(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0135125
(43) 공개일자 2014년11월25일

(51) 국제특허분류(Int. Cl.)
*G06F 17/00* (2006.01)
(21) 출원번호 10-2014-0058359
(22) 출원일자 2014년05월15일
심사청구일자 2014년05월15일
(30) 우선권주장
1020130055048 2013년05월15일 대한민국(KR)

(71) 출원인
**국립대학법인 울산과학기술대학교 산학협력단**
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
**남범석**
울산광역시 울주군 범서읍 구영로 75-25, 105동 501호 (울산구영동문굿모닝힐)
(74) 대리인
**특허법인충정**

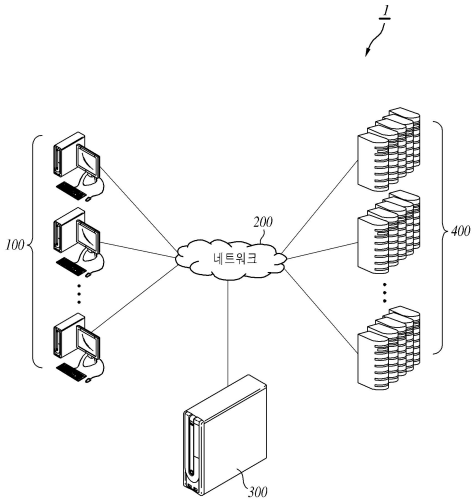
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **질의 할당 방법**

## (57) 요 약

질의 할당 방법이 제공되며, 적어도 하나의 클라이언트로부터 질의 데이터를 수신하는 단계, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버에 할당하기 위하여, 적어도 하나의 어플리케이션 서버의 EMA(Exponential Moving Average)를 계산하는 단계, 적어도 하나의 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 파악하는 단계, 계산된 EMA와 부하에 기반하여, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당하는 단계를 포함한다.

*대 표 도* - 도1



이 발명을 지원한 국가연구개발사업

| | |
|---|---|
| 과제고유번호 | 1415122825 |
| 부처명 | 지식경제부 |
| 연구관리전문기관 | 한국산업기술평가관리원 |
| 연구사업명 | 산업원천기술개발사업 |
| 연구과제명 | 차세대 메모리 기반의 스마트 디바이스용 임베디드 시스템 소프트웨어 원천기술 개발 |
| 기 여 율 | 1/1 |
| 주관기관 | 국립대학법인 울산과학기술대학교 산학협력단 |
| 연구기간 | 2012.06.01 ~ 2013.05.31 |

## *특허청구의 범위*

**청구항 1**

질의 할당 서버에서 실행되는 질의 할당 방법에 있어서,

적어도 하나의 클라이언트로부터 질의 데이터를 수신하는 단계;

상기 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버에 할당하기 위하여, 상기 적어도 하나의 어플리케이션 서버의 EMA(Exponential Moving Average)를 계산하는 단계;

상기 적어도 하나의 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 파악하는 단계;

상기 계산된 EMA와 부하에 기반하여, 상기 수신된 질의 데이터를 상기 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당하는 단계;

를 포함하는, 질의 할당 방법.

**청구항 2**

제 1 항에 있어서,

상기 EMA를 계산하는 단계는,

상기 적어도 하나의 어플리케이션 서버에 EMA를 랜덤하게 부여하는 단계;

상기 수신된 질의 데이터의 다차원 중심 포인트와 상기 EMA 간 거리를 계산하는 단계;

를 포함하는 것인, 질의 할당 방법.

**청구항 3**

제 1 항에 있어서,

상기 계산된 EMA와 부하에 기반하여, 상기 수신된 질의 데이터를 상기 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당하는 단계는,

상기 계산된 EMA와 부하를 곱하여 계산된 총 EMA를 상기 적어도 하나의 어플리케이션에 부여하고,

상기 수신된 질의 데이터의 다차원 중심 포인트와, 상기 부여된 총 EMA 간 거리를 계산하고,

상기 계산된 거리가 가장 짧은 상기 어느 하나의 어플리케이션 서버에 상기 질의 데이터를 할당함으로써 수행되는 것인, 질의 할당 방법.

**청구항 4**

제 3 항에 있어서,

상기 다차원 중심 포인트와 상기 EMA 간 거리는, 유클라디언 거리로 정의되는 것인, 질의 할당 방법.

**청구항 5**

제 1 항에 있어서,

상기 질의 데이터는, 동일한 확률을 가지고 상기 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당되는 것인, 질의 할당 방법.

**청구항 6**

제 1 항에 있어서,

상기 질의 데이터는, 상기 적어도 하나의 어플리케이션 서버에 이미 할당된 질의 데이터와의 유사도를 고려하여, 상기 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당되는 것인, 질의 할당 방법.

**청구항 7**

제 1 항에 있어서,

상기 질의 데이터가 할당된 어플리케이션 서버의 EMA 및 부하에 기초한 최종 EMA를 업데이트하는 단계

를 더 포함하는 것인, 질의 할당 방법.

**청구항 8**

제 1 항에 있어서,

상기 질의 데이터가 수신되는 경우, 상기 질의 데이터를 상기 질의 할당 서버의 큐에 저장된 가장 오래된 질의 데이터와 대체하여 저장하는 단계;

를 더 포함하는 것인, 질의 할당 방법.

**청구항 9**

제 1 항에 있어서,

상기 계산된 EMA와 부하에 기반하여, 상기 수신된 질의 데이터를 상기 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당하는 단계는,

하기 수학식을 이용하여, 상기 질의 데이터를 상기 어느 하나의 어플리케이션 서버에 할당하는 것인, 질의 할당 방법:

$$EMA_{i^*} = \alpha q + (1-\alpha) EMA_{i^*}$$

$$Bound(i,j) : Bound(j,i) = Load(j) : Load(i)$$

여기서, EMAi*는 i번째 어플리케이션 서버의 EMA를 의미하고, α는 0 내지 1의 가중치를 의미하고, q는 질의 데이터의 다차원 중심 포인트를 의미하고, Bound(i,j)는 i번째 어플리케이션 서버와 j번째 어플리케이션 서버 간의 EMA 거리를 의미하고, Bound(j,i)는 j번째 어플리케이션 서버와 i번째 어플리케이션 서버 간의 EMA 거리를 의미하고, Load(j)는 j번째 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 의미하고, Load(i)는 i번째 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 의미한다.

**청구항 10**

제1항 내지 제9항의 방법을 프로세서에서 실행시키기 위한 프로그램을 기록한 프로세서가 읽을 수 있는 기록매체.

## *명 세 서*

## 기 술 분 야

[0001] 본 발명은 질의 할당 방법에 관한 것이다.

## 배 경 기 술

[0002] 최근, 분산 네트워크는 대규모 데이터를 저장하고, 검색하여 처리하는데 다양하게 이용되고 있으며, 분산 네트워크에 질의 데이터를 어떻게 할당하고 분산할 것인지는 다양하게 연구되고 있다.

[0003] 이때, 질의 데이터는 각 분산 자원에 부하를 줄이는 방법으로 할당되고 있다. 이와 관련하여, 한국공개특허인 제2003-0045987호(2003.06.12 공개)에는 라운드-로빈 방식의 스케줄링 방법을 개시한다.

[0004] 다만, 라운드-로빈 방식의 스케줄링 방법은, 분산 네트워크 내에서 캐시 히트율(Cache Hit Rate)을 무시하므로 데이터의 재사용률이 적고, 분산 네트워크에 작업의 수를 균등하게 분배하는 것만으로는 분산 네트워크를 효율적으로 활용하는 방법이 아니다.

## 발명의 내용

### 해결하려는 과제

[0005] 본 발명의 일 실시예는, 질의 데이터의 중심 포인트와, 복수의 어플리케이션 서버가 가지는 EMA(Exponential Moving Average) 간의 거리가 최소화되는 서버를 선택할 뿐만 아니라, 복수의 어플리케이션에 할당된 부하에 대응하는 질의의 수까지 고려하여 EMA 영역을 설정함으로써, 로드 밸런싱과 캐시 히트율을 동시에 고려하여 질의 데이터가 처리되도록 질의 데이터를 어플리케이션 서버에 할당할 수 있는, 질의 할당 방법을 제공할 수 있다. 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

### 과제의 해결 수단

[0006] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 일 실시예는, 적어도 하나의 클라이언트로부터 질의 데이터를 수신하는 단계, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버에 할당하기 위하여, 적어도 하나의 어플리케이션 서버의 EMA(Exponential Moving Average)를 계산하는 단계, 적어도 하나의 어플리케이션 서버에 할당된 작업량인 부하(Load)에 대응하는 질의의 수를 파악하는 단계, 계산된 EMA와 부하에 기반하여, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당하는 단계를 포함한다.

### 발명의 효과

[0007] 전술한 본 발명의 과제 해결 수단 중 어느 하나에 의하면, 분산 환경에서 시스템에 과부하를 야기하지 않으면서도 로드 밸런싱과 캐시 히트율을 동시에 고려하여 질의가 처리되도록 할 수 있다.

## 도면의 간단한 설명

[0008] 도 1은 본 발명의 일 실시예에 따른 질의 할당 시스템을 설명하기 위한 구성도이다.

도 2는 도 1에 도시된 질의 할당 서버를 설명하기 위한 구성도이다.

도 3은 도 1에 도시된 질의 할당 서버에서 EMA 포인트와 질의 간의 거리를 계산하는 것을 도시한 도면이다.

도 4는 도 1에 도시된 질의 할당 서버에서 부하를 고려하여 EMA 포인트와 질의 간의 거리를 계산하는 것을 도시한 도면이다.

도 5는 도 1에 도시된 질의 할당 서버에서 어플리케이션 서버의 부하를 밸런싱하는 일 실시예를 도시한 도면이다.

도 6은 도 1에 도시된 질의 할당 서버에서 EMA 포인트 및 부하에 기반하여 질의를 할당하는 알고리즘의 일 실시예를 도시한다.

도 7은 본 발명의 일 실시예에 따른 질의 할당 방법의 성능 평가 그래프를 도시한 도면이다.

도 8은 본 발명의 일 실시예에 따른 질의 할당 방법을 설명하기 위한 동작 흐름도이다.

## *발명을 실시하기 위한 구체적인 내용*

[0009] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0010] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미하며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0011] 이하 첨부된 도면을 참고하여 본 발명을 상세히 설명하기로 한다.

[0012] 도 1은 본 발명의 일 실시예에 따른 질의 할당 시스템을 설명하기 위한 구성도이다. 본 발명의 일 실시예에 따른 질의 할당 시스템(1)은, 적어도 하나의 클라이언트(100), 질의 할당 서버(300) 및 적어도 하나의 어플리케이션 서버(400)를 포함할 수 있다. 다만, 도 1의 질의 할당 시스템(1)은 본 발명의 일 실시예에 불과하므로, 도 1을 통해 본 발명이 한정 해석되는 것은 아니다.

[0013] 이때, 도 1의 각 구성요소들은 일반적으로 네트워크(network, 200)를 통해 연결될 수 있다. 예를 들어, 도 1에 도시된 바와 같이, 네트워크(200)를 통하여 적어도 하나의 클라이언트(100)와 질의 할당 서버(300)가 연결될 수 있다. 또한, 질의 할당 서버(300)는 네트워크를 통하여 적어도 하나의 어플리케이션 서버(400)와 연결될 수 있다. 그리고, 적어도 하나의 클라이언트(100)는 네트워크(200)를 통하여 적어도 하나의 어플리케이션 서버(400)와 연결될 수 있다.

[0014] 여기서, 네트워크(200)는 단말들 및 서버들과 같은 각각의 노드 상호 간에 정보 교환이 가능한 연결 구조를 의미하는 것으로, 이러한 네트워크(200)의 일 예는, 인터넷(Internet), LAN(Local Area Network), Wireless LAN(Wireless Local Area Network), WAN(Wide Area Network), PAN(Personal Area Network), 3G, 4G, LTE, Wi-Fi 등이 포함되나 이에 한정되지는 않는다. 도 1에 도시된 적어도 하나의 클라이언트(100), 질의 할당 서버(300) 및 적어도 하나의 어플리케이션 서버(400)는, 도 1에 도시된 것들로 한정 해석되는 것은 아니다.

[0015] 적어도 하나의 클라이언트(100)는, 적어도 하나의 어플리케이션 서버(400)로 적어도 하나의 질의를 전송할 수 있다. 그리고, 적어도 하나의 클라이언트(100)는, 적어도 하나의 어플리케이션 서버(400)에서 질의를 처리한 결과인 응답 데이터를 수신할 수 있다.

[0016] 질의 할당 서버(300)는, 적어도 하나의 클라이언트(100)로부터 질의 데이터를 수신할 수 있다. 그리고, 질의 할당 서버(300)는, 적어도 하나의 어플리케이션 서버(400)의 EMA(Exponential Moving Average, 기하 평균 이동) 포인트 또는 적어도 하나의 어플리케이션 서버(400)에 이미 할당된 질의 데이터에 대응하는 부하(Load)를 고려하여, 적어도 하나의 클라이언트(100)로부터 수신된 질의 데이터를 할당할 수 있다.

[0017] 적어도 하나의 어플리케이션 서버(400)는, 적어도 하나의 클라이언트(100)로부터 수신되고, 질의 할당 서버(300)에서 분배된 질의 데이터를 처리할 수 있다. 그리고, 적어도 하나의 어플리케이션 서버(400)는 질의 데이터의 처리 결과인 질의에 대한 응답 데이터를, 적어도 하나의 클라이언트(100)로 전송할 수 있다.

[0018] 도 2는 도 1에 도시된 질의 할당 서버를 설명하기 위한 구성도이고, 도 3은 도 1에 도시된 질의 할당 서버에서 EMA 포인트와 질의 간의 거리를 계산하는 것을 도시한 도면이고, 도 4는 도 1에 도시된 질의 할당 서버에서 부하를 고려하여 EMA 포인트와 질의 간의 거리를 계산하는 것을 도시한 도면이다.

[0019] 도 2를 참조하면, 본 발명의 일 실시예에 따른 질의 할당 서버(300)는, 수신부(410), 계산부(420), 파악부(430), 할당부(440), 업데이트부(450)를 포함할 수 있다.

[0020] 수신부(410)는, 적어도 하나의 클라이언트(100)로부터 질의 데이터를 수신할 수 있다.

[0021] 계산부(420)는, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버(400)에 할당하기 위하여, 적어도 하나

의 어플리케이션 서버(400)의 EMA(Exponential Moving Average)를 계산할 수 있다. 여기서, 계산부(420)는, 적어도 하나의 어플리케이션 서버(400)에 EMA를 랜덤하게 부여하고, 수신된 질의 데이터의 다차원 중심 포인트와 EMA 간 거리를 계산할 수 있다.

[0022] 이하에서, EMA의 개념에 대하여 간략하게 설명한다.

[0023] EMA는 장기 또는 단기 추세 변동을 파악하기 위한 통계적인 방법인데, 계산부(420)에서는 수신된 질의 데이터에 가중치를 두고, 가중 평균을 구하는데 사용할 수 있다. 즉, 캐시(Cache)된 데이터는 다차원 좌표로 표현될 수 있는데, 다차원 중심 포인트를 캐시 데이터의 EMA 값을 구하는데 사용할 수 있다. 시간의 변동에 따른 EMA는 하기 수학식 1과 같이 표현될 수 있다.

*수학식 1*

[0024] $$EMA[t]=\alpha \cdot p[t]+(1-\alpha) \cdot EMA[t-1]$$

[0025] 여기서, t는 시간을 의미하고, EMA[t]는 시간에 따라 변하는 EMA를 의미하고, α는 0 내지 1 사이의 값을 가지는 가중치를 의미한다. 여기서, EMA에 대한 수식 및 정의는 당업자에게 자명하므로 설명을 생략하기로 한다.

[0026] 다시, 도 2의 계산부(320)의 설명으로 돌아와서, 본 발명에 따른 질의 할당 방법에서는, 계산부(320)는, 분산 캐시 인프라구조를 가지는 적어도 하나의 어플리케이션 서버(400)가, 분포된 질의가 변하는 상황에서도 적응할 수 있도록 EMA 포인트를 할당할 수 있다. 즉, n 개의 어플리케이션 서버(400)와 하나의 질의 할당 서버(300)가 존재한다고 가정하면, 계산부(320)는, EMA 포인트의 초기화를 위해서 n 개의 어플리케이션 서버(400)에 부여할 EMA 변수를 랜덤하게 결정할 수 있다.

[0027] 본 발명의 일 실시예에 따른 질의 할당 방법은, i) EMA와 ii) 부하를 모두 고려할 수 있다. 우선, i) EMA 만을 이용하여 질의를 할당하는 방법을 설명한 후, i) EMA 와 ii) 부하를 모두 고려한 방법을 이후에 설명하기로 한다.

[0028] 여기서, 계산부(320)는, i) EMA 만을 이용해서 질의를 할당할 수도 있다. 즉, 계산부(320)는, 질의 데이터를 q라고 가정하면, q의 중심 포인트를 계산하고, q의 중심 포인트와 가장 가까운 EMA 포인트를 가지는 어플리케이션 서버(400)를 검색할 수 있다. 여기서, 질의 데이터와 EMA 포인트 간의 거리는 유클라디언 거리로 정의될 수 있다. 이때, i) EMA 만을 이용하여 질의를 할당하는 방법을, 도 3을 통하여 설명하기로 한다. 여기서, (a) EMA는 각각의 중심 포인트를 가지고 있고, 각각의 영역을 가지고 있는데, 질의 데이터(Query, q)가 수신되는 경우, EMA 포인트와 질의 데이터 간의 거리가 가장 가까운 EMA를 가지는 어플리케이션 서버(400)로 질의가 할당되게 된다. 즉, (b) $EMA_k$가 가지는 영역은 Bound(k,k+1)이고, $EMA_{k+1}$이 가지는 영역은 Bound(k+1,k)라고 가정하자. 이때, 질의 데이터의 중심 포인트가 $EMA_k$의 영역 내에 존재하는 경우, $EMA_k$와 질의 데이터의 중심 포인트 간의 거리가, $EMA_{k+1}$와 질의 데이터의 중심 포인트 간의 거리보다 짧으므로, 질의 데이터는 $EMA_k$로 할당되게 된다. 이는, i) EMA만을 고려한 결과로서, 이후 i)EMA와 ii)부하를 모두 고려한 결과와 비교될 것이다.

[0029] 다시, 도 2로 돌아와서, 만약, 계산부(320)의 계산 결과로 질의 데이터인 q가 어플리케이션 서버(400)인 i에 할당이 되는 경우에는, i인 어플리케이션 서버(400)의 EMA를 하기 수학식 2를 통하여 계산할 수 있고, i인 어플리케이션 서버(400)의 EMA는 업데이트될 수 있다.

*수학식 2*

[0030] $$EMA_{i^*}=\alpha q+(1-\alpha)EMA_{i^*}$$

[0031] 여기서, EMAi*는 i번째 어플리케이션 서버의 EMA를 의미하고, α는 0 내지 1의 가중치를 의미하고, q는 질의 데이터의 다차원 중심 포인트를 의미한다.

[0032] 한편, i) EMA와 ii) 부하를 모두 고려하기 위해서는, 적어도 하나의 어플리케이션 서버(400)에 각각 할당된 질의 데이터에 대응하는 부하도 파악해야 한다. 따라서, 파악부(430)는, 적어도 하나의 어플리케이션 서버(400)에 할당된 작업량인 부하에 대응하는 질의의 수를 파악할 수 있다.

[0033] 할당부(440)는, 계산된 EMA와 부하에 기반하여, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버(400) 중 어느 하나의 어플리케이션 서버(400)에 할당할 수 있다. 여기서, 할당부(440)는, 하기 수학식 3을 통하여 EMA와 부하를 모두 고려할 수 있다. 즉, 할당부(440)는, EMA와 부하를 곱한 결과를 이용할 수 있다.

*수학식 3*

[0034] $$Bound(i,j):Bound(j,i)=Load(j):Load(i)$$

[0035] 여기서, Bound(i,j)는 i번째 어플리케이션 서버와 j번째 어플리케이션 서버 간의 EMA 거리를 의미하고, Bound(j,i)는 j번째 어플리케이션 서버와 i번째 어플리케이션 서버 간의 EMA 거리를 의미하고, Load(j)는 j번째 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 의미하고, Load(i)는 i번째 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 의미한다.

[0036] 정리하면, 할당부(440)는, 계산된 EMA와 부하를 곱하여 계산된 총 EMA를 적어도 하나의 어플리케이션(400)에 부여하고, 수신된 질의 데이터의 다차원 중심 포인트와, 부여된 총 EMA 간 거리를 계산하고, 계산된 거리가 가장 짧은 어느 하나의 어플리케이션 서버(400)에 질의 데이터를 할당함으로써 수행될 수 있다. 여기서, 다차원 중심 포인트와 EMA 간 거리는, 유클라디언 거리로 정의될 수 있다. 이때, 총 EMA는 갱신된 EMA로 정의한다.

[0037] 할당부(440)에서, i) EMA와 ii) 부하를 모두 고려하여 질의를 할당하는 방법을 도 4를 참조하여 설명한다. 도 4의 (a)를 보면, $EMA_1$은 부하가 15이고, $EMA_2$는 부하가 5이므로, $EMA_1$의 영역과 $EMA_2$의 영역을 5:15로 설정할 수 있다. 여기서, 각각의 EMA를 중심으로 형성된 원은 아폴로니우스 원일 수 있다. 즉, EMA1에는, $EMA_2$보다 이미 부하가 3배나 많이 걸려있으므로, $EMA_1$의 영역을 $EMA_2$의 영역보다 부하의 비에 반비례하도록 설정함으로써, $EMA_1$의 영역 내에 새로운 질의 데이터가 위치할 확률을 줄이는 것이다. 즉, (b) $EMA_k$가 가지는 영역은 Bound(k,k+1)이고, $EMA_{k+1}$이 가지는 영역은 Bound(k+1,k)라고 가정하자. 이때, 각 EMA가 가지는 영역은, 이미 각 EMA에 걸린 부하가 고려되어 있다. 즉, 부하가 30이 걸린 $EMA_k$는 1 만큼의 영역을 가지고 있고, 부하가 10이 걸린 $EMA_{k+1}$은 3 만큼의 영역을 가지고 있다. 이때, 질의 데이터의 중심 포인트가 $EMA_{k+1}$의 영역 내에 존재하는 경우, $EMA_k$와 질의 데이터의 중심 포인트 간의 거리가, $EMA_{k+1}$와 질의 데이터의 중심 포인트 간의 거리보다 짧지만, 부하를 고려한 영역이 $EMA_k$:$EMA_{k+1}$=1:3으로 설정되어 있고, 이미 질의 데이터는 $EMA_{k+1}$의 영역 내에 존재하므로, 새로운 질의 데이터는 $EMA_{k+1}$로 할당되게 된다.

[0038] 도 2로 다시 돌아와서, 업데이트부(450)는, 질의 데이터가 할당된 어플리케이션 서버(400)의 EMA 및 부하에 기초한 최종 EMA를 업데이트할 수 있다. 이때, EMA는 수학식 2 및 수학식 3에 의해 계산될 수 있다. 여기서, 질의 데이터는, 동일한 확률을 가지고 적어도 하나의 어플리케이션 서버(400) 중 어느 하나의 어플리케이션 서버에 할당될 수 있다. 즉, 질의 데이터는, 적어도 하나의 어플리케이션 서버(400)에 이미 할당된 질의 데이터와의 유사도를 고려하여, 적어도 하나의 어플리케이션 서버(400) 중 어느 하나의 어플리케이션 서버에 할당될 수 있다. 이는, 캐시 히트율을 높이기 위함일 수 있다. 그리고, 질의 데이터가 수신되는 경우, 질의 데이터는 질의 할당 서버(400)의 큐에 저장된 가장 오래된 질의 데이터와 대체하여 저장될 수 있다.

[0039] 이와 같은 도 2 내지 도 4의 질의 할당 방법 대해서 설명되지 아니한 사항은 앞서 도 1을 통해 질의 할당 방법에 대하여 설명된 내용과 동일하거나 설명된 내용으로부터 용이하게 유추 가능하므로 이하 설명을 생략하도록 한다.

[0040] 도 5는 도 1에 도시된 질의 할당 서버에서 어플리케이션 서버의 부하를 밸런싱하는 일 실시예를 도시한 도면이다. 도 5를 참조하면, 본 발명의 일 실시예에 따른 질의 할당 방법을, 최소-최대 최적화(min-max optimization) 관점으로 표현한 도면이다. 여기서, 선형 질의 공간과 단일 질의 분산을 가정하고, EMA가 E2인 제 2 어플리케이션 서버(400)는 부하가 최대로 걸려있다고 가정한다. 즉, 제 2 어플리케이션 서버(400)에 걸린

부하 Load(2)가 제 1 어플리케이션 서버(400)에 걸린 부하 Load(1)나, 제 3 어플리케이션 서버(400)에 걸린 부하 Load(3)보다 크다고 가정한다. 그리고, m1은 E1보다는 E2에 가깝고, m2는 E3보다는 E2에 가깝다고 가정한다. 즉, r2<r1이고, r3<r4이므로, r2+r3<r1+r4이고, 제 2 어플리케이션 서버(400)에 질의 데이터가 할당될 확률은, 제 1 어플리케이션 서버(400)에 질의 데이터가 할당될 확률이나, 제 3 어플리케이션 서버(400)에 질의 데이터가 할당될 확률보다 낮게 된다. 따라서, 제 2 어플리케이션 서버(400)에 걸린 부하는 줄어들 수 있다.

[0041] 이와 같은 도 5의 질의 할당 방법에 대해서 설명되지 아니한 사항은 앞서 도 1 내지 도 4를 통해 질의 할당 방법에 대하여 설명된 내용과 동일하거나 설명된 내용으로부터 용이하게 유추 가능하므로 이하 설명을 생략하도록 한다.

[0042] 도 6은 도 1에 도시된 질의 할당 서버에서 EMA 포인트 및 부하에 기반하여 질의를 할당하는 알고리즘의 일 실시예를 도시한다. 본 발명의 일 실시예에 따른 알고리즘은, 도 6의 알고리즘의 한정되는 것은 아니고, 본 발명의 따른 i) EMA 또는 ii) 부하를 고려하여 적어도 하나의 어플리케이션 서버에, 적어도 하나의 클라이언트의 질의 데이터를 분산하여 할당하는 알고리즘이면 어느 것이든 가능하다는 것은 당업자에게 자명하다 할 것이다.

[0043] 이와 같은 도 6의 질의 할당 방법에 대해서 설명되지 아니한 사항은 앞서 도 1 내지 도 5를 통해 질의 할당 방법에 대하여 설명된 내용과 동일하거나 설명된 내용으로부터 용이하게 유추 가능하므로 이하 설명을 생략하도록 한다.

[0044] 도 7 및 도 8은 본 발명의 일 실시예에 따른 질의 할당 방법의 성능 평가 그래프를 도시한 도면이다. 도 7 및 도 8에서는, 본 발명의 일 실시예에 따른 질의 할당 방법의 성능을 측정하기 위하여, 다양한 질의 분포(uniform, normal, Zipf)를 가진 질의 세트를 사용하였다. 여기서, 도 7 및 도 8의 실험은, 41대의 리눅스 클러스터 머신과, 복수의 머신에 대한 성능 평가를 위하여 시뮬레이터를 사용하였다. 여기서, 실험에 사용한 머신은 dual QuadCore Xeon E5506 2.13 GHz CPUs가 장착되었으며, 각 노드당 12 GB의 메모리를 사용하였고, 각 노드는 Gigabit switched Ethernet으로 연결된다.

[0045] 도 7을 참조하면, 어플리케이션 서버의 수가 증가함에 따라, (a) 질의 응답 시간, (b) 캐시 히트율, (c) 로드 밸런싱의 변화를 도시한다. 이때, 캐시 히트율은, 이전 질의 데이터를 처리함으로써 사용되었던 데이터가 재사용되는 확률을 의미한다.

[0046] 여기서, 질의 응답 시간은, 질의 할당 서버가 질의 데이터를 수신받고, 어플리케이션 서버에 질의 데이터가 할당되고, 어플리케이션 서버가 할당된 질의 데이터의 처리를 완료할 때까지의 시간을 의미한다. 여기서, 각각의 어플리케이션 서버는, 100개의 질의 결과를 저장할 수 있는 LRU 캐시를 가지고 있다.

[0047] (a) 어플리케이션 서버가 증가함에 따라, 질의 응답 시간이 줄어드는 것을 알 수 있다. 즉, 복수의 어플리케이션 서버는 더 많은 캐시 공간을 제공할 수 있다. 여기서, 본 발명의 일 실시예에 따른 방법은, BEMA(Balancing EMA)로 표기되는데, 어플리케이션 서버의 수가 늘어날수록 본 발명의 BEMA의 질의 처리 시간이 가장 빠른 것을 알 수 있다. 또한, (b) 본 발명의 BEMA가 어플리케이션 서버의 수가 증가함에 따라 캐시 히트율도 함께 가장 많이 증가하는 것을 알 수 있다. 그리고, (c) 본 발명의 BEMA가 어플리케이션의 수가 적든 많든 간에 가장 좋은 부하 분산율을 가지는 것을 알 수 있다.

[0048] 도 8은 본 발명의 일 실시예에 따른 질의 할당 방법을 설명하기 위한 동작 흐름도이다. 도 8을 참조하면, 질의 할당 서버는, 적어도 하나의 클라이언트로부터 질의 데이터를 수신한다(S7100).

[0049] 그리고 나서, 질의 할당 서버는, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버에 할당하기 위하여, 적어도 하나의 어플리케이션 서버의 EMA(Exponential Moving Average)를 계산한다(S7200).

[0050] 또한, 질의 할당 서버는, 적어도 하나의 어플리케이션 서버에 할당된 작업량인 부하에 대응하는 질의의 수를 파악한다(S7300).

[0051] 마지막으로, 질의 할당 서버는, 계산된 EMA와 부하에 기반하여, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당한다(S7400).

[0052] 도 8을 통하여, 본 발명의 일 실시예에 따른 질의 할당 방법 대해서 설명되지 아니한 사항은 앞서 도 1 내지 도 7을 통해 질의 할당 방법에 대하여 설명된 내용과 동일하거나 설명된 내용으로부터 용이하게 유추 가능하므로 이하 설명을 생략하도록 한다.

[0053] 상술한 단계들(S7100~S7400)간의 순서는 예시일 뿐, 이에 한정되지 않는다. 즉, 상술한 단계들(S7100~S7400)간의 순서는 상호 변동될 수 있으며, 이중 일부 단계들은 동시에 실행되거나 삭제될 수도 있다.

[0054] 도 8을 통해 설명된 일 실시예에 따른 질의 할당 방법은, 컴퓨터에 의해 실행되는 애플리케이션이나 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

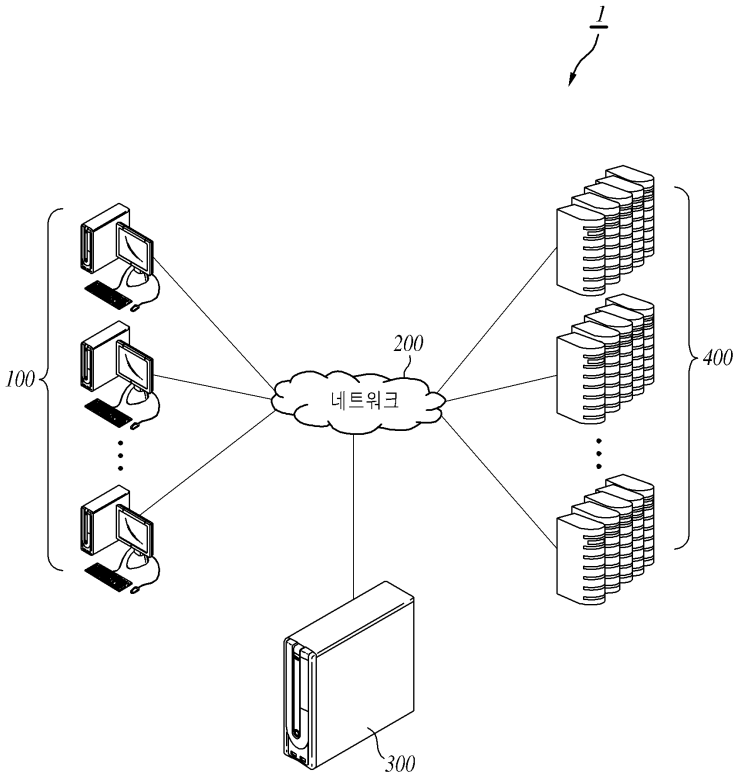
[0055] 전술한 본 발명의 일 실시예에 따른 질의 할당 방법은, 단말기에 기본적으로 설치된 애플리케이션(이는 단말기에 기본적으로 탑재된 플랫폼이나 운영체제 등에 포함된 프로그램을 포함할 수 있음)에 의해 실행될 수 있고, 사용자가 애플리케이션 스토어 서버, 애플리케이션 또는 해당 서비스와 관련된 웹 서버 등의 애플리케이션 제공 서버를 통해 마스터 단말기에 직접 설치한 애플리케이션(즉, 프로그램)에 의해 실행될 수도 있다. 이러한 의미에서, 전술한 본 발명의 일 실시예에 따른 질의 할당 방법은 단말기에 기본적으로 설치되거나 사용자에 의해 직접 설치된 애플리케이션(즉, 프로그램)으로 구현되고 단말기에 등의 컴퓨터로 읽을 수 있는 기록매체에 기록될 수 있다.

[0056] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

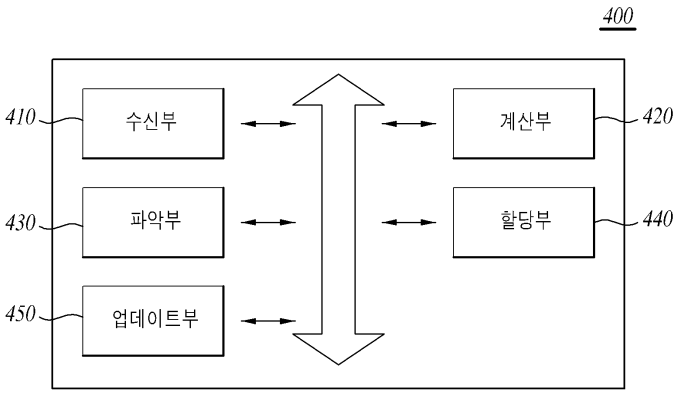
[0057] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

## 도면

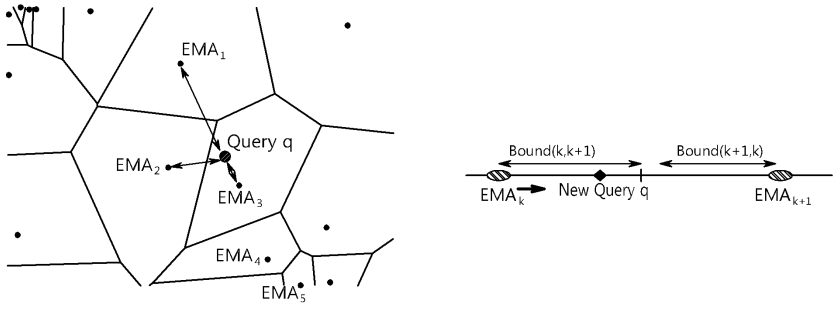
### 도면1


1
100
200
네트워크
400
300

### 도면2

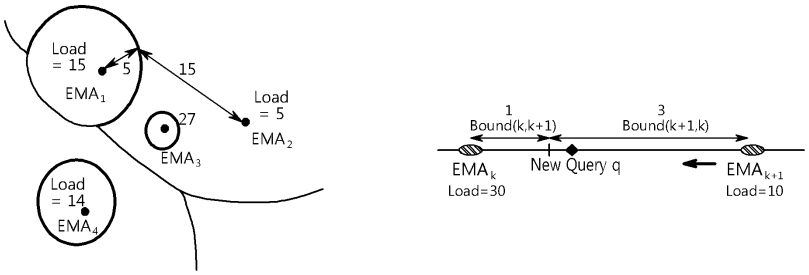

400
410
수신부
420
계산부
430
파악부
440
할당부
450
업데이트부

## 도면3



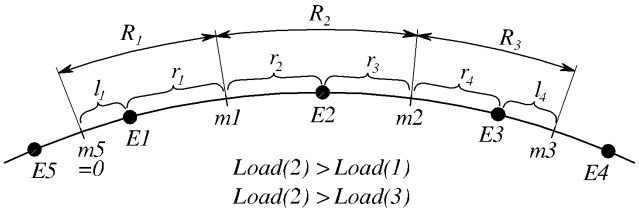

(a) (b)

## 도면4




(a) (b)

## 도면5

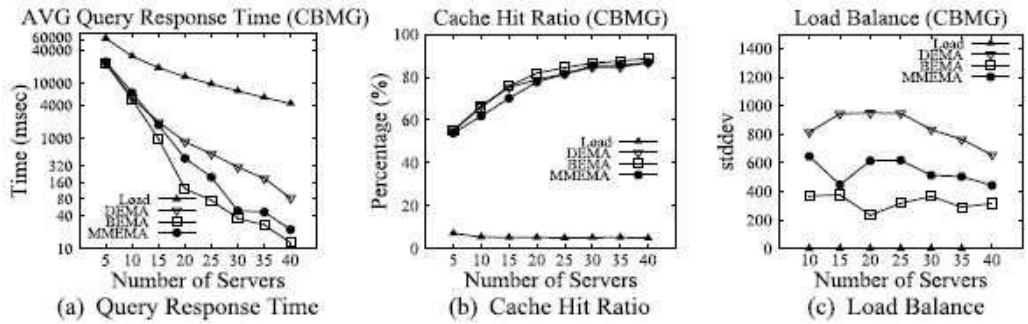



## 도면6

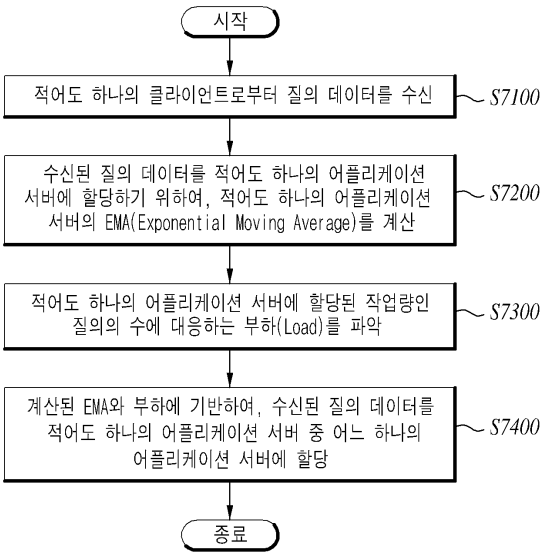
**Algorithm 1** BEMA Algorithm

```
1: INPUT: a client query Q
2: MinDistance ← MaxNum
3: for s = 1 to NumberOfApplicationServers do
4:     if EMA[s] is not initialized then
5:         forward query Q to server s.
6:         EMA[s] ← Q;
7:         return;
8:     else
9:         Load[s] ← GetCurrentLoadOfServer(s);
10:        WeightedDistance ← EuclideanDistance(EMA[s], Q) · Load[s];
11:        if WeightedDistance < MinDistance then
12:            SelectedServer ← s
13:            MinDistance ← WeightedDistance
14:        end if
15:    end if
16: end for
17: forward query Q to SelectedServer.
18: EMA[SelectedServer] ← (α · Q) + (1 − α) · EMA[SelectedServer]
```

도면7


AVG Query Response Time (CBMG)
Time (msec)
Number of Servers
Load
DEMA
BEMA
MMEMA
(a) Query Response Time
Cache Hit Ratio (CBMG)
Percentage (%)
Number of Servers
(b) Cache Hit Ratio
Load Balance (CBMG)
stddev
Number of Servers
(c) Load Balance

도면8


시작
적어도 하나의 클라이언트로부터 질의 데이터를 수신
S7100
수신된 질의 데이터를 적어도 하나의 어플리케이션 서버에 할당하기 위하여, 적어도 하나의 어플리케이션 서버의 EMA(Exponential Moving Average)를 계산
S7200
적어도 하나의 어플리케이션 서버에 할당된 작업량인 질의의 수에 대응하는 부하(Load)를 파악
S7300
계산된 EMA와 부하에 기반하여, 수신된 질의 데이터를 적어도 하나의 어플리케이션 서버 중 어느 하나의 어플리케이션 서버에 할당
S7400
종료