

Web상에서 원격관리가 가능한 실시간 차트 시스템의 설계 및 구현

박철우, 김성, 박규석
경남대학교 컴퓨터공학과

Design and Implementation of A Remote controllable Real-Time Chart System in The Web

Choi-woo Park, Sung Kim, Kyoo-seok Park
Dept. of Computer Engineering, Kyungnam University

요약

최근 들어 급속한 인터넷 사이트와 이용자들의 증가로 인해 많은 정보 제공 사이트들은 이용자들의 요구에 상응하는 정보를 제공하기 위하여 보다 이해하기 쉬운 그림, 도표, 차트 등의 그래픽으로 처리된 정보들을 생성하고 관리한다. 본 연구에서는 이용자들의 요구에 부합되며 인터넷 상에서 원격 관리가 가능한 실시간 차트서버와 차트 클라이언트, Java application 내부에서 처리되는 java chart component 등을 제안하고 설계 및 구현한다.

1. 서론

근래에 와서 인터넷의 확산에 따라 많은 회사 및 개인 홈페이지들이 생겨나고 있으나, 초기에는 텍스트 위주의 정보교환이나 의사 전달과 같은 커뮤니케이션 기능이 주된 역할을 하였기 때문에 문자위주의 정보가 대부분이었다.

정보 이용자들은 점차 숫자위주의 통계 정보 보다 빠른 이해가 가능한 그림, 차트, 도표 등의 2차 적으로 생산된 정보를 선호하게 되었으며, 이미 이와 같은 실시간 정보를 제공하는 사이트가 많이 생기고 있다.

기존의 실시간 차트의 경우 특정용도로 작성되어 누적되는 통계정보를 응용프로그램이나 web상의 CGI(Common Gateway Interface)프로그램에 의해 animation image를 생성하는 방법을 사용하므로, browser에서 refresh시키는 방법으로 정보갱신이 이루어지고 있기 때문에 실시간 정보의 제공 시 접속 때마다 server에서 차트를 작성하게 되어 server의 효율이 떨어지며, 또한 여러 가지 범용 그래프를 많이 쓰는 경우 소프트웨어 재사용성이 떨어지고, 대규모 data를 취급할 때 CGI를 통한 원격 관리가 어려운 단점이 있다. [1][2][3]

본 논문에서는 인터넷을 이용하여 사용자의 요구에 부응할 수 있는 java 기술을 사용한 범용 차트 component와 차트 구성에 필요한 모든 설정 및 data를 객체화 하여 전송하는 방식의 실시간 차트 시스템을 제안한다.

java의 플랫폼 독립적인 특성을 이용하여 java application에 간단히 삽입할 수 있는 component 형태로 설계 및 구현하였으며, java RMI(Remote Method Invocation)기술을 사용하여 차트 server와 client 간의 통신상의 보안 문제를 해결하였고, client와 server manager는 browser를 통해 실행되는 java applet 형태로 구성하여 CGI원격관리기술 보다 다양한 형태의 접근이 가능한 실시간 차트정보 시스템을 설계 및 구현하였다.

제안 시스템은 지속적인 갱신이 일어나는 증권정보, 득표현황, 정보처리상황 등의 실시간 정보를 보다 다양한 형태로 신속하게 전달할 수 있으며, 관리자는 원격지에서 신속히 최신 자료의 입력 및 기타 차트형태 변경 등을 할 수 있어, 정보이용자들의 요구에 좀더 질 높은 정보를 제공하게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문

에서 사용된 기법에 관련된 연구를 살펴보고, 3장에서는 원격 관리가 가능한 실시간 차트 시스템을 설계하고, 4장에서는 구현 및 평가, 5장에서는 결론을 내린다.

2. 관련연구

2.1 Chart 구현 기술

업무용으로 가장 많이 활용되는 막대차트, 꺾은선차트, 원차트를 작성하는 기본적인 알고리즘은 다음과 같다.

데이터의 표준화: 데이터를 화면상에 적절한 좌표로 변환하는 작업으로, 각 축에 대하여 식1과 같이 표준화 비율을 계산한 다음, 식2에서 데이터 값에 곱해주어 정확한 좌표 범위를 계산한다.

표준화 비율; 화면좌표높이 / (최대값-최소값) (식1)
 표준화 데이터; 실제의 데이터 * 표준화 비율 (식2)

차트 생성: 표준화된 데이터를 받아, 서로 다른 데이터 군을 반복적으로 입력하여 해당 좌표에 출력한다. 막대차트의 경우 하나의 군에 두개나 그 이상의 막대가 그려지는 경우, 계산된 표준화 데이터를 화면에 균형 있게 표시하기 위해 다시 여백과 막대와 막대사이의 공간을 지정하는 인자로 나눈다. 또한 데이터 군을 구별하기 위한 색상을 표시해야 한다.

눈금표시: 차트 크기에 따라 2~5등분하는 주단위를 계산하고, 주단위의 5등분하는 보조단위를 계산한다. 최대·최소값의 최상위 자리수를 단위를 기준으로 5등분 하여 주단위를 계산한다.

값 표시: 각 막대를 그린 후 이미 계산된 데이터 값에 따라 양수인 경우 막대 위쪽(+a)에 음수인 경우 막대 아래쪽(-a)에 데이터 값을 넣는다.

범례표시: 여러 데이터 군이 동시에 표시되는 경우 각 막대에 색과 나타내고자 하는 데이터의 항목을 차트 밖의 영역에 표시하거나, 데이터를 가리지 않는 범위에서 차트 안에 표시한다.[4]

2.2 RMI(Remote Method Invocation)

RMI를 이용하여 클라이언트 컴퓨터에서 실행되는 프로그램이 원격서버 컴퓨터에 위치한 객체의 메소드를 호출하여 서로 원하는 데이터를 전송할 수 있다.

그림1의 구조도에서 stub은 원격 객체를 대변하는 클라이언트 측의 대리자로서 원격 객체가 구현하는

인터페이스를 정의, 프로그램 내에서 지역적인 다른 객체처럼 참조되어 서버측 객체와 접속을 유지한다.

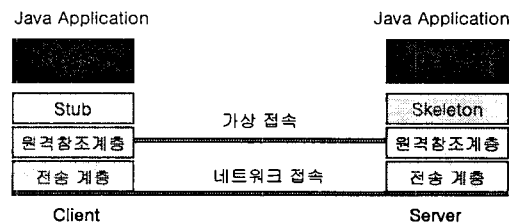
원격 객체가 호출 되면 stub은 파라미터들을 직렬화 시킨 후 정렬 스트림(marshal stream)형태로 원격 참조계층으로 전달한 후, 원격 메소드 수행이 끝나면 다시 원격참조계층으로 부터 정렬스트림을 받아 역직렬화를 통해 값을 리턴한다.

정렬 스트림은 서버측의 RRL과 클라이언트측의 RRL과의 통신상에 사용된다.

skeleton은 서버측 RRL과 인터페이스하는 서버측의 생성체로서 클라이언트 측 RRL에게 메소드 호출 요청을 받으면 서버측 RRL은 원격 메소드로 전송되어 온 파라미터들을 복원시킨 후 skeleton이 서버측에 구현된 실제 객체를 호출한다. 원격 객체에서 리턴 값을 받아 다시 정렬 스트림으로 정렬 후 서버측의 RRL에 전달한다.

원격참조계층은 특정한 stub 또는 skeleton 모델에 종속적이지 않은 독립 참조 프로토콜을 관리하여 다른 두 계층에 영향을 주지 않고 RRL교체가 가능하도록 하며, 하위 운송 인터페이스와 작용하여 stub과 skeleton 계층에 스트림을 제공한다.

전송계층은 TCP/IP 및 기타 프로토콜을 이용하여 클라이언트와 서버사이의 접속을 만들고, 기존의 접속을 관리 및 자신의 주소 공간에 상주하는 원격 객체를 처리한다. 전송계층이 클라이언트측 RRL에서 요청을 접수하면 원격 객체에 적합한 RMI서버를 검색하여, 그 서버에 대한 소켓을 만들고 이 접속을 클라이언트 측 RRL에 전달한후 내부 테이블에 그 원격 객체의 참조를 추가시켜 연결을 확립한다.[5]

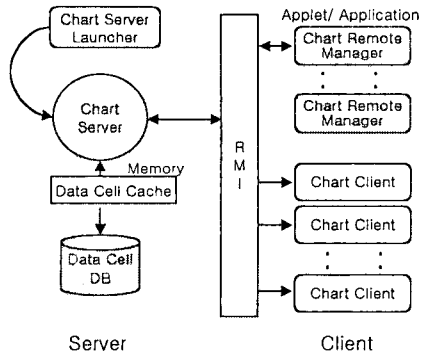


[그림 1] RMI 계층 구조

3. 시스템 설계

3.1 시스템 구성

제안하는 실시간 차트 시스템의 구성도는 그림2와 같다.

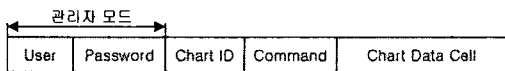


[그림 2] 실시간 Chart system 구성도

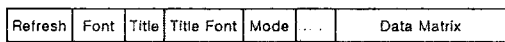
그림2에서 나타낸 것과 같이 서버를 구동하는 chart server launcher 및 각 차트의 목록 및 data를 관리하며 client에 data를 제공하는 chart server, 각종 차트를 관리하는 chart remote manager, server로부터 data를 받아오는 chart client, 그리고, 차트 data를 이용하여 차트를 생성하는 java chart component로 구성되어 있다.

3.2 Remote 객체 구조 및 Chart data cell 구조

각각의 차트에 따라 그림3과 같은 remote 객체를 원격 메소드 호출을 통해 생성, 삭제, 갱신하게 되며 각 remote 객체는 그림4와 같은 chart data cell을 가지게 되고, cell은 차트를 그리기 위한 data, title, font, color, 차트종류 등의 많은 필드를 가진다.



[그림 3] Remote object structure

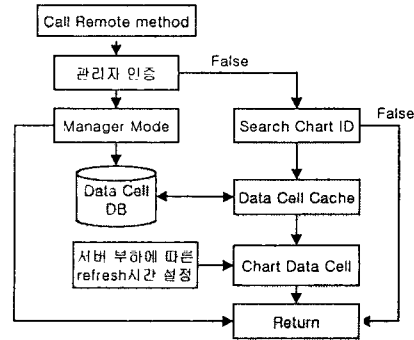


[그림 4] Chart data cell structure

3.3 Chart Server

다수의 chart data cell을 관리하고, client에 data cell를 제공해주는 chart server는 그림5와 같다. 원격 메소드 호출을 통한 데이터접근이 이루어지면 관리자는 인증과정을 거쳐 관리자 모드로 전환하여 data를 관리하게 된다. 일반 chart client는 인증과정에서 인식하게 되고, remote 객체의 chart ID에 따라 DB로부터 각 차트정보를 리턴 한다. 이때, 서버 내에 호출중인 client 수에 따라 refresh time을 조절하여 서버의 client 부하를 줄이고, 메모리의 일정부

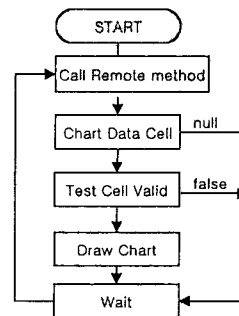
분을 cache공간으로 사용하여 DB access를 줄인다.



[그림 5] Chart server의 흐름도

3.4 Chart Client

서버에 자신의 chart ID의 data를 원격 메소드 호출을 통하여 리턴 받아, chart component를 이용하여 java application이나 java applet에 실질적인 차트를 출력하는 module이며 그림6과 같이 구성된다. 입력된 chart ID는 원격 메소드 호출의 매개변수로 사용되며, chart ID에 따른 data cell을 받게 된다. 리턴된 값이 null이면 chart ID가 없는 번호이거나 원격 메소드 호출시 문제가 생긴 것이며, 정상적으로 data를 받을 경우 chart component에 넘겨 차트를 완성한다. 각 chart는 refresh time을 가지고 있으며, 서버로부터 데이터리턴 시 마다 매번 설정되어 다음에 접속할 시점을 관리하게 된다.



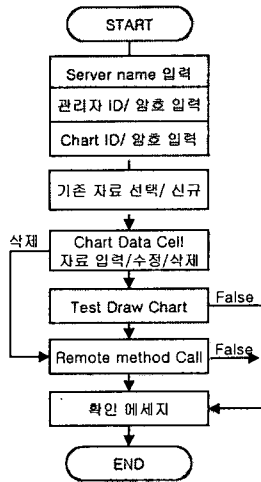
[그림 6] Chart client의 흐름도.

3.5 Chart Remote Manager

browser 또는 java application에서 생성하고자 하는 차트의 자료를 입력, 삭제, 수정하는 역할을 한다.

그림7에서와 같이 server name, 관리자 계정 및 암호를 입력한 후 미리보기 메소드를 실행시켜, 작성된 차트를 예상할 수 있고, 데이터유형을 최소화 할

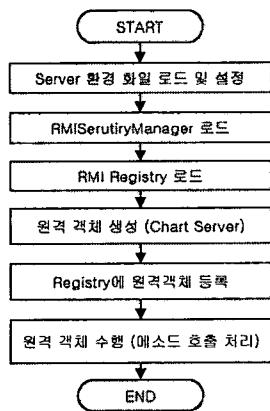
수 있다. 미리보기가 성공하면, 전송 함수를 실행할 수 있으며, 원격 메소드 호출을 통하여 data cell DB를 갱신하게 된다.



[그림 7] Chart remote manager의 흐름도.

3.7 Chart Server Launcher

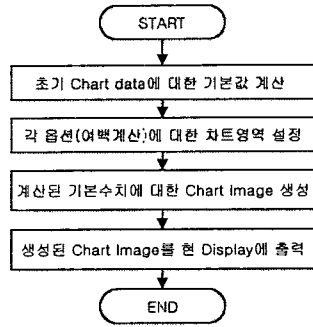
서버를 구동하기 위해 원격 객체를 등록 시키는 역할의 module로서 RMI를 사용한 프로그램의 전형적인 형태이다. 그림8에서와 같이 server에 필요한 환경변수파일을 읽어 IP, server name 등의 초기 정보를 설정한다. 그 후, RMI SerutiryManager와 RMI Registry를 로드 하게 되고, 차트서버의 역할을 하는 원격 객체를 생성하고 등록하게 된다. 등록된 원격 객체가 종료 할 때까지 chart server launcher는 계속 대기 상태가 된다.



[그림 8] Chart server launcher의 흐름도.

3.6 Java Chart Component

chart 데이터cell 클래스를 받아 실제적으로 화면에 차트를 그리는 component:로서, 그림9에서와 같이 4 단계로 구현된다.



[그림 9] Chart component의 흐름도.

먼저 chart data을 이용하여 차트 영역의 최대·최소 data값, 주 단위 계산 등의 기초 자료를 계산한다. 다음으로 현재의 최대 차트영역의 정보로부터 각종 여백(제목, 계열값등)을 고려하여 실제 차트가 표시될 영역을 사용될 차트 영역으로 계산한다.

그 다음은 계산된 정보를 이용하여 각 차트영역에 대한 데이터비율을 이용하여 실제 차트를 생성한다.

차트가 갱신될 때 화면의 번쩍임을 막기 위하여 버퍼 이미지에 먼저 모든 작업을 완료한 후 이미지를 화면에 복사하는 방법으로 출력한다.

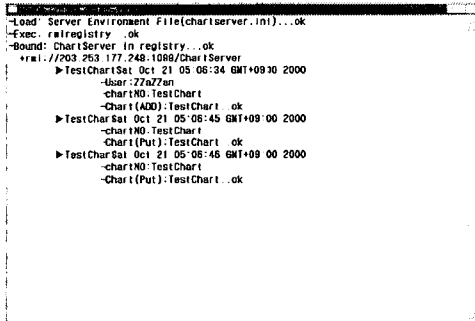
4. 구현 및 평가

본 시스템은 플랫폼 독립적인 java application의 형태로 application 및 applet 코드 내에 삽입되어 어떠한 환경이든지 자유롭게 이용 가능하도록 설계하였다.

개발 도구는 JDK 1.2.2 SE를 사용하였고, 윈도우 98 환경에서 구현하였다.

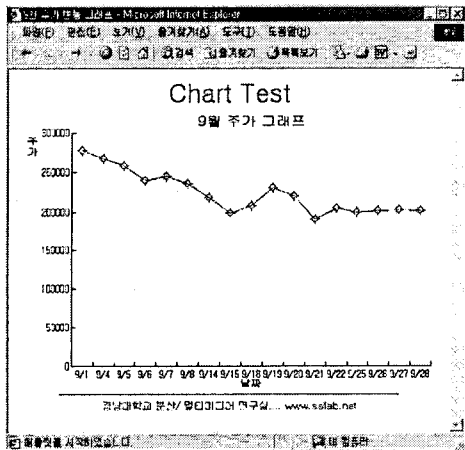
remote manager 및 chart client는 HTML내에 삽입되는 java applet을 이용하여 browser를 통한 데이터조작이 간편하고, 모든 java application내에 삽입될 수 있어, 소프트웨어 재사용성이 높다.

그림10은 launcher에 의하여 구동되는 chart server의 초기 실행화면을 나타낸 것이다.

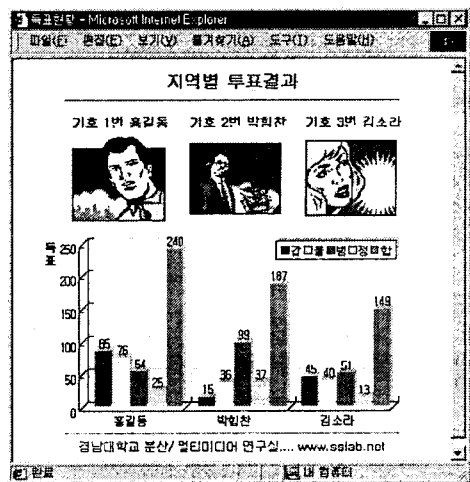


[그림 10] Chart server 초기 실행화면

그림11과 그림12는 이용자들이 최종적으로 접하게 되는 차트 client 화면의 한 예를 보인 것이다.

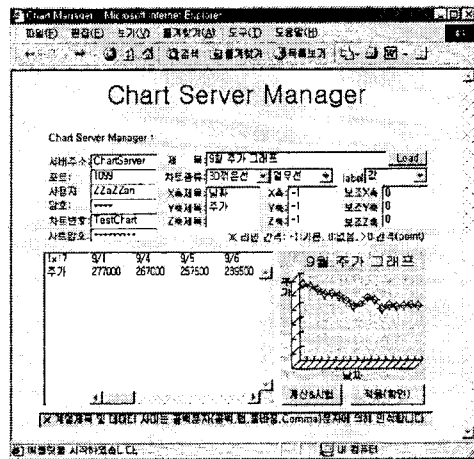


[그림 11] Chart server 초기 실행화면



[그림 12] Chart server 초기 실행화면

그림13은 web상에서 차트 정보를 관리하는 화면의 한 종류이며, 내부에 chart component를 두어 data를 적용한 결과를 미리 확인할 수 있다.



[그림 13] Chart server manager

기존의 web상에 CGI를 통한 animation 이미지 방식의 장점은 순간에 갱신되는 data가 적을 경우 이미지가 미리 생성되어 같은 이미지를 모든 client가 접속하여 가져가게 되고 browser는 단순히 image file만 출력하면 되므로 싼 비용으로 높은 효과를 낼 수 있지만, 순간에 갱신되는 data가 많을 경우 data에 따라 매번 서버에서 차트가 생성되어 server의 효율이 급격히 떨어지게 된다.

본 시스템은 image file 대신 차트를 작성할 수 있는 최소의 data만을 client에 전송하게 되고, 클라이언트의 java applet에 의하여 차트를 생성하게 되어 server의 부하를 줄이고, browser의 크기가 변경될 경우 동적으로 차트를 다시 그리는 벡터방식의 처리가 가능하므로 특정부분 확대·축소 등의 여러 가지 기능이 추가될 수 있어서, 기존의 bitmap 방식보다 정교한 차트를 생성할 수 있다.

또한, 모든 java application은 module화 되어 최소 작업으로 차트 시스템을 구축할 수 있도록 하여 기존의 CGI로 개발할 때 보다 시스템 개발시간을 단축할 수 있어 재사용성이 높으며, java RMI 기술을 사용하여 객체전달을 하므로 기존의 socket 통신보다 강력한 보안이 이루어질 수 있다.

또한, java application에 의한 차트 server의 data 갱신 및 직접적인 DBMS 접근이 가능하므로 대용량의 차트를 관리할 경우 기존의 CGI의 server에 접근

하는 방식으로 관리할 수 없는 부분까지 관리가능 하다.

5. 결론

제안된 시스템은 기존의 CGI방법에 의한 bitmap 차트의 실시간 처리에 대한 단점을 보완하기 위하여 java RMI기술을 사용하여 server application과 client application 간의 최소 정보만을 주기적으로 통신하도록하여 server의 부하율 및 client 갱신시간을 자유롭게 설정할 수 있게 함으로써 보다 효율적인 실시간 차트정보 시스템을 구현할 수 있게 하였다.

앞으로의 연구 방향은 보다 다양한 옵션들의 차트를 구현하는 것과, 쓰이지 않은 차트영역을 삭제하여 보다 쉽게 이해할 수 있는 차트를 생성하며, server와 client간의 효율적인 통신으로 network 및 server 부하를 최소화 할 수 있는 실시간 시스템을 구축하는 것이다.

[참고문헌]

- [1] "Using CGI", pp.439-507, Jeffy Dwight and Michael Erwin, Que, 1996. 10.
- [2] HTML을 지원하는 라이브러리를 이용한 웹 문서 생성 시스템의 설계 및 구현, v. 3, n. 4 pp.375-383, 김현중, 정보과학회논문지, 1997. 08.
- [3] "PHP 웹프로그래밍 가이드" pp.783-826, 이승혁, 마이트Press, 2000. 5.
- [4] "C프로그램 이렇게 잔다", pp.691-733, 황희용, 교학사, 1993. 3.
- [5] "Inside Secrets Java 2 Developer's Handbook" pp.648-693, Philip Heller & Simon Roberts, 삼각형프레스, 1999. 5.