

## 웹 브라우저 기반의 네트워크 컴퓨터 아키텍처 설계

송민규\*

### 요약

컴퓨터 및 네트워크 기술의 발전에 힘입어 이전에는 불가능하던 컴퓨팅 패러다임이 현실화되고 있다. 이른바, 네트워크 컴퓨터로 지칭되는 컴퓨팅 방식에서 컴퓨터 시스템 및 리소스는 이전의 그것과는 비교할 수 없을 정도로 확장 가능하다[1]. 네트워크에 접속되어 있는 여러 컴퓨터 시스템들의 무리는 하나의 거대한 가상 컴퓨터에 해당하며, 사용자는 네트워크 상에 분산된 방대한 리소스를 효율적으로 활용할 수 있다. 네트워크 상의 시스템에서 필요한 작업을 수행하고 리소스를 활용할 수 있기에 클라이언트를 경량화하는 것이 가능하다. 이러한 클라이언트는 쉘-클라이언트로서 불리고 있는데 네트워크 상의 시스템과 리소스를 활용함으로써 필요한 작업을 수행한다는 점에서 네트워크 컴퓨터와도 일맥상통한다고 할 수 있다[2]. 네트워크 컴퓨터 패러다임에서 네트워크는 애플리케이션 전달을 위한 기반 플랫폼으로서 작동하며 클라이언트 상에서 원격의 서버 시스템에 대한 접근 및 애플리케이션 실행하는 것을 가능하게 한다[3]. 이를 구현하는 기술에는 여러 가지가 있지만 본 논문에서는 웹, X 윈도우, Pyjamas 기술을 활용할 것이다. 이를 통하여 사용자는 마치 로컬에서와 같이 서버 시스템의 애플리케이션을 활용할 수 있으며 보안 및 유지관리에 있어 효율을 극대화할 수 있다. 본 논문에서는 이러한 네트워크 컴퓨터 구현을 위한 기반 환경으로 웹 브라우저를 지정하였으며, 클라이언트에서 웹 브라우저를 통하여 서버 상의 애플리케이션을 효율적으로 활용할 수 있는 아키텍처를 설계하고자 한다.

## Design of the Network Computer Architecture based on a Web Browser

Min-Gyu Song\*

### Abstract

By the advancement of computer & network technology, the paradigm of 'Network Computer' has been realizing'. In what is called network computer, computer system and computing resource is incomparably seem to be expanded compared with conventional network technology[1]. Network connected computer system constitute a massive virtual computer, it is possible for people to use an enormous amount of computing resource distributed widely through the network. It is also possible that we make client lightweight by the use of computer system & all shared computing resources on the network in our computer processing and we call this type of client system as thin-client. Thin-client and network computer are on and the same network paradigm in that both paradigm featuring the active use of computer system and resource on the network[2]. In network computer paradigm, network itself is regarded as a basic platform for the transfer of application, so it is possible that client access remote server system to run remote applications through the network[3]. In this paper, we propose the system architecture for the implementation of network computer by the use of Web browser, X window system and Pyjamas. By the use of network computer proposed in this paper, it is possible for people to run application on the server system as if he run local application, and it is expected to improve the security and maintenance efficiency.

**Keywords :** Network Computer, Server Based Computing, Thin-Client, Web Browser, X window System, Pyjamas

※ 제일저자(First Author) : 송민규

접수일:2008년 11월 20일, 완료일:2006년 12월 26일

\* 한국천문연구원 전파천문연구부

[mksong@kasi.re.kr](mailto:mksong@kasi.re.kr)

## 1. 서론

지난 수 십여 년 간 컴퓨터 및 네트워크 기술은 비약적으로 발전하였다. 1960년대 메인프레임을 기반으로 발전한 컴퓨터는 80년대의 PC(Personal Computer)를 거쳐 이제는 썬-클라이언트 형태로 진화해가고 있다[2]. 이처럼 컴퓨터의 발전 형태에 크나큰 영향을 미친 요소로 네트워크 기술을 언급하지 않을 수 없는데, PC와 썬-클라이언트에서 각각 클라이언트/서버, 서버 기반의 컴퓨팅 적용되어 컴퓨터 시스템이 구현되었다. 향후 IT 기술 발전에 있어 컴퓨터와 네트워크의 상호보완적 관계는 더욱 긴밀해질 것이며 그 최종적인 형태는 네트워크와 컴퓨터가 합쳐진 형태의 네트워크 컴퓨터가 될 것으로 전망되고 있다[1]. 네트워크 컴퓨터 패러다임에서 네트워크는 애플리케이션과 데이터가 전달되는 하나의 플랫폼으로 발전하고 있다. 원격의 서버 상에 위치한 애플리케이션을 로컬에서와 같이 실행할 수 있기 때문에 클라이언트는 고사양일 필요가 없으며 날씬한(thin) 형태로 변경하는 것이 가능하다[4]. 네트워크 컴퓨터 패러다임에서 기존의 단일 시스템, 독립된 물리적 시스템으로 한정되던 작업 영역은 네트워크 영역으로 확장된다. 이전에 한정된 용량, 한정된 성능의 단일 시스템에 국한되던 사용자 시스템은 네트워크를 기반으로 하나의 가상 컴퓨터로 진화해가고 있으며 각 사용자는 네트워크 투명하게, 필요로 하는 애플리케이션과 데이터를 효율적으로 활용하는 것이 가능하다. 일명 클라우드 컴퓨팅(Cloud Computing)이라 불리는 이러한 방법을 통하여 사용자는 언제 어디서라도 네트워크에 접속되어 있다면 서버 시스템의 애플리케이션을 효율적으로 활용할 수 있으며 이를 통하여 작업 효율의 극대화를 얻을 수 있다[5].

본 논문에서는 이러한 네트워크 상의 애플리케이션, 데이터를 효율적으로 활용할 수 있는 네트워크 컴퓨터의 구현 방안에 대해 논의해보기로 한다. 네트워크 컴퓨터에서 서버는 사용자가 애플리케이션에 효율적으로 접근, 활용할 수 있도록 인터페이스를 제공하여야 한다. 본 논문에서는 이러한 인터페이스 구현을 위한 방안으로 웹 서버와 X 윈도우 그리고 AJAX 애플리케이션

개발을 위한 파이썬 기반의 프레임워크인 Pyjamas를 활용하였다. 더불어 서버 상의 애플리케이션을 원격에서 보다 효율적으로 활용할 수 있도록 X 윈도우 기술을 웹 브라우저 상에 접목시켰다. 이러한 일련의 기술을 통하여 사용자에게 해당하는 클라이언트 시스템은 웹 브라우저만을 가지고 네트워크 상의 애플리케이션과 데이터에 접근할 수 있으며 시간과 공간의 제약 없이 원하는 작업을 효율적으로 수행하는 것이 가능하다.

본 논문은 다음의 순서에 따라 구성되었다. 서론에 이어 2장에서는 서버 기반 컴퓨팅과 네트워크 컴퓨터의 개념 및 원리에 대해 살펴볼 것이다. 3장에서는 그에 연관된 기술 및 문제점을 소개한 후 이를 극복할 수 있는 방안에 대해 기술하고자 한다. 4장에서는 네트워크 기반의 컴퓨팅 구현에 있어 기존 단점을 극복할 수 있는 웹 서버 구축을 위시한 윈도우 관리자, X 윈도우 시스템, Pyjamas 등의 기술에 대해 알아보려고 하며 5장에서는 이를 기반으로 클라이언트에서 원격의 애플리케이션을 실행할 수 있는 네트워크 컴퓨터 아키텍처를 설계 및 구현하고자 한다. 그리고 마지막 결론 부분에서 이러한 논의 및 전개를 기반으로 네트워크 컴퓨터의 가능성을 조망함으로써 본 논문을 마무리하고자 한다.

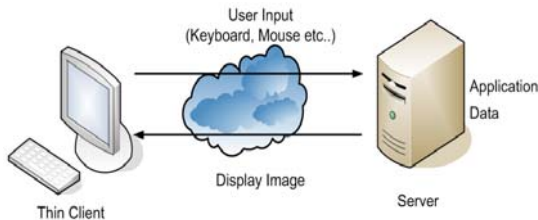
## 2. 네트워크 컴퓨터의 개념 및 원리

본 절에서는 네트워크 컴퓨터의 근간이라 할 수 있는 서버 기반 컴퓨팅에 대해 먼저 살펴보고자 한다. 애플리케이션과 데이터가 서버에 저장되는 서버 기반의 컴퓨팅을 통하여 네트워크 컴퓨터의 구성 및 메커니즘에 먼저 접근하고자 한다.

### 2.1 서버 기반 컴퓨팅

1981년 IBM에서 출시한 PC(Personal Computer)는 컴퓨터 시스템 활용에 있어 말할 수 없는 변화를 야기시켰다. CPU, HDD, 메모리 등 별도의 디바이스를 탑재한 PC의 등장으로 사용자들은 전산실이라는 제한된 환경에서 관리자의 승인을 받아 시스템을 활용하던 한계를 극복하게

되었다. 하지만 PC의 경우 세 가지 크나큰 단점을 가지고 있다. 먼저 초기 도입 비용이 크다는 것이 그 첫 번째이고 시스템 유지 관리, 소프트웨어 업그레이드 등의 TCO(Total Cost of Ownership)가 크다는 것이 두 번째, 그리고 데이터 보안에 취약하다는 것이 그 세 번째에 해당한다 [2]. 일반 사용자의 경우 PC 상에서 수행하는 작업의 상당수가 웹 서핑, 메일 작성, 문서 작업 등이라는 점에서 기존의 PC는 오버 스펙이라 할 수 있다. 또한 대기업의 경우 한해 시스템 유지 관리 비용은 날이 갈수록 증가하는 추세에 있으며 주요 정보의 유출로 인하여 각 기업체에서는 이를 방지하기 위한 보안 강화에 역점을 기울이고 있는 상황이다[4]. 이에 따라 대두된 것이 서버 기반의 컴퓨팅으로서 네트워크 기술에 발전에 따라 그 구현이 가능하게 되었다.



(그림 1) 서버 기반 컴퓨팅의 메커니즘

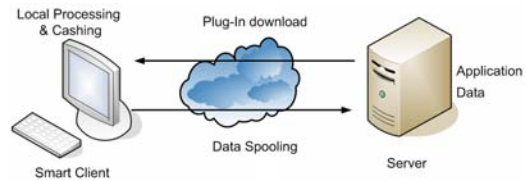
서버 기반의 컴퓨팅에서 애플리케이션과 데이터는 기존의 클라이언트/서버 방식과는 달리 서버에 저장된다. 클라이언트는 서버 측으로 사용자 입력을 전송하고 서버로부터는 디스플레이 이미지를 전송받는다. 이를 통하여 마치 로컬에서와 같이 원하는 작업을 수행할 수 있으며 관련 프로토콜에 VNC(Virtual Network Computing), RDP(Remote Desktop Protocol), ICA(Independent Computing Architecture)등이 있다. 서버의 애플리케이션을 업그레이드함으로써 네트워크 상의 모든 클라이언트는 최신의 애플리케이션을 활용할 수 있으며 이러한 측면에서 서버 기반 컴퓨팅은 유지, 보수의 측면에서 최적의 선택이라 할 수 있다[2][4]. 서버 기반 컴퓨팅의 개략도를 도시하면 위 (그림 1)과 같다.

2.2 플러그-인 기반의 컴퓨팅

서버 기반의 컴퓨팅에서 클라이언트의 입력 신호는 서버로 전달된다. 모든 작업 및 연산처리는

애플리케이션, 데이터가 저장된 서버에서 이루어지며 서버의 디스플레이 이미지는 클라이언트로 전달되었다. 입력 신호와 디스플레이 이미지가 전달되는 서버 기반 컴퓨팅과는 달리 네트워크 컴퓨터에서는 실행 가능한 애플리케이션이 클라이언트로 전달된다[6]. 일반적으로 이는 웹 애플리케이션의 형태로서 실행되는데 서버와의 통신에 기반하며 사용자는 이를 통하여 원하는 작업을 수행하게 된다. 모든 연산 처리가 전적으로 서버에 의존하는 서버 기반 컴퓨팅과 달리 플러그-인 기반의 컴퓨팅에서 클라이언트는 연산의 일부분을 담당하며 그를 위한 임의의 애플리케이션이 웹 브라우저와 상호작용하며 설치 및 운용된다. 이에 관련된 네트워크 기술로 ActiveX, 자바 애플릿, 플래시 등이 있으며 이를 통하여 클라이언트는 임의의 애플리케이션을 웹 브라우저 상에서 실행하는 것이 가능하다.

이러한 플러그-인 기술 기반의 컴퓨팅 개념을 그림으로 도시해보면 아래와 같다[6].



(그림 2) 플러그-인 기술을 활용한 네트워크 컴퓨팅

3. 기존 기술의 한계 및 극복 방안

본 절에서는 네트워크 컴퓨터를 구현함에 있어 관계되는 각 기술에 대해 조망한 후 그 단점 및 한계에 대해 살펴보고자 한다. 그리고 이에 대한 분석을 통하여 향후 네트워크 컴퓨터의 발전 방향, 가능성과 아울러 필요한 네트워크 기술에 대해서도 예측하는 것이 가능할 것으로 생각한다.

3.1 기존의 연관 기술 및 문제점

네트워크 컴퓨터 또는 썬-클라이언트 구현에 있어 현재 가장 널리 활용되는 기술로는 이전 섹션에서도 언급한 바와 같이 서버 기반 컴퓨팅

기술을 들 수 있다. 클라이언트에서 서버에 접속하여 서버의 디스플레이 이미지를 전송받는 방식이 현재 일반적으로 통칭되는 서버 기반 컴퓨팅이며 가장 널리 활용되는 서비스로는 마이크로소프트사의 터미널 서비스, 시트릭스사의 메타프레임을 들 수 있다. 원격 시스템 접속을 위한 별도의 프로토콜을 통하여 서버의 디스플레이 이미지는 클라이언트로 보내어지며 클라이언트 사용자는 마치 서버 앞에 앉아 작업하는 것 같은 기능을 얻을 수 있다.[4]. 서버 상의 애플리케이션 활용 및 데이터 접근을 지원한다는 점에서 네트워크 컴퓨터의 기능을 떠올릴 수도 있지만 이는 완전한 서버 기반 컴퓨팅이라 볼 수 없으며, 성능면에 있어서도 비효율적이라 할 수 있다. 서버 기반 컴퓨팅의 한 형태로 현재 가장 널리 활용되고 있는 터미널 서비스를 예로 설명하자면, 클라이언트는 실제로 애플리케이션이 아닌 서버의 디스플레이 이미지만을 전달받는다. 네트워크 컴퓨터가 “애플리케이션이 네트워크를 통하여 전달되며 원격의 애플리케이션을 로컬 상에서 효율적으로 실행하며 그를 위한 플랫폼으로 네트워크가 진화된다”는 특성을 염두에 두었을 때, 현재의 서버 기반 컴퓨팅은 분명 한계점을 가지고 있다. 또한 서버의 한 애플리케이션을 실행하는데 서버 전체의 디스플레이 이미지를 전송받는 현재의 방식은 비효율적이기도 하거니와 네트워크 대역폭 측면에 있어서도 지양되어야 할 것이다.

네트워크 컴퓨터 구현에 관련된 두 번째 기술로 플러그-인 기술을 들 수 있는데 ActiveX, 자바 애플릿 등을 그 예에 해당한다[6]. 이들 기술의 경우 네트워크를 경유하여 애플리케이션 호출 및 실행 메커니즘이 구현된다는 점에서 현 서버 기반 컴퓨팅의 디스플레이 이미지 전송 방식보다 네트워크 컴퓨팅의 근본 취지에 부합된다고 할 수 있다. 서버 시스템에 대한 접근 및 애플리케이션 실행이 웹 브라우저를 통하여 이루어진다는 것 또한 네트워크와 웹 브라우저의 발전적 측면에서 보다 적절한 방식이라 할 수 있다. 하지만 이 방식도 활용에 있어 여러 문제점을 안고 있는데 플러그-인 프로그램의 웹 브라우저 탑재로 인한 Fat-Client화, 보안 취약성 증가를 그 대표적인 문제점으로 꼽을 수 있다[7]. ActiveX와 자바 애플릿의 경우 네트워크 상

에서 구현하고자 하는 프로그램의 일부가 해당 웹 브라우저에 설치된다. 특이나 ActiveX의 경우 애플리케이션 실행에 필요한 라이브러리(DLL 파일), 컨트롤 프로그램(OCX 파일) 등이 캐비닛 파일(CAB)의 형태로 시스템 디스크에 설치된다는 점에서 보안 취약성으로 이어질 수 밖에 없으며 실제로 이로 인하여 마이크로소프트사에서는 ActiveX를 더 이상 지원하지 않기로 한 것이다.

보안에 있어서의 문제점 이외에도 ActiveX는 네트워크 컴퓨터의 대안으로 근본적인 한계를 갖고 있다. ActiveX 활용을 통하여 텍스트 위주로 활용되던 기존의 웹 브라우저를 다양한 애플리케이션이 실행되는 가상 데스크탑(Virtual Desktop)으로 업그레이드 하기는 하였지만, 웹 브라우저 상에서 애플리케이션이 실행되기 위해서는 OCX 컨트롤을 비롯한 다수의 애플리케이션들이 클라이언트 시스템 내에 설치되어야 한다[8]. 애플리케이션과 데이터가 서버에 위치하는 것이 네트워크 컴퓨터의 기본 개념이라는 것을 염두 하였을 때, 시스템 디스크에 임의의 프로그램이 설치되며 시스템 리소스를 건드리는 ActiveX는 네트워크 컴퓨터 구현을 위한 바람직한 대안이라 할 수 없으며 이전의 클라이언트/서버 방식에 웹 브라우저를 결합한 형태에 지나지 않는다는 것을 알 수 있다. 자바 애플릿의 경우도 웹 브라우저 상의 애플리케이션 기능 구현을 위하여 웹 브라우저 내에 가상 머신(Virtual Machine)과 애플리케이션이 적재된다는 점에 있어서는 ActiveX와 대동소이하다 할 수 있다[9]. 다만 시스템의 리소스를 직접 건드리는 ActiveX와 달리 자바 애플릿은 그 적용 범위가 웹 브라우저에 한정되기 때문에 썬-클라이언트 측면에서 보다 가능성있고 안정적인 보안 특성을 제공한다고 할 수 있다.

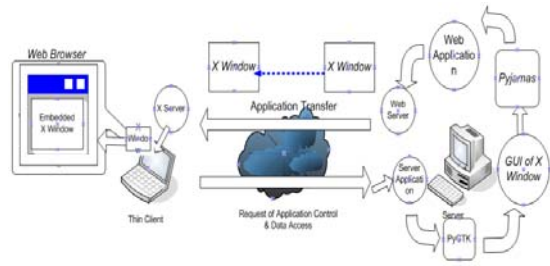
### 3.2 문제점 개선 및 해결 방안

이전 절에서 네트워크 컴퓨터를 구현함에 있어 서버 기반 컴퓨팅 및 플러그-인 기술의 한계와 문제점에 대해 살펴보았다. 네트워크 컴퓨터는 서버에 저장된 애플리케이션과 데이터를 사용자가 로컬에서와 같이 활용하는 것을 전제로 하는데, 서버 디스플레이 이미지 전송의 서버 기반 컴퓨팅이나 클라이언트 내 별도의 프로그램

이 설치되어야 하는 ActiveX, 자바 애플릿의 경우는 그를 구현함에 있어 한계에 다다름을 알 수 있었다. 본 논문에서는 그를 극복하기 위한 방안으로서 웹 브라우저, X Window, Pyjamas를 활용하고자 하며 이를 통하여 원격의 사용자가 웹 브라우저를 통하여 서버 상의 애플리케이션을 효율적으로 활용할 수 있도록 할 것이다. 본 논문에서 네트워크 컴퓨터를 구현함에 있어 상기 각 요소의 기능을 간략히 요약해보면 아래와 같다.

- 웹 브라우저 - 네트워크 상의 서버 시스템에 대한 접속 및 애플리케이션 활용이 이루어지는 작업 공간으로서 그 자체가 하나의 클라이언트에 해당
- X window - 클라이언트에서 원격의 애플리케이션 실행 및 데이터 접근을 로컬에서와 같이 구현가능 하도록 함[10]
- Pyjamas - 구글 웹킷에 기반하며 크로스-브라우저 웹 애플리케이션 개발을 위한 API로서 파이썬에 기반한 웹 애플리케이션 개발을 지원[11]

본 논문에서 제안하는 네트워크 컴퓨터 설계에서 사용자는 웹 브라우저를 기반으로 원격의 서버 시스템에 저장된 애플리케이션을 실행하고 데이터를 활용하게 된다. 이를 효율적으로 구현하기 위한 한 방법으로서 본 논문에서는 마치 로컬에서와 같이 그래픽한 형태의 작업 환경을 지원하는 X Window 기술을 활용하였고 이를 웹 브라우저에 디스플레이시킴으로써 웹 브라우저를 네트워크 기반의 썬-클라이언트로 업그레이드 하도록 하였다. 웹 브라우저가 네트워크 상의 시스템에 접속하면 기본적으로 해당 시스템 로그인을 위한 Xterm이 디스플레이 된다. 반면 임의의 웹 서버에 해당 웹 페이지가 담겨있는 X Window가 사용자의 웹 브라우저에 디스플레이 된다. 네트워크 컴퓨터 아키텍처 설계를 위한 각 구성 요소 및 동작 메커니즘을 그림으로 나타내면 (그림 3)과 같다. 이는 논문의 서두에 언급한 네트워크 컴퓨터의 두 가지 형태를 모두 갖는다 할 수 있다.



(그림 3). 네트워크 컴퓨터 설계를 위한 각 구성 요소 및 동작 메커니즘

로컬 상의 서버에서와 같은 사용자 인터페이스를 클라이언트에 제공하기 때문에 서버 기반 컴퓨팅의 주요 특징인 서버 디스플레이 이미지를 전송받는 것은 물론, 웹 페이지를 실행함에 있어 서버 측의 프로그램을 호출하기 때문에 애플리케이션 전달 효과도 얻을 수 있다[6]. 서버 측 웹 브라우저를 X Window의 형태로 클라이언트 측의 웹 브라우저와 결합시키는 것은 보안의 측면에 있어서도 매우 큰 이점을 제공한다. 애플리케이션 실행 영역으로까지 웹 브라우저의 기능을 확장시켰지만, 클라이언트 시스템의 보안에 위해요소로 작용하는 ActiveX 컨트롤을 안전하게 활용하는 것이 가능하며, 클라이언트 시스템에는 외부의 프로그램이나 파일이 설치되는 것을 차단할 수도 있다. 이러한 네트워크 컴퓨터 아키텍처를 통하여 웹 브라우저에서 네트워크 상의 애플리케이션과 데이터를 활용함은 물론, 네트워크 기반의 가상 컴퓨터를 구현할 수 있다. 뿐만 아니라 저사양의 컴퓨터라도 웹 브라우저를 기반으로 최대의 보안과 성능을 제공하는 썬-클라이언트를 구현하는 것이 가능하다.

#### 4. 네트워크 컴퓨터 설계를 위한 제반 요소의 구성 및 메커니즘

본 절에서는 네트워크 컴퓨터를 구현함에 있어 수반되는 상기 요소에 대해 보다 세부적으로 살펴보고자 한다. 이를 통하여 네트워크 컴퓨터의 동작 메커니즘을 보다 명확히 이해할 수 있음은 물론, 단계적으로 네트워크 컴퓨터의 실체에 접근할 수 있을 것이라 생각한다.

#### 4.1 웹 기반의 작업 관리자

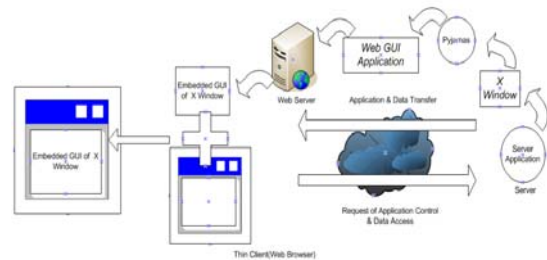
네트워크의 발전 및 관련 기술의 진보에 따라 컴퓨터에서 네트워크 기능의 비중은 갈수록 높아지고 있다. 필요한 연산 및 작업 수행에 있어 단일 시스템에 국한되지 않고 네트워크 상의 애플리케이션, 데이터 등을 가용 리소스로 적극 활용하는 형태로 발전하고 있으며 그 중심에는 웹 브라우저가 존재한다. 단순히 텍스트 위주의 HTML 페이지를 표시하던 한계를 넘어 현재의 웹 브라우저는 다양한 애플리케이션이 실행되며 실제 작업이 이루어지는, 그 자체가 하나의 클라이언트 시스템으로 진화하고 있으며, 웹 브라우저를 통하여 사용자는 네트워크 차원의 가상 컴퓨터를 접하게 된다. 웹 브라우저는 다수의 애플리케이션 윈도우가 실행되고 사용자 작업이 이루어지는 웹 브라우저 기반의 윈도우 매니저이자, 웹탑(WebTop)의 형태로 진화하고 있으며 사용자는 마치 데스크탑에서 애플리케이션을 실행하는 것 같이 웹 브라우저 공간에서 관련 작업을 수행하는 수준으로까지 발전하게 되었다 [12].

본 논문에서는 네트워크 컴퓨터 아키텍처를 설계하고 그를 구현함에 있어서 웹 브라우저 상에 윈도우 관리자 특성을 부여할 것이다. 각 사용자가 클라이언트에 별도의 프로그램이나 하드웨어 없이 웹 브라우저 상에서 원하는 작업을 수행하도록 하는 것이 우리가 설계하고자 하는 시스템의 첫 번째 기능이며 원격 서버상의 애플리케이션 및 데이터를 효율적으로 활용하도록 하는 것이 두 번째 기능에 해당한다. 본 논문에서는 이를 구현하기 위한 기반으로 웹 브라우저를 적극 활용하고자 한다.

#### 4.2 X 윈도우 시스템

X 윈도우 시스템(X Windows System, 흔히 X11, X라고 알려져 있음)은 XFree86(<http://www.xfree86.org>) 그룹이 개발한 그래픽 사용자 인터페이스를 제공하는 데스크톱 작업 환경으로서, 레드햇을 비롯한 많은 리눅스 배포판에서 기본 X 윈도우 시스템으로 활용되고 있다. 서버와 클라이언트의 네트워크 모델 지원을 위하여 X서버, X 클라이언트가 서로 분리되도록 설계되었으며 이를 통하여 X 서버의 프로그램을 X 윈도우 클라이언트가 이용할 수 있는 특성을 제공한다. [13].

사용자는 X 윈도우 시스템을 통하여 각종 애플리케이션 실행 및 데이터 처리를 수행할 수 있다. 단순히 Telnet이나 SSH 등의 접속에 기반한 텍스트 터미널이 아니라 실제 로컬 상에서 작업하는 것 같이 원격에서 GUI 기반의 애플리케이션을 활용할 수 있다. 본 논문에 X 윈도우의 이러한 기능을 구현함으로써 앞서 2장에서 살펴본 두 가지 효과(서버 디스플레이 이미지 전송, 네트워크를 통한 애플리케이션 실행)를 모두 얻을 수 있다. 네트워크를 경유한 원격에서의 애플리케이션 실행 및 데이터 접근이 네트워크 컴퓨터의 전제조건이라는 점에서 X 윈도우의 이러한 특성은 네트워크 컴퓨터를 구현함에 있어 크나큰 이점을 제공한다. 때문에 본 논문에서는 웹 브라우저에 X 윈도우를 접목함으로써 원격 애플리케이션 실행 및 데이터 접근이라는 네트워크 컴퓨터의 특성을 구현할 것이며 이에 대한 개략도를 그림으로 나타내면 아래와 같다.



(그림 4) 웹 브라우저와 X 윈도우 GUI의 결합

#### 4.3 Pyjamas

Pyjamas는 파이썬으로 AJAX 애플리케이션 개발을 위한 웹 개발 프레임워크로서, 파이썬 프로그램을 자바스크립트 프로그램으로 변환하는 컴파일러로서의 기능을 수행한다. 기존에 GWT (Google Web Toolkit: 구글 웹 툴킷)을 통하여 자바 기반으로 Ajax 애플리케이션을 개발하던 것과 같이 개발자는 Pyjamas을 통하여 파이썬으로 Ajax 애플리케이션을 개발할 수 있으며 이 과정에서 클래스와 모듈을 적절히 활용함으로써 HTML이나 자바스크립트에 신경쓰지 않고 웹 애플리케이션을 구현할 수 있다[14]. 네트워크 컴퓨터 설계에 있어 본 논문에서는 GTK 기반의 리눅스 GUI를 웹 브라우저에 포팅하기 위한 용도로 Pyjamas를 활용할 것이다. 원격의 애플리



케이션과 데이터 접근에 있어 본 논문에서는 X 윈도우 접근 방식을 활용할 것이고 그 필요성 및 효과에 대해서는 이미 이전 절에서 기술하였다. Pyjamas에서는 PyGTKWeb이라는 패키지를 통하여 GTK 기반의 GUI를 웹 브라우저 상에 디스플레이시킬 수 있다[15]. GTK 애플리케이션의 라이브러리를 할 수 있는 gtk.py, glib.py, gobject.py에 대응하는 래퍼를 통하여 데스크탑 GUI를 기존 GTK 애플리케이션에 대한 변경없이 웹 브라우저 상의 애플리케이션으로 변화시킬 수 있다. X 윈도우의 대표적인 형태로 Xterm과 Xbrowser를 들 수 있으며 이들은 GUI의 한 형태에 해당한다. 본 논문에서는 Xterm과 Xbrowser를 GTK 기반으로 구현할 것이며 이를 웹 상에 디스플레이함으로써 궁극적으로 웹 브라우저를 통한 원격 애플리케이션 실행, 데이터 접근을 구현하기로 한다.

### 5. 네트워크 컴퓨터 설계 및 구현

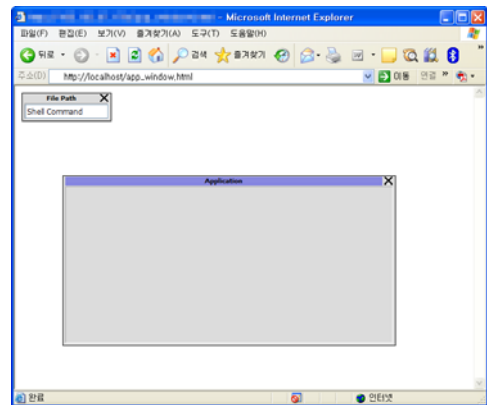
본 절에서는 앞서 살펴본 네트워크 컴퓨터 구성을 위한 사항을 기반으로 실제 네트워크 컴퓨터를 설계 및 구현하기 위한 방안에 대해 기술하기로 한다.

#### 5.1 네트워크 컴퓨터 아키텍처 설계

본 논문에서 제안한 네트워크 컴퓨터를 구현함에 있어 X Window[13], GTK, Pyjamas[14], 자바 스크립트 기반의 윈도우 관리자 기술을 활용하고자 한다. 작업 수행에 필요한 애플리케이션과 데이터는 서버 상에 위치하며 사용자는 웹 브라우저에 기반하여 이를 활용하게 된다. 작업 수행에 필요한 애플리케이션과 데이터가 저장된 네트워크 컴퓨터는 다수의 클라이언트에서 웹 브라우저를 통하여 접속할 수 있도록 다양한 인터페이스를 지원하며 해당 시스템에 대한 애플리케이션 접근은 웹 브라우저를 통한 X 윈도우의 형태로 제공된다.

4장 서두에서 네트워크 컴퓨터의 구성 요건에서 이미 언급하였듯이 네트워크 컴퓨터의 구현 및 활용에 있어 웹 서버, 웹 브라우저는 절대적인 역할을 한다. 네트워크로 모든 시스템이 연결되고 시스템 리소스가 통합되는 가상 컴퓨팅 환경에서 네트워크 상의 외부 시스템에 존재하는

애플리케이션 및 데이터에 대한 접근성은 높아질 수 밖에 없으며 사용자는 웹 브라우저를 사용하여 마치 로컬에서와 같이 시스템 접속 및 애플리케이션 호출을 수행한다. 네트워크 컴퓨터 패러다임에서 그 자체가 경량화된 형태의 썬-클라이언트로서 웹 브라우저는 네트워크 상 상에서 접근 가능한 리소스를 활용하는 기반이라 할 수 있으며, 다양한 애플리케이션 및 데이터 특성을 갖춘 웹 서버와의 통신은 더욱 중요해진다 할 수 있다. 하지만 본 논문에서는 기존에 웹 서버 상의 리소스로 제한된 웹 브라우저의 한계를 넘어 시스템 전체 영역에 걸쳐 애플리케이션 및 데이터를 활용할 수 있도록 시스템을 설계하고자 한다. 이를 구현하기 위하여 클라이언트는 두 가지 방법을 통하여 서버 시스템에 접속해야 하며 그를 위한 첫 번째 방법으로 웹 브라우저를 통한 웹 서버 접속이 있다. HTTP 프로토콜 방식의 세션 연결에 해당하며, 본 방식에서 사용자가 웹 서버에 접속함과 동시에 웹 브라우저 상에는 두 개의 웹 기반 위젯이 아래와 같이 생성된다. 또한 이에 관련된 자바 스크립트 소스를 나타내보면 아래와 같다.



(그림 5-1) 웹 기반의 애플리케이션 실행을 위한 윈도우 관리자

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

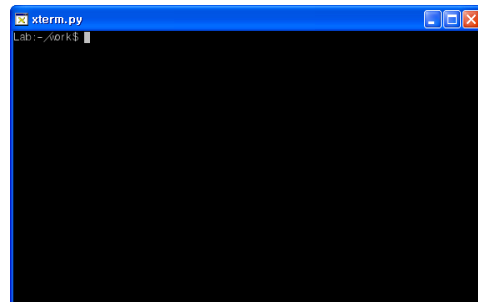
```
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <link rel="stylesheet" type="text/css"
href="/wtk-cvs/wtk-dhtml.css">
    <script type="text/javascript"
src="/wtk-cvs/wtk-dhtml.js"></script>
    <script type="text/javascript"
src="/wtk-cvs/tests/test_window.js"></script>
    <script>
function onLoad()
{
  WtkInit();
  try {
    run_test();
  } catch (e) {
    print("Got exception: " + e +
"\n");
    if(e.stack) {
      print("*Stack Trace*\n" +
e.stack + "\n");
    }
  }
}
....
```

(그림 5-2) 웹 기반의 윈도우 프레임구성을 위한 개략적 소스코드

네트워크 컴퓨터의 애플리케이션, 데이터는 그 저장된 위치, 실행 방법에 따라 크게 두 가지로 분류될 수 있다. HTML 파일로서 웹 서버를 통하여 접속 가능한 파일이 그 첫 번째 유형에 해당하며 그 이외에 일반 폴더에 위치하는 파일들이 두 번째 유형에 해당된다. 웹 서버 상 특정 디렉토리 이하의 HTML 파일은 웹 브라우저의 주소창을 통하여 접근 및 실행이 가능하다. 하지만 타 폴더에 위치한 일반 파일의 접근 및 실행을 지원할 수 있도록 초기 시스템 접근 시 해당 명령을 입력할 수 있는 별도의 위젯을 만들어 놓은 것이며, (그림 5-1) 좌측상단의 위젯이 그에 해당한다. 크기가 다소 큰 직사각형 모양의 위젯은 서버의 Xterm 영역에 해당한다. 작은 위젯에 사용자가 어떠한 명령을 입력하느냐에 따라 해당 애플리케이션이 웹 브라우저의 내부에 추가로 디스플레이되며, 이를 통하여 사용자는

서버의 애플리케이션과 데이터를 효율적으로 활용할 수 있다.

클라이언트의 브라우저 내로 Xterm, Xbrowser 등의 X Window가 발송되어야 함에 따라 본 논문에서는 이들 윈도우 GUI를 GTK를 기반으로 작성하였다, 하지만 기본적으로 GTK 윈도우는 데스크탑 기반이기 때문에 본 논문에서는 이를 웹 애플리케이션으로 변환하기 위하여 AJAX 프로그램 패키지인 Pyjamas를 활용하였다[15]. Pyjamas은 웹킷에 기반을 두고 있으며 pygtk2, python-qt4, wxWidget와 같이 크로스-플랫폼 애플리케이션 개발을 위한 API를 지원한다. 사용자가 원하면 완벽한 HTML 페이지, 스타일 시트를 로드할 수 있으며 자바스크립트를 실행한다. Pyjamas에서는 실제로 GTK 애플리케이션의 라이브러리에 해당하는 gtk.py, glib.py, gobject.py에 대응하는 래퍼에 기반한 컴파일을 통하여 기존 pygtk2 애플리케이션을 변경하지 않고도 웹 애플리케이션을 구현한다. 우리는 GTK로 작성된 아래의 X 윈도우를 클라이언트의 웹 브라우저로 전송함에 있어 이러한 특성을 활용하였고 먼저, GTK 기반으로 작성된 X 윈도우와 그에 대한 소스를 살펴보면 아래와 같다.



(그림 6-1) 웹 서버로 전달될 GUI 기반의 Xterm

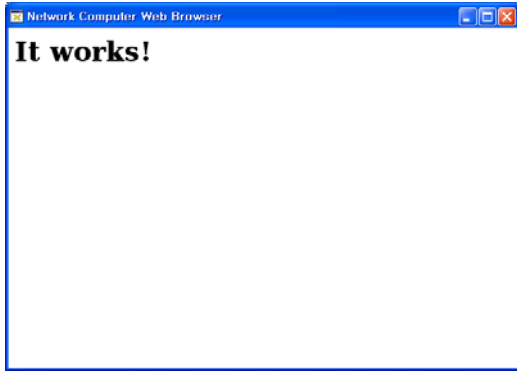
```
if __name__ == '__main__':
    vte=vte.Terminal()
    vte.connect("child-exited", lambda
term: gtk.main_quit())
    vte.fork_command()
    window = gtk.Window()
    window.add(vte)
```



```

window.connect('delete-event',
lambda window, event: gtk.main_quit())
window.show_all()
gtk.main()
.....
    
```

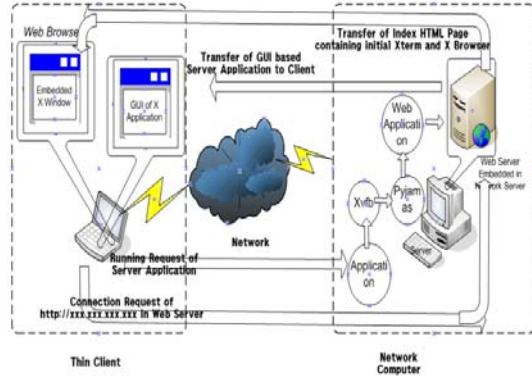
(그림 6-2) GTK 기반의 Xterm 구성



(그림 7) Localhost 접속 시 웹 브라우저로 전송될 GTK 기반의 X 윈도우

위 코드에서 볼 수 있듯이 Xterm과 웹 브라우저를 GTK를 기반으로 X 윈도우로 구현함에 있어 각각 파이썬의 vte 모듈과 mozembed 모듈을 활용하였다. 이를 통하여 각 윈도우에 터미널 애플레이터와 HTML 렌더링의 웹 페이지를 구현하였다.

클라이언트에서 웹 브라우저를 통하여 서버상의 애플리케이션을 호출하면 해당 애플리케이션의 X 윈도우가 서버 측으로 전송되어야 하는데 이를 위해서는 로컬 상에 X 서버가 구동되어야 한다. 본 논문에서는 이를 위한 방안으로 웹 브라우저 상의 X 윈도우에 X 가상 프레임 버퍼(Xvfb: X Virtual Frame Buffer)를 적용하였으며 이를 통하여 서버의 애플리케이션을 웹 브라우저 상에서 활용 가능 하도록 하였다. 이는 VNC, 메타프레임 등 서버 기반컴퓨팅과 같은 서버의 이미지 전달 방식이 아닌 실제 원격 애플리케이션을 웹 브라우저를 통하여 제공한다는 것에 의미가 있으며 이를 기반으로 한 네트워크 컴퓨터의 최종 아키텍처를 그림으로 도시하면 아래와 같다.



(그림 8)네트워크 컴퓨터 구성을 위한 네트워크 열계

## 6. 결론

고성능의 서버급 컴퓨터의 등장과 그를 잇는 네트워크 기술의 발전은 컴퓨팅 활용에 있어서도 크나큰 변혁을 초래하였다. 단일 시스템 내에서 독립된 형태로 동작하던 기존 방식 하에서 애플리케이션은 그 자체가 서버이자 클라이언트였으며 외부에 차단된 상태에서 다수의 기능과 애플리케이션을 갖춘 Fat-Client 형태로 발전하였다. 필요한 기능을 자체적으로 구현하고 다수의 애플리케이션을 제공하는 특성으로 인하여 시스템 사양은 높아질 수 밖에 없었고, 소프트웨어 유지 관리 과정에서 많은 번거로움이 발생하였다.

네트워크 기술의 발전으로 현재 네트워크는 더 이상 예전의 그 단순한 물리적 네트워크에서 벗어나 애플리케이션이 전달되고 구동시킬 수 있는 플랫폼으로 진화되고 있으며 이를 통하여 향후에는 단일 시스템 환경에서는 불가능하였던 컴퓨팅 구현이 가능할 것으로 예측되고 있다. 사용자 시스템에 없더라도 타 시스템에 구현된 애플리케이션을 호출함으로써 필요한 기능을 수행할 수 있게 되었고 이는 서버 기반의 컴퓨팅, Thin-Client의 형태로 나타나게 되었다. 서버 기반의 컴퓨팅에서 모든 애플리케이션과 데이터는 서버에 위치하며 클라이언트는 네트워크를 통하여 필요한 기능을 호출한다. 본 논문에서는 이러한 시스템을 구현하기 위한 첫 시발점으로 웹 브라우저를 가능성 있는 썬-클라이언트의 대

안으로 제시하였다. 웹 브라우저 상에서 사용자가 데스크탑 애플리케이션을 활용하는 것 같이 웹 애플리케이션을 활용하는 것이 본 논문에서 추구하고자 한 네트워크 컴퓨터의 형태로서 이를 위하여 본 논문에서는 웹 기반의 윈도우 관리자를 설계 및 구현하였다. 뿐만 아니라 단순한 웹 서버 접속에 국한하지 않고 서버 시스템의 애플리케이션을 효율적으로 활용할 수 있도록 그를 위한 별도의 명령어 윈도우(Command Window)를 웹 페이지에 구성하였다. 관건은 서버 상에서 호출된 애플리케이션을 X 윈도우의 형태로 클라이언트의 웹 브라우저에 띄어주는 것으로서 이를 위하여 GTK 기반의 X 윈도우 또한 제작하였다. Xvfb를 통하여 클라이언트 상에 X 서버를 구동함으로써 서버 측의 애플리케이션이 클라이언트 상에 디스플레이 되도록 하였으며 이를 통하여 웹 브라우저로 일원화되는 웹 기반의 썬-클라이언트이자 네트워크 컴퓨터를 설계할 수 있었다.

### 참 고 문 헌

[1] White Paper, "Thin-Client/Server Computing", Citrix Systems, Inc, 1998  
 [2] Andrej Volchkov, Server-Based Computing Opportunities, IT Pro, 2002.  
 [3] Jonathan M. Smith, "Selected Challenges in Computer Networking", IEEE Internet Computing, 1999.  
 [4] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper, "Virtual Network Computing", pp.Mobile Computing, 33-38, 1998  
 [5] Lin. G, Dasmalchi. G, Zhu. J, "Cloud Computing and IT as a Service: Opportunities and Challenges", 08, IEEE International Conference 2008.  
 [6] [http://www.pulsewan.com/data101/thin\\_client\\_basics.htm](http://www.pulsewan.com/data101/thin_client_basics.htm)  
 [7] Gary McGraw, Greg Morrisett, "Attacking Malicious Code: A Report to the Infosec Research Council," IEEE Software, vol. 17, no. 5, pp. 33-41, Sep/Oct, 2000  
 [8] Donis Marshall, "ActiveX/OLE Programming", R&D Books, 2000  
 [9] Rubin, A.D. Geer, D.E., Jr., "Mobile code security", Internet Computing, IEEE, 1998

[10] Chris Tyler, "X Power Tools", O'reilly, 2007  
 [11] <http://pyjs.org/>  
 [12] Nenad V. Nikolic, Milan V. Trajkovic, "Socratenon ? A Web-Based Training System with an Intellect", Proceedings of the 33rd Hawaii International Conference on Systems Sciences, IEEE Computer Society, 2000  
 [13] [en.wikipedia.org/wiki/X\\_Window\\_System](http://en.wikipedia.org/wiki/X_Window_System)  
 [14] <http://code.google.com/p/pyjamas/>  
 [15] [code.google.com/p/pyjamas/wiki/GsocLlpamies](http://code.google.com/p/pyjamas/wiki/GsocLlpamies)

### 송 민 규



2001년 : 강원대학교 전기공학과 (공학사)

2005년 : 강원대학교 전자공학과 대학원 (공학석사)

2002년~현재 : 한국천문연구원 전파천문연구부  
 관심분야 : 분산 컴퓨팅, 썬-클라이언트, 네트워크 컴퓨터, 원격 데스크탑, 그리드