



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0142074
(43) 공개일자 2024년09월30일

(51) 국제특허분류(Int. Cl.)
G06F 3/06 (2006.01) G06F 12/1009 (2016.01)
(52) CPC특허분류
G06F 3/0626 (2013.01)
G06F 12/1009 (2013.01)
(21) 출원번호 10-2023-0036585
(22) 출원일자 2023년03월21일
심사청구일자 2023년03월21일

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
신용삼
경기도 화성시 동탄순환대로19길 59 (목동, 힐스테이트 동탄) 2033동 903호
백웅기
울산광역시 울주군 언양읍 유니스트길 50
(뒷면에 계속)
(74) 대리인
특허법인 무한

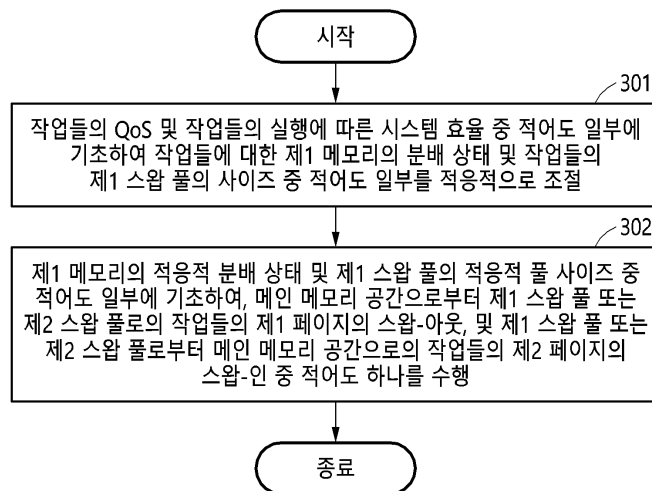
전체 청구항 수 : 총 25 항

(54) 발명의 명칭 메모리 관리 방법 및 그 방법을 이용하는 장치

(57) 요약

메모리 관리 방법 및 그 방법을 이용하는 장치가 제공된다. 그 방법은 작업들의 QoS 및 작업들의 실행에 따른 시스템 효율 중 적어도 일부에 기초하여 작업들에 대한 제1 메모리의 분배 상태 및 작업들의 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절하고, 제1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀 사이즈 중 적어도 일부에 기초하여, 메인 메모리 공간으로부터 제1 스왑 풀 또는 제2 스왑 풀로의 작업들의 제1 페이지의 스왑-아웃, 및 제1 스왑 풀 또는 제2 스왑 풀로부터 메인 메모리 공간으로의 작업들의 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 단계들을 포함할 수 있다.

대표도 - 도3



(52) CPC특허분류

G06F 3/0604 (2013.01)

G06F 3/0614 (2013.01)

G06F 3/0647 (2013.01)

G06F 3/0658 (2013.01)

G06F 3/0659 (2013.01)

G06F 2212/1016 (2013.01)

G06F 2212/1032 (2013.01)

G06F 2212/1044 (2013.01)

G06F 2212/401 (2013.01)

(72) 발명자

한명균

울산광역시 울주군 언양읍 유니스트길 50

오덕재

인천광역시 서구

박은성

울산광역시 울주군 언양읍 유니스트길 50

명세서

청구범위

청구항 1

메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리, 및 제2 스왑 풀을 포함하는 제2 메모리를 이용한 메모리 관리 방법에 있어서,

작업들의 QoS(quality of service) 및 상기 작업들의 실행에 따른 시스템 효율 중 적어도 일부에 기초하여 상기 작업들에 대한 상기 제1 메모리의 분배 상태 및 상기 작업들의 상기 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절하는 단계; 및

상기 제1 메모리의 적응적 분배 상태 및 상기 제1 스왑 풀의 적응적 풀 사이즈 중 적어도 일부에 기초하여, 상기 메인 메모리 공간으로부터 상기 제1 스왑 풀 또는 상기 제2 스왑 풀로의 상기 작업들의 제1 페이지의 스왑-아웃, 및 상기 제1 스왑 풀 또는 상기 제2 스왑 풀로부터 상기 메인 메모리 공간으로의 상기 작업들의 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 단계

를 포함하는 메모리 관리 방법.

청구항 2

제1항에 있어서,

상기 제1 페이지의 스왑-아웃 및 상기 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 단계는

상기 제1 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제1 스왑 풀 및 상기 제2 스왑 풀로부터 상기 제1 페이지의 스왑-아웃에 따른 상기 제1 페이지의 저장 위치를 선택하는 단계; 및

상기 제1 페이지의 상기 선택된 저장 위치에 기초하여 상기 제1 페이지의 스왑-아웃을 수행하는 단계

를 포함하는, 메모리 관리 방법.

청구항 3

제2항에 있어서,

상기 제1 페이지의 저장 위치를 선택하는 단계는

상기 제1 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제1 페이지의 제1 점수를 결정하는 단계; 및

상기 제1 페이지의 상기 제1 점수와 제1 임계치 간의 비교 결과에 따라 상기 제1 페이지의 저장 위치를 선택하는 단계

를 포함하고,

상기 제1 점수는

상기 제1 페이지의 핫니스(hotness)가 높을수록, 상기 제1 페이지의 압축률이 높을수록, 상기 제1 스왑 풀의 성능이 높을수록, 상기 제1 스왑 풀의 가용 공간이 클수록, 높게 결정되는,

메모리 관리 방법.

청구항 4

제1항에 있어서,

상기 제1 스왑 풀은 압축된 상태의 페이지들을 저장하는,

메모리 관리 방법.

청구항 5

제1항에 있어서,

상기 제1 페이지의 스왑-아웃 및 상기 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 단계는

상기 제2 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제2 페이지를 상기 메인 메모리 공간과 상기 제1 스왑 풀 또는 상기 제2 스왑 풀에 중복(duplicate)으로 저장할지에 관한 상기 제2 페이지의 스왑-인 유형을 결정하는 단계; 및

상기 제2 페이지의 상기 결정된 스왑-인 유형에 기초하여 상기 제2 페이지의 스왑-인을 수행하는 단계를 포함하는 메모리 관리 방법.

청구항 6

제5항에 있어서,

상기 제2 페이지의 스왑-인 유형을 결정하는 단계는

상기 제2 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제2 페이지의 제2 점수를 결정하는 단계; 및

상기 제2 페이지의 상기 제2 점수와 제2 임계치 간의 비교 결과에 따라 상기 제2 페이지의 스왑-인 유형을 결정하는 단계

를 포함하고,

상기 제2 점수는

상기 제2 페이지의 핫니스(hotness)가 높을수록, 상기 제2 페이지의 압축률이 높을수록, 상기 제2 페이지가 저장된 스왑 풀의 성능이 낮을수록, 상기 제2 페이지가 저장된 상기 스왑 풀의 가용 공간이 클수록, 높게 결정되는,

메모리 관리 방법.

청구항 7

제5항에 있어서,

상기 제2 페이지의 스왑-인 유형을 결정하는 단계는

상기 제2 페이지의 스왑-인이 상기 제2 페이지에 대한 쓰기 연산에 따른 상기 메인 메모리 공간 상의 메모리 폴트에 의해 발생한 경우, 상기 제2 페이지의 스왑-인 유형을 상기 메인 메모리 공간에 대한 단독 저장 유형으로 결정하는 단계를 포함하는,

메모리 관리 방법.

청구항 8

제1항에 있어서,

상기 제1 메모리의 상기 분배 상태 및 상기 제1 스왑 풀의 상기 사이즈 중 적어도 일부를 적응적으로 조절하는 단계는

적응 모드(adaptation mode)에서 상기 제1 메모리의 상기 분배 상태 및 상기 제1 스왑 풀의 상기 사이즈의 조절에 따른 상기 QoS 및 상기 시스템 효율 중 적어도 하나의 변화에 기초하여 상기 제1 메모리의 상기 적응적 분배 상태 및 상기 제1 스왑 풀의 상기 적응적 풀 사이즈를 결정하는 단계;

아이들 모드(idle mode)에서 상기 적응적 분배 상태 및 상기 적응적 풀 사이즈로 상기 제1 메모리 및 상기 제1 스왑 풀을 운영하면서 상기 QoS 및 상기 시스템 효율 중 적어도 일부의 변화를 모니터링하는 단계; 및

상기 QoS 및 상기 시스템 효율 중 적어도 일부의 변화에 기초하여 상기 적응 모드를 다시 수행하는 단계

를 포함하는, 메모리 관리 방법.

청구항 9

제8항에 있어서,

상기 작업들은

요구 QoS 레벨을 갖는 유형의 제1 작업 및 상기 요구 QoS 레벨을 갖지 않는 유형의 제2 작업을 포함하고,

상기 제1 메모리의 상기 적응적 분배 상태 및 상기 제1 스왑 풀의 상기 적응적 풀 사이즈를 결정하는 단계는

상기 제1 작업의 제1 QoS가 상기 제1 작업의 제1 요구 QoS 레벨을 만족하는 경우, 상기 제1 작업을 위한 상기 제1 메모리 내 제1 부분 공간의 사이즈를 줄이면서 상기 제1 작업의 상기 제1 부분 공간의 제1 적응적 메모리 사이즈를 탐색하는 단계; 및

상기 제1 적응적 메모리 사이즈의 조절에 따라 상기 제1 QoS가 상기 제1 요구 QoS 레벨을 만족하지 않는 경우, 상기 제1 작업을 위한 상기 제1 스왑 풀 내 제1 부분 풀의 사이즈를 조절하면서 상기 제1 작업의 제1 적응적 풀 사이즈를 탐색하는 단계를 포함하는,

메모리 관리 방법.

청구항 10

제9항에 있어서,

상기 제1 메모리의 상기 적응적 분배 상태 및 상기 제1 스왑 풀의 상기 적응적 풀 사이즈를 결정하는 단계는

상기 제1 메모리의 사이즈와 상기 제1 부분 공간의 상기 제1 적응적 메모리 사이즈 간의 차이에 기초하여 상기 제2 작업을 위한 상기 제1 메모리 내 제2 부분 공간의 제2 적응적 메모리 사이즈를 결정하는 단계; 및

상기 제1 QoS가 상기 제1 요구 QoS 레벨을 만족하는 상태에서 상기 시스템 효율이 증가하도록 상기 제2 작업을 위한 상기 제1 스왑 풀 내 제2 부분 풀의 사이즈를 조절하면서 상기 제2 작업의 제2 적응적 풀 사이즈를 탐색하는 단계

를 포함하는, 메모리 관리 방법.

청구항 11

제1항에 있어서,

상기 제1 메모리의 상기 분배 상태 및 상기 제1 스왑 풀의 상기 사이즈 중 적어도 일부를 적응적으로 조절하는 단계는

상기 제1 스왑 풀의 상기 사이즈의 감소가 요구되는 경우, 상기 제1 스왑 풀에 저장된 페이지들로부터 빅탐 페이지(victim page)를 선택하는 단계; 및

상기 빅탐 페이지에 대한 축출(eviction)을 수행하는 단계

를 포함하는, 메모리 관리 방법.

청구항 12

제11항에 있어서,

상기 빅탐 페이지에 대한 축출을 수행하는 단계는

상기 빅탐 페이지가 중복 페이지인 경우 상기 빅탐 페이지에 대한 비유효화를 수행하는 단계; 및

상기 빅탐 페이지가 중복 페이지가 아닌 경우 상기 빅탐 페이지를 상기 제2 스왑 풀에 저장하는 단계

를 포함하는, 메모리 관리 방법.

청구항 13

하드웨어와 결합되어 제1항 내지 제12항 중 어느 하나의 항의 방법을 실행시키기 위하여 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

청구항 14

메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리;

제2 스왑 풀을 포함하는 제2 메모리; 및

작업들의 QoS(quality of service) 및 상기 작업들의 실행에 따른 시스템 효율 중 적어도 일부에 기초하여 상기 작업들에 대한 상기 제1 메모리의 분배 상태 및 상기 작업들의 상기 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절하고,

상기 제1 메모리의 적응적 분배 상태 및 상기 제1 스왑 풀의 적응적 풀 사이즈 중 적어도 일부에 기초하여, 상기 메인 메모리 공간으로부터 상기 제1 스왑 풀 또는 상기 제2 스왑 풀로의 상기 작업들의 제1 페이지의 스왑-아웃, 및 상기 제1 스왑 풀 또는 상기 제2 스왑 풀로부터 상기 메인 메모리 공간으로의 상기 작업들의 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 프로세서

를 포함하는, 전자 장치.

청구항 15

제14항에 있어서,

상기 프로세서는, 상기 제1 페이지의 스왑-아웃 및 상기 제2 페이지의 스왑-인 중 적어도 하나를 수행하기 위해,

상기 제1 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제1 스왑 풀 및 상기 제2 스왑 풀로부터 상기 제1 페이지의 스왑-아웃에 따른 상기 제1 페이지의 저장 위치를 선택하고,

상기 제1 페이지의 상기 선택된 저장 위치에 기초하여 상기 제1 페이지의 스왑-아웃을 수행하는,

전자 장치.

청구항 16

제14항에 있어서,

상기 제1 스왑 풀은 압축된 상태의 페이지들을 저장하는,

전자 장치.

청구항 17

제14항에 있어서,

상기 프로세서는, 상기 제1 페이지의 스왑-아웃 및 상기 제2 페이지의 스왑-인 중 적어도 하나를 수행하기 위해,

상기 제2 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제2 페이지를 상기 메인 메모리 공간과 상기 제1 스왑 풀 또는 상기 제2 스왑 풀에 중복(duplicate)으로 저장할지에 관한 상기 제2 페이지의 스왑-인 유형을 결정하고,

상기 제2 페이지의 상기 결정된 스왑-인 유형에 기초하여 상기 제2 페이지의 스왑-인을 수행하는,

전자 장치.

청구항 18

제14항에 있어서,

상기 제1 메모리의 상기 분배 상태 및 상기 제1 스왑 풀의 상기 사이즈 중 적어도 일부를 적응적으로 조절하기 위해,

적응 모드(adaptation mode)에서 상기 제1 메모리의 상기 분배 상태 및 상기 제1 스왑 풀의 상기 사이즈의 조절에 따른 상기 QoS 및 상기 시스템 효율 중 적어도 하나의 변화에 기초하여 상기 제1 메모리의 상기 적응적 분배 상태 및 상기 제1 스왑 풀의 상기 적응적 풀 사이즈를 결정하고,

아이들 모드(idle mode)에서 상기 적응적 분배 상태 및 상기 적응적 풀 사이즈로 상기 제1 메모리 및 상기 제1 스왑 풀을 운영하면서 상기 QoS 및 상기 시스템 효율 중 적어도 일부의 변화를 모니터링하고,

상기 QoS 및 상기 시스템 효율 중 적어도 일부의 변화에 기초하여 상기 적응 모드를 다시 수행하는,

전자 장치.

청구항 19

제18항에 있어서,

상기 작업들은

요구 QoS 레벨을 갖는 유형의 제1 작업 및 상기 요구 QoS 레벨을 갖지 않는 유형의 제2 작업을 포함하고,

상기 프로세서는, 상기 제1 메모리의 상기 적응적 분배 상태 및 상기 제1 스왑 풀의 상기 적응적 풀 사이즈를 결정하기 위해,

상기 제1 작업의 제1 QoS가 상기 제1 작업의 제1 요구 QoS 레벨을 만족하는 경우, 상기 제1 작업을 위한 상기 제1 메모리 내 제1 부분 공간의 사이즈를 줄이면서 상기 제1 작업의 상기 제1 부분 공간의 제1 적응적 메모리 사이즈를 탐색하고,

상기 제1 적응적 메모리 사이즈의 조절에 따라 상기 제1 QoS가 상기 제1 요구 QoS 레벨을 만족하지 않는 경우, 상기 제1 작업을 위한 상기 제1 스왑 풀 내 제1 부분 풀의 사이즈를 조절하면서 상기 제1 작업의 제1 적응적 풀 사이즈를 탐색하고,

상기 제1 메모리의 사이즈와 상기 제1 부분 공간의 상기 제1 적응적 메모리 사이즈 간의 차이에 기초하여 상기 제2 작업을 위한 상기 제1 메모리 내 제2 부분 공간의 제2 적응적 메모리 사이즈를 결정하고,

상기 제1 QoS가 상기 제1 요구 QoS 레벨을 만족하는 상태에서 상기 시스템 효율이 증가하도록 상기 제2 작업을 위한 상기 제1 스왑 풀 내 제2 부분 풀의 사이즈를 조절하면서 상기 제2 작업의 제2 적응적 풀 사이즈를 탐색하는,

전자 장치.

청구항 20

제14항에 있어서,

상기 프로세서는, 상기 제1 메모리의 상기 분배 상태 및 상기 제1 스왑 풀의 상기 사이즈 중 적어도 일부를 적응적으로 조절하기 위해,

상기 제1 스왑 풀의 상기 사이즈의 감소가 요구되는 경우, 상기 제1 스왑 풀에 저장된 페이지들로부터 빅티م 페이지(victim page)를 선택하고,

상기 빅티م 페이지에 대한 축출(eviction)을 수행하는,

전자 장치.

청구항 21

메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리;

제2 스왑 풀을 포함하는 제2 메모리; 및

제1 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제1 스왑 풀 및 상기 제2 스왑 풀로부터 상기 제1 페이지의 스왑-아웃에 따른 상기 제1 페이지의 저장 위치를 선택하고,

상기 제1 페이지의 상기 선택된 저장 위치에 기초하여 상기 제1 페이지의 스왑-아웃을 수행하는, 프로세서

를 포함하는, 전자 장치.

청구항 22

제21항에 있어서,

상기 프로세서는, 상기 제1 페이지의 저장 위치를 선택하기 위해,

상기 제1 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제1 페이지의 제1 점수를 결정하고,

상기 제1 페이지의 상기 제1 점수와 제1 임계치 간의 비교 결과에 따라 상기 제1 페이지의 저장 위치를 선택하고,

상기 제1 점수는

상기 제1 페이지의 핫니스(hotness)가 높을수록, 상기 제1 페이지의 압축률이 높을수록, 상기 제1 스왑 풀의 성능이 높을수록, 상기 제1 스왑 풀의 가용 공간이 클수록, 높게 결정되는,

전자 장치.

청구항 23

메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리;

제2 스왑 풀을 포함하는 제2 메모리; 및

제2 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제2 페이지를 상기 메인 메모리 공간과 상기 제1 스왑 풀 또는 상기 제2 스왑 풀에 중복(duplicate)으로 저장할지에 관한 상기 제2 페이지의 스왑-인 유형을 결정하고,

상기 제2 페이지의 상기 결정된 스왑-인 유형에 기초하여 상기 제2 페이지의 스왑-인을 수행하는, 프로세서를 포함하는, 전자 장치.

청구항 24

제23항에 있어서,

상기 프로세서는, 상기 제2 페이지의 스왑-인 유형을 결정하기 위해,

상기 제2 페이지, 상기 제1 스왑 풀, 및 상기 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 상기 제2 페이지의 제2 점수를 결정하고,

상기 제2 페이지의 상기 제2 점수와 제2 임계치 간의 비교 결과에 따라 상기 제2 페이지의 스왑-인 유형을 결정하고,

상기 제2 점수는

상기 제2 페이지의 핫니스(hotness)가 높을수록, 상기 제2 페이지의 압축률이 높을수록, 상기 제2 페이지가 저장된 스왑 풀의 성능이 낮을수록, 상기 제2 페이지가 저장된 상기 스왑 풀의 가용 공간이 클수록, 높게 결정되는,

전자 장치.

청구항 25

제23항에 있어서,

상기 프로세서는, 상기 제2 페이지의 스왑-인 유형을 결정하기 위해,

상기 제2 페이지의 스왑-인이 상기 제2 페이지에 대한 쓰기 연산에 따른 상기 메인 메모리 공간 상의 메모리 폴트에 의해 발생한 경우, 상기 제2 페이지의 스왑-인 유형을 상기 메인 메모리 공간에 대한 단독 저장 유형으로 결정하는,

전자 장치.

발명의 설명

기술 분야

[0001] 아래 실시예들은 메모리 관리 방법 및 그 방법을 이용하는 장치에 관한 것이다.

배경 기술

[0002] 메모리 스와핑(memory swapping)은 컴퓨터 운영 체제에서 주 기억장치(예: RAM(random access memory))의 물리적 메모리보다 더 큰 가상 메모리를 제공할 수 있다. 시스템에서 RAM의 물리적 메모리가 부족할 때, RAM의 일부 데이터가 스왑 데이터로 보조 기억장치(예: SDD(solid state drive))로 임시로 이동할 수 있다. 이로써 시스템은 계속 실행될 수 있다. 메모리 스와핑을 통해 물리적 메모리의 확장 효과가 나타날 수 있고, 시스템은 스왑 공간을 가상 메모리처럼 사용할 수 있다. 스왑 데이터가 저장된 공간에 접근하는 것은 RAM에 저장된 데이터에 접근하는 것보다 느릴 수 있다. 메모리 스와핑은 운영 체제의 구성 요소로서 메모리 리소스를 효율적으로 관리하는데 사용될 수 있고, 성능과 메모리 사용 사이의 절충점을 제공할 수 있다.

발명의 내용

과제의 해결 수단

[0003] 일 실시예에 따르면, 메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리, 및 제2 스왑 풀을 포함하는 제2 메모리를 이용한 메모리 관리 방법은 작업들의 QoS(quality of service) 및 작업들의 실행에 따른 시스템 효율 중 적어도 일부에 기초하여 작업들에 대한 제1 메모리의 분배 상태 및 작업들의 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절하는 단계, 및 제1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀 사이즈 중 적어도 일부에 기초하여, 메인 메모리 공간으로부터 제1 스왑 풀 또는 제2 스왑 풀로의 작업들의 제1 페이지의 스왑-아웃, 및 제1 스왑 풀 또는 제2 스왑 풀로부터 메인 메모리 공간으로의 작업들의 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 단계를 포함한다.

[0004] 일 실시예에 따르면, 전자 장치는 메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리, 제2 스왑 풀을 포함하는 제2 메모리, 및 작업들의 QoS(quality of service) 및 작업들의 실행에 따른 시스템 효율 중 적어도 일부에 기초하여 작업들에 대한 제1 메모리의 분배 상태 및 작업들의 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절하고, 제1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀 사이즈 중 적어도 일부에 기초하여, 메인 메모리 공간으로부터 제1 스왑 풀 또는 제2 스왑 풀로의 작업들의 제1 페이지의 스왑-아웃, 및 제1 스왑 풀 또는 제2 스왑 풀로부터 메인 메모리 공간으로의 작업들의 제2 페이지의 스왑-인 중 적어도 하나를 수행하는 프로세서를 포함한다.

[0005] 일 실시예에 따르면, 전자 장치는 메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리, 제2 스왑 풀을 포함하는 제2 메모리, 및 제1 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제1 스왑 풀 및 제2 스왑 풀로부터 제1 페이지의 스왑-아웃에 따른 제1 페이지의 저장 위치를 선택하고, 제1 페이지의 선택된 저장 위치에 기초하여 제1 페이지의 스왑-아웃을 수행하는, 프로세서를 포함한다.

[0006] 일 실시예에 따르면, 전자 장치는 메인 메모리 공간 및 제1 스왑 풀을 포함하는 제1 메모리, 제2 스왑 풀을 포함하는 제2 메모리, 및 제2 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제2 페이지를 메인 메모리 공간과 제1 스왑 풀 또는 제2 스왑 풀에 중복(duplicate)으로 저장할지에 관한 제2 페이지의 스왑-인 유형을 결정하고, 제2 페이지의 결정된 스왑-인 유형에 기초하여 제2 페이지의 스왑-인을 수행하는, 프로세서를 포함한다.

도면의 간단한 설명

[0007] 도 1a는 일 실시예에 따른, 메모리 스와핑에 따른 페이지의 이동 경로를 예시적으로 나타내는 도면이다.

도 1b는 일 실시예에 따른, 각 작업을 위한 제1 메모리의 서브 영역들을 예시적으로 나타내는 도면이다.

도 2는 일 실시예에 따른, 메모리 스와핑을 수행하는 구성요소들을 예시적으로 나타내는 도면이다.

도 3은 일 실시예에 따른, 스왑 풀 사이즈의 조절, 스왑-아웃, 및 스왑-인을 포함하는 메모리 스와핑의 동작들

을 예시적으로 나타내는 플로우 차트이다.

도 4는 일 실시예에 따른, 스왑-아웃 동작을 예시적으로 나타내는 플로우 차트이다.

도 5는 일 실시예에 따른, 스왑-인 동작을 예시적으로 나타내는 플로우 차트이다.

도 6 내지 도 8은 일 실시예에 따른, 제1 메모리의 분배 상태 및 제1 스왑 풀의 사이즈의 적응적 조절을 위한 동작들을 예시적으로 나타내는 플로우 차트이다.

도 9는 일 실시예에 따른, 전자 장치의 구성요소들을 예시적으로 나타내는 블록도이다.

도 10은 일 실시예에 따른, 스왑-아웃 동작을 예시적으로 나타내는 플로우 차트이다.

도 10은 일 실시예에 따른, 스왑-인 동작을 예시적으로 나타내는 플로우 차트이다.

발명을 실시하기 위한 구체적인 내용

- [0008] 실시예들에 대한 특정한 구조적 또는 기능적 설명들은 단지 예시를 위한 목적으로 개시된 것으로서, 다양한 형태로 변경되어 구현될 수 있다. 따라서, 실제 구현되는 형태는 개시된 특정 실시예로만 한정되는 것이 아니며, 본 명세서의 범위는 실시예들로 설명한 기술적 사상에 포함되는 변경, 균등물, 또는 대체물을 포함한다.
- [0009] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 해석되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.
- [0010] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다.
- [0011] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 설명된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로써 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0012] 본 명세서에서, "A 또는 B 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.
- [0013] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 해당 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0014] 이하, 실시예들을 첨부된 도면들을 참조하여 상세하게 설명한다. 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고, 이에 대한 중복되는 설명은 생략하기로 한다.
- [0015] 도 1a는 일 실시예에 따른, 메모리 스와핑에 따른 페이지의 이동 경로를 예시적으로 나타내는 도면이다. 최근 기계 학습, 빅 데이터, 인-메모리 키-밸류 스토어(in-memory key-value store) 등 메모리 집약형 워크로드(workload)의 증가에 따라 메모리 요구량이 급격하게 증가하고 있다. 이러한 메모리 요구량은 메모리 스와핑을 통해 비용 효율적인 방식으로 충족될 수 있다. 메모리 스와핑(memory swapping), 간단히 스왑을 통해 메모리(예: 제1 메모리(110))의 페이지가 스왑 풀(swap pool)(예: 제1 스왑 풀(112) 또는 제2 스왑 풀(121))로 이동(예: 스왑-아웃(swap-out))하여 메모리에 상주하는 데이터가 압축 메모리(예: 제1 스왑 풀(112))나 디스크(예: 제2 스왑 풀(121)) 등에 저장될 수 있다. 압축 메모리(compressed memory) 또는 메모리 압축(memory compression)이라는 용어는 물리적인 메모리 장치 자체가 아닌 물리적인 메모리 장치 내 데이터(예: 페이지)를 압축하는 기술을 의미할 수 있다. 스왑을 통해 대용량 워킹-세트(working-set)를 갖는 워크로드의 효율적 실행이 지원될 수 있다.
- [0016] 메모리 스와핑을 위한 스왑 풀은 디스크 스왑, 압축 메모리 스왑(compressed memory swap, CMS) 등의 스왑 백엔드(swap backend)를 포함할 수 있다. 디스크 스왑은 HDD(hard disk drive), SSD(solid state drive) 등의 디스크 장치를 스왑 공간으로 사용하는 스왑 백엔드에 해당할 수 있다. CMS는 압축된 메모리를 스왑 공간으로 활

용하는 스왑 백엔드에 해당할 수 있다. CMS의 이용을 위해 페이지가 압축되어 CMS의 메모리 공간에 저장될 수 있다. CMS를 통해 디스크 I/O(input/output)가 감소되고 성능 향상이 이루어질 수 있다.

[0017] 도 1a를 참조하면, 제1 메모리(110)는 메인 메모리 공간(111) 및 제1 스왑 풀(112)을 포함할 수 있고, 제2 메모리(120)는 제2 스왑 풀(121)을 포함할 수 있다. 메인 메모리 공간(111) 및 제1 스왑 풀(112)은 각각 제1 메모리(110)의 물리적 영역에 형성될 수 있고, 제2 스왑 풀(121)은 제2 메모리(120)의 물리적 영역에 형성될 수 있다. 제1 스왑 풀(112)은 CMS 백엔드에 해당할 수 있고, 제2 스왑 풀(121)은 디스크 스왑 백엔드에 해당할 수 있다.

[0018] 제2 메모리(120)는 제1 메모리(110)에 비해 프로세서로부터 멀게 위치할 수 있다. 프로세서가 제1 메모리(110)의 페이지에 접근하는게 제2 메모리(120)의 페이지에 접근하는 것에 비해 시간 및/또는 비용이 더 적게 소요될 수 있다. 예를 들어, 제1 메모리(110)는 주 기억장치(예: RAM(random access memory))에 해당할 수 있고, 제2 메모리(120)는 보조 기억장치(예: HDD(hard disk drive), SSD(solid state drive)와 같은 디스크(disk))에 해당할 수 있다.

[0019] 제1 스왑 풀(112)은 압축된 페이지를 저장할 수 있다. 페이지가 메인 메모리 공간(111)으로부터 제1 스왑 풀(112)로 스왑-아웃되기 위해서는 페이지의 압축이 요구될 수 있고, 압축된 페이지가 제1 스왑 풀(112)로부터 메인 메모리 공간(111)으로 스왑-인되기 위해서는 압축된 페이지의 압축 해제가 요구될 수 있다. 제2 스왑 풀(121)은 비압축 상태의 페이지를 저장할 수 있다. 메인 메모리 공간(111)의 페이지는 제2 스왑 풀(121)로 스왑-아웃될 수 있고, 제2 스왑 풀(121)의 페이지는 메인 메모리 공간(111)으로 스왑-인될 수 있다.

[0020] 스왑 기술을 통해 대용량 워킹-세트를 갖는 워크로드가 효율적으로 실행될 수 있다. 실시예들에 따르면, 스왑 풀(예: 제1 스왑 풀(112))의 사이즈가 적응적으로 조절될 수 있다. 또한, 실시예들에 따르면, 스왑-아웃 및/또는 스왑-인이 이루어질 때, 페이지의 특성 및/또는 스왑 풀(예: 제1 스왑 풀(112), 제2 스왑 풀(121))의 특성이 고려될 수 있다. 실시예들에 따르면, 이러한 스왑 풀의 사이즈 조절이나 특성 기반의 스왑-인, 스왑-아웃을 통해 워크로드의 QoS(quality of service) 보장 및 시스템 효율의 극대화가 이루어질 수 있다.

[0021] 도 1b는 일 실시예에 따른, 각 작업을 위한 제1 메모리의 서브 영역들을 예시적으로 나타내는 도면이다. 도 1b를 참조하면, 제1 작업(151) 및 제2 작업(152)과 같은 복수의 작업들이 실행되는 경우, 제1 메모리(110)는 제1 부분 공간(110a) 및 제2 부분 공간(110b)과 같은 복수의 부분 공간들로 분할되어 복수의 작업들에 할당될 수 있다. 제1 작업(151)은 요구 QoS 레벨을 갖는 제1 유형의 작업에 해당할 수 있고, 제2 작업(152)은 요구 QoS 레벨을 갖지 않는 제2 유형의 작업에 해당할 수 있다. 요구 QoS 레벨은 해당 워크로드 및/또는 해당 작업의 정상적인 작동을 위해 요구되는 최소한의 QoS 레벨을 의미할 수 있다. 따라서, 요구 QoS 레벨을 갖는 작업의 경우 해당 요구 QoS 레벨을 만족하는 성능이 제공될 필요가 있다. 예를 들어, 제1 작업(151)은 LC 작업(latency-critical task)에 해당할 수 있고, 제2 작업(152)은 배치 작업(batch task)에 해당할 수 있다. 제1 작업(151) 및 제2 작업은 동일한 워크로드를 위한 것이거나 혹은 다른 워크로드들을 위한 것일 수 있다.

[0022] 일 실시예에 따르면, 적응적 조절 동작(161)을 통해 제1 작업(151)을 위한 제1 메모리(110) 내 제1 부분 공간(110a)과 제2 작업(152)을 위한 제1 메모리(110) 내 제2 부분 공간(110b)의 분배 상태가 최적화될 수 있다. 제1 부분 공간(110a)과 제2 부분 공간(110b)의 분배 상태에 대한 적응적 조절 동작(161)은 제1 부분 공간(110a)의 사이즈 조절 및 제2 부분 공간(110b)의 사이즈 조절을 포함할 수 있다. 제1 부분 공간(110a)의 사이즈는 제1 작업(151)의 QoS가 요구 QoS 레벨을 만족하는 조건 하에서 최소화될 수 있다. 제1 메모리(110)에서 제1 부분 공간(110a)을 제외한 나머지 부분은 제2 부분 공간(110b)으로 할당될 수 있다.

[0023] 일 실시예에 따르면, 적응적 조절 동작들(162, 163)을 통해 제1 메모리(110) 내 제1 스왑 풀(112)의 사이즈가 최적화될 수 있다. 제1 스왑 풀(112)의 사이즈 조절은 제1 작업(151)을 위한 제1 스왑 풀(112) 내 제1 부분 풀(112a)의 적응적 조절 동작(162) 및 제2 작업(152)을 위한 제1 스왑 풀(112) 내 제2 부분 풀(112b)의 적응적 조절 동작(163)을 포함할 수 있다. 제1 부분 풀(112a)의 사이즈는 QoS가 증가하도록 조절될 수 있다. 제2 부분 풀(112b)의 사이즈는 제1 작업(151)의 QoS가 요구 QoS 레벨을 만족하면서 시스템 효율이 최대화되도록 조절될 수 있다. 적응적 조절 동작들(161, 162, 163)에 대해서는 추후 더 상세히 설명한다.

[0024] 도 2는 일 실시예에 따른, 메모리 스와핑을 수행하는 구성요소들을 예시적으로 나타내는 도면이다. 도 2를 참조하면, 전자 장치(200)는 프로세서(210) 및 메모리(220)를 포함할 수 있다. 전자 장치(200)는 프로세서(210) 및 메모리(220)를 이용하여 운영 체제(230)를 구동할 수 있고, 운영 체제(230)를 이용하여 메모리(220)를 관리할 수 있다. 도 2의 점선(201)을 기준으로 좌측은 전자 장치(200)의 하드웨어 구성에 해당할 수 있고, 우측은

전자 장치(200)의 소프트웨어 구성에 해당할 수 있다.

- [0025] 메모리(220)는 제1 메모리(221)(예: 도 1의 제1 메모리(110)) 및 제2 메모리(223)(예: 도 1의 제2 메모리(120))를 포함할 수 있다. 전자 장치(200)는 제1 메모리(221)의 제1 스왑 풀(예: 도 1의 제1 스왑 풀(112)) 및 제2 메모리(222)의 제2 스왑 풀(예: 도 1의 제2 스왑 풀(121))을 이용하여 메모리 스와핑을 수행할 수 있다. 전자 장치(200)는 메모리 스와핑을 이용하여 워크로드를 처리할 수 있다.
- [0026] 운영 체제(230)는 특성 모니터(characteristic monitor)(231), 런타임 시스템(runtime system)(232), 메모리 매니저(memory manager)(233), 및 스왑 백엔드(swap backend)(234)를 포함할 수 있다. 특성 모니터(231)는 메모리 스왑과 관련된 다양한 정보를 모니터링할 수 있다. 예를 들어, 특성 모니터(231)는 워크로드의 QoS, 시스템 효율, 페이지 접근 빈도 등의 모니터링 작업을 수행할 수 있다. 런타임 시스템(232)은 모니터링된 정보를 기반으로 CMS 사이즈를 동적으로 조절할 수 있다. 메모리 매니저(233)는 스왑-인 및/또는 스왑-아웃을 통해 메모리 관리를 수행할 수 있다. 스왑 백엔드(234)는 각 백엔드(예: CMS 백엔드, 디스크 스왑 백엔드)를 위한 원시 함수(primitive function)를 제공할 수 있다. 예를 들어, 원시 함수는 load, store, invalidate, shrink 등을 포함할 수 있다.
- [0027] 특성 모니터(231), 런타임 시스템(232), 메모리 매니저(233), 및 스왑 백엔드(234)는 실시예들의 메모리 관리를 수행하는 예시적인 구성요소들에 해당하며, 실시예들의 메모리 관리는 다른 구성요소들에 의해 수행될 수 있다. 또한, 실시예들에 따른 특성 모니터(231), 런타임 시스템(232), 메모리 매니저(233), 및 스왑 백엔드(234)의 각 동작은 운영 체제(230)의 동작, 프로세서(210)의 동작, 및 전자 장치(200)의 동작으로 간주될 수 있다. 또한, 실시예들에 따라 특성 모니터(231), 런타임 시스템(232), 메모리 매니저(233), 및 스왑 백엔드(234)의 어느 하나의 동작으로 설명되는 경우에도, 특성 모니터(231), 런타임 시스템(232), 메모리 매니저(233), 및 스왑 백엔드(234)의 다른 하나 이상에 의해 해당 동작이 수행될 수 있다.
- [0028] 도 3은 일 실시예에 따른, 스왑 풀 사이즈의 조절, 스왑-아웃, 및 스왑-인을 포함하는 메모리 스와핑의 동작들을 예시적으로 나타내는 플로우 차트이다. 도 3을 참조하면, 단계(301)에서 전자 장치는 작업들의 QoS 및 작업들의 실행에 따른 시스템 효율(예: 시스템 스루풋(throughput)) 중 적어도 일부에 기초하여 작업들에 대한 제1 메모리의 분배 상태 및 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절한다. 제1 스왑 풀은 CMS 풀에 해당할 수 있다. 전자 장치는 QoS 및 시스템 효율 이외에 더 많은 정보를 활용하여 제1 스왑 풀의 사이즈를 적응적으로 조절할 수 있다. 예를 들어, 전자 장치는 작업들의 QoS, 작업들의 실행에 따른 시스템 효율, 및 작업들의 실행에 따른 부하(load)에 기초하여 제1 스왑 풀의 사이즈를 적응적으로 조절할 수 있다. 제1 스왑 풀의 사이즈 조절은 제1 메모리가 지속적으로 동작하고 있는 도중에 수행될 수 있다. 예를 들어, 워크로드의 작업들을 실행하기 위해 제1 메모리의 읽기 동작 및/또는 쓰기 동작이 수행될 수 있다. 이러한 제1 메모리의 읽기 동작 및/또는 쓰기 동작이 지속적으로 이루어지는 동안 제1 스왑 풀의 사이즈 조절이 이루어질 수 있다.
- [0029] 제1 스왑 풀의 사이즈 조절을 위한 동작 모드는 적응 모드(adaptation mode) 및 아이들 모드(idle mode)를 포함할 수 있다. 전자 장치는 적응 모드와 아이들 모드를 교차로 수행하면서 제1 메모리의 분배 상태 및 제1 스왑 풀의 사이즈를 조절할 수 있다. 전자 장치는 적응 모드에서 제1 메모리의 분배 상태 및 제1 스왑 풀의 사이즈의 조절에 따른 QoS 및 시스템 효율 중 적어도 하나의 변화에 기초하여 제1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀(pool) 사이즈를 결정하고, 아이들 모드에서 적응적 분배 상태 및 적응적 풀 사이즈로 제1 메모리 및 제1 스왑 풀을 운영하면서 QoS 및 시스템 효율 중 적어도 일부의 변화를 모니터링하고, QoS 및 시스템 효율 중 적어도 일부의 변화에 기초하여 적응 모드를 다시 수행할 수 있다.
- [0030] 작업들은 요구 QoS 레벨을 갖는 유형의 제1 작업 및 요구 QoS 레벨을 갖지 않는 유형의 제2 작업을 포함할 수 있다. 전자 장치는, 제1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀 사이즈를 결정하기 위해, 제1 작업의 제1 QoS가 제1 작업의 제1 요구 QoS 레벨을 만족하는 경우, 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 사이즈를 줄이면서 제1 작업의 제1 부분 공간의 제1 적응적 메모리 사이즈를 탐색하고, 제1 적응적 메모리 사이즈의 조절에 따라 제1 QoS가 제1 요구 QoS 레벨을 만족하지 않는 경우, 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 사이즈를 조절하면서 제1 작업의 제1 적응적 풀 사이즈를 탐색할 수 있다. 전자 장치는, 제1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀 사이즈를 결정하기 위해, 제1 메모리의 사이즈와 제1 부분 공간의 제1 적응적 메모리 사이즈 간의 차이에 기초하여 제2 작업을 위한 제1 메모리 내 제2 부분 공간의 제2 적응적 메모리 사이즈를 결정하고, 제1 QoS가 제1 요구 QoS 레벨을 만족하는 상태에서 시스템 효율이 증가하도록 제2 작업을 위한 제1 스왑 풀 내 제2 부분 풀의 사이즈를 조절하면서 제2 작업의 제2 적응적 풀 사이즈를 탐색할 수 있다.

- [0031] 전자 장치는, 제1 메모리의 분배 상태 및 제1 스왑 풀의 사이즈 중 적어도 일부를 적응적으로 조절하기 위해, 제1 스왑 풀의 사이즈의 감소가 요구되는 경우, 제1 스왑 풀에 저장된 페이지들로부터 빅탐 페이지(victim page)를 선택하고, 빅탐 페이지에 대한 축출(eviction)을 수행할 수 있다. 제1 스왑 풀의 사이즈의 감소가 요구되는 경우, 메모리 효율 관점에서 가장 유리한 방향으로 빅탐 페이지가 선택될 수 있다. 전자 장치는, 빅탐 페이지에 대한 축출을 수행하기 위해, 빅탐 페이지가 중복 페이지(duplicated page)인 경우 빅탐 페이지에 대한 비유효화(invalidation)를 수행하고, 빅탐 페이지가 중복 페이지가 아닌 경우 빅탐 페이지를 제2 스왑 풀에 저장할 수 있다. 제2 스왑 풀은 디스크 스왑 풀에 해당할 수 있다.
- [0032] 빅탐 페이지가 중복 페이지라면 빅탐 페이지가 메인 메모리 공간 및 제1 스왑 풀에 모두 저장되어 있는 상황이므로, 빅탐 페이지에 대한 비유효화를 통해 제1 스왑 풀의 빅탐 페이지는 효력을 잃을 수 있고, 메인 메모리 공간의 빅탐 페이지만 효력을 가질 수 있다. 이 경우, 제1 스왑 풀의 사이즈 감소에 따라 제1 스왑 풀의 빅탐 페이지는 제거될 수 있고, 빅탐 페이지의 데이터는 메인 메모리 공간 상에 유지될 수 있다. 빅탐 페이지가 중복 페이지가 아니라면, 빅탐 페이지의 데이터 손실을 막기 위해 빅탐 페이지를 제2 스왑 풀로 옮긴 뒤 제1 스왑 풀의 사이즈 감소가 이루어질 수 있다.
- [0033] 단계(302)에서 전자 장치는 1 메모리의 적응적 분배 상태 및 제1 스왑 풀의 적응적 풀 사이즈 중 적어도 일부에 기초하여, 메인 메모리 공간으로부터 제1 스왑 풀 또는 제2 스왑 풀로의 작업들의 제1 페이지의 스왑-아웃, 및 제1 스왑 풀 또는 제2 스왑 풀로부터 메인 메모리 공간으로의 작업들의 제2 페이지의 스왑-인 중 적어도 하나를 수행한다. 전자 장치는 제1 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제1 스왑 풀 및 제2 스왑 풀로부터 제1 페이지의 스왑-아웃에 따른 제1 페이지의 저장 위치를 선택하고, 제1 페이지의 선택된 저장 위치에 기초하여 제1 페이지의 스왑-아웃을 수행할 수 있다. 제1 스왑 풀은 압축된 상태의 페이지들을 저장할 수 있다.
- [0034] 전자 장치는, 제1 페이지의 저장 위치를 선택하기 위해, 제1 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제1 페이지의 제1 점수를 결정하고, 제1 페이지의 제1 점수와 제1 임계치 간의 비교 결과에 따라 제1 페이지의 저장 위치를 선택할 수 있다. 제1 페이지의 핫니스(hotness), 제1 페이지의 압축률, 제1 스왑 풀의 성능, 제1 스왑 풀의 가용 공간 등이 저장 위치 선택에 고려될 수 있다. 핫니스는 페이지 접근 빈도를 나타낼 수 있다. 제1 스왑 풀의 성능은 압축 및 압축 해제를 위한 지연(latency)이나 대역폭(bandwidth)을 포함할 수 있다. 제1 페이지의 핫니스가 높을수록, 제1 페이지의 압축률이 높을수록, 제1 스왑 풀의 성능이 높을수록, 제1 스왑 풀의 가용 공간이 클수록, 제1 점수가 높게 결정될 수 있다.
- [0035] 전자 장치는 제2 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제2 페이지를 메인 메모리 공간과 스왑 풀(제1 스왑 풀 또는 제2 스왑 풀)에 중복(duplicate)으로 저장할지에 관한 제2 페이지의 스왑-인 유형을 결정하고, 제2 페이지의 결정된 스왑-인 유형에 기초하여 제2 페이지의 스왑-인을 수행할 수 있다. 스왑-인 유형은 중복 저장 유형 및 단독 저장 유형을 포함할 수 있다. 중복 저장 유형에 따르면 제2 페이지는 메인 메모리 공간 및 스왑 풀에 중복으로 저장될 수 있고, 단독 저장 유형에 따르면 제2 페이지는 메인 메모리 공간에 단독으로 저장될 수 있다. 단독 저장 유형에 따르면 제2 스왑 풀의 제2 페이지는 비유효화 처리될 수 있다.
- [0036] 전자 장치는, 제2 페이지의 스왑-인 유형을 결정하기 위해, 제2 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제2 페이지의 제2 점수를 결정하고, 제2 페이지의 제2 점수와 제2 임계치 간의 비교 결과에 따라 제2 페이지의 스왑-인 유형을 결정할 수 있다. 제2 페이지의 핫니스, 제2 페이지의 압축률, 제2 페이지가 저장된 스왑 풀(제1 스왑 풀 또는 제2 스왑 풀)의 성능, 제2 페이지가 저장된 스왑 풀의 가용 공간 등이 스왑-인 유형의 결정에 고려될 수 있다. 제2 페이지의 핫니스가 높을수록, 제2 페이지의 압축률이 높을수록, 제2 페이지가 저장된 스왑 풀의 성능이 낮을수록, 제2 페이지가 저장된 스왑 풀의 가용 공간이 클수록, 제2 점수가 높게 결정될 수 있다.
- [0037] 전자 장치는, 제2 페이지의 스왑-인이 제2 페이지에 대한 쓰기 연산에 따른 메인 메모리 공간 상의 메모리 폴트에 의해 발생한 경우, 제2 페이지의 스왑-인 유형을 메인 메모리 공간에 대한 단독 저장 유형으로 결정할 수 있다. 중복 저장 유형에 따라 제2 페이지가 메인 메모리 공간과 스왑 풀(제1 스왑 풀 또는 제2 스왑 풀)에 중복으로 저장된 경우, 제2 페이지의 스왑-아웃이 필요하더라도 이미 스왑 풀에 제2 페이지가 저장되어 있으므로 제2 페이지를 메인 메모리로부터 스왑 풀로 스왑-아웃하는 작업이 생략될 수 있다. 제2 페이지를 메인 메모리 공간으로부터 스왑 풀로 스왑-아웃하는 작업 대신 메인 메모리 공간의 제2 페이지를 비유효화하는 동작이 수행될 수 있다. 스왑-아웃 작업의 생략은 제2 페이지의 데이터가 보존되는 경우에 제한적으로 적용될 수 있다.

[0038] 메인 메모리 공간에서 제2 페이지의 데이터가 보존되지 않고 변경된 경우 변경된 제2 페이지를 메인 메모리 공간으로부터 스왑 풀로 스왑-아웃하는 동작이 요구될 수 있다. 이러한 제2 페이지의 데이터 변경은 쓰기 동작을 통해 이루어질 수 있다. 보다 구체적으로, 어느 작업의 처리를 위해 프로세서를 통해 제2 페이지에 관한 연산이 요구될 수 있다. 메인 메모리 공간에 제2 페이지가 없다면 메모리 폴트가 발생할 수 있고, 제2 페이지가 메인 메모리 공간으로 스왑-인될 수 있다. 프로세서의 연산 결과에 따라 제2 페이지의 데이터는 변경되거나 혹은 변경되지 않을 수 있다. 제2 페이지와 관련된 연산이 제2 페이지에 대한 쓰기 연산에 해당한다면, 제2 페이지의 데이터는 변경될 수 있다. 제2 페이지의 데이터가 변경될 경우 변경된 제2 페이지의 추가적인 스왑-아웃이 필요하므로 제2 페이지를 중복 저장 유형으로 설정하는 것에 따른 효과를 얻을 수 없다. 제2 페이지의 스왑-인이 제2 페이지에 대한 쓰기 연산에 따른 것이라면, 제2 페이지의 스왑-인 유형은 단독 저장 유형으로 결정될 수 있다.

[0039] 도 4는 일 실시예에 따른, 스왑-아웃 동작을 예시적으로 나타내는 플로우 차트이다. 도 4를 참조하면, 단계(401)에서 전자 장치는 페이지 특성 및 스왑 풀 특성을 측정한다. 페이지 특성은 핫니스 및/또는 압축률을 포함할 수 있다. 스왑 풀 특성은 스왑 풀의 성능 및/또는 가용 공간을 포함할 수 있다. 단계(402)에서 전자 장치는 제1 스왑 풀에 가용 공간(F_{CMS})이 존재하는지 체크할 수 있다. 가용 공간(F_{CMS})이 존재하는 경우, 단계(403)에서 전자 장치는 측정된 특성을 이용하여 점수를 결정할 수 있다. 단계(404)에서 전자 장치는 점수와 임계치($\Theta_{preferCMS}$)를 비교할 수 있다. 점수가 임계치($\Theta_{preferCMS}$)보다 크다면, 단계(405)에서 전자 장치는 제1 스왑 풀을 저장 위치로 선택할 수 있다. 제1 스왑 풀에 가용 공간(F_{CMS})이 없거나, 점수가 임계치($\Theta_{preferCMS}$)보다 작다면, 단계(407)에서 전자 장치는 제2 스왑 풀을 저장 위치로 선택할 수 있다. 단계(406)에서 전자 장치는 선택된 저장 위치로 스왑-아웃을 수행할 수 있다.

[0040] 일 실시예에 따르면, 아래 표 1에 따라 스왑-아웃 동작이 수행될 수 있다.

표 1

```

01: procedure doSwapOut(task, page)
02:    $H \leftarrow \text{getPageHotness}(\text{task}, \text{page})$ 
03:    $R \leftarrow \text{getPageCompressionRatio}(\text{task}, \text{page})$ 
04:    $P_{CMS} \leftarrow \text{getCompressedMemorySwapPerformance}(\text{task}, \text{page})$ 
05:    $F_{CMS} \leftarrow \text{getCompressedMemorySwapFreeSize}()$ 
06:   if  $F_{CMS} > 0$  and  $\text{preferCompressedMemorySwap}(H, R, P_{CMS}, F_{CMS}) = \text{true}$  then
07:      $B \leftarrow \text{compressedMemorySwap}$ 
08:   else
09:      $B \leftarrow \text{diskSwap}$ 
10:    $\text{writeToSwap}(\text{page}, B)$ 

```

[0041]

[0042] doSwapOut(task, page)는 (task, page)의 스왑-아웃을 수행하는 함수를 나타낼 수 있다. task를 통해 워크로드의 특정 작업이 특정될 수 있고, page를 통해 해당 작업을 위한 특정 페이지가 특정될 수 있다. getPageHotness(task, page)는 (task, page)의 핫니스를 획득하는 함수, getPageCompressionRatio(task, page)는 (task, page)의 압축률을 획득하는 함수, getCompressedMemorySwapPerformance(task, page)는 (task, page)가 저장된 제1 스왑 풀의 성능을 획득하는 함수, getCompressedMemorySwapFreeSize()는 제1 스왑 풀의 가용 공간을 획득하는 함수를 나타낼 수 있다. 획득된 데이터는 핫니스(H), 압축률(R), 성능(P_{CMS}), 가용 공간(F_{CMS})으로 저장될 수 있다. 제1 스왑 풀의 성능은 압축 및 압축 해제를 위한 지연(latency)이나 대역폭(bandwidth)을 포함할 수 있다. 페이지들은 압축된 상태로 제1 스왑 풀에 저장될 수 있고, 압축 해제된 상태로 제1 스왑 풀로부터 로드될 수 있다. 제1 스왑 풀의 성능은 이러한 압축 및 압축 해제에 어느 정도의 지연이 발생하는지 및/또는 어느 정도의 대역폭으로 이러한 압축 및 압축 해제가 수행될 수 있는지를 나타낼 수 있다. 제1 스왑 풀의 성능이 높은 것은 압축 및 압축 해제 시 지연이 적고 대역폭이 큰 것을 의미할 수 있다. 제1 스왑 풀의 성능이 높을수록 제1 스왑 풀을 이용하는 이득이 크므로 점수를 높이는 요인으로 작용할 수 있다. preferCompressedMemorySwap(H, R, P_{CMS} , F_{CMS})는 핫니스(H), 압축률(R), 성능(P_{CMS}), 가용 공간(F_{CMS})의 입력에 따라 제1 스왑 풀이 저장 위치로 선택되었는지 반환하는 함수를 나타낼 수 있다. B는 백엔드, compressedMemorySwap은 제1 스왑 풀, diskSwap은 제2 스왑 풀을 나타낼 수 있다. 백엔드(B)를 통해 저장 위치가 제1 스왑 풀 또는 제2 스왑 풀로 특정될 수 있다.

[0043] 가용 공간(F_{CMS})이 0보다 크고, $\text{preferCompressedMemorySwap}(H, R, P_{CMS}, F_{CMS})$ 의 반환 값이 참(true)인 경우, 제 1 스왑 풀이 저장 위치로 선택될 수 있다. 가용 공간(F_{CMS})이 0보다 작거나, $\text{preferCompressedMemorySwap}(H, R, P_{CMS}, F_{CMS})$ 의 반환 값이 거짓(false)인 경우, 제2 스왑 풀이 저장 위치로 선택될 수 있다. $\text{writeToSwap}(\text{page}, B)$ 는 page를 B에 저장하는 함수를 나타낼 수 있다.

[0044] 일 실시예에 따르면, 아래 표 2에 따라 저장 위치가 선택될 수 있다.

표 2

```

01: procedure preferCompressedMemorySwap( $H, R, P_{CMS}, F_{CMS}$ )
02:    $\text{score} \leftarrow \alpha \cdot H + \beta \cdot R + \gamma_{CMS} \cdot P_{CMS} + \delta_{CMS} \cdot F_{CMS}$ 
03:   if  $\text{score} > \theta_{\text{preferCMS}}$  then
04:     return true
05:   else
06:     return false

```

[0046] 핫니스(H), 압축률(R), 성능(P_{CMS}), 가용 공간(F_{CMS})에 따라 점수가 결정될 수 있다. α , β , γ_{CMS} , δ_{CMS} 는 가중치를 나타낼 수 있다. $\theta_{\text{preferCMS}}$ 는 임계치를 나타낼 수 있다. 점수가 임계치($\theta_{\text{preferCMS}}$)보다 크다면 참이 반환될 수 있고, 작다면 거짓이 반환될 수 있다. 가중치들(α , β , γ_{CMS} , δ_{CMS}) 및 임계치($\theta_{\text{preferCMS}}$)는 실험을 통해 QoS를 만족하면서 시스템 효율이 최대화되는 값으로 결정될 수 있다. 도 5는 일 실시예에 따른, 스왑-인 동작을 예시적으로 나타내는 플로우 차트이다. 도 5를 참조하면, 단계(501)에서 전자 장치는 백엔드를 확인하고, 읽기 연산을 수행한다. 백엔드는 제1 스왑 풀 또는 제2 스왑 풀에 해당할 수 있다. 확인된 백엔드에 관한 읽기 연산이 수행될 수 있다. 읽기 연산을 통해 스왑-인이 이루어질 수 있다. 단계(502)에서 전자 장치는 페이지 특성 및 스왑 풀 특성을 측정한다. 페이지 특성은 핫니스 및/또는 압축률을 포함할 수 있다. 스왑 풀 특성은 스왑 풀의 성능 및/또는 가용 공간을 포함할 수 있다.

[0047] 단계(503)에서 전자 장치는 스왑-인이 쓰기 연산에 따른 접근인지 체크할 수 있다. 쓰기 연산에 따른 접근이 아니라면, 단계(504)에서 전자 장치는 측정된 특성을 이용하여 점수를 결정할 수 있다. 단계(505)에서 전자 장치는 점수와 임계치($\theta_{\text{duplication,B}}$)를 비교할 수 있다. 점수가 임계치($\theta_{\text{duplication,B}}$)보다 크다면, 단계(506)에서 전자 장치는 스왑-인 유형을 중복 저장 유형으로 결정할 수 있다. 스왑-인이 쓰기 연산에 따른 접근이거나, 점수가 임계치($\theta_{\text{duplication,B}}$)보다 작다면, 단계(507)에서 전자 장치는 스왑-인 유형을 단독 저장 유형으로 결정할 수 있다. 전자 장치는 결정된 스왑-인 유형에 따라 스왑-인을 수행할 수 있다.

[0048] 일 실시예에 따르면, 아래 표 3에 따라 스왑-인 동작이 수행될 수 있다.

표 3

```

01: procedure doSwapIn(task, page, accessType)
02:    $B \leftarrow \text{getSwapBackend}(\text{page})$ 
03:    $\text{readFromSwap}(\text{page}, B)$ 
04:    $H \leftarrow \text{getPageHotness}(\text{task}, \text{page})$ 
05:    $R \leftarrow \text{getPageCompressionRatio}(\text{page})$ 
06:    $P_B \leftarrow \text{getSwapBackendPerformance}(\text{page}, B)$ 
07:    $F_B \leftarrow \text{getSwapBackendFreeSize}(B)$ 
08:   if  $\text{accessType} = \text{WRITE}$  or  $\text{preferDuplication}(B, H, R, P_B, F_B) = \text{false}$  then
09:      $\text{invalidate}(\text{page})$ 

```

[0050] $\text{doSwapIn}(\text{task}, \text{page}, \text{accessType})$ 은 ($\text{task}, \text{page}, \text{accessType}$)의 스왑-인을 수행하는 함수를 나타낼 수 있다. task 를 통해 워크로드의 특정 작업이 특정될 수 있고, page 를 통해 해당 작업을 위한 특정 페이지가 특정될 수 있고, accessType 을 통해 doSwapIn 의 접근 유형이 특정될 수 있다. $\text{getSwapBackend}(\text{page})$ 는 page 가 저장된 백엔드를 확인하는 함수, $\text{readFromSwap}(\text{page}, B)$ 은 (page, B)의 읽기 연산을 수행하는 함수를 나타낼 수 있다. B 는 page 의 백엔드를 나타낼 수 있다. 백엔드는 제1 스왑 풀 또는 제2 스왑 풀에 해당할 수 있

다. $\text{getPageHotness}(\text{task}, \text{page})$ 는 $(\text{task}, \text{page})$ 의 핫니스를 획득하는 함수, $\text{getPageCompressionRatio}(\text{page})$ 는 (page) 의 압축률을 획득하는 함수, $\text{getCompressedMemorySwapPerformance}(\text{page}, B)$ 는 page 가 저장된 백엔드(B)의 성능을 획득하는 함수, $\text{getCompressedMemorySwapFreeSize}(B)$ 는 백엔드(B)의 가용 공간을 획득하는 함수를 나타낼 수 있다. 획득된 데이터는 핫니스(H), 압축률(R), 성능(P_B), 가용 공간(F_B)으로 저장될 수 있다.

[0051] WRITE는 쓰기 연산을 나타낼 수 있다. $\text{preferDuplication}(B, H, R, P_B, F_B)$ 는 백엔드(B), 핫니스(H), 압축률(R), 성능(P_B), 가용 공간(F_B)의 입력에 따라 스왑-인 유형이 중복 저장 유형으로 설정되는 것이 적합한지 반환하는 함수를 나타낼 수 있다. 접근 유형이 쓰기 연산에 해당하거나 $\text{preferDuplication}(B, H, R, P_B, F_B)$ 의 반환 값이 거짓인 경우, $\text{invalidate}(\text{page})$ 가 수행될 수 있다. $\text{invalidate}(\text{page})$ 는 page 를 비유효화하는 함수를 나타낼 수 있다.

[0052] 일 실시예에 따르면, 아래 표 4에 따라 스왑-인 유형이 결정될 수 있다.

표 4

```

01: procedure preferDuplication( $B, H, R, P_B, F_B$ )
02:    $\text{score} \leftarrow \alpha \cdot H + \beta \cdot R - \gamma_B \cdot P_B + \delta_B \cdot F_B$ 
03:   if  $\text{score} > \theta_{\text{duplication}, B}$  then
04:     return true
05:   else
06:     return false

```

[0053]

[0054] 백엔드(B), 핫니스(H), 압축률(R), 성능(P_B), 가용 공간(F_B)에 따라 점수가 결정될 수 있다. $\alpha, \beta, \gamma_B, \delta_B$ 는 가중치를 나타낼 수 있다. $\theta_{\text{duplication}, B}$ 는 임계치를 나타낼 수 있다. 점수가 임계치($\theta_{\text{duplication}, B}$)보다 크다면 참이 반환될 수 있고, 작다면 거짓이 반환될 수 있다. 가중치들($\alpha, \beta, \gamma_B, \delta_B$) 및 임계치($\theta_{\text{duplication}, B}$)는 실험을 통해 QoS를 만족하면서 시스템 효율이 최대화되는 값으로 결정될 수 있다. 도 6 내지 도 8은 일 실시예에 따른, 제1 메모리의 분배 상태 및 제1 스왑 풀의 사이즈의 적응적 조절을 위한 동작들을 예시적으로 나타내는 플로우 차트이다. 도 6을 참조하면, 단계(601)에서 전자 장치는 QoS 및 시스템 효율을 측정할 수 있다. 전자 장치는 시스템 부하를 더 측정할 수 있다. 단계(602)에서 전자 장치는 현재 동작 모드가 적응 모드인지 체크한다.

[0055] 현재 동작 모드가 적응 모드(adaptation mode)인 경우, 단계(603)에서 전자 장치는 제1 메모리의 적응적 배분 상태 및 제1 스왑 풀의 적응적 사이즈를 결정한다. 전자 장치는 제1 메모리의 배분 상태 및 제1 스왑 풀의 사이즈의 조절에 따른 QoS 및 시스템 효율 중 적어도 하나의 변화에 기초하여 제1 메모리의 적응적 배분 상태 및 제1 스왑 풀의 적응적 풀 사이즈를 결정할 수 있다. 전자 장치는 적응적 배분 상태 및 적응적 풀 사이즈의 결정에 시스템 부하를 더 고려할 수 있다. 단계(604)에서 전자 장치는 적응 중단 필요 시 현재 동작 모드를 아이들 모드(idle mode)로 변경할 수 있다. 아래에서 다시 설명하겠지만, 제1 부분 공간의 사이즈(M_C)의 변화가 멈추거나, 제1 부분 풀의 사이즈(S_C)의 변화가 멈추거나, 시스템 효율이 감소하는 경우에 적응 중단 식별자(stopSubPhase)에 참의 값이 대입되고 적응 중단이 이루어질 수 있다. 단계(605)에서 전자 장치는 QoS 및 시스템 효율을 업데이트할 수 있다. 전자 장치는 시스템 부하를 더 업데이트할 수 있다.

[0056] 현재 동작 모드가 아이들 모드인 경우, 단계(607)에서 전자 장치는 결정된 적응적 배분 상태 및 적응적 풀 사이즈로 제1 메모리 및 제1 스왑 풀을 운영할 수 있다. 단계(608)에서 전자 장치는 QoS 및 시스템 효율의 변화에 대응하여 현재 동작 모드를 적응 모드로 변경할 수 있다. 전자 장치는 QoS 및/또는 시스템 효율의 변화가 큰 경우 현재 동작 모드를 적응 모드로 변경할 수 있다.

[0057] 동작(606)에서 전자 장치는 실행 중지 명령이 있는지 체크할 수 있다. 실행 중지 명령에 따라 제1 메모리의 분배 상태 조절 및 제1 스왑 풀의 사이즈 조절이 중지될 수 있다. 실행 중지 명령이 없다면 전자 장치는 단계(601)를 다시 수행할 수 있다. 전자 장치는 단계(604)를 통해 현재 동작 모드가 아이들 모드로 변경되기 전까지 단계들(601, 602, 603, 604, 605)을 반복적으로 수행하면서 제1 스왑 풀의 적응적 사이즈를 결정할 수 있다.

[0058] 일 실시예에 따르면, 아래 표 5에 따라 제1 메모리의 분배 상태 조절 및 제1 스왑 풀의 사이즈 조절이 수행될

수 있다.

표 5

```

01: global variables
02:  $T_{LC}$ ,  $T_{Batch}$ 
03:  $L_{prev}$ ,  $L_{curr}$ ,  $\Delta L$ 
04:  $M_{LC,curr}$ ,  $M_{LC,next}$ 
05:  $S_{LC,curr}$ ,  $S_{LC,next}$ ,  $S_{Batch,curr}$ ,  $S_{Batch,next}$ 
06:  $Q_{prev}$ ,  $Q_{curr}$ ,  $\Delta Q$ 
07:  $E_{prev}$ ,  $E_{curr}$ ,  $\Delta E$ 
08:  $\vec{H}$ , phase, subphase, d
09: CMSAdaptationCount, stopSubphase
10: procedure Main
11:  $T_{LC} \leftarrow \text{getLatencyCriticalTask}()$ 
12:  $T_{Batch} \leftarrow \text{getBatchTask}()$ 
13: initVariables()
14: phase  $\leftarrow$  adaptation, subphase  $\leftarrow$  exploreMemorySize
15: while true do
16:    $M_{LC,curr} \leftarrow M_{LC,next}$ 
17:    $S_{LC,curr} \leftarrow S_{LC,next}$ ,  $S_{Batch,curr} \leftarrow S_{Batch,next}$ 
18:   sleep( $t_{measure}$ )
19:    $L_{curr} \leftarrow \text{measureLoad}()$ 
20:    $Q_{curr} \leftarrow \text{measureQoS}()$ 
21:    $E_{curr} \leftarrow \text{measureEfficiency}()$ 
22:    $\Delta L \leftarrow L_{curr} - L_{prev}$ ,  $\Delta Q \leftarrow Q_{prev} - Q_{curr}$ 
23:    $\Delta E \leftarrow E_{curr} - E_{prev}$ 
24:   if phase = adaptation then
25:     if subphase = exploreMemorySize then
26:       exploreMemorySize()
27:     else
28:       exploreBatchCMSSize()
29:   else
30:     idlePhase()
31:    $L_{prev} \leftarrow L_{curr}$ ,  $Q_{prev} \leftarrow Q_{curr}$ ,  $E_{prev} \leftarrow E_{curr}$ 

```

global variables는 전역 변수를 선언하는 함수를 나타낼 수 있다. T_{LC} 는 요구 QoS 레벨을 갖는 제1 작업(예: LC 작업)의 식별 정보, T_{Batch} 는 요구 QoS 레벨을 갖지 않는 제2 작업(예: 배치 작업)의 식별 정보, L_{prev} 는 이전 부하, L_{curr} 은 현재 부하, ΔL 은 부하 변화, $M_{LC,curr}$ 는 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 현재 사이즈, $M_{LC,next}$ 는 제1 부분 공간의 다음 사이즈, $S_{LC,curr}$ 은 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 현재 사이즈, $S_{LC,next}$ 는 제1 부분 풀의 다음 사이즈, $S_{Batch,curr}$ 은 제2 작업을 위한 제1 메모리 내 제2 부분 풀의 현재 사이즈, $S_{Batch,next}$ 는 제2 부분 풀의 다음 사이즈, Q_{prev} 는 이전 QoS, Q_{curr} 은 현재 QoS, ΔQ 은 QoS 변화, E_{prev} 는 이전

시스템 효율, E_{curr} 은 현재 시스템 효율, ΔE 은 시스템 효율 변화, $\vec{H}$ 는 히스토리 데이터, phase는 현재 동작 모드, subphase는 현재 서브 동작 모드, d는 탐색 방향, CMSAdaptationCount는 카운트 식별자, stopSubphase는 적응 중단 식별자를 나타낼 수 있다. Main은 제1 메모리의 배분 상태 및 제1 스왑 풀의 사이즈의 적응적 조절을 위한 메인 함수에 해당할 수 있다. 실행 중지 명령이 있기 전까지 Main을 통해 제1 메모리의 배분 상태 조절 및 제1 스왑 풀의 사이즈 조절이 이루어질 수 있다. getLatencyCriticalTask() 및 getBatchTask()를 통해 제1 작업의 식별 정보(T_{LC}) 및 제2 작업의 식별 정보(T_{Batch})가 결정될 수 있다. getLatencyCriticalTask()는 제1 작업의 식별 정보(T_{LC}) 얻는 함수, getBatchTask()는 제2 작업의 식별 정보(T_{Batch})를 얻는 함수를 나타낼 수 있다. initVariables()를 통해 변수 초기화가 수행될 수 있다. initVariables()는 변수들을 초기화하는 함수를 나타낼 수 있다. 현재 동작 모드(phase)는 적응 모드(adaptation)으로 설정될 수 있고, 현재 서브 동작 모드(subphase)는 메모리 사이즈 탐색 모드(exploreMemorySize)로 설정될 수 있다.

while true do는 무한 반복 루프를 나타낼 수 있다. 제1 부분 공간의 현재 사이즈($M_{LC,curr}$)에 제1 부분 공간의 다음 사이즈($M_{LC,next}$)가 입력되고, 제1 부분 풀의 현재 사이즈($S_{LC,curr}$)에 제1 부분 풀의 다음 사이즈($S_{LC,next}$)가 입력되고, 제2 부분 풀의 현재 사이즈($S_{Batch,curr}$)에 제2 부분 풀의 다음 사이즈($S_{Batch,next}$)가 입력될 수 있다.

$\text{sleep}(t_{\text{measure}})$ 는 t_{measure} 동안 대기하는 함수에 해당할 수 있다. t_{measure} 는 변수 측정에 걸리는 시간을 나타낼 수 있다. $\text{measureLoad}()$ 를 통해 부하가 측정되고, $\text{measureQoS}()$ 를 통해 QoS가 측정되고, $\text{measureEfficiency}()$ 를 통해 시스템 효율이 측정될 수 있다. $\text{measureLoad}()$ 를 통해 측정된 값에 따라 현재 부하(L_{curr})가 설정될 수 있고, $\text{measureQoS}()$ 를 통해 측정된 값에 따라 현재 QoS(Q_{curr})가 설정될 수 있고, $\text{measureEfficiency}()$ 를 통해 측정된 값에 따라 현재 시스템 효율(E_{curr})이 설정될 수 있다. 현재 부하(L_{curr})와 이전 부하(L_{prev}) 간의 차이에 따라 부하 변화(ΔL)가 결정될 수 있고, 현재 QoS(Q_{curr})와 이전 QoS(Q_{prev}) 간의 차이에 따라 QoS 변화(ΔQ)가 결정될 수 있고, 현재 시스템 효율(E_{curr})과 이전 시스템 효율(E_{prev}) 간의 차이에 따라 시스템 효율 변화(ΔE)가 결정될 수 있다.

[0062] 현재 동작 모드(phase)가 적응 모드(adaptation)에 해당하고, 현재 서브 동작 모드(subphase)가 메모리 사이즈 탐색 모드(exploreMemorySize)에 해당하는 경우, $\text{exploreMemorySize}()$ 가 실행될 수 있다. $\text{exploreMemorySize}()$ 는 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 사이즈, 제2 작업을 위한 제1 메모리 내 제2 부분 공간의 사이즈, 및 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 사이즈를 결정하는 함수에 해당할 수 있다. 제1 부분 공간의 사이즈 및 제2 부분 공간의 사이즈를 통해 제1 메모리의 분배 상태가 결정될 수 있다. 현재 동작 모드(phase)가 적응 모드(adaptation)에 해당하고, 현재 서브 동작 모드(subphase)가 메모리 사이즈 탐색 모드(exploreMemorySize)에 해당하지 않는 경우, $\text{exploreBatchCMSSize}()$ 가 실행될 수 있다. $\text{exploreBatchCMSSize}()$ 는 제2 작업을 위한 제1 메모리 내 제2 부분 풀의 사이즈를 결정하는 함수에 해당할 수 있다.

[0063] 현재 동작 모드(phase)가 적응 모드(adaptation)에 해당하지 않는 경우 $\text{idlePhase}()$ 가 실행될 수 있다. $\text{idlePhase}()$ 는 적응적 분배 상태 및 적응적 풀 사이즈로 제1 메모리 및 제1 스왑 풀은 운영하는 함수에 해당할 수 있다. $\text{exploreMemorySize}()$, $\text{exploreBatchCMSSize}()$, 또는 $\text{idlePhase}()$ 에 따른 동작들이 종료되면, 이전 부하(L_{prev})가 현재 부하(L_{curr})로 업데이트되고, 이전 QoS(Q_{prev})가 현재 QoS(Q_{curr})로 업데이트되고, 이전 시스템 효율(E_{prev})이 현재 시스템 효율(S_{curr})로 업데이트될 수 있다.

[0064] 일 실시예에 따르면, 아래 표 6에 따라 변수 초기화가 수행될 수 있다.

표 6

01:	procedure initVariables()
02:	$L_{\text{prev}} \leftarrow 0; L_{\text{curr}} \leftarrow 0; \Delta L \leftarrow 0$
03:	$M_{\text{LC,curr}} \leftarrow M_{\text{max}}; M_{\text{LC,next}} \leftarrow M_{\text{max}}$
04:	$S_{\text{LC,curr}} \leftarrow S_{\text{init}}; S_{\text{LC,next}} \leftarrow S_{\text{init}}$
05:	$S_{\text{Batch,curr}} \leftarrow S_{\text{init}}; S_{\text{Batch,next}} \leftarrow S_{\text{init}}$
06:	$Q_{\text{prev}} \leftarrow 0; Q_{\text{curr}} \leftarrow 0; \Delta Q \leftarrow 0$
07:	$E_{\text{prev}} \leftarrow 0; E_{\text{curr}} \leftarrow 0; \Delta E \leftarrow 0$
08:	$\vec{H} \leftarrow \text{initBoundedBuffer}(\text{size}=N, \text{sortKey}=\text{efficiency})$
09:	$d \leftarrow \text{increase}$
10:	$\text{CMSAdaptationCount} \leftarrow 0$
11:	$\text{stopSubphase} \leftarrow \text{false}$

[0065]

[0066] 이전 부하(L_{prev}), 현재 부하(L_{curr}), 부하 변화(ΔL), 이전 QoS(Q_{prev}), 현재 QoS(Q_{curr}), QoS 변화(ΔQ), 이전 시스템 효율(E_{prev}), 현재 시스템 효율(S_{curr}), 시스템 효율 변화(ΔE), 및 카운트 식별자(CMSAdaptationCount)는 0으로 초기화될 수 있다. 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 현재 사이즈($M_{\text{LC,curr}}$) 및 제1 부분 공간의 다음 사이즈($M_{\text{LC,next}}$)는 최대 메모리 사이즈(M_{max})로 초기화될 수 있다. 최대 메모리 사이즈(M_{max})는 적정 값으로 미리 설정될 수 있다. 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 현재 사이즈($S_{\text{LC,curr}}$), 제1 부분 풀의 다음 사이즈($S_{\text{LC,next}}$), 제2 작업을 위한 제1 메모리 내 제2 부분 풀의 현재 사이즈($S_{\text{Batch,curr}}$), 및 제2 부분 풀의 다음 사이즈($S_{\text{Batch,next}}$)에 초기 사이즈(S_{init})가 입력될 수 있다. 히스토리 데이터($\vec{H}$)는 $\text{initBoundedBuffer}(\text{size}=N, \text{sortKey}=\text{efficiency})$ 를 통해 초기화될 수 있다. $\text{initBoundedBuffer}(\text{size}=N, \text{sortKey}=\text{efficiency})$ 는 N의 사이즈를 갖고 콘텐츠를 시스템 효율로 정렬하는 버퍼를 생성하는 함수에 해당할 수

있다. 탐색 방향(d)은 증가 방향으로 초기화될 수 있다. 적응 중단 식별자(stopSubphase)는 거짓으로 초기화될 수 있다. 히스토리 데이터($\vec{H}$)는 N개의 상태들(states)을 저장할 수 있다. 각 상태는 $\vec{s} = \{L, M_{LC}, S_{LC}, S_{Batch}, Q, E\}$ 와 같이 정의될 수 있다. $\vec{s}$ 는 상태, L은 부하, M_{LC} 는 제1 부분 공간의 사이즈, S_{LC} 는 제1 부분 풀의 사이즈, S_{Batch} 는 제2 부분 풀의 사이즈, Q는 QoS, E는 시스템 효율을 나타낼 수 있다. 전자 장치는 도 7a 및 도 7b에 따라 단계(603)를 수행할 수 있다. 도 7a는 표 5의 메모리 사이즈 탐색 모드(exploreMemorySize)에 해당할 수 있고, 도 7b는 표 5의 제2 부분 풀 사이즈 탐색 모드(exploreBatchCMSSize)에 해당할 수 있다. 메모리 사이즈 탐색 모드(exploreMemorySize)를 통해 제1 작업을 위한 제1 부분 공간의 사이즈, 제2 작업을 위한 제2 부분 공간의 사이즈, 및 제1 작업을 위한 제1 부분 풀의 사이즈의 탐색이 이루어질 수 있다. 제2 부분 풀 사이즈 탐색 모드(exploreBatchCMSSize)를 통해 제2 작업을 위한 제2 부분 풀의 사이즈의 탐색이 이루어질 수 있다.

[0067] 도 7a를 참조하면, 단계(701)에서 전자 장치는 현재 QoS의 적정 여부를 체크할 수 있다. 현재 QoS와 목표 QoS 간의 차이가 임계치(예: Θ_{QoS})보다 작다면, 전자 장치는 현재 QoS가 적정하다고 결정할 수 있다. 단계(702)에서 전자 장치는 히스토리 데이터를 업데이트할 수 있다. 전자 장치는 현재 부하, 제1 부분 공간의 현재 사이즈, 제1 부분 풀의 현재 사이즈, 제2 부분 풀의 현재 사이즈, 현재 QoS, 및 현재 시스템 효율을 포함하는 상태(state)를 히스토리 데이터에 저장할 수 있다.

[0068] 단계(703)에서 전자 장치는 요구 QoS 레벨에 기초하여 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 사이즈를 탐색한다. 전자 장치는 제1 작업의 QoS가 제1 작업의 요구 QoS 레벨을 만족하는 경우, 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 사이즈를 줄이면서 제1 작업의 제1 부분 공간의 제1 적응적 메모리 사이즈를 탐색할 수 있다. 단계(704)에서 전자 장치는 탐색 방향에 기초하여 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 사이즈를 탐색한다. 전자 장치는 제1 적응적 메모리 사이즈의 조절에 따라 제1 작업의 QoS가 요구 QoS 레벨을 만족하지 않는 경우, 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 사이즈를 조절하면서 제1 작업의 제1 적응적 풀 사이즈를 탐색할 수 있다.

[0069] 단계(705)에서 전자 장치는 사이즈 조절을 종료할지 결정한다. 적응 중단 식별자(stopSubphase)가 참인 경우 사이즈 조절이 종료될 수 있다. 적응 중단 식별자(stopSubphase)가 참인 경우, 단계(706)에서 전자 장치는 히스토리 데이터에 기초하여 제1 작업을 위한 제1 적응적 메모리 사이즈 및 제1 적응적 풀 사이즈를 결정할 수 있다. 일 실시예에 따르면, 전자 장치는 히스토리 데이터로부터 가장 높은 시스템 효율을 갖는 상태의 제1 적응적 메모리 사이즈 및 제1 적응적 풀 사이즈를 선택할 수 있다.

[0070] 단계(707)에서 전자 장치는 제1 작업 및 제2 작업의 제1 메모리의 분배 상태를 조절한다. 사이즈 조절이 진행 중인 경우, 제1 부분 공간의 사이즈를 탐색하는 과정의 중간 결과에 따라 제1 작업을 위한 제1 부분 공간의 사이즈가 결정될 수 있고, 제1 메모리의 사이즈와 제1 부분 공간의 사이즈 간의 차이에 기초하여 제2 작업을 위한 제2 부분 공간의 사이즈가 결정될 수 있다. 사이즈 조절이 종료된 경우, 탐색 결과에 해당하는 제1 적응적 메모리 사이즈에 따라 제1 부분 공간의 사이즈가 결정될 수 있고, 제1 메모리의 사이즈와 제1 적응적 메모리 사이즈 간의 차이에 기초하여 제2 부분 공간의 제2 적응적 메모리 사이즈가 결정될 수 있다. 제1 메모리의 분배 상태는 제1 부분 공간의 사이즈 및 제2 부분 공간의 사이즈에 대응할 수 있다.

[0071] 단계(708)에서 제1 작업의 제1 부분 풀 사이즈를 조절한다. 사이즈 조절이 진행 중인 경우, 제1 부분 풀의 사이즈를 탐색하는 과정의 중간 결과에 따라 제1 작업을 위한 제1 부분 풀의 사이즈가 결정될 수 있다. 사이즈 조절이 종료된 경우, 탐색 결과에 해당하는 제1 적응적 풀 사이즈에 따라 제1 부분 풀의 사이즈가 결정될 수 있다.

[0072] 일 실시예에 따르면, 아래 표 7에 따라 제1 메모리의 분배 상태 및 제1 부분 풀의 사이즈가 결정될 수 있다.

표 7

```

01: procedure exploreMemorySize()
02:   if satisfyQoS( $Q_{curr}$ ) then
03:      $\vec{H}.insert(\{L_{curr}, M_{LC,curr}, S_{LC,curr}, S_{Batch,curr}, Q_{curr}, E_{curr}\})$ 
04:      $slack \leftarrow (Q_{target} - Q_{curr}) / Q_{target}$ 
05:     if  $slack > 0$  and  $slack < \theta_{slack}$ 
06:        $M_{LC,next} \leftarrow M_{LC,curr}$ 
07:     else
08:        $M_{LC,next} \leftarrow \max(M_{LC,curr} - memoryStepSize, M_{min})$ 
09:       if  $M_{LC,next} = M_{LC,curr}$  then
10:         stopSubphase  $\leftarrow$  true
11:       else
12:         if CMSAdaptationCount = 1 and  $\Delta Q < 0$  then
13:            $d \leftarrow$  getOppositeDirection( $d$ )
14:            $S_{LC,next} \leftarrow$  getNextCMSSize( $S_{LC,curr}$ )
15:           if  $S_{LC,next} = S_{LC,curr}$  then
16:             stopSubphase  $\leftarrow$  true
17:           CMSAdaptationCount  $\leftarrow$  CMSAdaptationCount + 1
18:         if stopSubphase = true then
19:           if  $\vec{H} \neq \emptyset$  then
20:              $(M_{LC,next}, S_{LC,next}) \leftarrow$  getLCSizeWithMaxEfficiency( $\vec{H}$ )
21:              $d \leftarrow$  increase
22:             CMSAdaptationCount  $\leftarrow$  0
23:             stopSubphase  $\leftarrow$  false
24:             subphase  $\leftarrow$  exploreBatchCMSSize
25:           else
26:              $(M_{LC,next}, S_{LC,next}) \leftarrow (M_{LC,curr}, S_{LC,curr})$ 
27:             phase  $\leftarrow$  idle
28:           adjustMemorySize( $T_{LC}, M_{LC,next}$ )
29:           adjustMemorySize( $T_{Batch}, M_{total} - M_{LC,next}$ )
30:           adjustCMSSize( $T_{LC}, S_{LC,next}$ )

```

[0073]

[0074]

exploreMemorySize()는 제1 작업을 위한 제1 메모리 내 제1 부분 공간의 사이즈, 제2 작업을 위한 제1 메모리 내 제2 부분 공간의 사이즈, 및 제1 작업을 위한 제1 스왑 풀 내 제1 부분 풀의 사이즈를 결정하는 함수에 해당할 수 있다. 메모리 사이즈 탐색 모드(exploreMemorySize)를 통해 제1 작업의 제1 부분 공간의 사이즈 및 제1 스왑 풀의 사이즈를 조절하여 QoS를 만족하는 제1 부분 공간의 최소 사이즈의 탐색이 이루어질 수 있다. satisfyQoS(Q_{curr})은 현재 QoS(Q_{curr})가 적정한지 결정하는 함수에 해당할 수 있다. 현재 QoS(Q_{curr})와 목표 QoS(Q_{target}) 간의 차이가 임계치(예: θ_{QoS})보다 작다면, 현재 QoS(Q_{curr})는 적정하다고 결정될 수 있다. 현재 QoS(Q_{curr})가 적정하다는 것은 현재 QoS(Q_{curr})가 QoS 조건을 만족한다는 것을 의미할 수 있고, 현재 QoS(Q_{curr})가 부적정하다는 것은 현재 QoS(Q_{curr})가 QoS 조건을 위배한다는 것을 의미할 수 있다. 목표 QoS(Q_{target}) 및 QoS 조건은 워크로드 및/또는 작업의 특성에 따라 결정될 수 있다.

[0075]

현재 QoS(Q_{curr})가 적정한 경우, 히스토리 데이터($\vec{H}$)의 업데이트 및 제1 부분 공간의 사이즈 탐색이 이루어질 수 있다. $\vec{H}.insert(\{L_{curr}, M_{LC,curr}, S_{LC,curr}, S_{Batch,curr}, Q_{curr}, E_{curr}\})$ 는 현재 부하(L_{curr}), 제1 부분 공간의 현재 사이즈($M_{LC,curr}$), 제1 부분 풀의 현재 사이즈($S_{LC,curr}$), 제2 부분 풀의 현재 사이즈($S_{Batch,curr}$), 현재 QoS(Q_{curr}), 현재 시스템 효율(E_{curr})을 히스토리 데이터($\vec{H}$)에 삽입하는 함수에 해당할 수 있다. satisfyQoS(Q_{curr})가 참인 경우, 현재 부하(L_{curr}), 제1 부분 공간의 현재 사이즈($M_{LC,curr}$), 제1 부분 풀의 현재 사이즈($S_{LC,curr}$), 제2 부분 풀의 현재 사이즈($S_{Batch,curr}$), 현재 QoS(Q_{curr}), 현재 시스템 효율(E_{curr})이 하나의 상태(state)로 히스토리 데이터($\vec{H}$)에 삽입될 수 있다.

[0076]

slack은 현재 QoS(Q_{curr})로부터 목표 QoS(Q_{target})까지의 잔여 비율을 나타낼 수 있다. 현재 QoS(Q_{curr})와 목표 QoS(Q_{target})에 기초하여 잔여 비율(slack)이 결정될 수 있다. 잔여 비율(slack)이 0보다 크고, 잔여 비율(slack)이 임계치(θ_{slack})보다 작은 경우, 제1 부분 공간의 현재 사이즈($M_{LC,curr}$)가 제1 부분 공간의 다음 사이즈

($M_{LC,next}$)로 결정될 수 있다. 이에 따라 제1 부분 공간의 현재 사이즈($M_{LC,curr}$)와 제1 부분 공간의 다음 사이즈($M_{LC,next}$)가 동일해질 수 있다. 제1 부분 공간의 현재 사이즈($M_{LC,curr}$)와 제1 부분 공간의 다음 사이즈($M_{LC,next}$)가 동일한 것은 적응 중단 조건에 해당할 수 있다. 제1 부분 공간의 사이즈 변화가 멈춘 상태이므로 제1 부분 공간의 사이즈 탐색이 종료될 수 있다.

[0077] 잔여 비율(slack)이 0보다 작거나, 잔여 비율(slack)이 임계치(Θ_{slack})보다 큰 경우, 새로운 값이 제1 부분 공간의 다음 사이즈($M_{LC,next}$)로 설정될 수 있다. 새로운 값은 제1 부분 공간의 현재 사이즈($M_{LC,curr}$)로부터 memoryStepSize를 뺀 값, 및 최소 메모리 사이즈(M_{min}) 중에 큰 값에 해당할 수 있다. 최소 메모리 사이즈(M_{min})는 적정 값으로 미리 설정될 수 있다. 제1 부분 공간의 현재 사이즈($M_{LC,curr}$)와 제1 부분 공간의 다음 사이즈($M_{LC,next}$)가 동일한 경우 적응 중단 식별자(stopSubphase)는 참으로 설정될 수 있다.

[0078] 현재 QoS(Q_{curr})가 부적정한 경우, 제1 부분 풀의 사이즈 탐색이 이루어질 수 있다. 제1 부분 영역의 현재 사이즈($M_{LC,curr}$)에서 QoS를 만족하는 제1 부분 풀의 사이즈의 탐색이 이루어질 수 있다. 카운트 식별자(CMSAdaptationCount)가 0의 값을 갖는 경우의 측정 데이터(예: QoS)와 카운트 식별자(CMSAdaptationCount)가 1의 값을 갖는 경우의 측정 데이터(예: QoS)에 따라 사이즈의 탐색 방향이 설정될 수 있다. 사이즈의 탐색 방향은 초기에 설정된 증가 방향으로 유지되거나, 혹은 반대 설정을 통해 감소 방향으로 변경될 수 있다.

[0079] 카운트 식별자(CMSAdaptationCount)의 값이 1이고, QoS 변화(ΔQ)가 0 미만인 경우, 탐색 방향(d)이 반대로 설정될 수 있다. 카운트 식별자(CMSAdaptationCount)의 값이 1인 경우는 제1 부분 풀의 사이즈 변화에 따른 최초의 QoS 변화(ΔQ)를 확인할 수 있는 상황일 수 있는데, 처음부터 QoS가 감소하는 경우 탐색 방향(d)을 반대로 바꿔서 QoS의 증가를 도모할 수 있다. getOppositeDirection(d)는 탐색 방향(d)을 반대로 바꾸는 함수에 해당할 수 있다. getNextCMSSize($S_{LC,curr}$)의 리턴 값에 따라 제1 부분 풀의 다음 사이즈($S_{LC,next}$)가 설정될 수 있다. getNextCMSSize($S_{LC,curr}$)에 대해서는 추후 더 상세히 설명한다. 제1 부분 풀의 현재 사이즈($S_{LC,curr}$)와 제1 부분 풀의 다음 사이즈($S_{LC,next}$)가 동일한 경우 적응 중단 식별자(stopSubphase)는 참으로 설정될 수 있다. 다음 탐색 시기를 위해 카운트 식별자(CMSAdaptationCount)는 1만큼 증가될 수 있다.

[0080] 적응 중단 식별자(stopSubphase)가 참으로 설정되면 제1 작업을 위한 제1 부분 공간의 제1 적응적 메모리 사이즈 및 제1 작업을 위한 제1 부분 풀의 제1 적응적 풀 사이즈가 결정될 수 있다. getLCSizeWithMaxEfficiency($\vec{H}$)는 히스토리 데이터($\vec{H}$)에서 가장 높은 시스템 효율을 갖는 상태의 제1 부분 공간의 현재 사이즈($M_{LC,curr}$) 및 제1 부분 풀의 현재 사이즈($S_{LC,curr}$)를 획득하는 함수에 해당할 수 있다. 히스토리 데이터($\vec{H}$)가 공집합이 아니라면, getLCSizeWithMaxEfficiency($\vec{H}$)를 통해 획득된 값에 따라 제1 부분 공간의 다음 사이즈($M_{LC,next}$) 및 제1 부분 풀의 다음 사이즈($S_{LC,next}$)가 설정될 수 있고, 이에 따라 제1 부분 공간의 제1 적응적 메모리 사이즈 및 제1 부분 풀의 제1 적응적 풀 사이즈가 설정될 수 있다. 이때, 탐색 방향(d)은 증가 방향으로 초기화될 수 있고, 카운트 식별자(CMSAdaptationCount)는 0으로 초기화될 수 있고, 적응 중단 식별자(stopSubphase)는 거짓으로 초기화될 수 있고, 현재 서브 동작 모드(subphase)는 제2 부분 풀 사이즈 탐색 모드(exploreBatchCMSSize)로 설정될 수 있다.

[0081] 히스토리 데이터($\vec{H}$)가 공집합이라면, 제1 부분 공간의 현재 사이즈($M_{LC,curr}$) 및 제1 부분 풀의 현재 사이즈($S_{LC,curr}$)에 따라 제1 부분 공간의 다음 사이즈($M_{LC,next}$) 및 제1 부분 풀의 다음 사이즈($S_{LC,next}$)가 설정될 수 있다. 히스토리 데이터($\vec{H}$)가 공집합인 경우 히스토리 데이터($\vec{H}$)로부터 데이터 추출이 이루어지지 않는다는 점에서 히스토리 데이터($\vec{H}$)가 공집합이 아닌 경우와 다를 수 있다. 이 경우, 현재 동작 모드(phase)가 아이들 모드(idle)로 전환됨에 따라, 제2 부분 풀 사이즈 탐색 모드(exploreBatchCMSSize) 없이 제1 부분 공간의 현재 사이즈($M_{LC,curr}$) 및 제1 부분 풀의 현재 사이즈($S_{LC,curr}$)로 제1 메모리 및 제1 스왑 풀의 운영이 이루어질 수 있다.

[0082] adjustMemorySize(T_{LC} , $M_{LC,next}$)에 따르면 식별 정보(T_{LC})에 의해 식별되는 제1 작업을 위한 제1 부분 공간의 물리

적인 사이즈가 제1 부분 공간의 다음 사이즈($M_{LC,next}$)로 조절될 수 있다. $adjustMemorySize(T_{Batch}, M_{total} - M_{LC,next})$ 에 따르면 식별 정보(T_{Batch})에 의해 식별되는 제2 작업을 위한 제2 부분 공간의 물리적인 사이즈가 총 메모리 사이즈(M_{total})로부터 제1 부분 공간의 다음 사이즈($M_{LC,next}$)를 뺀 값으로 조절될 수 있다. 총 메모리 사이즈(M_{total})는 제1 작업 및 제2 작업을 위해 할당된 메모리 공간의 전체 사이즈를 나타낼 수 있다. $adjustCMSSize(T_{LC}, S_{LC,next})$ 에 따르면 식별 정보(T_{LC})에 의해 식별되는 제1 작업을 위한 제1 부분 폴의 물리적 사이즈가 제1 부분 폴의 다음 사이즈($S_{LC,next}$)로 조절될 수 있다.

[0083] 일 실시예에 따르면, 아래 표 8에 따라 제1 부분 폴의 다음 사이즈가 결정될 수 있다.

표 8

01:	procedure getNextCMSSize(S_{curr})
02:	if CMSAdaptationCount = 0 then
03:	if $S_{curr} + CMSStepSize > S_{max}$ then
04:	$d \leftarrow getOppositeDirection(d)$
05:	if $d = \text{increase}$ then
06:	return $\min(S_{curr} + CMSStepSize, S_{max})$
07:	else
08:	return $\max(S_{curr} - CMSStepSize, S_{min})$

[0084]

[0085] getNextCMSSize(S_{curr})는 현재 사이즈(S_{curr})(예: 제1 부분 폴의 현재 사이즈($S_{LC,curr}$))에 따라 다음 사이즈(S_{next})(예: 제1 부분 폴의 다음 사이즈($S_{LC,next}$))를 위한 값을 반환하는 함수에 해당할 수 있다. 카운트 식별자(CMSAdaptationCount)가 0의 값을 갖고, 현재 사이즈(S_{curr})와 CMSStepSize의 합이 최대 폴 사이즈(S_{max})보다 크다면 탐색 방향(d)은 반대로 설정될 수 있다. 최대 폴 사이즈(S_{max})는 적정 값으로 미리 설정될 수 있다. CMSStepSize는 사이즈 조절 폭을 나타낼 수 있다. 탐색 방향(d)이 증가 방향에 해당하는 경우, 현재 사이즈(S_{curr})와 CMSStepSize의 합, 및 최대 사이즈(S_{max}) 중에 작은 값이 리턴될 수 있다. 탐색 방향(d)이 증가 방향이 아닌 경우, 현재 사이즈(S_{curr})로부터 CMSStepSize를 뺀 값, 및 최소 사이즈(S_{min}) 중에 큰 값이 리턴될 수 있다. 도 7b를 참조하면, 단계(711)에서 전자 장치는 현재 QoS의 적정 여부를 체크할 수 있다. 단계(712)에서 전자 장치는 히스토리 데이터를 업데이트할 수 있다. 전자 장치는 현재 부하, 제1 부분 공간의 현재 사이즈, 제1 부분 폴의 현재 사이즈, 제2 부분 폴의 현재 사이즈, 현재 QoS, 및 현재 시스템 효율을 포함하는 상태(state)를 히스토리 데이터에 저장할 수 있다. 단계(713)에서 전자 장치는 탐색 방향에 기초하여 제2 작업을 위한 제1 스왑 폴 내 제2 부분 폴의 사이즈를 탐색한다. 전자 장치는 제1 작업의 QoS가 요구 QoS 레벨을 만족하는 상태에서 시스템 효율이 증가하도록 제2 작업을 위한 제1 스왑 폴 내 제2 부분 폴의 사이즈를 조절하면서 제2 작업의 제2 적응적 폴 사이즈를 탐색할 수 있다.

[0086] 단계(714)에서 전자 장치는 사이즈 조절을 종료할지 결정한다. 적응 중단 식별자(stopSubphase)가 참이거나, 제2 부분 폴의 사이즈(S_{Batch})의 변화가 멈추는 경우 사이즈 조절이 종료될 수 있다. 사이즈 조절이 종료되는 경우, 단계(715)에서 전자 장치는 히스토리 데이터에 기초하여 제2 작업을 위한 제2 적응적 폴 사이즈를 결정할 수 있다. 일 실시예에 따르면, 전자 장치는 히스토리 데이터로부터 가장 높은 시스템 효율을 갖는 상태의 제2 적응적 폴 사이즈를 선택할 수 있다.

[0087] 단계(716)에서 제2 작업의 제1 부분 폴 사이즈를 조절한다. 사이즈 조절이 진행 중인 경우, 제2 부분 폴의 사이즈를 탐색하는 과정의 중간 결과에 따라 제2 작업을 위한 제2 부분 폴의 사이즈가 결정될 수 있다. 사이즈 조절이 종료된 경우, 탐색 결과에 해당하는 제2 적응적 폴 사이즈에 따라 제2 부분 폴의 사이즈가 결정될 수 있다.

[0088] 일 실시예에 따르면, 아래 표 9에 따라 제2 부분 폴의 사이즈가 결정될 수 있다.

표 9

```

01: procedure exploreBatchCMSSize()
02:   if satisfyQoS( $Q_{curr}$ ) then
03:      $\vec{H}.insert(\{L_{curr}, M_{LC,curr}, S_{LC,curr}, S_{Batch,curr}, Q_{curr}, E_{curr}\})$ 
04:     if CMSAdaptationCount = 1 then
05:       if  $\Delta E < 0$  then
06:          $d \leftarrow getOppositeDirection(d)$ 
07:       else if CMSAdaptationCount > 1 then
08:         if  $\Delta E < 0$  then
09:           stopSubPhase  $\leftarrow$  true
10:          $S_{Batch,next} \leftarrow getNextCMSSize(S_{Batch,curr})$ 
11:         if stopSubPhase = true or  $S_{Batch,next} = S_{Batch,curr}$  then
12:            $S_{Batch,next} \leftarrow getBatchCMSSizeWithMaxEfficiency(\vec{H})$ 
13:           phase  $\leftarrow$  idle
14:         adjustCMSSize( $T_{Batch}, S_{Batch,next}$ )
15:         CMSAdaptationCount  $\leftarrow$  CMSAdaptationCount + 1

```

[0089]

[0090]

exploreBatchCMSSize()는 제2 작업을 위한 제1 스왑 풀 내 제2 부분 풀의 사이즈를 결정하는 함수에 해당할 수 있다. 제2 부분 풀 사이즈 탐색 모드(exploreBatchCMSSize)를 통해 제1 작업의 QoS가 요구 QoS 레벨을 만족하는 상태에서 시스템 효율이 증가하도록 제2 부분 풀의 사이즈 탐색이 이루어질 수 있다. 제2 작업은 요구 QoS 레벨을 갖지 않기 때문에, 제1 작업의 요구 QoS 레벨이 고려될 수 있다. satisfyQoS(Q_{curr})은 현재 QoS(Q_{curr})가 적절한지 결정하는 함수에 해당할 수 있다. 현재 QoS(Q_{curr})가 적절한 경우, 히스토리 데이터($\vec{H}$)의 업데이트가 이루어질 수 있다. $\vec{H}.insert(\{L_{curr}, M_{LC,curr}, S_{LC,curr}, S_{Batch,curr}, Q_{curr}, E_{curr}\})$ 는 현재 부하(L_{curr}), 제1 부분 공간의 현재 사이즈($M_{LC,curr}$), 제1 부분 풀의 현재 사이즈($S_{LC,curr}$), 제2 부분 풀의 현재 사이즈($S_{Batch,curr}$), 현재 QoS(Q_{curr}), 현재 시스템 효율(E_{curr})을 히스토리 데이터($\vec{H}$)에 삽입하는 함수에 해당할 수 있다.

[0091]

카운트 식별자(CMSAdaptationCount)가 0의 값을 갖는 경우의 측정 데이터(예: 시스템 효율)와 카운트 식별자(CMSAdaptationCount)가 1의 값을 갖는 경우의 측정 데이터(예: 시스템 효율)에 따라 사이즈의 탐색 방향이 설정될 수 있다. 사이즈의 탐색 방향은 초기에 설정된 증가 방향으로 유지되거나, 혹은 반대 설정을 통해 감소 방향으로 변경될 수 있다.

[0092]

카운트 식별자(CMSAdaptationCount)의 값이 1이고, 시스템 효율 변화(ΔE)가 0 미만인 경우, 탐색 방향(d)이 반대로 설정될 수 있다. 카운트 식별자(CMSAdaptationCount)의 값이 1인 경우는 제2 부분 풀의 사이즈 변화에 따른 최초의 시스템 효율 변화(ΔE)를 확인할 수 있는 상황일 수 있는데, 처음부터 시스템 효율이 감소하는 경우 탐색 방향(d)을 반대로 바꿔서 시스템 효율의 증가를 도모할 수 있다. getOppositeDirection(d)는 탐색 방향(d)을 반대로 바꾸는 함수에 해당할 수 있다. 카운트 식별자(CMSAdaptationCount)의 값이 1보다 큰 경우는 한번 정해진 탐색 방향(d)으로의 탐색이 이루어지고 있는 상황일 수 있는데, 시스템 효율 변화(ΔE)가 0 미만인 것은 최적화가 이루어진 것을 의미할 수 있다. 이 경우, 적응 중단 식별자(stopSubPhase)에 참의 값이 대입되고 적응 중단이 이루어질 수 있다.

[0093]

getNextCMSSize($S_{Batch,curr}$)의 리턴 값에 따라 제2 부분 풀의 다음 사이즈($S_{Batch,next}$)가 설정될 수 있다. getNextCMSSize($S_{Batch,curr}$)에 대해서는 표 8의 설명이 적용될 수 있다.

[0094]

적응 중단 식별자(stopSubphase)가 참이거나, 제1 부분 풀의 다음 사이즈($S_{Batch,next}$)와 현재 사이즈(S)가 동일한 경우, 제2 작업을 위한 제2 부분 풀의 제2 적응적 풀 사이즈가 결정될 수 있다. getBatchCMSSizeWithMaxEfficiency($\vec{H}$)는 히스토리 데이터($\vec{H}$)에서 가장 높은 시스템 효율을 갖는 상태의 제2 부분 풀의 현재 사이즈($S_{Batch,curr}$)를 획득하는 함수에 해당할 수 있다. getLCSizeWithMaxEfficiency($\vec{H}$)를 통해 획득된 값에 따라 제1 부분 풀의 다음 사이즈($S_{Batch,next}$)가 설정될 수 있고, 이에 따라 제2 부분 풀의 제2 적응적 풀 사이즈가 설정될 수 있다. 현재 동작 모드(phase)는 아이들 모드(idle)로 설정될 수 있다.

[0095] $\text{adjustCMSSize}(T_{\text{Batch}}, S_{\text{Batch, next}})$ 에 따르면 식별 정보(T_{Batch})에 의해 식별되는 제2 작업을 위한 제2 부분 풀의 물리적 사이즈가 제2 부분 풀의 다음 사이즈($S_{\text{Batch, next}}$)로 조절될 수 있다. 다음 탐색 시기를 위해 카운트 식별자($\text{CMSAdaptationCount}$)는 1만큼 증가될 수 있다.

[0096] 일 실시예에 따르면, 아래 표 10에 따라 아이들 모드가 수행될 수 있다.

표 10

[0097]

```

01: procedure idlePhase()
02:   if  $|\Delta L| > \theta_L$  or  $|\Delta Q| > \theta_Q$  or  $|\Delta E| > \theta_E$  then
03:     initVariables()
04:     phase  $\leftarrow$  adaptation
05:     subphase  $\leftarrow$  exploreMemorySize
    
```

[0098] $\text{idlePhase}()$ 는 적응적 사이즈로 제1 메모리 및 제1 스왑 풀을 운영하는 함수에 해당할 수 있다. 부하 변화(ΔL)가 임계치(θ_L)보다 크거나, QoS 변화(ΔQ)가 임계치(θ_Q)보다 크거나, 시스템 효율 변화(ΔE)가 임계치(θ_E)보다 큰 경우, $\text{initVariables}()$ 를 통해 변수들이 초기화될 수 있고, 현재 동작 모드(phase)는 적응 모드(adaptation)로 설정될 수 있고, 현재 서브 동작 모드(subphase)는 메모리 사이즈 탐색 모드(exploreMemorySize)로 설정될 수 있다. 도 8을 참조하면, 단계(801)에서 전자 장치는 제1 스왑 풀의 사이즈 감소가 필요한지 체크할 수 있다. 사이즈 감소가 필요하다면, 단계(802)에서 전자 장치는 빅티 페이지(victim page)에 대한 축출(eviction)을 수행할 수 있다. 전자 장치는 제1 스왑 풀에 저장된 페이지들로부터 빅티 페이지를 선택하고, 빅티 페이지에 대한 축출을 수행할 수 있다. 전자 장치는 빅티 페이지가 중복 페이지인 경우 빅티 페이지에 대한 비유효화를 수행하고, 빅티 페이지가 중복 페이지가 아닌 경우 빅티 페이지를 제2 스왑 풀에 저장할 수 있다. 사이즈 감소가 필요하지 않다면, 다시 말해 사이즈 증가가 필요하다면, 단계(804)에서 전자 장치는 메모리 리클레임(memory reclaim)을 요청할 수 있다. 단계(805)에서 전자 장치는 제1 스왑 풀의 사이즈를 조절할 수 있다.

[0099] 일 실시예에 따르면, 아래 표 11에 따라 제1 스왑 풀의 사이즈 조절이 수행될 수 있다.

표 11

[0100]

```

01: procedure adjustCMSSize(task,  $S_{\text{new}}$ )
02:    $S_{\text{curr}} \leftarrow \text{getCurrentCompressedMemorySwapSize}(\text{task})$ 
03:   if  $S_{\text{new}} < S_{\text{curr}}$  then
04:      $M \leftarrow S_{\text{curr}} - S_{\text{new}}$ 
05:     shrinkCompressedMemorySwap(task,  $M$ )
06:   else if  $S_{\text{new}} > S_{\text{curr}}$  then
07:      $F_{\text{memory}} \leftarrow \text{getFreeMemorySize}(\text{task})$ 
08:      $\text{sizeToReclaim} \leftarrow S_{\text{new}} - S_{\text{curr}} - F_{\text{memory}}$ 
09:     if  $\text{sizeToReclaim} > 0$  then
10:       requestMemoryReclaim(task,  $\text{sizeToReclaim}$ )
11:     compressedMemorySwap.setSize(task,  $S_{\text{new}}$ )
    
```

[0101] $\text{adjustCMSSize}(\text{task}, S_{\text{new}})$ 는 새로운 사이즈(S_{new})에 따라 작업(task)의 제1 스왑 풀의 부분 풀의 사이즈를 물리적으로 조절하는 함수에 해당할 수 있다. $\text{getCurrentCompressedMemorySwapSize}(\text{task})$ 는 작업(task)의 현재 제1 스왑 풀의 부분 풀의 사이즈를 획득하는 함수에 해당할 수 있다. $\text{getCurrentCompressedMemorySwapSize}(\text{task})$ 의 리턴 값은 현재 사이즈(S_{curr})로 저장될 수 있다. 현재 사이즈(S_{curr})가 새로운 사이즈(S_{new})보다 큰 경우, 다시 말해 제1 스왑 풀의 사이즈 감소가 필요한 경우, $\text{shrinkCompressedMemorySwap}(\text{task}, M)$ 을 통해 사이즈 조절이 이루어질 수 있다. $\text{shrinkCompressedMemorySwap}(\text{task}, M)$ 은 작업(task)의 제1 스왑 풀의 부분 풀로부터 M bytes 만큼의 공간을 확보하는 함수에 해당할 수 있다. M 은 현재 사이즈(S_{curr})와 새로운 사이즈(S_{new}) 간의 차이를 나타낼 수 있다. 새로운 사이즈(S_{new})가 현재 사이즈(S_{curr})보다 큰 경우, 다시 말해 제1 스왑 풀의 사이즈 증가가 필요한 경우, 메모리 리클레임이 요청될 수 있다. $\text{getFreeMemorySize}(\text{task})$ 는 작업(task)의 부분 풀에 추가 가능한 메인 메모리 공

간의 여유 공간을 리턴하는 함수에 해당할 수 있다. $\text{getFreeMemorySize}(\text{task})$ 의 리턴 값은 여유 공간(F_{memory})에 저장될 수 있다. 새로운 사이즈(S_{new})로부터 현재 사이즈(S_{curr})를 뺀 값으로부터 여유 공간(F_{memory})을 뺀 값이 리클레임 사이즈(sizeToReclaim)에 저장될 수 있다. 리클레임 사이즈(sizeToReclaim)가 0보다 크다면 $\text{requestMemoryReclaim}(\text{task}, \text{sizeToReclaim})$ 을 통해 운영체제 측에 대한 메모리 리클레임 요청이 수행될 수 있다. $\text{requestMemoryReclaim}(\text{task}, \text{sizeToReclaim})$ 는 리클레임 사이즈(sizeToReclaim)만큼의 추가적인 메모리 공간을 요청하는 함수에 해당할 수 있다. $\text{compressedMemorySwap.setSize}(\text{task}, S_{\text{new}})$ 는 작업(task)의 제1 스왑 폴의 부분을 새로운 사이즈(S_{new})로 설정하는 함수에 해당할 수 있다.

[0102] 일 실시예에 따르면, 아래 표 12에 따라 제1 스왑 폴의 사이즈 감소가 수행될 수 있다.

표 12

```

01: procedure shrinkCompressedMemorySwap(task, M)
02:   victimPageList  $\leftarrow$  getVictimPages(task, M)
03:   for page in victimPageList then
04:     if isDuplicatedPage(page) = true then
05:       invalidate(page)
06:     else
07:       decompress(page)
08:       evictToDisk(page)

```

[0103]

[0104]

$\text{shrinkCompressedMemorySwap}(\text{task}, M)$ 은 제1 스왑 폴의 부분 폴로부터 M bytes 만큼의 공간을 확보하는 함수에 해당할 수 있다. $\text{getVictimPages}(\text{task}, M)$ 는 M bytes 만큼의 빅티 페이지들을 획득하는 함수에 해당할 수 있다. $\text{getVictimPages}(\text{task}, M)$ 에 의해 획득된 페이지들은 리스트(victimPageList)에 저장될 수 있다. 리스트(victimPageList)의 각 페이지의 유형에 따라 비유효화 또는 축출이 수행될 수 있다. $\text{isDuplicatedPage}(\text{page})$ 는 page 가 중복 저장 유형에 해당하는지 리턴하는 함수에 해당할 수 있다. page 가 중복 저장 유형에 해당하는 경우 page 는 비유효화될 수 있다. page 가 중복 저장 유형에 해당하지 않는 경우, 다시 말해 단독 저장 유형에 해당하는 경우, page 의 압축이 해제될 수 있고, page 는 제2 스왑 폴에 저장될 수 있다. $\text{invalidate}(\text{page})$ 는 page 를 비유효화시키는 함수에 해당할 수 있고, $\text{decompress}(\text{page})$ 는 page 의 압축을 해제하는 함수에 해당할 수 있고, $\text{evictToDisk}(\text{page})$ 는 page 를 제2 스왑 폴로 축출하는 함수에 해당할 수 있다. 일 실시예에 따르면, 아래 표 13에 따라 빅티 페이지가 선택될 수 있다.

표 13

```

01: procedure getVictimPages(task, M)
02:   candidatePageList  $\leftarrow$  getCandidatePages(task)
03:   for page in candidatePageList then
04:      $H \leftarrow$  getPageHotness(page)
05:      $R \leftarrow$  getPageCompressionRatio(page)
05:      $P_{\text{CMS}} \leftarrow$  getCompressedMemorySwapPerformance(page)
06:      $D \leftarrow$  isDuplicatedPage(page)
07:      $\text{page.score} \leftarrow$  calculateScore( $H, R, P_{\text{CMS}}, D$ )
08:   sortPageListByScore(candidatePageList)
09:   return getLowScorePages(candidatePageList, M)

```

[0105]

[0106]

$\text{getVictimPages}(\text{task}, M)$ 는 M bytes 만큼의 빅티 페이지들을 획득하는 함수에 해당할 수 있다. $\text{getCandidatePages}(\text{task})$ 는 후보 페이지를 획득하는 함수에 대응할 수 있다. 오버헤드의 최소화를 위해 제1 스왑 폴의 전체 페이지가 아닌 후보 페이지로부터 빅티가 선택될 수 있다. 후보 페이지는 운영체제가 제공하는 비활성 리스트(inactive list) 기반으로 결정될 수 있다. 후보 페이지들은 리스트(candidatePageList)에 저장될 수 있다. 리스트(candidatePageList)의 각 후보 페이지로부터 빅티 페이지가 선택될 수 있다. 빅티 페이지는 특성 기반으로 선택될 수 있다. $\text{getPageHotness}(\text{page})$ 는 page 의 핫니스를 획득하는 함수, $\text{getPageCompressionRatio}(\text{page})$ 는 page 의 압축률을 획득하는 함수, $\text{getCompressedMemorySwapPerformance}(\text{page})$ 는 page 가 저장된 제1 스왑 폴의 성능을 획득하는 함수, $\text{isDuplicatedPage}(\text{page})$ 는 page 가 중복 저장 유형에

해당하는지 획득하는 함수를 나타낼 수 있다. 획득된 데이터는 핫니스(H), 압축률(R), 성능(P_{CMS}), 중복 유형(D)으로 저장될 수 있다.

[0107] calculateScore(H, R, P_{CMS} , D)는 핫니스(H), 압축률(R), 성능(P_{CMS}), 중복 유형(D)의 입력에 따라 점수를 반환하는 함수를 나타낼 수 있다. 점수는 page.score로 저장될 수 있다. sortPageListByScore(candidatePageList)는 리스트(candidatePageList)의 후보 페이지들을 점수로 정렬하는 함수에 해당할 수 있다. getLowScorePages(candidatePageList, M)은 후보 페이지들로부터 점수가 낮은 순서로 M bytes 만큼의 후보 페이지들을 획득하는 함수를 나타낼 수 있다. getLowScorePages(candidatePageList, M)의 리턴 값을 통해 빅팁 페이지가 선택될 수 있다.

[0108] 일 실시예에 따르면, 아래 표 14에 따라 점수가 결정될 수 있다.

표 14

01: procedure calculateScore(H, R, P_{CMS}, D) 02: score $\leftarrow \alpha \cdot H + \beta \cdot R + \gamma_{CMS} \cdot P_{CMS} + \epsilon \cdot D$ 03: return score

[0109]

[0110] 핫니스(H), 압축률(R), 성능(P_{CMS}), 중복 유형(D)에 따라 점수가 결정될 수 있다. α , β , γ_{CMS} , ϵ 은 가중치를 나타낼 수 있다. 도 9는 일 실시예에 따른, 전자 장치의 구성요소들을 예시적으로 나타내는 블록도이다. 도 9를 참조하면, 전자 장치(900)는 프로세서(910), 메모리(920), 카메라(930), 저장 장치(940), 입력 장치(950), 출력 장치(960) 및 네트워크 인터페이스(970)를 포함할 수 있으며, 이들은 통신 버스(980)를 통해 서로 통신할 수 있다. 예를 들어, 전자 장치(900)는 이동 전화, 스마트 폰, PDA, 넷북, 태블릿 컴퓨터, 랩톱 컴퓨터 등과 같은 모바일 장치, 스마트 워치, 스마트 밴드, 스마트 안경 등과 같은 웨어러블 디바이스, 데스크탑, 서버, 데이터 센터 등과 같은 컴퓨팅 장치, 텔레비전, 스마트 텔레비전, 냉장고 등과 같은 가전 제품, 도어 락 등과 같은 보안 장치, 자율주행 차량, 스마트 차량 등과 같은 차량의 적어도 일부로 구현될 수 있다. 전자 장치(900)는 도 2의 전자 장치(200)를 구조적 및/또는 기능적으로 포함할 수 있다.

[0111] 프로세서(910)는 전자 장치(900) 내에서 실행하기 위한 기능 및 명령어들을 실행한다. 예를 들어, 프로세서(910)는 메모리(920) 또는 저장 장치(940)에 저장된 명령어들을 처리할 수 있다. 프로세서(910)는 도 1 내지 도 8, 도 10, 및 도 11의 동작을 수행할 수 있다. 메모리(920)는 컴퓨터 판독가능한 저장 매체 또는 컴퓨터 판독가능한 저장 장치를 포함할 수 있다. 메모리(920)는 프로세서(910)에 의해 실행하기 위한 명령어들을 저장할 수 있고, 전자 장치(900)에 의해 소프트웨어 및/또는 애플리케이션이 실행되는 동안 관련 정보를 저장할 수 있다.

[0112] 카메라(930)는 사진 및/또는 비디오를 촬영할 수 있다. 저장 장치(940)는 컴퓨터 판독가능한 저장 매체 또는 컴퓨터 판독가능한 저장 장치를 포함한다. 저장 장치(940)는 메모리(920)보다 더 많은 양의 정보를 저장하고, 정보를 장기간 저장할 수 있다. 예를 들어, 저장 장치(940)는 자기 하드 디스크, 광 디스크, 플래시 메모리, 플로피 디스크 또는 이 기술 분야에서 알려진 다른 형태의 비휘발성 메모리를 포함할 수 있다.

[0113] 입력 장치(950)는 키보드 및 마우스를 통한 전통적인 입력 방식, 및 터치 입력, 음성 입력, 및 이미지 입력과 같은 새로운 입력 방식을 통해 사용자로부터 입력을 수신할 수 있다. 예를 들어, 입력 장치(950)는 키보드, 마우스, 터치 스크린, 마이크로폰, 또는 사용자로부터 입력을 검출하고, 검출된 입력을 전자 장치(900)에 전달할 수 있는 임의의 다른 장치를 포함할 수 있다. 출력 장치(960)는 시각적, 청각적 또는 촉각적인 채널을 통해 사용자에게 전자 장치(900)의 출력을 제공할 수 있다. 출력 장치(960)는 예를 들어, 디스플레이, 터치 스크린, 스피커, 진동 발생 장치 또는 사용자에게 출력을 제공할 수 있는 임의의 다른 장치를 포함할 수 있다. 네트워크 인터페이스(970)는 유선 또는 무선 네트워크를 통해 외부 장치와 통신할 수 있다.

[0114] 도 10은 일 실시예에 따른, 스왑-아웃 동작을 예시적으로 나타내는 플로우 차트이다. 전자 장치는 제1 메모리, 제2 메모리, 및 프로세서를 포함할 수 있다. 제1 메모리는 메인 메모리 공간 및 제1 스왑 풀을 포함할 수 있다. 제2 메모리는 제1 메모리에 비해 프로세서로부터 멀게 위치하고 제2 스왑 풀을 포함할 수 있다.

[0115] 도 10을 참조하면, 전자 장치는 단계(1010)에서 제1 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제1 스왑 풀 및 제2 스왑 풀로부터 제1 페이지의 스왑-아웃에 따른 제1 페이지의 저장 위치를

선택하고, 단계(1020)에서 제1 페이지의 선택된 저장 위치에 기초하여 제1 페이지의 스왑-아웃을 수행한다.

- [0116] 프로세서는, 제1 페이지의 저장 위치를 선택하기 위해, 제1 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제1 페이지의 제1 점수를 결정하고, 제1 페이지의 제1 점수와 제1 임계치 간의 비교 결과에 따라 제1 페이지의 저장 위치를 선택할 수 있다. 제1 점수는 제1 페이지의 핫니스(hotness)가 높을수록, 제1 페이지의 압축률이 높을수록, 제1 스왑 풀의 성능이 높을수록, 제1 스왑 풀의 가용 공간이 클수록, 높게 결정될 수 있다.
- [0117] 그 밖에, 도 1 내지 도 9, 및 도 11의 설명이 도 10의 스왑-아웃 동작에 적용될 수 있다.
- [0118] 도 11은 일 실시예에 따른, 스왑-인 동작을 예시적으로 나타내는 플로우 차트이다. 전자 장치는 제1 메모리, 제2 메모리, 및 프로세서를 포함할 수 있다. 제1 메모리는 메인 메모리 공간 및 제1 스왑 풀을 포함할 수 있다. 제2 메모리는 제1 메모리에 비해 프로세서로부터 멀게 위치하고 제2 스왑 풀을 포함할 수 있다.
- [0119] 도 11을 참조하면, 전자 장치는 단계(1110)에서 제2 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제2 페이지를 메인 메모리 공간과 제1 스왑 풀 또는 제2 스왑 풀에 중복(duplicate)으로 저장할지에 관한 제2 페이지의 스왑-인 유형을 결정하고, 제2 페이지의 결정된 스왑-인 유형에 기초하여 제2 페이지의 스왑-인을 수행한다.
- [0120] 프로세서는, 제2 페이지의 스왑-인 유형을 결정하기 위해, 제2 페이지, 제1 스왑 풀, 및 제2 스왑 풀 중 적어도 하나의 특성에 기초하여, 제2 페이지의 제2 점수를 결정하고, 제2 페이지의 제2 점수와 제2 임계치 간의 비교 결과에 따라 제2 페이지의 스왑-인 유형을 결정할 수 있다. 제2 점수는 제2 페이지의 핫니스(hotness)가 높을수록, 제2 페이지의 압축률이 높을수록, 제2 페이지가 저장된 스왑 풀의 성능이 낮을수록, 제2 페이지가 저장된 스왑 풀의 가용 공간이 클수록, 높게 결정될 수 있다.
- [0121] 프로세서는, 제2 페이지의 스왑-인 유형을 결정하기 위해, 제2 페이지의 스왑-인이 제2 페이지에 대한 쓰기 연산에 따른 메인 메모리 공간 상의 메모리 폴트에 의해 발생한 경우, 제2 페이지의 스왑-인 유형을 메인 메모리 공간에 대한 단독 저장 유형으로 결정할 수 있다.
- [0122] 그 밖에, 도 1 내지 도 10의 설명이 도 11의 스왑-인 동작에 적용될 수 있다.
- [0123] 이상에서 설명된 실시예들은 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치, 방법 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0124] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 저장될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0125] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 저장할 수 있으며 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와

같은 광기록 매체(optical media), 플로피컬 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.

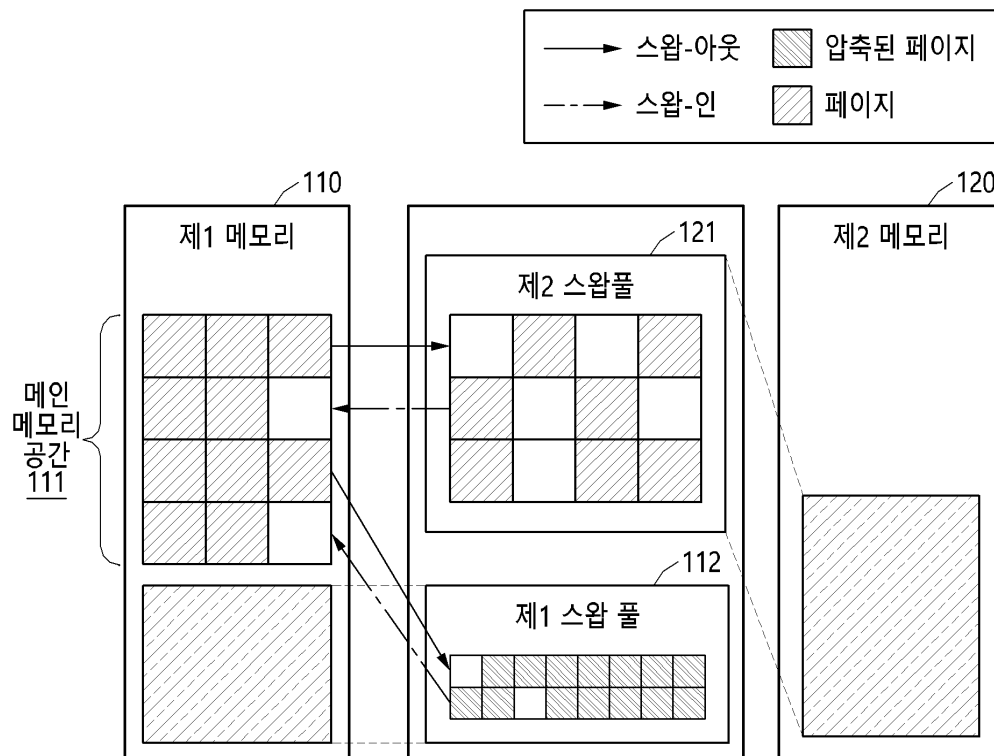
[0126] 위에서 설명한 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 또는 복수의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0127] 이상과 같이 실시예들이 비록 한정된 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 이를 기초로 다양한 기술적 수정 및 변형을 적용할 수 있다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

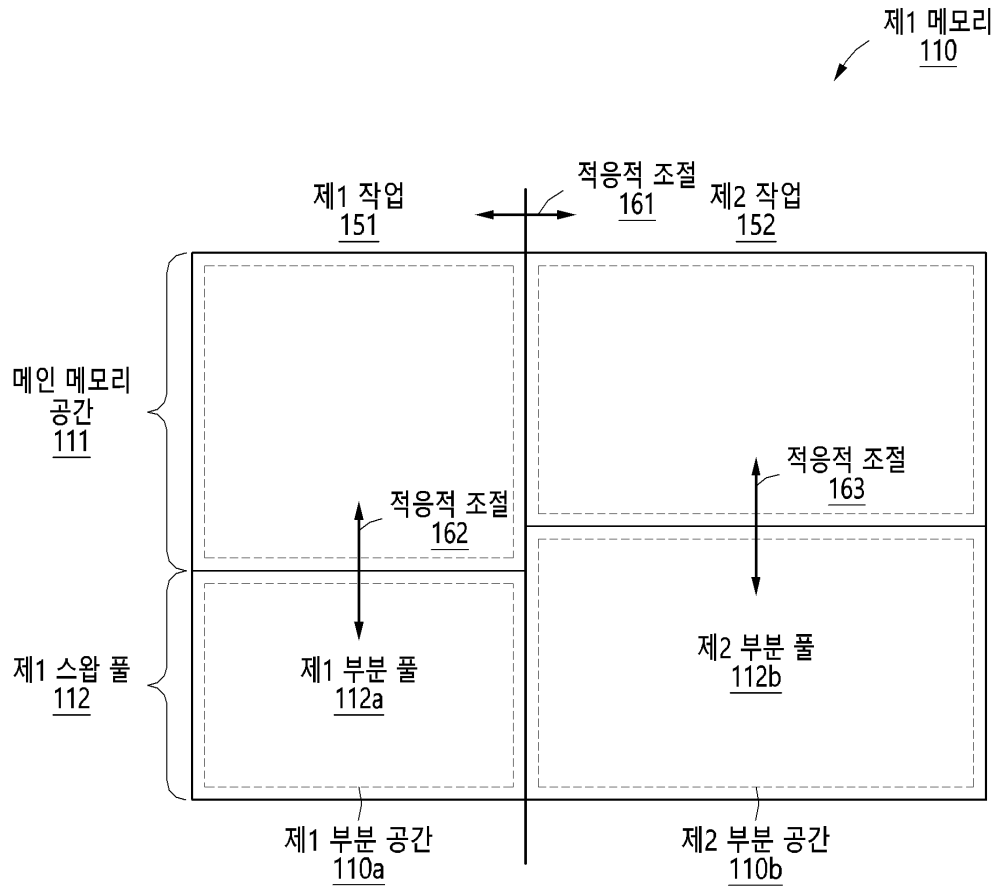
[0128] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

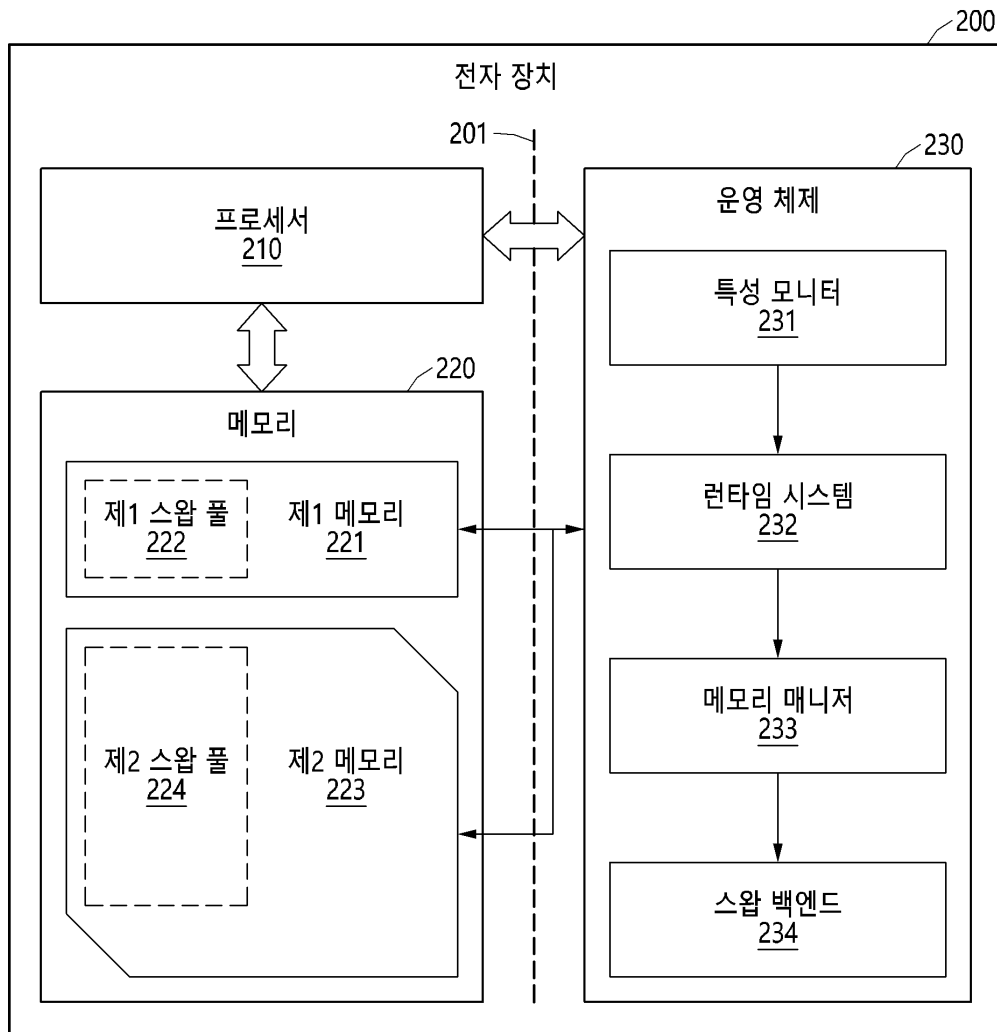
도면1a



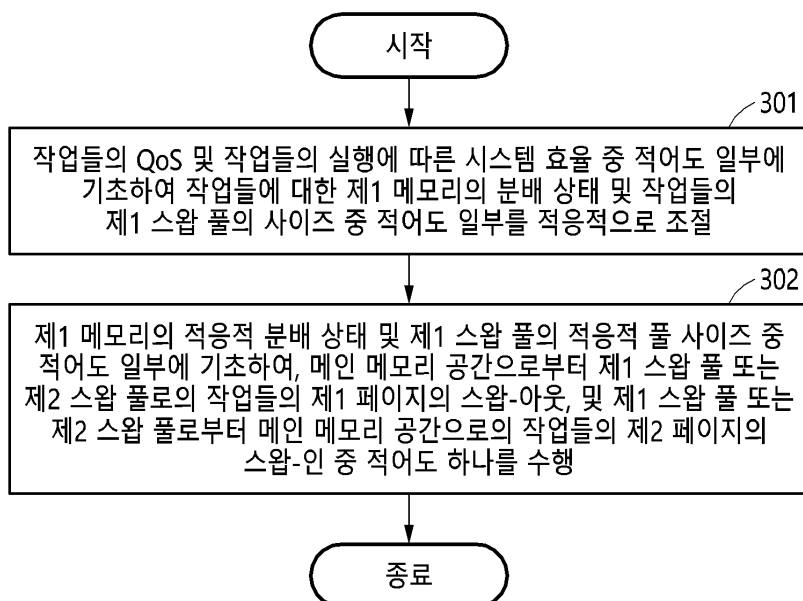
도면 1b



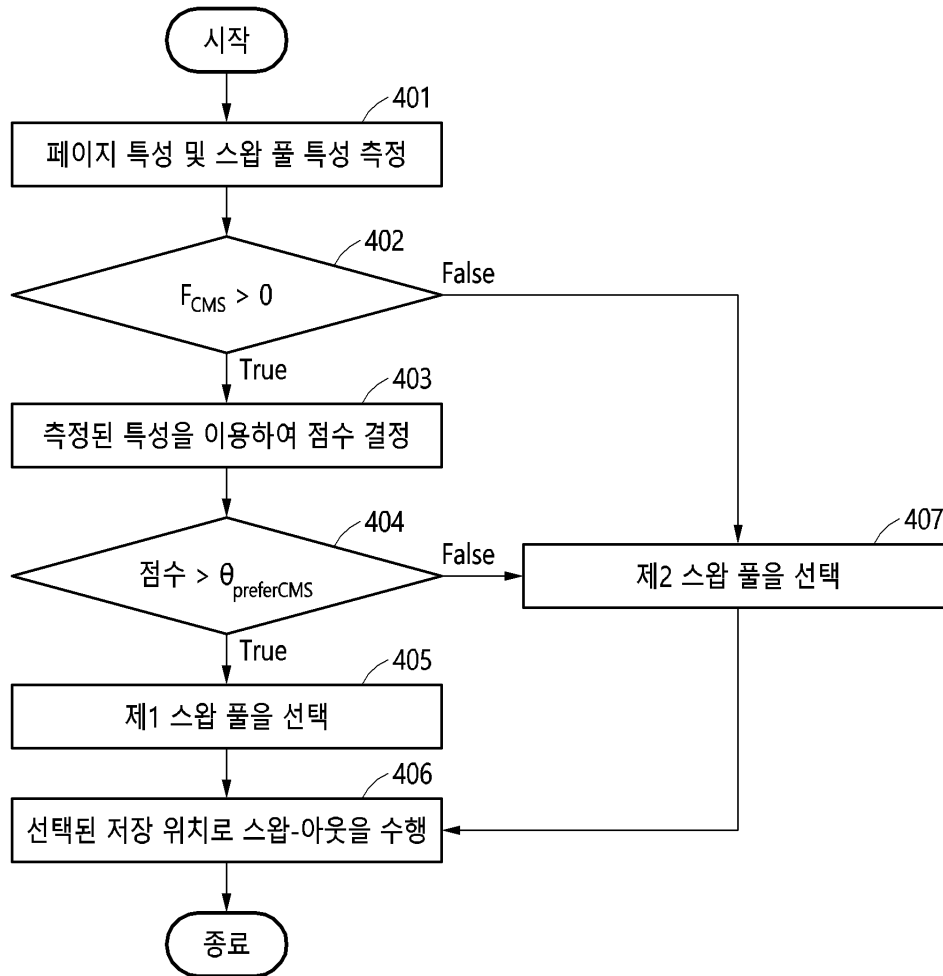
도면2



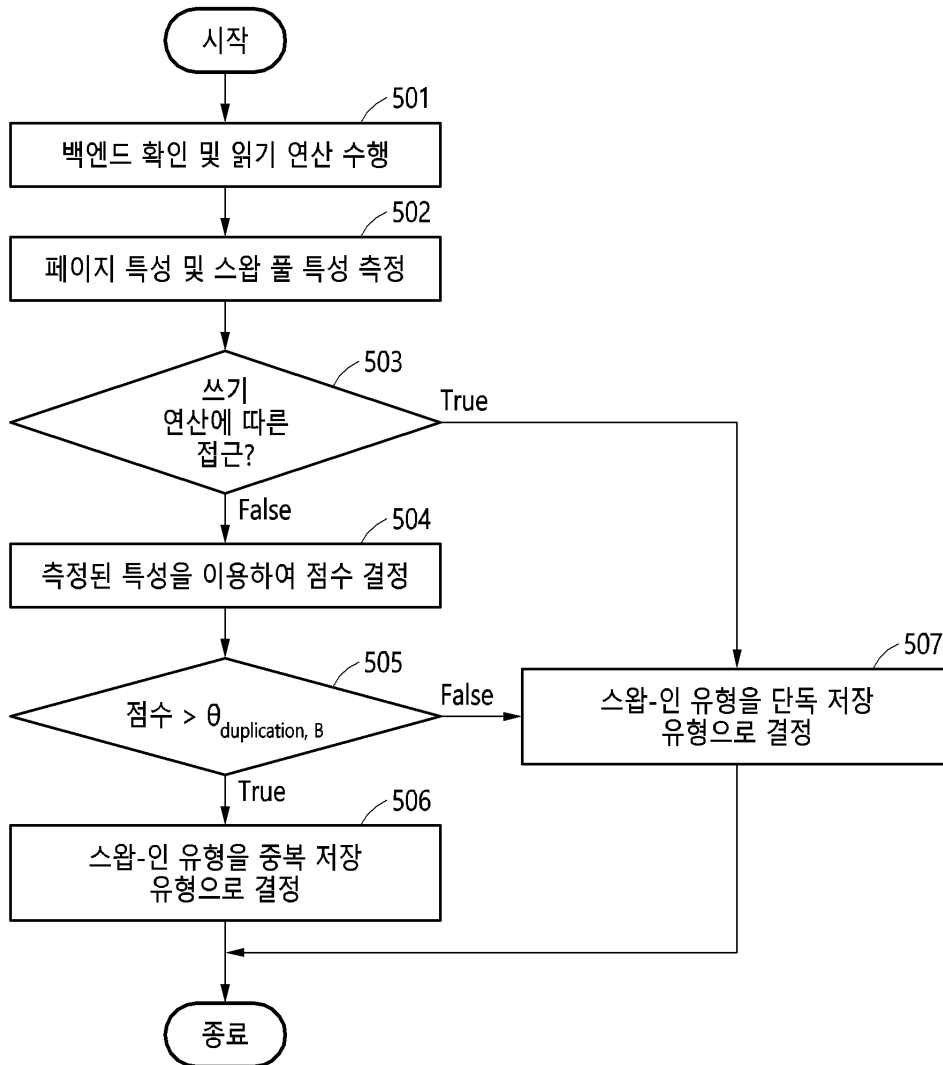
도면3



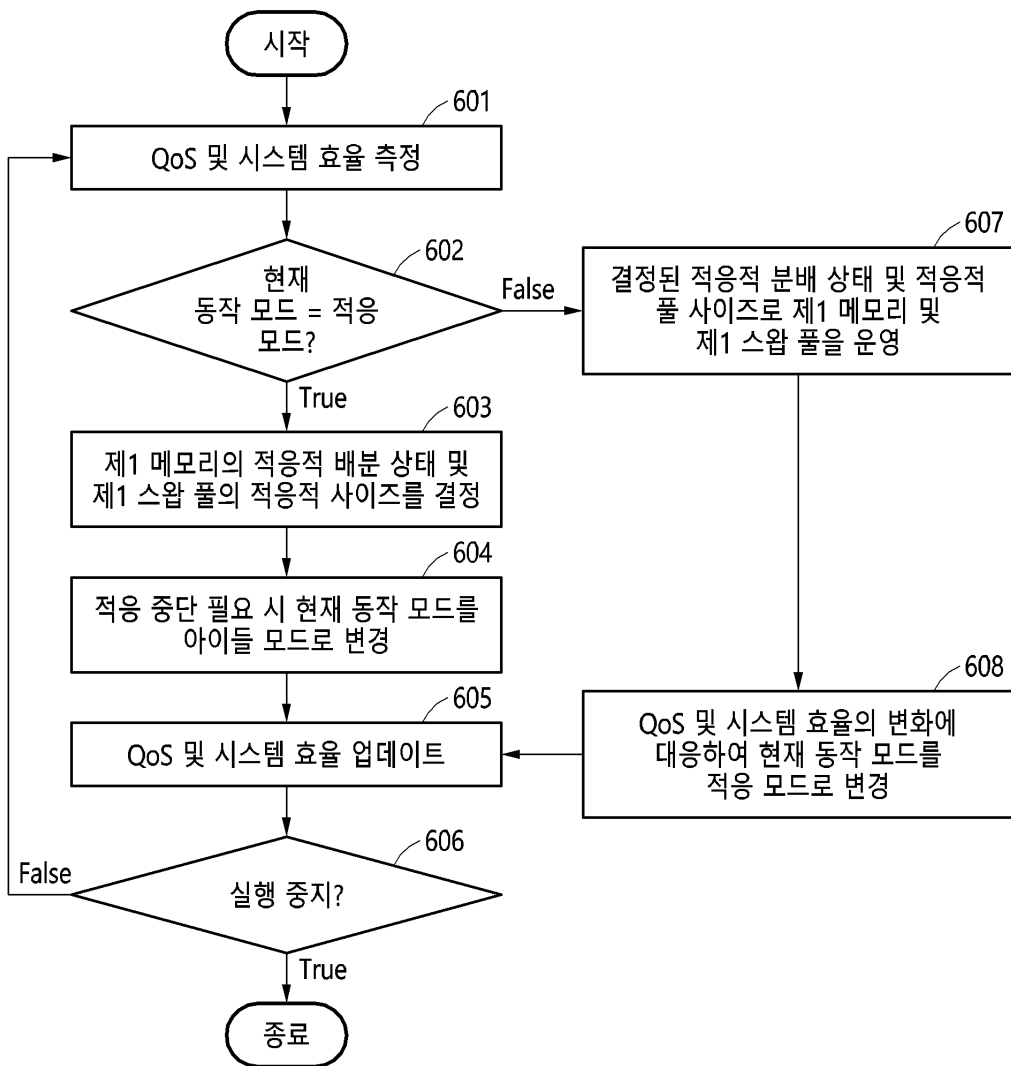
도면4



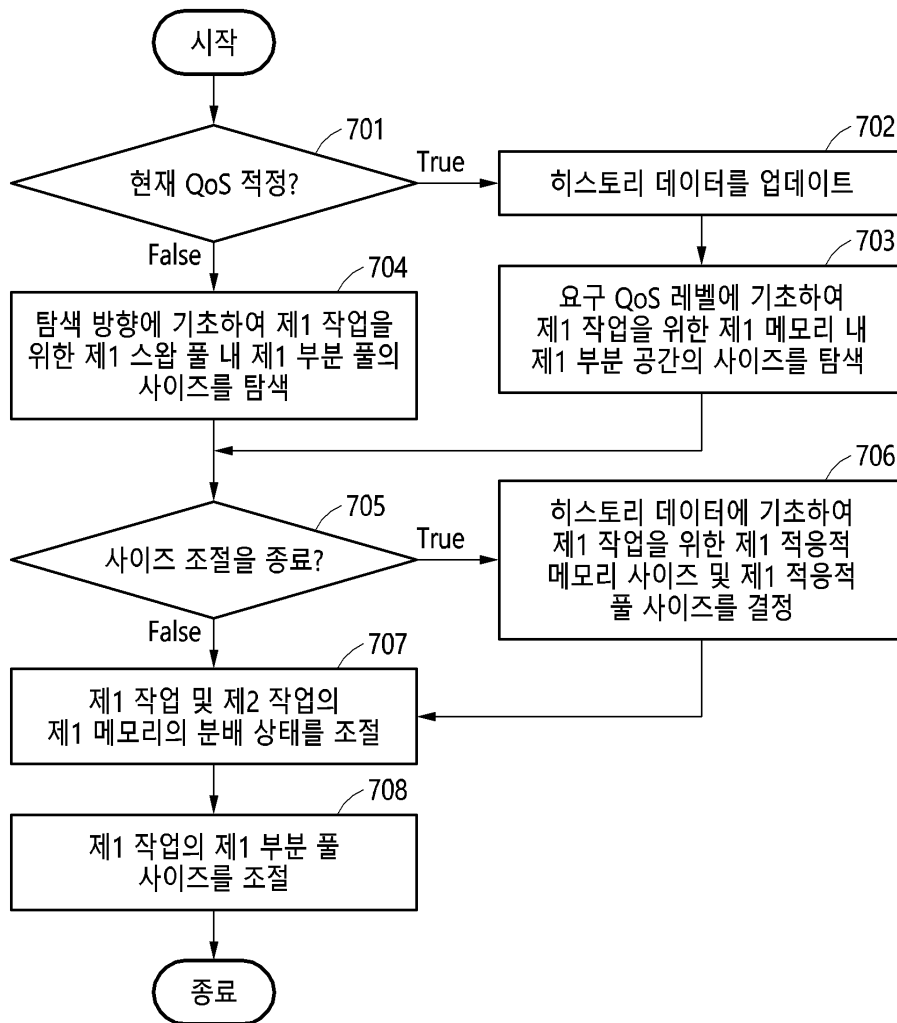
도면5



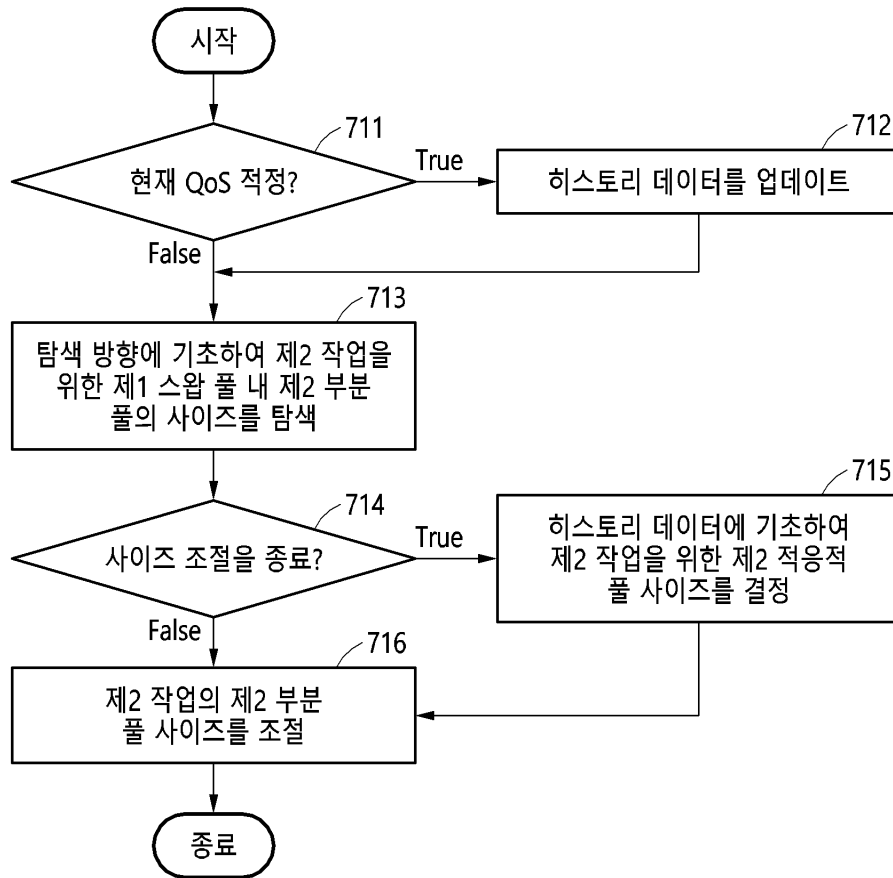
도면6



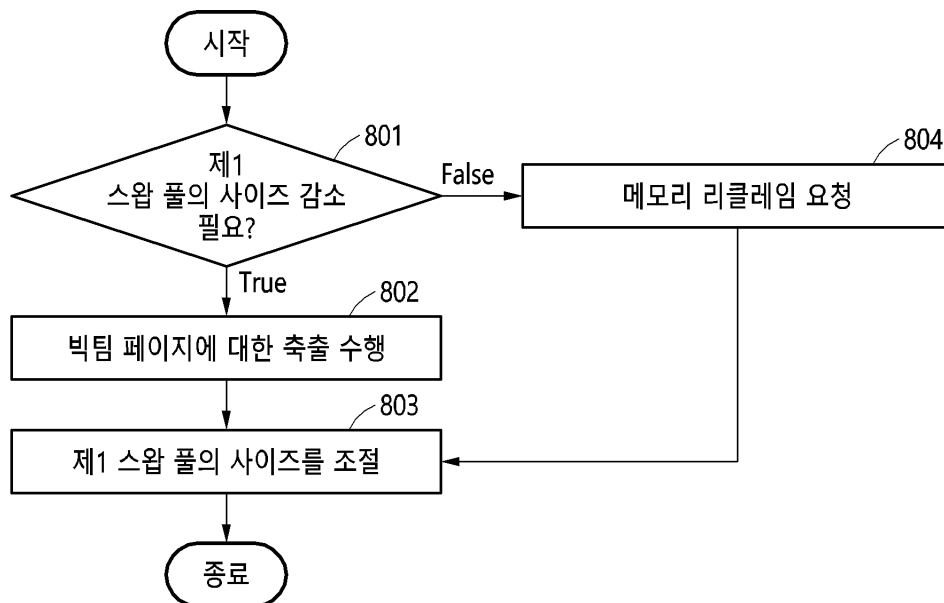
도면7a



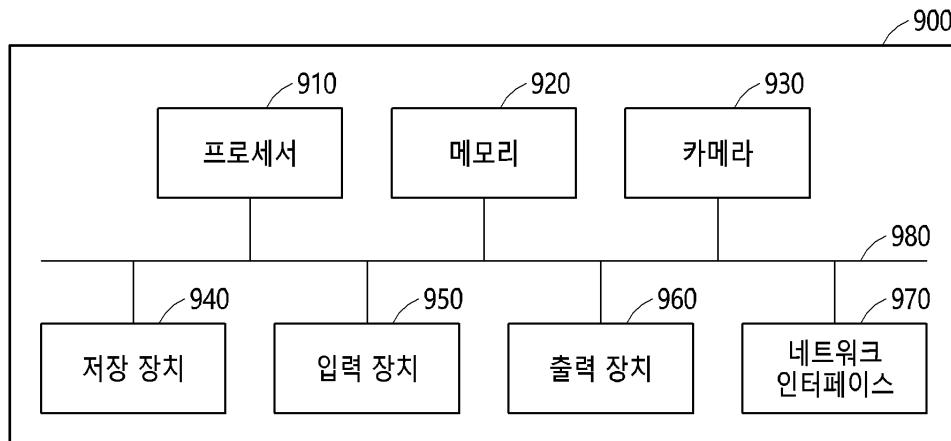
도면7b



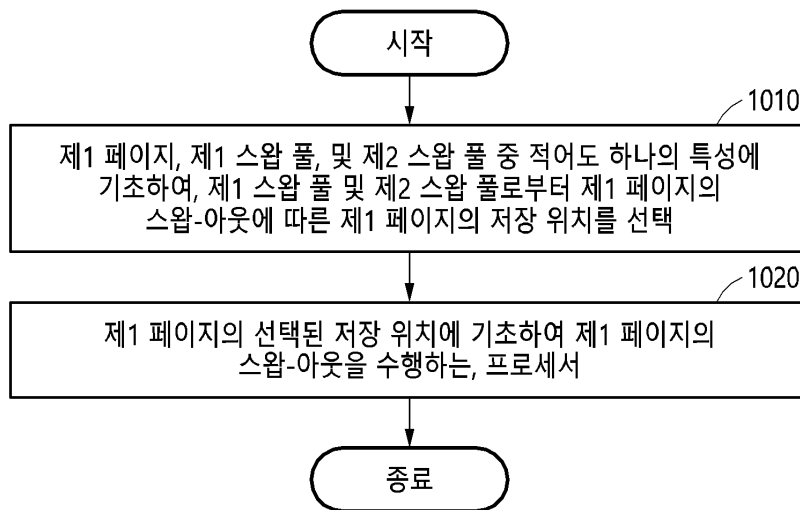
도면8



도면9



도면10



도면11

