 `result`에 돌려진다. `mysql_free_result()` 함수의 별도의 예제는 <부록 5>에서 다른 예제에 함께 보여준다.

```

for(i=0;i<9;i++)
    for(j=i+1;j<10;j++)
        if(a[i]>a[j])
            {
                tmp=a[i];
                a[i]=a[j];
                a[j]=tmp;
            }
    printf("\nsorting result:\n");
for(i=0;i<10;i++)
{
    printf("%12d",a[i]);
    printf("\n");
}

sprintf(query,"insert into ok2 (null, '%s' , '%s' , '%s' ,'%s',
'%s' , '%s', '%s','%s','%s','%s'),
one, two, three, four, five, six, seven, eight, nine, ten);

if(mysql_real_query(&mysql, query, strlen(query)))
{
    printf("success");
    printf("error on: %d\n %s\n", mysql_error(&mysql),
mysql_error(&mysql));
    exit(1);
}

```

<그림 7> 알고리즘 계산 및 데이터베이스 입력

<그림 7>에서는 읽어들인 값을 알고리즘의 계산순서에 의해 계산되는 순서를 설명하고 있다. 본 논문의 가장 중요한 부분으로 C알고리즘에서 계산된 결과값을 MYSQL데이터베이스 서버에 입력하기 위해 sprintf()문을 사용하였다. sprintf()문은 PHP3이상 버전에서도 사용되며 Perl, Visual Basic, XML, COBAR 등에도 사용되는 것으로 sprintf() 문의 주요 기능은 printf문과 마찬가지로 형식화된 출력을 한다. 단, printf문과 다른 점은 출력이 stdout(화면)으로 나가는 것이 아니고 문자열로 나간다. 첫번째 인수 buffer가 sprintf함수에 의한 출력을 저장할 장소이며 format이 서식문자열이다. printf와 마찬가지로 format 문자열내의 서식 개수와 일치하는 수의 뒤따르는 인수가 있어야 하며 대응되는 서식과 인수끼리는 데이터형이 같아야 한다. 서식 및 출력 형식 지정에 관한 세부적인 사항은 printf문과 동일하다. sprintf() 문에 의해 리턴되는 값은 문자열로 출력된 실제 문자 수를 리턴 하되 NULL 문자는 제외한다. 에러발생시는 EOF를 리턴한다.

```
gcc -o 실행파일명 파일이름 -lmysqlclient -L/usr/local/
mysql/lib/mysql -I/usr/local/mysql/include/mysql/
```

<그림 8> 컴파일 방법

<그림 8>의 컴파일에 있어 보통의 C언어의 컴파일과는 많은 차이를 보인다. 위 <그림 8>은 MYSQL의 API를 이용한 것으로 소켓 개념이 들어가 있으며 헤더파일 또한 MYSQL의 헤더 파일을 이용하므로 컴파일시 헤더파일의 경로를 같이 지정해주어야 한다.

-I는 헤더 파일의 경로를 지정하는 것이며 위의 헤더파일이 존재하지 않는 경우 파일이름.conc:2:mysql.h : “그런 파일이나 디렉토리 없음” 이라는 메시지를 출력한다. 이 경우에는 mysql.h의 경로를 찾을 수 없어서 나는

에러이므로 mysql.h가 어디 있는지 확인하고 -I 옵션으로 그 경로를 지정하면 된다. **“Id: can't open -mysqlclient; 그런 파일이나 디렉토리 없음”**이라는 메시지는 -L 옵션뒤에 붙은 라이브러리의 경로가 잘못되었기 때문이다. 이 경우에는 libmysqlclient.so 파일의 경로를 찾아서 확인하고 -L 옵션으로 그 경로를 지정하면 된다.

“Access defined for user: 'root@localhost' (Using password: YES)” 접근이 금지되었다는 메시지로서 사용자 ID를 잘못 입력하였거나 혹은 암호를 잘못 입력하였을 때 나오는 메시지이다. MYSQL의 mysql database의 user 테이블에서 찾아보고 그것을 참고로 하면 된다.

“./sql: error in loading shared libraries ibmysqlclient.so.6: cannot open shared object file: No such file or directory” 라는 메시지는 MYSQL의 라이브러리를 열지 못한다는 메시지이다. 컴파일 할 때 MYSQL의 동적 라이브러리를 사용하는데 동적 라이브러이므로 실행시에도 라이브러리가 필요하게 된다. libmysqlclient.so가 /usr/lib 혹은 /usr/lib/mysql 디렉터리에 존재 하지 않을 경우에 생기는 문제이다. 가장 간단한 해결법으로는 모든 MYSQL 라이브러리를 /usr/lib/나 /usr/local/lib 밑으로 복사하는 것인데 그렇게 좋은 방법은 아니다. 해결을 위한 두 가지 방법이 있는데 먼저 /etc/ld.so.conf 파일에 libmysqlclient.so가 있는 경로를 적어 준 후에 ldconfig 라는 명령을 프롬프트에서 실행하여 주면된다. 권한이 없는 일반 사용자라면 자신의 셸 환경 변수를 이용하면 된다. 각자의 셸에 맞게 LD_LIBRARY_PATH를 libmysqlclient.so가 있는 디렉터리로 지정을 해주면 된다. C 셸 사용자는 setenv LD_LIBRARY_PATH 경로명 본 셸 사용자 export LD_LIBRARY_PATH 경로명 이렇게 해주면 된다.

```
/tmp/ccPISyaz.o(.text+0xXX) :  
    undefined reference to 'mysql_real_connect'  
  
/tmp/ccPISyaz.o(.text_0xXc) :  
    undefined reference to 'mysql_error'  
  
/tmp/ccPISyaz.o(.text_0xXX) :  
    undefined reference to 'mysql_close'  
  
coollect2:ld returned 1 exit status.
```

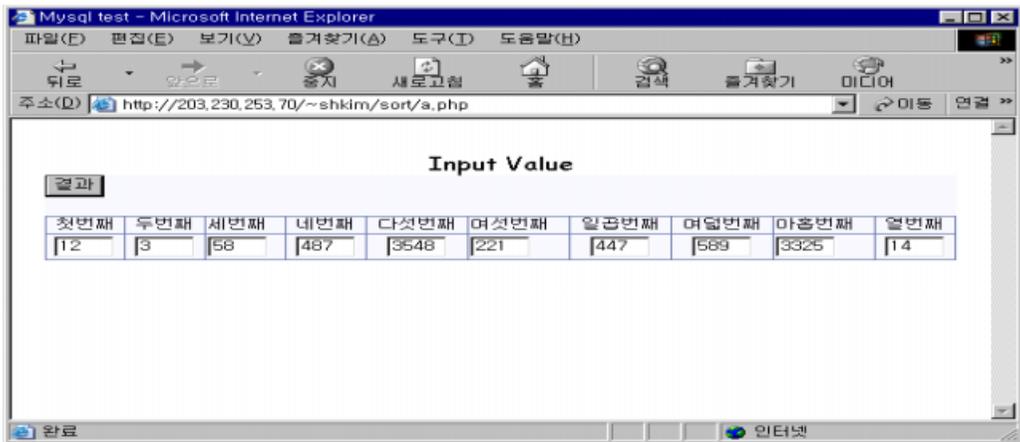
<그림 9> -lmysqlclient 옵션 에러 메시지

-lmysqlclient 옵션은 gcc 가 링크를 할 때 mysqlclient 라이브러리를 사용하라는 것으로 이 옵션이 빠질 경우 <그림 9>와 같은 메시지를 나타낸다.

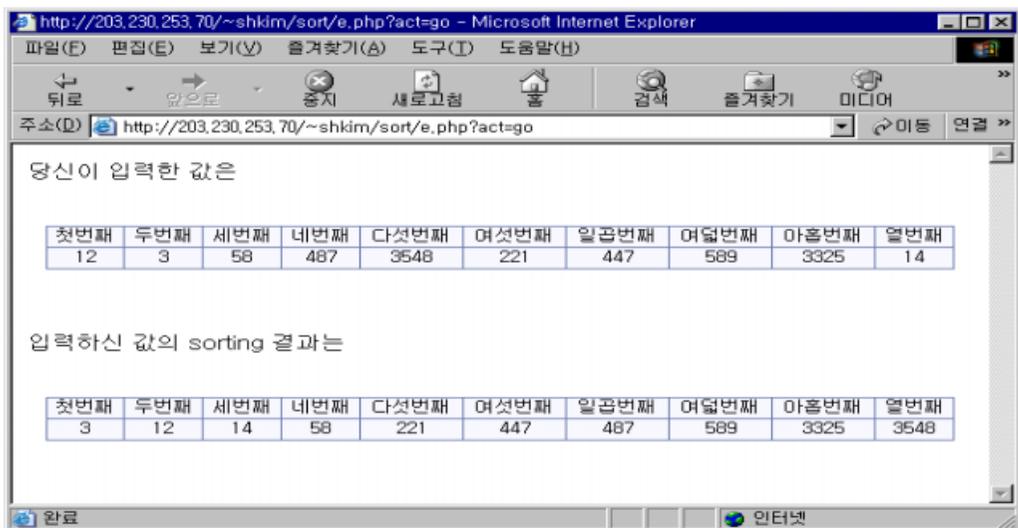
Ⅲ. 구현예제

본 장에서는 C API를 이용한 웹 기반의 Solver에 관한 2개의 구현예제를 소개하고자 한다.

3.1 소팅 프로그램



<그림 10> 자료의 입력 화면



<그림 11> 입력 값 확인과 결과 값 확인 화면

<그림 10>은 <부록1>에 수록되어 있는 input.php의 모습이다. 사용자가 소팅하고자 하는 자료 값을 입력하고 그 값을 전송해 주기전의 모습이다.

<그림 10>과 같이 입력하면 본 논문에서 개발한 C API를 통해 C 언어로 구성된 소팅 알고리즘을 수행한 후 그 결과 값을 웹상에서 <그림 11>과 같이 출력한다.

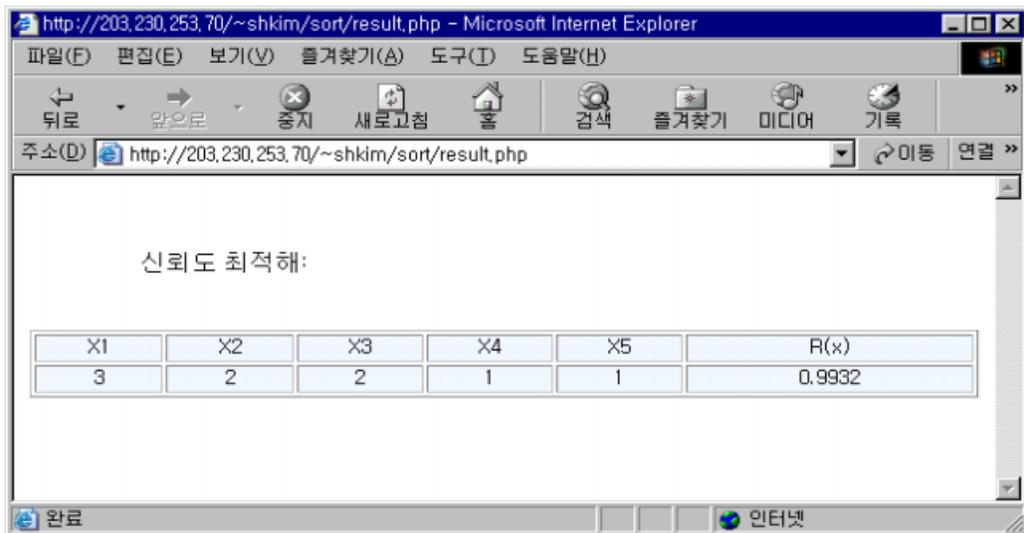
3.2 NRO(Network Reliability Optimizer) 알고리즘의 솔루션

본 논문에서 개발한 C API 프로그램을 이용하여 [18]에서 웹상에서 신뢰도 최적해를 제공해주지 못하는 단점을 극복하였다.

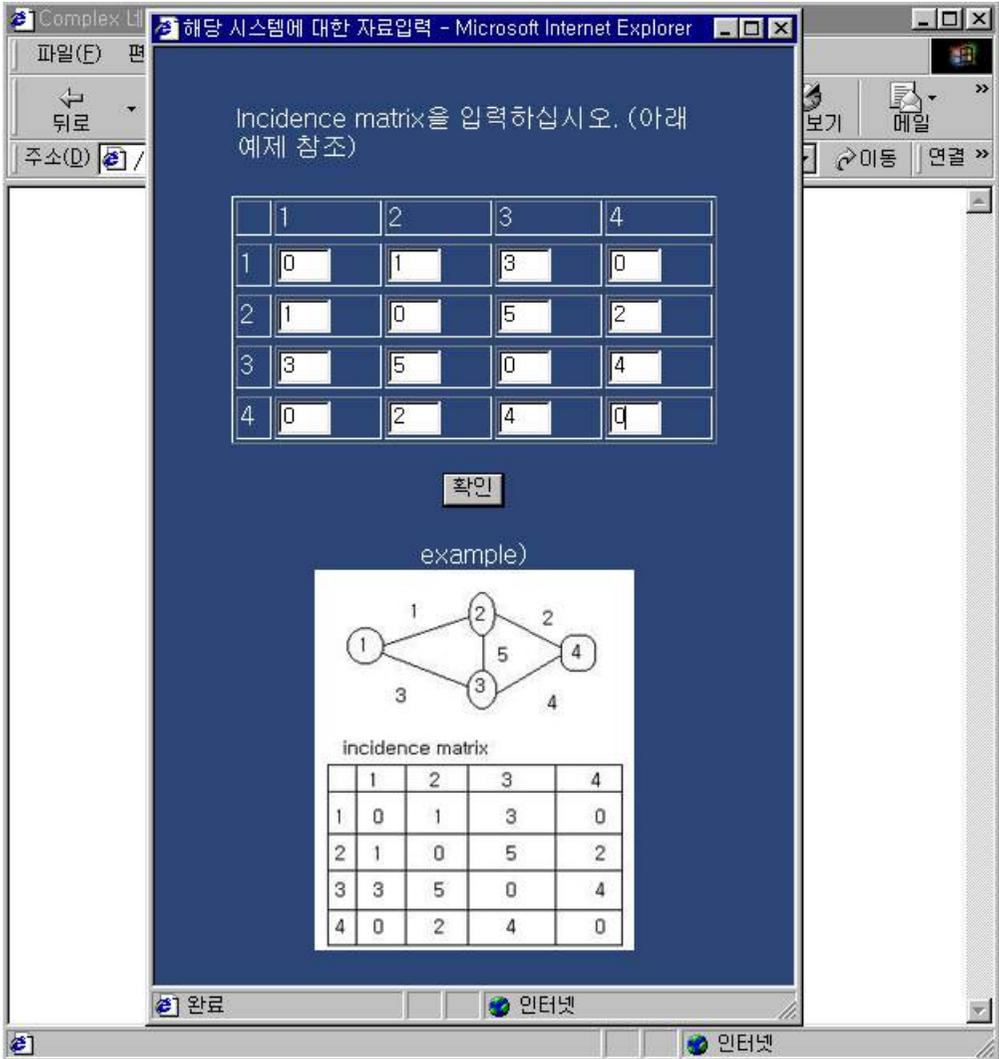
[18]에서처럼 <그림 13>의 incidence matrix의 자료를 입력하고 <표 2>와 같은 자료를 입력하면, 본 논문에서 개발한 C API 프로그램을 이용하여 <그림 12>와 같은 신뢰도 최적해가 웹상에서 편리하게 제공된다.

<i>i</i>	1	2	3	4	5
신뢰도	0.70	0.85	0.75	0.80	0.90
제약식	2	3	2	3	1

<표 2> 하부시스템의 부품의 자료 : $n = 5, m = 1$



<그림 12 > 신뢰도 최적해



<그림 13> incidence matrix 입력

IV. 결 론

본 논문에서는 서로 다른 운영체제간의 데이터의 흐름을 이용하여 웹상에서 알고리즘 솔루션을 구현하였다. 기존의 C 알고리즘은 개인 컴퓨터에서만 그 결과를 알아낼 수 있었으나 본 논문에서 개발한 C API는 웹상에서 솔루션을 제공함으로써 인터넷이 연결만 되어있다면 C 컴파일러 없이도 C 알고리즘의 솔루션을 구할 수 있다. 또한 서로 다른 운영체제에 존재하는 웹 서버와 데이터베이스 서버를 C API를 이용하여 서버간의 데이터를 상호 교환하게 하였다. CGI나 PHP의 문제점인 데이터의 전송방식(Post방식)으로 인한 해커들로부터의 해킹위험을 본 논문에서 개발한 C API가 제공하는 인터페이스에 의해 해커들로부터의 위험부담을 줄였다.

앞으로의 연구 과제는 C API를 WIN API처럼 사용자가 손쉽게 사용할 수 있도록 사용자 인터페이스를 개발하는 것이며, 웹상에서 입력되는 무한한 데이터를 데이터베이스에 입력할 수 있도록 MYSQL 데이터베이스에 삽입시키는 인터페이스의 개발이다.

[참 고 문 헌]

- [1] 최경락 “Advanced PHP programming Version4”, 삼각형플레스, 2001
- [2] David Lane, Hugh E.Williams “Web Database Applications with PHP & MySQL”, 대림, 2002
- [3] Kevin Tatroe, Rasmus Lerdorf “Programming PHP”, 열림기술, 2002
- [4] 허정수 “MySQL & Web DB연동”, 베스트북, 2001
- [5] 최완규 “Beginning PHP4”, 정보문화사, 2000
- [6] BLAKE SCHWENDIMAN “PHP4 DEVELOPER’S GUIDE”, 사이버출판사, 2002
- [7] DWIGHT&NILES “예제로 배우는 CGI 프로그래밍”, 인포북, 2001
- [8] WADE MAXFIED “MY SQL & PHP FROM SCRATCH”, 테크북, 2001
- [9] 정진호 “www.phpschool.com” New, Q&A board
- [10] www.database.sarang.net MySql(C API), study(windows2000서버에 php,mysql 설치), database자료(MySql function).
- [11] www.linux.co.kr Q&A(CGI, PHP, socket programming), 팀엔테크 (Fork 실행).
- [12] www.linux.sarang.net 질문과 답변(MySql, C API), 1원짜리 팀(PHP의 보안 문제해결책).
- [13] www.apache.kr.net CGIWRAP, 질문과답변.
- [15] www.terms.co.kr 인터넷, 컴퓨터관련 전자사전.
- [16] www.jointclub.net Programming in C, unix network 프로그래밍,
- [17] www.openmp.org Specifications-C/C++ version 2.0 (with change bars reflection from 1.0)
- [18] 원해연 “신뢰도 최적화 문제에 대한 Web-site Solver 의 개발에 관한 연구” 한국해양대학교 응용수학과대학원 석사논문, 2001

[부 록]

input.php

```
<html>
<head>
<title>Mysql test</title>
</head>

<body>
<center>
<br><font size=4 face='Comic Sans MS'>Input Value</font>

<table width=600 cellspacing=1 cellpadding=1 bgcolor=6b7bad>
<form method=post action=b.php>
<tr bgcolor=ghostwhite>
  <td align=center><font size=2>첫번째
  <td align=center><font size=2>두번째
  <td align=center><font size=2>세번째
  <td align=center><font size=2>네번째
  <td align=center><font size=2>다섯번째
  <td align=center><font size=2>여섯번째
  <td align=center><font size=2>일곱번째
  <td align=center><font size=2>여덟번째
  <td align=center><font size=2>아홉번째
  <td align=center><font size=2>열번째
</tr>

<tr bgcolor=ghostwhite>
  <td align=center><font size=2> <input type=text name=one size=4>
  <td align=center><font size=2><input type=text name=two size=4>
```

```

<td align=center><font size=2><input type=text name=three size=4>
<td align=center><font size=2><input type=text name=four size=4>
<td align=center><font size=2><input type=text name=five size=4>
<td align=center><font size=2><input type=text name=six size=4>
<td align=center><font size=2><input type=text name=seven size=4>
<td align=center><font size=2><input type=text name=eight size=4>
<td align=center><font size=2><input type=text name=nine size=4>
<td align=center><font size=2><input type=text name=ten size=4>
</tr>

<input type=submit value='결과'>

</form>
</body>
</html>

```

dbinsert.php

```

<?
$connect=mysql_connect("203.230.253.70","root","7878");
mysql_select_db("test1",$connect);
$query="delete from ok";
mysql_query($query,$connect);
$query="insert into ok(one,two,three,four,five,six,seven,eight,nine,ten)
values('$one','$two','$three','$four','$five','$six','$seven','$eight','$nine','$ten'
)";
mysql_query($query,$connect);

echo"
<script>
location= 'e.php';
</script>
";
?>

```

show.php

```
<?
if($act != "go")

{
    echo("<meta http-equiv='Refresh'
    content='0;URL=$PHP_SELF?act=go'>");
}

$connect=mysql_connect("localhost","root","7878");
mysql_select_db("test1",$connect);
$query="select *from ok";
$result=mysql_query($query,$connect);
$row=mysql_fetch_row($result);

echo"당신이 입력한 값은<br><br><br>";

echo"
<table width=600 align=center bgcolor=6b7bad cellpadding=1>

<tr bgcolor=ghostwhite>
    <td align=center><font size=2>첫번째
    <td align=center><font size=2>두번째
    <td align=center><font size=2>세번째
    <td align=center><font size=2>네번째
    <td align=center><font size=2>다섯번째
    <td align=center><font size=2>여섯번째
    <td align=center><font size=2>일곱번째
    <td align=center><font size=2>여덟번째
    <td align=center><font size=2>아홉번째
    <td align=center><font size=2>열번째
</tr>
```

```

<tr bgcolor=ghostwhite>
  <td align=center><font size=2>$row[0]
  <td align=center><font size=2>$row[1]
  <td align=center><font size=2>$row[2]
  <td align=center><font size=2>$row[3]
  <td align=center><font size=2>$row[4]
  <td align=center><font size=2>$row[5]
  <td align=center><font size=2>$row[6]
  <td align=center><font size=2>$row[7]
  <td align=center><font size=2>$row[8]
  <td align=center><font size=2>$row[9]
</tr>
</table><br><br><br>";

```

```

$list=file("/home/shkim/public_html/sort/data.txt");

```

```

echo"입력하신 값의 sorting 결과는 <br><br><br>";

```

```

echo"

```

```

<table width=600 align=center bgcolor=6b7bad cellspacing=1 cellpadding=1>

```

```

<tr bgcolor=ghostwhite>
  <td align=center><font size=2>첫 번째
  <td align=center><font size=2>두 번째
  <td align=center><font size=2>세 번째
  <td align=center><font size=2>네 번째
  <td align=center><font size=2>다섯 번째
  <td align=center><font size=2>여섯 번째
  <td align=center><font size=2>일곱 번째
  <td align=center><font size=2>여덟 번째
  <td align=center><font size=2>아홉 번째
  <td align=center><font size=2>열 번째
</tr>

```

```

<tr bgcolor=ghostwhite>

```

```

";
for($i=0;$i< count($list);$i++)

{
    echo "<td align=center><font size=2>$list[$i]";

}
?>

```

<부록 1> 소팅 프로그램에 사용된 PHP 소스

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <sys/wait.h>
#define WAIT_TIME 0.2
#define EXEC_NAME "/home/shkim/public_html/sort/call"

int load_ps ();
main ( void )
{
    int c_pid, stat;

    if ( fork () )
    {
        printf ( "\n watching...[%s]\n", EXEC_NAME );
        exit ( 1 );
    }
    c_pid = load_ps ( 0 );

```

```

while ( 1 )
{
    if ( waitpid ( c_pid, &stat, WNOHANG ) != 0 )
        c_pid = load_ps ( 1 );
    sleep ( WAIT_TIME );
}
}
int load_ps ( int method )
{
    int c_pid;
    char str[256];
    c_pid = fork ();

    if ( c_pid == 0 )
    {
        execl ( EXEC_NAME, (char *)0 );
        exit ( 1 );
    }

    if (method == 0)
    {
        sprintf (str, "Loading...[%s]: PID:%d", EXEC_NAME, c_pid );
    }
    else
        sprintf(str, "Reloading...[%s]: PID:%d", EXEC_NAME, c_pid );

    return c_pid;
}

```

```

mysql= mysql_connect(NULLL, myuser, mypass);
// localhost를 연결한다.

if(mysql==MYSQL_ERR)
{
    printf("MySQL 서버 연결 에러!\n");
    return;
}
mysql_select_db(mysql, "no_db");
// no_db 의 데이터베이스가 없다고 가정한다.
err_code= mysql_errno(mysql);
mysql_close(mysql);
// MySQL 서버와의 접속을 폐쇄한다.

printf("error code= ", err_code, "\n")

----- [ 출력결과 ] -----
error code= 1049

```

<부록 3> mysql_errno() 함수의 사용 예

```

mysql= mysql_connect(NULLL, myuser, mypass);
//localhost를 연결한다.

if(mysql!=MYSQL_ERR)
{
    printf("MySQL 서버 연결 에러!\n");
    return;
}
rt = mysql_select_db(mysql, "no_db");
// no_db의 데이터베이스가 없다고 가정한다.
errmsg= mysql_error(mysql);
mysql_close(mysql);
// MySQL 서버와의 접속을 폐쇄한다.

printf("rt= %d  errmsg=%s\n", rt,errmsg);

----- [ 출력결과 ] -----
rt= -1  errmsg= 데이터베이스 'no_db'는 알수 없음

```

<부록 4> mysql_error() 함수의 사용 예

```

// 다음 프로그램은 MYSQL 서버가 정상 작동하지 않거나 또는
// welhelp라는 데이터베이스가 없거나 addr 이라는 테이블이 없으면
// 에러가 발생한다.

mysql= mysql_connect(NULLL, myuser,mypass);
// localhost에 연결한다.

if(mysql==MYSQL_ERR)
{
    printf("MySQL 서버 연결 에러!\n");
    return(-1);
}
if(mysql_select_db(mysql, "welhelp")==MYSQL_ERR)
// welhelp database 선택한다.
{
    printf("welhelp database 선택 에러!\n");
    return(-1);
}
query= "select * from addr";
// addr 테이블의 모든 항목을 선택한다.
if(mysql_query(mysql, query)==MYSQL_ERR)
{
    printf("query error!!\n"); return(-1);
}
myres= mysql_store_result(mysql);
// 추출된 자료를 저장하고 포인터를 돌려준다.
row_count= mysql_num_rows(myres);
// 추출된 자료의 레코드 개수를 얻는다.

```

```

    field_count= mysql_num_fields(myres);
// 추출된 자료의 항목 개수를 얻는다.
    mysql_data_seek(myres, 1);
// 2번째 레코드로 이동한다.
    row_count-= 1;
// 추출된 레코드의 개수를 조정한다.

    for(i=0; i<row_count; i++)
    {
        row= mysql_fetch_row(myres);
// 저장된 자료에서 한 레코드를 읽는다.
        if( isarray(row) )
// 또는 if(row!=MYSQL_ERR)를 사용한다.
        for(j=0; j<field_count; j++) printf("\t", row[j]);
// 항목을 출력한다.
        printf();
    }

    mysql_free_result(myres);
// 임시 저장된 자료를 해제(free)한다.
    mysql_close(mysql);
// MYSQL 서버와의 접속을 해제한다.

```

```

----- [ 출력결과 ] -----
    한여름  35      00시 00동 456번지
    서지연  20      A시  B동 789번지

```

<부록 5> mysql_store_result() 사용예

```

a=sprintf("decimal= %d octal= %o hex= %x real= %f",123,123,123,123.4);
b=sprintf("(%.3ld)(%#o)(%#X)(%09.4b)", 1234567,123,123,96);
c=sprintf("(%.10.2f)(%.10.1f)(%010.2f)(%-10.2f)", 5.67,5.67,5.67,5.67);
d=sprintf("%.3ld", 123456789);
print("    ", a, "\n    ", b, "\n    ", c, "\n    ", d, "\n");

```

```

----- [ 출력결과 ] -----
decimal= 123 octal= 173 hex= 7b real= 123.400000
(1,234,567)(0173)(0X7B)(0110 0000)
(      5.67)(      5.7)(0000005.67)(5.67      )
123,456,789

```

<부록 6> sprintf() 문의 예