

# Multimedia Storage Server를 응용한 MMO

## 게임 서버 구조에 대한 설계

황요한<sup>0</sup> 김동균 장인걸, 신동일, 김동현  
세종대학교 컴퓨터공학과, 컴퓨터공학과, 디지털컨텐츠학과  
{xfilen<sup>0</sup>,kd999,ig8961}@gce.sejong.ac.kr, dshin@sejong.ac.kr, mustache@sejong.ac.kr

### Design of an Architecture for Massively Multiplayer Online Game Server using Multimedia Storage Server

Yo-Han Hwang<sup>0</sup> Dong-Kyun Kim In-Gaul Jang, Dong-il Shin, Dong-Hyun Kim  
Dept. of Computer Science, Dept. of Digital Contents, Sejong University

#### 요 약

온라인 게임(online game)은 초고속 통신망의 보급과 기존의 1인용 게임(Single-player game)의 인공지능에서 느낄 수 없었던 즐거움을 제공함으로써 급속도로 보급되었다. 또한 최근의 온라인 게임 시장은 그래픽 기술을 비롯한 하드웨어의 발달로 2D에서 3D로 옮겨가고 있다. 따라서 이러한 추세에 따라 온라인 게임은 보다 많은 멀티미디어 데이터를 필요로 하게 되었다. 이에 Multimedia Storage Server를 멀티플레이어 온라인 게임(Multi-player online game)에 적용함으로써 보다 효율적인 멀티플레이어 온라인 게임 서버를 구성하고자 한다.

#### 1. 서 론

게임은 남녀노소 누구나 사용하는 소프트웨어로서 자리매김하고 있다. 또한 네트워크 및 하드웨어의 발달로 기존의 인간과 컴퓨터의 대결로 이루어지던 게임에서 이제는 인간과 인간의 대결이 가능하게 됨으로써 기존의 게임에서 느낄 수 없었던 즐거움을 느낄 수 있게 되었다. 이러한 온라인 게임(Online Game)의 정의는 유무선으로 연결된 통신 네트워크 상에서 진행되는 게임을 의미하며, 보다 협의적 의미로는 다수의 사용자가 Client측의 단말기를 사용해 Game Server에 Online으로 접속하여 다수의 사용자와 일정시간동안 다양한 데이터를 주고받으며 진행되는 게임을 지칭한다[3].

온라인 게임은 반드시 게임서버에 접속하여야만 실행될 수 있고 로그인 과정을 거친 후에 게임을 즐기며 전세계 사용자들과 동료 또는 적으로 만나면서 가상 사회를 형성할 수 있다.

본 논문에서 제안하는 시스템은 2D에서 3D로 옮겨가고 있는 최근의 온라인 게임 시장을 반영하여 3D 그래픽과 다양한 멀티미디어 데이터를 사용하는, 동시에 다수의 사용자가 접속 가능한 MMORPG (Massively Multiplayer Online Role Playing Game)를 고려하여 설계되었다.

#### 2. 시스템 구조

만일 게임을 실행하기 위해 대용량의 파일을 다운로드 받아 설치해야 한다면 이는 게임에 대한 사용자의 접근성을 제한하는 요소로 작용할 것이다. 이것은 또한 사용자의 하드웨어 공간을 필요로 하므로 사용자에게 경제적 부담을 유발할 수도 있다.

제안할 시스템은, 대용량의 멀티미디어 데이터를 저장하고 있던 Multimedia Storage Server가 사용자의 요청(request)에 따라 데이터를 전송하게 되는 구조로, client는 CPU Server에서 수행된 결과와 Multimedia Storage Server로부터 받은 데이터를 이용하여 display 작업을 수행하게 된다.

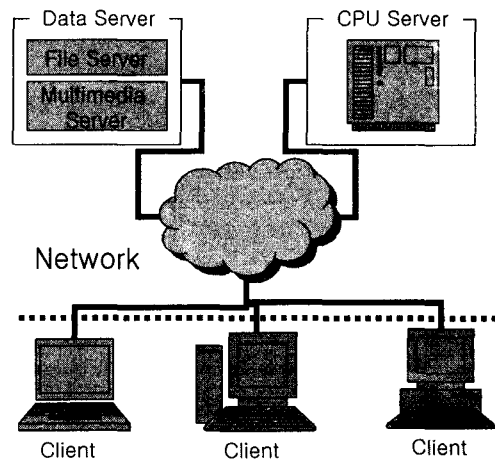


그림 1 Multi-player Online Game의 구조도

### 3. 구성요소

본 논문에서 제안하는 온라인 게임 서버 시스템은 Data Server, CPU Server, Client의 구성 요소로 이뤄진다. 각각의 구성요소들이 수행하는 기능을 살펴봄으로써 시스템의 특성에 대해 알아보도록 하자.

#### 3.1 Data Server

그림 1에서 살펴본 구조 중 Data Server의 보다 자세한 구성은 그림 2와 같으며 Game Engine, Level Data등의 정보를 저장하게된다.

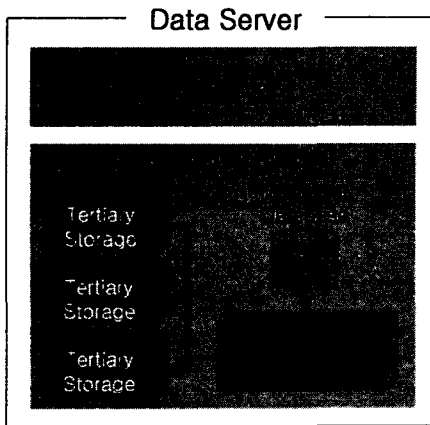


그림 2 Data Server 구조도

몇몇 유명한 게임을 소규모 LAN에서 측정 한 결과 트래픽(traffic)은 Update packet과 Game data 두 종류의 구별되는 형태로 나타났다[5].

Update packet은 사용자가 자신의 상태 정보를 다른 사용자에게 제공하기 위한 작은 크기(32-100 bytes)의 패킷(packet)으로 사용자 캐릭터의 움직임에 따라 그 발생 정도가 결정된다. peer-to-peer 게임의 경우 Update packet의 사이즈는 일정 수준으로 유지되나 client-server 구조에서는 사용자의 수에 따라 선형적으로 증가한다.

이에 반해 Game data는 많은 용량의 데이터를 특정 시간에 필요로 하여 폭발적인 트래픽을 발생시키는데 이것은 scenery data와 character model 등으로 구성된 Game data가 사용자의 cache에 로딩(load)되어야 하기 때문이다. Game data는 게임의 도입부(startup)나 한 level에서 다음 level로 넘어가는 경우 주로 발생하며 그 크기가 5Mbyte에 이르기기도 한다[6].

Update packet은 실시간 상호작용(real-time interactivity)이 강조되지만 Game data의 경우 그렇지 않으므로 두 종류의 데이터를 위해 각각을 실시간 상호작용을 보장할 수 있는 네트워크와 연결된 File Server와

많은 저장공간을 제공하는 Multimedia Server로 나누어 구성하도록 한다.

나누어 구성하는 또 다른 이유는 Update packet과 Game data가 텍스트 파일과 멀티미디어(video, audio) 파일과 같이 서로 다른 특성을 지닌 데이터로 구성되며, Game data의 경우 주로 읽기 전용으로 사용되는데 반해 Update packet의 경우 수시로 업데이트(update)를 위한 읽고,쓰기가 발생하는 등 사용형태가 다르기 때문이다. 따라서 각각의 서버에 서로 다른 disk scheduling 및 buffer를 적용함으로써 성능의 향상을 기대 할 수 있다. 가장 잘 알려진 real-time disk scheduling 알고리즘으로는 EDF(earliest deadline first)알고리즘이 있으나 과도한 seek time과 rotational latency가 발생하므로, 디스크 헤드가 backtracking할 수 있는 양을 제한하는 Scan 알고리즘과의 결합된 형태인 Scan-EDF 알고리즘이 보다 적합하다. 이를 제외한 다른 알고리즘의 경우 각각의 프로세서에 대해 round 단위로 서비스하는데 이의 가장 단순한 형태인 round-robin의 경우 각각의 round에서 고정된 순서에 의해 서비스함으로써 디스크의 상대적 위치의 이점을 활용하지 못하는 단점이 있다. 이런 이유로 data-placement 알고리즘과 결합하여 사용하기도 한다 [7].

MMORPG 게임의 경우 특정 기간에 특정 level에 접속하는 사용자가 많아지는 특징을 보이므로 Multimedia File Server에 최근에 사용된(LRU-based) level을 유지하는 방법도 생각해 볼 수 있다[6]. 또한 Tertiary Storage로 대용량의 저속의 장비를 사용하고 Multimedia File Server로 고속의 입출력이 가능한 장비를 이용한다면 가격 대 성능비에 따른 경제적 효과를 기대할 수 있다. 그러나 이런 계층 구조를 적용할 경우 구성의 복잡도(Complexity)가 증가하므로 얻을 수 있는 이익을 고려하여 적용하도록 한다.

#### 3.2 CPU Server

게임 로직(game logic)의 수행 및 사용자의 게임 속에서의 위치정보 등의 update를 수행하는 프로세서들의 조합이다. CPU Server는 많은 사용자들에 의해 사용될 공용의 프로세싱 자원을 제공하게 되는데 이러한 '중앙 집중적' 방식은 서버에 부담을 줄 수 있으나 제한한 시스템에서는 collision detection, hidden surface removal, polygon sort등의 많은 연산을 필요로 하여 병목(bottleneck)현상을 일으키는 display 작업을 제외한 게임 로직만을 수행하게 되므로 많은 부하가 제거되었다고 할 수 있다. 또한 이 경우 게임 로직이 서버에서 수행되므로 맵해과 같은 해킹으로부터 게임을 보호할 수 있는 효과를 기대해 볼 수 있다.

그림 3은 Multipalyer 온라인 게임에서 수행되는 main

loop의 전형이다.

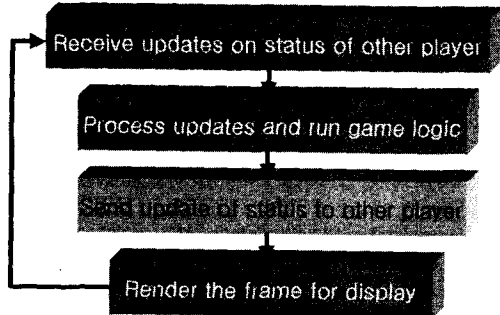


그림 3 Multiplayer 게임 세션에서의 task

### 3.3 Client

Client는 display 작업만을 수행하게 되는데 앞서도 언급하였지만 display 작업은 게임을 수행하는데 있어서 주요 병목현상을 유발한다. 또한 사용자와의 상호작용에도 영향을 미치므로 상호작용하는 실시간 온라인 게임을 위한 중요 요소이다. 예를 들어 설명하면 20FPS (Frame Per Second)로 수행되는 게임의 경우 각각의 frame을 그리기 위해 50ms의 시간이 걸리는데 이 50ms동안 사용자의 요구가 수용될 수 없으므로 50ms의 lag이 발생하게 되는 것이다 [9]. 이상에서 살펴본 시스템 구성요소들은 다음과 같은 과정을 거쳐 게임 서버로써의 기능을 수행되게 된다.

새로운 사용자가 게임의 새로운 session에 참여하게 되면 game engine은 Data Server에서 CPU Server로 로딩되고 새로운 session에 해당하는 level data 및 multimedia data는 CPU Server에 로딩(loading)된 game engine에 의해 수행된 게임 로직(game logic)의 결과와 함께 client로 보내져 display 되게 된다.

### 4. 결론

최근의 온라인 게임 시장은 그래픽 기술을 비롯한 하드웨어의 발달로 2D에서 3D로 옮겨가고 있다. 따라서 이러한 추세에 따라 온라인 게임은 보다 많은 멀티미디어 데이터를 필요로 하게 되었다. 이에 Multimedia Storage Server를 멀티플레이어 온라인 게임(Multi-player online game)에 적용함으로써 보다 효율적인 멀티플레이어 온라인 게임 서버의 설계 방안에 대하여 기술하였다. 많은 저장 공간을 필요로 하는 특징을 갖고 있는 멀티미디어 데이터를 위한 전용 서버를 이용함으로써 멀티미디어 데이터를 위한 client측의 저장 공간을 절약할 수 있으며, 설치를 위해 다운로드 받아야 할 데이터의 양을

줄임으로써 게임에 대한 사용자의 접근성 향상을 이룰 수 있다.

향후에는 설계된 온라인 게임 서버를 구현하여 여러 가지 다른 구동 환경에서 멀티미디어 서버의 성능을 측정하여 각 게임 장르에 적합한 온라인 게임 서버 구축 방안에 대하여 연구하고자 한다.

### 5. 참고 문헌

- [1] 한국첨단게임산업협회, "온라인 게임 산업의 현황과 발전 방향", 2000.2.
- [2] (재)게임종합지원센터, "2001 대한민국 게임백서", 2001.
- [3] 주정규,이원규, "온라인 게임 서버기술 연구", 한국게임학회 하계 학술발표대회, pp.49-62, 2001.7.
- [4] 이동원, 최 성, "네트워크 게임 시스템 구축 연구", 한국 게임학회 하계 학술발표대회, pp.115-119, 2001.7.
- [5] R. A. Bangun & H. W. P. Beadle, "Traffic on a Client-Server Based Architecture for Multi-User Network Game Application", Proceedings of ICT'97, vol. 1, pp. 93-98, Melbourne, Victoria, Australia, April 1997.
- [6] Ricky A. Bangun et al, "A Network Architecture for Multiuser Networked Games on Demand", ICICS'97, vol. 3, pp.1815-1819, September 1997.
- [7] D. J. Gemmell et al, "Multimedia Storage Server: a Tutorial", Computer, vol. 28, no. 5. pp. 40-49, May 1995
- [8] P. V. Rangan et al, "Designing an On-Demand Multimedia Service", IEEE Comms. Magazine, vol. 30, no. 7, pp.56-64, July 1992.
- [9] Jonathan Blow et al, "A Look at Latency in Networked Games", Game Developer. Magazine, pp.28-40, July 1998.