

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일

2024년 12월 12일 (12.12.2024) WIPO | PCT



(10) 국제공개번호

WO 2024/253401 A1

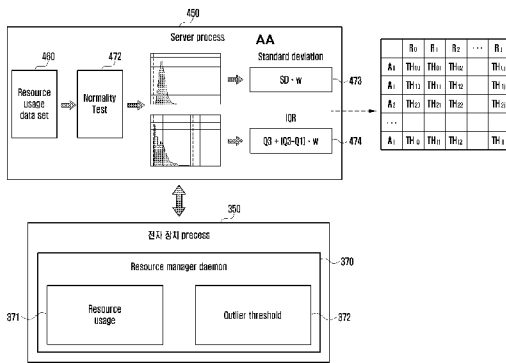
- (51) 국제특허분류:
G06F 11/30 (2006.01) G06F 9/50 (2006.01)
G06F 11/34 (2006.01)
- (21) 국제출원번호: PCT/KR2024/007587
- (22) 국제출원일: 2024년 6월 3일 (03.06.2024)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
10-2023-0072349 2023년 6월 5일 (05.06.2023) KR
10-2023-0082921 2023년 6월 27일 (27.06.2023) KR
- (71) 출원인: 삼성전자 주식회사 (SAMSUNG ELECTRONICS CO., LTD.) [KR/KR]; 16677 경기도 수원시 영통구 삼성로 129, Gyeonggi-do (KR).
- (72) 발명자: 박언상 (PARK, Unsang); 16677 경기도 수원시 영통구 삼성로 129, Gyeonggi-do (KR).
- (74) 대리인: 윤앤리특허법인(유한) (YOON & LEE INTERNATIONAL PATENT & LAW FIRM); 08502 서울특별시 금천구 가산디지털1로 226, 에이스하이엔드타워 5차 3층, 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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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 역내 권리의 보호를 위하여): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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: ELECTRONIC DEVICE AND METHOD FOR DETECTING APPLICATION EXCESSIVELY USING RESOURCE OF ELECTRONIC DEVICE

(54) 발명의 명칭: 전자 장치 및 전자 장치의 리소스를 과다 사용하는 어플리케이션을 감지하는 방법



- 350 ... Electronic device
- 370 ... Resource manager daemon
- 371 ... Resource usage
- 372 ... Outlier threshold
- 450 ... Server process
- 460 ... Resource usage data set
- 472 ... Normality test
- AA ... Standard deviation

(57) Abstract: An electronic device according to various embodiments of the present document may comprise a memory and a processor operatively connected to the memory. The memory may store a resource usage data set in which a plurality of applications and a plurality of resource usage amounts corresponding to the plurality of applications, respectively, are mapped and recorded. The memory may store instructions that, when executed by the processor, cause the electronic device to: if a first application is executed, identify the amount of use of at least one resource used by the executed first application; record information about the identified amount of use of the at least one resource in a resource usage data set stored in the memory; determine at least one outlier threshold value corresponding to the first application and at least one resource used when the first application is executed, respectively, through predetermined statistical analysis on the basis of the resource usage data set stored in the memory; and update at least one outlier threshold value configured with regard to the prestored first application and the at least one resource, respectively, on the basis of the determined at least one outlier threshold value. Various other embodiments are possible.



(57) 요약서: 본 문서의 다양한 실시예에 따른 전자 장치는, 메모리, 및 상기 메모리와 작동적으로 연결되는 프로세서를 포함할 수 있다. 상기 메모리는, 복수의 어플리케이션 및 상기 복수의 어플리케이션 각각에 대응하는 복수의 리소스 사용량을 맵핑하여 기록하는 리소스 사용 데이터 세트를 저장할 수 있다. 상기 메모리는, 상기 프로세서에 의해 실행 시에, 상기 전자 장치가, 제1 어플리케이션이 실행되는 경우, 상기 실행된 제1 어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 상기 확인된 적어도 하나의 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하고, 상기 메모리에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1 어플리케이션 및 상기 제1 어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정하고, 및 상기 결정된 적어도 하나의 이상치 임계 값에 기초하여, 기 저장된 상기 제1 어플리케이션 및 상기 적어도 하나의 리소스 각각에 대해 설정된 적어도 하나의 이상치 임계 값을 업데이트 하도록 하는 인스트럭션들을 저장할 수 있다. 그 외에 다양한 실시예가 가능하다.

명세서

발명의 명칭: 전자 장치 및 전자 장치의 리소스를 과다 사용하는 어플리케이션을 감지하는 방법

기술분야

- [1] 본 문서는 전자 장치에 관한 것이며, 예를 들어, 전자 장치의 리소스를 이용하는 다양한 어플리케이션을 설치 및 실행할 수 있는 전자 장치에 대한 것이다.

배경기술

- [2] 스마트폰, 태블릿 PC, 랩탑 PC, 스마트 와치와 같은 전자 장치는 다양한 종류의 운영 체제(또는 플랫폼)를 포함하며, 각 운영 체제 상에서 실행할 수 있는 다양한 어플리케이션을 설치 및 실행할 수 있다. 전자 장치의 사용자는 온라인 마켓을 통해 어플리케이션을 다운로드 받아 전자 장치 상에 설치할 수 있다. 이와 같은 마켓은 특별한 검증 절차 없이 개발자가 자유롭게 어플리케이션을 마켓에 등록할 수 있는 개방형 구조이다. 어플리케이션을 쉽게 개발할 수 있도록 지원하는 다양한 API(application program interface) 또는 SDK(software development kit)가 제공됨에 따라, 사용자는 직접 필요한 어플리케이션을 개발하여 마켓을 통해 공개할 수도 있다.
- [3] 전자 장치 상에서 어플리케이션을 실행하면 시스템 리소스를 사용할 수 있다. 전자 장치 상에서 다양한 어플리케이션이 동시에 포 그라운드 및 백 그라운드에서 실행될 수 있기 때문에, 특정 어플리케이션이 점유할 수 있는 시스템 리소스의 양은 제한될 수 있다. 마켓을 통해 공개되는 어플리케이션의 종류가 다양해짐에 따라, 이상 동작하는 어플리케이션도 늘어날 수 있다. 예를 들어, 개발자의 전문성 부족으로 인해, 특정 어플리케이션의 경우 시스템 리소스를 과다 사용하여 전자 장치의 전체적인 성능에 영향을 줄 수 있다. 따라서, 전자 장치 상에서 이와 같이 이상 동작하는 어플리케이션을 실시간으로 감지하는 기술이 요구된다.

발명의 상세한 설명

기술적 과제

- [4] 전자 장치는 이상 동작 하는 어플리케이션을 감지하기 위해, 리소스가 임계 값 이하로 부족해지는 지 여부를 모니터링 할 수 있다. 하지만, 종래의 기술은 이상 동작 여부를 결정하기 위한 임계 값을 고정된 값으로 사용 하였으며, 이에 따라 어플리케이션에 따라 실제 이상 동작을 감지하지 못하는 케이스들이 발생할 수 있다.

과제 해결 수단

- [5] 본 문서(disclosure)(또는 명세서(specification), 발명(invention))의 다양한 실시예에 따른 전자 장치는 메모리, 및 상기 메모리와 작동적으로 연결되는 프로세서를 포함할 수 있다.

- [6] 일 실시예에 따르면, 상기 메모리는, 복수의 어플리케이션 및 상기 복수의 어플리케이션 각각에 대응하는 복수의 리소스 사용량을 맵핑하여 기록하는 리소스 사용 데이터 세트를 저장할 수 있다.
- [7] 일 실시예에 따르면, 상기 메모리는, 상기 프로세서에 의해 실행 시에, 상기 전자 장치가, 제1어플리케이션이 실행되는 경우, 상기 실행된 제1어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 상기 확인된 적어도 하나의 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하도록 하는 인스트럭션들을 저장할 수 있다.
- [8] 일 실시예에 따르면, 상기 인스트럭션들은, 상기 전자 장치가, 상기 메모리에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 제1이상치 임계 값을 결정하고, 및 상기 메모리에 기 저장된 상기 제1어플리케이션 및 상기 적어도 하나의 리소스 각각에 대해 설정된 적어도 하나의 제2이상치 임계 값을 상기 결정된 적어도 하나의 제1이상치 임계 값에 기초하여 업데이트 하도록 하는 인스트럭션들을 더 포함할 수 있다.

발명의 효과

- [9] 본 문서의 다양한 실시예에 따르면, 다양한 어플리케이션을 설치 및 실행할 수 있는 전자 장치에서 리소스의 과다 사용의 기준이 되는 임계 값을 통계적 분석을 통해 결정함으로써, 이상 동작하는 어플리케이션을 정확하게 감지할 수 있는 전자 장치 및 전자 장치의 리소스를 과다 사용하는 어플리케이션을 감지하는 방법을 제공할 수 있다.

도면의 간단한 설명

- [10] 도 1은 다양한 실시예들에 따른, 네트워크 환경 내의 전자 장치의 블록도이다.
- [11] 도 2는 일 실시예에 따른 프로그램의 블록도이다.
- [12] 도 3은 일 실시예에 따른 서버 장치 및 복수의 전자 장치를 간략히 도시한 것이다.
- [13] 도 4는 일 실시예에 따른 서버 장치의 블록도이다.
- [14] 도 5는 일 실시예에 따른 서버 장치 및 전자 장치에서 실행되는 프로세스의 블록도이다.
- [15] 도 6은 일 실시예에 따른 리소스 사용 데이터 세트를 도시한 것이다.
- [16] 도 7은 일 실시예에 따른 서버 장치 및 전자 장치에서 실행되는 프로세스의 블록도이다.
- [17] 도 8a 내지 8d는 일 실시예에 따른 전자 장치에서 이상 동작 감지 시 제공하는 UI를 도시한 것이다.
- [18] 도 9a 내지 도 9d는 일 실시예에 따른 일 실시예에 따른 전자 장치가 어플리케이션의 이상 동작을 감지하는 방법의 흐름도이다.

- [19] 도 10은 일 실시예에 따른 서버 장치의 이상치 임계 값을 결정하는 방법의 흐름도이다.
- [20] 도 11은 일 실시예에 따른 전자 장치의 블록도이다.
- [21] 도 12는 일 실시예에 따른 전자 장치에서 실행되는 프로세스의 블록도이다.
- [22] 도 13은 일 실시예에 따른 전자 장치의 이상치 임계 값을 결정하는 방법의 흐름도이다.

발명의 실시를 위한 형태

- [23] 이하에서는 도면을 참조하여 본 문서의 다양한 실시예에 대하여 당해 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 문서에 개시되는 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 도면의 설명과 관련하여, 동일하거나 유사한 구성요소에 대해서는 동일하거나 유사한 참조 부호가 사용될 수 있다. 또한, 도면 및 관련된 설명에서는, 잘 알려진 기능 및 구성에 대한 설명이 명확성과 간결성을 위해 생략될 수 있다.
- [24] 도 1은, 다양한 실시예들에 따른, 네트워크 환경(100) 내의 전자 장치(101)의 블록도이다. 도 1을 참조하면, 네트워크 환경(100)에서 전자 장치(101)는 제 1 네트워크(198)(예: 근거리 무선 통신 네트워크)를 통하여 전자 장치(102)와 통신하거나, 또는 제 2 네트워크(199)(예: 원거리 무선 통신 네트워크)를 통하여 전자 장치(104) 또는 서버(108) 중 적어도 하나와 통신할 수 있다. 일 실시예에 따르면, 전자 장치(101)는 서버(108)를 통하여 전자 장치(104)와 통신할 수 있다. 일 실시예에 따르면, 전자 장치(101)는 프로세서(120), 메모리(130), 입력 모듈(150), 음향 출력 모듈(155), 디스플레이 모듈(160), 오디오 모듈(170), 센서 모듈(176), 인터페이스(177), 연결 단자(178), 햅틱 모듈(179), 카메라 모듈(180), 전력 관리 모듈(188), 배터리(189), 통신 모듈(190), 가입자 식별 모듈(196), 또는 안테나 모듈(197)을 포함할 수 있다. 어떤 실시예에서는, 전자 장치(101)에는, 이 구성요소들 중 적어도 하나(예: 연결 단자(178))가 생략되거나, 하나 이상의 다른 구성요소가 추가될 수 있다. 어떤 실시예에서는, 이 구성요소들 중 일부들(예: 센서 모듈(176), 카메라 모듈(180), 또는 안테나 모듈(197))은 하나의 구성요소(예: 디스플레이 모듈(160))로 통합될 수 있다.
- [25] 프로세서(120)는, 예를 들면, 소프트웨어(예: 프로그램(140))를 실행하여 프로세서(120)에 연결된 전자 장치(101)의 적어도 하나의 다른 구성요소(예: 하드웨어 또는 소프트웨어 구성요소)를 제어할 수 있고, 다양한 데이터 처리 또는 연산을 수행할 수 있다. 일 실시예에 따르면, 데이터 처리 또는 연산의 적어도 일부로서, 프로세서(120)는 다른 구성요소(예: 센서 모듈(176) 또는 통신 모듈(190))로부터 수신된 명령 또는 데이터를 휘발성 메모리(132)에 저장하고, 휘발성 메모리(132)에 저장된 명령 또는 데이터를 처리하고, 결과 데이터를 비휘발성 메모리(134)에 저장할 수 있다. 일 실시예에 따르면, 프로세서(120)는 메인 프로세서(121)(예: 중

양 처리 장치 또는 어플리케이션 프로세서) 또는 이와는 독립적으로 또는 함께 운영 가능한 보조 프로세서(123)(예: 그래픽 처리 장치, 신경망 처리 장치(NPU: neural processing unit), 이미지 시그널 프로세서, 센서 허브 프로세서, 또는 커뮤니케이션 프로세서)를 포함할 수 있다. 예를 들어, 전자 장치(101)가 메인 프로세서(121) 및 보조 프로세서(123)를 포함하는 경우, 보조 프로세서(123)는 메인 프로세서(121)보다 저전력을 사용하거나, 지정된 기능에 특화되도록 설정될 수 있다. 보조 프로세서(123)는 메인 프로세서(121)와 별개로, 또는 그 일부로서 구현될 수 있다.

- [26] 보조 프로세서(123)는, 예를 들면, 메인 프로세서(121)가 인액티브(예: 슬립) 상태에 있는 동안 메인 프로세서(121)를 대신하여, 또는 메인 프로세서(121)가 액티브(예: 어플리케이션 실행) 상태에 있는 동안 메인 프로세서(121)와 함께, 전자 장치(101)의 구성요소들 중 적어도 하나의 구성요소(예: 디스플레이 모듈(160), 센서 모듈(176), 또는 통신 모듈(190))와 관련된 기능 또는 상태들의 적어도 일부를 제어할 수 있다. 일실시예에 따르면, 보조 프로세서(123)(예: 이미지 시그널 프로세서 또는 커뮤니케이션 프로세서)는 기능적으로 관련 있는 다른 구성요소(예: 카메라 모듈(180) 또는 통신 모듈(190))의 일부로서 구현될 수 있다. 일실시예에 따르면, 보조 프로세서(123)(예: 신경망 처리 장치)는 인공지능 모델의 처리에 특화된 하드웨어 구조를 포함할 수 있다. 인공지능 모델은 기계 학습을 통해 생성될 수 있다. 이러한 학습은, 예를 들어, 인공지능 모델이 수행되는 전자 장치(101) 자체에서 수행될 수 있고, 별도의 서버(예: 서버(108))를 통해 수행될 수도 있다. 학습 알고리즘은, 예를 들어, 지도형 학습(supervised learning), 비지도형 학습(unsupervised learning), 준지도형 학습(semi-supervised learning) 또는 강화 학습(reinforcement learning)을 포함할 수 있으나, 전술한 예에 한정되지 않는다. 인공지능 모델은, 복수의 인공 신경망 레이어들을 포함할 수 있다. 인공 신경망은 심층 신경망(DNN: deep neural network), CNN(convolutional neural network), RNN(recurrent neural network), RBM(restricted boltzmann machine), DBN(deep belief network), BRDNN(bidirectional recurrent deep neural network), 심층 Q-네트워크(deep Q-networks) 또는 상기 중 둘 이상의 조합 중 하나일 수 있으나, 전술한 예에 한정되지 않는다. 인공지능 모델은 하드웨어 구조 이외에, 추가적으로 또는 대체적으로, 소프트웨어 구조를 포함할 수 있다.
- [27] 메모리(130)는, 전자 장치(101)의 적어도 하나의 구성요소(예: 프로세서(120) 또는 센서 모듈(176))에 의해 사용되는 다양한 데이터를 저장할 수 있다. 데이터는, 예를 들어, 소프트웨어(예: 프로그램(140)) 및, 이와 관련된 명령에 대한 입력 데이터 또는 출력 데이터를 포함할 수 있다. 메모리(130)는, 휘발성 메모리(132) 또는 비휘발성 메모리(134)를 포함할 수 있다.
- [28] 프로그램(140)은 메모리(130)에 소프트웨어로서 저장될 수 있으며, 예를 들면, 운영 체제(142), 미들 웨어(144) 또는 어플리케이션(146)을 포함할 수 있다.

- [29] 입력 모듈(150)은, 전자 장치(101)의 구성요소(예: 프로세서(120))에 사용될 명령 또는 데이터를 전자 장치(101)의 외부(예: 사용자)로부터 수신할 수 있다. 입력 모듈(150)은, 예를 들면, 마이크, 마우스, 키보드, 키(예: 버튼), 또는 디지털 펜(예: 스타일러스 펜)을 포함할 수 있다.
- [30] 음향 출력 모듈(155)은 음향 신호를 전자 장치(101)의 외부로 출력할 수 있다. 음향 출력 모듈(155)은, 예를 들면, 스피커 또는 리시버를 포함할 수 있다. 스피커는 멀티미디어 재생 또는 녹음 재생과 같이 일반적인 용도로 사용될 수 있다. 리시버는 착신 전화를 수신하기 위해 사용될 수 있다. 일실시예에 따르면, 리시버는 스피커와 별개로, 또는 그 일부로서 구현될 수 있다.
- [31] 디스플레이 모듈(160)은 전자 장치(101)의 외부(예: 사용자)로 정보를 시각적으로 제공할 수 있다. 디스플레이 모듈(160)은, 예를 들면, 디스플레이, 홀로그램 장치, 또는 프로젝터 및 해당 장치를 제어하기 위한 제어 회로를 포함할 수 있다. 일실시예에 따르면, 디스플레이 모듈(160)은 터치를 감지하도록 설정된 터치 센서, 또는 상기 터치에 의해 발생하는 힘의 세기를 측정하도록 설정된 압력 센서를 포함할 수 있다.
- [32] 오디오 모듈(170)은 소리를 전기 신호로 변환시키거나, 반대로 전기 신호를 소리로 변환시킬 수 있다. 일실시예에 따르면, 오디오 모듈(170)은, 입력 모듈(150)을 통해 소리를 획득하거나, 음향 출력 모듈(155), 또는 전자 장치(101)와 직접 또는 무선으로 연결된 외부 전자 장치(예: 전자 장치(102))(예: 스피커 또는 헤드폰)를 통해 소리를 출력할 수 있다.
- [33] 센서 모듈(176)은 전자 장치(101)의 작동 상태(예: 전력 또는 온도), 또는 외부의 환경 상태(예: 사용자 상태)를 감지하고, 감지된 상태에 대응하는 전기 신호 또는 데이터 값을 생성할 수 있다. 일실시예에 따르면, 센서 모듈(176)은, 예를 들면, 제스처 센서, 자이로 센서, 기압 센서, 마그네틱 센서, 가속도 센서, 그립 센서, 근접 센서, 컬러 센서, IR(infrared) 센서, 생체 센서, 온도 센서, 습도 센서, 또는 조도 센서를 포함할 수 있다.
- [34] 인터페이스(177)는 전자 장치(101)가 외부 전자 장치(예: 전자 장치(102))와 직접 또는 무선으로 연결되기 위해 사용될 수 있는 하나 이상의 지정된 프로토콜들을 지원할 수 있다. 일실시예에 따르면, 인터페이스(177)는, 예를 들면, HDMI(high definition multimedia interface), USB(universal serial bus) 인터페이스, SD카드 인터페이스, 또는 오디오 인터페이스를 포함할 수 있다.
- [35] 연결 단자(178)는, 그를 통해서 전자 장치(101)가 외부 전자 장치(예: 전자 장치(102))와 물리적으로 연결될 수 있는 커넥터를 포함할 수 있다. 일실시예에 따르면, 연결 단자(178)는, 예를 들면, HDMI 커넥터, USB 커넥터, SD 카드 커넥터, 또는 오디오 커넥터(예: 헤드폰 커넥터)를 포함할 수 있다.
- [36] 햅틱 모듈(179)은 전기적 신호를 사용자가 촉각 또는 운동 감각을 통해서 인지할 수 있는 기계적인 자극(예: 진동 또는 움직임) 또는 전기적인 자극으로 변환할

- 수 있다. 일실시에에 따르면, 햅틱 모듈(179)은, 예를 들면, 모터, 압전 소자, 또는 전기 자극 장치를 포함할 수 있다.
- [37] 카메라 모듈(180)은 정지 영상 및 동영상을 촬영할 수 있다. 일실시에에 따르면, 카메라 모듈(180)은 하나 이상의 렌즈들, 이미지 센서들, 이미지 시그널 프로세서들, 또는 플래시들을 포함할 수 있다.
- [38] 전력 관리 모듈(188)은 전자 장치(101)에 공급되는 전력을 관리할 수 있다. 일실시에에 따르면, 전력 관리 모듈(188)은, 예를 들면, PMIC(power management integrated circuit)의 적어도 일부로서 구현될 수 있다.
- [39] 배터리(189)는 전자 장치(101)의 적어도 하나의 구성요소에 전력을 공급할 수 있다. 일실시에에 따르면, 배터리(189)는, 예를 들면, 재충전 불가능한 1차 전지, 재충전 가능한 2차 전지 또는 연료 전지를 포함할 수 있다.
- [40] 통신 모듈(190)은 전자 장치(101)와 외부 전자 장치(예: 전자 장치(102), 전자 장치(104), 또는 서버(108)) 간의 직접(예: 유선) 통신 채널 또는 무선 통신 채널의 수립, 및 수립된 통신 채널을 통한 통신 수행을 지원할 수 있다. 통신 모듈(190)은 프로세서(120)(예: 어플리케이션 프로세서)와 독립적으로 운영되고, 직접(예: 유선) 통신 또는 무선 통신을 지원하는 하나 이상의 커뮤니케이션 프로세서를 포함할 수 있다. 일실시에에 따르면, 통신 모듈(190)은 무선 통신 모듈(192)(예: 셀룰러 통신 모듈, 근거리 무선 통신 모듈, 또는 GNSS(global navigation satellite system) 통신 모듈) 또는 유선 통신 모듈(194)(예: LAN(local area network) 통신 모듈, 또는 전력선 통신 모듈)을 포함할 수 있다. 이들 통신 모듈 중 해당하는 통신 모듈은 제 1 네트워크(198)(예: 블루투스, WiFi(wireless fidelity) direct 또는 IrDA(infrared data association)와 같은 근거리 통신 네트워크) 또는 제 2 네트워크(199)(예: 레거시 셀룰러 네트워크, 5G 네트워크, 차세대 통신 네트워크, 인터넷, 또는 컴퓨터 네트워크(예: LAN 또는 WAN)와 같은 원거리 통신 네트워크)를 통하여 외부의 전자 장치(104)와 통신할 수 있다. 이런 여러 종류의 통신 모듈들은 하나의 구성요소(예: 단일 칩)로 통합되거나, 또는 서로 별도의 복수의 구성요소들(예: 복수 칩들)로 구현될 수 있다. 무선 통신 모듈(192)은 가입자 식별 모듈(196)에 저장된 가입자 정보(예: 국제 모바일 가입자 식별자(IMS))를 이용하여 제 1 네트워크(198) 또는 제 2 네트워크(199)와 같은 통신 네트워크 내에서 전자 장치(101)를 확인 또는 인증할 수 있다.
- [41] 무선 통신 모듈(192)은 4G 네트워크 이후의 5G 네트워크 및 차세대 통신 기술, 예를 들어, NR 접속 기술(new radio access technology)을 지원할 수 있다. NR 접속 기술은 고용량 데이터의 고속 전송(eMBB(enhanced mobile broadband)), 단말 전력 최소화 및 다수 단말의 접속(mMTC(massive machine type communications)), 또는 고신뢰도와 저지연(URLLC(ultra-reliable and low-latency communications))을 지원할 수 있다. 무선 통신 모듈(192)은, 예를 들어, 높은 데이터 전송률 달성을 위해, 고주파 대역(예: mmWave 대역)을 지원할 수 있다. 무선 통신 모듈(192)은 고주파 대역에서의 성능 확보를 위한 다양한 기술들, 예를 들어, 빔포밍

(beamforming), 거대 배열 다중 입출력(massive MIMO(multiple-input and multiple-output)), 전차원 다중입출력(FD-MIMO: full dimensional MIMO), 어레이 안테나(array antenna), 아날로그 빔형성(analog beam-forming), 또는 대규모 안테나(large scale antenna)와 같은 기술들을 지원할 수 있다. 무선 통신 모듈(192)은 전자 장치(101), 외부 전자 장치(예: 전자 장치(104)) 또는 네트워크 시스템(예: 제 2 네트워크(199))에 규정되는 다양한 요구사항을 지원할 수 있다. 일실시예에 따르면, 무선 통신 모듈(192)은 eMBB 실현을 위한 Peak data rate(예: 20Gbps 이상), mMTC 실현을 위한 손실 Coverage(예: 164dB 이하), 또는 URLLC 실현을 위한 U-plane latency(예: 다운링크(DL) 및 업링크(UL) 각각 0.5ms 이하, 또는 라운드 트립 1ms 이하)를 지원할 수 있다.

- [42] 안테나 모듈(197)은 신호 또는 전력을 외부(예: 외부의 전자 장치)로 송신하거나 외부로부터 수신할 수 있다. 일실시예에 따르면, 안테나 모듈(197)은 서브스트레이트(예: PCB) 위에 형성된 도전체 또는 도전성 패턴으로 이루어진 방사체를 포함하는 안테나를 포함할 수 있다. 일실시예에 따르면, 안테나 모듈(197)은 복수의 안테나들(예: 어레이 안테나)을 포함할 수 있다. 이런 경우, 제 1 네트워크(198) 또는 제 2 네트워크(199)와 같은 통신 네트워크에서 사용되는 통신 방식에 적합한 적어도 하나의 안테나가, 예를 들면, 통신 모듈(190)에 의하여 상기 복수의 안테나들로부터 선택될 수 있다. 신호 또는 전력은 상기 선택된 적어도 하나의 안테나를 통하여 통신 모듈(190)과 외부의 전자 장치 간에 송신되거나 수신될 수 있다. 어떤 실시예에 따르면, 방사체 이외에 다른 부품(예: RFIC(radio frequency integrated circuit))이 추가로 안테나 모듈(197)의 일부로 형성될 수 있다.
- [43] 다양한 실시예에 따르면, 안테나 모듈(197)은 mmWave 안테나 모듈을 형성할 수 있다. 일실시예에 따르면, mmWave 안테나 모듈은 인쇄 회로 기판, 상기 인쇄 회로 기판의 제 1 면(예: 아래 면)에 또는 그에 인접하여 배치되고 지정된 고주파 대역(예: mmWave 대역)을 지원할 수 있는 RFIC, 및 상기 인쇄 회로 기판의 제 2 면(예: 윗 면 또는 측면)에 또는 그에 인접하여 배치되고 상기 지정된 고주파 대역의 신호를 송신 또는 수신할 수 있는 복수의 안테나들(예: 어레이 안테나)을 포함할 수 있다.
- [44] 상기 구성요소들 중 적어도 일부는 주변 기기들간 통신 방식(예: 버스, GPIO(general purpose input and output), SPI(serial peripheral interface), 또는 MIPI(mobile industry processor interface))을 통해 서로 연결되고 신호(예: 명령 또는 데이터)를 상호간에 교환할 수 있다.
- [45] 일실시예에 따르면, 명령 또는 데이터는 제 2 네트워크(199)에 연결된 서버(108)를 통해서 전자 장치(101)와 외부의 전자 장치(104)간에 송신 또는 수신될 수 있다. 외부의 전자 장치(102, 또는 104) 각각은 전자 장치(101)와 동일한 또는 다른 종류의 장치일 수 있다. 일실시예에 따르면, 전자 장치(101)에서 실행되는 동작들의 전부 또는 일부는 외부의 전자 장치들(102, 104, 또는 108) 중 하나 이상의 외부의 전자 장치들에서 실행될 수 있다. 예를 들면, 전자 장치(101)가 어떤 기

능이나 서비스를 자동으로, 또는 사용자 또는 다른 장치로부터의 요청에 반응하여 수행해야 할 경우에, 전자 장치(101)는 기능 또는 서비스를 자체적으로 실행시키는 대신에 또는 추가적으로, 하나 이상의 외부의 전자 장치들에게 그 기능 또는 그 서비스의 적어도 일부를 수행하라고 요청할 수 있다. 상기 요청을 수신한 하나 이상의 외부의 전자 장치들은 요청된 기능 또는 서비스의 적어도 일부, 또는 상기 요청과 관련된 추가 기능 또는 서비스를 실행하고, 그 실행의 결과를 전자 장치(101)로 전달할 수 있다. 전자 장치(101)는 상기 결과를, 그대로 또는 추가적으로 처리하여, 상기 요청에 대한 응답의 적어도 일부로서 제공할 수 있다. 이를 위하여, 예를 들면, 클라우드 컴퓨팅, 분산 컴퓨팅, 모바일 에지 컴퓨팅(MEC: mobile edge computing), 또는 클라이언트-서버 컴퓨팅 기술이 이용될 수 있다. 전자 장치(101)는, 예를 들어, 분산 컴퓨팅 또는 모바일 에지 컴퓨팅을 이용하여 초저지연 서비스를 제공할 수 있다. 다른 실시예에 있어서, 외부의 전자 장치(104)는 IoT(internet of things) 기기를 포함할 수 있다. 서버(108)는 기계 학습 및/또는 신경망을 이용한 지능형 서버일 수 있다. 일 실시예에 따르면, 외부의 전자 장치(104) 또는 서버(108)는 제 2 네트워크(199) 내에 포함될 수 있다. 전자 장치(101)는 5G 통신 기술 및 IoT 관련 기술을 기반으로 지능형 서비스(예: 스마트 홈, 스마트 시티, 스마트 카, 또는 헬스케어)에 적용될 수 있다.

[46] 도 2은 다양한 실시예에 따른 프로그램 모듈의 블록도이다.

[47] 일 실시예에 따르면, 프로그램(140)은 전자 장치(101)의 하나 이상의 리소스들을 제어하기 위한 운영 체제(142), 미들 웨어(144), 또는 상기 운영 체제(142) 상에서 실행 가능한 어플리케이션(146)을 포함할 수 있다. 운영 체제(142)는, 예를 들면, Android™, iOS™, Windows™, Symbian™, Tizen™, 또는 Bada™를 포함할 수 있다. 프로그램(140) 중 적어도 일부 프로그램은, 예를 들면, 제조 시에 전자 장치(101)에 프리로드되거나, 또는 사용자의 사용 환경에서 외부 전자 장치(예: 전자 장치(102) 또는 104), 또는 서버(108))로부터 다운로드되거나 갱신 될 수 있다.

[48] 운영 체제(142)는 전자 장치(101)의 시스템 리소스(예: 프로세스, 메모리, 또는 전원)를 제어(예: 할당 또는 회수)할 수 있다. 운영 체제(142)는, 추가적으로 또는 대체적으로, 전자 장치(101)의 다른 하드웨어 디바이스, 예를 들면, 입력 장치(150), 음향 출력 장치(155), 표시 장치(160), 오디오 모듈(170), 센서 모듈(176), 인터페이스(177), 햅틱 모듈(179), 카메라 모듈(180), 전력 관리 모듈(188), 배터리(189), 통신 모듈(190), 가입자 식별 모듈(196), 또는 안테나 모듈(197)을 구동하기 위한 하나 이상의 드라이버 프로그램들을 포함할 수 있다.

[49] 미들 웨어(144)는 어플리케이션(146)이 전자 장치(101)의 하나 이상의 리소스들이 제공하는 기능 또는 정보를 사용할 수 있도록 다양한 기능들을 어플리케이션(146)으로 제공할 수 있다. 미들 웨어(144)는, 예를 들면, 어플리케이션 매니저(201), 윈도우 매니저(203), 멀티미디어 매니저(205), 리소스 매니저(207), 파워 매니저(209), 데이터베이스 매니저(211), 패키지 매니저(213), 커넥티비티 매니저(215), noti피케이션 매니저(217), 로케이션 매니저(219), 그래픽 매니저(221), 시

큐리티 매니저(223), 통화 매니저(225), 또는 음성 인식 매니저(227)를 포함할 수 있다. 어플리케이션 매니저(201)는, 예를 들면, 어플리케이션(146)의 생명 주기를 관리할 수 있다. 윈도우 매니저(203)는, 예를 들면, 화면에서 사용되는 GUI 자원을 관리할 수 있다. 멀티미디어 매니저(205)는, 예를 들면, 미디어 파일들의 재생에 필요한 포맷을 파악하고, 해당 포맷에 맞는 코덱을 이용하여 미디어 파일의 인코딩 또는 디코딩을 수행할 수 있다. 리소스 매니저(207)는, 예를 들면, 어플리케이션(146)의 소스 코드 또는 메모리의 공간을 관리할 수 있다. 파워 매니저(209)는, 예를 들면, 배터리의 용량, 온도 또는 전원을 관리하고, 이 중 해당 정보를 이용하여 전자 장치(101)의 동작에 필요한 전력 정보를 결정 또는 제공할 수 있다. 일 실시예에 따르면, 파워 매니저(209)는 바이오스(BIOS: basic input/output system)와 연동할 수 있다.

- [50] 데이터베이스 매니저(211)는, 예를 들면, 어플리케이션(146)에서 사용될 데이터베이스를 생성, 검색, 또는 변경할 수 있다. 패키지 매니저(213)는, 예를 들면, 패키지 파일의 형태로 배포되는 어플리케이션의 설치 또는 갱신을 관리할 수 있다. 커넥티비티 매니저(215)는, 예를 들면, 전자 장치(101)와 외부 전자 장치 간의 무선 또는 유선 연결을 관리할 수 있다. 노티피케이션 매니저(217)는, 예를 들면, 발생된 이벤트(예: 통화, 메시지, 또는 알람)를 사용자에게 알리기 위한 기능을 제공할 수 있다. 로케이션 매니저(219)는, 예를 들면, 전자 장치(101)의 위치 정보를 관리할 수 있다. 그래픽 매니저(221)는, 예를 들면, 사용자에게 제공될 그래픽 효과 또는 이와 관련된 사용자 인터페이스를 관리할 수 있다. 시큐리티 매니저(223)는, 예를 들면, 시스템 보안 또는 사용자 인증을 제공할 수 있다. 통화(telephony) 매니저(225)는, 예를 들면, 전자 장치(101)의 음성 통화 또는 영상 통화 기능을 관리할 수 있다. 음성 인식 매니저(227)는, 예를 들면, 사용자의 음성 데이터를 서버(108)로 전송하고, 해당 음성 데이터에 기반하여 전자 장치(101)에서 수행될 기능에 대응하는 명령어(command) 또는 해당 음성 데이터에 기반하여 변환된 문자 데이터를 수신할 수 있다. 일 실시예에 따르면, 미들웨어(244)는 동적으로 기존의 구성요소를 일부 삭제하거나 새로운 구성요소들을 추가할 수 있다. 일 실시예에 따르면, 미들웨어(144)의 적어도 일부는 운영 체제(142)의 일부로 포함되거나, 또는 운영 체제(142)와는 별도의 소프트웨어로 구현될 수 있다.

- [51] 어플리케이션(146)은, 예를 들면, 홈(251), 다이얼러(253), SMS/MMS(255), IM(instant message)(257), 브라우저(259), 카메라(261), 알람(263), 연락처(265), 음성 인식(267), 이메일(269), 달력(271), 미디어 플레이어(273), 앨범(275), 워치(277), 헬스(279)(예: 운동량 또는 혈당 등을 측정), 또는 환경 정보(281)(예: 기압, 습도, 또는 온도 정보) 어플리케이션을 포함할 수 있다. 일 실시예에 따르면, 어플리케이션(146)은 전자 장치(101)와 외부 전자 장치 사이의 정보 교환을 지원할 수 있는 정보 교환 어플리케이션(미도시)을 더 포함할 수 있다. 정보 교환 어플리케이션은, 예를 들면, 외부 전자 장치로 지정된 정보(예: 통화, 메시지, 또는 알람)를 전달하기 위한 노티피케이션 릴레이 어플리케이션, 또는 외부 전자 장치를 관

리하기 위한 장치 관리 어플리케이션을 포함할 수 있다. noti피케이션 릴레이 어플리케이션은, 예를 들면, 전자 장치(101)의 다른 어플리케이션(예: 이메일 어플리케이션(269))에서 발생된 이벤트(예: 메일 수신)에 대응하는 알림 정보를 외부 전자 장치로 전달하거나, 또는 외부 전자 장치로부터 알림 정보를 수신하여 전자 장치(101)의 사용자에게 제공할 수 있다. 장치 관리 어플리케이션은, 예를 들면, 전자 장치(101)와 통신하는 외부 전자 장치 또는 그 일부 구성 요소(예: 표시 장치(160) 또는 카메라 모듈(180))의 전원(예: 턴-온 또는 턴-오프) 또는 기능(예: 표시 장치(160) 또는 카메라 모듈(180)의 밝기, 해상도, 또는 포커스)을 제어할 수 있다. 장치 관리 어플리케이션은, 추가적으로 또는 대체적으로, 외부 전자 장치에서 동작하는 어플리케이션의 설치, 삭제, 또는 갱신을 지원할 수 있다.

- [52] 도 3은 일 실시예에 따른 서버 장치 및 복수의 전자 장치를 간략히 도시한 것이다.
- [53] 도 3을 참조하면, 서버 장치(400)는 네트워크를 통해 복수의 전자 장치(300)와 연결될 수 있다. 전자 장치(300)는 스마트폰, 태블릿 PC, 랩탑 PC와 같은 장치일 수 있으며, 그 종류에는 정함이 없다. 전자 장치(300)는 도 1의 전자 장치(101)의 구성 및/또는 기능 중 적어도 일부를 포함할 수 있다.
- [54] 일 실시예에 따르면, 전자 장치(300)는 다양한 어플리케이션을 설치 및 실행할 수 있다. 예를 들어, 전자 장치(300)는 도 2의 어플리케이션(146) 중 적어도 일부를 실행할 수 있으며, 또한 어플리케이션 스토어(예: 플레이 스토어™)를 통해 다운로드 받아 설치한 적어도 하나의 어플리케이션을 실행할 수 있다. 전자 장치(300)는 포그라운드 및 백그라운드에서 복수의 어플리케이션을 동시에 실행할 수 있다.
- [55] 일 실시예에 따르면, 전자 장치(300)가 특정 어플리케이션을 실행하는 경우, 실행된 어플리케이션은 전자 장치(300)의 리소스(resource)(또는 시스템 리소스)를 사용할 수 있다. 여기서, 어플리케이션이 사용하는 전자 장치(300)의 리소스는 CPU(central processing unit), 메모리, I/O(input/output), 스레드(thread), 팝업 윈도우(popup window), 가상 디스플레이(virtual display), 알림(notification), 펜딩 인텐트(pending intent)를 예로 들 수 있으며, 이에 한정되지는 않는다. 실행되는 어플리케이션에 따라 사용하는 리소스의 종류는 다를 수 있으며, 동일한 어플리케이션의 경우에도 동작 상태 및/또는 환경 설정에 따라 사용하는 리소스의 종류가 다를 수 있다.
- [56] 일 실시예에 따르면, 전자 장치(300) 상에서 실행 중인 어플리케이션이 특정 리소스를 과도하게 사용하는 경우, 이상 동작이 발생할 수 있다. 어플리케이션 스토어에 업로드된 어플리케이션은 성능 및 안정성을 담보하지 않으며, 일부 어플리케이션은 일부 리소스를 과도하게 사용하는 이상 동작을 발생할 수 있다. 또한, 전자 장치(300)마다 하드웨어 성능 및 환경 설정이 상이하기 때문에, 대부분 전자 장치(300) 상에서 잘 동작하는 어플리케이션도 일부 전자 장치(300)에서 이상 동작을 발생시킬 수 있다.

- [57] 일 실시예에 따르면, 서버 장치(400)는 복수의 전자 장치(300)로부터 리소스 사용량 정보를 수신하고, 수신된 리소스 사용량 정보를 리소스 사용 데이터 세트에 누적 저장하고, 누적 저장된 리소스 사용 데이터 세트에 기초하여, 이상치 임계 값을 계산할 수 있다. 예를 들어, 전자 장치(300)가 특정 어플리케이션을 실행하는 경우, 적어도 하나의 리소스를 사용하게 되며, 실행된 어플리케이션 및 사용된 적어도 하나의 리소스의 사용량을 맵핑하여 저장할 수 있다. 전자 장치(300)는 어플리케이션이 리소스를 사용하는 경우마다 및/또는 서버 장치(400)로부터 요청을 수신하는 경우, 리소스 사용량 정보를 서버 장치(400)로 전송할 수 있다.
- [58] 일 실시예에 따르면, 서버 장치(400)는 여러 전자 장치(300)의 리소스 사용량 정보를 포함하는 리소스 사용 데이터 세트에 대해 통계적 분석을 이용하여, 이상치 임계 값을 계산할 수 있다. 서버 장치(400)는 각각의 어플리케이션 및 각각의 리소스에 대해 이상치 임계 값을 결정할 수 있다. 예를 들어, SNS 어플리케이션의 CPU 사용량, 메모리 사용량, pending intent 생성 개수 각각에 대해 통계적 분석을 통해 이상치 임계 값을 결정할 수 있다. 서버 장치(400)는 통계적 분석 시, IQR(inter quartile range) 및/또는 표준 편차(standard deviation)를 이용하여, 일반적인 사용량에 대비하여 오차가 큰 값을 이상치 임계 값으로 설정할 수 있다.
- [59] 일 실시예에 따르면, 서버 장치(400)는 계산된 이상치 임계 값을 각 전자 장치(300)에 제공할 수 있다. 전자 장치(300)는 서버 장치(400)로부터 수신한 이상치 임계 값으로 기 저장된 이상치 임계 값을 업데이트 할 수 있다. 전자 장치(300)는 특정 어플리케이션의 실행 시, 적어도 하나의 리소스의 사용량을 확인하고, 각각의 리소스의 사용량이 각 리소스의 이상치 임계 값 이상인지 확인할 수 있다. 예를 들어, SNS 어플리케이션이 실행되는 경우, CPU 사용량을 CPU 사용량의 이상치 임계 값과 비교하고, 메모리 사용량을 메모리 사용량의 이상치 임계 값과 비교하고, 및 pending intent 생성 개수를 pending intent 생성 개수의 이상치 임계 값과 비교할 수 있다.
- [60] 일 실시예에 따르면, 전자 장치(300)는 어플리케이션의 실행 시 사용되는 특정 리소스의 사용량이 이상치 임계 값 이상인 경우, 해당 어플리케이션의 실행으로 인해 이상 동작이 발생한 것으로 결정할 수 있다. 전자 장치(300)는 이상 동작이 발생한 것으로 결정되는 경우, 사용자에게 피드백을 제공하거나, 해당 어플리케이션을 강제 종료하거나, 및/또는 해당 어플리케이션을 삭제할 수 있다.
- [61] 일 실시예에 따르면, 이상치 임계 값을 결정하는 동작은 서버 장치(400)가 아닌 각각의 전자 장치(300)에 의해 수행될 수도 있다. 예를 들어, 전자 장치(300)는 메모리에 리소스 사용 데이터 세트를 저장하고, 각각의 어플리케이션이 실행되는 경우, 리소스 사용량 정보를 리소스 사용 데이터 세트에 기록할 수 있다. 전자 장치(300)는 리소스 사용 데이터 세트에 충분한 데이터가 수집되는 경우, 및/또는 정해진 주기마다, 통계적 분석을 통해 각 어플리케이션의 각 리소스에 대한 이상치 임계 값을 결정할 수 있다.

- [62] 본 문서에서는, 서버 장치(400)가 이상치 임계 값을 결정하는 실시예를 제1실시예로, 각 전자 장치(300)가 이상치 임계 값을 결정하는 실시예를 제2실시예로 지칭할 수 있다. 이하에서는 도 4 내지 도 10을 통해 제1실시예에 대해 설명하고, 도 11 내지 도 13을 통해 제2실시예에 대해 설명하기로 한다. 각 실시예에서 중복되거나, 다른 실시예의 내용으로부터 유추 가능한 내용은 그 설명을 생략할 수도 있다.
- [63] 도 4는 일 실시예에 따른 서버 장치의 블록도이다.
- [64] 도 4를 참조하면, 서버 장치(400)는 통신 인터페이스(430), 프로세서(410) 및 메모리(420)를 포함할 수 있다. 도시된 구성 중 일부가 생략 또는 치환되더라도 본 문서의 다양한 실시예들을 구현할 수 있다.
- [65] 일 실시예에 따르면, 통신 인터페이스(430)는 네트워크를 통한 다양한 전자 장치(예: 도 3의 전자 장치(300))와의 통신을 지원할 수 있다. 통신 인터페이스(430)는 HTTP(hypertext transfer protocol), REST(representational state transfer), MQTT(message queuing telemetry transport), 소켓(socket)과 같은 다양한 인터페이스를 제공할 수 있다. 서버 장치(400)는 통신 인터페이스(430)를 통해 전자 장치에 리소스 사용량 정보의 전송을 요청하고, 전자 장치로부터 리소스 사용량 정보를 수신할 수 있다.
- [66] 일 실시예에 따르면, 메모리(420)는 다양한 데이터를 일시적 또는 비일시적으로 저장할 수 있다. 메모리(420)는 RAM(random access memory), 가상 메모리, 캐시 메모리, 및/또는 플래시 메모리와 같은 다양한 종류의 메모리들을 포함할 수 있다. 메모리(420)는 서버 장치(400)(또는 프로세서(410))에 의해 실행될 수 있는 다양한 인스트럭션(instructions)들을 저장할 수 있다.
- [67] 일 실시예에 따르면, 메모리(420)는 복수의 전자 장치로부터 수신한 리소스 사용량 정보를 누적하여 저장하는 리소스 사용 데이터 세트를 저장할 수 있다. 리소스 사용 데이터 세트는 각각의 어플리케이션과 각각의 리소스의 사용량 정보를 맵핑해서 저장할 수 있다. 리소스 사용 데이터 세트는 전자 장치로부터 수신한 리소스 사용량의 RAW_DATA를 누적 저장하는 RAW_DATA set를 포함하며, 추가적으로, 리소스 사용량과 관련된 COUNT, SUM, SQUARE_SUM 및 FREQUENCY_TABLE과 같은 데이터를 누적 저장하는 COUNT set, SUM set, SQUARE_SUM set 및 FREQUENCY_TABLE set를 더 포함할 수 있다.
- [68] 일 실시예에 따르면, 프로세서(410)는 서버 장치(400)의 각 구성 요소들의 제어 및/또는 통신에 관한 연산이나 데이터 처리를 수행할 수 있는 구성으로써, 하나 이상의 프로세서들로 구성될 수 있다. 후술할 프로세서(410)(또는 서버 장치(400))의 동작들은 메모리(420)에 저장된 인스트럭션들을 실행함으로써 수행될 수 있다.
- [69] 일 실시예에 따르면, 프로세서(410)는 전자 장치로부터 수신한 리소스 사용 정보를 리소스 사용 데이터 세트에 저장하는 동작 및 리소스 사용 데이터 세트에

저장된 데이터로부터 이상치 임계 값을 생성하는 동작을 수행하기 위한 다양한 서버 프로세스 또는 스레드들을 실행할 수 있다.

- [70] 일 실시예에 따르면, 프로세서(410)는 통신 인터페이스(430)를 통해 전자 장치에 리소스 사용량 정보의 전송을 요청할 수 있다. 예를 들어, 프로세서(410)는 정해진 주기마다, 또는 리소스 사용 데이터 세트에 누적 기록된 데이터의 양에 기초하여 각각의 전자 장치에 리소스 사용량 정보의 전송을 요청할 수 있다.
- [71] 일 실시예에 따르면, 프로세서(410)는 통신 인터페이스(430)를 통해 전자 장치로부터 리소스 사용량 정보가 수신되는 경우, 이를 메모리(420)의 리소스 사용 데이터 세트에 저장할 수 있다. 리소스 사용 데이터 세트에 누적 저장되는 데이터의 구조에 대해서는 도 6을 통해 보다 상세히 설명하기로 한다.
- [72] 일 실시예에 따르면, 프로세서(410)는 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 특정 어플리케이션 및 특정 어플리케이션의 실행시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정할 수 있다. 프로세서(410)는 복수의 어플리케이션 각각에 대해 리소스 각각에 대응하는 이상치 임계 값을 결정하는 동작을 수행할 수 있다. i 개의 어플리케이션 및 j 개의 리소스에 대한 이상치 임계 값을 구하는 경우, App_0 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산하고, App_1 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산하고, ..., App_i 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산할 수 있다.
- [73] 일 실시예에 따르면, 프로세서(410)는 리소스 사용 데이터 세트의 특정 어플리케이션 및 특정 리소스의 데이터에 대하여, 정규성을 검증할 수 있다. 여기서, 정규성 검증은 Kolmogorov-Smirnov test, Shapiro-Wilk test, 및/또는 왜도, 첨도, 표준오차 검증에 기초할 수 있으며, 이에 한정되지는 않는다.
- [74] 일 실시예에 따르면, 프로세서(410)는 특정 리소스의 리소스 사용 데이터 세트의 데이터가 정규 분포를 따르는 경우, 해당 데이터들의 평균 및 표준 편차를 계산할 수 있다. 프로세서(410)는 표준 편차 및 해당 리소스에 대해 미리 정해진 가중치(예: w_1)에 기초하여 이상치 임계 값을 결정할 수 있다. 예를 들어, 프로세서(410)는 표준 편차와 가중치의 곱으로 이상치 임계 값을 결정할 수 있다. 여기서, 가중치 w_1 은 각 리소스 별로 서로 다른 값으로 결정될 수 있다.
- [75] 일 실시예에 따르면, 프로세서(410)는 특정 리소스의 리소스 사용 데이터 세트의 데이터가 정규 분포를 따르지 않는 경우, IQR(inter quartile range)을 계산하고, IQR 및 해당 리소스에 대해 미리 정해진 가중치에 기초하여 이상치 임계 값을 결정할 수 있다. 여기서, IQR은 각 데이터를 그 값에 따라 순차적으로(예: 오름차순으로) 정렬한 후, 하위 25%에 해당하는 데이터와 상위 25%에 해당하는 데이터 사이의 차이로 결정될 수 있다. 프로세서(410)는 Q3 값과 Q1 값을 구한 후 그 차

이로 IQR을 구할 수 있다. 프로세서(410)는 IQR과 가중치(예: w_2)를 곱한 값을 Q3 값에 더한 값을 이상치 임계 값으로 결정할 수 있다. 여기서, 가중치 w_2 는 각 리소스 별로 서로 다른 값으로 결정될 수 있다.

- [76] 일 실시예에 따르면, 프로세서(410)는 데이터에 대한 정규성 검증 없이, 모든 데이터에 대해 IQR 및 가중치 w_2 에 기초하여 이상치 임계 값을 계산할 수도 있다.
- [77] 일 실시예에 따르면, 프로세서(410)는 계산된 이상치 임계 값을 통신 인터페이스(430)를 통해 각각의 전자 장치로 전송할 수 있다. 전자 장치는 수신한 이상치 임계 값을 저장하고, 어플리케이션의 실행 시 사용된 각 리소스의 사용량을 각 이상치 임계 값과 비교하여, 이상 동작 여부를 결정할 수 있다.
- [78] 도 5는 일 실시예에 따른 서버 장치 및 전자 장치에서 실행되는 프로세스의 블록도이다.
- [79] 일 실시예에 따르면, 서버 장치(예: 도 3 및 도 4의 서버 장치(400))에서 실행되는 서버 프로세스(450)는 이상치 임계 값 계산기(outlier threshold calculator)(470)를 포함하고, 이상치 임계 값 계산기(470)는 서버 장치의 메모리(예: 도 4의 메모리(420))에 저장된 리소스 사용 데이터 세트(460)에 액세스 할 수 있다.
- [80] 일 실시예에 따르면, 전자 장치(예: 도 3의 전자 장치(300))에서 실행되는 전자 장치 프로세스(350)는 리소스 생성기(resource generator)(360) 및 리소스 매니저 데몬(resource manager daemon)(370)을 포함할 수 있다.
- [81] 일 실시예에 따르면, 리소스 생성기(360)는 어플리케이션이 실행되어 적어도 하나의 리소스를 사용하는 경우, 어플리케이션의 리소스 사용에 대응하는 이벤트를 생성하여, 리소스 매니저 데몬(370)으로 전달할 수 있다. 리소스 생성기(360)는 클래스(예: ResEvt.class(.java))(361), JNI(java native interface) 라이브러리(예: ResEvt JNI library(.cpp))(362) 및/또는 SO(shared object)(예: LibresEvt.so)(363) 형태로 제공될 수 있다. 리소스 생성기(360)는 시스템 프로세스의 일부로 실행될 수 있으며, 시스템 프로세스에서 리소스를 이용 중인 상황을 플러그-인(plug-in)을 통해 이벤트로 생성하여 리소스 매니저 데몬(370)으로 전달할 수 있다.
- [82] 일 실시예에 따르면, 리소스 매니저 데몬(370)은 리소스 사용 통계(371)를 기록하고, 이상치 임계 값(372)을 관리할 수 있다. 리소스 매니저 데몬(370)에서 리소스 생성기(360)로부터 어플리케이션 및 리소스 사용량 정보를 수신하고, 이를 리소스 사용 통계(371)에 기록할 수 있다. 리소스 사용 통계(371)에는 복수의 어플리케이션 및 각 어플리케이션이 사용한 각각의 리소스의 사용량 정보가 테이블 형태로 저장될 수 있다.
- [83] 일 실시예에 따르면, 리소스 매니저 데몬(370)은 리소스 생성기(360)로부터 이벤트를 수신하는 경우, 각각의 리소스 사용량을 각 리소스에 대응하는 이상치 임계 값과 비교할 수 있다. 비교 결과, 해당 이벤트의 어플리케이션이 특정 리소스를 이상치 임계 값 이상으로 사용한 것으로 확인되는 경우, 리소스 매니저 데몬(370)은 해당 어플리케이션이 리소스를 과다 사용하여 이상 동작하는 것으로 결

- 정할 수 있다. 이 경우, 리소스 매니저 데몬(370)은 해당 이벤트에 포함된 각 리소스의 사용량 정보를 리소스 사용 통계(371)에 기록하지 않을 수 있다.
- [84] 일 실시예에 따르면, 리소스 매니저 데몬(370)은 서버 장치로부터 데이터 전송 요청이 있는 경우, 및/또는 어플리케이션의 리소스 사용 이벤트가 전달될 때마다, 리소스 사용 통계에 누적 저장된 리소스 사용량 통계(371)를 서버 장치로 전송할 수 있다.
- [85] 일 실시예에 따르면, 리소스 매니저 데몬(370)은 특정 어플리케이션이 이상 동작하는 것으로 결정하는 경우, 해당 어플리케이션의 동작을 중지하고(또는 실행을 종료하고), 리소스를 회수할 수 있다. 리소스 매니저 데몬(370)은 해당 어플리케이션의 동작을 중지한 후 출력 장치(예: 디스플레이, 스피커, 햅틱)를 통해 사용자에게 이상 동작을 지시하는 알림을 제공하고, 해당 어플리케이션을 재시작하도록 유도할 수 있다. 리소스 매니저 데몬(370)은 해당 어플리케이션의 재시작 이후에도 리소스를 과다하게 사용하는 문제가 기준 횟수 이상으로 반복해서 발생하는 경우, 출력 장치를 통해 사용자에게 해당 어플리케이션의 삭제를 유도하는 사용자 인터페이스를 제공할 수 있다.
- [86] 일 실시예에 따르면, 리소스 매니저 데몬(370)은 서버 장치로부터 수신한 이상치 임계 값을 저장하고, 서버 장치로부터 새로운 이상치 임계 값이 수신되는 경우 미리 저장된 이상치 임계 값(372)을 업데이트 할 수 있다.
- [87] 일 실시예에 따르면, 서버 장치는 전자 장치로부터 리소스 사용량 정보가 수신되는 경우, 리소스 사용 데이터 세트(460)에 저장할 수 있다. 리소스 사용 데이터 세트(460)는 복수의 전자 장치로부터 획득된 리소스 사용량 정보를 누적해서 저장할 수 있으며, 각각의 어플리케이션과 각각의 리소스의 사용량 정보를 맵핑해서 저장할 수 있다.
- [88] 일 실시예에 따르면, 이상치 임계 값 계산기(470)는 리소스 사용 데이터 세트(460)에 기초하여, 미리 정해진 통계적 분석을 통해 특정 어플리케이션 및 특정 어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정할 수 있다. 이상치 임계 값 계산기(470)는 복수의 어플리케이션 각각에 대해 리소스 각각에 대응하는 이상치 임계 값을 결정하는 동작을 수행할 수 있다. 예를 들어, i 개의 어플리케이션 및 j 개의 리소스에 대한 이상치 임계 값을 구하는 경우, App_0 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산하고, App_1 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산하고, ..., App_i 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산할 수 있다.
- [89] 일 실시예에 따르면, 이상치 임계 값 계산기(470)는 특정 어플리케이션의 특정 리소스의 사용량 데이터의 정규성을 테스트하여, 정규 분포를 따르는 지 여부를 확인할 수 있다. 데이터가 정규 분포를 따르는 경우, 이상치 임계 값 계산기(470)

는 표준 편차를 계산하고, 표준 편차 및 미리 정해진 가중치에 기초하여 이상치 임계 값을 계산할 수 있다. 데이터가 정규 분포를 따르지 않는 경우, 이상치 임계 값 계산기(470)는 IQR(inter quartile range)를 계산하고, IQR 및 미리 정해진 가중치에 기초하여 이상치 임계 값을 계산할 수 있다.

- [90] 도 6은 일 실시예에 따른 리소스 사용 데이터 세트를 도시한 것이다.
- [91] 일 실시예에 따르면, 서버 장치는 복수의 전자 장치로부터 수신되는 리소스 사용량 정보를 리소스 사용 데이터 세트(460)에 누적해서 저장할 수 있다. 도 6은 각 전자 장치에서 실행되는 어플리케이션이 사용하는 리소스의 사용 통계를 모으기 위한 리소스 사용 데이터 세트(460)의 자료 구조를 도시하고 있다.
- [92] 도 6을 참조하면, 리소스 사용 데이터 세트(460)의 각 행은 전자 장치 상에서 실행될 수 있는 복수의 어플리케이션을 나타내고, 각 열은 각 어플리케이션이 사용할 수 있는 리소스(resource)를 나타낼 수 있다. 예를 들어, 어플리케이션들은 SNS 어플리케이션, 멀티 미디어 어플리케이션, 또는 게임 어플리케이션과 같이, 어플리케이션 스토어에서 제공되는 어플리케이션을 포함할 수 있으며, 리소스들은 어플리케이션의 실행 시 사용될 수 있는 시스템 리소스로써, CPU(central processing unit) 사용량, 메모리 사용량, I/O 사용량, 스레드 생성 개수, 팝업 윈도우 생성 개수, 가상 디스플레이 생성 개수, 알림(notification) 발생 횟수, 펜딩 인텐트(pending intent) 생성 개수를 예로 들 수 있다. 어플리케이션 및 리소스의 예는 상기 예시에 한정되지 않는다.
- [93] 도 6을 참조하면, 특정 어플리케이션(예: App5)의 특정 리소스(예: 리소스 1)에 대해 맵핑되어 저장된 리소스 사용량 데이터(465)는 보다 상세히 도시되어 있다. App5 및 리소스1의 리소스 사용량 데이터는 특정 전자 장치에서 App5의 실행 시 리소스1의 사용량을 획득하여 누적 기록한 것일 수 있다.
- [94] 특정 어플리케이션 및 특정 리소스에 대해 기록된 리소스 사용량 데이터(465)는 SUM, COUNT, SQUARE_SUM, FREQUENCY_TABLE, RAW_DATA를 포함할 수 있다.
- [95] 여기서, x_k 는 어플리케이션이 사용한 리소스의 크기이고, N 은 어플리케이션이 자원을 요청한 횟수 또는 샘플의 개수 일 수 있다. RAW_DATA는 x_k 의 집합일 수 있으며, FREQUENCY_TABLE은 x_k 의 계급(또는 구간) 및 계급별 도수를 저장하는 자료 구조일 수 있다. COUNT, SUM, SQUARE_SUM은 아래의 식 1, 식 2 및 식 3과 같이 계산될 수 있다.

[96] [수식1]

$$COUNT = N$$

[97] [수식2]

$$SUM = \sum_{k=1}^N X_k$$

[98] [수식3]

$$SQUARE_SUM = \sum_{k=1}^N (X_k)^2$$

[99] 식 1과 같이 COUNT는 N과 동일하고, 식 2와 같이 SUM은 해당 어플리케이션이 사용한 리소스의 크기의 합으로 계산될 수 있다. 식 3과 같이 SQUARE_SUM은 해당 어플리케이션이 사용한 리소스의 크기의 제곱의 합으로 계산될 수 있다.

[100] 일 실시예에 따르면, 서버 장치는 특정 어플리케이션 및 특정 리소스에 대해 여러 전자 장치로부터 수신한 데이터를 누적하여 리소스 사용 데이터 세트(460)에 저장할 수 있다. 예를 들어, 서버 장치는 k개의 전자 장치의 리소스 사용량과 관련된 COUNT, SUM, SQUARE_SUM 및 FREQUENCY_TABLE 값을 합산하여 각각 COUNT set, SUM set, SQUARE_SUM set, FREQUENCY_TABLE set로 아래의 식 4 내지 식 7과 같이 계산할 수 있다.

[101] [수식4]

$$COUNT\ set = \sum_{k=1}^N Terminal_k\ 의\ N$$

[102] [수식5]

$$SUM\ set = \sum_{k=1}^N Terminal_k\ 의\ SUM$$

[103] [수식6]

$$SQUARE_SUM\ set = \sum_{k=1}^N Terminal_k\ 의\ SQUARE_SUM$$

[104] [수식7]

$$FREQUENCY_TABLE\ set\ i\ 번째\ 계급\ 도수 = \sum_{k=1}^N Terminal_k\ 의\ FREQUENCY_TABLE\ i\ 번째\ 계급\ 도수$$

[105] 식 5와 같이, SUM set는 k개의 전자 장치에서 실행 어플리케이션이 사용한 리소스의 크기의 합으로 계산될 수 있다. 식 6과 같이, SQUARE_SUM set는 k개의 전자 장치 각각에서 해당 어플리케이션이 사용한 리소스의 크기의 제곱을 합한 값으로 계산될 수 있다. 식 7과 같이, FREQUENCY_TABLE set의 i번째 계급(또는 구간)의 도수는 k개의 전자 장치 각각에서 리소스 사용량이 해당 계급에 속하는 도수를 합한 값으로 계산될 수 있다. RAW_DATA set는 서버 프로세스에서 정규성 테스트를 위한 것으로, 각각의 전자 장치에 설치된 어플리케이션이 사용한 리소스 사용량 정보의 전체 또는 일부를 샘플링 한 자료인 RAW_DATA를 합산한 것일 수 있다.

[106] 표 1은 복수의 전자 장치로부터 App5(예: SNS 어플리케이션) 및 리소스1(예: pending intent)에 대해 수집한 리소스 사용량 정보 세트를 나타낸 것이다.

[107] [표1]

	CO UNT	SU M	SQU ARE_ SUM	RAW_DATA	FREQUENCY_TABLE						
					30 ~	40~	50~	60~	70~	80~	90~
Term inal 1	5	316	22462	33 52 98 56 77	1	0	2	0	1	0	1
Term inal 2	5	332	24762	53 96 32 88 63	1	0	1	1	0	1	1
Term inal 3	5	286	17746	83 38 68 42 55	1	1	1	1	0	1	0
Term inal 4	5	262	15768	61 31 86 51 33	2	0	1	1	0	1	0
Term inal 5	5	282	17152	81 53 34 65 49	1	1	1	1	0	1	0
Term inal 6	4	224	13502	79 58 51 36	1	0	2	0	1	0	0

[108] 서버 장치는 표 1과 같이 리소스 사용량 정보가 수신되는 경우, 각 전자 장치의 리소스 사용량 정보를 아래의 표 2와 같이 합산할 수 있다.

[109] [표2]

	CO UNT set	SU M set	SQU ARE_ SUM set	RAW_DATA set	FREQUENCY_TABLE set						
					30 ~	40~	50~	60~	70~	80~	90~
SERVER	29	17 02	111 392	33 52 98 56 77 53 96 32 88 63 83 38 68 42 55 61 31 86 51 33 81 53 34 65 49 79 58 51 36	7	2	8	4	2	4	2

[110] 서버 장치는 표 2와 같이 합산된 COUNT set, SUM set, SQUARE_SUM set, RAW_DATA set 및 FREQUENCY_TABLE set를 리소스 사용 데이터 세트(460)에 저장할 수 있다.

[111] 일 실시예에 따르면, 각각의 전자 장치는 특정 어플리케이션이 사용하는 특정 리소스에 대해 리소스 사용량 정보(예: RAW_DATA)만 서버 장치로 전송할 수 있으며, 이 경우 서버 장치는 수신한 RAW_DATA에 기반하여 해당 전자 장치의 COUNT, SUM, SQUARE_SUM, FREQUENCY TABLE과 같은 데이터를 계

산할 수 있다. 다른 실시예에 따르면, 각각의 전자 장치는 특정 어플리케이션이 사용하는 특정 리소스에 대해 리소스 사용량 정보를 획득한 후, COUNT, SUM, SQUARE_SUM, FREQUENCY TABLE과 같은 데이터를 계산하여 서버 장치에 전송할 수 있다.

- [112] 도 7은 일 실시예에 따른 서버 장치 및 전자 장치에서 실행되는 프로세스의 블록도이다.
- [113] 일 실시예에 따르면, 서버 프로세스(450)에서 리소스 사용 데이터 세트(460)는 다양한 전자 장치로부터 수신한 리소스 사용량 정보를 누적하여 저장할 수 있다. 서버 장치에서 저장하는 리소스 사용 데이터 세트(460)에 대해서는 도 6을 통해 상세히 설명한 바 있다.
- [114] 일 실시예에 따르면, 서버 프로세스(450)는 메모리(예: 도 4의 메모리(420))에 저장된 리소스 사용 데이터 세트(460)에 기초하여, 미리 정해진 통계적 분석을 통해 특정 어플리케이션 및 특정 어플리케이션이 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정할 수 있다.
- [115] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 리소스 사용 데이터 세트(460)에 저장된 리소스 사용량 정보 중 적어도 일부에 대해, 정규성 테스트(normality test)(472)를 수행하여, 해당 리소스 사용량 정보가 정규성을 갖는지 확인할 수 있다. 이상치 임계 값 계산기(460)는 리소스 사용량 정보가 정규성을 갖는 경우, 표준 편차를 계산하고 각 리소스 사용량 정보의 평균, 표준 편차 및 해당 리소스에 대해 미리 정해진 가중치에 기초하여 이상치 임계 값을 결정할 수 있다(473). 또한, 이상치 임계 값 계산기(460)는 리소스 사용량 정보가 정규성을 갖지 않는 경우, IQR(inter quartile range)을 계산하고, IQR 및 해당 리소스에 대해 미리 정해진 가중치에 기초하여 이상치 임계 값을 결정할 수 있다(474).
- [116] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 전자 장치의 리소스 매니저 데몬(370)에서 전송되는 각 어플리케이션의 각각의 리소스 사용량 통계(371)를 수집하고, 리소스 사용 데이터 세트(460)에 누적해서 저장할 수 있다. 예를 들어, 서버 장치는 정해진 주기에 따라 전자 장치에 리소스 사용량 정보를 전송하도록 요청하거나, 또는 전자 장치는 어플리케이션의 실행에 따른 리소스 사용 이벤트가 발생할 때마다 서버 장치에 리소스 사용량 정보를 전송할 수 있다.
- [117] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 복수의 전자 장치로부터 수집한 특정 어플리케이션이 사용하는 특정 리소스의 통계 정보(예: SUM set, COUNT set, SQUARE_SUM set, FREQUENCY_TABLE set)를 이용하여 통계 값을 구하고, 정규성을 검정(472)할 수 있다. 여기서, 정규성 검증은 Kolmogorov-Smirnov test, Shapiro-Wilk test, 및/또는 왜도, 첨도, 표준 오차 검증에 기초할 수 있으며, 이에 한정되지는 않는다.
- [118] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 데이터가 정규성을 갖는 경우, 표준 편차를 계산할 수 있다. 예를 들어, 표 2의 RAW_DATA에 포함된 29개 수치가 정규성을 갖는 경우, 29개 수치의 평균 및 표준 편차를 계산할 수 있다.

[119] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 계산된 표준 편차(SD) 및 해당 리소스에 대해 미리 정해진 가중치(w_1)에 기초하여 이상치 임계 값(outlier threshold)을 결정할 수 있다. 예를 들어, 리소스 사용량 정보가 정규성을 갖는 경우, 이상치 임계 값은 아래의 식 8과 같이 계산될 수 있다.

[120] [수학식 8]

[121] $AVG = \text{SUM set} / \text{COUNT set}$

[122] $\text{Outlier threshold} = AVG + SD * w_1$

[123] 여기서, AVG는 리소스 사용 데이터 세트에 기록된 각각의 리소스 사용량의 평균으로써, 기록된 SUM set에서 COUNT set를 나눈 값으로 계산될 수 있다. 가중치 w_1 은 각각의 리소스 별로 서로 다른 값으로 결정될 수 있다.

[124] 일 실시예에 따르면, 가중치 임계 값 계산기(460)는 데이터가 정규성을 갖는 경우, IQR(inter quartile range)을 계산할 수 있다. 여기서, IQR은 각 데이터를 그 값에 따라 순차적으로(예: 오름차순으로) 정렬한 후, 하위 25%에 해당하는 데이터와 상위 25%에 해당하는 데이터 사이의 차이로 결정될 수 있다.

[125] 아래의 표 3은 특정 어플리케이션의 특정 리소스 사용량에 대한 FREQUENCY_TABLE set를 도수 분포표로 나타낸 것이다.

[126] [표3]

계급 구간	도수	누적 도수
30 - 40	7	7
40 - 50	2	9
50 - 60	8	17
60 - 70	4	21
70 - 80	2	23
80 - 90	4	27
90 - 100	2	29
계	29	

[127] 상기 표 3의 데이터에서 오름차순으로 데이터를 정렬한 후 하위 25%의 데이터의 순번에 해당하는 Q1 index 및 상위 25%의 데이터의 순번에 해당하는 Q3 index는 다음의 식 9 및 식 10과 같이 계산될 수 있다.

[128] [수학식 9]

[129] $Q1 \text{ index} = (29+1) * 1/4 \approx 7$

[130] [수학식 10]

[131] $Q3 \text{ index} = (29+1) * 3/4 \approx 22$

[132] 또한, 하위 25%의 데이터의 값에 해당하는 Q1 값 및 상위 25%의 데이터의 값에 해당하는 Q3 값은 다음의 식 11과 같이 계산될 수 있다.

- [133] [수학식 11]
- [134] $Q1 \text{ value} = 30 + (40-30) * (Q1 \text{ index} - 0) / 7 = 40$
- [135] $Q3 \text{ value} = 70 + (80-70) * (Q3 \text{ index} - 21) / 2 = 75$
- [136] 상기 표 3의 예에서 Q1 값 및 Q3 값은 29개의 데이터를 오름차순으로 나열한 후 7번째 및 22번째 순서의 데이터를 찾는 것으로 구할 수 있다. 다만, 실제 서버 장치에서 획득하는 리소스 사용 데이터 세트(460)는 수많은 데이터를 포함하기 때문에, 모든 데이터를 오름차순으로 나열하여 각 순번에 맞는 데이터를 찾는 것이 연산 시간 측면에서 비효율적일 수 있다. 이에, 일 실시예에 따르면, 서버 장치는 식 11과 같은 방법으로, Q1 index, Q3 index가 속하는 계급(또는 구간)을 찾고, 해당 계급에서의 Q1 index, Q3 index의 순서를 보간법으로 유추하여, Q1 값 및 Q3 값을 계산할 수 있다.
- [137] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 IQR 및 해당 리소스에 대해 미리 정해진 가중치(w_2)에 기초하여 이상치 임계 값을 계산할 수 있다. 식 12는 IQR에 기초하여 이상치 임계 값을 계산하는 수식이다.
- [138] [수학식 12]
- [139] $\text{Outlier threshold} = Q3 + (Q3-Q1) * w_2$
- [140] 상기 표 3의 예에서 outlier threshold는 아래의 식 13과 같이 계산될 수 있다. 여기서, 가중치 w_2 는 1.5로 대입하였는데 이에 한정되지 않는다.
- [141] [수학식 13]
- [142] $\text{Outlier threshold} = 75 + (75-40) * 1.5 = 127.5$
- [143] 앞에서 설명한 실시예에 따르면, 이상치 임계 값 계산기(460)는 리소스 사용 데이터 세트(460)의 데이터들이 정규성을 갖는 지 여부에 따라 이상치 임계 값을 계산하는 방법을 다르게 결정하였다. 다른 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 정규성 테스트를 하지 않고, 모든 데이터에 대해 IQR에 기초하여 이상치 임계 값을 계산할 수도 있다.
- [144] 일 실시예에 따르면, 이상치 임계 값 계산기(460)는 리소스 사용 데이터 세트(460)에서 각각의 어플리케이션 및 이와 대응되는 각각의 리소스에 대해 이상치 임계 값을 계산할 수 있다. 예를 들어, 도 7에 도시된 바와 같이, A_0 내지 A_i 의 어플리케이션 및 R_0 내지 R_j 의 리소스 사용량이 리소스 사용 데이터 세트(460)에 기록된 경우, 이상치 임계 값 계산기(460)는 어플리케이션 A_i 및 리소스 R_j 에 대응하는 이상치 임계 값 TH_{ij} 를 각각의 어플리케이션 및 리소스에 대해 계산할 수 있다. 이 경우, 일부 어플리케이션은 리소스 중 일부를 사용하지 않을 수 있는데, 이 경우 이상치 임계 값 계산기(460)는 해당 어플리케이션 및 리소스에 대응하는 이상치 임계 값을 제외하거나, 0 이하의 값(예: -1)로 기록할 수 있다.
- [145] 도 8a 내지 8d는 일 실시예에 따른 전자 장치에서 리소스의 과다 사용으로 인한 이상 동작 감지 시 제공하는 UI를 도시한 것이다.

- [146] 일 실시예에 따르면, 전자 장치(300)는 특정 어플리케이션이 실행되는 경우, 실행된 어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 이 중 특정 리소스의 사용량이 해당 어플리케이션 및 리소스에 대해 설정된 이상치 임계 값(outlier threshold) 이상인 경우, 상기 실행된 어플리케이션으로 인해 이상 동작이 발생한 것으로 결정할 수 있다.
- [147] 일 실시예에 따르면, 전자 장치(300)는 이상 동작이 발생한 경우, 리소스를 과다 사용한 어플리케이션의 실행을 종료할 수 있다. 또한, 전자 장치(300)는 이상 동작이 발생한 어플리케이션을 지시하는 정보 및 임계 값을 초과하여 사용된 적어도 하나의 리소스를 지시하는 정보를 포함하는 알림을 출력 장치를 통해 제공할 수 있다. 여기서, 알림은 디스플레이 상에 시각적으로 제공되는 사용자 인터페이스, 햅틱을 이용한 진동 및 스피커를 이용한 오디오 정보 중 적어도 하나를 포함할 수 있다.
- [148] 도 8a 내지 8d는 이상 동작 감지 시 디스플레이를 통해 제공하는 UI를 도시하고 있다.
- [149] 일 실시예에 따르면, 전자 장치(300)는 특정 어플리케이션이 이상치 임계 값 이상으로 리소스를 과다 사용하여 이상 동작이 발생한 것으로 결정한 경우, 이상 동작을 지시하는 UI를 디스플레이 상에 표시할 수 있다. 예를 들어, 전자 장치(300)는 이상 동작을 지시하는 UI를 도 8a에 도시된 것처럼 화면 중앙 영역에 팝업 형태의 UI(810)로 표시하거나, 및/또는 도 8b에 도시된 것처럼 화면 상단에서 제공되는 상태 바(status bar) 상의 UI(820)를 통해 제공할 수 있다. 일 실시예에 따르면, 전자 장치(300)는 이상 동작을 발생시킨 어플리케이션을 강제로 종료하고, 이와 적어도 일부 동시에 도 8a 또는 도 8b와 같은 UI(810, 820)를 디스플레이를 통해 제공할 수 있다.
- [150] 도 8a 또는 도 8b와 같은 UI(810, 820)에서 사용자가 상세 확인 버튼(811, 821)을 선택하는 경우, 전자 장치(300)는 도 8c에 도시된 바와 같은 과다 사용된 리소스를 지시하는 정보를 UI(830)를 통해 제공할 수 있다.
- [151] 도 8c를 참조하면, 전자 장치(300)는 실행된 어플리케이션이 사용하는 리소스(예: pending intent, 팝업 윈도우(popup window), 가상 디스플레이(virtual display))의 정보를 제공하고, 이 중 이상치 임계 값 이상으로 사용된 리소스(예: pending intent)를 사용자가 인식할 수 있도록 제공할 수 있다.
- [152] 일 실시예에 따르면, 전자 장치(300)는 특정 어플리케이션의 실행 시 이상 동작의 발생으로 인해 종료한 횟수가 기준 횟수(예: 3회) 이상인 경우, 사용자에게 피드백을 제공할 수 있다. 여기서, 피드백은 해당 어플리케이션의 삭제를 유도하는 정보를 포함할 수 있다.
- [153] 도 8a 내지 8c와 같은 UI(810, 820, 830)가 표시된 상태에서 사용자가 정해진 버튼을 선택하거나, 및/또는 특정 어플리케이션의 이상 동작의 발생으로 인해 종료한 횟수가 기준 횟수 이상인 경우, 전자 장치(300)는 도 8d에 도시된 바와 같이, 어플리케이션의 삭제 여부를 선택할 수 있는 UI(840)를 디스플레이를 통해 제공할

- 수 있다. UI(840)가 표시된 상태에서 사용자가 확인(841)을 선택하는 경우, 전자 장치(300)는 해당 어플리케이션을 삭제할 수 있다.
- [154] 도 9a 내지 도 9d는 일 실시예에 따른 일 실시예에 따른 전자 장치가 어플리케이션의 이상 동작을 감지하는 방법의 흐름도이다.
- [155] 도 9a 내지 도 9d에 도시된 방법은 전자 장치의 프로세서(예: 도 1의 프로세서(120))에 의해 수행될 수 있으며, 예를 들어, 도시된 동작 중 적어도 일부는 도 5의 리소스 생성기(resource generator)(360) 및/또는 리소스 매니저 데몬(resource manager daemon)(370)에 의해 수행될 수 있다. 도시된 방법은 전자 장치(또는 전자 장치의 프로세서)에 의해 실행될 수 있는 인스트럭션들로 구현될 수 있다. 이러한 인스트럭션들은 컴퓨터로 판독 가능한 기록 매체(computer readable recording medium) 상에 저장될 수 있다.
- [156] 도 9a는 전자 장치가 리소스 매니저 데몬을 초기화 한 후 이상치 임계 값을 획득하여 업데이트 하는 방법을 도시하고 있다.
- [157] 일 실시예에 따르면, 동작 902에서, 전자 장치는 리소스 매니저 데몬을 초기화할 수 있다. 리소스 매니저 데몬은 어플리케이션의 실행 시 사용되는 적어도 하나의 리소스의 사용량 정보를 기록하고, 각 어플리케이션 및 각 리소스에 대응하는 이상치 임계 값을 관리할 수 있다.
- [158] 일 실시예에 따르면, 동작 904에서, 전자 장치는 이상치 임계 값을 0보다 작은 값(예: -1)으로 설정할 수 있다. 여기서, 이상치 임계 값을 0보다 작은 값으로 초기화 하는 이유는 특정 어플리케이션이 특정 리소스를 사용한 정보에 대한 통계적인 이상치를 구하기 이전의 상황이므로, 이상치 임계 값을 음수로 설정하여 어플리케이션의 실행 시 이상 동작으로 판단하지 않도록 하기 위함일 수 있다. 일 실시예에 따르면, 전자 장치는 특정 어플리케이션이 실행되어 리소스를 사용 시, 해당 리소스에 대해 이상치 임계 값이 0보다 작은 값으로 설정되어 있는 상황인 경우, 해당 리소스의 사용량과 무관하게 이상 동작이 아닌 것으로 판단할 수 있다. 다른 일 실시예에 따르면, 전자 장치는 리소스 매니저 데몬의 초기화 과정에서 이상치 임계 값을 0보다 작은 값이 아닌 통계적으로 무의미한 정도의 큰 값으로 설정할 수도 있다.
- [159] 일 실시예에 따르면, 동작 906에서, 전자 장치는 서버 장치로부터 이상치 임계 값이 수신되는 지 확인할 수 있다. 도 3 내지 도 7을 통해 설명한 바와 같이, 서버 장치는 다양한 전자 장치로부터 획득한 리소스 사용량 정보에 기초하여, 통계적 분석을 통해 각각의 어플리케이션 및 각각의 리소스에 대응하는 이상치 임계 값을 계산할 수 있다. 서버 장치는 리소스 사용 데이터 세트에 충분한 데이터가 쌓이는 경우 이상치 임계 값을 계산할 수 있으며, 계산된 이상치 임계 값을 각 전자 장치에 제공할 수 있다.
- [160] 일 실시예에 따르면, 서버 장치로부터 이상치 임계 값이 수신되는 경우, 동작 908에서, 전자 장치는 서버 장치로부터 전달받은 이상치 임계 값으로 기 저장된 이상치 임계 값을 업데이트 할 수 있다. 예를 들어, 리소스 매니저 데몬의 초기화

과정에서 0보다 작은 값(예: -1)으로 설정되어 있던 이상치 임계 값을 서버 장치로부터 획득한 이상치 임계 값으로 업데이트 할 수 있다. 이후, 전자 장치는 어플리케이션이 실행되어 적어도 하나의 리소스를 사용하는 경우, 각각의 리소스 사용량이 각각의 업데이트 된 이상치 임계 값 이상인지 여부를 확인할 수 있다.

- [161] 도 9b는 전자 장치가 리소스 사용량 정보를 서버 장치에 전송하고, 기 저장된 데이터를 초기화 하는 방법에 대해 도시하고 있다.
- [162] 일 실시예에 따르면, 동작 922에서, 전자 장치는 서버 장치로부터 데이터 전송 요청을 수신할 수 있다.
- [163] 일 실시예에 따르면, 동작 924에서, 전자 장치는 메모리에 기록되어 있는 리소스 사용량 정보를 서버 장치로 전송할 수 있다. 예를 들어, 전자 장치는 서버 장치의 데이터 전송 요청에 응답하여, 메모리에 기록되어 있는 각 어플리케이션의 각 리소스 사용량과 관련된 COUNT, SUM, SQUARE_SUM, FREQUENCY_TABLE, 및 RAW_DATA를 서버 장치에 전송할 수 있다. 리소스 사용량 정보에 포함되는 COUNT, SUM, SQUARE_SUM, FREQUENCY_TABLE, 및 RAW_DATA에 대해서는 앞서 도 6을 통해 설명한 바 있다. 일 실시예에 따르면, 전자 장치는 RAW_DATA만 메모리에 기록한 후 서버 장치에 전송할 수 있으며, 이 경우 서버 장치는 수신한 RAW_DATA에 기반하여 해당 전자 장치의 COUNT, SUM, SQUARE_SUM, FREQUENCY TABLE과 같은 데이터를 계산할 수 있다.
- [164] 일 실시예에 따르면, 동작 926에서, 전자 장치는 메모리에 기록되어 있던 COUNT, SUM, SQUARE_SUM의 값을 0으로 초기화 할 수 있다. 또한, 동작 928에서, 전자 장치는 FREQUENCY_TABLE 및 RAW_DATA를 초기화 할 수 있다. 전자 장치는 리소스 사용량 정보를 서버 장치에 전송한 후, 기록되어 있던 데이터를 초기화 할 수 있다.
- [165] 전자 장치는 리소스 사용량 정보를 초기화 한 후, 이후 어플리케이션의 실행에 따른 리소스 사용 이벤트가 발생하면, 사용된 리소스의 사용량 정보를 메모리에 기록할 수 있다.
- [166] 도 9c는 전자 장치가 리소스 사용량 정보를 메모리에 기록하는 방법에 대해 도시하고 있다.
- [167] 일 실시예에 따르면, 동작 942에서, 전자 장치는 특정 어플리케이션의 특정 리소스 사용에 대한 이벤트를 수신할 수 있다. 예를 들어, 어플리케이션이 실행되는 경우, 적어도 하나의 리소스(예: CPU 사용, 메모리 사용, pending intent 생성)를 사용할 수 있으며, 리소스 생성기(예: 도 5의 리소스 생성기(360))는 어플리케이션의 각 리소스 사용에 대한 이벤트를 리소스 매니저 데몬(예: 도 5의 리소스 매니저 데몬(370))으로 전달할 수 있다. 전자 장치는 특정 어플리케이션의 특정 리소스 사용량을 변수 R에 저장할 수 있다.
- [168] 일 실시예에 따르면, 동작 944에서, 전자 장치는 리소스 사용량이 현재의 이상치 임계 값 미만인지, 또는 이상치 임계 값이 0보다 작은 값인지 여부를 확인할 수 있다. 전자 장치는 리소스 매니저 데몬의 초기화 과정에서 이상치 임계 값을 0

보다 작은 값(예: -1)으로 설정할 수 있으며, 이상치 임계 값이 0보다 작은 값인 경우는 서버 장치로부터 획득된 이상치 임계 값이 없는 경우일 수 있다.

[169] 일 실시예에 따르면, 이상치 임계 값이 0보다 큰 값으로 설정되어 있고, 어플리케이션의 실행에 따라 사용된 리소스의 사용량이 현재 설정된 이상치 임계 값 이상인 경우, 동작 946에서, 전자 장치는 어플리케이션의 실행에 따라 리소스를 과다 사용하여 이상 동작이 발생한 것으로 결정할 수 있다. 이 경우, 전자 장치는 이상 동작의 발생에 대응하여, 어플리케이션을 종료하거나, 및/또는 이상 동작을 알리기 위한 알림을 출력 장치(예: 디스플레이, 햅틱, 스피커)를 통해 출력할 수 있다. 또한, 사용된 리소스의 사용량이 현재 설정된 이상치 임계 값 이상인 경우, 전자 장치는 동작 948 이하의 리소스 사용량 정보를 기록하는 동작을 수행하지 않을 수 있다.

[170] 일 실시예에 따르면, 현재 설정된 이상치 임계 값이 0보다 작은 수이거나, 어플리케이션의 실행에 따라 사용된 리소스 사용량이 이상치 임계 값 미만인 경우, 동작 948 이하의 리소스 사용량 정보를 기록하는 동작을 수행할 수 있다. 전자 장치는 동작 948에서, 특정 어플리케이션 및 특정 리소스에 대해 현재 메모리에 저장된 값인 COUNT에서 1을 더하고, SUM에 R을 더하고, SQUARE_SUM에 R^2 을 더할 수 있다. 여기서, R은 동작 942에서 실행된 어플리케이션의 사용량이므로, 동작 948에 따라 해당 어플리케이션의 리소스 사용량 정보가 메모리에 기록될 수 있다.

[171] 일 실시예에 따르면, 동작 950에서, 전자 장치는 현재 데이터의 저장이 요구되는지 확인할 수 있다. 예를 들어, 전자 장치는 동작 942에서 실행된 어플리케이션의 특정 리소스 사용량에 관련된 정보가 이미 메모리에 충분한 개수만큼 저장되어 있는 경우, 추가 데이터의 저장이 불필요한 것으로 판단할 수 있다. 또는, 전자 장치는 해당 어플리케이션 및/또는 해당 리소스에 대응하는 이상치 임계 값이 서버 장치로부터 제공되는지 여부에 기초하여 현재 데이터의 저장이 요구되는지 확인할 수 있다.

[172] 일 실시예에 따르면, 현재 데이터의 저장이 요구되는 것으로 판단되는 경우, 동작 952에서, 전자 장치는 RAW_DATA에 현재 데이터를 저장할 수 있다.

[173] 일 실시예에 따르면, 동작 954에서, 전자 장치는 FREQUENCY_TABLE에 현재의 리소스 사용량(또는 R 값)을 포함하는 계급의 도수를 1 증가시킬 수 있다. 예를 들어, 리소스 사용량의 수치 범위를 복수의 계급으로 분할할 수 있으며, 전자 장치는 복수의 계급 중 동작 942에서 획득한 리소스 사용량이 속하는 계급의 도수를 1 증가시킬 수 있다. 일 실시예에 따르면, 각 리소스의 사용량과 관련된 복수의 계급의 크기는 리소스의 종류 별로 임의로 결정될 수 있으며, 또는 서버 장치에 의해 지정될 수 있다.

[174] 일 실시예에 따르면, 전자 장치는 클래스(class)를 R 값을 포함하는 계급의 string 값으로 정의할 수 있다. FREQUENCY_TABLE에 해당 클래스의 값이 존재하지 않는 경우, 전자 장치는 해당 클래스의 도수를 0으로 설정할 수 있다. 전자

장치는 FREQUENCY_TABLE에 해당 클래스의 값이 존재하는 경우, 해당 클래스의 도수를 1 증가시킬 수 있다.

- [175] 도 9d는 전자 장치가 특정 어플리케이션의 실행에 따라, 이상 동작 발생 시의 동작을 도시하고 있다.
- [176] 일 실시예에 따르면, 동작 962에서, 전자 장치는 전자 장치는 특정 어플리케이션의 특정 리소스 사용에 대한 이벤트를 수신할 수 있다.
- [177] 일 실시예에 따르면, 동작 964에서, 전자 장치는 리소스 사용량이 현재의 이상치 임계 값 이상이고, 및 이상치 임계 값이 0 이상인지 여부를 확인할 수 있다. 이상치 임계 값이 0 이상인 경우는 서버 장치로부터 획득된 이상치 임계 값으로 설정된 것을 의미할 수 있다.
- [178] 일 실시예에 따르면, 동작 962의 이벤트의 리소스 사용량이 이상치 임계 값 미만이거나, 또는 현재 이상치 임계 값이 0보다 작은 수로 설정된 경우, 동작 966에서, 전자 장치는 이상 동작이 아닌 것으로 결정하고, 리소스 사용량을 메모리에 기록할 수 있다. 예를 들어, 전자 장치는 도 9c의 동작 948 이하의 리소스 사용량 정보를 기록하는 동작을 수행할 수 있다.
- [179] 일 실시예에 따르면, 동작 962의 이벤트의 리소스 사용량이 이상치 임계 값 이상인 경우, 전자 장치는 리소스의 과다 사용으로 인한 이상 동작으로 결정할 수 있다. 전자 장치는 동작 968에서, 이상 동작을 발생시킨 어플리케이션을 강제 종료할 수 있다.
- [180] 일 실시예에 따르면, 동작 970에서, 전자 장치는 해당 어플리케이션의 강제 종료 횟수가 기준 횟수 이상인지 확인할 수 있다. 여기서, 기준 횟수는 3회일 수 있으나, 이에 한정되지는 않는다.
- [181] 일 실시예에 따르면, 강제 종료 횟수가 기준 횟수 미만인 경우, 동작 974에서, 전자 장치는 어플리케이션의 이상 동작을 지시하는 UI를 디스플레이를 통해 제공할 수 있다. 예를 들어, 전자 장치는 이상 동작을 지시하는 UI를 도 8a와 같이 팝업 형태로 제공하거나, 및/또는 도 8b와 같이 상태 바(status bar) 상에서 제공할 수 있다. 또는, 전자 장치는 디스플레이가 아닌 다른 종류의 출력 장치(예: 햅틱, 스피커)를 통해 어플리케이션의 이상 동작을 지시하는 알림을 제공할 수 있다.
- [182] 일 실시예에 따르면, 강제 종료 횟수가 기준 횟수 이상인 경우, 동작 972에서, 전자 장치는 사용자에게 어플리케이션의 삭제를 유도하는 UI를 제공할 수 있다. 예를 들어, 전자 장치는 도 9d의 UI를 디스플레이를 통해 제공할 수 있다. 일 실시예에 따르면, 전자 장치는 동작 974에서 제공되는 UI 상에서 사용자가 정해진 버튼을 선택하는 경우, 동작 972의 어플리케이션의 삭제를 유도하는 UI를 제공할 수 있다.
- [183] 도 10은 일 실시예에 따른 서버 장치의 이상치 임계 값을 결정하는 방법의 흐름도이다.
- [184] 도 10에 도시된 방법은 서버 장치의 프로세서(예: 도 4의 프로세서(410))에 의해 수행될 수 있으며, 앞서 설명한 바 있는 기술적 특징에 대해서는 이하에서 그 설

명을 생략할 수도 있다. 도시된 방법은 서버 장치(예: 서버 장치(400)의 프로세서(410))에 의해 실행될 수 있는 인스트럭션들로 구현될 수 있다. 이러한 인스트럭션들은 컴퓨터로 판독 가능한 기록 매체(computer readable recording medium) 상에 저장될 수 있다.

- [185] 일 실시예에 따르면, 동작 1010에서, 서버 장치는 전자 장치로부터 특정 어플리케이션의 실행 시 사용되는 적어도 하나의 리소스의 사용량을 포함하는 리소스 사용량 정보를 수신할 수 있다. 예를 들어, 서버 장치는 특정 어플리케이션 및 특정 리소스에 대응하는 COUNT, SUM, SQUARE_SUM, FREQUENCY_TABLE, RAW_DATA를 수신할 수 있다. 여기서, COUNT는 어플리케이션이 자원을 요청한 횟수 또는 샘플의 개수이고, RAW_DATA는 획득된 리소스 사용량의 집합이고, SUM은 해당 어플리케이션이 사용한 리소스의 크기의 합이고, SQUARE_SUM은 해당 어플리케이션이 사용한 리소스의 크기의 제곱의 합일 수 있다. 일 실시예에 따르면, 서버 장치는 전자 장치로부터 RAW_DATA만 수신할 수 있으며, 이 경우 서버 장치가 수신한 RAW_DATA에 기초하여, COUNT, SUM, SQUARE_SUM, FREQUENCY_TABLE과 같은 다른 데이터를 계산할 수 있다.
- [186] 일 실시예에 따르면, 동작 1020에서, 서버 장치는 메모리에 저장된 리소스 사용 데이터 세트에 전자 장치로부터 수신한 리소스 사용량 정보를 추가할 수 있다. 예를 들어, 서버 장치는 리소스 사용 데이터 세트의 COUNT set, SUM set 및 SQUARE_SUM set에 각각 전자 장치로부터 수신한 COUNT, SUM 및 SQUARE_SUM을 더할 수 있다. 또한, 서버 장치는 RAW_DATA set에 전자 장치로부터 수신한 RAW_DATA를 추가할 수 있다. 서버 장치에서 누적해서 저장하는 리소스 사용 데이터 세트에 대해서는 도 6을 통해 상세히 설명한 바 있다.
- [187] 일 실시예에 따르면, 서버 장치는 전자 장치로부터 수신한 FREQUENCY_TABLE에 기초하여 FREQUENCY_TABLE set를 업데이트 할 수 있다. 예를 들어, 서버 장치는 클래스를 FREQUENCY_TABLE의 최초 계급의 스트링 값으로 결정하고, 수신한 FREQUENCY_TABLE에 해당 클래스가 존재하는 경우, FREQUENCY_TABLE set에서 해당 클래스의 도수 값에 수신한 FREQUENCY_TABLE에서 해당 클래스의 도수 값을 더할 수 있다. 이 후, 서버 장치는 클래스를 다음 계급의 스트링 값으로 결정하고, 상기와 같이 FREQUENCY_TABLE set에서 해당 클래스의 도수 값에 수신한 FREQUENCY_TABLE에서 해당 클래스의 도수 값을 더하는 동작을 더 추가할 클래스가 없을 때까지 반복 수행할 수 있다.
- [188] 일 실시예에 따르면, 동작 1030에서, 서버 장치는 리소스 사용 데이터 세트의 특정 어플리케이션 및 리소스 사용량에 대해 정규성 검증(normality test)을 수행할 수 있다. 여기서, 정규성 검증은 Kolmogorov-Smirnov test, Shapiro-Wilk test, 및/또는 왜도, 첨도, 표준 오차 검증에 기초할 수 있으며, 이에 한정되지는 않는다. 동작 1032에서, 서버 장치는 정규성 검증 결과 데이터가 정규 분포를 따르는지 확

인할 수 있다. 데이터가 정규 분포를 따르는 경우, 동작 1052 및 1054를 수행하고, 정규 분포를 따르지 않는 경우, 동작 1042, 1044 및 1046을 수행할 수 있다.

[189] 일 실시예에 따르면, 데이터가 정규분포를 따르는 경우, 동작 1052에서, 서버 장치는 해당 어플리케이션 및 리소스의 리소스 사용 데이터 세트에 포함된 각 데이터의 표준 편차를 계산할 수 있다. 예를 들어, 표준 편차는 식 14와 같이 계산될 수 있다.

[190] [수식14]

$$SD = \sqrt{\frac{SQUARE_SUM\ set}{COUNT\ set}}$$

[191] 일 실시예에 따르면, 동작 1054에서, 서버 장치는 계산된 표준 편차 및 해당 리소스에 대해 미리 정해진 가중치(w_1)에 기초하여, 이상치 임계 값을 계산할 수 있다. 예를 들어, 이상치 임계 값은 식 15와 같이 계산될 수 있다.

[192] [수학식 15]

[193] $AVG = SUM\ set / COUNT\ set$

[194] $Outlier\ threshold = AVG + SD * w_1$

[195] 여기서, AVG는 리소스 사용 데이터 세트에 기록된 각각의 리소스 사용량의 평균으로써, 기록된 SUM set에서 COUNT set를 나눈 값으로 계산될 수 있다. 가중치 w_1 은 각각의 리소스 별로 서로 다른 값으로 결정될 수 있다.

[196] 일 실시예에 따르면, 데이터가 정규분포를 따르지 않는 경우, 동작 1042에서, 서버 장치는 Q1 index 및 Q3 index를 계산할 수 있다. 여기서, Q1 index는 데이터를 오름차순으로 정렬한 후 하위 25%의 데이터의 순번에 해당하고, Q3 index는 상위 25%의 데이터의 순번에 해당할 수 있다. Q1 index 및 Q3 index는 각각 식 16 및 식 17에 따라 계산될 수 있다.

[197] [수학식 16]

[198] $Q1\ index = int(COUNT\ set + 1) * 1/4$

[199] [수학식 17]

[200] $Q3\ index = int(COUNT\ set + 1) * 3/4$

[201] 일 실시예에 따르면, 동작 1044에서, 서버 장치는 IQR(inter quartile range)를 계산할 수 있다. 여기서, IQR은 Q3 값과 Q1 값의 차이로 결정될 수 있다. 서버 장치는 Q3 index의 Q3 값 및 Q1 index의 Q1 값을 확인하고, 그 차이를 통해 IQR을 계산할 수 있다. 일 실시예에 따르면, 서버 장치는 Q1 index, Q3 index가 속하는 계급(또는 구간)을 찾고, 해당 계급에서의 Q1 index, Q3 index의 순서를 보간법으로 유추하여, Q1 값 및 Q3 값을 계산할 수 있다.

[202] 일 실시예에 따르면, 동작 1046에서, 서버 장치는 계산된 IQR 및 미리 정해진 가중치(예: w_2)에 기초하여 이상치 임계 값을 계산할 수 있다. 예를 들어, 이상치 임계 값은 식 18과 같이 계산될 수 있다.

[203] [수학식 18]

- [204] $\text{Outlier threshold} = Q3 + (Q3 - Q1) * w_2$
- [205] 가중치 w_2 는 각 리소스에 따라 달라질 수 있다.
- [206] 일 실시예에 따르면, 동작 1060에서, 서버 장치는 결정된 이상치 임계 값을 각 전자 장치로 전송할 수 있다. 전자 장치는 수신한 이상치 임계 값을 저장하고, 어플리케이션의 실행 시 사용된 각 리소스의 사용량을 각 이상치 임계 값과 비교하여, 이상 동작 여부를 결정할 수 있다.
- [207] 도 11은 일 실시예에 따른 전자 장치의 블록도이다.
- [208] 도 11을 참조하면, 일 실시예에 따른 전자 장치(1100)는 프로세서(1110), 메모리(1120), 통신 모듈(1130) 및 적어도 하나의 출력 장치(1140)를 포함할 수 있다. 도시된 구성 중 일부가 생략 또는 다른 구성으로 치환되더라도 본 문서의 다양한 실시예를 구현할 수 있다. 전자 장치(1100)는 도시된 구성 이외에도 도 1의 전자 장치(101)의 구성 및/또는 기능 중 적어도 일부를 더 포함할 수 있다. 도시된(또는 도시되지 않은) 전자 장치(1100)의 각 구성 중 적어도 일부는 상호 작동적으로 (operatively), 기능적으로(functionally) 및/또는 전기적으로 (electrically) 연결될 수 있다.
- [209] 일 실시예에 따르면, 전자 장치(1100)는 사용자에게 시각적, 청각적, 또는 촉각적인 알림(또는 피드백)을 제공하기 위한 적어도 하나의 출력 장치(1140)를 포함할 수 있다. 예를 들어, 적어도 하나의 출력 장치(1140)는 디스플레이(1150), 스피커(1160), 및 햅틱 모듈(1170)을 포함할 수 있다.
- [210] 일 실시예에 따르면, 디스플레이(1150)는 액정 디스플레이(liquid crystal display(LCD)), 발광 다이오드(light-emitting diode(LED)) 디스플레이, 유기 발광 다이오드(organic light-emitting diode(OLED)) 디스플레이 중 어느 하나로 구현될 수 있으며, 이에 한정되지는 않는다. 디스플레이(1150)는 사용자의 신체 일부(예: 손가락) 또는 입력 장치(예: 스타일러스 펜)를 이용한 터치 및/또는 근접 터치(또는 호버링) 입력을 감지하는 터치 스크린으로 구성될 수 있다. 디스플레이(1150)는 도 1의 디스플레이 모듈(160)의 구성 및/또는 기능 중 적어도 일부를 포함할 수 있다. 디스플레이(1150)는 적어도 일부가 플렉서블(flexible) 할 수 있으며, 폴더블(foldable) 디스플레이, 롤러블(rollable) 디스플레이로 구현될 수도 있다.
- [211] 일 실시예에 따르면, 햅틱 모듈(1170)은 모터와 같은 기계적인 구조를 통해 사용자가 인지할 수 있는 진동을 발생시킬 수 있다. 햅틱 모듈(1170)은 도 1의 햅틱 모듈(179)의 구성 및/또는 기능 중 적어도 일부를 포함할 수 있다.
- [212] 일 실시예에 따르면, 스피커(1160)는 오디오 신호를 외부로 출력할 수 있다. 스피커(1160)는 도 1의 음향 출력 모듈(155)의 구성 및/또는 기능 중 적어도 일부를 포함할 수 있다.
- [213] 일 실시예에 따르면, 전자 장치(1100)는 디스플레이(1150), 햅틱 모듈(1170), 스피커(1160) 이외에도 사용자가 다양한 감각을 통해 인지 가능한 알림을 제공할 수 있는 다른 종류의 출력 장치(1140)를 더 포함할 수 있다.

- [214] 일 실시예에 따르면, 통신 모듈(1130)은 외부 장치(예 도 3의 서버 장치(400))와의 무선 통신을 지원하기 위한 다양한 하드웨어 및/또는 소프트웨어 구성을 포함할 수 있다. 통신 모듈(1130)은 근거리 무선 통신(예: Wi-Fi, Bluetooth) 및 셀룰러 무선 통신(예: 4G LTE, 5G NR)을 지원할 수 있다. 통신 모듈(1130)은 도 1의 통신 모듈(190)의 구성 및/또는 기능 중 적어도 일부를 포함할 수 있다.
- [215] 일 실시예에 따르면, 메모리(1120)는 휘발성 메모리 및 비휘발성 메모리를 포함하여, 다양한 데이터들을 일시적 또는 영구적으로 저장할 수 있다. 메모리(1120)는 도 1의 메모리(130)의 구성 및/또는 기능 중 적어도 일부를 포함하고, 도 1의 프로그램(140) 및/또는 도 2의 프로그램(140) 중 적어도 일부를 저장할 수 있다.
- [216] 일 실시예에 따르면, 메모리(1120)는 프로세서(1110)에서 수행될 수 있는 다양한 인스트럭션(instruction)들을 저장할 수 있다. 이와 같은 인스트럭션들은 프로세서(1110)에 의해 인식될 수 있는 산술 및 논리 연산, 데이터 이동, 입출력과 같은 제어 명령을 포함할 수 있다.
- [217] 일 실시예에 따르면, 프로세서(1110)는 전자 장치(1100)의 각 구성 요소들의 제어 및/또는 통신에 관한 연산이나 데이터 처리를 수행할 수 있는 구성으로써, 하나 이상의 프로세서(1110)들로 구성될 수 있다. 프로세서(1110)는 메모리(1120), 출력 장치(1140), 및 통신 모듈(1130)을 포함하는 전자 장치(1100)의 각 구성 요소와 전기적으로, 기능적으로, 및/또는 작동적으로 연결될 수 있다. 프로세서(1110)는 도 1의 프로세서(120)의 구성 및/또는 기능 중 적어도 일부를 포함할 수 있다.
- [218] 일 실시예에 따르면, 전자 장치(1100)는 dual, quad, octa 코어(core)와 같은 멀티-프로세서 컴퓨터 아키텍처를 포함할 수 있다. 예를 들어, 전자 장치(1100)는 연산 작업과 같은 핵심적인 동작을 수행하는 복수의 코어를 하나의 집적회로 상에서 통합하여 프로세서(1110)를 구성할 수 있다. 일 실시예에 따르면, 프로세서(1110)는 싱글 코어 프로세서로 구현될 수도 있다.
- [219] 일 실시예에 따르면, 프로세서(1110)가 전자 장치(1100) 상에서 구현할 수 있는 연산 및 데이터 처리 기능에는 한정됨이 없을 것이나, 이하에서는 어플리케이션의 실행 시 리소스 사용 이벤트를 확인하여 리소스 사용량 정보를 메모리(1120)에 기록하고, 메모리(1120)에 저장된 리소스 사용 데이터 세트에 기초하여 각 리소스에 대응하는 이상치 임계 값을 계산하고, 및 특정 리소스 사용량이 이상치 임계 값 이상인 경우 이상 동작으로 결정하여 이상 동작의 발생에 대응하는 피드백을 제공하는 다양한 실시예들에 대해 설명하기로 한다. 후술할 전자 장치(1100) 또는 프로세서(1110)의 동작들은 메모리(1120)에 저장된 인스트럭션들을 로딩(loading)함으로써 수행될 수 있다.
- [220] 일 실시예에 따르면, 전자 장치(1100)는 다양한 어플리케이션을 설치 및 실행할 수 있다. 예를 들어, 전자 장치(1100)는 도 2의 어플리케이션(146) 중 적어도 일부를 실행할 수 있으며, 또한 어플리케이션 스토어(예: 플레이 스토어™)를 통해 다운로드 받아 설치한 적어도 하나의 어플리케이션을 실행할 수 있다. 전자 장치

(1100)는 포 그라운드 및 백 그라운드에서 복수의 어플리케이션을 동시에 실행할 수 있다.

- [221] 일 실시예에 따르면, 프로세서(1110)가 메모리(1120)에 저장된 특정 어플리케이션을 실행하는 경우, 실행된 어플리케이션은 전자 장치(1100)의 리소스를 사용할 수 있다. 여기서, 어플리케이션이 사용하는 전자 장치(1100)의 리소스는 CPU(central processing unit), 메모리, I/O, 스레드, 팝업 윈도우, 가상 디스플레이, 알림(notification), 펜딩 인텐트(pending intent)를 예로 들 수 있으며, 이에 한정되지는 않는다. 실행되는 어플리케이션에 따라 사용하는 리소스의 종류는 다를 수 있으며, 동일한 어플리케이션의 경우에도 동작 상태 및/또는 환경 설정에 따라 사용하는 리소스의 종류가 다를 수 있다.
- [222] 일 실시예에 따르면, 메모리(1120)는 복수의 어플리케이션 및 복수의 어플리케이션 각각에 대응하는 복수의 리소스 사용량을 맵핑하여 기록하는 리소스 사용 데이터 세트를 저장할 수 있다. 리소스 사용 데이터 세트는 각각의 어플리케이션과 각각의 리소스의 사용량 정보를 맵핑해서 저장할 수 있다. 또한, 리소스 사용 데이터 세트는 COUNT, SUM, SQUARE_SUM, FREQUENCY_TABLE과 같은 리소스 사용량과 관련된 데이터를 저장할 수 있다. 예를 들어, 리소스 사용 데이터 세트는 도 6에서 설명한 서버 장치 상에 저장되는 리소스 사용 데이터 세트와 자료 구조가 동일할 수 있다.
- [223] 일 실시예에 따르면, 프로세서(1110)는 특정 어플리케이션이 실행되는 경우, 실행된 어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인할 수 있다. 예를 들어, 사용자의 입력에 따라 특정 어플리케이션이 실행되는 경우, 실행된 어플리케이션은 적어도 하나의 리소스(예: CPU 사용, 메모리 사용, pending intent 생성)를 사용할 수 있으며, 프로세서(1110)는 리소스 사용에 관련된 이벤트를 확인할 수 있다.
- [224] 일 실시예에 따르면, 프로세서(1110)는 어플리케이션의 실행에 따라 사용된 적어도 하나의 리소스의 사용량 정보를 메모리(1120)에 저장된 리소스 사용 데이터 세트에 기록할 수 있다. 예를 들어, 리소스 사용 데이터 세트는 각각의 어플리케이션의 실행 시 사용되는 각각의 리소스의 사용량 정보를 누적해서 기록할 수 있으며, 프로세서(1110)는 상기 어플리케이션의 실행에 따라 사용된 적어도 하나의 리소스의 사용량 정보를 리소스 사용 데이터 세트에서 대응되는 어플리케이션 및 리소스의 데이터에 추가할 수 있다. 일 실시예에 따르면, 프로세서(1110)는 실행된 어플리케이션이 사용하는 적어도 하나의 리소스 중 그 사용량이 이상치 임계 값 이상인 것이 있는 경우, 이상 동작으로 판단할 수 있으며, 이 경우 리소스 사용 데이터 세트에 기록하는 동작은 수행하지 않을 수 있다.
- [225] 일 실시예에 따르면, 프로세서(1110)는 메모리(1120)에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 특정 어플리케이션 및 특정 어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값(예: 제1이상치 임계 값)을 결정할 수 있다. 예를

들어, 프로세서(1110)가 이상치 임계 값을 결정하는 방법은 앞서 도 4 내지 도 10을 통해 설명한 서버 장치가 이상치 임계 값을 결정하는 동작에 방법에 대응될 수 있다.

- [226] 일 실시예에 따르면, 프로세서(1110)는 리소스 사용 데이터 세트의 특징 어플리케이션 및 리소스 사용량에 대해 정규성 검증(normality test)을 수행할 수 있다. 여기서, 정규성 검증은 Kolmogorov-Smirnov test, Shapiro-Wilk test, 및/또는 왜도, 첨도, 표준 오차 검증에 기초할 수 있으며, 이에 한정되지는 않는다.
- [227] 일 실시예에 따르면, 프로세서(1110)는 데이터가 정규분포를 따르는 경우, 표준 편차 및 미리 정해진 가중치(w_1)에 기초하여 이상치 임계 값을 결정할 수 있다. 예를 들어, 프로세서(1110)는 정규 분포를 따르는 데이터들의 평균 및 표준 편차를 계산할 수 있다. 프로세서(1110)는 표준 편차와 가중치의 곱으로 이상치 임계 값을 결정할 수 있다. 여기서, 가중치 w_1 은 각 리소스 별로 서로 다른 값으로 결정될 수 있다.
- [228] 일 실시예에 따르면, 프로세서(1110)는 데이터가 정규분포를 따르지 않는 경우, IQR 및 해당 리소스에 대해 미리 정해진 가중치(w_2)에 기초하여 이상치 임계 값을 결정할 수 있다. 여기서, IQR은 각 데이터를 그 값에 따라 순차적으로(예: 오름차순으로) 정렬한 후, 하위 25%에 해당하는 데이터와 상위 25%에 해당하는 데이터 사이의 차이로 결정될 수 있다. 프로세서(1110)는 Q3 값과 Q1 값을 구한 후 그 차이로 IQR을 구할 수 있다. 프로세서(1110)는 IQR과 가중치(예: w_2)를 곱한 값을 Q3 값에 더한 값을 이상치 임계 값으로 결정할 수 있다. 여기서, 가중치 w_2 는 각 리소스 별로 서로 다른 값으로 결정될 수 있다.
- [229] 일 실시예에 따르면, 프로세서(1110)는 데이터에 대한 정규성 검증 없이, 모든 데이터에 대해 IQR 및 가중치 w_2 에 기초하여 이상치 임계 값을 계산할 수도 있다.
- [230] 일 실시예에 따르면, 프로세서(1110)는 이상치 임계 값(예: 제1이상치 임계 값)이 계산되는 경우, 해당 어플리케이션 및 리소스에 대해 메모리(1120)에 현재 저장된 이상치 임계 값(예: 제2이상치 임계 값)을 계산된 이상치 임계 값으로 업데이트 할 수 있다. 이상치 임계 값은 초기화 상태에서 0보다 작은 값으로 설정될 수 있으며, 이상치 임계 값이 계산되는 경우 그 값으로 업데이트 될 수 있다. 프로세서(1110)는 리소스 사용 데이터 세트에 충분한 양의 데이터가 수집되는 경우에, 이상치 임계 값을 계산하는 동작을 수행할 수 있다.
- [231] 일 실시예에 따르면, 프로세서(1110)는 특정 어플리케이션이 실행되어 적어도 하나의 리소스를 사용하는 경우, 리소스의 사용량이 해당 리소스에 대응하여 설정된 이상치 임계 값을 초과하는지 여부를 확인할 수 있다. 일 실시예에 따르면, 프로세서(1110)는 리소스 사용량이 이상치 임계 값을 초과하는 경우, 해당 어플리케이션의 실행에 따라 이상 동작이 발생한 것으로 결정할 수 있다.

- [232] 일 실시예에 따르면, 프로세서(1110)는 이상 동작을 발생시킨 어플리케이션을 강제로 종료할 수 있다. 또한, 프로세서(1110)는 도 8a 내지 도 8d에 도시된 것과 같이, 어플리케이션의 이상 동작을 지시하는 사용자 인터페이스를 디스플레이(1150)를 통해 제공할 수 있다. 또는, 프로세서(1110)는 햅틱 모듈(1170) 또는 스피커(1160)와 같은 다른 출력 장치(1140)를 통해 어플리케이션의 이상 동작을 지시하는 알림을 제공할 수 있다.
- [233] 일 실시예에 따르면, 프로세서(1110)는 이상 동작을 발생시킨 어플리케이션의 강제 종료 횟수가 기준 횟수 이상인지 확인할 수 있다. 여기서, 기준 횟수는 3회일 수 있으나, 이에 한정되지는 않는다. 프로세서(1110)는 강제 종료 횟수가 기준 이상인 경우, 해당 어플리케이션의 삭제를 유도하기 위한 사용자 인터페이스(예: 도 9d의 사용자 인터페이스)를 제공할 수 있다.
- [234] 도 12는 일 실시예에 따른 전자 장치에서 실행되는 프로세스의 블록도이다.
- [235] 도 12를 참조하면, 전자 장치의 프로세서(예: 도 11의 프로세서(1110))에서 실행되는 프로세스들(1200)은 리소스 매니저 데몬(resource manager daemon)(1210) 및 이상치 임계 값 계산기(outlier threshold calculator)(1250)를 포함할 수 있다.
- [236] 일 실시예에 따르면, 리소스 매니저 데몬(1210)은 어플리케이션이 실행되어 적어도 하나의 리소스를 사용하는 경우, 어플리케이션의 리소스 사용에 대응하는 이벤트를 확인할 수 있다. 예를 들어, 리소스 생성 이벤트는 클래스(예: ResEvt.class(java)), JNI(java native interface) 라이브러리(예: ResEvt JNI library(.cpp)) 및/또는 SO(shared object)(예: LibresEvt.so) 형태로 제공될 수 있다.
- [237] 일 실시예에 따르면, 리소스 매니저 데몬(1210)은 리소스 사용 통계(1220)를 기록하고, 이상치 임계 값(1230)을 관리할 수 있다. 리소스 매니저 데몬(1210)은 어플리케이션의 실행 시 어플리케이션 및 리소스 사용량 정보를 수신하고, 이를 리소스 사용 통계(1220)에 기록할 수 있다. 리소스 사용 통계(1220)에는 복수의 어플리케이션 및 각 어플리케이션이 사용한 각각의 리소스의 사용량 정보가 테이블 형태로 저장될 수 있다.
- [238] 일 실시예에 따르면, 이상치 임계 값 계산기(1250)는 메모리에 누적 저장된 리소스 사용 데이터 세트(1260)에 정해진 양의 데이터가 누적 저장된 경우, 이상치 임계 값을 계산하는 동작을 수행할 수 있다. 이상치 임계 값 계산기(1250)는 리소스 사용 데이터 세트(1260)에 기초하여, 미리 정해진 통계적 분석을 통해 특정 어플리케이션 및 특정 어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정할 수 있다. 이상치 임계 값 계산기(1250)는 복수의 어플리케이션 각각에 대해 리소스 각각에 대응하는 이상치 임계 값을 결정하는 동작을 수행할 수 있다. 예를 들어, i 개의 어플리케이션 및 j 개의 리소스에 대한 이상치 임계 값을 구하는 경우, App_0 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산하고, App_1 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계

값, ..., 리소스 j 에 대한 이상치 임계 값을 계산하고, ..., App_i 의 리소스 $_0$ 에 대한 이상치 임계 값, 리소스 $_1$ 에 대한 이상치 임계 값, ..., 리소스 j 에 대한 이상치 임계 값을 계산할 수 있다.

- [239] 일 실시예에 따르면, 이상치 임계 값 계산기(1250)는 특정 애플리케이션의 특정 리소스의 사용량 데이터의 정규성을 테스트(1270)하여, 정규 분포를 따르는 지 여부를 확인할 수 있다. 데이터가 정규 분포를 따르는 경우, 이상치 임계 값 계산기(1250)는 표준 편차를 계산하고, 표준 편차 및 미리 정해진 가중치에 기초하여 이상치 임계 값을 계산할 수 있다(1271). 데이터가 정규 분포를 따르지 않는 경우, 이상치 임계 값 계산기(1250)는 IQR(inter quartile range)를 계산하고, IQR 및 미리 정해진 가중치에 기초하여 이상치 임계 값을 계산할 수 있다(1272).
- [240] 이상치 임계 값 계산기(1250)가 이상치 임계 값을 계산하는 방법은 도 4 내지 도 10을 통해 설명한 서버 장치의 이상치 임계 값 계산기(1250)가 이상치 임계 값을 계산하는 방법에 대응될 수 있다.
- [241] 일 실시예에 따르면, 리소스 매니저 데몬(1210)은 리소스 사용 이벤트를 확인하는 경우, 각각의 리소스 사용량을 각 리소스에 대응하는 이상치 임계 값과 비교할 수 있다. 비교 결과, 해당 이벤트의 애플리케이션이 특정 리소스를 이상치 임계 값 이상으로 사용한 것으로 확인되는 경우, 리소스 매니저 데몬(1210)은 해당 애플리케이션이 리소스를 과다 사용하여 이상 동작하는 것으로 결정할 수 있다. 이 경우, 리소스 매니저 데몬(1210)은 해당 이벤트에 포함된 각 리소스의 사용량 정보를 리소스 사용 통계에 기록하지 않을 수 있다.
- [242] 일 실시예에 따르면, 리소스 매니저 데몬(1210)은 특정 애플리케이션이 이상 동작하는 것으로 결정하는 경우, 해당 애플리케이션의 동작을 중지하고(또는 실행을 종료하고), 리소스를 회수할 수 있다. 리소스 매니저 데몬(1210)은 해당 애플리케이션의 동작을 중지한 후 출력 장치(예: 디스플레이, 스피커, 햅틱)를 통해 사용자에게 이상 동작을 지시하는 알림을 제공하고, 해당 애플리케이션을 재시작하도록 유도할 수 있다. 리소스 매니저 데몬(1210)은 해당 애플리케이션의 재시작 이후에도 리소스를 과다하게 사용하는 문제가 기준 횟수 이상으로 반복해서 발생하는 경우, 출력 장치를 통해 사용자에게 해당 애플리케이션의 삭제를 유도하는 UI를 제공할 수 있다.
- [243] 도 13은 일 실시예에 따른 전자 장치의 이상치 임계 값을 결정하는 방법의 흐름도이다.
- [244] 도시된 방법은 전자 장치(예: 도 11의 전자 장치(1100))에 의해 수행될 수 있으며, 이하에서는 앞서 설명한 바 있는 기술적 특징에 대해서는 그 설명을 생략할 수도 있다.
- [245] 일 실시예에 따르면, 동작 1310에서, 전자 장치는 애플리케이션이 실행되는 경우, 실행된 애플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인할 수 있다. 예를 들어, 사용자의 입력에 따라 특정 애플리케이션이 실행되는 경우, 실행된 애플리케이션은 적어도 하나의 리소스(예: CPU 사용, 메모리 사용,

pending intent 생성)를 사용할 수 있으며, 전자 장치는 리소스 사용에 관련된 이벤트를 확인할 수 있다.

- [246] 일 실시예에 따르면, 동작 1320에서, 전자 장치는 리소스의 사용량 정보를 리소스 사용량 데이터 세트에 기록할 수 있다. 예를 들어, 리소스 사용 데이터 세트는 각각의 어플리케이션의 실행 시 사용되는 각각의 리소스의 사용량 정보를 누적해서 기록할 수 있으며, 전자 장치는 상기 어플리케이션의 실행에 따라 사용된 적어도 하나의 리소스의 사용량 정보를 리소스 사용 데이터 세트에서 대응되는 어플리케이션 및 리소스의 데이터에 추가할 수 있다.
- [247] 일 실시예에 따르면, 동작 1330에서, 전자 장치는 통계적 분석을 통해 어플리케이션 및 리소스 각각에 대응하는 이상치 임계 값을 결정할 수 있다.
- [248] 일 실시예에 따르면, 전자 장치는 리소스 사용 데이터 세트의 특정 어플리케이션 및 리소스 사용량에 대해 정규성 검증(normality test)을 수행할 수 있다. 특정 리소스 사용량 정보가 정규성을 갖는 것으로 확인되는 경우, 전자 장치는 리소스 사용량의 표준 편차를 계산하고, 계산된 표준 편차 및 미리 정해진 가중치(w_1)에 기초하여 이상치 임계 값을 결정할 수 있다. 특정 리소스 사용량 정보가 정규성을 갖지 않는 경우, 전자 장치는 리소스 사용량 정보의 IQR(inter quartile range)를 계산하고, IQR 및 미리 정해진 가중치(w_2)에 기초하여 이상치 임계 값을 결정할 수 있다. 여기서, 가중치(w_1) 및 가중치(w_2)는 각각의 리소스 별로 결정될 수 있다.
- [249] 일 실시예에 따르면, 전자 장치는 각각의 어플리케이션 및 리소스에 대응하는 각각의 이상치 임계 값을 결정할 수 있다.
- [250] 일 실시예에 따르면, 동작 1340에서, 전자 장치는 계산된 이상치 임계 값으로 기 저장된 이상치 임계 값을 업데이트 할 수 있다. 일 실시예에 따르면, 전자 장치는 초기화 단계에서 이상치 임계 값을 0보다 작은 값으로 설정할 수 있으며, 이후 이상치 임계 값이 계산되는 경우 계산된 값으로 업데이트 할 수 있다. 전자 장치는 리소스 사용 데이터 세트에 충분한 양의 데이터가 수집되는 경우, 이상치 임계 값을 계산하는 동작을 수행하고 계산된 이상치 임계 값을 다시 업데이트 할 수 있다.
- [251] 일 실시예에 따르면, 전자 장치는 특정 어플리케이션이 실행되는 경우, 실행된 어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 이 중 특정 리소스의 사용량이 해당 어플리케이션 및 리소스에 대해 설정된 이상치 임계 값 이상인 경우, 상기 실행된 어플리케이션으로 인해 이상 동작이 발생한 것으로 결정할 수 있다. 전자 장치는 이상 동작이 발생한 경우, 리소스를 과다 사용한 어플리케이션의 실행을 종료할 수 있다. 또한, 전자 장치는 이상 동작이 발생한 어플리케이션을 지시하는 정보 및 임계 값을 초과하여 사용된 적어도 하나의 리소스를 지시하는 정보를 포함하는 알람을 출력 장치를 통해 제공할 수 있다.
- [252] 본 문서의 다양한 실시예에 따른 전자 장치는, 메모리, 및 상기 메모리와 작동적으로 연결되는 프로세서를 포함할 수 있다.

- [253] 일 실시예에 따르면, 상기 메모리는, 복수의 어플리케이션 및 상기 복수의 어플리케이션 각각에 대응하는 복수의 리소스 사용량을 맵핑하여 기록하는 리소스 사용 데이터 세트를 저장할 수 있다.
- [254] 일 실시예에 따르면, 상기 프로세서는, 제1어플리케이션이 실행되는 경우, 상기 실행된 제1어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 상기 확인된 적어도 하나의 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록할 수 있다.
- [255] 일 실시예에 따르면, 상기 프로세서는, 상기 메모리에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 제1이상치 임계 값을 결정하고, 및 상기 메모리에 기 저장된 상기 제1어플리케이션 및 상기 적어도 하나의 리소스 각각에 대해 설정된 적어도 하나의 제2이상치 임계 값을 상기 결정된 적어도 하나의 이상치 임계 값에 기초하여 업데이트 하도록 설정될 수 있다.
- [256] 일 실시예에 따르면, 상기 메모리는, 상기 프로세서에 의해 실행 시에, 상기 전자 장치가, 제1어플리케이션이 실행되는 경우, 상기 실행된 제1어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 상기 확인된 적어도 하나의 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하고, 상기 메모리에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 제1이상치 임계 값을 결정하고, 및 상기 결정된 적어도 하나의 제1이상치 임계 값에 기초하여, 상기 메모리에 기 저장된 상기 제1어플리케이션 및 상기 적어도 하나의 리소스 각각에 대해 설정된 적어도 하나의 제2이상치 임계 값을 업데이트 하도록 하는 인스트럭션들을 저장할 수 있다.
- [257] 일 실시예에 따르면, 상기 프로세서는, 상기 실행된 제1어플리케이션이 사용하는 제1리소스의 사용량이 상기 제1어플리케이션 및 상기 제1리소스에 대응하여 설정된 이상치 임계 값을 초과하는 경우, 상기 제1리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하지 않도록 설정될 수 있다.
- [258] 일 실시예에 따르면, 상기 프로세서는, 상기 메모리에 저장된 리소스 사용 데이터 세트에서 제2어플리케이션에 대응되는 제2리소스의 사용량 정보가 기준 개수 이상으로 저장된 경우, 상기 제2어플리케이션 및 상기 제2리소스에 대응하는 이상치 임계 값을 결정하도록 설정될 수 있다.
- [259] 일 실시예에 따르면, 상기 프로세서는, 상기 제2어플리케이션에 대응되어 기록된 복수의 제2리소스의 사용량 정보의 IQR(inter quartile range)를 계산하고, 상기 계산된 IQR 및 상기 제2리소스에 대응하는 가중치에 기초하여, 상기 제2어플리케이션 및 상기 제2리소스에 대응하는 이상치 임계 값을 결정하도록 설정될 수 있다.

- [260] 일 실시예에 따르면, 상기 가중치는 각각의 리소스 별로 결정될 수 있다.
- [261] 일 실시예에 따르면, 상기 프로세서는, 상기 제2어플리케이션에 대응되어 기록된 복수의 제2리소스의 사용량 정보가 정규성을 갖는지 확인하고, 정규성을 갖는 것으로 확인되는 경우, 상기 복수의 제2리소스의 사용량 정보의 표준 편차를 계산하고, 상기 계산된 표준 편차에 기초하여 상기 제2어플리케이션 및 상기 제2리소스에 대응하는 이상치 임계 값을 결정하도록 설정될 수 있다.
- [262] 일 실시예에 따르면, 상기 프로세서는, 제3어플리케이션이 실행되는 경우, 상기 실행된 제3어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 상기 확인되는 적어도 하나의 리소스의 사용량 중 제3리소스의 사용량이 상기 제3어플리케이션 및 상기 제3리소스에 대응하여 현재 설정된 이상치 임계 값 이상인 경우, 상기 제3어플리케이션의 실행으로 인해 이상 동작이 발생한 것으로 결정하도록 설정될 수 있다.
- [263] 일 실시예에 따르면, 상기 프로세서는, 상기 이상 동작이 발생한 것으로 결정되는 경우, 상기 제3어플리케이션의 실행을 종료하도록 설정될 수 있다.
- [264] 일 실시예에 따르면, 상기 프로세서는, 상기 제3어플리케이션의 실행 시 이상 동작의 발생으로 인해 종료한 횟수가 기준 횟수 이상인 경우, 사용자에게 피드백을 제공하도록 설정될 수 있다.
- [265] 일 실시예에 따르면, 상기 전자 장치는 디스플레이를 더 포함하고, 상기 프로세서는, 상기 이상 동작이 발생된 제3어플리케이션을 지시하는 정보 및 상기 이상치 임계 값을 초과하여 사용된 적어도 하나의 리소스를 지시하는 정보를 포함하는 사용자 인터페이스를 상기 디스플레이를 통해 제공하도록 설정될 수 있다.
- [266] 본 문서의 다양한 실시예에 따른 서버 장치는, 통신 인터페이스, 메모리, 및 상기 통신 인터페이스 및 상기 메모리와 작동적으로 연결되는 프로세서를 포함할 수 있다.
- [267] 일 실시예에 따르면, 상기 프로세서는, 상기 통신 인터페이스를 통해, 복수의 전자 장치로부터 제1어플리케이션의 실행 시에 사용된 적어도 하나의 리소스의 사용량 정보를 수신하고, 상기 수신한 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하고, 상기 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정하고, 상기 통신 인터페이스를 통해, 상기 결정된 이상치 임계 값을 상기 복수의 전자 장치로 전송하도록 설정될 수 있다.
- [268] 일 실시예에 따르면, 상기 메모리는, 상기 프로세서에 의해 실행 시에, 상기 프로세서가, 상기 통신 인터페이스를 통해, 복수의 전자 장치로부터 제1어플리케이션의 실행 시에 사용된 적어도 하나의 리소스의 사용량 정보를 수신하고, 상기 수신한 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하고, 상기 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하

는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정하고, 상기 통신 인터페이스를 통해, 상기 결정된 이상치 임계 값을 상기 복수의 전자 장치로 전송하도록 하는 인스트럭션들을 저장할 수 있다.

- [269] 일 실시예에 따르면, 상기 프로세서는, 상기 메모리에 저장된 리소스 사용 데이터 세트에서 상기 제1어플리케이션에 대응되는 제1리소스의 사용량 정보가 기준 개수 이상으로 저장된 경우, 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하도록 설정될 수 있다.
- [270] 일 실시예에 따르면, 상기 프로세서는, 상기 제1어플리케이션에 대응되어 기록된 복수의 제1리소스의 사용량 정보의 IQR(inter quartile range)를 계산하고, 상기 계산된 IQR 및 상기 제1리소스에 대응하는 가중치에 기초하여, 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하도록 설정될 수 있다.
- [271] 일 실시예에 따르면, 상기 가중치는 각각의 리소스 별로 결정될 수 있다.
- [272] 일 실시예에 따르면, 상기 프로세서는, 상기 제1어플리케이션에 대응되어 기록된 복수의 제1리소스의 사용량 정보가 정규성을 갖는지 확인하고, 정규성을 갖는 것으로 확인되는 경우, 상기 복수의 제1리소스의 사용량 정보의 표준 편차를 계산하고, 상기 계산된 표준 편차에 기초하여 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하도록 설정될 수 있다.
- [273] 본 문서의 다양한 실시예에 따른, 서버 장치에 의해 수행되는 방법은, 복수의 전자 장치로부터 제1어플리케이션의 실행 시에 사용된 적어도 하나의 리소스의 사용량 정보를 수신하는 동작, 상기 수신한 리소스의 사용량 정보를 메모리에 저장된 리소스 사용 데이터 세트에 기록하는 동작, 상기 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정하는 동작, 및 상기 결정된 이상치 임계 값을 상기 복수의 전자 장치로 전송하는 동작을 포함할 수 있다.
- [274] 일 실시예에 따르면, 상기 적어도 하나의 이상치 임계 값을 결정하는 동작은, 상기 저장된 리소스 사용 데이터 세트에서 상기 제1어플리케이션에 대응되는 제1리소스의 사용량 정보가 기준 개수 이상으로 저장된 경우, 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하는 동작을 포함할 수 있다.
- [275] 일 실시예에 따르면, 상기 적어도 하나의 이상치 임계 값을 결정하는 동작은, 상기 제1어플리케이션에 대응되어 기록된 복수의 제1리소스의 사용량 정보의 IQR(inter quartile range)를 계산하는 동작, 및 상기 계산된 IQR 및 상기 제1리소스에 대응하는 가중치에 기초하여, 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하는 동작을 포함할 수 있다.
- [276] 일 실시예에 따르면, 상기 가중치는 각각의 리소스 별로 결정될 수 있다.

- [277] 일 실시예에 따르면, 상기 적어도 하나의 이상치 임계 값을 결정하는 동작은, 상기 제1어플리케이션에 대응되어 기록된 복수의 제1리소스의 사용량 정보가 정규성을 갖는지 확인하는 동작, 및 정규성을 갖는 것으로 확인되는 경우, 상기 복수의 제1리소스의 사용량 정보의 표준 편차를 계산하고, 상기 계산된 표준 편차에 기초하여 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하는 동작을 포함할 수 있다.
- [278] 본 문서의 다양한 실시예에 따른 전자 장치는, 디스플레이, 메모리, 및 상기, 디스플레이, 및 상기 메모리와 작동적으로 연결되는 프로세서를 포함할 수 있다.
- [279] 일 실시예에 따르면, 상기 메모리는, 복수의 어플리케이션 및 복수의 리소스에 각각 대응하는 복수의 이상치 임계 값을 저장할 수 있다.
- [280] 일 실시예에 따르면, 상기 프로세서는, 제1어플리케이션이 실행되는 경우, 상기 실행된 제1어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고, 상기 메모리에서, 상기 제1어플리케이션 및 상기 사용되는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 확인하고, 상기 확인된 적어도 하나의 리소스 중 제1리소스의 사용량을 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 제1이상치 임계 값과 비교하고, 및 상기 제1리소스의 사용량이 상기 제1이상치 임계 값 이상인 경우, 상기 제1어플리케이션의 실행으로 인해 이상 동작이 발생한 것으로 결정하도록 설정될 수 있다.
- [281] 일 실시예에 따르면, 상기 프로세서는, 상기 이상 동작이 발생한 것으로 결정되는 경우, 상기 제1어플리케이션의 실행을 종료하도록 설정될 수 있다.
- [282] 일 실시예에 따르면, 상기 프로세서는, 상기 제1어플리케이션의 실행 시 이상 동작의 발생으로 인해 종료한 횟수가 기준 횟수 이상인 경우, 사용자에게 피드백을 제공하도록 설정될 수 있다.
- [283] 본 문서에 개시된 다양한 실시예들에 따른 전자 장치는 다양한 형태의 장치가 될 수 있다. 전자 장치는, 예를 들면, 휴대용 통신 장치(예: 스마트폰), 컴퓨터 장치, 휴대용 멀티미디어 장치, 휴대용 의료 기기, 카메라, 웨어러블 장치, 또는 가전 장치를 포함할 수 있다. 본 문서의 실시예에 따른 전자 장치는 전술한 기기들에 한정되지 않는다.
- [284] 본 문서의 다양한 실시예들 및 이에 사용된 용어들은 본 문서에 기재된 기술적 특징들을 특정한 실시예들로 한정하려는 것이 아니며, 해당 실시예의 다양한 변경, 균등물, 또는 대체물을 포함하는 것으로 이해되어야 한다. 도면의 설명과 관련하여, 유사한 또는 관련된 구성요소에 대해서는 유사한 참조 부호가 사용될 수 있다. 아이টে에 대응하는 명사의 단수 형은 관련된 문맥상 명백하게 다르게 지시하지 않는 한, 상기 아이টে 한 개 또는 복수 개를 포함할 수 있다. 본 문서에서, "A 또는 B", "A 및 B 중 적어도 하나", "A 또는 B 중 적어도 하나", "A, B 또는 C", "A, B 및 C 중 적어도 하나", 및 "A, B, 또는 C 중 적어도 하나"와 같은 문구들 각각은 그 문구들 중 해당하는 문구에 함께 나열된 항목들 중 어느 하나, 또는 그들의 모든 가능한 조합을 포함할 수 있다. "제 1", "제 2", 또는 "첫째" 또는 "둘째"와 같은

용어들은 단순히 해당 구성요소를 다른 해당 구성요소와 구분하기 위해 사용될 수 있으며, 해당 구성요소들을 다른 측면(예: 중요성 또는 순서)에서 한정하지 않는다. 어떤(예: 제 1) 구성요소가 다른(예: 제 2) 구성요소에, "기능적으로" 또는 "통신적으로"라는 용어와 함께 또는 이런 용어 없이, "커플드" 또는 "커넥티드"라고 언급된 경우, 그것은 상기 어떤 구성요소가 상기 다른 구성요소에 직접적으로(예: 유선으로), 무선으로, 또는 제3구성요소를 통하여 연결될 수 있다는 것을 의미한다.

- [285] 본 문서의 다양한 실시예들에서 사용된 용어 "모듈"은 하드웨어, 소프트웨어 또는 펌웨어로 구현된 유닛을 포함할 수 있으며, 예를 들면, 로직, 논리 블록, 부품, 또는 회로와 같은 용어와 상호 호환적으로 사용될 수 있다. 모듈은, 일체로 구성된 부품 또는 하나 또는 그 이상의 기능을 수행하는, 상기 부품의 최소 단위 또는 그 일부가 될 수 있다. 예를 들면, 일 실시예에 따르면, 모듈은 ASIC(application-specific integrated circuit)의 형태로 구현될 수 있다.
- [286] 본 문서의 다양한 실시예들은 기기(machine)(예: 전자 장치(101)) 의해 읽을 수 있는 저장 매체(storage medium)(예: 내장 메모리(136) 또는 외장 메모리(138))에 저장된 하나 이상의 명령어들을 포함하는 소프트웨어(예: 프로그램(140))로서 구현될 수 있다. 예를 들면, 기기(예: 전자 장치(101))의 프로세서(예: 프로세서(120))는, 저장 매체로부터 저장된 하나 이상의 명령어들 중 적어도 하나의 명령어를 호출하고, 그것을 실행할 수 있다. 이것은 기기가 상기 호출된 적어도 하나의 명령어에 따라 적어도 하나의 기능을 수행하도록 운영되는 것을 가능하게 한다. 상기 하나 이상의 명령어들은 컴파일러에 의해 생성된 코드 또는 인터프리터에 의해 실행될 수 있는 코드를 포함할 수 있다. 기기로 읽을 수 있는 저장 매체는, 비일시적(non-transitory) 저장 매체의 형태로 제공될 수 있다. 여기서, '비일시적'은 저장 매체가 실제(tangible)하는 장치이고, 신호(signal)(예: 전자기파)를 포함하지 않는다는 것을 의미할 뿐이며, 이 용어는 데이터가 저장 매체에 반영구적으로 저장되는 경우와 임시적으로 저장되는 경우를 구분하지 않는다.
- [287] 일 실시예에 따르면, 본 문서에 개시된 다양한 실시예들에 따른 방법은 컴퓨터 프로그램 제품(computer program product)에 포함되어 제공될 수 있다. 컴퓨터 프로그램 제품은 상품으로서 판매자 및 구매자 간에 거래될 수 있다. 컴퓨터 프로그램 제품은 기기로 읽을 수 있는 저장 매체(예: compact disc read only memory(CD-ROM))의 형태로 배포되거나, 또는 어플리케이션 스토어(예: 플레이 스토어™)를 통해 또는 두 개의 사용자 장치들(예: 스마트폰들) 간에 직접, 온라인으로 배포(예: 다운로드 또는 업로드)될 수 있다. 온라인 배포의 경우에, 컴퓨터 프로그램 제품의 적어도 일부는 제조사의 서버, 어플리케이션 스토어의 서버, 또는 중계 서버의 메모리와 같은 기기로 읽을 수 있는 저장 매체에 적어도 일시 저장되거나, 임시적으로 생성될 수 있다.
- [288] 다양한 실시예들에 따르면, 상기 기술한 구성요소들의 각각의 구성요소(예: 모듈 또는 프로그램)는 단수 또는 복수의 개체를 포함할 수 있으며, 복수의 개체 중

일부는 다른 구성요소에 분리 배치될 수도 있다. 다양한 실시예들에 따르면, 전술한 해당 구성요소들 중 하나 이상의 구성요소들 또는 동작들이 생략되거나, 또는 하나 이상의 다른 구성요소들 또는 동작들이 추가될 수 있다. 대체적으로 또는 추가적으로, 복수의 구성요소들(예: 모듈 또는 프로그램)은 하나의 구성요소로 통합될 수 있다. 이런 경우, 통합된 구성요소는 상기 복수의 구성요소들 각각의 구성요소의 하나 이상의 기능들을 상기 통합 이전에 상기 복수의 구성요소들 중 해당 구성요소에 의해 수행되는 것과 동일 또는 유사하게 수행할 수 있다. 다양한 실시예들에 따르면, 모듈, 프로그램 또는 다른 구성요소에 의해 수행되는 동작들은 순차적으로, 병렬적으로, 반복적으로, 또는 휴리스틱하게 실행되거나, 상기 동작들 중 하나 이상이 다른 순서로 실행되거나, 생략되거나, 또는 하나 이상의 다른 동작들이 추가될 수 있다.

청구범위

- [청구항 1] 전자 장치에 있어서,
메모리; 및
상기 메모리와 작동적으로 연결되는 프로세서를 포함하고,
상기 메모리는,
복수의 어플리케이션 및 상기 복수의 어플리케이션 각각에 대응하는 복
수의 리소스 사용량을 맵핑하여 기록하는 리소스 사용 데이터 세트를 저
장하고,
상기 프로세서에 의해 실행 시에, 상기 전자 장치가,
제1어플리케이션이 실행되는 경우, 상기 실행된 제1어플리케이션이 사용
하는 적어도 하나의 리소스의 사용량을 확인하고,
상기 확인된 적어도 하나의 리소스의 사용량 정보를 상기 메모리에 저장
된 리소스 사용 데이터 세트에 기록하고,
상기 메모리에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진
통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의
실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나
의 제1이상치 임계 값을 결정하고, 및
상기 메모리에 기 저장된 상기 제1어플리케이션 및 상기 적어도 하나의
리소스 각각에 대해 설정된 적어도 하나의 제2이상치 임계 값을 상기 결
정된 적어도 하나의 제1이상치 임계 값에 기초하여 업데이트 하도록 하는
인스트럭션들을 저장하는 전자 장치.
- [청구항 2] 제 1항에 있어서,
상기 인스트럭션들은, 상기 전자 장치가,
상기 실행된 제1어플리케이션이 사용하는 제1리소스의 사용량이 상기 제
1어플리케이션 및 상기 제1리소스에 대응하여 설정된 이상치 임계 값을
초과하는 경우, 상기 제1리소스의 사용량 정보를 상기 메모리에 저장된
리소스 사용 데이터 세트에 기록하지 않도록 하는 인스트럭션들을 포함
하는 전자 장치.
- [청구항 3] 제 1항 또는 제 2항에 있어서,
상기 인스트럭션들은, 상기 전자 장치가,
상기 메모리에 저장된 리소스 사용 데이터 세트에서 제2어플리케이션에
대응되는 제2리소스의 사용량 정보가 기준 개수 이상으로 저장된 경우,
상기 제2어플리케이션 및 상기 제2리소스에 대응하는 이상치 임계 값을
결정하도록 하는 인스트럭션들을 포함하는 전자 장치.
- [청구항 4] 제 3항에 있어서,
상기 인스트럭션들은, 상기 전자 장치가,

상기 제2어플리케이션에 대응되어 기록된 복수의 제2리소스의 사용량 정보의 IQR(inter quartile range)를 계산하고,
 상기 계산된 IQR 및 상기 제2리소스에 대응하는 가중치에 기초하여, 상기 제2어플리케이션 및 상기 제2리소스에 대응하는 이상치 임계 값을 결정하도록 하는 인스트럭션들을 포함하는 전자 장치.

[청구항 5]

제 4항에 있어서,
 상기 가중치는 각각의 리소스 별로 결정되는 전자 장치.

[청구항 6]

제 3항에 있어서,
 상기 인스트럭션들은, 상기 전자 장치가,
 상기 제2어플리케이션에 대응되어 기록된 복수의 제2리소스의 사용량 정보가 정규성을 갖는지 확인하고,
 정규성을 갖는 것으로 확인되는 경우, 상기 복수의 제2리소스의 사용량 정보의 표준 편차를 계산하고, 상기 계산된 표준 편차에 기초하여 상기 제2어플리케이션 및 상기 제2리소스에 대응하는 이상치 임계 값을 결정하도록 하는 인스트럭션들을 포함하는 전자 장치.

[청구항 7]

제 1항에 있어서,
 상기 인스트럭션들은, 상기 전자 장치가,
 제3어플리케이션이 실행되는 경우, 상기 실행된 제3어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하고,
 상기 확인되는 적어도 하나의 리소스의 사용량 중 제3리소스의 사용량이 상기 제3어플리케이션 및 상기 제3리소스에 대응하여 현재 설정된 이상치 임계 값 이상인 경우, 상기 제3어플리케이션의 실행으로 인해 이상 동작이 발생한 것으로 결정하도록 하는 인스트럭션들을 포함하는 전자 장치.

[청구항 8]

제 7항에 있어서,
 상기 인스트럭션들은, 상기 전자 장치가,
 상기 이상 동작이 발생한 것으로 결정되는 경우, 상기 제3어플리케이션의 실행을 종료하도록 하는 인스트럭션들을 포함하는 전자 장치.

[청구항 9]

제 7항에 있어서,
 디스플레이를 더 포함하고,
 상기 인스트럭션들은, 상기 전자 장치가,
 상기 이상 동작이 발생된 제3어플리케이션을 지시하는 정보 및 상기 이상치 임계 값을 초과하여 사용된 적어도 하나의 리소스를 지시하는 정보를 포함하는 사용자 인터페이스를 상기 디스플레이를 통해 제공하도록 하는 인스트럭션들을 포함하는 전자 장치.

[청구항 10]

서버 장치에 있어서,
 통신 인터페이스;
 메모리; 및

상기 통신 인터페이스 및 상기 메모리와 작동적으로 연결되는 프로세서를 포함하고,

상기 메모리는, 상기 프로세서에 의해 실행 시에, 상기 서버 장치가, 상기 통신 인터페이스를 통해, 복수의 전자 장치로부터 제1어플리케이션의 실행 시에 사용된 적어도 하나의 리소스의 사용량 정보를 수신하고, 상기 수신한 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하고,

상기 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정하고,

상기 통신 인터페이스를 통해, 상기 결정된 이상치 임계 값을 상기 복수의 전자 장치로 전송하도록 하는 인스트럭션들을 포함하는 서버 장치.

[청구항 11]

제 10항에 있어서,

상기 인스트럭션들은, 상기 서버 장치가,

상기 메모리에 저장된 리소스 사용 데이터 세트에서 상기 제1어플리케이션에 대응되는 제1리소스의 사용량 정보가 기준 개수 이상으로 저장된 경우, 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하도록 하는 인스트럭션들을 포함하는 서버 장치.

[청구항 12]

제 11항에 있어서,

상기 인스트럭션들은, 상기 서버 장치가,

상기 제1어플리케이션에 대응되어 기록된 복수의 제1리소스의 사용량 정보의 IQR(inter quartile range)를 계산하고,

상기 계산된 IQR 및 상기 제1리소스에 대응하는 가중치에 기초하여, 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하도록 하는 인스트럭션들을 포함하는 서버 장치.

[청구항 13]

제 12항에 있어서,

상기 인스트럭션들은, 상기 서버 장치가,

상기 제1어플리케이션에 대응되어 기록된 복수의 제1리소스의 사용량 정보가 정규성을 갖는지 확인하고,

정규성을 갖는 것으로 확인되는 경우, 상기 복수의 제1리소스의 사용량 정보의 표준 편차를 계산하고, 상기 계산된 표준 편차에 기초하여 상기 제1어플리케이션 및 상기 제1리소스에 대응하는 이상치 임계 값을 결정하도록 하는 인스트럭션들을 포함하는 서버 장치.

[청구항 14]

서버 장치에 의해 수행되는 방법에 있어서,

복수의 전자 장치로부터 제1어플리케이션의 실행 시에 사용된 적어도 하나의 리소스의 사용량 정보를 수신하는 동작;

상기 수신한 리소스의 사용량 정보를 메모리에 저장된 리소스 사용 데이터 세트에 기록하는 동작;

상기 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 이상치 임계 값을 결정하는 동작; 및

상기 결정된 이상치 임계 값을 상기 복수의 전자 장치로 전송하는 동작을 포함하는 방법.

[청구항 15]

컴퓨터로 판독 가능한 비-일시적 기록 매체에 있어서,

복수의 어플리케이션 및 상기 복수의 어플리케이션 각각에 대응하는 복수의 리소스 사용량을 맵핑하여 기록하는 리소스 사용 데이터 세트를 저장하는 동작;

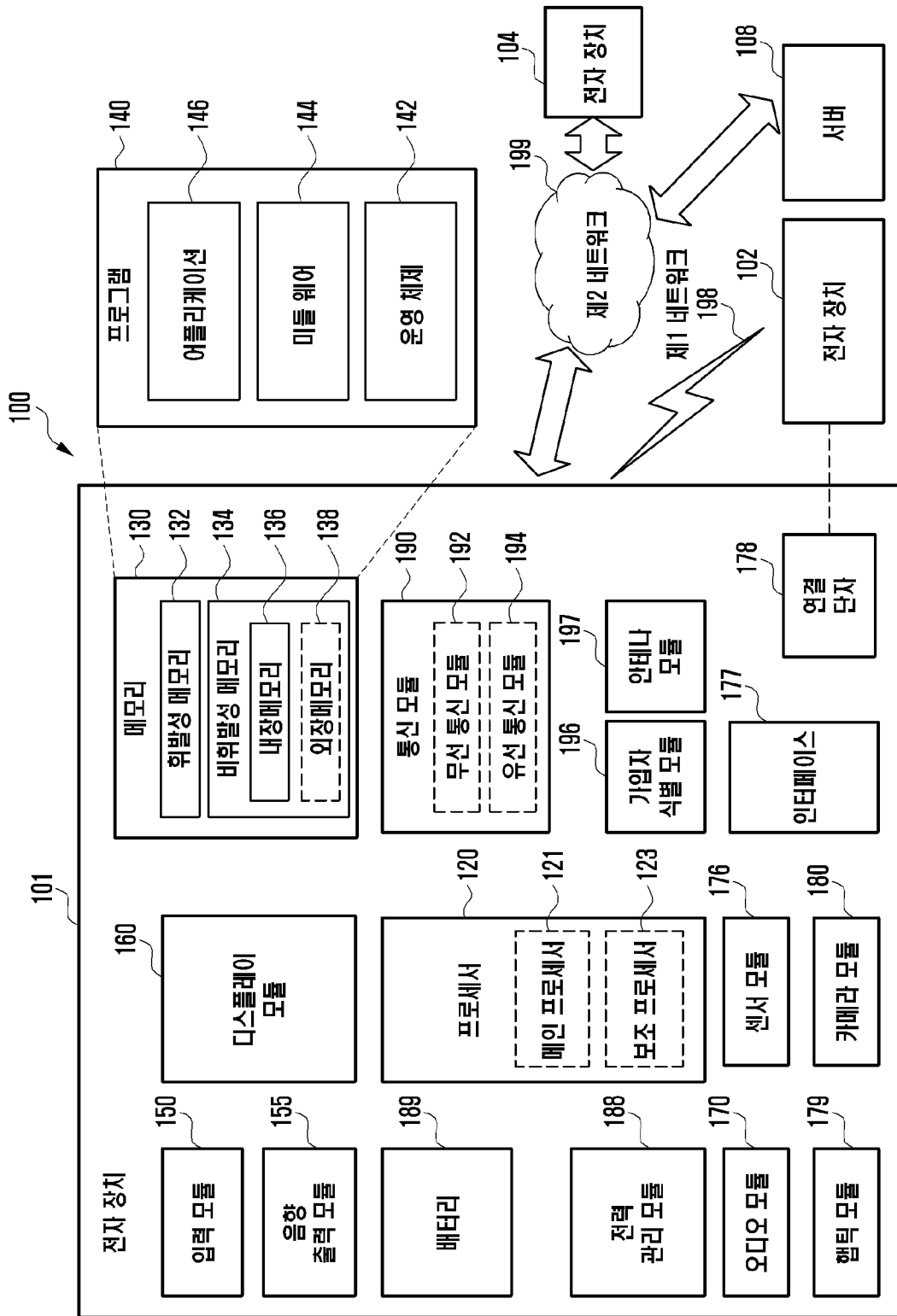
제1어플리케이션이 실행되는 경우, 상기 실행된 제1어플리케이션이 사용하는 적어도 하나의 리소스의 사용량을 확인하는 동작;

상기 확인된 적어도 하나의 리소스의 사용량 정보를 상기 메모리에 저장된 리소스 사용 데이터 세트에 기록하는 동작;

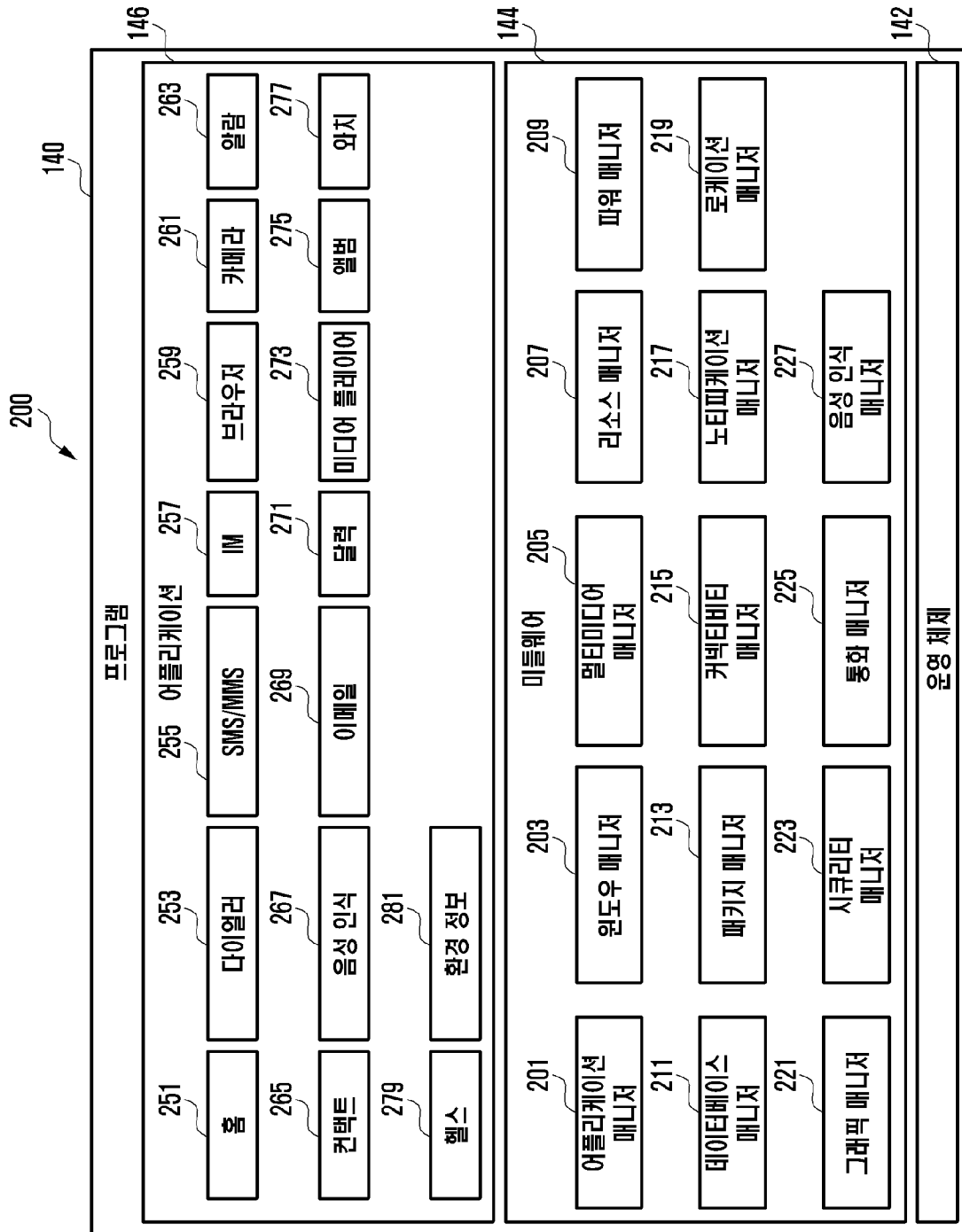
상기 메모리에 저장된 리소스 사용 데이터 세트에 기초하여, 미리 정해진 통계적 분석을 통해 상기 제1어플리케이션 및 상기 제1어플리케이션의 실행 시에 사용하는 적어도 하나의 리소스 각각에 대응하는 적어도 하나의 제1이상치 임계 값을 결정하는 동작;

기 저장된 상기 제1어플리케이션 및 상기 적어도 하나의 리소스 각각에 대해 설정된 적어도 하나의 제2이상치 임계 값을 상기 결정된 적어도 하나의 제1이상치 임계 값에 기초하여 업데이트 하는 동작을 수행하기 위한 인스트럭션들을 저장하는 기록 매체.

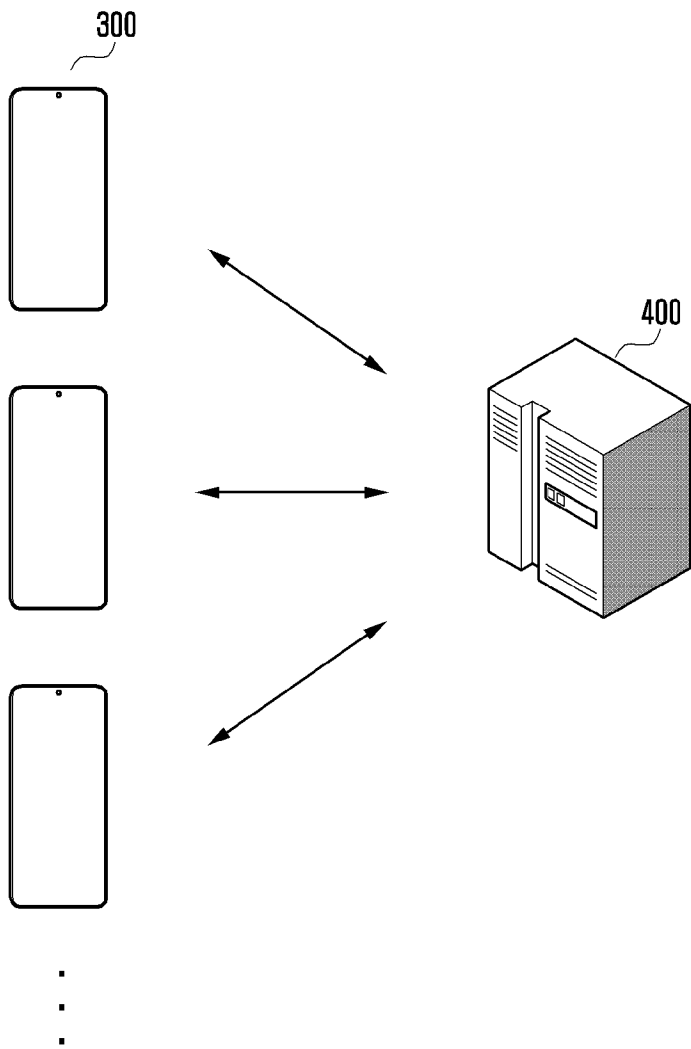
[도 1]



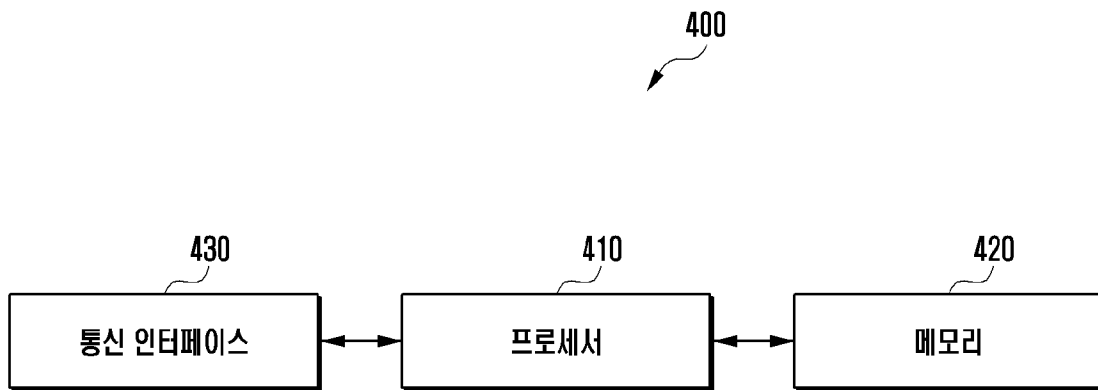
[도2]



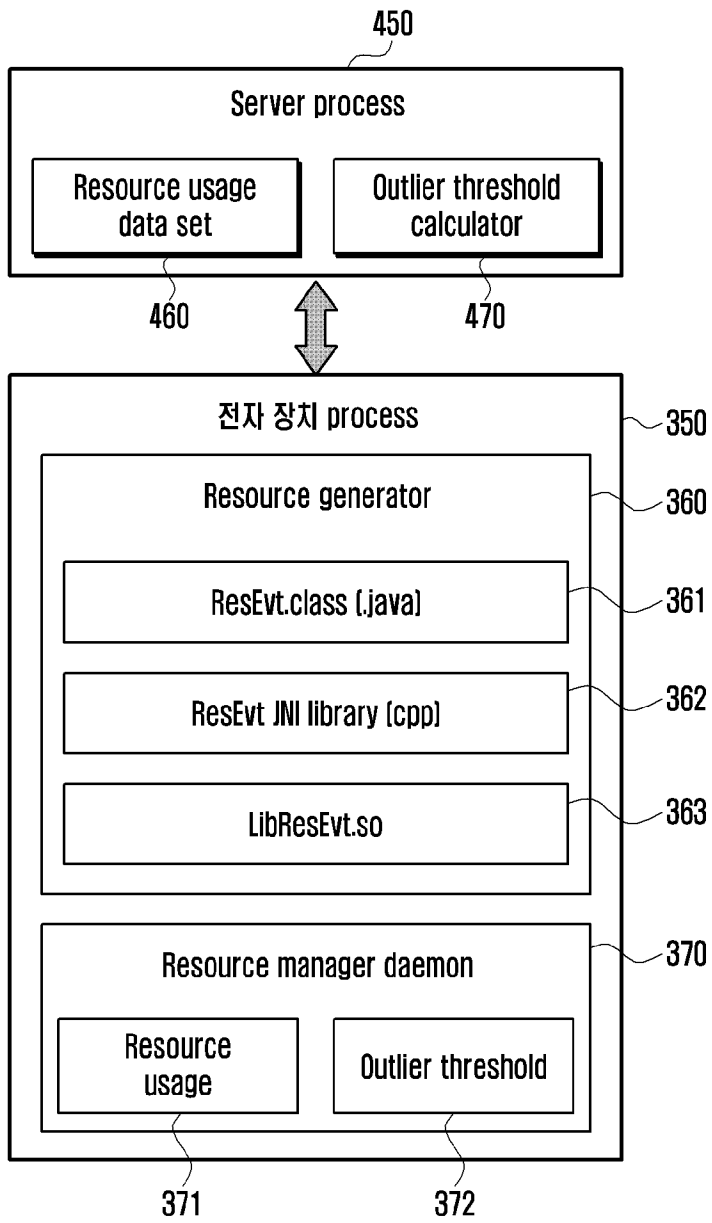
[도3]



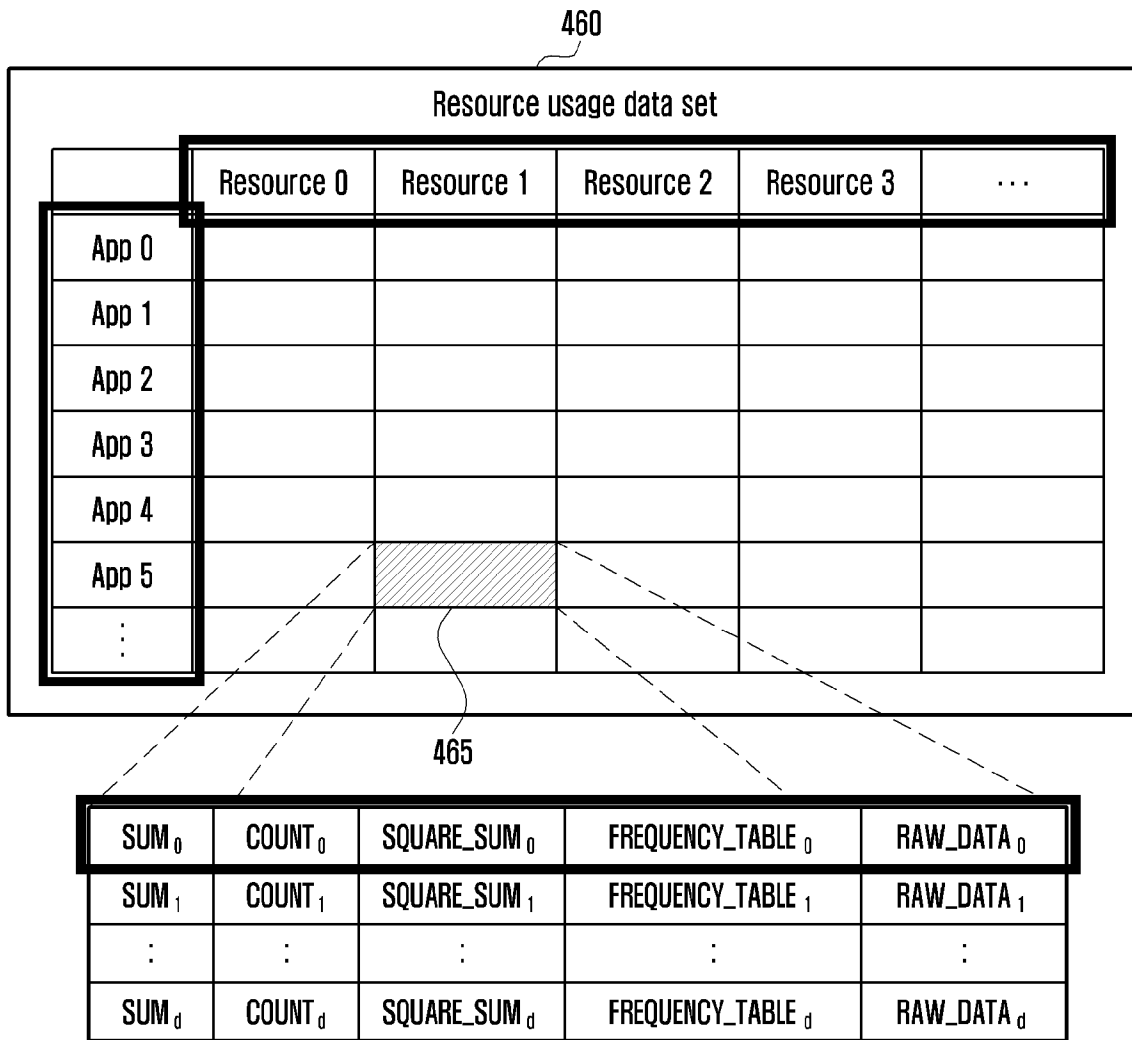
[도4]



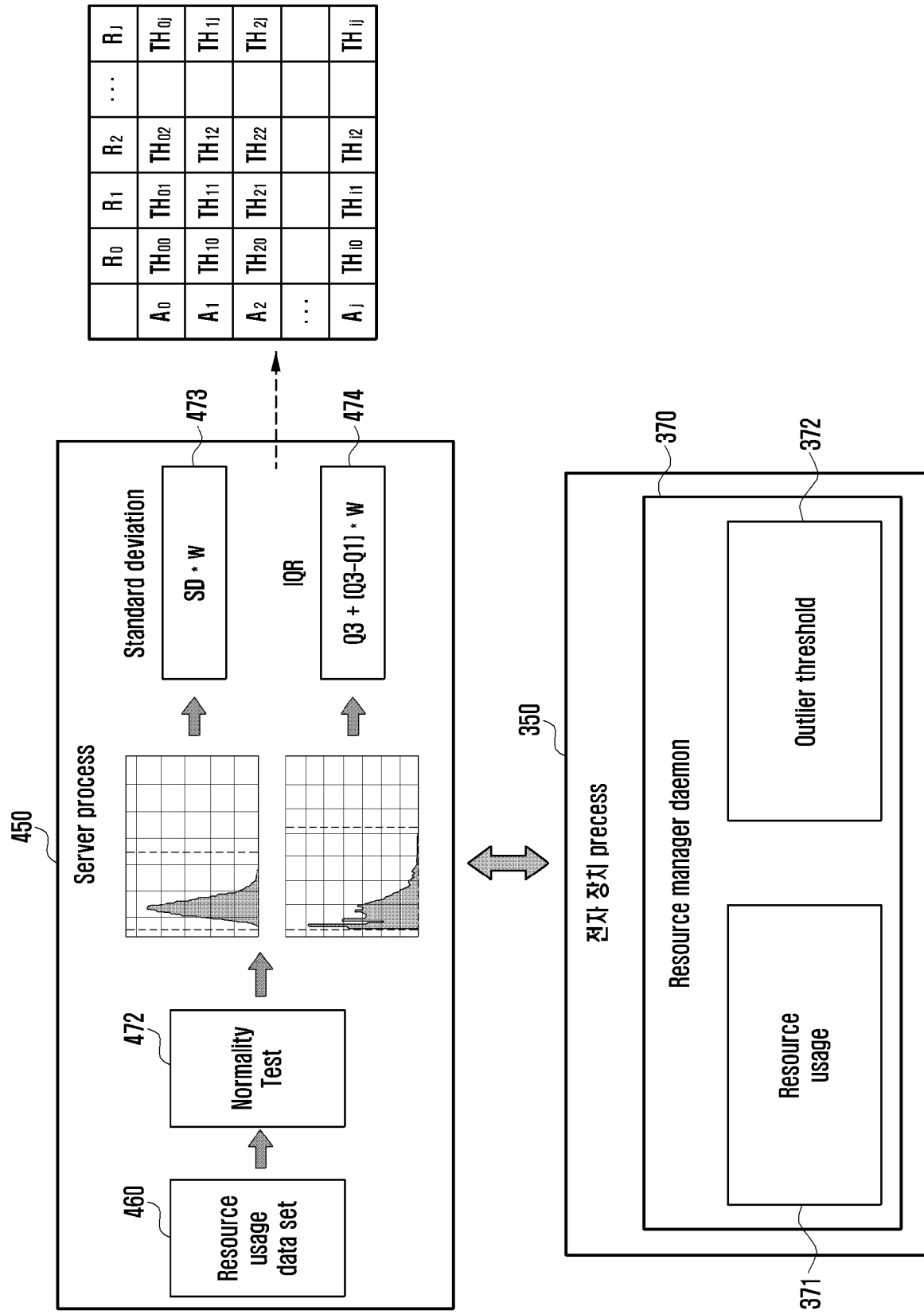
[도5]



[도6]



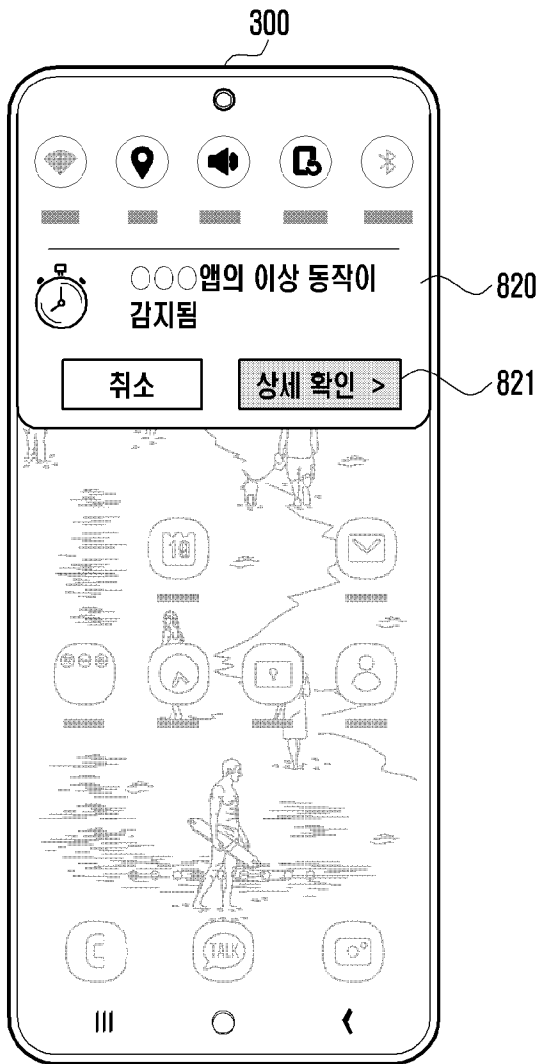
[도7]



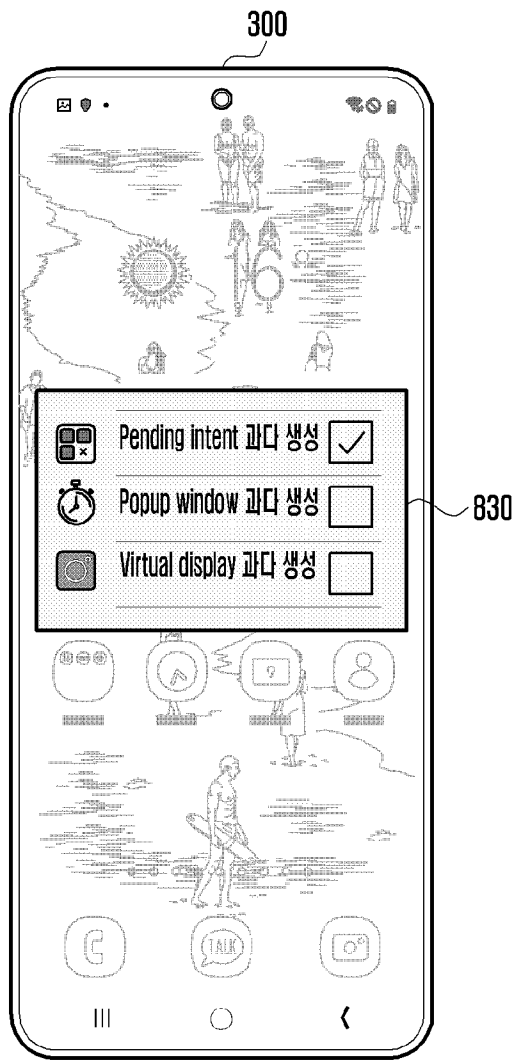
[도8a]



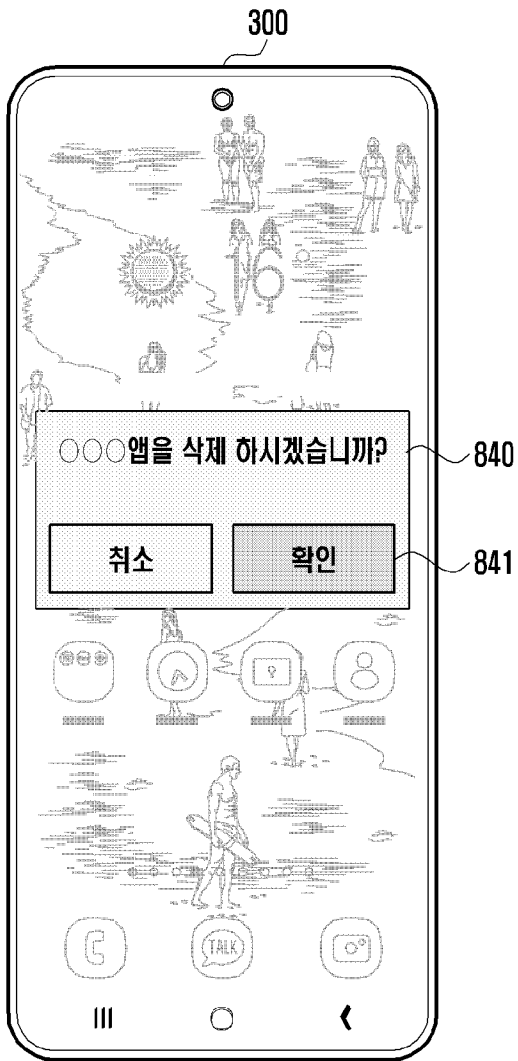
[도8b]



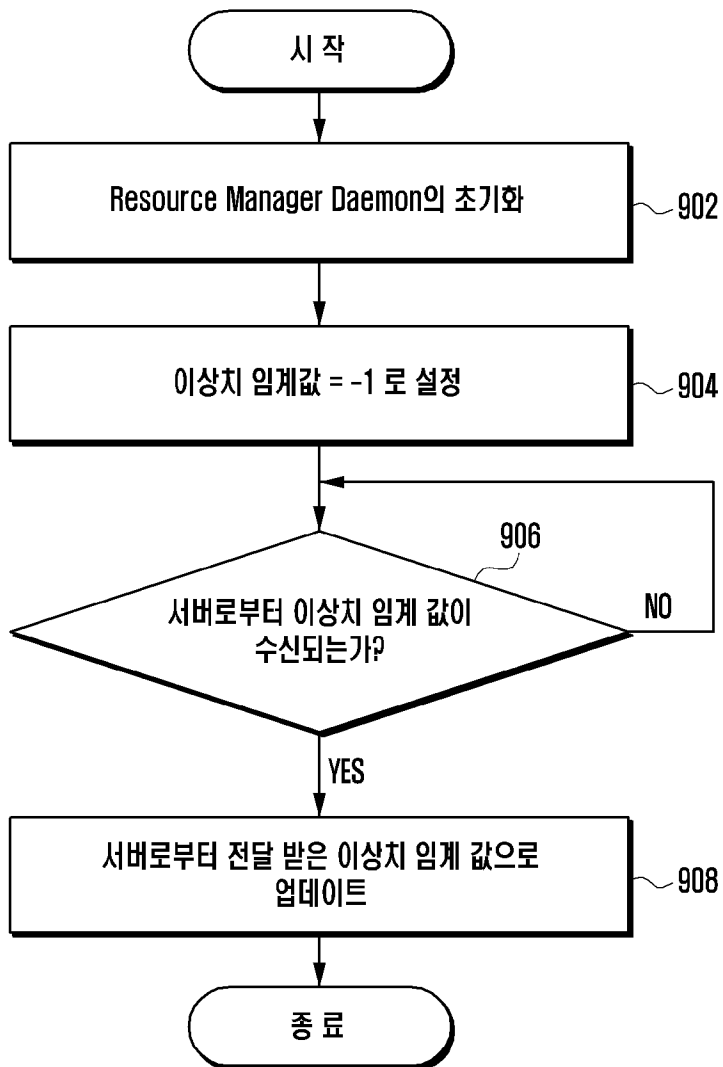
[도8c]



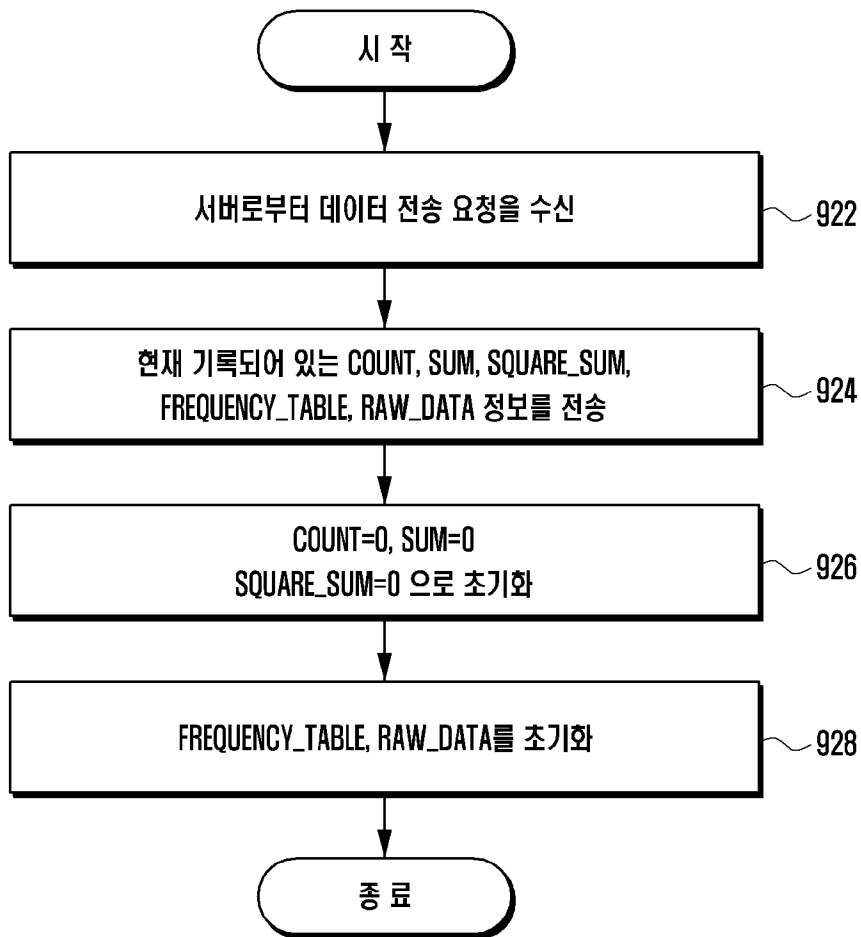
[도8d]



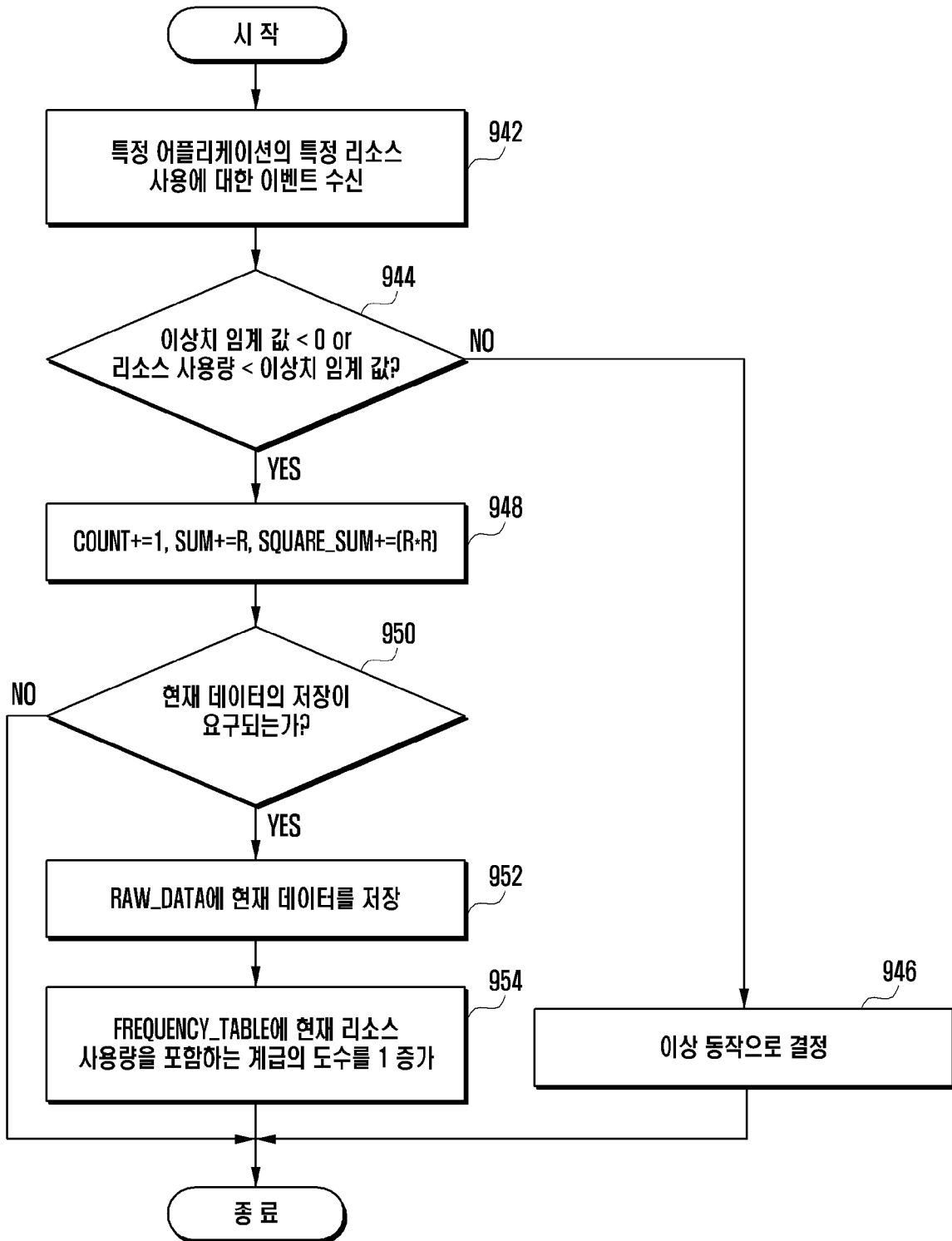
[도9a]



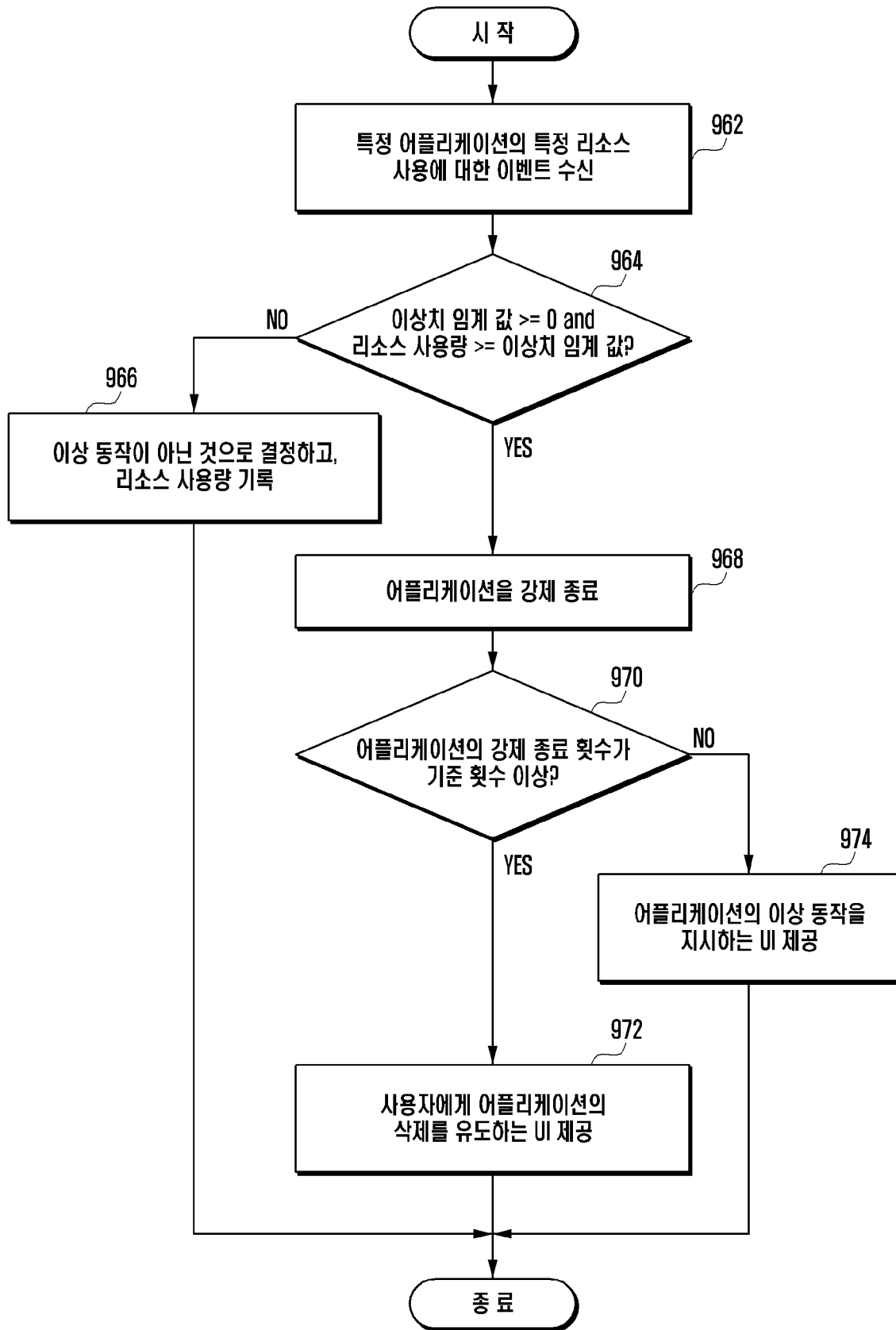
[도9b]



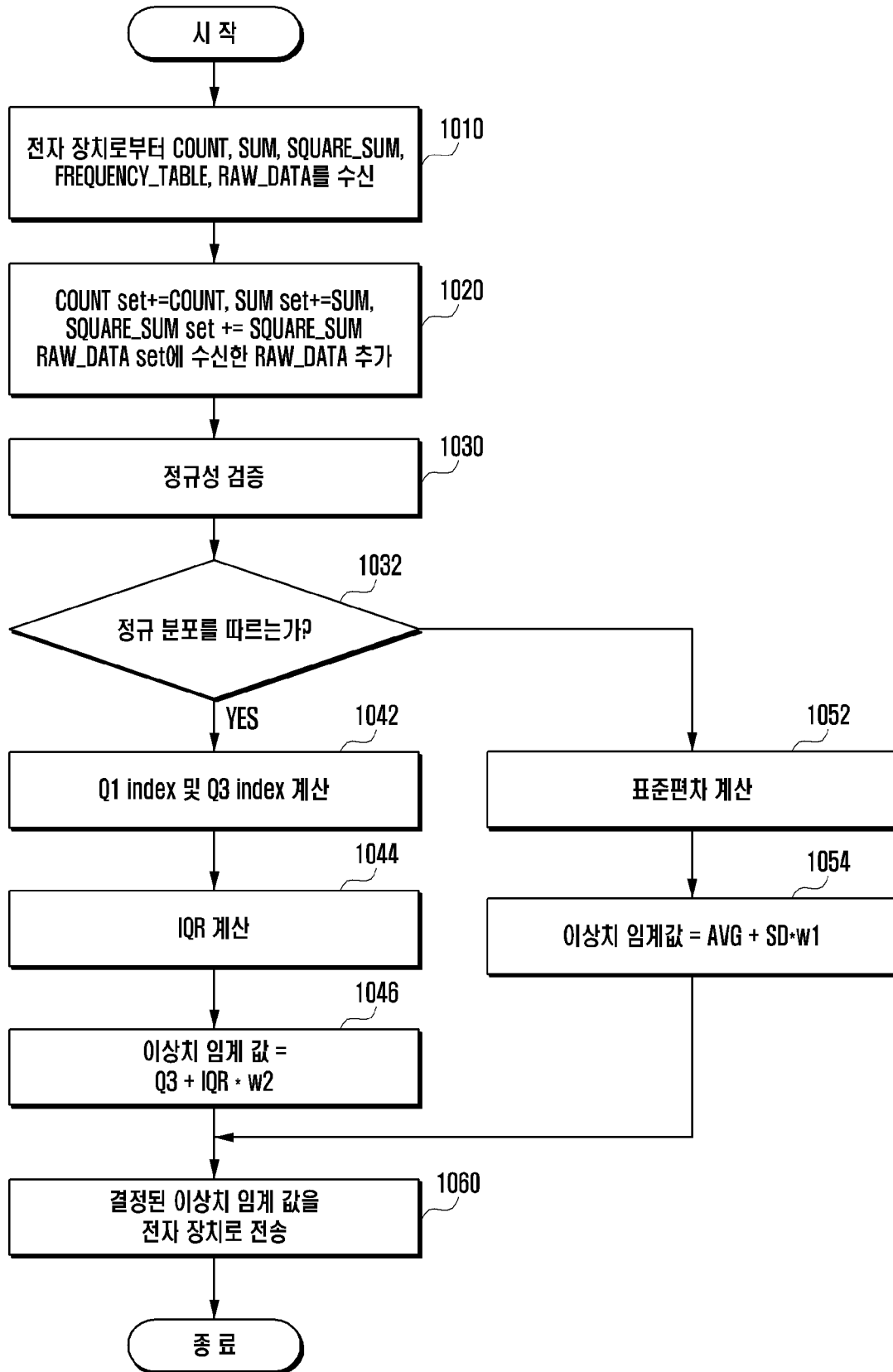
[도9c]



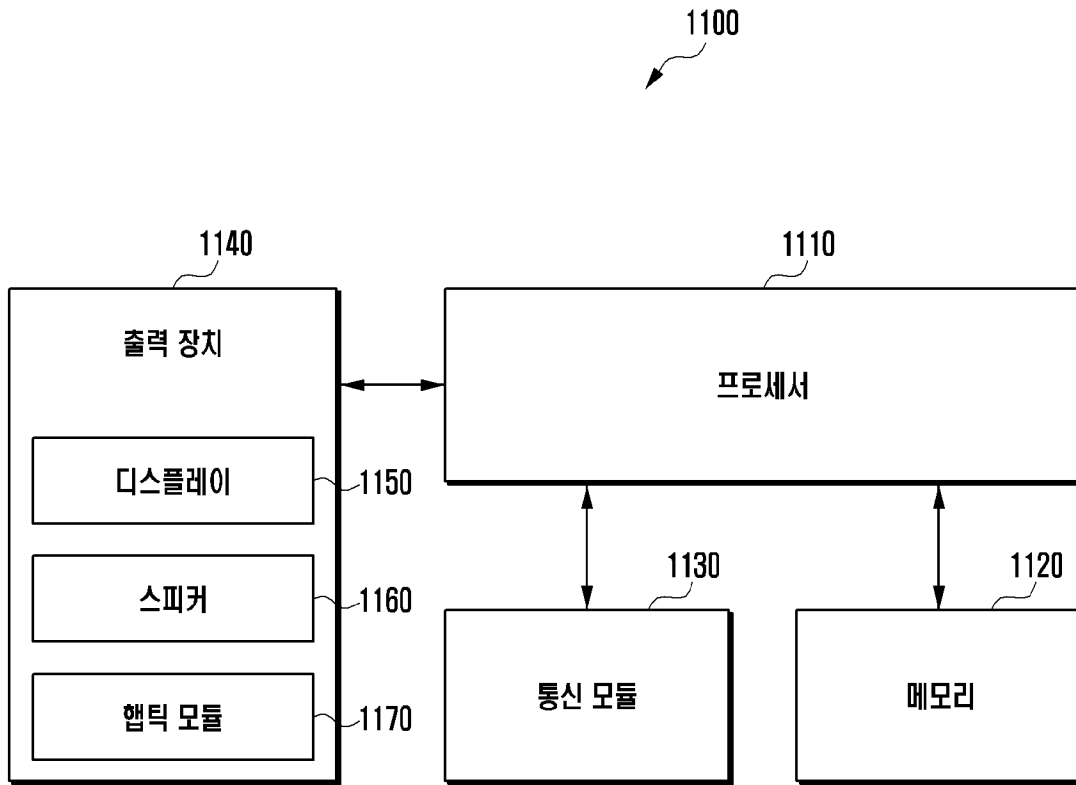
[도9d]



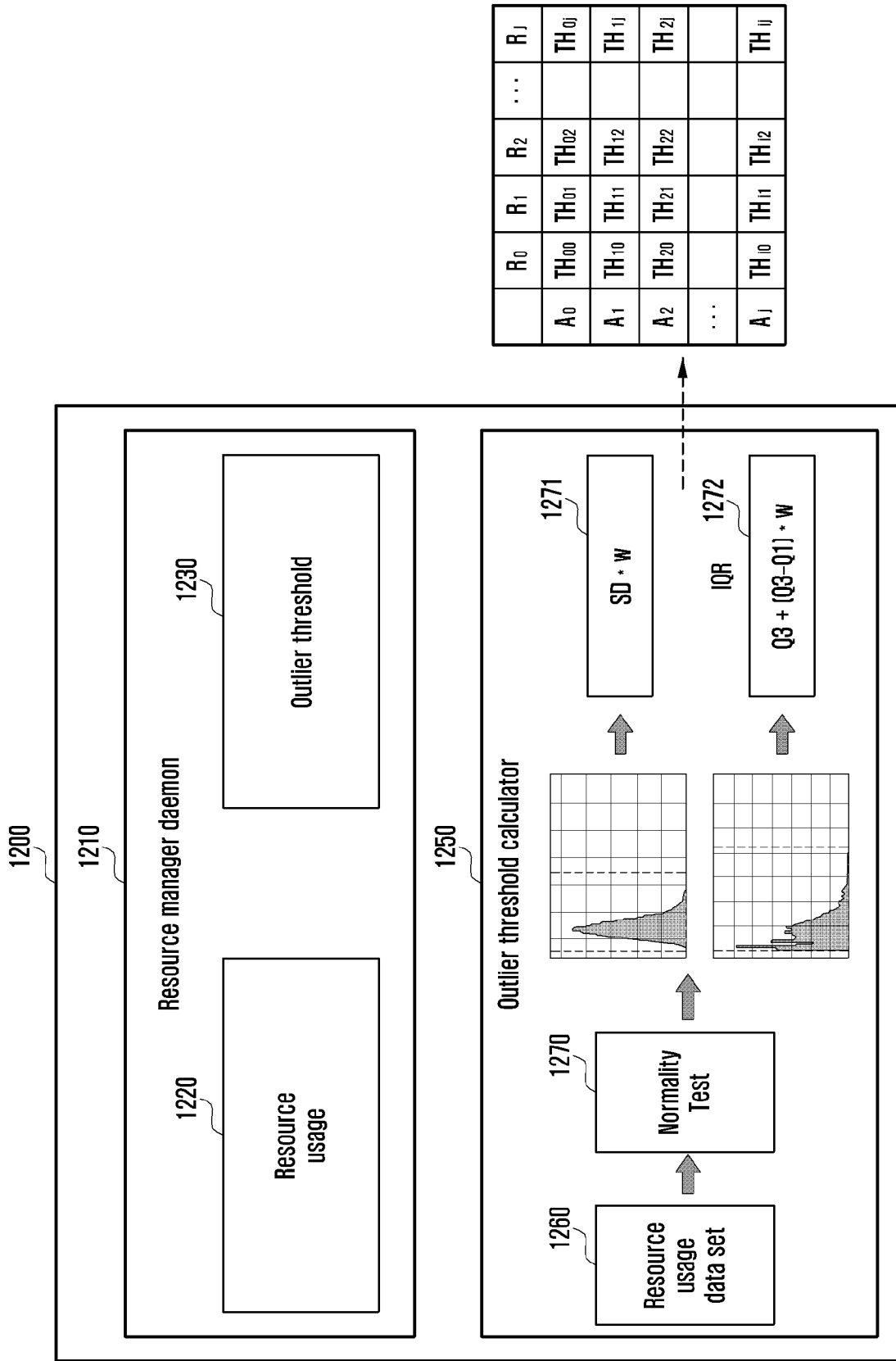
[도 10]



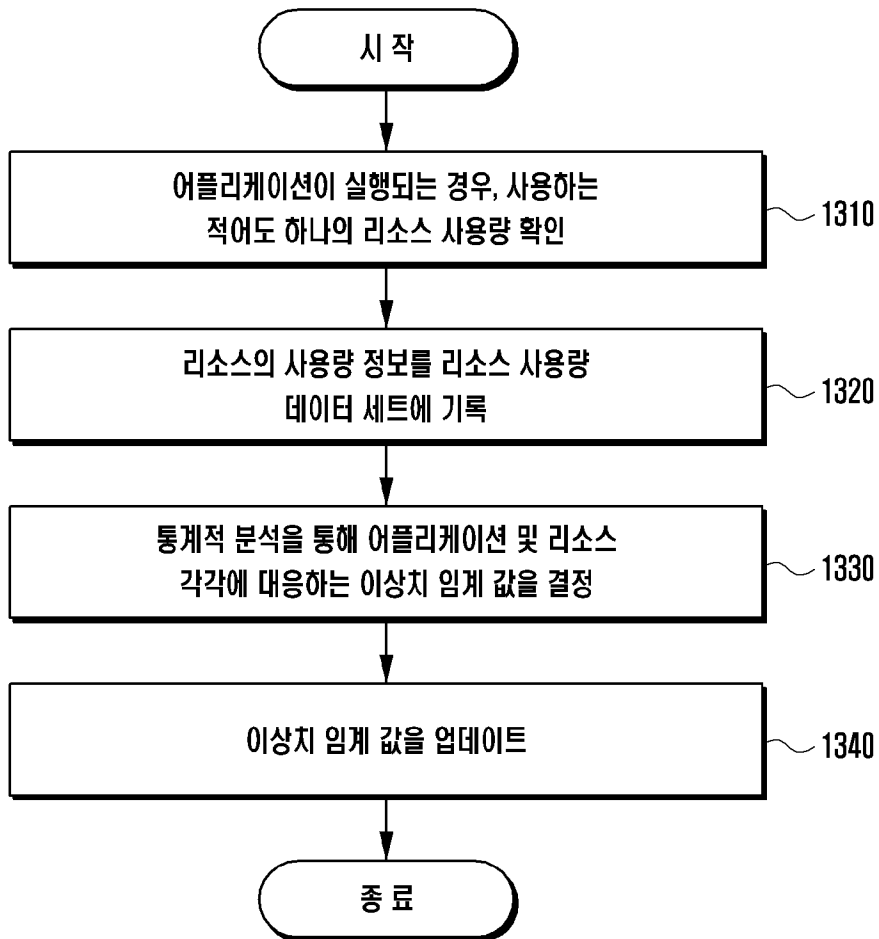
[도11]



[도 12]



[도 13]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2024/007587

A. CLASSIFICATION OF SUBJECT MATTER G06F 11/30(2006.01)i; G06F 11/34(2006.01)i; G06F 9/50(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 11/30(2006.01); G06F 11/07(2006.01); G06F 11/34(2006.01); G06F 12/02(2006.01); G06F 9/44(2006.01); G06F 9/50(2006.01); 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: 어플리케이션(application), 리소스(resource), 사용량(usage), 통계적 분석 (statistical analysis), 임계(threshold), 업데이트(update)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	KR 10-2020-0052361 A (HUAWEI TECHNOLOGIES CO., LTD.) 14 May 2020 (2020-05-14) See paragraphs [0006]-[0010], [0032]-[0099] and [0113]; and figures 2, 5-6 and 14.	1-3,6-11,14-15 4-5,12-13
A	KR 10-2017-0065085 A (SAMSUNG ELECTRONICS CO., LTD.) 13 June 2017 (2017-06-13) See paragraphs [0192] and [0223]; and figure 30.	1-15
A	KR 10-1987664 B1 (NAMU TECH CO., LTD. et al.) 11 June 2019 (2019-06-11) See paragraphs [0072]-[0082] and [0126]; and figure 15.	1-15
A	KR 10-2012-0026046 A (MICROSOFT CORPORATION) 16 March 2012 (2012-03-16) See paragraphs [0016] and [0037]; and figure 1.	1-15
A	US 2018-0332107 A1 (AMAZON TECHNOLOGIES, INC.) 15 November 2018 (2018-11-15) See paragraphs [0026] and [0060]; and figure 5.	1-15
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 09 September 2024		Date of mailing of the international search report 09 September 2024
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
Information on patent family members

International application No.

PCT/KR2024/007587

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
KR	10-2020-0052361	A	14 May 2020	CN	111527478	A	11 August 2020
				EP	3673375	A1	01 July 2020
				JP	2020-537215	A	17 December 2020
				JP	7120708	B2	17 August 2022
				KR	10-2339239	B1	14 December 2021
				US	11321210	B2	03 May 2022
				US	2020-0371892	A1	26 November 2020
				WO	2019-075399	A1	18 April 2019
KR	10-2017-0065085	A	13 June 2017	KR	10-2502569	B1	23 February 2023
				US	10474574	B2	12 November 2019
				US	2018-0357166	A1	13 December 2018
				WO	2017-095195	A1	08 June 2017
KR	10-1987664	B1	11 June 2019	CN	112437915	A	02 March 2021
				JP	2021-530802	A	11 November 2021
				US	2021-0279157	A1	09 September 2021
				WO	2020-017844	A1	23 January 2020
KR	10-2012-0026046	A	16 March 2012	CN	102414674	A	11 April 2012
				EP	2425349	A2	07 March 2012
				JP	2012-525641	A	22 October 2012
				JP	5608222	B2	15 October 2014
				KR	10-1600129	B1	04 March 2016
				US	2010-0281482	A1	04 November 2010
				US	8261266	B2	04 September 2012
				WO	2010-126805	A2	04 November 2010
US	2018-0332107	A1	15 November 2018	CN	104854563	A	19 August 2015
				EP	2898410	A1	29 July 2015
				JP	2015-532992	A	16 November 2015
				JP	6049887	B2	21 December 2016
				US	10015241	B2	03 July 2018
				US	10542079	B2	21 January 2020
				US	2014-0082165	A1	20 March 2014
				US	2014-0082614	A1	20 March 2014
				US	2016-0241637	A1	18 August 2016
				US	9135048	B2	15 September 2015
				US	9323577	B2	26 April 2016
				WO	2014-047073	A1	27 March 2014

A. 발명이 속하는 기술분류(국제특허분류(IPC)) G06F 11/30(2006.01)i; G06F 11/34(2006.01)i; G06F 9/50(2006.01)i		
B. 조사된 분야 조사된 최소문헌(국제특허분류를 기재) G06F 11/30(2006.01); G06F 11/07(2006.01); G06F 11/34(2006.01); G06F 12/02(2006.01); G06F 9/44(2006.01); G06F 9/50(2006.01); 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: 어플리케이션(application), 리소스(resource), 사용량(usage), 통계적 분석(statistical analysis), 임계(threshold), 업데이트(update)		
C. 관련 문헌		
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
X A	KR 10-2020-0052361 A (후아웨이 테크놀로지 컴퍼니 리미티드) 2020.05.14 단락 [0006]-[0010], [0032]-[0099], [0113]; 및 도면 2, 5-6, 14	1-3,6-11,14-15 4-5,12-13
A	KR 10-2017-0065085 A (삼성전자주식회사) 2017.06.13 단락 [0192], [0223]; 및 도면 30	1-15
A	KR 10-1987664 B1 (나무기술 주식회사 등) 2019.06.11 단락 [0072]-[0082], [0126]; 및 도면 15	1-15
A	KR 10-2012-0026046 A (마이크로소프트 코퍼레이션) 2012.03.16 단락 [0016], [0037]; 및 도면 1	1-15
A	US 2018-0332107 A1 (AMAZON TECHNOLOGIES, INC.) 2018.11.15 단락 [0026], [0060]; 및 도면 5	1-15
<input type="checkbox"/> 추가 문헌이 C(계속)에 기재되어 있습니다. <input checked="" type="checkbox"/> 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 2024년09월09일(09.09.2024)		국제조사보고서 발송일 2024년09월09일(09.09.2024)
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

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-2020-0052361 A	2020/05/14	CN 111527478 A	2020/08/11
		EP 3673375 A1	2020/07/01
		JP 2020-537215 A	2020/12/17
		JP 7120708 B2	2022/08/17
		KR 10-2339239 B1	2021/12/14
		US 11321210 B2	2022/05/03
		US 2020-0371892 A1	2020/11/26
		WO 2019-075399 A1	2019/04/18
KR 10-2017-0065085 A	2017/06/13	KR 10-2502569 B1	2023/02/23
		US 10474574 B2	2019/11/12
		US 2018-0357166 A1	2018/12/13
		WO 2017-095195 A1	2017/06/08
KR 10-1987664 B1	2019/06/11	CN 112437915 A	2021/03/02
		JP 2021-530802 A	2021/11/11
		US 2021-0279157 A1	2021/09/09
		WO 2020-017844 A1	2020/01/23
KR 10-2012-0026046 A	2012/03/16	CN 102414674 A	2012/04/11
		EP 2425349 A2	2012/03/07
		JP 2012-525641 A	2012/10/22
		JP 5608222 B2	2014/10/15
		KR 10-1600129 B1	2016/03/04
		US 2010-0281482 A1	2010/11/04
		US 8261266 B2	2012/09/04
		WO 2010-126805 A2	2010/11/04
US 2018-0332107 A1	2018/11/15	CN 104854563 A	2015/08/19
		EP 2898410 A1	2015/07/29
		JP 2015-532992 A	2015/11/16
		JP 6049887 B2	2016/12/21
		US 10015241 B2	2018/07/03
		US 10542079 B2	2020/01/21
		US 2014-0082165 A1	2014/03/20
		US 2014-0082614 A1	2014/03/20
		US 2016-0241637 A1	2016/08/18
		US 9135048 B2	2015/09/15
		US 9323577 B2	2016/04/26
WO 2014-047073 A1	2014/03/27		