



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0124181
(43) 공개일자 2016년10월26일

- (51) 국제특허분류(Int. Cl.)
G06F 12/1009 (2016.01) *G06F 12/02* (2006.01)
G06F 12/08 (2016.01)
- (52) CPC특허분류
G06F 12/1009 (2013.01)
G06F 12/023 (2013.01)
- (21) 출원번호 10-2016-7025623
- (22) 출원일자(국제) 2015년02월18일
심사청구일자 없음
- (85) 번역문제출일자 2016년09월19일
- (86) 국제출원번호 PCT/US2015/016237
- (87) 국제공개번호 WO 2015/175062
국제공개일자 2015년11월19일
- (30) 우선권주장
14/187,031 2014년02월21일 미국(US)

- (71) 출원인
마이크로소프트 테크놀로지 라이센싱, 엘엘씨
미국 위성턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
- (72) 발명자
박 예브게니 엠.
미국 위성턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이던츠
(8/1172) 마이크로소프트 코포레이션 내
이이군 메멧
미국 위성턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이던츠
(8/1172) 마이크로소프트 코포레이션 내
(뒷면에 계속)
- (74) 대리인
김태홍, 김진희

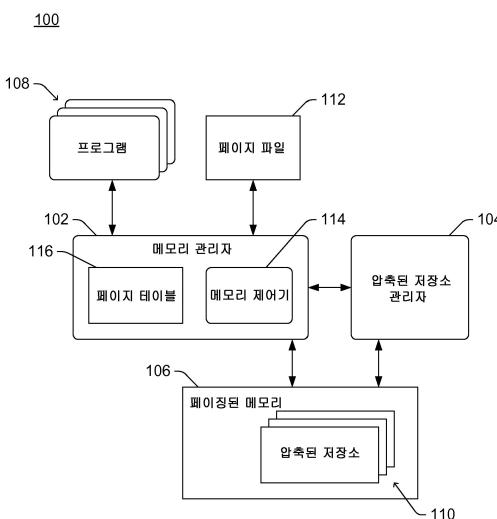
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 수정된 메모리 압축

(57) 요 약

수정된 메모리 페이지를 중 적어도 일부와 같은, 프로그램 프로세스의 작업 세트로부터의 메모리 페이지 세트가 페이지 파일에 써지기 전에 압축된 저장소로 압축되고, 그 후에 메모리 페이지들이 메모리 관리자에 의해 용도 변경될 수 있다. 압축된 저장소는 다수의 메모리 페이지들로 이루어져 있고, 압축된 저장소 메모리 페이지들은 페이지 파일에 써진 후에 메모리 관리자에 의해 용도 변경될 수 있다. 압축된 저장소로 압축된 메모리 페이지들에 대한 메모리 관리자로부터의 차후의 요청은 압축된 저장소 메모리 페이지들에 액세스하는 것(페이지 파일에 써진 경우 페이지 파일로부터 압축된 저장소 메모리 페이지들을 검색하는 것을 포함함), 요청된 메모리 페이지들을 압축 해제하는 것, 및 요청된 메모리 페이지들을 메모리 관리자로 반환하는 것에 의해 이행된다.

대 표 도 - 도1



(52) CPC특허분류

G06F 12/08 (2013.01)

G06F 2212/401 (2013.01)

(72) 발명자

왕 랜디

미국 위성턴주 98052-6399 레드몬드 원 마이크로소

프트 웨이 엘씨에이 - 인터내셔널 페이턴츠

(8/1172) 마이크로소프트 코포레이션 내

키샨 아룬 유.

미국 위성턴주 98052-6399 레드몬드 원 마이크로소

프트 웨이 엘씨에이 - 인터내셔널 페이턴츠

(8/1172) 마이크로소프트 코포레이션 내

명세서

청구범위

청구항 1

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법에 있어서,

수정된 제1 메모리 페이지 세트를 식별하는 단계;

상기 제1 메모리 페이지 세트를 제2 메모리 페이지 세트로 이루어져 있는 압축된 저장소(compressed store)로 압축하는 단계;

상기 제1 메모리 페이지 세트가 상기 압축된 저장소로 압축된 후에 메모리 관리자가 상기 제1 메모리 페이지 세트를 용도 변경할 수 있게 하는 단계; 및

상기 제1 메모리 페이지 세트를 페이지 파일(page file)에 쓰기(write out)보다는 상기 제2 메모리 페이지 세트를 상기 페이지 파일에 쓰는 단계

를 포함하는,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 2

제1항에 있어서,

상기 제2 메모리 페이지 세트가 상기 페이지 파일에 써진 후에 상기 메모리 관리자가 상기 제2 메모리 페이지 세트를 용도 변경할 수 있게 하는 단계를 추가로 포함하는,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 3

제1항에 있어서,

상기 제1 메모리 페이지 세트는 프로그램의 프로세스에 할당되어 있고, 상기 압축된 저장소는 상기 프로그램의 상기 프로세스와만 연관되어 있는 것인,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 4

제3항에 있어서,

상기 프로세스의 메모리 페이지들을 스왑아웃하기(out-swap)로 결정한 것에 응답하여,

상기 프로세스의 작업 세트(working set)에서 메모리 페이지들의 리스트를 식별하는 단계;

상기 식별된 메모리 페이지들을 압축하는 단계;

상기 압축된 식별된 메모리 페이지들을 상기 식별된 메모리 페이지들의 가상 메모리 주소에 의해 순차적 순서로 상기 압축된 저장소에 쓰는 단계;

상기 압축된 저장소를 위한 상기 페이지 파일 내의 공간을 예약하는 단계; 및

상기 압축된 저장소를 상기 페이지 파일 내의 상기 예약된 공간에 쓰는 단계

를 추가로 포함하는,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 5

제4항에 있어서,

상기 프로세스의 상기 메모리 페이지들을 스왑아웃한 후에 상기 프로세스의 메모리 페이지들을 스왑인하기(in-swap)로 결정한 것에 응답하여,

상기 압축된 저장소를 상기 페이지 파일로부터 검색하는 단계;

상기 검색된 압축된 저장소를 제3 메모리 페이지 세트에 저장하는 단계;

상기 제3 메모리 페이지 세트 내의 상기 압축된 메모리 페이지들을 압축 해제하는 단계; 및

상기 프로세스의 상기 작업 세트에 포함시키기 위해 상기 압축 해제된 메모리 페이지들을 반환하는(returning) 단계

를 추가로 포함하는,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 6

제1항에 있어서,

상기 제1 메모리 페이지 세트가 압축된 후에 그리고 상기 제2 메모리 페이지 세트가 상기 페이지 파일에 써졌는지 또는 상기 페이지 파일에 써진 후에 용도 변경되었는지에 관계없이 상기 제1 메모리 페이지 세트를 반환하는데 순방향 진행을 보장하는 단계를 추가로 포함하는,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 7

제1항에 있어서,

상기 제1 메모리 페이지 세트의 각각의 메모리 페이지는 적어도 유사한 우선순위 레벨들을 갖고, 상기 제2 메모리 페이지 세트의 각각의 메모리 페이지는 적어도 유사한 우선순위 레벨들을 갖는 것인,

컴퓨팅 디바이스에서 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키는 컴퓨터 구현 방법.

청구항 8

컴퓨팅 디바이스의 프로그램들에 할당하기 위해 이용가능한 메모리의 양을 증가시키도록 구성된 컴퓨팅 디바이스에 있어서,

메모리;

상기 메모리의 페이지들을 관리하도록 구성된 메모리 관리자; 및

압축된 저장소 관리자(compressed store manager)

를 포함하고,

상기 압축된 저장소 관리자는,

프로세스에 의해 수정된 제1 메모리 페이지 세트를 상기 프로세스와 연관된 압축된 저장소 - 상기 압축된 저장소는 제2 메모리 페이지 세트로 이루어져 있음 - 로 압축하는 동작;

상기 제1 메모리 페이지 세트의 각각의 메모리 페이지에 대해, 상기 메모리 페이지가 상기 압축된 저장소로 압축된 후에 상기 메모리 관리자가 상기 메모리 페이지를 용도 변경할 수 있게 하는 동작; 및

상기 제1 메모리 페이지 세트를 페이지 파일(page file)에 쓰는(write out) 일이 없는 경우 상기 제2 메모리 페이지 세트를 상기 페이지 파일에 쓰는 동작

에 의해 상기 컴퓨팅 디바이스의 상기 프로그램들에 할당하기 위해 상기 메모리 관리자에게 이용가능한 상기 메모리의 양을 증가시키도록 구성된 것인,

컴퓨팅 디바이스.

청구항 9

제8항에 있어서,

상기 압축된 저장소는 상기 프로그램의 상기 프로세스와만 연관되는 것인,

컴퓨팅 디바이스.

청구항 10

제9항에 있어서,

상기 압축된 저장소 관리자는 또한,

상기 프로세스의 메모리 페이지들을 스왑아웃하기로 결정한 것에 응답하여,

상기 프로세스의 작업 세트에서 메모리 페이지들의 리스트를 식별하고,

상기 식별된 메모리 페이지들을 압축하며,

상기 압축된 식별된 메모리 페이지들을 상기 식별된 메모리 페이지들의 가상 메모리 주소에 의해 순차적 순서로
상기 압축된 저장소에 쓰고,

상기 압축된 저장소를 위한 상기 페이지 파일 내의 공간을 예약하며,

상기 압축된 저장소를 인접하게 상기 페이지 파일 내의 상기 예약된 공간에 쓰고;

상기 프로세스의 상기 메모리 페이지들을 스왑아웃한 후에 상기 프로세스의 메모리 페이지들을 스왑인하기로 결정한 것에 응답하여,

상기 압축된 저장소를 상기 페이지 파일로부터 검색하고,

상기 검색된 압축된 저장소를 제3 메모리 페이지 세트에 저장하며,

상기 제3 메모리 페이지 세트 내의 상기 압축된 메모리 페이지들을 압축 해제하고,

상기 프로세스의 상기 작업 세트에 포함시키기 위해 상기 압축 해제된 메모리 페이지들을 반환하도록 구성되는 것인,

컴퓨팅 디바이스.

발명의 설명

배경기술

[0001] 컴퓨터 기술이 진보함에 따라, 컴퓨터 프로세서의 성능이 향상되었다. 그렇지만, 이러한 컴퓨터 프로세서의 성능이 때때로 컴퓨터에서의 다른 병목 현상들에 의해 방해된다. 예를 들어, 하드 디스크 드라이브로부터 RAM(random access memory)으로의 데이터 전송의 속도가 컴퓨터 성능에서의 병목 현상이다. 컴퓨터에서의 병목 현상의 영향을 감소시키는 하나의 방식은 RAM에 보다 많은 데이터를 저장하는 것이다. 그렇지만, 컴퓨터에 아주 많은 양의 RAM을 사용하는 것이 전형적으로 엄청난 비용이 소요될 정도로 RAM의 비용이 여전히 높다.

발명의 내용

[0002] 이 발명의 내용은 이하에서 발명을 실시하기 위한 구체적인 내용에 추가로 기술되는 선택된 개념들을 간략화된 형태로 소개하기 위해 제공된다. 이 발명의 내용은 청구된 발명 요지의 핵심적인 특징들 또는 필수적인 특징들을 언급하기 위한 것이 아니며, 청구된 발명 요지의 범주를 제한하기 위해 사용되기 위한 것도 아니다.

[0003] 하나 이상의 양태들에 따르면, 수정된 제1 메모리 페이지 세트가 식별된다. 제1 메모리 페이지 세트가 제2 메모리 페이지 세트로 이루어져 있는 압축된 저장소(compressed store)로 압축된다. 제1 메모리 페이지 세트가 압축된 저장소로 압축된 후에, 메모리 관리자는 제1 메모리 페이지 세트를 용도 변경(repurpose)할 수 있다. 그에 부가하여, 제1 메모리 페이지 세트를 페이지 파일(page file)에 쓰기(write out)보다는 제2 메모리 페이지

세트가 페이지 파일에 써진다.

[0004] 하나 이상의 양태들에 따르면, 컴퓨팅 디바이스는 메모리, 메모리의 페이지들을 관리하도록 구성된 메모리 관리자, 및 압축된 저장소 관리자(compressed store manager)를 포함한다. 압축된 저장소 관리자는 프로세스에 의해 수정된 제1 메모리 페이지 세트를 프로세스와 연관된 압축된 저장소로 압축하도록 구성되어 있고, 압축된 저장소는 제2 메모리 페이지 세트로 이루어져 있다. 압축된 저장소 관리자는 또한, 제1 메모리 페이지 세트의 각각의 메모리 페이지에 대해, 메모리 관리자가 메모리 페이지가 압축된 저장소로 압축된 후에 메모리 페이지를 용도 변경할 수 있게 하고 제1 메모리 페이지 세트를 페이지 파일에 쓰는 일이 없는 경우 제2 메모리 페이지 세트를 페이지 파일에 쓸 수 있게 하도록 구성되어 있다.

도면의 간단한 설명

[0005] 유사한 특징들을 참조하기 위해 도면들 전체에 걸쳐 동일한 번호가 사용된다.

도 1은 하나 이상의 실시예들에 따른, 수정된 메모리 압축을 이용하는 예시적인 시스템을 나타낸 도면.

도 2는 하나 이상의 실시예들에 따른, 메모리 페이지들의 예시적인 수명 주기를 나타낸 도면.

도 3은 하나 이상의 실시예들에 따른, 페이징된 메모리(paged memory)의 예시적인 레코드를 나타낸 도면.

도 4는 하나 이상의 실시예들에 따른, 메모리 페이지들을 압축하는 것의 일 예를 나타낸 도면.

도 5는 하나 이상의 실시예들에 따른, 메모리 페이지들을 압축하는 예시적인 프로세스를 나타낸 플로우차트.

도 6은 하나 이상의 실시예들에 따른, 압축된 메모리 페이지들을 검색하는 예시적인 프로세스를 나타낸 플로우차트.

도 7은 하나 이상의 실시예들에 따른, 예시적인 메모리 페이지 맵(memory page map)을 나타낸 도면.

도 8은 하나 이상의 실시예들에 따른, 예시적인 영역 맵(region map)을 나타낸 도면.

도 9는 하나 이상의 실시예들에 따른, 작업 세트(working set)를 스왑아웃(out-swapping)하는 것의 일 예를 나타낸 도면.

도 10은 하나 이상의 실시예들에 따른, 작업 세트를 스왑아웃하는 예시적인 프로세스를 나타낸 플로우차트.

도 11은 하나 이상의 실시예들에 따른, 작업 세트를 스왑인(in-swapping)하는 예시적인 프로세스를 나타낸 플로우차트.

도 12는 본원에 기술되는 다양한 기법들을 구현할 수 있는 하나 이상의 컴퓨팅 시스템들 및/또는 디바이스들을 나타내는 예시적인 컴퓨팅 디바이스를 포함하는 예시적인 시스템을 나타낸 도면.

발명을 실시하기 위한 구체적인 내용

[0006] 수정된 메모리 압축이 본원에서 논의된다. 시스템에서의 메모리는 메모리 페이지(또는 간단히 페이지)라고 지칭되는 다수의 블록들 또는 부분들로 이루어져 있다. 수정된 작업 세트에서의 메모리 페이지들 중 적어도 일부와 같은, 프로그램의 작업 세트로부터의 메모리 페이지 세트가 페이지 파일에 써지기 전에 압축된 저장소로 압축된다. 압축된 저장소가 페이지 파일에 써졌는지에 상관없이, 메모리 페이지들이 압축된 저장소로 압축된 후에 메모리 관리자에 의해 용도 변경될 수 있다. 압축된 저장소 자체는 다수의 메모리 페이지들로 이루어져 있고, 압축된 저장소 메모리 페이지들은 페이지 파일에 써진 후에 메모리 관리자에 의해 용도 변경될 수 있다. 압축된 저장소로 압축된 메모리 페이지들에 대한 메모리 관리자로부터의 차후의 요청은 압축된 저장소 메모리 페이지들에 액세스하는 것(페이지 파일에 써진 경우 압축된 저장소 메모리 페이지들을 페이지 파일로부터 검색하는 것을 포함함), 요청된 메모리 페이지들을 압축 해제하는 것, 및 요청된 메모리 페이지들을 메모리 관리자로 반환하는 것에 의해 이행된다.

[0007] 그에 부가하여, 프로그램이 일시 중단되는 때와 같은 특정의 상황에서, 메모리 페이지의 작업 세트가 스왑아웃된다. 메모리 페이지들의 작업 세트를 스왑아웃하는 것은 프로그램의 프로세스의 작업 세트에서 메모리 페이지들의 리스트를 식별하는 것, 식별된 메모리 페이지들을 압축하는 것, 및 압축된 식별된 메모리 페이지들을 식별된 메모리 페이지들의 가상 메모리 주소에 의해 순차적 순서로 압축된 저장소에 쓰는 것을 포함한다. 압축된 저장소를 위한 페이지 파일 내의 공간이 예약되고, 압축된 저장소가 페이지 파일 내의 예약된 공간에 써진다.

[0008] 프로그램이 재개되어야 한다는 차후의 결정에 응답하여, 프로그램에 대한 메모리 페이지들의 작업 세트가 스왑인된다. 메모리 페이지들의 작업 세트를 스왑인하는 것은 압축된 저장소를 페이지 파일로부터 검색하는 것 및 압축된 저장소를 압축된 저장소 메모리 페이지 세트에 저장하는 것을 포함한다. 압축된 저장소 메모리 페이지들이 압축 해제되고, 프로그램의 작업 세트에 포함시키기 위해 메모리 관리자에 반환된다.

[0009] 도 1은 하나 이상의 실시예들에 따른, 수정된 메모리 압축을 이용하는 예시적인 시스템(100)을 나타낸 것이다. 시스템(100)은 메모리 관리자(102), 압축된 저장소 관리자(104), 및 페이징된 메모리(106)를 포함한다. 일반적으로, 메모리 관리자(102)는 페이징된 메모리(106)에 데이터를 저장하는 것을 관리한다. 메모리 관리자(102)는 페이징된 메모리(106)의 부분들을 시스템(100)에서의 다양한 프로그램들(108)에 할당한다. 프로그램(108)은, 실행 중일 때, 본원에서 프로세스(실행 중인 프로그램(108)의 인스턴스임)라고도 지칭되고, 따라서 프로그램(108)에 할당된 페이징된 메모리(106)의 부분들이 프로그램(108)의 프로세스에 할당된다고도 말해진다. 프로세스에 할당된 페이지들은 그 프로세스에 의해 소유되거나 그에 전용되고, 그 프로세스에 의해 사용되며, 다른 프로세스에 의해서는 사용되지 않는다(그리고 따라서 개인 페이지(private page)라고도 지칭됨). 프로그램(108)은 애플리케이션, 운영 체제 프로그램, 또는 다른 커널 또는 모듈일 수 있다. 메모리 관리자(102)는 데이터를 페이징된 메모리(106)로부터 검색하고 데이터를 페이징된 메모리(106)에 쓰라는 요청들을 이 프로그램들(108)로부터 수신한다. 페이징된 메모리(106)는, 예를 들어, 휘발성 메모리(예컨대, RAM) 또는 비휘발성 메모리(예컨대, 플래시 메모리)와 같은, 임의의 유형의 CPU(Central Processing Unit) 주소지정가능 메모리일 수 있다.

[0010] 메모리 관리자(102)는 또한 페이징된 메모리(106)의 하나 이상의 부분들을 압축된 저장소 관리자(104)에 할당한다. 개별적으로 예시되어 있지만, 압축된 저장소 관리자(104)는 메모리 페이지들을 할당하기 위한 프로그램(108)으로서 취급된다. 압축된 저장소 관리자(104)는 데이터를 압축된 형태로 저장하기 위한 하나 이상의 압축된 저장소들(110), 예를 들어, 각각의 프로그램(108)에 대해 하나의 압축된 저장소(110)를 발생시킨다. 시스템(100)은 또한, 메모리 페이지들이 저장될 수 있는 저장 디바이스 상의 파일인, 페이지 파일(112)을 포함한다. 페이지 파일(112)이 저장되는 저장 디바이스는 시스템(100)에서의 보조 저장 디바이스이고, 하나 이상의 실시예들에서, 페이징된 메모리(106)보다 더 느린 읽기/쓰기 시간을 갖지만 그보다 더 큰 저장 용량을 갖는다. 페이지 파일(112)이 저장되는 저장 디바이스는, 예를 들어, 플래시 메모리(예컨대, SSD(solid state disk)) 또는 자기 디스크일 수 있다. 단일의 페이지 파일(112)이 도 1에 예시되어 있지만, 유의할 점은 다수의 페이지 파일들이 시스템(100)에 포함될 수 있다는 것이다.

[0011] 메모리 관리자(102)는, 페이징(paging)을 사용하여, 페이징된 메모리(106)를 관리한다. 메모리 관리자(102)는 메모리(106)(예컨대, RAM)를, 특정의(예컨대, 고정된) 데이터 크기 단위인, 페이지들로 편성한다. 페이징이라는 동작은 데이터가 메모리(106)에 있지 않을 때 데이터를, 시스템(100)에서의 페이지 파일(112)인, 백킹 파일(backing file)로부터 페이지 단위로 읽는 것을 지칭한다. 페이징이라는 동작은 또한 더티(dirty)(수정된) 데이터를 다시 페이지 파일(112)에 페이지 단위로 쓰는 것을 지칭한다. 메모리 페이지는 따라서 페이지 파일 백킹된 메모리 페이지(page file backed memory page)라고도 지칭된다. 이러한 페이징 기법은 본 기술 분야의 통상의 기술자에게 널리 공지되어 있다.

[0012] 메모리 관리자(102)는 메모리 관리자(102)의 기능을 수행하는 동작을 하는 메모리 제어기(114), 및 페이징된 메모리(106)의 메모리 페이지들에 관한 다양한 정보의 레코드인 페이지 테이블(116)을 포함한다. 페이지 테이블(116)은 메모리 페이지들이 임의의 주어진 때에 어디에 저장되어 있는지를 나타내는 정보를 포함한다. 이하에서 더 상세히 논의되는 바와 같이, 메모리 페이지들은 전형적으로 물리 메모리(페이징된 메모리(106))에 또는 압축된 저장소(110)(그 자체가 물리 메모리 또는 페이지 파일(112)에 저장될 수 있음)에 저장되지만, 어떤 상황에서, 압축되지 않은 메모리 페이지들이 페이지 파일(112)에 저장될 수 있다. 물리 메모리에 저장된 메모리 페이지에 대해, 메모리 관리자(102)는 메모리 페이지들에 직접 액세스한다. 압축된 저장소(110)에 저장된 메모리 페이지에 대해, 메모리 관리자(102)는 압축된 저장소 관리자(104)에게 메모리 페이지들을 요청하고, 압축된 저장소 관리자(104)는 적절한 경우 메모리 페이지들을 검색하고 압축 해제하며, 압축 해제된 메모리 페이지들을 메모리 관리자(102)에 반환한다. 압축되지 않은 메모리 페이지가 페이지 파일(112)에 저장되어 있는 경우, 메모리 관리자는 메모리 페이지를 페이지 파일(112)로부터 검색한다.

[0013] 페이징된 메모리(106)는 각각이 임의의 주어진 때에 다수의 상이한 유형의 페이지들 중 하나로서 분류될 수 있는 다수의 페이지들을 포함하고, 이 분류는 시간에 따라 변할 수 있다. 한 유형의 메모리 페이지는 프로그램(108)에 할당되어 있고 프로그램(108)에 의해 현재 사용 중인 메모리 페이지이고, 이 유형의 메모리 페이지는 프로그램(108)의 프로세스의 작업 세트(프로그램(108)의 작업 세트라고도 지칭됨)에서의 작업 세트 페이지

(working set page)(또는 대안적으로, 할당된 페이지(assigned page))라고 지칭된다. 현재 사용 중인 메모리 페이지는 문턱값 양의 시간(예컨대, 이전 20초) 내에 프로그램(108)에 의해 액세스되었거나 메모리 페이지가 메모리 관리자(102)에 의해 용도 변경되어서는 안되는 방식으로 프로그램(108)에 의해 다른 방법으로 액세스되고 있는 메모리 페이지를 말한다. 메모리 페이지를 용도 변경한다는 것은 메모리 페이지가, 동일하거나 상이한 프로그램에 대한 상이한 데이터를 저장하는 것과 같은, 다른 무언가를 위해 또는 시스템에서의 다른 용도를 위해 메모리 관리자(102)에 의해 재사용된다는 것을 말한다.

[0014] 다른 유형의 메모리 페이지는 프로그램(108)에 할당되어 있고 프로그램(108)에 의해 현재 사용 중이 아닌 메모리 페이지이고, 이 유형의 메모리 페이지는 대기 페이지(standby page)(또는 대안적으로, 재할당가능 페이지(re-assignable page))라고 지칭된다. 현재 사용 중이 아닌 메모리 페이지는 내용이 수정되지 않았고(마지막 수정 이후에 페이지 파일(112) 또는 압축된 저장소(110)에 써졌고) 문턱값 양의 시간(예컨대, 이전 20초) 내에 프로그램(108)에 의해 액세스되지 않았거나 메모리 페이지가 메모리 관리자(102)에 의해 용도 변경될 수 있는 방식으로 프로그램(108)에 의해 다른 방법으로 액세스되고 있는 페이지를 말한다. 메모리 페이지는, 예를 들어, 상이한 유형의 페이지(예컨대, 작업)에 할당되는 것, 상이한 프로그램에 할당되는 것 등에 의해 용도 변경될 수 있다.

[0015] 다른 유형의 메모리 페이지는 내용이 수정되었지만 아직 페이지 파일(112)에 복사되지 않은 메모리 페이지이고, 이 유형의 메모리 페이지는 수정된 페이지(또는 대안적으로, 더티 페이지)라고 지칭된다. 수정된 메모리 페이지는 메모리 관리자(102)에 의해 용도 변경되어서는 안된다. 그렇지만, 이하에서 보다 상세히 논의되는 바와 같이, 수정된 메모리 페이지를 압축하고 압축된 저장소(110)에 저장한 후에, 메모리 페이지의 분류가 (예컨대, 대기 페이지 또는 사용가능 페이지(free page)로) 변경될 수 있다.

[0016] 다른 유형의 메모리 페이지는 프로그램(108)에 현재 할당되어 있지 않고 메모리 관리자(102)가 그에 대해 무엇이든(용도 변경되는 것을 포함함) 할 수 있는 메모리 페이지이다. 이 유형의 메모리 페이지는 사용가능 페이지(또는 대안적으로, 할당 해제된 페이지(de-allocated page))라고 지칭된다.

[0017] 메모리 관리자(102)는 페이지된 메모리(106)의 메모리 페이지들이 언제 프로그램(108)에 할당되는지, 어느 메모리 페이지들이 어느 프로그램들(108)에 할당되는지, 프로그램(108)에 이전에 할당된 어느 메모리 페이지들이 용도 변경되어야 하는지 등을 결정하기 위해 다양한 규칙들 또는 기준들을 사용한다. 메모리 관리자(102)는 또한 메모리 페이지의 유형에 대한 분류를 결정하고, 적절한 경우 메모리 페이지의 분류를 변경한다. 하나 이상의 실시예들에서, 메모리 관리자(102)는 페이지된 메모리(106)의 어느 메모리 페이지들이 어느 유형의 페이지로서 분류되는지의 리스트 또는 다른 레코드를 유지한다. 예를 들어, 메모리 관리자(102)는 작업 세트 페이지들의 리스트 또는 다른 레코드, 대기 페이지들의 리스트 또는 다른 레코드, 수정된 페이지들의 리스트 또는 레코드, 사용가능 페이지들의 리스트 또는 레코드 등을 유지할 수 있다.

[0018] 시스템(100)은 하나 이상의 디바이스들을 나타낸다. 하나 이상의 실시예들에서, 시스템(100)에 예시된 컴포넌트들(102 내지 108)은 동일한 컴퓨팅 디바이스의 일부로서 포함되어 있다. 대안적으로, 컴포넌트들은 2 개 이상의 디바이스들에 걸쳐 분산되어 있을 수 있다.

[0019] 도 2는 하나 이상의 실시예들에 따른, 메모리 페이지들의 예시적인 수명 주기(200)를 나타낸 것이다. 메모리 페이지가 프로그램에 할당되고, 프로그램의 작업 세트(202)에 포함될 수 있다. 프로그램의 작업 세트(202)는 프로그램의 작업 세트 페이지들을 포함한다. 메모리 관리자(102)는 다양한 때에 그리고, 메모리 페이지들을 다른 프로그램들이 이용할 수 있게 만들기 위한 것과 같은, 다양한 이유로 프로그램의 작업 세트(202) 내의 메모리 페이지들을 트리밍(trim)(축소(reduce)라고도 지칭됨)할 수 있다. 작업 세트(202)에 써지고(수정되고) 그로부터 트리밍된 메모리 페이지들은 메모리 페이지들의 수정된 리스트(206)로 이동(204)된다.

[0020] 압축된 저장소 관리자(104)는 수정된 리스트(206) 내의 메모리 페이지들을 압축하고 압축된 메모리 페이지들을 압축된 저장소(110)의 메모리 페이지들에 추가(208)한다. 압축된 저장소 관리자(104)는, 규칙적인 또는 불규칙적인 간격으로 메모리 페이지들을 압축하는 것, 적어도 문턱 개수의 메모리 페이지들이 수정된 리스트에 포함된 것에 응답하여 메모리 페이지들을 압축하는 것, 메모리 관리자(102)로부터의 요청에 응답하여 메모리 페이지들을 압축하는 것, 프로세스의 메모리 페이지들이 이하에서 보다 상세히 논의되는 바와 같이 스왑아웃되어야 한다는 결정에 응답하여 메모리 페이지들을 압축하는 것 등과 같이, 수정된 리스트 내의 메모리 페이지들을 언제 압축해야 하는지의 타이밍을 상이한 방식으로 결정할 수 있다. 압축되어 압축된 저장소(110)에 추가된 후에, 수정된 리스트(206) 내의 메모리 페이지들은 대기 리스트(212)로 이동(210)된다. 대기 리스트(212)는 대기 페이지들의 리스트이지만, 메모리 페이지들이 대안적으로 사용가능 페이지들의 리스트로 이동될 수 있다.

- [0021] 압축된 저장소(110) 자체는 다수의 메모리 페이지(본원에서 압축된 저장소 메모리 페이지라고도 지칭됨)로 이루어져 있고, 이 압축된 저장소 메모리 페이지들은 원하는 바에 따라 압축된 저장소(110) 또는 메모리 관리자(102)에 의해 수정된 리스트(206)로 이동(214)될 수 있다. 그렇지만, 압축된 저장소 메모리 페이지들은 원하는 바에 따라 오랜 시간 동안(어쩌면 무기한으로) 수정된 리스트로 이동될 필요는 없다. 압축된 저장소 메모리 페이지들이 압축된 저장소의 작업 세트에 남아 있을 수 있다.
- [0022] 압축된 저장소 메모리 페이지들이 수정된 리스트(206)로 이동되기 위해, 메모리 관리자(102)는 압축된 저장소 메모리 페이지들을 페이지 파일(112)에 쓴다(216). 이와 같이, 작업 세트(202)로부터의 메모리 페이지는, 압축되어 압축된 저장소 메모리 페이지에 저장된 후에, 페이지 파일(112)에 써질 수 있다. 유의할 점은 압축된 저장소 관리자(104)가 수정된 리스트(206) 상의 압축된 저장소 메모리 페이지들을 압축하려고 시도하지 않는다는 것이다. 압축된 저장소 관리자(104)는, 메모리 페이지들이 할당되어 있는 프로세스에 기초하여(예컨대, 압축된 저장소 관리자(104)와 같은 시스템 프로세스에 할당되는 경우, 메모리 페이지들을 압축하려는 어떤 시도도 행해지지 않음), 압축된 저장소 메모리 페이지들의 레코드를 유지하는 것에 의해, 기타와 같은 상이한 방식들로, 압축하려고 시도해서는 안되는 메모리 페이지들을 식별할 수 있다.
- [0023] 수정된 리스트(206)로 이동되는 작업 세트(202)로부터의 메모리 페이지가 예상된 양의 시간 내에(예컨대, 문턱값 양의 시간 내에) 압축될 수 없거나 압축된 저장소(110)의 메모리 페이지에 다른 방식으로 추가될 수 없는 상황, 또는 시스템(100)이 다양한 정책 이유들로 메모리 페이지를 압축하지 않기로 선택하는 상황이 발생할 수 있다. 시스템(100)에서의 과도한 메모리 사용, 시스템(100)에서의 과도한 프로세서 사용, 시스템(100)에 적용되는 다양한 정책들 등으로 인해, 이러한 상황이 가끔씩 발생할 수 있다. 이러한 상황에서, 메모리 관리자(102)는 압축되지 않은 메모리 페이지를 페이지 파일(112)에 쓴다(218). 이러한 메모리 페이지들은 따라서 압축된 저장소(110)의 메모리 페이지에 포함되지 않지만, 메모리 관리자(102)는 이러한 메모리 페이지들이 압축된 저장소 관리자(104)보다는 페이지 파일(112)로부터 이용가능하다는 레코드를 (예컨대, 페이지 테이블(116)에) 유지한다.
- [0024] 도 1로 돌아가서, 페이지된 메모리(106)의 페이지들 각각은 대응하는 우선순위 레벨을 갖는다. 특정 유형의 메모리 페이지들만이 우선순위 레벨을 가질 수 있거나, 대안적으로 모든 유형의 메모리 페이지들이 우선순위 레벨을 가질 수 있다. 메모리 관리자(102)는 각각의 메모리 페이지에 대한 우선순위 레벨을 설정하고, 메모리 페이지에 대한 우선순위 레벨을 각종의 상이한 방식으로 결정할 수 있다. 하나 이상의 실시예들에서, 메모리 페이지에 대한 우선순위 레벨은 운영 체제에 의해 메모리 페이지의 할당을 요청한 프로세스 또는 스레드에 할당된 우선순위 레벨이다. 프로그램(108)으로부터의 요청, 메모리 페이지가 특정 유형의 검색(예컨대, 페이지들의 투기적 검색(speculative retrieval))과 연관되어 있는 것 등과 같은, 다른 기준들이 또한 메모리 페이지에 대한 우선순위 레벨을 결정하는 데 사용될 수 있다. 메모리 관리자(102)는 또한 시간에 따라 메모리 페이지들의 우선순위 레벨을 변경할 수 있다(예컨대, 문턱값 양의 시간 동안 프로그램(108)에 의해 액세스되지 않는 메모리 페이지들의 우선순위 레벨을 낮추는 것, 페이지 파일(112)에 써졌다가 대기 리스트(212)로 이동된 압축된 저장소 메모리 페이지들의 우선순위 레벨을 낮추는 것 등).
- [0025] 우선순위 레벨이 시스템(100)에서 다양한 방식으로 사용될 수 있다. 하나 이상의 실시예들에서, 메모리 페이지들이 그들의 우선순위 레벨에 기초하여 페이지 파일(112)에 써진다. 보다 낮은 우선순위인, 페이지 파일(112)에 써질 메모리 페이지들은 보다 높은 우선순위인 메모리 페이지들보다 앞서 페이지 파일에 써진다. 이것은 보다 높은 우선순위의 메모리 페이지들이 보다 낮은 우선순위의 메모리 페이지들보다 더 오래 물리 메모리에 남아 있게 할 수 있고, 보다 높은 우선순위의 메모리 페이지들이 보다 낮은 우선순위의 메모리 페이지들보다 먼저 페이지 파일에 써지는(그리고 물리 메모리에 더 이상 없는) 우선순위 역전 시나리오를 방지한다. 이와 유사하게, 하나 이상의 실시예들에서, 메모리 페이지들이 그들의 우선순위 레벨에 기초하여 대기 리스트로부터 용도 변경된다. 보다 낮은 우선순위인 대기 리스트 상의 메모리 페이지들은 보다 높은 우선순위인 메모리 페이지들보다 앞서 용도 변경된다. 이것은 보다 높은 우선순위의 메모리 페이지들이 보다 낮은 우선순위의 메모리 페이지들보다 더 오래 대기 리스트 상에 남아 있을 수 있게 한다.
- [0026] 도 3은 하나 이상의 실시예들에 따른, 페이지된 메모리의 예시적인 레코드(300)를 나타낸 것이다. 다수의(X 개의) 우선순위 레벨들(302)의 세트가 예시되어 있고, 우선순위 레벨들(302)의 세트는 페이지들에 대응할 수 있는 우선순위 레벨들을 포함한다. 페이지된 메모리의 레코드(300)는, 예를 들어, 도 1의 페이지된 메모리(106)의 수정된 페이지들의 레코드이다. 보다 큰 숫자를 갖는 우선순위 레벨은 보다 작은 숫자를 갖는 우선순위 레벨보다 더 높은 우선순위일 수 있거나(예컨대, 우선순위 레벨 4는 우선순위 레벨 1보다 더 높은 우선순위일 수 있음), 그 반대일 수 있다. 대안적으로, 우선순위 레벨을 식별하기 위해, 글자 또는 다른 문자와 같은, 숫자

이외의 라벨이 사용될 수 있다.

[0027] 각각의 우선순위 레벨(302)에 대해, 레코드(300)는 그 우선순위 레벨을 가지는 0 개 이상의 메모리 페이지들의 세트를 식별할 수 있다. 예를 들어, 페이지들(302)은 우선순위 레벨 1에 대응하고, 페이지들(306)은 우선순위 레벨 3에 대응하며, 이하 마찬가지이다. 레코드(300)는, 특정의 우선순위 레벨에 대응하는 메모리 페이지들을 식별해주는 리스트 또는 다른 데이터 구조, 상이한 우선순위 레벨에 대응하는 메모리 페이지들에 대해 상이한 저장 구역(예컨대, 상이한 캐시)을 사용하는 것 등과 같은, 각종의 상이한 방식들 중 임의의 방식으로 유지될 수 있다.

[0028] 도 1로 돌아가서, 압축된 저장소(110)는 메모리 관리자(102)에 의해 압축된 저장소 관리자(104)에 할당된 하나 이상의 메모리 페이지들로서 구현된다. 이 메모리 페이지들은 다양한 기능을 제공하는 압축된 저장소 관리자(104)의 사용자 모드 가상 주소 공간에 있는 메모리 페이지이다. 압축된 저장소(110)를 구현하는 메모리 페이지들은 커널 가상 주소 공간을 사용하지 않으며, 권한이 없는 사용자 모드 프로그램이 메모리 페이지 상의 데이터에 액세스할 수 없기 때문에 데이터 노출을 방지하기 위해 보안을 제공한다. 게다가, 압축된 저장소 관리자(104)의 사용자 모드 가상 주소 공간에 있는 메모리 페이지들을 사용하는 것에 의해, 메모리 관리자(102) 및 압축된 저장소 관리자(104)는 기존의 메모리 관리자 API(application programming interface)들을 사용하여 메모리 페이지들을 제어하여, 압축된 저장소 관리자(104)가 원하는 바에 따라 그의 백킹 데이터(backing data)의 페이징 가능성(pageability), 스왑 능력(swapping capabilities) 등을 제어할 수 있게 한다.

[0029] 하나 이상의 실시예들에서, 압축된 저장소 관리자(104)는 페이징된 메모리(106)의 페이지들을 압축하고 압축된 페이지들을 압축된 저장소(110)의 영역에 저장하는데, 이는 메모리 페이지들을 압축된 저장소(110)로 또는 압축된 저장소(110)의 영역으로 압축하는 것이라고도 지칭된다. 압축된 저장소 관리자(104)는 각각의 프로그램(108)에 대한(그리고 그와 연관된) 별개의 압축된 저장소(110)를 발생시키고, 특정의 프로그램(108)으로부터의 수정된 리스트(206)로부터의 압축된 메모리 페이지들을 특정의 프로그램과 연관된 압축된 저장소(110)의 영역들에 추가한다. 대안적으로, 압축된 저장소 관리자(104)는 압축된 저장소들(110)을 각각의 프로그램(108)에 대한 하나의 압축된 저장소(110)와 상이한 세분성으로 발생시키고 유지할 수 있다(예컨대, 압축된 저장소 관리자(104)는 프로그램들(108)의 임의의 세트에 대해 또는 시스템 내의 페이징 가능한 메모리 모두에 대해 압축된 저장소들(110)을 발생시키고 유지할 수 있다).

[0030] 압축된 저장소(110)의 각각의 영역은 메모리 관리자(102)에 의해 압축된 저장소 관리자(104)에 할당되는 페이징된 메모리(106)의 다수의 메모리 페이지들로 이루어져 있다. 예를 들어, 영역은 128 킬로바이트일 수 있고 메모리 페이지들 각각은 4 킬로바이트일 수 있지만, 이 값들은 예이고 다른 영역 및/또는 페이지 크기들이 대안적으로 사용될 수 있다. 특정의 영역으로 압축되고 있는 페이징된 메모리(106)의 페이지들은 적어도 유사한 우선순위 레벨들을 가지며, 페이징된 메모리(106)의 영역을 이루고 있는 메모리 페이지들은 그 영역으로 압축된 메모리 페이지들과 적어도 유사한 우선순위 레벨들을 가진다. 적어도 유사한 우선순위 레벨들을 가지는 메모리 페이지들은 동일하거나 유사한 우선순위 레벨들을 가지는 메모리 페이지들을 지칭한다.

[0031] 유의할 점은 압축된 저장소 관리자(104) 또는 압축된 저장소(110)에 의해 수행되는 것으로 본원에서 논의되는 동작들이 압축된 저장소 관리자(104) 및 압축된 저장소(110) 중 하나 또는 둘 다에 의해 수행될 수 있다는 것과, 압축된 저장소 관리자(104)에 의해 수행되는 것으로 본원에서 논의되는 동작들이 대안적으로 압축된 저장소(110)에 의해 수행될 수 있다는 것과, 압축된 저장소(110)에 의해 수행되는 것으로 본원에서 논의되는 동작들이 대안적으로 압축된 저장소 관리자(104)에 의해 수행될 수 있다는 것이다. 하나 이상의 실시예들에서, 압축된 저장소 관리자(104)는 각각의 압축된 저장소(110)를 관리하는 일, 읽기 및 쓰기 요청을 적절한 압축된 저장소(110)로 라우팅하는 일, 압축된 저장소(110)에 대한 메모리를 관리하는 일(예컨대, 메모리 페이지들이 압축된 저장소(110)에 할당되도록 요청하는 일) 등을 맡고 있다. 이러한 상황에서, 각각의 압축된 저장소(110)는 단순히 메모리 페이지 세트일 수 있다. 대안적으로, 압축된 저장소(110)에 대한 메모리를 관리하는(예컨대, 페이징된 메모리(106)의 페이지 세트가 메모리 관리자(102)에 의해 압축된 저장소(110)에 할당되도록 요청하는) 제어 및 논리의 적어도 일부는 압축된 저장소(110) 자체에 의해 구현될 수 있다. 이러한 상황에서, 압축된 저장소(110)는 하나 이상의 메모리 페이지들은 물론, 압축된 저장소(110)에 대한 메모리를 관리하는 제어 및 논리를 구현하는 압축된 저장소 모듈을 포함한다.

[0032] 하나 이상의 실시예들에서, 압축된 저장소(110)의 특정의 영역으로 압축되고 있는 페이징된 메모리(106)의 페이지들은 동일한 우선순위 레벨을 가지며, 상이한 우선순위 레벨들에 대해 상이한 영역들이 사용된다. 압축된 저장소(110)의 한 영역을 이루고 있는 메모리 페이지들은 그 영역으로 압축되는 메모리 페이지들과 동일한 우선순

위 레벨을 갖는다. 예를 들어, 압축된 저장소 관리자(104)는 도 3의 페이지들(304)을, 압축된 저장소 관리자(104)에 할당된 메모리 페이지들로 이루어져 있고 우선순위 레벨 1을 가지는 하나의 영역으로 압축할 수 있다. 다른 예로서, 압축된 저장소 관리자(104)는 도 3의 페이지들(306)을, 압축된 저장소 관리자(104)에 할당된 메모리 페이지들로 이루어져 있고 우선순위 레벨 3을 가지는 다른 영역으로 압축할 수 있다. 유의할 점은, 각각의 우선순위 레벨에 대해, 압축된 저장소 관리자(104)가 그 우선순위 레벨에 대응하는 메모리 페이지들로 이루어져 있는 0 개 이상의 영역들을 생성할 수 있다는 것이다.

[0033] 대안적으로, 압축된 저장소(110)의 특정의 영역으로 압축되고 있는 페이지된 메모리(106)의 페이지들은 유사하지만 꼭 동일할 필요는 없는 우선순위 레벨들을 가질 수 있다. 어느 우선순위 레벨들이 유사한지는, 우선순위 레벨들이 서로로부터 문턱값 개수(예컨대, 1 개 또는 2 개)의 레벨들 내에 있는지에 기초하는 것과 같은, 상이한 방식들로 결정될 수 있다. 예를 들어, 우선순위 레벨 1과 우선순위 레벨 2는 유사할 수 있고 따라서 우선순위 레벨 1의 페이지들과 우선순위 레벨 2의 페이지들이 동일한 영역으로 압축될 수 있지만, 우선순위 레벨 1과 우선순위 레벨 5는 유사하지 않을 수 있고 따라서 우선순위 레벨 1의 페이지들과 우선순위 레벨 5의 페이지들이 동일한 영역으로 압축되지 않는다.

[0034] 이와 유사하게, 페이지된 메모리(106)의 한 영역을 이루고 있는 메모리 페이지들은 그 영역으로 압축되는 메모리 페이지들과 유사하지만 꼭 동일할 필요는 없는 우선순위 레벨들을 가질 수 있다. 예를 들어, 압축된 저장소(110)의 한 영역을 이루고 있는 메모리 페이지들은 우선순위 레벨 3을 가질 수 있고, 그 영역으로 압축되는 메모리 페이지들은 우선순위 레벨 2 또는 우선순위 레벨 3을 가질 수 있다. 게다가, 영역이 동일하거나 유사한 우선순위 레벨의 메모리 페이지들로 이루어져 있을 수 있다. 예를 들어, 영역이 우선순위 레벨 2를 가지는 몇몇 메모리 페이지들과 우선순위 레벨 3을 가지는 몇몇 메모리 페이지들로 이루어져 있을 수 있다.

[0035] 영역을 생성하기 위해, 압축된 저장소 관리자(104)는 페이지된 메모리(106)의 페이지 세트가 메모리 관리자(102)에 의해 관리자(104)에 할당되도록 요청한다. 할당된 페이지 세트는 작업 세트 페이지로서 분류되고, 압축된 저장소 관리자(104)는 할당된 페이지 세트를 특정의 압축된 저장소(110)와 연관시킨다. 압축된 저장소 관리자(104)는 수정된 리스트(206) 상의 메모리 페이지들을 압축하고 그 압축된 페이지들을 압축된 저장소 관리자(104)에 할당된 페이지 세트에 저장한다.

[0036] 이와 같이, 예를 들어, 특정의 페이지 세트가 적어도 유사한 우선순위 레벨들에 대응하는 부가의 페이지 세트로 압축된다. 그 특정의 페이지 세트가 압축된 후에, 부가의 페이지 세트도 수정된 페이지로서 분류된다. 이와 같이, 압축된 저장소 관리자(104)는 그 특정의 페이지 세트를 동일한 유형의(수정된) 그리고 적어도 유사한 우선순위 레벨들에 대응하는 상이한 페이지 세트로 압축한다. 그렇지만, 압축된 저장소 관리자(104)가 그 특정의 페이지 세트를 압축하기 때문에, 그 특정의 페이지 세트 내의 페이지들의 양이 전형적으로 압축된 페이지 세트 내의 페이지들의 양보다 더 크다. 압축된 페이지 세트가 이어서 페이지 파일(112)에 써질 수 있다. 압축된 페이지 세트가 언제 페이지 파일(112)에 써지는지(또는 압축된 페이지 세트가 페이지 파일(112)에 써지는지)를 결정하기 위해 각종의 상이한 정책들이 적용될 수 있다.

[0037] 도 4는 하나 이상의 실시예들에 따른, 메모리 페이지들을 압축하는 것의 일 예를 나타낸 것이다. 페이지된 메모리(106)는, 402에 예시된 바와 같이, 동일한 우선순위 레벨(예컨대, 앞서 논의된 바와 같이, 우선순위 레벨 3)을 가지는 5 개의 메모리 페이지들(306)을 포함한다. 압축된 저장소 관리자(104)는 메모리 페이지들(306)과 적어도 유사한 우선순위 레벨들을 가지는 2 개의 메모리 페이지들(404)을 포함하는 압축된 저장소의 영역을 할당받는다. 압축된 저장소 관리자(104)는, 406에 예시된 바와 같이, 메모리 페이지들(306)을 메모리 페이지들(404)로 압축한다. 압축 후에, 메모리 페이지들(306)의 압축된 버전이 페이지된 메모리(106)에 메모리 페이지들(404)로서 저장되기 때문에, 메모리 관리자(102)는 메모리 페이지들(306)을 용도 변경할 수 있다. 이와 같이, 압축 후에, 페이지된 메모리(106)가 메모리 페이지들(404)은 저장하지만 메모리 페이지들(306)은 저장할 필요가 없다.

[0038] 도 5는 하나 이상의 실시예들에 따른, 메모리 페이지들을 압축하는 예시적인 프로세스(500)를 나타낸 플로우차트이다. 프로세스(500)는, 도 1의 압축된 저장소 관리자(104)와 같은, 압축된 저장소 관리자에 의해 수행되며, 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합으로 구현될 수 있다. 프로세스(500)가 한 세트의 동작들로서 도시되어 있고, 다양한 동작들의 작업들을 수행하기 위한 도시된 순서로 제한되지 않는다. 프로세스(500)는 메모리 페이지들을 압축하는 예시적인 프로세스이며; 상이한 도면들을 참조한 메모리 페이지들을 압축하는 것에 대한 부가의 논의가 본원에 포함되어 있다.

[0039] 프로세스(500)에서, 수정된 리스트 상의 하나 이상의 메모리 페이지들의 세트가 식별된다(동작(502)). 하나 이

상의 메모리 페이지들의 세트가, 메모리 페이지들이 수정된 리스트에 추가되는 순서로 식별되는 것, 페이지들의 우선순위 레벨에 기초하는 것(예컨대, 보다 낮은 우선순위의 페이지들이 보다 높은 우선순위의 페이지들보다 먼저 식별됨) 등과 같은, 다양한 방식으로 식별될 수 있다.

[0040] 동작(502)에서 식별된 메모리 페이지 세트가 페이지정된 메모리의 압축된 저장소의 한 영역으로 압축된다(동작(504)). 압축된 저장소의 이 영역은 동작(502)에서 식별된 메모리 페이지 세트와 적어도 유사한 우선순위 레벨들을 가지는 하나 이상의 압축된 저장소 메모리 페이지들로 이루어져 있다. 동작(504)에서의 압축의 일부로서, 압축된 저장소 관리자는 메모리 페이지 세트의 각각의 메모리 페이지의 압축된 버전이 그 영역에서의 어디에 저장되는지의 레코드를 유지한다. 이 레코드는, 메모리 페이지의 각각의 압축된 버전에 대해, 어느 하나 이상의 압축된 저장소 메모리 페이지들이 메모리 페이지의 압축된 버전을 저장하는지의 레코드, 메모리 페이지의 각각의 압축된 버전에 대해, 메모리 페이지의 압축된 버전을 저장하는 영역에서의 주소 범위의 레코드 등과 같은, 다양한 형태들을 취할 수 있다.

[0041] 메모리 페이지를 압축한다는 것은 메모리 페이지의 내용(예컨대, 메모리 페이지에 저장된 데이터)을 압축하는 것을 말한다. 각종의 상이한 공개 및/또는 독점적 압축 기법들 중 임의의 것이 메모리 페이지를 압축하는 데 사용될 수 있다. 예를 들어, 메모리 페이지가 런 길이(run-length) 인코딩 압축 알고리즘, LZW 압축 알고리즘, 허프만 코딩 알고리즘 등을 사용하여 압축될 수 있다. 상이한 유형의 콘텐츠에 대한 상이한 압축 알고리즘들과 같은, 다수의 상이한 압축 알고리즘이 선택적으로 이용될 수 있다. 예를 들어, 영숫자 및/또는 텍스트 콘텐츠에 대해 하나의 압축 알고리즘이 사용될 수 있고, 영상 콘텐츠에 대해 다른 압축 알고리즘이 사용될 수 있으며, 오디오 콘텐츠에 대해 다른 압축 알고리즘이 사용될 수 있다. 메모리 페이지를 압축하는 데 사용되는 압축 기법은 전형적으로 무손실 압축 알고리즘이지만, 특정의 상황에서(예컨대, 특정의 영상 및/또는 오디오 콘텐츠에 대해) 손실 압축 알고리즘이 대안적으로 사용될 수 있다. 특정의 압축 알고리즘이 또한 디바이스 의존적일 수 있고, 여기서 보다 빠른 디바이스(예컨대, 보다 강력한 하나 이상의 프로세서들을 가짐)는 보다 잘 압축하는, 프로세서를 보다 많이 사용하는 압축 알고리즘(more processor-intensive compression algorithm)으로 구성될 수 있는 반면, 보다 느린 디바이스(예컨대, 덜 강력한 하나 이상의 프로세서들을 가짐)는 그만큼 잘 압축하지 못하는, 프로세서를 보다 적게 사용하는 알고리즘(less processor-intensive algorithm)으로 구성될 수 있다. 특정의 압축 알고리즘이 또한 저장 디바이스 성능과의 트레이드오프에 기초할 수 있다. CPU가 저장 디바이스보다 대폭 성능이 우수한 경우, 더 잘 압축하는(그리고 따라서 보다 적은 저장 디바이스 I/O를 유발시키는), 프로세서를 보다 많이 사용하는 압축 알고리즘이 사용될 수 있다.

[0042] 하나 이상의 실시예들에서, 메모리 페이지들이 동작(504)에서 개별적으로 압축된다. 대안적으로, 2 개 이상의 메모리 페이지들이 한꺼번에 압축될 수 있다(예컨대, 압축 알고리즘이 개별적으로보다는 한꺼번에 2 개 이상의 메모리 페이지들에 대해 실행될 수 있다).

[0043] 메모리 페이지 세트의 부가 처리가 또한 동작(504)에서 선택적으로 수행될 수 있다. 이 부가 처리는 메모리 페이지 세트의 콘텐츠를 (예컨대, 대칭 키 암호화, 공개 키 암호화 등을 사용하여) 암호화하는 것, 메모리 페이지 세트에 대한 CRC(cyclic redundancy check) 데이터와 같은 오류 검출 데이터를 발생시키는 것, 메모리 페이지 세트에 대한 HMAC(Hash Message Authentication Code) 데이터와 같은 인증 정보를 발생시키는 것 등일 수 있다.

[0044] 동작(502)에서 식별되는 메모리 페이지 세트는 압축된 저장소로 압축된 후에 대기 리스트로 이동되거나(동작(506)), 대안적으로, 사용가능 리스트(free list)로 이동된다. 메모리 페이지 세트가 다양한 방식들로, 예컨대, 압축된 저장소 관리자(104)가 메모리 페이지 세트가 해제(free)될 수 있다는 것을 메모리 관리자(102)에 통지한 것에 응답하여, 대기 리스트(또는 사용가능 리스트)로 이동될 수 있다. 동작(502)에서 식별된 메모리 페이지 세트가 대기 리스트(또는 사용가능 리스트)로 이동되고 용도 변경될 수 있는데, 그 이유는 동일한 메모리 페이지의 2 개의 버전(하나의 버전은 압축되어 있지 않고 하나의 버전은 압축되어 있음)이 페이지정된 메모리에 유지되는 것이 페이지정된 메모리의 효율적인 사용이 아닐 수 있으므로 그렇게 할 필요가 없기 때문이다.

[0045] 압축된 저장소의 영역을 이루고 있는 압축된 저장소 메모리 페이지들이 메모리 관리자에 의해 페이지 파일에 써진다(동작(508)). 유의할 점은 압축된 저장소 메모리 페이지들이 즉각 페이지 파일에 써져야 할 필요는 없다는 것이다. 오히려, 압축된 저장소 메모리 페이지들을 쓰는 것이, 메모리 관리자(102)가 다른 프로그램(108)에 할당하기 위해 메모리 페이지들을 해제하고자 할 때, 압축된 저장소 관리자(104)가 압축된 저장소 메모리 페이지들을 쓰고자 할 때 등과 같은, 나중의 때까지 지연될 수 있다. 또한 유의할 점은 압축된 저장소 메모리 페이지들을 페이지 파일에 쓰는 속도, 효율 등을 개선시키기 위해 임의의 수의 기법들이 구현될 수 있다는 것이다(예

컨대, 압축된 저장소 메모리 페이지들이 인접한 청크(contiguous chunk)들에 써질 수 있고, 기타일 수 있음).

[0046] 추가로 유의할 점은, 동작(502)에서 식별된 메모리 페이지 세트를 페이지 파일에 쓰는 것보다는, 압축된 저장소 메모리 페이지들이 페이지 파일에 써진다(508)는 것이다. 이와 같이, 메모리 페이지 세트의 압축 해제된 버전을 페이지 파일에 쓰지 않고(또는 그러한 일이 없을 때), 동작(502)에서 식별된 메모리 페이지 세트의 압축된 버전이 페이지 파일에 써진다.

[0047] 영역을 이루고 있는 압축된 저장소 메모리 페이지들이 페이지 파일에 써지면, 압축된 저장소 메모리 페이지들은 대기 리스트 상에서 압축된 저장소 메모리 페이지들의 우선순위 레벨(또는 대안적으로 보다 낮은 레벨)에 놓여진다. 이 시점에서, 압축된 저장소 메모리 페이지들은 시스템(100)에서 이용가능한 메모리로서 취급되고, 메모리 관리자(102)에 의해 원하는 바에 따라 언제라도 용도 변경될 수 있다.

[0048] 하나 이상의 실시예들에서, 압축된 저장소 관리자는, 압축된 저장소 메모리 페이지가 실질적으로 채워질 때까지, 메모리 관리자가 압축된 저장소 메모리 페이지를 쓸 수 있게 하지 않는다. 압축된 저장소 메모리 페이지가 실질적으로 채워진다는 것은 부가의 압축된 메모리 페이지들이 저장될 수 있는 저장 공간이 메모리 페이지에 거의 남아 있지 않은 것을 지칭한다. 예를 들어, 압축된 저장소 메모리 페이지의 적어도 문턱값 양의 저장 공간이 압축된 메모리 페이지들을 저장하는 데 사용되었다.

[0049] 유의할 점은, 본원에서의 논의의 일부가 하나 이상의 프로세스들의 메모리 페이지들을 압축하는 것에 관한 것이지만, 본원에서 논의되는 기법들이 사용자 모드 페이징된 메모리 및 커널 모드 페이징된 메모리를 포함할 수 있는 모든 페이징 가능 메모리에 대한 메모리 페이지들을 압축하는 데 유사하게 적용된다는 것이다. 이러한 상황에서, 페이지 파일 전체가 메모리에 의해 압축된 저장소에 표현되고, 압축된 저장소는 기본적으로 모든 수정된 페이지들이 압축된 저장소를 다 써버릴 때 페이지 파일을 관리한다.

[0050] 도 6은 하나 이상의 실시예들에 따른, 압축된 메모리 페이지들을 검색하는 예시적인 프로세스(600)를 나타낸 플로우차트이다. 프로세스(600)는, 도 1의 압축된 저장소 관리자(104)와 같은, 압축된 저장소 관리자에 의해 수행되며, 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합으로 구현될 수 있다. 프로세스(600)가 한 세트의 동작들로서 도시되어 있고, 다양한 동작들의 작업들을 수행하기 위한 도시된 순서로 제한되지 않는다. 프로세스(600)는 압축된 메모리 페이지들을 검색하는 예시적인 프로세스이며; 상이한 도면들을 참조한 압축된 메모리 페이지들을 검색하는 것에 대한 부가의 논의가 본원에 포함되어 있다.

[0051] 메모리 페이지에 대한 요청이 메모리 관리자로부터 수신된다(동작(602)). 이 요청은, 예를 들어, 적어도 하나의 메모리 페이지가 압축된 저장소(110)로 압축된 후에 메모리 페이지 내의 내용에 대한 액세스를 요청하는 프로그램(108)으로부터의 요청에 응답한 것일 수 있다. 요청은 적어도 하나의 메모리 페이지의 식별자(키(key)라고도 지칭됨)를 포함한다. 메모리 관리자는 단일의 메모리 페이지 또는 다수의 메모리 페이지들을 요청할 수 있다.

[0052] 요청에 응답하여, 요청된 메모리 페이지가 페이징된 메모리에서의 압축된 저장소 메모리 페이지에 있는지에 대한 검사가 행해진다(동작(604)). 앞서 논의된 바와 같이, 압축된 저장소 메모리 페이지들이 페이지 파일(112)에 즉각 써질 필요는 없으며, 따라서 요청된 메모리 페이지가 페이징된 메모리(106)의 압축된 저장소 메모리 페이지에서 여전히 이용가능할 수 있다. 유의할 점은 압축된 저장소 메모리 페이지가 압축된 저장소 관리자(104)의 작업 세트에 또는 다양한 리스트들(예컨대, 수정된 리스트 또는 대기 리스트) 상에 있을 수 있지만 여전히 페이징된 메모리에 있을 수 있다는 것이다.

[0053] 요청된 메모리 페이지가 페이징된 메모리에서의 압축된 저장소에 있으면, 요청된 메모리 페이지가 압축 해제되고(동작(606)) 메모리 관리자로 반환된다(동작(608)). 압축 해제되고 반환된 후에, 압축된 메모리 페이지가 압축된 저장소(110)의 영역으로부터 삭제되고, 따라서 메모리 페이지의 압축되지 않은 형태와 압축된 형태 사이의 메모리 페이지의 중복을 회피한다. 압축된 저장소 메모리 페이지의 내용이 압축 해제되고 반환된 후에, 압축된 저장소 메모리 페이지가 용도 변경되거나 해제될 수 있다.

[0054] 요청된 메모리 페이지가 압축 해제되는 방식은 요청된 메모리 페이지를 압축하는 데 사용된 기법에 적어도 부분적으로 기초하여 달라질 수 있다. 메모리 페이지를 복호화하는 것, 오류 검출 데이터를 검증하는 것, 인증 정보를 검증하는 것, 이 검증이 실패하면 (가능한 경우) 데이터를 정정하는 것 등과 같은, 메모리 페이지에 대해 수행되는 임의의 다른 처리가 또한 동작(606)에서 검증되거나 실행 취소된다. 오류 검출 데이터 또는 인증 정보가 검증될 수 없고 그리고/또는 데이터가 정정될 수 없으면, 압축 해제된 요청된 메모리 페이지 대신에 요청된 메모리 페이지가 오염되었다는 표시가 메모리 관리자로 반환될 수 있다.

[0055] 요청된 메모리 페이지가 페이지된 메모리에서의 압축된 저장소에 없으면, 요청된 메모리 페이지를 포함하는 압축된 저장소 메모리 페이지가 페이지 파일로부터 검색된다(동작(610)). 압축된 저장소 메모리 페이지를 검색하는 것은 압축된 저장소 메모리 페이지를 페이지 파일로부터 읽는 것 및 압축된 저장소 메모리 페이지를 페이지된 메모리(106)의 압축된 저장소(110)에 저장하는 것을 포함한다. 요청된 메모리 페이지는 그러면 페이지된 메모리에서의 압축된 저장소(110)에 있고, 따라서 요청된 메모리 페이지가 압축 해제되고(동작(606)) 메모리 관리자로 반환된다(동작(608)).

[0056] 유의할 점은 동작(604) 및 동작(610)이 한꺼번에 수행될 수 있다는 것이다. 예를 들어, 압축된 저장소 관리자(104)는 어느 메모리 페이지들이 어느 압축된 저장소 메모리 페이지들로 압축되는지의 레코드를 유지한다. 동작(602)에서의 요청에 응답하여, 압축된 저장소 관리자(104)는 요청된 메모리 페이지가 압축되어 있는 압축된 저장소 메모리 페이지를 읽는다. 압축된 저장소 메모리 페이지가 페이지된 메모리(106)의 압축된 저장소(110)에 있으면, 압축된 저장소 관리자(104)는 페이지된 메모리(106)로부터 압축된 저장소 메모리 페이지를 읽는다. 압축된 저장소 메모리 페이지가 페이지된 메모리(106)의 압축된 저장소(110)에 있지 않으면, 페이지 폴트(page fault)가 발생하고, 메모리 관리자(102)는 압축된 저장소 메모리 페이지를 페이지 파일로부터 검색하고 압축된 저장소 메모리 페이지를 페이지된 메모리(106)에 놓으며, 이어서 압축된 저장소 관리자는 페이지된 메모리(106)로부터 압축된 저장소 메모리 페이지를 읽는다.

[0057] 또한 유의할 점은 압축된 저장소 관리자(104)가 선택적으로 요청된 메모리 페이지에 부가하여 하나 이상의 메모리 페이지들을 압축 해제하고 반환하기로(그리고 필요한 경우 페이지 파일로부터 검색하기로) 할 수 있다는 것이다. 예를 들어, 압축된 저장소 관리자(104)는 메모리 페이지들에 대한 장래의 메모리 관리자 요청들을 감소시키는 것에 의해 성능을 향상시키기 위해 요청된 메모리 페이지의 근방에 있는(예컨대, 그로부터 문턱값 주소 범위 내에 있는) 하나 이상의 메모리 페이지들을 압축 해제하고 반환하기로 할 수 있다.

[0058] 추가로 유의할 점은, 본원에서 논의되는 메모리 페이지들의 압축 및 메모리 페이지들의 검색과 함께, 압축된 저장소 관리자(104)가 메모리 관리자로부터의 쓰기(압축) 또는 읽기(검색) 요청의 순방향 진행(forward progress)을 보장하는 동작을 한다는 것이다. 압축된 저장소 관리자(104)에 의해 보장되는 순방향 진행은, 적어도 부분적으로 메모리 페이지들의 압축으로 인해 어떤 메모리 페이지들도 이용가능하지 않기 때문에, 시스템(100)에서의 교착 상태(deadlock)를 방지하는 것을 말한다. 압축된 저장소 관리자(104)는, 요청된 메모리 페이지가 이미 압축되어 페이지 파일(112)에 저장되어 있는지에 관계없이 그리고 요청된 메모리 페이지가, 압축되어 페이지 파일(112)에 저장된 후에, 용도 변경되었는지에 관계없이, 이 순방향 진행을 보장하는 동작을 한다. 하나 이상의 실시예들에서, 수정된 리스트 상의 하나 이상의 압축되지 않은 메모리 페이지들을 페이지 파일에 쓰라고 메모리 관리자에 통보함으로써 하나 이상의 메모리 페이지들을 해제하는 것에 의해 메모리 페이지들을 압축하기 위한 순방향 진행이 보장된다. 적어도 하나의 이러한 읽기가 진행되는 데 필요한 자원들을 항상 가지도록 하기 위해 선행적으로(up-front) 필요한 자원(예컨대, 압축된 저장소 관리자의 작업 세트 내의 하나 이상의 메모리 페이지들)을 사전 할당하는 것에 의해 메모리 페이지들을 검색하기 위한 순방향 진행이 보장된다.

[0059] 도 1로 돌아가서, 각각의 메모리 페이지는 메모리 페이지들이 서로 구별될 수 있게 하는 그와 연관된 식별자(키라고도 지칭됨)를 갖는다. 이 식별자는, 메모리 관리자(102) 또는 압축된 저장소 관리자(104)와 같은, 다양한 컴포넌트들 중 임의의 것에 의해 할당될 수 있다. 메모리 페이지의 압축된 버전은 메모리 페이지 자체보다 페이지된 메모리(106)에서 보다 적은 공간을 차지하고, 따라서 압축된 저장소 관리자(104)에 의해 동일한 데이터 단위 구조가 사용되지 않는다. 예를 들어, 메모리 관리자(102)는 페이지된 메모리(106)를 메모리 페이지들에 의해 관리할 수 있지만, 압축된 저장소 관리자(104)는 그러하지 않을 것이다.

[0060] 압축된 저장소 관리자(104)는 다양한 압축된 메모리 페이지들이 압축된 저장소들(110)의 어느 영역들에 저장되는지를 식별해주는 메모리 페이지 맵(memory page map)을 유지한다. 도 7은 하나 이상의 실시예들에 따른, 예시적인 메모리 페이지 맵(702)을 나타낸 것이다. 압축된 저장소(110)로 압축된 각각의 메모리 페이지에 대해, 메모리 페이지 맵(702)은 그 압축된 메모리 페이지가 메모리 페이지들의 영역들에서 어디에 저장되는지를 식별해주는 대응하는 레코드(704)를 유지한다. 압축된 저장소(110)는 하나 이상의(n 개의) 상이한 영역들(706(1), ..., 706(n))로서 예시되어 있으며, 그 영역들 각각 자체는 앞서 논의된 바와 같이 다수의 메모리 페이지들로 이루어져 있다. 하나 이상의 실시예들에서, 페이지된 메모리(106)에서의 각각의 압축된 저장소(110)에 대해 개별적인 메모리 페이지 맵이 유지된다. 대안적으로, 다수의 압축된 저장소들에 대한 메모리 페이지 맵들이 단일의 메모리 페이지 맵으로 결합될 수 있고, 레코드(704)는 압축된 메모리 페이지가 저장되어 있는 압축된 저장소(110)의 식별자를 포함할 수 있다.

- [0061] 각각의 레코드(704)는 페이지 키 필드(710), 영역 번호 필드(712), 오프셋 필드(714), 크기 필드(716), 및 플래그(들) 필드(718)를 포함한다. 페이지 키 필드(710)는 대응하는 압축된 메모리 페이지의 식별자를 저장한다. 이 식별자는 각종의 상이한 형태들 중 임의의 형태를 취할 수 있다. 예를 들어, 이는 프로세스 식별자 및 가상주소, 레코드 식별자 등일 수 있다.
- [0062] 영역 번호 필드(712)는 대응하는 압축된 메모리 페이지가 저장되는 특정의 영역(706)의 식별자를 저장한다. 전형적으로는 번호이지만, 이 영역이 각종의 다른 방식들 중 임의의 방식으로 식별될 수 있을 것이다. 오프셋 필드(714)는 대응하는 압축된 메모리 페이지에 대한 데이터의 저장이 시작되는 특정의 영역(706)에 대한 오프셋의 식별자를 저장한다. 대안적으로, 영역 및 오프셋 식별자보다는, 대응하는 압축된 메모리 페이지가 저장되는 압축된 부분에서의 장소의 임의의 다른 식별자(들)를 저장하는 하나 이상의 필드들이 사용될 수 있을 것이다.
- [0063] 크기 필드(716)는 대응하는 압축된 메모리 페이지의 크기(예컨대, 바이트 단위로 되어 있음)의 식별자를 저장한다. 영역, 특정의 영역에 대한 오프셋, 및 압축된 메모리 페이지의 크기가 주어지면, 압축된 메모리 페이지가 저장되는 메모리 주소 범위가 즉각 식별될 수 있다. 플래그(들) 필드(718)는 대응하는 압축된 메모리 페이지에 관한 다양한 상태 정보를 저장한다. 이러한 플래그들의 예는 어느 메모리 페이지들이 압축되도록 선택되는지를 결정하는 데 사용되는 메모리 페이지에 대한 선택 우선순위, 메모리 페이지에 대응하는 우선순위 레벨 등을 포함한다.
- [0064] 메모리 페이지 맵(702)이 일 예에 불과하다는 것을 잘 알 것이다. 다른 실시예들에서, 부가 필드들이 맵(702)의 각각의 레코드에 포함될 수 있다. 예를 들어, CRC 데이터와 같은 오류 검출 데이터, HMAC 데이터와 같은 인증 정보, 메모리 페이지의 사용 이력 및/또는 빈도수를 기술하는 정보 등이 맵(702)의 각각의 레코드에 포함될 수 있다. 게다가, 필드들(710 내지 718) 중 하나 이상이 맵(702)의 레코드들로부터 생략될 수 있다. 그에 부가하여, 유의할 점은 필드들(710 내지 718) 중 하나 이상에서의 데이터가 별개의 레코드에 저장되는 것이 아니라 페이징된 메모리(106)에서의 메모리 페이지에 저장될 수 있다는 것이다.
- [0065] 메모리 페이지 맵(702)에 부가하여, 압축된 저장소 관리자(104)는 또한 어떤 메모리 페이지가 영역의 특정의 장소에 저장되는지를 식별해주는 영역 맵을 유지한다. 이와 같이, 메모리 페이지 맵(702)은 식별된 압축된 메모리 페이지가 어디에 저장되는지를 식별해주는 동작을 하는 반면, 영역 맵은 영역의 주어진 장소에 어느 압축된 메모리 페이지가 저장되는지를 식별해주는 동작을 한다.
- [0066] 도 8은 하나 이상의 실시예들에 따른, 예시적인 영역 맵(802)을 나타낸 것이다. 상이한 메모리 페이지가 저장되어 있는 영역에 대한 각각의 오프셋에 대해, 영역 맵(802)은 그 오프셋에 저장된 대응하는 압축된 메모리 페이지를 식별해주는 레코드(804)를 유지한다. 각각의 레코드(804)는 페이지 키 필드(806), 영역 번호 필드(808), 및 오프셋 필드(810)를 포함한다.
- [0067] 페이지 키 필드(806)는 도 7의 레코드(704)의 페이지 키 필드(710)와 유사한, 대응하는 압축된 메모리 페이지의 식별자를 저장한다. 영역 번호 필드(808)는 레코드(704)의 영역 번호 필드(712)와 유사한, 대응하는 압축된 메모리 페이지가 저장되는 특정의 영역(706)의 식별자를 저장한다. 오프셋 필드(810)는 레코드(704)의 오프셋 필드(714)와 유사한, 대응하는 압축된 메모리 페이지에 대한 데이터의 저장이 시작되는 특정의 영역(706)에 대한 오프셋의 식별자를 저장한다. 대안적으로, 영역 및 오프셋 식별자보다는, 대응하는 압축된 메모리 페이지가 저장되는 압축된 부분에서의 장소의 임의의 다른 식별자(들)를 저장하는 하나 이상의 필드들이 사용될 수 있을 것이다.
- [0068] 영역 맵(802)이 일 예에 불과하다는 것을 잘 알 것이다. 다른 실시예들에서, 부가 필드들이 맵(802)의 각각의 레코드에 포함될 수 있고, 그리고/또는 필드들(806 내지 810) 중 하나 이상이 맵(802)의 레코드들로부터 생략될 수 있다. 그에 부가하여, 유의할 점은 필드들(806 내지 810) 중 하나 이상에서의 데이터가 별개의 레코드에 저장되는 것이 아니라 도 1의 페이징된 메모리(106)에서의 메모리 페이지에 저장될 수 있다는 것이다.
- [0069] 도 7의 메모리 페이지 맵(702) 및 도 8의 영역 맵(802)은 각종의 상이한 방식들 중 임의의 방식으로 구현될 수 있다. 하나 이상의 실시예들에서, 이 맵들 각각은 개개의 레코드들에 액세스할 때에 증가된 성능을 위해 B+-트리로서 구현된다. 대안적으로, AVL 트리, 해시 테이블, 어레이 등과 같은 각종의 다른 종래의 데이터 구조들 중 임의의 것이 이 맵들을 구현하는 데 사용될 수 있다.
- [0070] 도 1로 돌아가서, 메모리 관리자(102)가 프로그램(108)에 대한 작업 세트 전체를 페이징된 메모리(108) 밖으로 이동시키고자 하는 상황들이 발생할 수 있다. 하나 이상의 실시예들에서, 프로그램(108)이 시스템(100)에서 일시 중단되고 있을 때 이러한 상황들이 일어난다. 프로그램(108)을 일시 중단시키는 프로세스의 일부로서, 메모

리 페이지들의 작업 세트가 압축되고 압축된 저장소(110)에 저장되는데, 이는 작업 세트를 스왑아웃하는 것이라고도 지칭된다. 이 압축된 저장소(110)가 이어서 페이지 파일(112)에 써질 수 있다. 프로그램(108)이 차후에 재개되는 경우(예컨대, 더 이상 일시 중단되지 않는 경우), 압축된 저장소(110)가 페이지 파일(112)로부터 검색되고(압축된 저장소(110)가 페이지 파일(112)에 써졌던 경우), 압축된 저장소(110) 내의 메모리 페이지들이 압축 해제되고 작업 세트(202)로 반환되는데, 이는 작업 세트를 스왑인하는 것이라고도 지칭된다.

[0071] 도 9는 하나 이상의 실시예들에 따른, 작업 세트를 스왑아웃하는 것의 일 예를 나타낸 것이다. 도 9에서, 프로그램(108)의 특정의 프로세스(프로세스 X)의 작업 세트(202)가 예시되어 있다. 작업 세트(202)를 스왑아웃하는 것은, 시작 가상 주소 및 길이(예컨대, 그 시작 가상 주소에서 시작하는 작업 세트 내의 순차적 메모리 페이지들의 개수)의 리스트와 같은, 메모리 페이지들의 리스트(902)를 발생시키는 것에 의해 수행된다. 리스트는, 가장 작은 주소부터 가장 큰 주소(이 주소들은 프로세스가 실행되는 메모리 공간의 가상 주소임)까지와 같이, 순차적으로 정렬된다. 작업 세트(202)의 메모리 페이지들이 수정된 리스트(206)에 이 동일한 순차적 순서로 추가된다. 게다가, 도시된 바와 같이, 작업 세트(202)의 메모리 페이지들이 이 동일한 순차적 순서로 압축되고, 프로그램(108)과 연관된 압축된 저장소(110)에 이 동일한 순차적 순서로 써진다. 이와 같이, 작업 세트(202) 내의 메모리 페이지들이 작업 세트(202)에서 인접해 있지 않을 수 있더라도, 메모리 페이지들이 압축된 저장소(110)에서 인접한 장소들에 써진다.

[0072] 압축된 저장소(110)의 총 크기가 또한 결정되고, 압축된 저장소(110)를 저장하기에 충분한 페이지 파일(112) 내의 공간이 예약된다. 압축된 저장소(110)가 페이지 파일(112)에 써질 때, 압축된 저장소(110)가 인접하여 페이지 파일(112)에 써진다. 페이지들을 페이지 파일의 인접한 순차적으로 정렬된 부분에 쓰는 것은, 페이지들이 차후의 스왑인 동작 동안 작업 세트(202) 내로 다시 읽혀질 때, 차후의 읽기 동작이 페이지 파일(112)의 큰 순차적으로 정렬된 블록을 읽을 수 있게 할 수 있다.

[0073] 도 10은 하나 이상의 실시예들에 따른, 작업 세트를 스왑아웃하는 예시적인 프로세스(1000)를 나타낸 플로우차트이다. 프로세스(1000)는, 도 1의 압축된 저장소 관리자(104)와 같은, 압축된 저장소 관리자에 의해 수행되며, 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합으로 구현될 수 있다. 프로세스(1000)가 한 세트의 동작들로서 도시되어 있고, 다양한 동작들의 작업들을 수행하기 위한 도시된 순서로 제한되지 않는다. 프로세스(1000)는 작업 세트를 스왑아웃하는 예시적인 프로세스이며; 상이한 도면들을 참조한 작업 세트를 스왑아웃하는 것에 대한 부가의 논의가 본원에 포함되어 있다.

[0074] 프로세스(1000)에서, 하나 이상의 페이지들을 프로세스의 작업 세트로부터 페이지 파일 내로 스왑아웃하기로 결정한다(동작(1002)). 이 결정은, 프로세스가 비활성이거나 일시 중단되어 있다는 결정, 프로세스와 연관된 하나 이상의 스크린들이 특정의 양의 시간 동안 활성이 아니었다는 결정, 프로세스가 특정의 양의 시간 동안 백그라운드(background)에 있었다는 결정, 프로세스가 특정의 양의 시간 동안 특정의 개수의 페이지들을 사용하지 않았다는 결정, 컴퓨팅 시스템 전체가 일시 중단되었거나 비활성이라는 결정 등과 같이, (예컨대, 메모리 관리자(102) 또는 다른 정책 관리자에 의해) 다양한 기준들에 기초하여 행해질 수 있다.

[0075] 하나 이상의 페이지들을 스왑아웃하기로 결정되었으면, 하나 이상의 후보 페이지들이 프로세스의 작업 세트로부터 식별되고, 이 후보 메모리 페이지들의 리스트가 발생된다(동작(1004)). 후보 메모리 페이지들의 리스트가, 가장 작은 가상 주소부터 가장 큰 가상 주소까지와 같이, 순차적으로 정렬된다. 하나 이상의 실시예들에서, 메모리 관리자(102)는 작업 세트의 각각의 페이지를 분석하고, 특정의 기준들에 기초하여 각각의 페이지가 스왑아웃을 위한 후보인지를 결정한다. 하나 이상의 실시예들에서, 스왑아웃을 위한 후보들은 작업 세트(202) 내의 개인 페이지 및/또는 페이지 파일 배킹된 공유 페이지(page file backed shared page)를 포함할 수 있다. 후보 페이지들이 또한 상이한 방식으로, 예컨대, 그 페이지들이 클린(clean)(압축된 저장소(110)에 써졌지만 이러한 쓰기 이후에 수정되지 않은 페이지, 따라서 물리 메모리에서의 페이지의 현재 버전이 압축된 저장소(110)에서의 페이지와 동일함)인지에 기초하여, 식별될 수 있다. 게다가, 하나 이상의 실시예들에서, 페이지가 스왑아웃을 위한 후보인지를 결정할 때, 페이지가 메모리에 잠금(lock)되어 있는지 여부가 고려될 수 있다.

[0076] 동작(1004)에서 발생된 리스트 상의 후보 메모리 페이지들이 수정된 리스트(206)에 추가된다(동작(1006)). 후보 메모리 페이지들이, 동작(1004)에서 발생된 리스트 상에서 나오는 것과 동일한 순차적 순서로, 수정된 리스트에 추가된다.

[0077] 수정된 리스트(206)에서의 후보 메모리 페이지들이 압축되고 프로세스인 실행 중인 프로그램과 연관된 압축된 저장소(110)에 써진다(동작(1008)). 도 5와 관련한 상기 논의와 유사하게, 각종의 방식들 중 임의의 방식으로 압축이 수행될 수 있다. 도 5와 관련한 상기 논의와 유사하게, 다양한 다른 동작들(예컨대, 메모리 페이지들의

내용을 암호화하는 것, 오류 검출 데이터를 발생시키는 것 등)이 또한 메모리 페이지들에 대해 선택적으로 수행될 수 있다. 압축된 메모리 페이지들이, 동작(1004)에서 발생된 리스트 상에서 나오는 그리고 수정된 리스트(206)에서 나오는 이 동일한 순차적 순서로, 압축된 저장소(110)에 써진다. 이와 같이, 작업 세트(202) 내의 메모리 페이지들이 작업 세트(202)에서 인접해 있지 않을 수 있더라도, (예컨대, 도 9에 도시된 바와 같이) 메모리 페이지들이 압축된 저장소(110)에서 인접한 장소들에 써진다. 후보 메모리 페이지가 압축된 저장소(110)에 써진 후에, 후보 메모리 페이지가 작업 세트(202)로부터 제거된다(예컨대, 그리고 사용가능 리스트에 추가된다).

[0078] 압축된 저장소(110)를 저장하기에 충분한 페이지 파일(112) 내의 공간이 예약된다(동작(1010)). 페이지 파일(112) 내의 공간을 예약한다는 것은 페이지 파일(112)의 일부를 확보하거나 압축된 저장소(110)를 쓰는 것은 용인하지만 다른 데이터를 쓰는 것은 용인하지 않는 것으로 그에 표시하는 것을 지칭한다. 페이지 파일(112) 내의 공간이 각종의 상이한 방식으로, 예컨대, 페이지 파일의 부분들 및 이러한 부분들이 예약되어 있는지(그리고 예약되어 있는 경우, 그들이 예약되어 있는 압축된 저장소(110))를 식별해주는 할당 테이블(allocation table) 또는 다른 코드를 유지하는 것에 의해, 예약될 수 있다. 예약하는 것은, 예를 들어, 메모리 관리자(102) 또는 압축된 저장소 관리자(104)에 의해 수행될 수 있다.

[0079] 어떤 나중의 때에, 압축된 스왑아웃된 페이지들을 포함하는 압축된 저장소(110)를 페이지 파일(112) 내의 그의 예약된 공간에 쓰는 것에 대한 결정이 (예컨대, 메모리 관리자(102) 또는 압축된 저장소 관리자(104)에 의해) 행해진다(동작(1012)). 하나 이상의 실시예들에서, 이 결정은 (동작(1002)에서) 스왑아웃하기로 하는 결정을 가져온 기준들이 여전히 유효한 특정 문턱값 양의 시간이 경과했다는 결정에 기초하여 행해질 수 있다. 예를 들어, 프로세스가 여전히 비활성이거나 일시 중단되어 있는 특정의 양의 시간(예컨대, 5분)이 경과할 수 있다. 그 결정이 또한 하나 이상의 다른 프로그램들(108)이 사용하기 위한 보다 많은 물리 메모리가 필요하다는 메모리 관리자(102)에 의한 결정에 기초하여 행해질 수 있다.

[0080] 압축된 저장소를 스왑 파일(swap file)에 쓰기로 하는 동작(1012)에서의 결정에 응답하여, 압축된 저장소 메모리 페이지들이 페이지 파일 내의 예약된 공간에 써진다(동작(1014)). 하나 이상의 실시예들에서, 압축된 저장소 메모리 페이지들 모두가 제거되고 하나 이상의 쓰기 동작들에서 페이지 파일(112)에 써진다. 대안적으로, 압축된 저장소 메모리 페이지들의 일부분만이 제거되고 페이지 파일(112)에 써질 수 있다. 일부 실시예에서, 써지는 후보 페이지들은 메모리의 부족에 기초하여(예컨대, 메모리 관리자가 보다 많은 물리 메모리를 필요로 하는 것에 기초하여) 결정될 수 있다. 페이지 파일(112)에 써진 후에, 압축된 저장소 메모리 페이지들은 대기 리스트 상에서 압축된 저장소 메모리 페이지들의 우선순위 레벨(또는 대안적으로 보다 낮은 레벨)에 놓여진다. 이 시점에서, 압축된 저장소 메모리 페이지들은 시스템에서 이용가능한 메모리로서 취급되고, 메모리 관리자(102)에 의해 원하는 바에 따라 언제라도 용도 변경될 수 있다.

[0081] 동작(1012)으로 돌아가서, 어떤 상황에서, 압축된 저장소를 쓰지 않기로 하는 결정이 행해질 수 있다. 이러한 상황에서, 압축된 저장소를 위한 예약된 공간은 스왑인이 일어날 때까지 유지된다. 하나 이상의 실시예들에서, 특정의 문턱값 양의 시간 후에 (예컨대, 동작(1002)에서) 페이지들을 스왑아웃하기로 하는 초기 결정을 가져온 조건들이 더 이상 존재하지 않는 경우, 압축된 저장소를 쓰지 않기로 하는 결정이 행해질 수 있다. 예를 들어, 프로세스가 다시 활성일 수 있거나 더 이상 일시 중단되지 않을 수 있다. 다른 상황에서, 그 후보 페이지들이 페이지 파일에 써지기 전에 페이지들을 스왑인하기로 하는 결정이 있을 수 있다.

[0082] 도 11은 하나 이상의 실시예들에 따른, 작업 세트를 스왑인하는 예시적인 프로세스(1100)를 나타낸 플로우차트이다. 프로세스(1100)는, 도 1의 압축된 저장소 관리자(104)와 같은, 압축된 저장소 관리자에 의해 수행되며, 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합으로 구현될 수 있다. 프로세스(1100)가 한 세트의 동작들로서 도시되어 있고, 다양한 동작들의 작업들을 수행하기 위한 도시된 순서로 제한되지 않는다. 프로세스(1100)는 작업 세트를 스왑인하는 예시적인 프로세스이며; 상이한 도면들을 참조한 작업 세트를 스왑인하는 것에 대한 부가의 논의가 본원에 포함되어 있다.

[0083] 프로세스(1100)에서, 프로세스에 대한 작업 세트로부터 이전에 스왑아웃된 압축된 저장소를 스왑인하기로 결정된다(동작(1102)). 하나 이상의 실시예들에서, 이 결정이 메모리 관리자(102) 또는 다른 정책 관리자에 의해 행해지고, 다양한 기준들에 기초하여 행해질 수 있다. 이러한 기준들은 프로세스가 더 이상 활성이 아니거나 일시 중단되지 않는다는 표시를 수신하는 것, 도 10의 동작(1002)에서 스왑아웃하기로 하는 결정을 가져온 그 조건들의 부존재(cessation) 등을 포함한다.

[0084] 스왑아웃된 압축된 저장소 메모리 페이지들이 물리 메모리에 남아 있는지의 결정이 행해진다(동작(1104)). 어

면 상황에서, 압축된 저장소 메모리 페이지들이 페이지 파일에 써졌을 수 있지만 물리 메모리에 남아 있다(예컨대, 페이지들이 대기 리스트 상에 캐시되었고 아직 용도 변경되지 않은 경우). 이러한 상황에서, 압축된 저장소 메모리 페이지들을 페이지 파일로부터 읽는 것보다는, 물리 메모리에 남아 있는 이 압축된 저장소 메모리 페이지들이 압축 해제될 수 있다.

[0085] 스왑아웃된 압축된 저장소 메모리 페이지가 물리 메모리에 없으면, 메모리 페이지가 페이지 파일로부터 검색된다(동작(1106)). 압축된 저장소 메모리 페이지들을 검색하는 것은 압축된 저장소 메모리 페이지들을 페이지 파일로부터 읽는 것 및 압축된 저장소 메모리 페이지들을 페이지징된 메모리(106)의 압축된 저장소(110)에 저장하는 것을 포함한다. 하나 이상의 실시예들에서, 압축된 저장소 메모리 페이지들을 이와 같이 읽어들이는 것은 페이지 파일로부터 압축된 저장소 메모리 페이지들의 큰 블록을 읽어들이고, 따라서 페이지 파일 내의 임의의 장소들로부터 보다 작은 블록을 읽는 것보다 더 효율적이다. 게다가, 페이지들이 페이지 파일 내의 예약된 인접한 블록에 써졌고 순차적인 가상 주소 순서로 써졌기 때문에, 다수의 페이지들의 클러스터가 이어서 큰 블록으로 그리고 순차적 가상 주소 순서로 읽혀질 수 있어, 보다 효율적인 읽기 동작을 제공할 수 있다.

[0086] 압축된 저장소(110)의 압축된 저장소 메모리 페이지들에 있는 스왑아웃된 메모리 페이지들이 압축 해제되고(동작(1108)) 메모리 관리자로 반환된다(동작(1110)). 압축 해제된 메모리 페이지들을 메모리 관리자로 반환하는 것은 압축 해제된 메모리 페이지들이 프로그램의 작업 세트에 포함될 수 있게 한다. 도 6과 관련한 상기 논의와 유사하게, 각종의 방식들 중 임의의 방식으로 압축 해제가 수행될 수 있다. 도 6과 관련한 상기 논의와 유사하게, 다양한 다른 동작들(예컨대, 메모리 페이지들의 내용을 복호화하는 것, 오류 검출 데이터를 검증하는 것 등)이 또한 메모리 페이지들에 대해 선택적으로 수행될 수 있다. 압축 해제되고 반환된 후에, 압축된 메모리 페이지가 압축된 저장소(110)의 영역으로부터 삭제된다. 압축된 저장소 메모리 페이지의 내용이 압축 해제되고 반환된 후에, 압축된 저장소 메모리 페이지가 용도 변경되거나 해제될 수 있다.

[0087] 페이지 파일 내의 압축된 저장소를 위한 예약된 공간이 또한 해제된다(동작(1112)). 페이지 파일 내의 예약된 공간이 이어서 다른 데이터를 저장하는 데 이용 가능하다.

[0088] 동작(1104)으로 돌아가서, 스왑아웃된 메모리 페이지들 중 임의의 것이 물리 메모리에 남아 있으면, 그 메모리 페이지들이 압축 해제되고(동작(1108)) 메모리 관리자로 반환된다(동작(1110)). 그 스왑아웃된 압축된 메모리 페이지들을 저장하는 압축된 저장소 메모리 페이지들이 동작(1106)에서 페이지 파일로부터 읽혀질 필요가 없는 데, 그 이유는 그들이 물리 메모리에 남아 있기 때문이다.

[0089] 유의할 점은 동작(1106)에서 스왑아웃된 압축된 저장소 메모리 페이지들을 검색하는 것 및 동작(1108)에서 메모리 페이지들을 압축 해제하는 것이 병렬로 수행될 수 있다는 것이다. 하나 이상의 실시예들에서, 다른 스왑아웃된 압축된 저장소 메모리 페이지들이 페이지 파일로부터 읽혀지고 있는 동안 그와 동시에 페이지 파일로부터 검색된 메모리 페이지들은 압축 해제된다. 메모리 페이지들을 압축 해제하고 압축된 저장소 메모리 페이지들을 페이지 파일로부터 검색하기 위해 다수의 스크래드들이 사용될 수 있다. 예를 들어, 시스템(100) 내의 하나의 프로세서 코어에서 실행 중인 하나의 스크래드는 압축된 저장소 메모리 페이지들을 페이지 파일로부터 검색하는 것을 관리할 수 있고, 시스템(100) 내의 나머지 프로세서 코어들에서 동시에 또는 병렬로 실행 중인 스크래드들은 메모리 페이지들의 압축 해제를 관리할 수 있다.

[0090] 이와 같이, 압축된 저장소 메모리 페이지들을 스왑인하는 것의 성능이 본원에서 논의되는 기법들을 사용하여 향상될 수 있다. 메모리 페이지들이 압축되고, 따라서 압축이 없는 경우 읽혀지는 것보다 더 적은 수의 메모리 페이지들이 읽혀진다. 게다가, 압축 해제가 메모리 페이지들을 읽는 것과 병렬로 수행될 수 있고, 다수의 압축 해제 스크래드들이 동시에 실행될 수 있어, 압축 해제된 메모리 페이지들이 메모리 관리자로 반환될 수 있는 속도를 추가로 증가시킨다.

[0091] 본원에서 논의되는 기법들은 또한 다양한 부가 기능을 지원한다. 메모리 페이지들이 압축되는 것으로 인해, 페이지 파일을 저장하는 저장 디바이스에 대한 읽기와 쓰기의 횟수가 감소될 수 있다. 저장 디바이스에 대한 읽기 및 쓰기의 횟수를 감소시키는 것은 저장 디바이스의 수명 증가를 가져올 수 있음은 물론, 저장 디바이스 임력/출력으로 인한 병목 현상을 감소시킬 수 있다. 게다가, 메모리 페이지들의 압축은 메모리 관리자에 의해 프로그램에 할당하기 위해 이용 가능한 메모리 페이지들의 개수의 증가를 가져올 수 있다.

[0092] 게다가, 스왑아웃된 압축된 저장소 메모리 페이지들은 메모리 페이지들의 압축 해제된 버전보다 더 적은 수의 페이지들을 차지한다. 스왑아웃된 압축된 저장소 메모리 페이지들이 앞서 논의된 바와 같이 페이지 파일(112)에 써지고 대기 리스트에 추가되며, 그들의 압축된 속성으로 인해, 보다 많은 수의 압축된 저장소들이 대기 리

스트 상에 유지될 수 있을 것이다. 이것은, 일시 중단된 프로그램이 재개되는 경우, 프로그램에 대한 압축된 저장소 메모리 페이지들이 물리 메모리에 있을 가능성을 증가시킬 수 있고, 따라서 페이지 파일로부터 검색되는 경우보다 더 빨리 스왑인될 수 있다.

[0093] 그에 부가하여, 압축된 메모리 페이지들은 그 메모리 페이지들이 압축 이전에 있었던 것과 유사한 우선순위 레벨들에서 압축된 저장소들에 저장된다. 메모리 페이지들의 적어도 유사한 우선순위 레벨들을 유지하는 것에 의해, 상당히 더 높은 우선순위의 메모리 페이지들이 보다 낮은 우선순위의 메모리 페이지보다 먼저 페이지 파일에 써지는 우선순위 역전 시나리오가 방지된다.

[0094] 특정의 기능이 특정의 모듈들을 참조하여 본원에서 논의되지만, 유의할 점은 본원에서 논의되는 개개의 모듈들의 기능이 다수의 모듈들로 분할될 수 있고 그리고/또는 다수의 모듈들의 적어도 일부 기능이 단일의 모듈로 결합될 수 있다는 것이다. 그에 부가하여, 동작을 수행하는 것으로 본 명세서에서 논의된 특정의 모듈은 동작을 수행하는 그 특정의 모듈 자체, 또는 대안적으로, 동작을 수행하는(또는 그 특정의 모듈과 함께 동작을 수행하는) 다른 컴포넌트 또는 모듈을 호출하거나 다른 방식으로 그에 액세스하는 그 특정의 모듈을 포함한다. 이와 같이, 동작을 수행하는 특정의 모듈은 동작을 수행하는 그 특정의 모듈 자체 및/또는 동작을 수행하는 그 특정의 모듈에 의해 호출되거나 다른 방식으로 액세스되는 다른 모듈을 포함한다.

[0095] 도 12는 본원에 기술되는 다양한 기법들을 구현할 수 있는 하나 이상의 시스템들 및/또는 디바이스들을 나타내는 예시적인 컴퓨팅 디바이스(1202)를 포함하는 전체적으로 1200으로 나타낸 예시적인 시스템을 예시한 것이다. 컴퓨팅 디바이스(1202)는, 예를 들어, 서비스 공급자의 서버, 클라이언트와 연관되어 있는 디바이스(예컨대, 클라이언트 디바이스), 온칩 시스템, 및/또는 임의의 다른 적당한 컴퓨팅 디바이스 또는 컴퓨팅 시스템일 수 있다.

[0096] 예시적인 컴퓨팅 디바이스(1202)는, 예시된 바와 같이, 서로 통신 가능하게 결합되어 있는 처리 시스템(1204), 하나 이상의 컴퓨터 판독가능 매체(1206), 및 하나 이상의 I/O 인터페이스들(1208)을 포함한다. 비록 도시되어 있지는 않지만, 컴퓨팅 디바이스(1202)는 다양한 컴포넌트들을 서로 결합시키는 시스템 버스 또는 다른 데이터 및 명령 전송 시스템을 추가로 포함할 수 있다. 시스템 버스는 각종의 버스 아키텍처들 중 임의의 것을 이용하는 메모리 버스 또는 메모리 제어기, 주변기기 버스, USB(universal serial bus), 및/또는 프로세서 또는 로컬 버스와 같은, 상이한 버스 구조들 중 임의의 것 또는 그의 임의의 조합을 포함할 수 있다. 제어 및 데이터 라인들과 같은 각종의 다른 예들이 또한 생각되고 있다.

[0097] 처리 시스템(1204)은 하드웨어를 사용하여 하나 이상의 동작들을 수행하는 기능을 나타낸다. 그에 따라, 처리 시스템(1204)은 프로세서들, 기능 블록들 등으로서 구성될 수 있는 하드웨어 요소들(1210)을 포함하는 것으로 예시되어 있다. 이것은 하나 이상의 반도체들을 사용하여 형성되는 ASIC(application specific integrated circuit) 또는 다른 논리 디바이스로서 하드웨어로 구현하는 것을 포함할 수 있다. 하드웨어 요소들(1210)은 그들이 형성되는 재료들 또는 그들에서 이용되는 처리 메커니즘들에 의해 제한되지는 않는다. 예를 들어, 프로세서들은 반도체(들) 및/또는 트랜지스터들(예컨대, 전자 집적 회로(IC)들)로 이루어져 있을 수 있다. 이와 관련하여, 프로세서 실행가능 명령어는 전자적으로 실행가능한 명령어일 수 있다.

[0098] 컴퓨터 판독가능 매체(1206)는 메모리/저장소(1212)를 포함하는 것으로 예시되어 있다. 메모리/저장소(1212)는 하나 이상의 컴퓨터 판독가능 매체와 연관된 메모리/저장 용량을 나타낸다. 메모리/저장소(1212)는 휘발성 매체(RAM(random access memory) 등) 및/또는 비휘발성 매체(ROM(read only memory), 플래시 메모리, 광 디스크, 자기 디스크, 기타 등등)를 포함할 수 있다. 메모리/저장소(1212)는 고정식 매체(fixed media)(예컨대, RAM, ROM, 고정식 하드 드라이브 등)는 물론, 이동식 매체(예컨대, 플래시 메모리, 이동식 하드 드라이브, 광 디스크 등)를 포함할 수 있다. 컴퓨터 판독가능 매체(1206)는, 이하에서 더 기술하는 바와 같이, 각종의 다른 방식들로 구성될 수 있다.

[0099] 입출력 인터페이스(들)(1208)는 사용자가 컴퓨팅 디바이스(1202)에 명령들 및 정보를 입력할 수 있게 하고 또한 정보가 다양한 입출력 디바이스들을 사용하여 사용자 및/또는 다른 컴포넌트들 또는 디바이스들에 제시될 수 있게 하는 기능을 나타낸다. 입력 디바이스들의 예는 키보드, 커서 제어 디바이스(예컨대, 마우스), (예컨대, 음성 입력을 위한) 마이크로폰, 스캐너, 터치 기능(예컨대, 물리적 터치를 검출하도록 구성되어 있는 용량성 또는 다른 센서), (예컨대, 제스처로서 터치를 수반하지 않는 움직임을 검출하기 위해 가시 광장 또는 적외선 주파수와 같은 비가시 광장을 이용할 수 있는) 카메라 등을 포함한다. 출력 디바이스들의 예는 디스플레이 디바이스(예컨대, 모니터 또는 프로젝터), 스피커, 프린터, 네트워크 카드, 촉각 반응 디바이스(tactile-response device) 등을 포함한다. 이와 같이, 컴퓨팅 디바이스(1202)는 사용자 상호작용을 지원하기 위해 이하에서 더

기술되는 바와 같이 각종의 방식들로 구성될 수 있다.

[0100] 컴퓨팅 디바이스(1202)는 또한 저장소 관리자(1214)를 포함한다. 저장소 관리자(1214)는 앞서 논의된 바와 같이 다양한 수정된 메모리 압축 기능을 제공한다. 저장소 관리자(1214)는, 예를 들어, 도 1의 압축된 저장소 관리자(104)를 구현할 수 있다. 컴퓨터 판독가능 매체(1206)와 분리되어 예시되어 있지만, 유의할 점은 저장소 관리자(1214)가 대안적으로 컴퓨터 판독가능 매체(1206)의 일부로서 구현될 수 있다는 것이다.

[0101] 다양한 기법들이 일반적으로 소프트웨어, 하드웨어 요소들, 또는 프로그램 모듈들과 관련하여 본 명세서에 기술되어 있을 수 있다. 일반적으로, 이러한 모듈들은 특정의 작업들을 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로그램, 객체, 요소, 컴포넌트, 데이터 구조 등을 포함한다. "모듈", "기능", 및 "컴포넌트"라는 용어는, 본 명세서에서 사용되는 바와 같이, 일반적으로 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합을 나타낸다. 본원에 기술되는 기법들이 플랫폼 독립적(platform-independent)이며, 이는 그 기법들이 각종의 프로세서들을 가지는 각종의 컴퓨팅 플랫폼들에서 구현될 수 있다는 것을 의미한다.

[0102] 기술된 모듈들 및 기법들의 구현은 어떤 형태의 컴퓨터 판독가능 매체들 상에 저장되거나 그를 통해 전송될 수 있다. 컴퓨터 판독가능 매체는 컴퓨팅 디바이스(1202)에 의해 액세스될 수 있는 각종의 매체를 포함할 수 있다. 제한이 아닌 예로서, 컴퓨터 판독가능 매체는 "컴퓨터 판독가능 저장 매체" 및 "컴퓨터 판독가능 신호 매체"를 포함할 수 있다.

[0103] "컴퓨터 판독가능 저장 매체"는, 단순 신호 전송, 반송파, 또는 신호 자체와 달리, 정보의 영속적인 저장 및/또는 유형적인 저장을 가능하게 하는 매체 및/또는 디바이스들을 지칭한다. 이와 같이, 컴퓨터 판독가능 저장 매체는 비신호 전달 매체(non-signal bearing media)를 지칭한다. 컴퓨터 판독가능 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 논리 요소/회로, 또는 다른 데이터와 같은 정보를 저장하기 위해 적당한 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체 및/또는 저장 디바이스와 같은 하드웨어를 포함한다. 컴퓨터 판독가능 저장 매체의 예는 원하는 정보를 저장하는 데 적당하고 컴퓨터에 의해 액세스될 수 있는, RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 다른 광 저장소, 하드 디스크, 자기 카세트, 자기 테이프, 자기 디스크 저장소 또는 다른 자기 저장 디바이스, 또는 다른 저장 디바이스, 유형적 매체(tangible media), 또는 제조 물품을 포함할 수 있지만, 이들로 제한되지 않는다.

[0104] "컴퓨터 판독가능 신호 매체"는 명령어들을, 예컨대, 네트워크를 통해, 컴퓨팅 디바이스(1202)의 하드웨어로 전송하도록 구성되어 있는 신호 전달 매체(signal-bearing medium)를 지칭한다. 신호 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 다른 데이터를 반송파, 데이터 신호, 또는 다른 전송 메커니즘과 같은 피변조 데이터 신호(modulated data signal)에 구현할 수 있다. 신호 매체는 또한 임의의 정보 전달 매체(information delivery media)를 포함한다. "피변조 데이터 신호"라는 용어는 신호의 특성을 중 하나 이상이 정보를 그 신호에 인코딩하는 방식으로 설정되거나 변경된 신호를 의미한다. 제한이 아닌 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 연결(direct-wired connection)과 같은 유선 매체와, 음향, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 포함한다.

[0105] 이전에 기술된 바와 같이, 하드웨어 요소(1210) 및 컴퓨터 판독가능 매체(1206)는, 일부 실시예에서, 본원에 기술되는 기법들의 적어도 일부 양태들을 구현하기 위해 이용될 수 있는, 명령어, 모듈, 프로그램가능 디바이스 논리, 및/또는 하드웨어 형태로 구현되는 고정식 디바이스 논리를 나타낸다. 하드웨어 요소는 집적 회로 또는 온칩 시스템, ASIC(application-specific integrated circuit), FPGA(field-programmable gate array), CPLD(complex programmable logic device), 및 실리콘(silicon) 또는 다른 하드웨어 디바이스에서의 다른 구현들의 컴포넌트들을 포함할 수 있다. 이와 관련하여, 하드웨어 요소는 하드웨어 요소에 의해 구현되는 명령어, 모듈, 및/또는 논리에 의해 정의되는 프로그램 작업을 수행하는 처리 디바이스는 물론, 실행을 위한 명령어들을 저장하는 데 이용되는 하드웨어 디바이스(예컨대, 이전에 기술된 컴퓨터 판독가능 저장 매체)로서 동작할 수 있다.

[0106] 상기한 것들의 조합들이 또한 본원에 기술되는 다양한 기법들 및 모듈들을 구현하는 데 이용될 수 있다. 그에 따라, 소프트웨어, 하드웨어, 또는 프로그램 모듈 및 기타 프로그램 모듈이 어떤 형태의 컴퓨터 판독가능 저장 매체 상에 구현된 하나 이상의 명령어들 및/또는 논리로서 그리고/또는 하나 이상의 하드웨어 요소들(1210)에 의해 구현될 수 있다. 컴퓨팅 디바이스(1202)는 소프트웨어 및/또는 하드웨어 모듈들에 대응하는 특정의 명령어들 및/또는 기능들을 구현하도록 구성될 수 있다. 그에 따라, 컴퓨팅 디바이스(1202)에 의해 실행가능한 모듈과 같은 모듈들을 소프트웨어로서 구현하는 것이 적어도 부분적으로 하드웨어로, 예컨대, 처리 시스템의 컴퓨

터 판독가능 저장 매체 및/또는 하드웨어 요소들(1210)의 사용을 통해 달성될 수 있다. 명령어들 및/또는 기능들은 본 명세서에 기술된 기법, 모듈, 및 예를 구현하기 위해 하나 이상의 제조 물품(예를 들어, 하나 이상의 컴퓨팅 디바이스(1202) 및/또는 처리 시스템(1204))에 의해 실행가능/동작가능할 수 있다.

[0107] 도 12에 추가로 예시되어 있는 바와 같이, 예시적인 시스템(1200)은 PC(personal computer), 텔레비전 디바이스, 및/또는 모바일 디바이스 상에서 애플리케이션을 실행할 때 매끄러운 사용자 경험을 위한 유비쿼터스 환경을 가능하게 한다. 서비스 및 애플리케이션은 애플리케이션을 이용하는 것, 비디오 게임을 플레이하는 것, 비디오를 시청하는 것 등을 하면서 하나의 디바이스로부터 다음 디바이스로 전환할 때 공통의 사용자 경험을 위해 3 가지 환경들 모두에서 실질적으로 유사하게 실행된다.

[0108] 예시적인 시스템(1200)에서, 다수의 디바이스들이 중앙 컴퓨팅 디바이스를 통해 상호연결된다. 중앙 컴퓨팅 디바이스는 다수의 디바이스들에 로컬일 수 있거나 다수의 디바이스들로부터 원격지에 위치될 수 있다. 하나 이상의 실시예들에서, 중앙 컴퓨팅 디바이스는 네트워크, 인터넷, 또는 다른 데이터 통신 링크를 통해 다수의 디바이스들에 연결되는 하나 이상의 서버 컴퓨터들의 클라우드일 수 있다.

[0109] 하나 이상의 실시예들에서, 이 상호연결 아키텍처는, 다수의 디바이스들의 사용자에게 공통의 매끄러운 경험을 제공하기 위해, 기능이 다수의 디바이스들을 거쳐 전달될 수 있게 한다. 다수의 디바이스들 각각은 상이한 물리적 요구사항 및 능력을 가질 수 있고, 중앙 컴퓨팅 디바이스는 디바이스에 맞게 조정되어 있음과 동시에 모든 디바이스들에 여전히 공통인 경험을 디바이스에 전달하는 것을 가능하게 하기 위해 플랫폼을 사용한다. 하나 이상의 실시예들에서, 한 부류의 대상 디바이스(target device)들이 생성되고, 디바이스들의 일반 부류(generic class)에 따라 경험이 조정된다. 디바이스들의 물리적 특징, 사용 유형, 또는 다른 공통 특성에 의해 한 부류의 디바이스들이 정의될 수 있다.

[0110] 다양한 구현들에서, 컴퓨팅 디바이스(1202)는, 컴퓨터(1216), 모바일(1218), 및 텔레비전(1220) 용도에 대해서와 같이, 각종의 상이한 구성들을 가질 수 있다. 이 구성들 각각은 일반적으로 상이한 구조 및 능력을 가질 수 있는 디바이스들을 포함하고, 따라서 컴퓨팅 디바이스(1202)는 상이한 디바이스 부류들 중 하나 이상에 따라 구성될 수 있다. 예를 들어, 컴퓨팅 디바이스(1202)는 개인용 컴퓨터, 데스크톱 컴퓨터, 멀티 스크린 컴퓨터, 랩톱 컴퓨터, 넷북 등을 포함하는 컴퓨터(1216) 부류의 디바이스로서 구현될 수 있다.

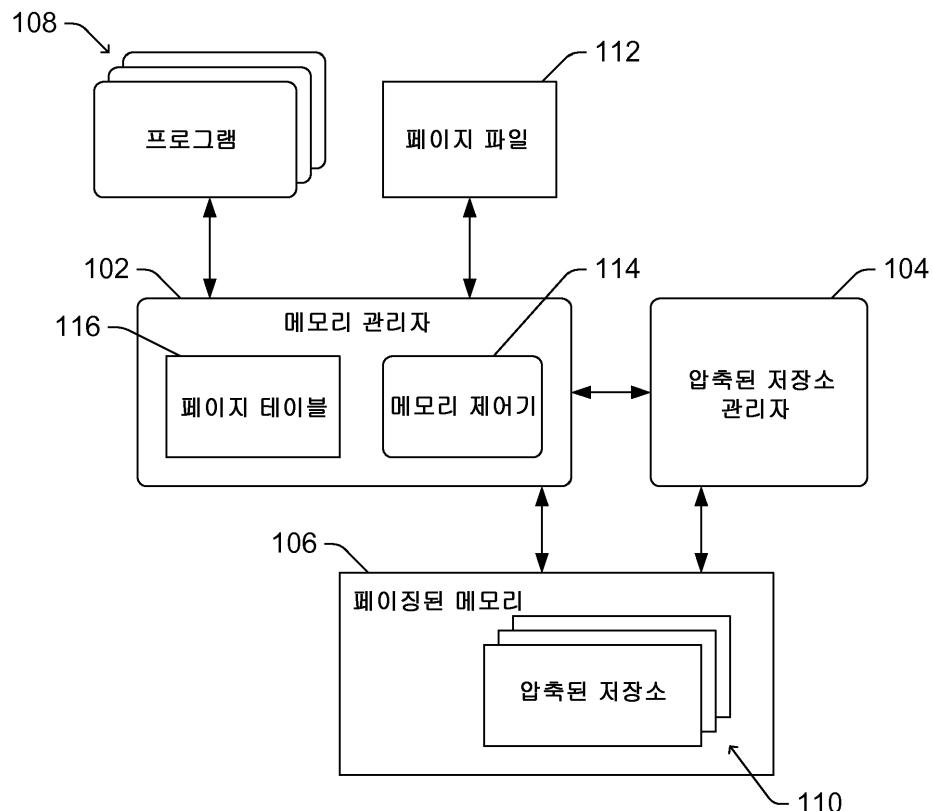
[0111] 컴퓨팅 디바이스(1202)는 또한 휴대폰, 휴대용 음악 플레이어, 휴대용 게임 디바이스, 태블릿 컴퓨터, 멀티 스크린 컴퓨터 등과 같은 모바일 디바이스들을 포함하는 모바일(1218) 부류의 디바이스로서 구현될 수 있다. 컴퓨팅 디바이스(1202)는 또한 통상의 시청 환경에서 일반적으로 대형 화면을 가지거나 그에 연결되어 있는 디바이스들을 포함하는 텔레비전(1220) 부류의 디바이스로서 구현될 수 있다. 이 디바이스들은 텔레비전, 셋톱 박스, 게임 콘솔 등을 포함한다.

[0112] 본원에 기술되는 기법들은 컴퓨팅 디바이스(1202)의 이 다양한 구성들에 의해 지원될 수 있고, 본 명세서에 기술된 기법들의 특정의 예로 제한되지 않는다. 이 기능은 또한 전부 또는 일부가, 이하에 기술되는 바와 같이, 예컨대, 플랫폼(1224)을 통해 "클라우드"(1222)를 거쳐, 분산 시스템을 사용하여 구현될 수 있다.

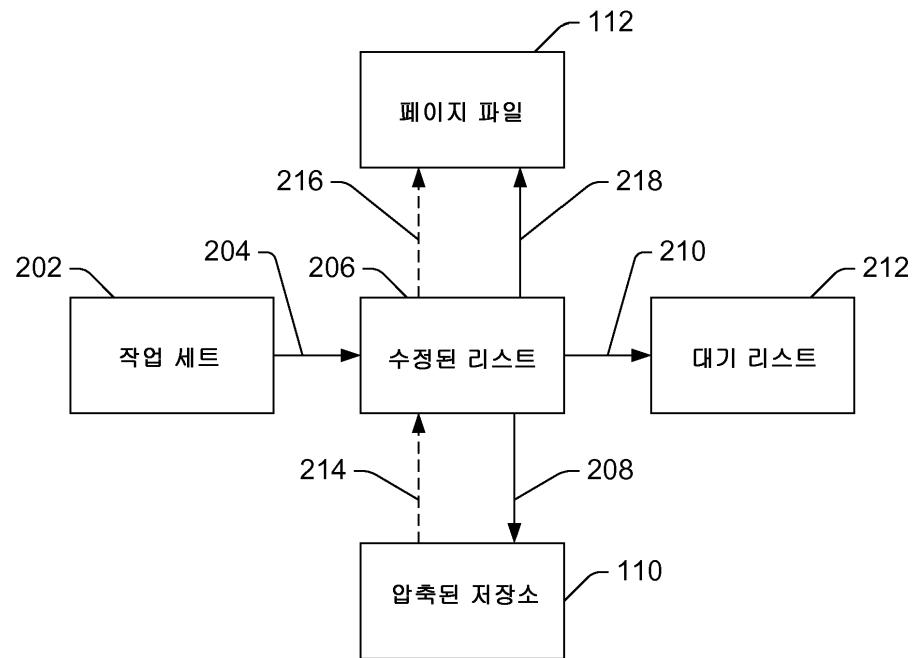
[0113] 클라우드(1222)는 자원(1226)을 포함하고 그리고/또는 자원(524)에 대한 플랫폼(1224)을 나타낸다. 플랫폼(1224)은 클라우드(1222)의 하드웨어 자원(예컨대, 서버) 및 소프트웨어 자원의 기본 기능을 추상화한다. 자원(1226)은 컴퓨팅 디바이스(1202)로부터 원격지에 있는 서버 상에서 컴퓨터 처리가 실행되는 동안 이용될 수 있는 애플리케이션들 및/또는 데이터를 포함할 수 있다. 자원(1226)은 또한 인터넷을 거쳐 그리고/또는 가입자 네트워크(셀룰러 또는 Wi-Fi 네트워크 등)를 통해 제공되는 서비스들을 포함할 수 있다.

[0114] 플랫폼(1224)은 컴퓨팅 디바이스(1202)를 다른 컴퓨팅 디바이스들과 연결시키는 자원 및 기능을 추상화할 수 있다. 플랫폼(1224)은 또한 플랫폼(1224)을 통해 구현되는 자원(122)에 대한 봉착된 요구에 대응하는 레벨의 스케일을 제공하기 위해 자원의 스케일링을 추상화하는 역할을 할 수 있다. 그에 따라, 상호연결된 디바이스 환경에서, 본원에 기술되는 기능의 구현은 시스템(1200) 전체에 걸쳐 분산되어 있을 수 있다. 예를 들어, 기능이 부분적으로 컴퓨팅 디바이스(1202)에서는 물론 클라우드(1222)의 기능을 추상화하는 플랫폼(1224)을 통해서도 구현될 수 있다.

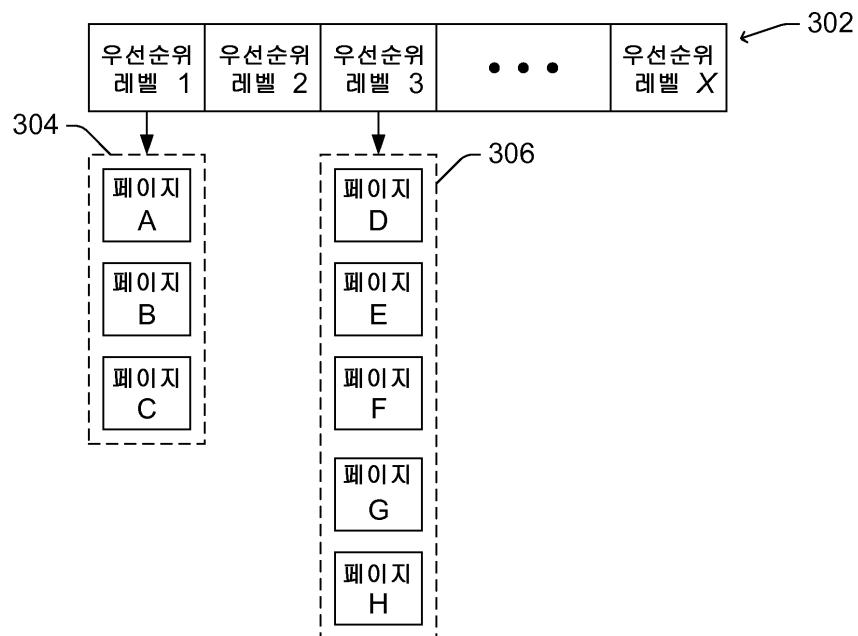
[0115] 발명 요지가 구조적 특징 및/또는 방법 동작과 관련하여 기술되어 있지만, 첨부된 청구항에 한정되어 있는 발명 요지가 앞서 기술된 구체적인 특징 또는 동작으로 꼭 제한될 필요는 없다는 것을 잘 알 것이다. 오히려, 앞서 기술된 구체적인 특징 및 동작은 청구항들을 구현하는 예시적인 형태로서 개시되어 있다.

도면**도면1**100

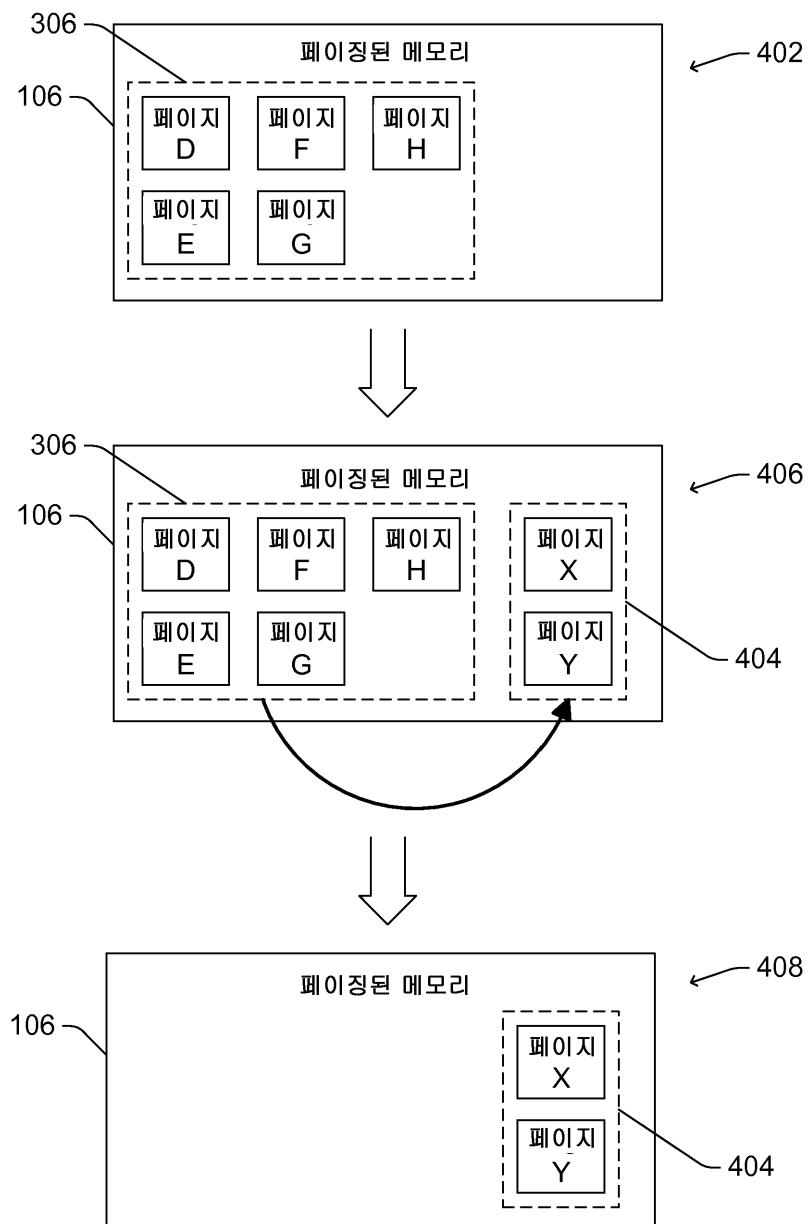
도면2

200

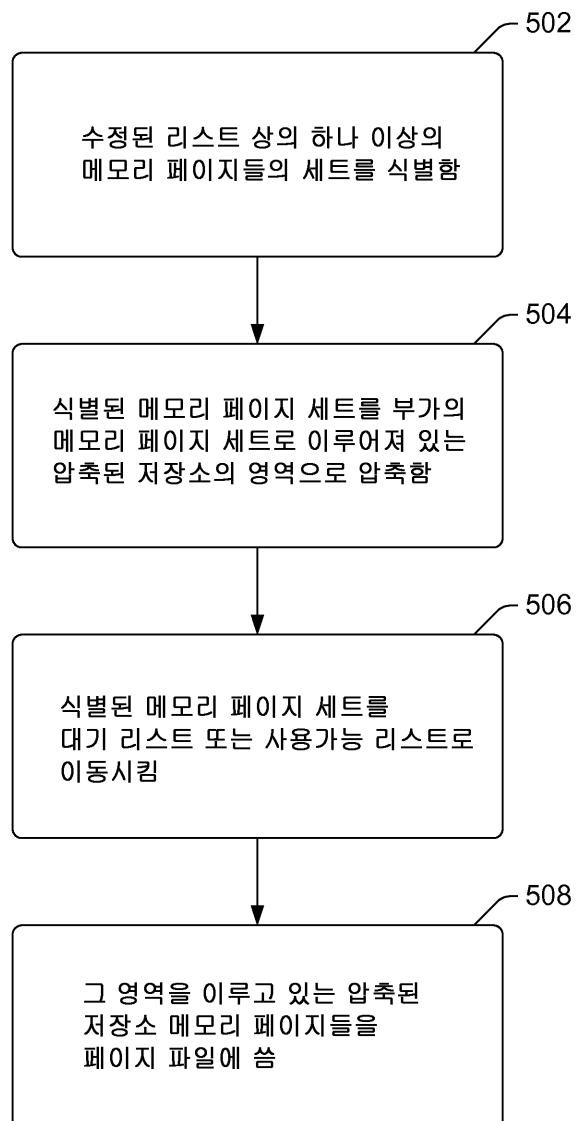
도면3

300

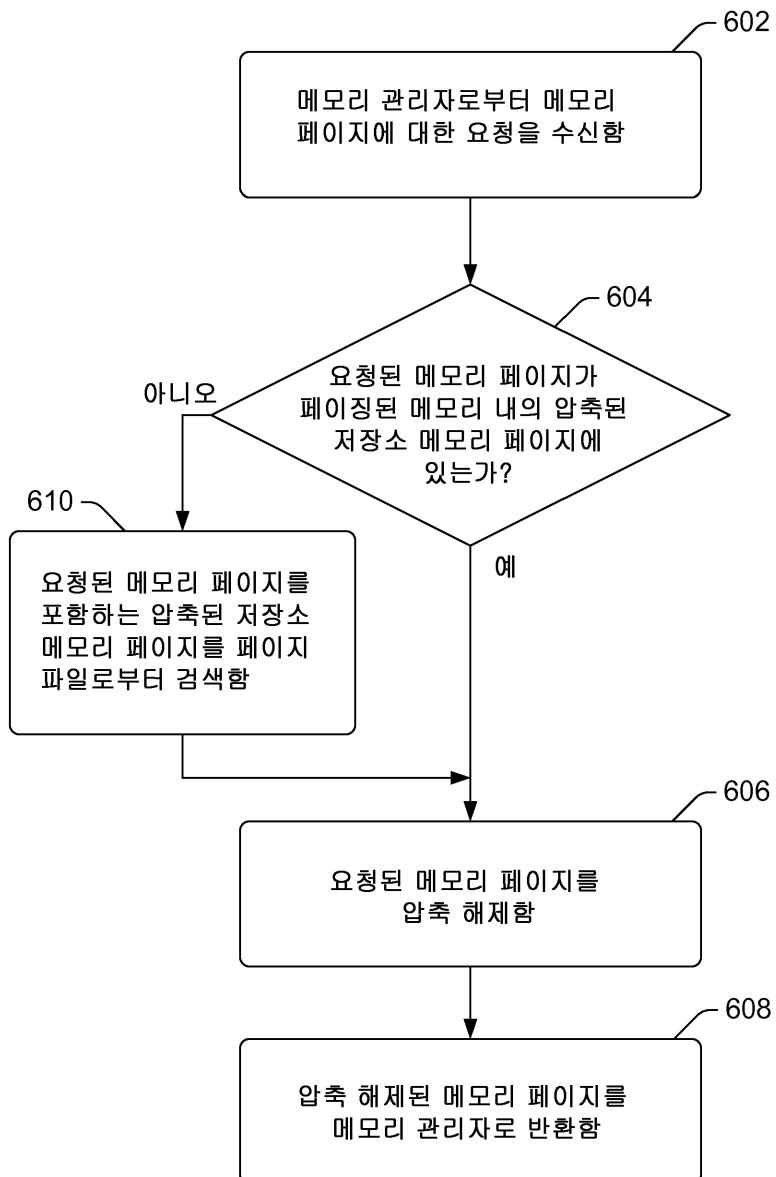
도면4



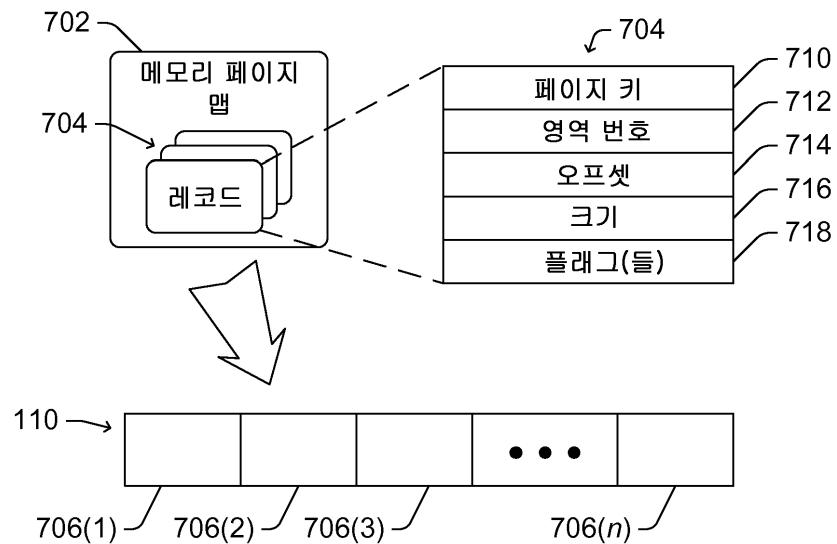
도면5



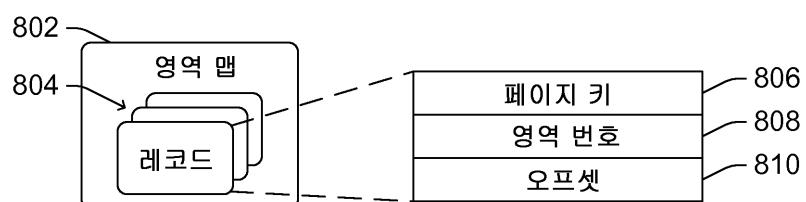
도면6

600

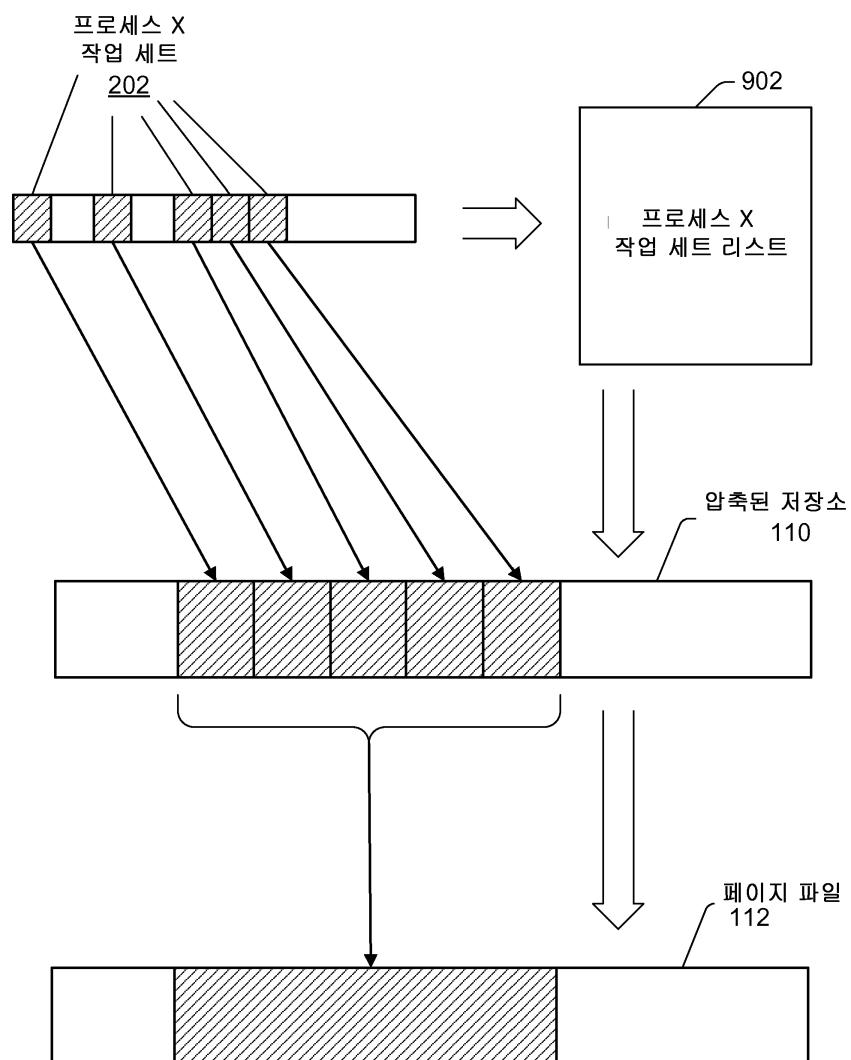
도면7



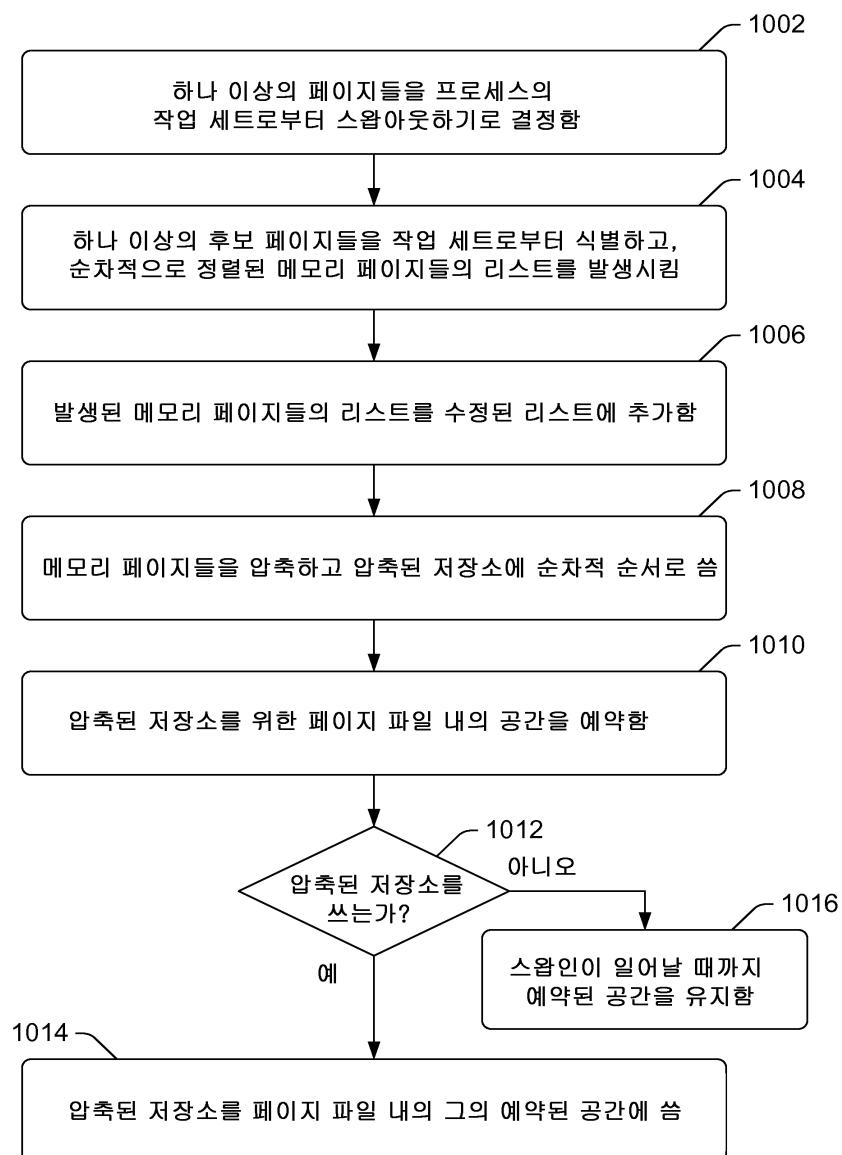
도면8



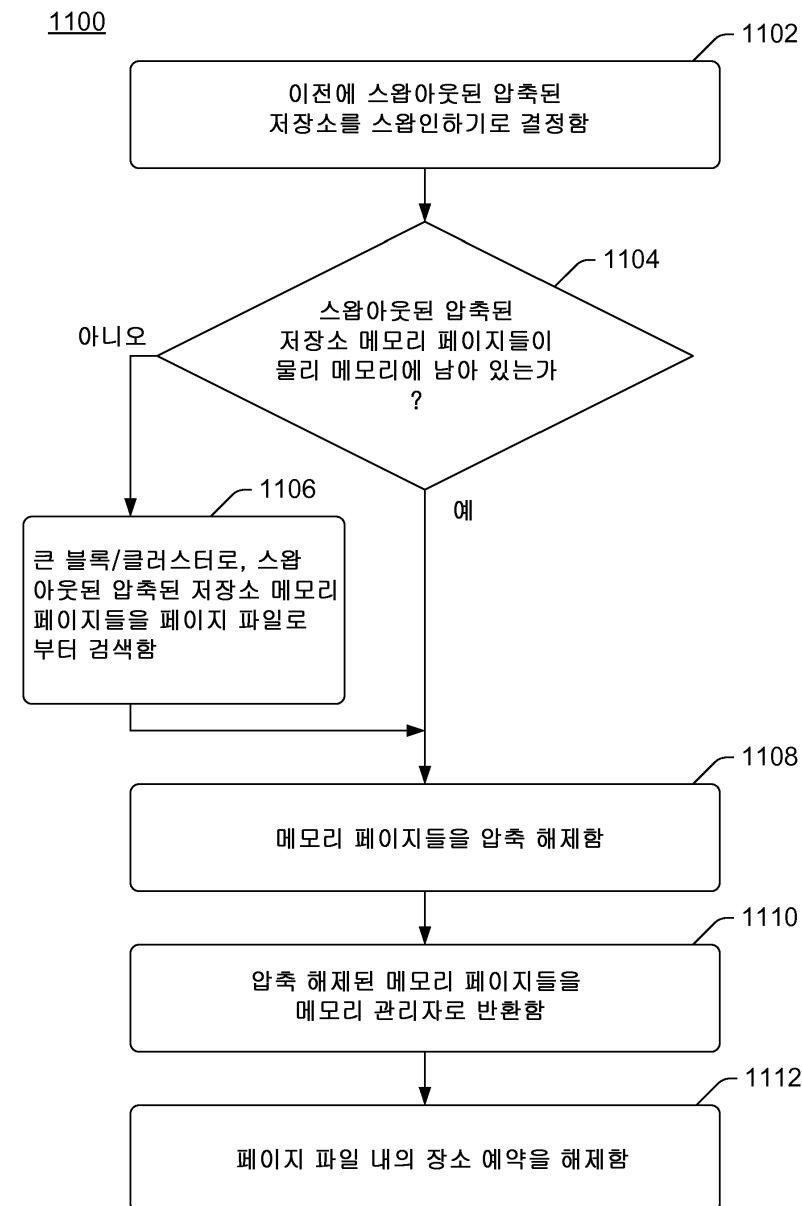
도면9



도면10

1000

도면11



도면12

1200