

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국



(10) 국제공개번호

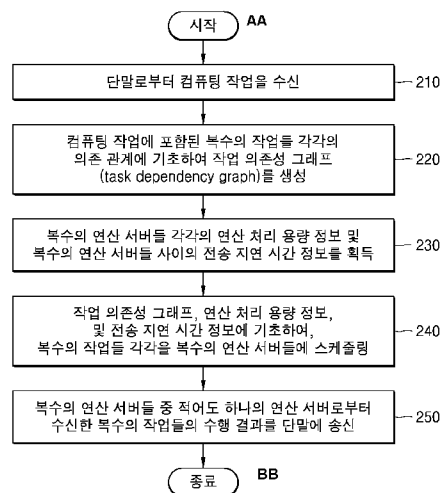
(43) 국제공개일
2020 년 5 월 14 일 (14.05.2020) WIPO | PCT

WO 2020/096239 A1

- (51) 국제특허분류: **G06F 9/48** (2006.01) **G06F 9/455** (2006.01) **BUSINESS FOUNDATION**) [KR/KR]; 02841 서울시 성북구 안암로 145, Seoul (KR).
G06F 9/50 (2006.01)
- (21) 국제출원번호: PCT/KR2019/014040
- (22) 국제출원일: 2019 년 10 월 24 일 (24.10.2019)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보: 10-2018-0135343 2018 년 11 월 6 일 (06.11.2018) KR
- (71) 출원인: 삼성전자 주식회사 (SAMSUNG ELECTRONICS CO., LTD.) [KR/KR]; 16677 경기도 수원시 영통구 삼성로 129, Gyeonggi-do (KR). 고려대학교 산학협력단 (KOREA UNIVERSITY RESEARCH AND
- (72) 발명자: 권기석 (KWEON, Kisuk); 16677 경기도 수원시 영통구 삼성로 129, Gyeonggi-do (KR). 고한열 (KO, Haneul); 02841 서울시 성북구 안암로 145, Seoul (KR). 백상현 (PACK, Sangheon); 02841 서울시 성북구 안암로 145, Seoul (KR). 이재욱 (LEE, Jaewook); 02841 서울시 성북구 안암로 145, Seoul (KR). 김준우 (KIM, Joon-woo); 02841 서울시 성북구 안암로 145, Seoul (KR). 태유진 (TAE, Yujin); 02841 서울시 성북구 안암로 145, Seoul (KR).
- (74) 대리인: 리앤록 특허법인 (Y.P.LEE,MOCK & PARTNERS); 06292 서울시 강남구 언주로30길 13 대림아크로텔 12층, Seoul (KR).
- (81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT,

(54) Title: METHOD AND DEVICE FOR ARRANGING SCHEDULE FOR COMPUTING TASK TO SERVER ON BASIS OF TASK DEPENDENCY

(54) 발명의 명칭: 작업 의존성에 기초하여 컴퓨팅 작업을 서버에 스케줄링하는 방법 및 장치



210 ... Receive computing task from terminal
220 ... Generate task dependency graph on basis of dependency relations between multiple tasks included in computing task
230 ... Acquire information about operation process capacity of each of multiple operation servers and time information about delays in transmission between multiple operation servers
240 ... Arrange schedules for multiple tasks to multiple operation servers on basis of task dependency graph, information about operation process capacities, and time information about delays in transmission
250 ... Transmit, to terminal, results of performance of multiple tasks, received from at least one of multiple operation servers
AA ... Start
BB ... End

(57) Abstract: The present disclosure relates to: a method for arranging a schedule for a computing task comprising multiple tasks to multiple operation servers; and a control server. The control server according to certain embodiments comprises a transceiver unit, a memory, and at least one processor performing control to: receive a computing task from a terminal; generate a task dependency graph on the basis of dependency relations between multiple tasks included in the computing task; acquire information about the process operation capacity of each of multiple operation servers and time information about delays in transmission between the multiple operation servers; arrange schedules for the multiple tasks to the multiple operation servers on the basis of the task dependency graph, the information about operation process capacities, and the time information about delays in transmission; and transmit, to the terminal,

[다음 쪽 계속]

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, 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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 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))

results of the performance of the multiple tasks, received from at least one of the multiple operation servers.

(57) 요약서: 본 개시는 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 복수의 연산 서버들에 스케줄링하는 방법 및 제어 서버에 관한 것으로, 일부 실시예에 따른 제어 서버는, 송수신부; 메모리; 및 단말로부터 컴퓨팅 작업을 수신하고, 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하고, 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하고, 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신하도록 제어하는 적어도 하나의 프로세서를 포함한다.

명세서

발명의 명칭: 작업 의존성에 기초하여 컴퓨팅 작업을 서버에 스케줄링하는 방법 및 장치

기술분야

- [1] 본 개시의 다양한 실시예들은 무선통신시스템에 대한 것으로서, 보다 구체적으로는 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 복수의 연산 서버들에 스케줄링하는 장치 및 그 동작 방법에 관한 것이다.

배경기술

- [2] 4G 통신 시스템 상용화 이후 증가 추세에 있는 무선 데이터 트래픽 수요를 충족시키기 위해, 개선된 5G 통신 시스템 또는 pre-5G 통신 시스템을 개발하기 위한 노력이 이루어지고 있다. 이러한 이유로, 5G 통신 시스템 또는 pre-5G 통신 시스템은 4G 네트워크 이후 (Beyond 4G Network) 통신 시스템 또는 LTE 시스템 이후 (Post LTE) 이후의 시스템이라 불리어지고 있다. 높은 데이터 전송률을 달성하기 위해, 5G 통신 시스템은 초고주파(mmWave) 대역 (예를 들어, 60기가(60GHz) 대역과 같은)에서의 구현이 고려되고 있다. 초고주파 대역에서의 전파의 경로손실 완화 및 전파의 전달 거리를 증가시키기 위해, 5G 통신 시스템에서는 빔포밍(beamforming), 거대 배열 다중 입출력(massive MIMO), 전차원 다중입출력(Full Dimensional MIMO: FD-MIMO), 어레이 안테나(array antenna), 아날로그 빔형성(analog beam-forming), 및 대규모 안테나 (large scale antenna) 기술들이 논의되고 있다. 또한 시스템의 네트워크 개선을 위해, 5G 통신 시스템에서는 진화된 소형 셀, 개선된 소형 셀 (advanced small cell), 클라우드 무선 액세스 네트워크 (cloud radio access network: cloud RAN), 초고밀도 네트워크 (ultra-dense network), 기기 간 통신 (Device to Device communication: D2D), 무선 백홀 (wireless backhaul), 이동 네트워크 (moving network), 협력 통신 (cooperative communication), CoMP (Coordinated Multi-Points), 및 수신 간섭제거 (interference cancellation) 등의 기술 개발이 이루어지고 있다. 이 밖에도, 5G 시스템에서는 진보된 코딩 변조(Advanced Coding Modulation: ACM) 방식인 FQAM (Hybrid FSK and QAM Modulation) 및 SWSC (Sliding Window Superposition Coding)과, 진보된 접속 기술인 FBMC(Filter Bank Multi Carrier), NOMA(non-orthogonal multiple access), 및 SCMA(sparse code multiple access) 등이 개발되고 있다.
- [3] 한편, 인터넷은 인간이 정보를 생성하고 소비하는 인간 중심의 연결 망에서, 사물 등 분산된 구성 요소들 간에 정보를 주고 받아 처리하는 IoT(Internet of Things, 사물인터넷) 망으로 진화하고 있다. 클라우드 서버 등과의 연결을 통한 빅데이터(Big data) 처리 기술 등이 IoT 기술에 결합된 IoE (Internet of Everything) 기술도 대두되고 있다. IoT를 구현하기 위해서, 센싱 기술, 유무선 통신 및

네트워크 인프라, 서비스 인터페이스 기술, 및 보안 기술과 같은 기술 요소들이 요구되어, 최근에는 사물간의 연결을 위한 센서 네트워크(sensor network), 사물 통신(Machine to Machine, M2M), MTC(Machine Type Communication)등의 기술이 연구되고 있다. IoT 환경에서는 연결된 사물들에서 생성된 데이터를 수집, 분석하여 인간의 삶에 새로운 가치를 창출하는 지능형 IT(Internet Technology) 서비스가 제공될 수 있다. IoT는 기존의 IT(information technology)기술과 다양한 산업 간의 융합 및 복합을 통하여 스마트홈, 스마트 빌딩, 스마트 시티, 스마트 카 혹은 커넥티드 카, 스마트 그리드, 헬스케어, 스마트 가전, 첨단의료서비스 등의 분야에 응용될 수 있다.

- [4] 이에, 5G 통신 시스템을 IoT 망에 적용하기 위한 다양한 시도들이 이루어지고 있다. 예를 들어, 센서 네트워크(sensor network), 사물 통신(Machine to Machine, M2M), MTC(Machine Type Communication)등의 기술이 5G 통신 기술인 빔 포밍, MIMO 및 어레이 안테나 등의 기법에 의해 구현되고 있는 것이다. 앞서 설명한 빅데이터 처리 기술로써 클라우드 무선 액세스 네트워크(cloud RAN)가 적용되는 것도 5G 기술과 IoT 기술 융합의 일 예라고 할 수 있을 것이다.
- [5] 상술한 것과 이동통신 시스템의 발전에 따라 다양한 서비스를 제공할 수 있게 됨으로써, 이러한 서비스들을 원활하게 제공하기 위한 방안이 요구되고 있다.
- [6] 초저지연(Ultra Low Latency) 서비스 제공을 위하여 3GPP에서는 라디오 Radio Frame 구조를 변경하여 TTI(Transmission Time Interval)를 1ms 이하로 정의하고, HARQ 기법을 조정하고, 최초 접속 절차 및 스케줄링을 개선하는 등의 방법을 통해 통신 지연 시간을 줄이는 표준화를 진행하고 있다. ETSI(European Telecommunication Standards Institute)에서 진행하는 모바일 에지 컴퓨팅 또는 다중 액세스 에지 컴퓨팅(MEC: Mobile Edge computing 또는 Multi-access Edge Computing) 기술은 초저지연 서비스 실현을 위한 효과적인 기술로 전망되고 있다.

발명의 상세한 설명

과제 해결 수단

- [7] 본 개시의 다양한 실시예들은 무선통신시스템에서, 컴퓨팅 작업에 포함된 작업들 각각의 의존 관계, 연산 서버들 각각의 연산 처리 용량 정보, 및 연산 서버들 사이의 전송 지연 시간 정보에 기초하여 복수의 작업들을 포함하는 컴퓨팅 작업을 복수의 연산 서버들에 스케줄링함으로써 컴퓨팅 작업의 수행 시간을 감소시키기 위한, 무선통신시스템에서 신호를 송수신하는 방법 및 장치를 제공할 수 있다.

도면의 간단한 설명

- [8] 도 1은 일부 실시예에 따른 제어 서버 및 단말의 동작을 설명하기 위한 개념도이다.
- [9] 도 2는 일부 실시예에 따른 제어 서버가 컴퓨팅 작업을 연산 서버들에

스케줄링하는 방법을 설명하기 위한 흐름도이다.

[10] 도 3은 일부 실시예에 따른 작업 의존성 그래프를 설명하기 위한 도면이다.

[11] 도 4는 일부 실시예에 따른 제어 서버가 작업들의 연산 복잡도를 조정하여 펼쳐진(spread) 작업 의존성 그래프를 생성하는 방법을 설명하기 위한 도면이다.

[12] 도 5는 일부 실시예에 따른 제어 서버가 컴퓨팅 작업을 연산 서버들에 스케줄링하는 방법을 설명하기 위한 순서도이다.

[13] 도 6은 일부 실시예에 따른 제어 서버, 단말, 및 연산 서버들이 신호를 송수신하는 방법을 설명하기 위한 도면이다.

[14] 도 7은 일부 실시예에 따른 제어 서버가 컴퓨팅 작업을 연산 서버들에 할당하고 각 작업이 수행되는 시간을 결정하는 방법을 설명하기 위한 흐름도이다.

[15] 도 8은 일부 실시예에 따른 제어 서버가 복수의 작업들이 수행되는 시간을 결정하는 방법을 설명하기 위한 도면이다.

[16] 도 9는 일부 실시예에 따른 단말이 제어 서버에 컴퓨팅 작업을 오프로딩하는 방법을 설명하기 위한 흐름도이다.

[17] 도 10은 일부 실시예에 따른 제어 서버의 블록도이다.

[18] 도 11은 일부 실시예에 따른 단말의 블록도이다.

발명의 실시를 위한 최선의 형태

[19] 일부 실시예에 따라 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 복수의 연산 서버들에 스케줄링하는 제어 서버는, 송수신부; 메모리; 및 단말로부터 컴퓨팅 작업을 수신하고, 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하고, 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하고, 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신하도록 제어하는 적어도 하나의 프로세서를 포함한다.

[20] 일부 실시예에 따른 프로세서는, 단말로부터 컴퓨팅 작업을 수신하고, 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하고, 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하고, 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신하도록 제어할 수 있다.

[21] 일부 실시예에 따른 프로세서는, 복수의 작업들 각각의 연산 복잡도를 식별하고, 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속

작업(sub-task)으로 변환함으로써, 복수의 작업들 각각의 연산 복잡도를 조정하고, 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프를 생성할 수 있다.

- [22] 일부 실시예에 따른 프로세서는, 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우, 제1 작업이 할당된 제1 연산 서버에 제2 작업을 할당하고, 제1 연산 서버에 제2 작업을 할당할 수 없는 경우, 전송 지연 시간 정보에 기초하여 제2 작업을 할당할 수 있는 연산 서버 중 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 제2 작업을 할당할 수 있다.
- [23] 일부 실시예에 따른 프로세서는, 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우, 제1 작업이 할당된 제1 연산 서버에 제2 작업을 할당하고, 제1 연산 서버에 제2 작업을 할당할 수 없는 경우, 전송 지연 시간 정보에 기초하여 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 제2 작업을 할당하고, 제2 연산 서버에 제2 작업을 할당할 수 없는 경우, 연산 처리 용량 정보에 기초하여 복수의 연산 서버들 중 가장 큰 연산 처리 용량을 가진 제3 연산 서버에 제2 작업을 할당할 수 있다.
- [24] 일부 실시예에 따르면, 제어 서버는 복수의 연산 서버들 각각은 복수의 VM(virtual machine)들을 포함하고, 프로세서는, 복수의 작업들 각각을 복수의 연산 서버들에 할당하고 복수의 작업들 각각이 수행되는 시간을 결정함으로써 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 제1 연산 서버의 제1 VM에 할당된 제1 작업 및 제2 작업에 대해, 제1 작업이 수행되는 시간과 제2 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 제2 작업을 제1 연산 서버의 제2 VM에 할당하고 제2 작업이 수행되는 시간을 결정하거나, 제1 작업이 수행된 후 제2 작업이 수행되도록 제2 작업이 수행되는 시간을 결정할 수 있다.
- [25] 일부 실시예에 따른 프로세서는, 복수의 작업들 각각을 복수의 연산 서버들에 할당하고 복수의 작업들 각각이 수행되는 시간을 결정함으로써 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 복수의 작업들 각각의 연산 복잡도를 식별하고, 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 복수의 작업들 각각의 연산 복잡도를 조정하고, 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 따라, 펼쳐진(spread) 작업 의존성 그래프를 생성하고, 펼쳐진 작업 의존성 그래프에 기초하여 복수의 작업들 각각을 복수의 연산 서버들에 할당하고, 연산 복잡도가 조정되지 않은 복수의 작업들 각각의 의존 관계에 기초하여 생성된 작업 의존성 그래프에 기초하여 복수의 작업들 각각이 수행되는 시간을 결정할 수 있다.
- [26] 일부 실시예에 따른 제어 서버가 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 복수의 연산 서버들에 스케줄링하는 방법은, 단말로부터 컴퓨팅 작업을 수신하는 단계; 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하는 단계; 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송

지연 시간 정보를 획득하는 단계; 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하는 단계; 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신하는 단계를 포함한다.

- [27] 일부 실시예에 따라 제어 서버에 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 오프로딩하는 단말은, 송수신부; 메모리; 및 제어 서버에 컴퓨팅 작업을 송신하고, 제어 서버 또는 적어도 하나의 연산 서버로부터 복수의 작업들의 수행 결과를 수신하도록 제어하는 적어도 하나의 프로세서를 포함한다. 전송한 복수의 작업들의 수행 결과는, 복수의 작업들이 복수의 작업들 각각의 의존 관계에 기초하여 생성된 작업 의존성 그래프(task dependency graph), 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보에 기초하여 스케줄링된 복수의 연산 서버로부터 복수의 작업들이 수행된 결과이다.

발명의 실시를 위한 형태

- [28] LTE 이후의 향후 통신 시스템으로서, 5G 통신시스템(또는 NR)은 사용자 및 서비스 제공자 등의 다양한 요구 사항을 자유롭게 반영할 수 있어야 하기 때문에 다양한 요구사항을 만족하는 서비스가 지원되어야 한다. 5G 통신시스템을 위해 고려되는 서비스로는 증가된 모바일 광대역 통신(enhanced Mobile Broadband: eMBB), 대규모 기계형 통신(massive machine type communication: mMTC), 초신뢰 저지연 통신(Ultra Reliability Low Latency Communication: URLLC) 등이 있다.
- [29] 이하 본 개시는 무선통신시스템에서 단말이 기지국으로부터 방송 정보를 수신하기 위한 기술에 대해 설명한다. 본 개시는 4G 시스템 또는 4G 시스템 이후 보다 높은 데이터 송신률을 지원하기 위한 5G 통신 시스템을 IoT 기술과 융합하는 통신 기법 및 그 시스템에 관한 것이다. 본 개시는 5G 통신 기술 및 IoT 관련 기술을 기반으로 지능형 서비스(예를 들어, 스마트 홈, 스마트 빌딩, 스마트 시티, 스마트 카 또는 커넥티드 카, 헬스케어, 디지털 교육, 소매업, 보안 및 안전 관련 서비스 등)에 적용될 수 있다.
- [30] 이하, 본 개시의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [31] 실시예를 설명함에 있어서 본 개시가 속하는 기술 분야에 익히 알려져 있고 본 개시와 직접적으로 관련이 없는 기술 내용에 대해서는 설명을 생략한다. 이는 불필요한 설명을 생략함으로써 본 개시의 요지를 흐리지 않고 더욱 명확히 전달하기 위함이다.
- [32] 마찬가지로 이유로 첨부된 도면에 있어서 일부 구성요소는 과장되거나 생략되거나 개략적으로 도시되었다. 또한, 각 구성요소의 크기는 실제 크기를 전적으로 반영하는 것이 아니다. 각 도면에서 동일한 또는 대응하는 구성 요소에는 동일한 참조 번호를 부여하였다.
- [33] 본 개시의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과

함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 개시는 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 개시의 개시가 완전하도록 하고, 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 개시는 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.

- [34] 이때, 처리 흐름도 도면들의 각 블록과 흐름도 도면들의 조합들은 컴퓨터 프로그램 인스트럭션들에 의해 수행될 수 있음을 이해할 수 있을 것이다. 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서를 통해 수행되는 그 인스트럭션들이 흐름도 블록(들)에서 설명된 기능들을 수행하는 수단을 생성하게 된다. 이들 컴퓨터 프로그램 인스트럭션들은 특정 방식으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 흐름도 블록(들)에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다. 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 흐름도 블록(들)에서 설명된 기능들을 실행하기 위한 단계들을 제공하는 것도 가능하다.
- [35] 또한, 각 블록은 특정된 논리적 기능(들)을 실행하기 위한 하나 이상의 실행 가능한 인스트럭션들을 포함하는 모듈, 세그먼트 또는 코드의 일부를 나타낼 수 있다. 또, 몇 가지 대체 실행 예들에서는 블록들에서 언급된 기능들이 순서를 벗어나서 발생하는 것도 가능함을 주목해야 한다. 예컨대, 잇달아 도시되어 있는 두 개의 블록들은 사실 실질적으로 동시에 수행되는 것도 가능하고 또는 그 블록들이 때때로 해당하는 기능에 따라 역순으로 수행되는 것도 가능하다.
- [36] 이때, 본 실시예에서 사용되는 '~부'라는 용어는 소프트웨어 또는 FPGA(Field Programmable Gate Array) 또는 ASIC(Application Specific Integrated Circuit)과 같은 하드웨어 구성요소를 의미하며, '~부'는 어떤 역할들을 수행한다. 그렇지만 '~부'는 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. '~부'는 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 '~부'는 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들

및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들, 및 변수들을 포함한다. 구성요소들과 '~부'들 안에서 제공되는 기능은 더 작은 수의 구성요소들 및 '~부'들로 결합되거나 추가적인 구성요소들과 '~부'들로 더 분리될 수 있다. 뿐만 아니라, 구성요소들 및 '~부'들은 디바이스 또는 보안 멀티미디어카드 내의 하나 또는 그 이상의 CPU들을 재생시키도록 구현될 수도 있다. 또한 실시예에서 '~부'는 하나 이상의 프로세서를 포함할 수 있다.

- [37] 도 1은 일부 실시예에 따른 제어 서버(110) 및 단말(120)의 동작을 설명하기 위한 개념도이다.
- [38] ETSI에서 진행하는 모바일 에지 컴퓨팅 또는 다중 액세스 에지 컴퓨팅(MEC: Mobile Edge computing 또는 Multi-access Edge Computing, 이하 MEC) 기술은 초저지연 서비스 실현을 위한 기술이다.
- [39] MEC는 모바일 네트워크의 가장자리, 무선 액세스 네트워크(RAN)의 내부, 및 모바일 가입자들과 근접한 곳 등에 IT 서비스 환경 및 클라우드 컴퓨팅 능력을 제공하기 위해 개발되었다. MEC의 목표는 지연 시간(latency)을 줄이고, 고효율적인 네트워크 운영 및 서비스 제공을 보장하고, 향상된 사용자 경험을 제공하는 것이다. MEC의 액세스는 와이파이(wifi) 및 고정 네트워크들과 같은 셀룰러 이외의 다른 유형의 액세스 또한 포함할 수 있다.
- [40] 한편, MEC가 실제 네트워크에 초저지연 서비스의 요구사항을 만족시키기 위해서는 이동성 지원, 자원 관리 등의 해결해야 하는 여러 쟁점들이 존재한다. 특히 최근에는 모바일 디바이스의 부족한 성능을 보완 하기 위해 모바일 디바이스가 수행할 연산을 MEC의 에지 클라우드로 오프로딩하는 기법들이 많이 연구되고 있다. 하지만 MEC의 에지 클라우드가 일반적으로 제한된 컴퓨팅 자원을 가지고 있으며 미래의 초저지연 서비스(예를 들어, AR, VR, 홀로그램 등)들이 복잡한 연산을 요구한다는 점에서, 낮은 처리 시간을 보장하며 에지 클라우드를 이용해 오프로딩 서비스를 제공하기 위한 기법이 필요하다.
- [41] 개시된 다양한 실시예는 제어 서버가 단말로부터 오프로딩 서비스 요청을 받아 복수의 연산 서버들로 컴퓨팅 작업을 분산적으로 병렬 처리하는 방법 및 장치를 제공한다.
- [42] 도 1을 참조하면, 단말(120)은 단말(120)이 수행하기 어려운 컴퓨팅 작업에 대해 제어 서버(110)로 오프로딩 요청을 할 수 있다. 제어 서버(110)는 단말(120)로부터 오프로딩 요청 및 복수의 작업을 포함하는 컴퓨팅 작업을 수신하는 것에 대응하여, 복수의 연산 서버들(130)에 작업을 스케줄링하여 전달할 수 있다. 복수의 연산 서버들(130)은 스케줄링에 따라 작업을 수행하고 수행한 결과를 제어 서버(110) 또는 단말(120)로 전달할 수 있다.
- [43] 제어 서버(110)는 스케줄링을 수행하는 다양한 서버일 수 있다. 예를 들어, 제어 서버(110)는 기지국(eNB 또는 gNB)일 수 있고, 스케줄링을 수행하는 데이터

센터일 수도 있다. 물론 전술한 예시에 한정되지 않고 제어 서버는 다양한 서버를 포함할 수 있다.

- [44] 복수의 연산 서버들(130) 각각은 연산 또는 작업을 수행하는 다양한 서버일 수 있다. 예를 들어, 복수의 연산 서버들(130) 각각은 기지국(eNB 또는 gNB)일 수 있고, AP(Access Point)일 수도 있고, 데이터 센터일 수도 있다. 또한, 복수의 연산 서버들(130) 각각은 단말(120) 부근에 설치된 에지 클라우드 장치일 수 있다. 물론 전술한 예시에 한정되지 않고 연산 서버는 다양한 서버를 포함할 수 있다.
- [45] 복수의 연산 서버들(130)에 포함된 각 연산 서버는 적어도 하나의 VM을 구동할 수 있고, 적어도 하나의 프로세서 및 VM을 구동하기 위한 하이퍼바이저를 포함할 수 있다. 각 연산 서버는 다른 연산 서버와 유선 또는 무선으로 신호를 송수신할 수 있다. 예를 들어, 제1 연산 서버(131), 제2 연산 서버(132), 제3 연산 서버(133)는 유무선으로 작업 또는 작업 수행 결과를 송수신할 수 있다. 복수의 연산 서버들(130)에 포함된 연산 서버의 수는 물론 도 1에 도시된 바에 의해 한정되지 않고 다양하게 정해질 수 있다.
- [46] 제어 서버(110)는 단말(120)의 위치에 기초하여 단말(120) 근처에 위치하는 복수의 연산 서버들(130)을 에지 클러스터로 설정할 수 있다. 단말(120)은 제어 서버(110)에 컴퓨팅 작업의 오프로딩 요청을 함으로써, 단말(120) 근처에 위치한 에지 클러스터에서 수행된 작업 결과를 수신할 수 있다.
- [47] 도 2는 일부 실시예에 따른 제어 서버가 컴퓨팅 작업을 연산 서버들에 스케줄링하는 방법을 설명하기 위한 흐름도이다.
- [48] 210 단계에서, 제어 서버는 단말로부터 컴퓨팅 작업을 수신할 수 있다.
- [49] 단말은 복수의 작업들을 포함하는 컴퓨팅 작업을 제어 서버에 송신할 수 있다. 단말은 컴퓨팅 작업과 함께 컴퓨팅 작업에 대한 오프로딩 요청을 제어 서버에 송신할 수 있다.
- [50] 컴퓨팅 작업의 종류는 다양하게 정해질 수 있다. 예를 들어, 컴퓨팅 작업은 단말에 설치된 어플리케이션에서 제공되는 서비스가 수행되기 위한 연산 집합일 수 있다. 물론 이에 한정되지 않고 컴퓨팅 작업은 다양한 종류의 작업을 포함할 수 있다.
- [51] 220 단계에서, 제어 서버는 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성할 수 있다.
- [52] 작업 의존성 그래프(task dependency graph)는 복수의 작업들의 의존관계를 나타내는 그래프이다. 예를 들어, 제2 작업이 수행되기 위해 반드시 제1 작업이 먼저 수행되어야 한다면, 제2 작업은 제1 작업과 의존관계에 있다고 정의할 수 있다. 또는 제2 작업은 제1 작업과 의존성이 있다고도 정의할 수 있다.
- [53] 예를 들어, 제어 서버는 단말로부터 받은 컴퓨팅 작업을 분석함으로써 컴퓨팅 작업에 포함된 복수의 작업들 사이의 의존관계를 파악하고, 파악한 의존관계에 기초하여 작업 의존성 그래프를 생성할 수 있다.
- [54] 일부 실시예에 따른 제어 서버는 복수의 작업들 각각의 연산 복잡도가 조정된,

펼쳐진(spread) 작업 의존성 그래프를 생성할 수 있다. 제어 서버는 펼쳐진 작업 의존성 그래프에 기초하여 복수의 작업들 각각을 복수의 연산 서버에 할당할 수 있다.

- [55] 230 단계에서, 제어 서버는 복수의 연산 서버들 각각의 연산 처리 용량 정보(processing capacity) 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득할 수 있다.
- [56] 연산 처리 용량 정보는 연산 서버가 작업을 수행하는 능력에 대한 정보 또는 연산 서버가 작업을 처리할 수 있는 자원을 현재 얼마나 확보하고 있는지에 대한 정보를 의미할 수 있다. 예를 들어, 연산 처리 용량 정보는 연산 서버가 소정의 시간에 처리할 수 있는 연산의 수, 연산 서버에 시간별 작업이 할당된 스케줄링 정보, 연산 서버에서 작업을 할당 받을 수 있는 시간 구간에 대한 정보 중 적어도 하나를 포함할 수 있다. 예를 들어, 작업을 할당 받을 수 있는 시간 구간은 연산 서버가 아직 작업을 할당 받지 않아 연산 서버의 작업 수행이 가능한 시간 구간일 수 있다.
- [57] 연산 서버 사이의 전송 지연 시간 정보는 서로 다른 제1 연산 서버에서 제2 연산 서버로 신호를 전송하는데 걸리는 시간을 의미할 수 있다. 예를 들어, 전송 지연 시간 정보는 제1 연산 서버의 작업 수행 결과를 제2 연산 서버로 전송하는데 걸리는 시간을 의미할 수 있다.
- [58] 복수의 연산 서버들 각각은 물리적으로 이격되어 위치할 수 있다. 복수의 연산 서버들 각각이 물리적으로 이격되어 위치하는 경우, 연산 서버들 사이에 유선 또는 무선으로 신호를 송수신하는 데에 전송 지연 시간이 발생할 수 있다. 또한 복수의 연산 서버들 중 적어도 일부는 하나의 장치에 설치되어 있을 수 있다. 하나의 장치에 설치된 적어도 일부의 연산 서버들 간에 서로 신호를 송수신할 때에도 전송 지연 시간은 발생할 수 있다.
- [59] 일부 실시예에 따르면, 복수의 연산 서버들 중 적어도 일부는 각각 복수의 가상 머신(VM:Virtual Machine)을 구동할 수 있다. 예를 들어, 하나의 VM은 하나의 시간 구간 동안 하나의 작업을 수행할 수 있다. 복수의 VM을 구동하는 연산 서버는 각각의 VM을 구동하여 작업을 수행함으로써 하나의 시간 구간 동안 복수의 작업을 수행할 수 있다.
- [60] 일부 실시예에 따르면, 연산 서버의 연산 처리 용량 정보는 연산 서버가 구동하는 VM의 개수를 포함할 수 있다. 또한, 연산 처리 용량 정보는 연산 서버에 포함된 각각의 VM이 작업을 수행하는 능력에 대한 정보 또는 각각의 VM이 작업을 처리할 수 있는 자원을 현재 얼마나 확보하고 있는지에 대한 정보를 의미할 수 있다. 예를 들어, 연산 처리 용량 정보는 각각의 VM이 소정의 시간에 처리할 수 있는 연산의 수, 각각의 VM에 시간별 작업이 할당된 스케줄링 정보, 각각의 VM이 작업을 할당 받을 수 있는 시간 구간에 대한 정보 중 적어도 하나를 포함할 수 있다.
- [61] 일부 실시예에 따르면, 제어 서버는 복수의 연산 서버들로부터 복수의 연산

서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 수신할 수 있다. 예를 들어, 제어 서버는 주기적으로 복수의 연산 서버들로부터 연산 처리 용량 정보 및 전송 지연 시간 정보를 수신할 수 있다. 또한 제어 서버는 복수의 연산 서버들로부터 수신한 연산 처리 용량 정보 및 전송 지연 시간 정보를 제어 서버에 저장하고, 저장된 정보를 로딩함으로써 연산 처리 용량 정보 및 전송 지연 시간 정보를 획득할 수도 있다. 또한, 제어 서버는 복수의 연산 서버들로부터 갱신된 연산 처리 용량 정보 또는 갱신된 전송 지연 시간 정보를 수신하는 것에 대응하여 제어 서버에 저장된 연산 처리 용량 정보 및 전송 지연 시간 정보를 갱신할 수 있다. 물론 전송한 방법은 예시일 뿐, 제어 서버가 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하는 방법은 전송한 방법에 한정되지 않는다.

- [62] 240 단계에서, 제어 서버는 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링할 수 있다.
- [63] 제어 서버가 복수의 작업들을 스케줄링 하는 동작은 각 작업을 연산 서버에 할당하는 동작 및 각 작업이 수행되는 시간을 결정하는 동작을 포함할 수 있다.
- [64] 일부 실시예에 따른 제어 서버는 복수의 작업들을 작업 의존성 그래프의 각 스텝 별로 복수의 연산 서버들에 스케줄링할 수 있다. 제어 서버는 어느 하나의 스텝에 포함된 적어도 하나의 작업을 적어도 하나의 연산 서버에 스케줄링할 수 있다.
- [65] 제어 서버는 작업 의존성 그래프 또는 펼쳐진 작업 의존성 그래프에 기초하여, 각 작업을 의존 관계에 따라 할당할 수 있다. 제어 서버가 작업 의존성 그래프의 각 스텝 별로 작업을 의존 관계에 따라 할당하는 방법은 도 5를 참조하여 자세히 설명한다.
- [66] 일부 실시예에 따르면, 제어 서버는 복수의 작업들을 복수의 연산 서버에 할당하는 단계에서 각 작업이 수행되는 시간을 결정할 수 있다. 예를 들어, 제어 서버는 각 작업의 소요 시간 및 전송 지연 시간 정보에 기초하여 각 작업이 수행되는 시간을 결정하거나, 펼쳐진 작업 의존성 그래프의 각 스텝을 기준으로 각 작업이 수행되는 시간을 결정할 수 있다.
- [67] 일부 실시예에 따른 제어 서버는 복수의 작업들이 복수의 연산 서버에 할당된 후 각 작업이 수행되는 시간을 결정할 수 있다. 제어 서버가 각 작업이 수행되는 시간을 결정하는 동작은 도 7을 참조하여 자세히 설명한다.
- [68] 250 단계에서, 제어 서버는 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신할 수 있다.
- [69] 작업 의존성 그래프에서 첫 번째 스텝 및 마지막 스텝은 각각 하나의 작업을 포함할 수 있다. 즉 단말로부터 수신한 컴퓨팅 작업에 포함된 복수의 작업들은 작업 의존성 그래프의 마지막 스텝에서 하나의 작업으로 수렴될 수 있다.
- [70] 마지막 스텝의 작업은 하나의 연산 서버에 할당될 수 있다. 또는, 예를 들어

마지막 스텝의 작업의 연산 처리량이 큰 경우, 마지막 스텝의 작업은 복수의 연산 서버에 할당될 수도 있다.

- [71] 마지막 스텝의 작업이 할당된 적어도 하나의 연산 서버는 작업을 수행한 후 제어 서버로 작업 수행 결과를 송신할 수 있고, 또는 단말로 직접 작업 수행 결과를 송신할 수도 있다. 제어 서버는 적어도 하나의 연산 서버로부터 수신한 작업 수행 결과를 단말에 송신할 수 있다.
- [72] 도 2에 도시된 각 단계가 수행되는 순서는 도2에 도시된 바에 의해 한정되지 않고 다양하게 정해질 수 있다. 예를 들어, 230 단계는 210 단계 이전에 수행되거나, 220단계 이전에 수행될 수도 있다.
- [73] 개시된 다양한 실시예에 따른 제어 서버는, 각 작업의 의존성, 연산 서버들의 연산 처리 용량, 및 연산 서버들 사이의 전송 지연 시간을 고려하여 복수의 작업을 분산 병렬 처리함으로써, 낮은 스케줄링 복잡도를 갖는 동시에 컴퓨팅 작업이 수행되는 완료 시간을 줄일 수 있다. 즉, 제어 서버는 많은 연산량을 요구하는 초저지연 서비스를 낮은 처리 시간과 함께 제공할 수 있다. 예를 들어, 제어 서버는 작업 의존성을 고려하지 않는 Dependency-Unaware 기법 또는 연산 서버의 연산 처리 용량만을 고려하는 Capacity-Based 기법에 비하여 컴퓨팅 작업이 수행되는 완료 시간을 줄일 수 있다.
- [74] 또한, 복수의 연산 서버들이 단말 근처에 위치하는 엣지 서버들인 경우, 제어 서버가 복수의 연산 서버들에 스케줄링된 작업을 송신하는 데에 소요되는 전송 지연 시간, 연산 서버들이 제어 서버 또는 단말에 작업 수행 결과를 송신하는 데에 소요되는 전송 지연 시간 등을 줄일 수 있어, 컴퓨팅 작업이 수행되는 완료 시간을 더욱 줄일 수 있다.
- [75] 도 3은 일부 실시예에 따른 작업 의존성 그래프(300)를 설명하기 위한 도면이다.
- [76] 도 3을 참조하면, 일부 실시예에 따른 제어 서버가 생성한 작업 의존성 그래프(300)는 복수의 스텝들을 포함할 수 있다. 각 스텝은 적어도 하나의 작업을 포함할 수 있다. 즉, 제어 서버는 단말로부터 수신한 전체 컴퓨팅 작업을 복수의 스텝들로 조정하고 각 스텝에 적어도 하나의 작업을 배치함으로써 작업 의존성 그래프(300)를 생성할 수 있다.
- [77] 작업 의존성 그래프(300)에서 원은 1개의 작업 단위를 나타낸다. 작업 단위는 제어 서버가 작업을 용이하게 스케줄링하기 위해 복수의 작업들을 논리적으로 또는 개념적으로 나눈 것일 수 있다. 또는 1개의 작업 단위는 하나의 작업을 의미할 수도 있다.
- [78] 작업 의존성 그래프(300)에서 화살표는 의존성을 나타낸다. 예를 들어, 작업 302 및 작업 303은 각각 작업 301과 의존성을 갖는다. 의존관계는 1:1, 1:n, n:1, m:n 등 다양하게 정해질 수 있다. 각 작업이 갖는 다양한 의존관계의 일 예가 작업 의존성 그래프(300)에서 스텝 s-1과 스텝s에 도시되어 있다.
- [79] 전체 컴퓨팅 작업은 작업 의존성 그래프(300)에서 하나의 작업으로부터

- 시작해서 복수의 작업들로 전개된 후 다시 하나의 작업으로 수렴될 수 있다. 도 3에는 작업 의존성 그래프(300)에서 첫 번째 스텝인 스텝 1의 작업은 작업 301이고 마지막 스텝인 스텝 M의 작업은 작업 304인 일 예가 도시되어 있다.
- [80] 작업 의존성 그래프(300)의 각 작업의 연산 복잡도는 다양할 수 있다. 제어 서버는 각 작업의 연산 복잡도를 조정함으로써 펼쳐진 작업 의존성 그래프를 생성할 수 있다.
- [81] 도 4는 일부 실시예에 따른 제어 서버가 작업들의 연산 복잡도를 조정하여 펼쳐진(spread) 작업 의존성 그래프를 생성하는 방법을 설명하기 위한 도면이다.
- [82] 도 4의 401은 도 3의 작업 의존성 그래프(300)의 스텝 s, s+1, s+2만을 도시한 것이다. 도 4의 401을 참조하면, 작업 의존성 그래프에 포함된 각 작업은 서로 다른 연산 복잡도를 가질 수 있다. 예를 들어, 작업 411은 1의 연산 복잡도, 작업 422는 2의 연산 복잡도를 가질 수 있다.
- [83] 연산 복잡도는 각 작업이 수행되는 데에 필요한 연산량을 의미할 수 있다. 연산량이 크면 작업 소요 시간이 클 수 있다. 복수의 작업을 작업 의존성 그래프의 스텝별로 연산 서버에 할당하는 경우, 각 스텝에 포함된 작업들의 연산 복잡도가 서로 상이하다면, 전체 컴퓨팅 작업 완료 시간이 오래 걸릴 수 있다. 예를 들어, 도 4의 401의 스텝 s에서 연산 복잡도가 1인 작업 411을 수행한 연산 서버 또는 VM은, 연산 복잡도가 2인 작업 412가 수행되기 까지 대기한 후 스텝 s+1의 작업으로 전개될 수 있고, 따라서 대기 시간이 소요된다.
- [84] 일부 실시예에 따른 제어 서버는 적어도 하나의 작업의 연산 복잡도를 조정하여 펼쳐진(spread) 작업 의존성 그래프(402)를 생성함으로써, 복수의 작업을 작업 의존성 그래프의 스텝별로 연산 서버에 할당하는 경우 전체 컴퓨팅 작업 완료 시간을 줄일 수 있다.
- [85] 작업의 연산 복잡도를 조정하는 것은 각 작업의 연산 복잡도를 동일 또는 유사한 정도로 조정하는 것을 의미할 수 있다. 예를 들어, 제어 서버는 기준 연산 복잡도를 1로 설정하고, 모든 작업이 0.8~1.2 사이의 연산 복잡도를 갖도록 연산 복잡도를 조정할 수 있다. 물론 이에 한정되지 않지 않고, 제어 서버는 모든 작업이 일정 범위 사이의 연산 복잡도를 가지도록 조정할 수 있다.
- [86] 일부 실시예에 따른 제어 서버는 연산 복잡도가 큰 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써 복수의 작업들의 연산 복잡도를 조정할 수 있다. 예를 들어, 도 4를 참조하면, 제어 서버는 연산 복잡도가 2인 작업 412를, 각각 연산 복잡도가 1인 부속 작업 421 및 부속 작업 422로 변환함으로써 연산 복잡도가 조정하여 펼쳐진 작업 의존성 그래프(402)를 생성할 수 있다.
- [87] 부속 작업으로 변환된 작업들은 서로 부속 관계를 갖는다. 펼쳐진 작업 의존성 그래프(402)에는 부속 관계 및 부속 작업들이 표시되어 있다. 한편, 부속 관계를 갖는 작업들은 작업 의존성 그래프의 연산 복잡도를 조정하기 위하여 서로 논리적으로 또는 개념적으로 분리된 것일 뿐, 분리되기 전에는 하나의 작업이므로, 같은 연산 서버 또는 같은 VM에서 수행된다.

- [88] 도 5는 일부 실시예에 따른 제어 서버가 컴퓨팅 작업을 연산 서버들에 스케줄링하는 방법을 설명하기 위한 순서도이다.
- [89] 일부 실시예에 따른 제어 서버는 펼쳐진 작업 의존성 그래프의 각 스텝별로, 각 스텝에 포함된 각 작업을 적어도 하나의 연산 서버에 스케줄링할 수 있다. 도 5는 펼쳐진 작업 의존성 그래프의 스텝 s의 작업 k를 연산 서버에 스케줄링하는 방법의 예시를 설명한다.
- [90] 501 단계에서, 제어 서버는 작업 k가 다른 작업과 부속 관계를 갖는지 여부를 판단할 수 있다. 예를 들어, 제어 서버는 스텝 s 이전의 적어도 하나의 스텝 중 작업 k와 부속 관계를 갖는 작업 n이 있는지 판단할 수 있다.
- [91] 502 단계에서, 제어 서버는 부속 관계를 갖는 작업들은 같은 연산 서버의 같은 VM에 할당할 수 있다. 예를 들어, 제어 서버는 작업 k와 부속 관계를 갖는 작업 n이 할당된 연산 서버의 VM에 작업 k를 할당할 수 있다.
- [92] 부속 관계를 갖는 작업들은 작업 의존성 그래프의 연산 복잡도를 조정하기 위하여 펼쳐진 작업 의존성 그래프에서 분리된 것이고, 부속 관계의 특성상 서로 다른 VM에서 수행될 수 없다. 따라서 부속 관계를 갖는 작업들은 같은 연산 서버의 같은 VM에 할당된다.
- [93] 503 단계에서, 제어 서버는 작업 k가 이전 스텝의 작업m에 대해 의존성을 갖는지 여부를 판단할 수 있다. 예를 들어, 제어 서버는 스텝 s 이전의 적어도 하나의 스텝 중 작업 k와 의존성을 갖는 작업 m이 있는지 여부를 판단할 수 있다.
- [94] 또한 제어 서버는 작업 k와 의존성을 갖는 복수의 작업 중 하나의 작업인 m을 선택할 수 있다.
- [95] 펼쳐진 작업 의존성 그래프에 따르면, 이전 스텝의 작업에 대해 의존성을 갖지 않는 작업은 첫 번째 스텝의 첫 번째 작업뿐이다. 첫 번째 작업을 제외한 나머지 작업들은 적어도 하나의 작업에 대해 의존성을 갖는다.
- [96] 504 단계에서, 작업 k가 이전 스텝의 작업m에 대해 의존성을 갖지 않는 경우, 제어 서버는 작업 k를 연산 처리 용량이 가장 큰 연산 서버에 할당할 수 있다. 즉, 작업 k가 첫 번째 스텝의 첫 번째 작업인 경우, 제어 서버는 작업 k를 연산 처리 용량이 가장 큰 연산 서버에 할당한다.
- [97] 예를 들어, 연산 처리 용량이 가장 큰 연산 서버는 복수의 연산 서버들 중 가장 많은 VM을 구동하는 연산 서버 또는 소정의 시간에 처리할 수 있는 연산의 수가 가장 많은 서버일 수 있다.
- [98] 제어 서버는 연산 처리 용량이 가장 큰 연산 서버에 첫 번째 스텝의 첫 번째 작업을 할당함으로써, 이후의 스텝들에서 서로 의존성을 갖는 작업들을 하나의 연산 서버에 할당하는 가능성을 높일 수 있다. 서로 의존성을 갖는 작업들이 하나의 연산 서버에 할당되는 경우 연산 서버간의 전송 지연 없이 서로 의존성을 갖는 작업들을 수행함으로써 컴퓨팅 작업 완료 시간을 줄일 수 있다.
- [99] 505 단계에서, 제어 서버는 작업 m이 할당된 연산 서버에 작업 k를 할당 가능한지 여부를 판단할 수 있다.

- [100] 예를 들어 작업 m 이 할당된 제1 연산 서버의 모든 VM이 스텝 s 에 포함된 작업을 할당 받은 경우, 작업 m 이 할당된 제1 연산 서버에 작업 k 를 할당 불가능할 수 있다.
- [101] 작업 m 이 할당된 제1 연산 서버의 VM들 중 적어도 하나의 VM이 스텝 s 에 포함된 작업을 할당받지 않은 경우, 작업 m 이 할당된 제1 연산 서버에 작업 k 를 할당 가능할 수 있다.
- [102] 506 단계에서, 작업 m 이 할당된 연산 서버에 작업 k 가 할당 가능한 경우, 제어 서버는 전술한 연산 서버에 작업 k 를 할당할 수 있다.
- [103] 예를 들어, 제어 서버는 작업 m 이 할당된 제1 연산 서버에서 스텝 s 에 포함된 작업을 할당받지 않은 적어도 하나의 VM에 작업 k 를 할당할 수 있다.
- [104] 507 단계에서, 작업 m 이 할당된 연산 서버에 작업 k 가 할당 불가능한 경우, 제어 서버는 작업 m 이 할당된 연산 서버로부터 전송 지연 시간이 가장 짧은 연산 서버에 작업 k 를 할당 가능한지 여부를 판단할 수 있다.
- [105] 제어 서버는 복수의 연산 서버들 사이의 전송 지연 시간 정보에 기초하여, 작업 m 이 할당된 제1 연산 서버로부터 전송 지연 시간이 가장 짧은 제2 연산 서버를 식별할 수 있다.
- [106] 예를 들어, 제2 연산 서버의 모든 VM이 스텝 s 에 포함된 작업을 할당 받은 경우, 제2 연산 서버에 작업 k 를 할당 불가능할 수 있다.
- [107] 일부 실시예에 따른 제어 서버는, 제2 연산 서버에 작업 k 를 할당 불가능한 경우, 단계 504로 진행할 수 있다. 즉, 제어 서버는 작업 k 를 연산 처리 용량이 가장 큰 연산 서버에 할당할 수 있다. 제어 서버는 연산 처리 용량이 가장 큰 연산 서버에 작업 k 를 할당함으로써, 이후의 스텝들에서 작업 k 와 의존성을 갖는 작업들을 하나의 연산 서버에 할당하는 가능성을 높일 수 있다.
- [108] 또는, 일부 실시예에 따른 제어 서버는 제2 연산 서버에 작업 k 를 할당 불가능한 경우, 작업 m 이 할당된 제1 연산 서버로부터 짧은 전송 지연 시간을 갖는 순서대로 다른 연산 서버에 작업 k 의 할당을 시도할 수 있다. 즉, 제어 서버는 작업 k 를 할당할 수 있는 연산 서버 중 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제3 연산 서버를 식별하고, 제3 연산 서버에 작업 k 를 할당할 수 있다.
- [109] 508 단계에서, 작업 m 이 할당된 연산 서버로부터 전송 지연 시간이 가장 짧은 연산 서버에 작업 k 를 할당 가능한 경우, 제어 서버는 전술한 연산 서버에 작업 k 를 할당할 수 있다.
- [110] 작업 k 가 어느 하나의 연산 서버에 할당되면 제어 서버는 다음으로 작업 $k+1$ 의 할당하기 위해 도 5에 도시된 단계들을 수행할 수 있다. 스텝 s 의 모든 작업이 적어도 하나의 연산 서버에 할당되면 제어 서버는 다음으로 스텝 $s+1$ 에 포함된 작업을 할당하기 위해 도 5에 도시된 단계들을 수행할 수 있다.
- [111] 개시된 다양한 실시예에 따른 제어 서버는 도 5에 도시된 방법과 같이, 펼쳐진 의존성 그래프에 포함된 각 스텝별로, 각 작업의 의존성, 연산 서버들의 연산 처리 용량, 및 연산 서버들 사이의 전송 지연 시간을 고려하여 복수의 작업을

- 복수의 연산 서버에 할당함으로써, 낮은 스케줄링 복잡도를 갖는 동시에 컴퓨팅 작업이 수행되는 완료 시간을 줄일 수 있다.
- [112] 도 6은 일부 실시예에 따른 제어 서버(110), 단말(120), 및 연산 서버들(130)이 신호를 송수신하는 방법을 설명하기 위한 도면이다.
- [113] 도 6에서 설명할 내용 중 도 2에서 설명한 내용과 중복되는 내용은 간단히 설명하도록 한다.
- [114] 601 단계에서, 제어 서버(110)는 복수의 연산 서버들(130)로부터 제어 서버(110)에 연산 서버의 연산 처리 용량 정보 및 연산 서버들 사이의 전송 지연 시간 정보를 수신할 수 있다.
- [115] 602 단계에서, 제어 서버(110)는 단말(120)로부터 복수의 작업들을 포함하는 컴퓨팅 작업에 대한 오프로딩 요청을 수신할 수 있다.
- [116] 603 단계에서, 제어 서버(110)는 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성할 수 있다.
- [117] 604 단계에서, 제어 서버(110)는 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링할 수 있다.
- [118] 605 단계에서, 제어 서버(110)는 스케줄링에 따라 복수의 작업들을 복수의 연산 서버들(130)에 분배할 수 있다.
- [119] 606 단계에서, 연산 서버들(130)은 스케줄링에 따라 복수의 작업들을 수행할 수 있다.
- [120] 607 단계에서, 적어도 하나의 연산 서버는 복수의 작업들의 수행 결과를 제어 서버(110) 또는 단말(120)에 송신할 수 있다. 전체 컴퓨팅 작업의 마지막 스텝의 작업을 수행한 적어도 하나의 연산 서버는 제어 서버(110)로 작업들의 수행 결과를 송신하거나, 또는 단말(120)로 직접 작업 수행 결과를 송신할 수 있다.
- [121] 608 단계에서, 제어 서버(110)가 적어도 하나의 연산 서버로부터 복수의 작업들의 수행 결과를 수신한 경우, 제어 서버(110)는 수신한 결과를 단말(120)에 송신할 수 있다.
- [122] 도 6의 601 단계, 602 단계, 603 단계, 604 단계, 608 단계의 자세한 내용은 각각 도 2의 230 단계, 210 단계, 220 단계, 240 단계, 250 단계에서 자세히 설명하였다.
- [123] 도 7은 일부 실시예에 따른 제어 서버가 컴퓨팅 작업을 연산 서버들에 할당하고 각 작업이 수행되는 시간을 결정하는 방법을 설명하기 위한 흐름도이다.
- [124] 일부 실시예에 따른 제어 서버는 복수의 작업들이 복수의 연산 서버에 할당된 후 각 작업이 수행되는 시간을 결정할 수 있다. 예를 들어, 제어 서버는 도 5에 도시된 방법에 의해 복수의 작업들이 할당된 후, 각 작업이 수행되는 시간을 결정할 수 있다.
- [125] 도 7에서 설명할 내용 중 도 2에서 설명한 내용과 중복되는 내용은 간단히 설명하도록 한다.

- [126] 710 단계에서, 제어 서버는 단말로부터 컴퓨팅 작업을 수신할 수 있다.
- [127] 720 단계에서, 제어 서버는 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프를 생성할 수 있다.
- [128] 730 단계에서, 제어 서버는 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득할 수 있다.
- [129] 740 단계에서, 제어 서버는 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 할당할 수 있다.
- [130] 750 단계에서, 제어 서버는 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각이 수행되는 시간을 결정할 수 있다.
- [131] 각 작업이 수행되는 시간은 전체 컴퓨팅 작업의 첫 번째 작업이 수행되기 시작하는 시각을 기준(예를 들어, 0초)으로 하여 각 작업이 수행되는 시각을 의미할 수 있다.
- [132] 일부 실시예에 따른 제어 서버는 740 단계에서 펼쳐진 작업 의존성 그래프의 각 스텝별로 복수의 작업들을 복수의 연산 서버에 할당할 수 있다. 제어 서버가 복수의 작업들을 스텝별로 복수의 연산 서버에 할당하는 단계에서 각 작업이 수행되는 시간을 결정하지 않거나, 각 동작의 소요 시간에 기초하여 각 작업이 수행되는 시간을 결정한 경우, 연산 서버 간 신호 송수신 시 발생하는 전송 지연 시간의 영향으로 작업들이 동일한 시간 구간 동안 하나의 M에 중복되어 할당되는 경우가 발생할 수도 있다.
- [133] 제어 서버는 제1 연산 서버의 제1 VM에 할당된 제1 작업 및 제2 작업에 대해, 제1 작업이 수행되는 시간과 제2 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 제2 작업을 제1 연산 서버의 제2 VM에 할당하고 제2 작업이 제2 VM에서 수행되는 시간을 결정하거나, 제1 작업이 수행된 후 제2 작업이 수행되도록 제1 VM에서 제2 작업이 수행되는 시간을 결정할 수 있다.
- [134] 제어 서버는 작업 의존성 그래프의 각 스텝별로 복수의 작업들 각각이 수행되는 시간을 결정할 수 있다. 제어 서버가 각 스텝별로 각 작업이 수행되는 시간을 결정하는 경우, 부속 관계를 갖는 작업들이 서로 다른 VM에 할당되지 않도록, 펼쳐진 작업 의존성 그래프가 아니라 연산 복잡도가 조정되지 않은 작업 의존성 그래프에 기초하여 각 작업이 수행되는 시간을 결정할 수 있다. 즉, 제어 서버는 도 3의 작업 의존성 그래프(300)에 기초하여 각 작업이 수행되는 시간을 결정할 수 있다.
- [135] 제어 서버는 각 작업이 수행되는 시간을 결정함으로써 전체 컴퓨팅 작업의 완료 시간을 알 수 있고, 작업들이 동일한 시간 구간 동안 하나의 VM에 중복되어 할당되지 않도록 스케줄링할 수 있다.
- [136] 한편, 일부 실시예에 따른 제어 서버는 작업을 할당하는 740 단계에서 각 VM에 동시에 할당되는 작업이 발생하지 않도록 각 작업이 수행되는 시간을 결정할

수도 있다. 예를 들어, 제어 서버는 펼쳐진 작업 의존성 그래프의 각 스텝별로 스텝의 시작 시간 및 종료 시간을 복수의 연산 서버들에 포함된 모든 VM에 대해 동일하게 설정할 수 있다.

- [137] 760 단계에서, 제어 서버는 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신할 수 있다.
- [138] 도 7의 710 단계, 720 단계, 730 단계, 740 단계, 760 단계의 자세한 내용은 각각 도 2의 210 단계, 220 단계, 230 단계, 240 단계, 250 단계에서 자세히 설명하였다.
- [139] 도 8은 일부 실시예에 따른 제어 서버가 복수의 작업들이 수행되는 시간을 결정하는 방법을 설명하기 위한 도면이다.
- [140] 도 8은 도 7의 750 단계에서 제어 서버가 작업 의존성 그래프의 스텝별로 각 작업이 수행되는 시간을 결정하는 예시를 도시한다. 즉, 도 8은 스텝 s 의 작업 k 가 수행되는 시간을 결정하는 방법의 예시를 설명한다.
- [141] 도 8에 도시된 각 변수의 의미는 다음 [표 1]과 같다.

[142] [표 1]

변수	정의
$r_{e,i}$	연산 서버 e 의 VM- i 에 할당된 작업이 모두 끝나는 시간
$r_{e,i'}$	연산 서버 e 의 VM- i' 에 할당된 작업이 모두 끝나는 시간
$b_{s,k}$	스텝 s 의 작업 k 의 연산이 시작되는 시간
$f_{s,k}$	스텝 s 의 작업 k 의 연산이 끝나는 시간
$f_{s',k'}$	스텝 s 의 작업 k 와 의존성이 있는 스텝 s' 의 작업 k' 의 연산이 끝나는 시간
e'	스텝 s 의 작업 k 와 의존성이 있는 스텝 s' 의 작업 k' 가 할당된 연산 서버
$l_{e',e}$	연산 서버 e' 와 연산 서버 e 사이의 전송 지연 시간

- [144] 도 8에 도시된 방법을 수행하기 전에, 즉 첫 번째 스텝의 첫 번째 작업이 수행되는 시간을 결정하기 전에, 제어 서버는 복수의 연산 서버에 작업을 할당하는 단계에서 각 작업의 소요시간 및 전송 지연 시간 정보에 기초하여 각 작업이 수행되는 시간을 결정할 수 있다. 즉, 제어 서버는 도 8에 도시된 방법을

수행하기 전에, [표 1]에 도시된 모든 변수 중 $r_{e,i}$ 및 $r_{e,i'}$ 를 제외한 모든 변수를 각 작업의 소요시간 및 전송 지연 시간 정보에 기초하여 결정할 수 있다.

- [145] 801 단계에서, 제어 서버는 $r_{e,i} \leq \max(f_{s',k'} + l_{e',e})$ 인지 여부를 판단할 수 있다. $\max(f_{s',k'} + l_{e',e})$ 은 스텝 s의 작업 k와 의존성이 있는 스텝 s-1까지의 적어도 하나의 작업의 연산이 끝나는 시각과, 전송한 적어도 하나의 작업이 할당된 연산 서버 e'로부터 작업 k가 할당된 연산 서버로의 전송 지연 시간을 합산한 값이 가장 큰 값을 의미한다.
- [146] 스텝 s의 작업 k와 의존성이 있는 스텝 s'의 작업 k'가 할당된 연산 서버 e'와 작업 k가 할당된 연산 서버 e가 같은 경우, 즉 전송 지연 시간이 발생하지 않은 경우, $l_{e',e}=0$ 이다. e'와 e가 같지 않은 경우 $l_{e',e}>0$ 이다.
- [147] $r_{e,i} \leq \max(f_{s',k'} + l_{e',e})$ 인 경우, 연산 서버 e의 VM-i에 스텝 s-1까지 할당된 작업이 끝난 후 작업 k를 수행하게 되므로 VM-i에 작업 k가 다른 작업과 동시에 할당되지 않는다.
- [148] $r_{e,i} > \max(f_{s',k'} + l_{e',e})$ 인 경우, 연산 서버 e의 VM-i에 스텝 s-1까지 할당된 작업이 끝나기 전에 작업 k를 수행하게 되므로, 즉 작업을 할당하는 단계에서 결정된 $b_{s,k}$ 가 $r_{e,i}$ 보다 작으므로, VM-i에 작업 k가 수행되는 시간과 다른 작업이 수행되는 시간이 적어도 일부 중복된다.
- [149] 제어 서버가 첫 번째 스텝의 첫 번째 작업이 수행되는 시각을 결정하는 경우, 즉 $s=1$ 이고 $k=1$ 인 경우, 아직 VM-i에 할당된 작업이 없으므로 $r_{e,i}=0$ 이다. 따라서 $s=1$ 이고 $k=1$ 인 경우에는 $r_{e,i} \leq \max(f_{s',k'} + l_{e',e})$ 이 늘 성립한다.
- [150] 802 단계에서, $r_{e,i} \leq \max(f_{s',k'} + l_{e',e})$ 인 경우, 제어 서버는 $b_{s,k}$, $f_{s,k}$ 를 결정하고, $r_{e,i}$ 를 갱신할 수 있다.
- [151] 제어 서버가 첫 번째 스텝의 첫 번째 작업이 수행되는 시각을 결정하는 경우, 즉 $s=1$ 이고 $k=1$ 인 경우에는, $b_{s,k}$, $f_{s,k}$ 및 $r_{e,i}$ 는 이전 스텝에 대한 고려 없이 결정되어 있고 갱신할 필요가 없다.
- [152] 작업 k가 첫 번째 스텝의 첫 번째 작업이 아닌 경우, 제어 서버는 $\max(f_{s',k'} + l_{e',e})$ 값을 작업 k가 시작되는 시각인 $b_{s,k}$ 로 결정하고, $b_{s,k}$ 에 작업 k의 작업 소요 시간을 합산한 값을 $f_{s,k}$ 로 결정할 수 있다. 제어 서버는 결정된 $f_{s,k}$ 값을 $r_{e,i}$ 로 결정함으로써 $r_{e,i}$ 를 갱신할 수 있다.
- [153] 803 단계에서, $r_{e,i} > \max(f_{s',k'} + l_{e',e})$ 인 경우, 즉 VM-i에 작업 k가 수행되는 시간과 다른 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 제어 서버는 연산 서버 e에서 작업이 VM-i보다 먼저 끝나는 VM-i'가 있는지 여부, 즉 $r_{e,i'} \leq r_{e,i}$ 인 VM-i'가 있는지 여부를 판단할 수 있다.
- [154] 804 단계에서, 연산 서버 e에서 작업이 VM-i보다 먼저 끝나는 VM-i'가 없는 경우, 제어 서버는 $r_{e,i}$ 이후로 $b_{s,k}$, $f_{s,k}$ 를 결정하고, $r_{e,i}$ 를 갱신할 수 있다.
- [155] 예를 들어 제어 서버는 $r_{e,i}$ 값을 $b_{s,k}$ 로 결정하고, $b_{s,k}$ 에 작업 k의 작업 소요 시간을 합산한 값을 $f_{s,k}$ 로 결정할 수 있다. 제어 서버는 결정된 $f_{s,k}$ 값을 $r_{e,i}$ 로

결정함으로써 $r_{e,i}$ 를 갱신할 수 있다.

[156] 즉, 작업 할당 단계에서 할당된 VM-i에 작업k가 수행되는 시간과 다른 작업이 수행되는 시간이 적어도 일부 중복되고, 작업이 VM-i보다 먼저 끝나는 VM-i'가 없는 경우, 제어 서버는 VM-i의 모든 작업이 끝난 이후 작업 k가 수행되도록 작업 k를 스케줄링할 수 있다.

[157] 805 단계에서, 연산 서버 e에서 작업이 VM-i보다 먼저 끝나는 VM-i'가 있는 경우, 제어 서버는 작업 k를 VM-i'에 스케줄링할 수 있다.

[158] 제어 서버가 작업 k를 작업 할당 단계에서 할당된 VM-i와 다른 VM-i'에 작업 k를 스케줄링하는 경우, 연산 서버 e 내에서 작업 k를 할당하는 VM만 조정되므로, 스텝 s에서 작업 k가 수행되는 데에 추가적인 전송 지연 시간이 발생하지 않을 수 있다.

[159] 예를 들어, 제어 서버는 연산 서버 e 내에서 작업이 가장 빨리 끝나는 VM에 작업 k를 스케줄링할 수 있다.

[160] 806 단계에서, 제어 서버는 $r_{e,i'} \leq \max(f_{s',k} + l_{e',e})$ 인지 여부를 판단할 수 있다.

[161] $r_{e,i'} \leq \max(f_{s',k} + l_{e',e})$ 인 경우, 연산 서버 e'의 VM-i'에 할당된 작업이 끝난 후 작업 k를 수행하게 되므로 VM-i'에 작업k가 다른 작업과 동시에 할당되지 않는다.

[162] $r_{e,i'} > \max(f_{s',k} + l_{e',e})$ 인 경우, 연산 서버 e'의 VM-i'에 할당된 작업이 끝나기 전에 작업 k를 수행하게 되므로, 즉 작업을 할당하는 단계에서 결정된 $b_{s,k}$ 가 $r_{e,i'}$ 보다 작으므로, VM-i'에 작업k가 수행되는 시간과 다른 작업이 수행되는 시간이 적어도 일부 중복된다.

[163] 807 단계에서, $r_{e,i'} \leq \max(f_{s',k} + l_{e',e})$ 인 경우, 제어 서버는 $b_{s,k}$, $f_{s,k}$ 를 결정하고, $r_{e,i'}$ 를 갱신할 수 있다.

[164] 예를 들어 제어 서버는 $\max(f_{s',k} + l_{e',e})$ 값을 작업 k가 시작되는 시각인 $b_{s,k}$ 로 결정하고, $b_{s,k}$ 에 작업 k의 작업 소요 시간을 합산한 값을 $f_{s,k}$ 로 결정할 수 있다. 제어 서버는 결정된 $f_{s,k}$ 값을 $r_{e,i'}$ 로 결정함으로써 $r_{e,i'}$ 를 갱신할 수 있다.

[165] 808 단계에서, $r_{e,i'} > \max(f_{s',k} + l_{e',e})$ 인 경우, 제어 서버는 $r_{e,i'}$ 이후로 $b_{s,k}$, $f_{s,k}$ 를 결정하고, $r_{e,i'}$ 를 갱신할 수 있다.

[166] 예를 들어 제어 서버는 $r_{e,i'}$ 값을 $b_{s,k}$ 로 결정하고, $b_{s,k}$ 에 작업 k의 작업 소요 시간을 합산한 값을 $f_{s,k}$ 로 결정할 수 있다. 제어 서버는 결정된 $f_{s,k}$ 값을 $r_{e,i'}$ 로 결정함으로써 $r_{e,i'}$ 를 갱신할 수 있다.

[167] 제어 서버는 도 8을 참조하여 설명한 바와 같이, 제1 연산 서버의 제1 VM에 할당된 제1 작업 및 제2 작업에 대해, 제1 작업이 수행되는 시간과 제2 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 제2 작업을 제1 연산 서버의 제2 VM에 할당하고 제2 작업이 제2 VM에서 수행되는 시간을 결정하거나, 제1 작업이 수행된 후 제2 작업이 수행되도록 제1 VM에서 제2 작업이 수행되는 시간을 결정할 수 있다. 즉 제어 서버는 작업들이 동일한 시간 구간 동안 하나의 VM에 중복되어 할당되지 않도록 스케줄링할 수 있다.

- [168] 또한 제어 서버는 각 작업이 수행되는 시간을 결정함으로써 전체 컴퓨팅 작업의 완료 시간을 알 수 있다. 예를 들어, 전체 컴퓨팅 작업의 마지막 스텝의 작업의 $r_{e,i}$ 가 전체 컴퓨팅 작업의 완료 시간을 나타낼 수 있다.
- [169] 도 9는 일부 실시예에 따른 단말이 제어 서버에 컴퓨팅 작업을 오프로딩하는 방법을 설명하기 위한 흐름도이다.
- [170] 910 단계에서, 단말은 제어 서버에 복수의 작업들을 포함하는 컴퓨팅 작업을 송신할 수 있다.
- [171] 단말은 컴퓨팅 작업과 함께 컴퓨팅 작업에 대한 오프로딩 요청을 제어 서버에 송신할 수 있다. 컴퓨팅 작업의 종류는 다양하게 정해질 수 있다. 예를 들어, 컴퓨팅 작업은 단말에 설치된 어플리케이션에서 제공되는 서비스가 수행되기 위한 연산 집합일 수 있다. 물론 이에 한정되지 않고 컴퓨팅 작업은 다양한 종류의 작업을 포함할 수 있다.
- [172] 920 단계에서, 단말은 제어 서버 또는 적어도 하나의 연산 서버로부터 복수의 작업들의 수행 결과를 수신할 수 있다.
- [173] 복수의 작업들의 수행 결과는, 일부 실시예에 따른 제어 서버가 도 1 내지 도 8을 참조하여 설명한 다양한 방법에 의해 복수의 작업들을 복수의 연산 서버들에 스케줄링하고, 복수의 연산 서버들이 스케줄링에 따라 연산을 수행한 결과일 수 있다. 예를 들어, 복수의 작업들의 수행 결과는, 복수의 작업들이 복수의 작업들 각각의 의존 관계에 기초하여 생성된 작업 의존성 그래프(task dependency graph), 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보에 기초하여 스케줄링된 복수의 연산 서버로부터 복수의 작업들이 수행된 결과일 수 있다.
- [174] 도 10은 일부 실시예에 따른 제어 서버(1000)의 블록도이다.
- [175] 도 10을 참조하면, 제어 서버(1000)는 송수신부(1020), 프로세서(1010) 및 메모리(1030)를 포함할 수 있다. 상기 실시예들에서 제안한 제어 서버의 스케줄링 방법에 따라, 송수신부(1020), 프로세서(1010) 및 메모리(1030)가 동작할 수 있다. 다만, 일부 실시예에 따른 제어 서버(1000)의 구성 요소가 전술한 예에 한정되는 것은 아니다. 다른 실시예에 따라, 제어 서버(1000)는 전술한 구성 요소들 보다 더 많은 구성 요소를 포함하거나 더 적은 구성 요소를 포함할 수도 있다. 뿐만 아니라 특정한 경우 송수신부(1020), 프로세서(1010) 및 메모리(1030)가 하나의 칩(chip) 형태로 구현될 수도 있다.
- [176] 송수신부(1020)는 적어도 하나의 단말 및 복수의 연산 서버들과 신호를 송수신할 수 있다. 신호는 제어 정보 및 데이터를 포함할 수 있다. 이를 위해, 송수신부(1020)는 송신되는 신호의 주파수를 상승 변환 및 증폭하는 RF 송신기 및 수신되는 신호를 저 잡음 증폭하고 주파수를 하강 변환하는 RF 수신기 등으로 구성될 수 있다. 다만, 이는 일 실시예일뿐, 송수신부(1020)의 구성 요소가 RF 송신기 및 RF 수신기로 한정되는 것은 아니다.
- [177] 또한, 송수신부(1020)는 무선 채널을 통해 신호를 수신하여 프로세서(1010)로

출력하고, 프로세서(1010)로부터 출력된 신호를 무선 채널을 통해 송신할 수 있다.

- [178] 프로세서(1010)는 상술한 본 개시의 실시예에 따라 제어 서버(1000)가 동작할 수 있도록 일련의 과정을 제어할 수 있다. 예를 들어, 프로세서(1010)는 전술한 실시예들 중 적어도 하나 이상의 신호 송수신 방법을 수행할 수 있다.
- [179] 일부 실시예에 따른 프로세서(1010)는, 단말로부터 컴퓨팅 작업을 수신하고, 컴퓨팅 작업에 포함된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하고, 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하고, 작업 의존성 그래프, 연산 처리 용량 정보, 및 전송 지연 시간 정보에 기초하여, 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 복수의 작업들의 수행 결과를 단말에 송신하도록 제어할 수 있다.
- [180] 일부 실시예에 따른 프로세서(1010)는, 복수의 작업들 각각의 연산 복잡도를 식별하고, 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 복수의 작업들 각각의 연산 복잡도를 조정하고, 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프를 생성할 수 있다.
- [181] 일부 실시예에 따른 프로세서(1010)는, 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우, 제1 작업이 할당된 제1 연산 서버에 제2 작업을 할당하고, 제1 연산 서버에 제2 작업을 할당할 수 없는 경우, 전송 지연 시간 정보에 기초하여 제2 작업을 할당할 수 있는 연산 서버 중 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 제2 작업을 할당할 수 있다.
- [182] 일부 실시예에 따른 프로세서(1010)는, 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우, 제1 작업이 할당된 제1 연산 서버에 제2 작업을 할당하고, 제1 연산 서버에 제2 작업을 할당할 수 없는 경우, 전송 지연 시간 정보에 기초하여 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 제2 작업을 할당하고, 제2 연산 서버에 제2 작업을 할당할 수 없는 경우, 연산 처리 용량 정보에 기초하여 복수의 연산 서버들 중 가장 큰 연산 처리 용량을 가진 제3 연산 서버에 제2 작업을 할당할 수 있다.
- [183] 일부 실시예에 따르면, 제어 서버(1000)는 복수의 연산 서버들 각각은 복수의 VM(virtual machine)들을 포함하고, 프로세서(1010)는, 복수의 작업들 각각을 복수의 연산 서버들에 할당하고 복수의 작업들 각각이 수행되는 시간을 결정함으로써 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 제1 연산 서버의 제1 VM에 할당된 제1 작업 및 제2 작업에 대해, 제1 작업이 수행되는 시간과 제2 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 제2 작업을 제1 연산 서버의 제2 VM에 할당하고 제2 작업이 수행되는 시간을

결정하거나, 제1 작업이 수행된 후 제2 작업이 수행되도록 제2 작업이 수행되는 시간을 결정할 수 있다.

- [184] 일부 실시예에 따른 프로세서(1010)는, 복수의 작업들 각각을 복수의 연산 서버들에 할당하고 복수의 작업들 각각이 수행되는 시간을 결정함으로써 복수의 작업들 각각을 복수의 연산 서버들에 스케줄링하고, 복수의 작업들 각각의 연산 복잡도를 식별하고, 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 복수의 작업들 각각의 연산 복잡도를 조정하고, 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 따라, 펼쳐진(spread) 작업 의존성 그래프를 생성하고, 펼쳐진 작업 의존성 그래프에 기초하여 복수의 작업들 각각을 복수의 연산 서버들에 할당하고, 연산 복잡도가 조정되지 않은 복수의 작업들 각각의 의존 관계에 기초하여 생성된 작업 의존성 그래프에 기초하여 복수의 작업들 각각이 수행되는 시간을 결정할 수 있다.
- [185] 메모리(1030)는 제어 서버(1000)에서 획득되는 신호에 포함된 제어 정보 또는 데이터를 저장할 수 있으며, 프로세서(1010)의 제어에 필요한 데이터 및 프로세서(1010)에서 제어 시 발생하는 데이터 등을 저장하기 위한 영역을 가질 수 있다. 예를 들어, 메모리(1030)는 합산 가중치 결정을 위해 기 설정된 송신 안테나 가중치 정보 및 수신 안테나 가중치 정보를 저장할 수 있다. 또한, 메모리(1030)는 결정된 합산 가중치에 관한 정보를 저장할 수 있다.
- [186] 이러한 메모리(1030)는 롬(ROM) 또는/및 램(RAM) 또는/및 하드디스크 또는/및 CD-ROM 또는/및 DVD 등의 다양한 형태로 구성될 수 있다.
- [187] 도 11은 일부 실시예에 따른 단말(1100)의 블록도이다.
- [188] 도 11을 참조하면, 단말(1100)은 송수신부(1120), 프로세서(1110) 및 메모리(1130)를 포함할 수 있다. 상기 실시예들에서 제안한 단말의 스케줄링 방법에 따라, 송수신부(1120), 프로세서(1110) 및 메모리(1130)가 동작할 수 있다. 다만, 일부 실시예에 따른 단말(1100)의 구성 요소가 전술한 예에 한정되는 것은 아니다. 다른 실시예에 따라, 단말(1100)은 전술한 구성 요소들 보다 더 많은 구성 요소를 포함하거나 더 적은 구성 요소를 포함할 수도 있다. 뿐만 아니라 특정한 경우 송수신부(1120), 프로세서(1110) 및 메모리(1130)가 하나의 칩(chip) 형태로 구현될 수도 있다.
- [189] 송수신부(1120)는 제어 서버 및 복수의 연산 서버들과 신호를 송수신할 수 있다. 신호는 제어 정보 및 데이터를 포함할 수 있다. 이를 위해, 송수신부(1120)는 송신되는 신호의 주파수를 상승 변환 및 증폭하는 RF 송신기 및 수신되는 신호를 저 잡음 증폭하고 주파수를 하강 변환하는 RF 수신기 등으로 구성될 수 있다. 다만, 이는 일 실시예일뿐, 송수신부(1120)의 구성 요소가 RF 송신기 및 RF 수신기로 한정되는 것은 아니다.
- [190] 또한, 송수신부(1120)는 무선 채널을 통해 신호를 수신하여 프로세서(1110)로 출력하고, 프로세서(1110)로부터 출력된 신호를 무선 채널을 통해 송신할 수

있다.

- [191] 프로세서(1110)는 상술한 본 개시의 실시예에 따라 단말(1100)이 동작할 수 있도록 일련의 과정을 제어할 수 있다. 예를 들어, 프로세서(1110)는 전술한 실시예들 중 적어도 하나 이상의 신호 송수신 방법을 수행할 수 있다.
- [192] 일부 실시예에 따른 프로세서(1110)는, 제어 서버에 컴퓨팅 작업을 송신하고, 제어 서버 또는 적어도 하나의 연산 서버로부터 복수의 작업들의 수행 결과를 수신하도록 제어할 수 있다. 복수의 작업들의 수행 결과는, 복수의 작업들이 복수의 작업들 각각의 의존 관계에 기초하여 생성된 작업 의존성 그래프(task dependency graph), 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보에 기초하여 스케줄링된 복수의 연산 서버로부터 복수의 작업들이 수행된 결과일 수 있다.
- [193] 메모리(1130)는 단말(1100)에서 획득되는 신호에 포함된 제어 정보 또는 데이터를 저장할 수 있으며, 프로세서(1110)의 제어에 필요한 데이터 및 프로세서(1110)에서 제어 시 발생하는 데이터 등을 저장하기 위한 영역을 가질 수 있다. 예를 들어, 메모리(1130)는 합산 가중치 결정을 위해 기 설정된 송신 안테나 가중치 정보 및 수신 안테나 가중치 정보를 저장할 수 있다. 또한, 메모리(1130)는 결정된 합산 가중치에 관한 정보를 저장할 수 있다.
- [194] 이러한 메모리(1130)는 롬(ROM) 또는/및 램(RAM) 또는/및 하드디스크 또는/및 CD-ROM 또는/및 DVD 등의 다양한 형태로 구성될 수 있다.
- [195] 한편, 본 명세서와 도면에 개시된 본 개시의 실시예들은 본 개시의 기술 내용을 쉽게 설명하고 본 개시의 이해를 돕기 위해 특정 예를 제시한 것일 뿐이며, 본 개시의 범위를 한정하고자 하는 것은 아니다.
- [196] 즉 본 개시의 기술적 사상에 바탕을 둔 다른 변형예들이 실시 가능하다는 것은 본 개시의 속하는 기술 분야에서 통상의 지식을 가진 자에게 자명한 것이다. 또한 상기 각각의 실시예는 설명의 편의를 위하여 구분된 것으로, 필요에 따라 서로 조합되어 운용할 수 있다. 예컨대, 본 개시의 각각의 실시예의 일부분들이 서로 조합되어 제어 서버와 단말이 운용될 수 있다. 또한, 실시예들은 다른 시스템, 예를 들어, LTE 시스템, 5G 또는 NR 시스템 등에도 상술한 실시예의 기술적 사상에 바탕을 둔 다른 변형예들이 실시 가능할 수 있다.
- [197] 개시된 실시예에 따른 장치는 프로세서, 프로그램 데이터를 저장하고 실행하는 메모리, 디스크 드라이브와 같은 영구 저장부(permanent storage), 외부 장치와 통신하는 통신 포트, 터치 패널, 키(key), 버튼 등과 같은 사용자 인터페이스 장치 등을 포함할 수 있다.
- [198] 개시된 실시예에 따른 장치의 동작방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 전술한 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 전술한 매체에 기록되는 프로그램 명령은 본 개시를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터

소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 전술한 컴퓨터 판독 가능 매체는 컴퓨터 프로그램 제품에 포함될 수 있다.

- [199] 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.
- [200] 개시된 실시예는 기능적인 블록 구성들 및 다양한 처리 단계들로 나타내어질 수 있다. 이러한 기능 블록들은 특정 기능들을 실행하는 다양한 개수의 하드웨어 또는/및 소프트웨어 구성들로 구현될 수 있다. 예를 들어, 개시된 실시예는 하나 이상의 마이크로프로세서들의 제어 또는 다른 제어 장치들에 의해서 다양한 기능들을 실행할 수 있는, 메모리, 프로세싱, 로직(logic), 룩업 테이블(look-up table) 등과 같은 직접 회로 구성들을 채용할 수 있다. 개시된 실시예의 구성 요소들이 소프트웨어 프로그래밍 또는 소프트웨어 요소들로 실행될 수 있는 것과 유사하게, 개시된 실시예는 데이터 구조, 프로세스들, 루틴들 또는 다른 프로그래밍 구성들의 조합으로 구현되는 다양한 알고리즘을 포함하여, C, C++, 자바(Java), 어셈블러(assembler) 등과 같은 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 기능적인 측면들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다. 또한, 개시된 실시예는 전자적인 환경 설정, 신호 처리, 및/또는 데이터 처리 등을 위하여 종래 기술을 채용할 수 있다.

청구범위

- [청구항 1] 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 복수의 연산 서버들에 스케줄링하는 제어 서버에 있어서,
송수신부;
메모리; 및
단말로부터 상기 컴퓨팅 작업을 수신하고,
상기 컴퓨팅 작업에 포함된 상기 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하고,
상기 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하고,
상기 작업 의존성 그래프, 상기 연산 처리 용량 정보, 및 상기 전송 지연 시간 정보에 기초하여, 상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 스케줄링하고,
상기 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 상기 복수의 작업들의 수행 결과를 상기 단말에 송신하도록 제어하는 적어도 하나의 프로세서를 포함하는, 제어 서버.
- [청구항 2] 제1항에 있어서, 상기 적어도 하나의 프로세서는,
상기 복수의 작업들 각각의 연산 복잡도를 식별하고,
상기 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 상기 복수의 작업들 각각의 연산 복잡도를 조정하고,
상기 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 기초하여 상기 작업 의존성 그래프를 생성하는, 제어 서버.
- [청구항 3] 제1항에 있어서, 상기 적어도 하나의 프로세서는,
상기 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우,
상기 제1 작업이 할당된 제1 연산 서버에 상기 제2 작업을 할당하고,
상기 제1 연산 서버에 상기 제2 작업을 할당할 수 없는 경우, 상기 전송 지연 시간 정보에 기초하여 상기 제2 작업을 할당할 수 있는 연산 서버 중 상기 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 상기 제2 작업을 할당하는, 제어 서버.
- [청구항 4] 제1항에 있어서, 상기 적어도 하나의 프로세서는,
상기 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우,
상기 제1 작업이 할당된 제1 연산 서버에 상기 제2 작업을 할당하고,
상기 제1 연산 서버에 상기 제2 작업을 할당할 수 없는 경우, 상기 전송 지연 시간 정보에 기초하여 상기 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 상기 제2 작업을 할당하고,
상기 제2 연산 서버에 상기 제2 작업을 할당할 수 없는 경우, 상기 연산

[청구항 5]

처리 용량 정보에 기초하여 상기 복수의 연산 서버들 중 가장 큰 연산 처리 용량을 가진 제3 연산 서버에 상기 제2 작업을 할당하는, 제어 서버.
제1항에 있어서,
상기 제어 서버는, 상기 복수의 연산 서버들 각각은 복수의 VM(virtual machine)들을 포함하고,
상기 적어도 하나의 프로세서는,
상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 할당하고 상기 복수의 작업들 각각이 수행되는 시간을 결정함으로써 상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 스케줄링하고,
제1 연산 서버의 제1 VM에 할당된 제1 작업 및 제2 작업에 대해, 상기 제1 작업이 수행되는 시간과 상기 제2 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 상기 제2 작업을 제1 연산 서버의 제2 VM에 할당하고
상기 제2 작업이 수행되는 시간을 결정하거나, 상기 제1 작업이 수행된 후
상기 제2 작업이 수행되도록 상기 제2 작업이 수행되는 시간을 결정하는,
제어 서버.

[청구항 6]

제1항에 있어서, 상기 적어도 하나의 프로세서는,
상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 할당하고 상기 복수의 작업들 각각이 수행되는 시간을 결정함으로써 상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 스케줄링하고,
상기 복수의 작업들 각각의 연산 복잡도를 식별하고, 상기 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 상기 복수의 작업들 각각의 연산 복잡도를 조정하고, 상기 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 따라,
펼쳐진(spread) 작업 의존성 그래프를 생성하고, 상기 펼쳐진 작업 의존성 그래프에 기초하여 상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 할당하고,
연산 복잡도가 조정되지 않은 상기 복수의 작업들 각각의 의존 관계에 기초하여 생성된 상기 작업 의존성 그래프에 기초하여 상기 복수의 작업들 각각이 수행되는 시간을 결정하는, 제어 서버.

[청구항 7]

제어 서버에 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 오프로딩하는 단말에 있어서,
송수신부;
메모리; 및
상기 제어 서버에 상기 컴퓨팅 작업을 송신하고,
상기 제어 서버 또는 적어도 하나의 연산 서버로부터 상기 복수의 작업들의 수행 결과를 수신하도록 제어하는 적어도 하나의 프로세서를 포함하고,
상기 수행 결과는, 상기 복수의 작업들이 상기 복수의 작업들 각각의 의존

관계에 기초하여 생성된 작업 의존성 그래프(task dependency graph), 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보에 기초하여 스케줄링된 상기 복수의 연산 서버로부터 상기 복수의 작업들이 수행된 결과인, 단말.

[청구항 8] 제어 서버가 복수의 작업들을 포함하는 컴퓨팅 작업(task)을 복수의 연산 서버들에 스케줄링하는 방법에 있어서,
단말로부터 상기 컴퓨팅 작업을 수신하는 단계;
상기 컴퓨팅 작업에 포함된 상기 복수의 작업들 각각의 의존 관계에 기초하여 작업 의존성 그래프(task dependency graph)를 생성하는 단계;
상기 복수의 연산 서버들 각각의 연산 처리 용량 정보 및 복수의 연산 서버들 사이의 전송 지연 시간 정보를 획득하는 단계;
상기 작업 의존성 그래프, 상기 연산 처리 용량 정보, 및 상기 전송 지연 시간 정보에 기초하여, 상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 스케줄링하는 단계;
상기 복수의 연산 서버들 중 적어도 하나의 연산 서버로부터 수신한 상기 복수의 작업들의 수행 결과를 상기 단말에 송신하는 단계를 포함하는, 스케줄링 방법.

[청구항 9] 제8항에 있어서,
상기 작업 의존성 그래프를 생성하는 단계는,
상기 복수의 작업들 각각의 연산 복잡도를 식별하는 단계;
상기 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 상기 복수의 작업들 각각의 연산 복잡도를 조정하는 단계; 및
상기 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 기초하여 상기 작업 의존성 그래프를 생성하는 단계를 포함하는, 스케줄링 방법.

[청구항 10] 제8항에 있어서,
상기 스케줄링하는 단계는, 상기 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우,
상기 제1 작업이 할당된 제1 연산 서버에 상기 제2 작업을 할당하고,
상기 제1 연산 서버에 상기 제2 작업을 할당할 수 없는 경우, 상기 전송 지연 시간 정보에 기초하여 상기 제2 작업을 할당할 수 있는 연산 서버 중 상기 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 상기 제2 작업을 할당하는 단계를 포함하는, 스케줄링 방법.

[청구항 11] 제8항에 있어서,
상기 스케줄링하는 단계는, 상기 복수의 작업들 중 제2 작업이 제1 작업과 의존 관계를 갖는 경우,
상기 제1 작업이 할당된 제1 연산 서버에 상기 제2 작업을 할당하고,
상기 제1 연산 서버에 상기 제2 작업을 할당할 수 없는 경우, 상기 전송

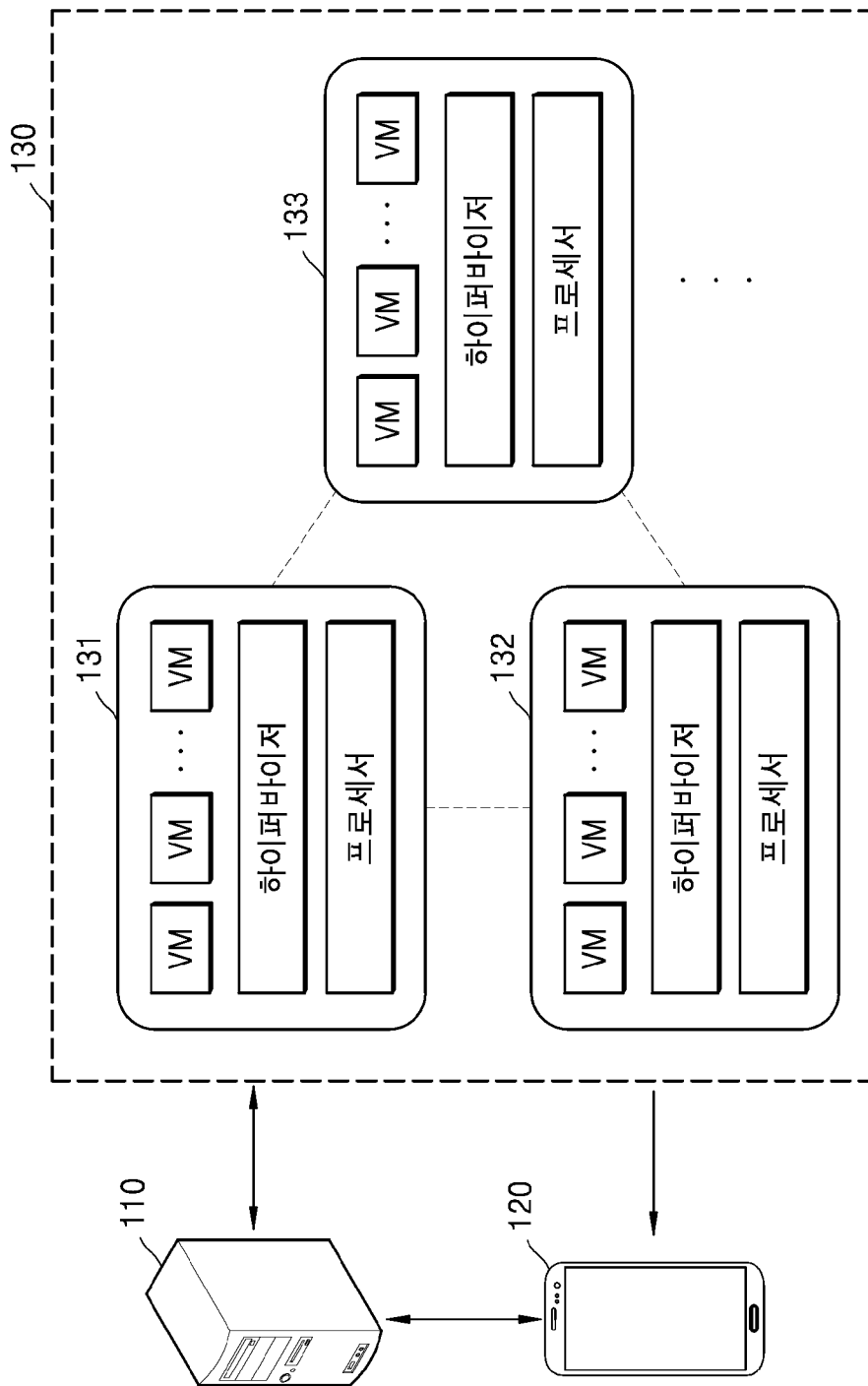
지연 시간 정보에 기초하여 상기 제1 연산 서버로부터의 전송 지연 시간이 가장 짧은 제2 연산 서버에 상기 제2 작업을 할당하고, 상기 제2 연산 서버에 상기 제2 작업을 할당할 수 없는 경우, 상기 연산 처리 용량 정보에 기초하여 상기 복수의 연산 서버들 중 가장 큰 연산 처리 용량을 가진 제3 연산 서버에 상기 제2 작업을 할당하는 단계를 포함하는, 스케줄링 방법.

[청구항 12] 제8항에 있어서,
상기 복수의 연산 서버들 각각은 복수의 VM(virtual machine)들을 포함하고,
상기 스케줄링하는 단계는,
상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 할당하는 단계; 및
상기 복수의 작업들 각각이 수행되는 시간을 결정하는 단계를 포함하고,
상기 수행되는 시간을 결정하는 단계는,
제1 연산 서버의 제1 VM에 할당된 제1 작업 및 제2 작업에 대해, 상기 제1 작업이 수행되는 시간과 상기 제2 작업이 수행되는 시간이 적어도 일부 중복되는 경우, 상기 제2 작업을 제1 연산 서버의 제2 VM에 할당하고
상기 제2 작업이 수행되는 시간을 결정하거나, 상기 제1 작업이 수행된 후
상기 제2 작업이 수행되도록 상기 제2 작업이 수행되는 시간을 결정하는,
스케줄링 방법.

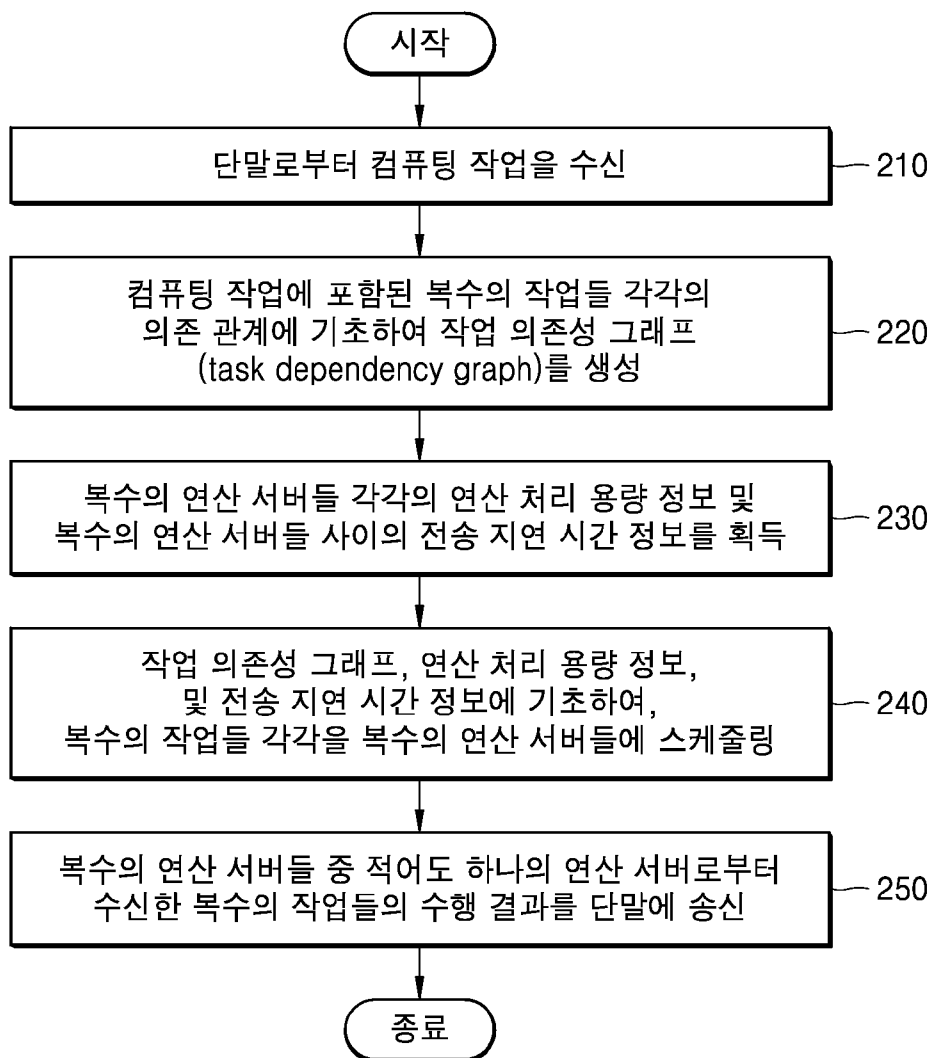
[청구항 13] 제12항에 있어서,
상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 할당하는 단계는,
상기 복수의 작업들 각각의 연산 복잡도를 식별하는 단계;
상기 복수의 작업들 중 적어도 하나의 작업을 적어도 하나의 부속 작업(sub-task)으로 변환함으로써, 상기 복수의 작업들 각각의 연산 복잡도를 조정하는 단계;
상기 연산 복잡도가 조정된 복수의 작업들 각각의 의존 관계에 따라, 펼쳐진(spread) 작업 의존성 그래프를 생성하는 단계; 및
상기 펼쳐진 작업 의존성 그래프에 기초하여 상기 복수의 작업들 각각을 상기 복수의 연산 서버들에 할당하는 단계를 포함하고,
상기 수행되는 시간을 결정하는 단계는,
연산 복잡도가 조정되지 않은 상기 복수의 작업들 각각의 의존 관계에 기초하여 생성된 상기 작업 의존성 그래프에 기초하여 상기 복수의 작업들 각각이 수행되는 시간을 결정하는 단계를 포함하는, 스케줄링 방법.

[청구항 14] 제8항의 방법을 수행하도록 하는 프로그램이 저장된 기록매체를 포함하는 컴퓨터 프로그램 제품.

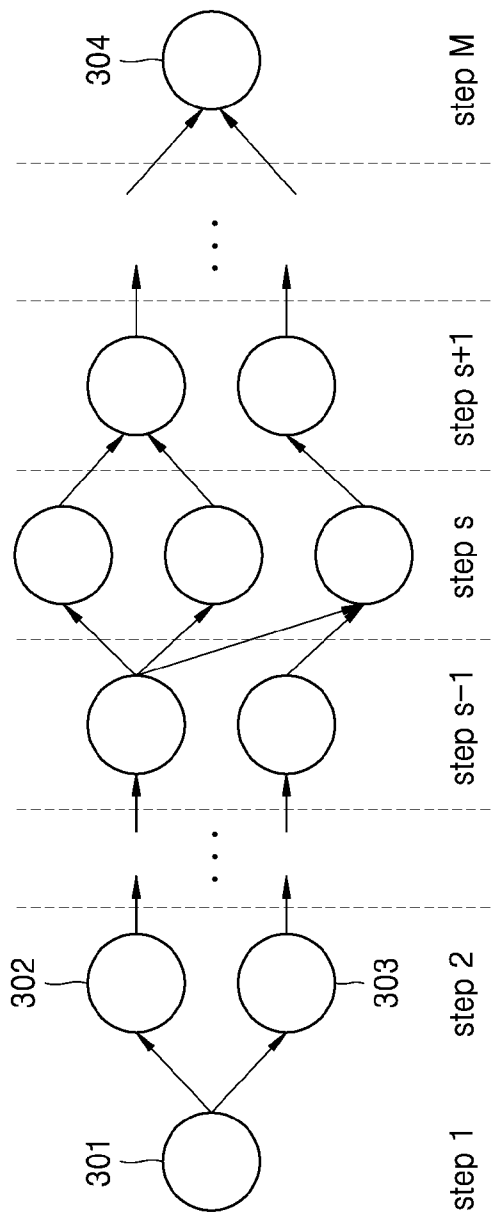
[도 1]



[도2]

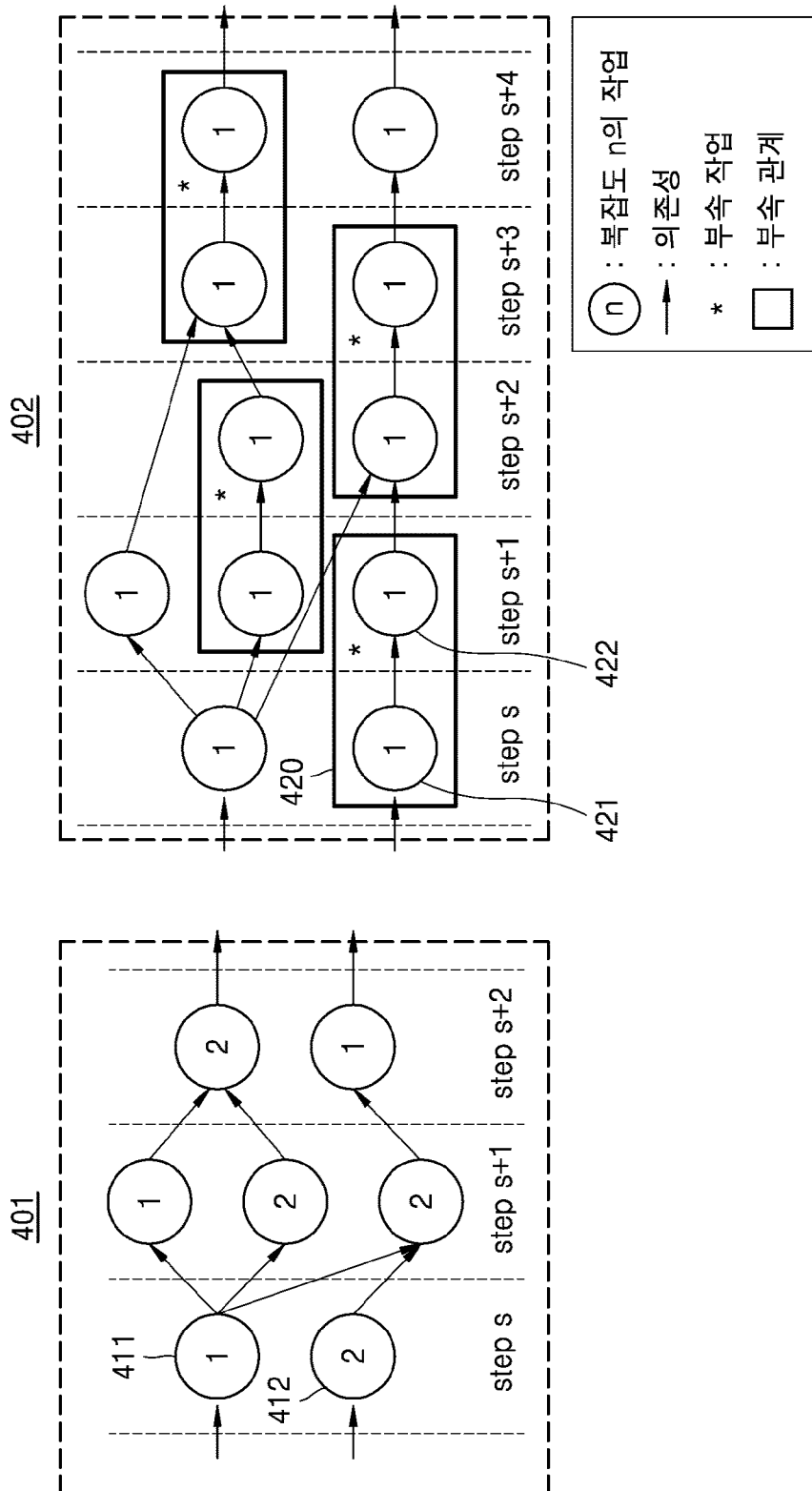


[도3]

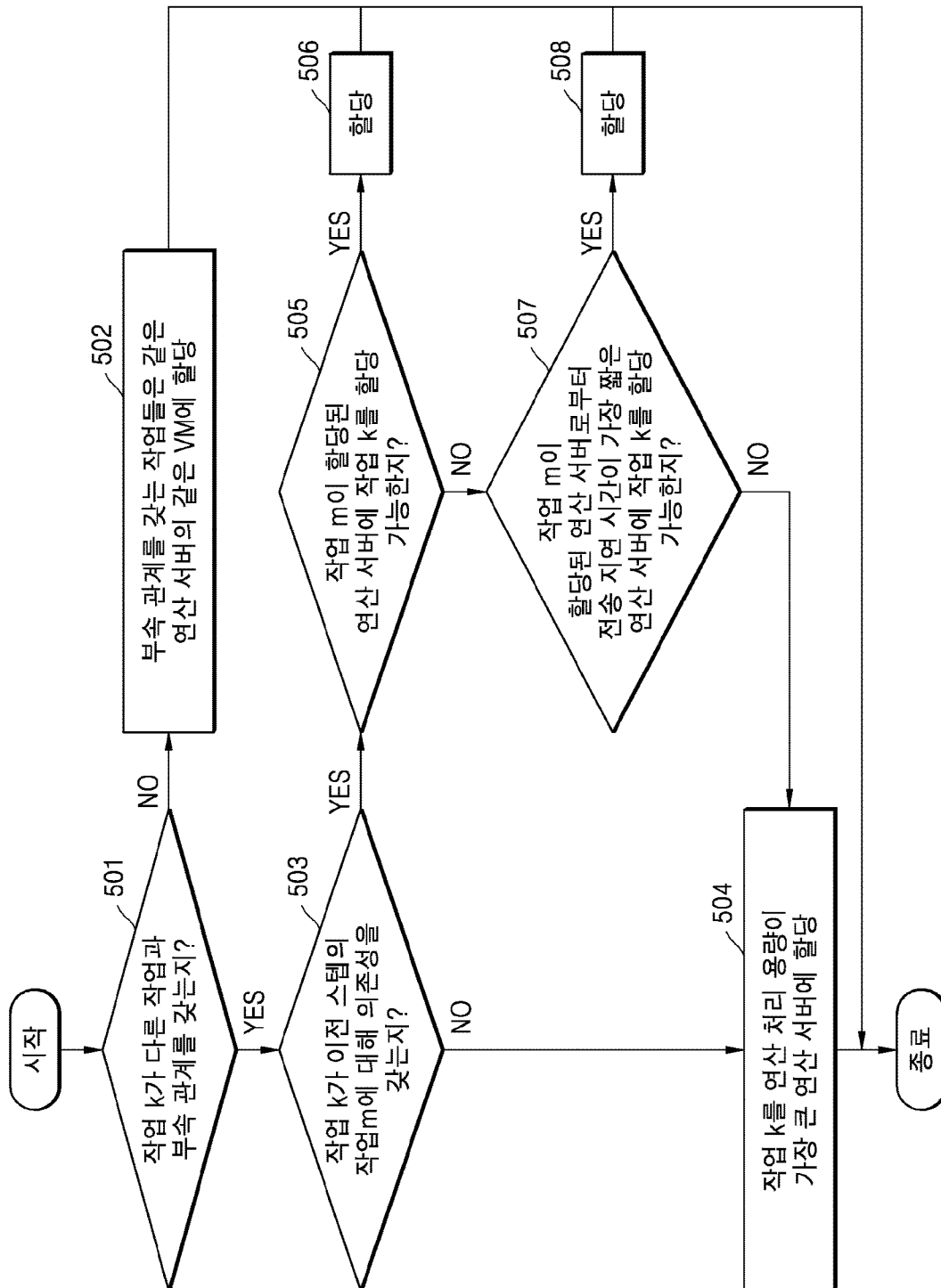


300

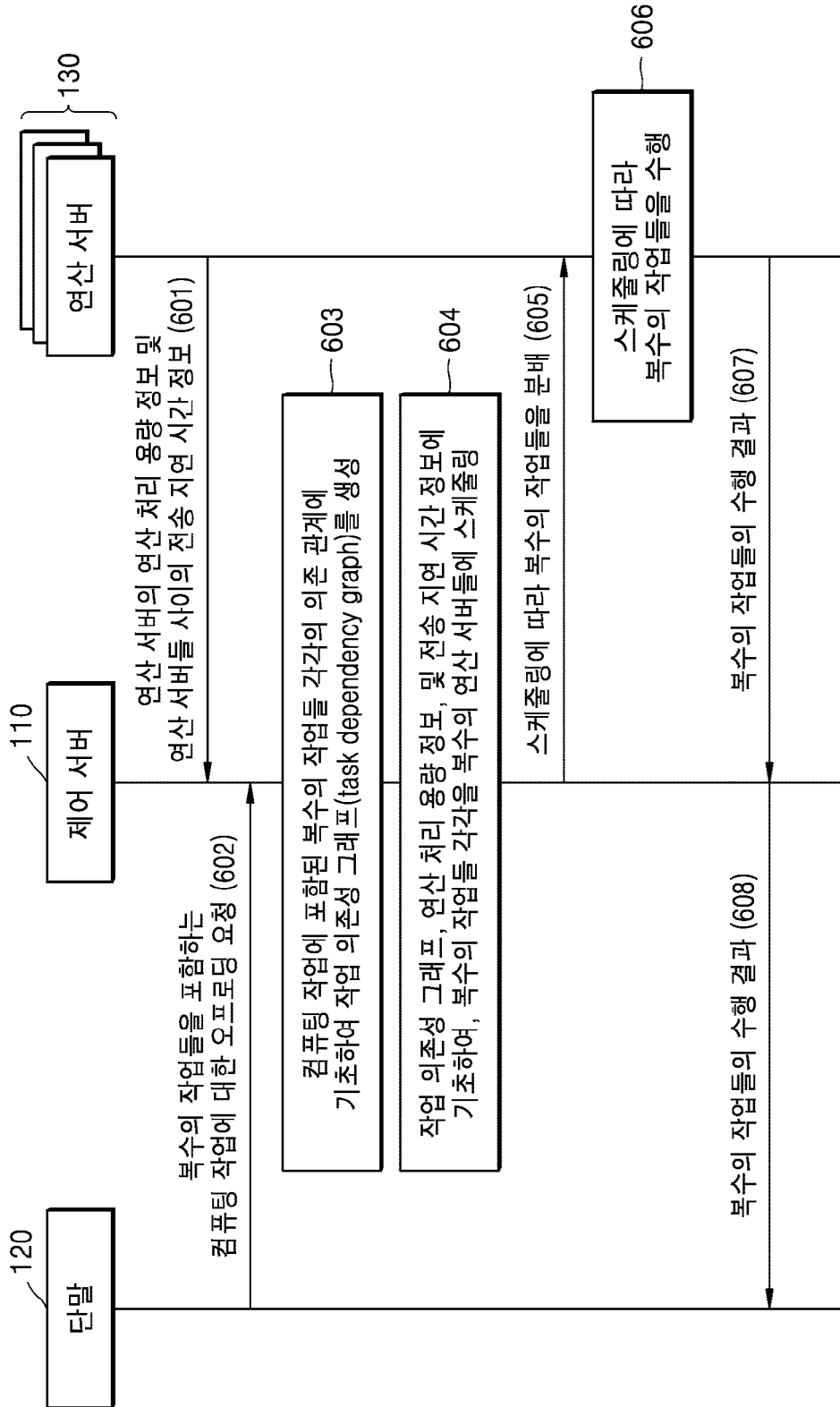
[도4]



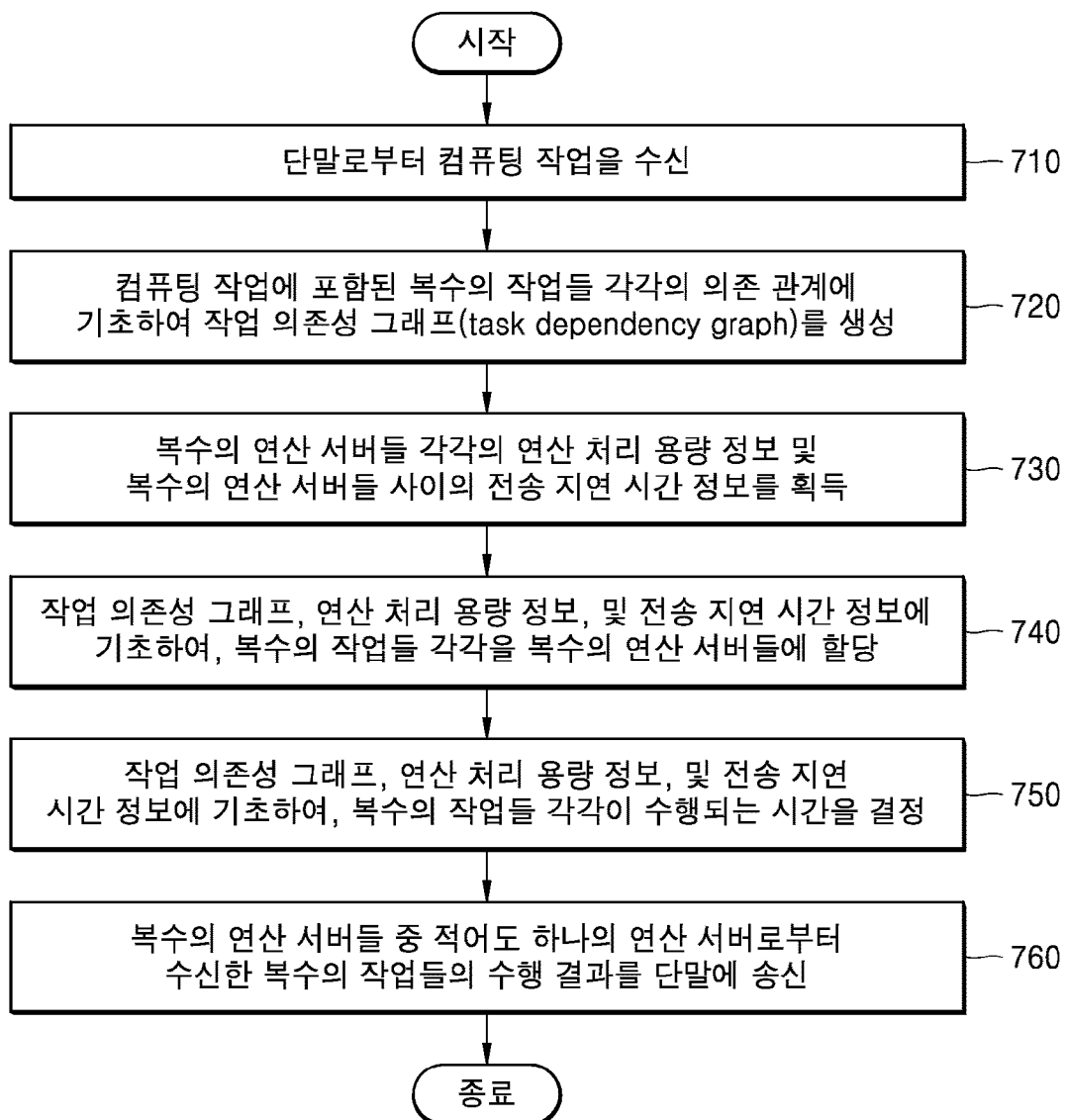
[도5]



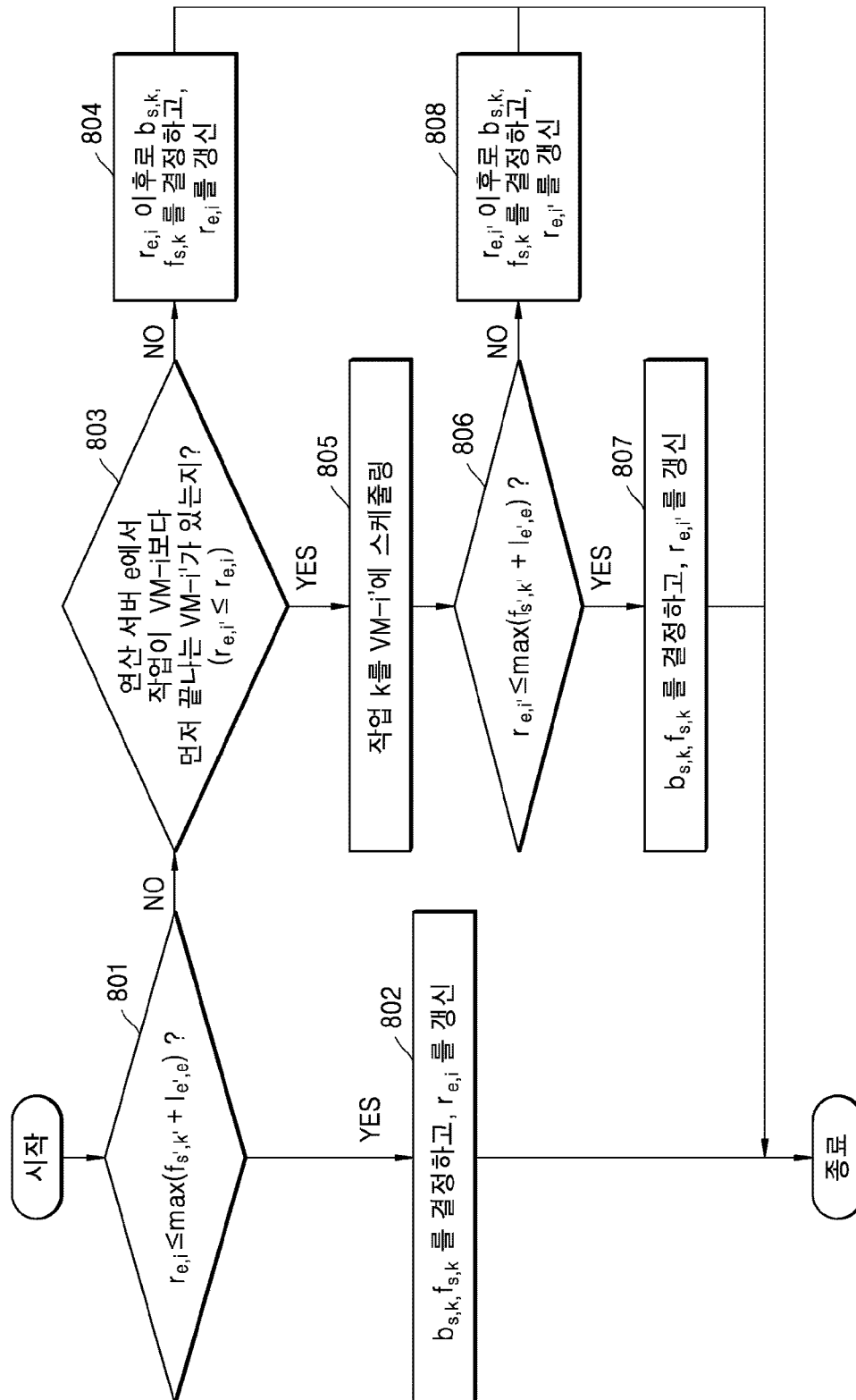
[도6]



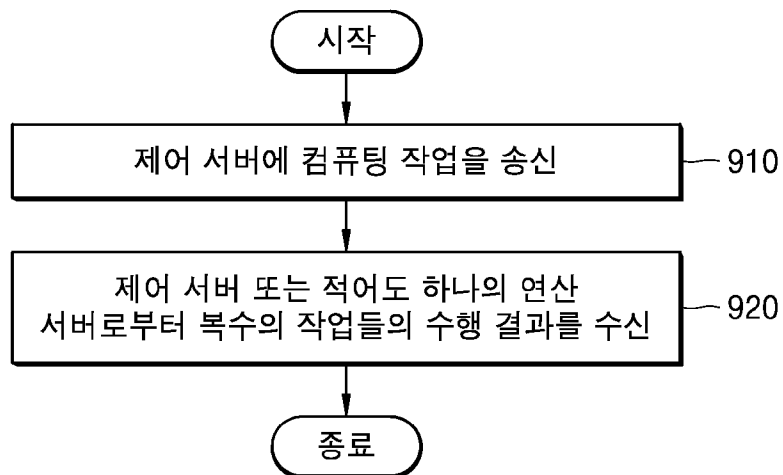
[도7]



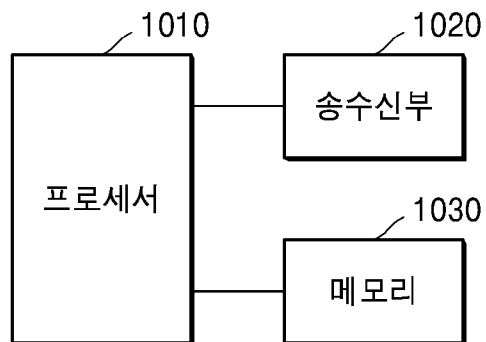
[도8]



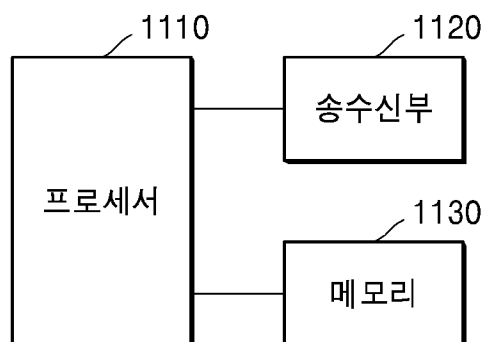
[도9]



[도10]

1000

[도11]

1100

INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2019/014040**A. CLASSIFICATION OF SUBJECT MATTER*****G06F 9/48(2006.01)i, G06F 9/50(2006.01)i, G06F 9/455(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 9/48; G06F 11/20; G06F 11/36; G06F 15/16; G06F 17/30; G06F 9/44; G06F 9/455; G06F 9/46; G06F 9/50

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: Internet of Things (IoT), Server, Task, Dependency, Capacity, Transmission delay, Complexity, Distribute, Parallel

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2011-243120 A (NEC CORP.) 01 December 2011 See paragraphs [0018]-[0055]; claim 1; and figures 1-3.	1-14
Y	JP 2016-162067 A (NEC CORP.) 05 September 2016 See paragraphs [0018]-[0024]; and claim 1.	1-14
Y	KR 10-2017-0044320 A (BD INC.) 25 April 2017 See paragraphs [0059]-[0060]; and figure 4.	2,6,9,13
Y	KR 10-2015-0009662 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 27 January 2015 See paragraphs [0027]-[0036]; and figure 1.	5,12-13
A	US 2014-0108861 A1 (HADAPT, INC.) 17 April 2014 See paragraphs [0033]-[0048]; and figure 1.	1-14



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

“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

30 JANUARY 2020 (30.01.2020)

Date of mailing of the international search report

30 JANUARY 2020 (30.01.2020)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex Daejeon Building 4, 189, Cheongsu-ro, Seo-gu,
Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2019/014040

Patent document cited in search report	Publication date	Patent family member	Publication date
JP 2011-243120 A	01/12/2011	JP 5482442 B2	07/05/2014
JP 2016-162067 A	05/09/2016	JP 6459633 B2	30/01/2019
KR 10-2017-0044320 A	25/04/2017	KR 10-1794016 B1	08/11/2017
KR 10-2015-0009662 A	27/01/2015	KR 10-1941282 B1	23/01/2019
		US 2015-0026306 A1	22/01/2015
US 2014-0108861 A1	17/04/2014	US 9934276 B2	03/04/2018
		WO 2014-062637 A2	24/04/2014
		WO 2014-062637 A3	12/09/2014

A. 발명이 속하는 기술분류(국제특허분류(IPC))

G06F 9/48(2006.01)i, G06F 9/50(2006.01)i, G06F 9/455(2006.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

G06F 9/48; G06F 11/20; G06F 11/36; G06F 15/16; G06F 17/30; G06F 9/44; G06F 9/455; G06F 9/46; G06F 9/50

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: 사물인터넷(IoT), 서버(Server), 작업(Task), 의존성(Dependency), 용량(Capacity), 전송 지연(Transmission delay), 복잡도(Complexity), 분산(distribute), 병렬(parallel)

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
Y	JP 2011-243120 A (NEC CORP.) 2011.12.01 단락 [0018]-[0055]; 청구항 1; 및 도면 1-3	1-14
Y	JP 2016-162067 A (NEC CORP.) 2016.09.05 단락 [0018]-[0024]; 및 청구항 1	1-14
Y	KR 10-2017-0044320 A (주식회사 비디) 2017.04.25 단락 [0059]-[0060]; 및 도면 4	2,6,9,13
Y	KR 10-2015-0009662 A (한국전자통신연구원) 2015.01.27 단락 [0027]-[0036]; 및 도면 1	5,12-13
A	US 2014-0108861 A1 (HADAPT, INC.) 2014.04.17 단락 [0033]-[0048]; 및 도면 1	1-14

☐ 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 이후 “X”에 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

2020년 01월 30일 (30.01.2020)

국제조사보고서 발송일

2020년 01월 30일 (30.01.2020)

ISA/KR의 명칭 및 우편주소



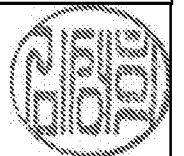
대한민국 특허청
(35208) 대전광역시 서구 청사로 189,
4동 (둔산동, 정부대전청사)

팩스 번호 +82-42-481-8578

심사관

김성희

전화번호 +82-42-481-3516



국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
JP 2011-243120 A	2011/12/01	JP 5482442 B2	2014/05/07
JP 2016-162067 A	2016/09/05	JP 6459633 B2	2019/01/30
KR 10-2017-0044320 A	2017/04/25	KR 10-1794016 B1	2017/11/08
KR 10-2015-0009662 A	2015/01/27	KR 10-1941282 B1	2019/01/23
		US 2015-0026306 A1	2015/01/22
US 2014-0108861 A1	2014/04/17	US 9934276 B2	2018/04/03
		WO 2014-062637 A2	2014/04/24
		WO 2014-062637 A3	2014/09/12