

# 온라인 게임 서버의 기술 동향

## Technical Trend of Online Game Server

양광호(K.H. Yang)	3D 협동공간연구팀 선임연구원
심광현(K.H. Shim)	3D 협동공간연구팀 선임연구원
고동일(D.I. Ko)	3D 협동공간연구팀 연구원
박일규(I.K. Park)	3D 협동공간연구팀 연구원
김종성(J.S. Kim)	3D 협동공간연구팀 선임연구원, 팀장

엔터테인먼트 산업으로서의 컴퓨터 게임은 대중의 감각에 의하여 유지되고, 대중의 감각을 최대한 활성화하는 게임이 지속적으로 출시되어야 한다. 게임의 품질은 게임 디자이너의 창작력을 최대한 현실화 시켜주는 컴퓨터 기술에 종속된다. 그러나 현재, 플레이어 수가 수천 명 이상으로 확장되는 대규모 시스템의 범주로 가면, 당장 기술 장벽에 부딪히게 된다. 본 고에서 게임 네트워킹의 특성을 고찰한 후, 산업계의 출시 제품들의 동향을 분석한다. 특히, 기술 개발 경쟁이 심화되고 있는 대규모 멀티플레이어 온라인 게임 서버 동향을 분석한다.

## I. 서론

컴퓨터 게임의 시초는 1961년 MIT에서 만든 Spacewar!이다[2]. 네트워크 게임의 시초는 SGI의 Flight/Dogfight(1984/1985)이고, 본격적인 것으로는 id Software의 Doom이며 1993년경에 전세계적으로 1,500만 본의 셰어웨어(shareware) 프로그램이 사용될 만큼 인기를 누렸다. 1998년 Battle.net 무료서비스로 StarCraft와 Diablo를 히트시킨 Blizzard Entertainment는 StarCraft만으로도 160만 카피를 판매했다[3]. 당시 Battle.net은 피크 타임에 5만 참여자를 자랑하기도 했다.

한편, 텍스트 기반 MUD 게임은 3D 기술과 접목되어, 최근에는 수십만 명이 동시 참여 가능한 게임(Massively Multiplayer Online Game; 대규모 멀티플레이어 온라인 게임)으로 발전하고 있다. 온라인 게임 시장의 잠재적 성장 가능성을 토대로 마이

크로소프트를 비롯하여 거대 게임 업체가 온라인 게임에 참여하고 있다.

본 고에서는 멀티플레이어 온라인 게임 서버의 기술 동향에 관하여 기술한다. 게임 디자이너의 창작력을 현실화하는 장애 요인은 대규모 플레이어간 네트워킹 및 고성능 서버에서 발생하고 있다는 데에서 게임 네트워킹의 특성을 고찰한 후, 산업계의 출시 제품들의 동향을 분석한다. 특히, 기술 개발 경쟁이 심화되고 있는 대규모 멀티플레이어 온라인 게임 동향을 분석한다.

## II. 온라인 게임 서버 기술

### 1. 기술 개요

온라인 게임의 품질 결정 인자의 하나는 네트워크 성능이며, 대역폭과 지연성의 두 가지 척도가 있

다. 대역폭은 게임의 스케일러빌리티(scalability; 플레이어 수의 증가성)를, 지연성은 게임의 응답성을 좌우한다. 이들 양자는 하부 통신 링크가 제공하는 한정된 자원이며, 게임 정보를 이 한정 자원을 활용하여 플레이어들에게 잘 분산시킴으로써 게임의 품질을 향상시키고자 함이 온라인 게임, 특히 게임 서버 기술의 역할이라 하겠다.

최적의 네트워크 성능을 유지하기 위해서는 해당 게임에 적합한 게임 통신 구조를 택하여야 하며, 보통 참여하는 플레이어의 수에 의하여 <표 1>과 같이 결정된다[9].

<표 1> 게임 통신 구조 및 특징

구분	플레이어 수	특징
P2P 구조 (Peer-to-Peer)	16명이 상한	낮은 지연성, 플레이어 수에 따라 대역폭 제한
CS(Client-server) 구조	16 ~ 200명	메시지 필터링 도입, 근접 플레이어 수 제한, 게임 월드 영역 분할(정적, 동적), 게임 진행 동기화와 게임 로직 갱신 등이 용이함
Hybrid 구조	16 ~ 200명	서버를 중심으로 대부분 통신은 P2P[5],[6]
분산 서버 구조	200명 이상	스케일러빌리티, 서버와 서버간 통신 부하

온라인 게임에 플레이어가 어떻게 참여하는가에 따라 <표 2>처럼, drop-in(상시로 출입)과 session-oriented(게임 개시 때에만 참여 가능)으로 나눌 수 있다.

또, 인터넷 게임은 게임의 진행 시간이 단기간인가, 영속적(persistent)인가로 구분할 수 있으며, 영속적 게임의 경우 게임 상태 유지를 위한 서버가 필요하고 전원 중단, 시스템 크래시 등에 대한 대책이 수립되어야 한다.

플레이어가 게임 상대를 어떻게 만나는가에 따라 로비에 등록된 참여자 대기열 순서로 팀을 구성하는 대기열식, 게임 의사를 등록 후 선택 받기를 기다리는 선택식, 주로 영속적 게임의 경우 사용하며 게임 공간 배회 중 상대를 만나게 되는 공간 배회식 등으로 구분한다[10].

<표 2> 게임 참여 방식

구분	특징
드롭인 방식	플레이어는 진행중인 게임에 수시 출입 서버가 상태 정보 관리, 전달할 상태 정보 양이 관건
세션	모든 플레이어가 게임 동시 참여, 사전 랑데부 및 조정 절차 필요. 플레이어 대기 시간을 요함, 일정한 참여 수 확보가 관건

이상에서 온라인 게임의 관련 기술을 개괄적으로 설명하였으며 이 중에서 가장 중요한 사항은 최적의 네트워크 성능을 유지하는 것이다. 어떠한 경우이든 round trip latency 250msec를 보장할 수 있어야 한다[9]. 다음 절에서 최적의 네트워크 성능을 유지하는 기술에 관하여 고찰한다.

## 2. 동적 상태 관리 및 스케일러빌리티

게임 공간에 함께 참여하는 플레이어들이 상호간 같은 시각, 같은 공간에 있으며 그 공간에서 자신들이 하는 일들을 포함해서 그 공간에서 일어나는 사건들을 공유할 수 있을 때 몰입감(immersion), 현실감을 느낄 수 있고 비로소 멀티플레이 게임을 진행할 수가 있다. 이는 참여하는 플레이어 컴퓨터들이 동적으로 변화하는 게임 공간을 동시 공유함으로써 이루어지며 이를 동적 상태 관리(dynamic state management)라고 한다.

지리적으로 분산된 여러 대의 컴퓨터로 진행되는 온라인 게임에서는 네트워크 지연성과 대역폭이라는 제약 조건이 존재하기 때문에 동적 상태 관리의 완전도가 높으면 높을수록 플레이어들이 느끼는 현실감을 저하된다. 반면, 플레이어의 현실감을 높이기 위해서는 동적 상태 관리의 완전도를 낮추어야만 한다. 이를 consistency-throughput tradeoff라고 하며, <표 3>은 consistency throughput 스펙트럼의 양극단을 나타내며, 게임 공간의 빈번한 동적 상태 변화를 실시간으로 유지하면서 동시에 컴퓨터들 간에 동적 상태 일관성을 완전하게 유지할 수 없다는 것을 말한다.

<표 3> consistency-throughput tradeoff

시스템 특성	완전 일관성	높은 상태 갱신율
View 일관성	모든 컴퓨터 동일	수신상태 데이터의존
동적 변화 지원 정도	낮음, 프로토콜에 의하여 제한됨	높음, 대역폭에 의존
네트워크 특성	저지연성, 고신뢰성	헤테로네트워크 가능
수용 플레이어 수	작음	많음

즉, 모든 컴퓨터가 완전히 동일한 상태를 갖출 것이나, 아니면 컴퓨터별로는 동적 상태가 다르더라도 그 시점에서 가장 근사한 상태를 유지하도록 할 것이냐를 나타낸다. 완전한 일관성을 강조하게 되면, 한 컴퓨터에서 상태를 변경할 때, 그 상태가 다른 컴퓨터에서도 동일하게 갱신되었다는 것을 확인하기 위한 프로토콜이 처리되므로 자연스럽게 상태 갱신율이 떨어지게 되고 게임의 실시간성을 유지하기 곤란하게 된다. 반대로 높은 상태 갱신율을 강조하게 되면, 각 플레이어 컴퓨터에서의 게임 상태는 수신된 데이터에 의해 변경되므로, 패킷이 경유하는 네트워크의 품질에 따라 컴퓨터 간에 게임 상태가 다를 수 있고 결과적으로 서로 다른 게임 상태에서 게임을 진행하게 될 수가 있다.

따라서 온라인 게임 개발에 있어서는 게임 공간 상태의 일관성과 높은 상태 갱신율을 동시에 고려하여야 하며 다음의 3가지 기법을 적절히 활용할 수 있다(<표 4> 참조).

스케일러빌리티란 규모 증가의 영향을 말하며, 스케일러블 시스템이란 규모 증가와 관련한 비용이 상대적으로 경미한 것을 말한다. 온라인 게임에서는 주로 동시 사용자의 증가에 쉽게 대응할 수 있는 시스템을 말하며, 온라인 게임 서비스를 개시하는 때는 수천 명 규모일지라도 인기 상승에 따른 수십만 명의 규모에도 대응할 수 있는 스케일러블 시스템을 구현하여야 하며, 본 고에서는 이 “스케일러블” 형용사 대신에 “대규모”라는 말을 사용하기로 한다.

대규모 시스템에서는 네트워크 대역폭(광역 망 및 가입자 망)과 프로세서 성능(서버 및 클라이언트 컴퓨터)의 자원 관리 방식이 핵심 기술이며, 얼마나

<표 4> 상태 관리 기법

구분	특징
중앙 관리 (Centralized Repository)	중앙의 프로세스가 공유 상태 통제 록킹, 순서화 등의 갱신 관리 모든 컴퓨터에서 동일 갱신 값 유지 클라이언트 캐시로 접근 고속화 CS 구조, P2P 토폴로지상 가상적 중앙 관리 구조[21] 네트워크 오버헤드 과다, 스케일러블하지 못함
상태 정보 전파	Doom, Diablo에서 채택, 대부분의 게임은 시간적 동기성이 데이터 정확성보다 중요함 Blind broadcasting, 비신뢰성 UDP 전파 신속화 갱신 관리를 위한 Lock 관리 서버 Proxy update/Ownership transfer 상태 갱신과 관련한 보수적(Conservative) 전략/낙천적(Optimistic) 전략 대역폭 소비 과다, 네트워크 Jitter 현상 발생
데드레커닝 (Dead Reckoning)	송신측은 필수 정보만 송신, 수신측에서 예측 복원(Prediction, Convergence) 대역폭 소비 최소화

이 자원에 대한 요구를 최소로 하면서 온라인 게임 성능을 최대로 유지하느냐가 관건이다. 자원 소모의 가장 중요한 인자는 플레이어이다. 플레이어가 신규로 참여할 때 마다 공유 데이터의 양과 게임 공간에서의 인터랙션이 증가하고, 관련 상태 정보 전달 메시지로 인한 네트워크 대역폭 소모가 증가하고, 증가된 공유데이터와 인터랙션을 처리해야 하기 때문에 각 플레이어 컴퓨터의 부하가 증가하게 된다.

자원과 자원 소모를 요구하는 인자들을 다음과 같이 표현할 수 있다[1].

$$\text{Resource} = M \times H \times B \times T \times P$$

여기서,

M: 게임 공간에서 처리되는 메시지 수

H: 각 메시지 당 목적지 컴퓨터 수

B: 목적지별 메시지 당 요구 평균 대역폭

T: 패킷이 목적지에 도달하는 시간성

P: 메시지 당 처리 프로세서 사이클 값

온라인 게임에서는 이 자원 소모를 최소로 하여야, 여유있는 자원에 의하여 네트워크 로드와 프로세스 성능이 개선되어 전체 온라인 게임의 성능이

개선되는 것이다. 따라서 위 공식에서 우변의 변수 값을 낮추는 것이 필요하나, 현실적으로는 한 변수의 값을 개선하면 반드시 다른 변수 값이 악화된다. 따라서 대규모 온라인 게임에서는 최상의 성능을 보장하는 일반적인 해법은 아직 발견되고 있지 않으며, 온라인 게임의 특성과 그 특성에 따라 어떤 변수가 병목으로 작용하는가를 분석해 내고 그 병목을 해소하는 기법들을 선택적으로 사용할 수 밖에 없다. <표 5>는 이러한 자원 관리를 위하여 활용할 수 있는 기법들의 예이다.

<표 5> 온라인 게임 자원 관리 기법

기법	결정 변수	비고
패킷압축	B 감소   P 증가	패킷 사이즈 축소
패킷집적	M, B 감소   P, T 증가	복수의 갱신 메시지를 한 개 메시지로 통합
멀티캐스팅	H 감소   M 증가	엔터티 소스, 게임 공간 영역별로 그룹 형성, 부분 엔터티 그룹에 갱신 메시지 송출
관심영역 필터	H 감소   M, P, T 증가	필터링 조건 지정에 따른 메시지 송출
상세화 레벨	H, B 감소   M, P 증가	거리에 따른 엔터티 상태 갱신 정도 변경
서버 클러스터	P 감소   T 증가	클라이언트의 다중 서버 분산 서버-서버 메시지 부하

스케일러빌리티를 위하여 게임 공간을 서버 클러스터 상에 분산함으로써 대규모 플레이어가 참여할 수 있도록 하는 게임 서버 클러스터 기술은 특히 중요하다. 근래 대규모 온라인 게임은 보통 수십 대의 서버로 구성되는 서버 클러스터에서 구현하고 있다. 한 대의 컴퓨터가 갖는 동시 접속 플레이어 수의 제한을 극복하는 한편, 게임 공간과 플레이어를 여러 대의 서버로 분산시킴으로써 부하를 줄여서 일정한 게임 성능을 유지하기 위한 것이다.

분산 서버 구조에서는 다음의 두 가지 게임 공간 관리 기술이 사용된다.

- ① Zoned World: 게임 공간을 미리 분할해 놓고 분할 영역별로 관리 서버를 할당함. 한 영역에서 다른 영역으로의 이동은 포털이라는 특정한 이동 지점을 경유해서 이루어짐
- ② Seamless World: 게임 공간의 각 영역은 관리

하는 관리 서버가 게임 진행 시에 그 영역에서의 로드 계산에 따라 동적으로 정해지며 플레이어는 다른 영역으로 이동할 때에 포털을 경유하지 않고 자유롭게 이동할 수 있음

여러 서버 상에서 게임 월드를 관리하기 위하여는 공간을 가장 기본 단위인 셀(cell)로 나누고 각 셀에서의 부하를 실시간으로 모니터링하여 한 개 서버가 관리할 수 있는 인접 셀들을 묶어서 셀 파티션(cell partition)을 구성한 후 서버에 할당한다. 따라서 각 서버가 관리하는 셀은 셀의 부하 및 참여자 분포에 의하여 동적으로 변화하게 된다. 또한 플레이어가 영역간을 이동하면서 다른 서버가 관리하는 부분 가지 영역을 심리스로 관리하는 것이 필요하다.

서버 클러스터로 게임을 진행하는 때는 플레이어가 고품질의 게임을 영위하는 한편으로 게임 상태가 일관성 있게 관리되어야 하므로 서버가 많아지면서 서버간 통신 오버헤드가 증가하게 된다. 이 오버헤드를 줄이기 위하여 게임 국면 마다 플레이어간 인터랙션을 최소화 하는 한편 인터랙션의 국소성(locality of interaction)을 최대화하는 설정을 도입하기도 하며, 그 설정의 극단적인 형태가 MUD에서의 던전(동굴)이다.

### 3. 부가적 고려 사항

상기 게임 상태 유지의 최적화는 온라인 게임에 따라 선별적으로 채용하여야 하는 일차적인 요구사항이며, 이외에도 <표 6>과 같은 이차적인 요구사항을 고려하여야 한다.

### 4. 대표적 서버 엔진, 기술동향

지금까지 온라인 게임은 유능한 창작력을 가진 게임 디자이너와 탁월한 코딩 실력을 갖춘 프로그래머가 단기간에 각자의 힘을 규합하여 게임을 제작하는 것이 위주가 되었고, 다른 가용 프로그램을 활용하기 보다는 스크래치부터 자작하는 관행이 지배하였다. 그러다 점차 게임 엔진을 도입하여 소프트웨

<표 6> 부가적 고려 사항

구분	비고
인터랙션 지속	옵션 스크린, 특정 게임 모드로 메시지 송수신 장애 방지. 메시지 블로킹시 상대 플레이어에 통지, 플레이어 상호간의 활동 상태 파악 수단
음성 및 기타 통신 수단	음성 및 채팅 채널 제공, 녹음/비디오 리플레이, 음성 왜곡(수줍음 기 플레이어 지원) 등의 수단
플레이어와 서버 Drop-out 대책	Graceful handling. AI를 활용한 네트워크 장애시 잠정적인 플레이어 대역 활동 등 상대에 드롭아웃 사실 통지
Drop-in spectator	게임 초심자 게임 진행 관찰, 방관자/트레이너/고객서비스/친구 등 역할, 정보 노출 대책 수립, 활동 범위, 통신 능력 제한
서비스제공자에 대한 고려	서비스 제공자의 서비스 플랫폼과 통합 용이한 소프트웨어 구조. 랑데부/채팅/빌링/계정 로그인/암호 등 관련 통합 용이성. 고객서비스/온라인 시험/베타 프로그램/시범서비스/토너먼트/플레이어랭킹관련 설비 등에 대한 고려. 게임 데이터/플레이어 정보의 엔터프라이즈 DB 저장. 웨어웨어/프리웨어 유통시 정품과의 관계
On-line Automatic Maintenance and Upgrade	비즈니스 S/W는 제품의 유지보수/버전업이 서비스 및 마케팅 전략. 네트워크 게임도 차츰 인터넷을 통한 유지 보수 및 갱신 도입(데이터와 패치 코드 다운로드). 자동화된 AS 및 업그레이드[7],[8]
Robustness	서버 다운 대책, 사용자 손실 및 사업 실패 방지
Performance	일반 소프트웨어는 성능보다는 정확성이 중요. 한 게임이 다른 게임보다 빠르게 작동한다는 사실은 게임 히트 요인
보안 대책	게임 치팅(cheating) 전문 웹사이트 등장 현실 시기같은 영속적 게임에서 수천 명에게 손실, 가입자 손실은 곧 수익 손실과 직결[11]

어를 재활용하는 한편, 제작 기간을 단축하는 사례가 늘고 있다. Quake 엔진처럼 시장에서 한 번 히트한 제품을 기반으로 후속 히트작에 대한 기대감이 높기 때문이다.

또 한편으로 다른 비즈니스 소프트웨어 분야처럼 지네릭(generic) 요구사항을 미리 갖춘 엔진을 기반으로 게임을 제작하는 경향도 일고 있다. NDL사의 '페르시아의 왕자 3D'는 Motion Factory사의 모티베이트 캐릭터 엔진을 이용하기도 하였다.

향후, 게임 제작에 게임 엔진을 활용하는 경향은 일반화될 전망이다.

현재 시장에 출시된 게임 엔진은 유형별로 다음과 같이 분류할 수 있다.

<표 7> 출시 게임 서버 엔진의 분류

구분	범위	제품명	특징(개발사)
소규모	서버 엔진	DirectPlay	음성통신, 고성능 스트리밍 (마이크로소프트)
		NetZ	부하분산, 결합 허용 ProkSim Software Inc.
		TRIBES	데이터 전달 특성별 전달관리 Dynamics
		VR-1	패킷 집적/우선순위/유틸리티 VR-1, Inc.
		Terraplay	서비스제공자/ISP 지원 하부구조 Terraplay AB
		HLA 엔진	RTI 커널 기반 MAK, Cybernet
소규모	게임 엔진	QUAKE	CS 구조, 게임 개발 지원도구 Id Software Inc.
		UNREAL	CS 구조, 게임 개발 지원도구 Epic Games Inc.
		LithTech	게임 개발 지원도구, 전체 장르 LithTech Inc.
대규모	서버 엔진	RTIME	분산서버, 대규모 하부 구조 Rtime Inc.
		Turbine	분산서버, 대규모 게임개발 지원 Turbine Entertainment SW
	게임 엔진	BigWorld	분산서버, 대규모 게임개발 지원 Microforte Pty. Ltd.

- 소규모 서버 엔진: 온라인 게임에서 네트워킹 및 상태 정보 전달이라는 기본 기능을 갖추고 있음. 게임을 제작하기 위해서는 이 기본 기능 위에서 해당 기능에 요구되는 일관성 유지 등 고급 기능을 덧붙여 제작 추가하여야 함
- 소규모 게임 엔진: 장르별로 게임 개발에 필요한 서버 및 렌더링 등의 모든 기능을 구비하고 있음. 개발자는 월드 맵, 스크립팅 정도로 게임을 제작할 수 있음
- 대규모 서버 엔진: 대규모 플레이어 온라인 게임을 개발할 수 있는 스케일러블 구조를 가지고 있음. 그러나 구체적으로 게임을 제작하기 위해서는 해당 게임에 맞는 추가 모듈의 개발이 요구됨
- 대규모 게임 엔진: 대규모 플레이어 온라인 게임에 필요한 모듈이 거의 모두 구비되어 있음. 최소의 추가 개발이 요구됨

이러한 분류에 따라 출시되어 있는 온라인 게임

들은 <표 7>과 같이 분류할 수 있다.

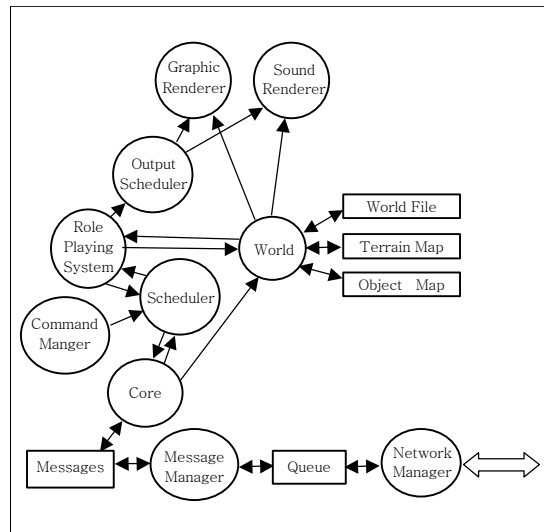
한 개의 컴퓨터 게임이 등장하여 시장을 석권하는가 싶어도 곧 새로운 특징을 구현한 다른 컴퓨터 게임이 나타나면 시장을 내어주어야 한다. 그 특징은 Quake는 Half-life에게, Doom은 Thief에게, Tomb Raider는 Vampire에게 밀린 것처럼, 디자이너 창작력의 우수성으로 인한 게임의 묘미에서의 차이 및 그 디자이너의 창작력을 유감없이 현실화 해주는 컴퓨터 기술의 차이이다. 엔터테인먼트 산업으로서의 컴퓨터 게임은 대중의 감각에 의하여 유지되므로 대중의 감각을 최대한 활성화하는 게임이 지속적으로 출시되어야 한다[12],[13]. 실감 영상을 만드는 3D 그래픽스 기술, NPC(Non Playable Character)를 구동하는 인공지능 기술을 비롯하여, 멀티플레이어 게임을 가능하게 하는 네트워킹 및 게임 서버 기술 등의 게임 구현 기술 위에서 우수한 게임 콘텐츠가 제작될 수 있다.

이 중에서 현재 기술적 장벽으로 인식되고 있는 부분이 네트워킹 및 게임 서버 기술이다. 소규모 멀티플레이어 온라인 게임에 대하여는 가용 기술이 많이 있기 때문에, 차별성 있는 게임 아이디어만 있으면 단기간에 경쟁력 있는 게임을 제작하여 출시할 수 있다. 그러나, 플레이어 수가 수천 명 이상으로 확장되는 대규모 시스템 범주로 가면, 당장 기술 장벽에 부딪히게 된다.

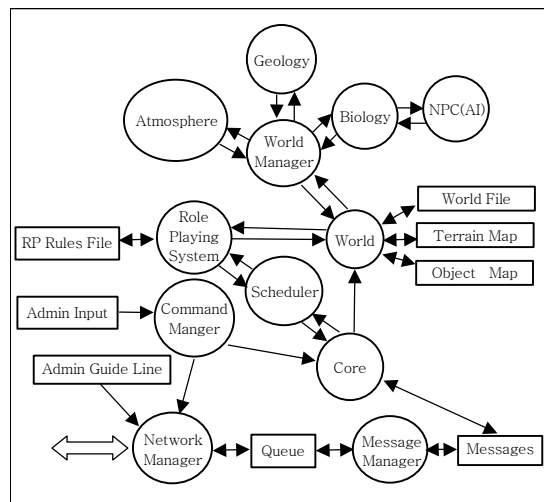
서버 기술의 중요성을 보이기 위하여 롤플레이어 시스템에서의 DFD를 (그림 1)과 (그림 2)에 도시한다 [22]. 그림에서 알 수 있듯이 수천/수만 개의 클라이언트의 이벤트 및 객체를 서버에서 일관되게 관리하는 것은 상당한 기술적인 도전 사항인 것이다. 이 대규모 플레이어 온라인 게임분야는 Ultima Online을 시발점으로 첨단 온라인 게임의 각축장이 되고 있다. 다음 절에서 대규모 멀티플레이어 온라인 게임에 관하여 기술한다.

### 5. 대규모 참여자 온라인 게임

대규모 플레이어 온라인 게임(MMPOG, Mas-



(그림 1) MMPOG 클라이언트 DFD



(그림 2) MMPOG 서버 DFD

sively Multiplayer Online Game: MMOG)은 MMORPG(Massively Multiplayer Online Role Playing Game, 대규모 멀티플레이어 온라인 롤플레이어 게임)라고 부르기도 한다. 후자의 경우는 MMORPG가 그 게임의 전반적인 진행 기본이 RPG(Role Playing Game)와 흡사한 면을 갖고 있다는 데서 붙인 이름이며, 그러나 게임 장르가 RPG 요소 외에도 슈팅, 모험 등의 요소를 포함하고 있어 RPG라고 단정할 수 없는 까닭에 전자로 통칭하는 경우가 많이지

고 있다.

원래 RPG는 탁상용으로 소개되었는데, 게임 보드와 소도구를 갖추고 플레이어들이 역할을 나누어 갖고 게임 마스터의 진행에 따라 주사위를 굴러가며 순서에 따라 게임을 진행하는데 세계적으로 VAMPIRE: THE MASQUERADE가 히트를 한 적이 있다. 탁상 RPG에 관하여 참고문헌[15]를 참고하라. 이 RPG가 컴퓨터화 되기도 하고 또한 MUD(Multi User Dungeons), MOO(MUD Object Oriented)로 진전하였으며 플레이어들은 텍스트 기반 롤플레이팅 게임을 진행하였다.

그리고 3D 그래픽스가 텍스트 기반의 MUD와 접목이 되었는데 그 효시가 3DO사의 Meridian 59 이다. 그래픽은 조잡하였지만 플레이어들은 탁상 RPG에서처럼 주사위 순서를 기다리지 않고도 자유롭게 움직일 수 있게 되었고, MUD에서처럼 “너는 쥐를 잡았고 2,398 포인트의 대미지를 입혔다”라는 텍스트 문자 대신에 직접 3D 화면에서 괴물과 상대하게 되었다.

그리고 1997년에 Origin Systems사에 의하여 (UO)Ultima Online이 등장하였으며 해당년도 말까지 전세계적으로 70,000 타이틀을 판매하였다[17]. 이 UO를 비롯하여 동시 플레이어가 수천 명 이상인 게임을 소위 MMPOG로 분류하며 현재 탑랭킹에 올라 있는 제품을 <표 8>에 도시한다.

MMPOG의 특징은 평균 체류 시간이 길고(보통 수 시간), 시작하여 멈추지 않는, 플레이어가 나가고 없어도 계속 진행되는 영속적 게임 공간에서 플레이어들끼리 채팅을 하고, 경제 시스템, 생태계를 구비한 가상 환경에서 게임을 진행한다는 것이다. 가상 공간은 광범위하여 플레이어 아바타가 종단하는 데에만 수 시간이 걸릴 정도이다. 최대 동시 사용자 수가 수만 명에, 평균 동시 사용자 수도 만 여 명 대이다. 서비스 운용 면에서도 CD 구입 대금뿐만 아니라 월정 회비를 기본으로 하기 때문에 좋은 비즈니스 모델을 제공하고 있다. <표 8> 외에도 2000년 3월 현재 조사에 의하면 약 70개의 MMPOG 상용 제품을 개발 진행중인 것으로 보도되고 있다[19].

<표 8> 탑랭킹 MMPOG

구분	게임 명	비고
Origin Systems	UO	출시 당시 39대의 서버 사용(중세 브리타니카 지형, 15개 도시, 9개 성당, 7개 지하실, 광야지대), 평균 동시 사용자 수 5,800명, 2.5D, 심리스 월드
마이크로소프트	Asheron's call	1999. 11. 출시, 동시 사용자 수 3천 명, 제작사: Turbine Entertainment, 3D, 심리스 월드, Zone 이동시 래그 감소
소니	EverQuest	현재 70만 카피 판매, 월정 \$9.89, 2001/1 현재 피크 동시 사용자 수 81,185명, 평균 6~7만 유지[18] 38개의 서버 사용, 게임 월드 당 3천 명 수용 3D, Zoned world
세가 소프트웨어	10SIX	2000년 하반기 출시[16] 1,000,000명 동시 참여 가능
국내	리니지, 바람의 나라	수천 명 동시 참여[14]

### III. 결론

이상에서 온라인 게임 서버 기술에 대한 동향을 기술하였다. Interactive Digital Software Association에 따르면, 콘솔게임 및 PC 게임을 합할 때에 '98년도에 이미 미국의 영화 산업을 앞지르는 63억 달러를 기록한 만큼 게임 산업은 엔터테인먼트 분야의 주류를 형성하였다[20]. 엔터테인먼트 산업을으로서의 컴퓨터 게임은 대중의 감각에 의하여 유지되고, 대중의 감각을 최대한 활성화하는 게임이 지속적으로 출시되어야 한다. 게임의 품질은 게임 디자이너의 창작력을 최대한 현실화 시켜주는 컴퓨터 기술에 종속된다. 그러나 현재, 플레이어 수가 수천 명 이상으로 확장되는 대규모 시스템 범주로 가면, 당장 기술 장벽에 부딪히게 된다.

대규모 멀티플레이어 온라인 게임은 2000년 3월 현재 UO, EQ, AC의 가입자 수는 약 400,000 여 명으로 집계되어, 베스트 셀러 스탠드 얼론 게임과는 아직 수익 면에서 훨씬 못미치고 있지만, 소니 온라인 엔터테인먼트 집계에 따르면 2002년경에 총 온라인 게임 시장은 2,700만 여 명에 이를 것으로 보고 있다.

이처럼, 온라인 게임 서버 기술은 시장 및 첨단 기술 개발 측면에서 중대한 도전장을 제시하고 있는 셈이다. 그러나, 전통적으로는 게임 기술 개발이 시장의 요구에 주도되고 있었기 때문에 기술 체계화 및 보편화 보다는 업체 고유 기술로 폐쇄화되는 경향이 짙었다. 반면 기술 공개 및 교류가 보다 활발한 학계, 연구계에서는 분산 VR의 연구가 진행되어 왔으며, 여기서는 범용 데이터 처리가 주안점으로, 제한된 데이터를 다루는 게임과는 차이가 있지만, 대규모 참여자를 대상으로 한다는 점에서는 온라인 게임과의 공통의 관심사항이 될 수도 있다. 사실 분산 VR은 응용요구가 제한(군사용 시뮬레이션 분야만이 대규모의 참여자를 요구하고 있음, DIS: 300명 규모, 현재 100,000~300,000명 참여자를 위한 후속 기술 개발이 지속되고 있음)되어 왔으나, 온라인 게임에서는 그것 자체가 핵심이슈이기 때문에 상호 보완적이라 할 수 있으며, 이러한 맥락에서 산, 학, 연의 공조가 성과를 맺을 수 있는 분야라고 할 수 있다.

게임 엔진 전문 회사인 Numerical Design의 조사에 의하면 게임 개발 비용 중 40~70%는 자체 게임 엔진을 개발하는 데 소요된다고 한다[4]. 재사용 가능한 게임 엔진을 사용한다면 개발비를 급격히 절감할 수 있다는 얘기이다. 국내 게임 산업의 육성 관점에서도 국산 게임 서버 엔진 기술의 개발은 절실하다고 하겠다.

특히 서버 기술은 게임 서버 클러스터 및 고속 네트워크 링크 설비 등으로 고가의 투자가 요구되므로 중소 게임 업체에서 확보하기에 애로 사항이 많은 분야이다. 서버 기술은 이론 및 실험실 연구 외에 대규모 참여자가 참여하는 현장 시험이 무엇보다도 중요하다. 대규모 참여자의 안정적인 상호 작용을 지원하는 서버 기술을 개발하고 게임 업계에서 경제적으로 도입 사용할 수 있도록 해야 한다.

향후, 당분간 분산 서버 기반으로 스케일러빌리티를 높여 나가는 것이 과제이겠지만, 분산 가상 현실 분야에서도 제기되고 있듯이 장기적으로는 3D 가상 공간에서 전인류가 만날 수 있는 기술을 개발하는 것이다. 이러한 글로벌한 규모에서는 기술적으로는 분

산 서버가 갖는 구조적 제약사항을 극복할 수 있는 완전 분산 처리 구조가 요구된다. 그 완전 분산 처리 구조가 어떤 모습이 될 것인지는 지금부터 풀어야 할 연구개발 과제이다. 글로벌한 규모에서는 게임 공간의 운용도 한 업체의 범주를 넘어 여러 업체가 상호 운용할 수 밖에 없는 상황이 된다. 게임 산업계로 볼 때에는 먼 미래의 얘기이지만 분산 VR의 입장에서 볼 때에 그것은 바로 당면한 과제이다.

## 참 고 문 헌

- [1] S. Singhal and M. Zyda, *Networked Virtual Environments Design and Implementation*, Addison-Wesley, 1999.
- [2] <http://lcs.www.media.mit.edu/groups/el/projects/sp-acewar/>
- [3] <http://www.salon.com/tech/feature/1999/04/21/battle.net/>
- [4] <http://www.ndl.com/wpapers/bizcase.html>
- [5] S. Singhal *et al.*, InVerse: Designing an Interactive Universe Architecture for Scalability and Extensibility, HPDC '97, *IEEE Computer Society*, Aug. 1997.
- [6] [http://www.gamasutra.com/features/20000121/upton\\_pfv.htm](http://www.gamasutra.com/features/20000121/upton_pfv.htm)
- [7] Wats en, Kent and Zyda, Michael. "Bamboo - A Portable System for Dynamically Extensible, Networked, Real-Time, Virtual Environments," *Proceedings of VRAIS 98*, Atlanta, GA, Mar. 16-19, 1998, pp. 252 - 259.
- [8] Groove Networks, Inc., <http://www.Groovenetworks.com/>
- [9] Yu-Shen Ng, Designing Fast-Action Games For The Internet, Gamasutra, Sep. 5, 1997.
- [10] Ben Calica, Multi-player Lobbying, Gamasutra, Sep. 4, 1998.
- [11] Andrew, Chris Kirmse, Security in Online Game, Gamasutra, July 7, 1997.
- [12] Chris Crawford, The Art of Computer Game Programming, <http://members.nbci.com/kalid/art/art.html>
- [13] Costikyan, I Have No Words & I Must Design, Costikyan, I Have No Words & I Must Design, <http://www.stanford.edu/~kiky/KSLStuff/IHaveNoWords.html>



- [14] 라도삼, 가상공간의 전경과 삶의 단편들: '리니지'를 중심으로, 한국언론정보학회 추계 학술 대회, 1999.
- [15] RPG Information, <http://www.rpgconkor.net/>
- [16] 10SIX, <http://www.heat.net/10sixchannel/10sixfrontpage.html>
- [17] Scott S. Fisher, Real-time Interactive Graphics in Computer Gaming, ACM SIGGRAPH, May 1998.
- [18] [http://www.verant.com/press\\_releases.html#7](http://www.verant.com/press_releases.html#7)
- [19] <http://rpgvault.ign.com/features/specials/gdc2000.shtml>
- [20] <http://www.gamecenter.com/News/Item/0,3,02782,00.html>
- [21] C. Carlsson and O. Hagsand, DIVE-A Platform for Multiuser Virtual Environment, Computer & Graphics, Nov. 1993.
- [22] <http://www.arianne.cx/download>