



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0153592  
(43) 공개일자 2022년11월18일

- (51) 국제특허분류(Int. Cl.)  
A63F 13/352 (2014.01) A63F 13/335 (2014.01)  
A63F 13/77 (2014.01)
- (52) CPC특허분류  
A63F 13/352 (2015.01)  
A63F 13/335 (2015.01)
- (21) 출원번호 10-2022-7031766
- (22) 출원일자(국제) 2021년03월17일  
심사청구일자 없음
- (85) 번역문제출일자 2022년09월14일
- (86) 국제출원번호 PCT/US2021/022815
- (87) 국제공개번호 WO 2021/188711  
국제공개일자 2021년09월23일
- (30) 우선권주장  
16/821,716 2020년03월17일 미국(US)

- (71) 출원인  
벨브 코퍼레이션  
미국 워싱턴 (우편번호 98004) 벨뷰 엔이 포츠 스트리트 10400 스위트 1400
- (72) 발명자  
그리페, 피에르-롭 미구엘  
미국 98004 워싱턴 벨뷰 노스이스트 4번 스트리트 10400
- (74) 대리인  
양영준

전체 청구항 수 : 총 20 항

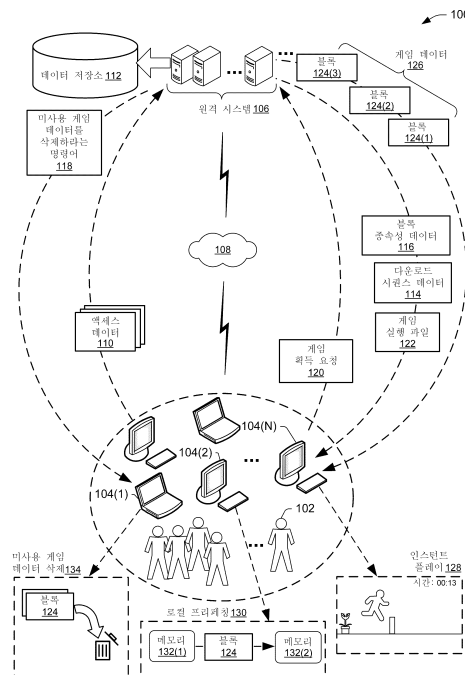
(54) 발명의 명칭 비디오 게임의 인스턴트 플레이를 위한, 그리고 클라이언트 측 게임 데이터의 폐기와 프리페치를 위한 파일 시스템 판독 작동 추적

(57) 요약

비디오 게임(들)의 게임 실행 파일을 런닝(running)하는 클라이언트 머신은 게임 세션 동안 게임 실행 파일에 의해 수행된 판독 작동을 추적하도록 구성된 파일 시스템 프록시 컴포넌트를 활용하여, 추적된 판독 작동에 기초하여 액세스 데이터를 생성하고, 액세스 데이터를 원격 시스템에 리포트할 수 있다. 이 원격 측정(telemetry) 접근

(뒷면에 계속)

대표도 - 도1



방식은, 원격 시스템이 다수의 클라이언트 머신에 의해서 보고된 액세스 데이터를 수집하고, 클라이언트 시스템 컨피규레이션에 따라 액세스 데이터를 카탈로그화하고, 액세스 데이터를 분석하는 것을 허용하여, 다양한 게임 관련 기능을 구현하기 위해서 클라이언트 머신에 의해서 사용 가능한 데이터를 생성할 수 있으며, 이 기능은, 한정됨 없이, 비디오 게임의 "인스턴트 플레이", 로컬 메모리 리소스를 확보하기 위해 게임 데이터의 미사용 (unused) 블록의 폐기, 및/또는 게임 플레이 동안 레이턴시를 감소시키기 위한 게임 데이터의 로컬 프리페치 (local prefetching)를 포함한다.

(52) CPC특허분류

*A63F 13/77* (2015.01)

*A63F 2300/534* (2013.01)

*A63F 2300/535* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

클라이언트 머신으로서,

하나 이상의 프로세서; 및

상기 하나 이상의 프로세서에 의해서 실행될 때, 상기 하나 이상의 프로세서로 하여금,

상기 클라이언트 머신 상에서 비디오 게임을 플레이하기 위해 상기 비디오 게임의 게임 실행 파일을 실행하고,

상기 클라이언트 머신의 파일 시스템에 대해 상기 게임 실행 파일에 의해 수행되는 제1 관독 작동 및 제2 관독 작동을 결정하고 - 상기 제1 관독 작동은 상기 비디오 게임을 위한 게임 데이터의 제1 블록을 관독할 것을 요청하고, 상기 제2 관독 작동은 상기 게임 데이터의 제2 블록을 관독할 것을 요청함 -,

상기 관독 작동들에 적어도 부분적으로 기초하여 액세스 데이터를 생성하고 - 상기 액세스 데이터는,

상기 게임 데이터의 상기 제1 블록의 제1 식별자;

상기 게임 실행 파일의 실행 동안, 상기 게임 데이터의 상기 제1 블록이 상기 제1 관독 작동에 적어도 부분적으로 기초하여 액세스된 제1 시간;

상기 게임 데이터의 상기 제2 블록의 제2 식별자; 및

상기 게임 실행 파일의 상기 실행 동안, 상기 게임 데이터의 상기 제2 블록이 상기 제2 관독 작동에 적어도 부분적으로 기초하여 액세스된 제2 시간을 상술함 -, 및

상기 액세스 데이터, 상기 비디오 게임의 식별자, 및 상기 클라이언트 머신의 컨피규레이션(configuration)을 원격 시스템에 송신하게 하는 컴퓨터 실행 가능 명령어를 저장하는 비일시적 컴퓨터 관독 가능 매체를 포함하는 클라이언트 머신.

#### 청구항 2

제1항에 있어서, 상기 클라이언트 머신의 상기 컨피규레이션은 상기 클라이언트 머신과 연관된 하드웨어, 소프트웨어, 또는 펌웨어의 유형, 버전 또는 특성을 상술하는, 클라이언트 머신.

#### 청구항 3

제1항에 있어서, 상기 클라이언트 머신은 비휘발성 메모리를 더 포함하되, 상기 비디오 게임은 제1 비디오 게임이고, 상기 게임 실행 파일은 상기 제1 비디오 게임의 제1 게임 실행 파일이고, 상기 게임 데이터는 상기 제1 비디오 게임을 위한 제1 게임 데이터이고, 상기 컴퓨터 실행 가능 명령어는, 상기 하나 이상의 프로세서에 의해서 실행될 때, 상기 하나 이상의 프로세서로 하여금 추가로,

제2 비디오 게임을 획득하기 위한 요청을 상기 원격 시스템에 송신하고 - 상기 요청은 상기 클라이언트 머신의 상기 컨피규레이션을 포함함 -,

상기 원격 시스템으로부터,

상기 제2 비디오 게임의 제2 게임 실행 파일; 및

상기 클라이언트 머신의 상기 컨피규레이션과 연관된 다운로드 시퀀스(sequence) 데이터 - 상기 다운로드 시퀀스 데이터는 상기 제2 비디오 게임을 위한 제2 게임 데이터의 블록들의 시퀀스를 상술함 -;를 수신하고,

상기 다운로드 시퀀스 데이터에 상술된 상기 시퀀스에 따라 상기 제2 게임 데이터의 상기 블록들을 상기 비휘발성 메모리에 다운로드하기 시작하고,

상기 클라이언트 머신 상에서 상기 제2 비디오 게임을 실행하기 위해 상기 제2 게임 실행 파일을 실행하게 하는, 클라이언트 머신.

**청구항 4**

제3항에 있어서, 상기 컴퓨터 실행 가능 명령어는, 상기 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서로 하여금 추가로,

상기 파일 시스템에 대해 상기 제2 게임 실행 파일에 의해 수행되는 제3 판독 작동을 수신하고 - 상기 제3 판독 작동은 상기 제2 게임 데이터의 블록을 판독할 것을 요청함 -,

상기 제2 게임 데이터의 상기 블록이 상기 비휘발성 메모리에 다운로드되었다고 결정하고,

상기 파일 시스템을 사용하여 상기 제2 게임 데이터의 상기 블록을 판독하게 하는, 클라이언트 머신.

**청구항 5**

제3항에 있어서, 상기 컴퓨터 실행 가능 명령어는, 상기 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서로 하여금 추가로,

상기 파일 시스템에 대해 상기 제2 게임 실행 파일에 의해 수행되는 제3 판독 작동을 수신하고 - 상기 제3 판독 작동은 상기 제2 게임 데이터의 블록을 판독할 것을 요청함 -,

상기 제2 게임 데이터의 상기 블록이 상기 비휘발성 메모리에 다운로드되는 것이 완료되지 않았다고 결정하고,

상기 제2 게임 데이터의 상기 블록에 대한 제2 요청을 상기 원격 시스템에 송신하고,

상기 원격 시스템으로부터 상기 제2 게임 데이터의 상기 블록을 수신하고,

상기 파일 시스템을 사용하여 상기 제2 게임 데이터의 상기 블록을 판독하게 하는, 클라이언트 머신.

**청구항 6**

제1항에 있어서, 상기 클라이언트 머신은 상기 게임 데이터를 저장하는 비휘발성 메모리를 더 포함하되, 상기 게임 데이터의 복수의 제1 블록은 적어도 상기 게임 데이터의 상기 제1 블록 및 상기 게임 데이터의 상기 제2 블록을 포함하고, 상기 컴퓨터 실행 가능 명령어는, 상기 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서로 하여금 추가로,

상기 비휘발성 메모리로부터 상기 게임 데이터의 하나 이상의 제2 블록을 삭제하라는 명령어를 상기 원격 시스템으로부터 수신하고 - 상기 게임 데이터의 상기 하나 이상의 제2 블록은 임계 시간 기간 또는 임계 수의 게임 세션 동안 상기 클라이언트 머신 상에서 상기 게임 실행 파일에 의해 액세스되지 않은 상기 게임 데이터의 미사용 블록을 나타냄 -,

상기 비휘발성 메모리로부터 상기 게임 데이터의 상기 하나 이상의 제2 블록을 삭제하게 하는, 클라이언트 머신.

**청구항 7**

제1항에 있어서, 상기 클라이언트 머신은,

제1 속도로 판독 액세스를 제공하도록 구성되고, 상기 게임 데이터를 저장하는 제1 메모리; 및

상기 제1 속도보다 더 빠른 제2 속도로 판독 액세스를 제공하도록 구성된 제2 메모리를 더 포함하되,

상기 컴퓨터 실행 가능 명령어는, 상기 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서로 하여금 추가로,

상기 원격 시스템으로부터, 상기 클라이언트 머신의 상기 컨피규레이션과 연관된 블록 종속성 데이터를 수신하고 - 상기 블록 종속성 데이터는 상기 게임 데이터의 상기 블록들 중 두 개 이상의 블록 사이의 개별 연관성을 상술함 -,

이벤트를 검출하고,

상기 블록 종속성 데이터에 상기 이벤트와 연관된 것으로서 상술된 상기 게임 데이터의 상기 제1 블록 또는 상기 게임 데이터의 상기 제2 블록 중 적어도 하나를 상기 제2 메모리에 캐시(cache)하고,

상기 제1 판독 작동 또는 상기 제2 판독 작동 중 적어도 하나를 수신하고,

상기 제2 메모리로부터 상기 게임 데이터의 상기 제1 블록 또는 상기 게임 데이터의 상기 제2 블록 중 적어도 하나를 판독하게 하는, 클라이언트 머신.

### 청구항 8

제1항에 있어서, 상기 클라이언트 머신은,

제1 속도로 판독 액세스를 제공하도록 구성되고, 상기 게임 데이터를 저장하는 제1 비휘발성 메모리;

상기 제1 속도보다 더 빠른 제2 속도로 판독 액세스를 제공하도록 구성된 제2 비휘발성 메모리; 및

상기 제2 속도보다 더 빠른 제3 속도로 판독 액세스를 제공하도록 구성된 휘발성 메모리를 더 포함하되,

상기 컴퓨터 실행 가능 명령어는, 상기 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서로 하여금 추가로,

상기 원격 시스템으로부터, 상기 클라이언트 머신의 상기 컨피규레이션과 연관된 블록 종속성 데이터를 수신하고 - 상기 블록 종속성 데이터는 상기 게임 데이터의 상기 블록들 중 두 개 이상의 블록 사이의 개별 연관성을 상술함 -,

이벤트를 검출하고,

상기 블록 종속성 데이터에 상기 이벤트와 연관된 것으로서 상술된 상기 게임 데이터의 상기 제1 블록을 상기 제2 비휘발성 메모리에 캐시하고,

상기 블록 종속성 데이터에 상기 이벤트와 연관된 것으로서 상술된 상기 게임 데이터의 상기 제2 블록을 상기 휘발성 메모리에 캐시하고,

상기 제1 판독 작동을 수신하고,

상기 게임 데이터의 상기 제1 블록을 상기 제2 비휘발성 메모리로부터 판독하고,

상기 제2 판독 작동을 수신하고,

상기 게임 데이터의 상기 제2 블록을 상기 휘발성 메모리로부터 판독하게 하는, 클라이언트 머신.

### 청구항 9

방법으로서,

클라이언트 머신의 프로세서가, 상기 클라이언트 머신 상에서 비디오 게임을 플레이하기 위해 상기 비디오 게임의 게임 실행 파일을 실행하는 단계;

상기 프로세서가, 상기 클라이언트 머신의 파일 시스템에 대해 상기 게임 실행 파일에 의해 수행되는 제1 판독 작동 및 제2 판독 작동을 결정하는 단계 - 상기 제1 판독 작동들은 상기 비디오 게임을 위한 게임 데이터의 제1 블록을 판독할 것을 요청하고, 상기 제2 판독 작동은 상기 게임 데이터의 제2 블록을 판독할 것을 요청함 -;

상기 프로세서가, 상기 판독 작동들에 적어도 부분적으로 기초하여 액세스 데이터를 생성하는 단계 - 상기 액세스 데이터는,

상기 게임 데이터의 상기 제1 블록의 제1 식별자;

상기 게임 실행 파일의 실행 동안, 상기 게임 데이터의 상기 제1 블록이 상기 제1 판독 작동에 적어도 부분적으로 기초하여 액세스된 제1 시간;

상기 게임 데이터의 상기 제2 블록의 제2 식별자; 및

상기 게임 실행 파일의 상기 실행 동안, 상기 게임 데이터의 상기 제2 블록이 상기 제2 판독 작동에 적어도 부분적으로 기초하여 액세스된 제2 시간을 상술함 -; 및

상기 액세스 데이터, 상기 비디오 게임의 식별자, 및 상기 클라이언트 머신의 컨피규레이션을 원격 시스템에 송신하는 단계를 포함하는 방법.

**청구항 10**

제9항에 있어서, 상기 게임 데이터의 복수의 제1 블록은 적어도 상기 게임 데이터의 상기 제1 블록 및 상기 게임 데이터의 상기 제2 블록을 포함하고, 상기 방법은,

상기 클라이언트 머신의 상기 비휘발성 메모리로부터 상기 게임 데이터의 하나 이상의 제2 블록을 삭제하라는 명령어를 상기 원격 시스템으로부터 수신하는 단계 - 상기 게임 데이터의 상기 하나 이상의 제2 블록은 임계 시간 기간 또는 임계 수의 게임 세션 동안 상기 클라이언트 머신 상에서 상기 게임 실행 파일에 의해 액세스되지 않은 상기 게임 데이터의 미사용 블록을 나타냄 -; 및

상기 비휘발성 메모리로부터 상기 게임 데이터의 상기 하나 이상의 제2 블록을 삭제하게 하는 단계를 더 포함하는, 방법.

**청구항 11**

제9항에 있어서, 상기 비디오 게임은 제1 비디오 게임이고, 상기 게임 실행 파일은 상기 제1 비디오 게임의 제1 게임 실행 파일이고, 상기 게임 데이터는 상기 제1 비디오 게임을 위한 제1 게임 데이터이고, 상기 방법은,

제2 비디오 게임을 획득하기 위한 요청을 상기 원격 시스템에 송신하는 단계 - 상기 요청은 상기 클라이언트 머신의 상기 컨피규레이션을 포함함 -;

상기 원격 시스템으로부터,

상기 제2 비디오 게임의 제2 게임 실행 파일; 및

상기 클라이언트 머신의 상기 컨피규레이션과 연관된 다운로드 시퀀스 데이터 - 상기 다운로드 시퀀스 데이터는 상기 제2 비디오 게임을 위한 제2 게임 데이터의 블록들의 시퀀스를 상술함 -;를 수신하는 단계;

상기 다운로드 시퀀스 데이터에 상술된 상기 시퀀스에 따라 상기 제2 게임 데이터의 상기 블록들을 상기 클라이언트 머신의 비휘발성 메모리에 다운로드하기 시작하는 단계; 및

상기 프로세서가, 상기 클라이언트 머신 상에서 상기 제2 비디오 게임을 실행하기 위해 상기 제2 게임 실행 파일을 실행하는 단계를 더 포함하는, 방법.

**청구항 12**

제11항에 있어서,

상기 프로세서가, 상기 파일 시스템에 대해 상기 제2 게임 실행 파일에 의해 수행되는 제3 판독 작동을 수신하는 단계 - 상기 제3 판독 작동은 상기 제2 게임 데이터의 블록을 판독할 것을 요청함 -;

상기 제2 게임 데이터의 상기 블록이 상기 비휘발성 메모리에 다운로드되었다고 결정하는 단계; 및

상기 프로세서가, 상기 파일 시스템을 사용하여 상기 제2 게임 데이터의 상기 블록을 판독하는 단계를 더 포함하는, 방법.

**청구항 13**

제11항에 있어서,

상기 프로세서가, 상기 파일 시스템에 대해 상기 제2 게임 실행 파일에 의해 수행되는 제3 판독 작동을 수신하는 단계 - 상기 제3 판독 작동은 상기 제2 게임 데이터의 블록을 판독할 것을 요청함 -;

상기 제2 게임 데이터의 상기 블록이 상기 비휘발성 메모리에 다운로드되는 것이 완료되지 않았다고 결정하는 단계;

상기 제2 게임 데이터의 상기 블록에 대한 제2 요청을 상기 원격 시스템에 송신하는 단계;

상기 원격 시스템으로부터 상기 제2 게임 데이터의 상기 블록을 수신하는 단계; 및

상기 프로세서가, 상기 파일 시스템을 사용하여 상기 제2 게임 데이터의 상기 블록을 판독하는 단계를 더 포함하는, 방법.

**청구항 14**

제9항에 있어서, 상기 게임 데이터는 상기 클라이언트 머신의 제1 메모리에 저장되고, 상기 제1 메모리는 제1 속도로 판독 액세스를 제공하도록 구성되고, 상기 방법은,

상기 원격 시스템으로부터, 상기 클라이언트 머신의 상기 컨피규레이션과 연관된 블록 종속성 데이터를 수신하는 단계 - 상기 블록 종속성 데이터는 상기 게임 데이터의 상기 블록들 중 두 개 이상의 블록 사이의 개별 연관성을 상술함 -;

상기 프로세서가 이벤트를 검출하는 단계;

상기 블록 종속성 데이터에 상기 이벤트와 연관되는 것으로서 상술된 상기 게임 데이터의 상기 제1 블록 또는 상기 게임 데이터의 상기 제2 블록 중 적어도 하나를, 상기 제1 속도보다 더 빠른 제2 속도로 판독 액세스를 제공하도록 구성된 상기 클라이언트 머신의 제2 메모리에 캐싱하는 단계;

상기 프로세서가 상기 제1 판독 작동 또는 상기 제2 판독 작동 중 적어도 하나를 수신하는 단계; 및

상기 프로세서가, 상기 제2 메모리로부터 상기 게임 데이터의 상기 제1 블록 또는 상기 게임 데이터의 상기 제2 블록 중 적어도 하나를 판독하는 단계를 더 포함하는, 방법.

**청구항 15**

제14항에 있어서, 상기 이벤트를 검출하는 단계는, 상기 게임 실행 파일이 상기 게임 데이터의 상기 제1 블록을 판독할 것을 요청하였다는 것을 검출하는 단계를 포함하고, 상기 캐싱하는 단계는, 상기 게임 데이터의 상기 제2 블록이 상기 게임 데이터의 상기 제1 블록과 연관된다는 것을 상술하는 상기 블록 종속성 데이터에 적어도 부분적으로 기초하여 상기 제2 메모리에 상기 게임 데이터의 상기 제2 블록을 캐싱하는 단계를 포함하는, 방법.

**청구항 16**

방법으로서,

원격 시스템이, 클라이언트 시스템 컨피규레이션을 공통으로 갖는 다수의 클라이언트 머신으로부터 비디오 게임과 연관된 액세스 데이터를 수신하는 단계 - 상기 액세스 데이터는 상기 다수의 클라이언트 머신 중 개별 머신에 대해 적어도,

개별 클라이언트 머신 상에서 상기 비디오 게임의 게임 실행 파일의 실행 동안 상기 게임 실행 파일에 의해 액세스된 상기 비디오 게임을 위한 게임 데이터의 제1 블록의 제1 식별자;

상기 개별 클라이언트 머신 상에서 상기 게임 실행 파일의 상기 실행 동안, 상기 게임 데이터의 상기 제1 블록이 상기 게임 실행 파일에 의해서 액세스된 제1 시간;

상기 개별 클라이언트 머신 상에서 상기 게임 실행 파일의 상기 실행 동안, 상기 게임 실행 파일에 의해 액세스된 상기 게임 데이터의 제2 블록의 제2 식별자; 및

상기 개별 클라이언트 머신 상에서 상기 게임 실행 파일의 상기 실행 동안, 상기 게임 데이터의 상기 제2 블록이 상기 게임 실행 파일에 의해서 액세스된 제2 시간을 상술함 -;

상기 액세스 데이터를 분석하는 단계;

상기 원격 시스템의 프로세서가, 상기 액세스 데이터를 분석하는 단계에 적어도 부분적으로 기초하여,

적어도 상기 게임 데이터의 상기 제1 블록 및 상기 게임 데이터의 상기 제2 블록이 상기 클라이언트 시스템 컨피규레이션을 갖는 클라이언트 머신들에 다운로드되는 시퀀스를 상술하는 다운로드 시퀀스 데이터; 또는

상기 게임 데이터의 상기 제1 블록과 상기 게임 데이터의 상기 제2 블록 사이의 연관성, 또는 상기 게임 데이터의 상기 제1 블록 또는 상기 게임 데이터의 상기 제2 블록과 이벤트 사이의 연관성을 상술하는 블록 종속성 데이터 중 적어도 하나를 포함하는 데이터를 생성하는 단계;

상기 원격 시스템이 상기 클라이언트 시스템 컨피규레이션을 갖는 하나 이상의 클라이언트 머신에 상기 데이터를 송신하는 단계를 포함하는, 방법.

**청구항 17**

제16항에 있어서, 상기 데이터는 상기 다운로드 시퀀스 데이터를 포함하고, 상기 방법은,

상기 원격 시스템이, 상기 클라이언트 시스템 컨피규레이션을 갖는 클라이언트 머신으로부터, 상기 비디오 게임을 획득하기 위한 요청을 수신하는 단계;

상기 원격 시스템이, 상기 비디오 게임의 상기 게임 실행 파일을 상기 클라이언트 머신으로 송신하는 단계; 및 적어도 상기 게임 데이터의 상기 제1 블록과 상기 게임 데이터의 상기 제2 블록을, 상기 다운로드 시퀀스 데이터에 상술된 상기 시퀀스에 따라, 상기 클라이언트 머신에 다운로드하는 단계를 더 포함하는, 방법.

**청구항 18**

제16항에 있어서, 상기 데이터는 상기 다운로드 시퀀스 데이터를 포함하고, 상기 액세스 데이터를 분석하는 단계는,

상기 게임 데이터의 상기 제1 블록과 연관된 제1 통계 및 상기 게임 데이터의 상기 제2 블록과 연관된 제2 통계를 계산하는 단계;

상기 제1 통계 및 상기 제2 통계에 적어도 부분적으로 기초하여 상기 시퀀스를 결정하는 단계를 포함하는, 방법.

**청구항 19**

제16항에 있어서, 상기 데이터는 상기 블록 종속성 데이터를 포함하고, 상기 액세스 데이터를 분석하는 단계는,

상기 게임 데이터의 상기 제1 블록과 상기 게임 데이터의 상기 제2 블록 사이의 연관성을 결정하는 단계; 또는

컨택스트 큐와 상기 게임 데이터의 상기 제1 블록 또는 상기 게임 데이터의 상기 제2 블록 중 적어도 하나 사이의 연관성을 결정하는 단계 중 적어도 하나를 포함하는, 방법.

**청구항 20**

제16항에 있어서,

상기 다수의 클라이언트 머신 중 하나로부터 수신된 상기 액세스 데이터에 적어도 부분적으로 기초하여, 상기 게임 데이터의 하나 이상의 블록이 임계 시간 기간 또는 임계 수의 게임 세션 동안 상기 다수의 클라이언트 머신 중 상기 하나에서 상기 게임 실행 파일에 의해 액세스되지 않았다고 결정하는 단계;

상기 원격 시스템이, 상기 다수의 클라이언트 머신 중 상기 하나의 비휘발성 메모리로부터, 상기 게임 데이터의 상기 하나 이상의 블록을 삭제하라는 명령어를 상기 다수의 클라이언트 머신 중 상기 하나로 송신하는 단계를 더 포함하는, 방법.

**발명의 설명**

**기술 분야**

[0001] 관련 출원의 교차 참조

[0002] 본 출원은, 2020년 3월 17일자로 출원된, 발명의 명칭이 "TRACKING FILE SYSTEM READ OPERATIONS FOR INSTANT PLAY OF VIDEO GAMES, AND FOR CLIENT-SIDE DISCARDING AND PREFETCHING OF GAME DATA"인 미국 특허 출원 제 16/821,716호의 우선권을 주장하는 PCT 출원이며, 이 미국 특허 출원의 전체 내용이 본원에 인용되어 포함된다.

**배경 기술**

[0003] 개인용 컴퓨터(PC: personal computer) 비디오 게임의 배포를 위한 서비스는, 비디오 게임의 디지털 사본을 PC에 배포하기 위해서 네트워크 액세스 가능한 컴퓨팅 플랫폼을 활용한다. 예를 들어, 사용자는 배포 서비스를 통해 사용 가능한 비디오 게임을 구매할 수 있고, 비디오 게임은 인터넷을 통해 원격 컴퓨팅 시스템으로부터 사용자의 PC로 다운로드될 수 있다. 오늘날의 많은 비디오 게임은 상대적으로 크고, 결과적으로, 비디오 게임을 다운로드하기 위해서 상당한 양의 시간이 걸릴 수 있고, 또한, 비디오 게임이 사용자의 PC에 다운로드되면, 비디

오 게임을 위한 게임 데이터 전부를 저장하기 위해서 PC 상의 상당한 양의 디스크 저장 공간을 차지할 수 있다. 예를 들어, 비디오 게임은 100 기가바이트(GB: gigabyte)를 초과하는 게임 데이터를 포함할 수 있고, 이는, 사용자 네트워크 연결의 다운로드 속도에 따라, 기존 기술을 사용하여 다운로드하기 위해서 몇 시간이 걸릴 수 있다. 비디오 게임의 개발자가, PC에 다운로드된 게임 데이터의, 전부는 아니지만, 일부로 비디오 게임이 플레이 가능하도록 게임 코드를 작성하지 않는 한, 사용자는 게임 다운로드가 완료될 때까지 게임을 플레이하기 위해 기다려야 한다. 또한, 대부분의 게임은 매우 크기 때문에, 사용자는, PC 상에서 가장 많은 저장 공간을 제공하는 하드 디스크 드라이브(HDD: hard disk drive) 상에 게임을 저장하는 것을 주로 선택한다. 그러나, 현재 대용량 HDD의 가용성에도 불구하고, 로컬 저장 용량은 여전히 제한적이며, HDD의 관독 액세스 속도가 상대적으로 느리기 때문에, 게임 세션 동안 PC가 HDD로부터 게임 데이터를 로드할 때 레이턴시(latency)가 발생할 수 있다.

[0004] 이러한 시스템 및 다른 시스템을 개선시키고 향상시키기 위한 기술적 솔루션들이 본원에서 제공된다.

**도면의 간단한 설명**

[0005] 상세한 설명은 수반된 도면을 참조하여 설명된다. 도면에서, 참조 번호의 맨 왼쪽 숫자(들)는 이 참조 번호가 처음으로 나타나는 도면을 식별한다. 여러 도면에서의 동일한 참조 번호의 사용은 유사하거나 동일한 구성 요소 또는 특징부를 나타낸다.

도 1은 본원에 설명된 기술을 구현하도록 구성된 비디오 게임 배포 플랫폼을 포함하는 예시적인 환경을 예시하는 다이어그램이다.

도 2는, 클라이언트 머신의 예시적인 컴포넌트를 예시하는 블록도, 및 액세스 데이터를 생성하도록 파일 시스템 관독 작동을 추적하기 위한, 그리고 액세스 데이터를 원격 시스템에 전송하기 위한, 예시적인 프로세스의 흐름도를 도시한다.

도 3은 원격 시스템의 예시적인 컴포넌트를 예시하는 블록도, 및 클라이언트 머신으로부터 액세스 데이터를 수신하기 위한, 그리고 본원에 설명되는 기술을 구현하기 위해 하나 이상의 사용자에게 걸쳐 액세스 데이터를 분석하기 위한 예시적인 프로세스의 흐름도를 도시한다.

도 4는 게임 데이터의 블록들을 블록들의 특정 시퀀스(sequence)로 다운로드하기 위해 클라이언트 머신에 의해 사용되기 위한 다운로드 시퀀스 데이터를, 다수의 클라이언트 머신으로부터 수신된 액세스 데이터에 기초하여, 생성하기 위한 예시적인 프로세스의 흐름도이다.

도 5는, 게임플레이 동안 레이턴시를 감소시키도록 게임 데이터의 블록을 프리페치(prefetch)하기 위해 클라이언트 머신에 의해 사용되기 위한 블록 중속성 데이터를, 다수의 클라이언트 머신으로부터 수신된 액세스 데이터에 기초하여, 생성하기 위한 예시적인 프로세스의 흐름도이다.

도 6은, 클라이언트 머신으로부터 수신된 액세스 데이터에 기초하여, 게임 데이터의 미사용 블록을 결정하고, 게임 데이터의 미사용 블록을 폐기하도록 클라이언트 머신에 지시하기 위한 예시적인 프로세스의 흐름도이다.

도 7은 클라이언트 머신의 비휘발성 메모리로부터 게임 데이터의 미사용 블록을 폐기하기 위한 예시적인 프로세스의 흐름도이다.

도 8은 게임 다운로드가 완료되기 전에 클라이언트 머신 상에서 비디오 게임을 실행하기 위한 예시적인 프로세스의 흐름도이다.

도 9는 게임플레이 동안 레이턴시를 감소시키기 위해 게임 데이터의 블록을 프리페치하기 위한 예시적인 프로세스의 흐름도이다.

**발명을 실시하기 위한 구체적인 내용**

[0006] 비디오 게임 배포 플랫폼은 사용자가 비디오 게임을 획득하는 것, 비디오 게임의 디지털 사본을 자신의 각각의 클라이언트 머신에 다운로드하는 것, 인스톨(install)된 클라이언트 애플리케이션을 사용하여 비디오 게임을 실행하는 것을 허용할 것이다. 사용자가 자신의 클라이언트 머신 상에서 비디오 게임을 플레이할 때, 각각의 클라이언트 머신 상에서 런닝(running)되는 비디오 게임의 게임 실행 파일은 클라이언트 머신의 파일 시스템을 사용하여 비디오 게임을 위한 게임 데이터의 블록을 판독할 것을 지속적으로 요청할 수 있다. 본원에서 사용되는 "게임 실행 파일"은, 실행될 때, 클라이언트 머신의 입력 디바이스(들)를 통해 제공되는 사용자 입력에 기초하여 프레임을 렌더링함으로써 사용자가 클라이언트 머신 상에서 비디오 게임을 플레이하는 것을 허용하는 비디오 게

임의 실행 가능한 코드(예를 들어, 하나 이상의 실행 가능한 파일)를 의미한다. 본원에서 사용되는 "게임 데이터"는 일련의 프레임에 걸쳐 비디오 게임을 위한 게임 실행 파일의 실행 동안 게임 실행 파일에 의해 판독되는 데이터를 의미한다. 게임 데이터는, 무엇보다도, 비디오 게임을 위한 그래픽을 렌더링하고, 게임플레이 로직을 계산하고, 그리고/또는 비디오 게임을 위한 물리학을 계산하기 위해서 사용될 수 있다. 게임 데이터의 예는, 한정됨 없이, 텍스처, 가상 객체, 지도, 게임 캐릭터, 가상 객체의 매개변수와 다른 특징 및/또는 가상 게임 월드 등을 포함한다. 이러한 게임 데이터는, 게임 데이터를 다수의 블록(예를 들어, 균일한 크기의 블록들)으로 분할함으로써 메모리에 저장될 수 있고, 클라이언트 머신의 파일 시스템은, 게임 데이터의 이러한 블록이 클라이언트 머신의 로컬 메모리에 저장되고 이로부터 검색되는 방식을 제어하도록 구성된다. 비디오 게임이 클라이언트 머신 상에서 플레이될 때, 게임 데이터의 블록(들)을 판독하라는, 비디오 게임의 게임 실행 파일(들)에 의해 만들어진 각각의 요청은 본원에서 "판독 작동"으로 지칭된다. 따라서, 비디오 게임의 게임 실행 파일은 게임 세션 전체에 걸쳐서 파일 시스템에 대한 일련의 판독 작동을 수행할 수 있다.

[0007] 본원에 설명된 기술, 디바이스, 및 시스템은, 게임 세션 동안 비디오 게임의 게임 실행 파일에 의해 수행된 판독 작동을 추적하고, 추적된 판독 작동에 기초하여 액세스 데이터를 생성하고, 액세스 데이터를 원격 시스템에 리포트하기 위해서 클라이언트 머신의 파일 시스템 프로시 컴포넌트를 사용하는 것에 관한 것이다. 클라이언트 머신에 의해 구현되는 예시적인 프로세스는, 클라이언트 머신 상에서 비디오 게임을 플레이하기 위해 비디오 게임의 게임 실행 파일을 실행하는 것, 클라이언트 머신의 파일 시스템에 대한 게임 실행 파일에 의해 수행되는 판독 작동을 결정하는 것, 판독 작동에 적어도 부분적으로 기초하여 액세스 데이터를 생성하는 것, 및 원격 시스템에 액세스 데이터, 비디오 게임의 식별자 및 클라이언트 머신의 컨피규레이션을 송신하는 것을 포함할 수 있다. 게임 실행 파일에 의해 수행되는 판독 작동은 비디오 게임을 위한 게임 데이터의 블록을 판독할 것을 요청할 수 있다. 따라서, 클라이언트 머신에 의해 생성된 액세스 데이터는 (i) 게임 세션 동안 액세스된 게임 데이터의 블록의 식별자, 및 (ii) 게임 실행 파일의 실행 동안, 게임 데이터의 블록이 판독 작동에 적어도 부분적으로 기초하여 액세스된 시간을 상술할 수 있다.

[0008] 이 원격 측정(telemetry) 접근 방식은, 원격 시스템이 다수의 클라이언트 머신에 의해서 보고된 액세스 데이터를 수집하고, 클라이언트 시스템 컨피규레이션에 따라 액세스 데이터를 카탈로그화하고, 액세스 데이터를 분석하는 것을 허용하여, 다양한 게임 관련 기능을 구현하기 위해서 클라이언트 머신에 의해서 사용 가능한 데이터를 생성할 수 있으며, 이 기능은, 한정됨 없이, 비디오 게임의 "인스턴트 플레이(instant play)", 로컬 메모리 리소스를 확보하기 위해 게임 데이터의 미사용(unused) 블록의 폐기, 및/또는 게임 플레이 동안 레이턴시를 감소시키기 위한 게임 데이터의 로컬 프리페치(local prefetching)를 포함한다. 원격 시스템에 의해서 구현될 예시적인 프로세스는, 공통으로 클라이언트 시스템 컨피규레이션을 갖는 다수의 클라이언트 머신으로부터 비디오 게임과 연관된 액세스 데이터를 수신하는 것, 액세스 데이터를 분석하는 것, 액세스 데이터의 분석에 적어도 부분적으로 기초하여 데이터를 생성하는 것, 클라이언트 시스템 컨피규레이션을 갖는 하나 이상의 클라이언트 머신에 데이터를 송신하는 것을 포함할 수 있다. 원격 시스템에 의해서 수신된 액세스 데이터는, 클라이언트 머신들 중 개별 머신에 대해서, (i) 개별 클라이언트 머신 상에서 게임 실행 파일을 실행하는 동안 비디오 게임의 게임 실행 파일에 의해 액세스되었던 비디오 게임을 위한 게임 데이터의 블록의 식별자, 및 (ii) 게임 실행 파일의 실행 동안, 게임 데이터의 블록이 게임 실행 파일에 의해 액세스되었던 시간을 상술할 수 있다. 또한, 액세스 데이터를 분석하는 것에 적어도 부분적으로 기초하여 원격 시스템에 의해서 생성된 데이터는, (i) 게임 데이터의 블록들 중 적어도 일부가 클라이언트 시스템 컨피규레이션을 갖는 클라이언트 머신들에 다운로드되는 시퀀스를 상술하는 다운로드 시퀀스 데이터; 또는 (ii) 게임 데이터의 두 개 이상의 블록들 사이의 개별 연관성을 상술하는 블록 종속성 데이터 중 적어도 하나를 포함할 수 있다.

[0009] 상술된 바와 같이, 활성화될 수 있는 하나의 예시적인 게임 관련 기능은 "인스턴트 플레이" 기능이다. 본원에 설명된 인스턴트 플레이 기능은, 비디오 게임을 획득하는 사용자가, 비디오 게임을 획득한 때, 그리고 게임 데이터가 사용자의 클라이언트 머신으로 다운로드를 완료하기 전에, 비디오 게임의 플레이를 시작하는 것을 허용할 수 있다. 따라서, 사용자는, 비디오 게임의 게임 세션을 시작하기 전에, 비디오 게임이 다운로드를 완료할 때까지 기다릴 필요가 없다. 인스턴트 플레이 기능을 가능하게 하기 위해, 원격 시스템은 다수의 클라이언트 머신으로부터, 특정 비디오 게임과 관련된 액세스 데이터(본원에 설명된 바와 같음)를 수신할 수 있고, 원격 시스템은, 액세스 데이터에 기초하여, 비디오 게임을 위한 게임 데이터의 블록들의 시퀀스를 상술하는 다운로드 시퀀스 데이터를 생성할 수 있다. 이 시퀀스는, 먼저 액세스될 가능성이 더 높은 블록을 시퀀스의 시작 쪽으로 위치시키고, 먼저 액세스될 가능성이 낮은 블록을 시퀀스의 끝 쪽으로 위치시킬 수 있다. 이러한 방식으로, 게임 데이터의 블록들이 클라이언트 머신으로 다운로드되기 시작하면, 게임 세션 동안 초기에 액세스될 가능성이 가장 높은 게임 데이터의 블록은, 게임 세션 동안 초기에 액세스될 가능성이 낮은 게임 데이터의 다른 블록을 다

운로드하기 전에, 클라이언트 머신의 비휘발성 메모리에 저장된다. 이것은, 게임 데이터의 블록들이 여전히 클라이언트 머신의 비휘발성 메모리에 다운로드되는 동안에도, 사용자가 비디오 게임의 획득 시 비디오 게임의 플레이를 시작할 수 있는 것을 허용한다. 사실, 본원에 설명된 기술 및 시스템은, 게임 데이터의 제1 블록이 사용자의 클라이언트 머신으로 다운로드되는 것을 완료하기 전에도, 사용자가 비디오 게임의 플레이를 시작하는 것을 허용한다. 이것은 클라이언트 머신 상에서 파일 시스템 프록시 컴포넌트를 사용하는 것에 의해서 적어도 부분적으로 가능해지며, 이 컴포넌트는 게임 실행 파일에 의해 수행된 판독 작동을 수신하고, 게임 데이터의 요청된 블록이 비휘발성 메모리에 다운로드되었는지 또는 블록이 여전히 다운로드를 완료해야 하는지 여부를 결정하도록 구성된다. 블록이 비휘발성 메모리에서 "실현"되면, 게임 데이터의 블록이 비휘발성 메모리로 다운로드되는 것을 완료하였다는 것을 의미하며, 게임 실행 파일은 파일 시스템을 사용하여 게임 데이터의 블록을 판독할 수 있다. 블록이 비휘발성 메모리로 다운로드되는 것을 완료하지 않은 경우, 파일 시스템 프록시 컴포넌트는 판독 작동을 인터셉트하고, 게임 데이터의 미실현 블록을 원격 시스템으로부터 요청할 수 있고, 일단 블록이 원격 시스템으로부터 수신되면, 블록은 파일 시스템을 사용하여 판독될 수 있다. 게임 데이터의 미실현 블록이 원격 시스템으로부터 검색되는 동안, 비디오 게임의 실행에서 일시 중지가 있을 수 있지만, 게임 데이터의 다운로드가 비디오 게임의 획득 시 시작된다고 가정하고, 게임 데이터의 블록들이, 게임 세션 동안 게임 실행 파일이 데이터의 블록들에 액세스하는 시퀀스와 일치하는 시퀀스로 다운로드된다고 가정하면, 이러한 일이 자주 또는 항상 발생할 가능성은 낮다.

[0010] 활성화될 수 있는 다른 예시적인 게임 관련 기능은 게임 데이터의 사용되지 않는 블록을 폐기함으로써 클라이언트 머신 상에서 로컬 메모리 리소스를 확보하는 것이다. 로컬 메모리 리소스를 확보함으로써, 클라이언트 머신은 다른 비디오 게임의 게임 데이터 및/또는 일반적으로 다른 데이터를 위해서 활용될 수 있는 유용한 저장 용량을 회수할 수 있다. 게임 데이터의 클라이언트 측 폐기를 가능하게 하기 위해, 클라이언트 머신(비휘발성 메모리에 저장된 비디오 게임의 게임 데이터를 구비함)은 비디오 게임의 게임 실행 파일을 실행할 수 있고, 클라이언트 머신은, 판독 작동이 클라이언트 머신의 파일 시스템에 대해 게임 실행 파일에 의해 수행될 때, 하나 이상의 게임 세션에 걸쳐 액세스 데이터를 생성할 수 있다. 이 액세스 데이터는, 하나 이상의 게임 세션 동안 액세스된 게임 데이터의 제1 서브세트의 블록뿐만 아니라, 게임 세션(들) 동안 이러한 블록이 액세스된 시간을 상술할 수 있다. 원격 시스템은 클라이언트 머신으로부터 액세스 데이터를 수신할 수 있고, 원격 시스템은, 적어도 임계 시간 기간 또는 임계 수의 게임 세션 동안 블록이 클라이언트 머신에서 게임 실행 파일에 의해 액세스되지 않은 것에 기초하여 "미사용" 블록으로 분류될 수 있는 게임 데이터의 하나 이상의 제2 블록을 액세스 데이터에 기초하여 결정할 수 있다. 이 경우에, 원격 시스템은, 비휘발성 메모리로부터 게임 데이터의 하나 이상의 제2 블록을 삭제할 것을 클라이언트 머신에 지시하는 명령어를 클라이언트 머신에 보낼 수 있다. 클라이언트 머신이 비휘발성 메모리로부터 이러한 미사용 블록을 삭제하면, 클라이언트 머신의 저장 용량이 증가될 수 있다. 또한, 게임 실행 파일이 삭제된 게임 데이터의 블록을 판독할 것을 요청하는 경우, 클라이언트 머신의 파일 시스템 프록시 컴포넌트는 주문형(on-demand) 방식으로 게임 데이터의 블록을 원격 시스템으로부터 요청할 수 있으며, 로컬 메모리에 블록을 저장하는 것과 비교하여 약간의 레이턴시가 추가된다.

[0011] 활성화될 수 있는 다른 예시적인 게임 관련 기능은, 게임 세션 동안 게임 데이터를 로드할 때 레이턴시를 줄이기 위해 게임 데이터의 블록을 로컬 프리페칭(local prefetching)하는 것이다. 로컬 프리페칭 기능을 가능하게 하기 위해, 원격 시스템은 다수의 클라이언트 머신으로부터, 특정 비디오 게임과 관련된 액세스 데이터(본원에 설명된 바와 같음)를 수신할 수 있고, 원격 시스템은, 액세스 데이터에 기초하여, 게임 데이터의 둘 이상의 블록 사이의 개별 연관성을 서술하는 블록 종속성 데이터를 생성할 수 있다. 예를 들어, 블록 종속성 데이터는, 게임 세션 동안 게임 데이터의 제1 블록이 액세스될 때마다, 게임 데이터의 제2 블록이 일반적으로 임계 시간 기간 내에 액세스되는 것을 나타낼 수 있다. 이러한 방식으로, 블록 종속성 데이터는, 원격 시스템에서 수신된 액세스 데이터에 나타나는 액세스 패턴에 기초하여 두 개 이상의 블록의 세트들 간의 블록간 관계를 설명할 수 있다. 원격 시스템은 비디오 게임이 인스톨된 클라이언트 머신에 블록 종속성 데이터를 송신할 수 있고, 클라이언트 머신이 비디오 게임의 게임 실행 파일을 실행할 때, 클라이언트 머신은, 게임 데이터가 지속적으로 저장된 메모리(예를 들어, HDD, SD 카드와 같은 비휘발성 메모리)보다 더 빠른 판독 액세스 속도를 제공하는 로컬 메모리에 블록을 캐싱함으로써, 게임 데이터의 블록을 프리페칭할 수 있다. 이러한 로컬 프리페칭은, 게임 실행 파일이 게임 데이터의 블록을 판독할 것을 요청할 때, 로드 시간의 레이턴시를 감소시킬 수 있다.

[0012] 본원에 설명된 기술 및 시스템은, 오늘날 게임 개발자가 비디오 게임을 만드는 방식을 변경하지 않으면서, 클라이언트 머신의 게임플레이 기능을 개선할 수 있다. 예를 들어, 본원에 설명된 파일 시스템 프록시 컴포넌트와 같은 클라이언트 측 컴포넌트를 구현하는 것은, 클라이언트 머신의 사용자가 비디오 게임의 획득 시 비디오 게임을 플레이하는 것을 허용하며, 이는, 사용자가 게임 세션을 시작하기 전에 게임이 다운로드되는 것을 (잠재적으

로 몇 시간 동안) 기다릴 필요가 없다는 것을 의미한다. 이것은 또한, 먼저 액세스될 가능성이 가장 높은 블록을 시퀀스의 시작부에 위치시키는 블록의 시퀀스로 게임 데이터가 지능적으로 다운로드되는 것을 허용하며, 이는 게임 데이터의 다운로드가 진행되는 동안 게임 플레이 중에 레이턴시를 감소시키는 것을 돕는다. 클라이언트 머신의 게임플레이 기능은, 추가적으로, 또는 대안적으로, 블록 종속성 데이터에 따라 게임 세션 동안 게임 데이터를 프리페치함으로써 개선될 수 있다. 이것은, 클라이언트 머신이 비디오 게임의 게임 실행 파일에 의해서 수행되는 일련의 판독 작동에서 다음에 액세스될 가능성이 높은 게임 데이터의 블록을 예측할 수 있고, 이러한 블록이, 게임 데이터가 지속되는 비휘발성 메모리보다 더 빠른 판독 액세스 속도(예를 들어, 랜덤 액세스 메모리(RAM: random access memory)와 같은 휘발성 메모리)를 제공하는 로컬 메모리에 캐시될 수 있기 때문이다. 이것은, 비디오 게임이 게임 데이터를 요청할 때, 빠른 액세스를 위해 게임 데이터를 "프라임(prime)"한다.

[0013] 본원에 설명된 기술 및 시스템은, 추가적으로 또는 대안적으로, 하나 이상의 디바이스가 적어도 메모리 리소스에 관해서 리소스를 절약하는 것을 허용한다. 예를 들어, 하나 이상의 클라이언트 머신에 의해서 생성된 액세스 데이터를 사용하여 게임 데이터의 미사용 블록을 결정하고, 게임 데이터의 미사용 블록을 삭제하면 클라이언트 머신에서 로컬 메모리 리소스를 확보할 수 있다. 예를 들어, 클라이언트 머신 및/또는 원격 시스템이, 추적된 파일 시스템 판독 작동과 관련된 액세스 데이터로부터, 사용자가 싱글 플레이어 모드에서 게임을 플레이한 적이 없고, 항상 멀티 플레이어 모드에서 게임을 플레이한다고 결정하는 경우, 비디오 게임을 위한 싱글 플레이어 게임 데이터는, 이러한 블록들이 적어도 임계 시간 기간 또는 임계 수의 게임 세션 동안 사용되지 않았다는 결정에 기초하여, 클라이언트 머신의 비휘발성 메모리로부터 삭제될 수 있다.

[0014] 도 1은 본원에 설명된 기술을 구현하도록 구성된 비디오 게임 배포 플랫폼을 포함하는 예시적인 환경(100)을 예시하는 다이어그램이다. 커뮤니티의 사용자(102)(때때로 "고객"으로 지칭됨)는 하나 이상의 클라이언트 머신(104)과 연관될 수 있다. 따라서, 도 1에 도시된 클라이언트 머신(104(1)-(N))은, 비디오 게임과 같은 프로그램을 실행하기 위해 사용자 커뮤니티(또는 고객 기반)에 의해 활용될 수 있는 컴퓨팅 디바이스를 나타낸다. 클라이언트 머신(104)은, PC, 데스크톱 컴퓨터, 랩톱 컴퓨터, 모바일 폰(예를 들어, 스마트 폰), 태블릿 컴퓨터, 휴대용 디지털 단말기(PDA), 웨어러블 컴퓨터(예를 들어, 가상 현실(VR: virtual reality) 헤드셋, 증강 현실(AR: augmented reality) 헤드셋, 스마트 안경 등), 차량 내(예를 들어 자동차 내) 컴퓨터, 텔레비전(스마트 텔레비전), 셋톱 박스(STB: set-top-box), 게임 콘솔, 및/또는 임의의 유사한 컴퓨팅 디바이스를, 한정됨 없이, 포함하는, 연관된 디스플레이 상에 그래픽을 처리하여 렌더링하도록, 그리고 네트워크(들)를 통해 데이터를 송신/수신하도록 구성된 임의의 적합한 유형의 컴퓨팅 디바이스로서 구현될 수 있다.

[0015] 클라이언트 머신(104)의 컨피규레이션(configuration)은 다양할 수 있다. 예를 들어, 클라이언트 머신(104)의 서브세트는 각각, 특정 유형, 버전 또는 특성의 하드웨어(예를 들어, 중앙 처리 장치(CPU: central processing unit) 모델, 그래픽 처리 장치(GPU: graphics processing unit) 모델 등) 및/또는 특정 유형, 버전 또는 특성의 펌웨어 및/또는 소프트웨어(예를 들어, 일 버전의 그래픽 드라이버, 인스톨 스크립트에 사용되는 다운로드 가능한 콘텐츠(DLC: downloadable content) 패키지, 비디오 게임 클라이언트가 런닝되는 언어 등)를 사용할 수 있다. 클라이언트 머신(104)의 하드웨어, 펌웨어, 및/또는 소프트웨어의 이러한 양태 및 다른 양태는, 본원에서 사용되는 바와 같은, 클라이언트 머신(104)의 "컨피규레이션"(이는 때때로 "클라이언트 시스템 컨피규레이션"으로 지칭됨)을 구성하고, 이들은 비디오 게임 배포 플랫폼 상에 있는 비디오 게임을 위한 게임 데이터(데이터의 블록으로 분할됨)를 다운로드하기 위해서 사용되는 유한한 세트의 데포(depot)을 결정할 수 있다. 따라서, 클라이언트 머신(104)의 서브세트들은 공통 클라이언트 시스템 컨피규레이션을 공유할 수 있고, 클라이언트 시스템 컨피규레이션은 클라이언트 머신(104)의 이러한 서브세트들 간에 상이할 수 있다. 클라이언트 시스템 컨피규레이션의 관점에서 상이한 클라이언트 머신들(104)은 동일한 비디오 게임에 대해서도 게임 데이터의 블록들을 상이하게 다운로드, 저장 및/또는 액세스할 수 있다. 한 쌍의 클라이언트 머신(104)이 공통의 클라이언트 시스템 컨피규레이션을 갖는지 여부를 결정하는 것은, 머신들(104)이 임계 수의 공통의 유형, 버전 또는 특성의 하드웨어, 소프트웨어 또는 펌웨어를 공유하는 것에 기초할 수 있다. 예를 들어, 만약 2개의 클라이언트 머신(104)이 인스톨 스크립트에 대해 적어도 동일한 DLC 패키지를 사용하는 경우, 다른 양태에서 몇 가지 차이점에도 불구하고(예를 들어, 상이한 GPU 모델 등), 동일한 클라이언트 시스템 컨피규레이션을 갖는 것으로 간주될 수 있다.

[0016] 다시 도 1을 참조하면, 클라이언트 머신(104)은 컴퓨터 네트워크(108)를 통해 원격 컴퓨팅 시스템(106)(때때로, "원격 시스템(106)"으로 축약됨)과 통신할 수 있다. 컴퓨터 네트워크(108)는 인터넷, 다른 유형의 데이터 및/또는 음성 네트워크, 유선 인프라(예를 들어, 동축 케이블, 광섬유 케이블 등), 무선 인프라(예를 들어, 무선 주파수(RF), 셀룰러, 위성 등) 및/또는 기타 연결 기술을, 한정됨 없이, 나타내고/내거나 포함할 수 있다. 원격 시스템(106)은, 일부 예에서, 컴퓨터 네트워크(108)를 통해 유지되고 액세스 가능한 네트워크 액세스 가능 컴퓨

팅 플랫폼의 일부일 수 있다. 이와 같은 네트워크 액세스 가능 컴퓨팅 플랫폼은 "주문형 컴퓨팅", "서비스로서의 소프트웨어(SaaS: software as a service)", "플랫폼 컴퓨팅", "네트워크 액세스 가능 플랫폼", "클라우드 서비스", "데이터 센터" 등과 같은 용어를 사용하여 지칭될 수 있다. 일반적으로, 원격 시스템(106)은 클라이언트 머신(104)으로부터 액세스 데이터(110)를 수집하도록 구성되고, 수신하는 액세스 데이터(110)를 데이터 저장소(112) 내에서 카탈로그화(예를 들어, 조직화, 범주화, 분류 등)하도록 구성된다. 원격 시스템(106)은 또한, 본원에 설명된 다양한 게임 관련 특징부를 구현하기 위해 클라이언트 머신(104)에 의해서 사용 가능한 데이터를 생성하도록 액세스 데이터(110)를 분석하게끔 구성될 수 있다. 예를 들어, 원격 시스템(106)은 다운로드 시퀀스 데이터(114) 및/또는 블록 종속성 데이터(116)를 생성하도록 구성될 수 있고, 원격 시스템(106)은, 본원에 설명된 바와 같이, 이 데이터를 클라이언트 머신(104)에 배포할 수 있다. 원격 시스템(106)은, 추가적으로 또는 대안적으로, 하나 이상의 클라이언트 머신(104) 상에 있는 게임 데이터의 미사용 블록을 결정하기 위해 액세스 데이터(110)를 분석하도록 구성될 수 있고, 클라이언트 머신(들)(104) 상의 로컬 메모리 리소스를 확보하기 위해 게임 데이터의 미사용 블록을 삭제하라는 명령어(118)를 클라이언트 머신(들)(104)에 보낼 수 있다.

[0017] 일부 실시형태에서, 원격 시스템(106)은 클라이언트 머신(104)에 프로그램(및 데이터)을 배포(예를 들어, 다운로드)하기 위한 배포 서비스부로서 작용하거나 이에 대한 액세스를 갖는다. 일 예에서, 클라이언트 머신(104)은 클라이언트 애플리케이션을 클라이언트 머신 상에 인스톨할 수 있다. 비디오 게임 클라이언트일 수 있는 클라이언트 애플리케이션(예를 들어, 비디오 게임을 플레이하기 위한 게임 소프트웨어)은, 클라이언트 애플리케이션이 인스톨되는 클라이언트 머신(104) 상에서 비디오 게임과 같은 프로그램을 실행하도록 구성될 수 있다. 클라이언트 애플리케이션이 인스톨된 상태에서, 클라이언트 머신(104)은 원격 시스템(106)으로부터 컴퓨터 네트워크(108)를 통해 프로그램(예를 들어, 비디오 게임)을 다운로드하는 능력을 가질 수 있다. 프로그램(예를 들어, 비디오 게임)이 클라이언트 머신(104)에 다운로드 및 실행되기 위해 개별적으로 구매될 수 있는 직접 구매 모델, 구독 기반 모델, 프로그램이 일정 기간 동안 임대 또는 리스되는 콘텐츠 배포 모델과 같은 임의의 유형의 콘텐츠 배포 모델이 이러한 목적을 위해 활용될 수 있다. 따라서, 개별 클라이언트 머신(104)은, 클라이언트 애플리케이션을 로딩함으로써 실행 가능한 하나 이상의 인스톨된 비디오 게임을 포함할 수 있고, 이러한 비디오 게임은 실행 중에 디스플레이 상에 그래픽을 렌더링할 수 있다. 일 예에서, 사용자(102)는, 예컨대, 비디오 게임의 실행을 시작하기 위해 비디오 게임 클라이언트를 로딩하고 원하는 비디오 게임을 선택함으로써, 사용자가 원격 시스템(106)으로부터 획득(예를 들어, 구매, 임대, 리스 등)하고 다운로드한 다수의 비디오 게임 중 하나를 플레이하도록 선택할 수 있다.

[0018] 사용자들(102) 중 일부가 원격 시스템(106)에 의해 이 사용자들(102)에게 사용 가능하게 된 비디오 게임을 플레이하는 도 1의 실시예를 고려한다. 비디오 게임이 사용자들(102)의 각각의 클라이언트 머신(104)에서 플레이될 때, 각각의 클라이언트 머신(104) 상에서 런닝되는 게임 실행 파일은, 비디오 게임을 위한 게임 데이터의 블록을 판독할 것을 파일 시스템에 계속해서 요청할 수 있다. 게임 실행 파일에 의해 행해진 판독 작동은 액세스 데이터(110)를 생성하기 위해 클라이언트 머신(104)의 파일 시스템 프록시 컴포넌트에 의해 추적될 수 있다. 예를 들어, 클라이언트 머신(104(1)) 상에서 런닝되는 게임 실행 파일은, 예컨대, 하드 디스크 드라이브(HDD: hard disk drive)의 섹터로부터, 또는 클라이언트 머신(104(1))에 제거 가능하게 결합된 시큐어 디지털(SD: secure digital) 카드의 섹터로부터 게임 데이터의 블록을 판독함으로써, 게임 실행 동안, 클라이언트 머신(104(1))의 하나 이상의 로컬 메모리 리소스로부터 게임 데이터의 블록을 판독할 것을 클라이언트 머신(104(1))의 파일 시스템에 요청할 수 있다. 따라서, 클라이언트 머신(104(1)) 상에서 실행되는 비디오 게임을 위한 게임 실행 파일은, 게임 세션 전체에 걸쳐 파일 시스템에 대해 일련의 판독 작동을 수행할 수 있고, 클라이언트 머신(104(1))의 파일 시스템 프록시 컴포넌트는 액세스 데이터(110)를 생성하기 위해 판독 작동을 추적할 수 있다.

[0019] 액세스 데이터(110)는 (i) 비디오 게임을 위한 게임 데이터의 액세스된 블록(예를 들어, 게임 세션 동안 게임 실행 파일에 의해서 액세스된 블록)을 식별하는 블록 식별자뿐만 아니라 (ii) 게임 데이터의 액세스된 블록이 판독 작동을 수행하는 게임 실행 파일에 의해서 액세스되었던, 게임 실행 파일의 실행 중의 시간을 상술할 수 있다. 언급된 바와 같이, 비디오 게임을 위한 게임 데이터는 텍스트, 가상 객체, 지도, 게임 캐릭터 등을 포함할 수 있다. 일부 실시형태에서, 게임 데이터는 비휘발성 메모리의 섹터(예를 들어, HDD, SD 카드 등의 섹터)에 걸쳐 저장될 수 있고, 섹터 내의 게임 데이터의 블록으로 조직화될 수 있다. 블록을 사용하는 것은, 파일 시스템의 연속 저장 공간을 사용하여 모든 파일들을 저장할 필요를 피하면서, 다양한 크기의 파일들을 처리하는 유연한 방식일 수 있다. 게임 데이터의 각각의 블록은 블록 식별자(예를 들어, 숫자)를 사용하여 참조되고/되거나 위치될 수 있다. 일반적으로, 클라이언트 머신(104)의 하나 이상의 프로세서는 하나 이상의 장치 드라이버에 의해 지시되는 바와 같이 판독, 쓰기, 및/또는 다른 저장 작동을 수행할 수 있다. 일부 실시형태에서, 클라이언트 머신(104)은, 게임플레이 동안 게임 실행 파일에 의해 행해진 판독 작동에 기초하여 어느 블록들이 액세스되었

고 이들이 어디(예를 들어, 어떤 섹터)로부터 액세스되었는지를 결정하기 위해 게임 데이터 블록과 비휘발성 메모리의 섹터 사이의 매핑을 생성할 수 있다. 예를 들어, 게임 실행 파일이 비휘발성 메모리의 제1 섹터에 저장된 게임 데이터에 액세스하도록 판독 작동을 수행하는 경우, 제1 섹터에 저장되는 게임 데이터의 특정 블록(들)의 식별자(들)는 액세스 데이터(110)에 상술될 수 있다. 또한, 액세스 데이터(110)에 상술된 액세스 시간은, 블록들이 게임 세션 동안 액세스된 순서(또는 시퀀스)(예를 들어, 블록 A가 먼저 액세스되었고, 이어서 블록 D가 액세스되었고, 이어서 블록 F가 액세스되었고, 등등) 뿐만 아니라 액세스의 상대적 시간(예를 들어, 게임 세션 안으로 4분, 게임 세션 안으로 9분, 게임 세션 안으로 1시간, 등등)을 결정하는 것을 허용할 수 있다.

[0020] 원격 시스템(106)에 의해 수신된 액세스 데이터(110)는, 고유한 클라이언트 시스템 컨피규레이션에 따라 그리고/또는 대응되는 비디오 게임의 비디오 게임 ID에 따라 데이터 저장소(112)에 카탈로그화될 수 있다. 액세스 데이터(110)는, 액세스 데이터(110)를 전송한 클라이언트 머신(104) 상에서 실행되는 비디오 게임 클라이언트에 로그인한 사용자의 사용자 계정과 관련하여 추가로 저장될 수 있다. 도 1은, 클라이언트 머신(104)으로부터 수신하는 액세스 데이터(110)를 저장하거나, 카탈로그화하거나, 또는 달리 조직화하기 위해 원격 시스템(106)에 의해 유지되는 데이터 저장소(112)를 도시한다. 데이터 저장소(112)는, 클라이언트 시스템 컨피규레이션 및 비디오 게임 ID의 고유한 조합과 각각 연관된 그룹들(또는 버킷들) 안으로 액세스 데이터(110)를 조직화할 수 있다. 다시 말해서, 데이터 저장소(112)에 있는 액세스 데이터(110)의 각각의 버킷은 본원에 설명된 바와 같이, 특정 프로그램(예를 들어, 비디오 게임) 및 특정 클라이언트 시스템 컨피규레이션에 결부(tie)될 수 있다.

[0021] 도 1은, 클라이언트 머신(104(N))이 원격 시스템(106)으로부터 비디오 게임을 획득(예를 들어, 구매, 임대, 리스 등)하기 위한 요청(120)을 전송하는 것을 도시한다. 예를 들어, 인스톨된 비디오 게임 클라이언트를 통해 자신의 사용자 계정에 로그인된 클라이언트 머신(104(N))의 사용자는 비디오 게임을 구매하기 위해 원격 시스템(106)을 통해 거래를 실시할 수 있다. 요청(120)에 응답하여, 클라이언트 머신(104(N))은 원격 시스템(106)으로부터 비디오 게임의 게임 실행 파일(122)을 수신할 수 있고, 클라이언트 머신(104)은 또한, 획득된 비디오 게임에 대해서, 그리고 클라이언트 머신(104)의 특정 클라이언트 시스템 컨피규레이션에 대해서 다운로드 시퀀스 데이터(114) 및/또는 블록 종속성 데이터(116)를 수신할 수 있다. 따라서, 요청(120)이 클라이언트 머신(104(N))의 컨피규레이션을 포함할 수 있으며, 이는 원격 시스템(106)에, 특정 클라이언트 시스템 컨피규레이션에 대해 이용 가능할 수 있는 블록 종속성 데이터(116) 및 다운로드 시퀀스 데이터(114)를 찾도록 지시한다는 것이 이해될 것이다.

[0022] 도 1은 또한, 원격 시스템(106)이 다운로드 시퀀스 데이터(114)에 상술된 블록들의 시퀀스에 따라, 획득된 비디오 게임을 위한 게임 데이터(126)의 블록들(124)을 다운로드하기 시작하는 것을 도시한다. 도 1은, 제1 블록(124(1))이 다운로드되고, 이어서 제2 블록(124(2))이 다운로드되고, 이어서 제3 블록(124(3))이 다운로드되는 등의 예를 도시한다. 3개의 블록(124)이 클라이언트 머신(104(N))에 다운로드되는 것으로서 도 1에 도시되지만, 다운로드 시퀀스가, 블록(124(3)) 이후에 다운로드되는 추가 블록을 포함하여, 임의의 수의 블록(124)을 포함할 수 있다는 것이 이해될 것이다. 클라이언트 머신(104(N))은 블록(124)을 HDD, SD 카드 등과 같은 클라이언트 머신(104(N))의 비휘발성 메모리에 다운로드할 수 있다.

[0023] 도 1에 도시된 바와 같이, 클라이언트 머신(104(N))은 인스턴트 플레이(128) 기능을 구현할 수 있으며, 이 기능으로, 클라이언트 머신(104(N))은 게임 데이터(126)의 블록(124)의 다운로드 전 또는 다운로드 동안, 획득된 비디오 게임에 대한 게임 실행 파일(122)의 실행을 시작할 수 있다. 예를 들어, 클라이언트 머신(104(N))은, 게임 데이터의 제1 블록(124(1))이 다운로드되기 전에 게임 실행 파일(122)의 실행을 시작할 수 있다. 게임 실행 파일(122)은, 사용자(102)가 새로 획득된 비디오 게임을 론치(launch)하기 위해 마우스 및/또는 키보드, 게임 컨트롤러 등을 사용하여 사용자 입력을 제공하는 바와 같은, 클라이언트 머신(104(N))에 의해 수신된 사용자 입력에 응답하여 시작될 수 있다. 이러한 의미에서, 사용자(102)가 비디오 게임 플레이를 시작할 수 있는 시기에 관해 사용자(102)에게 주어진 제한이 없을 수 있다. 결과적으로, 사용자(102)는 게임의 플레이를 시작할 수 있고, 게임 실행 파일(122)은, 게임 데이터의 제1 블록(124(1))이 클라이언트 머신(104(N))으로 다운로드되기 전에, 실행을 시작할 수 있다. 사용자(102)가 비디오 게임을 획득한 후 일정 기간을 기다리기로 선택하는 경우, 게임 실행 파일(122)은, 게임 데이터(126)의 적어도 하나의 블록(124(1))이 클라이언트 머신(104(N))으로 다운로드된 후에 실행을 시작할 수 있다.

[0024] 일부 실시형태에서, 클라이언트 머신(104(N)) 상에서 런닝되는 비디오 게임 클라이언트는, 블록(124)의 다운로드를 시작한 후 미리 결정된 시간이 경과할 때까지, 또는 미리 결정된 이벤트가 발생될 때까지(예를 들어, 사용자(102)가 게임의 플레이를 시작하는 것을 허용하기 전에 임계 수의 블록들(124)이 다운로드될 때까지 대기함으

로써), 게임 실행 파일(122)이 시작되는 것을 방지하도록 구성될 수 있다. 미리 결정된 시간 또는 이벤트는, 비디오 게임이 인스턴트 플레이(128) 기능에 얼마나 적합한지에 기초하여 원격 시스템(106)에 의해 결정될 수 있고, 원격 시스템(106)은, 비디오 게임이 획득되면, 미리 결정된 시간이 경과할 때까지 기다리라는, 그리고/또는 게임 실행 파일(122)이 클라이언트 머신(104(N)) 상에서 런닝되는 것을 허용하기 전에 미리 결정된 이벤트의 발생을 기다리라는 지시를 비디오 게임 클라이언트에게 보낼 수 있다. 일부 실시형태에서, 원격 시스템(106)은, 예컨대, "최선의 사용자 경험을 위해서, 다운로드를 시작한 후 5분 정도를 [비디오 게임 X]를 플레이하기 위해 기다리는 것이 권장됩니다(For the best user experience, we recommend waiting 5 minutes after starting the download to play [Video Game X])"라는 권장 사항을 표시함으로써, 클라이언트 머신(104(N))을 통해 사용자(102)에게 권장 사항을 출력하기 위해 클라이언트 머신(104(N))에 데이터를 보낼 수 있다. 일 실시예에서, 비디오 게임이 인스턴트 플레이(128) 기능에 적합하다면, 원격 시스템(106)은 클라이언트 머신(104(N))에게, "이 게임은 인스턴트 플레이를 위한 준비가 되어 있으므로, 지금 바로 플레이를 시작할 수 있습니다. 즐거운 시간 되세요!(This game is ready for instant play, so you can start playing right now. Enjoy!)"라는 알림을 출력하도록 지시할 수 있다.

[0025] 도 1에 도시된 바와 같이, 다른 클라이언트 머신(104(2))은, 클라이언트 머신(104(2))이 게임플레이 동안의 레이턴시를 감소시키기 위해 게임 데이터(126)의 하나 이상의 실행된 블록(124)을 프리페치(prefetch)할 수 있는 로컬 프리페칭(local prefetching)(130) 특징을 구현할 수 있다. 예를 들어, 비디오 게임을 위한 게임 데이터(126)는 제1 속도로 판독 액세스를 제공하는 제1 메모리(132(1))에 저장될 수 있다. 이 제1 메모리(132(1))는, 게임 데이터(126)가 지속되는 HDD, SD 카드 등과 같은 비휘발성 메모리를 나타낼 수 있다. 클라이언트 머신(104(2))은 또한, 제1 속도보다 더 빠른 제2 속도로 판독 액세스를 제공하는 제2 메모리(132(2))를 포함할 수 있다. 이러한 제2 메모리(132(2))는 추가의 비휘발성 메모리(예를 들어, SSD)이거나, 또는 휘발성 메모리(예를 들어, RAM과 같은 작업 메모리)일 수 있다. 어느 경우든, 블록 종속성 데이터(116)는, 블록(들)(124)이 게임 실행 파일(122)에 의해 다음으로 판독될 가능성이 높은지 여부를 결정하기 위해 클라이언트 머신(104(2))에 의해 사용될 수 있고, 만약 그렇다면, 블록(들)(124)은 제2 메모리(132(2))에 캐시(cache)될 수 있어, 게임 실행 파일(122)이 결국 블록(들)(124)을 판독할 것을 요청할 때, 상대적으로 느린 제1 메모리(132(1))로부터 블록(들)(124)에 액세스하기 보다는, 블록(들)(124)이 제2 메모리(132(2))로부터 빠르게 액세스될 수 있다. 아래에서 더 상세히 설명되는 바와 같이, 다수의 상이한 로컬 메모리 리소스(132)는 로컬 프리페칭(130) 특징의 일부로서 게임 데이터(126)의 블록(124)을 캐시하기 위해 활용될 수 있으며, 이는 전체 대역폭을 개선하고, 예를 들어, 작업 메모리에 배타적으로 블록(124)을 캐싱하는 것보다 훨씬 더 레이턴시를 감소시킬 수 있다. 이것은, 전체 대역폭을 개선하고 레이턴시를 훨씬 더 많이 감소시키기 위해서, 게임 실행 파일(122)이, 제1 메모리(132(1))를 포함하는 상이한 로컬 저장 리소스(132)로부터 병렬로 판독될 수 있다는 개념에 기초한다.

[0026] 도 1에 도시된 바와 같이, 또 다른 클라이언트 머신(104(1))은 미사용 게임 데이터 삭제(134) 기능을 구현할 수 있으며, 이 기능에 의해서, 클라이언트 머신(104(1))은 클라이언트 머신(104(1))의 비휘발성 메모리에 저장된 게임 데이터(126)의 하나 이상의 블록(124)을 삭제하라는 명령어(118)를 원격 시스템(106)으로부터 수신하고, 응답으로, 클라이언트 머신(104(1))은 클라이언트 머신(104(1)) 상의 로컬 메모리를 확보하기 위해 하나 이상의 블록(124)을 삭제할 수 있다. 예를 들어, 원격 시스템(106)은, 클라이언트 머신(104(1))으로부터 수신된 액세스 데이터(110)로부터, 클라이언트 머신(104(1))의 사용자(102)가 비디오 게임의 싱글 플레이어 모드를 플레이한 적이 없고, 결과적으로, 싱글 플레이어 모드로 비디오 게임을 플레이하기 위해서 사용 가능한 비디오 게임을 위한 게임 데이터(126)의 블록(124)이 임계 시간 기간 또는 임계 수의 게임 세션 동안 게임 실행 파일(122)에 의해 액세스되지 않았다고 결정할 수 있다. 따라서, 원격 시스템(106)은, 이전에 클라이언트 머신(들)(104)에 다운로드되었고 "미사용" 블록(124)으로서 분류 가능한 것으로 결정된 게임 데이터(126)의 하나 이상의 미사용 블록(124)을 클라이언트 머신(들) 상의 비휘발성 메모리로부터 삭제하도록 클라이언트 머신(들)(104)에 지시하는 명령어(118)를 클라이언트 머신(104(1))과 같은 하나 이상의 클라이언트 머신(104)에 전송할 수 있다. 명령어(118)는, 클라이언트 머신(104(1))이 게임 데이터(126)의 정확한 블록(124)을 삭제하여 클라이언트 머신(104(1)) 상에서 게임을 플레이하는 사용자 경험을 손상시키지 않으면서 로컬 메모리 용량을 확보할 수 있도록 미사용 블록(124)의 식별자를 상술할 수 있다.

[0027] 도 2는, 액세스 데이터(110)를 생성하도록 파일 시스템 판독 작동을 추적하기 위한, 그리고 액세스 데이터(110)를 원격 시스템(106)에 전송하기 위한, 예시적인 프로세스(200)의 흐름도 및 클라이언트 머신(104)의 예시적인 컴포넌트를 예시하는 블록도를 도시한다. 예시된 구현예에서, 클라이언트 머신(104)은, 다른 컴포넌트 중에서, 중앙 처리 유닛(들)(CPU(들)) 및 그래픽 처리 유닛(들)(GPU(들)) 등과 같은 하나 이상의 프로세서(202), 하나 이상의 입력 디바이스(204), 하나 이상의 출력 디바이스(206), 비일시적 컴퓨터 판독 가능 매체(208), 로컬

메모리(132), 및 통신 인터페이스(들)(210)를 포함한다.

[0028] 비밀시적 컴퓨터 판독 가능 매체(208)는, 컴퓨터 판독 가능 명령어, 데이터 구조, 프로그램 모듈 또는 다른 로직 및/또는 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 착탈식 및 비착탈식 매체, 휘발성 및 비휘발성 메모리를 포함할 수 있다. 이러한 메모리는, RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, 디지털 다목적 디스크(DVD: digital versatile disk) 또는 기타 광 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 디바이스, 또는 기타 자기 저장 디바이스, RAID 저장 시스템, 또는 원하는 정보를 저장하는 데 사용될 수 있으며 컴퓨팅 디바이스에 의해서 액세스될 수 있는 임의의 다른 매체를 포함하나, 이에 한정되지 않는다. 컴퓨터 판독 가능 매체(208)는 컴퓨터 판독 가능 저장 매체(CRSM: computer-readable storage media)로서 구현될 수 있으며, 이는 컴퓨터 판독 가능 매체(208) 상에 저장된 명령어를 실행하기 위해 프로세서(들)(202)에 의해 액세스 가능한 임의의 사용 가능한 물리적 매체일 수 있다. 하나의 기본 구현예에서, CRSM은 랜덤 액세스 메모리(RAM: random access memory) 및 플래시 메모리를 포함할 수 있다. 다른 구현예에서, CRSM은, 읽기 전용 메모리(ROM: read-only memory), 전기적으로 지울 수 있는 프로그래밍 가능한 읽기 전용 메모리(EEPROM: electrically erasable programmable read-only memory), 또는 원하는 정보를 저장하기 위해서 사용될 수 있고, 프로세서(들)(202)에 의해서 액세스될 수 있는 임의의 다른 유형(tangible)의 매체를 포함할 수 있으나, 이에 한정되지 않는다. 또한, 로컬 메모리(132)가 컴퓨터 판독 가능 매체(208)와 별개인 것으로 도시되지만, 컴퓨터 판독 가능 매체(208) 및 로컬 메모리(132(1), 132(2) 및/또는 132(3)) 중 임의의 하나 이상은, 일부 구현예에서, 동일한 메모리, 또는 동일한 메모리의 적어도 부분을 나타낼 수 있다.

[0029] 이제 도 2에 도시된 프로세스(200)가 참조된다. 본원에 설명된 프로세스는 하드웨어, 소프트웨어, 펌웨어 또는 이들의 조합(때때로 본원에서 "로직"으로 지칭됨)으로 구현될 수 있는 일련의 작동을 나타내는 논리 흐름 그래프의 블록 집합으로서 예시된다. 소프트웨어의 맥락에서, 블록은, 하나 이상의 프로세서에 의해 실행될 때 설명된 작동을 수행하는 컴퓨터 실행 가능 명령어를 나타낸다. 일반적으로, 컴퓨터 실행 가능 명령어는, 특정 기능들을 수행하거나 특정된 추상적 데이터 유형을 구현하는 루틴, 프로그램, 오브젝트, 구성요소, 데이터 구조 등을 포함한다. 작동이 설명되는 순서는 제한으로서 해석되게 하려는 것이 아니고, 임의의 수의 설명된 블록들은 임의의 순서로 그리고/또는 병렬로 결합되어서 프로세스를 구현할 수 있다.

[0030] 논의의 목적을 위해, 프로세스(200)를 수행하는 클라이언트 머신(104)이 비디오 게임 클라이언트(212)를 이전에 인스톨했다고 가정하며, 이 비디오 게임 클라이언트는 컴퓨터 판독 가능 매체(208)에 저장되고 비디오 게임을 위한 게임 실행 파일(122)을 실행하도록 구성된 클라이언트 애플리케이션을 나타낸다. 클라이언트 머신(104)의 사용자(102)는 비디오 게임을 획득(예를 들어, 구매, 임대, 리스 등)할 수 있으며, 이 비디오 게임은, 획득되면, 비휘발성 메모리에 인스톨(예를 들어, 원격 시스템(106)로부터 다운로드)되고 유지될 수 있다. 일부 실시형태에서, 제1 로컬 메모리(132(1))(때때로 "제1 메모리(132(1))"로 지칭됨)는 비휘발성 메모리(예를 들어, HDD, SD 카드 등)를 나타낼 수 있고, 제1 메모리(132(1))는 제1 속도로 판독 액세스를 제공한다. 게임 데이터(126)의 블록(124)이 원격 시스템(106)으로부터 다운로드될 때, 게임 데이터(126)의 블록(124)은 제1 메모리(132(1))로 다운로드될 수 있다. 한편, 제2 메모리(132(2))는 추가적인 비휘발성 메모리(예를 들어, 솔리드 스테이트 드라이브(SSD: solid-state drive))를 나타낼 수 있고, 제2 메모리(132(2))는 제1 메모리(132(1))에 의해 제공되는 제1 속도보다 더 빠른 제2 속도로 판독 액세스를 제공할 수 있다. 추가로, 제3 메모리(132(3))는 휘발성 메모리(예를 들어, RAM과 같은 작업 메모리)를 나타낼 수 있고, 제3 메모리(132(3))는 제2 속도보다 더 빠른 제3 속도로 판독 액세스를 제공할 수 있다. 클라이언트 머신(104)이 (예를 들어, 제2 메모리(132(2))를 생략함으로써) 도 2에 도시된 것보다 더 적은 로컬 메모리(132) 리소스, 또는 추가적인 로컬 메모리(132) 리소스를 구현할 수 있다는 것이 이해될 것이다.

[0031] 214에서, 비디오 게임 클라이언트(212)는 클라이언트 머신(104) 상에서 비디오 게임의 게임 실행 파일(122)을 실행할 수 있다. 예를 들어, 클라이언트 머신(104)의 사용자(102)는 비디오 게임 클라이언트(212)를 로드할 수 있고, 로드된 비디오 게임 클라이언트(212)는 사용자(102)에게 (게임 실행 파일(122)의 실행을 통해) 이전에 다운로드된 비디오 게임을 실행하고/하거나, 원격 시스템(106)으로부터 새로운 비디오 게임을 획득할 수 있는 능력을 제공할 수 있다. 게임 실행 파일(122)은 예컨대, 제3 메모리(132(3))와 같은 작업 메모리에 로드될 수 있으며, 여기서, 게임 실행 파일은, 무엇보다도, 클라이언트 머신(104)의 디스플레이 상에 그래픽을 렌더링하기 위해 실행된다. 게임 세션 중에 스타트업 단계와 플레이타임 단계가 있을 수 있다. 게임 세션 동안, 게임 실행 파일(122)은 입력 디바이스(들)(204)(예를 들어, 마우스 및/또는 키보드, 게임 컨트롤러, 헤드 마운트 디스플레이(HMD: head-mounted display), 마이크(들) 등)로부터 입력 데이터를 수신하도록 구성될 수 있고, 클라이언트 머신(104)의 디스플레이(들)(즉, 출력 디바이스(들)(206)) 상에 비디오 게임 콘텐츠의 다음 프레임을 렌더링하

기 위해 액세스할 게임 데이터(126)의 블록(124)을 결정할 수 있다. 예를 들어, 게임 실행 파일(122)은, 게임 세계의 어느 부분이 다가오는 프레임에서 렌더링될지 뿐만 아니라, 어느 객체 및/또는 텍스처가 다가오는 프레임에서 렌더링될지 결정하도록 구성될 수 있고, 다가오는 프레임을 제시하기 위해 게임 데이터(126)의 대응되는 블록(124)을 판독하도록 판독 작동을 이슈할 수 있다.

[0032] 216에서, 클라이언트 머신(104) 상에서 실행되는 파일 시스템 프록시 컴포넌트(218)는, 게임 실행 파일(122)에 의해 행해지는 클라이언트 머신(104)의 파일 시스템(220)에 대한 판독 작동(예를 들어, 제1 판독 작동, 제2 판독 작동 등)을 결정(예를 들어, 수신, 모니터링, 인터셉트 등)할 수 있다. 파일 시스템(220)은 비디오 게임을 위한 게임 데이터(126)의 블록(124)을 포함하는 데이터가 저장되고 검색되는 방법을 제어하도록 구성될 수 있다. 비디오 게임을 위한 게임 데이터(126) 모두가 제1 메모리(132(1))와 같은 로컬 메모리 리소스에서 실현되는지 여부에 상관없이, 파일 시스템 프록시 컴포넌트(218)는, 게임 데이터(126)의 어느 블록(124)(예를 들어, 파일)이 클라이언트 머신(104)의 비휘발성 메모리 상에(예를 들어, HDD, SD 카드 상에) 존재하는지에 관해 비디오 게임에 "라이(lie)"하도록 구성될 수 있다. 예를 들어, 비디오 게임 클라이언트(212)는, 파일 시스템 프록시 컴포넌트(218)를 통해, 비디오 게임의 식별자에 기초하여, 비휘발성 메모리에 저장되기로 되어 있는 게임 파일의 목록, 게임 파일의 크기, 및 가능하게는 게임 파일이 저장될 비휘발성 메모리의 섹터를 알 수 있다. 이것은 비디오 게임(예를 들어, 게임 실행 파일(122))이, 예를 들어, 비디오 게임을 위한 게임 데이터(126)의 모든 블록(124)이 제1 메모리(132(1))에 저장된다고 생각하게 만든다. 파일 시스템 프록시 컴포넌트(218)는, 필요한 경우, 이 정보를 가공하고 게임 실행 파일(122)에 표시하기 위한 파일 시스템(220)의 확장일 수 있다. 어느 경우든, 비디오 게임을 위한 게임 데이터(126)의 모든 블록(124)이 클라이언트 머신(104)의 제1 메모리(132(1))에 저장되고 이로부터 액세스 가능한 실시예를 고려한다. 이 실시예에서, 파일 시스템 프록시 컴포넌트(218)는, 게임 실행 파일(122)에 의해 행해지는 파일 시스템(220)에 대한 판독 작동을 단순히 모니터링하는 패스쓰루(pass-through)로서 작용할 수 있다. 본원에서 더 자세히 설명되는 바와 같이, 게임 데이터(126)의 특정 블록(124)은 제1 메모리(132(1))로부터 프리페치되고, 제2 메모리(132(2)) 또는 제3 메모리(132(3)) 중 적어도 하나에 캐시될 수 있으며, 제2 및 제3 메모리 모두는 제1 메모리(132(1))에 의해 제공되는 판독 액세스 속도보다 더 빠른 판독 액세스 속도를 제공한다. 파일 시스템(220)은 게임 데이터(126)의 블록(124)이 임의의 주어진 순간에 저장되는 위치를 추적할 수 있고, 게임 세션 동안 게임 실행 파일(122)에 의해 수행되는 판독 작동을 제공하도록 적절한 메모리 리소스(132)로부터 게임 데이터(126)의 블록(124)에 액세스할 수 있다.

[0033] 222에서, 파일 시스템 프록시 컴포넌트(218)는 게임 실행 파일(122)로부터 수신하는 판독 작동에 적어도 부분적으로 기초하여 액세스 데이터(110)를 생성할 수 있다. 본원의 다른 곳에서 설명된 바와 같이, 이러한 액세스 데이터(110)는 (i) 게임 세션 동안 게임 실행 파일(122)에 의해 액세스되었던 게임 데이터(126)의 블록(124)의 식별자, 및 (ii) 게임 실행 파일(122)의 실행 동안, 게임 데이터(126)의 액세스된 블록(124)이 게임 실행 파일(122)에 의해 액세스되었던 시간을 상술할 수 있다. 예를 들어, 게임 데이터(126)의 2개의 블록(124)과 관련하여, 액세스 데이터(110)는 (i) 게임 데이터의 제1 블록의 제1 식별자, (ii) 게임 실행 파일의 실행 동안, 게임 데이터의 제1 블록이 제1 판독 작동에 적어도 부분적으로 기초하여 액세스되었던 제1 시간, (iii) 게임 데이터의 제2 블록의 제2 식별자, 및 (iv) 게임 실행 파일의 실행 동안, 게임 데이터의 제2 블록이 제2 판독 작동에 적어도 부분적으로 기초하여 액세스되었던 제2 시간을 상술할 수 있다. 일부 실시형태에서, 액세스된 각각의 블록(124)에 대한 액세스 시간은, 게임 세션의 시작으로부터 측정된 시간으로서 액세스 데이터(110)에 표현될 수 있다(예를 들어, 블록 A는 게임 세션 시작 후 4분에 액세스되었고, 블록 D는 게임 세션 시작 후 13분에 액세스되었고, 등등).

[0034] 224에서, 클라이언트 머신(104)은 액세스 데이터(110)를, 예컨대, 통신 인터페이스(들)(210)를 통해 그리고 컴퓨터 네트워크(108)를 거쳐, 원격 시스템(106)에 전송할 수 있다. 통신 인터페이스(들)(210)는 다수의 유형의 유선 및/또는 무선 또는 라디오(radio) 기술을 구현할 수 있다. 예를 들어, 통신 인터페이스(들)(210)는 저전력 블루투스(BLE: bluetooth low energy) 라디오, Wi-Fi 라디오, 및/또는 셀룰러 라디오 등과 같은 라디오를 구현할 수 있다. 통신 인터페이스(들)(210)가, 네트워크, 연결된 주변 디바이스, 또는 다른 무선 네트워크와 통신하는 플러그인 네트워크 디바이스에 대한 유선 연결을 용이하게 하는 물리적 포트를 더 포함할 수 있다는 것이 이해될 것이다.

[0035] 서브블록 226에 도시된 바와 같이, 액세스 데이터(110)는 224에서, 클라이언트 머신(104)의 컨피규레이션 및 비디오 게임의 식별자와 함께 전송될 수 있다. 또한, 액세스 데이터(110)는, 액세스 데이터(110)가 생성될 때 원격 시스템(106)으로 액세스 데이터(110)를 스트리밍함으로써, 그리고/또는 이벤트에 대한 응답으로, 실시간 또는 실질적으로 실시간과 같은 임의의 적절한 시간에 원격 시스템(106)으로 송신될 수 있다(예를 들어, 주기적으

로, 리소스 소비의 임계 백분율 미만과 같이 처리 리소스 소비가 상대적으로 낮은 아이들링 시간 동안, 네트워크 연결이 재개될 때(예를 들어, 오프라인으로 게임을 플레이한 후), 클라이언트 머신(104)이 게임 실행 파일(122)의 실행을 중지한 후(예를 들어, 사용자(102)가 비디오 게임을 나감으로써 게임 세션을 종료한 후) 등). 액세스 데이터(110)는 또한, 예컨대, 액세스 데이터(110)를 아티팩트로서 원격 시스템(106)에 송신함으로써, 클라이언트 머신(104) 상에서의 추적된 판독 작동으로부터 기인되는 메타데이터를 전송하기 위한 임의의 적절한 포맷으로 송신될 수 있다. 프로세스(200)는 원격 시스템(106)에서 액세스 데이터(110)를 수집하기 위한 "원격 측정(telemetry)" 접근 방식을 나타낸다. 다수의 클라이언트 머신(104)이 프로세스(200)를 수행하고 있을 수 있다는 것을 고려하면, 원격 시스템(106)은 블록 224에서, 다양한 클라이언트 시스템 컨피규레이션을 갖는 아주 많은 클라이언트 머신(104)으로부터 송신(예를 들어, 업로드, 리포트 등)되는 액세스 데이터(110)를 수집할 수 있다.

[0036] 도 3은 원격 시스템(106)의 예시적인 컴포넌트를 예시하는 블록도, 및 클라이언트 머신(104)으로부터 액세스 데이터(110)를 수신하기 위한, 그리고 본원에 설명되는 기술을 구현하기 위해 하나 이상의 사용자(102)에 걸쳐 액세스 데이터(110)를 분석하기 위한 예시적인 프로세스(300)의 흐름도를 도시한다. 예시된 구현예에서, 원격 시스템(106)은, 다른 컴포넌트 중에서, 하나 이상의 프로세서(302), 메모리(304)(또는 비일시적 컴퓨터 판독 가능 매체(304)), 및 통신 인터페이스(들)(306)를 포함한다. 메모리(304)(또는 비일시적 컴퓨터 판독 가능 매체(304))는 컴퓨터 판독 가능 명령, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성 메모리, 착탈식 및 비착탈식 매체를 포함할 수 있다. 이러한 메모리는, RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, 디지털 다목적 디스크(DVD) 또는 기타 광 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 디바이스, 또는 기타 자기 저장 디바이스, RAID 저장 시스템, 또는 원하는 정보를 저장하는 데 사용될 수 있으며 컴퓨팅 디바이스에 의해서 액세스될 수 있는 임의의 다른 매체를 포함하나, 이에 한정되지 않는다. 컴퓨터 판독 가능 매체(304)는 컴퓨터 판독 가능 저장 매체(CRSM)로서 구현될 수 있으며, 이는 메모리(304)에 저장된 명령어를 실행하기 위해 프로세서(들)(302)에 의해 액세스 가능한 임의의 이용 가능한 물리적 매체일 수 있다. 하나의 기본 구현예에서, CRSM은 랜덤 액세스 메모리(RAM) 및 플래시 메모리를 포함할 수 있다. 다른 구현예에서, CRSM은, 읽기 전용 메모리(ROM), 전기적으로 지울 수 있는 프로그래밍 가능한 읽기 전용 메모리(EEPROM), 또는 원하는 정보를 저장하기 위해서 사용될 수 있고, 프로세서(들)(302)에 의해서 액세스될 수 있는 임의의 다른 유형(tangible)의 매체를 포함할 수 있으나, 이에 한정되지 않는다. 다운로드 시퀀스 컴포넌트(308), 블록 종속성 컴포넌트(310), 및/또는 게임 데이터 사용 컴포넌트(312)는, 프로세서(들)(302)에 의해 실행될 때, 원격 시스템(106)으로 하여금 본원에 설명된 기술 및 작동을 수행하게 하는 메모리(304)에 저장된 명령어를 나타낼 수 있다. 예를 들어, 다운로드 시퀀스 컴포넌트(308)는, 본원에 설명된 바와 같이, 액세스 데이터(110)에 기초하여 다운로드 시퀀스 데이터(114)를 생성하도록 구성될 수 있다. 블록 종속성 컴포넌트(310)는, 본원에 설명된 바와 같이, 액세스 데이터(110)에 기초하여 블록 종속성 데이터(116)를 생성하도록 구성될 수 있다. 게임 데이터 사용 컴포넌트(312)는, 여기에 설명된 바와 같이, 사용자별 또는 클라이언트별 머신(104) 기반으로 게임 데이터(126)의 미사용 블록(124)을 결정하도록 구성될 수 있다.

[0037] 통신 인터페이스(들)(306)는 다수의 유형의 유선 및/또는 무선 또는 라디오(radio) 기술을 구현할 수 있다. 예를 들어, 통신 인터페이스(들)(306)는 저전력 블루투스(BLE) 라디오, Wi-Fi 라디오, 및/또는 셀룰러 라디오 등과 같은 라디오를 구현할 수 있다. 통신 인터페이스(들)(306)가, 네트워크, 연결된 주변 디바이스, 또는 다른 무선 네트워크와 통신하는 플러그인 네트워크 디바이스에 대한 유선 연결을 용이하게 하는 물리적 포트를 더 포함할 수 있다는 것이 이해될 것이다.

[0038] 이제 도 3에 도시된 프로세스(300)가 참조된다. 314에서, 원격 시스템(106)은 클라이언트 머신(114)으로부터 액세스 데이터(110)를 수신할 수 있다(예를 들어, 위에서 설명된 프로세스(200)를 사용하는 "원격 측정" 접근 방식의 일부로서). 수신된 액세스 데이터(110)는, 특정 클라이언트 머신(104)에 대해, 비디오 게임의 게임 실행 파일(112)에 의해 특정 클라이언트 머신(104) 상에서 액세스되었던 비디오 게임을 위한 게임 데이터(126)의 블록(124)의 식별자, 및 블록(124)이 게임 세션 동안(예를 들어, 비디오 게임의 게임 실행 파일(들)(122)의 실행 동안) 액세스되었던 시간을 상술할 수 있다. 서브블록 316에 의해 도시된 바와 같이, 액세스 데이터(110)는 특정 클라이언트 머신(104)의 컨피규레이션과 함께, 그리고 액세스 데이터(110)의 생성 동안 실행되었던 비디오 게임의 식별자와 함께 수신될 수 있다.

[0039] 318에서, 원격 시스템(106)은, 클라이언트 시스템 컨피규레이션 및 게임 식별자에 의해(또는 이에 따라) 블록 314에서 수신한 액세스 데이터(110)를 카탈로그화할 수 있다. 예를 들어, 데이터 저장소(112)는, 게임 ID 및 클

라이언트 시스템 컨피규레이션의 고유한 조합에 의해 분류되는 다수의 그룹 또는 버킷(320(1)-(M))을 포함할 수 있다. 예를 들어, 클라이언트 시스템 컨피규레이션 "1"을 갖는 제1 클라이언트 머신(104(1))을 취하면, 제1 액세스 데이터(110)가 블록 314에서 제1 클라이언트 머신(104(1))으로부터 수신될 때, 제1 액세스 데이터(110)는 블록 318에서 제1 버킷(320(1)(또는 그룹))으로 카탈로그화될 수 있고, 이 버킷(320(1))은, 서버블록 316에서 제1 클라이언트 머신(104(1))으로부터 수신된 제1 클라이언트 머신(104(1))의 클라이언트 시스템 컨피규레이션 및 비디오 게임의 게임 ID로 태그될 수 있다. 마찬가지로, 제2 클라이언트 머신(104(2))은 제2 액세스 데이터(110)를 송신할 수 있고, 이 제2 액세스 데이터(110)는 블록 318에서 제2 버킷(320(2))으로 카탈로그화될 수 있으며, 이는, 서버블록 316에서 제2 클라이언트 머신(104(2))으로부터 수신된 제2 클라이언트 머신(104(2))의 제2 클라이언트 시스템 컨피규레이션 "2" 및 비디오 게임의 게임 ID로 태그될 수 있다. 이것은, 액세스 데이터(110)를 리포트하는 클라이언트 머신들(104)의 수에 따라, 클라이언트 머신들(104)이 상이한 클라이언트 시스템 컨피규레이션을 갖는지 여부에 따라, 그리고/또는 얼마나 많은 상이한 비디오 게임들이 이러한 클라이언트 머신들(104) 상에서 실행되고 있는지에 따라 임의의 수 "M" 개의 버킷에 대해 계속될 수 있다.

[0040] 322에서, 원격 시스템(106)은 블록 314에서 수신된 액세스 데이터(110)에 대한 분석을 수행하여 데이터(예를 들어, 분석의 결과)를 생성할 수 있다. 이 생성된 데이터는, 한정됨 없이, 하나 이상의 비디오 게임에 대한 다운로드 시퀀스 데이터(114), 하나 이상의 비디오 게임에 대한 블록 종속성 데이터(116), 및/또는 사용자별/클라이언트 머신별 기반 게임 데이터(126)의 미사용 블록(124)의 결정을 포함할 수 있다.

[0041] 324에서, 원격 시스템(106)은 트리거 이벤트(들)가 발생했는지 여부를 결정할 수 있다. 트리거 이벤트는 다양할 수 있지만, 비제한적으로, 비디오 게임을 획득(예를 들어, 구매, 임대, 리스 등)하라는 요청(120)을 클라이언트 머신(104)으로부터 수신하는 것, 비디오 게임에 대한 새로운 다운로드 시퀀스 데이터(114) 및/또는 새로운 블록 종속성 데이터(116)를 생성하는 것, 임계 시간 기간 또는 임계 수의 게임 세션 동안 액세스되지 않은 클라이언트 머신(104) 상의 게임 데이터(126)의 미사용 블록(124)이 있다고 결정하는 것, 일정 기간의 경과 등을 포함할 수 있다.

[0042] 원격 시스템(106)이, 블록 324에서, 트리거 이벤트가 발생하지 않았다고 결정하면, 프로세스(300)는 블록 324로부터 다시 블록 314로의 "아니오(NO)" 경로를 따를 수 있으며, 블록 314에서 원격 시스템(106)은 클라이언트 머신(104)으로부터 액세스 데이터(110)를 계속 수집/수신한다. 원격 시스템(106)이, 블록 324에서, 트리거 이벤트가 발생했다고 결정하면, 프로세스(300)는 블록 324로부터 블록 326으로의 "예(YES)" 경로를 따를 수 있으며, 블록 326에서 원격 시스템(106)은 하나 이상의 클라이언트 머신(104)에 데이터를 송신할 수 있다. 예를 들어, 트리거 이벤트가, 클라이언트 머신(104)이 비디오 게임을 획득하는 것을 포함하는 경우, 원격 시스템(106)은 비디오 게임에 대한 다운로드 시퀀스 데이터(114)를 검색할 수 있고, 다운로드 시퀀스 데이터(114)를 비디오 게임을 위한 게임 실행 파일(122)과 함께 클라이언트 머신(104)에 송신할 수 있다. 추가적으로 또는 대안적으로, 원격 시스템(106)은 비디오 게임에 대한 블록 종속성 데이터(116)를 검색할 수 있고, 블록 종속성 데이터(116)를 클라이언트 머신(들)(104)에 송신할 수 있다. 또 다른 실시예로서, 트리거 이벤트가 클라이언트 머신(104) 상에 게임 데이터(126)의 미사용 블록(들)(124)이 존재한다고 결정하는 것을 포함하는 경우, 원격 시스템(106)은 게임 데이터의 미사용 블록(들)을 삭제하라는 명령어를 송신할 수 있다.

[0043] 일부 실시형태에서, 도 2에 도시된 클라이언트 머신(104)의 컴포넌트 중 적어도 일부는 원격 시스템(106)의 컴포넌트(들)로서 구현될 수 있다는 것이 이해될 것이다. 예를 들어, 게임 실행 파일(122)은 원격 시스템(106) 상에서 (예를 들어, 비디오 게임 스트리밍 서비스의 일부로서) 실행될 수 있고, 게임 실행 파일(122)은 비디오 게임을 플레이하는 동안, 사용자 입력을 나타내는 데이터를 클라이언트 머신(104)으로부터 네트워크(108)를 통해 수신할 수 있다. 이 시나리오에서, 비디오 게임을 위한 게임 데이터(126)는 또한 원격 시스템(106)에 저장될 수 있고, 파일 시스템 프록시 컴포넌트(218)는, 게임플레이 동안 액세스되는 게임 데이터(126)의 블록(124)을 모니터링하기 위해 게임 실행 파일(122)로부터 판독 작동을 수신하는 원격 시스템(106)의 컴포넌트일 수 있다. 이 컨피규레이션에서, 클라이언트 머신(104)은, 비디오 게임 세션 동안 프로세스의 대부분이 클라우드에서 수행되는 쉘 클라이언트로서 작용할 수 있다. 다른 실시예에서, 다운로드 시퀀스 컴포넌트(308), 블록 종속성 컴포넌트(310), 및/또는 게임 데이터 사용 컴포넌트(312)는 클라이언트 머신(104)의 컴포넌트일 수 있다. 이러한 실시예에서, 클라이언트 머신(104)은 원격 시스템(106)에 의해 수집된 다른 클라이언트 머신(104)에 의해 생성된 액세스 데이터(110)를 수신할 수 있고, 클라이언트 머신(104)은 자신의 액세스 데이터(110) 및/또는 다른 클라이언트 머신(104)에 의해 생성된 액세스 데이터(110)에 대한 분석을 수행하여, 클라이언트 머신(104) 상에 저장된 게임 데이터(126)의 다운로드 시퀀스 데이터(114), 블록 의존 데이터(116), 및/또는 미사용 블록(124)을 결정할 수 있다.

- [0044] 도 4는 게임 데이터(126)의 블록들(124)을 블록들(124)의 특정 시퀀스로 다운로드하기 위해 클라이언트 머신(104)에 의해 사용되기 위한 다운로드 시퀀스 데이터(114)를, 다수의 클라이언트 머신(104)으로부터 수신된 액세스 데이터(110)에 기초하여, 생성하기 위한 예시적인 프로세스(400)의 흐름도이다.
- [0045] 402에서, 원격 시스템(106)은 다수의 클라이언트 머신(104)으로부터 비디오 게임과 연관된 액세스 데이터(110)를 수신할 수 있다. 블록 402에서 수신된 액세스 데이터(110)는, 클라이언트 머신들(104) 중 개별 머신에 대해서, (i) 개별 클라이언트 머신 상에서 게임 실행 파일(122)을 실행하는 동안 비디오 게임의 게임 실행 파일(122)에 의해 액세스되었던 비디오 게임과 연관된 게임 데이터(126)의 액세스된 블록(124), 및 (ii) 게임 실행 파일(122)의 실행 동안, 게임 데이터(126)의 액세스된 블록(124)이 게임 실행 파일(122)에 의해 액세스되었던 시간을 상술할 수 있다. 서브블록 404에 의해 도시된 바와 같이, 액세스 데이터(110)는 액세스 데이터(110)를 송신한 클라이언트 머신(104)의 컨피규레이션, 및/또는 액세스 데이터(110)가 생성되었을 때 클라이언트 머신(104) 상에서 실행된 비디오 게임의 게임 식별자와 함께 수신될 수 있다.
- [0046] 406에서, 원격 시스템(106)은, 액세스 데이터(110)에 적어도 부분적으로 기초하여, 비디오 게임의 게임 식별자 및 클라이언트 시스템 컨피규레이션과 연관된 다운로드 시퀀스 데이터(114)를 생성할 수 있다. 다운로드 시퀀스 데이터(114)는 비디오 게임을 위한 게임 데이터(126)의 블록들(124)의 시퀀스를 상술할 수 있고, 이 시퀀스는, 게임 데이터의 블록들(124)이 특정 클라이언트 시스템 컨피규레이션을 갖는 클라이언트 머신(104)으로 다운로드 되는 순서를 나타낼 수 있다. 비디오 게임을 위한 다운로드 시퀀스는 임의의 적절한 방식으로 결정될 수 있다. 서브블록 408 및 410에 의해 도시된 바와 같이, 다운로드 시퀀스 데이터(114)의 생성은 다수의 하위 작동을 포함할 수 있다.
- [0047] 408에서, 비디오 게임을 플레이한 N명("N"은 양의 정수임)의 대응되는 사용자(102)의 N개의 게임 세션을 통해 이러한 블록(124)이 액세스된(액세스 데이터(110)에 상술된 바와 같이) 때(예를 들어, 시간, 위치/순서/순위 등)를 적어도 부분적으로 기초하여 비디오 게임을 위한 게임 데이터(126)의 적어도 일부 블록(124)에 대한 통계가 계산될 수 있다. 예를 들어, 비디오 게임을 플레이한 N명의 사용자(102)와 연관된 액세스 데이터(110)에 기초하여, 원격 시스템(106)은, 게임 세션 동안(예를 들어, 초기 게임 세션 동안) N명의 사용자(102) 각각에 의해 이러한 블록(124)이 액세스된 시간에 기초하여 통계(예를 들어, 평균값)를 결정할 수 있다. 원격 시스템(106)은, 예를 들어, (예를 들어, N명의 상이한 사용자에게 의해서 블록 A가 액세스된 다수의 상이한 시간에 기초하여) 블록 A에 대한 평균 액세스 시간이 초기 게임 세션에서 30분이었다는 것, 블록 B에 대한 평균 액세스 시간이 초기 게임 세션에서 45분이었다는 것, 블록 C에 대한 평균 액세스 시간이 초기 게임 세션에서 2시간이었다는 것, 등등을 결정할 수 있다. 원격 시스템(106)은, 추가로 또는 대안적으로, 블록 A가 초기 게임 세션 동안 사용자 A에 의해 액세스된 첫 번째 블록이었다는 것, 블록 A가 초기 게임 세션 동안 사용자 B에 의해서 액세스된 다섯 번째 블록이었다는 것, 블록 A가 초기 게임 세션 동안 사용자 C에 의해서 액세스된 일곱 번째 블록이었다는 것 등등을 N명의 사용자에게 대해서 결정할 수 있고, 다음으로, 원격 시스템(106)은, 블록이 N개의 상이한 사용자(102)에 의해 액세스된 때에 관한 이러한 정보에 적어도 부분적으로 기초하여 블록 A에 대한 통계(예를 들어, 평균 위치/순서/순위)를 결정할 수 있다. 이것은 게임 세션 동안 액세스되었던 다른 블록(124)에 대해 반복될 수 있다. 이러한 분석은 적어도 초기 게임 세션 동안 수행될 수 있다. 분석은, 게임 데이터(126)의 어느 블록(124)이 게임 세션(들) 동안 일반적으로 액세스되는지를 결정하기 위해 임의의 후속 게임 세션에 대해서도 수행될 수 있다.
- [0048] 410에서, 계산된 통계에 기초하여, 비디오 게임을 위한 게임 데이터(126)의 다수의 블록(124)에 대한 다운로드 시퀀스가 결정될 수 있다. 다운로드 시퀀스에 상술된 블록들(124)은 게임 데이터의 블록들(124) 중 일부(예를 들어, 블록(124)의 전체 수의 부분)일 수 있거나, 다운로드 시퀀스는 게임 데이터(126)의 블록들(124) 전부를 포함할 수 있다. 주어진 비디오 게임에 대해, 플레이타임의 초기 기간 동안 액세스될 가능성이 있는 블록들(124)의 초기 수와 플레이타임의 초기 기간의 지속 기간 사이의 비율은 다소 편향될 수 있다는 것이 이해될 것이다. 다시 말해서, 2 기가바이트의 게임 데이터(126)는 특정 비디오 게임에 대해 초기 6시간의 플레이시간을 제공할 수 있고, 게임 데이터(126)의 나머지는 이 초기 플레이 기간 동안 액세스될 가능성이 낮을 수 있다. 이것은, 특정 게임이 "인스턴트 플레이"(128) 기능에 매우 적합할 수 있고, 이 비율에 기초하여 인스턴트 플레이에 매우 적합한 "좋은 인스턴트 플레이 게임"으로서 플래그(flag)될 수 있다는 것을 의미한다. 다시 말해, 상대적으로 적은 양의 게임 데이터(126)에 의해 상대적으로 긴 초기 플레이 기간이 지원될 수 있는 경우 특정 게임이 플래그될 수 있다. 예를 들어, 싱글 플레이어 게임은, 새로운 플레이어가 동일한 레벨을 먼저 플레이하도록 요구될 수 있기 때문에, 레이턴시가 거의 또는 전혀 없는 "인스턴트 플레이"(128) 기능에 매우 적합할 수 있다. 싱글 플레이어 게임의 경우, 원격 시스템(106)은, 게임을 처음으로 플레이하는 모든 또는 거의 모든 플레이어가

플레이타임의 처음 X시간 내에 동일한 블록(124)에 액세스한다고 결정할 수 있고, 이 패턴은 일 세트의 N명의 플레이어에 걸쳐 실질적으로 선형이고 일관적일 수 있다 - 예를 들어, 모든 N 플레이어는 블록 A에 액세스한 다음, 블록 H, 블록 Q 등에 액세스할 수 있다. 블록들(124)이 액세스되는 정확한 시간들에 불일치가 있을 수 있지만, 주어진 비디오 게임에 대한 플레이타임의 초기 기간에 걸쳐 액세스되는 블록들(124)의 공통 순서가 있을 수 있다. 대조적으로, 멀티플레이어 게임은 거대한 지도를 구비하는 오픈 게임 월드(open game world)를 가질 수 있다. 예를 들어, N명의 사용자에 걸쳐, N명의 사용자는 처음에 오픈 월드의 상이한 위치들에서 게임 안으로 드랍(drop)될 수 있고, 각각의 게임 실행 파일(122)은, 매칭된 플레이어가 게임 안으로 드랍된 오픈 월드의 상이한 위치에 대응되는 게임 데이터(126)의 상이한 블록(124)에 액세스할 것이다. 이것은 인스턴트 플레이(128) 기능에 적합하지 않은 게임의 예이다.

[0049] 비디오 게임이 인스턴트 플레이(128) 기능을 지원할 수 있다고 가정하면, 다운로드 시퀀스는 블록 A, 블록 D, 블록 G, 블록 C, 블록 F, 블록 E 등의 순서를 상술할 수 있다. 따라서, 임의의 비디오 게임에 대해, 그리고 임의의 클라이언트 시스템 컨피규레이션에 대해, 원격 시스템(106)은, 적어도 플레이타임의 일 기간(예를 들어, 플레이타임의 처음 6시간) 동안, 평균 사용자(102)가 게임 데이터의 특정 서브세트의 블록(124)을 독점적으로 액세스할 가능성이 높고, 이 플레이타임의 기간 동안 특정 시퀀스로 이러한 블록들(124)에 액세스할 가능성이 높다고 결정할 수 있다. 일부 실시형태에서, 다운로드 시퀀스가 사용자 특정적일 수 있다는 것이 이해될 것이다. 즉, 동일한 비디오 게임 및 동일한 클라이언트 시스템 컨피규레이션에 대해, 제1 다운로드 시퀀스가 제1 사용자(102)에 대해 결정될 수 있고, 제2 다운로드 시퀀스가 제2 사용자(102)에 대해 결정될 수 있으며, 제2 다운로드 시퀀스는 제1 다운로드 시퀀스와 상이하다. 이것은 사용자-특정 액세스 데이터(110)에 나타난 고유한 액세스 패턴에 기초할 수 있다.

[0050] 412에서, 원격 시스템(106)은 비디오 게임을 획득하기 위한 요청(120)을 클라이언트 머신(104)으로부터 수신할 수 있으며, 요청(120)은 클라이언트 머신(104)의 컨피규레이션을 포함할 수 있다. 예를 들어, 비디오 게임 클라이언트(212)를 통해 자신의 사용자 계정에 로그인된 클라이언트 머신(104)의 사용자(102)는 비디오 게임을 구매하기 위해 원격 시스템(106)을 통해 거래를 실시할 수 있다.

[0051] 414에서, 원격 시스템(106)은 비디오 게임의 게임 실행 파일(122)을 클라이언트 머신(104)에 송신할 수 있다. 이 게임 실행 파일(122)은 클라이언트 머신(104)에서 비디오 게임 플레이를 시작하기 위해 클라이언트 머신(104)의 프로세서(들)(202)에 의해 실행 가능한 실행 코드(예를 들어, 파일(들))일 수 있다. 서브블록 416에 의해 도시된 바와 같이, 원격 시스템(106)은, 다운로드 시퀀스 데이터(114)에 의해 상술된 시퀀스로, 클라이언트 머신(104)이 획득된 비디오 게임을 위한 게임 데이터(126)의 블록들(124)의 다운로드를 시작하도록, 클라이언트 머신(104)에 다운로드-시퀀스 데이터(114)를 송신할 수 있다.

[0052] 418에서, 원격 시스템(106)은, 다운로드-시퀀스 데이터(114)에 상술된 시퀀스에 따라 게임 데이터의 블록들(124)을 클라이언트 머신(104)으로 다운로드할 수 있다. 이것은 클라이언트 머신(104)에 대해 달성 가능한 다운로드 링크 데이터 전송 속도에 따라 다소 시간이 걸릴 수 있다. 따라서, 다운로드되는 블록들(124)의 시퀀스 때문에, 먼저 액세스될 가능성이 가장 높은 블록들(124)이 다른 블록들보다 먼저 클라이언트 머신(104)의 로컬 메모리(132)에서 실현되고, 다른 블록들은, 비디오 게임의 게임 실행 파일(122)이 다른 블록들을 관독할 것을 요청할 때까지, 클라이언트 머신(104)으로의 다운로드를 완료했을 가능성이 높다.

[0053] 도 5는, 게임플레이 동안 레이턴시를 감소시키도록 게임 데이터(126)의 블록(124)을 프리페치하기 위해 클라이언트 머신(104)에 의해 사용되기 위한 블록 종속성 데이터(116)를, 다수의 클라이언트 머신(104)으로부터 수신된 액세스 데이터(110)에 기초하여, 생성하기 위한 예시적인 프로세스(500)의 흐름도이다.

[0054] 502에서, 원격 시스템(106)은 다수의 클라이언트 머신(104)으로부터 비디오 게임과 연관된 액세스 데이터(110)를 수신할 수 있다. 블록 502에서 수신된 액세스 데이터(110)는, 클라이언트 머신들(104) 중 개별 머신에 대해서, (i) 개별 클라이언트 머신 상에서 게임 실행 파일(122)을 실행하는 동안 비디오 게임의 게임 실행 파일(122)에 의해 액세스되었던 비디오 게임과 연관된 게임 데이터(126)의 액세스된 블록(124), 및 (ii) 게임 실행 파일(122)의 실행 동안, 게임 데이터(126)의 액세스된 블록(124)이 게임 실행 파일(122)에 의해 액세스되었던 시간을 상술할 수 있다. 서브블록 504에 의해 도시된 바와 같이, 액세스 데이터(110)는 액세스 데이터(110)를 송신한 클라이언트 머신(104)의 컨피규레이션, 및/또는 액세스 데이터(110)가 생성되었을 때 클라이언트 머신(104) 상에서 실행된 비디오 게임의 게임 식별자와 함께 수신될 수 있다.

[0055] 506에서, 원격 시스템(106)은, 액세스 데이터(110)에 적어도 부분적으로 기초하여, 비디오 게임의 게임 식별자 및 클라이언트 시스템 컨피규레이션과 연관된 블록 종속성 데이터(116)를 생성할 수 있다. 블록 종속성 데이터

(116)는 비디오 게임을 위한 게임 데이터(126)의 둘 이상의 블록(124) 사이의 개별 연관성을 상술할 수 있다. 달리 말하면, 원격 시스템(106)은, 특정 이벤트가 발생하는 경우 액세스될 게임 데이터(126)의 하나 이상의 블록(124)을 예측할 수 있다. 이러한 방식으로, 게임 데이터(126)의 블록들(124) 사이의 그리고/또는 블록(124)과 컨텍스트 큐(contextual cue) 사이의 의존성(예를 들어, 분기, 트리형 의존성)의 맵이 액세스 데이터(110)에 기초하여 구축될 수 있다. 서브블록들(508, 510)에 의해 도시된 바와 같이, 블록 종속성 데이터(116)의 생성은 하나 이상의 서브작동을 포함할 수 있다.

[0056] 508에서, 원격 시스템(106)은, 비디오 게임을 플레이한 N명의 사용자(102)로부터 수신된 액세스 데이터(110)에 나타난 액세스 패턴(예를 들어, 액세스 시간)에 기초하여, 비디오 게임을 위한 게임 데이터(126)의 블록(124)과 컨텍스트 큐 사이의 연관성을 결정할 수 있다. 예를 들어, 원격 시스템(106)은, 특정 컨텍스트 큐가 검출될 때마다(예를 들어, 사용자(102)가 비디오 게임의 라이브러리 페이지로 네비게이트할 때마다), 게임 데이터(126)의 특정 블록(124)이 일반적으로(예를 들어, N명의 사용자 평균) 임계 시간 기간 내에 액세스된다고 결정할 수 있다. 이러한 유형의 상관관계는 액세스 데이터에 상술된 액세스 시간에 적어도 부분적으로 기초하여 결정될 수 있다. 예를 들어, N명의 사용자에 대해, 블록 A가, N명의 사용자가 비디오 게임의 라이브러리 페이지를 네비게이트하는 처음 15초 이내에 가장 많이 액세스되는 블록인 경우, 블록 A는 이러한 유형의 컨텍스트 큐와 연관될 수 있다.

[0057] 510에서, 원격 시스템(106)은, 비디오 게임을 플레이한 N명의 사용자(102)로부터 수신된 액세스 데이터(110)에 나타난 액세스 패턴(예를 들어, 액세스 시간)에 기초하여, 비디오 게임을 위한 게임 데이터(126)의 블록(124)의 쌍들 사이의 연관성을 결정할 수 있다. 예를 들어, 원격 시스템(106)은, 비디오 게임을 플레이한 N명의 사용자(102)로부터 수신된 액세스 데이터(110)에 기초하여, 게임 세션 동안 게임 데이터(126)의 제1 블록(124(1))이 액세스될 때마다, 게임 데이터(126)의 제2 블록(124(2))이 일반적으로 임계 시간 기간 내에 액세스된다고 결정할 수 있다. 이러한 방식으로, 원격 시스템(106)은, 액세스 데이터(110)에 나타난 액세스 패턴에 기초하여, 두 개 이상의 블록(124)의 그룹들 사이의 블록간 관계를 결정할 수 있다.

[0058] 일부 실시형태에서, 블록 종속성 데이터(116)가 사용자 특정적일 수 있다는 것이 이해될 것이다. 즉, 동일한 비디오 게임 및 동일한 클라이언트 시스템 컨피규레이션에 대해, 블록 종속성의 제1 맵은 제1 사용자(102)에 대해 결정될 수 있다, 블록 종속성의 제2 맵은 제2 사용자(102)에 대해 결정될 수 있으며, 제2 맵은 제1 맵과 상이하다. 이것은 사용자-특정 액세스 데이터(110)에 나타난 고유한 액세스 패턴에 기초할 수 있다.

[0059] 512에서, 원격 시스템(106)은 블록 종속성 데이터(116)를 하나 이상의 클라이언트 머신(104)으로 송신할 수 있다. 블록 종속성 데이터(116)는 다양한 트리거 이벤트에 응답하여 송신될 수 있다. 예를 들어, 원격 시스템(106)은, 클라이언트 머신(104)의 사용자(102)가 비디오 게임을 획득(예를 들어, 구매, 임대, 리스 등)한 것에 응답하여 비디오 게임을 위한 블록 종속성 데이터(116)를 클라이언트 머신(104)에 송신할 수 있다. 다른 실시예로서, 블록 506에서 생성된 블록 종속성 데이터(116)는 비디오 게임을 위한 블록 종속성 데이터(116)의 이전 버전에 대해 새로운 데이터(또는 업데이트된 데이터)를 나타낼 수 있고, 원격 시스템(106)은, 블록 506에서 새로운/업데이트된 블록 종속성 데이터(116)를 생성하는 것에 응답하여, 블록 종속성 데이터(116)를 비디오 게임의 소유자와 연관된 것으로 알려진 클라이언트 머신(104)에 송신할 수 있다. 또 다른 실시예로서, 사용자는 자신의 클라이언트 머신(104)에서 구현된 로컬 프리젠템 기능을 갖는 것을 옵트인(opt-in)할 수 있고, 옵트인에 응답하여, 원격 시스템(106)은 옵트인된 사용자의 클라이언트 머신(104)에 블록 종속성 데이터(116)를 송신할 수 있다.

[0060] 도 6은, 클라이언트 머신(104)으로부터 수신된 액세스 데이터(110)에 기초하여, 게임 데이터(126)의 미사용 블록(124)을 결정하고, 게임 데이터(126)의 미사용 블록(124)을 폐기하도록 클라이언트 머신(104)에 지시하기 위한 예시적인 프로세스(600)의 흐름도이다.

[0061] 602에서, 원격 시스템(106)은 클라이언트 머신(104)으로부터 비디오 게임과 연관된 액세스 데이터(110)를 수신할 수 있다. 블록 602에서 수신된 액세스 데이터(110)는, (i) 클라이언트 머신(104) 상에서 게임 실행 파일(122)을 실행하는 동안 비디오 게임의 게임 실행 파일(122)에 의해 액세스되었던 비디오 게임과 연관된 게임 데이터(126)의 액세스된 블록(124), 및 (ii) 게임 실행 파일(122)의 실행 동안, 게임 데이터(126)의 액세스된 블록(124)이 게임 실행 파일(122)에 의해 액세스되었던 시간을 상술할 수 있다. 원격 시스템(106)이 동일한 클라이언트 머신(104)으로부터 이전에 과거의 어떤 시간(들)에 액세스 데이터(110)를 수신했을 수 있다는 것이 이해될 것이다. 따라서, 원격 시스템(106)은 클라이언트 머신(104) 상에서 플레이되는 다수의 게임 세션에 걸쳐 생성되는 액세스 데이터(110)에 대한 액세스를 가질 수 있다. 서브블록 604에 의해 도시된 바와 같이, 액세스 데

이터(110)는 액세스 데이터(110)를 송신한 클라이언트 머신(104)의 컨피규레이션, 및/또는 액세스 데이터(110)가 생성되었을 때 클라이언트 머신(104) 상에서 실행된 비디오 게임의 게임 식별자와 함께 수신될 수 있다.

[0062] 606에서, 원격 시스템(106)은, 액세스 데이터(110)(및 아마도 과거에 클라이언트 머신(104)로부터 수신된 추가 액세스 데이터(110))에 기초하여, 클라이언트 머신(104)의 비휘발성 메모리(예를 들어, 제1 메모리 132(1))에 현재 저장되어 있는 비디오 게임을 위한 게임 데이터(126)의 임의의 블록(124)이 게임 데이터(126)의 "미사용" 블록(124)으로서 분류될 수 있는지 여부를 결정할 수 있다. 서브블록 608에 의해 도시된 바와 같이, 이 결정은, 개별 블록(124)이 임계 시간 기간 동안 또는 임계 수의 게임 세션 동안 클라이언트 머신(104) 상의 비디오 게임의 게임 실행 파일(들)(122)에 의해 액세스되지 않았는지 여부를 결정하는 것을 포함할 수 있다. 예를 들어, 게임 데이터(126)의 블록 X-Z가 과거 P 개의 게임 세션("P"는 임의의 적절한 정수임) 또는 마지막 Q 주(week)("Q"는 임의의 적절한 정수, 예컨대, 지난 10주) 동안 액세스되지 않은 경우, 블록 X-Z가 미사용 블록일 것으로 결정할 수 있다.

[0063] 일부 실시형태에서, 다른 인자(factor)가 게임 데이터(126)의 블록(124)을 미사용 블록으로서 분류할 때 고려된다. 예를 들어, 원격 시스템(106)은, 특정 비디오 게임의 N명의 사용자에게 걸쳐, 게임 데이터(126)의 어떤 블록(들)(124)이 게임 실행 파일(122)에 의해 한 번 액세스되고 이후에 다시는 액세스되지 않는다는 것을 결정할 수 있고, 원격 시스템(106)은 특정 블록(들)(124)이 이미 한 번 액세스되었다고 결정할 수 있고, 만약 그렇다면, 블록(들)(124)이 게임 실행 파일(122)에 의해 얼마나 최근에 액세스되었는지에 관계없이, 블록(들)(124)을 미사용 인 것으로서 지칭할 수 있다. 미사용 블록(124)을 결정하기 위해 다른 휴리스틱(heuristic)이 사용될 수 있다.

[0064] 블록 606에서의 결정이 클라이언트 머신(104)의 비휘발성 메모리(예를 들어, 제1 메모리(132(1)))에 저장된 게임 데이터(126)의 블록(124) 중 어느 것도 미사용 블록(124)으로 분류될 수 없다는 것인 경우, 프로세스(600)는 블록 606으로부터 블록 602로의 "아니오(NO)" 경로를 따를 수 있으며, 블록 602에서 원격 시스템(106)이 클라이언트 머신(104)으로부터 추가 액세스 데이터(110)를 수신(또는 수신 대기)할 수 있다. 블록 606에서의 결정이 클라이언트 머신(104)의 비휘발성 메모리에 저장된 게임 데이터(126)의 하나 이상의 블록(124)이 미사용 블록(124)으로서 분류될 수 있다는 것인 경우, 프로세스(600)는 블록 606으로부터 블록 610으로의 "예(YES)" 경로를 따를 수 있다.

[0065] 610에서, 원격 시스템(106)은, 클라이언트 머신(104)의 비휘발성 메모리로부터 게임 데이터(126)의 미사용 블록(들)(124)을 삭제하라는 명령어(118)를 클라이언트 머신(104)에 송신할 수 있다. 이러한 명령어(118)는 삭제될 블록(124)의 식별자를 포함할 수 있다. 일단 삭제되면, 클라이언트 머신(104)의 로컬 메모리 리소스(132)가 확보되어 로컬 메모리 용량을 증가시키고 잠재적으로 여기에 다른 데이터를 저장할 수 있다.

[0066] 도 7은 클라이언트 머신(104)의 비휘발성 메모리로부터 게임 데이터의 미사용 블록을 폐기하기 위한 예시적인 프로세스(700)의 흐름도이다. 도 2 및 도 7의 오프 페이지 참조 "A"에 의해 도시된 바와 같이, 프로세스(700)는, 클라이언트 머신(104)이 원격 시스템(106)에 액세스 데이터(110)를 송신한 후 프로세스(200)의 블록 224로부터 계속될 수 있다.

[0067] 702에서, 클라이언트 머신(104)은, 비디오 게임을 위한 게임 데이터(126)의 하나 이상의 블록(124)을 비휘발성 메모리로부터 삭제하라는 명령어(118)를 원격 시스템(106)으로부터 수신할 수 있다. 원격 시스템(106)은, 클라이언트 머신에 의해 이전에 송신된 액세스 데이터(110)로부터, 삭제될 블록(들)(124)이 임계 시간 기간 또는 임계 수의 게임 세션 동안 액세스되지 않은 게임 데이터(126)의 미사용 블록(124)을 나타낸다고 결정할 수 있다.

[0068] 704에서, 클라이언트 머신(104)은, 원격 시스템(106)으로부터 수신된 명령어(118)에 기초하여 게임 데이터(126)의 하나 이상의 블록(124)을 비휘발성 메모리(예를 들어, 제1 메모리(132(1)))로부터 삭제할 수 있다. 서브블록 706에 의해 도시된 바와 같이, 클라이언트 머신(104)은, 게임 데이터(126)의 블록(들)(124)을 폐기하기 전에 클라이언트 머신(104)이 재부팅될 때까지 대기하도록 구성될 수 있다. 이러한 방식으로, 블록(들)(124)을 삭제하는 것은, 예를 들어, 사용자(102)가 비디오 게임(또는 다른 비디오 게임)을 현재 플레이하고 있다면, 어떤 문제도 일으키지 않는다. 게임 데이터(126)의 나머지 블록(124)이 미사용 블록(124)의 삭제 후에도 비휘발성 메모리에 지속된다는 것이 이해될 것이다. 따라서, 로컬 메모리 리소스(132)은 미래에 액세스될 가능성이 낮은 게임 데이터(126)의 블록(124)을 폐기함으로써 확보될 수 있다. 예시적인 사용 사례에서, 사용자(102)는 자신의 클라이언트 머신(104)에 비디오 게임 X를 인스톨했을 수 있고, 이렇게 한 이후로, 사용자(102)는 비디오 게임 X를 온라인에서만 플레이하고 있으며, 이는 사용자(102)가 싱글 플레이어 모드에서 비디오 게임 X를 플레이한 적이 없다는 것을 의미한다. 클라이언트 머신(104)에 의해 원격 시스템(106)으로 리포트된 액세스 데이터(110)로부터, 원격 시스템(106)은, 비디오 게임 X가 클라이언트 머신(104) 상에 인스톨된 이후로 수 주

및/또는 다수의 게임 세션에 걸쳐, 사용자(102)가 비디오 게임 X의 싱글 플레이어 모드를 위한 게임 데이터(126)의 블록(124)에 액세스한 적이 없다고 결정할 수 있다. 이 경우에, 원격 시스템(104)은 (이 게임 데이터(126)에 대응되는 하나 이상의 블록(124)을 식별함으로써) 싱글 플레이어 게임 데이터(126)를 폐기하라는 명령어(118)를 송신할 수 있고, 클라이언트 머신(104)은 이를 수신할 수 있고, 클라이언트 머신(104)은 클라이언트 머신(104)의 비휘발성 메모리(예를 들어, 제1 메모리(132(1)))의 공간을 확보하기 위해 이러한 블록(124)을 삭제할 수 있다.

[0069] 삭제 후, 비디오 게임은 정상적으로 실행될 수 있고, 게임 실행 파일(122)에 의해 수행되는 판독 작동은 계속 수행될 수 있다. 게임 실행 파일(122)이 클라이언트 머신(104)의 비휘발성 메모리로부터 삭제된 게임 데이터(126)의 블록(124)을 판독할 것을 요청하는 희박한 경우에, 파일 시스템 프록시 컴포넌트(218)는 요구 시 원격 시스템(106)으로부터 이러한 블록을 검색할 수 있다. 일부 실시형태에서, 삭제된 블록(124)이 게임 실행 파일(122)에 의해 판독되도록 요청되면, 이러한 블록(124)은 클라이언트 머신(104)의 비휘발성 메모리로 재다운로드될 수 있다. 본원에 설명된 다른 프로세스 및/또는 기술과 함께, 프로세스(700)의 구현은 클라이언트 머신(104) 상의 귀중한 디스크 공간을 회수할 수 있다(예를 들어, 대략 수십 기가바이트의 디스크 공간).

[0070] 도 8은 게임 다운로드가 완료되기 전에 클라이언트 머신(104) 상에서 비디오 게임을 실행하기 위한 예시적인 프로세스(800)의 흐름도이다. 이것은, 프로세스(800)가, 게임 데이터(126)의 클라이언트 머신(104)으로의 다운로드를 기다리지 않으면서, 비디오 게임의 획득 시 사용자(102)의 비디오 게임 플레이를 허용할 수 있기 때문에, 본원에서 때때로 "인스턴트 플레이"(128) 특징으로 지칭된다.

[0071] 802에서, 클라이언트 머신(104)은 비디오 게임을 획득(예를 들어, 구매, 임대, 리스 등)하기 위한 요청(120)을 원격 시스템(106)에 송신할 수 있다. 서브블록 804에 의해 도시된 바와 같이, 요청(120)은, 원격 시스템(106)이 클라이언트 시스템 컨피규레이션에 대한 정확한 다운로드 시퀀스 데이터(114)를 검색하도록 클라이언트 머신(104)의 컨피규레이션을 포함할 수 있다.

[0072] 806에서, 클라이언트 머신(104)은, 비디오 게임을 플레이하도록 클라이언트 머신(104)의 프로세서(들)(202)에 의해 실행 가능한 실행 코드(예를 들어, 하나 이상의 파일)와 같은 비디오 게임의 게임 실행 파일(들)(122)을 원격 시스템(106)으로부터 수신할 수 있다. 서브블록 808에 의해 도시된 바와 같이, 클라이언트 머신(104)은 또한, 클라이언트 머신(104)의 컨피규레이션과 연관된 다운로드 시퀀스 데이터(114)를 수신할 수 있다. 이러한 다운로드 시퀀스 데이터(114)는 비디오 게임을 위한 게임 데이터(126)의 블록들(124)의 시퀀스를 상술할 수 있다.

[0073] 810에서, 클라이언트 머신(104)은, 다운로드-시퀀스 데이터(114)에 상술된 시퀀스에 따라, 비디오 게임을 위한 게임 데이터(126)의 블록들(124)을 클라이언트 머신(104)의 비휘발성 메모리(예를 들어, 제1 메모리 132(1))로 다운로드하기 시작할 수 있다.

[0074] 812에서, 클라이언트 머신(104)은 비디오 게임을 시작하기 위해 게임 실행 파일(122)을 실행할 수 있다. 사용자(102)가 비디오 게임을 시작할 수 있고, 게임 실행 파일(122)이 게임 데이터(126)의 제1 블록(124) 다운로드가 완료되기 전에도 실행을 시작할 수 있다는 점이 이해될 것이다. 게임 실행 파일의 실행은 비디오 게임을 시작하기 위해 클라이언트 머신(104)에 의해 검출된 사용자 입력에 응답하여 발생할 수 있다. 예를 들어, 클라이언트 머신(104)의 사용자는, 사용자가 원격 시스템(106)을 통해 방금 획득한 비디오 게임을 시작하기 위해 비디오 게임 클라이언트(212)의 사용자 인터페이스 요소를 선택할 수 있다. 언급된 바와 같이, 일부 실시형태에서, 원격 시스템(106)은, 예컨대, "최선의 사용자 경험을 위해서, 다운로드를 시작한 후 T분 정도를 [비디오 게임 X]를 플레이하기 위해 기다리는 것이 권장됩니다(For the best user experience, we recommend waiting T minutes after starting the download to play [Video Game X])"라는 권장 사항을 표시함으로써, 클라이언트 머신(104)을 통해 사용자(102)에게 권장 사항을 출력하기 위해 클라이언트 머신(104)에 데이터를 보낼 수 있다. 일부 실시형태에서, 비디오 게임 클라이언트(212)는, 최적의 사용자 경험을 제공하도록 임계 시간이 경과한 이후까지 그리고/또는 임계 수의 블록(124)이 다운로드를 완료할 때까지 게임 실행 파일(122)의 실행을 방지할 수 있다. 대안적으로, 비디오 게임이 특히 인스턴트 플레이(128) 기능에 적합하다면, 원격 시스템(106)은 클라이언트 머신(104)에게, "이 게임은 인스턴트 플레이를 위한 준비가 되어 있으므로, 지금 바로 플레이를 시작할 수 있습니다. 즐거운 시간 되세요!(This game is ready for instant play, so you can start playing right now. Enjoy!)"라는 알림을 출력하도록 지시할 수 있고/있거나, 비디오 게임 클라이언트(212)는, 사용자(102)가 비디오 게임을 시작할 수 있는 시기에 어떠한 제한도 두지 않을 수 있다. 사실, 비디오 게임을 시작할 때(예를 들어, 월드(world)를 탐험하고 설정을 할 때) 약간의 레이턴시가 문제가 되지 않는 일부 플레이어에게는, 즉시, 가능하게는 게임 데이터(126)가 다운로드를 완료하기 전에 플레이를 시작하는 것이 더 나을 수 있다.

- [0075] 814에서, 클라이언트 머신(104)의 파일 시스템 프록시 컴포넌트(218)는 클라이언트 머신(104)의 파일 시스템(220)에 대한 게임 실행 파일(122)에 의해 행해지는 제1 판독 작동을 수신할 수 있다. 이러한 제1 판독 작동은 비디오 게임을 위한 게임 데이터(126)의 제1 블록(124(1))을 판독하도록 요청할 수 있다. 게임 데이터(126)의 다운로드되는 블록 814에서 판독 작동(들)이 수신될 때에 진행 중이기 때문에(매우 큰 비디오 게임을 가정함), 게임 실행 파일(122)이 게임 데이터(126)의 미실현 블록(124)을 판독할 것을 요청할 가능성이 있으며, "미실현"은 게임 데이터의 블록(124)이 아직 클라이언트 머신(104)의 비휘발성 메모리(예를 들어, 제1 메모리(132(1)))에 다운로드되지 않은 것을 의미한다. 일부 실시형태에서, 파일 시스템 프록시 컴포넌트(218)는, 요청된 블록(들)(124)이 비휘발성 메모리에서 실현되는지, 또는 블록(들)(124)이 여전히 다운로드될(예를 들어, 다운로드를 완료할) 필요가 있는지에 대한 결정을 내리기 위해 판독 작동을 일시적으로(예를 들어, 간단히) 차단할 수 있다.
- [0076] 따라서, 816에서, 파일 시스템 프록시 컴포넌트(218)는, 게임 데이터(126)의 제1 블록(124(1))이 클라이언트 머신(104)의 비휘발성 메모리(NVM: non-volatile memory)에 다운로드되었는지 여부(즉, 첫 번째 블록(124(1))이 비휘발성 메모리에서 "실현"되었는지)를 결정할 수 있다. 게임 데이터(126)의 제1 블록(124(1))이 블록 814에서 판독 작동을 수신하는 시점에 클라이언트 머신(104)의 비휘발성 메모리에서 실현된다면, 프로세스(800)는 블록 816으로부터 블록 818로의 "예(YES)" 경로를 따를 수 있다.
- [0077] 718에서, 클라이언트 머신(104)의 파일 시스템 프록시 컴포넌트(218)는 판독 작동을 차단 해제하고, 파일 시스템(220)은 로컬 메모리 리소스(132)로부터 게임 데이터(126)의 제1 블록(124(1))을 판독하기 위해서 사용될 수 있다. 우연히 제1 블록(124(1))이 실현되고, 이를 제1 메모리(132(1))보다 더 빠른 판독 액세스 속도를 제공하는 제2 메모리(132(2)) 또는 제3 메모리(132(3))에 캐시함으로써 프리페치된 경우, 제1 블록(124(1))은 캐시된 더 빠른 메모리(132(2)/(3))로부터 판독될 수 있다. 블록 816에서, 게임 데이터(126)의 제1 블록(124(1))이 클라이언트 머신(104)의 비휘발성 메모리에서 실현되지 않는다면, 이는 제1 블록(124(1))이 비휘발성 메모리(예를 들어, 제1 메모리(132(1)))로의 다운로드를 완료하지 않았음을 의미하며, 프로세스(800)는 블록 816으로부터 블록 820으로의 "아니오(NO)" 경로를 따를 수 있다.
- [0078] 820에서, 클라이언트 머신(104)은 게임 데이터(126)의 제1 블록(124(1))에 대한 제2 요청을 원격 시스템(106)에 송신할 수 있다. 언급된 바와 같이, 파일 시스템 프록시 컴포넌트(218)는 판독 작동을 인터셉트하고 차단할 수 있다. 일부 실시형태에서, 판독 작동과 연관된 호출은 클라이언트 머신(104)의 커널(Kernel) 내에서 인터셉트되어 차단될 수 있고, 파일 시스템 프록시 컴포넌트(218)는, 제1 블록(124(1))이 실현되지 않은 것으로 결정하는 경우, 제1 블록(124(1))을 다운로드하기 위해 비디오 게임 클라이언트(212) 안으로 폴백을 할 수 있다.
- [0079] 822에서, 클라이언트 머신(104)은 컴퓨터 네트워크(108)를 통해 원격 시스템(106)으로부터 게임 데이터(126)의 제1 블록(124(1))을 수신할 수 있다. 게임 데이터(126)의 제1 블록(124(1))의 수신에 이어, 게임 데이터(126)의 제1 블록(124(1))은 파일 시스템(220)을 사용하여 판독될 수 있다. 따라서, 파일 시스템 프록시 컴포넌트(218)는 적어도 게임 데이터(126)의 실현되지 않은 블록(124)에 대해, 이 블록(124)이 원격 시스템(106)으로부터 수신될 때까지, 판독 작동을 일시적으로 차단할 수 있다. 이것이 미실현 블록(124)을 판독할 때 레이턴시를 추가할 수 있지만, 클라이언트 머신(104)의 비휘발성 메모리에서 이미 실현된 블록(124)을 판독하는 레이턴시와 비교하여, 그럼에도 불구하고, 이 구성은 비디오 게임을 위한 모든 게임 데이터(126)를 다운로드하지 않으면서 비디오 게임이 플레이되는 것을 허용한다. 사실, 이것은, 어떠한 게임 데이터(126)도 클라이언트 머신(104)에 저장되지 않고서, 비디오 게임이 플레이되는 것을 허용한다. 그러나, 컴퓨터 네트워크(108)를 통해 게임 데이터(126)의 블록(124)을 검색함으로써 레이턴시가 추가되기 때문에, 게임 실행과 병렬로 게임 데이터(126)를 다운로드하는 동작은, 게임 데이터(126)의 블록(124)의, 전부는 아니더라도, 적어도 일부가, 게임 실행 파일(122)이 이러한 블록(124)을 판독할 것을 요청할 때까지 비휘발성 메모리에서 실현되어, 레이턴시를 감소시키는 것을 보장한다. 즉, 원격 시스템(106)은, 동일한 클라이언트 시스템 컨피그레이션을 갖는 다른 클라이언트 머신(104)으로부터 수신한 액세스 데이터(110)에 기초하여, 다운로드 시퀀스를 지능적으로 결정하기 때문에, 게임 데이터(126)의 미실현 블록(124)을 판독할 것을 요청하는 판독 작동을 수신할 가능성은 상대적으로 낮고, 발생한다고 하더라도, 자주 발생하지 않을 것으로 예상된다. 대신, 주어진 블록(124)에 대한 판독 작동이 게임 실행 파일(122)에 의해 수행될 때까지, 주어진 블록(124)이 비휘발성 메모리에 이미 다운로드되고 실현되어, 게임 실행 파일(122)이 컴퓨터 네트워크(108)를 통해 원격 시스템(106)으로부터 동일한 블록(124)을 검색하는 데 관련되는 레이턴시보다 더 짧은 레이턴시로 클라이언트 머신(104)의 로컬 메모리 리소스(132)로부터 블록(124)을 판독할 가능성이 크다.
- [0080] 824에서, 비디오 게임을 위한 게임 세션이 종료되어야 하는지에 대한 결정이 이루어진다. 예를 들어, 사용자

(102)가 비디오 게임을 종료하면, 프로세스(800)는 블록 824로부터 블록 826으로의 "예" 경로를 따를 수 있으며, 블록 826에서 비디오 게임 클라이언트(212)가 게임 실행 파일(122)의 실행을 중지할 수 있다. 블록 824에서, 게임 세션이 종료되어서는 안 되는 경우(예를 들어, 사용자(102)가 게임을 계속 플레이하는 경우), 프로세스(800)는 블록 824로부터 블록 814로의 "아니오" 경로를 따를 수 있으며, 블록 814에서 추가 판독 작동이 수신될 수 있다. 프로세스(800)의 일 부분은 사용자(102)가 게임을 계속 플레이할 때 이러한 방식으로 반복될 수 있으며, 게임 데이터(126)의 미실현 블록(124)은, 필요한 경우, 요구 시 원격 시스템(106)으로부터 검색된다. 파일 시스템 프록시 컴포넌트(218)가 비휘발성 메모리에서 사용 가능한 블록(124)에 대해 비디오 게임에 "라이(lie)"하는 이러한 접근 방식을 사용하여, 비디오 게임의 기준 프레임으로부터, 게임의 콘텐츠가 제1 메모리(132(1))에 저장된 것처럼 보인다는 것이 이해될 수 있다. 비디오 게임은, 게임 데이터(126)가 로컬 메모리 리소스(132)로부터 사용 가능하다고 예상할 수 있고, 클라이언트 머신(104)으로부터 클라이언트 시스템(104)으로의 가변 판독 액세스 속도에 익숙해질 수 있거나, 또는 다양한 인자(예를 들어, 다른 런닝 프로세스에 의해 소비되는 리소스 등)에 기초하여, 단일 클라이언트 머신(104)에 대한 판독 액세스 속도의 가변성이 있을 수 있다. 이것은 모두, 프로세스(800)의 접근이 광역 네트워크(108)를 통해 게임 데이터(124)의 미실현 블록(124)을 검색하기 위해, 때때로, 약간 느릴 가능성이 있음에도 불구하고, 비디오 게임으로 하여금 크래시되게 하지 않을 것이라는 것을 의미한다.

[0081] 프로세스(800)를 사용하여, 사용자(102)는 게임을 획득한 후 자신이 원하는 즉시 비디오 게임의 플레이를 시작할 수 있다. 또한, 프로세스(800)의 기술은 게임 애그노스틱(agnostic)이며, 이는, 게임 개발자가 인스턴트 플레이(128) 기능을 허용하기 위해 어떤 작업도 할 필요가 없음을 의미하며, 오히려, 이것은, 비디오 게임을 클라이언트 시스템 컨피규레이션의 이기종 모집단(heterogenous population)에 배포하는 비디오 게임 배포 플랫폼 상의 모든 게임에 대한 밸류어드(value add)이다.

[0082] 도 9는 게임플레이 동안 레이턴시를 감소시키기 위해 게임 데이터(126)의 블록(124)을 프리페치하기 위한 예시적인 프로세스(900)의 흐름도이다.

[0083] 902에서, 클라이언트 머신(104)은 원격 시스템(106)으로부터 클라이언트 머신(104)의 구성과 연관된 블록 종속성 데이터(116)를 수신할 수 있다. 이러한 블록 종속성 데이터(116)는 비디오 게임을 위한 게임 데이터(126)의 둘 이상의 블록(124) 사이의 개별 연관성을 상술할 수 있다.

[0084] 904에서, 클라이언트 머신(104)은 클라이언트 머신(104) 상에서 비디오 게임을 시작하기 위해 비디오 게임의 게임 실행 파일(122)을 실행할 수 있다. 게임 실행 파일(122)은, 비디오 게임이 클라이언트 머신(104) 상에 인스톨되어 있기 때문에, 클라이언트 머신(104) 상에서 이미 사용 가능하다. 게임 실행 파일(122)의 실행은 클라이언트 머신(104)의 사용자(102)가 게임의 플레이를 시작하는 것을 허용한다. 또한, 게임 실행 파일(122)의 실행은 비디오 게임을 시작하기 위해 클라이언트 머신(104)에 의해 검출된 사용자 입력에 응답하여 발생할 수 있다. 예를 들어, 클라이언트 머신(104)의 사용자(102)는 비디오 게임을 시작하기 위해 비디오 게임 클라이언트(212)의 사용자 인터페이스 요소를 선택할 수 있다.

[0085] 906에서, "프리페치 캐시"에 임계값 초과 용량이 있는지 여부에 관한 결정이 이루어질 수 있다. 본원에서 사용되는 "프리페치 캐시"는, 게임 데이터(126)의 프리페치된 블록(124)을 위해서 할당된(또는 예약된) 캐시를 의미한다. 본원에 설명된 바와 같이, 제1 메모리(132(1))보다 더 빠른 판독 액세스 속도를 제공하는 제2 메모리(132(2)) 및/또는 제3 메모리(132(3))는 프리페치 캐시를 제공할 수 있고, 로컬 메모리(132(2)/(3)) 중 하나 또는 둘 모두의 프리페치 캐시는 100메가바이트와 같은 임의의 적절한 양의 데이터로 설정될 수 있다. 본 실시예에서, 게임 실행 파일(122)이 프리페치된 게임 데이터(126)를 판독할 것을 요청하기 전에, 게임 세션 동안 100메가바이트만큼의 게임 데이터(126)가 프리페치될 수 있다. 100메가바이트는 단지 일례일 뿐이고, 프리페치 캐시는 더 적거나 더 많은 양의 데이터로 설정될 수 있다. 따라서, 블록 906에서, 프리페치 캐시(들)가 얼마나 가득 차 있는지를 결정하기 위해 프리페치 캐시(들)가 모니터링될 수 있고, 캐시 용량이 임계 양을 충족하거나 초과하는 경우, 게임 데이터(126)의 추가 프리페치가 트리거될 수 있다. 블록 906에서 평가되는 프리페치 캐시(들)의 이러한 임계 양의 용량은 임의의 적절한 양으로 설정될 수 있다. 예를 들어, 게임 데이터(126)의 개별 블록(124)의 크기가 4096바이트인 경우, 임계 양의 용량은, 게임 데이터(126)의 블록(124)을 프리페치하기에 충분한 공간인 4096바이트의 용량만큼 적게 설정될 수 있다. 그러나, 임계 양의 용량은, 프리페치 알고리즘을 통한 실행 동안, 다수의 블록(124)이 프리페치될 수 있는 것을 보장하기 위해 더 큰 수의 바이트/메가바이트로 설정될 수 있다.

[0086] 예시적인 사용 사례에서, 클라이언트 머신(104)의 사용자(102)는 게임의 라이브러리 페이지로 네비게이트했을

수 있고, 게임 데이터(126)의 일 세트의 블록(124)은 클라이언트 머신(104)의 제2 메모리(132(2)) 및/또는 제3 메모리(132(3))의 100 메가바이트의 프리페치 캐시를 채우도록 프리페치되었을 수 있다. 사용자(102)가 비디오 게임을 계속 플레이함에 따라, 파일 시스템(220)은, 프리페치된 블록(124) 중 하나 이상을 판독하기 위해 게임 실행 파일(122)에 의해 행해지는 판독 작동과 연관된 호출을 수신할 수 있다. 이러한 블록(124)을 검색하기 위해 클라이언트 머신(104)의 제1 메모리(132(1))에 액세스하는 대신, 파일 시스템(220)은, 프리페치된 블록(132)이 캐시되는 제2 메모리(132(2)) 또는 제3 메모리(132(3)) 중 적어도 하나에 액세스할 수 있고, 결과적으로, 프리페치 캐시(들)의 용량이 증가하기 시작한다. 프리페치 캐시의 이 용량이 임계값을 충족하거나 초과하면, 프리페치 캐시를 채우기 위해 추가 프리페치가 트리거될 수 있다. 따라서, 블록 906에서, 프리페치 캐시(들)에 임계 양 초과 용량이 있는 경우, 프로세스(900)는 블록 906으로부터 블록 908로의 "예" 경로를 따를 수 있다.

[0087] 908에서, 클라이언트 머신(104)의 로직은, 프리페치할 게임 데이터(126)의 블록(들)(124)을 결정하기 위해서 사용되는 이벤트(들)를 검출할 수 있다. 예를 들어, 서브블록 910에서, 검출된 이벤트는 컨텍스트 큐(예를 들어, 사용자가 비디오 게임의 라이브러리 페이지로 네비게이트하는 것)일 수 있다. 다른 실시예로서, 서브블록 912에서, 검출된 이벤트는, 게임 데이터(126)의 특정 블록(124)을 판독하기 위해 게임 실행 파일(122)에 의해 행해지는 판독 작동을 수신하는 파일 시스템 프록시 컴포넌트(218)일 수 있다.

[0088] 914에서, 클라이언트 머신(104)의 로직은 블록 중속 데이터(116)를 사용하여, 검출된 이벤트와 연관된 게임 데이터(126)의 하나 이상의 블록(124)을 식별할 수 있다. 예를 들어, 블록 908에서 검출된 이벤트가, 게임 실행 파일(122)이 게임 데이터(126)의 블록 A를 판독할 것을 요청한 것인 경우, 그리고 블록 중속성 데이터(116)가, 블록 B가 블록 A의 액세스로부터 임계 시간 기간 내에 게임 실행 파일(122)에 의해 액세스될 가능성이 있다고 나타내는 경우, 로직은 블록 914에서 블록 B를 식별할 수 있다.

[0089] 916에서, 식별된 블록(들)(124)은, 게임 데이터(126)가 클라이언트 머신(104) 상에 지속되는 제1 메모리(132(1))(예를 들어, HDD, SD 카드 등과 같은 비휘발성 메모리)보다 더 빠른 판독 액세스 속도를 제공하는 로컬 메모리(132)에 캐시될 수 있다. 따라서, 식별된 블록(들)(124)은 제 2 메모리(132(2))(예를 들어, SSD와 같은 추가 비휘발성 메모리) 또는 제 3 메모리(132(3))(예를 들어, RAM과 같은 휘발성 작업 메모리) 중 적어도 하나에 캐시될 수 있다. 서브블록 918에 의해 도시된 바와 같이, 게임 데이터(126)의 다수의 블록(124)이 914에서 식별되면, 블록 124은, 예컨대, 제2 메모리(132(2))에 제2 블록(124(2))을 캐싱하는 것, 제3 메모리(132(3))에 제3 블록(124(3))을 캐싱하는 것 등등에 의해서, 다수의 상이한 프리페치 캐시에 캐싱될 수 있다. 이러한 방식으로, 게임 데이터(126)의 프리페치된 블록들(124)은 로드 밸런싱(load balancing) 기술을 사용하여 다수의 로컬 메모리 리소스(132)의 다수의 프리페치 캐시에 걸쳐 분산될 수 있으며, 여기서 일부 게임 데이터(126)는 제2 메모리(132(2))에 캐시되고 대략 동일한 양의 게임 데이터(126)는 제3 메모리(132(3))에 캐시되지만, 제3 메모리(132(3))가 제2 메모리(132(2))보다 더 빠른 판독 액세스 속도를 제공한다. 이러한 방식으로, 프리페치된 게임 데이터(126)는 로드 밸런싱되어, 게임 데이터(126)가 게임 플레이 동안 병렬로 임의의 모든 로컬 메모리 리소스(132)로부터 액세스될 수 있다는 점에서 시스템의 전체 처리량을 개선할 수 있다. 이것은, 이러한 로드 밸런싱 방식에서, 판독 작동의 1%가 제1 메모리(132(1))로부터 게임 데이터(126)에 액세스할 것으로 예측되고, 판독 작동의 9%가 제2 메모리(132(2))로부터 게임 데이터(126)에 액세스할 것으로 예측되고, 판독 작동의 90%가 제3 메모리(132(3))로부터 게임 데이터(126)에 액세스할 것으로 예측되는 경우일 수 있다.

[0090] 920에서, 클라이언트 머신(104)의 파일 시스템 프록시 컴포넌트(218)는 파일 시스템(220)에 대한 게임 실행 파일(122)에 의해 행해지는 판독 작동을 수신할 수 있으며, 판독 작동은 비디오 게임을 위한 게임 데이터(126)의 특정 블록(124)을 판독할 것을 요청한다.

[0091] 922에서, 요청된 블록(124)이 비교적 빠른 판독 액세스 속도를 제공하는 로컬 메모리 리소스(예를 들어, 제2 메모리 132(2) 또는 제3 메모리 132(3))의 프리페치 캐시에 캐시되는지 여부에 대한 결정이 내려질 수 있다(예를 들어, 파일 시스템(220)에 의해). 블록(124)이 프리페치 캐시에 캐시되면, 프로세스(900)는 블록 922로부터 블록 924로의 "예" 경로를 따를 수 있으며, 블록 924에서 블록(124)이 프리페치 캐시로부터 판독될 수 있다. 예를 들어, 게임 실행 파일(122)은, 블록(124)이 캐시된 위치에 따라, 제2 메모리(132(2)) 또는 제3 메모리(132(3))로부터 게임 데이터(126)의 블록(124)을 판독할 수 있다. 블록(124)이 프리페치 캐시에 캐시되면, 프로세스(900)는 블록 922로부터 블록 926로의 "아니오" 경로를 따를 수 있으며, 블록 926에서 블록(124)은, 게임 데이터(126)가 클라이언트 머신(104) 상에 지속될 수 있는 제1 메모리(132(1))로부터 판독될 수 있다.

[0092] 928에서, 비디오 게임을 위한 게임 세션이 종료되어야 하는지에 대한 결정이 이루어진다. 예를 들어, 사용자(102)가 비디오 게임을 종료하면, 프로세스(900)는 블록 928로부터 블록 930으로의 "예" 경로를 따를 수

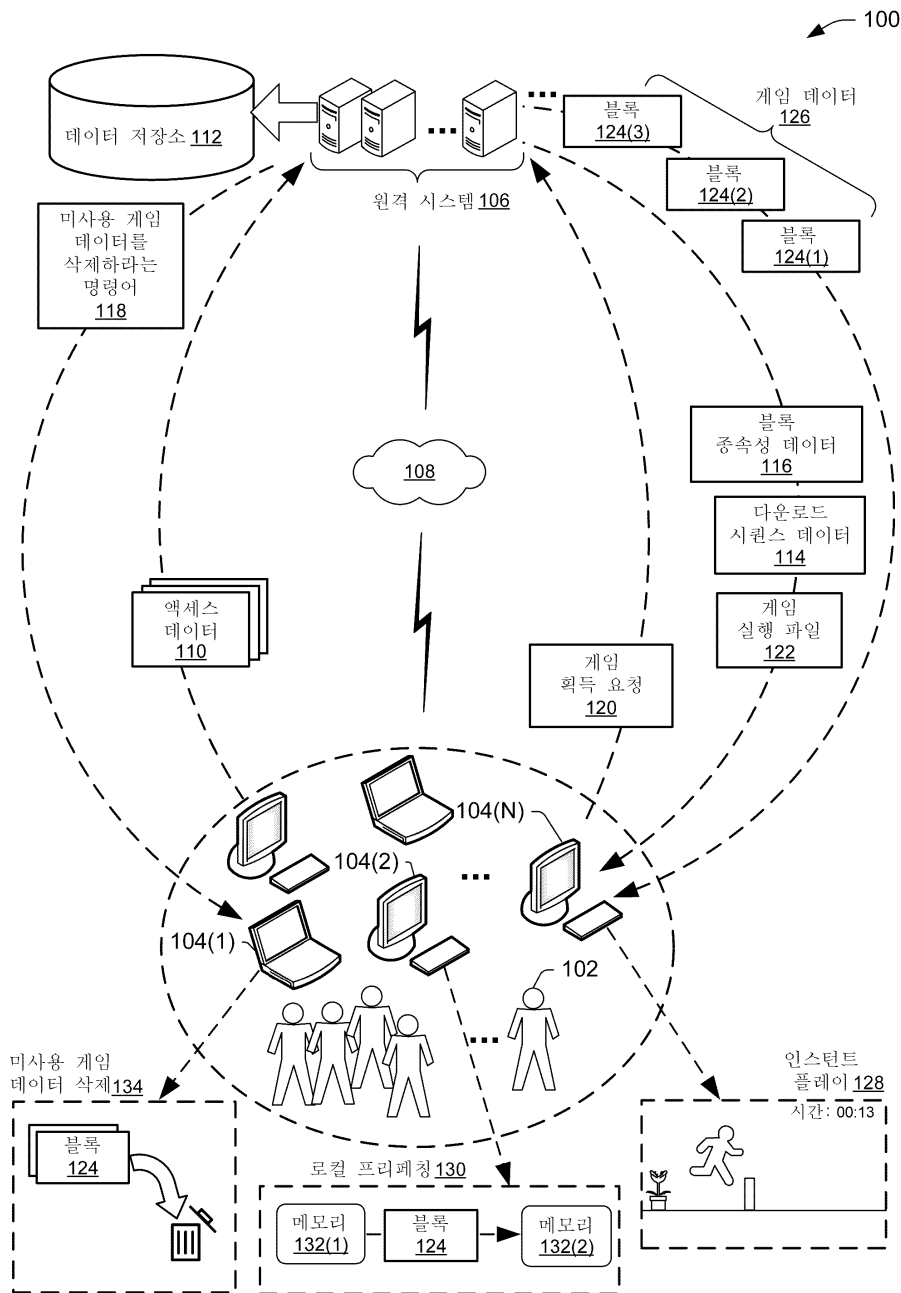
있으며, 블록 930에서 비디오 게임 클라이언트(212)가 게임 실행 파일의 실행을 중지할 수 있다. 일부 실시형태에서, 사용자(102)가 게임 세션을 종료하고 클라이언트 머신(104)을 재부팅하면, 제2 메모리(132(2))로 프리페치된 게임 데이터(126)는, 제2 메모리(132(2))가 추가의 비휘발성 메모리(예를 들어, SSD)를 나타내는 경우, 재부팅 시 제2 메모리(132(2))의 프리페치 캐시에 여전히 저장되어 있을 수 있다. 이러한 방식으로, 사용자(102)가 재부팅 후 다른 게임 세션을 시작하면, 제2 메모리(132(2))에 여전히 캐시되어 있는 게임 데이터(126)의 블록들(124)은, 후속 게임 세션 동안 게임 실행 파일(122)에 의해 요청되는 경우, 제2 메모리로부터 검색될 수 있다. 블록 928에서, 게임 세션이 종료되어서는 안 되는 경우(예를 들어, 사용자(102)가 게임의 플레이를 계속하는 경우), 프로세스(900)는 블록 928로부터 블록 906으로의 "아니오" 경로를 따를 수 있으며, 블록 906에서 프리페치 캐시(들)의 용량이, 블록 906로부터 시작하여 프로세스(900)의 일부를 반복하기 위해 다시 평가된다. 906에서, 프리페치 캐시(들)에 임계 양 초과 용량이 없다면(예를 들어, 프리페치 캐시(들)이 프리페치된 게임 데이터(126)로 너무 가득 찬 것으로 간주되는 경우), 프로세스(900)는 블록 906으로부터 직접 블록 920으로의 "아니오" 경로를 따를 수 있으며, 블록 920에서, 판독 작동은, 프리페치 캐시(들)에서 게임 데이터(126)를 프리페치하지 않으면서, 파일 시스템 프록시 컴포넌트(218)에 의해 수신된다.

[0093] 프로세스(900)는, 게임 데이터(126)가 클라이언트 머신(104) 상에 지속되는 제1 메모리(132(1))로부터 독립적으로 게임 데이터(126)에 액세스하는 것과 비교하여, 전반적인 레이턴시를 감소시킴으로써 게임 실행 동안 로딩 시간을 감소시킬 수 있다. 이러한 방식으로, 비교적 큰 비디오 게임이 여전히 제1 메모리(132(1))(예를 들어, HDD, SD 카드 등)에 저장될 수 있으나, 게임 데이터(126)는, 게임 실행 파일(122)에 의해 블록(124)이 요청되기 전에 제3 메모리(132(2))(예를 들어, RAM과 같은 비휘발성 작업 메모리) 및/또는 제2 메모리(132(2))(예를 들어, SSD)에 게임 데이터(126)의 블록(124)을 캐싱함으로써 지능적으로 프리페치될 수 있기 때문에, 게임 실행 파일(122)이 게임 데이터(126)의 블록(124)을 판독할 것을 요청할 때까지, 블록(124)은 판독 액세스 시간을 감소시키도록 프리페치 캐시에 이미 캐시되었을 것이다.

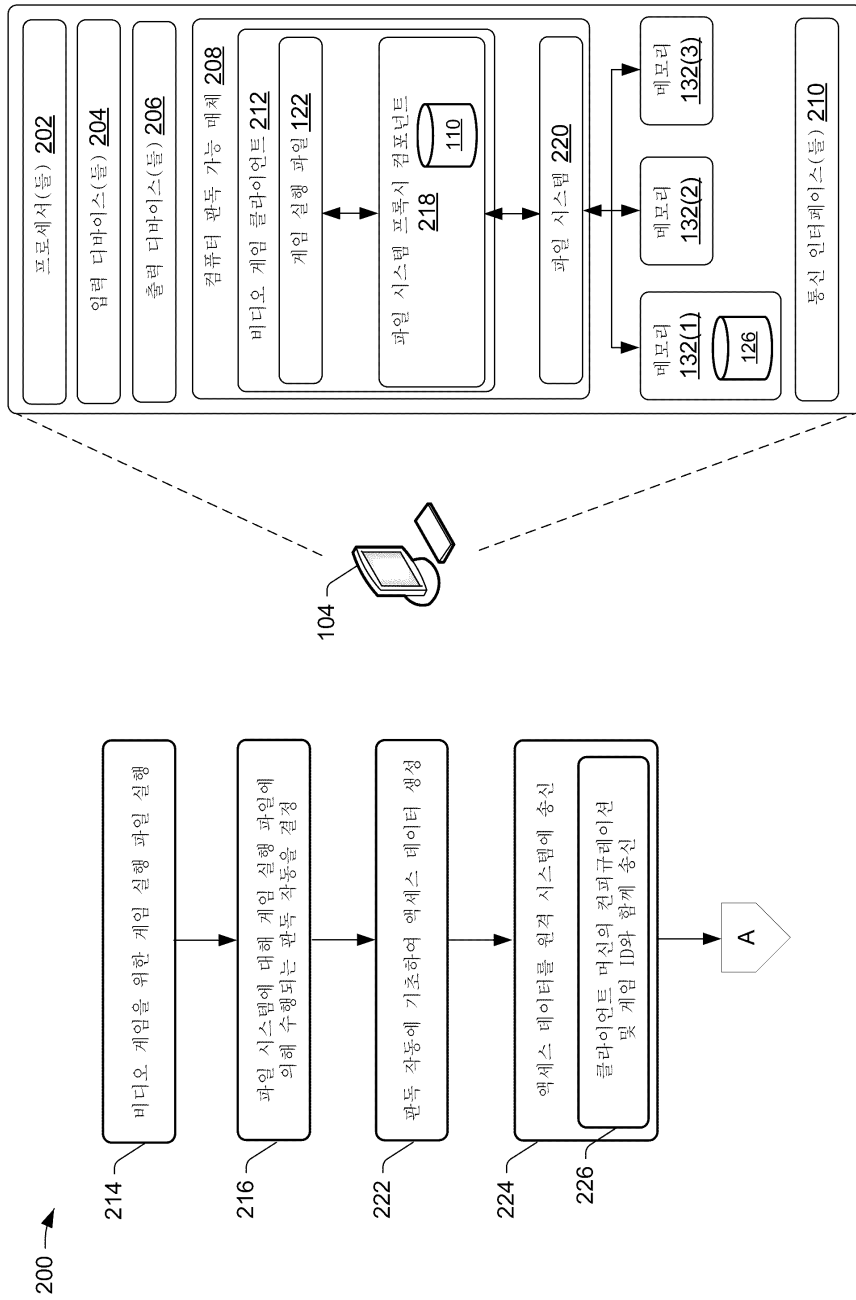
[0094] 첨부된 청구범위에 정의된 특허 대상은 구조적 특징들에 특정되는 술어로 설명되었지만, 설명된 이 특정 특징들에 반드시 한정되지 않는다는 것이 이해될 것이다. 오히려, 특정 특징들은 청구범위를 구현하는 예시적인 형태로서 개시된다.

도면

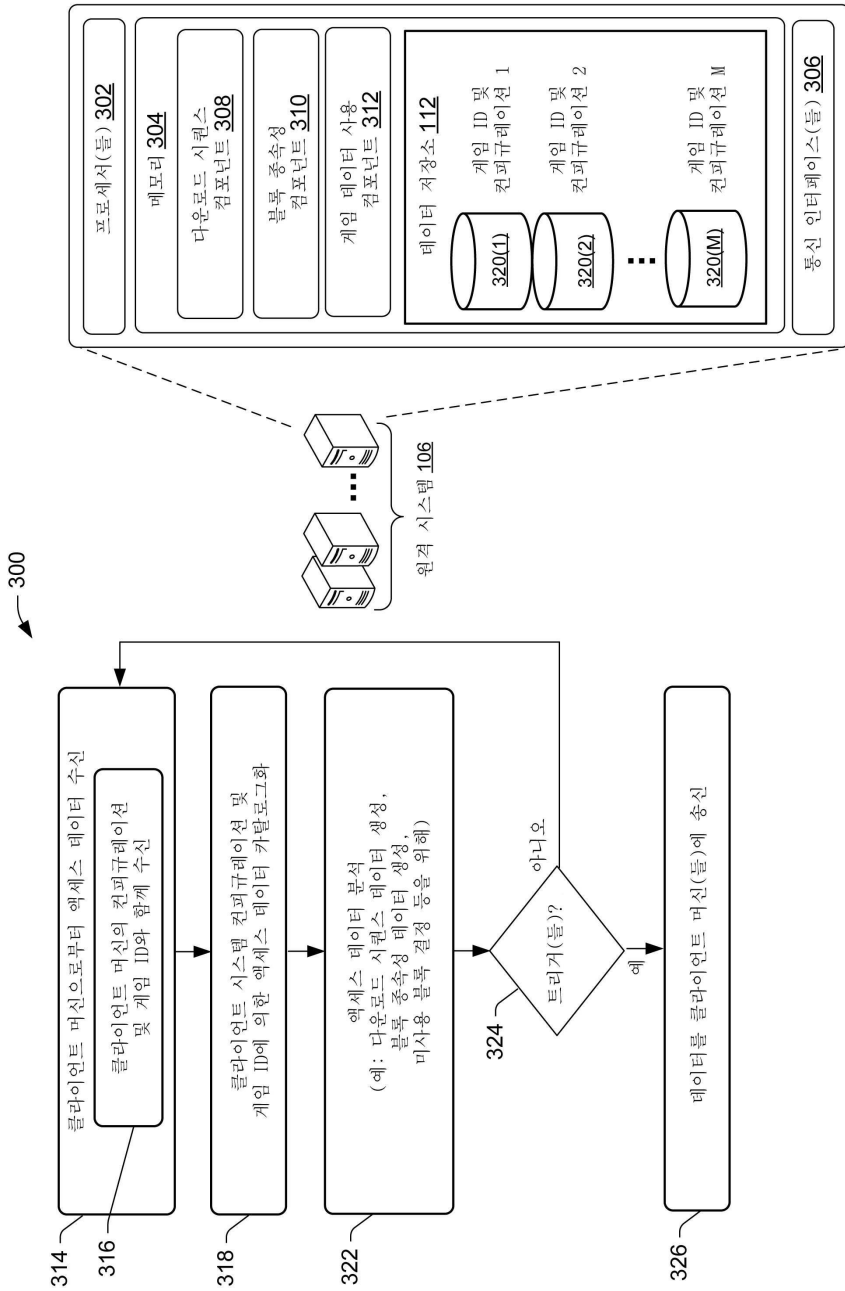
도면1



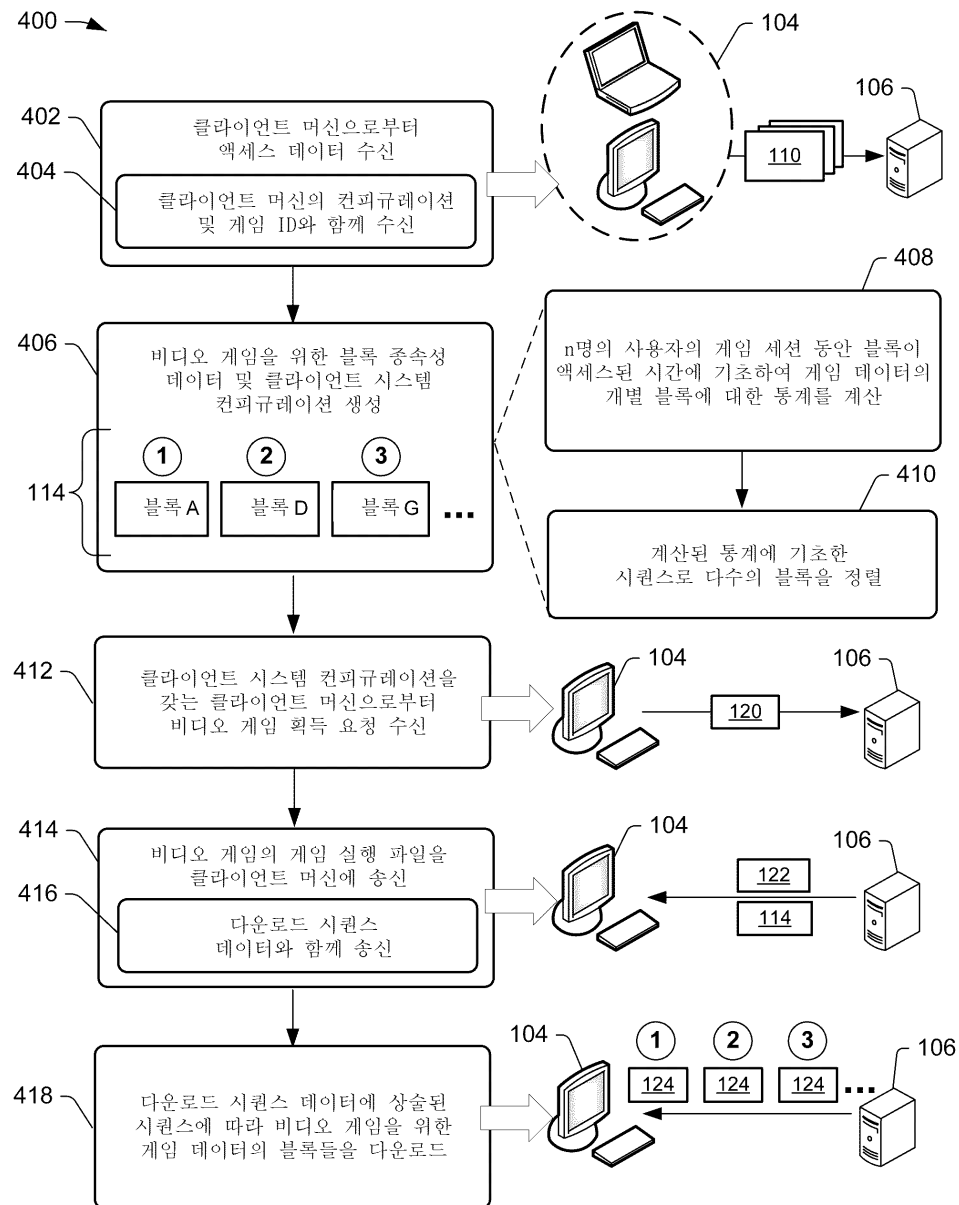
도면2



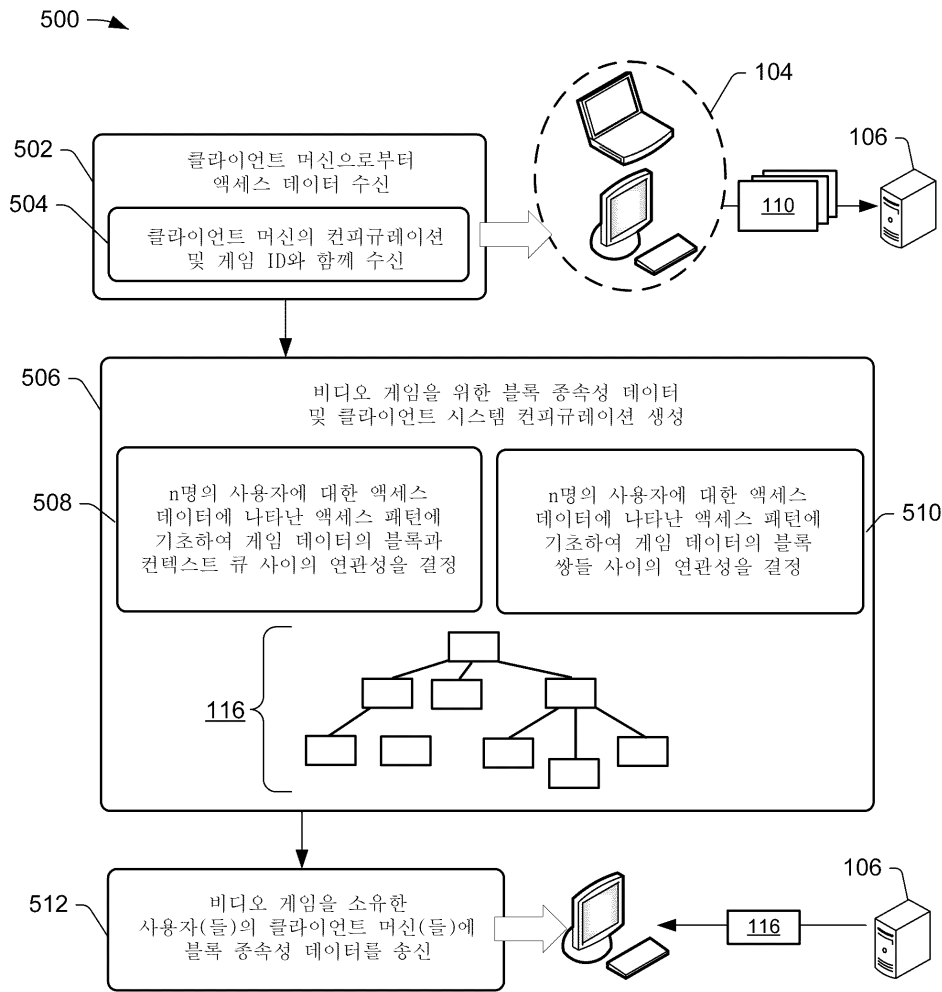
도면3



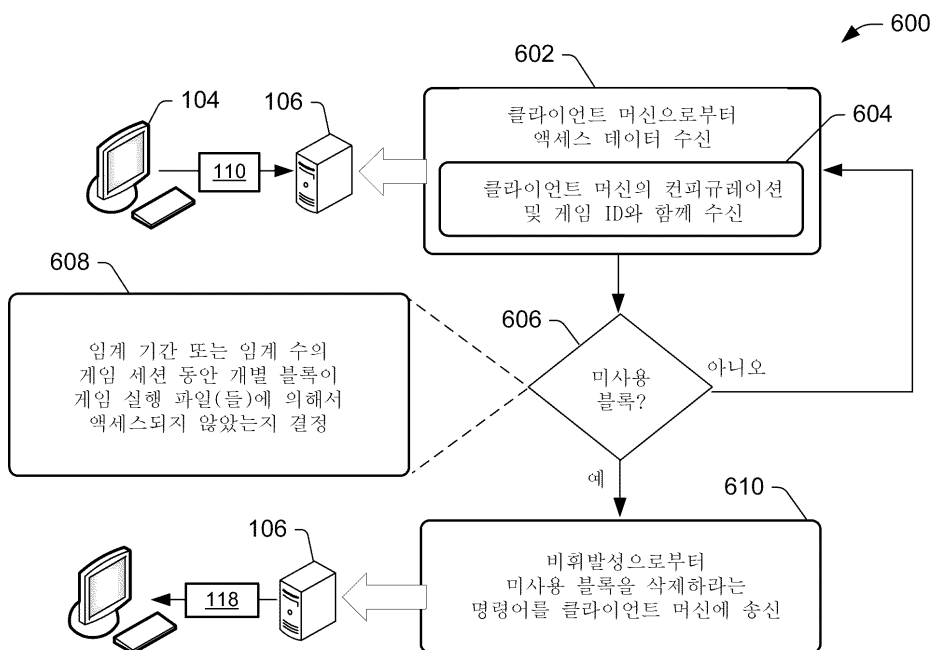
도면4



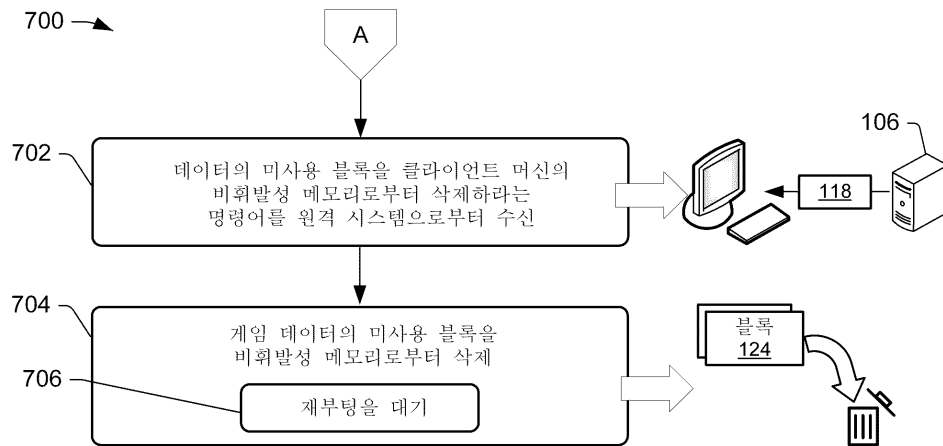
도면5



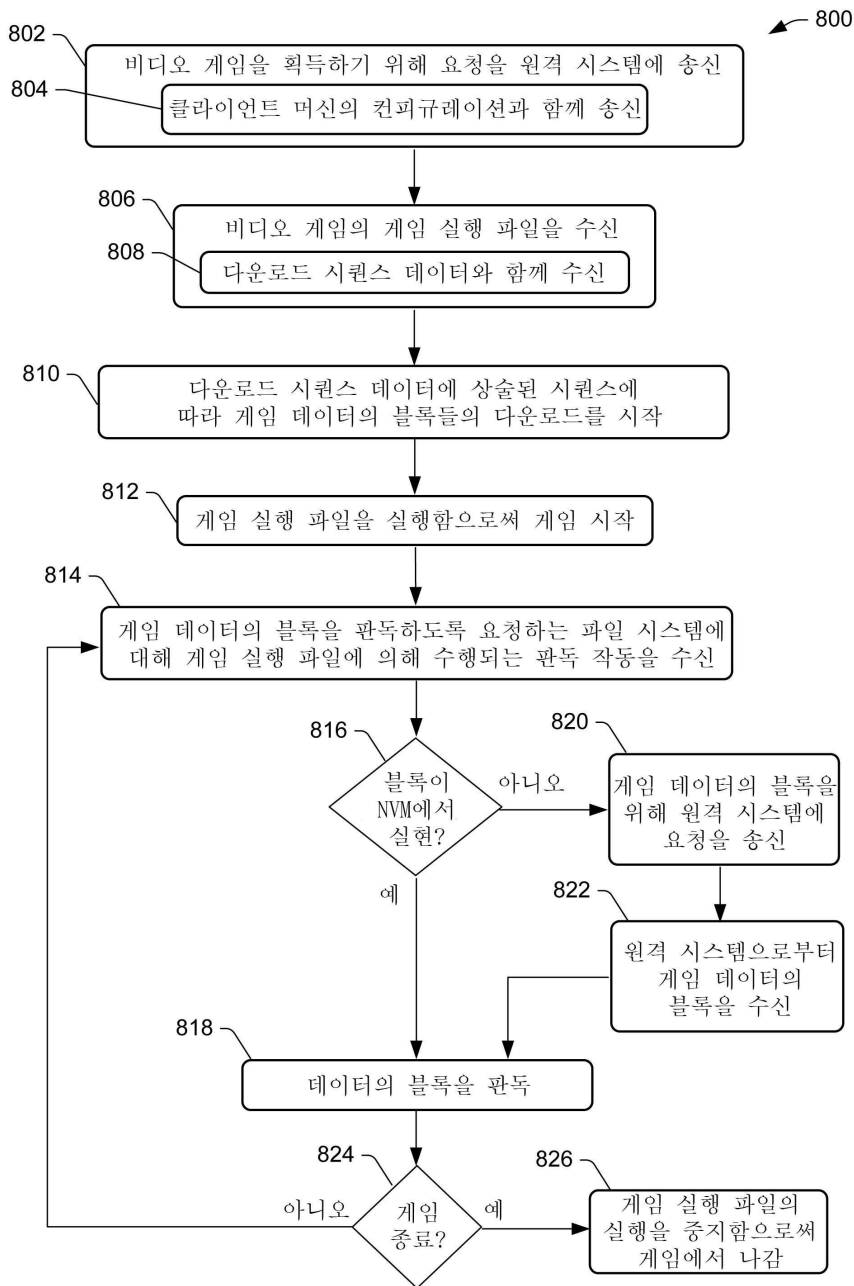
도면6



도면7



도면8



도면9

