

온라인 게임 클라이언트를 위한 안전한 패치 방안

이현진*, 송병근, 서재현
인제대학교 전산학과
e-mail: cyberpd@chollian.net

Patching Solution For Online Game Client

Hyun-Jin Lee*, Byung-Kuen Song, Jai-Hyun Seu
Computer Science, Inje University

요 약

기존의 온라인 게임 클라이언트의 패치 방법은 비효율적이고, 클라이언트 데이터의 무결성을 보장하지 못하였다. 본 논문에서는 온라인 게임 클라이언트를 위한 안전한 패치 방안에 대하여 알아보고, 구현을 위한 기술적 방안으로써 MD5 메시지 축약 알고리즘의 활용에 대해 기술한다.

1. 서론

최근의 온라인 게임들은 단일 유저를 위한 PC 패키지 게임처럼 화려한 영상과, 사실성 있는 소리들을 제공하며, 대부분이 3D 화면을 제공한다. 따라서 고음질의 효과음 및 배경음악, 세부적인 모델링 데이터 및 고화질 텍스처가 사용되며, 이러한 데이터 파일들은 큰 용량을 가진다. 또한, 방대한 지역의 지형 데이터가 이용되고, 수많은 아이템 및 몬스터가 등장하므로 온라인 게임 클라이언트가 가지고 있어야 하는 데이터의 개수 및 용량은 예전과 비교할 수 없을 만큼 늘어났다.

PC 패키지 게임은 대부분이 완성되어 출시가 된다. 그러나 온라인 게임은 대부분이 베타 테스트 단계부터 게임 유저에게 공개된다. 게임 유저는 지속적으로 개발되는 온라인 게임을 플레이하게 되므로, 데이터 파일의 변경이 자주 발생하게 된다.

또한, 온라인 게임은 게이머들의 피드백에 따른 변화가 많다. 개발자들은 게이머들의 요구사항들을 수용하기 위해 그래픽, 데이터 수치, 아이템, 지형, 기능 등을 자주 바꾸거나 추가한다.

그러므로 대부분의 온라인 게임 클라이언트는 많은 수의 변경 또는 추가되는 데이터 파일들을 효율

적으로 관리하기 위해 자동 패치 시스템을 가지고 있다.

그러나 현재 대부분의 온라인 게임들이 가지고 있는 패치 시스템은 비효율적이고, 클라이언트 데이터가 손상되었을 경우에 이를 해결 할 수 있는 시스템도 많지 않다.

기존의 온라인 게임들이 가지고 있는 대부분의 패치 시스템은 버전에 따른 데이터 변경, 추가 목록을 가지고 클라이언트 패치 프로그램에서 변경 또는 추가된 파일들을 다운로드 하게 된다.

또한, 만일 게임 유저의 PC가 논리적 또는 물리적 원인으로 인하여 하드 디스크가 손상되고, 게임 클라이언트의 데이터가 변경 또는 유실될 경우에 클라이언트 프로그램은 정상적으로 수행되지 못하고 예기치 못한 오류가 발생하기도 한다.

본 논문은 먼저 기존의 온라인 게임 클라이언트의 패치 방법에 대해 알아보고, 발생하는 문제점을 분석해 본다. 그리고 온라인 게임 클라이언트의 데이터를 패치하기 위한 안전한 방안에 대하여 알아보고 문제점을 해결하기 위한 기술적 방법으로 MD5 메시지 축약 알고리즘[1]의 이용방법에 대해 살펴본다.

2. 기존 방법

기존의 온라인 게임 클라이언트의 패치 방법은 이전 버전과 새로운 버전 사이의 변경되거나 추가된 파일을 다운로드하는 것이다.

패치 서버에는 클라이언트의 버전에 따라서 변경되거나 추가된 파일을 다운받을 수 있도록 표 1, 2와 같은 패치 파일 목록을 가지고 있다.

패치 파일	파일 버전
./v1.1/map001.dat	1.1
./v1.1/char001.dat	1.1

표 1. 1.0 버전에서 1.1 버전으로 패치할 파일 목록

패치 파일	파일 버전
./v1.2/map001.dat	1.2
./v1.2/char002.dat	1.2

표 2. 1.1 버전에서 1.2 버전으로 패치할 파일 목록

1.0 버전에서 1.1 버전으로 패치를 할 경우에 패치 서버는 접속한 클라이언트에게 표 1의 패치 파일 목록을 전송하고, 클라이언트는 목록에 있는 파일을 다운로드한다. 1.1 버전에서 1.2 버전으로 패치를 할 경우에는 클라이언트가 표 2 목록의 파일을 다운로드 한다. 1.0 버전에서 1.2 버전으로 패치를 하기 위해서 패치 서버는 표 1과 표 2의 목록을 합치고, 중복된 파일은 낮은 버전을 삭제하여 표 3의 목록을 생성하여 클라이언트에게 전송한다.

패치 파일	파일 버전
./v1.2/map001.dat	1.2
./v1.1/char001.dat	1.1
./v1.2/char002.dat	1.2

표 3. 1.0 버전에서 1.2 버전으로 패치할 파일 목록

위와 같은 방법으로 클라이언트는 패치 목록을 받아서 필요한 파일을 다운로드하게 된다. 이것을 수동 코드로 나타내면 소스 1과 같다.

```
// 패치 서버에 등록된 버전
Version ServerVer = GetServerVersion();
// 클라이언트의 버전
Version ClientVer = GetClientVersion();
```

```
FileList patchlist; // 패치 파일 목록
Version CurrentVer;
Version NextVer;

// 현재 서버 버전이 더 높을 경우 목록 갱신
while( ServerVer > CurrenttVer )
{
    // 패치될 다음 버전을 얻는다.
    NextVer = GetNextVersion(ClientVer);
    // 패치될 다음 버전의 파일 목록을 얻는다.
    FileList = GetPatchFileList(NextVer);
    // 현재 버전을 다음 버전으로 변경한다.
    CurrenttVer = NextVer;
}

// 패치할 파일 목록을 정렬한다.
// 패치할 같은 파일 이름이 있을 경우,
// 낮은 버전의 목록을 제거한다.
FileList->Sort();
```

소스 1. patch server pseudo code

이러한 일반적인 패치 방법을 순서대로 정리하면 다음과 같다.

< 일반적인 패치 단계 >

- ① 패치 서버 접속, 버전 정보 전송
- ② 버전에 따른 전송할 파일 목록 전송
- ③ 목록에 따른 변경파일 요청
- ④ 변경 파일 전송

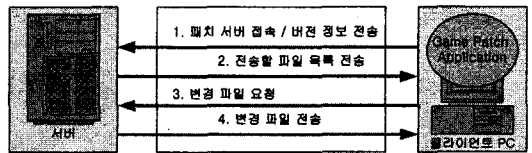


그림 1. 일반적인 패치 단계

이러한 패치 방법은 크게 두 가지 문제점이 발생한다. 첫째, 클라이언트의 데이터가 변경 또는 삭제 되었을 때, 이를 확인할 수가 없다. 만일 게이머의 PC에 문제가 생겨 클라이언트 데이터가 손실되었을 경우, 클라이언트 프로그램이 실행할 때 오류를 발생시킬 것이다. 이는 클라이언트 데이터의 무결성을 보장할 수 있는 방법이 없기 때문이다. 둘째, 패치가 될 때 마다, 패치 서버는 모든 버전간의 변경 파일과 목록을 유지해야 한다. 이는 패치 서버에 불필요한 데이터를 보관하도록 하며, 패치 데이터 관리에 비효율적이다.

3. 향상된 방법

위에서 기술한 문제점들을 해결하고, 온라인 게임 클라이언트의 정상적인 수행으로 신뢰성 있는 게임 진행 되도록 하기 위한 방법으로, MD5 메시지 축약 알고리즘을 이용한 클라이언트 데이터 검사가 있다.

MD5는 길이에 상관없는 하나의 입력 데이터로부터 128비트의 축약된 메시지를 만듦으로써 데이터 무결성을 검증하는데 사용되는 알고리즘이다. MD5는 IETF의 RFC1321에 명세가 되어있다. MD5 명세에 따르면, MD5 알고리즘에 입력된 임의의 두 개의 메시지가 동일한 메시지 축약의 결과를 가질 확률은 264분의 1이다. 또는 어떤 메시지 축약을 통해 잘못된 메시지가 만들어질 확률은 2128분의 1이다. 그러므로 패치 서버와 클라이언트의 데이터 파일의 MD5 메시지 축약 값을 비교하면, 파일의 작은 변화라도 확인할 수가 있다.

MD5를 이용하여 클라이언트의 데이터를 패치하는 방법은 다음과 같다.

패치 서버는 최신 클라이언트 데이터 파일들과 각 데이터 파일에 대한 MD5 값을 가지는 패치 목록을 가지고 있다. 클라이언트는 패치 서버에 접속하여 패치 목록 파일을 가져온다. 클라이언트는 자신이 가지고 있는 파일들의 MD5 체크섬 값을 검사하고, 패치 목록과 비교한다. 만약 체크섬 값이 틀릴 경우에 서버로부터 해당 파일을 다운로드 한다. 정리하면 다음과 같다.

< MD5 체크섬 방식의 패치 단계 >

- ① 패치 서버 접속
- ② MD5 체크섬 값을 포함하는 패치 파일 목록을 다운로드
- ③ 각각의 로컬 파일의 MD5 체크섬 값 확인
- ④ 변경 파일 요청
- ⑤ 변경 파일 다운로드

그림 2는 이것을 그림으로 나타낸 것이다.

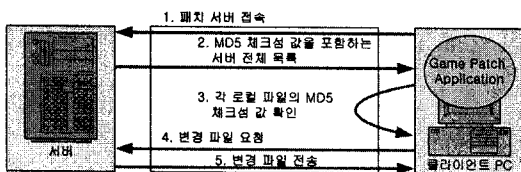


그림 2. MD5 체크섬 방식의 패치 단계

MD5 체크섬 확인 방식을 사용하면 클라이언트에서 최신의 파일을 검사하여 다운로드하게 되므로, 패치 서버는 각 버전 간의 변경 파일들과 목록을 유지하고 있을 필요가 없이 최신의 파일 목록만 가지고 있으면 된다. 또한, 클라이언트의 데이터 파일이 사용자 PC에서 손상되었을 때도 MD5 체크섬 값을 비교하여 정상적인 파일을 다운로드하게 되므로 클라이언트 데이터 파일의 무결성을 보장할 수가 있다.

그림 3은 패치 클라이언트의 수행 모습이다.

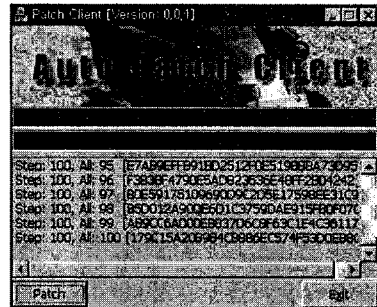


그림 3. 패치 클라이언트의 수행 모습

MD5를 이용하여 게임이 실행될 때 마다 모든 클라이언트 데이터 파일의 체크섬을 확인하는 것은 시간이 오래 걸린다. 이러한 문제는 클라이언트가 패치 서버에 접속하여 현재 게임 클라이언트가 최신 버전일 경우 자동으로 패치 목록을 다운로드 하지 않고, MD5 체크섬 확인은 옵션으로써 사용자가 선택할 수 있도록 하여 해결할 수 있다.

클라이언트 데이터의 무결성을 보장하기 위한 좀더 나은 방법은 클라이언트 프로그램과 패치 프로그램을 분리하는 것이다. 게임을 실행하기 위해서는 패치 프로그램을 실행하고, 패치 프로그램이 현재 버전 및 클라이언트 데이터를 확인하고, 클라이언트 프로그램을 실행하도록 하는 것이다. 클라이언트를 직접 실행하면 프로그램이 종료되도록 하여, 반드시 패치 서버에 접속하여 최신 버전인지를 확인하도록 하여 클라이언트의 무결성을 보장할 수가 있다.

4. 효율적 패치를 위한 추가 방안

게임 클라이언트는 실행될 때 마다 패치 서버에 접속되고, 동시에 수십 수백 개의 클라이언트가 패치 파일들에 대한 다운로드를 요청하게 된다. 이 경우 패치 서버를 늘리지 않고, 부하가 증가할 경우에

대비하기 좋은 방법은 프락시 서버를 이용하는 것이다. 패치 프로그램은 주로 FTP나 HTTP를 사용하므로 프락시 서버를 통하여 패치 서버의 처리 능력을 향상시킬 수가 있다.

패치 서버의 성능을 높이기 위한 것으로 패치 데이터를 압축하여 전송 용량을 줄이 방법도 있다. 이 경우 패치 서버의 패치 파일 목록에 압축 파일의 이름이 추가된다. 클라이언트는 패치 목록에서 MD5 값을 비교하고 압축된 데이터 파일을 다운로드 받는다. 공개용 압축 라이브러리인 ZLib[2]를 이용하여 이를 구현할 수 있다.

게임 데이터가 계속 변경되다 보면, 게이머의 PC에 더 이상 사용되지 않는 파일이 생길 수 있다. 이러한 파일들을 하드 드라이브의 공간을 낭비하게 되는데, 클라이언트의 데이터 파일을 패치 목록과 비교할 때 패치 목록에 없는 클라이언트 데이터 파일을 삭제함으로써 불필요한 파일들을 하드 드라이브에서 자동으로 삭제할 수 있다.

5. 결론

본 논문은 기존 온라인 게임 클라이언트의 패치 방법이 가지는 문제점들을 해결할 수 있는 방안으로써 MD5 메시지 축약 알고리즘을 이용한 파일 비교 방법을 기술하였다. 그러나 모든 게임 클라이언트의 데이터 파일을 MD5로 검사하는 것은 많은 시간이 소비된다. 이것은 클라이언트가 최신 버전일 경우 MD5 검사를 옵션으로써 게이머가 선택할 수 있게 하여 해결할 수가 있다. 파일 비교에 드는 시간적 비용을 줄이기 위해서 MD5외에 SHA-1[3], CRC64 등의 다른 해시 함수들을 이용할 수도 있을 것이다.

참고문헌

- [1] "The MD5 Message-Digest Algorithm", RFC1321, <http://www.ietf.org/rfc/rfc1321.txt>
- [2] "zlib compression library", Jean-loup Gailly and Mark Adler, <http://www.gzip.org/zlib>
- [3] "US Secure Hash Algorithm 1(SHA1)", RFC3174, <http://www.ietf.org/rfc/rfc3174.txt>
- [4] "공개용 압축 라이브러리를 이용한 파일 패키지 관리", 이현진, 송병근, 한국게임학회, 2003