



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2021년11월29일

(11) 등록번호 10-2332809

(24) 등록일자 2021년11월25일

(51) 국제특허분류(Int. Cl.)

G06F 9/54 (2018.01) G06F 9/50 (2018.01)

H04L 29/06 (2006.01) H04L 29/08 (2006.01)

(52) CPC특허분류

G06F 9/542 (2013.01)

G06F 9/5005 (2013.01)

(21) 출원번호 10-2017-7000516

(22) 출원일자(국제) 2015년07월06일

심사청구일자 2020년06월08일

(85) 번역문제출일자 2017년01월06일

(65) 공개번호 10-2017-0031131

(43) 공개일자 2017년03월20일

(86) 국제출원번호 PCT/US2015/039170

(87) 국제공개번호 WO 2016/007399

국제공개일자 2016년01월14일

(30) 우선권주장

14/326,239 2014년07월08일 미국(US)

(56) 선행기술조사문헌

US06829770 B

(뒷면에 계속)

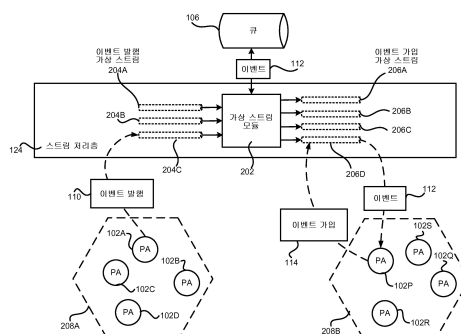
전체 청구항 수 : 총 20 항

심사관 : 유진태

(54) 발명의 명칭 가상 스트림 및 처리 에이전트를 이용한 스트림 기반 이벤트 처리 기법

(57) 요약

본 명세서에서는 가상 처리 에이전트를 이용해서 이벤트를 전달하고 스트림을 처리하는 기술이 개시된다. 큐에서 이벤트 발행을 수신하면, 런타임 시스템은, 발행된 이벤트에 관심은 있지만, 이 발행된 이벤트에 명백하게 가입하지는 않은 하나 이상의 가상 처리 에이전트를 식별한다. 이후, 발행된 이벤트의 이벤트 정보는 식별된 가상 처리 에이전트로 전달된다. 실제 전달 이전에, 런타임 시스템은 또한 가상 처리 에이전트가 가동되었는지 판정하고, 가동되지 않는 처리 에이전트를 가동시킨다. 수신한 이벤트 정보에 기초해서, 가상 처리 에이전트 중 일부는 가입을 명백하게 제시해서 큐로부터 이벤트를 더 수신하도록 결정할 수 있다. 명시적 가입은 런타임 시스템을 트리거해서 가입된 이벤트를 처리 에이전트에 전달하게 하고, 이는 명시적 가입이 수신되기 전에 큐에서 발행된 과거 이벤트를 포함할 수 있다.

대표도

- (52) CPC특허분류
H04L 65/60 (2013.01)
H04L 67/10 (2013.01)
- (72) 발명자
멜라메드 타미르
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 로디 로버트 루이스**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 칭 펠릭스**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 말리우크 마이클 윌리엄**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 히스카이 앤드류 알렉산더**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 클리오토 가브리엘**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 더런 조르겐**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- 겔러 알란 스튜어트**
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패이턴츠
 (8/1172) 마이크로소프트 테크놀로지 라이선싱, 엘 엘씨
- (56) 선행기술조사문헌
 US20090228563 A1
 US20130166580 A1
 US20110213753 A1
 US20080096664 A1

명세서

청구범위

청구항 1

하나 이상의 서버에서 실행되는 처리 에이전트에 이벤트를 전달하는 컴퓨터-구현된 방법으로서,

이벤트의 큐에서 이벤트의 발행(publication)을 수신하는 단계 - 상기 이벤트는 게임 애플리케이션과의 플레이어 상호 작용을 설명하는 정보를 포함함 - 와,

복수의 상이한 처리 에이전트들 중 처리 에이전트에 상기 이벤트를 매핑하는 단계 - 상기 처리 에이전트는 상기 이벤트를 수신하도록 명시적으로 가입되어 있지 않음 - 와,

상기 처리 에이전트가 가동되지 않는지 판정하는 단계와,

상기 처리 에이전트가 가동되지 않는다고 판정하는 것에 응답해서, 상기 처리 에이전트가 상기 게임 애플리케이션에 기능을 제공하는 데 이용가능 하도록 상기 처리 에이전트를 가동시키는 단계와,

상기 이벤트를 상기 처리 에이전트에 전달하는 단계와,

상기 처리 에이전트로부터 가입을 수신하는 단계와,

상기 이벤트의 큐에서 하나 이상의 추가 이벤트의 하나 이상의 추가 발행을 수신하는 단계 - 상기 가입 이후에 발생하는 상기 하나 이상의 추가 이벤트는 상기 처리 에이전트로부터 수신됨 - 와,

상기 가입에 적어도 부분적으로 기초하여, 상기 하나 이상의 추가 이벤트를 상기 처리 에이전트에 전달하는 단계와,

상기 가입이, 하나 이상의 과거 이벤트가 전달되도록 요청하는지를 판정하는 단계 - 상기 하나 이상의 과거 이벤트는 상기 처리 에이전트로부터 상기 가입을 수신하기 전에 상기 이벤트의 큐에서 발행되어 있음 - 와,

상기 가입이 상기 하나 이상의 과거 이벤트가 전달되도록 요청한다고 판정하는 것에 응답해서, 상기 하나 이상의 과거 이벤트를 상기 처리 에이전트에 전달하는 단계를 포함하는

컴퓨터-구현된 방법.

청구항 2

제 1 항에 있어서,

상기 이벤트의 큐에서 상기 이벤트의 발행을 수신하는 단계는,

상기 이벤트의 큐에서 상기 이벤트를 발행하는 요청을 수신하는 단계와,

상기 이벤트에 대응하는 가상 스트림을 가동시키는 단계와,

상기 가상 스트림을 통해서 상기 이벤트의 큐에 상기 이벤트를 발행하는 단계를 포함하는

컴퓨터-구현된 방법.

청구항 3

제 2 항에 있어서,

상기 가입은 상기 가상 스트림을 정의하는

컴퓨터-구현된 방법.

청구항 4

제 3 항에 있어서,

상기 처리 에이전트를 가동시키는 단계는 상기 처리 에이전트에 시스템 리소스를 할당하는 단계를 포함하는 컴퓨터-구현된 방법.

청구항 5

제 1 항에 있어서,

상기 처리 에이전트의 구성은 프로그램적으로 획득되거나 상기 처리 에이전트의 선언적 정의(declarative definition)로부터 결정되는

컴퓨터-구현된 방법.

청구항 6

제 1 항에 있어서,

상기 처리 에이전트는 상기 처리 에이전트가 가동되는지 또는 가동되지 않는지에 관계없이 상기 이벤트를 전달하기 위해 어드레싱 가능한

컴퓨터-구현된 방법.

청구항 7

제 1 항에 있어서,

상기 처리 에이전트는 특정 유형의 이벤트를 수신하도록 구성되고, 상기 가입은 상기 이벤트, 상기 하나 이상의 추가 이벤트 및 상기 하나 이상의 과거 이벤트를 포함하는 상기 이벤트의 큐에서 발행된 상기 특정 유형의 이벤트 전부를 요청하는

컴퓨터-구현된 방법.

청구항 8

제 1 항에 있어서,

가동 테이블을 사용하여 상기 이벤트를 상기 처리 에이전트에 매핑하는 단계를 더 포함하는

컴퓨터-구현된 방법.

청구항 9

제 1 항에 있어서,

상기 처리 에이전트는 상기 게임 애플리케이션의 개별 세션 내에서 발생하는 상이한 이벤트를 수신하도록 구성되는

컴퓨터-구현된 방법.

청구항 10

제 1 항에 있어서,

상기 처리 에이전트는 상기 게임 애플리케이션의 세션에 참여하는 개별 플레이어에 의해 발생한 상이한 이벤트를 수신하도록 구성되는

컴퓨터-구현된 방법.

청구항 11

제 1 항에 있어서,

상기 처리 에이전트는 상기 게임 애플리케이션의 세션에 참여하는 개별 장치에 의해 발생한 상이한 이벤트를 수신하도록 구성되는

컴퓨터-구현된 방법.

청구항 12

명령어를 가진 하나 이상의 컴퓨팅 장치를 포함하는 시스템으로서,

상기 명령어는 하나 이상의 프로세서에 의해 실행될 때 상기 하나 이상의 컴퓨팅 장치가,

이벤트의 발행을 수신하고 유지하는 큐를 관리 - 개별 이벤트는 게임 애플리케이션과의 플레이어 상호 작용을 설명하는 정보를 포함함 - 하고,

가상 스트림에서 이벤트의 발행을 수신하며,

상기 이벤트의 발행을 상기 큐로 포워딩하고,

복수의 상이한 처리 에이전트들 중 처리 에이전트에 상기 이벤트를 매핑 - 상기 처리 에이전트는 상기 이벤트를 수신하도록 명시적으로 가입되어 있지 않고, 상기 처리 에이전트는 상기 처리 에이전트가 가동되는지 또는 가동되지 않는지에 관계없이 상기 이벤트를 전달하기 위해 어드레싱 가능함 - 하며,

상기 처리 에이전트가 가동되지 않는지 판정하고,

상기 처리 에이전트가 가동되지 않는다고 판정하는 것에 응답해서, 상기 처리 에이전트가 상기 게임 애플리케이션에 기능을 제공하는 데 이용가능 하도록 상기 처리 에이전트를 가동시키며,

상기 이벤트를 상기 처리 에이전트에 전달하고,

상기 이벤트가 상기 처리 에이전트에 전달된 후에 상기 처리 에이전트로부터 상기 가상 스트림으로 명시적 가입을 수신하며,

상기 명시적 가입이 수신된 후에 상기 가상 스트림에서 하나 이상의 추가 이벤트의 하나 이상의 추가 발행을 수신하고,

상기 큐에 상기 하나 이상의 추가 이벤트의 상기 하나 이상의 추가 발행을 포워딩하며,

상기 명시적 가입에 적어도 부분적으로 기초하여, 상기 하나 이상의 추가 이벤트를 상기 처리 에이전트에 전달하고,

상기 명시적 가입이 수신되기 전에 상기 큐에서 과거 이벤트가 발행되었는지를 판정하며,

상기 큐에서 발행되었던 상기 과거 이벤트를 불러오고,

상기 과거 이벤트를 상기 처리 에이전트에 전달

하도록 구성되는,

시스템.

청구항 13

제 12 항에 있어서,

상기 처리 에이전트를 가동시키는 것은, 상기 처리 에이전트에 시스템 리소스를 할당하는 것을 포함하는 시스템.

청구항 14

제 12 항에 있어서,

상기 명령어는 또한 상기 하나 이상의 컴퓨팅 장치가,

소정 시구간 동안 상기 가상 스트림에서 어떤 이벤트도 수신되지 않으면 상기 가상 스트림에 할당된 시스템 리소스를 회수함으로써 상기 가상 스트림을 가동 중지시키고,

상기 처리 에이전트가 주어진 시구간 동안 유휴 상태가 되면 상기 처리 에이전트에 할당된 시스템 리소스를 회수함으로써 상기 처리 에이전트를 가동 중지시키도록 구성하는

시스템.

청구항 15

제 12 항에 있어서,

상기 처리 에이전트는 특정 유형의 이벤트를 수신하도록 구성되고, 상기 명시적 가입은 상기 이벤트, 상기 하나 이상의 추가 이벤트 및 상기 과거 이벤트를 포함하는 상기 큐에서 발행된 상기 특정 유형의 이벤트 전부를 요청하는

시스템.

청구항 16

제 12 항에 있어서,

상기 명령어는 또한 상기 하나 이상의 컴퓨팅 장치가 가동 테이블을 사용하여 상기 이벤트를 상기 처리 에이전트에 매핑하도록 구성하는

시스템.

청구항 17

시스템으로서,

하나 이상의 프로세서와,

컴퓨터 실행가능 명령어가 저장된 하나 이상의 컴퓨터 판독가능 저장 매체를 포함하되,

상기 컴퓨터 실행가능 명령어는, 상기 하나 이상의 프로세서에 의해 실행될 때, 상기 시스템으로 하여금,

이벤트의 큐에서 이벤트의 발행을 수신 - 상기 이벤트는 게임 애플리케이션과의 플레이어 상호 작용을 설명하는 정보를 포함함 - 하게 하고,

복수의 상이한 처리 에이전트들 중 처리 에이전트에 상기 이벤트를 매핑 - 상기 처리 에이전트는 상기 이벤트를 수신하도록 명시적으로 가입되어 있지 않음 - 하게 하며,

상기 처리 에이전트가 가동되지 않는지 판정하게 하고,

상기 처리 에이전트가 가동되지 않는다고 판정하는 것에 응답해서, 상기 처리 에이전트가 상기 게임 애플리케이션에 기능을 제공하는 데 이용가능 하도록 상기 처리 에이전트를 가동시키게 하며,

상기 이벤트를 상기 처리 에이전트에 전달하게 하고,

상기 처리 에이전트로부터 가입을 수신하게 하며,

상기 이벤트의 큐에서 하나 이상의 추가 이벤트의 하나 이상의 추가 발행을 수신 - 상기 가입 이후에 발생하는 상기 하나 이상의 추가 이벤트는 상기 처리 에이전트로부터 수신됨 - 하게 하고,

상기 가입에 적어도 부분적으로 기초하여, 상기 하나 이상의 추가 이벤트를 상기 처리 에이전트에 전달하게 하며,

상기 가입이, 하나 이상의 과거 이벤트가 전달되도록 요청하는지를 판정 - 상기 하나 이상의 과거 이벤트는 상기 처리 에이전트로부터 상기 가입을 수신하기 전에 상기 이벤트의 큐에서 발행되었음 - 하게 하고,

상기 가입이 상기 하나 이상의 과거 이벤트가 전달되도록 요청한다고 판정하는 것에 응답해서, 상기 하나 이상의 과거 이벤트를 상기 처리 에이전트에 전달하게 하는

시스템.

청구항 18

제 17 항에 있어서,

상기 처리 에이전트는 상기 게임 애플리케이션의 개별 세션 내에서 발생하는 상이한 이벤트를 처리하는 것을 담당하는

시스템.

청구항 19

제 17 항에 있어서,

상기 처리 에이전트는 상기 게임 애플리케이션의 세션에 참여하는 개별 플레이어에 의해 발생한 상이한 이벤트를 처리하는 것을 담당하는

시스템.

청구항 20

제 17 항에 있어서,

상기 처리 에이전트는 상기 게임 애플리케이션의 세션에 참여하는 개별 장치에 의해 발생한 상이한 이벤트를 처리하는 것을 담당하는

시스템.

발명의 설명

기술 분야

[0001]

스트림 처리 시스템에서 스트림 처리 유닛은 다른 처리 유닛과 통신해서 처리할 이벤트를 수신할 수 있다. 이는, 처리 유닛 사이에서 직접 통신을 통해서 행해질 수도 있고, 혹은 처리 유닛이 특정 이벤트에 가입하고 그 이벤트가 발행되면 이를 수신하는 발행-가입 모델을 통해서 행해질 수도 있다.

배경 기술

[0002] 스트림 처리 시스템의 규모가 예컨대, 수백만개의 처리 유닛이 포함될 정도로 증가됨에 따라서, 처리 유닛 사이에서 직접 통신 기술을 이용하는 것은 불가능할 수 있다. 발행-가입 모델은 실현이 용이해 보일 수 있지만, 종래의 발행-가입 모델에서는, 처리 유닛은, 이벤트와 관련된 어떤 정보를 수신하기 전에 명시적으로(explicitly) 이벤트 가입을 제시해야 한다. 이는, 경우에 따라서는, 그 이벤트가 발생하기 전에는 처리 유닛이 어떤 이벤트에 가입해야 하는지 예측할 수 없기 때문에, 문제가 될 수 있다. 예컨대, 게임 애플리케이션의 맵 상에서 발생하는 이벤트를 처리하도록 구성된 처리 유닛의 경우, 게임이 맵 상에서 시작되기 전까지, 어떤 게임에 가입했는지 알지 못한다. 이런 상황에서 처리 유닛이 관심 있는 이벤트를 수신하기 위해서는 모든 가능 이벤트에 가입해야 하며, 이는 대규모의 스트림 처리 시스템에서는 불가능하고 효율적이지 않다.

발명의 내용

해결하려는 과제

[0003] 본 개시는 이러한 사항을 고려해서 제시되는 것이다.

과제의 해결 수단

[0004] 이하, 가상 처리 에이전트를 이용한 이벤트 전달 및 스트림 처리 기술을 개시한다. 하나 이상의 가상 처리 에이전트가 생성되어서, 스트림 이벤트를 수신해서 처리하도록 구성될 수 있다. 스트림 처리 시스템 내의 다양한 개체로부터 이벤트 발행을 수신하는데, 큐(queue)와 같은 메커니즘이 사용 및/또는 구성될 수 있다. 발행된 이벤트는 또한, 이후에 필요시에 취득 및 전달될 수 있도록 유지 혹은 저장될 수 있다. 이벤트 발행을 수신하면, 런타임 시스템은, 이 발행된 이벤트를 수신하는 것에 관심을 있지만, 발행된 이벤트에 명시적으로 가입하지는 않은 가상 처리 에이전트를 하나 이상 식별할 수 있다. 이후, 이 식별된 가상 처리 에이전트에, 발행된 이벤트에 관한 이벤트 정보가 전달될 수 있다. 런타임 시스템은 또한, 이벤트 혹은 이벤트 정보를 실제로 전달하기 전에, 이 가상 처리 에이전트가 가동되었는지 판정하고, 처리 에이전트가 가동되지 않았다면 처리 에이전트를 가동시킬 수 있다. 이후, 이 가동된 처리 에이전트에 이벤트 혹은 그 안에 포함된 이벤트 정보가 전달될 수 있다.

발명의 효과

[0005] 이 수신한 이벤트 혹은 이벤트 정보에 기초해서, 가상 처리 에이전트는, 가입을 명시적으로 제시해서 큐로부터 하나 이상의 이벤트를 수신하는 것으로 결정할 수 있다. 명시적(explicit) 가입은, 가입된 이벤트가 처리 에이전트로 전달되도록 트리거하며, 여기에는 명시적 가입이 제시되기 전에 큐에서 발행되었던 과거 이벤트가 포함될 수 있다. 이러한 기술을 이용함으로써, 가상 처리 에이전트가 이벤트가 큐에서 발행되기 전에는 알 수 없을 이벤트를 가입, 수신 및 처리할 수 있다. 가상 처리 에이전트를 이용해서 이벤트를 가입, 수신 및 처리함으로써, 컴퓨팅 리소스의 확장 가능성, 신뢰성 및 유연성이 개선될 수 있다.

[0006] 상술한 청구 대상이 컴퓨터-제어형 장치, 컴퓨터 프로세스, 컴퓨팅 시스템으로서, 혹은 컴퓨터 판독 가능 저장매체와 같은 제조 물품으로서 구현될 수도 있다는 점을 이해할 것이다. 이러한 특징 및 다른 특징은 이하의 상세한 설명을 읽고 관련 도면을 참조함으로써 명백할 것이다.

[0007] 본 개요는 이하의 상세한 설명에서 더 설명되는 개념 중 일부를 선택해서 개략적으로 제공하고자 하는 것이다. 본 개요는 청구 대상의 주요한 특성 혹은 필수적인 특징을 나타내는 것이 아니며, 또한 청구 대상의 범주를 한정하는데 사용하고자 하는 것도 아니다. 나아가, 청구 대상은 본 개시 중 일부에서 언급한 단점을 모두 혹은 일부 해결하고 있는 구현예로 한정되는 것은 아니다.

도면의 간단한 설명

[0008] 도 1은 가상 처리 에이전트를 이용해서 스트림 처리하는, 본 명세서에 개시된 하나의 메커니즘을 전체적으로 나타내는 컴퓨터 시스템 도면,

도 2는 큐 및 가상 스트림의 측면들을 나타내는 블록도,

도 3은 가상 처리 에이전트에 이벤트를 전달하는 하나의 예시적인 처리의 다른 측면을 나타내는 흐름도,

도 4는 가상 처리 에이전트에 의한 이벤트 가입 및 처리의 하나의 예시적인 처리의 측면들을 나타내는 흐름도,

도 5a 내지 도 5d는 서로 다른 시점에, 큐에서 이벤트가 발행되는 것에 응답해서 예시적인 가상 처리 에이전트

에 의해 행해지는 동작을 나타내는 도면,

도 6은 본 명세서에 개시된 기술을 구현할 수 있는 컴퓨팅 시스템의 예시적인 컴퓨터 하드웨어 아키텍처를 나타내는 컴퓨터 아키텍처 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0009] 가상 처리 에이전트를 이용해서 이벤트를 전달하고 스트림을 처리하는 기술이 개시된다. 본 명세서에 개시된 기술에서, 하나 이상의 가상 처리 에이전트는 "처리 에이전트" 혹은 "PA"라고도 하며, 스트림 처리를 위한 처리 유닛으로서 사용될 수 있다. 이들 가상 처리 에이전트는 하나 이상의 서버에서 실행될 수 있고, 처리 에이전트가 가동되는지 아닌지 여부와 무관하게, 동작 및 상호 작용을 어드레싱할 수 있다. 각각의 가상 처리 에이전트는 다양한 이벤트를 발행, 수신 및/또는 처리하도록 구성될 수 있다. 이벤트는 큐에서 발행될 수 있고, 이후에 취득될 수 있도록, 큐에 유지 혹은 저장될 수 있다.
- [0010] 일단 이벤트가 발행되어서 큐에서 수신되면, 발행된 이벤트를 수신할 하나 이상의 가상 처리 에이전트가 식별될 수 있다. 식별되는 처리 에이전트에는 발행된 이벤트를 수신하는 것으로 명시적으로 가입하지 않은 처리 에이전트("묵시적(implicit) 가입자 처리 에이전트" 혹은 "묵시적 가입자"라고도 함)가 포함될 수 있다. 묵시적 가입자의 식별은, 비한정 예로서, 가상 처리 에이전트에 의해 수행될 액션, 가상 처리 에이전트의 입력과 출력, 가상 처리 에이전트의 상태, 및 가상 처리 에이전트의 잠재적인 다른 구성을 포함한, 가상 처리 에이전트의 세팅 혹은 구성에 기초해서 수행될 수 있다. 가상 처리 에이전트의 구성은 처리 에이전트의 선언적 정의(declarative definition)로부터 유도될 수도 있고, 혹은 프로그램적으로 획득될 수도 있다.
- [0011] 발행된 이벤트나, 혹은 이 발행된 이벤트에 포함되어 있거나 혹은 발행된 이벤트와 관련된 정보("이벤트 정보"라고도 함)는, 식별된 묵시적 가입자 처리 에이전트에 전달될 수 있다. 실제로 전달되기 전에, 가상 처리 에이전트를 지원하고 관리하는 런타임 시스템이, 묵시적 가입자 처리 에이전트가 가동되었는지 판정하고, 처리 에이전트가 가동되지 않았다면 처리 에이전트를 가동시킬 수 있다. 이후 이벤트 혹은 이벤트 정보가 이 가동된 처리 에이전트에 전달되고, 여기서 이벤트 혹은 이벤트 정보가 더 처리될 수 있다. 유사하게, 이벤트 발행이 큐에서 수신되면, 필요에 따라서, 발행된 이벤트를 수신하도록 명시적으로 가입된 하나 이상의 처리 에이전트가 식별 및 가동될 수 있다. 이후 발행된 이벤트는 처리를 위해서 명시적 가입자 처리 에이전트에 전달될 수 있다.
- [0012] 묵시적 가입자 처리 에이전트는 수신한 이벤트 혹은 이벤트 정보에 기초해서, 이벤트가 처리될 필요가 있는지 판정할 수 있으며, 이들 이벤트가 큐에서 발행될 때 수신하도록 명시적 가입을 제시할 수 있다. 처리 에이전트의 구성에 따라서, 명시적 가입에 명시되어 있는 이벤트에는, 큐에서 이미 발생해서 발행된 과거 이벤트가 포함될 수 있다. 이 시나리오에서는, 큐에서 발행된 이벤트가 유지 혹은 저장되어 있기 때문에, 큐는 이들 가입된 과거 이벤트를 취득해서 처리 에이전트에 전달할 수 있다. 큐는 또한, 이벤트가 큐에서 더 발행됨에 따라서, 가입된 이벤트를 처리 에이전트에 전달할 수 있다.
- [0013] 다른 측면에 따르면, 큐 내의 이벤트는 하나 이상의 가상 스트림으로 매핑 혹은 추출될 수 있으며, 각각의 가상 스트림은 큐에서 발행되는 이벤트의 서브셋을 포함할 수 있다. 처리 에이전트 및/또는 다른 개체에 의해 수행되는 이벤트 발행 및 이벤트 가입은, 큐가 아닌 가상 스트림과 관련될 수 있다. 가상 스트림 각각은, 거기서 발행될 이벤트에 의해서 및/또는 처리 에이전트가 수신에 관심이 있는 타입의 이벤트에 의해서 정의될 수 있다. 이와 같이, 가상 스트림은 유동적일 수 있으며, 가상 스트림의 존재는 발행된 이벤트 및/또는 전달된 이벤트에 따라서 달라질 수 있다. 본 명세서에 개시된 기술의 이러한 측면들 및 다른 측면들과 관련된 추가적인 세부 사항을 도 1 내지 도 6을 참조로 제공한다.
- [0014] 본 명세서에 개시된 청구 대상이, 컴퓨터 시스템 상의 운영 체제 및 애플리케이션 프로그램의 실행과 함께 실행되는 프로그램 모듈과 관련해서 설명되지만, 당업자라면 다른 타입의 프로그램 모듈과 조합해서 다른 구현이 수행될 수 있다는 것을 이해할 것이다. 일반적으로, 프로그램 모듈은 루틴, 프로그램, 컴포넌트, 데이터 구조, 그리고 특정 태스크를 수행하거나 특정한 압축 데이터 타입을 구현하는 다른 타입의 구조를 포함한다. 나아가, 당업자라면 본 명세서에 개시된 청구 대상이, 휴대형 장치, 멀티프로세서 시스템, 마이크로프로세서 기반 혹은 프로그래밍 가능 가전 제품, 미니컴퓨터, 미니프레임 컴퓨터 등을 포함한, 다른 컴퓨터 시스템 구성을 이용해서 실행될 수도 있다는 것을 이해할 것이다.
- [0015] 이하의 상세한 설명에서, 본 명세서의 일부인 첨부 도면을 참조하며, 이 도면은 예시적인 특정한 측면 혹은 일례로서 도시된 것이다. 전체 도면에서 같은 참조 번호는 같은 구성 요소를 가리키며, 이들 도면에는, 이벤트를

가입, 수신 및 처리하는 컴퓨팅 시스템 및 방법의 측면들이 도시되어 있다.

- [0016] 도 1을 참조하며, 여기에는 본 명세서에 개시되는 측면에 따라서 포함된 컴퓨터 시스템 및 다수의 컴포넌트에 관한 세부 사항이 도시되어 있다. 구체적으로, 도 1은 가상 처리 에이전트를 이용해서 이벤트를 전달하고 스트림을 처리하는 스트림 처리 시스템(100)의 측면을 나타내고 있다. 도 1에 도시된 바와 같이, 스트림 처리 시스템(100)은 그 안에 포함된 다양한 개체들 사이의 통신을 지원하고 용이하게 하는 런타임 시스템(108)을 포함할 수 있으며, 이들 개체는 가상 처리 에이전트(102A~102N)(각각을 가상 처리 에이전트(102)라고 하기도 하고, 이들 모두를 가상 처리 에이전트(102)라고 총칭하기도 함), 큐 시스템(104), 클라이언트 컴퓨팅 장치(116) 및 서버 컴퓨팅 장치(118)를 포함하지만, 이것으로 한정되는 것은 아니다.
- [0017] 런타임 시스템(108)은 분산된 서버의 그룹(도 1에는 도시되어 있지 않음)을 포함할 수 있으며, 이는 런타임 기초층(122)을 실행해서, 함께 "가상화된 분산 컴포넌트"("가상 컴포넌트"라고도 함)를 제공하고 지원하도록 구성될 수 있다. 여기서 사용되는 용어 '컴포넌트'는, 소프트웨어 객체나, 분산된 객체, 에이전트, 액터 등과 같은, 개별적으로 어드레싱 가능한 다른 종류의 분리된 개체와 같은 소프트웨어 컴포넌트를 가리킨다. 런타임 시스템(108)이 지원하는 가상화된 분산 컴포넌트는, 메모리 내 인스턴스화의 수명에 무관하게 존재하며, 따라서 런타임 시스템(108) 내의 임의의 개개의 서버의 수명과 무관하다. 가상 컴포넌트의 메모리 내 인스턴스가 없다면, 컴포넌트로 송신되는 메시지가, 런타임 기초층(122)으로 하여금 새로운 인스턴스를 이용 가능 서버에 생성하게 하는데, 이 서버는 런타임 시스템(108) 내의 임의의 이용 가능 서버일 수 있다. 이와 같이, 가상 컴포넌트의 식별에 의해서, 가상 컴포넌트는, 시스템에서 가동되는지의 여부에 무관하게, 상호 작용 및/또는 동작을 항상 어드레싱할 수 있다.
- [0018] 런타임 기초층(122)은 또한 가상 컴포넌트의 가동 및 가동 중지를 관리하도록 구성될 수 있다. 런타임 기초층(122)은 또한 가상 컴포넌트를 사용하는 프로그램에 투명하게 가상 컴포넌트들로의 통신을 관리하도록 구성될 수 있다. 이를 달성하기 위해서, 런타임 기초층(122)은 가상 컴포넌트의 임의의 가동된 인스턴스의 하나 이상의 서버 위치를 나타내는 정보를 유지하거나, 가동된 가상 컴포넌트 인스턴스에 하나 이상의 통신을 송신하거나, 혹은 가동 중지된 상태에 있는 가상 컴포넌트로의 통신을, 그 가상 컴포넌트의 인스턴스가 통신을 수신하기 위해서 가동 상태로 될 때까지 보존할 수 있다. 따라서 런타임 기초층(122)은, 시스템 내의 임의의 가능한 가상 컴포넌트를, 가동되고 있는 것처럼 즉, 메모리에 있는 것처럼, 호출할 수 있게 하는 가상의 '컴포넌트 공간'을 개발자에게 제공할 수 있다.
- [0019] 가상 컴포넌트가 가동되고 있던 서버가 고장 나면, 런타임 시스템(108) 내의 다른 서버에서 가상 컴포넌트를 재가동시킴으로써 가상 컴포넌트가 복구될 수 있다. 이와 같이, 가상 컴포넌트는 런타임 기초층(122)에 기초해서 개발된 애플리케이션에 의해서 관리될 필요가 없고 및/또는 명시적으로 다시 생성될 필요가 없다.
- [0020] 컴포넌트의 가상화는, 가상 컴포넌트로부터 현재 실행되고 있는 물리적인 인스턴스화/가동으로 매핑하는 인디렉션(indirection)의 레벨에 기초하고 있다. 이 인디렉션의 레벨은, 가상 컴포넌트 배치와 부하 밸런싱, 미사용 가상 컴포넌트의 가동 중지, 및 서버 고장 이후의 가상 컴포넌트 복구와 같은, 애플리케이션 개발자가 해결해야 했던 많은 문제들을 처리할 기회를, 런타임 기초층(122)에 제공할 수 있다. 따라서, 런타임 기초층(122)이 제공하는 가상 컴포넌트는, 부하 밸런싱 및 고장으로부터의 투명한 복구에 있어서의 런타임 유연성을 가능하게 해서, 프로그래밍 모델을 간략화할 수 있다. 런타임 기초층(122)에 대한 추가적인 세부 사항은, 2014년 3월 26일에 출원되어서 함께 계류중인 미국 특허 출원 제 14/228,129 호 "Virtualized Components in Computing Systems"에 개시되어 있다.
- [0021] 본 명세서에 개시된 기술의 측면에 따르면, 스트림 처리 시스템(100) 내의 가상 처리 에이전트(102)는 런타임 시스템(108)의 런타임 기초층(122)에 의해 제공 및 지원되는 가상 컴포넌트로서 구성될 수 있다. 이러한 구성에 기초해서, 가상 처리 에이전트(102)는 가동되고 있는지 여부에 무관하게, 항상 상호 작용 및/또는 동작을 어드레싱할 수 있다. 가상 처리 에이전트(102)가 가동되고 있던 서버가 고장나면, 가상 처리 에이전트(102)는 런타임 시스템(108)의 다른 서버 상에서 재가동됨으로써 복구될 수 있다.
- [0022] 일부 측면에서, 가상 처리 에이전트(102)는 각각이 특정한 액션을 수행하도록 구성될 수도 있고, 함께 대규모 소프트웨어 애플리케이션의 기능을 제공할 수도 있다. 예컨대, 컴퓨터 게임 애플리케이션에서, 게임 애플리케이션의 각각의 게임에 대해서 가상 처리 에이전트(102)가 생성되어서 각각의 게임을 관리하도록 구성될 수 있다. 가상 처리 에이전트(102)는 게임 애플리케이션의 게임을 호스팅하고 있는 각각의 맵에 대해서 생성될 수도 있다. 개개의 가상 처리 에이전트(102)는 또한, 게임 애플리케이션의 다양한 게임에 참여하고 있는 개개의 플레이어들을 관리하도록 생성 및 관리될 수 있다. 가상 처리 에이전트(102)에 의해 수행될 액션, 가상 처리 에이전트

(102)의 입력과 출력, 및/또는 다른 구성을 포함한, 가상 처리 에이전트(102)의 구성은, 가상 처리 에이전트(102)의 선언적 정의를 통해서 명시될 수도 있고, 혹은 프로그램적으로 획득될 수도 있다.

[0023] 스트림 처리를 위한 적절한 시스템을 제공하기 위해서, 스트림 처리 시스템(100) 내의 다양한 가상 처리 에이전트(102)는 서로 통신할 필요가 있다. 예컨대, 가상 처리 에이전트(102)의 상태를 업데이트하고, 다른 가상 처리 에이전트(102)를 가동하며, 및/또는 다른 가상 처리 에이전트(102)에 의해 수행될 특정 액션을 트리거하기 위해서, 가상 처리 에이전트(102) 사이에서 메시지가 통신될 수 있다. 본 명세서에 개시된 일부 측면에 따르면, 가상 처리 에이전트(102)로부터 시스템 내의 다른 가상 처리 에이전트(102) 및/또는 다른 개체로 통신될 메시지는, 큐 시스템(104)에 의해 구현되는 큐(106) 내의 이벤트(112)로서 발행될 수 있다.

[0024] 큐 시스템(104)은, 스트림 처리 시스템(100) 내의 다양한 개체로부터의 이벤트 발행을 수신하고 유지하도록 큐(106)를 실행하는, 하나 이상의 서버를 포함할 수 있다. 큐 시스템(104) 내의 서버와 런타임 시스템(108) 내의 서버는 같은 서버일 수도 있고 다른 서버일 수도 있으며, 큐 시스템(104) 내의 서버와 런타임 시스템(108) 내의 서버는 웹 서버, 애플리케이션 서버, 네트워크 장비, 전용 컴퓨터 하드웨어 장치, PC가 될 수도 있고, 이들의 임의의 조합 및/또는 공지된 다른 컴퓨팅 장치가 될 수도 있다는 것을 이해할 것이다. 서버는, LAN, WAN, 인터넷, 혹은 컴퓨팅 장치를 접속시키는, 공지된 임의의 다른 네트워킹 토폴로지가 될 수 있는 네트워크(도 1에서는 도시 생략)를 통해서 접속될 수 있다.

[0025] 큐(106)에서 발행되는 이벤트(112)는, 가상 처리 에이전트(102), 클라이언트 컴퓨팅 장치(116), 서버 컴퓨팅 장치(118) 및/또는 스트림 처리 시스템(100) 내의 다른 잠재적인 개체가 될 수 있는, 이벤트 송신 개체에서 발생되거나 혹은 관찰되는 이벤트와 관련된 다양한 정보를 포함할 수 있다. 이벤트(112)에 포함되는 정보로는, 이벤트가 발생된 개체의 신원, 이벤트의 시간, 이벤트의 위치, 이벤트에 포함된 개체의 신원 및/또는 이벤트를 묘사하는데 사용될 수 있는 다른 정보를 들 수 있지만, 이것으로 한정되는 것은 아니다. 이벤트(112)는, 런타임 시스템(108)을 통해서 이벤트 발행(110)을 송신함으로써 큐(106)에서 발행될 수 있다. 이벤트 발행(110)은, 그 이벤트가 발행될 것이라는 표시 및 이벤트(112)가 언제 어디서 발행될 것인지를 나타내는 데이터 등과 같은 추가 정보와 함께, 대응하는 이벤트(112)에 포함된 모든 정보를 포함할 수 있다. 또 다른 측면에서는, 큐 시스템(104)은 또한 발행된 이벤트(112)를, 큐(106)나, 혹은 큐 시스템(104) 및/또는 런타임 시스템(108)이 액세스 가능한 다른 저장 위치에 유지 혹은 저장할 수 있다.

[0026] 다른 가상 처리 에이전트(102) 및/또는 스트림 처리 시스템(100) 내의 다른 이벤트 발생 개체로부터, 처리를 위해서 이벤트를 수신하기 위해서, 가상 처리 에이전트(102)는 하나 이상의 이벤트 가입(114)을 송신해서 가입하여, 큐(106) 내의 발행된 이벤트를 수신한다. 이러한 가상 처리 에이전트(102)를, 이들 이벤트의 "명시적 가입자 처리 에이전트(102)" 혹은 "명시적 가입자(102)"라고도 한다. 비한정의 예로서, 이벤트 가입(114)은, 가상 처리 에이전트(102)가 수신에 관심이 있는 이벤트의 타입 혹은 콘텐츠를 나타내는 데이터를 포함할 수 있다. 이벤트 가입(114)은 또한 가입한 이벤트가 발행할 기간을 나타낼 수 있다. 이벤트 가입(114)은 또한, 가상 처리 에이전트(102)로 전달될 이벤트를 필터링하는데 사용될 수 있는 하나 이상의 서술(predicates) 혹은 규칙을 명시할 수 있다. 이벤트 가입(114)에는, 가상 처리 에이전트(102)가 수신에 관심이 있는 이벤트를 나타내는 다른 다양한 데이터가 포함될 수 있다는 것을 이해할 것이다.

[0027] 이하 상세하게 설명하는 바와 같이, 런타임 시스템(108)은 또한 스트림 처리층(124)을 실행하도록 구성될 수 있으며, 이는 "스트림 처리층(124)" 혹은 "런타임 스트림 처리층(124)"이라고도 할 수 있다. 런타임 스트림 처리층(124)은 스트림 처리 시스템(100) 내의 개체 사이에서의 이벤트 통신을 관리하도록 구성될 수 있으며, 이는 큐(106) 내의 이벤트 발행(110)을 관리하는 것, 가상 처리 에이전트(102) 및/또는 다른 이벤트 가입자로서의 이벤트(112)의 전달을 조정하는 것, 이벤트 가입(114)을 수신하고 처리하는 것, 그리고 잠재적인 다른 동작을 포함하지만 이것으로 한정되는 것은 아니다.

[0028] 런타임 스트림 처리층(124)은 하나 이상의 명시적 가입자 처리 에이전트(102)로부터 이벤트 가입(114)을 수신하고, 가입한 이벤트(112)가 큐(106)에서 발행되면, 가입한 이벤트(112)를 대응하는 명시적 가입자 처리 에이전트(102)에 송신할 수 있다. 다른 측면에 따르면, 런타임 스트림 처리층(124)은 또한 발행된 이벤트(112) 혹은 이 이벤트(112)에 포함된 이벤트 정보를 이벤트(112)에 명시적으로 가입하지 않았지만 발행된 이벤트(112)에 포함된 정보를 수신하는 것에 관심이 있는 하나 이상의 가상 처리 에이전트(102)에 송신할 수 있으며, 이 에이전트(102)는 발행된 이벤트(112)의 "묵시적 가입자 처리 에이전트(102)" 혹은 "묵시적 가입자(102)"라고도 한다.

[0029] 묵시적 가입자 처리 에이전트(102)는, 가상 처리 에이전트(102)에 의해 수행될 액션, 가상 처리 에이전트(102)의 입력과 출력, 가상 처리 에이전트(102)의 상태, 및/또는 가상 처리 에이전트(102)의 다른 구성과 같은, 가상

처리 에이전트(102)의 시험 구성 혹은 세팅을 통해서 식별될 수 있다. 예컨대, 가상 처리 에이전트(102)에 의해 수행될 액션이, 개시되었거나, 언급되었거나, 혹은 발행된 이벤트(112)와 관련된 객체 혹은 개체를 포함하는 경우에, 가상 처리 에이전트(102)는 묵시적 가입자 처리 에이전트(102)로서 식별될 수 있다. 또한, 발행된 이벤트(112)에 포함된 데이터가 가상 처리 에이전트(102)에 의해 소비 혹은 처리되는 경우에, 가상 처리 에이전트(102)는 묵시적 가입자 처리 에이전트(102)로서 식별될 수 있다. 이들 예는 단지 예시적인 것으로 한정적 의미가 아니라는 것을 이해할 것이다. 묵시적 가입자 처리 에이전트(102)를 나타내는 방법이 사용될 수도 있다.

[0030] 런타임 스트림 처리층(124)이 일단 묵시적 가입자 처리 에이전트(102)를 식별하면, 이벤트(112)가 런타임 기초층(122)을 통해서 묵시적 가입자 처리 에이전트(102)에 전달될 수 있다. 상술한 바와 같이, 스트림 처리 시스템(100) 내의 가상 처리 에이전트(102)는, 런타임 시스템(108)의 런타임 기초층(122)에 의해 지원되는 가상 컴포넌트로서 구현될 수 있다. 런타임 스트림 처리층(124)의 관점에서, 가상 처리 에이전트(102)는, 가상 처리 에이전트(102)가 시스템에서 가동되고 있는지 여부에 무관하게, 항상 이벤트(112)의 전달을 위해 어드레싱될 수 있다. 따라서 런타임 스트림 처리층(124)은 전달을 위해서 이벤트(112)를 런타임 기초층(122)에 패스할 수 있다.

[0031] 런타임 기초층(122)은 이벤트(112)를 수신하면, 묵시적 가입자 처리 에이전트(102)가 가동되었는지 여부, 즉 묵시적 가입자 처리 에이전트(102)가 메모리에 로딩되었고 동작할 준비가 되어 있는지 여부를 판정할 수 있다. 묵시적 가입자 처리 에이전트(102)가 가동되지 않았다면, 런타임 기초층(122)은 묵시적 가입자 처리 에이전트(102)를 가동시키고, 이벤트(112)를 가동된 묵시적 가입자 처리 에이전트(102)에 전달할 수 있다. 유사하게, 런타임 기초층(122)은 이벤트(112)를 명시적 가입자 처리 에이전트(102)에 전달할 때, 명시적 가입자 처리 에이전트(102)가 가동되었는지 판정하고, 필요에 따라서 이벤트(112)를 전달하기 전에 명시적 가입자 처리 에이전트(102)를 가동시킬 수 있다.

[0032] 가동되었을 때의 가상 처리 에이전트(102)의 구현 및 가상 처리 에이전트(102)의 상태에 따라서는, 가동된 묵시적 가입자 처리 에이전트(102) 중 일부는 아직, 이벤트(112)를 수용 및 처리하도록 구성되지 않을 수 있다. 예컨대, 묵시적 가입자 처리 에이전트(102)는 가동시에 이벤트 핸들러가 아직 구성되지 않았을 수 있다. 이러한 시나리오에서, 이벤트(112)에 포함된 데이터 전부 혹은 그 일부를 포함하는 이벤트 정보는 재배열되어서, 묵시적 가입자 처리 에이전트(102)가 허용 가능한 형태로 묵시적 가입자 처리 에이전트(102)에 전달될 수 있다. 다른 시나리오에서, 이벤트(112)에 포함된 상세한 데이터를 송신하지 않고도, 이벤트(112)의 발생을 나타내는 이벤트 정보를 가상 처리 에이전트(102)에 송신하기에 충분할 수 있다.

[0033] 수신한 이벤트(112) 혹은 이벤트 정보에 기초해서, 일부 묵시적 가입자 처리 에이전트(102)는 런타임 스트림 처리층(124)에 하나 이상의 이벤트 가입(114)을 제시해서, 큐(106) 내의 발행된 하나 이상의 이벤트를 처리를 위해 수신하도록 결정할 수 있다. 상기 간략하게 설명한 바와 같이, 묵시적 가입자 처리 에이전트(102)가 제시한 이벤트 가입(114)에 포함된 이벤트는, 이벤트 가입(114)이 수신되어서 처리되기 전에 큐(106)에 발행된 과거 이벤트를 포함할 수 있다. 런타임 스트림 처리층(124)은 대응하는 이벤트가 유지되는 큐 혹은 다른 저장부로부터 과거 이벤트를 취득하고, 이 취득한 이벤트를 묵시적 가입자 처리 에이전트(102)에 전달할 수 있다. 묵시적 가입자 처리 에이전트(102)가 이벤트 가입(114)을 제시할 때, 묵시적 가입자 처리 에이전트(102)는 가입된 이벤트에 대한 명시적 가입자 처리 에이전트(102)로 전환되고, 상기 설명한 명시적 가입자 처리 에이전트(102) 취급 방식과 유사하게 취급될 수 있다.

[0034] 이상 가상 처리 에이전트(102)가 이벤트 발행(110)을 송신하고, 이벤트 가입(114)을 제시하며, 및/또는 이벤트(112)를 수신하는 것으로 주로 설명했지만, 스트림 처리 시스템(100) 내의 다양한 다른 개체가 이들 동작을 수행할 수도 있다는 것을 이해할 것이다. 예컨대, 컴퓨터 게임 애플리케이션의 사용자는, 게임의 점수판 정보와 같은, 그 게임과 관련된 정보를 수신하는 것에 관심을 가질 수 있다. 사용자는 이러한 정보를, 이벤트 가입(114)을 런타임 시스템(108)에 송신해서 클라이언트 컴퓨팅 장치(116)를 통해서 획득할 수 있으며, 클라이언트 컴퓨팅 장치(116)는 PC, 랩톱, 노트북, PDA, 게임 콘솔, 셋톱 박스, e-리더, 가전 제품 장치, 스마트폰, 태블릿 컴퓨팅 장치, 서버 컴퓨터 혹은 런타임 시스템(108)과 통신할 수 있는 임의의 다른 컴퓨팅 장치가 될 수 있다.

[0035] 유사하게, 서버 컴퓨팅 장치(118)는 서버 컴퓨터, 웹 서버, 애플리케이션 서버, 네트워크 장비, 전용 컴퓨터 하드웨어 장치, PC가 될 수도 있고, 이들의 임의의 조합 및/또는 공지된 다른 컴퓨팅 장치가 될 수도 있으며, 여기서 게임이 실행되고, 이벤트 발행(110)을 송신해서 큐(106) 내에 게임 시작 이벤트(112)를 발행할 수 있다. 서버 컴퓨팅 장치(118)는 또한, 처리에 관심이 있는 이벤트(112)를 수신하기 위해서 하나 이상의 이벤트 가입(114)을 송신할 수 있다. 스트림 처리 시스템(100) 내의 다른 개체는 또한 상술한 것과 유사한 방식으로 이벤트(112)의 발행, 수신 및/또는 가입을 행하도록 구성될 수 있다.

- [0036] 도 1에서는 주로 스트림 처리 시스템(100) 내의 다양한 개체로부터 이벤트 발행(110)을 수신해서 유지하는데 큐(106)를 사용하는 것을 나타내었지만, 다른 다양한 메커니즘이 유사한 방식으로 사용될 수도 있다는 것을 이해할 것이다. 실제로, 데이터를 이후 처리를 위해서 저장하고 유지할 수 있는 가상의 임의의 데이터 구조가, 큐(106)와 관련해서 상술한 것과 유사한 방식으로 사용될 수 있다. 이 응용에는 이벤트 발행(110)을 유지하고 저장하는 이러한 데이터 구조를 모두 포함하도록 했다.
- [0037] 이하 도 2를 참조해서, 큐(106) 및 런타임 스트림 처리층(124)의 다른 측면을 나타내는 블록도가 설명된다. 상기 간략하게 설명한 바와 같이, 런타임 스트림 처리층(124)은 이벤트 발행(110) 및 이벤트 가입(114)을 처리하기 위해 가상 스트림을 제공 및 지원할 수 있다. 가상 스트림을 이용함으로써, 가상 처리 에이전트(102) 및/또는 다른 개체가 제시한 이벤트 발행(110) 및 이벤트 가입(114)은 큐(106)가 아닌 가상 스트림을 향할 수 있다. 그 결과, 가상 처리 에이전트(102)는 큐(106)의 존재를 인지할 필요가 없다. 도 2는 런타임 스트림 처리층(124)에 의해 추출된 다양한 가상 스트림과 가상 스트림들 사이의 상호 작용, 큐(106) 및 가상 처리 에이전트(102)를 나타내고 있다.
- [0038] 도 2에 도시된 바와 같이, 런타임 스트림 처리층(124)은, 가상 스트림과 관련된 기능을 제공 및 지원하고, 큐(106) 사이의 이벤트의 매핑 및 가상 스트림을 유지하기 위한 가상 스트림 모듈(202)을 포함할 수 있다. 가상 스트림 모듈(202)이 제공하는 가상 스트림은 이벤트 발행 가상 스트림(204A~204C)(이는 개별적으로 이벤트 발행 가상 스트림(204)이라고 할 수도 있고 혹은 함께 이벤트 발행 가상 스트림(204)이라고 할 수도 있다) 및 이벤트 가입 가상 스트림(206A~206D)(이는 개별적으로 이벤트 가입 가상 스트림(206)이라고 할 수도 있고 혹은 함께 이벤트 가입 가상 스트림(206)이라고 할 수도 있다)을 포함할 수 있다.
- [0039] 이벤트 발행가상 스트림(204) 각각은 발행될 이벤트에 의해 정의될 수 있고, 이벤트 가입 가상 스트림(206) 각각은 가입하는 개체가 수신에 관심이 있는 이벤트에 의해 정의될 수 있다. 예컨대, 가상 처리 에이전트(102A)는 이벤트 발행(110)을 런타임 스트림 처리층(124)에 송신해서 이벤트(112)를 발행할 수 있다. 이벤트 발행(110)은 이벤트 발행 가상 스트림(204C)을 통해서 런타임 스트림 처리층(124)으로 송신될 수 있다. 이벤트 발행(110)이 런타임 스트림 처리층(124)에서 수신되기 전에는 이벤트 발행 가상 스트림(204C)은 존재하지 않고, 이벤트 발행(110)이 도달하면 이벤트 발행 가상 스트림(204C)이 생성될 수 있다. 환언하면, 이벤트 발행(110) 혹은 그 안에 포함된 이벤트는, 이벤트 발행(110)을 송신받은 이벤트 발행 가상 스트림(204)이 존재하는 것으로 정의해서 그렇게 할 수 있다. 유사하게, 다른 가상 처리 에이전트(102)는 또한 대응하는 이벤트 발행 가상 스트림(204)에 이벤트 발행(110)을 제시해서 이벤트를 발행할 수 있다. 가상 스트림 모듈(202)은 이벤트 발행 가상 스트림(204)을 통해서 이벤트 발행(110)을 수신한 이후에, 이벤트 발행(110)에 명시된 이벤트를 발행을 위해서 큐(106)에 송신할 수 있다.
- [0040] 유사하게, 가상 처리 에이전트(102P)가 수신에 관심이 있는 이벤트를 나타내는 이벤트 가입(114)을 제시하면, 이러한 이벤트 가입(114)은 이벤트 가입 가상 스트림(206D)가 이벤트 가입(114)에 의해 명시된 이벤트에 대응하는 것으로 정의할 수 있다. 그 결과, 이벤트 가입(114)을 제시함으로써, 가상 처리 에이전트(102P)는 이벤트 가입(114)에 의해 정의된 이벤트 가입 가상 스트림(206D)에 가입하는 것과 동등하게 된다. 가상 처리 에이전트(102)가 복수의 이벤트 가입(114)을 런타임 스트림 처리층(124)에 제시함으로써 복수의 이벤트 가입 가상 스트림(206)에 가입할 수 있다는 것을 이해할 것이다. 일단 런타임 스트림 처리층(124)이 이벤트 가입(114)을 수신하면, 이벤트 가입 가상 스트림(206D)이 생성되고 존재할 수 있다. 이벤트 가입(114)에 기초해서, 가상 스트림 모듈(202)은 큐(106)에 발행된 이벤트(112)를 취득하고, 이 이벤트(112)를 이벤트 가입 가상 스트림(206D)에 푸시할 수 있으며, 이는 이후에 가상 처리 에이전트(102P)로 전달될 수 있다.
- [0041] 예컨대, 가상 처리 에이전트(102A)는 게임을 처리하고, 이벤트 발행(110)을 런타임 스트림 처리층(124)에 송신해서 게임에서 발생하는 이벤트(112)를 발행하도록 구성될 수 있다. 이벤트 발행(110)이 제시되면, 게임 이벤트 스트림(204C)이 생성되고, 이벤트(112)를 수신해서 발행할 수 있다. 게임 이벤트 스트림(204C)을 통해서 수신한 이벤트(112)는 다시 발행 및 저장을 위해서 큐(106)로 송신될 수 있다. 여기서, 런타임 스트림 처리층(124)도, 가상 처리 에이전트(102P)로부터 이벤트 가입(114)을 수신해서, 게임 내의 모든 킬 이벤트를 수신하도록 가입했을 수 있다. 이러한 이벤트 가입(114)은 킬 이벤트 스트림(206D)을 생성시켰을 수 있다.
- [0042] 가상 스트림 모듈(202)은 게임 이벤트 스트림(204C) 내의 이벤트 발행(110)을 수신하면, 킬 이벤트가 있다면, 게임 이벤트 스트림(204C)을 통해서 발행되는 킬 이벤트가 가상 처리 에이전트(102P)로 전달되도록 킬 이벤트 스트림(206D)으로 포워딩되어야 하는지 검출할 수 있다. 이러한 검출은 예컨대, 상술한 바와 같이, 큐(106) 내의, 그리고 가상 스트림 모듈(202)이 유지하고 있는 가상 스트림 내의 이벤트의 매핑을 이용함으로써 수행될 수

있다. 이어서 게임 이벤트 스트림(204C)을 통해서 발행되는 각각의 킬 이벤트(112)의 경우, 가상 스트림 모듈(202)은 큐(106)로부터 킬 이벤트(112)를 취득하고, 이를 킬 이벤트 스트림(206D)에 푸시해서 가상 처리 에이전트(102P)에 전달하게 한다. 상술한 예는 단지 예시적인 것으로, 다양한 다른 이벤트(112), 이벤트 발행(110), 이벤트 가입(114) 및/또는 다른 데이터는 가상 스트림(204, 206) 및/또는 런타임 스트림 처리층(124)을 통해서 가상 처리 에이전트(102) 및 큐(106) 사이에서 통신될 수 있다는 것을 이해할 것이다.

[0043] 다른 측면에 따르면, 가상 처리 에이전트(102)가 제시하는 이벤트 가입(114)은, 대응하는 이벤트 가입 가상 스트림(206)을 더 정의하는데 사용될 수 있는 하나 이상의 서술을 명시할 수 있다. 이 서술은 적절한 시점에 런타임 스트림 처리층(124)에 의해서 처리되어서, 이벤트 가입 가상 스트림(206)으로 푸시되는 이벤트를 필터링하는데 사용될 수 있는 하나 이상의 필터를 생성하는 것과 같이, 이벤트 가입 가상 스트림(206)을 구체화할 수 있다. 서술을 이용함으로써, 가상 처리 에이전트(102)로 송신되는 이벤트를 더 구체화해서, 따라서 가상 처리 에이전트(102)로 불필요한 이벤트가 전달되는 것을 방지한다.

[0044] 이상으로부터 알 수 있는 바와 같이, 가상 스트림(204, 206)은 매우 유동적이고 유전하며, 필요할 때마다 생성될 수 있어서, 스트림 처리 시스템(100)의 확장 가능성을 크게 증가시킨다. 다른 측면에 따르면, 가상 스트림(204, 206)은 논리적으로 항상 존재할 수 있지만, 시스템 리소스는 가상 스트림(204, 206) 내에 처리할 이벤트가 있을 때에만 가상 스트림(204, 206)에 할당된다. 가상 스트림(204, 206)은 일정 기간 동안 가상 스트림(204, 206) 내에 어떤 이벤트도 수신되지 않으면 자신의 리소스가 회수(reclaim)되어서 가동 중지될 수 있다. 이 기간을 사전 결정될 수도 있고, 혹은 시스템 리소스의 사용량과 같은 시스템의 상태에 따라서 유동적으로 조정될 수도 있다. 이러한 구현 방식은 가상 스트림이 세밀하게 만들어지는 시나리오에서 바람직할 수 있다. 예컨대, 애플리케이션 시스템 내의 가상 스트림은 사용자-기반, 장치-기반 및/또는 세션-기반으로 구성될 수 있다. 대형 애플리케이션 시스템에서, 사용자의 수, 장치의 수 및/또는 세션의 수, 그리고 가상 스트림의 수는 수백만 혹은 심지어 수십억개의 규모가 될 수 있다. 그러나, 이들 가상 스트림과 관련된 이벤트는 드물게 및/또는 간헐적으로 발생할 수 있으며, 한번은 가상 스트림의 작은 부분만이 활성화될 필요가 있다. 이러한 시스템에서, 가상 스트림(204, 206) 내에 처리할 이벤트가 있을 때만 리소스를 가상 스트림에 할당함으로써, 시스템의 효율 및 확장 가능성을 크게 높일 수 있다.

[0045] 나아가, 가상 스트림 모듈(202)은 큐(106)와 상호 작용하기 전에 가상 처리 에이전트(102)와의 통신을 통합(aggregate)하도록 더 구성될 수 있다. 예컨대, 가상 스트림 모듈(202)은 다수의 가상 처리 에이전트(102)로부터 다수의 이벤트 발행(110)을 수신할 수 있다. 이들 이벤트 발행(110)은 예컨대 멀티플렉싱을 통해서 통합되어서, 큐(106)로 통신되어, 하나의 통신 접속을 통해서 다수의 이벤트(112)를 통해서 발행될 수 있다. 유사하게, 다수의 가상 처리 에이전트(102)로부터 수신한 다수의 이벤트 가입(114)은 처리되고 통합되어서, 다수의 가상 처리 에이전트(102)로 송신될 이벤트(112)는 큐(106)와의 하나의 통신 접속을 통해서 큐(106)로부터 취득될 수 있다.

[0046] 상기 예에서는, 다수의 통신이 통합되어서 하나의 통신 접속을 통해서 큐(106)로 통신되는 것으로 설명되었지만, 런타임 스트림 처리층(124)과 큐(106) 사이의 통신을 위해서 하나 이상의 통신 접속이 성립될 수 있다는 것을 이해할 것이다. 그러나, 성립되는 통신 접속의 수는, 가상 처리 에이전트(102)로부터의 이벤트 발행(110) 및/또는 이벤트 가입(114)과 같은 통신 요청의 수보다 훨씬 적을 수 있다. 스트림 처리 시스템(100)의 규모가 증가할수록 통신 종합의 이점은 훨씬 더 커질 수 있다. 예컨대, 가상 처리 에이전트(102)가 수백만개인 스트림 처리 시스템(100)의 경우에, 가상 처리 에이전트(102)로부터 통신 요청은 수천만개의 규모가 있을 수 있으며, 큐(106)가 가상 처리 에이전트(102) 각각과의 직접 TCP/IP 접속을 다수 성립시키는 것은 일반적으로 불가능하다. 통신 통합을 통해서, 가상 스트림 모듈(202)은 큐(106)로의 직접 TCP/IP 접속을 수천개로 감소시킬 수 있으며, 이로써 통신 리소스의 소비를 실질적으로 감소시킨다.

[0047] 도 1 및 2에서 이벤트(112)를 발행하고 유지하는 데 하나의 큐(106)가 사용되는 것으로 도시되어 있지만, 다수의 큐(106)가 사용될 수 있다는 것을 이해할 것이다. 이러한 구현예에서, 가상 스트림 모듈(202)은 또한 다수의 큐(106)를 관리하고, 이들 큐(106)와 다양한 가상 스트림(204, 206) 사이의 이벤트의 매핑을 유지하도록 구성될 수 있다. 클라이언트 컴퓨팅 장치 및 서버 컴퓨팅 장치와 같은, 스트림 처리 시스템(100) 내의 다른 개체가 이벤트 발행(110)을 통해서 이벤트를 발행할 수도 있고, 이벤트 가입(114)을 제시하며, 및/또는 가상 스트림(204, 206)을 통해서 이벤트(112)를 수신할 수 있다는 것을 이해할 것이다.

[0048] 도 2는 또한 가상 처리 에이전트(102)를 조직하고 관리하며, 처리 에이전트 컨테이너(208A-208B)(이는 개별적으로 처리 에이전트 컨테이너(208)라고 할 수도 있고, 함께 처리 에이전트 컨테이너(208)라고 할 수도 있다)를 이

용하는 측면을 나타내고 있다. 처리 에이전트 컨테이너(208)는, 자신으로부터 분리되어서 이격되어 있는 가상 처리 에이전트(102)에 독립된 컴퓨팅 환경을 제공할 수 있다. 가상 처리 에이전트(102)를 호스팅하기 위해서, 가상 머신과 같은 다양한 컴퓨팅 리소스가 처리 에이전트 컨테이너(208)에 할당될 수 있다. 각각의 처리 에이전트 컨테이너(208)의 크기는 또한 다른 처리 에이전트 컨테이너(208)로부터 독립해서 결정될 수 있다. 이벤트(112)를 발행하는 가상 처리 에이전트(102)와 이벤트(112)에 가입해서 이를 수신하는 가상 처리 에이전트(102)는, 같은 처리 에이전트 컨테이너(208) 내에 있을 수도 있고, 2개의 서로 다른 처리 에이전트 컨테이너(208) 내에 있을 수도 있다는 점을 이해할 것이다. 가상 처리 에이전트(102)가 처리 에이전트 컨테이너(208)의 사용을 포함하지 않는 다른 방식으로 조직되고 관리될 수 있다는 것을 이해할 것이다.

[0049] 도 3은 가상 처리 에이전트(102)에 이벤트를 전달하는 하나의 예시적인 루틴(300)을 나타내는 흐름도이다. 일부 구현예에서, 루틴(300)은 도 1과 관련해서 상기 설명한 런타임 시스템(108)의 런타임 스트림 처리층(124)에 의해 수행된다. 그러나, 루틴(300)은, 런타임 시스템(108)의 다른 모듈 및/또는 컴포넌트에 의해서나, 혹은 도 1과 관련해서 상기 설명한 스트림 처리 시스템(100) 내의 다른 개체의 모듈 및/또는 컴포넌트에 의해서 수행될 수도 있다는 것을 이해할 것이다.

[0050] 도 3 등과 관련해서 설명되는 논리적인 동작은 (1) 컴퓨팅 시스템에서 실행되는 컴퓨터 구현식 액트나 프로그램 모듈의 시퀀스로서, 및/또는 (2) 컴퓨팅 시스템 내의 상호 접속된 머신 논리 회로나 논리 모듈로서 구현된다는 것을 이해할 것이다. 구현은, 컴퓨팅 시스템의 성능 및 다른 요건에 따른 선택의 문제이다. 따라서, 본 명세서에서 설명되는 논리적인 동작은, 스테이트, 동작, 구조적 장치, 액트 또는 모듈로 다양하게 불린다. 이들 스테이트, 동작, 구조적 장치, 액트 및 모듈은 소프트웨어로, 펌웨어로, 특수 목적 디지털 로직으로, 그리고 이들의 임의의 조합으로 구현될 수 있다. 또한, 도면에 도시되거나 본 명세서에 개시된 동작보다 더 많은 혹은 더 적은 동작이 수행될 수도 있다는 것을 이해할 것이다. 이들 동작은 본 명세서에 개시된 순서와는 다른 순서로 수행될 수도 있다. 또한, 개시된 방법 각각은 임의의 시점에서 종료될 수 있으며, 전체가 반드시 수행되어야 하는 것은 아니라는 것을 이해할 것이다.

[0051] 방법의 동작 모두 혹은 일부, 및/또는 실질적으로 동등한 동작은, 이하 정의되는 바와 같이, 컴퓨터-저장 매체에 포함된 컴퓨터-판독 가능 명령을 실행함으로써 수행될 수 있다. 상세한 설명 및 청구항에서 사용되는 용어 '컴퓨터-판독 가능 명령어' 및 이와 유사한 표현은, 루틴, 애플리케이션, 애플리케이션 모듈, 프로그램 모듈, 프로그램, 컴포넌트, 데이터 구조, 알고리즘 등을 넓게 포함하는 것으로서 사용된다. 컴퓨터-판독 가능 명령은, 싱글-프로세서나 멀티프로세서 시스템, 미니컴퓨터, 메인프레임 컴퓨터, PC, 휴대형 컴퓨팅 장치, 마이크로프로세서-기반 프로그램 가능 가전 제품, 및 이들의 조합을 포함하는, 다양한 시스템 구성으로 구현될 수 있다.

[0052] 루틴(300)은 동작 302로부터 시작하며, 여기서 런타임 시스템(108)의 스트림 처리층(124)은, 대응하는 이벤트 발행 가상 스트림(204) 내에서 이벤트 발행(110)을 수신할 수 있다. 상술한 바와 같이, 이벤트 발행 가상 스트림(204)은 수신한 이벤트 발행(110)에 기초해서 유동적으로 생성될 수 있다. 이후, 루틴(300)은 동작 304로 넘어가며, 여기서 런타임 스트림 처리층(124)은 이벤트 발행(110)에 명시된 이벤트(112)를 발행을 위해서 큐(106)에 송신한다. 런타임 스트림 처리층(124)은 또한 이벤트(112)가 이후에 취득될 수 있도록, 큐(106)에 지시해서 이벤트(112)를 유지 혹은 저장하게 할 수 있다. 일부 구현예에서, 이벤트(112)는 큐(106) 및/또는 런타임 스트림 처리층(124)이 액세스 가능한 저장 장치 내에 유지 혹은 저장될 수 있다.

[0053] 루틴(300)은 동작 304로부터 동작 306으로 넘어가며, 여기서 런타임 스트림 처리층(124)은 큐(106)로부터 이벤트(112)를 취득할 수 있고, 루틴(300)은 다시 동작 308로 넘어가며, 여기서 이벤트(112)에 기초해서 명시적 가입자 처리 에이전트(102)가 식별될 수 있다. 상술한 바와 같이, 명시적 가입자 처리 에이전트(102)는, 이벤트(112)에는 명백하게 가입하지 않았지만 이벤트(112)에 포함된 정보를 수신하는 것에 관심이 있는 가상 처리 에이전트(102)이다. 명시적 가입자 처리 에이전트(102)는, 가상 처리 에이전트(102)에 의해 수행될 액션, 가상 처리 에이전트(102)의 입력과 출력, 가상 처리 에이전트(102)의 상태, 및/또는 가상 처리 에이전트(102)의 다른 구성과 같은, 가상 처리 에이전트(102)의 시험 구성 혹은 세팅을 통해서 식별될 수 있다. 예컨대, 가상 처리 에이전트(102)에 의해 수행될 액션이, 개시되었거나, 언급되었거나, 혹은 발행된 이벤트(112)와 관련된 객체 혹은 개체를 포함하는 경우에, 가상 처리 에이전트(102)는 묵시적 가입자 처리 에이전트(102)로서 식별될 수 있다. 또한, 발행된 이벤트(112)에 포함된 데이터가 가상 처리 에이전트(102)에 의해 소비 혹은 처리되는 경우에, 가상 처리 에이전트(102)는 묵시적 가입자 처리 에이전트(102)로서 식별될 수 있다.

[0054] 일부 구현예에서, 런타임 스트림 처리층(124)은 다양한 타입의 이벤트 혹은 대응하는 이벤트 가입 가상 스트림(206)을 하나 이상의 묵시적 가입자 처리 에이전트(102)에 매핑하는 묵시적 가동 테이블을 유지할 수 있다. 예

시적인 묵시적 가동 테이블이 도 6에 도시되어 있으며 이후에 설명한다. 묵시적 가동 테이블은 사전 생성되어서, 필요시에 런타임 스트림 처리층(124)에 의해 탐색될 수 있다. 묵시적 가동 테이블은, 주어진 이벤트 혹은 이벤트 정보에 대한 하나 이상의 묵시적 가입자 처리 에이전트(102)를 출력할 수 있는 런타임 스트림 처리층(124)에서 실행되고 있는 기능 혹은 모듈을 통해서 구현될 수도 있다. 묵시적 가동 테이블이, 상술한 것과 동일한 원리에 따르면서 다른 방식으로 구현될 수도 있다는 점에 주의한다.

[0055] 루틴(300)은 동작 308로부터 동작 310으로 넘어가고, 여기서 동작 308에서 식별된 묵시적 가입자 처리 에이전트(102)는 예컨대, 이벤트(112) 혹은 이벤트(112)의 이벤트 정보를 묵시적 가입자 처리 에이전트(102)에 송신함으로써, 이벤트(112)를 통지받을 수 있다. 상술한 바와 같이, 이벤트 정보는 이벤트(112)에 포함된 정보나 이벤트(112)와 관련된 정보를 포함할 수 있다. 묵시적 가입자 처리 에이전트(102)의 성향(nature)에 따라서, 일부 묵시적 가입자 처리 에이전트(102)로 송신되는 이벤트 정보는 이벤트(112)에 포함된 데이터 모두 혹은 일부를 포함할 수 있다. 다른 묵시적 가입자 처리 에이전트(102)의 경우에, 이벤트 정보는 이벤트(112)의 존재 여부에 대한 표시만을 포함할 수도 있다. 도 4와 관련해서 상기 상세하게 설명한 바와 같이, 수신한 이벤트(112) 혹은 이벤트 정보에 기초해서, 묵시적 가입자 처리 에이전트(102)는 이벤트를 더 수신하고 및/또는 동작을 더 수행하기 위해서 가입할지 여부를 결정할 수 있다.

[0056] 루틴(300)은 동작(310)으로부터 동작(312)으로 넘어가고, 여기서 런타임 스트림 처리층(124)은 하나 이상의 묵시적 가입자 처리 에이전트(102)로부터 이벤트 가입(114)을 수신해서, 대응하는 이벤트 가입 가상 스트림(206)에 가입할 수 있다. 루틴(300)은 동작(314)으로 넘어가고 여기서, 런타임 스트림 처리층(124)은 명시적 이벤트 가입(114)에 명시된 이벤트가, 큐(106) 내의 이미 발생했고 발행된 과거 이벤트를 포함하고 있는지 판정할 수 있다. 명시적 이벤트 가입(114)이 과거 이벤트를 포함하는 것으로 판정되면, 루틴(300)은 동작(316)으로 넘어가고, 여기서 런타임 스트림 처리층(124)은 큐(106) 및/또는 과거 이벤트를 저장하고 있는 다른 저장 장치로부터 과거 이벤트를 취득할 수 있고, 이 취득한 과거 이벤트를 대응하는 가상 처리 에이전트(102)로 전달할 수 있다.

[0057] 동작(316)으로부터, 즉 동작(314)에서 명시적 이벤트 가입(114)이 과거 이벤트를 포함하지 않는다고 판정되면, 루틴(300)은 동작(318)으로 넘어가고, 여기서 런타임 스트림 처리층(124)은 유지하고 있는 명시적 가입을 갱신할 수 있고, 및/또는 이벤트 발행 가상 스트림(204), 이벤트 가입 가상 스트림(206) 및 큐(106) 내의 이벤트의 매핑을 갱신하는 등의, 수신한 명시적 이벤트 가입(114)에 기초한 다른 동작을 수행할 수 있다.

[0058] 루틴(300)은 동작(306)으로부터 동작(320)으로 넘어갈 수 있다. 동작(320)에서, 런타임 스트림 처리층(124)은 가상 처리 에이전트(102)가 이전에 제시한 이벤트 가입(114)에 기초해서 하나 이상의 명시적 가입자 처리 에이전트(102)를 식별할 수 있다. 루틴(300)은 다시 동작(322)으로 넘어가고, 여기서 이벤트(112)가 명시적 가입자 처리 에이전트(102)로 전달될 수 있다. 루틴(300)은 동작(318) 혹은 동작(322)으로부터 동작(324)으로 넘어가고, 여기서 런타임 스트림 처리층(124)이 이벤트를 더 전달해야 하는지 판정할 수 있다. 런타임 스트림 처리층(124)이 이벤트를 더 전달해야 한다면, 루틴(300)은 동작(302)으로 돌아가서 처리 및 전달을 위해서 이벤트 발행(110)을 더 수신한다. 더 전달할 필요가 없다면, 루틴(300)은 동작(326)으로 넘어가고 여기서 루틴(300)은 종료된다.

[0059] 상술한 이벤트 전달 처리에서, 이벤트(112)를 명시적 가입자 처리 에이전트(102)에 전달하면, 및/또는 이벤트(112)나 이벤트 정보를 명시적 가입자 처리 에이전트(102)에 송신하면, 런타임 기초층(122)은 대응하는 처리 에이전트(102)가 가동되었는지 여부를 우선 판정할 수 있다는 것을 이해할 것이다. 가동되지 않은 처리 에이전트(102)가 존재한다면, 런타임 기초층(122)은 처리 에이전트(102)를 가동시키고, 이후에 이벤트(112) 혹은 이벤트 정보를 가동된 처리 에이전트(102)에 전달할 수 있다.

[0060] 여기서, 묵시적 가입자 처리 에이전트(102)와 같은 처리 에이전트(102)는, 처리 에이전트(102)가 처리해야 하는 스트림(206) 내의 제 1 이벤트(112)의 도달시에 시스템 리소스를 할당하는 것을 포함해서, 유동적으로 가동될 수 있다. 따라서, 메모리 및 CPU와 같은 시스템 리소스는 제 1 이벤트(112)가 도달하기 전에는 처리 에이전트(102)에 할당될 필요가 없을 수 있다. 유사하게, 처리 에이전트(102)가 예컨대, 일정 기간 동안 유휴 상태가 되면, 처리 에이전트(102)가 가입한 스트림(206)으로부터 어떠한 이벤트(112)도 처리를 위해서 수신하지 않기 때문에, 처리 에이전트(102)는 투명하게 가동 중지되어서 자신의 시스템 리소스가 회수될 수 있다. 처리 에이전트(102)를 가동 중지하는 일정 기간은 사전 결정될 수도 있고, 시스템 리소스의 사용과 같은 시스템의 상태에 따라서 유동적으로 조정될 수도 있다.

[0061] 도 4는 이벤트 처리 및 가입을 행하는 하나의 예시적인 루틴(400)의 측면을 나타내는 흐름도이다. 일부 구현에서, 루틴(400)은 도 1 및 2와 관련해서 상기 설명한 가상 처리 에이전트(102)에 의해 수행된다. 그러나, 루틴

(400)은 도 1에 도시된 스트림 처리 시스템(100) 내의 다른 개체에 의해서 수행될 수도 있다는 것을 이해할 것이다.

[0062] 루틴(400)은 동작(402)에서 시작되며, 여기서 목시적 가입자 처리 에이전트(102)는 이벤트(112) 혹은 이벤트(112)의 이벤트 정보를 런타임 스트림 처리층(124)로부터 수신하고, 다시 수신한 이벤트(112) 혹은 이벤트 정보를 처리할 수 있다. 루틴(400)은 동작(402)으로부터 동작(404)으로 넘어가고, 여기서 목시적 가입자 처리 에이전트(102)는, 수신한 이벤트(112) 혹은 이벤트 정보에 기초해서 이벤트(112)에 더 가입해야 하는지 판정할 수 있다. 목시적 가입자 처리 에이전트(102)가 이벤트에 더 가입해야 한다고 판정되면, 루틴은 동작(406)으로 넘어가서, 목시적 가입자 처리 에이전트(102)는 자신이 수신에 관심이 있는 이벤트가 존재하는 하나 이상의 이벤트 가입 가상 스트림(206)을 식별할 수 있다. 동작(408)에서, 목시적 가입자 처리 에이전트(102)는 이벤트 가입(114)을 런타임 스트림 처리층(124)에 제시함으로써 식별된 이벤트 가입 가상 스트림(206)에 가입할 수 있다. 이벤트 가입(114)을 제시함으로써, 목시적 가입자 처리 에이전트(102)는 제시한 이벤트 가입(114)와 관련해서 명시적 가입자 처리 에이전트(102)로 전환될 수 있다.

[0063] 루틴(400)은 동작(408)으로부터 동작(410)으로 넘어가고, 여기서 명시적 가입자 처리 에이전트(102)는 큐(106) 내에 발행된 가입 이벤트를 수신할 수 있다. 상술한 바와 같이, 동작(408)에서 제시된 이벤트 가입(114)에 과거 이벤트가 명시되어 있다면, 동작(410)에서 수신한 이벤트는 이들 과거 이벤트를 포함할 수 있다. 이후에 루틴(400)은 동작(412)으로 넘어가고, 여기서 수신한 이벤트가 처리될 수 있다. 루틴(400)은 또한 동작(414)으로 넘어가고, 여기서 명시적 가입자 처리 에이전트(102)는 처리할 이벤트가 더 필요한지 판정될 수 있다. 이벤트가 더 필요하다고 판정되면, 루틴(400)은 동작(410)으로 돌아가고, 여기서 명시적 가입자 처리 에이전트(102)는 이벤트(112)를 처리를 위해 더 수신할 수 있다. 동작(414)에서 더 이상 이벤트가 필요없다고 판정되거나, 혹은 동작(404)에서 목시적 가입자 처리 에이전트(102)가 추가 이벤트에 가입할 필요가 없다고 판정되면, 루틴(400)은 동작(416)으로 넘어가고 여기서 루틴(400)은 종료된다.

[0064] 도 5a 내지 5d는 본 명세서에 개시된 하나 이상의 측면에 따르면, 이벤트(112)가 서로 다른 시점에 큐(106) 내에 발행되는 것에 응답해서, 가상 처리 에이전트(102)에 의해 수행되는 동작을 나타낸다. 도 5a에 도시된 바와 같이, 런타임 시스템(108)의 런타임 스트림 처리층(124)은, 상이한 타입의 이벤트(112) 혹은 이들의 대응하는 이벤트 가입 가상 스트림(206)을 가상 처리 에이전트(102)에 매핑한 목시적 가동 테이블(506)을 유지하거나 혹은 생성할 수 있다. 도 5a에 도시된 예시적인 목시적 가동 테이블(506)에 따라서, 게임 스트림(504)의 목시적 가입자 처리 에이전트(102)로서 맵 처리 에이전트, 게임 처리 에이전트, 점수판 처리 에이전트 및/또는 우수 플레이어 처리 에이전트가 식별될 수 있다. 유사하게, 플레이어 스트림(520)의 경우에, 잠재적인 목시적 가입자 처리 에이전트(102)는 플레이어 처리 에이전트, 게임 처리 에이전트 및/또는 우수 플레이어 처리 에이전트를 포함할 수 있다. 목시적 가동 테이블(506)에 기초해서, 런타임 스트림 처리층(124)은 이벤트(112)가 수신되어서 큐(106) 내에 발행됨에 따라서, 하나 이상의 목시적 가입자 처리 에이전트(102)를 식별해서 가동시킬 수 있다.

[0065] 시점 T0에, 예컨대, 이벤트 발행 가상 스트림(204)을 통해서 게임 시작 이벤트(502)가 큐(106)에서 발행된다. 게임 시작 이벤트(502)에 포함된 데이터는 플레이어 A-F가 현재 맵 H에서 게임 X를 시작했다는 것을 나타낸다. 런타임 스트림 처리층(124)은 큐(106)로부터 게임 시작 이벤트(502)를 취득하고, 게임 시작 이벤트(502)에 대응하는 게임 스트림(504)의 목시적 가입자 처리 에이전트(102)를 식별할 수 있다. 목시적 가동 테이블(506)에 기초해서, 런타임 스트림 처리층(124)은, 게임 스트림(504)의 목시적 가입자 처리 에이전트(102)가, 맵 G에서의 킬 수를 계산할 맵 처리 에이전트(508), 맵 H에서의 킬 수를 계산할 맵 처리 에이전트(510), 우수 플레이어(C, M, S)의 킬 수를 계산할 우수 플레이어 처리 에이전트(512)를 포함하는지 판정할 수 있다. 이후, 런타임 스트림 처리층(124)은 게임 시작 이벤트(502)에 기초해서 게임 시작 이벤트 정보(514)를 생성하고, 이를 게임 스트림(504)에 푸시해서 식별된 목시적 가입자 처리 에이전트(508, 510, 512)에 전달하게 한다.

[0066] 도 5b에 도시된 바와 같이, 시점 T1에, 게임 시작 이벤트 정보(514)가 목시적 가입자 처리 에이전트(508, 510, 512)에 전달되어서 처리될 수 있다. 처리 에이전트(508)에서는, 게임 시작 이벤트 정보(514)에 나타나 있는, 게임 X와 관련된 맵 H이 맵 처리 에이전트(508)가 관심을 갖는 맵 G와 다르다고 판정될 수 있다. 그 결과, 처리 에이전트(508)는 수신한 게임 시작 이벤트 정보(514)와 관련해서 아무것도 행하지 않는다. 그러나, 맵 처리 에이전트(510)에서, 게임 시작 이벤트 정보(514)에 포함된 맵 H가 맵 처리 에이전트(510)가 관심을 갖는 맵이라고 판정되고, 따라서 맵 처리 에이전트(510)는 가입(520)을 송신해서 게임 X 킬 스트림(530)에 가입하여 게임 X의 모든 이벤트를 수신할 수 있다.

[0067] 유사하게, 우수 플레이어 처리 에이전트(512)는 처리 중인 우수 플레이어 중 하나를 게임 X가 포함한다고 판정

하고, 따라서 가입(524)을 제시하여 게임 X 플레이어 C 킬 스트림(532)에 가입하여 게임 X의 플레이어 C에 의한 모든 킬 이벤트를 수신할 수 있다. 게임 시작 이벤트 정보(514)가 처리 에이전트(508, 510, 512)에 전달되어서 처리되고 있지만, 플레이어 C에 의한 킬 이벤트(514), 플레이어 A를 시상하기 위한 시상 이벤트(516) 및 플레이어 F에 의한 킬 이벤트(518)와 같이, 더 많은 이벤트가 큐(106)에서 발행될 수 있다는 점에 주의한다.

[0068] 도 5c에 도시된 바와 같은 시점 T2에서, 런타임 스트림 처리층(124)은, 맵 처리 에이전트(510) 및 우수 플레이어 처리 에이전트(512)가 제시한 이벤트 가입을 처리할 수 있다. 가입(522)이 게임 X의 개시 이후의 모든 킬 이벤트를 요청하고 있기 때문에, 런타임 스트림 처리층(124)은 이러한 가입이 가입(522)이 수신되기 전에 발행된 과거의 킬 이벤트(514, 518)를 포함한다고 판정할 수 있다. 유사하게, 런타임 스트림 처리층(124)은 또한 가입(524)이 과거의 킬 이벤트(514)를 포함한다고 판정할 수 있다. 따라서, 런타임 스트림 처리층(124)은 큐(106)로부터 킬 이벤트(514, 518)를 취득해서, 킬 이벤트(514, 518)를 게임 X 킬 스트림(530)을 통해서 맵 처리 에이전트(510)에 송신하고, 킬 이벤트(514)를 게임 X 플레이어 C 킬 스트림(532)을 통해서 맵 처리 에이전트(512)에 송신할 수 있다.

[0069] 도 5d에 도시된 바와 같이, 시점 T3에서, 새로운 킬 이벤트(528)가 큐(106)에서 발행된다. 런타임 스트림 처리층(124)은 큐(106)로부터 이 이벤트(528)를 취득하고, 이 새로운 킬 이벤트(528)가 게임 X 킬 스트림(530)에 속한다고 판정할 수 있다. 이후에 킬 이벤트(528)는 게임 X 킬 스트림(530)에 푸시되어서 맵 처리 에이전트(510)에 전달될 수 있다. 이벤트가 유사한 방식으로 큐(106)에서 더 발행됨에 따라서, 런타임 스트림 처리층(124)은 계속해서 더 많은 이벤트를 처리하고 전달할 수 있다.

[0070] 이상으로부터 알 수 있는 바와 같이, 묵시적 가입자 처리 에이전트(102)가 자신이 가입하지 않은 이벤트에 대한 정보를 수신할 수 있게 함으로써, 시스템의 매우 정밀한 스트림 처리 유닛 지원을 통해, 가상 처리 에이전트(102)가 이벤트 흐름에 기초해서 자동으로 가동될 수 있고, 이로써 프로그래밍 모델을 크게 간소화시킬 수 있다. 또한, 스트림으로 전달되는 이벤트의 시멘틱에 대한 스트림의 정의를 감소시킴으로써, 처리 에이전트(102)는 스트림-토폴로지를 인지할 필요가 없고, 대신 처리하고자 하는 특정 이벤트에만 집중할 수 있으므로, 프로그래밍 모델을 더 간소화시킬 수 있다. 나아가, 본 명세서에 개시된 기술을 이용하면, 토폴로지를 재검파일하거나 재전개(re-deploy)할 필요없이, 복잡한 이벤트 검출 및 통합 로직의 애드혹 정의가 유동적으로 시스템에 적용될 수 있다.

[0071] 도 6은 도 1에 도시된 컴포넌트를 저장 및 실행할 수 있는 컴퓨팅 장치(600)의 예시적인 컴퓨터 아키텍처를 나타내고 있다. 도 6에 도시된 컴퓨터 아키텍처는 종래의 서버, 데스크톱 컴퓨터, 랩톱, 태블릿, 패블릿, 네트워크 장비, PDA, e-리더, 디지털 셀룰러 폰 혹은 다른 컴퓨팅 장치가 될 수 있고, 이는 본 명세서에 개시된 임의의 소프트웨어 컴포넌트를 실행시키는데 사용될 수 있다. 예컨대, 도 6에 도시된 컴퓨터 아키텍처는 상술한 임의의 소프트웨어 컴포넌트를 실행시키는데 사용될 수 있다.

[0072] 컴퓨팅 장치(600)는 베이스보드(602) 즉 "마더보드"를 포함하고, 이는 다수의 컴포넌트나 장치가 시스템 버스나 다른 전기 통신 경로를 이용해서 접속될 수 있는 인쇄 회로 기판이다. 일 예시적인 측면에서, 하나 이상의 CPU(604)가 칩셋(606)과 함께 동작한다. CPU(604)는, 컴퓨팅 장치(600)의 동작에 필요한 산술 연산 및 논리 연산을 수행하는 표준 프로그래밍 가능 프로세서가 될 수 있다.

[0073] CPU(604)는 하나의 개별적인 물리적인 상태와 다음 상태를 구별해서 변경하는 스위칭 소자를 조작해서, 하나의 상태를 다음 상태로 천이시킴으로써 동작을 수행한다. 스위칭 소자는 일반적으로, 플립-플롭과 같이, 2개의 이진 상태 중 하나를 유지하는 전자 회로, 및 논리 게이트와 같이, 하나 이상의 다른 스위칭 소자의 상태의 논리적인 조합에 기초해서 출력 상태를 제공하는 전자 회로를 포함할 수 있다. 이들 기본 스위칭 소자를 조합해서, 레지스터, 가산기-감산기, 산술 논리 유닛, 부동 소수점 유닛 등을 포함한, 하나 이상의 복잡한 논리 회로를 생성할 수 있다.

[0074] 칩셋(606)은, CPU(604)와 베이스보드(602) 상의 나머지 컴포넌트 및 장치와의 사이의 인터페이스를 제공한다. 칩셋(606)은 컴퓨팅 장치(600) 내에서 주메모리로서 사용되는 RAM(608)에 대한 인터페이스를 제공할 수 있다. 칩셋(606)은, 또한 컴퓨팅 장치(600)의 시작 및 다양한 컴포넌트와 장치 사이의 정보 전달을 돕는 기초 루틴을 저장하는 ROM(610) 혹은 NVRAM과 같은 컴퓨터-판독 가능 저장 매체에 대한 인터페이스를 제공할 수 있다. 이 ROM(610) 혹은 NVRAM은 또한 본 명세서에 설명된 측면에 따른 컴퓨팅 장치(600)의 동작에 필요한 다른 소프트웨어 컴포넌트도 저장할 수 있다.

[0075] 컴퓨팅 장치(600)는, 네트워크를 통한 원격 컴퓨팅 장치 및 컴퓨터 시스템으로의 논리 접속을 이용하는, 네트워크

크화된 환경에서 동작할 수 있다. 칩셋(606)은, 기가비트 이더넷 어댑터와 같은, NIC(network interface controller)(612)를 통한 네트워크 접속을 제공하는 기능을 포함할 수 있다. NIC(612)는 네트워크(620)를 통해서 컴퓨팅 장치(600)를 다른 컴퓨팅 장치에 접속시킬 수 있다. 컴퓨팅 장치(600)에는, 컴퓨터를 다른 타입의 네트워크 및 원격 컴퓨터 시스템에 접속시키는 NIC(612)가 다수 존재할 수 있다는 것을 이해할 것이다. 컴퓨팅 장치(600)는 LAN(620)을 통해서, 원격 컴퓨터(650)와 같은 원격 서비스 및 서버와 통신할 수 있다.

[0076] 컴퓨팅 장치(600)는, 컴퓨팅 장치용 비휘발성 저장부를 제공하는 대용량 저장 장치(616)에 접속될 수 있다. 대용량 저장 장치(616)는, 본 명세서에 상세하게 설명된 시스템 프로그램, 애플리케이션 프로그램, 다른 프로그램 모듈 및 데이터를 저장할 수 있다. 대용량 저장 장치(616)는, 칩셋(606)에 접속된 저장 제어부(614)를 통해서 컴퓨팅 장치(600)에 접속될 수 있다. 대용량 저장 장치(616)는, 하나 이상의 물리적인 저장 유닛으로 이루어질 수 있다. 저장 제어부(614)는 SAS(serial attached SCSI) 인터페이스, SATA 인터페이스, FC(fiber channel) 인터페이스, 혹은 컴퓨터와 물리적인 저장 유닛 사이의 물리적인 접속 및 데이터 전송을 위한 다른 타입의 인터페이스를 통해서 물리적인 저장 유닛과 인터페이스할 수 있다. 대용량 저장 장치(616), 다른 저장 매체 및 저장 제어부(614)가 MMC(MultiMediaCard) 컴포넌트, eMMC 컴포넌트, SD(Secure Digital) 컴포넌트, PCI Express 컴포넌트 등을 포함할 수 있다는 것도 이해할 것이다.

[0077] 컴퓨팅 장치(600)는, 저장되는 정보를 반영하도록 물리적인 저장 유닛의 물리적인 상태를 변형하는 방식으로, 대용량 저장 장치(616)에 데이터를 저장할 수 있다. 물리적인 상태의 특정한 변형은, 본 명세서의 다양한 구현예의 다양한 요인에 따를 수 있다. 이러한 요인의 예로는, 물리적인 저장 유닛을 구현하는데 사용될 수 있는 기술, 대용량 저장 장치(616)가 주 저장부인지 부 저장부인지 여부 등을 들 수 있지만, 이것으로 한정되는 것은 아니다.

[0078] 예컨대, 컴퓨팅 장치(600)는 저장 제어부(614)를 통해서 명령어를 발행해서, 자기 디스크 드라이브 유닛 내의 특정 위치의 자기 특성, 광학식 저장 유닛 내의 특정 위치의 반사 특성이나 굴절 특성, 혹은 고체 상태 저장 유닛 내의 특정 캐패시터, 트랜지스터 혹은 다른 개개의 컴포넌트의 전기적인 특성을 변경함으로써 정보를 대용량 저장 장치(616)에 저장할 수 있다. 본 명세서의 사상 및 범주를 벗어나지 않고 물리적인 매체의 다른 변형이 가능하며, 상기 예시는 본 명세서를 용이하게 하기 위해서만 제공된다. 컴퓨팅 장치(600)는 또한, 물리적인 저장 유닛 내의 하나 이상의 특정 위치의 물리적인 상태나 특성을 검출함으로써 대용량 저장 장치(616)에서 정보를 판독할 수 있다.

[0079] 상술한 대용량 저장 장치(616)에 더해서, 컴퓨팅 장치(600)는 다른 컴퓨터-판독 가능 저장 매체에 액세스해서, 프로그램 모듈, 데이터 구조 혹은 다른 데이터와 같은 정보를 저장하고 취득할 수 있다. 따라서, 런타임 시스템(108) 및 다른 모듈이 대용량 저장 장치(616) 내에 저장된 데이터 및 소프트웨어로 도시되어 있지만, 런타임 시스템(108) 및/또는 다른 모듈은 적어도 부분적으로 컴퓨팅 장치(600)의 다른 컴퓨터-판독 가능 저장 매체에 저장될 수 있다는 것을 이해할 것이다. 본 명세서에 포함된 컴퓨터-판독 가능 매체의 설명이, 고체 상태 드라이브, 하드디스크 혹은 CD-ROM 드라이브와 같은 대용량 저장 장치를 참조하고 있지만, 당업자라면, 컴퓨터-판독 가능 매체는, 컴퓨팅 장치(600)에 의한 액세스가 가능한 임의의 이용 가능한 컴퓨터 저장 매체 혹은 통신 매체가 될 수 있다는 것을 이해할 것이다.

[0080] 통신 매체는 컴퓨터 판독 가능 명령어, 데이터 구조, 프로그램 모듈, 혹은 반송파나 다른 전송 매커니즘과 같은 변조된 데이터 신호의 다른 데이터를 포함하고, 또한 임의의 전달 매체를 포함한다. 용어 '변조된 데이터 신호'란, 신호에 정보를 인코딩하는 방식으로, 그 특성 중 하나 이상이 변경 혹은 설정된 신호를 의미한다. 예컨대, 통신 매체로는, 유선 네트워크나 다이렉트-유선 접속과 같은 유선 매체와, 음향, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 들 수 있지만, 이것으로 한정되는 것은 아니다. 컴퓨터-판독 가능 매체의 범주에는 이들의 임의의 조합도 포함된다.

[0081] 예컨대, 컴퓨터 저장 매체로는, 컴퓨터-판독 가능 명령어, 데이터 구조, 프로그램 모듈 혹은 다른 데이터와 같은 정보를 저장하는 임의의 방법이나 기술로 구현된, 휘발성 및 비휘발성 매체, 착탈 가능형 및 착탈 불가능형 매체를 들 수 있지만, 이것으로 한정되는 것은 아니다. 예컨대, 컴퓨터 매체로는 RAM, ROM, EPROM, EEPROM, 플래시 메모리나 다른 고체 상태 메모리 기술, CD-ROM, DVD, HD-DVD, BLU-RAY, 혹은 다른 광학식 저장부, 자기식 카세트, 자기식 테이프, 자기식 디스크 저장부나 다른 자기식 저장 장치, 혹은 소량의 정보를 저장하는데 사용될 수 있으며 컴퓨팅 장치(600)에 의한 액세스가 가능한 임의의 다른 매체를 들 수 있지만, 이것으로 한정되는 것은 아니다. 청구항에서, 용어 '컴퓨터 저장 매체', '컴퓨터-판독 가능 저장 매체' 및 그 유사한 매체는, 파형이나 신호 자체 및/또는 통신 매체는 포함하지 않는다.

- [0082] 대용량 저장 장치(616)는 컴퓨팅 장치(600)의 동작을 제어하는데 사용되는 운영 체제(622)를 저장할 수 있다. 일 측면에 따르면, 운영 체제는 LINUX 운영 체제를 포함한다. 다른 측면에 따르면, 운영 체제는 MICROSOFT사의 WINDOWS® 운영 체제를 포함한다. 또 다른 측면에 따르면, 운영 체제는 UNIX, 안드로이드, 윈도우 전화 혹은 iOS 운영 체제를 포함할 수 있다. 다른 운영 체제가 사용될 수도 있다는 것을 이해할 것이다. 대용량 저장 장치(616)는, 런타임 시스템(108) 및/또는 상술한 다른 임의의 소프트웨어 컴포넌트 및 데이터와 같은, 컴퓨팅 장치(600)가 사용하는 다른 운영 체제나 애플리케이션 프로그램 및 데이터를 저장할 수 있다. 대용량 저장 장치(616)는 또한 본 명세서에 명시적으로 나타내지 않은 다른 프로그램 및 데이터를 저장할 수도 있다.
- [0083] 일 측면에서, 대용량 저장 장치(616) 혹은 다른 컴퓨터 판독 가능 저장 매체에는 컴퓨터-실행 가능 명령어가 인코딩되고, 이는 컴퓨팅 장치(600)에 로딩될 때, 컴퓨터를, 다목적 컴퓨팅 시스템으로부터 본 명세서에 개시된 측면을 구현할 수 있는 특수 목적 컴퓨터로 변환시킨다. 컴퓨터-실행 가능 명령어는, 상술한 바와 같이, CPU(604)가 상태 사이에서 어떻게 천이하는지를 명시함으로써 컴퓨팅 장치(600)를 변환한다. 일 측면에 따르면, 컴퓨팅 장치(600)는 컴퓨터-실행 가능 명령어가 저장되어 있는 컴퓨터 판독 가능 저장 매체에 액세스하고, 컴퓨터-실행 가능 명령어는 도 3 및 4와 관련해서 상기 설명한 다양한 루틴을 수행한다. 컴퓨팅 장치(600)는 또한, 상술한 임의의 다른 컴퓨터-구현식 동작을 수행하는 컴퓨터 판독 가능 저장 매체를 포함할 수 있다.
- [0084] 컴퓨팅 장치(600)는 입력 장치(619)로부터의 입력을 수신해서 처리하는, 입출력 제어부(617)를 하나 이상 포함할 수 있다. 입력 장치(619)는 키보드, 마우스, 마이크, 헤드셋, 터치 패드, 터치 스크린, 전자 스타일러스, 혹은 임의의 다른 타입의 입력 장치와 같은, 다수의 입력 장치를 포함할 수 있다. 유사하게, 입출력 제어부(617)는 컴퓨터 모니터, 플랫 패널 디스플레이, 디지털 프로젝터, 프린터, 플로터, 혹은 다른 타입의 출력 장치와 같은, 디스플레이에 출력을 제공할 수 있다. 컴퓨팅 장치(600)는 도 6에 도시된 컴포넌트를 모두 포함하지 않을 수도 있고, 도 6에 분명하게 도시되지 않은 다른 컴포넌트를 포함할 수도 있으며, 혹은 도 6에 도시되지 않은 전혀 다른 아키텍처를 이용할 수도 있다는 점을 이해할 것이다.
- [0085] 본 명세서에 개시된 것은 이하의 예의 관점에서 고려될 수 있다.
- [0086] 예 1 : 이벤트를 전달하는 컴퓨터-구현된 방법으로서, 이 방법은, 이벤트의 발행을 수신하는 단계와, 처리 에이전트의 구성에 기초해서 이벤트와 관련된 정보를 수신할 처리 에이전트 - 이 처리 에이전트는 이벤트를 수신하도록 명시적으로 가입되어 있지 않음 - 를 식별하는 단계와, 처리 에이전트가 가동되는지 여부를 판정하는 단계와, 처리 에이전트가 가동되지 않는다고 판정되는 것에 응답해서, 처리 에이전트를 가동시키는 단계와, 이벤트와 관련된 정보를 처리 에이전트에 전달하는 단계와, 처리 에이전트로부터, 하나 이상의 이벤트에 대한 가입을 수신하는 단계와, 하나 이상의 이벤트를 처리 에이전트에 전달하는 단계를 포함한다.
- [0087] 예 2 : 예 1의 방법으로서, 가입이, 전달될 하나 이상의 과거 이벤트를 명시하고 있는지 여부를 판정하는 단계와, 가입이, 전달될 하나 이상의 과거 이벤트를 명시하고 있다고 판정되는 것에 응답해서, 하나 이상의 과거 이벤트를 처리 에이전트에 전달하는 단계를 더 포함한다.
- [0088] 예 3 : 예 1 및 2의 방법으로서, 이벤트의 발행을 수신하는 단계는, 이벤트를 큐에 발행하는 요청을 수신하는 단계와, 이벤트에 대응하는 가상 스트림을 가동시키는 단계와, 가상 스트림을 통해서 큐에 이벤트를 발행하는 단계를 포함한다.
- [0089] 예 4 : 예 1 내지 3 중 어느 하나의 방법으로서, 하나 이상의 이벤트에 대한 가입은 가상 스트림을 정의하고 있고, 가입은, 하나 이상의 이벤트에 대응하는 가상 스트림에 대한 가입이다.
- [0090] 예 5 : 예 1 내지 4 중 어느 하나의 방법으로서, 처리 에이전트를 가동시키는 단계는 처리 에이전트에 시스템 리소스를 할당하는 단계를 포함하고, 가상 스트림에 대한 가입은 가상 스트림을 가동시키고, 가상 스트림에 시스템 리소스가 할당되게 한다.
- [0091] 예 6 : 예 1 내지 5 중 어느 하나의 방법으로서, 일정 기간 동안 가상 스트림에 어떤 이벤트도 수신되지 않으면, 가상 스트림에 할당된 시스템 리소스를 회수함으로써 가상 스트림을 가동 중지시키는 단계와, 처리 에이전트가 소정 기간 동안 유휴 상태로 되어 있으면, 처리 에이전트에 할당된 시스템 리소스를 회수함으로써 처리 에이전트를 가동 중지시키는 단계를 더 포함한다.
- [0092] 예 7 : 예 1 내지 6 중 어느 하나의 방법으로서, 처리 에이전트의 구성은 프로그램적으로 획득되거나 혹은 처리 에이전트의 선언적 정의로부터 결정된다.
- [0093] 예 8 : 예 1 내지 7 중 어느 하나의 방법으로서, 처리 에이전트는, 처리 에이전트가 가동되는지 아닌지 여부와

무관하게 이벤트를 전달하도록 어드레싱될 수 있다.

- [0094] 예 9 : 컴퓨터-실행 가능 명령어가 저장된 컴퓨터-판독 가능 저장 매체로서, 명령어는 컴퓨터에 의해 실행될 때 컴퓨터로 하여금, 큐에서 발행되었지만 상기 처리 에이전트가 명시적으로 가입되어 있지 않은 이벤트를 수신할 처리 에이전트를 결정하게 하고 - 이벤트는 자신에 대응하는 제 1 가상 스트림을 통해서 큐에 발행됨 - , 처리 에이전트가 가동되는지 판정하게 하며, 처리 에이전트가 가동되지 않는다고 판정되는 것에 응답해서, 처리 에이전트를 가동하게 하고, 이벤트의 정보가 처리 에이전트에 전달되게 하고, 명시적 가입에 의해 정의된 제 2 가상 스트림에 대한 명시적 가입을 처리하게 하며 - 명시적 가입은, 이벤트 정보가 처리 에이전트에 전달되는 것에 응답해서, 처리 에이전트에 의해 제시됨 - , 제 2 가상 스트림 내의 이벤트가 처리 에이전트에 전달되게 한다.
- [0095] 예 10 : 예 9의 컴퓨터-판독 가능 저장 매체로서, 큐에서 발행되는 이벤트는 유지된다.
- [0096] 예 11 : 예 9 및 10의 컴퓨터-판독 가능 저장 매체로서, 명령어는 컴퓨터에 의해 실행될 때 컴퓨터로 하여금 또한, 제 2 가상 스트림이 큐에 발행된 과거 이벤트를 포함하는지 판정하게 하고, 제 2 가상 스트림이 큐에 발행된 과거 이벤트를 포함한다고 판정되는 것에 응답해서, 과거 이벤트를 제 2 가상 스트림에 푸시해서 처리 에이전트에 전달되게 한다.
- [0097] 예 12 : 예 9 내지 11의 컴퓨터-판독 가능 저장 매체로서, 명령어는 컴퓨터에 의해 실행될 때 컴퓨터로 하여금 또한, 일정 기간 동안 가상 스트림에 어떤 이벤트도 수신되지 않으면, 가상 스트림에 할당된 시스템 리소스를 회수함으로써 가상 스트림을 가동 중지하게 하고, 처리 에이전트가 소정 기간 동안 유휴 상태로 되어 있으면, 처리 에이전트에 할당된 시스템 리소스를 회수함으로써 처리 에이전트를 가동 중지하게 한다.
- [0098] 예 13 : 예 9 내지 12의 컴퓨터-판독 가능 저장 매체로서, 처리 에이전트를 가동시키는 것은, 처리 에이전트에 시스템 리소스를 할당하는 것을 포함하고, 제 2 가상 스트림에 대한 가입은 가상 스트림을 가동시키고, 제 2 가상 스트림에 시스템 리소스가 할당되게 하는 것을 포함한다.
- [0099] 예 14 : 예 9 내지 13의 컴퓨터-판독 가능 저장 매체로서, 처리 에이전트는, 처리 에이전트가 가동되는지 아닌지 여부와 무관하게 이벤트를 전달하도록 어드레싱될 수 있다.
- [0100] 예 15 : 예 9 내지 14의 컴퓨터-판독 가능 저장 매체로서, 처리 에이전트의 구성은 프로그램적으로 획득되거나 혹은 처리 에이전트의 선언적 정의로부터 결정된다.
- [0101] 예 16 : 런타임을 실행하는 하나 이상의 컴퓨팅 장치를 포함하는 시스템으로서, 런타임은, 이벤트의 발행을 수신하고 유지하는 큐를 관리하고, 제 1 가상 스트림에서 이벤트의 발행을 수신하며, 이벤트의 발행을 큐로 포워딩하고, 처리 에이전트의 구성에 기초해서, 이벤트의 이벤트 정보를 전달하는 처리 에이전트를 식별하며 - 처리 에이전트는, 이벤트를 수신하도록 명시적으로 가입되어 있지 않고, 처리 에이전트가 가동되는지 아닌지 여부와 무관하게 이벤트 정보를 전달하도록 어드레싱될 수 있음 - , 처리 에이전트가 가동되는지 판정하고, 처리 에이전트가 가동되지 않는다고 판정되는 것에 응답해서, 처리 에이전트를 가동시키며, 이벤트 정보를 처리 에이전트에 전달하고, 이벤트 정보를 처리 에이전트에 전달한 이후에, 처리 에이전트로부터, 제 2 가상 스트림의 이벤트에 대한 명시적 가입을 수신하며, 명시적 가입을 수신한 이후에, 명시적 가입이 큐에 발행된 과거 이벤트를 명시하고 있는지 여부를 판정하고, 명시적 가입이 과거 이벤트를 명시하고 있다고 판정되는 것에 응답해서, 큐에 발행된 과거 이벤트 중 하나를 취득하고, 과거 이벤트를 제 2 가상 스트림에 푸시하며, 제 2 가상 스트림 내의 이벤트를 처리 에이전트에 전달하도록 구성된다.
- [0102] 예 17 : 예 16의 시스템으로서, 처리 에이전트를 가동시키는 것은, 처리 에이전트에 시스템 리소스를 할당하는 것을 포함하고, 제 2 가상 스트림에 대한 가입은 가상 스트림을 가동시키고, 제 2 가상 스트림에 시스템 리소스가 할당되게 한다.
- [0103] 예 18 : 예 16 및 17의 시스템으로서, 하나 이상의 처리 에이전트를 실행하는 복수의 서버를 더 포함하고, 런타임은, 처리 에이전트가 가동되던 서버 중 하나가 고장난 이후에, 처리 에이전트를 다른 서버에서 재가동시킴으로써 처리 에이전트를 복구하도록 더 구성된다.
- [0104] 예 19 : 예 16 내지 18 중 어느 하나의 시스템으로서, 처리 에이전트의 구성은 프로그램적으로 획득되거나 혹은 처리 에이전트의 선언적 정의로부터 결정된다.
- [0105] 예 20 : 예 16 내지 19 중 어느 하나의 시스템으로서, 런타임은, 일정 기간 동안 제 2 가상 스트림에 어떤 이벤트도 수신되지 않으면, 가상 스트림에 할당된 시스템 리소스를 회수함으로써 가상 스트림을 가동 중지시키고, 처리 에이전트가 소정 기간 동안 유휴 상태로 되어 있으면, 처리 에이전트에 할당된 시스템 리소스를 회수함으

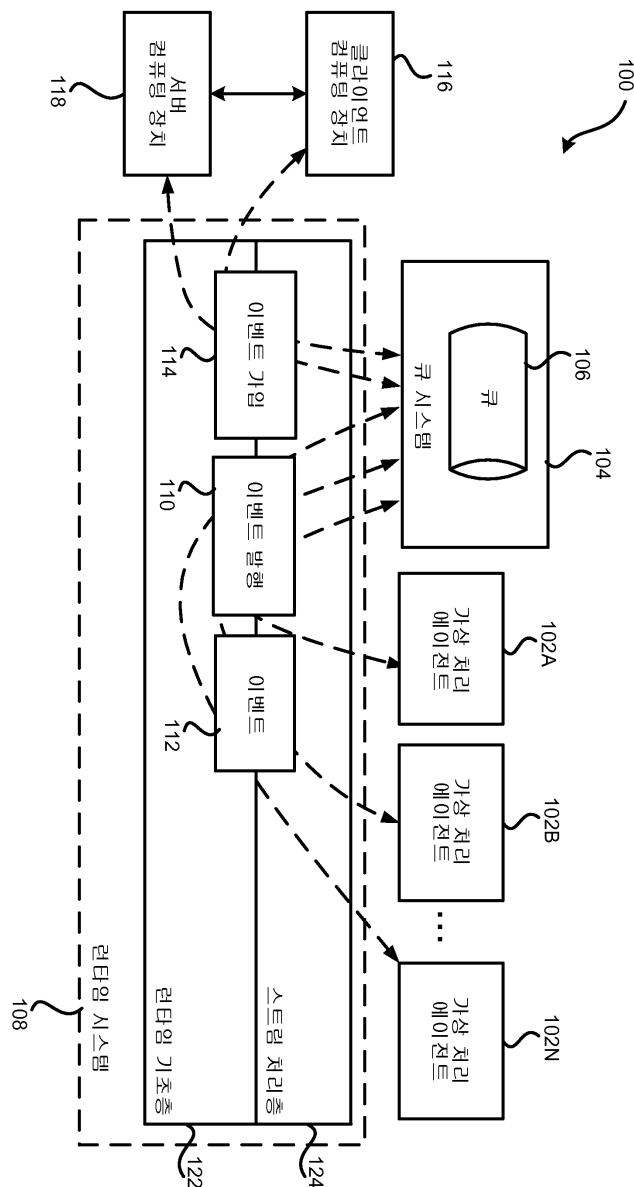
로써 처리 에이전트를 가동 중지시키도록 더 구성된다.

[0106] 상기 설명에 기초해서, 본 명세서에는 이벤트에 가입하고, 이벤트를 수신하고, 이벤트를 처리하는 개념 및 기술이 개시된다는 것을 이해할 것이다. 본 명세서에 개시된 청구 대상을 컴퓨터 구조 특성, 방법론적 액트 및 컴퓨터 판독 가능 매체로 특정된 표현으로 설명했지만, 첨부된 청구항에 정의된 방법이 반드시 본 명세서에 개시된 특정한 특성, 액트 및 매체로 한정되는 것은 아니다. 특정한 특성, 액트 및 매체는 청구항을 구현하는 예시적인 형태로서 개시된 것이다.

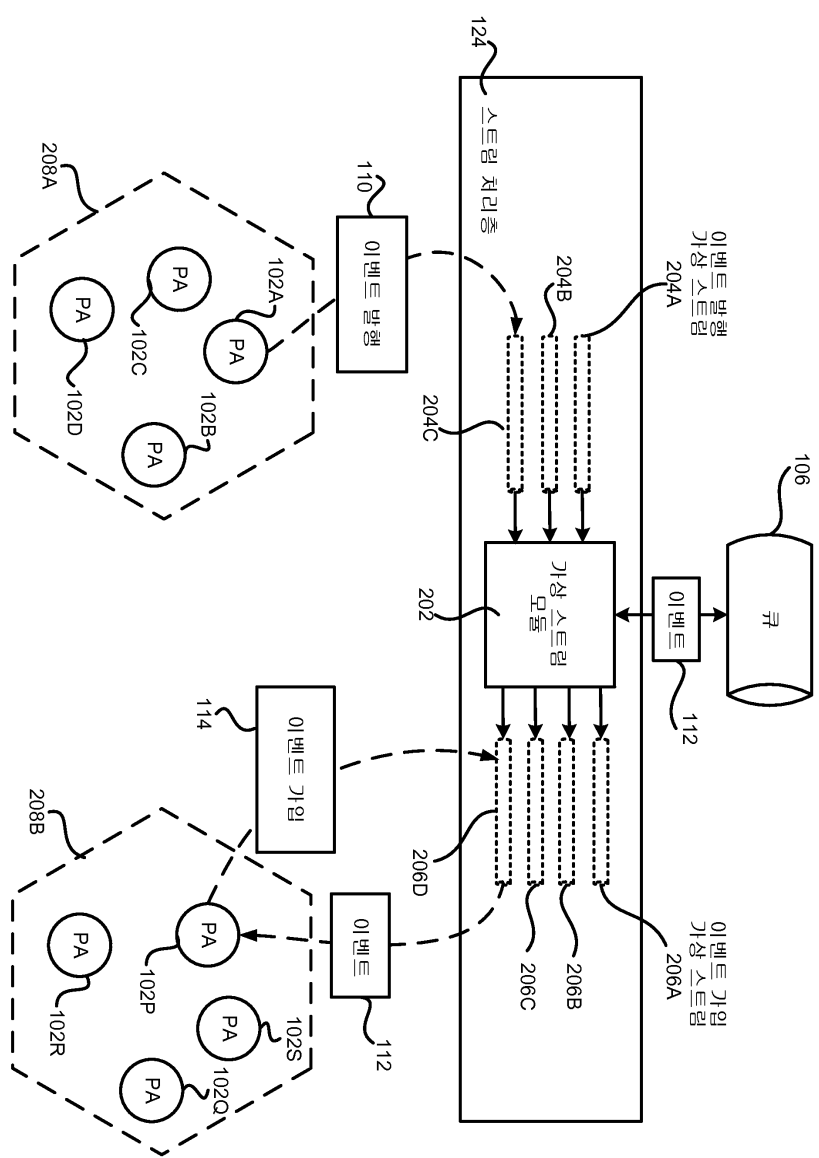
[0107] 상술한 청구 대상은 예시적인 것으로 한정 의미의 의미가 아니다. 도시되고 설명된 예시적인 측면이나 응용예를 따르지 않고, 청구항에 개시된 본 발명의 진정한 사상 및 범주로부터 벗어남없이, 본 명세서에 개시된 청구 대상에 대한 다양한 수정 및 변경이 있을 수 있다.

도면

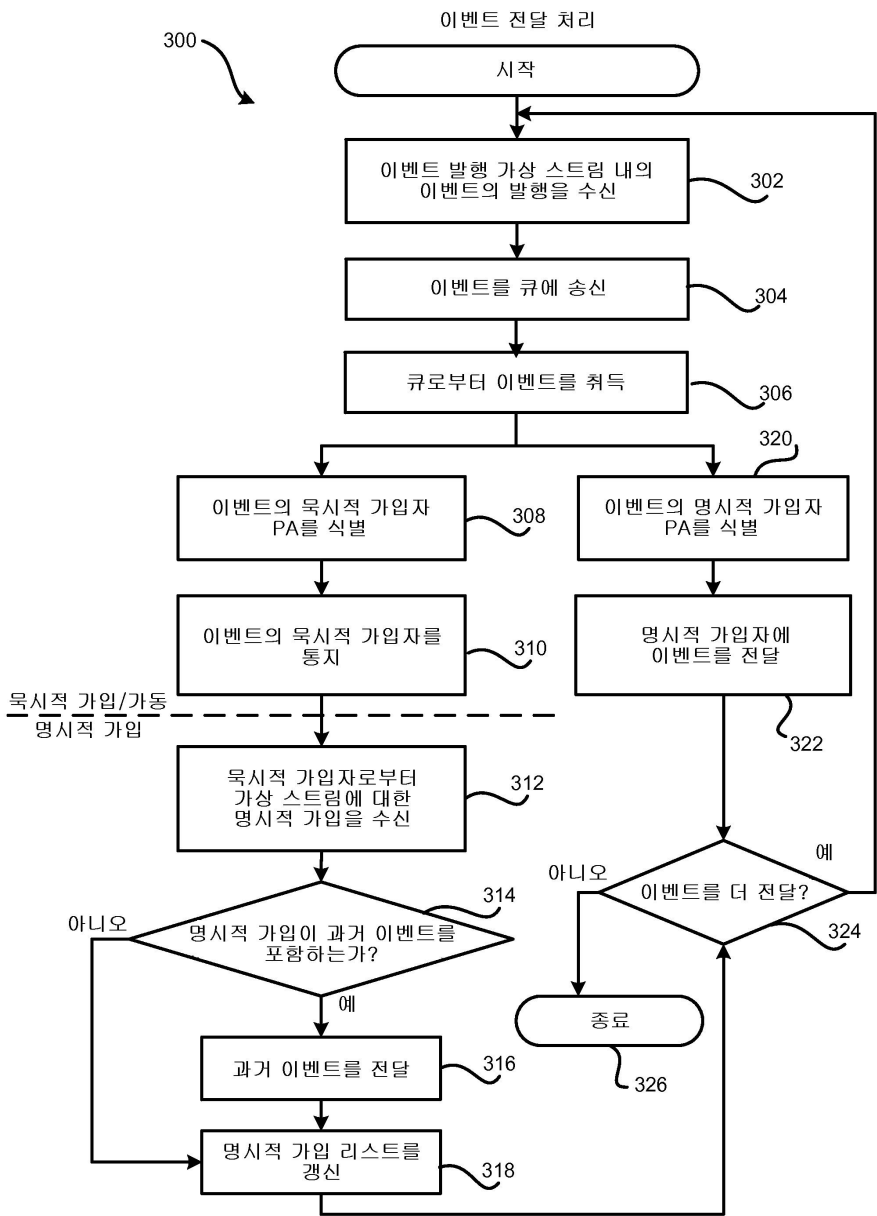
도면1



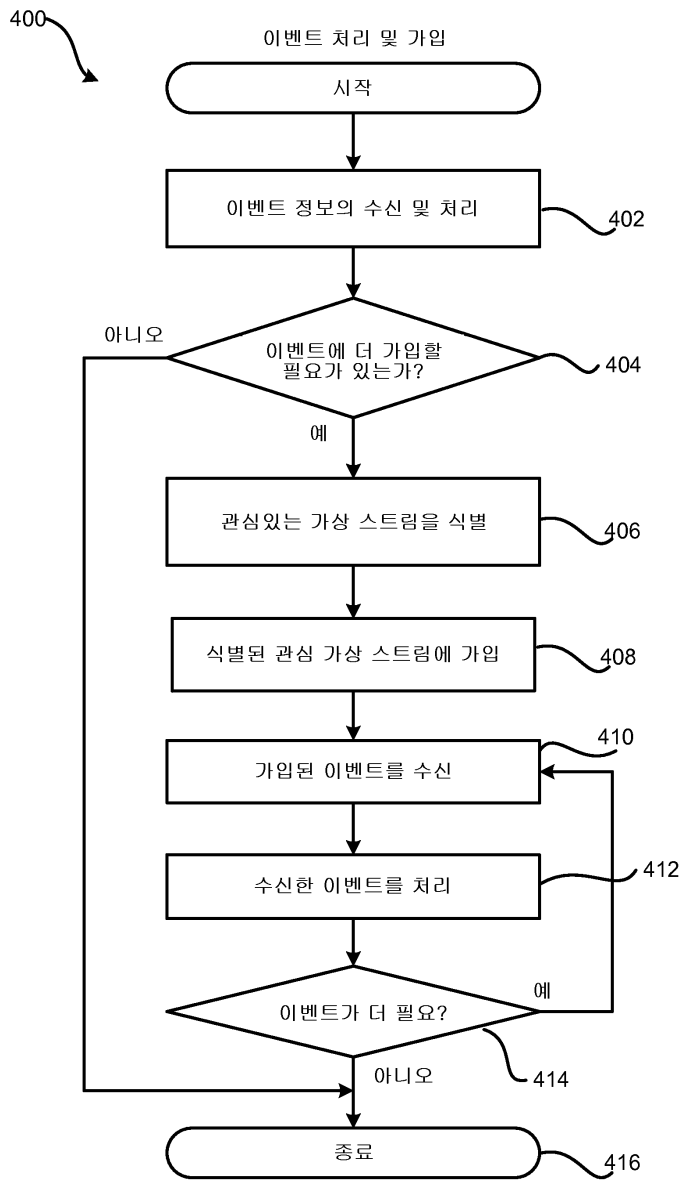
도면2



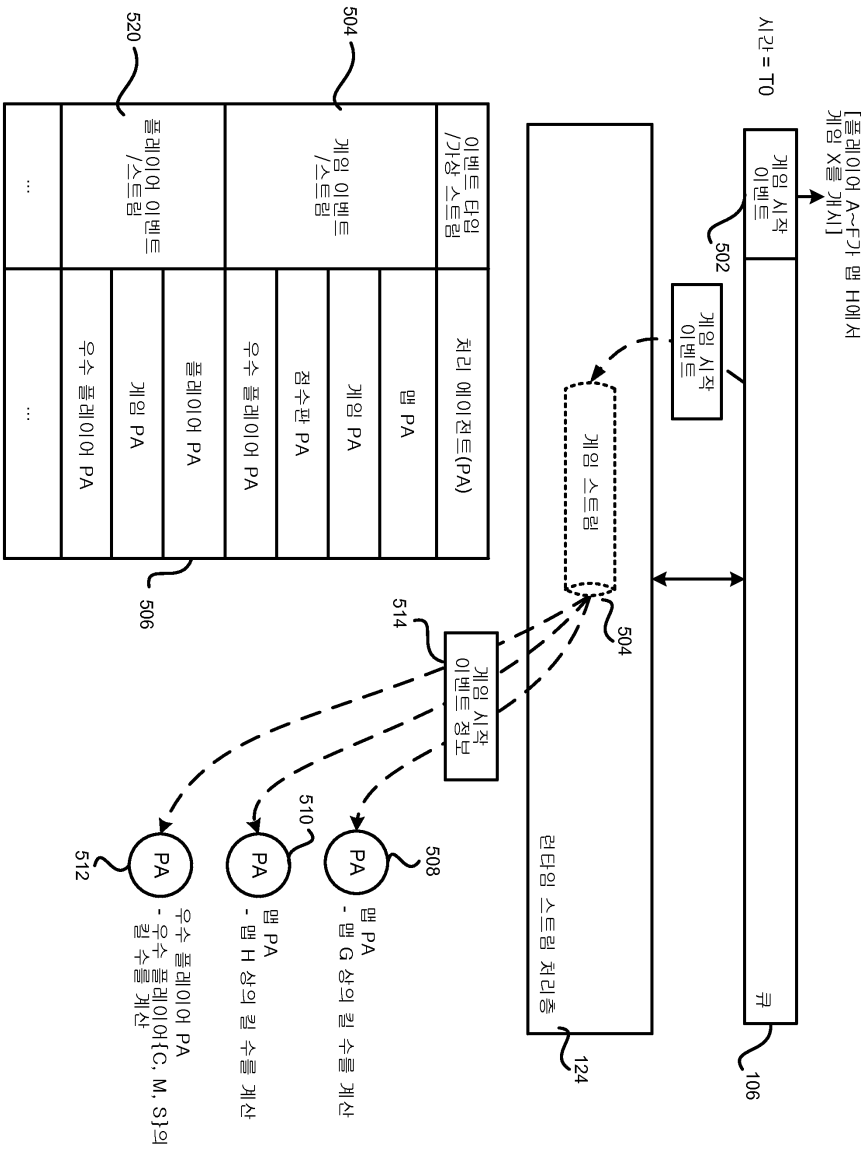
도면3



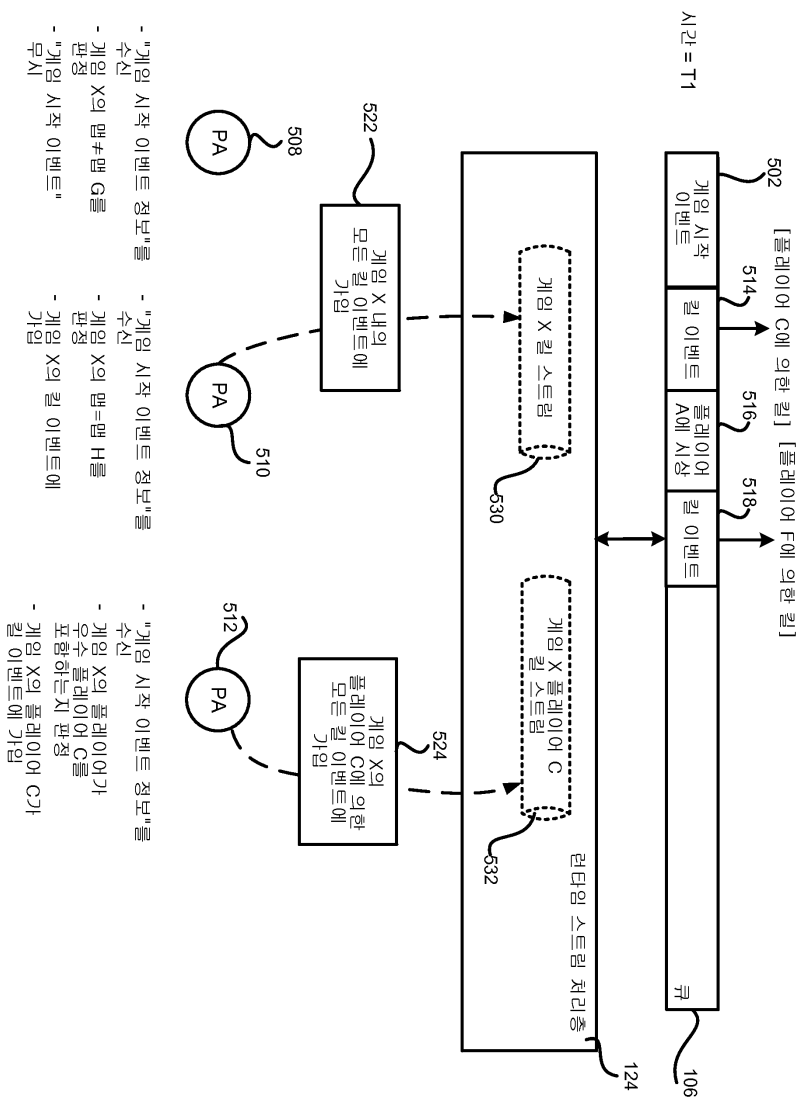
도면4



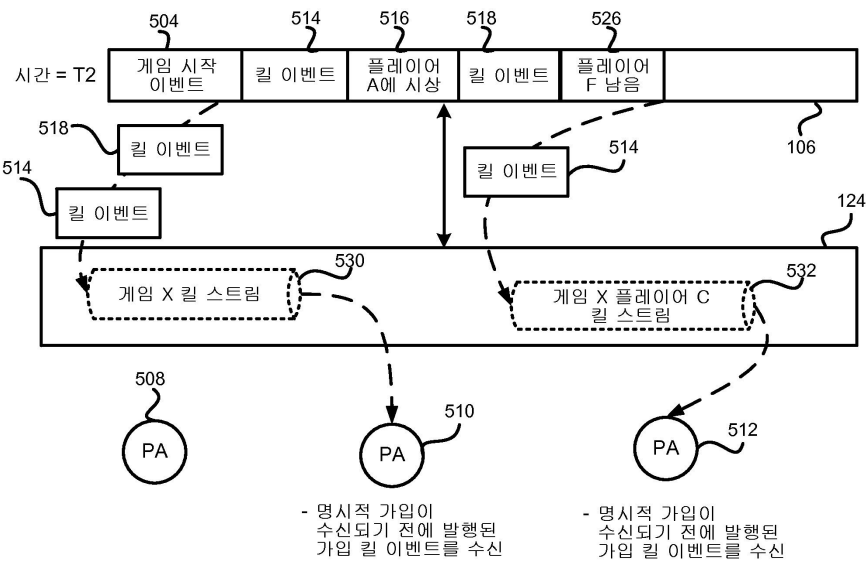
도면 5a



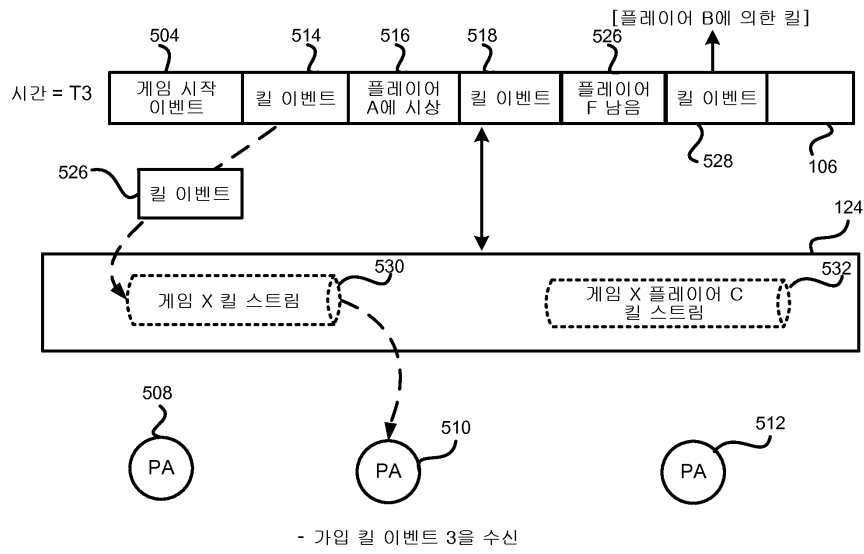
도면5b



도면5c



도면5d



도면6

