



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년06월03일
(11) 등록번호 10-1038963
(24) 등록일자 2011년05월30일

(51) Int. Cl.
G06F 12/08 (2006.01)
(21) 출원번호 10-2005-7018846
(22) 출원일자(국제출원일자) 2004년03월12일
심사청구일자 2008년06월25일
(85) 번역문제출일자 2005년10월04일
(65) 공개번호 10-2006-0006794
(43) 공개일자 2006년01월19일
(86) 국제출원번호 PCT/US2004/007655
(87) 국제공개번호 WO 2004/095291
국제공개일자 2004년11월04일
(30) 우선권주장
10/406,798 2003년04월02일 미국(US)
(56) 선행기술조사문헌
US20020129211 A1
US20030009623 A1
US20030009627 A1
전체 청구항 수 : 총 51 항

(73) 특허권자
인텔 코퍼레이션
미국 캘리포니아주 95052-8119 산타클라라
피.오.박스 58119 미션 칼리지 불바드 2200
(72) 발명자
나라드 찰스
미국 캘리포니아주 94024 로스 알토스 아보레텀
드라이브 5872
(74) 대리인
제일광장특허법인, 김원준

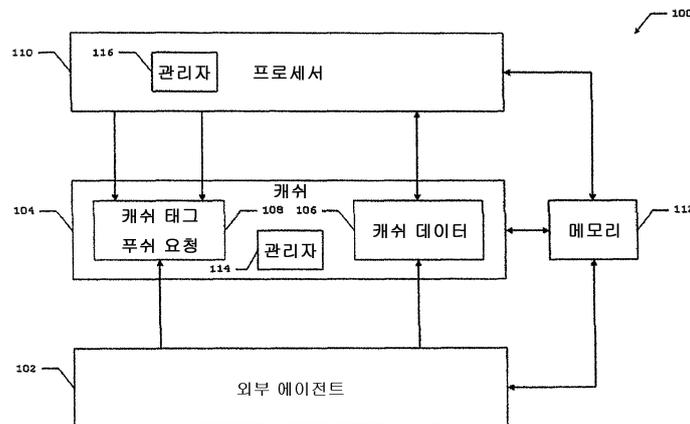
심사관 : 임영희

(54) 캐쉬 할당을 위한 장치, 시스템, 방법 및 기계 액세스가능 매체

(57) 요약

캐쉬 할당은 캐쉬 메모리와, 외부 에이전트가 캐쉬 메모리에 데이터가 위치되도록 요청하는 것을 허용하고, 프로세서가 캐쉬 메모리 내로 데이터를 풀링(pulling)하게 허용하도록 구성된 캐쉬 관리 메커니즘을 포함한다.

대표도 - 도1



특허청구의 범위

청구항 1

캐쉬 메모리 및 캐쉬 태그 어레이를 포함하는 캐쉬와,
 상기 캐쉬를 통해 주 메모리로의 액세스를 발행하도록 구성되는 프로세서와,
 캐쉬 관리 메커니즘을 포함하되,
 상기 캐쉬 관리 메커니즘은,

상기 주 메모리로 기록 액세스(write access)를 직접 발행할 수 있는 외부 에이전트가, 상기 캐쉬 메모리 내에 데이터가 위치되고 상기 캐쉬 태그 어레이 내에 상기 데이터와 연관된 캐쉬 태그가 위치될 것을 요청하는 것을 가능케 하되, 상기 외부 에이전트에 의한 요청은 상기 데이터를 식별하라는 상기 프로세서의 요청 및 상기 데이터를 식별하라는 캐쉬 요청에 응답하는 것이 아니고, 상기 외부 에이전트 요청에 응답하여 상기 캐쉬 관리 메커니즘이 캐쉬 라인을 희생시키고(victimize) 새로운 캐쉬 라인을 상기 데이터에 대해 할당하며,

상기 프로세서가 데이터로 하여금 상기 캐쉬 메모리로 풀링(pulling)되도록 하는 것을 가능케 하도록 구성되는 장치.

청구항 2

제 1 항에 있어서,
 상기 캐쉬 관리 메커니즘에 액세스가능하고, 데이터가 상기 캐쉬 메모리 내에 위치되는 시기를 결정하도록 구성되는 스로틀링(throttling) 메커니즘을 더 포함하는 장치.

청구항 3

제 1 항에 있어서,
 상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 포함된 데이터와 주 메모리에서 유지된 데이터의 카피 사이의 코히어런스(coherence)를 유지하도록 구성되는 장치.

청구항 4

제 3 항에 있어서,
 상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 포함된 데이터와 하나 이상의 다른 캐쉬에 포함된 데이터 사이의 코히어런스를 유지하도록 구성되는 장치.

청구항 5

제 4 항에 있어서,
 상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트로부터 상기 캐쉬 메모리로 전달된 데이터에 대응하는 상기 하나 이상의 다른 캐쉬 내의 데이터를 무효화시키도록 구성되는 장치.

청구항 6

제 4 항에 있어서,
 상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트로부터 상기 캐쉬 메모리로 전달된 데이터에 대응하는 상기

하나 이상의 다른 캐쉬 내의 데이터를 갱신하도록 구성되는 장치.

청구항 7

제 1 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트가 상기 캐쉬 메모리 내에 유지된 데이터의 카피를 저장하는 주 메모리를 갱신하는 것을 허용하도록 구성되는

장치.

청구항 8

삭제

청구항 9

제 1 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트가 상기 캐쉬 메모리에 포함된 현재 데이터가 덮어쓰이도록 (overwritten) 하는 것을 허용하도록 구성되는

장치.

청구항 10

제 9 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 위치한 데이터를 수정된 코히어런스 상태로 위치시키도록 구성되는

장치.

청구항 11

제 10 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 위치한 데이터를 배타적 코히어런스 상태로 위치시키도록 구성되는

장치.

청구항 12

제 10 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 위치한 데이터를 공유된 코히어런스 상태로 위치시키도록 구성되는

장치.

청구항 13

제 9 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 위치한 데이터를 클린(clean) 코히어런스 상태로 위치시키도록 구성되는

장치.

청구항 14

제 13 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 위치된 데이터를 배타적 코히어런스 상태로 위치시키도록 구성되는

장치.

청구항 15

제 13 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 캐쉬 메모리에 위치된 데이터를 공유된 코히어런스 상태로 위치시키도록 구성되는

장치.

청구항 16

제 1 항에 있어서,

상기 캐쉬 관리 메커니즘이 상기 외부 에이전트가 또한 데이터가 그 내부에 위치되도록 요청하는 것을 허용하도록 구성되는 적어도 하나의 다른 캐쉬 메모리를 더 포함하는

장치.

청구항 17

제 16 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트가 그 내부에 위치될 데이터를 위해 상기 적어도 하나의 다른 캐쉬 메모리 중 적어도 하나에서의 라인 할당을 요청하는 것을 허용하도록 구성되는

장치.

청구항 18

제 16 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트가 그 내부에 위치될 데이터를 위해 복수의 다른 캐쉬 메모리에서의 라인 할당을 요청하는 것을 허용하도록 구성되는

장치.

청구항 19

제 16 항에 있어서,

상기 캐쉬 관리 메커니즘은 또한 상기 외부 에이전트가 상기 다른 캐쉬 메모리 또는 캐쉬 메모리들에 포함된 현재의 데이터가 덮어쓰이도록 하는 것을 허용하도록 구성되는

장치.

청구항 20

제 1 항에 있어서,

상기 캐쉬 메모리는, 주 메모리를 모방하고, 상기 주 메모리에 액세스하고자 시도할 때 다른 캐쉬가 액세스할 수 있는 캐쉬를 포함하는

장치.

청구항 21

제 20 항에 있어서,

상기 캐쉬 메모리에 포함된 라인은 다른 캐쉬에 의한 판독 동작 이후에 할당 해제되는

장치.

청구항 22

제 20 항에 있어서,
 다른 캐쉬에 의한 관독 동작 이후에 라인이 공유된 상태로 변화되는
 장치.

청구항 23

제 1 항에 있어서,
 상기 외부 에이전트는 입/출력 장치를 포함하는
 장치.

청구항 24

제 1 항에 있어서,
 상기 외부 에이전트는 다른 프로세서를 포함하는
 장치.

청구항 25

제 1 항에 있어서,
 상기 데이터는 적어도 하나의 네트워크 통신 프로토콜 데이터 유닛의 적어도 일부의 데이터를 포함하는
 장치.

청구항 26

주 메모리로 기록 요청을 직접 발행할 수 있는 외부 에이전트가, 캐쉬 메모리 내에 데이터가 위치되고 캐쉬 태그 어레이 내에 연관된 태그가 위치되도록 하라는 요청을 발행하는 것을 가능케 하는 단계와,
 상기 외부 에이전트가 상기 캐쉬 메모리에 위치될 상기 데이터 및 상기 캐쉬 태그 어레이 내에 위치될 상기 태그를 제공하는 것을 가능케 하는 단계를 포함하되,
 상기 캐쉬 메모리 내에 데이터가 위치되고 캐쉬 태그 어레이 내에 연관된 태그가 위치되도록 하라는 상기 요청은 상기 데이터를 식별하라는 프로세서의 요청 및 상기 데이터를 식별하라는 캐쉬 요청에 응답하는 것이 아니고,
 데이터가 상기 캐쉬 메모리 내에 위치되도록 하라는 상기 요청에 응답하여, 캐쉬 관리 메커니즘이 캐쉬 라인을 희생시키고 새로운 캐쉬 라인을 상기 데이터에 대해 할당하며,
 상기 프로세서는 상기 캐쉬를 통해 상기 주 메모리로의 액세스를 발행하도록 구성되는
 방법.

청구항 27

제 26 항에 있어서,
 상기 프로세서가 데이터로 하여금 상기 캐쉬 메모리로 폴링되도록 하는 것을 가능케 하는 단계를 더 포함하는
 방법.

청구항 28

제 26 항에 있어서,
 상기 캐쉬 메모리가 상기 데이터에 대해 상기 캐쉬 메모리를 체크하고, 상기 캐쉬 메모리가 상기 데이터를 포함

하지 않는 경우, 주 메모리로부터 상기 데이터를 요청하는 것을 가능케 하는 단계를 더 포함하는 방법.

청구항 29

제 26 항에 있어서,
상기 외부 에이전트가 상기 캐쉬 메모리에 위치될 데이터를 제공할 시기를 결정하는 단계를 더 포함하는 방법.

청구항 30

제 26 항에 있어서,
상기 외부 에이전트가 상기 캐쉬 메모리 내의 데이터를 위한 위치를 선택할 것을 상기 캐쉬 메모리에 요청하는 것을 가능케 하는 단계를 더 포함하는 방법.

청구항 31

제 26 항에 있어서,
상기 캐쉬 메모리를 주 메모리 내의 데이터의 어드레스로 갱신하는 단계를 더 포함하는 방법.

청구항 32

제 26 항에 있어서,
상기 캐쉬 메모리를 상기 데이터의 상태로 갱신하는 단계를 더 포함하는 방법.

청구항 33

제 26 항에 있어서,
상기 외부 에이전트로부터, 주 메모리를 상기 데이터로 갱신하는 단계를 더 포함하는 방법.

청구항 34

실행가능한 인스트럭션을 저장하는 기계 액세스가능 매체에 있어서,
상기 인스트럭션은 기계로 하여금,
주 메모리로 기록 액세스를 직접 발행할 수 있는 외부 에이전트가, 캐쉬 메모리 내에 데이터가 위치되고 캐쉬 태그 어레이 내에 연관된 태그가 위치되도록 하라는 요청을 발행하는 것을 가능케 하고,
상기 외부 에이전트가 상기 캐쉬 메모리를 상기 데이터로 채우고 상기 캐쉬 태그 어레이를 상기 태그로 채우는 것을 가능케 하되,
상기 캐쉬 메모리 내에 데이터가 위치되도록 하라는 상기 요청은 상기 데이터를 식별하라는 프로세서의 요청 및 상기 데이터를 식별하라는 캐쉬 요청에 응답하는 것이 아니고,
데이터가 상기 캐쉬 메모리 내에 위치되도록 하라는 상기 요청에 응답하여, 캐쉬 관리 메커니즘이 캐쉬 라인을 희생시키고 새로운 캐쉬 라인을 상기 데이터에 대해 할당하며,
상기 프로세서는 상기 캐쉬를 통해 상기 주 메모리로의 액세스를 발행하도록 구성되는 기계 액세스가능 매체.

청구항 35

제 34 항에 있어서,

기계로 하여금, 상기 프로세서가 데이터로 하여금 상기 캐쉬 메모리로 폴딩되도록 하는 것을 추가로 가능케 하는

기계 액세스가능 매체.

청구항 36

제 34 항에 있어서,

기계로 하여금 또한, 캐쉬 메모리가 상기 데이터에 대해 상기 캐쉬 메모리를 체크하고, 상기 캐쉬 메모리가 상기 데이터를 포함하지 않는 경우, 주 메모리로부터 상기 데이터를 요청하는 것을 가능케 하는

기계 액세스가능 매체.

청구항 37

제 34 항에 있어서,

기계로 하여금 또한, 상기 외부 에이전트가 상기 캐쉬 메모리 내의 상기 데이터를 위한 위치를 선택할 것을 상기 캐쉬 메모리에게 요청하는 것을 가능케 하는

기계 액세스가능 매체.

청구항 38

캐쉬 메모리 및 캐쉬 태그 어레이를 포함하는 캐쉬와,

주 메모리로 기록 액세스를 직접 발행할 수 있는 외부 에이전트가, 상기 캐쉬 메모리의 하나의 라인을 희생 (victim)으로서 선택 -상기 라인은 데이터를 포함함- 하고, 상기 캐쉬 메모리 내의 상기 데이터를 상기 외부 에이전트로부터의 새로운 데이터로 대체하며, 상기 새로운 데이터와 연관된 태그를 상기 캐쉬 태그 어레이에 기록할 것을 상기 캐쉬 메모리에게 요청하는 것을 허용하도록 구성된 메모리 관리 메커니즘을 포함하되,

상기 외부 에이전트 요청은 상기 데이터를 식별하라는 프로세서의 요청 및 상기 데이터를 식별하라는 캐쉬 요청에 응답하는 것이 아니고,

상기 프로세서는 상기 캐쉬를 통해 상기 주 메모리로의 액세스를 발행하도록 구성되는

시스템.

청구항 39

제 38 항에 있어서,

상기 메모리 관리 메커니즘은 또한 상기 외부 에이전트가 상기 캐쉬 메모리를 상기 새로운 데이터의 주 메모리 내의 위치로 갱신하는 것을 허용하도록 구성되는

시스템.

청구항 40

제 39 항에 있어서,

상기 메모리 관리 메커니즘은 또한 외부 에이전트로 하여금 상기 주 메모리를 상기 새로운 데이터로 갱신하는 것을 허용하도록 구성되는

시스템.

청구항 41

제 39 항에 있어서,

프로세서와,

상기 프로세서에 포함되며, 상기 캐쉬 메모리에 대한 상기 프로세서의 액세스를 관리하도록 구성되는 캐쉬 관리 메커니즘을 더 포함하는

시스템.

청구항 42

제 39 항에 있어서,

적어도 하나의 추가적인 캐쉬 메모리를 더 포함하되, 상기 메모리 관리 메커니즘은 또한 상기 외부 에이전트가 그들 각각의 추가적인 캐쉬 메모리에서 라인을 할당하기 위해 상기 추가적인 캐쉬 메모리의 일부 또는 전부를 요청하는 것을 허용하도록 구성되는

시스템.

청구항 43

제 42 항에 있어서,

상기 메모리 관리 메커니즘은 또한 상기 외부 에이전트로부터의 상기 새로운 데이터에 대응하는 상기 추가적인 캐쉬 메모리 또는 메모리들 내의 데이터를 갱신하도록 구성되는

시스템.

청구항 44

제 39 항에 있어서,

상기 캐쉬 메모리에 포함된 데이터의 마스터 카피를 저장하도록 구성된 상기 주 메모리를 더 포함하는

시스템.

청구항 45

제 39 항에 있어서,

적어도 하나의 추가적인 외부 에이전트를 더 포함하되, 상기 메모리 관리 메커니즘은 상기 추가적인 외부 에이전트 각각이, 상기 캐쉬 메모리의 라인을 희생으로서 선택-상기 라인은 데이터를 포함함-하고, 상기 요청을 수행한 상기 추가적인 외부 에이전트로부터의 새로운 데이터로 상기 데이터를 대체할 것을 상기 캐쉬 메모리에 게 요청하는 것을 허용하도록 구성되는

시스템.

청구항 46

제 39 항에 있어서,

상기 외부 에이전트는 또한 상기 새로운 데이터의 단지 일부만을 상기 캐쉬 메모리 내로 푸쉬하도록 구성되는

시스템.

청구항 47

제 46 항에 있어서,

상기 새로운 데이터의 일부를 푸쉬하도록 구성된 네트워크 인터페이스를 더 포함하는

시스템.

청구항 48

제 46 항에 있어서,

상기 외부 에이전트는 또한 상기 캐쉬 메모리 내로 푸쉬되지 않은 새로운 데이터를 주 메모리 부분에 기록하도록 구성되는

시스템.

청구항 49

제 39 항에 있어서,

상기 데이터는 서술자(descriptor)를 포함하는

시스템.

청구항 50

적어도 하나의 물리 계층(PHY) 디바이스와,

상기 PHY를 통해 수신된 데이터에 대해 연결 계층 동작을 수행하기 위한 적어도 하나의 이더넷 매체 액세스 제어 어기(MAC)와,

주 메모리와,

캐쉬를 통해 상기 주 메모리로의 액세스를 발행하도록 구성되는 프로세서와,

상기 적어도 하나의 PHY 및 적어도 하나의 MAC를 통해 수신된 데이터의 적어도 일부가 상기 캐쉬 내에서 캐쉬될 것을 요청하기 위한 로직과,

상기 프로세서에 연결되는 상기 캐쉬를 포함하되,

상기 요청은 태그와 상기 데이터의 적어도 일부를 식별하라고 하며, 상기 요청 로직은 상기 데이터를 식별하라는 프로세서의 요청 및 상기 데이터를 식별하라는 캐쉬 요청에 응답하는 것이 아니고,

상기 요청 로직에 응답하여 상기 캐쉬 관리 메커니즘이 캐쉬 라인을 회생시키고 새로운 캐쉬 라인을 상기 데이터에 대해 할당하며,

상기 로직은 상기 주 메모리로 직접 기록할 수 있고,

상기 캐쉬는,

캐쉬 태그 어레이와,

캐쉬 메모리와,

캐쉬 관리 메커니즘

을 포함하며,

상기 캐쉬 관리 메커니즘은,

상기 요청에 응답하여, 상기 적어도 하나의 PHY 및 적어도 하나의 MAC를 통해 수신된 데이터의 적어도 일부를 상기 캐쉬 메모리 내에 위치시키고, 상기 태그를 상기 캐쉬 태그 어레이에 기록하며,

상기 캐쉬 메모리에 저장되지 않은 데이터에 대한 요청에 응답하여, 상기 프로세서로 하여금 데이터가 상기 캐쉬 메모리 내로 풀링되도록 하는 것을 허용하도록

구성되는

시스템.

청구항 51

제 50 항에 있어서,

상기 로직은 네트워크 프로세서에 의해 제공된 스레드(thread)들의 집합 중 적어도 하나의 스레드를 포함하는

시스템.

청구항 52

제 50 항에 있어서,

상기 캐쉬로부터 검색된 데이터에 대해 브리징, 라우팅, QoS(quality of service)의 결정, 흐름 결정 및 필터링을 포함하는 패킷 처리 동작들 중 적어도 하나를 수행하기 위한 로직을 더 포함하는 시스템.

명세서

배경 기술

[0001] 컴퓨터 시스템에서의 프로세서는 메모리 내의 요청된 위치에서의 데이터에 대한 요청을 발행할 수 있다. 프로세서는 우선, 주 메모리에 대한 전형적으로 더 느린 액세스를 통해서가 아닌, 프로세서와 밀접하게 관련된 메모리, 예를 들면, 캐쉬 내의 데이터를 액세스하고자 시도할 수 있다. 일반적으로, 캐쉬는 보다 크고, 보다 느린 주 메모리의 선택된 영역 또는 블록을 대행하는 메모리를 포함한다. 전형적으로, 캐쉬는 요구에 따라 채워지며, 물리적으로 프로세서에 보다 근접하고, 주 메모리보다 빠른 액세스 시간을 갖는다.

[0002] 메모리에 대한 프로세서의 액세스가 캐쉬에서 "미스(misses)", 예를 들면, 캐쉬 내에서 데이터의 카피를 찾을 수 없다면, 캐쉬는 주 메모리의 요청된 위치에서의 데이터를 모방하는 데이터를 저장하기 위해 캐쉬 내에서 위치를 선택하고, 요청된 위치에서의 데이터에 대한 요청을 주 메모리에게 발행하며, 선택된 캐쉬 위치를 주 메모리로부터의 데이터로 채운다. 또한, 캐쉬는 요청된 위치에 대해 공간적으로 근접하여 위치된 데이터를 요청하여, 요청 데이터가 때로는 동일하거나 공간적으로 근접한 메모리 위치로부터의 데이터에 대해 시간적으로 근접한 요청이 되도록 하는 프로그램으로서 저장함으로써, 공간적으로 근접한 데이터를 캐쉬에 포함시키는 효율성을 증가시킬 수 있다. 이러한 방식으로, 프로세서는 데이터에 대한 이러한 요청 및/또는 후속하는 요청을 위해 캐쉬 내의 데이터를 액세스할 수 있다.

발명의 상세한 설명

[0007] 도 1을 참조하면, 예시적인 시스템(100)은 캐쉬 메모리(104)("캐쉬(104)")의 라인의 할당을 요청할 수 있는 외부 에이전트(external agent)(102)를 포함한다. 외부 에이전트(102)는 캐쉬(104)에 포함된 데이터 메모리(106) 내에 데이터를 푸시하고, 캐쉬(104)에 포함된 태그 어레이(108) 내로 태깅(tagging)한다. 또한, 외부 에이전트(102)는 추가적인 국부적 및/또는 원격 캐쉬에서 라인 할당을 트리거하고/하거나 코히어런트 갱신 및/또는 코히어런트 무효화할 수 있다. 캐쉬(104)의 라인 할당을 트리거하고, 캐쉬(104) 내로의 데이터의 전달을 요청하기 위해, 외부 에이전트(102) 인에이블링함으로써, 제 1 캐쉬 액세스 미스와 관련된 패널티를 감소 또는 제거할 수 있다. 예를 들어, 프로세서(110)는 메모리(112) 내의 데이터를 외부 에이전트(102) 및 하나 이상의 다른 외부 에이전트(예를 들면, 입력/출력(I/O) 장치 및/또는 다른 프로세서)와 공유하고, 다른 에이전트에 의해 단지 기록된 데이터를 액세스하도록 캐쉬 미스를 초래할 수 있다. 캐쉬 관리 메커니즘(114)("관리자(114)")은, 공간 할당을 트리거링하고, 데이터를 캐쉬(104) 내로 전달하여, 외부 에이전트(102)가 프로세서(110)를 대신하여 데이터의 프리페치(prefetch)를 모방하도록 허용함으로써, 캐쉬 미스를 감소시키는 것을 돕는다. 전형적으로, 캐쉬 동작은 프로세서(110)에게 투명하다. 관리자(114)와 같은 관리자는 특정 캐쉬 및 메모리 전송의 협동적인 관리를 가능하게 하여, 2개의 에이전트 사이의 메모리 기반 메시지 통신의 성능을 향상시킨다. 관리자(114)는 수신 서술자 및 수신 버퍼의 선택된 부분을, 네트워크 인터페이스로부터 지정된 프로세서로 통신하는데 이용될 수 있다. 또한, 관리자(114)는 프로세서간(inter-processor) 또는 스레드간(inter-thread) 메시지의 비용을 최소화하는데 이용될 수 있다. 프로세서(110)는 관리자, 예를 들면, 캐쉬 관리 메커니즘(관리자)(116)을 또한 포함할 수 있다.

[0008] 관리자(114)는, 프로세서(110)의 요청에 따라 캐쉬(104)에서 데이터를 채우게 하는 것을 허용하며, 데이터를 채우는 것은 데이터를 캐쉬(104) 내로 풀링(pulling)하고, 데이터를 기록하거나, 데이터를 저장하는 것을 포함할 수 있다. 예를 들어, 프로세서(110)가 주 메모리(112)("메모리(112)") 내의 위치에서의 데이터에 대한 요청을 생성하고, 캐쉬(104) 내에서의 메모리 위치에 대한 프로세서(100)의 액세스가 미스되는 경우, 캐쉬(104)는 전형

적으로 관리자(114)를 이용하여, 메모리(112) 내의 요청된 위치에서의 데이터의 카피를 포함하기 위해 캐쉬(104) 내의 위치를 선택하고, 요청된 위치의 내용에 대한 요청을 메모리(112)에게 발행할 수 있다. 선택된 위치는 새롭게 할당된 라인에 의해 치환되거나 희생되는 상이한 메모리 위치를 나타내는 캐쉬 데이터를 포함할 수 있다. 코히어런트 멀티프로세서 시스템의 예에서, 메모리(112)에 대한 요청은 캐쉬(104)와는 상이한 프로세서 캐쉬와 같은, 메모리(112)가 아닌 에이전트로부터 만족될 수 있다.

[0009] 또한, 관리자(114)는 캐쉬(104) 내의 데이터의 카피가 메모리(112)에 아직 반영되지 않은 갱신 또는 수정을 포함하는 경우, 선택된 위치에서의 내용을 무시하거나 또는 선택된 위치에서의 내용을 메모리(112)에 되기록(writeback)함으로써, 외부 에이전트(102)가 캐쉬(104)를 트리거하여, 캐쉬(104)에 의해 선택된 캐쉬(104) 내의 위치에서의 현재 데이터를 희생시키는 것을 허용할 수 있다. 캐쉬(104)는 메모리(112)에 대한 희생 및 되기록을 수행하지만, 외부 에이전트(102)는 캐쉬(104)에 대한 요청을 전달하여 캐쉬(104)에 데이터를 저장함으로써, 이들 이벤트를 트리거할 수 있다. 예를 들어, 외부 에이전트(102)는 데이터를 캐쉬(104)에 저장하기 전에 메모리(112)에 대한 잠재적인 관독을 회피하면서, 캐쉬(104)에 저장될 데이터 및 데이터에 대한 어드레스 정보를 포함하는 푸쉬 코멘드를 전송할 수 있다. 외부 에이전트(102)로부터의 푸쉬 요청에서 지시된 메모리(106) 내의 위치를 나타내는 엔트리를 캐쉬(104)가 이미 포함하는 경우, 캐쉬(104)는 새로운 위치를 할당하지도 않고, 어떠한 캐쉬 내용도 희생시키지 않는다. 그 대신에, 캐쉬(104)는 매칭 태그를 갖는 위치를 이용하여, 외부 에이전트(102)로부터 푸쉬된 데이터를 갖는 대응하는 데이터를 덮어쓰기하며, 대응하는 캐쉬 라인 상태를 갱신한다. 코히어런트 멀티프로세서 시스템에서, 푸쉬 요청에서 지시된 위치에 대응하는 엔트리를 갖는 캐쉬(104)가 아닌 캐쉬가, 이들 엔트리를 무시하거나, 또는 이들을 푸쉬된 데이터 및 새로운 상태로 갱신함으로써, 시스템 캐쉬 코히어런스를 유지한다.

[0010] 프로세서(110)가 요구에 따라 캐쉬(104)를 채우도록 인에이블링하면서, 캐쉬(104)에 의한 라인 할당을 트리거하도록 외부 에이전트(102)를 인에이블링하는 것은, 중요한 새로운 데이터와 같은 중요한 데이터가 캐쉬(104) 내의 프로세서(110)에 시간적으로 보다 근접하여 선택적으로 위치되도록 허용함으로써, 프로세서 성능을 향상시킨다. 일반적으로, 라인 할당은 캐쉬 채움 동작 실행의 처리에서 희생시키기 위해 라인을 선택하고, 내용이 수정되는 경우 희생 캐쉬 내용을 주 메모리에 기록하고, 할당 에이전트에 의해 선택된 새로운 주 메모리 어드레스를 반영하고, 되기록 또는 캐쉬 코히어런스와 관련된 것과 같은 상태 정보를 반영하도록 필요에 따라 캐쉬 라인 상태를 갱신하기 위해 태그 정보를 갱신하고, 캐쉬 내의 대응하는 데이터 블록을 요청 에이전트에 의해 발행된 새로운 데이터로 대체하는 것 중 일부 또는 전부를 수행하는 것을 의미한다.

[0011] 데이터는 외부 에이전트(102)로부터 캐쉬(104)로, "더티(dirty)" 또는 "클린(clean)"으로서 전달될 수 있다. 데이터가 더티로서 전달된다면, 캐쉬(104)는 라인이 결국 캐쉬(104)로부터 희생될 때의 해당 메모리 위치를 나타내는 캐쉬 데이터의 현재 값으로 메모리(112)를 갱신한다. 데이터는 그것이 캐쉬(104) 내로 푸시된 후, 프로세서(110)에 의해 수정되거나 수정되지 않을 수 있다. 데이터가 클린으로서 전달된다면, 캐쉬(104)가 아닌 메커니즘, 즉 본 예에서의 외부 에이전트(102)는 메모리(112)를 데이터로 갱신할 수 있다. "더티" 또는 어떠한 동등한 상태는, 이러한 캐쉬가 현재 해당 데이터 위치에서 데이터의 가장 최근의 카피를 갖고 있음을 나타내고, 캐쉬(104)로부터 데이터가 방출될 때, 메모리(112)가 갱신되는 것을 보장할 책임이 있다. 멀티프로세서 코히어런트 시스템에서, 예를 들면, 다른 프로세서가 메모리(112) 내의 해당 위치에 기록하고자 시도하는 경우, 해당 캐쉬의 요청시에 상이한 캐쉬로 책임이 옮겨질 수 있다.

[0012] 캐쉬(104)는 데이터 메모리(106)로/로부터 데이터를 기록 및 관독할 수 있다. 또한, 캐쉬(104)는 태그 어레이(108)를 액세스하고, 상태 정보를 생성하여 수정하고, 태그를 생성하고, 희생이 되도록 할 수 있다.

[0013] 외부 에이전트(102)는 데이터의 중요 부분(예를 들면, 최초에 액세스된 부분, 빈번하게 액세스된 부분, 계속적으로 액세스되는 부분 등)에 대한 액세스 대기 시간을 숨기거나 감소시키면서, 캐쉬(104)를 통해 새로운 정보를 프로세서(110)에게 전송한다. 외부 에이전트(102)는 (예를 들면, 캐쉬(104)에서의) 데이터의 수신자에 보다 근접한 데이터를 전달하여, 수신자에 대한 메시징 비용을 감소시킨다. 강제된 미스(compelled misses)로 인해 스톨된(stalled) 프로세서(110)가 소비하는 시간량을 감소시키는 것은 프로세서 성능을 향상시킨다. 시스템(100)이 다수의 캐쉬를 포함한다면, 관리자(114)는 프로세서(110) 및/또는 외부 에이전트(102)가 일부의 또는 모든 캐쉬에서의 라인 할당을 요청하도록 허용할 수 있다. 대안적으로, 단지 선택된 캐쉬 또는 캐쉬들만이 푸쉬 데이터를 수신하고, 다른 캐쉬들은 캐쉬 코히어런스를 유지하기 위해, 예를 들면, 푸쉬 요청의 어드레스를 매칭시키는 태그를 포함하는 엔트리를 갱신 또는 무시함으로써, 적절한 동작을 취한다.

[0014] 외부 에이전트를 이용한 캐쉬 라인의 할당을 더 설명하기 전에, 시스템(100) 내의 요소들이 더 설명된다. 시

시스템(100) 내의 요소들은 다양함 방법으로 구현될 수 있다.

- [0015] 시스템(100)은 네트워크 시스템, 컴퓨터 시스템, 칩상의 고집적 I/O 서브시스템, 또는 다른 유사한 타입의 통신 또는 처리 시스템을 포함할 수 있다.
- [0016] 외부 에이전트(102)는 I/O 장치, 네트워크 인터페이스, 프로세서, 또는 캐쉬(104) 및 메모리(112)와 통신할 수 있는 다른 메커니즘을 포함할 수 있다. 일반적으로, I/O 장치는 데이터를 컴퓨터 시스템으로/으로부터 전달하는데 이용되는 장치를 포함한다.
- [0017] 캐쉬(104)는 메모리 액세스(예를 들면, 프로세서(110))와 저장 장치 혹은 주 메모리(예를 들면, 메모리(112))를 브리징할 수 있는 메모리 메커니즘을 포함할 수 있다. 전형적으로, 캐쉬(104)는 주 메모리보다 빠른 액세스 시간을 갖는다. 캐쉬(104)는 다수의 레벨을 포함할 수 있으며, 전용 캐쉬, 버퍼, 메모리 뱅크, 또는 다른 유사한 메모리 메커니즘을 포함할 수 있다. 캐쉬(104)는 독립 메커니즘을 포함할 수 있고, 또는 주 메모리의 준비된 섹션에 포함될 수도 있다. 전형적으로, 인스트럭션 및 데이터가 블록 내의 캐쉬(104)로/로부터 통신된다. 일반적으로, 블록은 그룹으로서 통신되거나 처리되는 비트 또는 바이트의 집합(collection)을 의미한다. 블록은 임의의 수의 워드를 포함할 수 있고, 워드는 임의의 수의 비트 또는 바이트를 포함할 수 있다.
- [0018] 데이터의 블록은 이더넷 또는 동기 광학 네트워크(Synchronous Optical Network; SONET) 프레임, TCP(Transmission Control Protocol) 세그먼트, IP(Internet Protocol) 패킷, 프래그먼트, ATM(Asynchronous Transfer Mode) 셀 등, 또는 그들의 일부분과 같은 하나 이상의 네트워크 통신 PDU(protocol data unit)의 데이터를 포함할 수 있다. 데이터의 블록은 서술자를 더 포함할 수 있다. 서술자는 외부 에이전트(102)와 같은 메시지 또는 패킷의 송신자가, 메시지 또는 PDU에 관한 정보를, 프로세서(110)와 같은 수신자에게 통신하기 위해 이용하는, 전형적으로 메모리 내의 데이터 구조이다. 서술자 내용은 메시지 또는 패킷을 포함하는 버퍼 또는 버퍼들의 위치(들), 버퍼(들) 내의 바이트의 수, 네트워크 포트가 이러한 패킷을 수신하는 신원(identification), 에러 지시 등을 포함할 수 있으나, 이것에 한정되는 것은 아니다.
- [0019] 데이터 메모리(106)는 주 메모리(예를 들면, 메모리(112))로부터 폐치된 데이터 정보를 저장하도록 구성된 캐쉬(104)의 일부를 포함할 수 있다.
- [0020] 태그 어레이(108)는 태그 정보를 저장하도록 구성된 캐쉬(104)의 일부를 포함할 수 있다. 태그 정보는 어느 주 메모리 어드레스가 데이터 메모리(106) 내의 대응하는 데이터 엔트리에 의해 표현되는지를 나타내는 어드레스 필드 및 대응하는 데이터 엔트리에 대한 상태 정보를 포함할 수 있다. 일반적으로, 상태 정보는 유효, 무효, 더티(대응하는 데이터 엔트리가, 주 메모리로부터 폐치된 이후에, 갱신 또는 수정된 것을 나타냄), 배제, 공유, 소유, 수정, 및 다른 유사한 상태들과 같은 데이터 상태를 나타내는 코드를 의미한다.
- [0021] 캐쉬(104)는 관리자(114)를 포함하며, 데이터 메모리(106) 및 태그 어레이(108)를 포함하는 단일의 메모리 메커니즘을 포함할 수 있고, 또는, 데이터 메모리(106) 및 태그 어레이(108)는 분리된 메모리 메커니즘일 수 있다. 데이터 메모리(106) 및 태그 어레이(108)가 분리된 메모리 메커니즘인 경우, "캐쉬(104)"는 데이터 메모리(106), 태그 어레이(108) 및 관리자(114) 중 적절한 것으로서 해석될 수 있다.
- [0022] 관리자(114)는 프로세서(110)가 아닌 에이전트에 의한 메모리에 대한 액세스에 응답하여, 요청된 어드레스를 태그와 비교하고, 히트(hit) 및 미스를 검출하고, 관독 데이터를 프로세서(110)에 제공하고, 프로세서(110)로부터 기록 데이터를 수신하고, 캐쉬 라인 상태를 관리하고, 코히어런트 동작을 지원하는 하드웨어 메커니즘을 포함할 수 있다. 또한, 관리자(114)는 외부 에이전트(102)로부터의 푸쉬 요청에 응답하기 위한 메커니즘을 포함한다. 또한, 관리자(114)는 프로세서(110)에 포함되거나 또는 프로세서(110)에 액세스가능한 소프트웨어와 같은, 캐쉬(104)의 관리를 제어할 수 있는 임의의 메커니즘을 포함할 수 있다. 그러한 소프트웨어는 캐쉬 초기화, 캐쉬 라인 무효화 또는 플러싱(flushing), 라인의 명시적 할당 및 다른 관리 기능과 같은 동작들을 제공할 수 있다. 관리자(116)는 관리자(114)와 유사하게 구성될 수 있다.
- [0023] 프로세서(110)는 마이크로프로세서 또는 CPU(central processing unit)와 같은 임의의 처리 메커니즘을 포함할 수 있다. 프로세서(110)는 하나 이상의 개별적인 프로세서를 포함할 수 있다. 프로세서(110)는 네트워크 프로세서, 범용 내장 프로세서 또는 다른 유사한 타입의 프로세서를 포함할 수 있다.
- [0024] 메모리(112)는 임의의 저장 메커니즘을 포함할 수 있다. 메모리(112)의 예에는, RAM(random access memory), DRAM(dynamic RAM), SRAM(static RAM), 플래쉬 메모리, 테이프, 디스크, 및 다른 타입의 유사한 저장 메커니즘이 포함된다. 메모리(112)는 하나의 저장 메커니즘, 예를 들면, 하나의 RAM 칩, 또는 저장 메커니즘들의 임의

의 조합, 예를 들면, SRAM과 DRAM을 둘다 포함하는 다수 RAM 칩을 포함할 수 있다.

- [0025] 도시된 시스템(100)은 설명을 용이하게 하기 위해 간략화된다. 시스템(100)은 하나 이상의 저장 메커니즘(캐쉬, 메모리, 데이터베이스, 버퍼 등), 브리지, 칩셋, 네트워크 인터페이스, 그래픽 메커니즘, 디스플레이 장치, 외부 에이전트, 통신 링크(버스, 무선 링크 등), 저장 제어기, 및 시스템(100)과 유사한 컴퓨터 시스템 또는 네트워크 시스템과 같은 시스템에 포함될 수 있는 다른 유사한 타입의 요소들과 같은 다수의 요소들을 포함할 수 있다.
- [0026] 도 2를 참조하면, 캐쉬 동작의 예시적인 처리(200)가 도시된다. 처리(200)는 도 1의 예시적인 시스템(100)에 포함된 요소들을 참조하여 기술되지만, 재구성되거나 재구성되지 않은, 동일한, 보다 많은, 또는 보다 적은 요소들을 포함하는 이러한 처리 또는 유사한 처리가, 시스템(100)에서, 또는 다른 유사한 시스템에서 수행될 수 있다.
- [0027] 시스템(100)에서의 에이전트는 요청을 발행한다(202). 요청 에이전트로서 지칭되는 에이전트는 외부 에이전트(102), 프로세서(110) 또는 다른 에이전트일 수 있다. 이러한 예시적인 설명에서, 외부 에이전트(102)가 요청 에이전트이다.
- [0028] 데이터에 대한 요청은 요청 에이전트로부터의 데이터를 캐쉬(104) 내에 위치시키기 위한, 캐쉬(104)에 대한 요청을 포함할 수 있다. 요청은 네트워크 수신 동작, I/O 입력, 프로세서간 메시지의 전달, 또는 다른 유사한 동작과 같은 동작의 결과일 수 있다.
- [0029] 전형적으로, 캐쉬(104)는 관리자(114)를 통해, 캐쉬(104)가 요청에서 지시된 메모리(112) 내의 위치를 나타내는 위치를 포함하는지 여부를 결정한다(204). 그러한 결정은 캐쉬(104)에 액세스하고, 전형적으로 요청 에이전트에 의해 제공되는 데이터의 메모리 어드레스에 대한 태그 어레이(108)를 체크함으로써 수행될 수 있다.
- [0030] 처리(200)가 다수의 프로세서 또는 조합 또는 프로세서 및 I/O 서브시스템의 도움을 받는 다수의 캐쉬를 포함하는 시스템에서 이용된다면, 임의의 프로토콜을 이용하여, 다수의 캐쉬를 체크하고, 각 메모리 어드레스의 코히어런트 버전을 유지할 수 있다. 캐쉬(104)는 캐쉬의 태그 어레이에서의 요청 데이터의 어드레스와 관련된 상태를 체크하여, 해당 어드레스에서의 데이터가 다른 캐쉬에 포함되어 있는지/있거나 해당 어드레스에서의 데이터가 다른 캐쉬에서 수정되었는지를 확인한다. 예를 들어, "배타적" 상태는 해당 어드레스에서의 데이터가 체크되는 캐쉬 내에만 포함됨을 나타낼 수 있다. 다른 예로서, "공유" 상태는 데이터가 적어도 하나의 다른 캐쉬에 포함될 수 있고, 다른 캐쉬는 요청 에이전트가 요청 데이터를 폐지하기 전에, 보다 현재의 데이터에 대해 체크될 필요가 있음을 나타낼 수 있다. 상이한 프로세서 및/또는 I/O 서브시스템이, 캐쉬 태그를 체크 및 갱신하기 위해, 동일하거나 상이한 기술을 이용할 수 있다. 데이터가 외부 에이전트의 요청시에 캐쉬 내로 전달될 때, 데이터는 하나 또는 다수의 캐쉬 내로 전달될 수 있고, 데이터가 명시적으로 전달되지 않는 캐쉬는 무효로 되거나 매칭 엔트리를 갱신하여, 시스템 코히어런스를 유지해야 한다. 데이터가 전달될 캐쉬 또는 캐쉬들은 요청에서 지시되거나, 또는 다른 수단에 의해 정적으로 선택될 수 있다.
- [0031] 태그 어레이(108)가 어드레스 및 위치가 유효하다는 지시를 포함한다면, 캐쉬 히트가 인식된다. 캐쉬(104)는 요청에서 지시된 위치를 나타내는 엔트리를 포함하고, 외부 에이전트(102)는 데이터를 캐쉬(104)로 푸쉬하며, 캐쉬(104) 내에 위치를 처음에 할당할 필요없이, 캐쉬 라인에서의 오래된 데이터를 덮어쓰기 한다. 외부 에이전트(102)는 공유 메모리를 통해 프로세서(110)로 통신되는 데이터의 일부 또는 전부를 캐쉬(104) 내로 푸쉬할 수 있다. 예를 들어, 요청 에이전트가 모든 데이터를 즉각적으로 또는 전혀 파싱할 수 없다면, 단지 일부의 데이터만이 캐쉬(104) 내로 푸쉬될 수 있다. 예를 들어, 네트워크 인터페이스는 수신 서술자 및 패킷 헤더 정보와 같은 리딩(leading) 패킷 내용만을 푸쉬할 수 있다. 외부 에이전트(102)가 데이터의 선택된 부분만을 푸쉬한다면, 전형적으로 푸쉬되지 않은 데이터의 다른 부분은 그 대신에 외부 에이전트(102)에 의해 메모리(112) 내로 기록된다. 더욱이, 외부 에이전트(102)에 의해 기록된 메모리(112) 내의 위치를 나타내는 캐쉬(104) 및 다른 캐쉬 내의 임의의 위치는 무효화되거나, 또는 새로운 데이터로 갱신되어, 시스템 코히어런스를 유지한다. 다른 캐쉬 내의 데이터의 카피는 무효화될 수 있으며, 캐쉬(104) 내의 캐쉬 라인은 "배타적"인 것으로서 표시되거나, 카피가 갱신되고, 캐쉬 라인이 "공유"된 것으로서 표시된다.
- [0032] 태그 어레이(108)가 유효 위치에 요청 어드레스를 포함하지 않는다면, 그것은 캐쉬 미스이며, 캐쉬(104)는 메모리(112) 내의 요청 위치를 나타내는 라인을 포함하지 않는다. 이러한 경우, 캐쉬(104)는 전형적으로 관리자(114)의 동작을 통해서, 푸쉬 데이터를 위치시킬 캐쉬(104) 내의 라인을 선택("할당")한다. 캐쉬 라인을 할당하는 것은, 위치를 선택하고, 캐쉬(104)가 메모리(112)로 되기록할 책임을 갖는 블록을 해당 위치가 포함하는지

를 결정하고, 만약 그렇다면 치환된(또는 "희생된") 데이터를 메모리(112)에 기록하고, 선택된 위치의 태그를 요청에서 지시된 어드레스 및 적절한 캐쉬 라인 상태로 갱신하고, 외부 에이전트(102)로부터의 데이터를 태그 어레이(108) 내의 선택된 태그 위치에 대응하는 데이터 어레이(106) 내의 위치로 기록하는 것을 포함한다.

[0033] 캐쉬(104)는 캐쉬(104) 내의(예를 들면, 데이터 메모리(106) 및 태그 메모리(108) 내의) 위치를 선택(206)함으로써 외부 에이전트(102)의 요청에 응답하여, 데이터의 카피를 포함할 수 있다. 이러한 선택은 할당이라고 지칭될 수 있으며, 선택된 위치는 할당된 위치라고 지칭될 수 있다. 할당된 위치가 유효 태그 및 메모리(112) 내의 상이한 위치를 나타내는 데이터를 포함한다면, 해당 내용은 "희생"이라고 지칭될 수 있으며, 캐쉬(104)로부터 그것을 제거하는 동작은 "희생시킨다" 라고 지칭될 수 있다. 희생 라인에 대한 상태는 라인이 희생될 때, 캐쉬(104)가 메모리(112) 내의 대응하는 위치를 희생 라인으로부터의 데이터로 갱신(208)할 책임이 있음을 나타낼 수 있다.

[0034] 캐쉬(104) 또는 외부 에이전트(102)는 외부 에이전트(102)로부터 캐쉬(104)로 푸쉬된 새로운 데이터로 메모리(112)를 갱신할 책임이 있을 수 있다. 새로운 데이터를 캐쉬(104) 내로 푸쉬할 때, 전형적으로, 시스템 내의 메모리 메커니즘들 사이, 예시적인 본 시스템(100)에서는 캐쉬(104)와 메모리(112) 사이에 코히어런스가 유지되어야 한다. 코히어런스는 예를 들면, 다른 메커니즘(들)에서의 상태를 "무효" 또는 다른 적절한 상태로 변경하고, 다른 메커니즘(들)을 수정된 데이터로 갱신하는 등의 동작에 의해, 수정을 반영하도록 다른 메모리 메커니즘에 위치되는 수정 데이터의 임의의 다른 카피를 갱신함으로써 유지된다. 캐쉬(104)는 데이터의 소유자로서 표시될 수 있으며, 새로운 데이터로 메모리(112)를 갱신(212)할 책임을 갖게 된다. 캐쉬(104)는 외부 에이전트(102)가 데이터를 캐쉬(104)로 푸쉬할 때 또는 이후의 시간에 메모리(112)를 갱신할 수 있다. 대안적으로, 데이터는 공유될 수 있고, 외부 에이전트(102)는 메커니즘, 즉 본 예에서는 메모리(112)를 갱신(214)하고, 캐쉬(104) 내로 푸쉬된 새로운 데이터로 메모리를 갱신할 수 있다. 그 후, 메모리(112)는 가장 현재 버전의 데이터의 카피를 포함할 수 있다.

[0035] 캐쉬(104)는 희생된 위치에 대한 태그 어레이(108)에서의 태그를, 요청에서 지시된 메모리(112) 내의 어드레스로 갱신(216)한다.

[0036] 캐쉬(104)는 희생된 위치에서의 내용을 외부 에이전트(102)로부터의 데이터로 대체(218)할 수 있다. 프로세서(110)가 캐쉬 계층 구조를 지원한다면, 외부 에이전트(102)는, 전형적으로 최외곽 층으로부터 시작되는 하나 이상의 레벨의 캐쉬 계층 구조 내로 데이터를 푸쉬할 수 있다.

[0037] 도 3을 참조하면, 캐쉬 동작의 다른 예시적인 처리(500)가 도시된다. 처리(500)는 캐쉬(104)에 대한 프로세서(110)의 액세스 및 캐쉬(104) 채움 요구의 예를 기술한다. 처리(500)는 도 1의 예시적인 시스템(100)에 포함된 요소들을 참조하여 기술되지만, 재구성되거나 또는 재구성되지 않은, 동일한, 보다 많거나 또는 보다 적은 요소를 포함하는 이러한 처리 또는 유사한 처리가 시스템(100) 내에 또는 다른 유사한 시스템 내에 형성될 수 있다.

[0038] 프로세서(110)가 캐쉬가능한 메모리 참조를 발행하는 경우, 해당 프로세서(110)의 메모리 액세스와 관련된 캐쉬(들)(104)은 그들의 관련된 태그 어레이(108)를 탐색하여, 요청된 위치가 그러한 캐쉬에서 현재 표현되고 있는지 여부를 결정한다(502). 캐쉬(들)(104)은, 예를 들어, 프로세서로부터의 기록을 허용하기 위해 라인이 정확한 코히어런트 상태에 있는 경우, 캐쉬(들)(104) 내의 참조된 엔트리가 요청 액세스에 대해 적절한 허가를 갖는지를 더 결정(504)한다. 메모리(112) 내의 위치가 캐쉬(104)에서 현재 표현되고, 정당한 허가를 갖는다면, "히트"가 검출되고, 캐쉬는 메모리(112) 내의 관련된 위치를 대신하여 프로세서로 데이터를 제공하거나 또는 프로세서로부터 데이터를 수신함으로써 요청을 서비스(506)한다. 태그 어레이(108) 내의 태그가, 요청 위치가 제공되지만 적절한 허가를 갖지 않음을 나타낸다면, 캐쉬 관리자(114)는, 예를 들면, 그것에 대한 기록을 인에이블링하기 위해 라인의 배타적인 소유권을 획득함으로써, 정당한 허가를 획득(508)한다. 캐쉬(104)가, 요청 위치가 캐쉬 내에 존재하지 않는 것으로 결정하는 경우, "미스"가 검출되고, 캐쉬 관리자(114)는 새로운 라인을 위치시킬 캐쉬(104) 내의 위치를 할당(510)하고, 적절한 허가를 갖는 메모리(112)로부터 데이터를 요청(512)하고, 데이터의 수신(514)시에, 데이터 및 관련된 태그를 캐쉬(104) 내의 할당된 위치에 위치시킬 것이다. 이들 사이의 코히어런스를 유지하는 복수의 캐쉬를 지원하는 시스템에서, 요청된 데이터는 메모리(112)로부터 발생되는 것이 아니라, 다른 캐쉬로부터 실제로 발생할 수 있다. 캐쉬(104) 내의 라인의 할당은, 해당 라인의 현재의 유효 내용을 희생시킬 수 있고, 전술한 바와 같은 희생의 되기록을 더 초래할 수 있다. 따라서, 처리(500)는 희생이 되기록을 요구하는지를 결정(512)하고, 만약 그렇다면, 메모리에 대한 희생된 라인의 되기록을 수행(514)한다.

- [0039] 도 4를 참조하면, 처리(300)는 스로틀링(throttling) 메커니즘이, 외부 에이전트(102)가 데이터를 캐쉬(104) 내로 푸쉬할 것인지/언제 푸쉬할 것인지에 대한 결정(302)을 어떻게 돕는지를 도시하고 있다. 스로틀링 메커니즘은, 외부 에이전트(102)가 캐쉬(104)를 압도하는 것을 방지하고, 또한, 시스템의 효율성을 저하시킬 수 있는 너무 많은 희생이 초래되지 않도록 방지할 수 있다. 예를 들어, 외부 에이전트(102)가 데이터를 캐쉬(104) 내로 푸쉬한다면, 그러한 푸쉬된 데이터는 프로세서(110)가 해당 위치를 액세스하기 전에 희생되고, 나중에 프로세서(110)는 요구시에 캐쉬(104) 내로 데이터가 역으로 되도록 잘못을 행할 것이며, 따라서, 프로세서(110)는 캐쉬 미스에 대해 대기 시간을 초래하고, 불필요한 캐쉬 및 메모리 트래픽을 초래할 수 있다.
- [0040] 외부 에이전트(102)가 데이터를 푸쉬하는 캐쉬(104)가 프로세서(110)에 대한 1차 데이터 캐쉬인 경우, 스로틀링 메커니즘은 경험(heuristics)을 이용하여, 보다 많은 데이터를 캐쉬(104) 내로 푸쉬하기 위해 그것이 외부 에이전트(102)에 대해 수용가능한지/언제 수용가능한지를 결정한다. 만약 그것이 수용가능한 시간이라면, 캐쉬(104)는 데이터를 포함하기 위해 캐쉬(104) 내의 위치를 선택(208)할 수 있다. 만약 그것이 현재 수용가능한 시간이 아니라면, 스로틀링 메커니즘은, (예를 들면, 요청이 수신된 시간에서의 능력 또는 자원 충돌에 근거하여) 경험을 이용하여 데이터를 유지(308) (또는, 데이터에 대한 그의 요청을 유지, 또는 외부 에이전트(102)가 이후의 시간에 재시도하도록 지시)할 수 있으며, 스로틀링 메커니즘은 그것이 수용가능한 시간인 것으로 결정한다.
- [0041] 캐쉬(104)가 전문화된 캐쉬인 경우, 스로틀링 메커니즘은 외부 에이전트(102)를 흐름 제어하기 위해 이용(306)되는 큐상에서의 임계값 검출과 같은 경험보다 더 결정론적인 메커니즘을 포함할 수 있다. 일반적으로, 큐는 요소들이 입력되었던 순서와 동일한 순서로 제거되는 데이터 구조를 포함한다.
- [0042] 도 5를 참조하면, 다른 예시적인 시스템(400)은 외부 에이전트(402)가 데이터를, 일반적으로 메모리(406)를 모방하는 주 메모리(406)("메모리(406)")와 동료 관계인 코히어런트 룩어사이드 버퍼(coherent lookaside buffer; CLB) 캐쉬 메모리(404)("CLB(404)") 내로 푸쉬하도록 허용할 수 있는 관리자(416)를 포함한다. 전형적으로, 버퍼는 일시적 저장 영역을 포함하고, 주 메모리, 예를 들면, 메모리(406)보다 낮은 대기 시간으로 액세스가능하다. CLB(404)는 프로세서(408)에 대해 메모리(406)보다 낮은 대기 시간 액세스를 제공하는 외부 에이전트(402)로부터 새롭게 도달되거나 또는 새롭게 생성된 데이터를 위한 스테이징(staging) 영역을 제공한다. 프로세서(408)가, 링 버퍼를 서비스할 때와 같은 알려진 액세스 패턴을 갖는 통신 메커니즘에서, CLB(404)를 이용하는 것은, 새로운 데이터를 액세스하는 것으로부터의 캐쉬 미스로 인한 스톱(stalls)을 감소시킴으로써, 프로세서(408)의 성능을 향상시킬 수 있다. CLB(404)는 다수의 에이전트 및/또는 프로세서 및 그들의 대응하는 캐쉬에 의해 공유될 수 있다.
- [0043] CLB(404)는 외부 에이전트(402)가 서술자 또는 버퍼 어드레스를 CLB(404)를 통해 프로세서(408)로 송신하기 위해 이용하는 시그널링 또는 통시 큐(410)와 결합된다. 큐(410)는 큐(410)가 가득 차는 경우, 그의 대응하는 CLB(404)가 가득 찬다는 점에서 흐름 제어를 제공한다. 큐(410)는 외부 에이전트(102)에게, 큐(410)가 "큐 풀(queue full)" 지시로 가득 차는 때에 통지한다. 마찬가지로, 큐(410)는 프로세서(408)에게, 큐가 "큐가 비어 있지 않음(queue not empty)" 지시를 갖는 적어도 하나의 비서비스된 엔트리를 가짐을 통지하고, 큐(410)에서 처리할 데이터가 있음을 신호한다.
- [0044] 외부 에이전트(402)는 큐(410) 내의 각 엔트리에 대한 데이터의 분량을 하나 이상의 캐쉬 라인에 푸쉬할 수 있다. 큐(410)는 X 엔트리를 포함하며, 여기서 X는 양의 정수와 동일하다. CLB(404)는 큐(410)를 링으로서 취급하면서, 포인터를 이용하여 할당을 위한 다음 CLB 엔트리를 지시한다.
- [0045] CLB(404)는 CLB 태그(412) 및 CLB 데이터(414)(각각 도 1의 태그 어레이(108) 및 데이터 메모리(106)와 유사함)를 포함하며, 태그 및 데이터를 각각 저장한다. CLB 태그(412) 및 CLB 데이터(414)는 각각 Y 블록의 데이터를 포함하며, 여기서, Y는 X*Y와 동일한 전체 엔트리 수에 대한 큐(410)에서의 각 데이터 엔트리에 대해, 양의 정수와 동일하다. 태그(412)는 태그에 의해 표현된 순차적인 캐쉬 블록의 수의 각 엔트리에 대한 지시를 포함할 수 있으며, 또는 해당 정보가 암시적일 수 있다. 프로세서(408)가, 외부 에이전트(402)가 CLB(404)로 푸쉬한 데이터의 라인으로 캐쉬를 채우기 위해 메모리 판독을 발행할 때, CLB(404)는 푸쉬된 데이터로 개입될 수 있다. CLB는 각각의 통지에 대해 프로세서(408)에게 Y 블록까지의 데이터를 전달할 수 있다. 각 블록은 그 어드레스가 CLB 태그(412) 내에 유효한 것으로서 저장 및 표시된 어드레스들 중 하나와 매칭되는 캐쉬 라인 채움 요청에 응답하여, CLB(404)로부터 프로세서(408)로 전달된다.
- [0046] CLB(404)는 1회 판독 방안을 가지므로, 일단 프로세서 캐쉬가 CLB 데이터(414)로부터 데이터 엔트리를 판독하면, CLB(404)는 그 엔트리를 무효화할 수 있다(잊어 버림). Y가 "1"보다 크다면, CLB(404)는 해당 위치

가 액세스될 때 각 데이터 블록을 개별적으로 무효화시키고, 모든 "Y" 블록이 액세스되는 경우에만 대응 태그를 무효화시킨다. 프로세서(408)는 통지와 관련된 모든 Y 블록을 액세스할 것이 요구된다.

[0047] 시스템(400)에 포함된 요소들은 도 1의 시스템에 포함된 유사하게 지칭되는 요소들과 유사하게 구현될 수 있다. 시스템(400)은 시스템(100)에 대해 위에서 기술된 바와 같은 다소의 요소들을 포함한다. 더욱이, 일반적으로, 시스템(400)은 외부 에이전트(402)가 캐쉬(104) 대신에 CLB(404) 내로 데이터를 푸쉬하고, 요청된 데이터가 CLB(404) 내에 존재할 때, 프로세서(408)가 CLB(404)로부터 캐쉬를 요구에 따라 채우는 것을 제외하고는, 도 2 및 3에서의 예와 유사하게 동작한다.

[0048] 설명된 기법들은 임의의 특정한 하드웨어 또는 소프트웨어 구성에 한정되지 않으며, 그러한 기법들은 매우 다양한 컴퓨팅 또는 처리 환경에서 응용될 수도 있다. 예를 들어, 네트워크 PDU를 처리하기 위한 시스템이 하나 이상의 물리 계층(PHY) 장치(예를 들면, 배선, 광학, 또는 무선 PHY) 및 하나 이상의 연결 계층 장치(예를 들면, 이더넷 매체 액세스 제어기(MAC) 또는 SONET 프레임)를 포함할 수 있다. 수신 로직(예를 들면, 수신 하드웨어, 프로세서, 또는 스레드)은 PDU에 포함된 데이터의 배치 또는 전송한 바와 같은 캐쉬 동작에서의 데이터의 서술자를 요청함으로써, PHY 및 연결 계층 장치를 통해 수신된 PDU상에서 동작할 수 있다. 후속하는 로직(예를 들면, 상이한 스레드 또는 프로세서)이 캐쉬를 통해 PDU 관련 데이터를 신속하게 액세스하고, 다른 동작들 중에서, 브리징, 라우팅, QoS(quality of service)의 결정, (예를 들면, 소스 및 목적지 어드레스 및 PDU의 포트들에 근거한) 흐름 결정, 또는 필터링과 같은 패킷 처리 동작을 수행할 수 있다. 그러한 시스템은 RISC(Reduced Instruction Set Computing) 프로세서의 집합을 특징으로 하는 네트워크 프로세서(NP)를 포함할 수 있다. NP 프로세서의 스레드는 전송한 바와 같은 수신 로직 및 패킷 처리 동작을 수행할 수 있다.

[0049] 그러한 기법들은 하드웨어, 소프트웨어, 또는 그 둘의 조합으로 구현될 수 있다. 그러한 기법은 모바일 컴퓨터, 정적 컴퓨터, 네트워킹 장비, PDA(personal digital assistants), 및 각각 프로세서, 프로세서에 의해 관독가능한 저장 매체(휘발성 및 비휘발성 메모리 및/또는 저장 소자를 포함함), 적어도 하나의 입력 장치 및 하나 이상의 출력 장치를 포함하는 유사한 장치들과 같은 프로그램가능 기계상에서 실행되는 프로그램으로 구현될 수 있다. 입력 장치를 이용하여 입력된 데이터에 프로그램 코드를 적용하여, 기술된 기능들을 수행하여, 출력 정보를 생성한다. 출력 정보는 하나 이상의 출력 장치에 적용된다.

[0050] 각각의 프로그램은 기계 시스템과 통신하도록, 하이 레벨 절차 또는 객체 지향적 프로그래밍 언어로 구현될 수 있다. 그러나, 원하는 경우, 프로그램은 어셈블리 또는 기계 언어로 구현될 수 있다. 임의의 경우에, 언어는 컴파일되거나 해석된 언어일 수 있다.

[0051] 각각의 그러한 프로그램은 이 문서에서 기술된 절차들을 수행하기 위해 컴퓨터에 의해 저장 매체 또는 장치가 관독될 때, 기계를 구성 및 동작하기 위한 일반적 용도 또는 특수한 용도의 프로그램가능 기계에 의해 관독가능한 저장 매체 또는 장치, 예를 들면, CD-ROM, 하드디스크, 자기 디스켓 또는 유사한 매체 또는 장치에 저장될 수 있다. 또한, 시스템은 프로그램으로 구성된 기계 관독가능 저장 매체로서 구현되는 것으로 고려될 수 있으며, 여기서 저장 매체는 기계가 특정한 사전정의된 방법으로 동작하도록 구성된다.

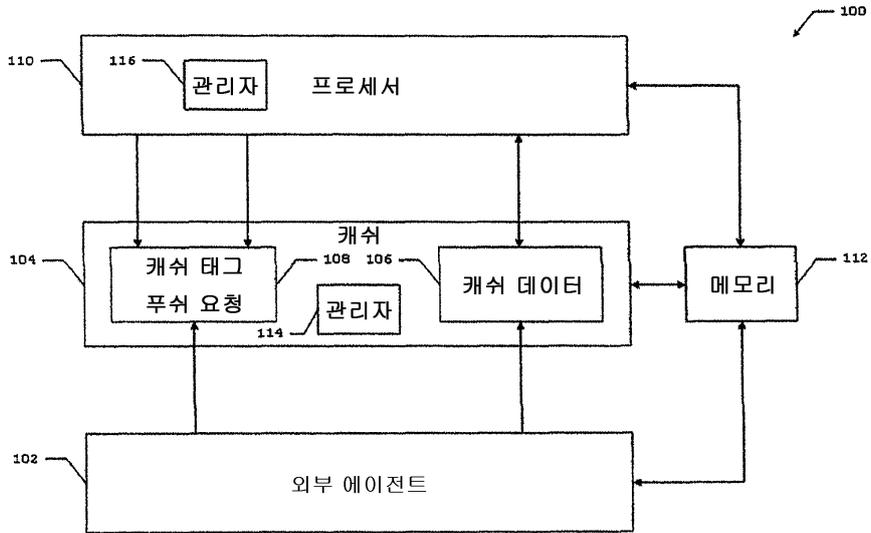
[0052] 다른 실시예들도 이하의 특허 청구 범위의 영역에 속한다.

도면의 간단한 설명

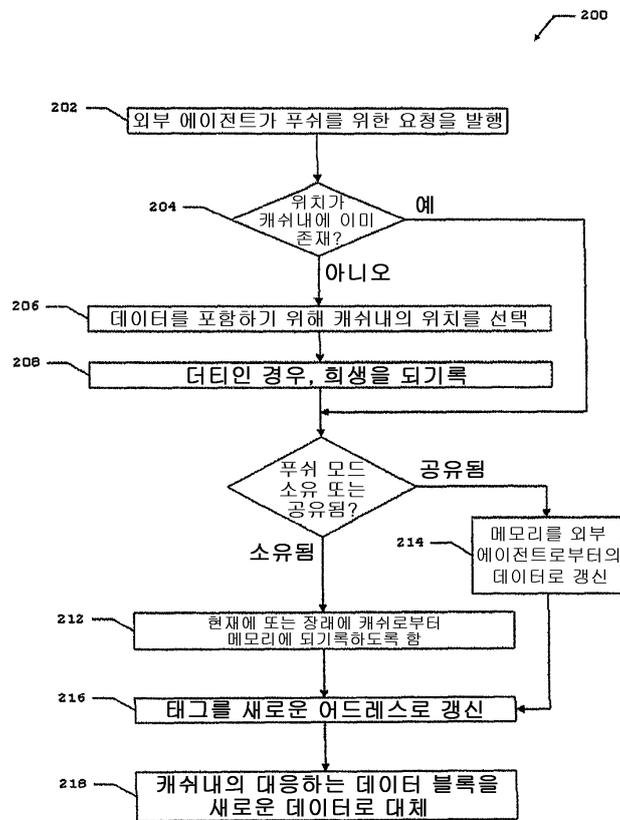
- [0003] 도 1은 캐쉬를 포함하는 시스템의 블록도이다.
- [0004] 도 2 및 3은 메모리 메커니즘을 채우는 처리를 도시하는 흐름도이다.
- [0005] 도 4는 메모리 메커니즘을 채우는 처리의 일부를 도시하는 흐름도이다.
- [0006] 도 5는 코히어런트 록어사이드 버퍼를 포함하는 시스템의 블록도이다.

도면

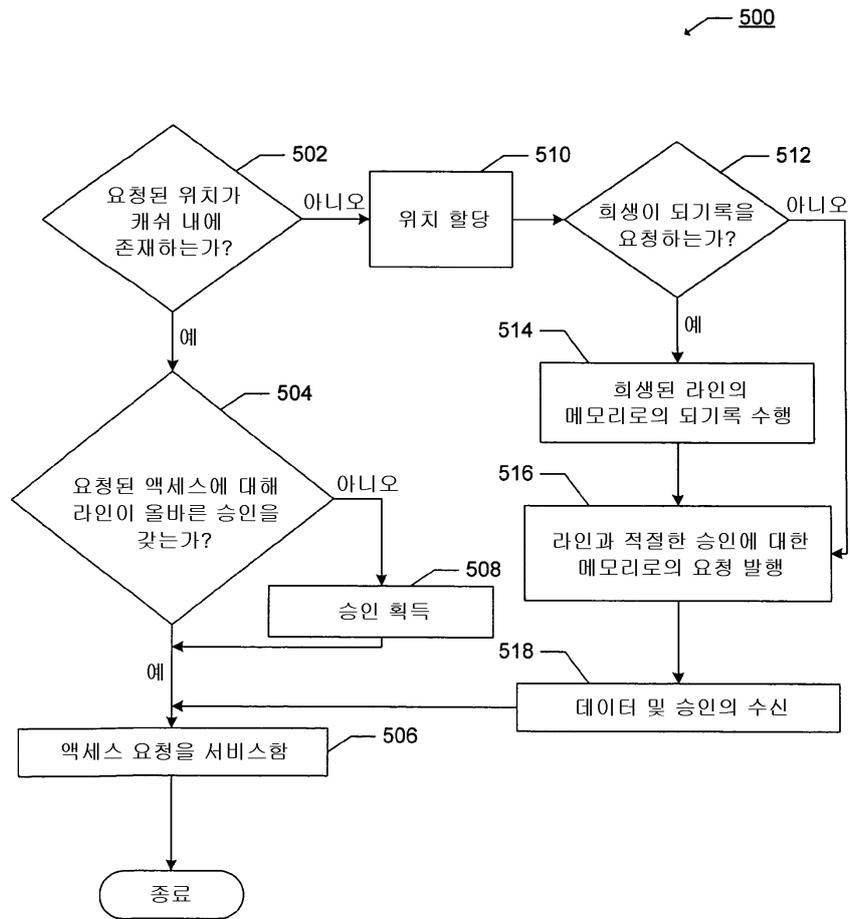
도면1



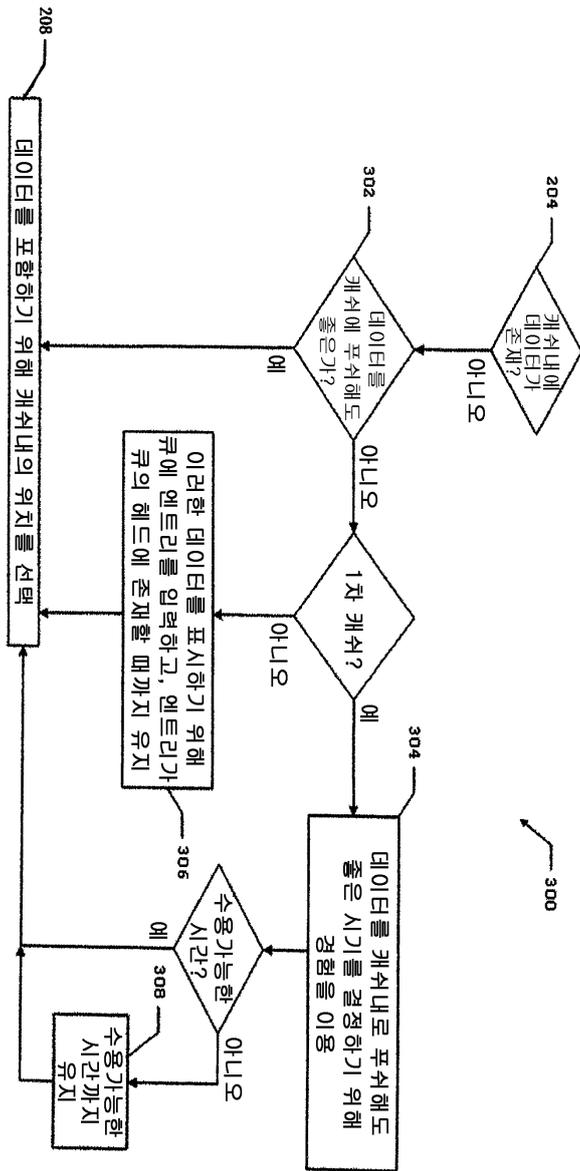
도면2



도면3



도면4



도면5

