

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2022년 4월 14일 (14.04.2022) WIPO | PCT



(10) 국제공개번호
WO 2022/075528 A1

- (51) 국제특허분류:
G06F 9/50 (2006.01) H04L 29/08 (2006.01)
- (21) 국제출원번호: PCT/KR2020/017849
- (22) 국제출원일: 2020년 12월 8일 (08.12.2020)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
10-2020-0129920 2020년 10월 8일 (08.10.2020) KR
- (71) 출원인: 인하대학교 산학협력단 (INHA-INDUSTRY PARTNERSHIP INSTITUTE) [KR/KR]; 22212 인천시 미추홀구 인하로 100, Incheon (KR).
- (72) 발명자: 송민석 (SONG, Minseok); 22212 인천시 미추홀구 인하로 100, Incheon (KR).
- (74) 대리인: 양성보 (YANG, Sungbo); 06099 서울시 강남구 선릉로125길 14 삼성빌딩 2층 피앤티특허법률사무소, Seoul (KR).
- (81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU,

ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

- (84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

(54) Title: TASK OFFLOADING FOR MAXIMIZING REWARDS UNDER EDGE SERVER LIMIT POWER IN MOBILE EDGE COMPUTING

(54) 발명의 명칭: 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩

Algorithm 1: Maximum allowable utilization determination(MUD) algorithm

```
Output:  $Y_j$ , ( $j = 1, \dots, N^{ES}$ );
1 Boolean variable: flag  $\leftarrow$  TRUE;
2 Temporary variables:  $P_{ij}^{algo}$  and  $V_{ij}^{algo}$ , ( $i = 1, \dots, N^{task}$ ), ( $j = 1, \dots, N^{ES}$ );
3 Temporary variable:  $P \leftarrow 0$ ;
4 Temporary variable:  $\forall i, X_i^{imp} \leftarrow 0$ ;
5 Set of task vertexes:  $S^{task} = \{\tau_1, \dots, \tau_{N^{task}}\}$ ;
6 while flag do
7   for  $\forall \tau_i \in S^{task}$  do
8     Calculate  $P_{ij}^{algo}$  and  $V_{ij}^{algo}$  using Equations (5) and (6);
9   end
10  Find the highest value of  $V_{ij}^{algo}$ , (say,  $V_{H,M}^{algo}$ );
11  if  $P^{used} + P_{H,M}^{algo} \leq P^{limit}$  then
12     $P^{used} \leftarrow P^{used} + P_{H,M}^{algo}$ ;
13     $X_H^{imp} \leftarrow M$ ;
14     $S^{task} \leftarrow S^{task} - \{\tau_H\}$ ;
15  else
16    flag  $\leftarrow$  FALSE;
17  end
18  if  $S^{task} = \emptyset$  then
19    flag  $\leftarrow$  FALSE;
20  end
21 end
22 for  $j = 1$  to  $N^{ES}$  do
23    $Y_j \leftarrow \sum v_{ix_j^{imp}} = \sum U_i$ ;
24 end
```

(57) Abstract: Disclosed is task offloading for maximizing rewards under edge server limit power in mobile edge computing. A task offloading method performed by a task offloading system implemented by a computer, according to one embodiment, may comprise the steps of: determining an edge server to execute an offloaded task on the basis of power information of each edge server; and offloading a task according to assignment information set for the determined edge server.

(57) 요약서: 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩이 개시된다. 일 실시예에 따른 컴퓨터로 구현되는 작업 오프로딩 시스템에 의해 수행되는 작업 오프로딩 방법은, 각각의 엣지 서버의 전력 정보에 기초하여 오프로드된 작업을 실행할 엣지 서버를 결정하는 단계; 및 상기 결정된 엣지 서버에 설정된 할당 정보에 따라 작업을 오프로딩하는 단계를 포함할 수 있다.

WO 2022/075528 A1

명세서

발명의 명칭: 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩

기술분야

- [1] 아래의 설명은 모바일 엣지 컴퓨팅에서 엣지 서버에게 작업을 오프로딩 하는 기술에 관한 것이다.

[2]

배경기술

- [3] 네트워크 및 시스템 기술의 향상은 무선 네트워크에 의존하는 애플리케이션의 실현 가능성을 증가시켰다. 센서, 스마트 개인용 기기, 운전자 없는 차량 등 연결된 사물인터넷(IoT) 기기를 기반으로 한 증강현실 및 스마트 교통 관제 시스템이 대표적이다. 스마트 IoT 디바이스는 빠르게 증가하고 있으며, 2030년까지는 스마트시티 애플리케이션에서 5000억대의 스마트 IoT 디바이스가 필요할 것으로 예측된다. 이러한 애플리케이션은 스마트 IoT 디바이스에 의해 생성될 방대한 양의 데이터를 처리해야 할 것이다.
- [4] 일부 IoT 디바이스는 빠른 연산이 필요한 작업을 수행해야 한다. 또한 정기적으로 처리해야 하는 센서 데이터의 정기적인 수집을 수반한다. 컴퓨팅 기능이 제한되면 이러한 작업을 클라우드 서버로 오프로드해야 할 경우가 있다. 그러나 클라우드에서 중앙 서버를 중심으로 코어 네트워크가 부과하는 지연은 클라우드 기반 접근법이 적절하지 않을 수 있음을 의미한다. 신속한 처리가 필요한 IoT 기반 시스템을 위해서 엣지 컴퓨팅 서비스를 통해 문제를 해결한다. 클라우드 가장자리에 위치한 엣지 서버(ES)는 빠른 처리와 더불어 중앙 서버의 부하 감소 효과를 산출하므로, 엣지 컴퓨팅에 대한 수요는 빠르게 증가할 것으로 예상된다. 엣지 컴퓨팅 서비스는 각 기지국(BS)에 엣지 서버를 배치하거나 엣지 서버를 여러 기지국에 연결하여 구현할 수 있다. 이를 모바일 엣지 컴퓨팅(MEC)이라고 한다. 이러한 기지국의 범위 내에 위치한 IoT 장치는 기지국에서 엣지 서버로 작업을 오프로드하거나 액세스할 수 있다.
- [5] EIP(Edge Infrastructure Providers)는 고객에게 엣지 컴퓨팅 서비스를 제공하기 시작했다. EIP는 클라우드 가장자리에 엣지 서버를 설치하고 이를 사용하여 고객에게 컴퓨팅 성능을 제공한다. 고객 간에 엣지 서버 공유 자원 활용을 개선하고 EIP에 대한 재정적 수익을 얻는다. EIP가 고객과 맺은 계약상 약정에 따라 서버에서 운영하는 우선순위가 결정된다. EIP에 대한 총 보상의 최대화라는 맥락 안에서 어떤 작업이 오프로드 되는지를 결정할 필요가 있다.
- [6] 엣지 서버는 원격지 마이크로 데이터센터로서 전력 소비량은 10kW급이 될 것으로 예상된다. 그러한 서버 수천 대가 EIP의 주요 관심사인 메가와트를 소비하게 되므로, 전력 제한이 반드시 필요하다. 엣지 서버에서 끌어온 전력이

사전 정의된 전력 제한치 미만을 유지한다. 서버가 소비하는 전력은 CPU 활용률에 따라 달라진다. 이는 서버의 작업량에 따라 결정되며, 엣지 서버가 수용하는 오프로드된 작업의 수에 따라 결정되는 것을 의미한다.

- [7] 이에 따라 작업 오프로딩 기술을 사용하여 모바일 기기에 의해 사용되는 에너지를 줄이기 위한 기술이 요구된다.

[8]

발명의 상세한 설명

기술적 과제

- [9] 엣지 서버에 의해 소비되는 전력을 제한하면서 발생한 이익과 관련된 보상을 최대화하는 방법 및 시스템을 제공할 수 있다. 상세하게는, 엣지 서버의 전력 특성을 검토하여 최적화 문제를 공식화하고, 각 오프로드된 작업을 실행할 엣지 서버를 결정함에 따라 설정된 각 엣지 서버의 최대 허용 CPU 활용률에 기초하여 각 엣지 서버 최대 허용 CPU 활용률과 관련된 전력 소비량을 고려하여 보상을 예측하고, 최소 비용 최대 흐름(MCMF) 그래프를 사용하여 작업 할당을 수행하는 방법 및 시스템을 제공할 수 있다.

[10]

기술적 해결방법

- [11] 컴퓨터로 구현되는 작업 오프로딩 시스템에 의해 수행되는 작업 오프로딩 방법은, 각각의 엣지 서버의 전력 정보에 기초하여 오프로드된 작업을 실행할 엣지 서버를 결정하는 단계; 및 상기 결정된 엣지 서버에 설정된 할당 정보에 따라 작업을 오프로딩하는 단계를 포함할 수 있다.
- [12] 상기 작업을 오프로딩하는 단계는, 상기 각각의 엣지 서버의 한도 전력 하에서 엣지 서버 제공자의 보상을 최대화하기 위한 상기 각각의 엣지 서버의 최대 허용 가능한 사용률과 작업 할당을 포함하는 할당 정보를 결정하는 단계를 포함할 수 있다.
- [13] 상기 작업을 오프로딩하는 단계는, 상기 각각의 엣지 서버의 최대 허용 가능한 사용률을 결정하는 동시에 전력 요구량에 대한 엣지 서버 제공자의 보상 비율이 높은 작업의 오프로드에 높은 우선 순위를 부여하는 단계를 포함할 수 있다.
- [14] 상기 작업을 오프로딩하는 단계는, 상기 각각의 엣지 서버의 전력 제약조건을 충족하면서 최대 허용 가능한 사용률의 값들을 탐색하는 단계를 포함할 수 있다.
- [15] 상기 작업을 오프로딩하는 단계는, 작업 할당으로부터 전력 증가량과 보상 증가량을 반복적으로 계산하여 전력 제약조건이 충족되고, 각각의 엣지 서버에서 최대 허용 가능한 사용률이 결정될 때까지 전력 비율에 더 높은 보상을 주는 최대 허용 가능한 사용률 값에 우선순위를 부여하는 단계를 포함할 수 있다.
- [16] 상기 작업을 오프로딩하는 단계는, 상기 각각의 엣지 서버로 작업 할당하기 위한 각각의 작업에 대한 값을 탐색하는 단계를 포함할 수 있다.

- [17] 상기 작업을 오프로딩하는 단계는, 최소 비용 최대 흐름 그래프를 사용하여 작업 분할을 허용하는 방법 또는 작업 분할을 허용하지 않는 방법 중 어느 하나의 방법을 이용하여 엣지 서버로의 작업 할당을 수행하는 단계를 포함할 수 있다.
- [18] 상기 작업 분할을 허용하는 방법은, 잔차 그래프가 상기 최소 비용 최대 흐름 그래프에서 생성되고 각 노드 사이의 흐름을 표현하기 위한 이진 함수의 값과 비용을 초기화하는 과정, 최단 경로 빠른 알고리즘을 사용하여 최소 비용 흐름을 도출함에 따라 상기 잔차 그래프에 경로를 생성하고, 상기 흐름을 갱신하는 경로 증강 과정 및 정점 사이에 흐름이 존재할 경우, 작업을 구성하는 하위 작업 중 하나를 처리하기 위해 엣지 서버를 선정함으로써 상기 하위 작업이 할당된 엣지 서버의 지수값을 갱신하는 마무리 과정을 수행할 수 있다.
- [19] 상기 작업 분할을 허용하지 않는 방법은, 상기 최소 비용 최대 흐름 그래프에 작동되고, 소스 정점에서 싱크 정점으로 최소 비용 흐름을 계산하는 최소 비용 흐름 과정, 상기 최소 비용 흐름과 연관된 작업 정점을 제거하고 용량 사용량과 작업이 할당된 엣지 서버의 지수값을 갱신하고 흐름과 연관된 엣지들을 제거하는 과정을 수행할 수 있다.
- [20] 컴퓨터로 구현되는 작업 오프로딩 시스템은, 각각의 엣지 서버의 전력 정보에 기초하여 오프로드된 작업을 실행할 엣지 서버를 결정하는 서버 결정부; 및 상기 결정된 엣지 서버에 설정된 할당 정보에 따라 작업을 오프로딩하는 작업 오프로딩부를 포함할 수 있다.

[21]

발명의 효과

- [22] 본 발명은 각 서버의 최대 활용률을 결정하는 동시에 전력 요구량에 대한 엣지 서버 제공자의 보상 비율이 높은 작업의 오프로드에 높은 우선 순위를 두고, 최소 비용 최대 흐름 그래프를 사용하여 작업 할당 문제를 모델링함에 따라 가장 높은 보상을 검색하여 서버 용량의 영향을 받는 작업을 오프로드함으로써 동일한 전력 제약 조건 하에서 대체 계획보다 더 높은 보상을 달성할 수 있다.

[23]

도면의 간단한 설명

- [24] 도 1은 일 실시예에 따른 모바일 엣지 컴퓨팅 구조를 설명하기 위한 도면이다.
- [25] 도 2는 일 실시예에 따른 작업 오프로딩 시스템의 구성을 설명하기 위한 블록도이다.
- [26] 도 3은 일 실시예에 따른 작업 오프로딩 시스템의 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩 방법을 설명하기 위한 흐름도이다.
- [27] 도 4는 일 실시예에 따른 작업 오프로딩 시스템에서 최대 허용 사용률 결정(maximum utilization determination: MUD) 알고리즘에 대한 의사 코드를

나타낸 도면이다.

[28] 도 5는 일 실시예에 따른 작업 오프로딩 시스템에서 엣지 서버 할당 문제를 흐름 네트워크로 모델링하는 것을 설명하기 위한 예이다.

[29] 도 6은 일 실시예에 따른 작업 오프로딩 시스템에서 사이의 흐름을 표현하기 위한 잔차 그래프의 예이다.

[30] 도 7은 일 실시예에 따른 작업 오프로딩 시스템에서 작업 분할이 허용된 경우의 엣지 서버 할당 알고리즘에 대한 의사 코드를 나타낸 도면이다.

[31] 도 8은 일 실시예에 따른 작업 오프로딩 시스템에서 작업 할당 없는 엣지 할당 알고리즘에 대한 의사 코드를 나타낸 도면이다.

[32] 도 9 내지 도 14는 일 실시예에 따른 작업 오프로딩 시스템의 실험 결과를 설명하기 위한 그래프이다.

[33]

발명의 실시를 위한 최선의 형태

[34] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[35]

[36] 모바일 엣지 컴퓨팅에서 작업은 모바일 장치에서 네트워크 가장자리에 있는 서버로 오프로드됨으로써 중앙 서버에 도달하는 데 필요한 대기 시간을 발생시키지 않고 작업 처리 속도를 높인다. 단, 엣지 서버에 의해 사용되는 힘은 상당하며 비용이 효율적일 필요가 있다. 실시예에서는 제한된 전력 소모량, 서버 처리 용량, 무선 네트워크 커버리지 내에서 엣지 서버 제공자의 보상을 최대화할 목적으로 작업을 서버로 오프로드하는 방안을 설명하기로 한다.

[37] 도 1은 일 실시예에 따른 모바일 엣지 컴퓨팅 구조를 설명하기 위한 도면이다.

[38] 엣지 서버들은 기지국들의 커버리지 안에 있는 IoT 장비들에서 오프로드된 작업들을 처리하기 위해 적어도 하나 이상의 기지국으로 할당될 수 있다. 만약 한 작업이 엣지 서버에서 EIP('엣지 서버 제공자'로 기재하기로 함)로 오프로드 되면, 그 엣지 서버 제공자는 엣지 서버에 의해 수행된 계산량에 기반하여 보상을 받을 수 있다. 그렇지 않으면, 오프로드된 작업들은 클라우드의 중앙 서버로 넘겨진다.

[39] 한편, SPEC 전력위원회는 엣지 서버들의 에너지 소비량을 측정하는 도구들과 표준들을 제공하고 있다. 2006년부터 엣지 서버들을 위한 에너지 측정값들을 수집했고 에너지 소비와 CPU사용률 사이의 관계에 대해 분석하였으며, 이는 서버 특성에 의해 강력히 좌우되는 것으로 밝혀졌다.

[40] 감시, 공기질 모니터링 및 교통량 모니터링과 같은 많은 IoT 애플리케이션들은 주기적으로 처리되어야 하는 센서 데이터의 주기적인 수집이 필요하다. 작업 오프로딩을 위해, 각각의 작업은 일정시간 동안 요구되는 계산량을 계산하고, 엣지 서버에서 제공하는 프로세서 용량의 일정 비율을 받을 수 있다. 작업 사용량은 작업에 필요한 계산량이 분자로, 일정시간 동안 엣지 서버에 의해

제공받은 총 계산량이 분모로 표현될 수 있다. 그 후, 각각의 작업은 각 주기 동안 할당된 프로세서 용량을 사용할 수 있다.

- [41] 다른 엣지 서버들이 다른 처리용량을 가질 수 있기 때문에, 각 작업의 사용량을 최고 처리용량을 갖는 엣지 서버에 대하여 정규화할 수 있다. 만약, NES가 엣지 서버의 수라면, $C_j^{\max}$ ($i=1, \dots, \text{NES}$) 를 I^{high} 의 지수를 가진 최고 처리용량을 갖는

엣지 서버와 비교하여 최대 처리용량을 나타내기 위해 사용할 수 있다. 그 후, 작업 τ_i 의 정규화 처리된 U_i ($i=1, \dots, N^{\text{task}}$)는 엣지 서버 I^{high} 에서 실행된다는 사실을 기반으로 결정될 수 있다. 여기서, N^{task} 는 총 작업 수를 나타낸다. 쉽게 설명하기 위하여, U_i 와 $C_j^{\max}$ 가 정수 값을 갖는다고 가정하자. 이를 위해,

그레인 크기를 최소 할당 단위로 도입하고 $C_j^{\max}$ 와 U_i 를 그레인의 배수로

표현할 수 있다. 예를 들면, 만약, 그레인의 크기가 0.01이면, $C_j^{\max} = 100$ 이 된다.

- [42] X_i , ($X_i = 0, \dots, N^{\text{ES}}$)를 작업 τ_i ($i=1, \dots, N^{\text{task}}$)가 할당된 엣지 서버의 지수라고 하자. 어떠한 엣지 서버로도 할당되지 못하는 작업 τ_i 을 처리하는 클라우드의 중앙 서버를 나타내기 위해 지수 0을 도입할 수 있다. 요약하자면, 만약 $X_i = 0$ 이면, 작업 τ_i 는 클라우드 중앙 서버에 의해 처리될 수 있다. 반면에, 만약 $X_i = j$ 이면, 작업 τ_i 는 엣지 서버 j 에 의해 처리될 수 있다. 모든 작업에 대하여 합쳐진 총 사용량은 처리용량을 초과할 수 없으므로 $\sum_{i, X_i=j} U_i \leq C_j^{\max}$ 가 된다.

이에 따라, 엣지 서버에서의 총 CPU 사용률 U_j^{total} 은 다음과 같이 표현될 수 있다.

- [43] 수학적 식 1:

[44]
$$U_j^{\text{total}} = \frac{\sum_{i, X_i=j} U_i}{C_j^{\max}}$$

- [45] 각 엣지 서버에 의해 소비되는 전력량을 제한하기 위하여, CPU 사용률을 제한할 수 있다. 이를 위해, 각 엣지 서버 j 에 대해 최대 허용 가능한 CPU 사용률 Y_j 를 도입할 수 있다. 여기에서, CPU 사용률 Y_j 는 0과 $C_j^{\max}$ 사이의 정수이다.

이때, 각 엣지 서버 j 에서 총 CPU 사용률 Y_j 를 초과해서는 안된다. 만약, CPU 사용률 Y_j 가 0이면, 각 엣지 서버 j 는 어떠한 작업도 처리할 수 없다. 만약, $Y_j = C_j^{\max}$ 라면, 각 엣지 서버 j 의 처리용량을 완전히 사용할 수 있다.

- [46] 활성 전력은 엣지 서버가 연산 작업을 진행하고 있는 동안 소비된 전력이고, 이는 CPU 사용률에 따라 비선형적으로 달라진다. 연산이 진행되지 않을 경우에도, 엣지 서버는 고정된 유휴 전력을 사용한다. 이에 따라, 각 엣지 서버 j 에서 사용률이 x ($0 \leq x \leq 1$)일 때, 활성 전력 계수를 표현하기 위해 $\alpha_{j,x}$ 를 사용하고

각 엣지 서버 j 의 유틸리티 계수를 표현하기 위해 β_j 를 사용할 수 있다. 이에, 각 엣지 서버 j 에 의해 도출된 전력 $P_j(x)$ 를 사용률 x 에 따라 다음과 같이 정의될 수 있다.

[47] 수학적식 2:

$$[48] \quad P_j(x) = \alpha_{j,x} x + \beta_j (1 - x)$$

[49] 도 2는 일 실시예에 따른 작업 오프로딩 시스템의 구성을 설명하기 위한 블록도이고, 도 3은 일 실시예에 따른 작업 오프로딩 시스템의 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩 방법을 설명하기 위한 흐름도이다.

[50] 작업 오프로딩 시스템(100)의 프로세서는 서버 결정부(210) 및 작업 오프로딩부(220)를 포함할 수 있다. 이러한 프로세서의 구성요소들은 작업 오프로딩 시스템에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및 프로세서의 구성요소들은 도 3의 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩 방법이 포함하는 단계들(310 내지 320)을 수행하도록 작업 오프로딩 시스템을 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

[51] 프로세서는 모바일 엣지 컴퓨팅에서 엣지 서버 한도 전력 하에서 보상을 최대화하는 작업 오프로딩 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 작업 오프로딩 시스템(100)에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 작업 오프로딩 시스템(100)을 제어할 수 있다. 이때, 프로세서 및 프로세서가 포함하는 서버 결정부(210) 및 작업 오프로딩부(220) 각각은 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(310 내지 320)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.

[52] 서버 결정부(210)는 각각의 엣지 서버의 전력 정보에 기초하여 오프로드된 작업을 실행할 엣지 서버를 결정할 수 있고(310), 작업 오프로딩부(220)는 결정된 엣지 서버에 설정된 할당 정보에 따라 작업을 오프로딩할 수 있다(320).

[53] 실시예에서는 다음에서 설명하는 3가지 제약조건 하에서 전체 보상을 극대화하기 위하여 적절한 엣지 서버로 각각의 작업을 할당할 수 있다.

[54] 엣지 서버 처리용량: 각각의 엣지 서버에서의 총 사용합계는 최대 허용 사용량은 넘어서는 안된다.

[55] 수학식 3:

$$[56] \quad \forall j, U_j^{\text{total}} \leq Y_j$$

[57] 엡지 서버 커버리지: 엡지 서버는 엡지 서버 영역 내에 존재하는 기기들에서 전송되는 작업만 수신할 수 있다. 그 작업이 엡지 서버에 의해 받아들여질 수 있는지 여부를 나타내는 이진 상수 H_{ij} 를 도입할 수 있다. 만약, $H_{ij}=1$ 이라면, 각 엡지 서버 j 는 작업 τ_i 를 오프로드 하는 요청을 받아들일 수 있다. 만약, $H_{ij}=0$ 이라면, 각 엡지 서버 j 는 작업 τ_i 를 오프로드 하는 요청을 받아들일 수 없다. 이에 따라, 작업 τ_i 은 $H_{ij}=1$ 일 경우에만, 엡지 서버 X 로 할당될 수 있다.

[58] 전력: P^{limit} 는 사용이 허용되는 모든 엡지 서버의 최대 총 전력이다.

[59] 수학식 4:

$$[60] \quad \sum_{j=1}^{N^{\text{ES}}} P_j(U_j^{\text{total}}) \leq P^{\text{limit}}$$

[61] 이에, 각 엡지 서버에서 작업 τ_i 을 처리할 때, 엡지 서버 제공자에 의해 받은 보상인 R_{ij} 를 도입하기로 한다. 그리고 나서, 엡지 서버 할당 문제와 최대 허용 사용량 결정 문제를 공식화할 수 있다. 이는, 모든 엡지 서버의 보상들의 합계를 극대화하기 위한 목적으로 각각의 작업 τ_i 에서의 값 X_i , ($X_i=0, \dots, N^{\text{ES}}$)를 찾는 것과 모든 엡지 서버 j 에서의 최대 허용 가능한 사용률 Y_j , ($Y_j=0, \dots, C_j^{\text{max}}$)를 찾는 것으로 구성될 수 있다.

$$[62] \quad \text{Maximize} \quad \sum_{i=1}^{N^{\text{task}}} R_{i,X_i}$$

$$[63] \quad \text{subject to} \quad \forall i, H_{i,X_i} = 1, (i = 1, \dots, N^{\text{task}})$$

$$\forall j, U_j^{\text{total}} \leq Y_j, (j = 1, \dots, N^{\text{ES}})$$

$$\sum_{j=1}^{N^{\text{ES}}} P_j(U_j^{\text{total}}) \leq P^{\text{limit}}$$

[64] 도 4를 참고하면, 최대 허용 사용률 결정(maximum utilization determination: MUD) 알고리즘에 대한 의사 코드를 나타낸 도면이다.

[65] 작업 오프로딩 시스템은 최대 허용 가능한 사용률을 결정할 수 있다. 최적화 문제를 해결하기 위하여, 각각의 작업 τ_i 에 대한 값 X_i 과 각각의 엡지 서버 j 에 대한 최대 허용 가능한 사용률 Y_j 을 결정해야 한다. 각각의 엡지 서버 j 는 C_j^{max} 처리용량을 갖고 있으며, X_i 와 Y_j 의 모든 조합들을 고려하는 것은 비실용적이다. 이에 따라 이를 2개의 하위 문제로 나눈다. 사용률 결정 문제(UDP: utilization determination problem,)는 에너지 제약조건을 충족하면서

최대 허용 가능한 사용률 y_j 의 값들을 찾는 것이다. 엣지 서버 할당 문제(EAP: edge server allocation problem,)는 각각의 작업 τ_i 을 엣지 서버로 할당하는 X_i 값들을 찾는 것이다.

[66] 사용률 결정 문제에 대하여 설명하기로 한다.

[67] 엣지 서버 j 에서 가능한 보상들 $\sum_{v_i, X_i=j} R_{i,j}$ 과 그에 해당하는 에너지

요구조건들이 제공된다면, 사용률 결정 문제는 다중 선택 배낭 문제(multiple-choice knapsack problem, MKCP)로 축소될 수 있다. 과부하 없이 아이템들을 배낭에 넣었을 때 얻는 총 이익을 극대화시키기 위해서, 다수의 아이템들을 포함하는 몇몇 클래스들의 각각에 대해 가중치와 이익이 있는 하나의 아이템을 선택해야 한다.

[68] 사용률 결정 문제에서 클래스는 $C_j^{\max}$ 사용률 값을 각각 갖고 있는 엣지

서버이며, 각 사용률 값은 총합이 P^{limit} 를 초과할 수 없는 전력과 극대화 되어야

하는 보상과 연관되어 있다. MCKP는 NP-난해 이다. 하지만 이익-가중치 비율에 기반을 둔 그리디 알고리즘(greedy algorithm)은 보통 잘 수행된다. 사용률 결정 문제는 작업 할당으로부터 전력과 보상 증가량을 반복적으로 계산하여 부등식 (4)에서의 전력 제약조건이 충족되고 각각의 엣지 서버 j 에서 최대 허용 가능한 사용률 y_j 이 결정될 때까지 전력 비율에 더 높은 보상을 주는 사용률 값에

우선순위를 부여할 수 있다.

[69] 다음은, 엣지 서버 할당 문제에 대하여 설명하기로 한다.

[70] 작업 오프로딩 시스템은 최소 비용 최대 흐름 그래프(MCMF)를 사용하여 엣지 서버로의 작업 할당을 모델링할 수 있다. 실시예에서는 가능한 2가지 엣지 할당 알고리즘을 고려할 수 있다. 작업 분할을 허용하는 알고리즘과 허용하지 않는 알고리즘이다. 만약 작업 분할이 허용된다면, 생성된 하위작업들이 다른 엣지 서버에 할당될 수 있다. 모델링을 통해 작업 τ_i 는 N_i^{sub} 개의 하위 작업 $\tau_{i,m}$ 으로

분할될 수 있다. 작업 오프로딩 시스템은 각각의 하위 작업의 CPU 사용이 정수 값을 갖도록 조건화 하였고(작업의 사용률 값과 마찬가지로), 따라서 작업 τ_i 가 분할될 수 있는 최대 하위 작업 수는 U_i 이다. $X_{i,m}$ ($X_{i,m} = 0, \dots, N^{\text{ES}}$)이 하위

작업 $\tau_{i,m}$ 이 할당된 엣지 서버의 지수라고 하자. 최소 비용 최대 흐름 그래프

그래프에서 최적의 정수 기반 흐름 집합을 찾을 수 있고, 이는 최대 가능한 흐름 전달 비용을 최소화한다. 이러한 흐름들에 부합하는 $X_{i,m}$ 값들을 결정하기 위해

이 알고리즘을 사용할 수 있다.

[71] 작업 분할이 허용되지 않아 각각의 작업이 단일 엣지 서버에서 실행되어야 할

때, EAP는 이익을 극대화시키기 위해서 다른 가중치와 이익을 갖는 아이템들이 제한된 용량을 갖는 배낭으로 할당되어야 하는 다수 배낭 문제(MKP)로 축소될 수 있다. 다수 배낭 문제에서 아이템들은 작업이고, 엣지 서버 할당 문제의 목적은 각각의 엣지 서버에서 처리용량에 따른 총 보상을 극대화시키는 것이다. 다중 선택 배낭 문제(MKCP)처럼 다수 배낭 문제(MKP)는 NP난해이다. 작업 오프로딩 시스템은 처리용량이 남지 않을 때까지 최대 보상값을 기반으로 하여 작업들을 엣지 서버로 할당하는 그리디 알고리즘을 다시 사용할 수 있다.

[72] 다음으로, 최대 허용 사용률 결정 알고리즘에 대하여 설명하기로 한다.

[73] 3가지 새로운 변수를 필요로 하는 y_i 값을 결정해야 한다. 먼저 알고리즘이

진행되는 동안 X_i 의 값들을 저장하기 위해 각 작업 τ_i 에서 일시적인 변수

x_i^{tmp} 를 도입한다. 그리고 $p_{i,j}^{algo}$ 는 엣지 서버 j 에 의해 도출된 전력 증가량을

나타내고, 새로운 작업 τ_i 가 여기에 할당될 때 이는 다음과 같이 표현된다.

[74] 수학적식 5:

$$[75] \quad p_{i,j}^{algo} = P_j \left(\frac{\sum v_m x_m^{tmp=j} U_m}{C_j^{max}} + \frac{U_{i,j}}{C_j^{max}} \right) - P_j \left(\frac{\sum v_m x_m^{tmp=j} U_m}{C_j^{max}} \right)$$

[76] 여기에서 $P_j \left(\frac{\sum v_m x_m^{tmp=j} U_m}{C_j^{max}} \right)$ 는 작업 τ_i 이 각 엣지 서버 j 에게 할당되기 전에,

엣지 서버 j 에 의해 도출된 전력이고, $P_j \left(\frac{\sum v_m x_m^{tmp=j} U_m}{C_j^{max}} + \frac{U_{i,j}}{C_j^{max}} \right)$ 는 그 후에 엣지

서버 j 에 의해 도출된 전력이다.

[77] $v_{i,j}^{algo}$ 는 작업 τ_i 을 엣지 서버 j 에게 할당함으로써 발생하는 보상의 증가량과

수학적식 5의 증가량 간의 비율을 나타내는 특정 보상이다. 이 특정 보상은 다음과 같이 표현될 수 있다.

[78] 수학적식 6:

$$[79] \quad v_{i,j}^{algo} = \frac{R_{i,j}}{p_{i,j}^{algo}}$$

[80] 도 4의 최대 허용 사용률 결정(maximum utilization determination: MUD)

알고리즘의 목적은 적은 전력 소비량으로 큰 보상을 얻는 것이다. 최대 허용

사용률 결정 알고리즘은 각 i 와 j 에서의 전력 필요량 $p_{i,j}^{algo}$ 과 특정 보상 $v_{i,j}^{algo}$ 을

계산한다. 그리고 나서 $i=M$ 이고 $j=H$ 일 때의 최대값 $v_{i,j}^{algo}$ 을 결정한다. 그리고

임시 변수 x_H^{tmp} 를 1로 설정하여 τ_M 을 할당된 것으로 표시하고 총 전력 요구량

p_{used} 를 갱신한다. 이 과정은 최대 전력 조건이 충족되지 않을 때 혹은 작업이 더 이상 남지 않을 때까지 반복된다.

[81] 도 5는 일 실시예에 따른 작업 오프로딩 시스템에서 엣지 서버 할당 문제를 흐름 네트워크로 모델링하는 것을 설명하기 위한 예이다.

[82] 엣지 서버 할당에 대하여 설명하기로 한다.

[83] 작업 오프로딩 시스템은 최소 비용 최대 흐름 그래프(MCMF)를 작성할 수 있다. 엣지 서버 할당 문제(EAP)는 최소 비용 최대 흐름 문제로 효과적으로 표현될 수 있다. 이는 직접적으로 엣지에 연결되었고 소스, 싱크 그리고 중계 노드가 있는 흐름 네트워크인 최대 비용 최대 흐름 그래프를 생성함으로써 설명될 수 있다. 소스 노드는 외부로 향하는 엣지만 있으며 싱크 노드는 내부로 향하는 엣지만 있다. 노드 a에서 노드 b로의 엣지는 용량과 비용이 있다. 최소 비용 최대 흐름 그래프의 문제는 적은 비용에서 총 흐름을 극대화시키는 소스 노드에서 싱크 노드로 가는 흐름들의 집합을 찾는 것이다.

[84] 일례로, 엣지 서버 할당 문제는 도 5에서 제시된 것처럼 다음과 같은 4가지 종류의 정점을 갖는 흐름 네트워크로 모델링될 수 있다. 단일 소스 정점 s, v_i 가 작업 τ_i 에 상응하는 작업 정점 $v_1, \dots, v_{N^{task}}, w_j$ 가 엣지 서버 j에 상응하는 엣지 서버의 정점 $v_1, \dots, v_{N^{task}}, w_j$ 가 엣지 서버 j에 상응하는 엣지 서버의 정점 $w_1, \dots, w_{N^{ES}}$, 그리고 단일 싱크 정점 t이다. 이제, 엣지 서버 j에서 사용된 처리용량 단위당 보상을 다음과 같이 정의할 수 있다.

$$[85] \quad R_{i,j}^{unit} = \frac{R_{i,j}}{U_i}$$

[86] 이 경우, 흐름 네트워크를 생성할 때의 규칙은 다음과 같이 묘사할 수 있다.

[87] 소스 정점 s는 모든 작업 정점에 연결된다. 각 엣지(s, v_i)는 용량 U_i 과 비용 0을 가진다.

[88] 각 정점 $v_i, (i=1, \dots, N^{task})$ 는 $H_{i,j}=1$ 일 때, 정점 w_j 에 연결된다. 각 엣지는 용량 U_i 와 비용 $-R_{i,j}^{unit}$ 을 가진다.

[89] 각 정점 $w_j, (j=1, \dots, N^{ES})$ 는 싱크 정점 t에 연결된다. 각 엣지(w_j, t)는 용량 Y_j 와 비용 0을 가진다.

[90] 엣지 서버 할당 문제는 각 엣지 서버에서의 사용 조건을 만족하면서 총 보상을 극대화시키는 것을 목적으로 하며, 작업 분할이 허용될 때는 값 $x_{i,m}$ 을 결정하고, 작업 분할이 허용되지 않을 때는 x_i 를 결정하여 해결할 수 있다. 총 보상을 극대화시키는 것은 총 비용 $\sum -R_{i,j}^{unit} U_i$ 을 최소화시키는 것에 해당된다.

- [91] 만약 소스 노드에서 싱크 노드로의 작업 τ_i 에 대한 흐름 용량이 U_i 라면, 각 엣지 서버의 정점 w_j 에서 싱크 정점으로의 총 흐름은 엣지 서버 j 의 용량 제약조건인 γ_j 를 초과할 수 없다. $H_{ij}=1$ 인 경우에만 V_i 와 w_j 사이의 엣지가 존재하기 때문에 엣지 서버 커버리지 제약조건은 최소 비용 최대 흐름 그래프에 표현될 수 있다.
- [92] 도 7은 일 실시예에 따른 작업 오프로딩 시스템에서 작업 분할이 허용된 경우의 엣지 서버 할당 알고리즘에 대한 의사 코드를 나타낸 도면이다.
- [93] 작업 분할이 허용될 때의 엣지 서버 할당 알고리즘(EAA-TS)은 알고리즘은 도 7에 제시되어 있다. EAA-TS는 감소하는 비용으로 소스에서 싱크로의 길을 찾고, 이는 또한 최단 경로 빠른 알고리즘(shortest path faster algorithm: SPFA)을 사용하여 용량 조건을 만족하며 이는 잘 알려진 벨만-포드 알고리즘의 개선된 형태이다.
- [94] 실시예에서는 정점(노드) a 와 b 사이의 흐름을 표현하기 위해서 이진 함수 $f(a, b)$ 를 도입하기로 한다. $f(a, b)$ 값이 1이라면, 흐름이 존재한다. 알고리즘 실행 시 선택된 경로는 최종 흐름의 일부분이 아니기 때문에 최단 경로 빠른 알고리즘은 역방향 경로가 필요하며 이는 취소될 수 있다. 따라서 각 엣지가 2개의 정점 사이에 허용될 수 있는 추가적인 흐름을 나타내고 역방향 경로가 허용되는 잔차 그래프를 작성해야 한다.
- [95] 예를 들어, v_1 에서 w_1 으로 흐름이 있도록 하기 위해 $f(v_1, w_1) = 1$ 이라고 가정해보자. 이에 해당하는 잔차 그래프가 도6에 표시되어 있다. 모든 엣지들 $s \rightarrow v_1 \rightarrow w_1 \rightarrow t$ 에 대한 용량 값은 1만큼 감소했지만, 역방향 경로의 엣지들의 용량은 1만큼 증가했다. 이 역방향 경로 $C(w_1, v_1)$ 에 대한 비용 $-R_{v_1, w_1}^{unit}$ 은 $-C(v_1, w_1)$ 로 설정된다. 역방향 경로는 알고리즘에 의해 선택될 수 있고, 이 경우 최초의 경로 선정은 취소된다. 최단 경로 빠른 알고리즘은 가능한 흐름이 없어질 때까지 잔차 그래프에서 작동한다.
- [96] EAA-TS는 최단 경로 빠른 알고리즘 이전과 이후에 작동하는 초기화, 경로 증강 그리고 마무리 과정을 포함한다.
- [97] 초기화: 잔차 그래프가 최소 비용 최대 흐름 그래프(라인 1)에서 생성되고 흐름 $f(a, b)$ 의 값과 비용 $C(a, b)$ 이 초기화 된다(라인5-9).
- [98] 경로 증강: 최소 비용 최대 흐름 그래프를 사용해서 최소 비용 흐름을 구한다(라인 11). 이에 해당하는 경로가 잔차 그래프에 생성된다(라인 15). 그리고 흐름이 갱신된다(라인 17-18). 이 과정은 G^R 에 생성되는 증강 경로 p 가 없어질 때까지 반복된다(라인 12).
- [99] 마무리: V_i 와 w_j 사이에 흐름이 있으면, τ_i 에 있는 하위 작업 중 하나를

처리하기 위해 엣지 서버 j 가 선정되고, $\chi_{i,m}$ 값은 이에 따라 갱신된다(라인 21-29).

- [100] 도 8은 일 실시예에 따른 작업 오프로딩 시스템에서 작업 할당 없는 엣지 할당 알고리즘에 대한 의사 코드를 나타낸 도면이다.
- [101] 작업 분할 없는 엣지 할당 알고리즘(EAA-NTS)은 다음과 같이 최소 비용 최대 흐름 그래프에서 작동된다.
- [102] 최소 비용 흐름: 소스 정점에서 싱크 정점으로 최소 비용 흐름 f 가 계산된다(라인5).
- [103] 갱신 및 제거: 최소 비용 흐름과 연관된 작업 정점이 제거되고, 용량 사용량 C_j^{imp} 과 지수 X_i 가 갱신되며 흐름과 연관된 엣지들이 제거된다(라인 6-16). 예를 들어, 최소 비용 흐름이 도5에서 $s \rightarrow v_1 \rightarrow w_2 \rightarrow t$ 라고 가정하자. 그렇다면 τ_1 에 해당되는 정점은 들어오는 엣지 및 나가는 엣지와 함께 제거된다. τ_1 이 엣지 서버 2로 할당되었기 때문에 X_1 값은 2로 갱신된다. 엣지 서버 2에 의해 증가된 처리 사용량을 기록하기 위해서 임시 변수 C_2^{imp} 는 U_1 만큼 증가되며 이는 Y_j 를 초과해서는 안된다. 이러한 2가지 과정은 모든 작업들이 검토될 때까지 반복된다.
- [104] 도 9 내지 도 14는 일 실시예에 따른 작업 오프로딩 시스템의 실험 결과를 설명하기 위한 그래프이다.
- [105] 1. 시뮬레이션 설정
- [106] 전력 소비량과 총 보상에 대한 우리의 스킴을 평가하기 위해서 시뮬레이션이 진행될 수 있다. 이를 위해서 2019년과 2020년 사이에 공개된 4개의 상업용 서버의 에너지 특성을 모델링할 수 있다. 이들 상업용 서버의 특성들 중 하나가 각 엣지 서버로 무작위 할당되었다.
- [107] 일례로, 호주 멜버른에서 핸드폰 네트워크 장소 데이터의 데이터베이스를 사용하여 엣지 서버와 IoT 장비 사이의 공간 관계를 모델링 했다. 이는 815개의 무선 액세스 포인트 (WAPs)와 125개의 기지국의 위도, 경도, 그리고 IP 주소를 포함하며 이를 사용하여 WAPs와 각 기지국 사이의 거리를 계산할 수 있다.
- [108] 각 기지국이 엣지 서버 커버리지 안에서 작업을 수행할 수 있는 단일 엣지 서버를 갖고 있다고 가정할 것이다. 그리고 멜버른 데이터베이스의 WAPs의 위치정보를 취득하여 IoT 장비의 위치를 나타낼 것이다. 만약 작업을 오프로드 하고 싶어하는 장비에 연결된 WAP가 ES에 연결된 기지국(BS)의 커버리지 안에 있다면, $H_{ij}=1$ 이다 (그렇지 않으면 0). 각 BS의 커버리지는 450m와 750m사이에 무작위로 선정된 거리로 표현된다.
- [109] 다양한 변수의 효과를 살펴보기 위해, 우리는 작업, 보상 및 작업 사용량을 변화시켰고, 이들은 표 1에 나타난 것처럼 서로 다른 범위를 갖고 있다. 각

WAP는 γ^T 와 δ^T 사이의 임의의 작업 수를 생성하는 IoT 장비들로 연결될 수 있다. 모든 엣지 서버 j 에 대하여 동일하게, 작업 τ_i 에 대한 기본 보상은 γ^R 과 δ^R 사이에 임의로 선정된 정수값이고, 작업 τ_i 에 의해 요구되는 사용량 U_i 는 γ^U 와 δ^U 사이에 임의로 선정된 정수값이다. 다르게 서술되지 않는 경우에, 각 변수에 대한 기본 범위는 표1과 같다.

[110] 표 1 매개변수 설정

[111]

매개변수	범위 묘사	기본 범위	실험에서 사용된 범위
WAP 당 작업 수	(γ^T, δ^T)	(5,10)	(3,6), (5,10), (7,14), (9,18)
보상	(γ^R, δ^R)	(1,15)	(1,5), (1,15), (1,25), (1,35)
작업 사용량 U_i	(γ^U, δ^U)	(3,8)	(1,6), (3,8), (5,10), (7,12), (1, 12)

[112] 도 8을 참고하면, 2가지 스킴, 즉, EAA-TS (알고리즘 (도면 7)) 그리고 EAA-NTS (알고리즘 (도면 8))를 평가한다. 둘 다 각 엣지 서버에서 최대 허용 가능한 사용을 결정하기 위해 최대 허용 사용률 결정 알고리즘을 사용했다. 부등식 (4)를 만족시키기 위해 전력 소비량을 제한하면서 엣지 서버에 작업을 할당하는 다음과 같은 4가지 방식으로 이들 스킴을 비교한다.

[113] RR + HR(라운드-로빈 할당 + 최대 보상)은 라운드 로빈 방식으로 작업을 할당하며 부등식 (4)가 충족되지 않을 때까지 최고 보상을 산출하는 작업을 연속적으로 선정한다.

[114] RA+HR(임의 작업 할당 + 최대 보상)은 부등식 (4)가 충족 되지 않을 때까지 임의로 선정된 작업들에 대해 높은 보상을 갖는 할당을 연속적으로 선정한다.

[115] MUD+HR(MUD+높은 보상)은 높은 보상을 산출하는 작업 할당이 먼저 수행되도록 값이 설정되는 것을 제외하고 최대 허용 사용률 결정 알고리즘과 유사하다. 값들은 부등식 (4)가 충족되지 않을 때까지 연속적으로 갱신 된다.

[116] MUD+HUR(MUD + 높은 기본 보상)은 값이 설정된 것을 제외하고 MUD+HR과 유사하다. 값들은 부등식 (4)가 충족되지 않을 때까지 연속적으로 갱신된다.

[117] 첫 2가지 스킴은 균형을 선호하고, MUD+HR은 높은 보상을 목적으로 하며, MUD+HUR은 사용량에 따른 높은 보상 비율을 목적으로 한다.

[118]

[119] 2. 작업 분할 효과

[120] 모든 엣지 서버가 완전히 사용 될 때 총 전력 소비량의 비율로 전력 한계 값들을

표현한다. 총 보상에 대한 작업 분할 효과를 살펴보기 위해 40%, 60%, 80%라는 3가지 합계 보상을 살펴보면서 표 1의 각각의 변수에 대한 다양한 범위 값들의 모든 조합들에 대해 EAA-TS와 EAA-NTS를 비교할 수 있다.

- [121] 표 1은 (γ^U , δ^U)과 p_{ratio} 의 다른 값들에 대한 EAA-TS와 EAA-NTS의 결과 간의 평균 비율 차이를 나타낸다. EAA-TS와 EAA-NTS 결과 간의 비율 차이가 다른 매개변수들의 값에 상관없이 무시할 수 있음을 발견할 수 있다. 예를 들어 $\delta^U = 12$ 일 때 0.33%를 넘지 않았고, $\delta^U = 6$ 일 때 0.48%를 넘지 않았다. 하지만, γ^U 이 증가함에 따라 차이가 커져가는 것을 확인할 수 있다. 그 이유를 다음과 같이 설명될 수 있다. EAA-NTS는 하나의 작업 전체를 처리하기 위해 남은 용량이 부족한 경우 엡지 서버에 작업을 할당할 수 없으므로, 사용률이 낮으며, 이는 주로 최소 작업 사용량 크기가 1을 초과할 때 발생한다. γ^U 이 증가함에 따라, 이 둘 사이의 차이는 줄어들고, 이는 가용 전력이 낮을 때 작업 분할이 효과적이라는 것을 의미한다.

- [122] 표 2: (γ^U , δ^U)과 p_{ratio} 의 다른 값들에 대한 EAA-NTS와 EAA-TS 비교

[123]

$2^*_{p_{ratio}}$	(γ^U , δ^U)				
	(1,6)	(1,12)	(3,8)	(5,10)	(7,12)
40%	0.01%	0.07%	2.87%	7.22%	7.78%
60%	0.04%	0.04%	1.95%	3.53%	5.35%
80%	0.01%	0.01%	1.15%	3.04%	4.57%

- [124] 3. 총 보상에 대한 전력 제한 효과

- [125] 총 보상에 대한 전력 제한 효과를 살펴보았다. 이제부터 작업 분할을 허용하지 다른 스킴들(RR+HR, RA+HR, MUD+HR 및 MUD+HUR)과 비교할 때 EAA-NTS를 기본 스킴으로 사용할 것이다. 도9는 표 1의 각 매개변수에 대한 기본 범위가 사용되었을 때 모든 5개 스킴에 대한 보상을 나타내고, 도면10은 EAA-NTS를 기준으로 정규화된 결과를 나타낸다.

- [126] 전력 제한이 더 높아지면 더 큰 보상을 제공한다. 또한, EAA-NTS는 모든 전력 제한과 관련하여 높은 보상을 산출한다. 이 보상들은 다른 스킴에 비교해서 7%에서 80%까지 높다. 이러한 차이는 더 낮은 전력 제한에서 더 크게 나타난다. MUD+HUR 스킴은 MUD+HR 보다 높은 보상을 산출하며 이는 높은 값을 갖는 작업이 먼저 할당되어야 한다는 것을 의미한다. 하지만 MUD+HUR 작업은 엡지 서버 특성을 고려하지 않는다. 이는 평균적으로 EAA-NTS 보상의 70%, 그리고 최소 전력 제한에서는 40% 수준만 산출할 뿐이다.

[127]

- [128] 4.작업 수에 대한 총 보상

[129] 총 보상이 어떻게 각 WAP내에서 생성된 작업 수에 의존하는지 알아볼 수 있다. 도 11은 3가지 p_{ratio} 값의 평균 보상 결과인 40%, 60%, 80% 를 보여준다.

다시 EAA-NTS가 다른 스킴에 비해 좋은 성과를 보여주었고, 작업의 수가 증가할 수록 차이가 커지는 경향을 보였다. 이를 초기 선정 시에 높은 보상을 갖는 작업이 더 많이 가능하다는 것으로 해석할 수 있다. 예를 들어, 작업 수 범위가 (3,6)에서 (9,18)로 변화했을 때 EAA-NTS에 의한 보상이 5060만큼 증가한 반면에, MUD+HUR로 얻은 보상은 2684만큼 증가함을 확인할 수 있다.

[130]

[131] 5. 보상 범위의 효과

[132] 다양한 보상 범위 (γ_R, δ_R)의 효과를 살펴보았다. 도12는 40%, 60% 그리고

80%일 때 평균 보상 결과를 나타낸다. 다시 EAA-NTS가 γ_R 과 δ_R 간의 차이가

커지면서 마진이 증가하여 다른 스킴들보다 우수한 성능을 나타내는 것을 확인할 수 있다. 작업들에 대해 넓은 보상 범위를 갖는 것이 더 큰 최적화 범위를 부여하며, EAA-NTS이 이를 잘 활용하는 것으로 보여진다.

[133]

[134] 6. 작업 사용량 범위 효과

[135] 작업 사용량 값의 범위에 따라 총 보상이 어떻게 달라지는지 확인할 수 있다. 도 13은 다수의 다른 범위(γ_R, δ_R)에 대한 총 보상과 3가지 p_{ratio} 값 40%, 60%

그리고 80%에 대한 평균값을 나타낸다. 더 적은 수의 작업이 엣지 서버로 할당될 수 있기 때문에, 총 보상은 범위가 (1,6)에서 (3,8)로 증가했을 때 약간 증가한다. 하지만 이 작업들은 더 높은 보상을 받는다. 하지만, 그 범위 밖에서 U_i 의 값은 보상 결과에 거의 영향을 주지 못한다.

[136]

[137] 7. EAA-NTS와 같은 보상을 얻기 위해 다른 스킴이 필요로 하는 전력 소비량

[138] p_{ratio} 의 다양한 값들에 대해서 EAA-NTS를 사용했을 때 얻어지는 총 보상을

결정할 수 있다. 동일한 보상을 얻기 위해서 다른 스킴들이 필요로 하는 전력을 찾을 수 있다. 도 14는 각각의 경우를 EAA-NTS에 의해 도출된 전력으로 정규화한 결과를 나타낸다. 동일한 보상을 얻기 위해서 다른 스킴들은 EAA-NTS보다 13%과 75% 더 높은 전력을 필요로 하는 것을 확인할 수 있다.

예를 들어, MUD+HUR은 EAA-NTS보다 13%에서 29% 많은 전력을 필요로 했고, MUD+HR은 19%에서 30% 더 많은 전력을 필요로 했다. $p_{ratio}=60\%, 70\%$ 및

80%일 때, RR+HR과 RA+HR은 모든 엣지 서버들이 완전히 사용되었음에도 불구하고 EAA-NTS와 같은 보상을 산출하지 못함을 확인할 수 있다.

[139]

- [140] 일 실시예에 따르면, 엡지 서버가 소비하는 전력을 제한하면서, 엡지 서버에 의해 획득되는 총 보상을 극대화하는 스킴을 제안할 수 있다. 상세하게는, 먼저 각각의 엡지 서버의 프로세서 용량과 사용량에 기반하여 엡지 서버가 산출하는 전력을 조사하고, 총 보상을 극대화 하기 위한 목적을 가지고 각 엡지 서버의 최대 허용 가능한 사용률과 작업 할당을 결정하는 최적화 문제를 공식화 할 수 있다. 최대 허용 가능한 사용률을 결정하기 위해서, 엡지 서버에서 최소의 총 전력 소비량으로 최고 보상을 얻으려고 하는 그리디 알고리즘을 사용할 수 있다. 작업 할당 문제를 최소 비용 최대 흐름 그래프로 공식화하고 그래프에서 최고 보상 경로를 탐색하는 2가지 알고리즘을 제안하였다. 하나의 알고리즘은 작업 분할을 허용하고, 다른 하나의 알고리즘은 작업 분할을 허용하지 않는다.
- [141] 또한, 작업 수, 전력 제한 값, 보상과 작업 사용량 범위, 그리고 전력 소비량에 대한 보상 비율을 다양화 하면서 총 보상에 대한 우리 스킴을 평가하기 위해 시뮬레이션을 실시할 수 있다. 실험 결과는 실시예에서 제안된 스킴이 큰 보상을 획득하기 위해 가용 전력을 잘 활용하는 것으로 나타났다. 제안된 스킴의 보상은 다른 스킴들을 사용하여 얻어진 보상보다 7%에서 80%만큼 높았고, 다른 스킴들은 같은 보상을 얻기 위해서 제안된 스킴보다 13% 에서 75% 정도 더 많은 전력이 필요했다. 실시예에서 제안된 스킴은 가용 전력이 감소되었을 때, 각 작업에 할당된 보상이 매우 가변적일 때, 그리고 많은 작업들이 있을 때 특히 효과적이다. 작업 분할은 최소 작업 사용량 크기가 1일 때는 큰 차이가 없지만, 평균 작업이 높은 CPU 처리용량을 필요로 할 때 효과적이다.
- [142] 증가하는 IoT 작업 수를 지원하기 위해서 엡지 컴퓨팅에 대한 요구가 급격하게 증가하며 엡지 서버에 의한 전력 소비량을 증가시키는 결과를 초래했다. 실시예에서 제시한 실험 결과는 제안된 스킴이 엡지 서버의 효과적인 전력 관리에 기여할 수 있다는 것을 나타낸다.

[143]

발명의 실시를 위한 형태

- [144] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로

설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

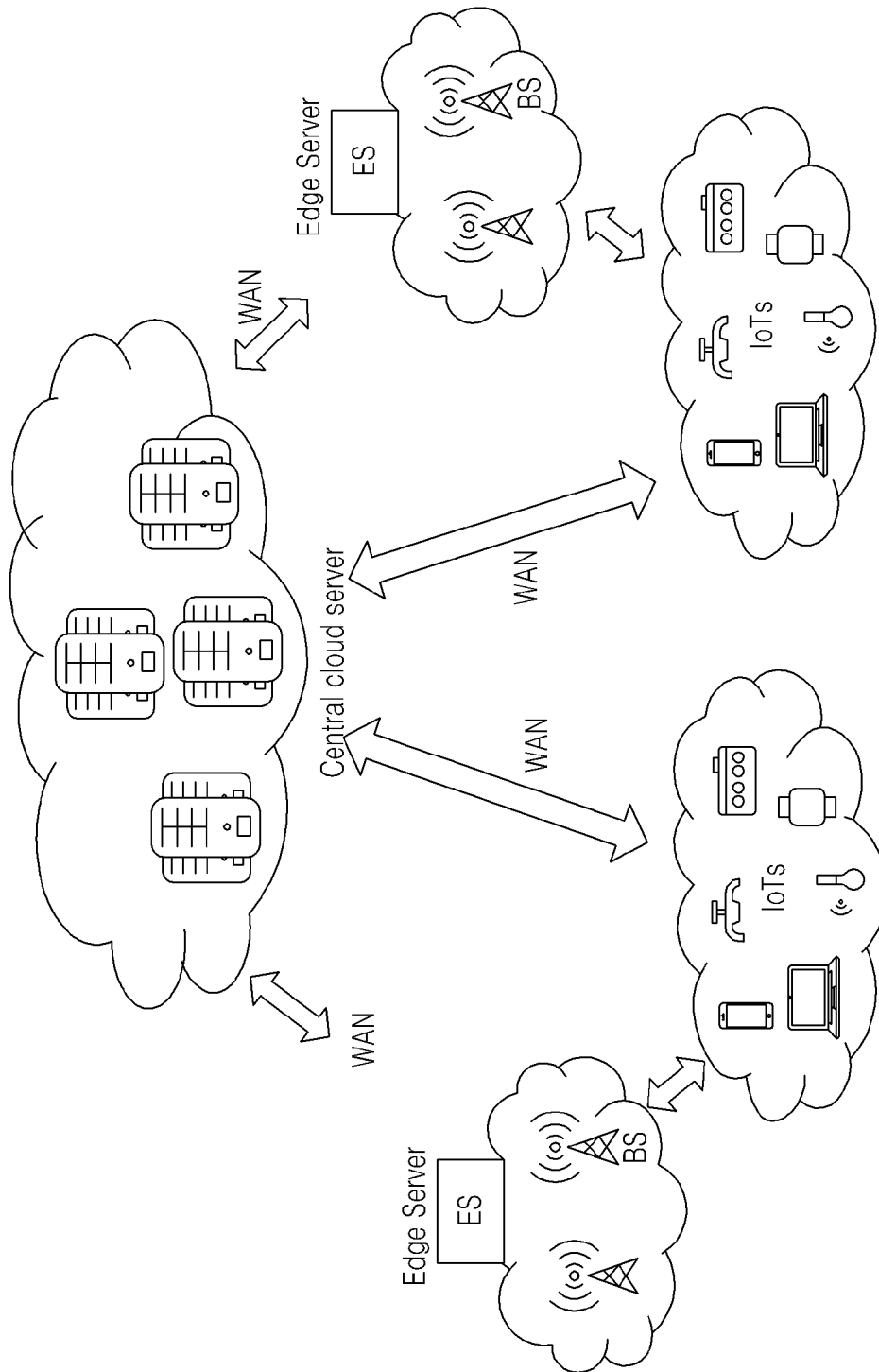
- [145] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [146] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.
- [147] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.
- [148] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

청구범위

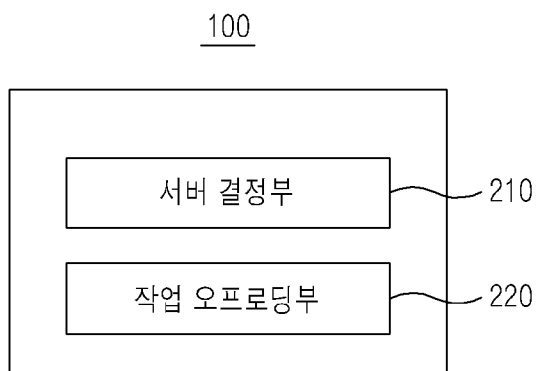
- [청구항 1] 컴퓨터로 구현되는 작업 오프로딩 시스템에 의해 수행되는 작업 오프로딩 방법에 있어서,
각각의 엡지 서버의 전력 정보에 기초하여 오프로드된 작업을 실행할 엡지 서버를 결정하는 단계; 및
상기 결정된 엡지 서버에 설정된 할당 정보에 따라 작업을 오프로딩하는 단계를 포함하는 작업 오프로딩 방법.
- [청구항 2] 제1항에 있어서,
상기 작업을 오프로딩하는 단계는,
상기 각각의 엡지 서버의 한도 전력 하에서 엡지 서버 제공자의 보상을 최대화하기 위한 상기 각각의 엡지 서버의 최대 허용 가능한 사용률과 작업 할당을 포함하는 할당 정보를 결정하는 단계를 포함하는 작업 오프로딩 방법.
- [청구항 3] 제1항에 있어서,
상기 작업을 오프로딩하는 단계는,
상기 각각의 엡지 서버의 최대 허용 가능한 사용률을 결정하는 동시에 전력 요구량에 대한 엡지 서버 제공자의 보상 비율이 높은 작업의 오프로드에 높은 우선 순위를 부여하는 단계를 포함하는 작업 오프로딩 방법.
- [청구항 4] 제1항에 있어서,
상기 작업을 오프로딩하는 단계는,
상기 각각의 엡지 서버의 전력 제약조건을 충족하면서 최대 허용 가능한 사용률의 값들을 탐색하는 단계를 포함하는 작업 오프로딩 방법.
- [청구항 5] 제4항에 있어서,
상기 작업을 오프로딩하는 단계는,
작업 할당으로부터 전력 증가량과 보상 증가량을 반복적으로 계산하여 전력 제약조건이 충족되고, 각각의 엡지 서버에서 최대 허용 가능한 사용률이 결정될 때까지 전력 비율에 더 높은 보상을 주는 최대 허용 가능한 사용률 값에 우선순위를 부여하는 단계를 포함하는 작업 오프로딩 방법.
- [청구항 6] 제1항에 있어서,
상기 작업을 오프로딩하는 단계는,
상기 각각의 엡지 서버로 작업 할당하기 위한 각각의 작업에 대한 값을 탐색하는 단계를 포함하는 작업 오프로딩 방법.

- [청구항 7] 제6항에 있어서,
상기 작업을 오프로딩하는 단계는,
최소 비용 최대 흐름 그래프를 사용하여 작업 분할을 허용하는 방법 또는
작업 분할을 허용하지 않는 방법 중 어느 하나의 방법을 이용하여 엷지
서버로의 작업 할당을 수행하는 단계
를 포함하는 작업 오프로딩 방법.
- [청구항 8] 제7항에 있어서,
상기 작업 분할을 허용하는 방법은,
잔차 그래프가 상기 최소 비용 최대 흐름 그래프에서 생성되고 각 노드
사이의 흐름을 표현하기 위한 이진 함수의 값과 비용을 초기화하는 과정,
최단 경로 빠른 알고리즘을 사용하여 최소 비용 흐름을 도출함에 따라
상기 잔차 그래프에 경로를 생성하고, 상기 흐름을 갱신하는 경로 증강
과정 및 정점 사이에 흐름이 존재할 경우, 작업을 구성하는 하위 작업 중
하나를 처리하기 위해 엷지 서버를 선정함으로써 상기 하위 작업이
할당된 엷지 서버의 지수값을 갱신하는 마무리 과정을 수행하는
작업 오프로딩 방법.
- [청구항 9] 제7항에 있어서,
상기 작업 분할을 허용하지 않는 방법은,
상기 최소 비용 최대 흐름 그래프에 작동되고, 소스 정점에서 싱크
정점으로 최소 비용 흐름을 계산하는 최소 비용 흐름 과정, 상기 최소
비용 흐름과 연관된 작업 정점을 제거하고 용량 사용량과 작업이 할당된
엷지 서버의 지수값을 갱신하고 흐름과 연관된 엷지들을 제거하는
과정을 수행하는
작업 오프로딩 방법.
- [청구항 10] 컴퓨터로 구현되는 작업 오프로딩 시스템에 있어서,
각각의 엷지 서버의 전력 정보에 기초하여 오프로드된 작업을 실행할
엷지 서버를 결정하는 서버 결정부; 및
상기 결정된 엷지 서버에 설정된 할당 정보에 따라 작업을 오프로딩하는
작업 오프로딩부
를 포함하는 작업 오프로딩 시스템.

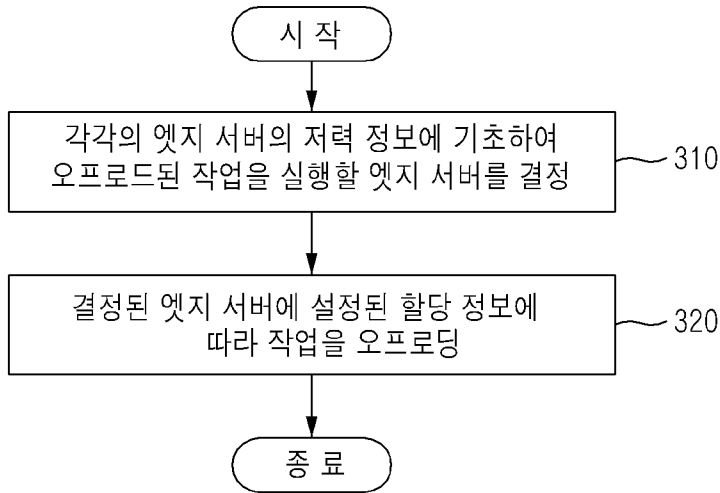
[도1]



[도2]



[도3]



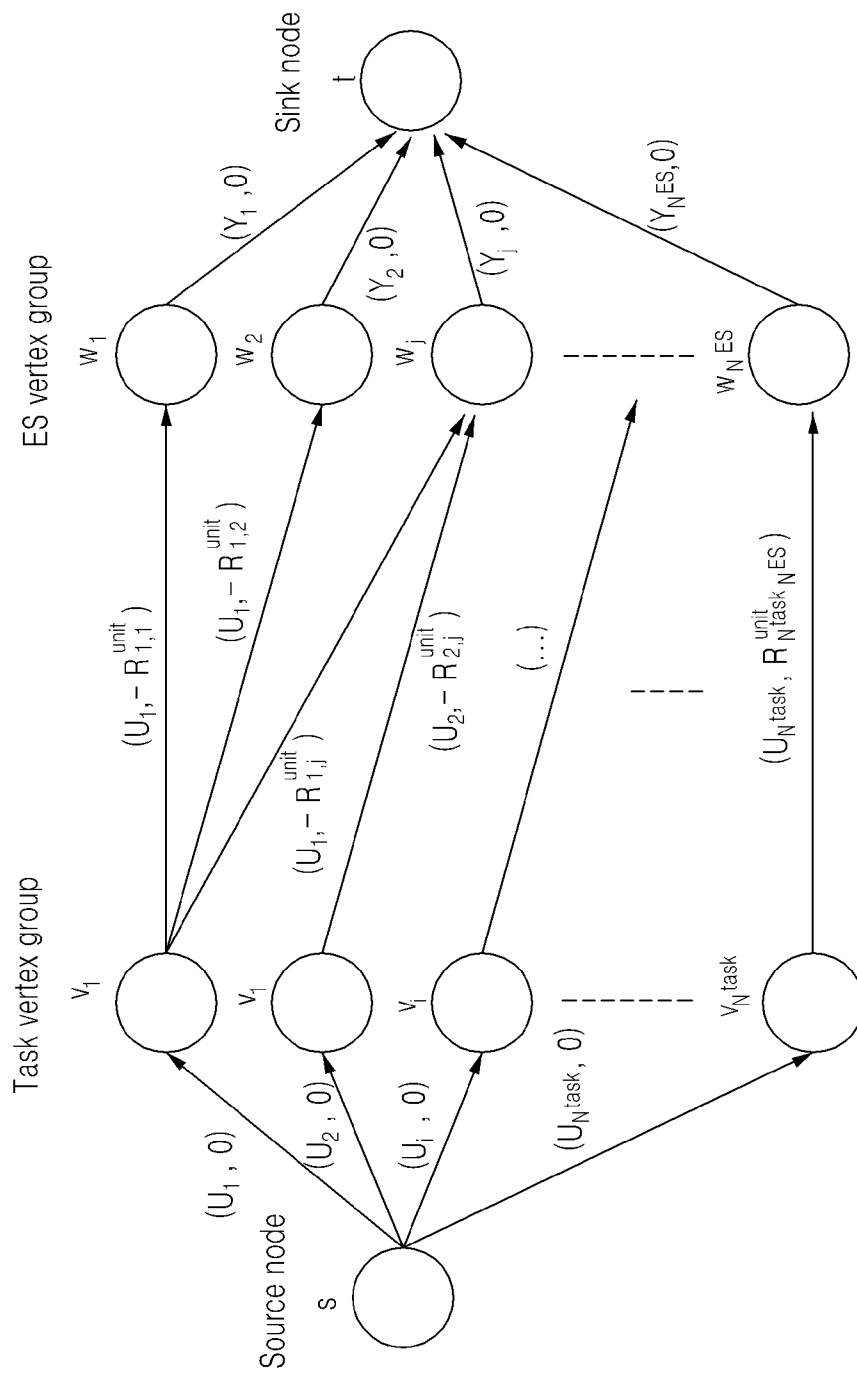
[도4]

Algorithm 1: Maximum allowable utilization determination(MUD) algorithm

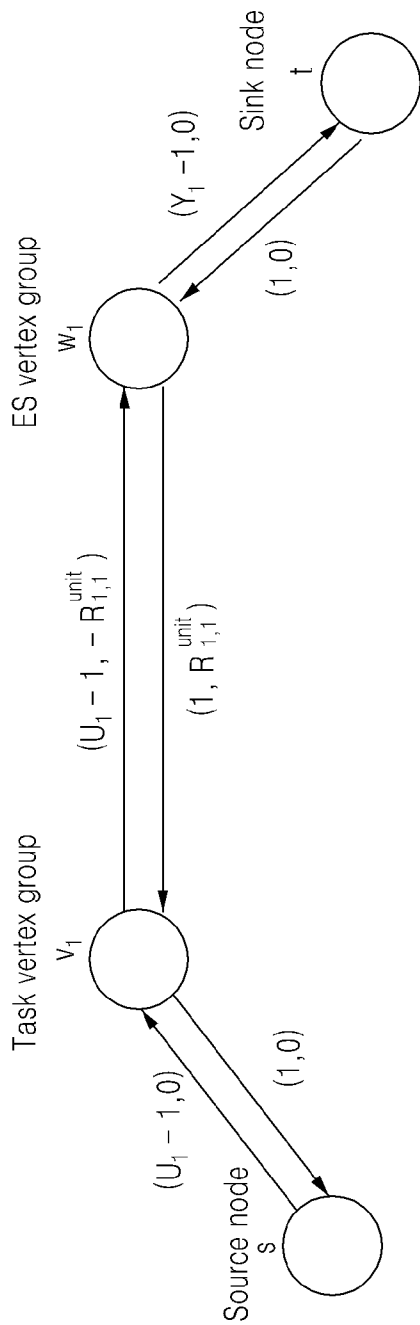
Output: Y_j , ($j = 1, \dots, N^{ES}$);

- 1 Boolean variable: $\text{flag} \leftarrow \text{TRUE}$;
- 2 Temporary variables: $P_{i,j}^{\text{algo}}$ and $V_{i,j}^{\text{algo}}$,
($i = 1, \dots, N^{\text{task}}$), ($j = 1, \dots, N^{ES}$)
- 3 Temporary variable: $P \leftarrow 0$;
- 4 Temporary variable: $\forall i, X_i^{\text{tmp}} \leftarrow 0$;
- 5 Set of task vertexes: $S^{\text{task}} = \{\tau_1, \dots, \tau_{N^{\text{task}}}\}$;
- 6 while flag do
- 7 for $\forall \tau_i \in S^{\text{task}}$ do
- 8 Calculate $P_{i,j}^{\text{algo}}$ and $V_{i,j}^{\text{algo}}$ using Equations (5)
and (6);
- 9 end
- 10 Find the highest value of $V_{i,j}^{\text{algo}}$, (say, $V_{H,M}^{\text{algo}}$);
- 11 if $P^{\text{used}} + P_{H,M}^{\text{algo}} \leq P^{\text{limit}}$ then
- 12 $P^{\text{used}} \leftarrow P^{\text{used}} + P_{H,M}^{\text{algo}}$;
- 13 $X_H^{\text{tmp}} \leftarrow M$;
- 14 $S^{\text{task}} \leftarrow S^{\text{task}} - \{\tau_H\}$;
- 15 else
- 16 $\text{flag} \leftarrow \text{FALSE}$;
- 17 end
- 18 if $S^{\text{task}} = \emptyset$ then
- 19 $\text{flag} \leftarrow \text{FALSE}$;
- 20 end
- 21 end
- 22 for $j = 1$ to N^{ES} do
- 23 $Y_j \leftarrow \sum v_i X_i^{\text{tmp}} = \sum U_i$;
- 24 end

[도5]



[도6]



[도7]

Algorithm 2: Edge Allocation Algorithm with Task Splitting (EAA-TS).

Input: A flow network G in Fig. 3;
Output: $\forall j, m, X_{i,m}$ for $\tau_{i,m}$;

- 1 Residual network G^R of G ;
- 2 Temporary variables for all the edges, (a,b) in network G^R : $f(a,b)$, $f(b,a)$, $C(a,b)$ and $C(b,a)$;
- 3 Temporary variable: $l_i^{\text{sub}} \leftarrow 1$ ($i = 1, \dots, N^{\text{task}}$);
- 4 Construct a residual graph G^R based on G ;
- 5 for all edges (a,b) in G^R do
- 6 $C(b,a) \leftarrow -C(a,b)$;
- 7 $f(a,b) \leftarrow 0$;
- 8 $f(b,a) \leftarrow 0$;
- 9 end
- 10 while TRUE do
- 11 Run the SPFA algorithm to find an augmenting path p from s to t using the minimum cost from G^R ;
- 12 if there is no augmenting path p from G^R then
- 13 break ;
- 14 end
- 15 Augment all the flows along p to network G^R ;
- 16 for all edges $(a,b) \in p$ do
- 17 $f(a,b) \leftarrow f(a,b) + 1$;
- 18 $f(b,a) \leftarrow f(b,a) - 1$;
- 19 end
- 20 end
- 21 for $j = 1$ to $j = N^{\text{ES}}$ do
- 22 for all edges from task to ES vertexes, (v_i, w_j) in G^R do
- 23 while $f(v_i, w_j) > 0$ do
- 24 $X_{i, l_i^{\text{sub}}} \leftarrow j$;
- 25 $l_i^{\text{sub}} \leftarrow l_i^{\text{sub}} + 1$;
- 26 $f(v_i, w_j) \leftarrow f(v_i, w_j) - 1$;
- 27 end
- 28 end
- 29 end

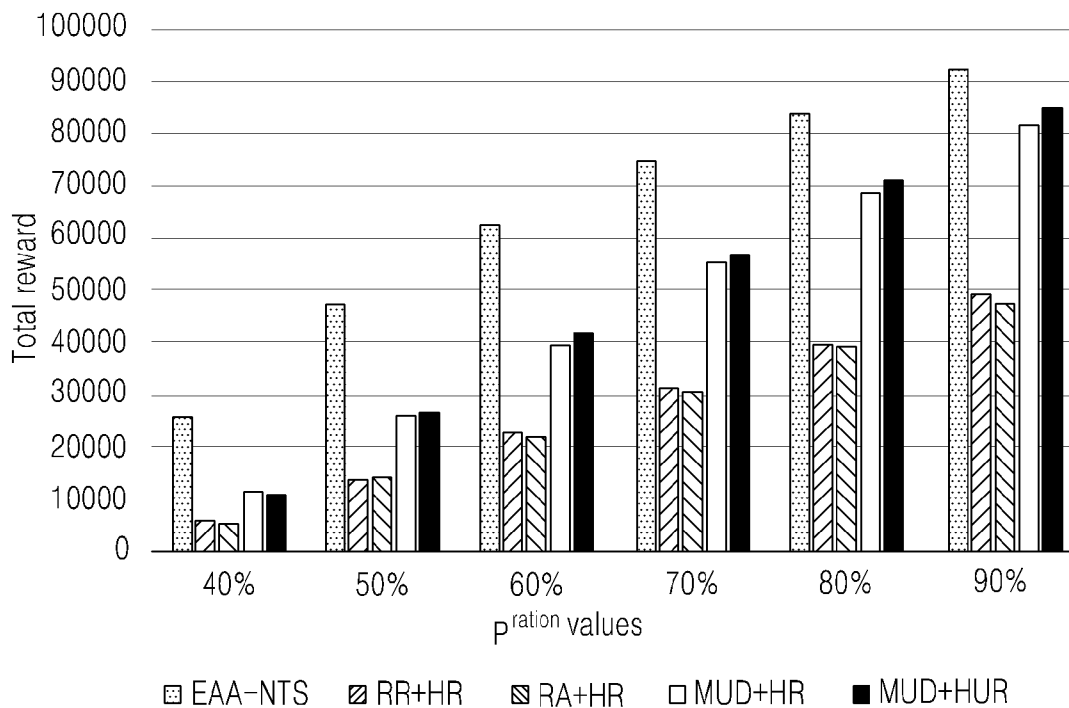
[도8]

Algorithm 3: Edge Allocation Algorithm for no task splitting (EAA-NTS).

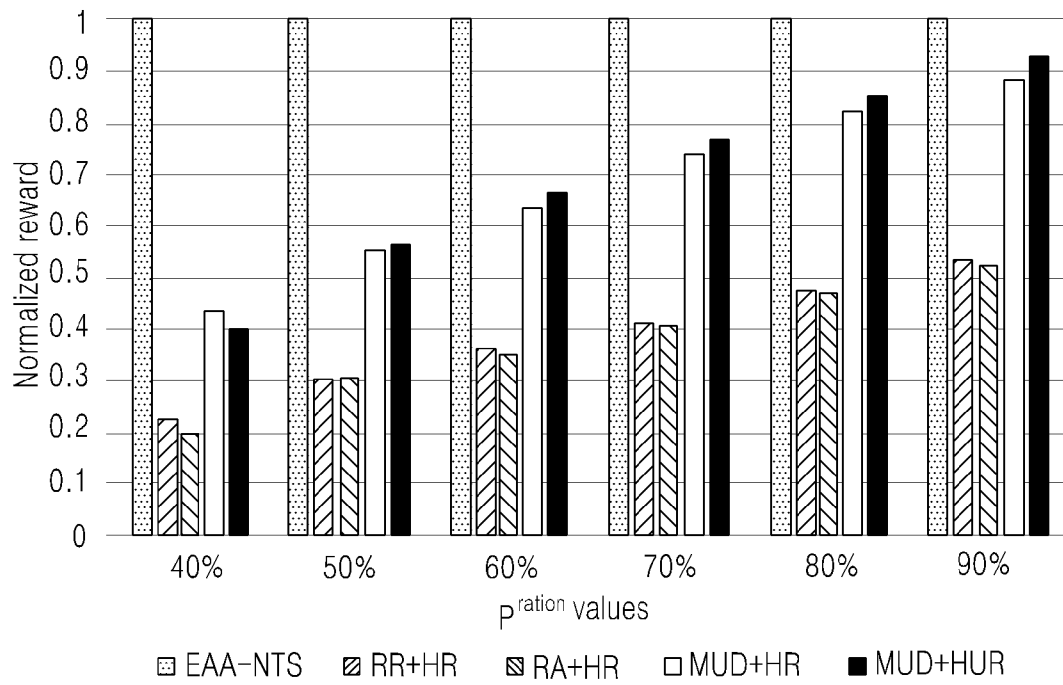
Input: A flow network G in Fig. 3;
Output X_i for τ_i , ($i = 1, \dots, N^{\text{task}}$);

- 1 Temporary variable; $C_j^{\text{tmp}} \leftarrow 0$, ($j = 1, \dots, N^{\text{ES}}$);
- 2 $S^{\text{task}} \leftarrow \{v_1, \dots, v_{N^{\text{task}}}\}$;
- 3 $\forall i, X_i \leftarrow 0$;
- 4 while $S^{\text{task}} \neq \emptyset$ do
- 5 Find the minimum-cost flow with the lowest values
of $-R_{i,j}^{\text{unit}}$, (say, $s \rightarrow v_L \rightarrow w_M \rightarrow t$);
- 6 if $C_M^{\text{tmp}} + U_L \leq Y_j$ then
- 7 $X_M \leftarrow M$;
- 8 $C_M^{\text{tmp}} \leftarrow C_M^{\text{tmp}} + U_L$
- 9 Eliminate an edge from s to v_L and all the
outgoing edges from v_L ;
- 10 $S^{\text{task}} \leftarrow S^{\text{task}} - \{v_L\}$;
- 11 else
- 12 Eliminate an outgoing edge from v_L to w_M ;
- 13 if there is no outgoing edge from v_L then
- 14 $S^{\text{task}} \leftarrow S^{\text{task}} - \{v_L\}$;
- 15 end
- 16 end
- 17 end

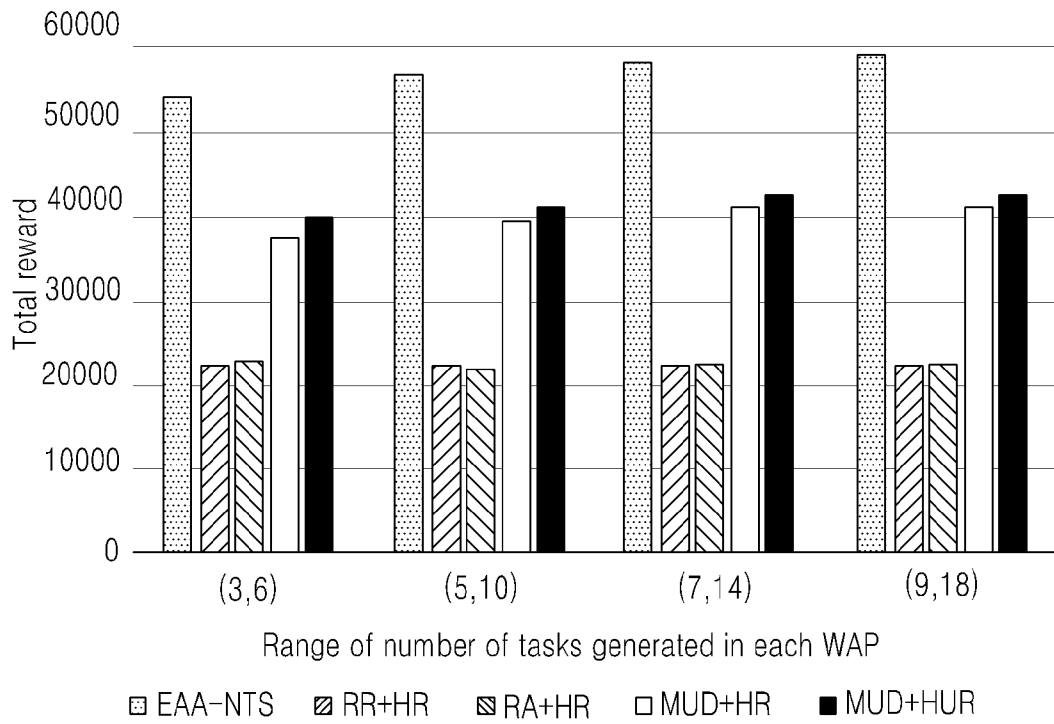
[도9]



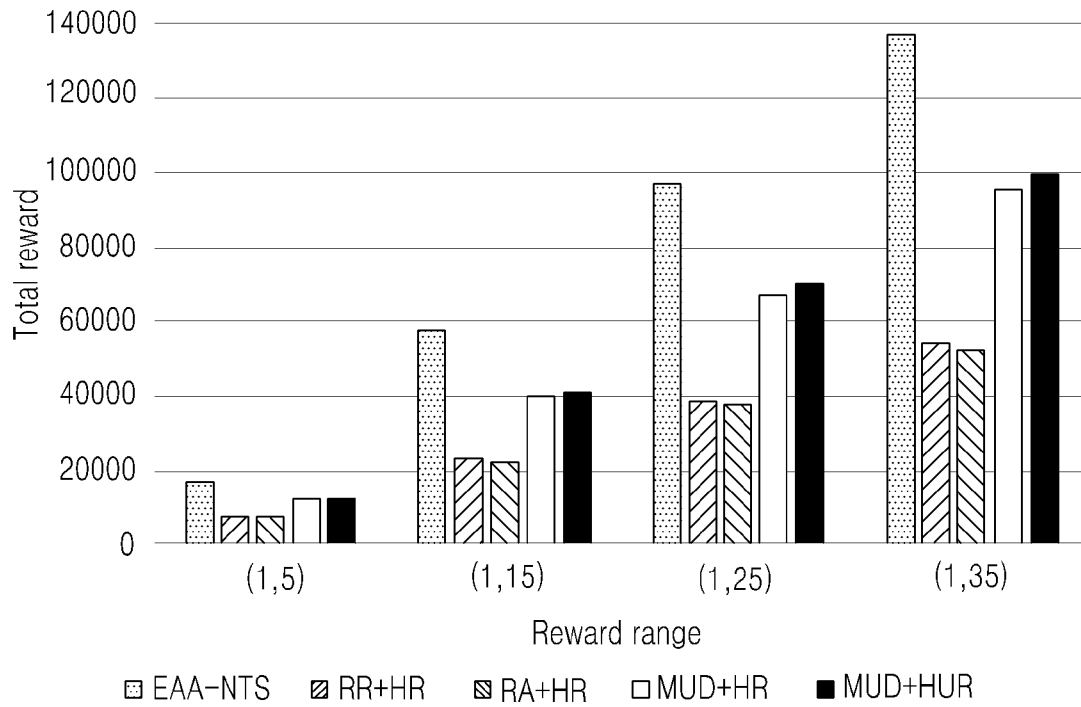
[도10]



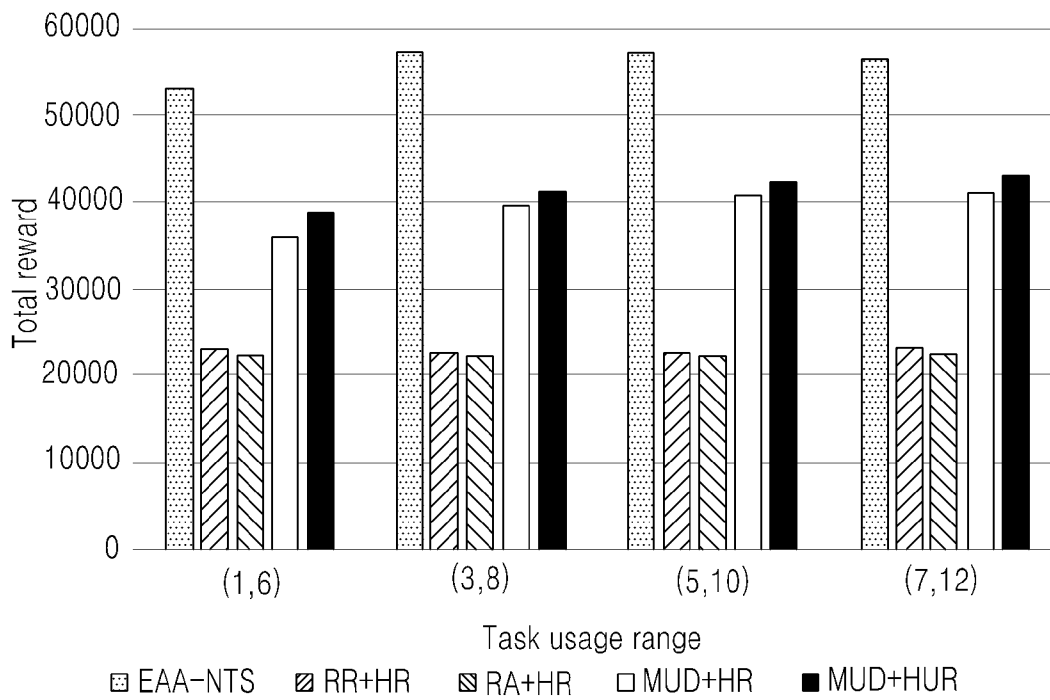
[도11]



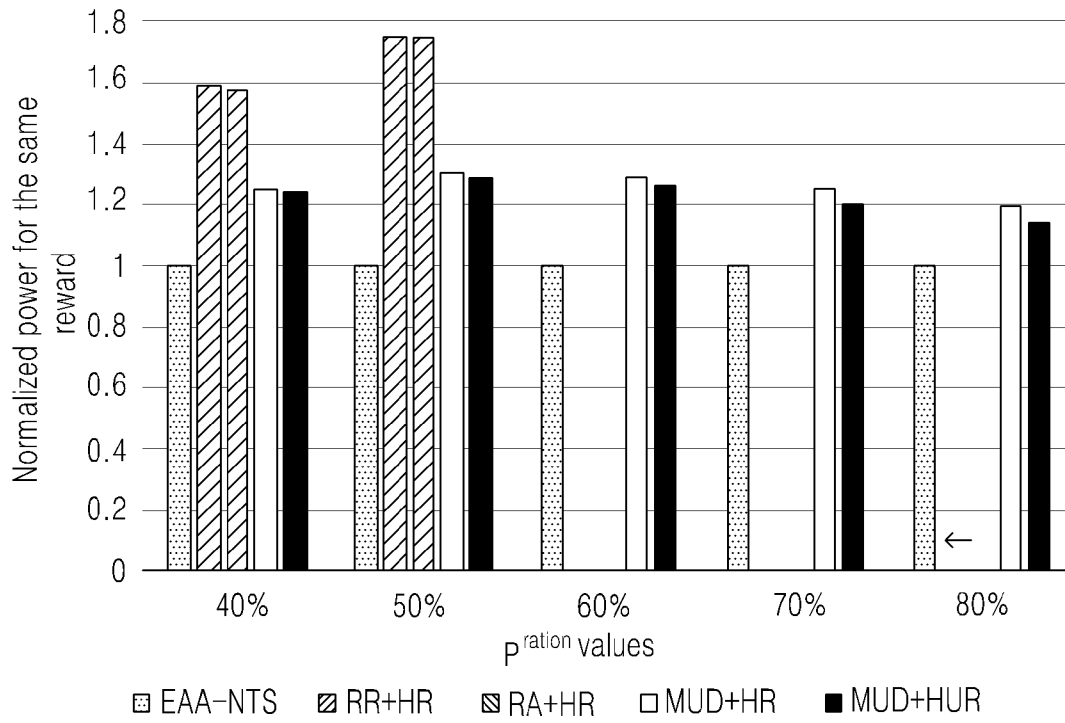
[도12]



[도13]



[도14]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2020/017849

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/50(2006.01)i; H04L 29/08(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/50(2006.01); H04L 12/923(2013.01); H04L 29/08(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models: IPC as above

Japanese utility models and applications for utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & keywords: 엣지 서버(edge server), 오프로딩(offloading), 보상(reward), 소모 전력(power consumption), 작업 할당(task allocation)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2020-0076875 A1 (INTEL IP CORPORATION) 05 March 2020 (2020-03-05) See paragraphs [0149]-[0151].	1,4,6,10
Y		2,3
A		5,7-9
Y	YUAN, Haitao et al. Profit-Maximized Collaborative Computation Offloading and Resource Allocation in Distributed Cloud and Edge Computing Systems. In: IEEE Transactions on Automation Science and Engineering, pp. 1-11, 14 July 2020, E-ISSN: 1558-3783. See page 1.	2,3
A	WO 2020-023115 A1 (FUTUREWEI TECHNOLOGIES, INC.) 30 January 2020 (2020-01-30) See paragraphs [0036]-[0136] and claims 1-20.	1-10
A	US 2019-0141120 A1 (INTEL CORPORATION) 09 May 2019 (2019-05-09) See paragraphs [0012]-[0039] and claims 1-20.	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

24 June 2021

Date of mailing of the international search report

25 June 2021

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon Building 4, 189 Cheongsaro, Seo-gu, Daejeon 35208

Facsimile No. +82-42-481-8578

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2020/017849

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2020-0109272 A (SAMSUNG ELECTRONICS CO., LTD.) 22 September 2020 (2020-09-22) See paragraphs [0019]-[0098] and claims 1-20.	1-10

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2020/017849

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)		Publication date (day/month/year)
US	2020-0076875	A1	05 March 2020	US	10440096 B2	08 October 2019
				US	2018-0183855 A1	28 June 2018
WO	2020-023115	A1	30 January 2020	US	2021-0136142 A1	06 May 2021
US	2019-0141120	A1	09 May 2019	None		
KR	10-2020-0109272	A	22 September 2020	US	2020-0296609 A1	17 September 2020
				WO	2020-185000 A1	17 September 2020

A. 발명이 속하는 기술분류(국제특허분류(IPC)) G06F 9/50(2006.01)i; H04L 29/08(2006.01)i		
B. 조사된 분야 조사된 최소문헌(국제특허분류를 기재) G06F 9/50(2006.01); H04L 12/923(2013.01); H04L 29/08(2006.01) 조사된 기술분야에 속하는 최소문헌 이외의 문헌 한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC 일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC 국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우)) eKOMPASS(특허청 내부 검색시스템) & 키워드: 엣지 서버(edge server), 오프로딩(offloading), 보상(reward), 소모 전력(power consumption), 작업 할당(task allocation)		
C. 관련 문헌		
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
X	US 2020-0076875 A1 (INTEL IP CORPORATION) 2020.03.05 단락 [0149]-[0151]	1,4,6,10
Y		2,3
A		5,7-9
Y	HAITAO YUAN 등. Profit-Maximized Collaborative Computation Offloading and Resource Allocation in Distributed Cloud and Edge Computing Systems, In: IEEE Transactions on Automation Science and Engineering, pp. 1-11, 2020.07.14, E-ISSN: 1558-3783 페이지 1	2,3
A	WO 2020-023115 A1 (FUTUREWEI TECHNOLOGIES, INC.) 2020.01.30 단락 [0036]-[0136] 및 청구항 1-20	1-10
A	US 2019-0141120 A1 (INTEL CORPORATION) 2019.05.09 단락 [0012]-[0039] 및 청구항 1-20	1-10
☒ 추가 문헌이 C(계속)에 기재되어 있습니다. ☒ 대응특허에 관한 별지를 참조하십시오.		
* 인용된 문헌의 특별 카테고리: “A” 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌 “D” 본 국제출원에서 출원인이 인용한 문헌 “E” 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌 “L” 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌 “O” 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌 “P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌		
“T” 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌 “X” 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다. “Y” 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다. “&” 동일한 대응특허문헌에 속하는 문헌		
국제조사의 실제 완료일 2021년06월24일(24.06.2021)		국제조사보고서 발송일 2021년06월25일(25.06.2021)
ISA/KR의 명칭 및 우편주소 대한민국 특허청 (35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사) 팩스 번호 +82-42-481-8578		심사관 양정록 전화번호 +82-42-481-5709

국 제 조 사 보 고 서
대응특허에 관한 정보

국제출원번호

PCT/KR2020/017849

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
US 2020-0076875 A1	2020/03/05	US 10440096 B2	2019/10/08
		US 2018-0183855 A1	2018/06/28
WO 2020-023115 A1	2020/01/30	US 2021-0136142 A1	2021/05/06
US 2019-0141120 A1	2019/05/09	없음	
KR 10-2020-0109272 A	2020/09/22	US 2020-0296609 A1	2020/09/17
		WO 2020-185000 A1	2020/09/17