



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0042876
(43) 공개일자 2015년04월21일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01)</p> <p>(52) CPC특허분류
G06F 17/30445 (2013.01)</p> <p>(21) 출원번호 10-2015-7008410(분할)</p> <p>(22) 출원일자(국제) 2010년12월23일
심사청구일자 없음</p> <p>(62) 원출원 특허 10-2012-7016852
원출원일자(국제) 2010년12월23일
심사청구일자 2015년03월02일</p> <p>(85) 번역문제출일자 2015년04월01일</p> <p>(86) 국제출원번호 PCT/US2010/061979</p> <p>(87) 국제공개번호 WO 2011/079251
국제공개일자 2011년06월30일</p> <p>(30) 우선권주장
61/289,778 2009년12월23일 미국(US)</p> | <p>(71) 출원인
아브 이니티오 테크놀로지 엘엘시
미국 02421 매사추세츠주 렉싱턴 스프링 스트리트 201</p> <p>(72) 발명자
스탠필 크레이그 더블유.
미국 01773 매사추세츠주 린콜린 허클베리 힐 로드 43
맥린 존
영국 스코틀랜드 피에이5 9비제이 엘더스리 엘더 그로브 뉴랜드크랙스 에비뉴 34</p> <p>(74) 대리인
유미특허법인</p> |
|---|--|

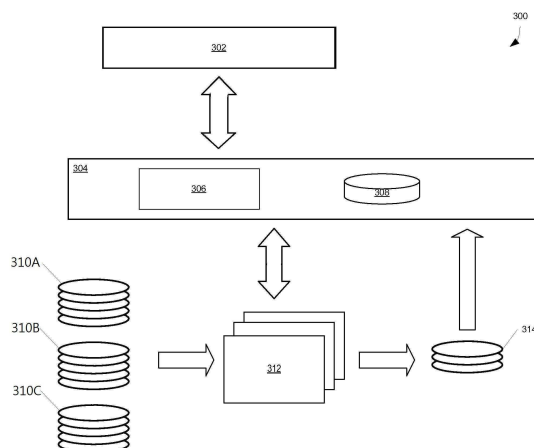
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **쿼리 관리**

(57) 요약

하나 이상의 데이터 소스 상에서 수행되는 쿼리 관리는 저장 매체(308)에 적어도 제1 쿼리를 저장하는 단계, 처리할 상기 제1 쿼리를 선택하는 단계, 쿼리 엔진(312)에, 제1 쿼리 구간 동안, 상기 하나 이상의 데이터 소스의 제1 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계, 상기 제1 데이터 부분 상에서 상기 제1 쿼리를 처리한 것을 기초로 상기 쿼리 엔진으로부터 결과 데이터(314)를 수신하는 단계, 상기 제1 쿼리 구간 이후, 상기 저장 매체에 상기 제1 쿼리의 상태를 저장하는 단계, 상기 쿼리 엔진에, 상기 제1 쿼리 구간 이후 제2 쿼리 구간 동안 제1 쿼리를 처리할 것을 지시하는 단계, 및 상기 쿼리 엔진에, 상기 제2 쿼리 구간 이후 제3 쿼리 구간 동안, 상기하나 이상의 데이터 소스의 제2 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계를 포함한다.

대표도 - 도3



명세서

청구범위

청구항 1

발명의 상세한 설명에 기재된, 또는 도면에 도시된 바와 같은 장치.

발명의 설명

기술분야

[0001] 본 출원은 2009년 12월 23일 출원된 미국 가출원 제61/289,778호에 대해 우선권을 주장하며, 참조로 본 명세서에 포함된다.

[0002] 본 발명은 쿼리(query) 관리에 관한 것이다.

배경기술

[0003] 일부 데이터 스토리지(storage) 시스템(예를 들어, 데이터 베이스)은 매우 많은 쿼리(query) 처리를 지원하는데 탁월한 방법으로 저장된 다량의 데이터를 저장한다. 예를 들어, 일부 시스템은 병렬 저장 장치, 병렬 쿼리 처리 엔진, 또는 양자 모두의 사용을 통해 병렬 처리 능력을 가진다.

발명의 내용

과제의 해결 수단

[0004] 본 발명의 일 실시예는 일반적으로, 하나 이상의 데이터 소스 상에서 수행되는 쿼리 관리 방법으로, 상기 방법은 저장 매체에 적어도 제1 쿼리를 저장하는 단계, 처리할 상기 제1 쿼리를 선택하는 단계, 쿼리 엔진에, 제1 쿼리 구간 동안, 상기 하나 이상의 데이터 소스의 제1 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계, 상기 제1 데이터 부분 상에서 상기 제1 쿼리를 처리한 것을 기초로 상기 쿼리 엔진으로부터 결과 데이터를 수신하는 단계, 상기 제1 쿼리 구간 이후, 상기 저장 매체에 상기 제1 쿼리의 상태를 저장하는 단계, 상기 쿼리 엔진에, 상기 제1 쿼리 구간 이후 제2 쿼리 구간 동안 제1 쿼리를 처리할 것을 지시하는 단계 및 상기 쿼리 엔진에, 상기 제2 쿼리 구간 이후 제3 쿼리 구간 동안, 상기하나 이상의 데이터 소스의 제2 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계를 포함한다.

[0005] 본 발명의 일 실시예는 하나 이상의 다음과 같은 특징을 포함한다.

[0006] 상기 방법은 상기 저장 매체에 상기 제1 쿼리에 연관된 우선순위를 저장하는 단계 및 처리할 상기 제1 쿼리를 선택하기에 앞서, 상기 제1 쿼리에 연관된 우선순위를 변경하는 단계를 포함하고, 처리할 상기 제1 쿼리를 선택하는 단계는 부분적으로 상기 우선순위를 기초로 하여 쿼리를 선택한다.

[0007] 제1 쿼리 구간은 기설정된 시간의 양으로 정의된다.

[0008] 제1 쿼리의 우선순위는, 하나 이상의 데이터 소스의 얼마나 많은 데이터가, 상기 제1 쿼리 구간 동안 상기 제1 쿼리가 실행되는 상기 제1 데이터 부분에 포함되는지에 영향을 준다.

[0009] 제1 쿼리를 저장하는 단계는, 상기 제1 쿼리를 제공한 요청자에게 통지되기 전에 이용 가능한 결과 데이터 양의 통지 문턱값을 저장하는 단계를 포함한다.

[0010] 상기 방법은 상기 결과 데이터 양이 상기 통지 문턱값을 초과하는 경우, 상기 요청자에게 통지하는 단계를 더 포함하며, 상기 제1 쿼리의 상태를 저장하는 단계는 상기 쿼리 엔진으로부터 수신한 상기 결과 데이터 양을 저장하는 단계를 포함한다.

[0011] 상기 방법은 상기 요청자로부터의 요청에 따라 결과 데이터를 반환하는 단계 및 상기 요청자에 반환되는 결과 데이터 양을 상기 저장 매체에 저장하는 단계를 더 포함한다.

[0012] 상기 쿼리의 선택은 상기 쿼리 엔진으로부터 수신된 결과 데이터의 양 및 상기 요청자에 반환되는 결과 데이터

의 양에 기초한다.

- [0013] 제1 쿼리의 상태를 저장하는 단계는, 상기 쿼리 엔진에 상기 제1 쿼리를 보유할 것을 지시하는 단계 및 상기 제1 쿼리가 보유된 후, 상기 제1 쿼리의 상태를 저장하는 단계를 포함한다.
- [0014] 상기 제1 쿼리에, 상기 제2 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계는, 저장된 상기 제1 쿼리의 상태를 로딩하는 단계 및 상기 쿼리 엔진에, 상기 제1 쿼리를 재개할 것을 지시하는 단계를 포함한다.
- [0015] 상기 제1 쿼리의 상태를 저장하는 단계는 오프셋(offset)을 2차 색인에 저장하는 단계를 포함한다.
- [0016] 2차 색인은 블록 압축된 색인 파일이다.
- [0017] 상기 방법은 제1 쿼리를 다수의 서브 쿼리로 분할하는 단계 및 쿼리 엔진에, 상기 서브 쿼리의 적어도 일부를 동시에 처리할 것을 지시하는 단계를 포함한다.
- [0018] 제2 쿼리는, 상기 제1 쿼리 구간이 시작된 후 수신되어 저장 매체에 저장된다.
- [0019] 상기 제2 쿼리는, 상기 제1 쿼리 구간이 시작되기 전 수신되어 저장 매체에 저장된다.
- [0020] 본 발명의 다른 실시예는, 일반적으로, 하나 이상의 데이터 소스 상에서 수행되는 쿼리 관리를 위한 컴퓨터 프로그램을 저장하는 컴퓨터 판독가능한 매체이다. 상기 컴퓨터 프로그램은 상기 컴퓨터가, 저장 매체에 적어도 제1 쿼리를 저장하고, 처리할 상기 제1 쿼리를 선택하고, 쿼리 엔진에, 제1 쿼리 구간 동안, 상기 하나 이상의 데이터 소스의 제1 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하고, 상기 제1 데이터 부분 상에서 상기 제1 쿼리를 처리한 것을 기초로 상기 쿼리 엔진으로부터 결과 데이터를 수신하고, 상기 제1 쿼리 구간 이후, 상기 저장 매체에 상기 제1 쿼리의 상태를 저장하고, 상기 쿼리 엔진에, 상기 제1 쿼리 구간 이후 제2 쿼리 구간 동안 제1 쿼리를 처리할 것을 지시하며, 상기 쿼리 엔진에, 상기 제2 쿼리 구간 이후 제3 쿼리 구간 동안, 상기하나 이상의 데이터 소스의 제2 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하도록 한다.
- [0021] 본 발명의 다른 실시예는, 일반적으로, 하나 이상의 데이터 소스 상에서 수행되는 쿼리 관리 시스템이다. 상기 시스템은, 적어도 제1 쿼리를 저장하는 저장 매체를 포함한다. 상기 시스템은 상기 하나 이상의 데이터 소스 상에서 쿼리를 처리하도록 구성된 쿼리 엔진을 포함한다. 또한, 상기 시스템은, 처리할 상기 제1 쿼리를 선택하고, 쿼리 엔진에, 제1 쿼리 구간 동안, 상기 하나 이상의 데이터 소스의 제1 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하고, 상기 제1 데이터 부분 상에서 상기 제1 쿼리를 처리한 것을 기초로 상기 쿼리 엔진으로부터 결과 데이터를 수신하고, 상기 제1 쿼리 구간 이후, 상기 저장 매체에 상기 제1 쿼리의 상태를 저장하고, 상기 쿼리 엔진에, 상기 제1 쿼리 구간 이후 제2 쿼리 구간 동안 제1 쿼리를 처리할 것을 지시하며, 상기 쿼리 엔진에, 상기 제2 쿼리 구간 이후 제3 쿼리 구간 동안, 상기하나 이상의 데이터 소스의 제2 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하도록 구성되는, 서버를 포함한다.
- [0022] 본 발명의 다른 실시예는, 일반적으로 하나 이상의 데이터 소스 상에서 수행되는 쿼리 관리 시스템이다. 상기 시스템은 적어도 제1 쿼리를 저장하는 저장 매체를 포함한다. 상기 시스템은 상기 하나 이상의 데이터 소스 상에서 쿼리를 처리하도록 구성된 쿼리 엔진을 포함한다. 상기 시스템은 저장 매체에서 쿼리를 관리하는 관리 수단을 포함하며, 상기 쿼리를 관리하는 것은 쿼리 엔진에, 제1 쿼리 구간 동안, 상기 하나 이상의 데이터 소스의 제1 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계, 상기 제1 데이터 부분 상에서 상기 제1 쿼리를 처리한 것을 기초로 상기 쿼리 엔진으로부터 결과 데이터를 수신하는 단계, 상기 제1 쿼리 구간 이후, 상기 저장 매체에 상기 제1 쿼리의 상태를 저장하는 단계, 상기 쿼리 엔진에, 상기 제1 쿼리 구간 이후 제2 쿼리 구간 동안 제1 쿼리를 처리할 것을 지시하는 단계 및 상기 쿼리 엔진에, 상기 제2 쿼리 구간 이후 제3 쿼리 구간 동안, 상기하나 이상의 데이터 소스의 제2 데이터 부분 상에서 상기 제1 쿼리를 처리할 것을 지시하는 단계를 포함한다.

발명의 효과

- [0023] 본 발명은 다음과 같은 하나 이상의 장점을 포함할 수 있다.
- [0024] 쿼리에 연관된 우선순위를 부분적으로 기초로 하여 쿼리를 선택하는 것은 병렬 쿼리 처리 시스템에서 처리 효율을 향상시킬 수 있도록 한다. 구간을, 쿼리의 부분이 부분적으로 처리되고 보유될 수 있는 시간으로 조각화하는 것은 시스템 내에서, 특히 높은 우선 순위의 쿼리에 있어서, 쿼리가 더 빨리 처리되도록 하고, 일부 잠재적인 백로그를 줄인다.

[0025] 본 발명의 다른 특징 및 장점은 이하의 설명 및 청구항으로부터 도출될 수 있다.

도면의 간단한 설명

[0026] 도 1 및 도 2는 쿼리(query) 처리를 표현하는 개략도이다.

도 3은 데이터 스토리지(storage)시스템의 블록도이다.

도 4는 압축된 색인 데이터 스토리지의 개략도이다.

도 5A, 5B, 6A, 6B 및 7A 내지 7D는 쿼리 처리에 연관된 시간 간격을 나타내는 플롯이다.

도 8은 쿼리 처리의 조각화 개략도이다.

도 9는 압축된 색인 데이터 스토리지의 쿼리 처리를 나타내는 개략도이다.

도 10 및 도 11은 쿼리 관리 프로세스의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0027] **1. 개요**

[0028] 도 1을 참조하면, 분산된 쿼리(query) 관리에서 몇 가지 문제점이 발생할 수 있다. 예를 들어, 쿼리가 선입선출법으로 데이터 스토리지 시스템의 쿼리 엔진에 전달되면, 시스템은 백로그(backlogged) 될 수 있다. 일부 경우에 있어서, 전달된 쿼리는 리소스(resource)를 적게 필요로 하면서 빠르게 실행되는 쇼트 쿼리(short query, 102, 104, 108, 112, 118), 실행되는데 긴 시간을 필요로 하며 다량의 시스템 자원을 활용하는 롱 쿼리(long query, 110, 114, 116), 및 상기 쇼트 쿼리 및 롱 쿼리 사이의 어딘가에 배치되는 쿼리를 포함할 수 있다. 쿼리가 실행되기 전에 특정 쿼리가 요구하는, 시스템 리소스의 양을 미리 설정하는 것은 실용적이지 않을 수 있다. 도 1은 다수의 쿼리 엔진을 사용하여 쿼리를 처리하는 시스템의 일 실시예를 나타낸다. 쿼리는, 데이터 스토리지 시스템의 쿼리 서버(100) 상에서 실행하는 쿼리 엔진에 의해 처리될 기회를 기다리는 대기열(queue, 101)에 비동기적으로(asynchronously) 수신되고 저장된다. 이 실시예에서는, 먼저 롱 쿼리 116이 처리되기 위해 제1 쿼리 엔진(120)에 할당되고, 잠시 후 쇼트 쿼리 118이 처리되기 위해 제2 쿼리 엔진(122)에 할당된다. 도 2를 참조하면, 잠시 후 쇼트 쿼리 118이 완료되고 라인에 있던 다음 쿼리인 롱 쿼리 114가 비어있는 쿼리 엔진(122)에 할당된다. 이 시점에서, 남아있는 쿼리 102, 104, 108, 110, 112는, 롱 쿼리 116 및 114 중 하나가 처리를 완료하고 쿼리 엔진의 처리 리소스를 해방시킬 때까지 기다린다. 이러한 현상은 쇼트 쿼리의 지연 시간을 증가시키고, 빠른 응답이 기대되는 쿼리에서는 수용할 수 없는 지연을 유발할 수 있다.

[0029] 도 3을 참조하면, 데이터 스토리지 시스템(300)은 예를 들어, 웹 서비스와 같은 프론트-엔드 서비스(front-end service, 302)를 제공하고, 쿼리를 실행하기 위한 요청을 수신한다. 중개 서버(304)는 다수의 쿼리 엔진들(312)에 의한 쿼리 실행 일정을 잡는다. 각각의 쿼리는 할당된 기간(period) 동안 실행되는 것이 허용되며, 상기 기간은 예를 들어, 시간(예를 들어, CPU 클럭(clock)에 의해 측정되는), 지속기간(duration), 처리되는 열(row)의 수, 또는 검색되는 열의 수로 측정될 수 있다. 쿼리 엔진(312)은 하나 이상의 데이터 소스(310A, 310B, 310C)로부터 데이터에 액세스하여, 쿼리를 처리하여 결과 세트(314)를 생성한다. 데이터 소스를 제공하는 스토리지 장치는 예를 들어, 중개 서버(304, 예를 들어 하드 드라이브)를 구현하는 컴퓨터에 연결된 스토리지 매체 상에 저장되는 시스템(300)과 근거리로 배치될 수 있으며, 또는 원격 연결로 통신하는 원격 시스템(예를 들어, 메인프레임) 상에서 호스트 역할을 하는 중개 서버(304)와 원격리로 배치될 수 있다.

[0030] 중개 서버(304)는 결과 세트(314)를 관리한다. 중개 서버(304)는 예를 들어, 쿼리의 우선순위, 요청된 열의 수, 쿼리로부터 반환되는 열의 수, 요청자(requerster)로 반환되는 열의 수, 쿼리가 사용되는 방법에 대한 지시, 한번에 요청되는 열의 수, 쿼리 엔진에 의해 쿼리가 최근 실행된 시간, 및 상태 지시자 등 쿼리에 관한 추가적인 정보를 저장할 수 있다. 상태 지시자는 쿼리가 대기중이거나, 실행중이거나, 보류되거나, 중지되거나, 또는 완료되는지를 나타낼 수 있다. 일부 구성에 있어서, 쿼리 상태는 또한 쿼리 처리 과정에서 발생한 오류 조건의 존재 여부를 나타낼 수 있다. 대기 상태의 쿼리는 실행될 자격이 있는 것으로, 현재 실행 중인 것은 아니다. 실행 상태에 있는 쿼리는 현재 쿼리 엔진에 의해 처리되고 있는 것이다. 보류 상태의 쿼리는, 중개 서버가 클라이언트에 의해 현재 요청되는 열의 수를 이미 반환하였으므로, 실행될 자격이 없다. 중지 상태의 쿼리는, 높은 우선순위의 쿼리가 선점하였으므로, 실행될 자격이 없다. 완료 상태의 쿼리는 실행이 완료된 것이다. 일부 구성에서, 다른 추가적인 상태가 지원될 수 있다.

- [0031] 중개 서버(304)는 또한 요청자가, 쿼리 결과가 처리될 준비가 되었음을 언제, 어떻게 통지해야하는지를 식별하는 정보를 저장할 수 있다. 일부 구성에 있어서, 다수의 쿼리 엔진은 단일 쿼리의 상이한 부분 상에서 독립적으로 작동할 수 있다. 각각의 쿼리 엔진은 중개 데이터베이스를 독립적으로 업데이트한다. 통지 이벤트는 예를 들어, 결합된 쿼리 엔진이 요청된 열의 수를 반환할 때와 같이, 트리거링 이벤트가 한번 발생하면 트리거 될 수 있다.
- [0032] 일부 구현들에서, 중개 서버는 중개 논리 모듈(306) 및 중개 데이터 베이스(308)를 포함한다. 일부 구현들에서, 중개 논리 모듈(306)은 개별적인 쿼리 엔진(312), 프론트-엔드 서비스(302)에 포함될 수 있으며, 다수의 서비스들 사이에서 분할 될 수 있다. 쿼리 엔진(312)은, 결과 세트(314) 내의 가능한 열의 수에 관해 중개 데이터베이스(308)를 업데이트 할 수 있다.
- [0033] 일부 구현들에서, 데이터 소스(310)는, 도 4를 참조하면, 압축된 색인 데이터 스토리지(402)를 포함한다. 압축된 색인 데이터 스토리지는 예를 들어, 파일에 저장된 다수의 압축된 데이터 블록들(404)을 포함한다. 각각의 압축된 데이터블록은 적어도 하나의 색인(406)에 연관되며, 상기 색인은 압축된 데이터 블록 내의 데이터의 위치를 활성화(enable)한다. 일부 구현들에서, 원시 색인은 제1 키(key, 예를 들어, 원시 키)를 기초로 검색되도록 제공되며, 하나 이상의 제2 색인은 다른 키들(예를 들어, 외부 키)을 기초로 검색되도록 제공된다. 일부 색인은 각각의 키 값이 고유한, 대체 키(surrogate key)로 구성될 수 있으며, 다른 색인은 데이터 세트 내에서 각각의 키 값이 고유하지 않은, 자연 키(natural key)를 기초로 할 수 있다. 일부 구현들에서, 자연 색인은 결합되어 단일 통합형 색인을 생성할 수 있다. 압축된 색인 데이터 스토리지 기술 및 시스템은 본 명세서에 참조로서 병합된 미국 특허 출원 2008/0104149호에서 더 상세하게 설명된다.
- [0034] **2. 쿼리 조각화(Query Slicing)**
- [0035] 도 5A를 참조하면, 일련의 쿼리들(A 502, B 504, C 506, 및 D 508)이, 상이한 쿼리에 연관된 구간을 도시한 플롯으로 도시된다. 만약 쿼리들이 전달되는 순서에 따라 실행된다면, 쿼리 A는 구간 502동안 실행되어 완료되고, 이후 쿼리 B는 구간 504동안 실행되어 완료되며, 뒤이어 쿼리 C가 구간 506동안, 쿼리 D가 구간 508 동안 실행된다. 이러한 조건하에서, 쿼리 A는, 시간 510에서 완료할 때까지 결과를 반환하지 않고, 쿼리 B는 시간 512에서 완료할 때까지 결과를 반환하지 않고, 쿼리 C는 시간 514에서 완료할 때까지 결과를 반환하지 않으며, 쿼리 D는 시간 516에서 완료할 때까지 결과를 반환하지 않을 것이다. 쿼리 D가 쇼트 쿼리 일지라도, 다른 롱 쿼리들 다음으로 배치되므로, 결과를 반환하는데 오랜 시간이 걸린다.
- [0036] 중개 서버(304)의 일부 구현들에서, 필연적으로 쿼리들은 순차적으로 실행하여 완료하는 대신, 중개 서버는 쿼리를 상이하고 작은 다수의 부분으로 나눈다. 쿼리 엔진(304)은 특정 구간 동안 쿼리를 실행하도록 지시를 받는다. 이러한 구간은 시간 간격, 반환되어야 하는 열의 수, 처리되는 열의 수, 또는 일부 다른 기준을 기초로 정의될 수 있다. 이러한 접근을 이용하여, 도 5B를 참조하면, 쿼리 A는 구간 528 동안 실행하고, 쿼리 B는 구간 530 동안 실행하고, 쿼리 C는 구간 532 동안 실행하고, 쿼리 D는 구간 534 동안 실행하며(완료되고), 이후 쿼리 A는 다시 제2 구간 동안 실행한다. 일부 경우에 있어서, 쿼리의 결과 일부는, 쿼리가 처리되는 각각의 구간 이후 쿼리를 제출하는 프로세스로 반환될 수 있다. 예를 들어, 쿼리 A의 결과 일부는 시간 520 이후 반환될 수 있으며, 쿼리 B, C 및 D의 결과 일부는 각각, 시간 522, 524 및 526 이후 반환될 수 있다. 쿼리를 작은 실행 구간으로 나눔으로써, 시스템(300)은, 시스템이 다른 쿼리를 실행하기 전에 쿼리가 완료되기를 기다려야하는 하는 경우보다 빠르게 더 많은 쿼리에 대한 결과를 생성할 수 있다. 추가적으로, 일부 쿼리는, 다른 쿼리와 지연을 교환함으로써, 이와 달리 쿼리를 완료시키는 것보다 빠르게 완료될 수 있다. 이러한 실시예에서, 쿼리 D는 시간 526에 완료되고, 쿼리 C는 시간 540에 완료되며, 쿼리 A는 시간 542에 완료되고, 쿼리 B는 시간 544에 완료된다. 따라서, 이러한 실시예에서, 쇼트 쿼리 C 및 D는 롱 쿼리 A 및 B를 지연시킨 대가로 더 빨리 완료될 수 있다.
- [0037] 어떻게 쿼리를 나눔지에 대한 결정은 시스템에서 요구되는 작동 특성에 의존할 수 있다. 예를 들어, 시간을 기초로 하는 쿼리 분할을 제공하는 것은, 각각의 쿼리가 특정 양만큼의 작업을 하도록 허용하는 것을 보장할 수는 있으나, 달력 시간으로 상기 작업이 얼마나 걸리지 여부 및 실행 구간에서 얼마나 많은 열이 반환될 수 있는 지에 대해서는 보장할 수 없다. 반면에, 다수의 열이 반환될 때까지 쿼리가 실행되도록 허용하는 것은, 다수의 결과를 생성하는데 얼마나 많은 실행 구간이 필요할지를 결정할 수 있으나, 상기 구간이 얼마나 지속되어야 하는지는 보장할 수 없다. 다수의 열이 처리될 때까지 쿼리가 실행되도록 허용하는 것은, 시스템이 쿼리를 완료하는데 얼마 많은 실행 구간이 필요할지를 식별하도록 허용할 수 있으나, 특정 수의 열 반환하는데 얼마나 많은 사이클이 요구되는지 또는 구체적으로 특정 실행 사이클이 얼마나 걸릴지는 언급할 수 없다.

- [0038] 쿼리를 처리하는 시간은, 단일 쿼리만이 처리되고 있을지라도, 실행 구간(또는 "쿼리 구간")으로 나누어질 수 있다. 쿼리 구간의 종료 시점에, 새로운 쿼리가 도달하면, 처리되는 쿼리는 보류되고, 다음 쿼리 구간이 새로운 쿼리를 처리하기 위해 사용된다. 그렇지 않고, 쿼리 구간의 종료 시점에 새로운 쿼리가 도달하지 않으면, 처리되는 쿼리는 추가 쿼리 구간 동안 계속하여 처리될 수 있다. 예를 들어, 도 6A의 실시예에서, 쿼리 B는, 쿼리 A가 처리되는 동안 시간 610에 도달하고, 도 6B의 실시예에서는, 쿼리 A 및 B 양자 모두가, 쿼리 A 또는 쿼리 B의 처리가 시작되기 전에 도달한다.
- [0039] 도 6A의 실시예에서, 쿼리 A는 구간 602 동안 실행하고, 만약 쿼리 A가 구간 602의 종료 시점에 완료되지 않는다면, 시스템은 쿼리 A가 추가 쿼리 구간 동안 처리되어야 하는지 여부 또는 처리되길 기다리는 다른 쿼리가 존재하는지 여부를 결정하기 위해 점검한다. 쿼리 B는 구간 602의 종료 시점에 아직 도달하지 않았으므로, 쿼리 A는 쿼리 구간 604 동안 처리된다. 유사하게, 쿼리 A는 또한 이후의 쿼리 구간 606 동안 처리된다. 그러나, 쿼리 구간 606의 종료 시점에, 시스템이 시간 610에 도달한 쿼리 B가 구간 608 동안 처리되어야 한다고 결정한다. 쿼리 A 및 B는, 이후 각각이 완료될 때까지 교번적 구간에서 처리된다(본 실시예에서는, 쿼리 A는 시간 612에서 완료되고, 쿼리 B는 시간 614에서 완료된다). 도 6B의 실시예에서는, 쿼리 A는 구간 620 동안 실행하고, 만약 쿼리 A가 구간 620의 종료 시점에 완료되지 않는다면, 시스템은 쿼리 A가 추가 쿼리 구간 동안 처리되어야 하는지 여부 또는 처리되길 기다리는 다른 쿼리가 존재하는지 여부를 결정하기 위해 점검한다. 쿼리 B가 구간 620의 종료 이전에 도달하였으므로, 쿼리 B는 쿼리 구간 622 동안 처리된다. 쿼리 A 및 B는 이후 각각이 완료될 때까지 교번적 구간에서 처리된다.
- [0040] 쿼리 구간의 종료 시점에서 쿼리를 보류하는 것은 쿼리의 상태를 중개 데이터베이스에 저장하는 단계를 포함한다. 일 구성에서, 일 구간 이후 쿼리 상태는 중개 데이터 베이스에 "보류"로 업데이트 될 수 있으며, 또는 쿼리를 나타내는 다른 상태는 실행될 자격이 없다. 소정의 구간 이후, 쿼리의 상태는, 쿼리를 다시 실행시키도록 활성화하기 위해 "대기"로 업데이트 될 수 있다. 다른 구성에서, 중개 서버는, 소정의 구간 이후 즉시 자동적으로 쿼리를 스케줄링한다.
- [0041] **3. 쿼리 우선순위 부여 및 재우선순위 부여**
- [0042] 중개 데이터베이스는 개별 쿼리에 연관된 우선순위를 저장할 수 있다. 우선권은, 주파수 및 쿼리를 실행하는 방법에 영향을 줄 수 있다. 도 7A를 참조하면, 높은 우선순위의 쿼리 A는 쿼리 B(구간 704 동안 처리됨) 또는 낮은 우선순위의 쿼리 C(구간 706 동안 처리됨)보다 긴 실행 구간 702를 제공받을 수 있다. 이러한 실시예에서, 높은 우선순위의 쿼리 A는 쿼리 B에 제공되는 실행 구간 704보다 긴 실행 구간 702를 제공받을 수 있으며, 쿼리 B는 낮은 우선순위의 쿼리 C에 제공되는 실행 구간 706보다 긴 실행 구간 704를 제공받을 수 있다. 또는, 도 7B를 참조하면, 높은 우선순위의 쿼리 A는 표준 우선순위의 쿼리 B(구간 710 동안 처리됨)보다 더 빈번한 실행 구간 708을 제공받을 수 있으며, 표준 우선순위의 쿼리 B는 낮은 우선순위의 쿼리 C(구간 712 동안 처리됨)보다 더 빈번한 실행 구간을 제공받을 수 있다. 도 7C를 참조하면, 일부 환경에 있어서, 쿼리 A는, 쿼리 A가 실행을 완료할 때까지(구간 714 이후) 다른 쿼리 B 및 C의 처리를 보류시키기에 충분히 높은 우선순위를 제공받을 수 있으며, 보류된 쿼리 B 및 C가 재개되는 실행 시점은 각각 구간 716 및 718 사이에서 교호한다.
- [0043] 중개 데이터베이스는 또한 쿼리가 실행되는 동안 쿼리에 재우선순위를 부여하는 것을 허용한다. 예를 들어, 도 7C를 참조하면, 높은 우선순위의 쿼리 A는 평균 우선순위의 쿼리 B(구간 722 동안) 및 낮은 우선순위의 쿼리 C(구간 724 동안)에 따라 중개 데이터 베이스에 의해 스케줄링된다(구간 720 동안). 시간 726에서, 높은 우선순위의 쿼리 A는 평균 우선순위 레벨로 우선순위가 재부여된다. 이 시점에서, 중개 데이터베이스는 새로운 우선순위 부여를 기초로 쿼리의 일정을 조정한다. 앞으로 재우선순위 부여 후, 현재 평균 우선순위의 쿼리 A는 평균 우선순위의 쿼리 B에 제공되는 구간 722와 유사한 크기의 실행 구간 728을 제공받는다.
- [0044] 재우선순위 부여는 요청 처리에 의한 결정에 기인하여 발생할 수 있으며, 또는 중개 서버 자체의 기준을 기초로 중개 서버 내에서 발생할 수 있다. 예를 들어, 중개 서버는 쿼리를 완료할 기한을 제공받을 수 있으며, 기한이 가까워져 옴에 따라, 중개 서버는 적절한 시점에 반드시 완료시키도록 쿼리의 우선순위를 증가시킬 수 있다. 일부 경우에 있어서, 중개 서버로 하여금 높은 우선순위의 트래픽을 점검하도록 하기 위한 단일 긴 실행 구간 대신, 쿼리는, 다수의 작은 실행 구간을 제공받을 수 있다. 다른 경우에 있어서, 중개 서버는, 높은 우선순위의 쿼리를 실행시키기 위해, 실행 중인 쿼리의 실행 구간을 중지시킬 수 있다.
- [0045] 일부 경우에 있어서, 이전 쿼리의 실행 전 또는 실행 중에, 쿼리들은 실행에 앞서 다음 구간에 들어가는 새로운 쿼리와 함께 스케줄링 될 수 있다. 일부 경우에 있어서, 실행을 위해 스케줄링되어야 하는 다음 쿼리는 선택

기준을 기초로 실행 바로 직전에 선택될 수 있다.

[0046] **4. 병렬 쿼리 처리**

[0047] 많은 시스템에 있어서, 다수의 쿼리를 동시에 실행하는 것이 유리할 수 있다. 예를 들어, 단일 시스템 상에서 실행되는 2개의 쿼리는, 시스템 상에서 실행되는 단일 쿼리 이상의 향상된 성능을 경험할 수 있다. 예를 들어, 하나의 쿼리가 하나의 리소스를 활용하는 동안, 다른 쿼리는 다른 컴퓨팅 리소스를 활용할 수 있으므로, 이러한 경우가 발생할 수 있다. 2개의 쿼리 양자 모두를 동시에 실행시킴으로써, 처리량이 향상된다. 일부 구현들에서, 도 8을 참조하면, 높은 우선순위의 쿼리(802)는 쿼리 조각들 804, 806, 808, 810, 812로 나누어진다. 각각의 조각은 분리된 쿼리 엔진 814, 816, 818, 820, 822에 의해 처리된다.

[0048] 높은 우선순위의 쿼리 802는 전술한 바와 같이 다수의 열을 기초로 조각화되어 처리될 수 있다. 분할 정보는, 쿼리를 완료하기 위해 얼마나 많은 실행 구간이 필요한지를 결정하기 위해, 쿼리의 타겟인, 압축된 색인 데이터 스토리지의 2차 색인과 비교될 수 있다. 이는 또한 압축된 색인 데이터 스토리지의 어느 부분이 각각의 쿼리 조각에 의해 처리되는지를 식별할 수 있다. 예를 들어, 도 9를 참조하면, 압축된 색인 파일(902)은 다수의 데이터 블록들(904, 906, 908, 910)을 포함하고, 각각의 데이터 블록은 다수의 데이터 기록을 포함한다. 압축된 색인 파일(902)은 데이터 블록을 참조하는 색인(912)에 연관된다. 일부 구성에 있어서, 색인은 각 데이터 블록에 대한 하나의 색인 기록(922)을 포함할 수 있으며, 다른 구성에 있어서, 색인(912)은 데이터 블록보다 적은 색인 기록(922)을 포함할 수 있다. 일부 구성에 있어서, 각 색인 기록(922)은 데이터 블록(904, 906, 908, 910)을 참조할 수 있으며, 다른 구성에 있어서, 각 색인 기록(922)은 데이터 블록의 제1 데이터 기록을 참조할 수 있다. 중개 서버는 색인(912)을 재검토하고, 색인 기록을 기초로 쿼리 실행 구간(또는 "쿼리 조각")을 결정한다. 이러한 실시예에서, 쿼리 엔진은 색인(912)을 기초로 4개의 쿼리 조각(914, 916, 918, 920)을 생성할 것을 선택한다. 하나의 쿼리 조각 914는 블록 1(904)로 시작하는 데이터 기록을 처리하여, 블록 10(미도시)의 종점에서 종료하고, 쿼리 조각 916은 블록 11(906)로 시작하는 데이터를 처리하여, 블록 20(미도시)의 종점에서 종료하며, 쿼리 조각 918은 블록 21(908)의 처리로 시작하여, 블록 30(미도시)의 종점에서 종료하며, 마지막 쿼리 조각 920은 블록 31(910)의 처리로 시작하여, 압축된 색인 파일(902)의 종점에서 처리를 종료한다. 이러한 실시예에서, 중개 서버는 임의의 수의 쿼리 조각을 생성할 것을 선택할 수 있으며, 오직 색인(912) 내의 색인 기록(922)의 수에만 제한된다.

[0049] 도 8을 참조하면, 쿼리의 각각의 조각은, 쿼리 엔진(814, 816, 818, 820, 822) 중 상이한 하나의 쿼리 엔진에 의해 동시에 처리될 수 있다. 예를 들어, 쿼리 조각 804는, 쿼리 조각 806이 쿼리 엔진 816에 의해 실질적으로 동시에 처리되는 동안, 쿼리 엔진 814에 의해 처리될 수 있다. 이와 동시에 쿼리 조각 808은 쿼리 엔진 818에 의해 처리될 수 있으며, 쿼리 조각 810은 쿼리 엔진 820에 의해, 쿼리 조각 812는 쿼리 엔진 822에 의해 처리될 수 있다. 각각의 쿼리 엔진은 그들의 쿼리 분할에 대한 결과 세트를 생성한다. 결과 세트 전체가 한번 생성되면, 결과 세트는 함께 결합되어, 전체 쿼리에 대한 완성된 결과 세트를 형성한다. 이러한 방법을 사용하여, 높은 우선순위의 쿼리는, 연산을 완료하는데 일반적으로 걸리는 시간의 일부 내에서 완료될 수 있다.

[0050] **5. 콜백(Callbacks)**

[0051] 시스템은, 사전에 설계된 기준에 의해 정의된 트리거가 충족되면 통지를 제공한다. 도 3을 참조하면, 새로운 쿼리가 프론트-엔드 서비스(302)에 제출될 때, 상기 제출은, 조건이 충족될 때, 중개 서버(304)가 요청자에 통지할 것을(프론트-엔드 서비스(302)를 통해) 요구하는 정보를 포함할 수 있다. 일부 구성에서, 상기 조건은 특정한 수의 결과 데이터 요소가 요청자에 의해 액세스될 준비가 되었을 때, 통지일 수 있다. 예를 들어, 요청자는, 100개의 결과 기록이 준비되었을 때, 통지될 수 있다. 일부 경우에 있어서, 요청자는, 통지 전에 준비되어야 하는 결과 데이터 요소의 수를 설계할 수 있다. 다른 경우에 있어서, 요청자는, 요청자가 통지되기 전에 반드시 충족시켜야 하는 다른 기준을 제공할 수 있다. 예를 들어, 요청자는, 쿼리가 보류되거나 또는 모든 처리가 완료되었을 때, 통지되기를 원할 수 있다. 일부 경우에 있어서, 트리거 기준은 중개 데이터베이스(308) 내에서 추적되는 정보를 명시하도록 제한될 수 있다. 다른 경우에 있어서, 트리거 기준은 제한되지 않을 수 있다. 트리거는 다수의 상이한 방법들에 있어서, 중개 서버(304)에 제공될 수 있다. 예를 들어, 트리거는, 각 쿼리 구간 또는 기설정된 응용 프로그램 인터페이스(application programming interface; API)에 부합하는 컴파일된 클래스를 이후에, 중개 서버(304)가 실행하는 스크립트로서 제공될 수 있다. 일부 경우에 있어서, 조건은 예를 들어, 100개의 결과 기록이 발견되는 조건과 같이 오직 한번 발생할 수 있다. 다른 구성에 있어서, 조건은 예를 들어, 100개의 추가적인 결과 기록이 발견될 때마다 통지를 요청하는 것과 같이 재발생될 수 있다.

[0052] 일부 경우에 있어서, 트리거 조건의 제출은 또한 동작 정의를 포함할 수 있다. 이러한 동작은 트리거에 따라

중개 데이터베이스(308)에 저장될 수 있다. 동작은 조건이 충족되었을 때, 중개 서버(304)가 어떻게 반응하는지를 정의한다. 동작은 예를 들어, 통지, 요약 등과 같은 가능한 동작의 기설정된 세트 중 하나일 수 있다. 동작은 중개 서버(304) 상에서 실행되는 스크립트 일 수 있다. 예를 들어, 하나의 동작은 쿼리 파라미터로서 반환된 결과를 사용하는 시스템에 추가적인 쿼리를 제출할 수 있다. 동작은 기 설정된 API의 형식을 따르는 컴파일된 클래스로서 제공될 수 있다.

[0053] **6. 쿼리 보류**

[0054] 일부 구현들에서, 중개 서버(304)는 쿼리의 처리를 보류할 수 있다. 중개 서버는 쿼리를 보류된 것으로 표시할 수 있으며, 쿼리가 재개될 때까지 처리되지 않을 수 있다. 쿼리 보류에 따라 쿼리 서버는 쿼리의 상태를 저장할 수 있다. 이러한 상태는 쿼리의 처리에 대한 지시이다. 예를 들어, 상태는 압축된 색인 데이터 저장소에 대한 오프셋일 수 있으며, 또는 상태는 b-트리에서 최근 처리된 노드를 포함할 수 있다.

[0055] 일부 경우에 있어서, 중개 서버는 자체의 결정에 의해 쿼리를 보류하기로 결정할 수 있다. 예를 들어, 쿼리 결과 세트 내의 다수의 기록 및 문턱값(threshold)을 초과하는 요청자에 전달되도록 대기하는 결과 세트 내의 열의 수를 생성할 때, 이러한 상황이 발생할 수 있다.

[0056] 예를 들어, 만약 사용자 인터페이스가 스크린을 실장하기 위해 데이터의 25개 열의 "페이지"를 요청할 수 있고, 시스템이 쿼리로부터 25 열의 데이터를 요청할 수 있다면, 쿼리를 제출하는 요청자는 고정된 수의 열의 전달을 후에 요청할 수 있다. 이후에, 만약 사용자가 더 많은 쿼리 결과를 보길 원한다는 것을 나타내면, 시스템은 다음 "페이지"의 쿼리 결과 또는 26에서 50까지의 결과를 요청할 수 있다. 중개 데이터베이스는 쿼리로부터 반환된 결과의 수 및 사용자에게 반환되는 결과의 수를 추적한다. 예를 들어, 쿼리는 300개의 열이 반환될 수 있으나, 25개의 열이 요청자에게 발송될 수 있다. 만약 쿼리로부터 반환되는 열의 수가 마진(margin)에 따라 요청자에게 발송되는 열의 수(예를 들어, 25, 50, 75 또는 100 열)를 초과하게 되면, 중개 데이터베이스는 해당 쿼리의 처리를 보류할 수 있다. 이것은 중개 데이터베이스내의 쿼리에 보류되었다는 것을 표시하거나 또는 쿼리의 다음 실행을 스케줄링하기에 앞서 점검하는 것을 통해 할 수 있을 것이다.

[0057] 일부 경우에 있어서, 문턱값은 시스템 300에 의해 정의되며, 다른 경우에 있어서, 문턱값은 쿼리가 어떻게 사용될지에 따라 각각의 쿼리에 대해 개별적으로 정의될 수 있다. 예를 들어, 쿼리의 결과가, 고정된 수의 아이템 표시와 함께 웹 페이지 상에서 데이터의 리스트를 표시하는데 사용될 경우, 그 쿼리는 네 페이지의 데이터가 대기하고 있는 경우 보류될 수 있다. 반면에, 쿼리의 결과가, 예를 들어 월말 보고와 같이 쿼리에 의해 반환되는 모든 데이터의 요약 보고를 생성하는데 사용될 경우, 그 쿼리는 보류되지 않을 수 있다. 일부 경우에 있어서, 문턱값은 열의 수에 의해 추정되어 요청자에게 통지하기 전에 수정할 수 있다.

[0058] 일부 경우에 있어서, 쿼리는, 중개 데이터베이스에서 쿼리의 상태 정보를 업데이트함으로써, 명시적으로 보류될 수 있다. 예를 들어, 쿼리는 보류되었다고 표시되어 높은 우선순위의 쿼리가 실행되도록 할 수 있다. 다른 경우에 있어서, 쿼리는, 중개 서버 스케줄링 알고리즘이 보류된 쿼리 상태를 갖는 쿼리가 스케줄링 되지 않도록 하기 때문에 암시적으로 보류될 수 있다. 쿼리 보류는, 쿼리가 제출되고, 이후 호출 프로그램이 쿼리가 완료되기 전에 종결할 때, 자원의 낭비를 최소화하는 추가적인 장점이 있다. 중개 서버는, 만약 요청자가 기 설정된 기간에서 결과에 액세스하지 않는다면, 쿼리 및 결과 세트를 삭제할 것을 선택할 수 있다.

[0059] **7. 중개 서버 처리**

[0060] 도 10을 참조하면, 흐름도 1000은 외부 요청 없이 쿼리의 처리를 보류할지 여부에 따른 결정을 포함하는, 중개 서버(304)의 작동의 예시적인 구성을 나타낸다.

[0061] 작동은 실행할 쿼리의 선택 단계(1002)를 포함한다. 일 실시예에서, 쿼리는 중개 서버에 의해 수립된 기설정된 스케줄의 일부로서 선택될 수 있다. 다른 실시예에서, 쿼리는 몇 가지 기준을 기초로 선택될 수 있으며, 상기 기준은 쿼리의 우선순위 및 마지막으로 쿼리가 실행된 시간을 포함할 수 있다. 일부 구성에 있어서, 중개 서버는 실행을 기다리는 쿼리들(예를 들어, 대기 상태) 상에서 반복할 수 있다. 각각의 쿼리는 쿼리 엔진 상에서 실행되도록 스케줄링된다. 대기중인 모든 쿼리들이 한번 실행되면, 중개 서버는 여전히 실행되길 기다리는 쿼리를 위해 상기 프로세스를 반복한다. 다른 구성에 있어서, 중개 서버는 가장 긴 실행 구간을 기다리는 쿼리를 선택한다. 다른 구성에 있어서, 중개 서버는 실행에 있어서 가장 높은 우선순위를 갖는 쿼리를 선택한다.

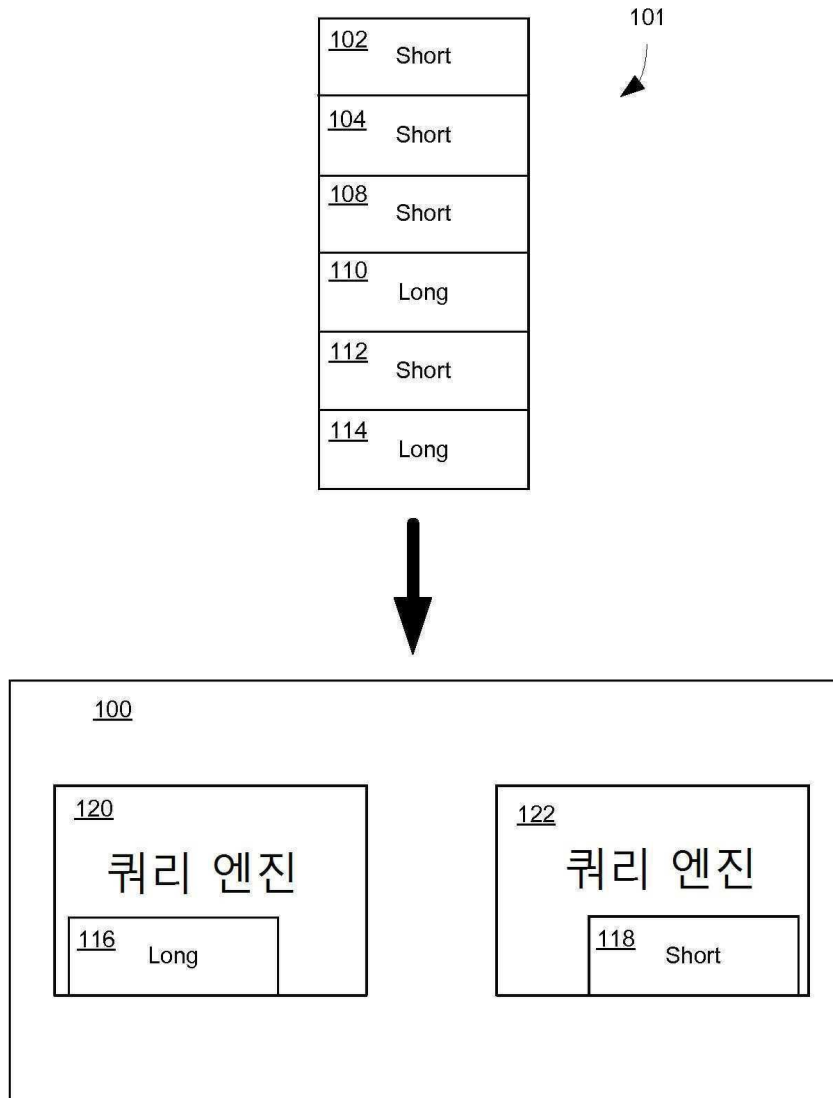
[0062] 작동은 또한 쿼리 엔진 상에서 쿼리를 실행하는 단계(1004)를 포함한다. 일 실시예에서, 선택된 쿼리는 데이터 레코드에 반하여 쿼리를 실행시키고, 결과 세트를 업데이트하며, 반환되는 열의 수의 중개 서버를 통지하는 쿼

리 엔진에 할당될 수 있다.

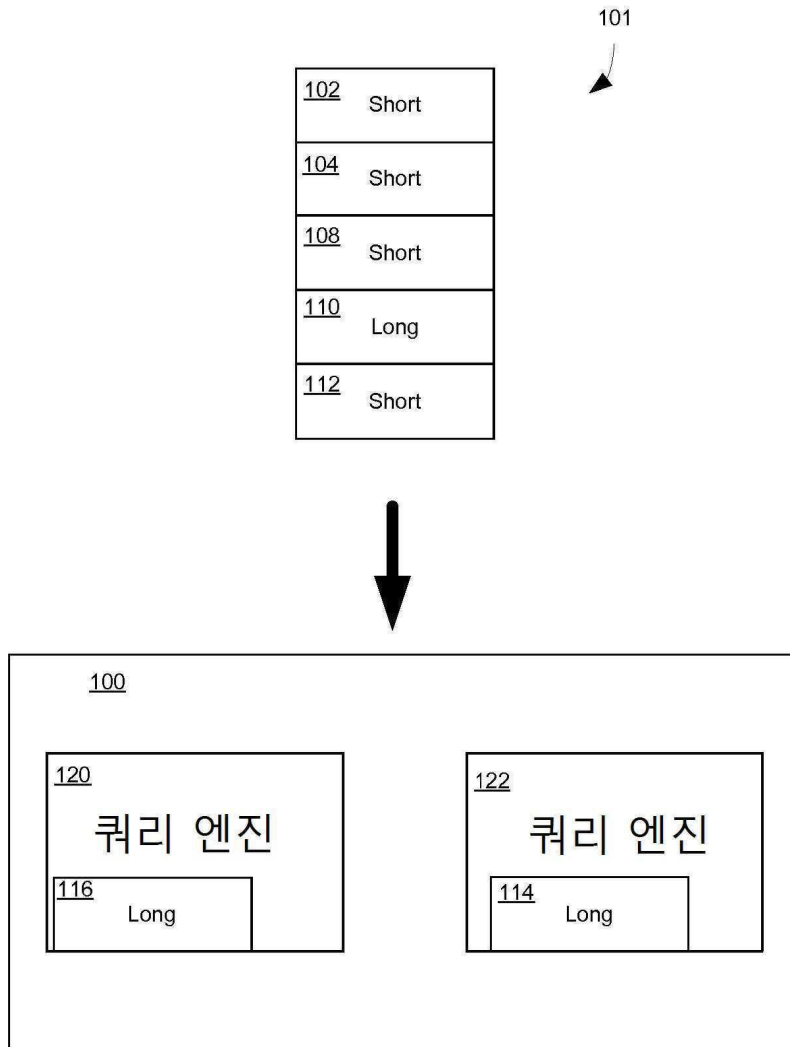
- [0063] 작동은 또한 전달되길 기다리는 열의 수를 점검하는 단계(1006)를 포함한다. 만약 전달되길 기다리는 열의 수가 통지된 문턱값을 초과하면, 중개 서버는 요청자에 대하여 콜백을 수행한다(1008).
- [0064] 작동은 또한, 만약 요청자가 액세스하기를 기다리는 열의 수가 보류된 문턱값을 초과하는지 여부를 점검하는 단계(1010) 및 만약 초과하는 경우 쿼리를 보류시키는 단계(1012)를 포함한다. 쿼리가 보류되거나 보류되지 않는 경우, 중개 서버는 선택된 다음 쿼리를 처리하기 위해 이동한다.
- [0065] 도 11을 참조하면, 흐름도 1100은 예를 들어, 콜백이 요청자에게 결과가 액세스할 준비가 되었음을 통지한 후, 쿼리에 의해 반환되는 결과 세트의 일부분에 액세스하는 요청자에 응답한, 중개 서버(304)의 작동의 일 구성예를 나타낸다.
- [0066] 작동은 쿼리로부터 결과를 요청하는 요청자(1102)를 포함한다. 일부 구성에서, 요청자는 예를 들어 25개의 열을 반환하는 요청과 같은, 반환되어야 하는 열의 수에 대한 지시를 발송할 수 있다. 다른 구성에서, 요청자는 특정 범위의 반환되어야 하는 결과를 요청할 수 있다. 예를 들어, 요청자는 50에서 126 사이의 결과가 반환되어야 함을 요청할 수 있다. 또 다른 구성에서, 요청하는 모든 집합된 결과가 반환되어야 함을 요청할 수 있다.
- [0067] 작동은 또한 결과를 반환하고 기록을 업데이트하는 단계(1104)를 포함한다. 요청에 응답하여, 중개 서버는 요청되는 열에 대한 액세스를 제공할 수 있다. 일부 구성에서, 중개 서버는 또한 쿼리가 여전히 추가 결과를 처리중임을 나타내는 지시를 요청자에게 발송할 수 있다. 다른 구성에서, 중개 서버는 또한 추가 결과는 즉시 전달 가능함을 나타내는 지시를 제공할 수 있다.
- [0068] 작동은 또한 쿼리가 현재 보류되는지 여부를 결정하기 위해 점검하는 단계(1106)를 포함할 수 있다. 만약 쿼리가 보류된다면, 제어는 다음 작동으로 넘어간다(1108). 그렇지 않다면 프로세스는 완료된다.
- [0069] 작동은 또한 전달되기를 기다리는 열의 수가 보류된 문턱값보다 작은지 여부를 결정하기 위해 점검하는 단계(1108)를 포함할 수 있다. 만약 그렇다면, 쿼리는 재개되고(1110), 중개 서버에 의해 처리되기 위해 스케줄링될 수 있다.
- [0070] 쿼리 관리 접근은 컴퓨터에서 실행되는 소프트웨어를 사용하여 구현될 수 있다. 예를 들어, 소프트웨어는 하나 이상의 프로세서, 하나 이상의 데이터 저장 시스템(휘발성 및 불휘발성 메모리 및/또는 저장 요소를 포함), 하나 이상의 입력 장치 또는 포트, 및 하나 이상의 출력 장치 또는 포트를 각각 포함하는 하나 이상의 프로그램화된 또는 프로그램가능한 컴퓨터 시스템(분산형, 클라이언트/서버, 또는 그리드(grid)와 같은 다양한 구조가 가능함)에서 실행되는 하나 이상의 컴퓨터 프로그램에서의 절차를 형성한다. 소프트웨어는, 예를 들어 연산 그래프의 구성 및 설정에 관련된 다른 서비스를 제공하는, 더 큰 프로그램의 하나 이상의 모듈을 구성할 수 있다. 그래프의 노드와 요소는 컴퓨터로 판독가능한 매체에 저장된 데이터 구조 또는 데이터 저장소에 저장된 데이터 모델에 부합하는 다른 구성의 데이터로서 구현될 수 있다.
- [0071] 소프트웨어는 범용 또는 전용의 프로그램가능한 컴퓨터로 판독가능한 CD-ROM과 같은 저장 매체에 제공되거나, 네트워크와 같은 통신 매체를 통해 실행가능한 컴퓨터로 전달(전파 신호로 부호화되는)될 수 있다. 모든 기능은, 전용의 컴퓨터상에서, 또는 코프로세서와 같은 전용의 하드웨어를 사용해서 수행될 수 있다. 소프트웨어는, 해당 소프트웨어에 의해 특정된 연산의 상이한 부분이 여러 컴퓨터에서 수행되는 분산 방식으로 구현되어도 된다. 이러한 각각의 컴퓨터 프로그램은, 본 명세서에서 설명하는 절차를 수행하도록 저장 매체 또는 디바이스가 컴퓨터 시스템에 의해 판독될 때에 컴퓨터를 구성 및 운영하기 위한, 범용 또는 전용의 프로그램가능한 컴퓨터에 의해 판독가능한 저장 매체 또는 디바이스(예를 들어, 고체 메모리 또는 매체, 자기 또는 광학 매체)에 저장되거나 다운로드되도록 하는 것이 바람직하다. 본 발명의 시스템은 컴퓨터 프로그램에 맞게 구성된, 컴퓨터로 판독가능한 저장 매체로서 구현될 수도 있는데, 이러한 저장 매체는 컴퓨터 시스템으로 하여금, 본 명세서에서 설명한 기능의 수행을 위해 특정되고 미리 정해진 방식으로 동작할 수 있도록 구성된다.
- [0072] 본 발명에 대하여 많은 실시예를 설명하였다. 그렇지만, 본 발명의 범위를 벗어남이 없이 다양한 변형이 가능하다는 것을 알 수 있을 것이다. 예를 들어, 상기 설명한 단계들 중 몇몇은 반드시 그 순서대로 수행되지 않아도 되며, 설명된 것과 다른 순서대로 수행되어도 된다.
- [0073] 이상의 설명은 청구범위에 의해 정해지는 본 발명의 범위를 제한하기 위한 것이 아니라 예시일 뿐이다. 예를 들어, 앞서 설명한 많은 기능적 단계들은 전체적인 과정에 실질적인 영향을 미치지 않으면서, 다른 순서로 수행되어도 된다. 다른 실시예는 이하의 청구범위에 포함된다.

도면

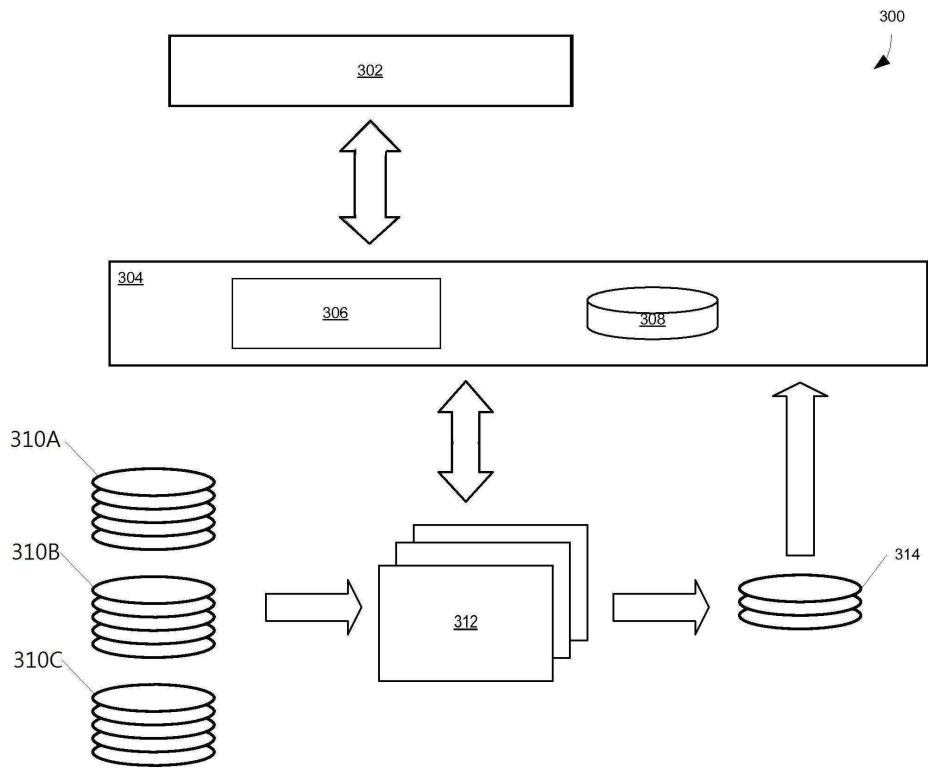
도면1



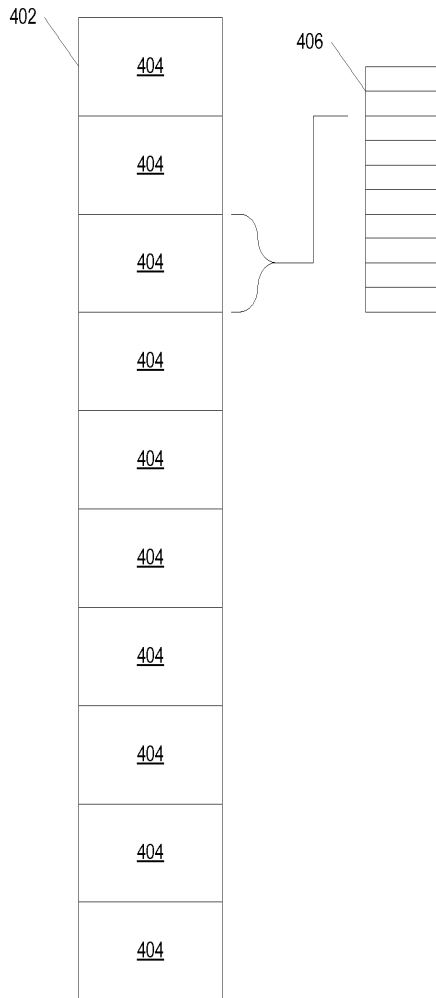
도면2



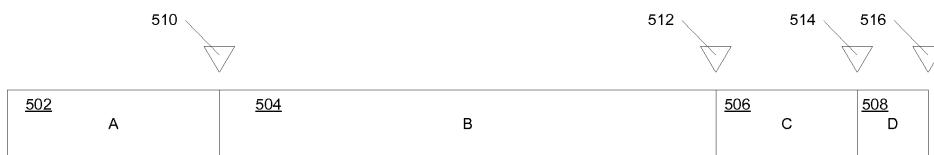
도면3



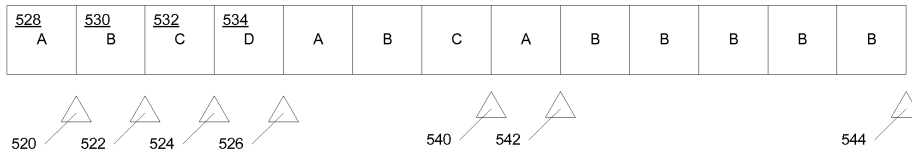
도면4



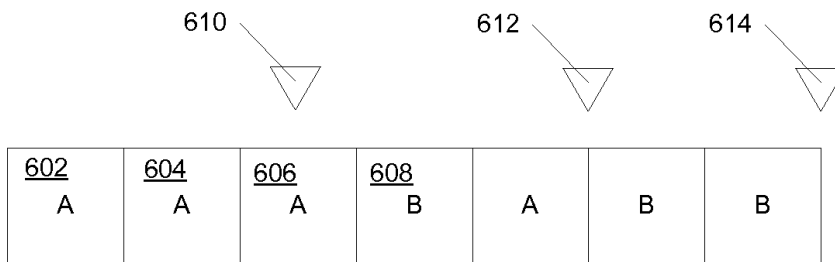
도면5a



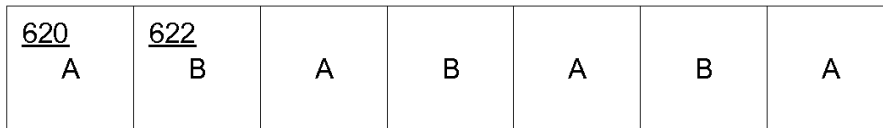
도면5b



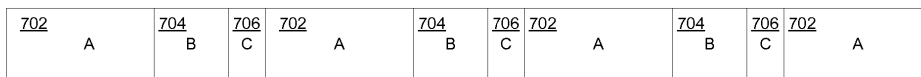
도면6a



도면6b



도면7a



도면7b

<u>708</u> A	<u>710</u> B	<u>708</u> A	<u>712</u> C	<u>708</u> A	<u>710</u> B	<u>708</u> A	<u>704</u> B	<u>712</u> C	<u>708</u> A	<u>710</u> B	<u>708</u> A	<u>712</u> C
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

도면7c

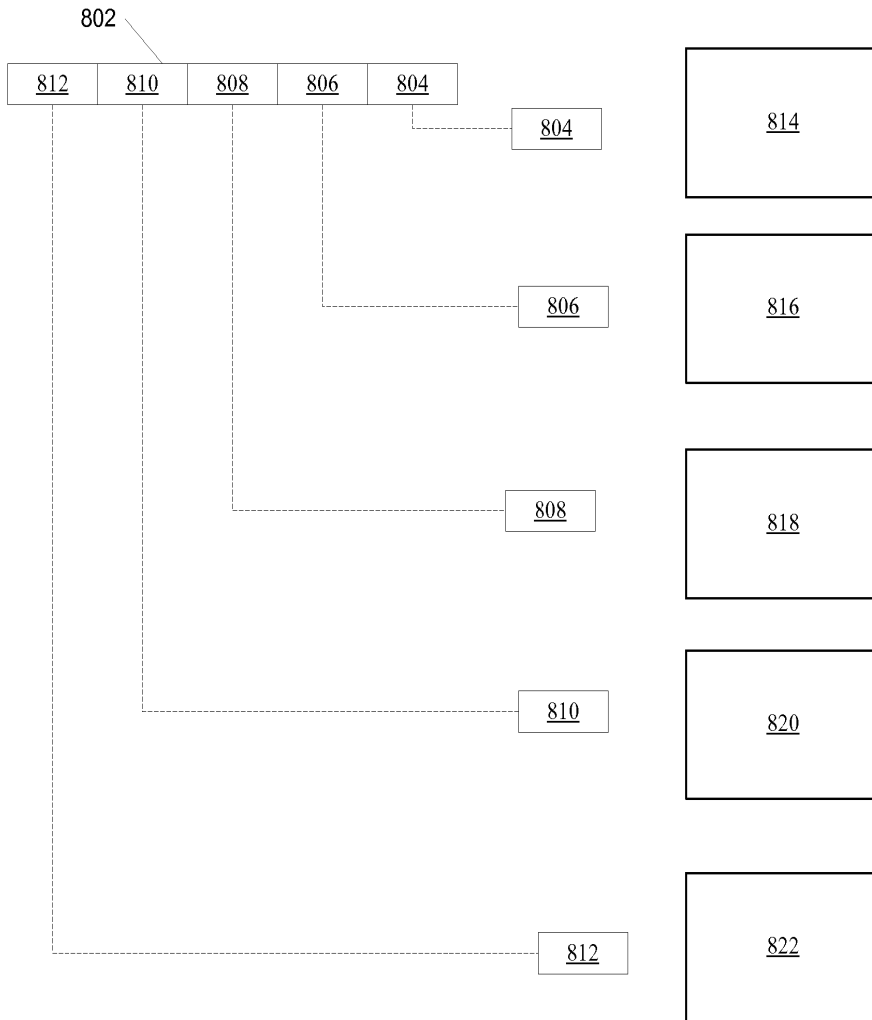
<u>714</u> A	<u>716</u> B	<u>718</u> C	<u>716</u> B	<u>718</u> C	<u>716</u> B	<u>718</u> C
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

도면7d

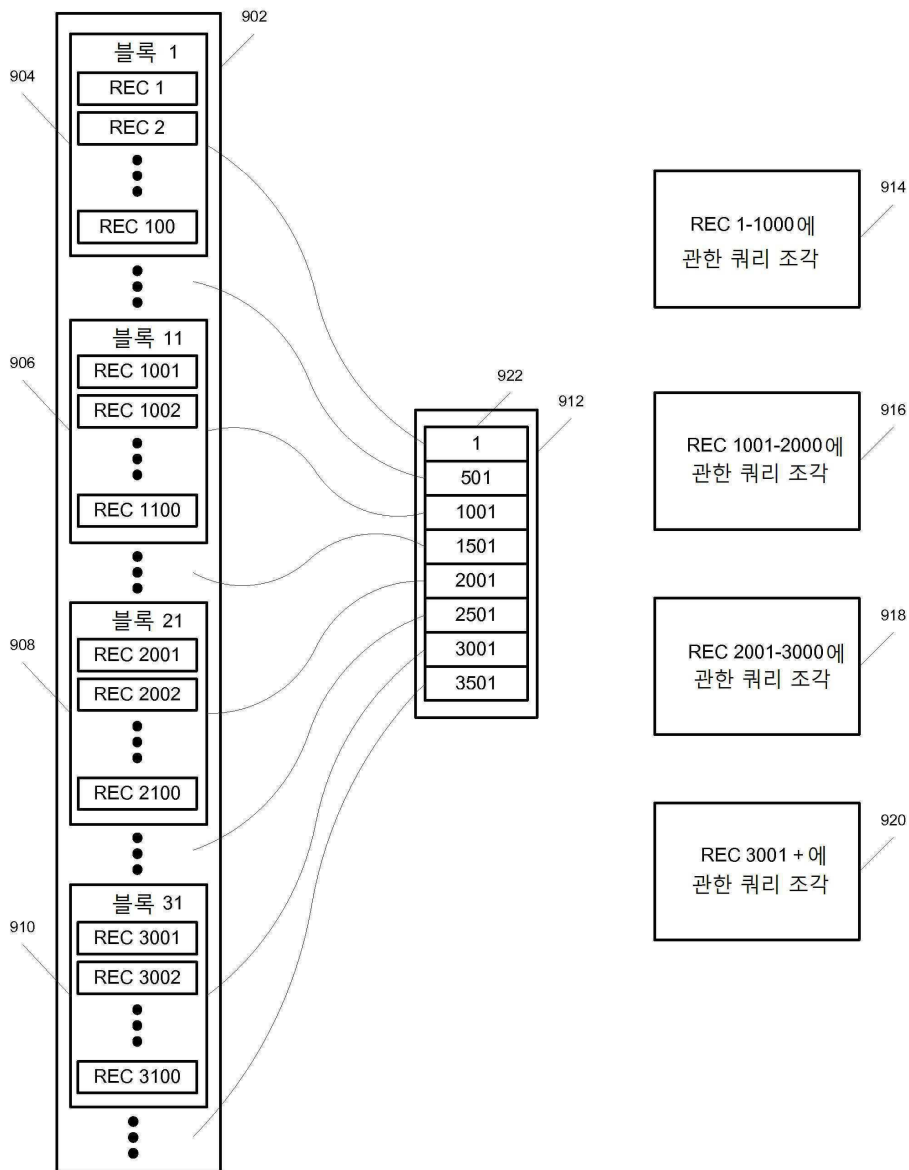
<u>720</u> A	<u>722</u> B	<u>724</u> C	<u>720</u> A	<u>722</u> B	<u>724</u> C	<u>728</u> A	<u>722</u> B	<u>724</u> C	<u>728</u> A	<u>722</u> B	<u>724</u> C	<u>728</u> A
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

726 

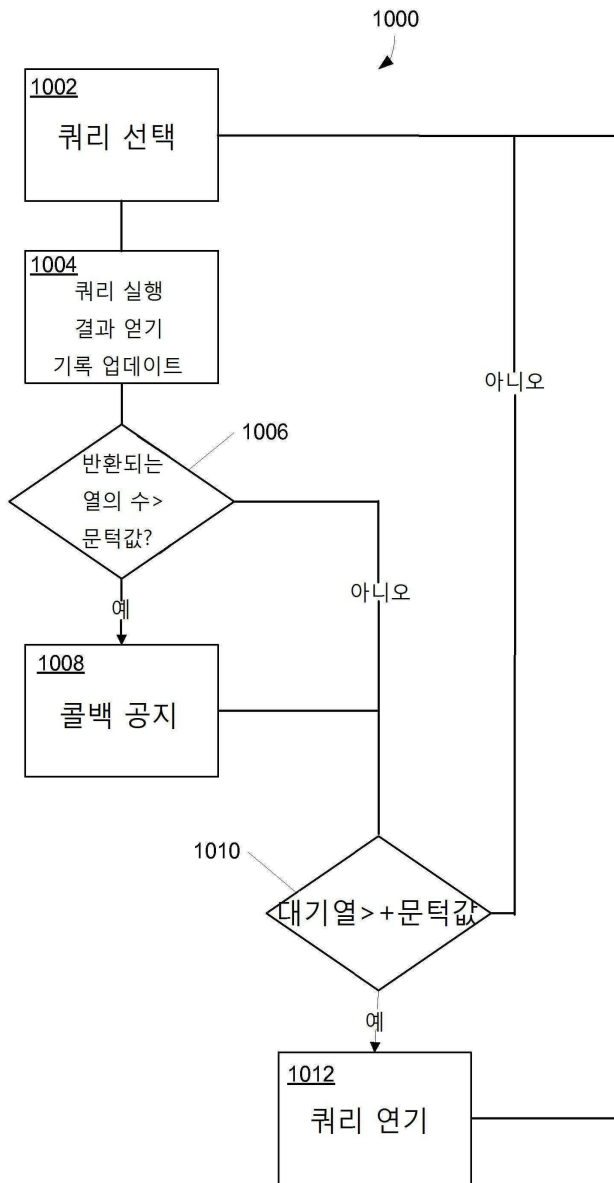
도면8



도면9



도면10



도면11

