온라인게임 서비스 안정화

Online Game Service Stabilization

융합 시대를 주도할 디지털콘텐츠 기술 특집

최용준 (Y.J. Choi) DC협동연구팀 연구원
박성수 (S.S. Park) DC협동연구팀 연구원
김재원 (J.W. Kim) DC협동연구팀 연구원
이범렬 (B.R. Lee) DC협동연구팀 팀장

목 차

* 본 연구는 온라인게임 산업 육성을 위한 정보통신부의 게임기술지원센터 운영 사업의 일환으로 수행되었음.

온라인게임에서 가장 많은 문제가 발생하는 시기는 오픈베타 이후의 일이다. 적은 인원으로 테스트를 진행하는 클로즈베타에서 경험하지 못한 게임의 버그가 발생할 수 있고 게임사용자의 폭주를 사전에 예상하지 못해 게임서비스가 마비되는 경우도 발생한다. 또는 불법해킹에 의해 서비스가 중단하는 사태도 발생한다. 게임개발사에서는 오픈베타에 앞서 여러 번의 클로즈베타 테스트를 진행하는 동안 게임 내의 버그를 잡는 과정을 거친다. 시스템 및 네트워크 설계, 게임 및 웹서비스 보안, CDN, 로드밸런싱, 콘텐츠동기화, 모니터링, 가상유저테스트 등의 기술은 온라인게임 서비스 안정화를 위한 필수적인 요소이고 향후 국내와 해외 상용화 서비스의 근간이 된다. 본 고는 온라인게임 서비스 안정화를 위해 필요한 기술적 고려사항에 대해 다루고자 한다.

Ⅰ. 서론

온라인게임에서 가장 많은 문제가 발생하는 시기는 오픈베타 이후의 일이다. 클로즈베타에서 경험하지 못한 게임의 버그가 발생할 수 있고 게임유저의 폭주로 인해 게임서비스가 중단되는 경우도 발생한다. 온라인게임은 일반 패키지 게임이나 콘솔 게임과 달리 지속적인 게임 업데이트가 필요한 서비스이다. 온라인게임 서비스는 클로즈베타 단계를 거친 후에 오픈베타를 진행하고 일정 시기가 지난 후 정액제 형태의 유료화나 아이템판매 등의 부분유료화를 진행한다.

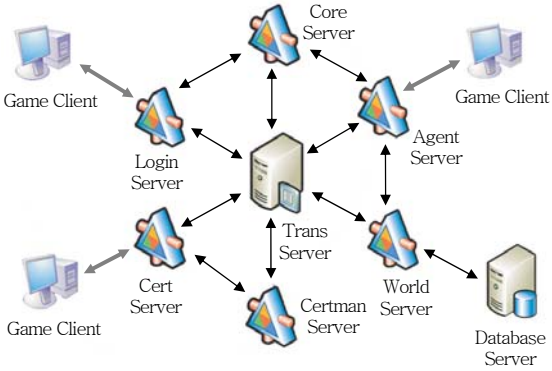
게임개발사에서는 클로즈베타 테스트를 2~3회 정도 진행하는 동안 게임 내의 버그를 잡는 과정을 거친다. 짧은 클로즈베타 테스트 기간에 얼마나 버그를 효율적으로 잡느냐가 향후 오픈베타 시에 사업의 성패와 직결되는 영향을 준다. 출시되는 온라인게임이 많다 보니 게임유저의 눈높이가 높아진 국내 환경에서는 클로즈베타 단계의 게임에 대해 높은 게임 완성도와 안정성의 확보를 요구하게 되었다.

또한 게임개발사에 자금을 투자하는 퍼블리셔, 포털에서도 여러 사항에 맞는 테스트 결과 리포트와 디버깅 리포트를 개발사에 요구하고 클로즈베타 결과 내용을 바탕으로 개발사와 계약하는 실정이다. 본 고에서는 온라인게임 서비스 안정화를 위해 필요한 기술적 고려사항에 대해 다루고자 한다.

Ⅱ. 안정화를 위한 기술적 고려사항

1. 시스템/네트워크 구성

게임서비스의 성능개선과 네트워크 병목현상을 최소화하기 위해 필요한 가장 기본적인 방법으로는 '서버 대 서버', '서버 대 클라이언트' 통신을 구분((그림 1) 참조)하여 서버간의 통신은 사설 네트워크 스위치를 이용하도록 구성하고 서버와 클라이언트 통신은 공인 네트워크 스위치를 통하도록 구성한다. 스위치와 연결되는 서버 네트워크 카드의 speed와



(그림 1) 게임 통신의 분산 구조

duplex 타입 설정을 확인하고 스위치에 연결된 각 포트의 error 패킷, CRC, collision을 반드시 확인해야 한다.

관리하고 있는 게임서버가 많을 경우에는 먼저 사설 네트워크 인프라 구성을 완료하고 사설 네트워크 DNS 환경을 구축한다. 사설 DNS 서버 구성[1]이 완료되면 다른 서버를 찾기 위한 이름풀이 방법으로 DNS를 이용할 수 있다. 윈도 서버의 통합계정관리, 보안관리, 중앙자원관리 기능을 이용하기 위해서는 LDAP 기반의 AD 환경을 구축해야 한다.

데이터베이스 서버의 보안을 높이고자 사설 네트워크만을 연결하고 인터넷 연결을 제거하여 사용한다. 그러나 사설 네트워크의 다른 서버에서 보안 취약점이 발생했을 경우 평소 윈도 업데이트를 하지 않고 있었던 데이터베이스 서버 또한 보안 취약점에 노출되게 된다. 이런 문제점을 해결하기 위해 내부 서버들의 보안 업데이트 및 백신 업데이트를 진행하기 위해 사설 네트워크 환경에 WSUS(윈도 업데이트 서버)[2]와 중앙백신서버를 구축하여 보안 문제를 사전에 예방할 수 있다.

2. 패치시스템

패치 서버는 게임의 버전을 관리하여 클라이언트의 버전을 최신 버전으로 유지시키는 역할을 한다. 게임 서버나 클라이언트는 필요에 따라 수시로 업데이트가 일어나는데, 클라이언트와 서버의 버전이 동일해야만 게임 실행에 문제가 없다. 버그 패치나 게

임 서버의 새로운 기능 추가를 실시간으로 업데이트 하는 것은 어려운 작업이며 실행 파일의 리소스가 변경되면 게임 서버 프로세스를 재실행해야 한다.

게임 개발사에서는 게임 서버의 패치를 위해 정기 점검 시간을 통해서 진행한다. 패치 서버는 주로 클라이언트에 설치된 파일을 최신 버전과 비교하여 이전 버전의 클라이언트가 게임에 접속할 경우 패치를 다운로드 받을 수 있는 경로를 제공하여 업데이트를 진행한다. 일반적으로 HTTP 프로토콜과 FTP 프로토콜을 사용하게 되는데 과거에는 패치 프로그램의 버전 관리와 빠른 전송속도를 제공하는 FTP 방식을 많이 사용하였지만 네트워크 방화벽 및 사용자 컴퓨터 보안이 강화되면서 외부 FTP 접속에 많은 문제가 발생하게 되었다.

HTTP 방식은 FTP 접속의 문제를 극복하였지만 업데이트 리스트를 별도 파일로 둬서 이를 비교한 후 다운로드 하는 방식이고 FTP 방식보다 다운로드 속도가 떨어진다는 단점이 있다. HTTP의 단점과 FTP의 단점을 극복하기 위해 양쪽 프로토콜을 다 지원하는 패치시스템을 만들기 위해 게임 클라이언트 패치 시에 최초 FTP 서버로 접속을 시도하고 접속이 되지 않을 경우 HTTP 서버로 접속을 시도하게끔 구성을 할 수 있다. 국내의 경우 게임 클라이언트와의 다운로드 서비스 속도 최적화를 위해 CDN 서비스를 이용할 수 있지만 해외의 경우 여러 경로(포털, ISP 서버)에 다운로드 서버를 구성해야 하는 상황이 있을 수 있으므로 양쪽 프로토콜을 지원하는 패치시스템에 대해서 고려해야 한다.

동남아시아의 경우 네트워크 인프라가 과거에 비해 많은 발전을 하였으나 하나의 ADSL 라인에 공유기를 연결하여 PC방과 같이 사용하므로 느린 네트워크 전송속도 환경을 생각한다면 패치에 의해 업데이트 되는 게임 클라이언트의 파일 크기를 개발단계에서 고려해서 크기를 최소화해야 한다. 게임 클라이언트의 패치를 압축파일의 형태로 제공하여 압축해제 시에 자동 업데이트 되도록 하는 방법도 있지만 버전관리의 문제점이 발생하게 된다.

3. CDN 서비스

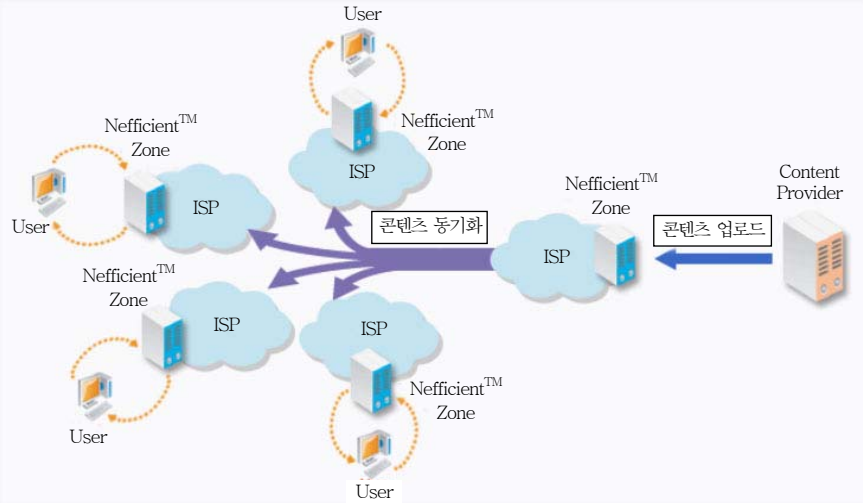
앞서 이야기한 패치서버는 게임 서버패치와 게임 클라이언트 패치 부분으로 구분되며 클라이언트 업데이트를 위해 기존 파일과 업데이트 리스트 정보를 비교한 후 다운로드 받아야 하는 파일의 정보와 다운로드 경로를 알려주는 역할을 한다.

CDN은 콘텐츠를 배달하고 분배하는 시스템이다. 인터넷 사용의 급증과 더불어 멀티미디어 콘텐츠가 일반화되면서 인터넷 사업자들은 더욱 양질의 서비스를 제공해야 되었고, 이에 따라 끊임없는 네트워크 및 시스템의 확장을 요구받게 되었다. 게임 서비스 업체에서는 게임 클라이언트((그림 2) 참조), 패치, 동영상 스트리밍, 웹콘텐츠 파일을 전달하기 위해 자체 서버와 네트워크 회선을 사용하는 대신 CDN 사업자와 계약을 통해 이러한 서비스를 진행하고 있다.

(그림 2) 게임 클라이언트 다운로드

CDN 서비스[3]의 주요 핵심기술로는 파일 동기화(sync)와 글로벌로드밸런싱(GSLB)으로 이루어져 있다((그림 3) 참조). 파일 동기화는 다수의 ISP/IDC에 서버팜을 구성하고 특정 서버에 업로드된 콘텐츠가 다수의 서버팜으로 자동 복제되는 구성을 의미한다. 복제되는 과정은 수동 또는 자동 서비스 형태로 제공된다. 인터넷이용자의 접속위치 IP 정보와 CDN 서버의 상태를 반영하여 글로벌로드밸런싱을 통해 최적의 전송 경로를 선택하여 콘텐츠를 제공한다.

콘텐츠를 제공하는 게임 업체에서는 CDN 서비스에 많은 비용을 지출하고 CDN 사업자의 경우도



(그림 3) CDN 서비스

네트워크 비용 절감을 위해 피어링(P2P) 기술과 그리드 기술을 CDN 서비스에 적용하고 있다. 게임업체에서는 비용최소화를 위해 웹하드 서비스를 제공하는 업체의 프로그램을 통해 다운로드중인 인터넷 사용자의 네트워크 대역폭을 사용하여 다운로드 요청을 처리하고 있다.

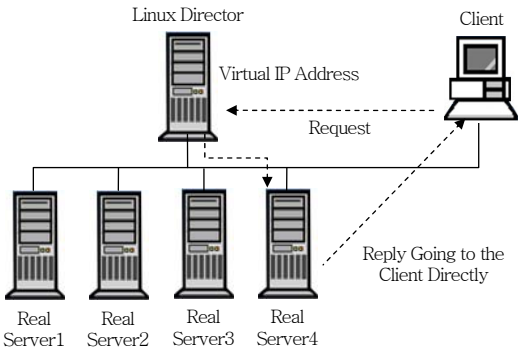
4. 콘텐츠 파일 복제 및 로드밸런싱

해외에 CDN 서비스를 진행하는 사업자가 없을 경우에는 자체 구축이 필요한데 이 과정에서 파일 복제 및 로드밸런싱에 대한 부분을 준비해야 한다.

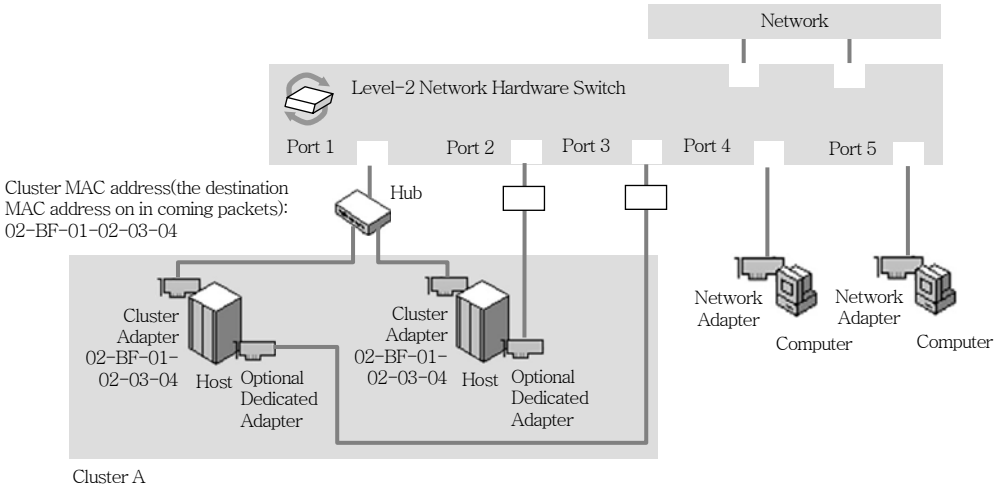
게임서버 또는 다운로드 서비스를 위한 IDC를 선정할 경우의 고려 사항은 게임이용자가 사용하는 네트워크 회선을 가장 많이 보유하고 있는 ISP/IDC 사업자의 서비스를 이용해야 서버와 클라이언트 통신 과정에서의 네트워크 지연을 감소할 수 있게 된다.

게임서비스의 로드밸런스 이용 부분은 login 서버의 앞단에 L4 스위치를 배치하여 다수의 login 서버로 클라이언트의 접속을 부하분산 할 수 있다. 다음으로는 다수의 웹서버와 패치 다운로드 서버 앞단에 L4 스위치를 배치하여 부하분산 할 수 있다. 하지만 L4 스위치의 경우 안정된 로드밸런싱의 기능을 제공하지만 고가의 비용이 소요되는 부분에서 단점이 있다. 외부 클라이언트의 접속을 다수의 서버로 부하분산시키는 기타 방법으로는 리눅스 기반의 LVS[4]와 윈도서버에 탑재된 기능인 NLB[5], 비용이 들지 않는 DNS 라운드로빈 방식이 있다.

L4 스위치의 대안으로 사용할 수 있는 LVS는 클라이언트의 요청 패킷을 대표 가상 IP 주소를 가진 LVS 서버가 리얼서버로 패킷을 전달한다. 이후 요청 패킷을 전달 받은 리얼서버는 클라이언트와 직접적인 통신을 진행한다. 이와 같은 절차의 통신을 다이렉트 라우팅 방식이라고 한다((그림 4) 참조). LVS는 요청 패킷의 분산만을 처리하므로 시스템부하가 거의 없고, 장애에 대비하여 LVS 서버를 이중화 구성이 가능하며, 주기적인 리얼서버의 HTTP 응답 문자스트링 체크를 통해 응답이 없을 경우 부하분산



(그림 4) 리눅스 기반의 로드밸런싱



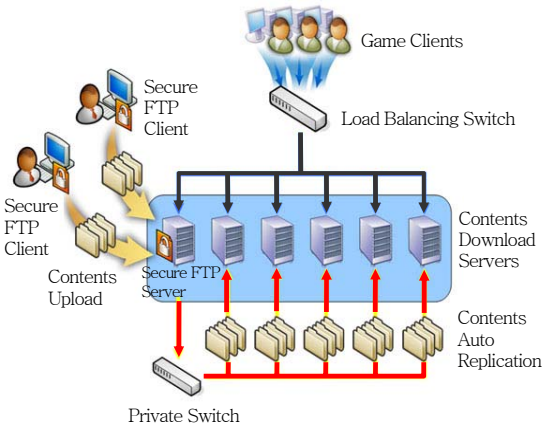
(그림 5) 윈도 기반의 로드밸런싱

리스트에서 장애가 발생한 서버를 제거하는 구성이 가능하다.

리눅스 기반의 LVS는 클라이언트의 요청 패킷을 분배해주는 별도 서버가 있었지만 윈도의 부하분산 기능인 NLB는 네트워크 카드 등록정보의 '네트워크 로드 균형 조정' 기능을 체크하여 가상 IP 주소인 클러스터 IP와 서버별로 고유한 전용 IP 설정, 포트 규칙 설정을 통해 로드밸런싱이 필요한 서비스를 등록할 수 있다((그림 5) 참조). 외부 클라이언트의 요청 패킷이 NLB 클러스터에까지 정상적으로 전달되기 위해서는 윈도 서버에서의 NLB 구성 작업과 별도로 L3 스위치에서 가상 IP 주소와 매칭된 MAC 주소를 정적(static)으로 반드시 등록해줘야 한다. 이런 과정을 마치면 각 클러스터 서버의 네트워크 카드는 상호 통신을 통해 상태 정보를 교환하고 외부의 요청 패킷을 포트규칙에 따라 분배하게 된다. 하지만 윈도 서버의 NLB는 리얼서버의 네트워크 카드 장애 감지가 가능하지만 애플리케이션 서비스의 장애 감지는 되지 않는다. 리얼서버의 윈도 애플리케이션 장애 감지를 사전에 할 수 있다면 저비용으로 로드분산 클러스터 서버(최대 32대)를 구축할 수 있고 클러스터 서버 각각의 네트워크 회선 대역폭을 합한 요청 트래픽 처리가 가능하다.

콘텐츠 파일 복제는 리눅스에서는 rsync[6]를 사용하고 윈도에서는 파일 복제 서비스가 있지만 마스터 서버의 콘텐츠가 변경될 경우 다수의 슬레이브 서버에 자동복제 구성을 하는 것에 다소 문제가 있다. 윈도 서버 환경에서의 파일 복제 툴의 대부분은 네트워크 공유를 통해서 파일 비교를 한 후 복제되는 방식과 마스터, 슬레이브에 파일 복제 에이전트를 설치하여 동기화에 필요한 통신을 하는 형태로 구분된다. 그 외 마스터 서버의 파일 변경 시에 FTP 방식으로 복제되는 구성도 있다.

다운로드 서비스((그림 6) 참조)를 자체적으로 구성할 경우에는 로드밸런서가 외부 클라이언트 요청 패킷의 부하분산을 담당하고, 파일의 업로드는 암호화되지 않은 FTP 방식보다는 SFTP를 지원하는 클



(그림 6) 다운로드 서비스 구성도

라이언트와 서버로 구성하여 외부에서 스니핑이 되지 않도록 구성한다. FTP는 서버 및 클라이언트의 방화벽 환경에 따라 active 또는 passive 모드 방식[7]으로 전환해야 하는데 이런 불편한 문제점을 해결하기 위해서도 단일 포트를 사용하는 SFTP을 사용해야 한다. 파일 업로드를 완료하면 콘텐츠 복제 툴이 사설 네트워크를 통해 다수의 서버로 콘텐츠 자동 동기화가 이루어지면 외부 클라이언트는 HTTP를 통해서 게임 패치 다운로드를 진행한다.

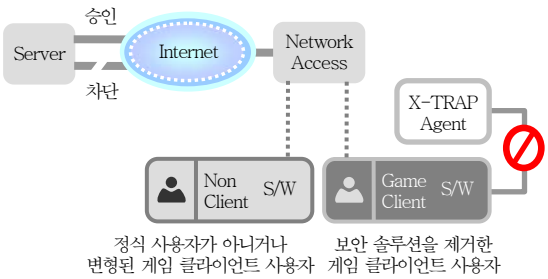
5. 온라인게임 보안솔루션

온라인게임 해킹에는 서버와 클라이언트 단계에서 이루어지는 해킹이 있다. 게임서버의 보안을 위해 네트워크 레벨의 방화벽, 운영체제 레벨의 호스트방화벽, 애플리케이션 보안 툴을 구성해야 한다.

SQL injection[8]과 같은 수법으로 게임 웹사이트의 웹 취약점을 통해서 접근하여 웹과 연동된 게임 데이터베이스까지 접근을 하는 경우가 있다. 여러 형태로 해킹된 게임서버의 프로그램을 취득한 해커는 게임의 실행 파일을 이용하여 프리(free)게임서버로 운영하여 게임업체에 금전적인 피해를 유발하기도 한다.

웹서비스 보안을 높이기 위해 웹소스 취약점 수정 및 웹방화벽 도입이 필요하고 게임 서버의 실행파일이 외부로 유출되더라도 다른 환경의 게임 서버에서 실행되지 않도록 서버 덤프 방지 프로그램을 사용해야 한다.

클라이언트 단계에서 실행하는 게임핵은 온라인게임 시장의 확대와 함께 진화를 거듭하고 있다. 대표적인 해킹툴로는 게임 캐릭터의 이동 속도나 공격속도를 증가시켜 주는 '스피드핵', 캐릭터의 행동 패턴을 설정해 몬스터와의 전투나 전리품 회수 등을 자동적으로 반복하게 해주는 '오토마우스, BOT', 서버와 클라이언트 사이에 거래되는 패킷 데이터를 해킹하여 수치를 높이는 '패킷핵', 여러 게임해킹 기능을 조작하기 쉬운 인터페이스로 제공하는 '전용핵' 등이 있다.



(그림 7) X-TRAP 서버 연동 크랙 방지 기능

다양한 게임핵의 행동 패턴을 연구/예측하고 이를 방어하는 보안 코드가 온라인게임 클라이언트에 탑재된다면 게임내의 부정한 행동을 차단할 수 있다. 온라인게임 보안솔루션으로는 nProtect의 'GameGuard'[9], 안철수연구소의 'HackShield'[10], 와이즈로직의 'X-TRAP'((그림 7) 참조)[11] 등이 있다.

6. 게임방화벽과 웹방화벽

방화벽은 기본적으로 관리자와 게임 클라이언트와의 통신에 필요한 서비스 포트를 제외한 모든 접근을 통제하고 분산서비스공격(DDOS)과 외부에서의 불법 스캐닝을 차단한다. 게임서버와 클라이언트는 패킷이란 매개체를 이용하여 상호간에 통신을 하는데 클라이언트에서 서버로 전송하는 패킷은 사용자에 의해 조작될 가능성이 있기 때문에 패킷의 암호화가 필요하다. 게임 통신은 작은 사이즈의 수많은 패킷이 전송되기 때문에 방화벽의 성능이 떨어질 경우에는 게임의 랙(lack) 현상을 유발하게 된다.

게임 개발사와 IDC의 네트워크 구간의 강력한 보안 통신을 위해서는 VPN 기능이 포함된 방화벽간 IPSec VPN 구성으로 암호화 터널링 통신을 제공할 수 있다. 이동 사용자를 위해서는 VPN 소프트웨어 프로그램의 인증을 거친 후 IDC 게임서비스 인프라에 접근하게끔 설정해야 한다.

게임 웹사이트의 경우 게임유저의 회원 가입과 게임에 대한 의견을 교환하는 장소로 사용된다. 웹서버는 서비스를 위해 80포트(HTTP)와 443포트(HTTPS)를 모든 IP에 대해 접근을 허용하고 있다. 웹 프로그램의 설정 오류나 개발 오류로 인한 웹 애

플리케이션 자체의 취약점을 이용한 홈페이지와 웹 서버 해킹이 시도될 수 있다. 방화벽, IDS, IPS 장비들은 네트워크 하위 계층을 주로 막는 데 사용되고 웹 해킹을 막는 데는 한계가 있다. 웹방화벽은 외부의 사용자 요청을 검사하고 적응학습 과정을 통해서 검사를 통과한 요청만 웹서버에 전달하고 응답을 전달하는 방식을 취하고 있다. 대부분의 웹방화벽은 웹사이트에 최적화하는 기간이 요구되며 네트워크 장비 타입과 호스트 운영체제에 설치되는 소프트웨어 종류가 있는데 무료 웹방화벽 소프트웨어로는 WebKnight가 있다.

7. 시스템/네트워크 모니터링

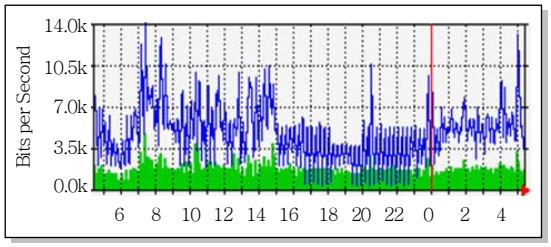
모니터링 솔루션은 점차 복잡하고 다양해지는 서비스 인프라에 대한 실시간 상태 감시 및 관리, 성능 진단 및 관리, 보안 및 장애 관리 서비스 등을 제공한다. 관리데이터는 SNMP 등과 같이 표준화된 프로토콜을 통해 수집하거나 지능형 에이전트와 같은 데이터 수집기를 통해 공급 받을 수 있다.

게임서비스를 운영하는 환경에서 최소한 네트워크 스위치 장비의 트래픽 사용량 정도는 확인해야 한다. 트래픽의 추이를 분석하고 게임 동시접속자 데이터를 비교하여 오픈 또는 상용화 서비스 시에 필요한 네트워크 대역폭을 산정할 수 있게 된다.

MRTG[12]는 장비의 CPU, 메모리, 스위치, 다양한 서버와 애플리케이션의 상태를 모니터링하는 툴이다. 우선 특정 네트워크 장비의 트래픽을 모니터링((그림 8) 참조)하고 싶을 경우 대상 장비에 SNMP 요청에 응답하기 위한 접근 허용 설정과 커뮤니티 이름을 설정해야 한다. MRTG가 구동될 서버에는 모니터링 장비에 정보를 요청하고, 그 값을

● 용 어 해 설 ●

SNMP(Simple Network Management Protocol): 단순 네트워크 관리 프로토콜(SNMP)은 TCP/IP 기반의 유형이 다른 네트워크에 맞는 효과적인 네트워크 관리 플랫폼을 설계하기 위해 1988년에 정의되어 1990년에 IAB에 의해 인터넷 표준으로 승인됨



	Max	Average	Current
In	4664.0b/s (0.0%)	1488.0b/s (0.0%)	1584.0b/s (0.0%)
Out	13.9kb/s (0.0%)	4768.0b/s (0.0%)	3552.0b/s (0.0%)

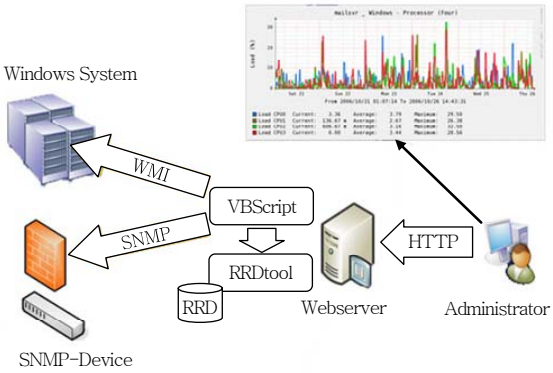
(그림 8) MRTG 네트워크 그래프

로그로 남긴 후, 그래프를 만드는 MRTG 프로그램을 설치해야 한다.

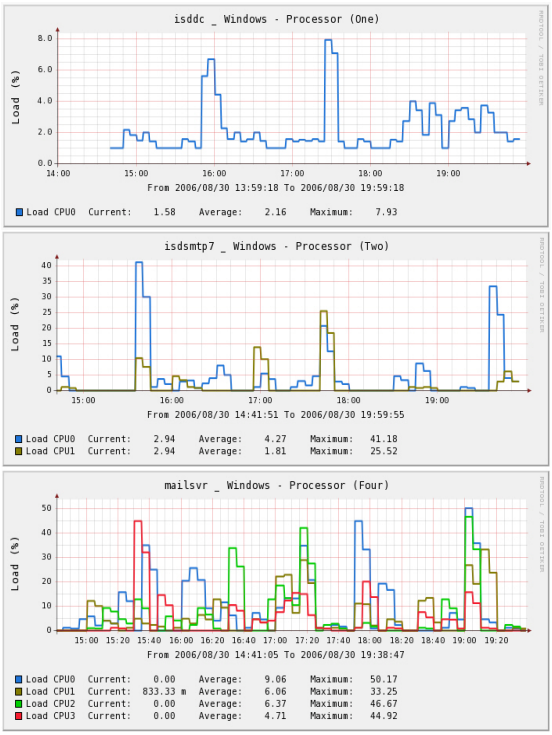
MRTG의 단점은 상태를 모니터링 하는 개체가 in/out 2개의 인자 밖에 없다는 점이다. 만약 4개 이상의 CPU를 보유한 서버의 CPU 로드를 모니터링 하기 위해서는 2개 이상의 그래프를 생성해야 하는 불편함이 있다. 또 다른 단점은 일간/주간/월간/연간의 그래프는 제공하지만 특정 기간 동안의 사용량 데이터를 분석하는 데에는 어려움이 있다.

RRDtool[13]은 MRTG의 그래프 기능과 로깅 기능을 강화하여 새로 구현된 도구이다. RRD는 네트워크 대역폭, 서버의 평균 부하 등과 같은 시간대별 데이터를 저장하고 표시하기 위한 시스템이다. RRD는 매우 간결한 방법으로 데이터를 저장하므로 시간 경과에 따라 파일 크기가 크게 늘어나지 않는다. RRD는 항상 일정한 데이터 밀도를 강제로 유지하기 위해 데이터를 처리함으로써, 유용한 그래프를 제공한다. 이를 위해서는 셸이나 펄, VBScript 또는 SNMP를 지원하는 네트워크 장비(스위치, 방화벽)에 주기적으로 요청 쿼리를 던지고 편리한 인터페이스를 제공하는 프론트엔드를 이용할 수 있다.

RRDtool이 설치된 서버에서 모니터링 하기를 원하는 개체의 성능을 측정하기 위해 VBScript를 수행한 결과값을 폴러가 다중 출력 필드에 파싱을 진행한다((그림 9) 참조). 최초에 RRDtool create 작업을 수행하여 RRD 데이터베이스 파일 안에 테이블을 생성한다. 그런 후 RRDtool 업데이트 예약 작업을 통해 주기적으로 사용량의 데이터가 저장된다.



(그림 9) RRDtool을 이용한 모니터링

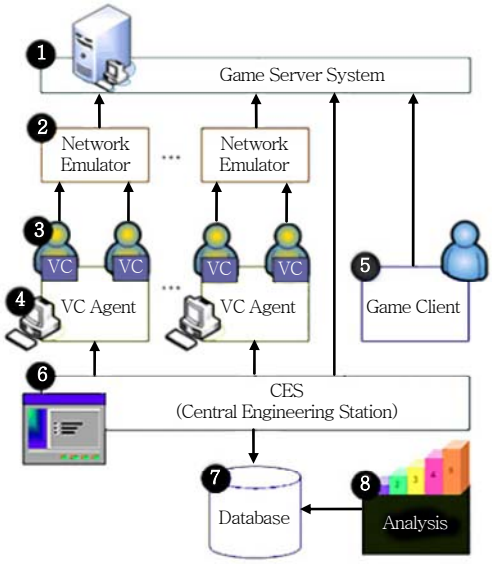


(그림 10) CPU LoadPercentage

(그림 10)은 원격 윈도 서버의 CPU 부하 퍼센트 값을 얻기 위해 WMI 클래스 개체 정보에 쿼리하기 위한 스크립트와 RRDtool을 사용하여 5분 주기로 결과값을 그래프의 형태로 표현한 것이다.

8. 게임시뮬레이터시스템

게임서비스의 오픈베타에 앞서 반드시 필요한 작업은 기존의 클로즈베타 테스트 결과를 기반으로 예상동접자에 맞는 게임시스템을 사전에 준비하는 일이다. 하지만 하나의 서버군이 최대로 수용할 수 있는 한계치를 클로즈베타 단계에서 경험하지 못하고 오픈베타를 맞이하게 된다면 게임동접자가 급격하게 증가될 경우 게임접속 중지 상황에 놓이게 된다.



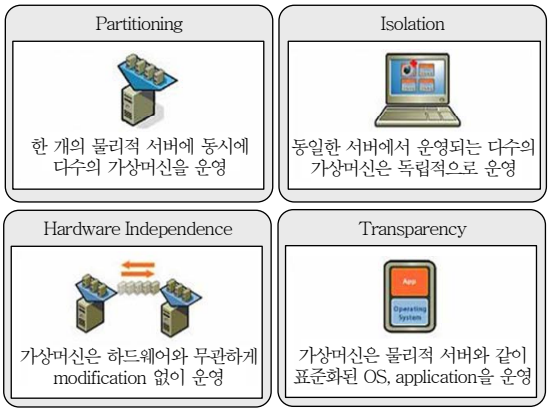
(그림 11) 게임시뮬레이터시스템 구성도

게임시뮬레이터시스템이란 온라인게임을 플레이하는 실제 수백, 수천의 사용자를 시뮬레이션 해주기 위해서 다수의 호스트들을 통합하여 제어하고 모니터링 해주는 소프트웨어 시스템이다.

대규모 사용자를 생성하여 시뮬레이션을 진행할 수 있는 확장 구성((그림 11) 참조)을 가지며 시뮬레이션 결과를 분석하기 위한 analysis, 반복적인 실험을 위한 스크립트, 게임서버군의 기본적인 성능지수(CPU, network, memory..)를 실시간으로 모니터링 할 수 있는 기능을 제공한다. 이런 기능들로 인해서 게임개발사에서는 동접자 증가의 추이를 분석하고 그에 맞는 시스템 증설계획을 세울 수 있게 된다.

9. 서버 가상화

게임 서버 중에서 패치, 인증, 로비 서버의 경우 초기 접속만을 처리하므로 서버에 부하가 크지 않



(그림 12) 가상화의 기능

다. 그런 이유 때문에 게임 서버군을 증설하여도 패치, 인증, 로비 서버는 여러 서버군에서 공용으로 사용이 가능하다. 게임 서버의 개발과정에서 향후 확장성을 고려해야 향후 확장에 대한 이슈가 발생될 때 유연하게 대처가 가능해진다.

가상화(virtualization) 기술은 시스템의 유휴자원을 좀 더 효율적으로 활용할 수 있는 방안을 제시한다. 가상화의 기능((그림 12) 참조)은 물리적인 하드웨어 박스를 가상적인 몇 개의 시스템으로 논리적 파티셔닝을 제공한다. 이를 통해 기존 시스템의 유휴자원을 가상 시스템 전용 자원으로 재활용하며 물리적인 하나의 하드웨어 장비의 시스템 자원 활용률을 향상시킨다.

가상화 기술을 적용시킨다면 패치, 인증, 로비 서버의 운영체제를 한 대의 물리적 하드웨어에서 운영이 가능할 것이다. Microsoft의 윈도 서버 2008 버전에서는 가상 머신 매니저로 '윈도 하이퍼바이저'라는 소프트웨어를 제공할 예정이다.

개발 테스트단계에 사용했던 가상화 서비스는 고사양의 서버에 탑재될 경우 게임서비스뿐만 아니라 서버를 사용하는 모든 분야에 적용되리라 생각된다.

Ⅲ. 결론

온라인게임을 안정적으로 서비스하기 위해서는 가장 먼저 게임 서버 애플리케이션의 안정성을 확보하는 일이다. 서버 다운을 일으키는 버그를 쉽게 찾아낼 수 있는 디버깅 피처를 반드시 갖추고 있어야 한다. 예외를 사용해서 서버에 오류가 발생했을 때의 상황을 개발자가 알 수 있는 형태로 남겨두게 되면 대부분의 버그는 쉽게 잡을 수 있다. 본문에서는 오픈베타 수준의 온라인게임 서비스 안정화를 위한 기술적 고려사항으로 다음의 항목을 다루었다.

- 시스템/네트워크 구성
- 패치시스템 및 CDN 서비스
- 콘텐츠 파일 복제 및 로드밸런싱
- 온라인게임 보안솔루션
- 게임방화벽과 웹방화벽
- 시스템/네트워크 모니터링
- 가상유저테스트를 위한 게임시뮬레이터시스템
- 서버 가상화

온라인게임이 지속적으로 업데이트 해야 하듯이 안정된 게임서비스를 위해서는 위에서 거론된 기술적 고려사항의 준비와 추가적으로 상용화에 대비한 운영시스템(GM 운영툴, 빌링시스템, CRM 시스템)이 통합적으로 갖추어져야 한다.

● 용 어 해 설 ●

WMI(Windows Management Instrumentation): 네트워크에서 분산되어 있는 다양한 시스템, 장치에 대한 관리 표준을 정하기 위해 웹기반의 엔터프라이즈 관리 표준을 제정했고 이것을 마이크로소프트에서 구현한 것이 WMI이다.

약 어 정 리

AD	Active Directory
ADSL	Asymmetric Digital Subscriber Line
CDN	Content Delivery Network
CPU	Central Processing Unit
CRC	Cyclic Redundancy Checking
CRM	Customer Relationship Management
DDOS	Distributed Denial-Of-Service attack
DNS	Domain Name System

FTP	File Transfer Protocol
GM	Game Master
GSLB	Global Server Load Balancing
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
IDC	Internet Data Center
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security protocol
ISP	Internet Service Provider
LDAP	Lightweight Directory Access Protocol
LVS	Linux Virtual Server
MAC	Media Access Control
MRTG	Multi Router Traffic Grapher
NLB	Network Load Balancing
P2P	Peer To Peer
RRD	Round Robin Database
SFTP	Secure File Transfer Protocol
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSH	Secure Shell
VPN	Virtual Private Network
WBEM	Web-Based Enterprise Management
WMI	Windows Management Instrumentation
WSUS	Windows Server Update Services

참 고 문 헌

[1] Private DNS Split-Brain DNS Server Configuration for ISPs Published June 2003, Microsoft Corporation, http://download.microsoft.com/download/6/6/0/6600f6f7-e3b7-466a-b7da-e088b94af122/split_brain_DNS.doc

[2] WSUS Windows Server Update Services(WSUS), http://technet.microsoft.com/enus/wsus/default.aspx

[3] CDN Service http://www.cdnetworks.co.kr/kor/tech/cdn_system.php

[4] LVS, http://www.linuxvirtualserver.org/

[5] NLB Using NLB with ISA Server Part 2: Layer 2 Fun with Unicast and Multicast Modes By Thomas Shinder, http://www.isaserver.org/articles/basicnlbpart2.html

[6] rsync, http://rsync.samba.org/

[7] FTP Active/Passive Mode How the FTP Protocol Challenges Firewall Security, By Stefaan Pouseele, http://www.isaserver.org/articles/How_the_FTP_protocol_Challenges_Firewall_Security.html

[8] SQL Injection Are your web applications vulnerable? By kevin Spett, http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf

[9] nProtect GameGuard, http://www.inca.co.kr

[10] Ahnlab HackShield, http://kr.ahnlab.com

[11] WiselogicX-Trap, http://www.wiselogic.co.kr

[12] MRTG, http://oss.oetiker.ch/mrtg/

[13] RRDtool, http://oss.oetiker.ch/rrdtool/