



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년03월02일

(11) 등록번호 10-1495767

(24) 등록일자 2015년02월16일

(51) 국제특허분류(Int. Cl.)

G06F 17/00 (2006.01) G06F 15/16 (2006.01)

(21) 출원번호 10-2013-0127523

(22) 출원일자 2013년10월25일

심사청구일자 2013년10월25일

(56) 선행기술조사문헌

KR101033813 B1

(73) 특허권자

경희대학교 산학협력단

경기도 용인시 기흥구 덕영대로 1732, 국제캠퍼스 내 (서천동, 경희대학교)

한국전자통신연구원

대전광역시 유성구 가정로 218 (가정동)

(72) 발명자

허의남

경기 용인시 기흥구 사은로126번길 46, 316동 1802호 (보라동, 현대모닝사이드1차아파트)

신영록

경기 수원시 영통구 태장로82번길 32, 104동 1006호 (망포동, 동수원엘지빌리지1차)

(뒷면에 계속)

(74) 대리인

특허법인 신지

전체 청구항 수 : 총 10 항

심사관 : 이석형

(54) 발명의 명칭 클라우드 컴퓨팅에서 중계자를 통한 데이터 분산 최적화 장치 및 그 방법

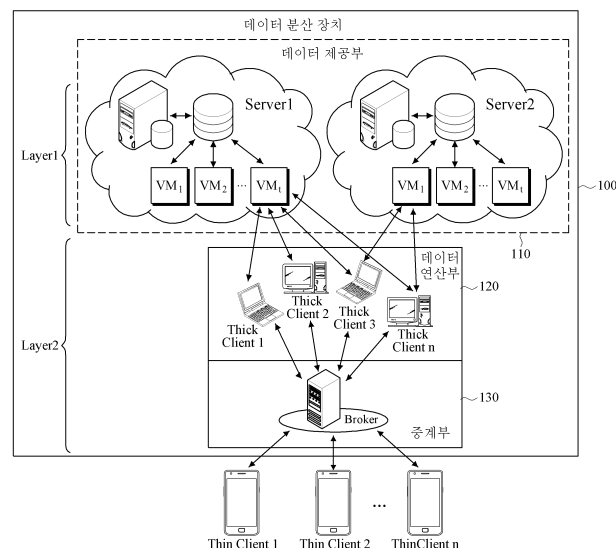
### (57) 요약

클라우드 컴퓨팅에 있어서 중간매개인 중계자를 통해 데이터 분할 및 분산을 최적화하여 효율적인 모바일 컴퓨팅을 지원하기 위한 데이터 분산 최적화 장치 및 그 동작 방법이 개시된다.

데이터 분산 최적화 장치는 데이터 제공부, 데이터 연산부, 중계부, 출력부를 포함 할 수 있다.

이를 통하여 기존의 모바일 컴퓨팅에 클라우드 환경뿐만 아니라 중계부라는 매개를 통하여 연결된 서비스 제공자들의 데이터 분산에 대한 최적의 서비스 조합을 통하여 높은 품질의 모바일 컴퓨팅 퍼포먼스를 낼 수 있다

대표도 - 도1b



(72) 발명자

**엄보윤**

대전 유성구 엑스포로 448, 209동 1006호 (전민동, 엑스포아파트)

**배성준**

대전 유성구 반석서로 109, 705동 1406호 (반석동, 반석마을7단지아파트)

**조철희**

대전 유성구 어은로 57, 105동 606호 (어은동, 한빛아파트)

**이현우**

서울 동작구 여의대방로44길 9, 103동 2206호 (대방동, 성원아파트)

**류원**

서울특별시 강동구 양재대로 1340 둔촌주공아파트 422-604호

이 발명을 지원한 국가연구개발사업

과제고유번호 13-912-03-901

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 정보통신방송기술개발사업(융합미디어원천)

연구과제명 상황인지형 Tele-Screen 시스템 기술개발

기 여 율 0.5/1

주관기관 한국전자통신연구원

연구기간 2012.03.01 ~ 2015.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호 1711001260

부처명 미래창조과학부

연구관리전문기관 한국정보통신산업진흥원

연구사업명 대학IT 연구센터육성 지원사업

연구과제명 실시간 모바일 클라우드 서비스 플랫폼 연구개발

기 여 율 0.5/1

주관기관 경희대학교 산학협력단

연구기간 2013.06.01 ~ 2013.12.31

## 특허청구의 범위

### 청구항 1

클라우드 컴퓨팅에 사용되는 데이터 분산장치에 있어서,

클라우드 서버에 저장된 데이터를, 씩 클라이언트의 대역폭에 따라 각각의 다른 크기의 블록으로 분할하고, 분할한 데이터 블록을 상기 씩 클라이언트에 할당 및 전송하는 데이터 제공부;

상기 데이터 블록을 수신하고, 이를 각각 연산하여 결과 데이터 블록을 생성할 수 있는 병렬로 연결된 복수의 씩 클라이언트를 포함하는 데이터 연산부; 및

소정의 알고리즘을 통해 필요한 상기 씩 클라이언트의 최소 수를 산출하고, 이 수만큼의 상기 씩 클라이언트로부터 연산된 결과 데이터 블록을 수신하고, 이를 결합하여 하나의 완전한 데이터를 생성해서 씩 클라이언트에 전송하는 중계부를 포함하는 것을 특징으로 하는 데이터 분산장치.

### 청구항 2

제1항에 있어서, 상기 데이터 제공부는,

클라우드 서버에 분산하여 처리할 데이터를 저장하여 이를 분산을 위해 공급하는 데이터 공급부;

상기 데이터 공급부로부터 수신한 데이터를 상기 데이터 제공부의 병렬연결된 씩 클라이언트들로 전송하는데 소요되는 시간이 각각 동일하도록, 소정의 알고리즘을 통하여 상기 씩 클라이언트들의 대역폭에 따라 각각 다른 크기의 데이터 블록으로 데이터를 분할하는 데이터 분할부; 및

상기 데이터블록을 각 씩 클라이언트들의 따라 할당 및 전송하는 데이터 할당부를 포함하는 것을 특징으로 하는 데이터 분산장치.

### 청구항 3

제1항에 있어서, 상기 데이터 연산부는,

상기 데이터 제공부에서 분할하여 씩 클라이언트에 할당한 데이터 블록을 각각 연산하여 결과 데이터블록을 생성하고, 이를 상기 중계부로 전송하는 것을 특징으로 하는 데이터 분산장치.

### 청구항 4

제1항에 있어서, 상기 중계부는,

사용자가 설정한 서비스 수준의 응답시간보다 더 짧은 응답시간 및 상기 씩 클라이언트보다 더 높은 연산능력을 가지는 것을 조건으로 하여, 이러한 조건을 충족함에 필요한 상기 필요 씩 클라이언트의 최소 수를 소정의 알고리즘을 통해 산출하는 최소수 산출부; 및

상기 산출된 최소 수만큼의 상기 씩 클라이언트에 결과 데이터 블록의 전송을 요청하고, 이에 따라 수신한 결과 데이터 블록을 하나의 완전한 데이터로 결합하여 상기 씩 클라이언트에 전송하는 데이터 결합부를 포함하는 것을 특징으로 하는 데이터 분산장치.

### 청구항 5

클라우드 서버에 데이터를 저장하는 단계;

상기 저장된 데이터를 씩 클라이언트의 대역폭에 따라 각각의 다른 크기의 블록으로 분할하는 단계;

상기 분할한 데이터 블록을 상기 씩 클라이언트에 할당하는 단계;

할당된 상기 데이터 블록을 수신하고, 이를 각각 연산하여 결과 데이터 블록을 생성하는 단계;

소정의 알고리즘을 통해 필요한 상기 씩 클라이언트의 최소 수를 산출하는 단계;

상기 산출된 결과에 따라 그 수만큼의 상기 씩 클라이언트에 상기 결과 데이터 블록을 요청하고, 이를 수신하여

데이터를 결합하는 단계; 및

상기 결합 된 데이터를 썩 클라이언트에게 전송하는 단계를 포함하는 데이터 분산방법.

#### 청구항 6

제5항에 있어서, 상기 대역폭에 따라 각각의 다른 크기의 블록으로 분할하는 단계는,

분할된 데이터를 병렬 연결된 상기 썩 클라이언트에 전송함에 있어, 전송시간이 각각 동일하도록, 소정의 알고리즘을 통해 상기 썩 클라이언트의 대역폭에 따라 각각 다른 크기의 데이터 블록으로 분할하는 데이터 분산방법.

#### 청구항 7

제5항에 있어서, 상기 썩 클라이언트에 할당하는 단계는,

상기 분할된 데이터 블록을 상기 썩 클라이언트로 전송함에 있어 일정한 전송시간을 유지하도록 각기 다른 크기의 상기 데이터블록을 각 썩 클라이언트들의 대역폭에 맞게 할당하여 전송하는 데이터 분산방법.

#### 청구항 8

제5항에 있어서, 상기 연산하여 결과 데이터 블록을 생성하는 단계는,

상기 분할되어 썩 클라이언트에 할당된 데이터 블록을 각 썩 클라이언트 별로 연산하여 상기 결과 데이터블록을 생성하는 데이터 분산방법.

#### 청구항 9

제5항에 있어서, 상기 썩 클라이언트의 최소 수를 산출하는 단계는,

사용자가 설정한 서비스 수준의 응답시간보다 더 짧은 응답시간 및 상기 썩 클라이언트보다 더 높은 연산능력을 가지는 것을 조건으로 하여, 이러한 조건을 충족 함에 필요한 상기 썩 클라이언트의 최소 수를 소정의 알고리즘을 통해 산출하는 데이터 분산방법.

#### 청구항 10

제5항에 있어서, 상기 결과 데이터 블록을 요청하고, 이를 수신하여 데이터를 결합하는 단계는,

상기 산출된 썩 클라이언트의 최소수의 결과에 따라 그 수만큼의 상기 썩 클라이언트에게 데이터 전송을 요청하고, 이에 따라 연산 된 데이터 블록을 수신하여 하나의 완전한 데이터를 생성하기 위해 결합하는 데이터 분산방법.

### 명세서

#### 기술분야

[0001] 데이터 분산 최적화 장치 및 그 동작 방법에 관한 것으로, 더욱 상세하게는 클라우드 컴퓨팅에 있어서 데이터 분산 최적화 장치 및 그 동작 방법에 관한 것이다.

#### 배경기술

[0002] 최근 스마트폰의 전세계 출하량은 PC의 출하량을 넘어섰으며, 이는 조만간 PC를 대신하여 모바일 장치를 이용한 인터넷 접속이 더 많을 것으로 예측된다. 그러나 이러한 모바일 컴퓨팅은 편리함과 유연성을 제공함에도 불구하고 느린 프로세서, 작은 메모리, 짧은 배터리 수명, 효율이 좋지않은 네트워크 연결과 같은 제한이 존재한다.

[0003] 이러한 제한을 해결하기 위해 클라우드 서버와 스토리지를 이용한 모바일 어플리케이션으로 모바일 장치에 모바일 접속의 이익이나 외부의 장치의 자원을 취득할 수 있는 협업을 제공하는 모바일 클라우드 컴퓨팅 기술이 발전하기 시작했다.

[0004] 그러나 이러한 기술의 발전에도 불구하고 아직까지 데이터 처리 시에 효율을 위한 대역폭차이에 따른 데이터 할

당방법 및 자원의 사용방안에 대한 고려가 많이 이루어지지 않고 있다.

## 발명의 내용

### 해결하려는 과제

[0005] 본 발명은 모바일 장치와 전형적인 컴퓨팅 기기인 노트북(Note book)이나 데스크 탑(Desk top) 컴퓨터 간의 협업을 통하여 더 뛰어난 작업효율의 모바일 컴퓨팅을 지원하는 것에 목적이 있다.

### 과제의 해결 수단

[0006] 본 발명의 실시 예에 따른 클라우드 컴퓨팅에 사용되는 데이터 분산장치는, 클라우드 서버에 저장된 데이터를, 썩 클라이언트의 대역폭에 따라 각각의 다른 크기의 블록으로 분할하고, 분할한 데이터 블록을 상기 썩 클라이언트에 할당 및 전송하는 데이터 제공부; 상기 데이터 블록을 수신하고, 이를 각각 연산하여 결과 데이터 블록을 생성할 수 있는 병렬로 연결된 복수의 썩 클라이언트를 포함하는 데이터 연산부; 및 소정의 알고리즘을 통해 필요한 상기 썩 클라이언트의 최소 수를 산출하고, 이 수만큼의 상기 썩 클라이언트로부터 연산된 결과 데이터 블록을 수신하고, 이를 결합하여 하나의 완전한 데이터를 생성해서 썩 클라이언트에 전송하는 중계부를 포함할 수 있다.

[0007] 그리고 상기 데이터 제공부는, 클라우드 서버에 분산하여 처리할 데이터를 저장하여 이를 분산을 위해 공급하는 데이터 공급부; 상기 데이터 공급부로부터 수신한 데이터를 상기 데이터 제공부의 병렬연결된 썩 클라이언트들로 전송하는데 소요되는 시간이 각각 동일하도록, 소정의 알고리즘을 통하여 상기 썩 클라이언트들의 대역폭에 따라 각각 다른 크기의 데이터 블록으로 데이터를 분할하는 데이터 분할부; 및 상기 데이터블록을 각 썩 클라이언트들의 따라 할당 및 전송하는 데이터 할당부를 포함할 수 있다.

[0008] 또한, 상기 데이터 연산부는, 상기 데이터 제공부에서 분할하여 썩 클라이언트에 할당한 데이터 블록을 각각 연산하여 결과 데이터블록을 생성하고, 이를 상기 중계부로 전송하는 것을 특징으로 하며, 상기 중계부는, 사용자가 설정한 서비스 수준의 응답시간보다 더 짧은 응답시간 및 상기 썩 클라이언트보다 더 높은 연산능력을 가지는 것을 조건으로 하여, 이러한 조건을 충족함에 필요한 상기 필요 썩 클라이언트의 최소 수를 소정의 알고리즘을 통해 산출하는 최소수 산출부; 및 상기 산출된 최소 수만큼의 상기 썩 클라이언트에 결과 데이터 블록의 전송을 요청하고, 이에 따라 수신한 결과 데이터 블록을 하나의 완전한 데이터로 결합하여 상기 썩 클라이언트에 전송하는 데이터 결합부를 포함하는 것을 특징으로 할 수 있다

[0009] 본 발명의 실시 예에 따른 데이터 분산방법은 클라우드 서버에 데이터를 저장하는 단계; 상기 저장된 데이터를 썩 클라이언트의 대역폭에 따라 각각의 다른 크기의 블록으로 분할하는 단계; 상기 분할한 데이터 블록을 상기 썩 클라이언트에 할당하는 단계; 할당된 상기 데이터 블록을 수신하고, 이를 각각 연산하여 결과 데이터 블록을 생성하는 단계; 소정의 알고리즘을 통해 필요한 상기 썩 클라이언트의 최소 수를 산출하는 단계; 상기 산출된 결과에 따라 그 수만큼의 상기 썩 클라이언트에 상기 결과 데이터 블록을 요청하고, 이를 수신하여 데이터를 결합하는 단계; 및 상기 결합된 데이터를 상기 썩 클라이언트에게 전송하는 단계를 포함할 수 있다.

[0010] 여기서, 상기 대역폭에 따라 각각의 다른 크기의 블록으로 분할하는 단계는,

[0011] 분할된 데이터를 병렬로 연결된 상기 썩 클라이언트에 전송함에 있어, 전송시간이 각각 동일하도록, 소정의 알고리즘을 통해 상기 썩 클라이언트의 대역폭에 따라 각각 다른 크기의 데이터 블록으로 분할할 수 있으며, 상기 썩 클라이언트에 할당하는 단계는, 상기 분할된 데이터 블록을 상기 썩 클라이언트로 전송함에 있어 일정한 전송시간을 유지하도록 각기 다른 크기의 상기 데이터블록을 각 썩 클라이언트들의 대역폭에 맞게 할당하여 전송할 수 있다.

[0012] 그리고 상기 연산하여 결과 데이터 블록을 생성하는 단계는, 상기 분할되어 썩 클라이언트에 할당된 데이터 블록을 각 썩 클라이언트 별로 연산하여 상기 결과 데이터블록을 생성할 수 있으며, 상기 썩 클라이언트의 최소 수를 산출하는 단계는, 사용자가 설정한 서비스 수준의 응답시간보다 더 짧은 응답시간 및 상기 썩 클라이언트보다 더 높은 연산능력을 가지는 것을 조건으로 하여, 이러한 조건을 충족 함에 필요한 상기 썩 클라이언트의 최소 수를 소정의 알고리즘을 통해 산출할 수 있다.

[0013] 또한, 상기 결과 데이터 블록을 요청하고, 이를 수신하여 데이터를 결합하는 단계는, 상기 산출된 썩 클라이언트의 최소수의 결과에 따라 그 수만큼의 상기 썩 클라이언트에게 데이터 전송을 요청하고, 이에 따라 연산된

데이터 블록을 수신하여 하나의 완전한 데이터를 생성하기 위해 결합할 수 있다.

### 발명의 효과

[0014] 본 발명의 실시 예에 따르면 모바일 장치가 직접 클라우드 서버에 접속하지 않고 중계자를 통하여 접속하므로 좀 더 빠른 처리속도가 기대되며, 클라우드 서버에서 노트북이나 데스크 탑 같은 로컬 디바이스로 데이터를 블록으로 분할하여 할당하는 것을 통해 높은 서비스 품질(높은 처리능력, 빠른 처리속도)에 도달할 수 있으며, 현재 모바일 클라우드 컴퓨팅에 있어서 보다 나은 자원 활용, 비용절감 및 전반적인 성능개선이 가능할 것으로 기대된다.

### 도면의 간단한 설명

[0015] 도 1a 및 도 1b는 본 발명의 실시 예에 따른 데이터 분산장치의 구성도이다.

더욱 상세하게는 도 1a는 본 발명의 실시 예에 따른 데이터 분산장치의 구성요소만 표시한 구성도이며, 도 1b는 본 발명의 실시 예에 따라 실제로 구현한 데이터 분산장치의 구성도이다.

도 2는 본 발명의 실시 예에 따른 데이터 제공부의 구성도이다.

도 3은 본 발명의 실시 예에 따른 중계부의 구성도이다.

도 4는 본 발명의 실시 예에 따른 필요 씩 클라이언트의 최소 수를 산출하는 알고리즘을 구현한 프로그래밍 코드이다.

도 5는 본 발명의 실시 예에 따른 데이터 분산방법에 대한 흐름도이다

### 발명을 실시하기 위한 구체적인 내용

[0016] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시 예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0017] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성 요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0018] 이하, 도면을 참조하여 본 발명의 실시 예에 따른 클라우드 컴퓨팅에서 중계자를 통한 데이터 분산 최적화 장치 및 그 방법에 대하여 설명한다.

[0019] 도 1a은 본 발명의 실시 예에 따른 데이터 분산장치(100)에 대한 구성도이다.

[0020] 도 1a는 본 발명의 실시 예에 따른 데이터 분산장치(100)의 구성요소만 표시한 구성도이다.

[0021] 도 1a을 참조하면, 데이터 분산장치(100)는 데이터 제공부(110), 데이터 연산부(120), 중계부(130)를 포함할 수 있다.

[0022] 데이터 제공부(110)는 데이터 공급부(111), 데이터 분할부(112), 데이터 할당부(113)을 포함할 수 있다.

[0023] 데이터 제공부(110)는, 클라우드 서버에 저장된 데이터를 데이터 연산부(120) 내 병렬로 연결된 씩 클라이언트들의 대역폭에 따라 분할하여 데이터 블록을 생성하고, 이를 각 씩 클라이언트에 할당하여 전송할 수 있다.

[0024] 데이터 제공부(110)의 데이터 블록의 분할과 할당방법에 대한 자세한 설명은 도 2를 참조하며 자세히 설명하도록 하겠다.

[0025] 데이터 연산부(120)는 복수의 씩 클라이언트를 포함할 수 있으며, 데이터 분석부로부터 할당된 데이터 블록은 전송받아 이를 연산하여 결과 데이터 블록을 생성하고 이를 중계부(130)로 전송할 수 있다.

[0026] 여기서, 씩 클라이언트란 팻 클라이언트(fat client)는 리치 클라이언트(rich client)라고도 하며 스스로 다량의 데이터를 처리하는 클라이언트로 서버에 꼭 의지하지는 않는다. 씩 클라이언트는 개인용 컴퓨터의 한 형태로 개인용 컴퓨터나 노트북 컴퓨터가 가장 흔하게 사용될 수 있다.

- [0027] 중계부(130)는 사용자가 설정한 서비스 수준(QoS, Quality of Service)의 응답시간보다 더 짧은 응답시간 및, 썬 클라이언트보다 더 많은 작업량 (Work Load), 즉 더 높은 연산능력을 가지는 것이라는 두 가지 조건을 충족하면서도, 자원의 낭비방지, 효율적인 자원 활용, 전반적인 성능개선 등을 위하여 소정의 알고리즘을 통해 꼭 필요한 썬 클라이언트의 최소 수를 산출하여, 그 수만큼의 썬 클라이언트들만 사용되도록 요청할 수 있다.
- [0028] 그리하여 이러한 요청을 받은 썬 클라이언트로부터 연산 된 결과 데이터 블록(Result data block)을 수신하여 데이터를 결합할 수 있다.
- [0029] 이때, 썬 클라이언트란, 자신의 계산 역할을 충족시키기 위해 다른 일부 컴퓨터(서버)에 크게 의존하는 컴퓨터를 의미한다. 스스로 역할을 감당하도록 설계된 컴퓨터를 가리키는 전통적인 썬 클라이언트와는 차이가 있고 역할은 데이터 영속성에서부터 실질적인 정보 처리에 이르기까지 다양하다
- [0030] 썬 클라이언트를 사용하는 컴퓨팅은 또한 연산에 필요한 자원 및 부대비용 비용을 낮추면서 높은 효율로 연산 결과물을 얻을 수 있는 하나의 방법이다. 이러한 실시 예에 따르면 최종 사용자에게 그래픽 사용자 인터페이스만을 제공하는 저사양의 휴대용 단말기를 썬 클라이언트로 사용하여 연산에 필요한 자원 및 비용을 감소하게 할 수 있다.
- [0031] 여기서, 중계부(130)는 실시 예에 따르면, 서버 등 데이터를 연산하고 저장할 수 있는 장치가 사용될 수도 있으며, 썬 클라이언트(데스크 탑 또는 노트북) 중의 하나를 중계부(130)로 지정하여 이와 같은 최소 수 산출 및 결과 데이터 결합 등을 수행하게 할 수 있다.
- [0032] 그리고, 중계부(130)의 필요 썬 클라이언트의 최소 수 산출 및 결과 데이터의 결합에 대한 자세한 설명은 도 3을 참조하며 자세히 설명하도록 하겠다.
- [0033] 도 1b는 본 발명의 실시 예에 따라 실제로 구현한 데이터 분산장치의 구성도이다
- [0034] 도 1b는 도 1a에서 설명한 각 부분이 본 발명의 실시 예에 따라 실제로 어떻게 연결되어 데이터가 전달되는지를 보여주기 위한 데이터 분산장치의 구성에 대한 실제 구현모습을 설명하고 있다.
- [0035] 본 발명의 실시 예에 따르면 데이터 제공부(110)의 데이터 공급부(111)는 서버를 통해 실제로 구현하여 운용될 수 있으며, 데이터 분할부(112) 및 데이터 할당부(113)는 서버의 연산능력을 통해 가상 장치(Virtual Machine)를 구현하여 이를 통해 데이터의 분할 및 할당을 수행하게 할 수 있다.
- [0036] 또한, 데이터 연산부(120)는 도 1b에 표시된 바와 같이 데스크 탑, 노트북 등의 장치를 병렬 연결하여 실제로 구현하여 운용할 수 있다.
- [0037] 이 때, 중계부(130)는 중계자를 두어 구현할 수 있는바, 이러한 중계자로는 서버 또는 데이터 연산부(120)와 같은 데스크 탑, 노트북 등의 연산이 가능한 장치로도 실제 구현이 가능하다.
- [0038] 도 2는 본 발명의 실시 예에 따른 데이터 제공부(110)에 대한 구성도이다.
- [0039] 도 2를 참조하면, 데이터 제공부(110)는 데이터 공급부(111), 데이터 분할부(112), 데이터 할당부(113)를 포함할 수 있다.
- [0040] 데이터 공급부(111)는, 클라우드 서비스 제공자로서 서버 등을 통해 데이터를 제공하는 역할을 할 수 있다.
- [0041] 이때, 클라우드 서비스 제공자는 데이터 서버를 이용하여 클라우드 서비스의 인프라를 구축하여, 이로 하여금 저장되어 있는 데이터를 제공할 수 있는 것을 의미한다.
- [0042] 여기서 서버란, 서버 프로그램이 실행되고 있는 컴퓨터 하드웨어를 서버라고 부르며 다른 프로그램으로 서비스를 제공하는 컴퓨터 프로그램을 말하기도 한다. 서버는 프린터 제어나 파일 관리 등 네트워크 전체를 감시·제어하거나, 메인 프레임이나 공동망을 통한 다른 네트워크와의 연결, 데이터·프로그램·파일 같은 소프트웨어 자원이나 모뎀·팩스·프린터 공유, 기타 장비 등 하드웨어 자원을 공유할 수 있도록 도와주는 역할을 한다. 서버는 클라이언트의 요청에 의하여 서비스를 하는데 이와 같이 구성된 시스템을 클라이언트/서버 시스템이라고 하며, 이는 하나 이상의 응용 프로그램을 상호 협력적인 환경에서 운영하는 분산처리 형태를 의미하고 일반 사용자들은 자신들의 컴퓨터(클라이언트)를 이용하여 서버에 접속하고 서버에서 제공하는 정보를 이용하게 된다.
- [0043] 데이터 분할부(112)는 데이터 공급부(111)로부터 전달받은 데이터를 썬 클라이언트의 대역폭에 따라 각각 다른 크기로 분할하여 데이터 블록을 생성할 수 있다.

- [0044] 여기서, 각각 다른 크기로 분할하여 데이터 블록을 생성하는 이유는 대역폭에 따라 데이터를 전달받아, 이 데이터를 병렬화된 썩 클라이언트에 전송하게 되는데, 이때 병목현상으로 인한 데이터 전송시간의 지연을 방지하기 위하여 각 썩 클라이언트들과 일정한 전송시간을 유지할 필요가 존재하기 때문이다.
- [0045] 그러므로 이렇게 일정한 전송속도를 유지하기 위해 데이터 공급부(111)로부터 전달받은 데이터를 썩 클라이언트의 대역폭에 따라 각각 다른 크기로 분할하여 데이터 블록을 생성할 수 있다. 그리고 이렇게 생성한 데이터 블록을 각 썩 클라이언트에 일정한 전송속도를 유지할 수 있도록 매칭(Matching)시켜 할당할 수 있다.
- [0046] 여기서 대역폭이란, 네트워크에서 이용할 수 있는 신호의 최고 주파수와 최저 주파수의 차이를 의미하며 일반적으로는 통신에서 이용 가능한 최대전송속도, 즉 정보를 전송할 수 있는 능력을 뜻하는 경우가 많고, 그 기본 단위로는 bps(bit per seconds)를 이용한다. 따라서 썩 클라이언트의 대역폭에 따라 데이터 블록의 크기를 나누는 것은 전송시간을 일정하게 유지하기 위함이다.
- [0047] 이때, 전송시간을 일정하게 유지되도록 하는 것은 전체적으로 데이터를 처리하는 시간이 늘어나게 되는 결과가 발생하는 것을 방지하기 위함이다. 최종적으로 이렇게 분할한 데이터를 출력하기 위해서는, 분할된 데이터 블록을 중계부(130)에서 다시 결합 하여야 한다.
- [0048] 그런데 어느 한 썩 클라이언트에서 전송시간이 오래 소요되는 경우에 병렬로 연결된 썩 클라이언트로부터 받은 데이터를 결합하기 위해서는 모든 데이터 블록이 다 필요한바, 이로 하여금 가장 오래 소요되는 데이터 블록이 연산 될 때까지 결합이 진행되지 못하므로 병목현상이 발생하게 된다.
- [0049] 그래서 이러한 병목현상으로 인해 데이터를 결합하는 단계에서 시간이 오래 소요되게 되므로, 이를 방지하기 위해 전송시간을 일정하게 유지하는 것이 필요하다.
- [0050] 전송시간을 일정하게 유지하기 위한 소정의 알고리즘은 썩 클라이언트는 대역폭에 따라 최대 전송속도가 달라지므로 전송속도가 빠른 썩 클라이언트에는 크기가 큰 데이터 블록을 할당하고 전송속도가 느린 썩 클라이언트에는 크기가 작은 데이터 블록을 할당하여 전송속도를 일정하게 맞추는 방법을 사용할 수 있다. 이를 공식으로 나타내면 다음과 같다.

### 수학식 1

$$\frac{w(ch_1)}{b_1} = \frac{w(ch_2)}{b_2} = \dots = \frac{w(ch_n)}{b_n}$$

$$Set S = w(block) = \sum_{i=0}^n w(ch_i) = t \sum_{i=0}^n b_i$$

$$Thus, w(ch_i) = t \times b_i = \frac{S}{\sum_{i=0}^n} \times b_i$$

- [0051]
- [0052] 수학식 1과 같은 공식이 성립하기 위해서는, 분할된 각각의 데이터 블록을 {ch1, ch2, ..., chn} 로 정의한다. 이때, 썩 클라이언트의 대역폭에 따라 각 데이터 블록은 따른 다른 크기(size)를 가질 수 있으며, w(chi) 는 데이터 블록의 크기로 정의하고. bi 는 데이터 제공부(110)로부터 데이터를 수신할 때의 썩 클라이언트의 네트워크 대역폭을 의미한다. SET S는 n개의 썩 클라이언트가 처리 할 수 있는 데이터의 총량의 크기(W(block))로 정의할 수 있다. 즉 총 n개의 썩 클라이언트가 t시간동안 처리 할 수 있는 데이터의 크기를 의미한다.
- [0053] 그러므로 데이터 블록을 데이터 제공부(110)로부터 썩 클라이언트로 전달하는데 소요되는 시간은 w(chi)/bi 이다. 병목현상을 방지하기 위하여 병렬연결 된 각 썩 클라이언트로 데이터 블록을 전송하는 시간은 상기 수학식 1의 t로 각 썩 클라이언트 별로 모두 같게 할 수 있다. 이렇게 각 전송시간 t동안 각 썩 클라이언트의 대역폭에 따라 처리할 수 있는 데이터 블록의 크기를 구할 수 있게 되고( $t \times b_i$ ), 이에 따라 데이터 블록을 분할 할 수 있다.
- [0054] 데이터 할당부(113)는 분할된 데이터 블록을 각 썩 클라이언트의 대역폭에 따라 할당할 수 있다.
- [0055] 여기서, 할당이란 분할 된 데이터 블록을 연산을 위하여 썩 클라이언트에 매칭(Matching)하여 전송하는 것을 의미하고. 이때 매칭은 일정한 전송속도를 유지하기 위하여 대역폭과 데이터 블록의 크기를 고려하여 서로 짝을

이루게 하는 것을 의미한다..

[0056] 도 3은 본 발명의 실시 예에 따른 중계부(130)에 대한 구성도이다.

[0057] 도 3을 참조하면, 중계부(130)는 최소수 산출부(131)과 데이터 결합부(132)를 포함할 수 있다.

[0058] 최소수 산출부(131)는 사용자가 설정한 서비스 수준(QoS)에서 요구하고 있는 응답시간보다 더 짧은 응답시간을 가지고, 썩 클라이언트 보다 더 많은 작업량, 즉 연산능력을 가지면서도, 자원의 효율적인 사용을 할 수 있도록 소정의 알고리즘을 통해 썩 필요한 썩 클라이언트의 수를 산출할 수 있다.

[0059] 여기서 서비스 수준이란, 동영상 등의 데이터 전송에 충분한 서비스 수준을 보증하기 위해 만족시켜야 하는 네트워크 서비스 규격으로, 사용자가 원하는 특정 네트워크의 안정된 서비스 보장을 위해 대역폭을 미리 예약하는 네트워크 기술을 뜻한다. 전송하는 우선순위에 따라 네트워크 트래픽을 분류하고 우선순위가 높은 트래픽을 먼저 전송하는 방법이 주로 사용된다. 이 규격에 따라 동영상·영상전화 등의 실시간 프로그램이 네트워크의 대역폭을 가장 효율적으로 사용할 수 있다. 또 일정 수준의 충분한 네트워크 자원을 보증함으로써 공유 네트워크로도 전용 네트워크 수준의 서비스를 제공할 수 있다.

[0060] 여기서 소정의 알고리즘이란 서비스 수준을 만족하는 썩 클라이언트의 최소수(m)를 계산하기 위하여 n개의 썩 클라이언트가 m개의 썩 클라이언트로부터 서비스를 받는다고 가정할 때, 각 썩 클라이언트는 각각의 작업량(workload), 즉 연산능력  $W_m$ 을 가지고 있으며 응답 시간은  $T_m^R$  이라고 정의한다. 이와 같이 썩 클라이언트에서의 연산능력은  $W_n$ , 응답 시간은  $T_n^R$  로 정의한다. 그리고 썩 클라이언트의 연산능력을 사용자가 원하는 서비스 수준(QoS)의 연산능력으로 나눈 서비스 수준 계수를  $a_m$ 으로, 썩 클라이언트에서 처리될 데이터의 사이즈를  $S_m$  정의하고, 노트북, 데스크 탑 컴퓨터와 같이 다양한 형태의 썩 클라이언트를 고려할 때, 필요한 썩 클라이언트의 최소 수를 구하는 방법은 다음과 같이 공식화될 수 있다.

## 수학식 2

$$\begin{aligned} & \text{Minimize } M \\ & \quad a_m \in \{0,1\} \\ & \text{subject to} \\ & \quad \sum_{n \in S_i^m} W_n < \sum_{m=1}^M a_m W_m, \\ & \quad \text{Min} \{T_n^R, \forall n \in S_i^m\} > T_m^R, \forall m. \end{aligned}$$

[0061]

[0062] 이러한 공식이 성립되는 이유는 다양한 썩 클라이언트들이 직접 연산을 처리할 때의 연산능력보다 중계부(130)에 접속하여 썩 클라이언트들을 통하여 연산하는 것의 연산능력이 더 높으면서도, 썩 클라이언트에서 직접 연산을 처리할 때 소요되는 응답시간보다 중계부(130)에 접속하여 썩 클라이언트들을 통하여 연산하는 것의 응답시간이 더 짧아야 한다는 조건( $\text{Min} \{T_n^R, \forall n \in S_i^m\} > T_m^R, \forall m$ )을 충족시키기 위해서이다.

[0063] 도 4는 본 발명의 실시 예에 따른 필요 썩 클라이언트의 최소 수를 산출하는 알고리즘을 구현한 프로그래밍 코드이다.

[0064] 도 4를 참조하면, 수학식 2와 같은 알고리즘을 통해 최소한의 필요 썩 클라이언트의 수를 구하는 것을 실제로 프로그램으로 구현하기 위하여, 도 4와 같은 프로그래밍 코드를 사용할 수 있다. 이러한 프로그래밍 코드는 상기 공식을 구현하기 위한 실시 예에 불과할 뿐 이에 한정되지 않는다.

[0065] 도 5는 본 발명의 실시 예에 따른 데이터 분산방법에 대한 흐름도이다

[0066] 도 5를 참조하면, 우선 클라우드 서버에 데이터를 저장한다(S10).

[0067] 그 후, 저장한 데이터를 데이터 연산부(120)로 전송함에 있어서 데이터 연산부(120) 내에 병렬로 연결된 각 썩 클라이언트가 일정한 전송시간을 가지게 하기 위하여, 소정의 알고리즘을 통해 썩 클라이언트의 대역폭에 따라 데이터를 분할하여 데이터 블록을 생성한다(S20).

[0068] 여기서 일정한 전송시간을 유지할 수 있도록 데이터를 분할하는 방법은, 상기 수학식 1의 공식을 사용하여 산출할 수 있다.

- [0069] 이때, 일정한 전송시간을 유지하는 이유는, 병렬로 연결된 썩 클라이언트들 사이에 전송시간에 차이가 나게 되면 병목현상이 발생하여 가장 늦은 썩 클라이언트의 전송시간으로 전송이 완료되기 때문에 추후 썩 클라이언트에 완전한 데이터를 생성하기 위해 결과데이터 블록을 결합함에 있어서도 데이터 블록을 연산이 늦어져 결과 데이터 블록의 생성이 늦어지면 그만큼 썩 클라이언트에 완전한 데이터를 전송하는데 걸리는 시간이 길어지므로 응답시간이 길어지는 문제가 발생하기 때문이다.
- [0070] 그 후 분할된 각 데이터 블록을 데이터 연산부(120)로 일정한 전송시간을 유지할 수 있게 전송하기 위하여 각 썩 클라이언트의 대역폭에 맞게 할당하여 데이터블록을 전송한다(S30).
- [0071] 여기서 할당이란, 전송시간을 일정하게 유지하게 하기 위하여 썩 클라이언트로의 데이터 전송에 대한 속도를 나타내는 대역폭을 기준으로 하여 전송속도가 빠른 썩 클라이언트에는 용량이 큰 데이터 블록을, 전송속도가 늦은 썩 클라이언트에는 용량이 작은 데이터 블록을 배정하는 것을 의미한다.
- [0072] 전송된 데이터 블록을 데이터 연산부(120) 내의 썩 클라이언트를 통해 연산하여 결과 데이터 블록을 생성한다(S40). 이렇게 생성된 결과 데이터 블록을 중계부(130)로 전송된다.
- [0073] 그 후, 몇 개의 썩 클라이언트들에게 결과 데이터 블록의 전송을 요청해야 할 것 인지에 대한 필요 썩 클라이언트의 최소 수를 산출할 수 있다(S50).
- [0074] 그 이유는, 중계부(130)가 데이터 연산부(120) 내의 썩 클라이언트로부터 결과 데이터 블록을 수신받기 위해서는 데이터 전송 요청을 하여야 하는바, 데이터 연산부(120) 내의 모든 썩 클라이언트에 데이터 전송 요청을 하는 것은 자원의 효율적인 사용 및 비용에 문제가 발생하므로 최소한의 썩 클라이언트에 데이터 전송을 요청하는 것이 합리적이기 때문이다.
- [0075] 그러나 너무 적은 수의 썩 클라이언트에 데이터 전송을 요청하는 경우, 썩 클라이언트가 스스로 데이터를 처리하는 것보다 낮은 연산능력을 가지게 되거나, 더 긴 응답시간이 걸리게 될 수도 있다,
- [0076] 이러한 경우 썩 클라이언트의 연산처리능력의 한계를 해결하기 위한 클라우드 컴퓨팅에 의미가 없어진다고 볼 수 있으므로 적어도 썩 클라이언트를 직접 사용하는 것보다 더 높은 연산능력 및 더 짧은 응답시간을 가질 수 있도록 필요 썩 클라이언트의 최소 수를 산출하는 방법을 사용할 수 있다.
- [0077] 이러한 필요 썩 클라이언트의 최소 수를 산출하는 방법은 상기 수학식 2의 공식에 따라 산출할 수 있고, 도면 4를 참조하여 기재된 프로그래밍 코드를 사용하는 실시 예에 따라 최소 수를 산출하는 소정의 알고리즘을 프로그램으로 구현할 수 있다.
- [0078] 이렇게 산출된 최소 수에 따라 그 수만큼의 썩 클라이언트씩 클라이언트씩 클라이언트송을 요청하고(S60), 이에 따라 전송된 결과 데이터 블록을 썩 클라이언트에 전송하여 사용자에게 결과를 출력할 수 있는 형식의 하나의 완전한 데이터로 결합할 수 있다(S70).
- [0079] 여기서 결합이란, 분할된 데이터 블록을 연산한 결과 데이터 블록을 하나로 통합하고 이를 썩 클라이언트에서 사용가능한 형식으로 가공하는 것을 의미한다.
- [0080] 이렇게 결합 된 데이터는 썩 클라이언트로 전송될 수 있다(S80).
- [0081] 본 발명의 실시 예는 이상에서 설명한 장치 및/또는 방법을 통해서만 구현이 되는 것은 아니며, 이상에서 본 발명의 실시 예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

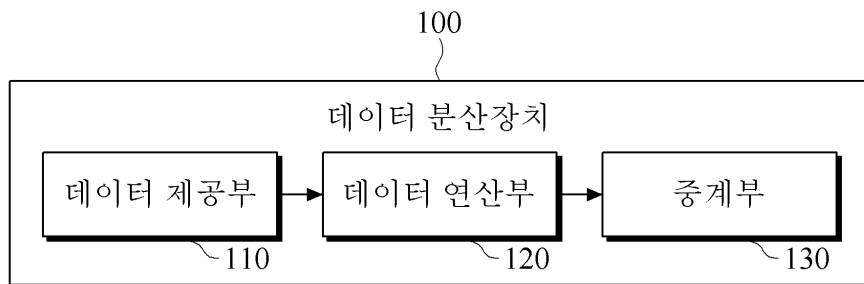
## 부호의 설명

- [0082] 100 : 데이터 분산장치  
110 : 데이터 제공부  
120 : 데이터 연산부  
130 : 중계부

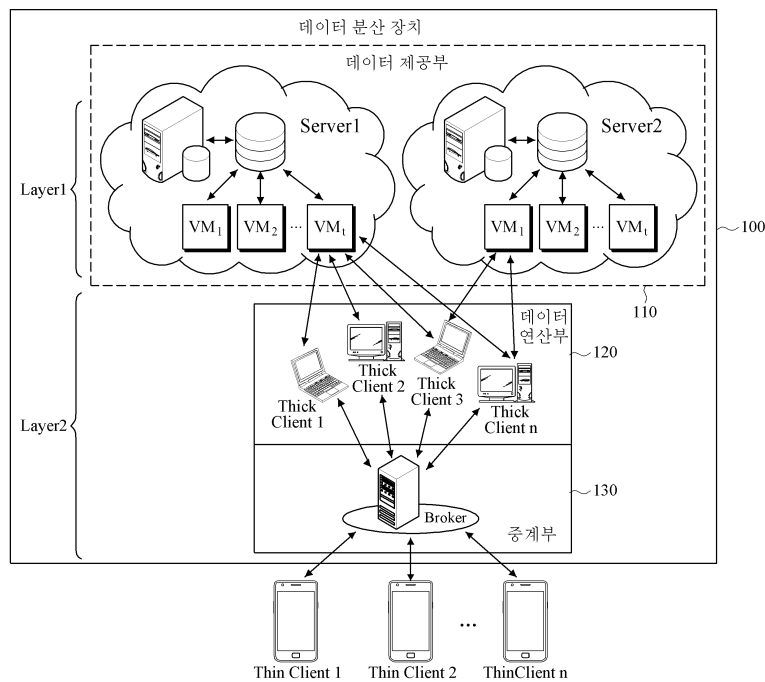
- 111 : 데이터 공급부
- 112 : 데이터 분할부
- 113 : 데이터 할당부
- 131 : 최소수 산출부
- 132 : 데이터 결합부

도면

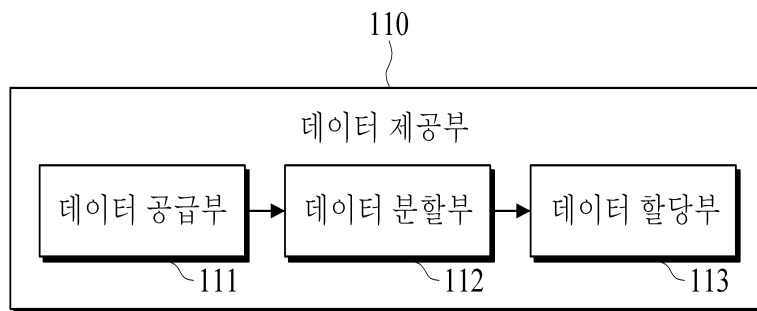
도면1a



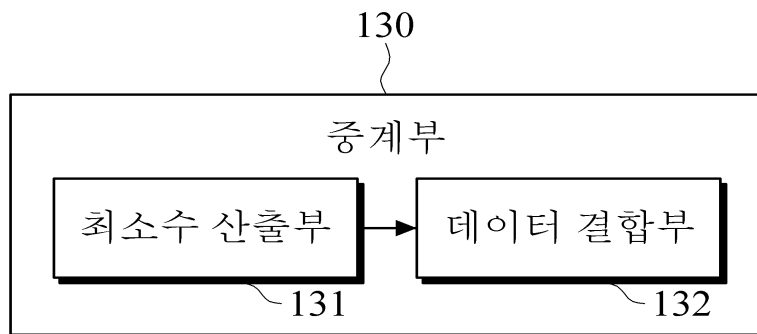
도면1b



도면2



도면3



## 도면4

## QoS를 만족시키기 위한 thick client 최소수 결정 알고리즘

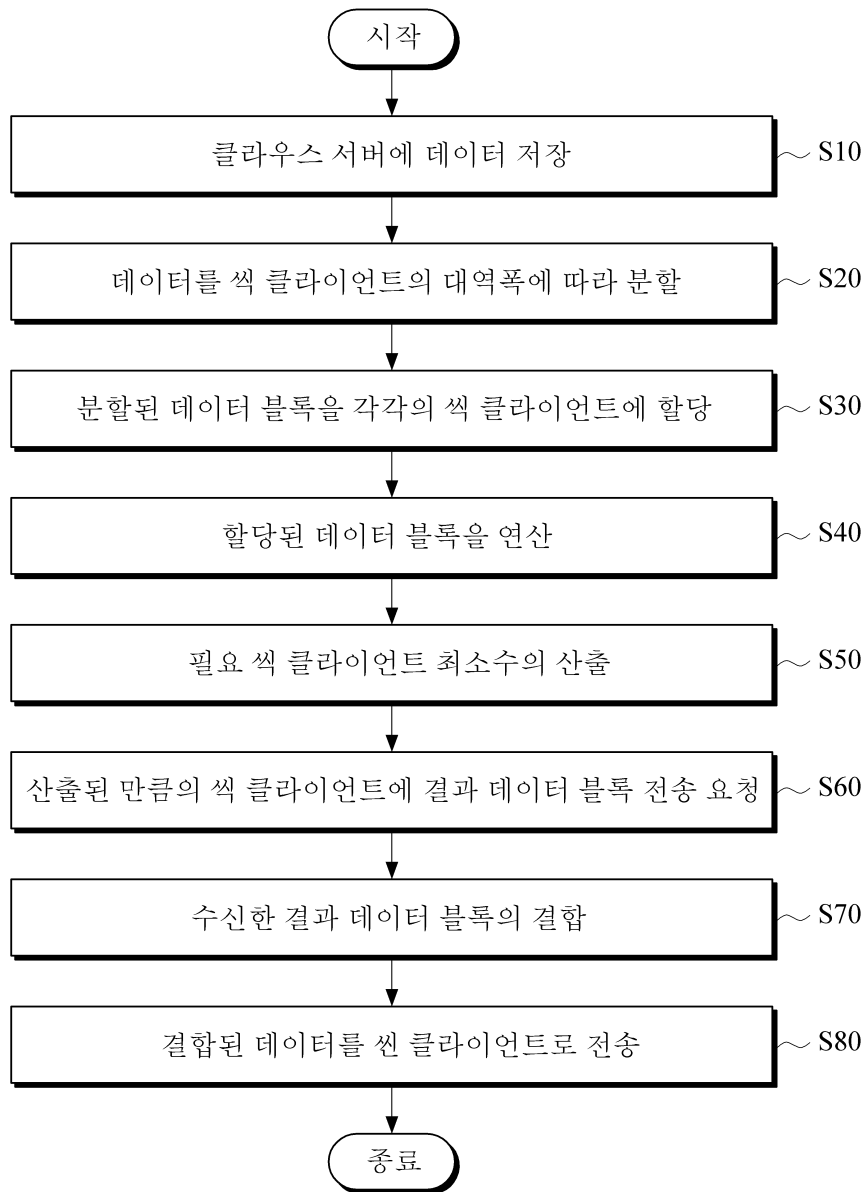
---

```

Input: (Wn, TnR, SIm)
Output: M // QoS를 만족시키는 thick client의 최소수
Function determineNumberThickClient
(Wn, TnR, SIm)
{
    Insert thick clients into a list L1
    Insert thin clients into a list L2
    Sort L1 according to the decreasing order of capacity
    Sort L2 according to the decreasing order of capacity
    for (int i = 0; i <= number of thin clients; i++)
    {
        firstItem = poplist(L1);
        workloadThick = firstItem.workload;
        workloadThin += L2 [i].workload;
        if !(L2[i].responseTime > firstItem.responseTime
            && workloadThick > workloadThin)
        {
            m++;
            workloadThin = L2 [i].workload;
        }
    }
    return m;
}

```

도면5



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 5

【변경전】

상기 썩 클라이언트

【변경후】

썩 클라이언트