에이전트에 의한 온라인게임 서버 테스트 자동화

정회원 이 헌 주*, 정 용 우*, 임 범 현*, 심 광 현*

Automated Testing of Online Game Servers by Agents

Hunjoo Lee*, Yong-Woo Jung*, Bum-Hyun Lim*, Kwang-Hyun Shim* ***Reguler Members***

요 약

게임 분야에서 온라인 게임은 높은 게임성과 안정적인 수익 모델로 인하여 많은 수가 개발되어 왔다. 특히, MMORPG는 게임의 특성상 많은 수의 사용자가 게임 서버에 접속하여 실시간으로 게임 서비스를 이용하게 된다. 따라서 게임 서버의 안정성은 매우 중요한 요소가 되므로 온라인 게임 개발 시에 장기간의 베타테스트를 통하여 게임 서버의 안정성을 실험하게 된다. 이러한 테스트는 많은 시간과 비용이 요구되는 과정이지만 반드시 필요한 과정이기도 하다. 본 논문은 온라인 게임의 서버를 테스트하기 위한 방법에 관한 것이다. 본 논문에서 제안된 시스템은 가상의 게임 클라이언트와 이들을 관리하는 에이전트를 이용하여 게임 서버를 자동으로 테스트할 수 있다. 제안된 방법을 온라인 게임 콘텐츠에 적용하여 다수의 동시 접속자 환경 하에서의 서버 성능 및 게임 관련 데이터를 효과적으로 모니터링 할 수 있음을 보였다.

Key Words : game server, online game, agent, client

ABSTRACT

In this paper, we present an efficient method for simulating massively virtual clients in an online game environment. Massively multi-player online games and other multi-user based networked applications are becoming more attractive to the gamer players. Such kind of technology has long been researched in the area called Networked Virtual Environments. In the game development process, a set of beta tests is used to ensure the stability of online game servers. A set of testing processes consumesa lot of development resources such as cost, time, and etc. The purpose of VENUS system is to provide an automated beta test environment to the game developers to efficiently test the online games to reduce development resources.

Ⅰ. 서 론

최근 게임 분야에서 온라인 게임은 양적으로나 질적으로 급성장을 보이고 있다. 뿐만 아니라 PC 기반의 온라인 게임 환경은 점차 게임 전용 콘솔, PDA, 휴대전화 등으로 그 영역을 넓혀가고 있는 추세이다[1]. 과거 온라인 게임 분야에서는 'Killer Application'이 확실치 않은 상황이었으나 Ultima Online, Everquest 등의 성공은 온라인게임 분야에서 MMORPG(Massively Multiplayer Online Role Playing Game) 장르가 가장 'Killer Application'에 근접하고 있음을 보여주게 되었다[2].

현재 다른 장르의 온라인게임들이 패키지 게임의 서비스 차원이거나 광고 효과를 노린 비교적 단순한 형태인 것에 비해, MMORPG는 게임 자체로 많은 수익 모델을 추구할 수 있게 됨으로 사용자에게 안정적인 서비스를 제공하는 것이 무엇보다 중요하게 되었다. 따라서 온라인 게임 서버의 안정성을 위하여 개발 과정에서 많은 테스트를 하게 된다[3]. 일반적으로 많이 사용되는 테스트 방법인 베타 테스

* 한국전자통신연구원 디지털콘텐츠연구단 게임기술개발센터
논문번호 : KICS2006-01-008, 접수일자 : 2005년 4월 12일, 최종논문접수일자 : 2006년 4월 18일

트는 게임의 상용화 직전에 게임 테스터를 모집하여 이들에게 시범 서비스를 제공하고 일정 기간 동안 시스템의 안정성을 검증하는 방식으로 이루어진다[4].

하지만 베타테스트는 몇 가지 비효율적인 문제점을 가지고 있다. 첫째로 테스트를 위하여 베타테스터를 모집하여야 하고, 게임의 테스트를 위해 장기간의 시간과 많은 비용이 소요된다. 둘째로 베타테스터는 게임의 문제점을 명확하게 제공할 수 없기 때문에 개발자가 정확히 상황을 파악하기 힘들다. 셋째로 게임의 진행 중에 실시간으로 문제가 발생되기 때문에 수정을 위해 문제가 발생된 상황을 정확히 재현하기 힘들다.

가상의 사용자를 이용한 서버 시험 기술은 파일 서버[5], 웹 서버[6] 등의 영역에서 연구가 되어 왔다. 또한 최근에는 온라인 게임 서버 분야에서도 베타테스트 시의 문제점을 보완하기 위하여 게임 서버 시험의 자동화 기술에 관하여 연구[7-9]되고 있다. 가상 사용자를 이용한 서버 시험 기술은 서버 테스트용 컴퓨터에서 가상의 이용자를 생성하여 서버 시스템에 부하를 주는 것이 일반적인 방법이다. 하지만 현재까지의 연구는 서버 테스트 시스템이 특정 게임에 종속되어 있으며, 또한 소규모 이용자를 대상으로만 시뮬레이션이 이루어지도록 되어 있다는 한계를 지니고 있다.

본 논문에서 제안하는 온라인 게임 서버 테스트를 위한 시스템은 가상의 사용자를 이용하여 대규모의 온라인 게임 서버를 자동으로 시험할 수 있는 시스템이다. 본 논문은 다음과 같이 구성된다. II장에서 제안하는 방법의 구조에 대하여 설명하고, III장에서는 제안된 시스템을 위하여 사용되는 프로토콜에 대하여 기술한다. IV장에서 온라인 게임을 대상으로 실험 및 적용한 결과를 기술하며, V장에서 결론 및 추후 연구과제에 대하여 기술하고 끝을 맺는다.

Ⅱ. 시스템 구조

본 논문에서 제안하는 방법은 그림 1과 같은형태의 구조를 가지고 있다.

시스템의 구성요소는 Virtual Client(VC), VC Agent, Central Engineering Station(CES), 게임 서버 등으로 이루어진다. VC는 가상의 사용자로 게임의 실제 이용자와 동일한 역할을 수행하며, 제안하는 시스템에서 제공하는 VC 엔진을 이용하여 VC

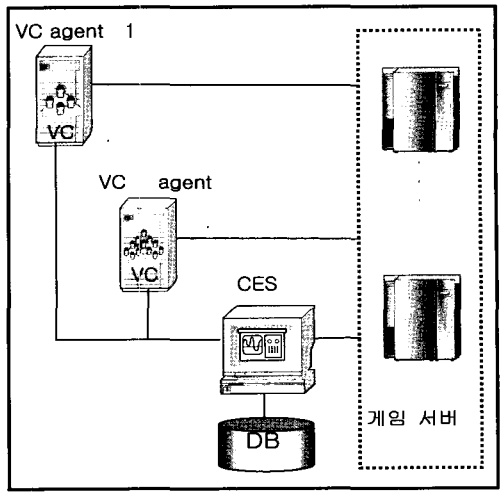


그림 1. 시스템 구조

를 자유롭게 만들 수 있다. VC agent는 VC를 구동하는 테스트 컴퓨터에서 데몬(Daemon) 프로그램으로 구동되며, CES로부터의 명령을 해석하여 VC를 제어하는 역할을 한다. CES는 사용자(게임 개발자 혹은 게임 테스터)의 입력을 받아 테스트 환경을 설정하고, VC Agent들을 관리 및 제어하며, 각 구성요소의 모니터링 결과를 DB에 저장하고 사용자에게 보여주는 역할을 한다.

본 시스템은 대규모의 사용자가 접속하는 상황을 시뮬레이션 할 수 있도록 하기 위하여 CES가 VC agent를 관리하고, VC agent는 VC를 관리하는 구조로 설계하였다. VC agent가 설치된 i번째 컴퓨터의 구동 가능한 총 VC의 수를 k_i라고 하면, 본 시스템이 구동 가능한 총 VC의 수 T는 다음과 같은 식으로 표현된다.

$$T = \sum_{i=1} k_i \tag{1}$$

(1)에서 보는 바와 같이 제안하는 시스템에서는 VC를 구동하는 컴퓨터 i를 증가시킴으로써 VC를 증가시킬 수 있다. 이러한 구조는 기존의 연구들이 하나의 컴퓨터에서 시험하던 것에 비해 월등히 많은 수의 VC를 이용하여 게임 서버를 시험할 수 있도록 해준다

Ⅲ. 통신 프로토콜

본 시스템에서는 세 가지의 통신 프로토콜을 사용한다. 첫째로 CES가 VC agent, Test Server

Cluster 및 Test Game Client와 통신을 하기 위한 VENUS Protocol, 둘째로 VC agent와 VC 사이의 통신을 담당하는 VC Engine Protocol, 셋째로 시스템 모니터링에 관련된 Observer Protocol 이다.

3.1 VENUS Protocol

그림 2에서 보듯이VENUS Protocol은 VC agent, Test Server Cluster 및 Test Game Client의 제어를 수행하기 위한 통신 프로토콜이다. VENUS Protocol의 세부 기능은 각 구성요소의 연결 초기화, 시뮬레이션 관리 등을 수행하는 것이다.

VENUS Protocol의 연결 초기화 기능은 CES가 각 구성요소와 TCP/IP 연결을 생성하고, VID (VENUS Identification)를 할당하는 기능이다. VID의 정의는 그림 3과 같다. VID의 Type과 Major Index는 CES에서 생성되고, Minor Index는 VC agent에서 생성된다.

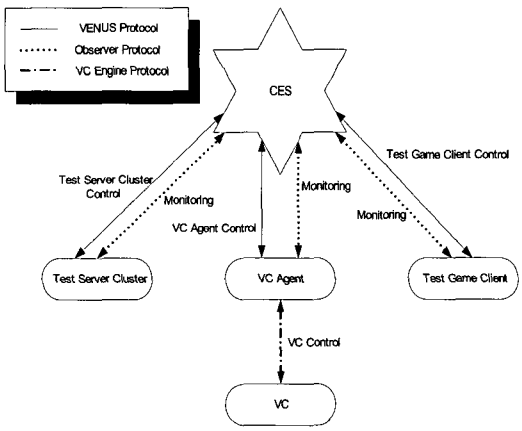


그림 2. VENUS Protocol

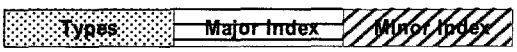


그림 3. VID 정의

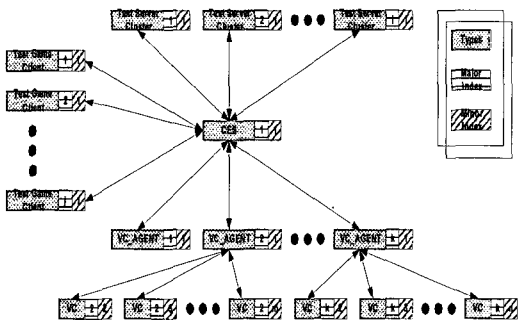


그림 4. VID의 예

그림 4는 각 구성요소의 수가 i(Test Server Cluster), j(Test Game Client), k(VC agent), m (VC), n(VC) 일 때 VID의 할당 예를 보이고 있다.

제안된 시스템은 VENUS Protocol의 시뮬레이션 관리 기능을 통하여 전체의 시뮬레이션을 설정하고 종료하게 된다. VENUS Protocol은 Test Server Cluster, Test Game Client 및 VC Agent 각각의 고유의 시뮬레이션 환경 설정 및 종료를 위한 통신 프로토콜로 이루어진다.

3.2 Observer Protocol

Observer Protocol은 일반적인 컴퓨터의 성능을 측정하고, 사용자가 게임 내부의 정보를 측정할 수 있도록 해준다. 또한 VC agent의 VC가 호스트로부터 받는 서비스의 처리시간을 측정할 수 있도록 해준다.

성능 측정 기능은 현재의 컴퓨터의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등을 포함한다. 제안된 시스템은 세가지 성능 측정치를 바탕으로 Test Server Cluster, Test Game Client 및 VC Agent를 구동하는 컴퓨터의 상태를 파악하게 된다. 특히 VC agent에서 성능 측정치는 VC를 구동하는 호스트 컴퓨터가 구동 가능한 최대 VC의 수를 결정하는데 사용된다. 한 호스트에서 구동되는 i번째 VC의 CPU, 메모리, 네트워크 사용량을 c_i, m_i, n_i이라고 할 때, VC가 사용하는 각 자원 별 총 성능 C_i, M_i, N_i을 식 (2)와 같이 정의한다.

$$C_i = \sum_{i=1} c_i \quad M_i = \sum_{i=1} m_i \quad N_i = \sum_{i=1} m_i \tag{2}$$

또한 C_i, M_i, N_i의 normalize 값은 각각 식 (3)과 같이 표현된다.

$$\bar{C}_i = \frac{C_i}{C_{\max}} \quad \bar{M}_i = \frac{M_i}{M_{\max}} \quad \bar{N}_i = \frac{N_i}{N_{\max}} \tag{3}$$

일반적으로 i가 증가함에 따라서 $\bar{C}_i$, $\bar{M}_i$, $\bar{N}_i$는 증가하게 된다. 따라서 하나의 호스트에서 구동 가능한 최대 VC의 수 $i_{\max}$는 식 (4)의 조건을 공통으로 만족하는 최대값이어야 한다.

$$\left|\bar{C}_i\right|_\infty \le 1 \quad \left|\bar{M}_i\right|_\infty \le 1 \quad \left|\bar{N}_i\right|_\infty \le 1 \tag{4}$$

제안된 시스템의 VC agent는 (4)를 이용하여 호

스트의 성능을 초과하는 VC의 생성 및 구동을 제한할 수 있다.

Observer Protocol은 Test Server Cluster의 게임 내부 정보를 모니터링 한다. 초기화 과정에서 모니터링 할 데이터의 형식을 Test Server Cluster가 CES에 전달하고, 해당 데이터가 발생할 때 데이터를 전송한다.

그림 5에서 구동과정을 보면 Test Server Cluster가 'Total Data Pattern'으로 전송할 사용자 데이터 패턴의 개수를 설정하고 전송할 데이터의 패턴 정의를 CES에 전송한다. CES는 내부에 데이터베이스를 생성, 추후에 들어올 데이터를 저장할 준비를 하게 된다. 준비 과정이 완료되면 Test Server Cluster가 예에서 보이는 형식의 데이터를 사용자가 지정한 시점에 CES로 전송하게 된다.

Observer Protocol에서는 VC가 Server에서 받는

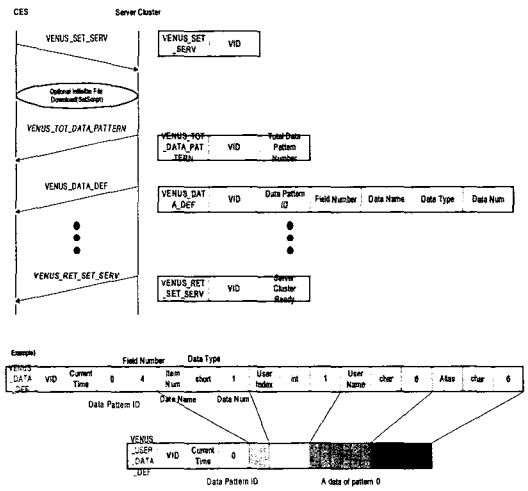


그림 5. User define data의 설정 및 예

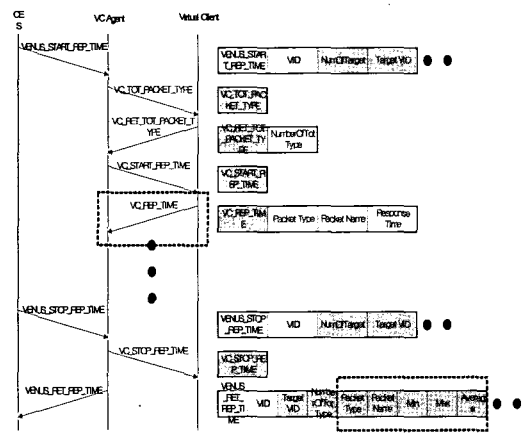


그림 6. 서비스 응답시간 측정

서비스의 시간을 측정하여 CES로 전송하는 기능을 가지고 있다. 이를 통해 Game Client가 Game Cluster에 Login, Logout, Moving Information, Inventory Data Update, Battle Data Update 등과 같은 다양한 클라이언트 서비스를 이용자의 필요에 따라 측정할 수 있다. 측정 결과를 바탕으로 Game Client가 정상적으로 게임을 진행하는데 필요한 Login Server, Game Server, DB Server 등의 각 서버 수를 조절할 수 있다.

3.3 VC Engine Protocol

VC Engine Protocol은 VC agent가 VC를 제어하는 데 사용되는 통신 프로토콜로 사용자는 VC 엔진을 이용하여 VC를 제작하고 본 시스템과 연동시키게 된다. VC 엔진은 앞에서 언급한 VENUS Protocol 및 Observer Protocol을 지원하는 기능과 VC 제어 명령 기능을 포함하고 있다.

제안된 시스템에서 정의한 VC 제어 명령은 VC 이동 명령을 'VC_CIRCULAR_MOVE'와 'VC_RECTANGLE_MOVE'의 2가지 형태로 정의하고 기타 VC 제어 명령을 'VC_ACTION'으로 정의한다. 또한 VC의 캐릭터 특성을 'VC_CHARACTERISTC'으로 정의한다.

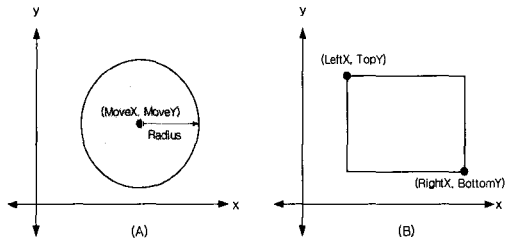


그림 7. VC_circular_move(A)와 VC_rectangle_ move(B)

VC_CIRCULAR_MOVE와 VC_RECTANGLE_MOVE 제어 명령은 현재의 VC의 위치에서 이동할 목표 지점을 묘사하는 것으로 목표지점을 원이나 사각형 영역으로 표현 하도록 하였다.

VC 이동 명령은 그림 8과 같이 목표지점에 도달하지 않은 상태에서 새로운 이동 명령이 내려오면 현재 좌표에서 해당 좌표로 이동을 하여야 한다.

'VC_CHARACTERISTIC'은 VC의 특성을 설정하기 위해 정의된다. 그림 9는 VC의 특성을 전투 호전성으로 이용할 때 사용한 예이다. VC 특성이 12일 때가 11일 때에 비하여 전투 호전성이 높다고 말할 수 있다. 전투 호전성을 바꿈으로 해서 Game

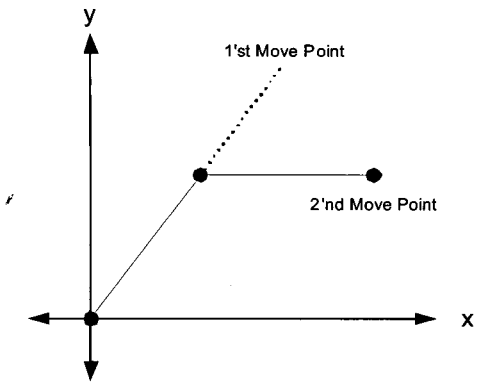


그림 8. VC의 이동

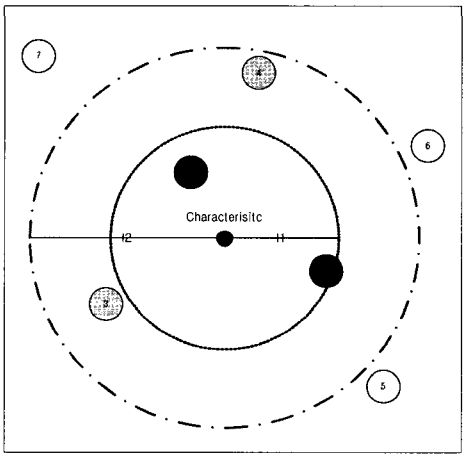


그림 9. VC_CHARACTERISTIC의 예

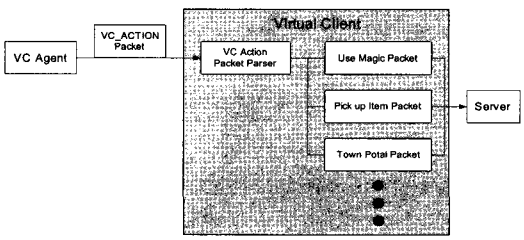


그림 10(a). VC_ACTION의 파싱 예

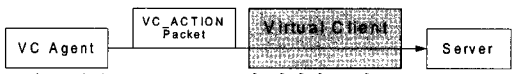


그림 10(b). VC_ACTION의 바이패스 예

Server와 DB Server에 전투 상황 및 결과를 증가시켜 안정성을 테스트할 수 있다.

'VC_ACTION'은 그 이외 VC를 제어하는데 있어 필요한 모든 명령을 사용자가 원하는 형태로 전송할 수 있다. VC agent는 CES에서 내려오는 'VC_ACTION' 관련 명령을 변형없이 전달하기 때문에 사용자는 그림 10의 (b)와 같이 Server가 알 수 있는 명령을 내려 보내거나, 그림 10의 (a)와 같이 들어온 'VC_ACTION'을 변형하여 사용할 수 있다.

Ⅳ. 실험 및 적용

본 장에서는 제안된 시스템을 간단한 온라인 게임을 이용하여 실험한 결과와, 베타테스트 중인 온라인 게임에 적용한 예에 대하여 기술한다.

4.1 실험

테스트 환경은 그림 11과 같이 8대의 VC 호스트와 1대의 서버 그리고 1대의 CES 및 게임 클라이언트용 컴퓨터가 사용되었다. 그림 12의 (a)는 각각의 VC 호스트에 100개(총 800개)의 VC를 생성한 후에 로그인, 로그아웃을 4회 반복시켰을 때의 Response Time, CPU, Network, Memory 사용량을 보이고 있다. Response Time의 서비스는 각 VC가 로그인하는데 걸린 평균시간, Echo 서비스에 걸리는 시간으로 설정되었다. 그림 12의 (a)에서 Server Cluster의 CPU 사용량은 10%-100%사이로 변동이 4회 발생한 곳이 로그인을 실행하는 곳으로 동일한 시간대에 Network 사용량이 0%-10%로 변동함을 볼 수 있다. 또한 Response Time에서 로그인에 걸리는 시간이 33ms93ms70ms66ms으로 변동하는 것을 볼 수 있다.

그림 12의 (b)는 VC를 그림 12의 (c)에서 보이는 자신이 현재 위치한 셀(cell) 영역 내에서 random하게 이동시켰을 때dp Test Server Cluster의 상황을 보이고 있다. CPU 사용량이 평균 70%, Network 사용량이 평균 20% 정도를 보이고 있으며 서버의 Echo 서비스 시간이 (a)에 비해 월등히 증가하고 있다. 이는 서버 내에 부하가 지속적으로 할

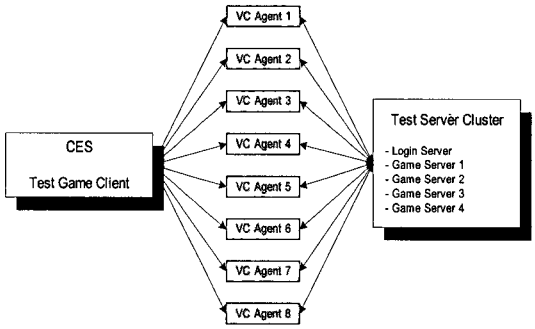
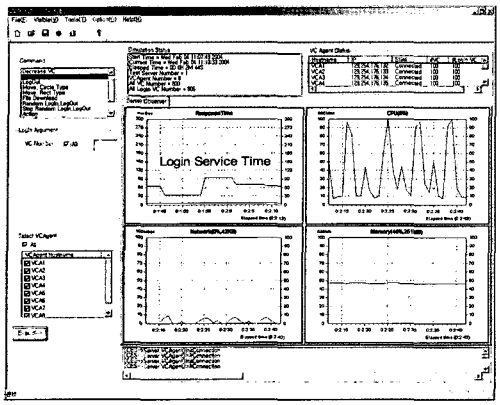
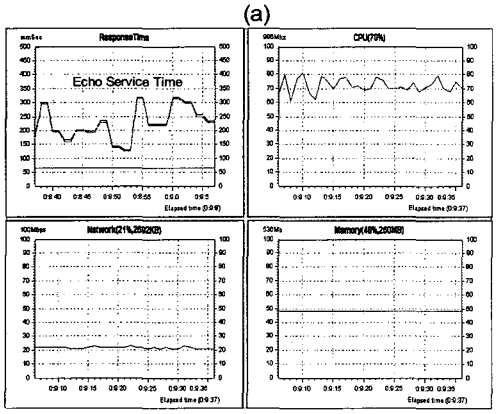


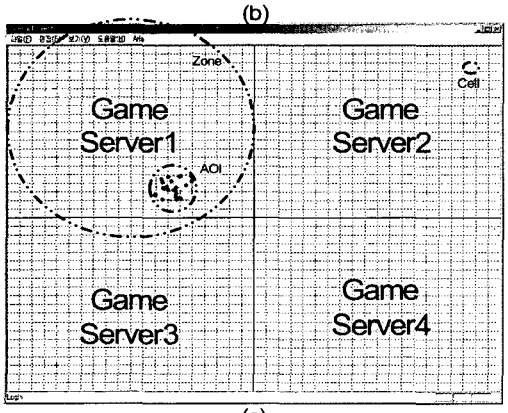
그림 11. 테스트 환경



(a)



(b)



(c)

그림 12. 실험 결과

당되면서 Echo 서비스에 시간지연이 나타남을 보이고 있는 것이다.

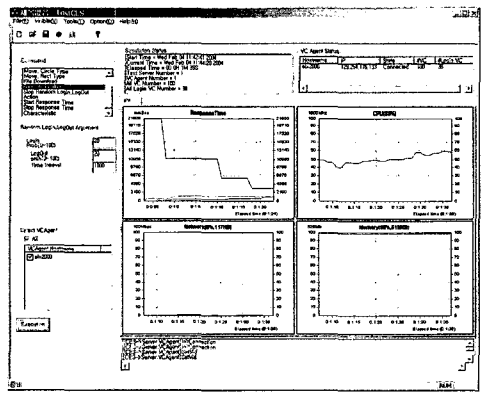
그림 12의 (c)는 실험에 사용된 Test Game Client를 보이고 있다. 각4개의 Game Server는 그림에서 보이는 4개의 game zone을 담당한다.

본 실험에서 제안된 시스템을 이용하여 게임 시스템의 로그인, 로그아웃의 안정성 테스트, random 로그인, 로그아웃 테스트, 셀 이동, game zone 이동 등을 수행하였다.

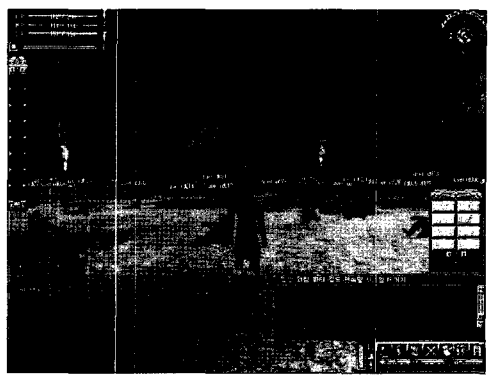
4.2 적용

다음은 제안된 시스템을 실제로 개발 중인 온라인 게임에 적용해 본 결과를 보인다. 적용된 온라인 게임의 game server는 크게 Login Server, World Server, Zone Server로 이루어진다. 이들의 안정성을 테스트하기 위해 Response Time은 하나의 로그인 과정에서 각 서버가 처리하는데 걸리는 시간을 표현하게 정의하였다.

그림 13의 (a)는 적용 게임에 VC를 로그인시킬 때의 CES를 보이고 있다. Login Server의Response Time이 차차 줄면서 World Server, Zone Server의 Response Time이 증가함을 알 수 있다. 그림 13의 (b)는 로그인 중인 VC를 Test Game Client를 통하여 본 것이다. VC_ACTION을 이용하여 전체 캐릭터를 제어하도록 하거나 VC_CIRCULAR_MOVE를 이용하여 특정 위치로 이동하도록 하는 등의 실험에 적용한 결과이다.



(a)



(b)

그림 13. 적용 예

V. 결 론

일반적으로 온라인 게임 시스템의 안정성을 검증하기 위하여 수행되는 테스트 방법은 시간과 비용 측면에서 효율적이지 않다. 본 논문에서는 이러한 문제를 해결하기 위하여 온라인 게임의 서버를 효율적으로 테스트하기 위한 시스템을 제안하였다.. 제안된 시스템은 대규모 사용자의 온라인 게임 환경을 시뮬레이션 할 수 있다. 다양한 시뮬레이션 환경을 지원할 수있도록 일반적인 VC의 제어 명령을 구현하였으며, 또한 각 구성 요소의 성능 지수 및 게임 내부의 정보를 모니터링 할 수 있도록 하였다. 그리고 사용자의 정의에 의하여 사용할 수 있는 Response Time 등을 통해 온라인 게임 시스템의 구축에 도움이 될 수 있도록 하였다.

향후에 VC를 지능적으로 구동시키기 위하여 인공지능 기술을 적용하는 연구가 필요하다. 또한 사용이 보다 편리하도록 사용자 인터페이스 및 시뮬레이션 환경 설정 과정을 보완하는 연구가 이루어질 예정이다.

참 고 문 헌

[1] DFCIntelligence, "The Online Game Market 2003", DFC Intelligence, Jun 2003.

[2] David Arneson, Thomas Ruwart, "A Test Bed for a High-Performance File Server", 'Putting all that Data to Work'. Proceedings., Twelfth IEEE Symposium on Mass Storage Systems, pp.26-29 Apr. 1993.

[3] Elbaum, S., Karre, S., Rothermel, G., "Improving web application testing with user session data", Proceedings. 25th International Conference on Software Engineering, 3-10, pp.49-59, May 2003.

[4] Larry Mellon, "Automated Testing of Massively Multi-Player Systems : Lessons Learned from The Sims Online", GDC 2003, Spring.

[5] Adobbati, R., A.N., Scholer, A., Tejada, S., Kaminka, G.A., Schaffer, S., Sollitto, C. "Gamebots : A 3D Virtual World Test-Bed for Multi-Agent Research", Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS, Montreal, Canada, 2001.

[6] Wu-chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole, "Provisioning Online Games : A Traffic Analysis of a Busy Counter-Strike Server", Proceedings of the second ACM SIGCOMM Workshop on Internet measurment workshop, Nov., 2002.

[7] K.H.Shim, etc., "Design and Analysis of State Update Rate Control Schemes for Performance Improvement in Networked Virtual Environments", Proc. of the SCI 2002, August, 2002.

[8] 정용우 등, "대규모 가상 이용자를 이용한 온라인 게임 시뮬레이션의 제작에 관한 연구", 한국컴퓨터게임학회 논문지, 제3호, pp.70-76, 2003.

[9] Andrew Rollings, Dave Morris, "Game Architecture and Design. Coriolis", 2000.

[10] Mark DeLoura, "Game Programming Gems 3", Charles Rivermedia, 2002.

[11] 배재환, "분산 게임 서버 시스템 설계에 관한 연구", 한국통신학회 논문지, 제28권 12B호, pp.1060-1065, 2003.

[12] Watt, A. & Policarpo, F., "3D Games: Real-Time Rendering and Software Technology" Addison-Wesley, 2001.

이 헌 주(Hunjoo Lee) 정회원

1991년 중앙대학교 컴퓨터공학과 학사

1993년 중앙대학교 컴퓨터공학과 석사

1998년 중앙대학교 컴퓨터공학과 박사

2001년~2002년 Iowa State University Post-Doc.

1998년~현재 한국전자통신연구원

<관심분야> 인공지능, 게임엔진

정 용 우(Yong-Woo Jung) 정회원

1999년 2월 충남대학교 메카트로닉스공학과 학사

2005년 2월 충남대학교 메카트로닉스공학과 석사

2002년~2005년 한국전자통신연구원

2005년~현재 LG전자 DS연구소

<관심분야> 네트워크, 로봇, 프로그래밍

임 범 현(Bum-hyun Lim) 정회원

1999년 2월 전북대학교 전자공학과 학사
2002년 2월 전북대학교 전자공학과 석사
2002년~현재 한국전자통신연구원
<관심분야> 시뮬레이션, 병렬처리, 가상현실

심 광 현(Kwang-Hyun Shim) 정회원

1993년 2월 KAIST 전기및전자공학과 석사
1998년 2월 KAIST 전기및전자공학과 박사
1998년~현재 한국전자통신연구원 크로스플랫폼게임연구팀 팀장
<관심분야> 네트워크 혼잡 제어, 실시간 통신 프로토콜, 네트워크 가상환경, 대규모 분산 서버