



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년05월10일
(11) 등록번호 10-1856454
(24) 등록일자 2018년05월03일

(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2018.01) G06F 9/50 (2018.01)
(52) CPC특허분류
G06F 9/4843 (2013.01)
G06F 9/505 (2013.01)
(21) 출원번호 10-2017-0028383
(22) 출원일자 2017년03월06일
심사청구일자 2017년03월06일
(56) 선행기술조사문헌
JP2013502642 A*
(뒷면에 계속)

(73) 특허권자
주식회사 티맥스데이터
경기도 성남시 분당구 정자일로 45(금곡동, 티맥스타워)
(72) 발명자
황창호
경기도 성남시 분당구 정자일로 30 계룡아파트 107동 903호
최승탁
경기도 성남시 분당구 성남대로 165 639호
(74) 대리인
이대호, 박건홍

전체 청구항 수 : 총 14 항

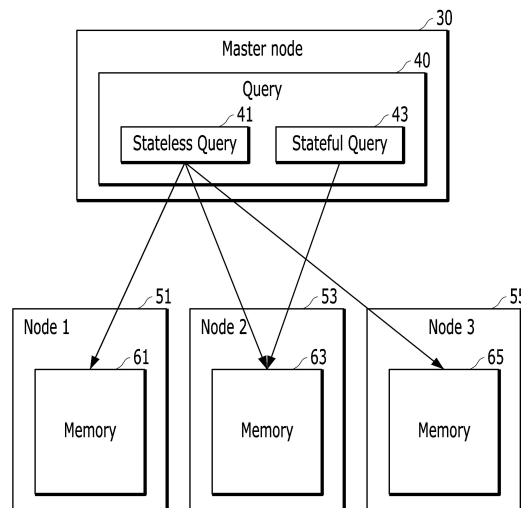
심사관 : 유진태

(54) 발명의 명칭 분산처리를 위한 컴퓨팅 장치

(57) 요약

본 개시의 일 실시예에 따라 인코딩된 명령들을 포함하는 컴퓨터 판독 가능 저장 매체에 저장된 컴퓨터 프로그램이 개시된다. 상기 컴퓨터 프로그램은 컴퓨터 시스템의 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 하나 이상의 프로세서들로 하여금 분산처리를 위한 이하의 단계들을 수행하도록 하며, 상기 단계들은: 프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하는 단계; 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계; 상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계; 및 각각의 노드에서 이벤트를 수신하는 단계를 포함한다.

대표도 - 도2



(56) 선행기술조사문헌

KR1020150104585 A*

KR1020160064569 A

KR1020090035545 A

WO2015149830 A1

JP2016504679 A

*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호 10051353

부처명 산업통상자원부

연구관리전문기관 한국산업기술평가관리원

연구사업명 우수기술연구센터(ATC) 기술개발사업

연구과제명 다양한 머신 데이터 분석을 위한 빅데이터 처리 토탈 플랫폼 개발

기 여 율 1/1

주관기관 ㈜티맥스데이터

연구기간 2015.06.01 ~ 2017.05.31

명세서

청구범위

청구항 1

인코딩된 명령들을 포함하는 컴퓨터 판독 가능 저장 매체에 저장된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 컴퓨터 시스템의 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 하나 이상의 프로세서들로 하여금 분산처리를 위한 이하의 단계들을 수행하도록 하며, 상기 단계들은:

프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하는 단계;

상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계;

상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계;

각각의 노드에서 이벤트를 수신하는 단계; 및

상기 분산처리 쿼리 및 상기 집중처리 쿼리 중 어느 하나에서 상기 이벤트의 처리 가부에 기초하여, 상기 분산처리 쿼리 및 상기 집중처리 쿼리 중 적어도 하나가 상기 이벤트를 처리하는 단계;

를 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 2

제 1 항에 있어서,

수신된 상기 이벤트를 상기 분산처리 쿼리가 처리할 수 있는 경우, 상기 분산처리 쿼리에서 상기 이벤트를 처리하는 단계;

를 더 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 3

제 1 항에 있어서,

상기 이벤트를 상기 분산처리 쿼리가 처리할 수 없는 경우, 상기 이벤트를 집중처리 노드로 전달할 것을 결정하는 단계;

를 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 4

제 1 항에 있어서,

상기 분산처리 쿼리는 각각의 이벤트에 대하여 연산을 처리하는 쿼리를 포함하고, 그리고 상기 집중처리 쿼리는 둘 이상의 이벤트를 종합하여 연산을 처리하는 쿼리를 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 5

제 1 항에 있어서,
 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계는,
 상기 분산처리 쿼리를 상기 분산처리 노드 각각의 메모리에 등록시키는 단계;
 를 포함하고, 그리고
 상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계는,
 상기 집중처리 쿼리를 상기 집중처리 노드의 메모리에 등록시키는 단계;
 를 포함하는,
 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 6

제 1 항에 있어서,
 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계는,
 상기 분산처리 쿼리에 상기 분산처리 쿼리가 처리할 수 없는 이벤트가 수신되는 경우, 해당 이벤트를 집중처리 노드로 전송하도록 하는 명령을 추가하여, 상기 분산처리 쿼리를 상기 분산처리 노드에 등록시키는 단계;
 를 포함하는,
 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 7

제 3 항에 있어서,
 상기 이벤트를 상기 분산처리 쿼리가 처리할 수 없는 경우, 상기 이벤트를 집중처리 쿼리로 전송할 것을 결정하는 단계는,
 상기 이벤트를 수신한 노드의 전용 어댑터(adapter)가 상기 집중처리 노드의 전용 어댑터에 상기 이벤트를 전송하는 단계;
 를 포함하는,
 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 8

제 7 항에 있어서,
 상기 이벤트를 수신한 노드의 전용 어댑터(adapter)가 상기 집중처리 노드의 전용 어댑터에 상기 이벤트를 전송하는 단계는,
 상기 이벤트를 수신한 노드와 상기 집중처리 노드가 동일한 노드인 경우, 메모리 상에서 상기 분산처리 쿼리에서 상기 집중처리 쿼리로 상기 이벤트를 전송하고, 그리고 상기 이벤트를 수신한 노드와 상기 집중처리 노드가 상이한 노드인 경우, TCP 또는 HTTP 프로토콜을 이용하여 상기 집중처리 쿼리로 상기 이벤트를 전송하는 단계;
 를 포함하는,
 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 9

제 1 항에 있어서,
상기 집중처리 노드는,
상기 둘 이상의 노드 중에 상기 분산처리 노드와 동일하거나 또는 별도의 노드이며 그리고 부하를 분산할 수 있도록 결정되는,
컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 10

제 2 항에 있어서,
상기 이벤트를 분산처리 쿼리가 처리한 경우, 상기 이벤트의 분산처리 쿼리 처리 결과를 상기 집중처리 노드에 전송하는 단계;
를 더 포함하는,
컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 11

제 2 항에 있어서,
집중처리 노드에서 상기 이벤트의 분산처리 쿼리 처리 결과를 수신하여 상기 집중처리 쿼리를 처리하는 단계;
를 더 포함하는,
컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 12

제 1 항에 있어서,
상기 쿼리는 CQL(Continuous Query Language)에 기초해서 생성되는,
컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 13

컴퓨팅 장치의 하나 이상의 프로세서에서 실행가능한 분산처리를 위한 방법으로서,
프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하는 단계;
상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계;
상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계;
각각의 노드에서 이벤트를 수신하는 단계; 및
상기 분산처리 쿼리 및 상기 집중처리 쿼리 중 어느 하나에서 상기 이벤트의 처리 가부에 기초하여, 상기 분산처리 쿼리 및 상기 집중처리 쿼리 중 적어도 하나가 상기 이벤트를 처리하는 단계;
를 포함하는,

컴퓨팅 장치의 하나 이상의 프로세서에서 실행가능한 분산처리를 위한 방법.

청구항 14

컴퓨팅 장치로서,
 하나 이상의 프로세서; 및
 상기 하나 이상의 프로세서에서 실행가능한 명령들을 저장하는 메모리;
 를 포함하고,
 상기 하나 이상의 프로세서는,
 프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하고,
 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키고,
 상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키고, 그리고
 각각의 노드에서 이벤트를 수신하며,
 상기 분산처리 쿼리 및 상기 집중처리 쿼리 중 어느 하나에서 상기 이벤트의 처리 가부에 기초하여, 상기 분산
 처리 쿼리 및 상기 집중처리 쿼리 중 적어도 하나가 상기 이벤트를 처리하는,
 컴퓨팅 장치.

발명의 설명

기술 분야

[0001] 본 개시는 컴퓨팅 장치에 관한 것으로서, 보다 구체적으로는 분산처리를 위한 컴퓨팅 장치에 관한 것이다.

배경 기술

[0003] 복잡 이벤트 처리(CEP: Complex Event Processing)은 실시간으로 발생하는 이벤트 중 의미 있는 것만을 추출하
 여 대응되는 액션을 수행하는 프로세스이다. 데이터의 스트림 데이터 처리에서는 입력된 데이터를 하드디스크
 (HDD)에 기록하지 않고, 메모리에서 데이터 처리가 이루어지므로 빠른 속도로 처리할 수 있다. 또한, 바로 직전
 에 처리한 데이터를 중간 데이터로서 가지고 있어 데이터를 모두 처리할 필요가 없으며, 메모리에 흘러가는 데
 이터에서 중간 데이터와의 차이가 난 데이터양만큼만 처리하면 된다.

[0004] 이러한 복잡 이벤트 처리에서는 실시간 성이 중요하므로 in-memory 처리 방식을 사용한다. 복잡 이벤트 처리에
 서도 분산처리 시스템처럼, 스케일 아웃(scale-out), 스케일 업(scale-up) 방식으로 확장성(scalability)을 지
 원한다. 스케일 업 방식은 복잡 이벤트 처리를 수행하는 컴퓨팅 장치의 성능 자체를 증가시키는 방식이고, 스케
 일 아웃 방식은 복잡 이벤트 처리를 수행하는 컴퓨팅 장치의 수를 늘려 처리 능력을 향상시키는 것이다.

[0005] 스케일 아웃 방식으로 확장하는 경우, 확장에 대한 비용(cost)이 스케일 업 방식보다 저렴한 장점이 있고, 전면
 장애의 가능성이 적은 장점이 있다. 그러나, 스케일 아웃 방식으로 확장하는 경우, 데이터의 정합성을 유지하여
 야 한다.

[0006] 선행 특허 문헌: JP2011-034255(2011.02.17)

발명의 내용

해결하려는 과제

[0008] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로 분산처리를 제공하는 솔루션을 제공하기 위한 것이다.

과제의 해결 수단

- [0010] 기술한 바와 같은 과제를 실현하기 위한 본 개시의 일 실시예에 따라 인코딩된 명령들을 포함하는 컴퓨터 관독 가능 저장 매체에 저장된 컴퓨터 프로그램이 개시된다. 상기 컴퓨터 프로그램은 컴퓨터 시스템의 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 하나 이상의 프로세서들로 하여금 분산처리를 위한 이하의 단계들을 수행하도록 하며, 상기 단계들은: 프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하는 단계; 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계; 상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계; 및 각각의 노드에서 이벤트를 수신하는 단계를 포함한다.
- [0011] 대안적으로, 수신된 상기 이벤트를 상기 분산처리 쿼리가 처리할 수 있는 경우, 상기 분산처리 쿼리에서 상기 이벤트를 처리하는 단계를 더 포함할 수 있다.
- [0012] 대안적으로, 상기 이벤트를 상기 분산처리 쿼리가 처리할 수 없는 경우, 상기 이벤트를 집중처리 노드로 전달할 것을 결정하는 단계를 포함할 수 있다.
- [0013] 대안적으로, 상기 분산처리 쿼리는 각각의 이벤트에 대하여 연산을 처리하는 쿼리를 포함하고, 그리고 상기 집중처리 쿼리는 둘 이상의 이벤트를 종합하여 연산을 처리하는 쿼리를 포함할 수 있다.
- [0014] 대안적으로, 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계는, 상기 분산처리 쿼리를 상기 분산처리 노드 각각의 메모리에 등록시키는 단계를 포함하고, 그리고 상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계는, 상기 집중처리 쿼리를 상기 집중처리 노드의 메모리에 등록시키는 단계를 포함할 수 있다.
- [0015] 대안적으로, 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계는, 상기 분산처리 쿼리에 상기 분산처리 쿼리가 처리할 수 없는 이벤트가 수신되는 경우, 해당 이벤트를 집중처리 노드로 전송하도록 하는 명령을 추가하여, 상기 분산처리 쿼리를 상기 분산처리 노드에 등록시키는 단계를 포함할 수 있다.
- [0016] 대안적으로, 상기 이벤트를 상기 분산처리 쿼리가 처리할 수 없는 경우, 상기 이벤트를 분산처리 쿼리로 전송할 것을 결정하는 단계는, 상기 이벤트를 수신한 노드의 전용 어댑터(adapter)가 상기 집중처리 노드의 전용 어댑터에 상기 이벤트를 전송하는 단계를 포함할 수 있다.
- [0017] 대안적으로, 상기 이벤트를 수신한 노드의 전용 어댑터(adapter)가 상기 집중처리 노드의 전용 어댑터에 상기 이벤트를 전송하는 단계는, 상기 이벤트를 수신한 노드와 상기 집중처리 노드가 동일한 노드인 경우, 메모리 상에서 상기 분산처리 쿼리에서 상기 집중처리 쿼리로 상기 이벤트를 전송하고, 그리고 상기 이벤트를 수신한 노드와 상기 집중처리 노드가 상이한 노드인 경우, TCP 또는 HTTP 프로토콜을 이용하여 상기 집중처리 쿼리로 상기 이벤트를 전송하는 단계를 포함할 수 있다.
- [0018] 대안적으로, 상기 집중처리 노드는, 상기 둘 이상의 노드 중에 상기 분산처리 노드와 동일하거나 또는 별도의 노드이며 그리고 부하를 분산할 수 있도록 결정될 수 있다.
- [0019] 대안적으로, 상기 이벤트를 분산처리 쿼리가 처리한 경우, 상기 이벤트의 분산처리 쿼리 처리 결과를 상기 집중처리 노드에 전송하는 단계를 더 포함할 수 있다.
- [0020] 대안적으로, 집중처리 노드에서 상기 이벤트의 분산처리 쿼리 처리 결과를 수신하여 쿼리를 처리하는 단계를 더 포함할 수 있다.
- [0021] 대안적으로, 상기 쿼리는 CQL(Continuous Query Language)에 기초해서 생성될 수 있다.
- [0022] 본 개시의 다른 일 실시 예에 따라 컴퓨팅 장치의 하나 이상의 프로세서에서 수행되는 분산처리를 위한 방법이 개시된다. 상기 방법은, 프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하는 단계; 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키는 단계; 상기 집중처리 쿼리를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키는 단계; 및 각각의 노드에서 이벤트를 수신하는 단계를 포함한다.
- [0023] 본 개시의 또 다른 일 실시예에 따라 컴퓨팅 장치가 개시된다. 상기 컴퓨팅 장치는 하나 이상의 프로세서; 및 상기 하나 이상의 프로세서에서 실행가능한 명령들을 저장하는 메모리를 포함하고, 상기 하나 이상의 프로세서는, 프로세서에서 프로세스에 포함된 쿼리(query)를 분석하여 분산처리 쿼리와 집중처리 쿼리로 구분하고, 상기 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(node) 각각에 등록시키고, 상기 집중처리 쿼리

를 상기 둘 이상의 노드 중 집중처리 노드에 등록시키고, 그리고 각각의 노드에서 이벤트를 수신한다.

발명의 효과

[0025] 본 개시는 분산처리를 제공하는 솔루션을 제공할 수 있다.

도면의 간단한 설명

[0027] 도 1 은 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 복잡 이벤트 처리를 수행하는 컴포넌트들을 나타낸 예시도이다.

도 2 는 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 쿼리의 등록을 수행하는 컴포넌트들을 나타낸 예시도이다.

도 3 은 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 분산처리를 수행하는 컴포넌트들을 나타낸 예시도이다.

도 4 는 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 수행되는 분산처리 방법의 순서도이다.

도 5 는 본 개시의 일 실시예에 따른 컴퓨팅 장치의 블록 구성도(block diagram)이다.

발명을 실시하기 위한 구체적인 내용

[0028] 다양한 실시예들이 이제 도면을 참조하여 설명되며, 전체 도면에서 걸쳐 유사한 도면번호는 유사한 구성요소를 나타내기 위해서 사용된다. 본 명세서에서, 다양한 설명들이 본 개시의 이해를 제공하기 위해서 제시된다. 그러나 이러한 실시예들은 이러한 구체적인 설명 없이도 실행될 수 있음이 명백하다. 다른 예들에서, 공지된 구조 및 장치들은 실시예들의 설명을 용이하게 하기 위해서 블록 다이어그램 형태로 제공된다.

[0029] 본 명세서에서 사용되는 용어 "컴포넌트", "모듈", "시스템" 등은 컴퓨터-관련 엔티티, 하드웨어, 펌웨어, 소프트웨어, 소프트웨어 및 하드웨어의 조합, 또는 소프트웨어의 실행을 지칭한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 처리과정(procedure), 프로세서, 객체, 실행 스레드, 프로그램, 및/또는 컴퓨터일 수 있지만, 이들로 제한되는 것은 아니다. 예를 들어, 컴퓨팅 장치에서 실행되는 애플리케이션 및 컴퓨팅 장치 모두 컴포넌트일 수 있다. 하나 이상의 컴포넌트는 프로세서 및/또는 실행 스레드 내에 상주할 수 있고, 일 컴포넌트는 하나의 컴퓨터 내에 로컬화될 수 있고, 또는 2개 이상의 컴퓨터들 사이에 분배될 수 있다. 또한, 이러한 컴포넌트들은 그 내부에 저장된 다양한 데이터 구조들을 갖는 다양한 컴퓨터 판독가능한 매체로부터 실행할 수 있다. 컴포넌트들은 예를 들어 하나 이상의 데이터 패킷들을 갖는 신호(예를 들면, 로컬 시스템, 분산 시스템에서 다른 컴포넌트와 상호작용하는 하나의 컴포넌트로부터의 데이터 및/또는 신호를 통해 다른 시스템과 인터넷과 같은 네트워크를 통해 전송되는 데이터)에 따라 로컬 및/또는 원격 처리들을 통해 통신할 수 있다.

[0030] 제시된 실시예들에 대한 설명은 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의 범위에서 해석되어야 할 것이다.

[0032] 도 1 은 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 복잡 이벤트 처리를 수행하는 컴포넌트들을 나타낸 예시도이다.

[0033] 본 개시의 일 실시예에 따라 복잡 이벤트 처리를 수행하기 위한 컴퓨팅 장치(1)는 프로세서(10) 및 메모리(20)를 포함한다. 본 개시의 노드(1)는 서버, 컴퓨팅 장치 등을 포함할 수 있다.

[0034] 본 개시의 일 실시예에 따른 컴퓨팅 장치(1)는 데이터 스트림 관리 시스템(DSMS: Data Stream Management System)을 구성할 수 있으며, 데이터 스트림의 저장을 위하여 데이터베이스 관리 시스템(DBMS: Database Management System)와 연결될 수 있다. 데이터 스트림 관리 시스템(DSMS)은 일반적으로 대용량의 스트림 데이터에 대하여 연속질의(Continuous query)를 수행할 수 있는 시스템을 포함할 수 있다. 스트림 데이터는 연속적인 이벤트를 포함하는 이벤트의 집합으로서, 예를 들어, 데이터베이스의 테이블과 같은 데이터를 포함할 수 있다. 전술한 스트림 데이터의 예시는 본 개시를 제한하지 않는다.

[0035] 프로세서(10)는 컴퓨팅 장치의 전반적인 동작을 제어하며, 하나 이상의 코어, 하나 이상의 스레드를 포함할 수

있다. 프로세서(10)는 컴퓨팅 장치의 중앙 처리 장치(CPU)를 포함할 수 있다.

[0036] 메모리(20)는 프로세서에 의해 실행가능한 프로그램들, 프로세스들, 명령들을 저장할 수 있다. 쿼리는 메모리상의 프로세스 엔진(21)에 인스턴스로 로딩되어 있을 수 있으며, 이벤트(23)가 발생할 때 마다 쿼리가 수행될 수 있다. 본 개시의 일 실시예에 따른 쿼리는 CQL(Continuous Query Language)에 기초해서 생성될 수 있다. 본 개시의 일 실시예에 따른 쿼리는 시스템에 미리 등록되어 있으며, 쿼리가 시스템에 미리 등록되어 있으므로 데이터 스트림(이벤트)은 발생과 동시에 처리될 수 있다. 이벤트는 발생한 순서에 따라서 순차적으로 처리될 수 있다. 따라서, 데이터를 저장하거나 데이터를 불러 오는 시간이 절약될 수 있어 전체 시스템의 처리 시간이 비약적으로 향상되어 빅데이터의 처리를 용이하게 수행할 수 있다. 데이터 저장을 위해서는 별도의 쿼리가 사용될 수 있다. 이벤트는 영구 저장 장치로부터 로드되어 메모리(20)에 로드될 수 있고, 네트워크 연결을 통해 수신되어 메모리(20)에 수신될 수도 있다.

[0037] 본 개시의 일 실시예에 따라, 쿼리가 엔진에 등록되어 메모리상에 인스턴스로 로딩되어 있고, 새로운 이벤트가 발생할 때 마다 쿼리가 수행되므로, 시스템의 처리속도가 향상될 수 있다.

[0039] 도 2 는 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 쿼리의 등록을 수행하는 컴포넌트들을 나타낸 예시도이다.

[0040] 마스터 노드(30)의 프로세서는 프로세스에 포함된 하나 이상의 쿼리(40)를 분석하여, 분산처리 쿼리(41)(stateless query)와 집중처리 쿼리(43)(stateful query)로 구분할 수 있다. 여기서 분산처리 쿼리(41)는 각각의 이벤트에 대하여 연산을 처리하는 쿼리를 포함하며, 집중처리 쿼리(43)는 둘 이상의 이벤트를 종합하여 연산을 처리하는 쿼리를 포함할 수 있다. 예를 들어, 분산처리 쿼리(41)는 필터링 등 하나의 이벤트에 대하여 판단 가능한 쿼리를 포함할 수 있다. 집중처리 쿼리는 grouping, aggregation, join 등 전체 데이터 스트림을 획득해야만 연산 가능한 쿼리를 포함할 수 있다. 데이터 스트림은 각각의 이벤트들의 집합을 포함할 수 있으며, 데이터베이스의 테이블에 대응될 수 있다. 분산처리 쿼리는 복수의 노드 각각에서 독립적으로 수행되어도 전체 결과가 달라지지 않는 쿼리를 포함할 수 있으며, 집중처리 쿼리는 분산된 노드에서 수행되는 경우 결과값이 달라질 수 있는 쿼리를 포함할 수 있다. 전술한 쿼리는 예시일 뿐이며 본 개시는 이에 제한되지 않는다.

[0041] 예를 들어, 최근 일주일 내에 행사 제품을 3 회 이상 구매한 고객에게 할인 쿠폰을 문자로 전송하는 시나리오를 포함하는 프로세스가 존재할 수 있다. 상기 프로세스는, “모든 구매 내역에서 행사 제품 구매 내역 만 필터링하는 쿼리(제 1 서브 쿼리)”, “고객별로 최근 일주일 내의 구매 횟수를 계산하는 쿼리(제 2 서브 쿼리)” 및 “구매 횟수가 3 회 이상인 고객 정보만 필터링하는 쿼리(제 3 서브 쿼리)” 를 포함할 수 있다. 본 예시에서, 제 1 서브 쿼리는 각각의 이벤트에 해당하는 구매 내역 각각을 보아 행사제품인지 아닌지 여부를 판단할 수 있으므로, 분산처리 쿼리(41)에 포함될 수 있다. 제 2 서브 쿼리는 둘 이상의 이벤트에 해당하는 고객 별 구매 내역 전체의 이벤트 스트림을 수신한 후 그 횟수를 계산하여야 하므로 집중처리 쿼리(43)에 포함될 수 있다. 제 3 서브 쿼리는 각각의 이벤트에 해당하는 고객의 구매 횟수 값 만으로 연산할 수 있으므로 분산처리 쿼리(41)에 포함될 수 있다.

[0042] 구체적인 쿼리의 예시는 다음과 같다.

[0043] 이벤트는 userID, itemName, timestamp 필드로 구성되어 있다고 가정한다.

[0045] (SubQuery#1 모든 구매 내역에서 행사제품 구매 내역만 필터링)

[0046] insert into TargetItemStream

[0047] select *

[0048] from PurchasedStream

[0049] where itemName == "EVENT_PRODUCT";

[0051] (SubQuery#2 고객별로 최근 일주일 내의 구매 횟수 계산)

[0052] insert into CheckStream

[0053] select userID, count(*) as numPurchased

[0054] group by userID

- [0055] from TargetItemStream>window.time(7 days);
- [0057] (SubQuery#3 구매 횟수가 3회 이상인 고객 정보만 필터링)
- [0058] insert into CouponStream
- [0059] select userID
- [0060] from CheckStream
- [0061] where numPurchased >= 3;
- [0062] 전술한 예시에서, 제 1 서브쿼리는 구매 내역 각각의 이벤트에서 각각의 구매 내역이 행사제품인지, 아닌지를 판단하는 것이므로 분산처리 쿼리에 해당한다. 제 2 서브쿼리는 제 1 서브쿼리의 처리 결과를 고객별로 합산하여야 하므로, 각 고객의 복수의 구매 내역(복수의 이벤트)을 획득하여 쿼리를 처리하므로, 집중처리 쿼리에 해당한다. 제 3 서브쿼리는 제 2 서브쿼리의 처리 결과(예를 들어, 각각의 고객의 구매 회수 값)에서, 구매 횟수가 3 이상인 것을 필터링하므로, 각각의 이벤트에 해당하는 고객의 구매 횟수가 3 이상인지 여부를 판단하는 쿼리이므로 분산처리 쿼리에 해당할 수 있다.
- [0063] 전술한 시나리오 및 쿼리는 예시일 뿐이며 본 개시는 이에 제한되지 않는다.
- [0064] 마스터 노드(30)는 쿼리를 컴파일링 하여, 실행 계획(execution plan)을 생성하고, 이를 분석하여 프로세스에 포함된 쿼리를 분산처리 쿼리와 집중처리 쿼리로 구분한다.
- [0065] 마스터 노드(40)는 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드(51, 53, 55) 각각에 등록시킬 수 있다. 분산처리 쿼리는 이벤트 각각에 대하여 연산을 수행하므로, 어떤 노드가 수행하는지에 무관하게 결과가 동일할 수 있다. 따라서, 분산처리 쿼리는 각각의 노드에서 분산처리될 수 있다. 그러므로, 마스터 노드(40)는 분산처리 쿼리(41)를 분산처리 노드(51, 53, 55) 각각의 메모리(61, 63, 65)에 등록시킬 수 있다.
- [0066] 마스터 노드(40)는 집중처리 쿼리(43)를 집중처리 노드(53)에 등록시킬 수 있다. 집중처리 쿼리(43)는 분산처리할 경우 결과가 달라질 수 있는 쿼리를 포함하므로, 하나의 노드에서 처리하도록 하나의 집중처리 노드(53)에 등록시킬 수 있다. 여기서, 집중처리 노드(53)는 전체 노드들 중 하나로서, 분산처리 노드들 중 하나일 수도 있고, 분산처리 노드들과 구분된 별도의 노드일 수 있다. 도 2의 예시에서는, 집중처리 노드가 분산처리 노드 중 하나인 실시예를 도시한다. 이때, 마스터 노드(30)는 각각의 노드의 성능, 각각의 노드의 부하 등에 기초하여 부하가 분산되도록 집중처리 노드를 결정할 수 있다.
- [0067] 마스터 노드(40)는 분산처리 노드 각각에 분산처리 쿼리(41)를 등록할 때, 분산처리 쿼리가 처리할 수 없는 이벤트가 수신되는 경우, 해당 이벤트를 집중처리 노드(53)로 전송하도록 하는 명령을 분산처리 쿼리(41)에 추가하여 등록시킬 수 있다. 분산처리 쿼리와 집중처리 쿼리는 각각의 쿼리가 처리되기 위한 이벤트가 서로 상이할 수 있다. 전술한 예시에서, 분산처리 노드(51)에 고객별 행사 제품의 구매 내역들을 포함하는 이벤트가 수신된 경우, 고객별 행사제품 구매 내역을 필터링하는 제 1 서브쿼리는 쿼리를 수행할 수 없다. 이 경우, 분산처리 노드(51)는 집중처리 노드(53)의 집중처리 쿼리가 이벤트에 대한 쿼리를 수행할 수 있도록, 자신이 처리할 수 없는 이벤트들을 집중처리 노드(53)로 전송할 수 있다. 마스터 노드(30)는 분산처리 쿼리(41)를 등록할 때, 자신이 처리할 수 없는 이벤트를 집중처리 노드(53)으로 전송하도록 하는 명령을 포함하여 쿼리를 분산처리 노드 각각에 등록시킬 수 있다.
- [0068] 마스터 노드(30)는 각각의 노드에 등록된 분산처리 쿼리와 집중처리 노드에 등록된 집중처리 노드 사이에, 이벤트를 전달할 수 있도록 전용 어댑터로 연결시킬 수 있다.
- [0069] 본 개시의 일 실시예에 따라, 분산처리 쿼리는 분산처리 노드 각각에 등록하여 이벤트를 분산처리 하도록 하고, 분산처리 쿼리의 처리 결과와, 분산처리 쿼리가 처리할 수 없는 이벤트를 집중처리 노드로 전송시켜 집중처리 쿼리가 처리하도록 함으로써, scale-out 방식으로 분산처리를 수행하면서도 데이터의 정합성을 유지할 수 있다.
- [0071] 도 3은 본 개시의 일 실시예에 따라 컴퓨팅 장치 상에서 분산처리를 수행하는 컴포넌트들을 나타낸 예시도이다.
- [0072] 도 2를 참조하여 전술한 바와 같이, 복수의 노드들에 분산처리 쿼리 및 집중처리 쿼리가 등록될 수 있다. 본 개시의 일 실시예에 따라 분산처리 시스템은 복수의 노드들(100, 200, 300)을 포함할 수 있다. 도 3은 제 2 노드(200)가 분산처리 노드 및 집중처리 노드의 역할을 모두 수행하는 것으로 도시되어 있으나, 분산처리 노드와

집중처리 노드는 상이한 노드에 위치할 수도 있다.

- [0073] 분산처리 노드(100, 200, 300)에서 이벤트를 수신한 경우, 수신된 이벤트를 분산처리 쿼리(111, 211, 311)가 처리할 수 있는 경우, 분산처리 쿼리에서 이벤트를 처리할 수 있다. 예를 들어, 전술한 예시에서 각각의 분산처리 쿼리들이 고객의 구매 내역 중 행사제품의 구매 내역을 필터링 하는 쿼리인 경우, 각각의 구매 내역이 이벤트로서 노드들(100, 200, 300)에 수신되는 경우, 분산처리 쿼리들은 각각의 구매 내역에 대하여 해당 구매 내역이 행사 제품의 구매 내역인지 여부를 판단할 수 있다. 전술한 쿼리는 예시일 뿐이며 본 개시는 이에 제한되지 않는다. 각각의 노드의 분산처리 엔진들(110, 210, 310)에 인스턴스화된 분산처리 쿼리들(111, 211, 311)은 각각의 이벤트를 처리할 수 있고, 처리 결과를 집중처리 노드(200)의 집중처리 엔진(230)에 전송할 수 있다.
- [0074] 분산처리 노드(100, 200, 300)에서 이벤트를 수신한 경우, 수신된 이벤트를 분산처리 쿼리(111, 211, 311)가 처리할 수 없는 경우, 분산처리 쿼리(111, 211, 311)는 상기 이벤트를 집중처리 노드(200)로 전달할 것을 결정할 수 있다. 수신된 이벤트를 분산처리 쿼리(111, 211, 311)가 처리할 수 없는 경우, 분산처리 쿼리(111, 211, 311)는 상기 이벤트를 집중처리 쿼리(231)가 처리하도록, 이벤트를 집중처리 엔진(230)으로 전달할 수 있다.
- [0075] 분산처리 엔진의 어댑터(113, 213, 313)를 통해, 집중처리 엔진의 어댑터(233)로 이벤트가 전송될 수 있다. 분산처리 쿼리가 처리 결과를 집중처리 노드로 전송하는 경우와, 분산처리 쿼리가 처리할 수 없는 이벤트를 집중처리 노드로 전송하는 경우, 분산처리 엔진의 전용 어댑터를 통해 집중처리 엔진의 전용 어댑터로 분산처리 쿼리의 결과값, 이벤트가 전달될 수 있다. 이 경우, 분산처리 엔진의 전용 어댑터와 집중처리 엔진의 전용 어댑터가 동일한 노드에 위치하는 경우(예를 들어, 제 2 노드(200)는 집중처리 노드와 분산처리 노드를 겸함), 메모리 상에서 분산처리 쿼리에서 집중처리 쿼리로 이벤트가 전달될 수 있다. 분산처리 엔진의 전용 어댑터와 집중처리 엔진의 전용 어댑터가 상이한 노드에 위치하는 경우(예를 들어, 제 1 노드(100)에서 이벤트를 수신하였으나, 분산처리 쿼리(111)가 처리할 수 없는 경우, 제 2 노드(200)로 이벤트가 전달됨), TCP 또는 HTTP 등의 네트워크 프로토콜을 이용하여 이벤트가 전달될 수 있다.
- [0076] 집중처리 노드(200)는 이벤트의 분산처리 쿼리의 처리 결과를 수신하여, 집중처리 쿼리를 처리할 수 있다. 전술한 예시에서, 집중처리 노드(200)는, 제 1, 제 2 및 제 3 노드의 분산처리 엔진으로부터, 모든 구매 내역에서 행사제품 구매 내역만 필터링된 분산처리 쿼리의 처리 결과를 수신할 수 있다. 집중처리 노드(200)는 행사제품의 모든 구매 내역을 수신하여, 이를 고객별로 분류하여 각 고객별 일주일 내의 구매 횟수를 계산할 수 있다. 전술한 프로세스의 쿼리는 예시일 뿐이며 본 개시는 이에 제한되지 않는다. 집중처리 노드(200)에서 수행된 집중처리 쿼리의 처리 결과를 분산처리 쿼리가 분산처리 할 필요가 있는 경우, 상기 집중처리 쿼리의 처리 결과는 각각의 분산처리 노드로 분배될 수 있다. 예를 들어, 전술한 예시에서 집중처리 쿼리의 쿼리 결과 일주일간 각 고객의 구매 횟수가 생성되고, 이는 제 3 서버 쿼리에 의해 분산처리 될 수 있는 이벤트이므로, 상기 집중 처리 쿼리의 쿼리 결과는 각각의 분산처리 노드로 분배될 수 있다.
- [0077] 본 개시의 일 실시예에 따라, 각각의 노드에서 분산하여 처리되어도 단일 노드에서 처리 한 것과 결과가 같은 쿼리는 분산처리 쿼리로 분류하고, 단일 노드에서 처리 하여야 올바른 처리 결과가 획득되는 쿼리는 집중처리 쿼리로 분류하여, 분산처리 쿼리는 분산처리 할 수 있도록 분산처리 노드 각각에 등록하고, 집중처리 쿼리는 단일 노드에서 처리 할 수 있도록 집중처리 노드에 등록하고, 분산처리 노드가 분산처리 쿼리의 결과 및 자신이 처리할 수 없는 이벤트를 집중처리 노드에 전송하도록 함으로써, 구현 cost 가 상대적으로 낮은 scale-out 방식의 분산 처리 클러스터링을 구현하여 성능을 높이면서도 정합성이 유지될 수 있다.
- [0079] 도 4 는 본 개시의 일 실시예 따라 컴퓨팅 장치 상에서 수행되는 분산처리 방법의 순서도이다.
- [0080] 마스터 노드의 프로세서에서 프로세스에 포함된 쿼리를 분석하여 분산처리 쿼리와 집중 처리 쿼리로 구분할 수 있다(510). 마스터 노드는 프로세스를 분석하여 실행 계획을 수립하며, 이때, 프로세스에 포함된 복수의 쿼리들을 분산처리 쿼리와 집중처리 쿼리로 분류할 수 있다. 분산처리 쿼리는 각각의 이벤트에 대하여 쿼리를 수행할 수 있는 쿼리로서, 복수의 노드에서 분산하여 처리하여도 결과값이 달라지지 않는 쿼리를 포함할 수 있다. 집중처리 쿼리는 둘 이상의 이벤트를 종합하여 연산을 처리하는 쿼리로서, 복수의 노드에서 분산하여 처리하는 경우 결과값이 달라질 수 있는 쿼리를 포함할 수 있다. 본 개시의 일 실시예에 따른 쿼리는 CQL(continuous query language)에 기초하여 생성될 수 있으며, 메모리상에 프로세스 엔진에 등록되어 인스턴스로 로딩되어 있어, 이벤트 발생시 즉시 쿼리의 처리가 가능할 수 있다.
- [0081] 마스터 노드는 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드 각각에 등록시킬 수 있다(530). 마스터 노드는 분산처리 쿼리를 둘 이상의 노드 중 하나 이상의 분산처리 노드 각각의 메모리에 등록시킬 수 있다

다. 분산처리 노드는 시스템의 하나 이상의 노드 전부 또는 집중처리 노드를 제외한 전부를 포함할 수 있다.

- [0082] 마스터 노드는 집중처리 쿼리를 둘 이상의 노드들 중 집중처리 노드에 등록시킬 수 있다(550). 집중처리 노드는 분산처리 노드에 포함될 수도 있고, 분산처리 노드와 구분된 별도의 노드일 수 있다. 마스터 노드는 부하가 분산될 수 있도록 집중처리 노드를 결정할 수 있다.
- [0083] 각각의 노드에서는 이벤트를 수신할 수 있다(570). 본 개시의 일 실시예에 따라, 각각의 쿼리는 메모리상에 인스턴스로 로딩되어 있으므로, 발생하는 이벤트는 각각의 노드의 메모리에 획득될 수 있다. 각각의 분산처리 노드의 분산처리 쿼리는 수신된 이벤트를 자신이 처리할 수 있는 경우, 이를 처리하여 분산처리 쿼리의 처리 결과를 집중처리 노드로 전송할 수도 있고, 수신된 이벤트를 자신이 처리할 수 없는 경우, 수신된 이벤트를 집중처리 쿼리가 처리하도록 집중처리 쿼리로 전송할 수도 있다.
- [0085] 도 5 는 본 개시의 일 실시예에 따른 컴퓨팅 장치의 블록 구성도(block diagram)이다.
- [0086] 도 5 는 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.
- [0087] 본 개시가 일반적으로 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어와 관련하여 기술되었지만, 당업자라면 본 개시가 기타 프로그램 모듈들과 결합되어 및/또는 하드웨어와 소프트웨어의 조합으로서 구현될 수 있다는 것을 잘 알 것이다.
- [0088] 일반적으로, 프로그램 모듈은 특정의 태스크를 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조, 기타 등등을 포함한다. 또한, 당업자라면 본 개시의 방법이 단일-프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론 퍼스널 컴퓨터, 핸드헬드 컴퓨팅 장치, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 기타 등등(이들 각각은 하나 이상의 연관된 장치와 연결되어 동작할 수 있음)을 비롯한 다른 컴퓨터 시스템 구성으로 실시될 수 있다는 것을 잘 알 것이다.
- [0089] 본 개시의 설명된 실시예들은 또한 어떤 태스크들이 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘다에 위치할 수 있다.
- [0090] 컴퓨터는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 일시적(transitory) 및 비일시적(non-transitory) 매체, 이동식 및 비-이동식 매체를 포함한다. 제한이 아닌 예로서, 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체 및 컴퓨터 판독가능 전송 매체를 포함할 수 있다. 컴퓨터 판독가능 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성 매체, 일시적 및 비-일시적 매체, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함하지만, 이에 한정되지 않는다.
- [0091] 컴퓨터 판독가능 전송 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터등을 구현하고 모든 정보 전달 매체를 포함한다. 피변조 데이터 신호라는 용어는 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 제한이 아닌 예로서, 컴퓨터 판독가능 전송 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들 중 임의의 것의 조합도 역시 컴퓨터 판독가능 전송 매체의 범위 안에 포함되는 것으로 한다.
- [0092] 컴퓨터(1102)를 포함하는 본 개시의 여러가지 측면들을 구현하는 예시적인 환경(1100)이 나타내어져 있으며, 컴퓨터(1102)는 처리 장치(1104), 시스템 메모리(1106) 및 시스템 버스(1108)를 포함한다. 시스템 버스(1108)는 시스템 메모리(1106)(이에 한정되지 않음)를 비롯한 시스템 컴포넌트들을 처리 장치(1104)에 연결시킨다. 처리 장치(1104)는 다양한 상용 프로세서들 중 임의의 프로세서일 수 있다. 듀얼 프로세서 및 기타 멀티프로세서 아키텍처도 역시 처리 장치(1104)로서 이용될 수 있다.
- [0093] 시스템 버스(1108)는 메모리 버스, 주변장치 버스, 및 다양한 상용 버스 아키텍처 중 임의의 것을 사용하는 로

컬 버스에 추가적으로 상호 연결될 수 있는 몇가지 유형의 버스 구조 중 임의의 것일 수 있다. 시스템 메모리(1106)는 판독 전용 메모리(ROM)(1110) 및 랜덤 액세스 메모리(RAM)(1112)를 포함한다. 기본 입/출력 시스템(BIOS)은 ROM, EPROM, EEPROM 등의 비휘발성 메모리(1110)에 저장되며, 이 BIOS는 시동 중과 같은 때에 컴퓨터(1102) 내의 구성요소들 간에 정보를 전송하는 일을 돕는 기본적인 루틴을 포함한다. RAM(1112)은 또한 데이터를 캐싱하기 위한 정적 RAM 등의 고속 RAM을 포함할 수 있다.

[0094] 컴퓨터(1102)는 또한 내장형 하드 디스크 드라이브(HDD)(1114)(예를 들어, EIDE, SATA)—이 내장형 하드 디스크 드라이브(1114)는 또한 적당한 새시(도시 생략) 내에서 외장형 용도로 구성될 수 있음—, 자기 플로피 디스크 드라이브(FDD)(1116)(예를 들어, 이동식 디스켓(1118)으로부터 판독을 하거나 그에 기록을 하기 위한 것임), 및 광 디스크 드라이브(1120)(예를 들어, CD-ROM 디스크(1122)를 판독하거나 DVD 등의 기타 고용량 광 매체로부터 판독을 하거나 그에 기록을 하기 위한 것임)를 포함한다. 하드 디스크 드라이브(1114), 자기 디스크 드라이브(1116) 및 광 디스크 드라이브(1120)는 각각 하드 디스크 드라이브 인터페이스(1124), 자기 디스크 드라이브 인터페이스(1126) 및 광 드라이브 인터페이스(1128)에 의해 시스템 버스(1108)에 연결될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(1124)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 그 둘다를 포함한다.

[0095] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등의 비휘발성 저장을 제공한다. 컴퓨터(1102)의 경우, 드라이브 및 매체는 임의의 데이터를 적당한 디지털 형식으로 저장하는 것에 대응한다. 상기에서의 컴퓨터 판독가능 매체에 대한 설명이 HDD, 이동식 자기 디스크, 및 CD 또는 DVD 등의 이동식 광 매체를 언급하고 있지만, 당업자라면 zip 드라이브(zip drive), 자기 카세트, 플래쉬 메모리 카드, 카트리지, 기타 등등의 컴퓨터에 의해 판독가능한 다른 유형의 매체도 역시 예시적인 운영 환경에서 사용될 수 있으며 또 임의의 이러한 매체가 본 개시의 방법들을 수행하기 위한 컴퓨터 실행가능 명령어를 포함할 수 있다는 것을 잘 알 것이다.

[0096] 운영 체제(1130), 하나 이상의 애플리케이션 프로그램(1132), 기타 프로그램 모듈(1134) 및 프로그램 데이터(1136)를 비롯한 다수의 프로그램 모듈이 드라이브 및 RAM(1112)에 저장될 수 있다. 운영 체제, 애플리케이션, 모듈 및/또는 데이터의 전부 또는 그 일부가 또한 RAM(1112)에 캐싱될 수 있다. 본 개시가 여러가지 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있다는 것을 잘 알 것이다.

[0097] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(1138) 및 마우스(1140) 등의 포인팅 장치를 통해 컴퓨터(1102)에 명령 및 정보를 입력할 수 있다. 기타 입력 장치(도시 생략)로는 마이크, IR 리모콘, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린, 기타 등등이 있을 수 있다. 이들 및 기타 입력 장치가 종종 시스템 버스(1108)에 연결되어 있는 입력 장치 인터페이스(1142)를 통해 처리 장치(1104)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스, 기타 등등의 기타 인터페이스에 의해 연결될 수 있다.

[0098] 모니터(1144) 또는 다른 유형의 디스플레이 장치도 역시 비디오 어댑터(1146) 등의 인터페이스를 통해 시스템 버스(1108)에 연결된다. 모니터(1144)에 부가하여, 컴퓨터는 일반적으로 스피커, 프린터, 기타 등등의 기타 주변 출력 장치(도시 생략)를 포함한다.

[0099] 컴퓨터(1102)는 유선 및/또는 무선 통신을 통한 원격 컴퓨터(들)(1148) 등의 하나 이상의 원격 컴퓨터로의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1148)는 워크스테이션, 컴퓨팅 디바이스 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서-기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있으며, 일반적으로 컴퓨터(1102)에 대해 기술된 구성요소들 중 다수 또는 그 전부를 포함하지만, 간략함을 위해, 메모리 저장 장치(1150)만이 도시되어 있다. 도시되어 있는 논리적 연결은 근거리 통신망(LAN)(1152) 및/또는 더 큰 네트워크, 예를 들어, 원거리 통신망(WAN)(1154)에의 유선/무선 연결을 포함한다. 이러한 LAN 및 WAN 네트워킹 환경은 사무실 및 회사에서 일반적인 것이며, 인트라넷 등의 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 해주며, 이들 모두는 전세계 컴퓨터 네트워크, 예를 들어, 인터넷에 연결될 수 있다.

[0100] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(1156)를 통해 로컬 네트워크(1152)에 연결된다. 어댑터(1156)는 LAN(1152)에의 유선 또는 무선 통신을 용이하게 해줄 수 있으며, 이 LAN(1152)은 또한 무선 어댑터(1156)와 통신하기 위해 그에 설치되어 있는 무선 액세스 포인트를 포함하고 있다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 모뎀(1158)을 포함할 수 있거나, WAN(1154) 상의 통신 컴퓨팅 디바이스에 연결되거나, 또는 인터넷을 통하는 등, WAN(1154)을 통해 통신을 설정

하는 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(1158)은 직렬 포트 인터페이스(1142)를 통해 시스템 버스(1108)에 연결된다. 네트워크화된 환경에서, 컴퓨터(1102)에 대해 설명된 프로그램 모듈들 또는 그의 일부분이 원격 메모리/저장 장치(1150)에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다.

[0101] 컴퓨터(1102)는 무선 통신으로 배치되어 동작하는 임의의 무선 장치 또는 개체, 예를 들어, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, PDA(portable data assistant), 통신 위성, 무선 검출가능 태그와 연관된 임의의 장비 또는 장소, 및 전화와 통신을 하는 동작을 한다. 이것은 적어도 Wi-Fi 및 블루투스 무선 기술을 포함한다. 따라서, 통신은 종래의 네트워크에서와 같이 미리 정의된 구조이거나 단순히 적어도 2개의 장치 사이의 애드혹 통신(ad hoc communication)일 수 있다.

[0102] Wi-Fi(Wireless Fidelity)는 유선 없이도 인터넷 등으로의 연결을 가능하게 해준다. Wi-Fi는 이러한 장치, 예를 들어, 컴퓨터가 실내에서 및 실외에서, 즉 기지국의 통화권 내의 아무 곳에서나 데이터를 전송 및 수신할 수 있게 해주는 셀 전화와 같은 무선 기술이다. Wi-Fi 네트워크는 안전하고 신뢰성있으며 고속인 무선 연결을 제공하기 위해 IEEE 802.11(a,b,g, 기타)이라고 하는 무선 기술을 사용한다. 컴퓨터를 서로에, 인터넷에 및 유선 네트워크(IEEE 802.3 또는 이더넷을 사용함)에 연결시키기 위해 Wi-Fi가 사용될 수 있다. Wi-Fi 네트워크는 비인가 2.4 및 5 GHz 무선 대역에서, 예를 들어, 11Mbps(802.11a) 또는 54 Mbps(802.11b) 데이터 레이트로 동작하거나, 양 대역(듀얼 대역)을 포함하는 제품에서 동작할 수 있다.

[0104] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 정보 및 신호들이 임의의 다양한 상이한 기술들 및 기법들을 이용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 위의 설명에서 참조될 수 있는 데이터, 지시들, 명령들, 정보, 신호들, 비트들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학장들 또는 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.

[0105] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 프로세서들, 수단들, 회로들 및 알고리즘 단계들이 전자 하드웨어, (편의를 위해, 여기에서 "소프트웨어"로 지칭되는) 다양한 형태들의 프로그램 또는 설계 코드 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 개시의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 애플리케이션에 대하여 다양한 방식으로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 개시의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.

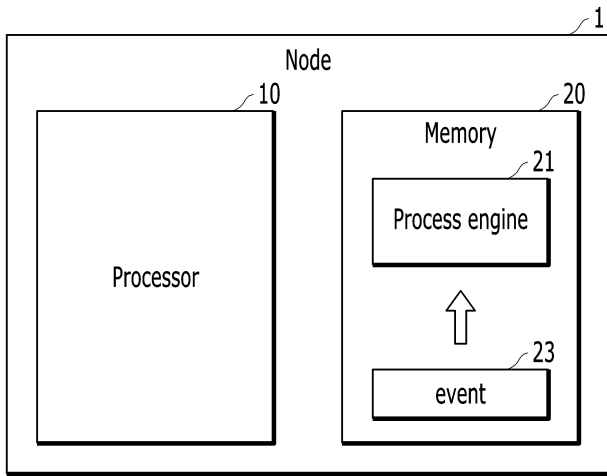
[0106] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조물품(article)으로 구현될 수 있다. 용어 "제조 물품"은 임의의 컴퓨터-관독가능 장치로부터 액세스 가능한 컴퓨터 프로그램, 캐리어, 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-관독가능 매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트립, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래쉬 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 키 드라이브, 등)를 포함하지만, 이들로 제한되는 것은 아니다. 또한, 여기서 제시되는 다양한 저장 매체는 정보를 저장하기 위한 하나 이상의 장치 및/또는 다른 기계-관독가능한 매체를 포함한다.

[0107] 제시된 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조는 예시적인 접근들의 일례임을 이해하도록 한다. 설계 우선순위들에 기반하여, 본 개시의 범위 내에서 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조가 재배열될 수 있다는 것을 이해하도록 한다. 첨부된 방법 청구항들은 샘플 순서로 다양한 단계들의 엘리먼트들을 제공하지만 제시된 특정한 순서 또는 계층 구조에 한정되는 것을 의미하지는 않는다.

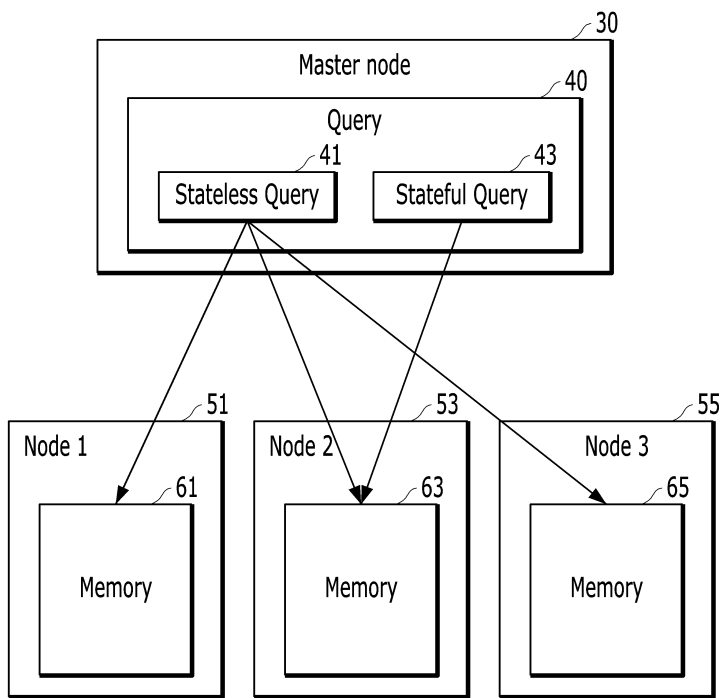
[0108] 제시된 실시예들에 대한 설명은 임의의 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면

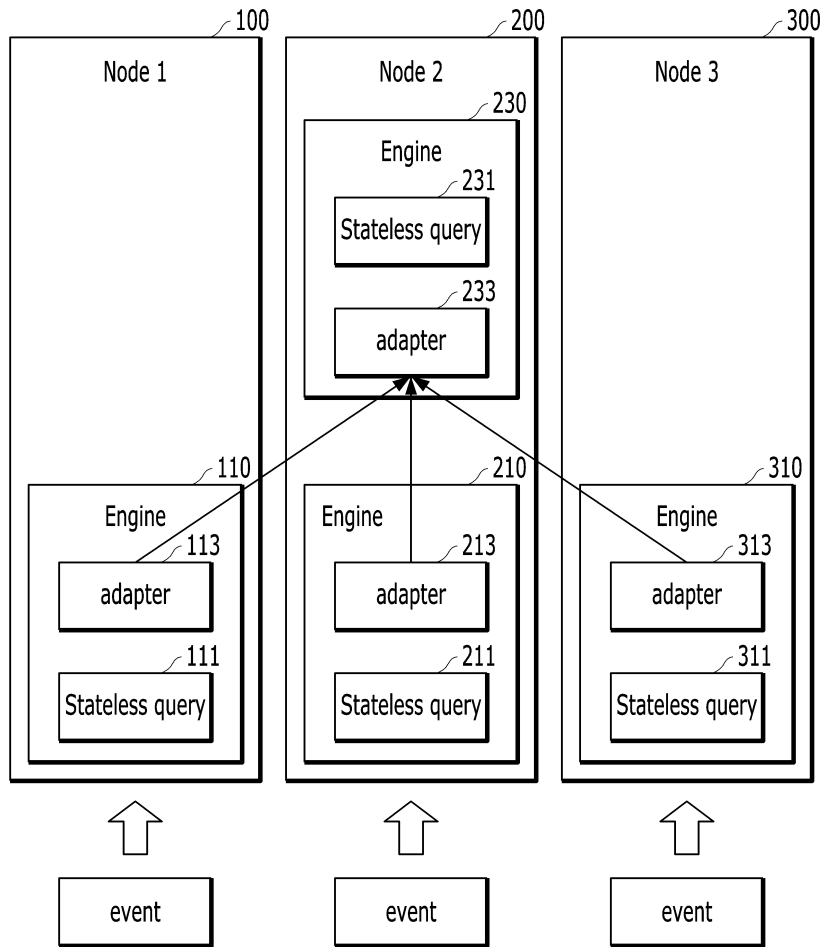
도면1



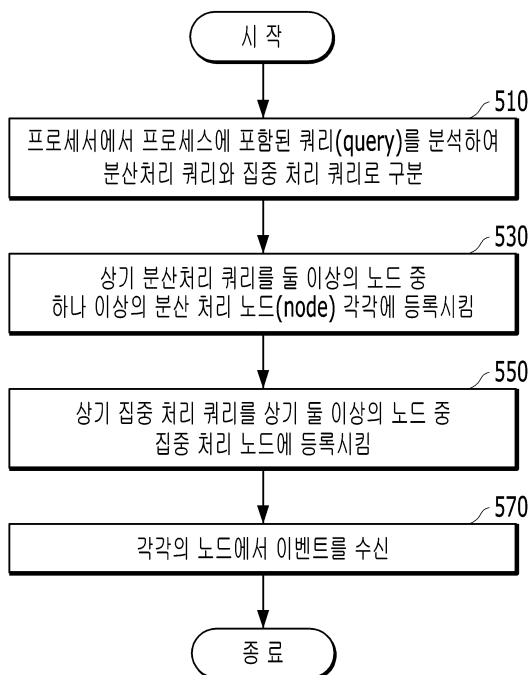
도면2



도면3



도면4



도면5

