



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0034440
(43) 공개일자 2018년04월04일

(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2018.01) G06F 9/52 (2018.01)
(52) CPC특허분류
G06F 9/4843 (2013.01)
G06F 9/52 (2013.01)
(21) 출원번호 10-2018-7002512
(22) 출원일자(국제) 2016년06월23일
심사청구일자 없음
(85) 번역문제출일자 2018년01월25일
(86) 국제출원번호 PCT/US2016/038925
(87) 국제공개번호 WO 2017/019212
국제공개일자 2017년02월02일
(30) 우선권주장
62/198,830 2015년07월30일 미국(US)
14/992,268 2016년01월11일 미국(US)

(71) 출원인
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
자오 한
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
몬테시노스 오르테고 파블로
미국 94539 캘리포니아주 프리몬트 미션 티에라 플레이스 118
(뒷면에 계속)
(74) 대리인
특허법인코리아나

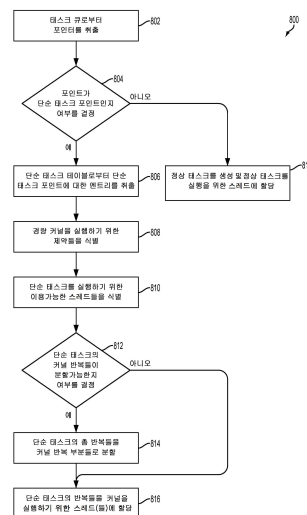
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 효율적인 병렬 컴퓨팅을 위한 단순화된 태스크-기반 런타임을 위한 방법

(57) 요약

양태들은 태스크 구조를 셋업하지 않고 스레드에 의해 직접 단순 태스크들로서 경량 커널들의 스케줄링 및 실행을 구현하기 위한 컴퓨팅 디바이스들, 시스템들, 및 방법들을 포함한다. 컴퓨팅 디바이스는 태스크 큐에서의 태스크 포인터가 경량 커널에 대한 단순 태스크 포인터인지 여부를 결정할 수도 있다. 컴퓨팅 디바이스는 스레드에 의한 실행을 위한 경량 커널을 위해 제 1 단순 태스크를 스케줄링할 수도 있다. 컴퓨팅 디바이스는 단순 태스크 테이블의 엔트리로부터, 경량 커널에 대한 커널 포인터를 추출할 수도 있다. 단순 태스크 테이블에서의 엔트리는 단순 태스크 포인터와 연관될 수도 있다. 컴퓨팅 디바이스는 경량 커널을 단순 태스크로서 직접 실행할 수도 있다.

대표도 - 도8



(72) 발명자

라만 아룬

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

로바트밀리 베흐남

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카스카발 게오르그 칼린

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

컴퓨팅 디바이스에서 경량 커널 (lightweight kernel) 들을 스케줄링 및 실행하는 방법으로서,

태스크 큐에서의 제 1 태스크 포인터가 경량 커널에 대한 단순 태스크 포인터 (simple task pointer) 인지 여부를 결정하는 단계;

상기 제 1 태스크 포인터가 단순 태스크 포인터라고 결정하는 것에 응답하여 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 단계;

단순 태스크 테이블의 엔트리로부터 상기 경량 커널에 대한 커널 포인터를 추출하는 단계로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 단계; 및

상기 경량 커널을 상기 제 1 단순 태스크로서 직접 실행하는 단계

를 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 단순 태스크의 실행을 완료하는 단계; 및

상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 단계

를 더 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 3

제 2 항에 있어서,

상기 경량 커널의 커널 반복들이 분할가능한지 여부를 결정하는 단계; 및

상기 경량 커널의 상기 커널 반복들이 분할가능하다고 결정하는 것에 응답하여 상기 경량 커널의 상기 커널 반복들을 반복 부분들로 분할하는 단계

를 더 포함하고,

상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 단계는 상기 제 1 단순 태스크에 적어도 하나의 반복 부분을 할당하는 단계를 포함하고;

상기 제 1 단순 태스크의 실행을 완료하는 단계는 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 반복들의 수와 동일한 수의 상기 제 1 단순 태스크의 실행들을 완료하는 단계를 포함하고; 그리고

상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 단계는 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 상기 반복들의 수의 완료를 반영하기 위해 상기 커널 반복 카운터의 상기 데이터를 업데이트하는 단계를 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 4

제 2 항에 있어서,

상기 커널 반복 카운터의 상기 데이터로부터 상기 제 1 단순 태스크의 모든 반복들이 완료되는지 여부를 결정하는 단계; 및

상기 제 1 단순 태스크의 반복들 모두가 완료된다고 결정하는 것에 응답하여 상기 단순 태스크 테이블의 상기 엔트리를 클리어하는 단계

를 더 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 5

제 1 항에 있어서,

상기 경량 커널을 실행하기 위한 제약을 식별하는 단계를 더 포함하고, 상기 제약은 메인 스레드, 임계 스레드, 및 비-임계 스레드를 포함하는, 상기 경량 커널을 실행하기 위한 지정된 스레드, 상기 경량 커널을 실행하기 위한 레이턴시 요건, 및 상기 경량 커널을 저장하는 메모리에 대한 상기 제 1 스레드를 실행하는 프로세서의 근접성 중 하나를 포함하고,

상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 단계는 상기 경량 커널을 실행하기 위한 상기 제약에 기초하여 상기 제 1 스레드를 선택하는 단계를 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 6

제 1 항에 있어서,

상기 태스크 큐에서의 제 2 태스크 포인터가 상기 경량 커널에 대한 상기 단순 태스크 포인터인지 여부를 결정하는 단계;

상기 제 2 태스크 포인터가 상기 단순 태스크 포인터라고 결정하는 것에 응답하여 제 2 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 2 단순 태스크를 스케줄링하는 단계;

상기 단순 태스크 테이블의 상기 엔트리로부터, 상기 경량 커널에 대한 상기 커널 포인터를 추출하는 단계로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 단계; 및

상기 경량 커널을 상기 제 2 단순 태스크로서 직접 실행하는 단계

를 더 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 7

제 6 항에 있어서,

상기 제 1 단순 태스크의 제 1 출력과 상기 제 2 단순 태스크의 제 2 출력을 결합하는 단계를 더 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 8

제 1 항에 있어서,

요청된 프로세스가 상기 경량 커널을 포함하는지 여부를 결정하는 단계;

상기 요청된 프로세스가 상기 경량 커널을 포함한다고 결정하는 것에 응답하여 상기 단순 태스크 테이블이 풀(full) 인지 여부를 결정하는 단계;

상기 단순 태스크 테이블이 풀이 아니라고 결정하는 것에 응답하여 상기 단순 태스크 테이블에서 상기 경량 커널에 대한 상기 엔트리를 생성하는 단계;

상기 엔트리와 연관된 상기 단순 태스크 포인터를 상기 태스크 큐에 추가하는 단계; 및

상기 단순 태스크 테이블이 풀이라고 결정하는 것에 응답하여 상기 경량 커널과 연관된 정상 태스크 포인터(normal task pointer)를 상기 태스크 큐에 추가하는 단계

를 더 포함하는, 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하는 방법.

청구항 9

컴퓨팅 디바이스로서,

제 1 프로세서를 포함하고,

상기 제 1 프로세서는,

태스크 큐에서의 제 1 태스크 포인터가 경량 커널에 대한 단순 태스크 포인터인지 여부를 결정하는 것;

상기 제 1 태스크 포인터가 단순 태스크 포인터라고 결정하는 것에 응답하여 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것;

단순 태스크 테이블의 엔트리로부터 상기 경량 커널에 대한 커널 포인터를 추출하는 것으로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 것; 및

상기 경량 커널을 상기 제 1 단순 태스크로서 직접 실행하는 것

을 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성된, 컴퓨팅 디바이스.

청구항 10

제 9 항에 있어서,

상기 제 1 프로세서는,

상기 제 1 단순 태스크의 실행을 완료하는 것; 및

상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 것

을 더 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 11

제 10 항에 있어서,

상기 제 1 프로세서는,

상기 경량 커널의 커널 반복들이 분할가능한지 여부를 결정하는 것; 및

상기 경량 커널의 상기 커널 반복들이 분할가능하다고 결정하는 것에 응답하여 상기 경량 커널의 상기 커널 반복들을 반복 부분들로 분할하는 것

을 더 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되고,

상기 제 1 프로세서는,

상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것이 상기 제 1 단순 태스크에 적어도 하나의 반복 부분을 할당하는 것을 포함하고;

상기 제 1 단순 태스크의 실행을 완료하는 것이 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 반복들의 수와 동일한 수의 상기 제 1 단순 태스크의 실행들을 완료하는 것을 포함하고; 그리고

상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 것이 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 상기 반복들의 수의 완료를 반영하기 위해 상기 커널 반복 카운터의 상기 데이터를 업데이트하는 것을 포함하도록

동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 12

제 10 항에 있어서,

상기 제 1 프로세서는,

상기 커널 반복 카운터의 상기 데이터로부터 상기 제 1 단순 태스크의 모든 반복들이 완료되는지 여부를 결정하는 것; 및

상기 제 1 단순 태스크의 반복들 모두가 완료된다고 결정하는 것에 응답하여 상기 단순 태스크 테이블의 상기 엔트리를 클리어하는 것

을 더 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 13

제 9 항에 있어서,

상기 제 1 프로세서에 통신가능하게 접속된 메모리; 및

상기 제 1 프로세서에 통신가능하게 접속된 제 2 프로세서

를 더 포함하고,

상기 제 1 프로세서는, 상기 경량 커널을 실행하기 위한 제약을 식별하는 것을 더 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되고, 상기 제약은 메인 스레드, 임계 스레드, 및 비-임계 스레드를 포함하는, 상기 경량 커널을 실행하기 위한 지정된 스레드, 상기 경량 커널을 실행하기 위한 레이턴시 요건, 및 상기 경량 커널을 저장하는 상기 메모리에 대한 상기 제 1 스레드를 실행하는 상기 제 2 프로세서의 근접성 중 하나를 포함하고, 그리고

상기 제 1 프로세서는, 상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것이 상기 경량 커널을 실행하기 위한 상기 제약에 기초하여 상기 제 1 스레드를 선택하는 것을 포함하도록 동작을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 14

제 9 항에 있어서,

상기 제 1 프로세서는,

상기 태스크 큐에서의 제 2 태스크 포인터가 상기 경량 커널에 대한 상기 단순 태스크 포인터인지 여부를 결정하는 것;

상기 제 2 태스크 포인터가 상기 단순 태스크 포인터라고 결정하는 것에 응답하여 제 2 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 2 단순 태스크를 스케줄링하는 것;

상기 단순 태스크 테이블의 상기 엔트리로부터, 상기 경량 커널에 대한 상기 커널 포인터를 추출하는 것으로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 것; 및

상기 경량 커널을 상기 제 2 단순 태스크로서 직접 실행하는 것

을 더 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 15

제 14 항에 있어서,

상기 제 1 프로세서는,

상기 제 1 단순 태스크의 제 1 출력과 상기 제 2 단순 태스크의 제 2 출력을 결합하는 것

을 더 포함하는 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 16

제 9 항에 있어서,

상기 제 1 프로세서는,

요청된 프로세스가 상기 경량 커널을 포함하는지 여부를 결정하는 것;

상기 요청된 프로세스가 상기 경량 커널을 포함한다고 결정하는 것에 응답하여 상기 단순 태스크 테이블이 풀인지 여부를 결정하는 것;

상기 단순 태스크 테이블이 풀이 아니라고 결정하는 것에 응답하여 상기 단순 태스크 테이블에서 상기 경량 커널에 대한 상기 엔트리를 생성하는 것;

상기 엔트리와 연관된 상기 단순 태스크 포인터를 상기 태스크 큐에 추가하는 것; 및

상기 단순 태스크 테이블이 풀이라고 결정하는 것에 응답하여 상기 경량 커널과 연관된 정상 태스크 포인터를 상기 태스크 큐에 추가하는 것

을 더 포함하는 동작을 수행하기 위해 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 17

컴퓨팅 디바이스로서,

태스크 큐에서의 제 1 태스크 포인터가 경량 커널에 대한 단순 태스크 포인터인지 여부를 결정하기 위한 수단;

상기 제 1 태스크 포인터가 단순 태스크 포인터라고 결정하는 것에 응답하여 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하기 위한 수단;

단순 태스크 테이블의 엔트리로부터 상기 경량 커널에 대한 커널 포인터를 추출하기 위한 수단으로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하기 위한 수단; 및

상기 경량 커널을 상기 제 1 단순 태스크로서 직접 실행하기 위한 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 18

제 17 항에 있어서,

상기 제 1 단순 태스크의 실행을 완료하기 위한 수단; 및

상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하기 위한 수단

을 더 포함하는, 컴퓨팅 디바이스.

청구항 19

제 18 항에 있어서,

상기 경량 커널의 커널 반복들이 분할가능한지 여부를 결정하기 위한 수단; 및

상기 경량 커널의 상기 커널 반복들이 분할가능하다고 결정하는 것에 응답하여 상기 경량 커널의 상기 커널 반복들을 반복 부분들로 분할하기 위한 수단

을 더 포함하고,

상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하기 위한 수단은 상기 제 1 단순 태스크에 적어도 하나의 반복 부분을 할당하기 위한 수단을 포함하고;

상기 제 1 단순 태스크의 실행을 완료하기 위한 수단은 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 반복들의 수와 동일한 수의 상기 제 1 단순 태스크의 실행들을 완료하기 위한 수단을 포함하고; 그리고

상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하기 위한 수단은 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 상기 반복들의 수의 완료를 반영하기 위해 상기 커널 반복 카운터의 데이터를 업데이트하기 위한 수단을 포함하는, 컴퓨팅 디바이스.

청구항 20

제 18 항에 있어서,

상기 커널 반복 카운터의 데이터로부터 상기 제 1 단순 태스크의 모든 반복들이 완료되는지 여부를 결정하기 위한 수단; 및

상기 제 1 단순 태스크의 반복들 모두가 완료된다고 결정하는 것에 응답하여 상기 단순 태스크 테이블의 상기 엔트리를 클리어하기 위한 수단

을 더 포함하는, 컴퓨팅 디바이스.

청구항 21

제 17 항에 있어서,

상기 경량 커널을 실행하기 위한 제약을 식별하기 위한 수단을 더 포함하고, 상기 제약은 메인 스레드, 임계 스레드, 및 비-임계 스레드를 포함하는, 상기 경량 커널을 실행하기 위한 지정된 스레드, 상기 경량 커널을 실행하기 위한 레이턴시 요건, 및 상기 경량 커널을 저장하는 메모리에 대한 상기 제 1 스레드를 실행하는 프로세서의 근접성 중 하나를 포함하고,

상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하기 위한 수단은 상기 경량 커널을 실행하기 위한 상기 제약에 기초하여 상기 제 1 스레드를 선택하기 위한 수단을 포함하는, 컴퓨팅 디바이스.

청구항 22

제 17 항에 있어서,

상기 태스크 큐에서의 제 2 태스크 포인터가 상기 경량 커널에 대한 상기 단순 태스크 포인터인지 여부를 결정하기 위한 수단;

상기 제 2 태스크 포인터가 상기 단순 태스크 포인터라고 결정하는 것에 응답하여 제 2 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 2 단순 태스크를 스케줄링하기 위한 수단;

상기 단순 태스크 테이블의 상기 엔트리로부터, 상기 경량 커널에 대한 상기 커널 포인터를 추출하기 위한 수단으로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하기 위한 수단;

상기 경량 커널을 상기 제 2 단순 태스크로서 직접 실행하기 위한 수단; 및

상기 제 1 단순 태스크의 제 1 출력과 상기 제 2 단순 태스크의 제 2 출력을 결합하기 위한 수단을 더 포함하는, 컴퓨팅 디바이스.

청구항 23

제 17 항에 있어서,

요청된 프로세스가 상기 경량 커널을 포함하는지 여부를 결정하기 위한 수단;

상기 요청된 프로세스가 상기 경량 커널을 포함한다고 결정하는 것에 응답하여 상기 단순 태스크 테이블이 풀인지 여부를 결정하기 위한 수단;

상기 단순 태스크 테이블이 풀이 아니라고 결정하는 것에 응답하여 상기 단순 태스크 테이블에서 상기 경량 커널에 대한 상기 엔트리를 생성하기 위한 수단;

상기 엔트리와 연관된 상기 단순 태스크 포인터를 상기 태스크 큐에 추가하기 위한 수단; 및

상기 단순 태스크 테이블이 풀이라고 결정하는 것에 응답하여 상기 경량 커널과 연관된 정상 태스크 포인터를 상기 태스크 큐에 추가하기 위한 수단

을 더 포함하는, 컴퓨팅 디바이스.

청구항 24

프로세서 실행가능 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체로서,

상기 프로세서 실행가능 명령들은, 컴퓨팅 디바이스의 제 1 프로세서로 하여금,

태스크 큐에서의 제 1 태스크 포인터가 경량 커널에 대한 단순 태스크 포인터인지 여부를 결정하는 것;

상기 제 1 태스크 포인터가 단순 태스크 포인터라고 결정하는 것에 응답하여 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것;

단순 태스크 테이블의 엔트리로부터 상기 경량 커널에 대한 커널 포인터를 추출하는 것으로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 것; 및

상기 경량 커널을 상기 제 1 단순 태스크로서 직접 실행하는 것을 포함하는 동작들을 수행하게 하도록 구성된, 비밀시적 프로세서 판독가능 저장 매체.

청구항 25

제 24 항에 있어서,
저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,
상기 제 1 단순 태스크의 실행을 완료하는 것; 및
상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서 판독가능 저장 매체.

청구항 26

제 25 항에 있어서,
저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,
상기 경량 커널의 커널 반복들이 분할가능한지 여부를 결정하는 것; 및
상기 경량 커널의 상기 커널 반복들이 분할가능하다고 결정하는 것에 응답하여 상기 경량 커널의 상기 커널 반복들을 반복 부분들로 분할하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되고,
저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,
상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것이 상기 제 1 단순 태스크에 적어도 하나의 반복 부분을 할당하는 것을 포함하고;
상기 제 1 단순 태스크의 실행을 완료하는 것이 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 반복들의 수와 동일한 수의 상기 제 1 단순 태스크의 실행들을 완료하는 것을 포함하고; 그리고
상기 단순 태스크 테이블의 상기 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 것이 상기 제 1 단순 태스크에 할당된 상기 적어도 하나의 반복 부분의 상기 반복들의 수의 완료를 반영하기 위해 상기 커널 반복 카운터의 상기 데이터를 업데이트하는 것을 포함하도록
동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서 판독가능 저장 매체.

청구항 27

제 25 항에 있어서,
저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,
상기 커널 반복 카운터의 상기 데이터로부터 상기 제 1 단순 태스크의 모든 반복들이 완료되는지 여부를 결정하는 것; 및
상기 제 1 단순 태스크의 반복들 모두가 완료된다고 결정하는 것에 응답하여 상기 단순 태스크 테이블의 상기 엔트리를 클리어하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서 판독가능 저장 매체.

청구항 28

제 24 항에 있어서,
저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금, 상기 경량 커널을 실행하기 위한 제약을 식별하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되고,
저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,

상기 제약이 메인 스레드, 임계 스레드, 및 비-임계 스레드를 포함하는, 상기 경량 커널을 실행하기 위한 지정된 스레드, 상기 경량 커널을 실행하기 위한 레이턴시 요건, 및 상기 경량 커널을 저장하는 메모리에 대한 상기 제 1 스레드를 실행하는 제 2 프로세서의 근접성 중 하나를 포함하고; 그리고

상기 제 1 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것이 상기 경량 커널을 실행하기 위한 상기 제약에 기초하여 상기 제 1 스레드를 선택하는 것을 포함하도록

동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 29

제 24 항에 있어서,

저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,

상기 태스크 큐에서의 제 2 태스크 포인터가 상기 경량 커널에 대한 상기 단순 태스크 포인터인지 여부를 결정하는 것;

상기 제 2 태스크 포인터가 상기 단순 태스크 포인터라고 결정하는 것에 응답하여 제 2 스레드에 의한 실행을 위한 상기 경량 커널을 위해 제 2 단순 태스크를 스케줄링하는 것;

상기 단순 태스크 테이블의 상기 엔트리로부터, 상기 경량 커널에 대한 상기 커널 포인터를 추출하는 것으로서, 상기 엔트리는 상기 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 것;

상기 경량 커널을 상기 제 2 단순 태스크로서 직접 실행하는 것; 및

상기 제 1 단순 태스크의 제 1 출력과 상기 제 2 단순 태스크의 제 2 출력을 결합하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 30

제 24 항에 있어서,

저장된 상기 프로세서 실행가능 명령들은, 상기 제 1 프로세서로 하여금,

요청된 프로세스가 상기 경량 커널을 포함하는지 여부를 결정하는 것;

상기 요청된 프로세스가 상기 경량 커널을 포함한다고 결정하는 것에 응답하여 상기 단순 태스크 테이블이 풀인지 여부를 결정하는 것;

상기 단순 태스크 포인터가 풀이 아니라고 결정하는 것에 응답하여 상기 단순 태스크 테이블에서 상기 경량 커널에 대한 상기 엔트리를 생성하는 것;

상기 엔트리와 연관된 상기 단순 태스크 포인터를 상기 태스크 큐에 부가하는 것; 및

상기 단순 태스크 테이블이 풀이라고 결정하는 것에 응답하여 상기 경량 커널과 연관된 정상 태스크 포인터를 상기 태스크 큐에 부가하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

발명의 설명

기술 분야

배경 기술

[0001]

관련 출원들

[0002]

본 출원은 35 U.S.C. § 119(e) 하에서, 2015년 7월 30일자로 출원된 발명의 명칭이 "Method For Simplified Task-based Runtime For Efficient Parallel Computing" 인 미국 가출원 제62/198,830호의 이익을 주장하고, 그 전체 내용은 이로써 참조로 통합된다.

- [0003] 배경
- [0004] 태스크 병렬성은 컴퓨테이션 코드들이 다수의 프로세서들에 걸쳐서 병렬화되는 병렬화의 형태이다. 병렬 컴퓨팅 환경에서 기본 스케줄링가능한 유닛으로서 서빙하는, 컴퓨테이션 태스크는, 소정의 입력들 및 출력들을 가지고 또는 이들 없이 컴퓨테이션 프로시저 (이하 "커널들" 로 지칭됨) 를 구현한다. 태스크-기반 병렬 프로그래밍 런타임은 프로그래머들이 태스크들의 형태로 알고리즘들을 표현하는 것을 허용하고, 스케줄러를 이용하여 다수의 프로세서들에 걸쳐서 태스크들을 분배하고 동기화 및 부하 밸런싱과 같은 메인테넌스 기능성들을 달성한다. 태스크-기반 런타임 시스템들이 발달하여 더 많은 피쳐들을 제공함에 따라, 태스크 관념들이 점차 복잡해져, 태스크 생성, 관리, 및 파괴에 상당한 오버헤드를 부과한다. 예를 들어, 태스크-기반 런타임 시스템들은 태스크를 셋업 시에 태스크가 이중의 디바이스 실행 경로에 속하는지 여부를 결정, 태스크의 라이프사이클을 트래킹하기 위한 태스크 레퍼런싱 및 언-레퍼런싱, 및 스케줄러로부터 독점적 소유권을 요청하는데 있어서 오버헤드를 초래한다.

- [0005] 태스크들을 생성, 디스패치 및 관리하는 오버헤드가 실제 컴퓨테이션에 맞먹기 때문에, 종래의 태스크-기반 런타임 시스템은 경량 커널 (lightweight kernel) 들에 상당한 오버헤드를 부가한다. 성능과 에너지 효율 양자 모두는 태스크 관리와 연관된 불가피한 오버헤드로 인해 손상된다. 자격을 갖춘 태스크-기반 런타임 시스템은 이들 제약들로 인해 병렬화가 더 낮은 빈도들로 발생하기 때문에 복잡한 의존성들 및 동기화 요건들을 가진 중량 커널 (heavyweight kernel) 들에 적합하다.

발명의 내용

해결하려는 과제

과제의 해결 수단

- [0006] 다양한 실시형태들의 방법들 및 장치들은 컴퓨팅 디바이스에서 경량 커널들을 스케줄링 및 실행하기 위한 회로들 및 방법들을 제공한다. 다양한 실시형태들은, 태스크 큐에서의 제 1 태스크 포인터가 경량 커널에 대한 단순 태스크 포인터 (simple task pointer) 인지 여부를 결정하는 것, 제 1 태스크 포인터가 단순 태스크 포인터라고 결정하는 것에 응답하여 제 1 스레드에 의한 실행을 위한 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것, 단순 태스크 테이블의 엔트리로부터 경량 커널에 대한 커널 포인터를 추출하는 것으로서, 엔트리는 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 것, 및 경량 커널을 제 1 단순 태스크로서 직접 실행하는 것을 포함할 수도 있다.
- [0007] 일부 실시형태들은 제 1 단순 태스크의 실행을 완료하는 것, 및 단순 태스크 테이블의 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 것을 더 포함할 수도 있다.
- [0008] 일부 실시형태들은 경량 커널의 커널 반복들이 분할가능한지 여부를 결정하는 것, 및 경량 커널의 커널 반복들이 분할가능하다고 결정하는 것에 응답하여 경량 커널의 커널 반복들을 반복 부분들로 분할하는 것을 더 포함할 수도 있다. 이러한 실시형태들에서, 제 1 스레드에 의한 실행을 위한 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것은 제 1 단순 태스크에 경량 커널의 적어도 하나의 반복 부분을 할당하는 것을 포함할 수도 있고, 제 1 단순 태스크의 실행을 완료하는 것은 제 1 단순 태스크에 할당된 적어도 하나의 반복 부분의 반복들의 수와 동일한 수의 제 1 단순 태스크의 실행들을 완료하는 것을 포함할 수도 있고, 그리고 단순 태스크 테이블의 엔트리의 커널 반복 카운터의 데이터를 업데이트하는 것은 제 1 단순 태스크에 할당된 적어도 하나의 반복 부분의 반복들의 수의 완료를 반영하기 위해 커널 반복 카운터의 데이터를 업데이트하는 것을 포함할 수도 있다.
- [0009] 일부 실시형태들은 커널 반복 카운터의 데이터로부터 제 1 단순 태스크의 모든 반복들이 완료되는지 여부를 결정하는 것, 및 제 1 단순 태스크의 반복들 모두가 완료된다고 결정하는 것에 응답하여 단순 태스크 테이블의 엔트리를 클리어하는 것을 더 포함할 수도 있다.
- [0010] 일부 실시형태들은 경량 커널을 실행하기 위한 제약을 식별하는 것을 더 포함할 수도 있다. 이러한 실시형태들에서, 그 제약은 메인 스레드, 임계 스레드, 및 비-임계 스레드를 포함하는, 경량 커널을 실행하기 위한 지정된 스레드, 경량 커널을 실행하기 위한 레이턴시 요건, 및 경량 커널을 저장하는 메모리에 대한 제 1 스레드를 실행하는 프로세서의 근접성 (proximity) 중 하나를 포함할 수도 있다. 이러한 실시형태들에서, 제 1 스레드에 의한 실행을 위한 경량 커널을 위해 제 1 단순 태스크를 스케줄링하는 것은 경량 커널을 실행하기 위한

제약에 기초하여 제 1 스레드를 선택하는 것을 포함할 수도 있다.

- [0011] 일부 실시형태들은 태스크 큐에서의 제 2 태스크 포인트가 경량 커널에 대한 단순 태스크 포인터인지 여부를 결정하는 것, 제 2 태스크 포인트가 단순 태스크 포인터라고 결정하는 것에 응답하여 제 2 스레드에 의한 실행을 위한 경량 커널을 위해 제 2 단순 태스크를 스케줄링하는 것, 단순 태스크 테이블의 엔트리로부터 경량 커널에 대한 커널 포인터를 추출하는 것으로서, 엔트리는 단순 태스크 포인터와 연관되는, 상기 커널 포인터를 추출하는 것, 및 경량 커널을 제 2 단순 태스크로서 직접 실행하는 것을 더 포함할 수도 있다.
- [0012] 일부 실시형태들은 제 1 단순 태스크의 제 1 출력과 제 2 단순 태스크의 제 2 출력을 결합하는 것을 더 포함할 수도 있다.
- [0013] 일부 실시형태들은 요청된 프로세스가 경량 커널을 포함하는지 여부를 결정하는 것, 요청된 프로세스가 경량 커널을 포함한다고 결정하는 것에 응답하여 단순 태스크 테이블이 풀 (full) 인지 여부를 결정하는 것, 단순 태스크 테이블이 풀이 아니라고 결정하는 것에 응답하여 단순 태스크 테이블에서 경량 커널에 대한 엔트리를 생성하는 것, 엔트리와 연관된 단순 태스크 포인터를 태스크 큐에 추가하는 것, 및 단순 태스크 테이블이 풀이라고 결정하는 것에 응답하여 경량 커널과 연관된 정상 태스크 포인터 (normal task pointer) 를 태스크 큐에 추가하는 것을 더 포함할 수도 있다.
- [0014] 다양한 실시형태들은 상기 설명된 실시형태 방법들 중 하나 이상의 동작들을 수행하기 위해 프로세서 실행가능 명령들로 구성된 프로세서를 포함하는 컴퓨팅 디바이스를 포함할 수도 있다.
- [0015] 다양한 실시형태들은 상기 설명된 실시형태 방법들 중 하나 이상의 기능들을 수행하기 위한 수단을 갖는 컴퓨팅 디바이스를 포함할 수도 있다.
- [0016] 다양한 실시형태들은 컴퓨팅 디바이스의 프로세서로 하여금, 상기 설명된 실시형태 방법들 중 하나 이상의 동작들을 수행하게 하도록 구성된 프로세서 실행가능 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체를 포함할 수도 있다.

도면의 간단한 설명

- [0017] 본 명세서에 통합되고 본 명세서의 일부를 구성하는 첨부한 도면들은, 본 발명의 예의 양태들을 예시하고, 위에서 주어진 일반적인 설명 및 아래에 주어지는 상세한 설명과 함께, 본 발명의 피쳐들을 설명하도록 서빙한다.
- 도 1 은 다양한 실시형태들을 구현하는데 적합한 컴퓨팅 디바이스를 예시하는 컴포넌트 블록 다이어그램이다.
- 도 2 는 다양한 실시형태들을 구현하는데 적합한 일 예의 멀티-코어 프로세서를 예시하는 컴포넌트 블록 다이어그램이다.
- 도 3 은 다양한 실시형태들을 구현하는데 적합한 일 예의 시스템 온 칩 (SoC) 을 예시하는 컴포넌트 블록 다이어그램이다.
- 도 4 는 다양한 실시형태들에 따른 일 예의 태스크 큐의 예시이다.
- 도 5 는 다양한 실시형태들에 따른 일 예의 단순 태스크 테이블의 예시이다.
- 도 6 은 다양한 실시형태들에 따른 런타임 단순 태스크 스케줄링 및 실행을 예시하는 프로세스 플로우 다이어그램이다.
- 도 7 은 단순 태스크를 큐잉하기 위한 일 실시형태 방법을 예시하는 프로세스 플로우 다이어그램이다.
- 도 8 은 단순 태스크를 스케줄링하기 위한 일 실시형태 방법을 예시하는 프로세스 플로우 다이어그램이다.
- 도 9 는 단순 태스크를 실행하기 위한 일 실시형태 방법을 예시하는 프로세스 플로우 다이어그램이다.
- 도 10 은 다양한 실시형태들에의 이용에 적합한 일 예의 모바일 컴퓨팅 디바이스를 예시하는 컴포넌트 블록 다이어그램이다.
- 도 11 은 다양한 실시형태들에의 이용에 적합한 일 예의 모바일 컴퓨팅 디바이스를 예시하는 컴포넌트 블록 다이어그램이다.
- 도 12 는 다양한 실시형태들에의 이용에 적합한 일 예의 서버를 예시하는 컴포넌트 블록 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

- [0018] 다양한 양태들은 첨부한 도면들을 참조하여 상세히 설명될 것이다. 가능하면 언제나, 동일한 참조 번호들은 동일하거나 또는 유사한 부분들을 지칭하기 위해 도면들 전반에 걸쳐 사용될 것이다. 특정한 예들 및 구현들에 이루어진 참조들은 예시적인 목적들을 위한 것이며, 본 발명 또는 청구항들의 범위를 제한하도록 의도되지 않는다.
- [0019] 용어들 "컴퓨팅 디바이스" 및 "모바일 컴퓨팅 디바이스"는 셀룰러 전화기들, 스마트폰들, 개인 또는 모바일 멀티-미디어 플레이어들, 개인 휴대 정보 단말기들 (PDA들), 랩톱 컴퓨터들, 태블릿 컴퓨터들, 스마트북들, 울트라북들, 팜-톱 컴퓨터들, 무선 전자 메일 수신기들, 멀티미디어 인터넷 가능 셀룰러 전화기들, 무선 게이밍 제어기들, 및 메모리, 및 멀티-코어 프로그래밍가능 프로세서를 포함하는 유사한 개인 전자 디바이스들 중 임의의 하나 또는 전부를 지칭하기 위해 본 명세서에서 상호교환가능하게 사용된다. 다양한 양태들은 제한된 메모리 및 배터리 리소스들을 갖는, 스마트폰들과 같은 모바일 컴퓨팅 디바이스들에 특히 유용하지만, 그 양태들은 일반적으로는 프로세서들의 전력 소비를 감소시키는 것이 모바일 컴퓨팅 디바이스의 배터리-동작 시간을 연장시킬 수 있는 제한된 전력 버짓 및 복수의 메모리 디바이스들을 구현하는 임의의 전자 디바이스에서 유용하다.
- [0020] 용어 "시스템-온-칩" (SoC)은 통상적으로, 그러나 비배타적으로, 하드웨어 코어, 메모리, 및 통신 인터페이스를 포함하는 상호접속된 전자 회로들의 세트를 지칭하기 위해 본 명세서에서 사용된다. 하드웨어 코어는 범용 프로세서, 중앙 프로세싱 유닛 (CPU), 디지털 신호 프로세서 (DSP), 그래픽스 프로세싱 유닛 (GPU), APU (accelerated processing unit), 보조 프로세서, 단일-코어 프로세서, 및 멀티-코어 프로세서와 같은 다양한 상이한 타입들의 프로세서들을 포함할 수도 있다. 하드웨어 코어는 다른 하드웨어 및 하드웨어 조합들, 이를테면 필드 프로그래밍가능 게이트 어레이 (FPGA), 주문형 집적 회로 (ASIC), 다른 프로그래밍가능 로직 디바이스, 이산 게이트 로직, 트랜지스터 로직, 성능 모니터링 하드웨어, 워치도그 (watchdog) 하드웨어, 및 타임 레퍼런스들을 추가로 구현할 수도 있다. 집적 회로들은 집적 회로의 컴포넌트들이 실리콘과 같은 반도체 재료의 단일 피스 상에 상주하도록 구성될 수도 있다.
- [0021] 현대의 컴퓨팅 디바이스들 상에서 실행하도록 구성된 많은 애플리케이션들은 의존성 및 동기화 요건이 충족되지 용이한 고도로 병렬화가능한, 경량 컴퓨테이션 프로시저들 (이하 "커널들"로 지칭됨)을 수반한다. 예를 들어, 일부 애플리케이션들은 각각의 반복이 매우 적은 컴퓨테이션 작업을 수행하는 백-투-백 루프 (back-to-back loop) 들의 실행을 포함한다. 종래의 태스크-기반 런타임 시스템에서 태스크들을 생성, 디스패치, 및 관리하는 오버헤드는 오버헤드가 실제 컴퓨테이션에 맞먹고 병렬화가 높은 레이트로 발생할 수도 있기 때문에, 이러한 경량 커널들을 병렬화하기 위해 상당한 오버헤드를 부가한다.
- [0022] 다양한 실시형태들은 런타임으로 정상 태스크 관리 스테이지들을 바이패스하고 정상 태스크 관리와 연관된 오버헤드를 감소시키는 단순 태스크 관리 방법들을 포함한다. 단순화된 태스크 관리는 런타임 관리된 단순 태스크 테이블에 의해 달성될 수도 있다. 단순 태스크 테이블의 각각의 엔트리는 경량 커널들에 대한 포인터들을 저장할 수도 있다. 단순 태스크는 단순 태스크 테이블에서의 엔트리에 대한 단순 태스크 포인터로 표현될 수도 있다. 이 메커니즘은 기존 스케줄링 로직에 개입하지 않고 런타임 스케줄러의 일 (job)을 단순화할 수도 있다. 스케줄러는 임의의 정상 태스크에 대한 바대로 태스크 큐로부터 태스크 포인터를 취출할 수도 있다. 태스크 포인터는 정상 태스크에 캐스트된 정상 태스크 포인터일 수도 있고, 스케줄러는 정상 프로시저들을 이용하여 정상 태스크를 스케줄링할 수도 있다. 태스크 포인터는 경량 커널에 대한 커널 포인터의 단순 태스크 테이블에서의 엔트리에 캐스트된 단순 태스크 포인터일 수도 있다. 단순 태스크 포인터에 대해, 스케줄러는 경량 커널을 프로세서 또는 프로세서 코어에 의해 단순 태스크로서의 직접 실행을 위한 스레드에 할당할 수도 있다. 단순 태스크는 단순 태스크가 정상 태스크인 것처럼 스케줄러가 스레드에 의한 실행을 위해 태스크를 스케줄링하는 것을 허용할 수도 있지만, 단순 태스크를 실행하기 위해 스레드는 태스크 구조를 인스턴트화하는 것, 태스크의 실행을 관리하는 것, 및 태스크 구조를 디스패치하는 것 없이 경량 커널을 직접 실행할 수도 있다.
- [0023] 도 1은 다양한 실시형태들에의 이용에 적합한 원격 컴퓨팅 디바이스 (50)와 통신하는 컴퓨팅 디바이스 (10)를 포함하는 시스템을 예시한다. 컴퓨팅 디바이스 (10)는 프로세서 (14), 메모리 (16), 통신 인터페이스 (18), 및 저장 메모리 인터페이스 (20)를 가진 SoC (12)를 포함할 수도 있다. 컴퓨팅 디바이스는 유선 또는 무선 모뎀과 같은 통신 컴포넌트 (22), 저장 메모리 (24), 무선 네트워크 (30)에 무선 접속 (32)을 확립하기 위한 안테나 (26), 및/또는 인터넷 (40)으로의 유선 접속 (44)에 접속하기 위한 네트워크 인터페이스 (28)를 더 포함할 수도 있다. 프로세서 (14)는 다수의 프로세서 코어들은 물론 다양한 하드웨어 코어들 중 임

의의 것을 포함할 수도 있다. SoC (12) 는 하나 이상의 프로세서들 (14) 을 포함할 수도 있다. 컴퓨팅 디바이스 (10) 는 1 초과의 SoC들 (12) 을 포함하여, 프로세서들 (14) 및 프로세서 코어들의 수를 증가시킬 수도 있다. 컴퓨팅 디바이스 (10) 는 또한, SoC (12) 와 연관되지 않은 하나 이상의 프로세서들 (14) 을 포함할 수도 있다. 개개의 프로세서들 (14) 은 도 2 를 참조하여 아래에 설명되는 바와 같이 멀티-코어 프로세서들일 수도 있다. 프로세서들 (14) 은 컴퓨팅 디바이스 (10) 의 다른 프로세서들 (14) 과 동일하거나 또는 상이할 수도 있는 특정 목적들을 위해 각각 구성될 수도 있다. 동일하거나 또는 상이한 구성들의 프로세서들 (14) 및 프로세서 코어들 중 하나 이상은 함께 그룹화될 수도 있다. 프로세서들 (14) 또는 프로세서 코어들의 그룹은 멀티-프로세서 클러스터로 지칭될 수도 있다.

[0024] SoC (12) 의 메모리 (16) 는 프로세서 (14) 에 의한 액세스를 위한 데이터 및 프로세서 실행가능 코드를 저장하기 위해 구성된 휘발성 또는 비휘발성 메모리일 수도 있다. 컴퓨팅 디바이스 (10) 및/또는 SoC (12) 는 다양한 목적들을 위해 구성된 하나 이상의 메모리들 (16) 을 포함할 수도 있다. 다양한 실시형태들에서, 하나 이상의 메모리들 (16) 은 휘발성 메모리들, 이를 테면 랜덤 액세스 메모리 (RAM) 또는 메인 메모리, 또는 캐시 메모리를 포함할 수도 있다. 이들 메모리들 (16) 은 비휘발성 메모리로부터 요청되어, 다양한 팩터들에 기초하여 미래의 액세스를 예상하여 비휘발성 메모리로부터 메모리들 (16) 로 로딩되는 제한된 양의 데이터 및/또는 프로세서 실행가능 코드 명령들, 및/또는 프로세서 (14) 에 의해 생성되고 비휘발성 메모리에 저장되지 않고 미래의 고속 액세스를 위해 일시적으로 저장된 중간 프로세싱 데이터 및/또는 프로세서 실행가능 코드 명령들을 일시적으로 보유하도록 구성될 수도 있다.

[0025] 메모리 (16) 는 프로세서들 (14) 중 하나 이상에 의한 액세스를 위해, 다른 메모리 (16) 또는 저장 메모리 (24) 와 같은 다른 메모리 디바이스로부터 메모리 (16) 로 로딩되는 프로세서 실행가능 코드를 적어도 일시적으로 저장하도록 구성될 수도 있다. 메모리 (16) 로 로딩된 프로세서 실행가능 코드는 프로세서 (14) 에 의한 함수의 실행에 응답하여 로딩될 수도 있다. 함수의 실행에 응답하여 메모리 (16) 에 프로세서 실행가능 코드를 로딩하는 것은, 요청된 프로세서 실행가능 코드가 메모리 (16) 에 위치되지 않기 때문에, 성공적이지 않거나, 또는 미스 (miss) 인 메모리 (16) 에 대한 메모리 액세스 요청으로부터 발생할 수도 있다. 미스에 응답하여, 다른 메모리 디바이스에 대한 메모리 액세스 요청이 요청된 프로세서 실행가능 코드를 다른 메모리 디바이스로부터 메모리 디바이스 (16) 로 로딩하기 위해 이루어질 수도 있다. 함수의 실행에 응답하여 메모리 (16) 에 프로세서 실행가능 코드를 로딩하는 것은 다른 메모리 디바이스에 대한 메모리 액세스 요청으로부터 발생할 수도 있고, 프로세서 실행가능 코드는 추후의 액세스를 위해 메모리 (16) 로 로딩될 수도 있다.

[0026] 통신 인터페이스 (18), 통신 컴포넌트 (22), 안테나 (26), 및/또는 네트워크 인터페이스 (28) 는 컴퓨팅 디바이스 (10) 가 무선 접속 (32) 을 경유한 무선 네트워크 (30), 및/또는 유선 네트워크 (44) 를 통해 원격 컴퓨팅 디바이스 (50) 와 통신하는 것을 가능하게 하기 위해 협심하여 작동할 수도 있다. 무선 네트워크 (30) 는 컴퓨팅 디바이스 (10) 에, 그 컴퓨팅 디바이스 (10) 가 원격 컴퓨팅 디바이스 (50) 와 데이터를 교환할 수도 있는 인터넷 (40) 에의 접속을 제공하기 위해, 예를 들어, 무선 통신을 위해 이용되는 무선 주파수 스펙트럼을 포함하는, 다양한 무선 통신 기술들을 이용하여 구현될 수도 있다.

[0027] 저장 메모리 인터페이스 (20) 및 저장 메모리 (24) 는 컴퓨팅 디바이스 (10) 가 비휘발성 저장 매체 상에 데이터 및 프로세서 실행가능 코드를 저장하는 것을 허용하기 위해 협심하여 작동할 수도 있다. 저장 메모리 (24) 는 프로세서들 (14) 중 하나 이상에 의한 액세스를 위해 저장 메모리 (24) 가 데이터 및/또는 프로세서 실행가능 코드를 저장할 수도 있는 메모리 (16) 의 다양한 실시형태들과 매우 유사하게 구성될 수도 있다. 비휘발성인 저장 메모리 (24) 는 컴퓨팅 디바이스 (10) 의 전력이 셧 오프 (shut off) 된 후라도 정보를 유지할 수도 있다. 전력이 다시 턴 온되고 컴퓨팅 디바이스 (10) 가 리부팅할 때, 저장 메모리 (24) 상에 저장된 정보는 컴퓨팅 디바이스 (10) 에 대해 이용가능할 수도 있다. 저장 메모리 인터페이스 (20) 는 저장 메모리 (24) 에의 액세스를 제어하고 프로세서 (14) 가 저장 메모리 (24) 로부터 데이터를 판독하고 저장 메모리 (24) 에 데이터를 기입하는 것을 허용할 수도 있다.

[0028] 컴퓨팅 디바이스 (10) 의 컴포넌트들 중 일부 또는 전부는 필요한 기능들을 여전히 서빙하면서 상이하게 배열 및/또는 결합될 수도 있다. 더욱이, 컴퓨팅 디바이스 (10) 는 컴포넌트들의 각각의 하나에 제한되지 않을 수도 있고, 각각의 컴포넌트의 다수의 인스턴스들은 컴퓨팅 디바이스 (10) 의 다양한 구성들에 포함될 수도 있다.

[0029] 도 2 는 다양한 실시형태들을 구현하는데 적합한 멀티-코어 프로세서 (14) 를 예시한다. 도 1 을 참조하면, 멀티-코어 프로세서 (14) 는 복수의 동종 또는 이종의 프로세서 코어들 (200, 201, 202, 203) 을 가질 수도 있

다. 프로세서 코어들 (200, 201, 202, 203) 은 단일 프로세서 (14) 의 프로세서 코어들 (200, 201, 202, 203) 이 동일한 목적을 위해 구성되고 동일하거나 또는 유사한 성능 특성들을 가질 수도 있다는 점에서 동종일 수도 있다. 예를 들어, 프로세서 (14) 는 범용 프로세서일 수도 있고, 프로세서 코어들 (200, 201, 202, 203) 은 동종의 범용 프로세서 코어들일 수도 있다. 대안적으로, 프로세서 (14) 는 그래픽스 프로세싱 유닛 또는 디지털 신호 프로세서일 수도 있고, 프로세서 코어들 (200, 201, 202, 203) 은 각각 동종의 그래픽스 프로세서 코어들 또는 디지털 신호 프로세서 코어들일 수도 있다. 참조의 용이함을 위해, 용어들 "프로세서" 및 "프로세서 코어" 는 본 명세서에서 상호교환가능하게 사용될 수도 있다.

[0030] 프로세서 코어들 (200, 201, 202, 203) 은 단일 프로세서 (14) 의 프로세서 코어들 (200, 201, 202, 203) 이 상이한 목적들을 위해 구성되고 및/또는 상이한 성능 특성들을 가질 수도 있다는 점에서 이종일 수도 있다. 이러한 이종의 프로세서 코어들의 일 예는 더 느린, 저전력 프로세서 코어들이 더 파워풀한 그리고 전력 소모적인 (power-hungry) 프로세서 코어들과 커플링될 수도 있는 "big.LITTLE" 아키텍처들로 알려진 것을 포함할 수도 있다. 이러한 이종의 프로세서 코어들의 이종성 (heterogeneity) 은 상이한 명령 세트 아키텍처, 파이프라인들, 동작 주파수들 등을 포함할 수도 있다.

[0031] 도 2 에 예시된 예에서, 멀티-코어 프로세서 (14) 는 4 개의 프로세서 코어들 (200, 201, 202, 203) (즉, 프로세서 코어 0, 프로세서 코어 1, 프로세서 코어 2, 및 프로세서 코어 3) 을 포함한다. 설명의 용이함을 위해, 본 명세서의 예들은 도 2 에 예시된 4 개의 프로세서 코어들 (200, 201, 202, 203) 을 지칭할 수도 있다. 그러나, 도 2 에 예시되고 본 명세서에서 설명된 4 개의 프로세서 코어들 (200, 201, 202, 203) 은 단지 일 예로서 제공될 뿐이고 다양한 양태들을 4-코어 프로세서 시스템으로 제한하려는 의도는 없다. 컴퓨팅 디바이스 (10), SoC (12), 또는 멀티-코어 프로세서 (14) 는 개별적으로 또는 조합하여 본 명세서에 예시 및 설명된 4 개의 프로세서 코어들 (200, 201, 202, 203) 보다 더 적거나 또는 더 많은 프로세서 코어들을 포함할 수도 있다.

[0032] 게다가, 멀티-코어 프로세서 (14) 는 프로세서 캐시 메모리 제어기 (204) 및 프로세서 캐시 메모리 (206) 를 포함할 수도 있다. 프로세서 캐시 메모리 (206) 는 메모리 (16) (도 1 참조) 와 유사하게 구성될 수도 있다. 프로세서 캐시 메모리 제어기 (204) 및 프로세서 캐시 메모리 (206) 는 프로세서 코어들 (200, 201, 202, 203) 이 휘발성 저장 매체 상에 제한된 양의 데이터 및/또는 프로세서 실행가능 코드를 액세스 및 일시적으로 보유하는 것을 허용하기 위해 협심하여 작동할 수도 있다. 프로세서 캐시 메모리 제어기 (204) 는 프로세서 캐시 메모리 (206) 에 대한 액세스를 제어하고 프로세서 코어들 (200, 201, 202, 203) 이 프로세서 캐시 메모리 (206) 로부터 판독하고 이에 기입하는 것을 허용할 수도 있다. 프로세서 코어들 (200, 201, 202, 203) 은 프로세서 캐시 메모리 (206) 를 공유할 수도 있고 및/또는 개개의 프로세서 코어들 (200, 201, 202, 203) 은 전용 프로세서 캐시 메모리 (206) 와 연관될 수도 있다. 프로세서 캐시 메모리 제어기 (204) 및 프로세서 캐시 메모리 (206) 의 단일 쌍이 예시의 용이함을 위해 이용되고, 멀티-코어 프로세서 (14) 상에 포함될 수도 있는 프로세서 캐시 메모리 제어기들 (204) 및 프로세서 캐시 메모리들 (206) 의 수를 제한하려는 의도는 없다.

[0033] 도 3 은 다양한 양태들을 구현하는데 적합한 일 예의 SoC (12) 를 예시한다. 도 1, 도 2, 및 도 3 을 참조하면, SoC (12) 는 프로세서 (14), SoC 캐시 메모리 제어기 (300), SoC 캐시 메모리 (302), 메인 메모리 제어기 (304), 메인 메모리 (306), 및 상기 설명된 SoC (12) 의 컴포넌트들과 같은 다른 컴포넌트들을 포함할 수도 있다. SoC (12) 는 또한 저장 메모리 제어기 (308) 및 저장 메모리 (24) 를 포함하거나 또는 이들에 통신하게 접속될 수도 있다. SoC (12) 의 컴포넌트들 및 SoC (12) 에 접속된 컴포넌트는 통신 버스 (310) 를 통해 서로에 접속될 수도 있다. SoC 캐시 메모리 (302), 메인 메모리 (306), 및 저장 메모리 (24) 의 각각은 데이터 및/또는 프로세서 실행가능 코드와 같은 메모리 콘텐츠들을 저장하도록 구성될 수도 있다. 메모리 콘텐츠들은 SoC 캐시 메모리 (302), 메인 메모리 (306), 및 저장 메모리 (24) 의 물리적 어드레스들에 의해 식별된 특정 로케이션들에 저장될 수도 있다.

[0034] 일 양태에서, 메모리들 (24, 302, 306) 에 대한 메모리 액세스 요청들은 메모리 액세스 요청의 요청된 메모리 콘텐츠들을 추출하기 위하여 개별의 메모리 (24, 302, 306) 의 물리적 어드레스로 트랜스레이트될 수도 있는 가상 어드레스를 이용하여 이루어질 수도 있다. 데이터 및/또는 프로세서 실행가능 코드 중 임의의 것의 저장 로케이션들은 시간에 따라 변할 수도 있다. 데이터 및/또는 프로세서 실행가능 코드와 연관된 물리적 어드레스들은 프로세서 (14) 에 의한 액세스를 위해 데이터 및/또는 프로세서 실행가능 코드의 로케이션들을 맵핑하는 데이터 구조에서 업데이트될 수도 있다.

[0035] SoC 캐시 메모리 (302) 는 메인 메모리 (306) 또는 저장 메모리 (24) 에 액세스하여 달성가능한 더 고속의 액세스

스를 위해 데이터 및/또는 프로세서 실행가능 코드를 일시적으로 저장하도록 구성될 수도 있다. SoC 캐시 메모리 (302) 는 CPU (14a), GPU (14b), 또는 APU (14c) 와 같은 단일 프로세서 (14) 에 의한 이용을 위해 전용되거나 또는 다수의 프로세서들 (14), 이를 테면 CPU (14a), GPU (14b), 및 APU (14c) 의 임의의 조합 및/또는 SoC (12) 의 서브시스템들 (미도시) 간에 공유될 수도 있다. SoC 캐시 메모리 (302) 는 메모리 (16) (도 1 참조) 및 프로세서 캐시 메모리 (206) (도 12 참조) 와 유사하게 구성될 수도 있다. SoC 캐시 메모리 제어기 (300) 는 다양한 프로세서들 (14a 내지 14c) 및 SoC (12) 의 서브시스템들 (미도시) 에 의해 SoC 캐시 메모리 (302) 에 대한 액세스를 관리할 수도 있다. SoC 캐시 메모리 제어기 (300) 는 또한, 프로세서 (14a 내지 14c) 에 의해 SoC 캐시 메모리 (302) 로부터 요청되지만, SoC 캐시 메모리 (302) 에서 발견되지 않아 캐시 미스를 초래할 수도 있는 메모리 콘텐츠들을 추출하기 위해 SoC 캐시 메모리 제어기 (300) 로부터 메인 메모리 (306) 및 저장 메모리 (24) 로의 액세스를 위한 메모리 액세스 요청들을 관리할 수도 있다.

[0036] 메인 메모리 (306) 는 저장 메모리 (24) 에 액세스할 때보다 더 고속의 액세스를 위해 데이터 및/또는 프로세서 실행가능 코드를 일시적으로 저장하도록 구성될 수도 있다. 메인 메모리 (306) 는 하나 이상의 SoC들 (12) 의 프로세서들 (14a 내지 14c) 및/또는 SoC (12) 의 서브시스템들 (미도시) 에 의한 액세스를 위해 이용가능할 수도 있다. 메인 메모리 제어기 (304) 는 다양한 프로세서들 (14a 내지 14c) 및 SoC (12) 및 컴퓨팅 디바이스의 서브시스템들 (미도시) 에 의해 메인 메모리 (306) 에 대한 액세스를 관리할 수도 있다. 메인 메모리 제어기 (304) 는 또한, 프로세서들 (14a 내지 14c) 또는 SoC 캐시 메모리 제어기 (300) 에 의해 메인 메모리 (306) 로부터 요청되지만, 메인 메모리 (306) 에서 발견되지 않아 메인 메모리 미스를 초래할 수도 있는 메모리 콘텐츠들을 추출하기 위해 메인 메모리 제어기 (304) 에 의한 저장 메모리 (24) 에의 액세스를 위한 메모리 액세스 요청들을 관리할 수도 있다.

[0037] 저장 메모리 (24) 는 컴퓨팅 디바이스가 전력공급되지 않을 때 유지를 위해 데이터 및/또는 프로세서 실행가능 코드의 지속적 저장을 제공하도록 구성될 수도 있다. 저장 메모리 (24) 는 SoC 캐시 메모리 (302) 및 메인 메모리 (306) 보다 더 많은 데이터 및/또는 프로세서 실행가능 코드를 저장하고, 그리고 프로세서들 (14a 내지 14c) 또는 SoC (12) 의 서브시스템들 (미도시) 에 의해 가까운 미래에 이용 또는 이용이 예측되지 않는 것들을 포함하는 데이터 및/또는 프로세서 실행가능 코드를 저장하기 위한 용량을 가질 수도 있다. 저장 메모리 (24) 는 하나 이상의 SoC들 (12) 의 프로세서들 (14a 내지 14c), 및/또는 SoC (12) 의 서브시스템들 (미도시) 에 의한 액세스를 위해 이용가능할 수도 있다. 저장 메모리 제어기 (308) 는 다양한 프로세서들 (14a 내지 14c) 및 SoC (12) 및 컴퓨팅 디바이스의 서브시스템들 (미도시) 에 의해 저장 메모리 (24) 에 대한 액세스를 관리할 수도 있다. 저장 메모리 제어기 (308) 는 또한, 프로세서들 (14a 내지 14c) 에 의해 SoC 캐시 메모리 (302) 또는 메인 메모리 (306) 로부터 요청되지만, SoC 캐시 메모리 (302) 또는 메인 메모리 (306) 에서 발견되지 않아 캐시 메모리 미스 또는 메인 메모리 미스를 초래할 수도 있는 메모리 콘텐츠들을 추출하기 위해 SoC 캐시 메모리 제어기 (300) 및 메인 메모리 제어기 (304) 로부터 저장 메모리 (24) 로의 액세스를 위한 메모리 액세스 요청들을 관리할 수도 있다.

[0038] SoC (12) 의 컴포넌트들의 일부 또는 전부는 필요한 기능들을 여전히 서빙하면서 상이하게 배열 및/또는 결합될 수도 있다. 더욱이, SoC (12) 는 컴포넌트들의 각각의 하나에 제한되지 않을 수도 있고, 각각의 컴포넌트의 다수의 인스턴스들은 SoC (12) 의 다양한 구성들에 포함될 수도 있다. SoC (12) 의 다양한 실시형태 구성들은 SoC (12) 와 분리되지만 통신 버스들 (310) 을 통해 SoC (12) 에 접속된 CPU (14a), GPU (14b), APU (14c), 메인 메모리 제어기 (304), 및 메인 메모리 (306) 와 같은 컴포넌트들을 포함할 수도 있다. 다양한 실시형태 구성들은 프로세서들 (14) 의 동종 또는 이종의 조합들을 포함하는, 임의의 단일 프로세서 (14) 또는 프로세서들 (14) 의 조합을 포함할 수도 있다. 유사하게, 컴퓨팅 디바이스 (10) (도 1) 는 SoC들 (12) 의 동종 또는 이종의 조합들을 포함하는, 임의의 SoC (12) 또는 SoC들 (12) 의 조합을 포함할 수도 있다.

[0039] 도 4 는 다양한 실시형태들에 따른 태스크 큐 (400) 의 일 예를 예시한다. 도 1 내지 도 4 를 참조하면, 태스크 큐 (400) 는 프로세서에 의한 실행을 위해 스케줄링될 다양한 프로세스들의 태스크들을 스케줄러에 표시하기 위한 태스크 포인터들의 콜렉션 및 조직화를 포함할 수도 있다. 태스크 큐 (400) 는 프로세서 (14) 에 의해 실행된 스케줄러에 의해 액세스가능한, 다양한 형태들의 메모리 (16, 24, 206, 306), 또는 전용 하드웨어, 이를 테면 레지스터에서 구현될 수도 있다. 다양한 실시형태들에서, 하나 이상의 스케줄러들 및 태스크 큐들 (400) 은 프로세서들 및 SoC들의 그룹들을 위해 컴퓨팅 디바이스 (10) 상에서 구현될 수도 있다. 태스크 큐 (400) 는 정상 태스크 포인터들 및 단순 태스크 포인터들을 포함하여, 다양한 태스크 포인터들을 저장하도록 구성된 슬롯들 (402, 404, 406, 408, 410) 을 포함할 수도 있다. 단순 태스크 포인터들은 도 5 를 참조하여 추가로 설명되는 바와 같이 그들이 경량 커널에 대한 데이터 엔트리에 캐스트되도록 구성될 수도 있다. 정

상 태스크 포인터들은 경량 커널과는 다른 임의의 커널을 위해 그들이 메모리 (16, 24, 206, 306) 에서의 로케이션에 캐스트되도록 구성될 수도 있다. 다양한 실시형태들에서, 단순 태스크 포인터들 및 정상 태스크 포인터들은 다른 타입의 태스크 포인터의 값과 오버랩하지 않는 값들에 캐스트될 수도 있다. 따라서, 스케줄러는 도 5 를 참조하여 추가로 설명되는 바와 같이, 태스크 포인터가 그들의 캐스트 값들에 의해 단순 태스크 포인터인지 또는 정상 태스크 포인터인지를 식별할 수도 있다.

[0040] 도 4 에 예시된 예에서, 태스크 큐 슬롯 (404) 은 정상 태스크 포인터 (정상 태스크 포인터 1) 를 포함하고, 태스크 큐 슬롯들 (402, 406, 408, 및 410) 은 단순 태스크 포인터들 (단순 태스크 포인터 1 을 포함하는 슬롯 (402), 단순 태스크 포인터 3 을 포함하는 슬롯들 (406 및 408), 및 단순 태스크 포인터 2 를 포함하는 슬롯 (410)) 을 포함한다. 태스크 큐 (400) 는 예시된 예에서 5 개의 슬롯들 (402 내지 410) 보다 더 크거나 또는 더 적은 임의의 수의 슬롯들을 포함할 수도 있다.

[0041] 다양한 실시형태들에서, 태스크 큐 (400) 는 선입 선출 (first-in first-out; FIFO) 방식으로 관리될 수도 있다. 이로써, 스케줄러는 가장 오래된 태스크 포인터를 갖는, 상부 슬롯, 이 예에서 슬롯 (402) 을 판독할 수도 있고, 상부 슬롯은 제거될 수도 있거나, 또는 오래된 태스크 포인터는 삭제 또는 덮어쓰여질 수도 있어, 나머지 태스크 포인터들은 태스크 큐 (400) 에서 그들 각각의 다음 슬롯들로 시프트 업된다. 태스크 포인터를, 그 연관된 태스크가 스케줄링될 때까지 제거 또는 교체하지 않는 정책과 같은 다른 태스크 큐 관리 정책들이 구현될 수도 있고, 태스크가 스케줄링될 수 없다면, 태스크 큐 (400) 는 실행될 수 없는 태스크 포인터가 태스크 큐 (400) 에서 다른 슬롯 (402 내지 410) 으로 이동되거나 또는 스케줄러가 태스크 큐 (400) 에서 다른 슬롯 (402 내지 410) 으로 일시적으로 스킵하도록 서포팅된다. 태스크 큐 관리 정책은 또한 우선순위 기반일 수도 있는데, 여기서 우선순위들은 태스크들 충돌들 및 의존성들에 기초할 수도 있는, 태스크의 임계성을 포함한, 다양한 팩터들에 의해 결정될 수도 있다. 우선순위들은 다음 슬롯 판독되는 슬롯에 영향을 주는 태스크 큐에서 각각의 태스크에 할당될 수도 있거나, 또는 우선순위들은 태스크들이 태스크 큐에서 순서화되는 순서에 영향을 줄 수도 있다.

[0042] 도 5 는 다양한 실시형태들에 따른 단순 태스크 테이블 (500) 의 일 예를 예시한다. 단순 태스크 테이블 (500) 은 다양한 형태들의 메모리 (16, 24, 206, 306) 에 저장된 데이터의 컬렉션 및 조직화를 포함할 수도 있다. 도 5 에 예시되고 본 명세서에서 설명된 단순 태스크 테이블 (500) 의 테이블 포맷은 데이터를 저장 및 조직화하기 위한 포맷의 일 예이고, 다른 데이터 구조들을 이용하여 데이터를 저장 및 조직화하기 위한 다른 실시형태들로서 제한되지 않는다.

[0043] 도 5 에 예시된 예에서, 단순 태스크 테이블 (500) 은 예를 들어, 각각의 엔트리에 대한 컬럼들 (502, 504, 506), 또는 경량 커널에 대한 태스크, 또는 단순 태스크의 로우, 508, 510, 512, 516 에서 커널 식별자 (ID), 커널 포인터, 및 커널 반복 카운터 간의 관계들을 조직화 및 유지할 수도 있다. 커널 식별자 컬럼 (502) 은 단순 태스크 테이블 (500) 에서의 단순 태스크에 대한 엔트리 (508 내지 516) 를 스케줄러에게 표시하도록 구성된 커널 식별자를 포함할 수도 있다. 스케줄러는 엔트리 (508 내지 516) 의 커널 식별자를 이용하여, 경량 커널이 실행을 위해 스케줄링될지 여부, 어느 경량 커널이 실행을 위해 스케줄링되어야 하는지, 및 경량 커널이 연관된 엔트리 (508 내지 516) 에 스케줄러를 안내하는 것에 의해 실행을 완료했는지 여부를 식별하는 것을 도울 수도 있다. 단순 태스크 포인터들은 최대가 커널 테이블에서의 엔트리들의 수일 수도 있는 정수 값들에 캐스트될 수도 있고, 정상 태스크 포인터들은 단순 태스크들의 정수 값들과 오버랩하지 않을 16 진법 포인터 값들에 캐스트될 수도 있다. 상이한 포인터들에 캐스트된 상이 값들에 기초하여, 스케줄러는 태스크 포인터가 단순 태스크 포인터인지 또는 정상 태스크 포인터인지를 식별할 수도 있다. 도 5 에 예시된 예에서, 단순 태스크 포인터에 캐스트된 값은 예를 들어, 정수 값일 수도 있는 단순 태스크 테이블 (500) 에서의 커널 식별자일 수도 있고, 정상 태스크 포인터는 예를 들어 16 진법 값일 수도 있는 물리적 또는 가상 메모리 어드레스에 캐스트될 수도 있다. 태스크 포인터에 캐스트된 값에 기초하여, 스케줄러는 태스크 포인터가 단순 태스크에 대한 단순 태스크 포인터인지 또는 정상 태스크에 대한 정상 태스크 포인터인지를 결정할 수도 있다.

[0044] 일단 스케줄러가 커널 식별자와 연관된 엔트리 (508 내지 516) 를 식별했다면, 스케줄러는 엔트리 (508 내지 516) 에 대한 커널 포인터 컬럼 (504) 으로부터의 커널 포인터 및 커널 반복 카운터 컬럼 (506) 으로부터의 커널 반복 카운터를 판독할 수도 있다. 커널 포인터는 단순 태스크로서 구현될 경량 커널을 추출하기 위한 물리적 또는 가상 메모리 어드레스를 스케줄러에게 제공할 수도 있다. 커널 반복 카운터는 단순 태스크를 위해 실행될 커널 실행 반복들의 총 수를 스케줄러에게 제공할 수도 있다.

[0045] 도 5 의 예에서, 커널 반복 카운터 컬럼 (506) 은 단순 태스크를 위해 실행될 커널 실행 반복들의 총 수를 표현

하는 정수 값들로서 도시된 커널 반복 카운터 값들을 포함한다. 다양한 실시형태들에서, 커널 반복 카운터 컬럼 (506) 및 커널 반복 카운터 값들은 스트링의 상이한 포지션들이 경량 커널의 상이한 특성들을 표현하는 다양한 심볼 표현들의 스트링을 포함할 수도 있다. 예를 들어, 스트링의 상이한 포지션들은 단순 태스크를 위해 실행될 커널 실행 반복들의 총 수, 단순 태스크를 위해 실행될 커널 실행 반복들의 총 수의 제수 (divisor) (예를 들어, 상수 값 (constant value) 또는 스레드 이용가능성에 의존한 값), 단순 태스크를 위해 실행된 커널 실행 반복들의 총 수, 및 경량 커널을 실행하기 위한 제약들을 표현할 수도 있다. 다양한 실시형태들에서, 경량 커널을 실행하기 위한 제약들은 메인 스레드, 임계 스레드, 비-임계 스레드를 포함하는 경량 커널을 실행하기 위한 스레드를 지정하는 것, 경량 커널을 실행하기 위한 레이턴시 요건, 경량 커널을 저장하는 메모리에 대한 스레드를 실행하는 프로세서의 근접성을 포함할 수도 있다. 경량 커널에 대해 실행하는 단순 태스크는 경량 커널의 실행 전과 후에 커널 반복 카운터 값을 판독 및 업데이트할 수도 있다.

[0046] 다양한 실시형태들에서, 단순 태스크 테이블 (500) 은 글로벌로 액세스가능한, 중앙집중된 단순 태스크 테이블 (500), 또는 글로벌로 또는 로컬로 액세스가능한 분배된 단순 태스크 테이블들 (500) 일 수도 있다. 분배된 단순 태스크 테이블 (500) 은 경량 커널들을 실행하도록 지정된 프로세서 (14) 의 메모리 (16, 24, 206, 306) 상과 같이, 경량 커널들의 실행 디바이스에 아주 근접하여 위치될 수도 있다. 예를 들어, GPU (14b) 는 전문화된 컴퓨테이션 태스크들을 프로세싱하기 위해 그 자신의 메모리 (16, 24, 206, 306) 에서 별도의 단순 태스크 테이블 (500) 을 이용할 수도 있다.

[0047] 다양한 실시형태들에서, 단순 태스크 테이블 (500) 은 단순 태스크들에 대해 단순 동기화 기능성들을 제공할 수도 있다. 예를 들어, 백-투-백 루프들을 실행하기 위해, 다수의 단순 태스크들은 동일한 경량 커널들을 실행할 수도 있다. 다수의 단순 태스크들은 완료할 다수의 반복들을 할당받을 수도 있고, 루프의 실행은 일단 모든 단순 태스크들이 그들의 할당된 반복들을 완료하면 완료할 수도 있다. 포크-조인 (fork-join) 동기화 패턴은 그 할당된 반복들의 실행을 완료하는 단순 태스크에 응답하여 커널 반복 카운터에서 데이터를 업데이트하는 것에 의해 단순 태스크 테이블 (500) 에의 부기 (bookkeeping) 를 통해 루프에 대해 구현될 수도 있다.

[0048] 도 5 에 예시된 예는 단순 태스크 테이블 (500) 에 N 개의 엔트리들 (508 내지 516) 을 포함한다. 다양한 실시형태들에서, N 의 값은 상이한 수들로 캡될 수도 있다. 단순 태스크 테이블 (500) 은 단순 태스크 테이블 (500) 에 N 개의 엔트리들 (508 내지 516) 이 있을 때 풀릴 수도 있고, 추가적인 엔트리들은 기존 엔트리 (508 내지 516) 가 기존 엔트리 (508 내지 516) 와 연관된 경량 커널에 대한 반복들의 총 수의 실행의 완료에 기초하여 무효화되거나 또는 제거될 때까지 단순 태스크 테이블에 부가되지 않을 수도 있다. 결과적으로, 경량 커널들은 이용가능한 어떤 공간도 없기 때문에 단순 태스크 테이블 (500) 에 입력되지 않을 수도 있고, 이들 입력되지 않은 (un-entered) 경량 커널들은 정상 태스크들로서 실행될 수도 있다.

[0049] 도 6 은 다양한 실시형태들에 따른 런타임 단순 태스크 스케줄링 및 실행의 프로세스 플로우의 예시이다. 스케줄러 (600) 및 스레드 (602) 는 컴퓨팅 디바이스에서 임의의 프로세서 또는 프로세서들 (14) 의 조합에 의해 실행될 수도 있다. 일부 실시형태들에서, 스케줄러 (600) 는 전용 하드웨어에서 구현될 수도 있다. 참조의 용이함을 위해, 스케줄러 모듈 뿐만 아니라 전용 스케줄러 하드웨어를 실행하는 프로세서(들)는 일반적으로 스케줄러 (600) 로 지칭된다. 게다가, 태스크 큐 (400), 단순 태스크 테이블 (500), 및 경량 커널 (604) 은 컴퓨팅 디바이스의 메모리들 (16, 24, 206, 306) 의 임의의 조합 상에 저장될 수도 있다. 다양한 실시형태들에서, 태스크 큐 (400), 단순 태스크 테이블 (500), 및 경량 커널 (604) 은 경량 커널 (604) 을 실행하기 위한 레이턴시 및 리소스 요건들을 감소시키기 위하여 스케줄러 (600) 및 스레드 (602) 를 실행하는 프로세서들 (14) 의 조합에 아주 근접하여 메모리들 (16, 24, 206, 306) 의 임의의 조합 상에 저장될 수도 있다. 다양한 실시형태들에서, 다수의 스케줄러들 (600), 스레드들 (602), 태스크 큐들 (400), 단순 태스크 테이블들 (500), 및 경량 커널들 (604) 은 컴퓨팅 디바이스 상에서 구현될 수도 있다.

[0050] 도 6 에 예시된 예의 실시형태에서, 스케줄러 (600) 는 태스크 큐 (400) 에 액세스하여 태스크 큐의 슬롯을 판독 (606) 하고 단순 태스크 포인터를 추출 (608) 할 수도 있다. 단순 태스크 포인터의 디렉션 (direction) 에 따라, 스케줄러 (600) 는 단순 태스크 테이블 (500) 에 액세스하여 단순 태스크 포인터에 의해 지정된 엔트리를 판독 (610) 하고 커널 포인터 및 커널 반복 카운터 값을 포함할 수도 있는 엔트리의 데이터를 추출 (612) 할 수도 있다. 스케줄러 (600) 는 단순 태스크를 실행하기 위한 이용가능한 스레드 (602) 를 식별하고 단순 태스크를 실행하기 위한 명령들 (614) 을 스레드 (602) 에 전달할 수도 있는데, 이는 경량 커널을 (정상 태스크 보다는) 단순 태스크로서 실행하기 위한 명령들 (614), 커널 포인터, 및 단순 태스크를 위해 실행될 커널 실행 반복들의 수를 포함할 수도 있다. 스케줄러 (600) 는 경량 커널을 실행하기 위한 스레드를 선택할 때, 커널 상호작용 카운터 데이터에 표시될 수도 있는, 경량 커널의 실행에 배치된 제약들을 고려할 수도 있다. 스레

드 (602) 는 메모리에 액세스하라는 수신된 명령들로부터의 정보를 이용하여 커널 포인터에 의해 지정된 메모리 로케이션을 판독 (616) 하고, 메모리 로케이션으로부터 경량 커널 (604) 을 추출 (618) 할 수도 있다. 스레드 (602) 는 경량 커널을 단순 태스크로서 직접 실행 (620) 할 수도 있다. 스레드 (602) 는 단순 태스크 테이블 (500) 에 액세스하여, 커널 반복 카운터 값의 데이터를 업데이트하는 것, 또는 엔트리를 무효화 또는 삭제하는 것을 포함할 수도 있는, 실행된 단순 태스크에 대한 엔트리를 업데이트 (622) 할 수도 있다.

[0051] 도 7 은 단순 태스크를 큐잉하기 위한 일 실시형태 방법 (700) 을 예시한다. 방법 (700) 은 프로세서 또는 프로세서 코어와 같은 범용 하드웨어에서 실행하는 소프트웨어를 이용하는 컴퓨팅 디바이스에서, 전용 하드웨어에서, 또는 프로세서 및 전용 하드웨어에서 실행하는 소프트웨어의 조합에서 실행될 수도 있다.

[0052] 블록 (702) 에서, 컴퓨팅 디바이스는 프로세스를 실행하라는 요청을 수신할 수도 있다. 요청은 컴퓨팅 디바이스 상에서 트리거된 하드웨어 신호로부터 인터프리팅되거나, 또는 컴퓨팅 디바이스 상에서 실행하는 오퍼레이팅 시스템 또는 프로그램으로부터의 호출을 포함할 수도 있다.

[0053] 결정 블록 (704) 에서, 컴퓨팅 디바이스는 프로세스들의 컴퓨테이션 프로시저가 단순 태스크로서 실행될 수 있는 경량 커널인지 여부를 결정할 수도 있다. 다양한 실시형태들에서, 경량 커널들인 프로세스의 엘리먼트들은 경량 커널들로서 식별가능한 것으로 프리프로그램될 수도 있고 컴퓨팅 디바이스는 실행을 위해 요청된 프로세스들의 경량 커널들의 통지를 받을 수도 있다. 다양한 실시형태들에서, 컴퓨팅 디바이스는 경량 커널들로서 컴퓨테이션 프로시저들의 타입들을 식별하도록 프리프로그램될 수도 있고, 컴퓨팅 디바이스는 엘리먼트들 중 임의의 엘리먼트가 경량 커널을 표시하는 타입인지 여부를 결정하기 위해 실행을 위해 요청된 프로세스들의 엘리먼트들을 검사할 수도 있다.

[0054] 실행을 위해 요청된 프로세스들이 단순 태스크로서 실행될 수 있는 경량 커널을 포함하지 않는다고 결정하는 것에 응답하여 (즉, 결정 블록 (704) = "아니오"), 컴퓨팅 디바이스는 블록 (718) 에서 정상 태스크 포인터를 실행될 커널에 대한 태스크 큐에 부가할 수도 있다.

[0055] 실행을 위해 요청된 프로세스들이 단순 태스크로서 실행될 수 있는 경량 커널을 포함한다고 결정하는 것에 응답하여 (즉, 결정 블록 (704) = "예"), 컴퓨팅 디바이스는 결정 블록 (706) 에서 경량 커널에 대한 엔트리가 단순 태스크 테이블에 존재하는지 여부를 결정할 수도 있다. 단순 태스크 테이블에서 경량 커널에 대한 엔트리에 대해 체크하는 것은 단순 태스크 테이블에서의 커널 포인터들에 캐스트된 메모리 로케이션을 경량 커널의 메모리 로케이션과 비교하는 것을 포함할 수도 있다.

[0056] 경량 커널에 대한 엔트리가 단순 태스크 테이블에 존재한다고 결정하는 것에 응답하여 (즉, 결정 블록 (706) = "예"), 컴퓨팅 디바이스는 결정 블록 (708) 에서 기존 엔트리가 경량 커널의 요청된 실행에 적합한지 여부를 결정할 수도 있다. 다양한 실시형태들에서, 경량 커널의 일부 실행들은 커널 반복 카운터에서 표현된 연관된 데이터에 의해 표시된 경량 커널의 실행을 위한 다양한 특성들에 기초하여 경량 커널의 다른 실행들과 상이할 수도 있다. 컴퓨팅 디바이스는 경량 커널에 대한 매칭 커널 포인터와 연관된 커널 반복 카운터에서 표현된 경량 커널의 실행을 위한 특성들을 이용하여 경량 커널에 대한 엔트리가 경량 커널의 요청된 실행을 위한 단순 태스크 포인터에 캐스트하는데 적합한지 여부를 결정할 수도 있다.

[0057] 기존 엔트리가 경량 커널의 요청된 실행에 적합하다고 결정하는 것에 응답하여 (즉, 결정 블록 (708) = "예"), 컴퓨팅 디바이스는 블록 (710) 에서 단순 태스크 테이블에서 경량 커널에 대한 기존 엔트리를 업데이트할 수도 있다. 기존 엔트리를 업데이트하는 것은 경량 커널의 실행의 반복들의 수와 같이, 커널 반복 카운터에서 표현된 연관된 데이터에 의해 표시된 경량 커널을 실행하기 위한 특성 데이터를 업데이트하는 것을 포함할 수도 있다.

[0058] 경량 커널에 대한 엔트리가 단순 태스크 테이블에 존재하지 않는다고 결정하는 것에 응답하여 (즉, 결정 블록 (706) = "아니오"), 또는 엔트리가 경량 커널의 실행에 적합하지 않다고 결정하는 것에 응답하여 (즉, 결정 블록 (708) = "아니오"), 컴퓨팅 디바이스는 결정 블록 (714) 에서 단순 태스크 테이블이 풀인지 여부를 결정할 수도 있다. 단순 태스크 테이블은 N 개의 엔트리를 유지하는 것이 가능한 제한된 사이즈일 수도 있다. 컴퓨팅 디바이스는 추가적인 엔트리들이 단순 태스크 테이블에 부가될 수도 있는지 여부를 결정하기 위해 단순 태스크 테이블의 용량과 기존 엔트리들의 수를 비교할 수도 있다.

[0059] 단순 태스크 테이블이 풀이라고 결정하는 것에 응답하여 (즉, 결정 블록 (714) = "예"), 컴퓨팅 디바이스는 블록 (718) 에서 정상 태스크 포인터를 경량 커널에 대한 태스크 큐에 부가할 수도 있다.

[0060] 단순 태스크 테이블이 풀이 아니라고 결정하는 것에 응답하여 (즉, 결정 블록 (714) = "아니오"), 컴퓨팅 디바

이스는 블록 (716) 에서 단순 태스크 테이블에서 경량 커널에 대한 엔트리를 생성할 수도 있다. 컴퓨팅 디바이스는 단순 태스크 테이블에서의 엔트리의 로케이션 및 단순 태스크로서 실행을 위한 경량 커널을 표시하는 고유 커널 식별자, 경량 커널에 대한 물리적 또는 가상 메모리 로케이션에 대한 커널 포인터, 및 경량 커널을 실행하기 위한 특성들을 특정하는 커널 반복 카운터 데이터를 가진 새로운 엔트리를 생성할 수도 있다.

[0061] 블록 (712) 에서, 컴퓨팅 디바이스는 블록 (710) 에서 업데이트되거나 또는 블록 (716) 에서 생성된 단순 태스크 포인터를 경량 커널에 대한 단순 태스크 테이블에서의 기존 엔트리에 대한 태스크 큐에 부가할 수도 있다. 단순 태스크 포인터는 기존 엔트리에 대한 커널 식별자에 캐스트될 수도 있다.

[0062] 도 8 은 단순 태스크를 스케줄링하기 위한 일 실시형태 방법 (800) 을 예시한다. 방법 (800) 은 프로세서 또는 프로세서 코어와 같은 범용 하드웨어에서 실행하는 소프트웨어를 이용하는 컴퓨팅 디바이스에서, 전용 하드웨어에서, 또는 프로세서 및 전용 하드웨어에서 실행하는 소프트웨어의 조합에서 실행될 수도 있다.

[0063] 블록 (802) 에서, 컴퓨팅 디바이스는 태스크 큐로부터 포인터를 추출할 수도 있다. 추출된 포인터는 본 명세서에서 논의한 바와 같이 선입 선출 태스크 큐 관리 정책, 이용가능성 기반 태스크 큐 관리 정책, 우선순위 기반 태스크 큐 관리 정책, 또는 이들 태스크 큐 관리 정책의 조합을 포함할 수도 있는, 태스크 큐 관리 정책에 의해 좌우될 수도 있다.

[0064] 결정 블록 (804) 에서, 컴퓨팅 디바이스는 추출된 포인터가 단순 태스크 포인터인지 여부를 결정할 수도 있다. 단순 태스크 포인터들 및 정상 태스크 포인터들은 다른 타입의 태스크 포인터의 값과 오버랩하지 않는 값들에 캐스트되어, 태스크 포인터가 그들 캐스트 값들에 의해 단순 태스크 포인터인지 또는 정상 태스크 포인터인지를 식별하기 위한 방식을 제공할 수도 있다. 다양한 실시형태들에서, 단순 태스크 포인터에 캐스트된 값은, 예를 들어, 정수 값일 수도 있는, 단순 태스크 테이블에서의 커널 식별자일 수도 있고, 정상 태스크 포인터는, 예를 들어, 16 진법 값일 수도 있는, 물리적 또는 가상 메모리 어드레스에 캐스트될 수도 있다. 태스크 포인터에 캐스트된 값에 기초하여, 컴퓨팅 디바이스는 태스크 포인터가 단순 태스크에 대한 단순 태스크 포인터인지 또는 정상 태스크에 대한 정상 태스크 포인터인지를 결정할 수도 있다.

[0065] 추출된 포인터가 단순 태스크 포인터가 아니라고 결정하는 것에 응답하여 (즉, 결정 블록 (804) = "아니오"), 컴퓨팅 디바이스는 블록 (818) 에서 정상 태스크를 생성하고 그 정상 태스크를 실행을 위한 스레드에 할당할 수도 있다.

[0066] 추출된 포인터가 단순 태스크 포인터라고 결정하는 것에 응답하여 (즉, 결정 블록 (804) = "예"), 컴퓨팅 디바이스는 블록 (806) 에서 단순 태스크 테이블로부터 단순 태스크 포인터와 연관된 엔트리를 추출할 수도 있다. 본 명세서에서 설명한 바와 같이, 단순 태스크 포인터는 단순 태스크 테이블에서 엔트리의 커널 식별자에 캐스트될 수도 있고, 따라서 컴퓨팅 디바이스는 추출된 단순 태스크 포인터에 캐스트된 값에 매칭하는 커널 식별자를 갖는 단순 태스크 테이블의 엔트리를 추출할 수도 있다. 엔트리를 추출하는 것은 또한, 엔트리의 커널 반복 카운터의 데이터 및 커널 포인터를 추출하는 것을 포함할 수도 있다.

[0067] 블록 (808) 에서, 컴퓨팅 디바이스는 단순 태스크 포인터와 연관된 엔트리의 경량 커널을 실행하기 위한 임의의 제약들을 식별할 수도 있다. 설명한 바와 같이, 경량 커널의 실행에 대한 제약들은 총 커널 반복들의 데이터에 포함될 수도 있다. 컴퓨팅 디바이스는 제약들의 심볼 표현들에 대한 총 커널 반복들의 데이터에서 지정된 로케이션들을 체크할 수도 있다. 제약들은 메인 스레드, 임계 스레드, 비-임계 스레드를 포함하는, 경량 커널을 실행하기 위한 스레드를 지정하는 것, 경량 커널을 실행하기 위한 레이턴시 요건, 경량 커널을 저장하는 메모리에 대한 스레드를 실행하는 프로세서의 근접성을 포함할 수도 있다.

[0068] 블록 (810) 에서, 컴퓨팅 디바이스는 단순 태스크로서 경량 커널을 실행하기 위한 이용가능한 스레드들을 식별하여, 정상 태스크를 생성하지 않고 경량 커널을 직접 실행할 수도 있다. 이용가능한 스레드들을 식별하는 것은 이용가능한 스레드가 경량 커널을 실행하기 위한 임의의 제약들을 충족하는지 여부를 고려할 수도 있다.

[0069] 결정 블록 (812) 에서, 컴퓨팅 디바이스는 단순 태스크로서 실행될 커널 반복들이 분할가능한지 여부를 결정할 수도 있다. 설명한 바와 같이, 단순 태스크를 위해 실행될 커널 실행 반복들의 총 수, 및 단순 태스크를 위해 실행될 커널 실행 반복들의 총 수의 제수를 포함하는 것은 총 커널 반복들의 데이터에 포함될 수도 있다. 컴퓨팅 디바이스는 커널 반복들이 분할가능한지 여부 및 이용가능한 스레드들에의 할당을 위해 커널 반복들을 분할하는 방법을 결정하기 위해 커널 실행 반복들의 총 수 및 커널 실행 반복들의 총 수의 제수를 위해 총 커널 반복들의 데이터에서의 지정된 로케이션들을 체크할 수도 있다.

[0070] 단순 태스크로서 실행될 커널 반복들이 분할가능하다고 결정하는 것에 응답하여 (즉, 결정 블록 (812) = "예"),

컴퓨팅 디바이스는 블록 (814) 에서 단순 태스크로서의 실행을 위한 총 반복들을 커널 반복 부분들로 분할할 수도 있다. 다양한 실시형태들에서, 커널 반복 부분들은 그들이 이용가능한 스레드들에 걸쳐서 커널 반복들을 대칭적으로 또는 비대칭적으로 확산하도록 존재할 수도 있다. 다양한 실시형태들에서, 커널 반복 부분들은 그들이 이용가능한 스레드들의 수보다 더 크거나, 동일하거나, 또는 더 적은 수의 커널 반복 부분들에서의 총 반복들 전부를 설명하도록 존재할 수도 있다.

[0071] 단순 태스크로서 실행될 커널 반복들이 분할가능하지 않다고 결정하는 것에 응답하여 (즉, 결정 블록 (812) = "아니오"), 또는 단순 태스크로서의 실행을 위한 총 반복들을 커널 반복 부분들로 분할한 후에, 컴퓨팅 디바이스는 블록 (816) 에서 단순 태스크들로서 실행될 커널 반복 부분들의 일부 또는 전부를 하나 이상의 이용가능한 스레드들에 할당할 수도 있다. 실행을 위한 커널 반복 부분들의 할당은 이용가능한 스레드가 경량 커널을 실행하기 위한 임의의 제약들을 충족하는지 여부를 고려할 수도 있다.

[0072] 도 9 는 단순 태스크를 실행하기 위한 일 실시형태 방법 (900) 을 예시한다. 방법 (900) 은 프로세서 또는 프로세서 코어와 같은 범용 하드웨어에서 실행하는 소프트웨어를 이용하는 컴퓨팅 디바이스에서, 전용 하드웨어에서, 또는 프로세서 및 전용 하드웨어에서 실행하는 소프트웨어의 조합에서 실행될 수도 있다.

[0073] 블록 (902) 에서, 컴퓨팅 디바이스는 경량 커널을 추출할 수도 있다. 컴퓨팅 디바이스는 단순 태스크 포인터와 연관된 엔트리의 커널 포인터를 이용하여 커널 포인터에 캐스트된 메모리 로케이션으로부터 경량 커널을 추출할 수도 있다.

[0074] 블록 (904) 에서, 컴퓨팅 디바이스는 경량 커널을 단순 태스크로서 실행할 수도 있다. 컴퓨팅 디바이스는 추출된 경량 커널을 이용하고 그리고 정상 태스크의 구성을 우선 생성하지 않고 경량 커널을 직접 실행하여, 실행 동안 정상 태스크의 충돌들 및 의존성들을 관리하는 것을 회피하고, 완료 시 정상 태스크를 디스패치하는 것에 의해 스레드에서의 단순 태스크의 할당에 응답할 수도 있다. 따라서, 경량 커널의 실행은 태스크처럼, 구체적으로는 단순 태스크를 이용하여 스케줄링될 수도 있지만, 단순 태스크로서 경량 커널을 실행하는 것은 정상 태스크들을 스케줄링 및 실행하기 위해 요구되는 리소스 소비의 오버헤드를 회피한다.

[0075] 블록 (906) 에서, 컴퓨팅 디바이스는 커널 반복 부분의 실행을 위해 로컬 반복 카운터를 업데이트할 수도 있다. 로컬 반복 카운터를 업데이트하는 것은 커널 반복 부분에서 실행되거나 또는 실행을 위해 남겨진 커널 반복들의 수를 표시하기 위해 알고리즘을 이용하는 것, 증분하는 것, 또는 감분하는 것을 포함할 수도 있다.

[0076] 결정 블록 (908) 에서, 컴퓨팅 디바이스는 커널 반복 부분이 완료되는지 여부를 결정할 수도 있다. 다양한 실시형태들에서, 컴퓨팅 디바이스는 로컬 반복 카운터에서의 단순 태스크의 실행된 반복들의 수를 커널 반복 부분의 반복들의 수와 비교할 수도 있거나, 또는 로컬 반복 카운터가 주어진 값, 이를 테면 0 과 동일한지 여부를 체크할 수도 있다. 커널 반복 부분이 완료되지 않는다고 결정하는 것에 응답하여 (즉, 결정 블록 (908) = "아니오"), 컴퓨팅 디바이스는 설명한 바와 같이 블록 (904) 에서 경량 커널을 단순 태스크로서 실행하고 블록 (906) 에서 커널 반복 부분의 실행을 위해 로컬 반복 카운터를 업데이트할 수도 있다.

[0077] 커널 반복 부분이 완료된다고 결정하는 것에 응답하여 (즉, 결정 블록 (908) = "예"), 컴퓨팅 디바이스는 블록 (910) 에서 단순 태스크 테이블에서 단순 태스크에 대한 엔트리의 총 반복 카운터를 업데이트할 수도 있다. 다양한 실시형태들에서, 총 반복 카운터를 업데이트하는 것은 완료된 커널 반복 부분들의 반복들의 수를 총 반복 카운터의 데이터에서 완료된 것으로 표현된 커널 반복들의 수에 추가하는 것, 또는 총 반복 카운터의 데이터에서 완료될 커널 반복들의 수로부터 완료된 커널 반복 부분들의 반복들의 수를 차감하는 것을 포함할 수도 있다. 다양한 실시형태들에서, 완료된 또는 완료될 커널 반복들의 수는 커널 반복들의 수에 의해, 커널 반복 부분들의 수에 의해, 또는 커널 반복들 또는 커널 반복 부분들을 표현하는 심볼 값에 의해 표현될 수도 있다.

[0078] 결정 블록 (912) 에서, 컴퓨팅 디바이스는 임의의 커널 반복 부분들이 실행을 위해 남아있는지 여부를 결정할 수도 있다. 실행되지 않은 커널 반복 부분은 특정한 스레드에 할당되거나 또는 스레드들의 그룹에 할당될 수도 있다. 커널 반복 부분이 실행을 위해 남아있다고 결정하는 것에 응답하여 (즉, 결정 블록 (912) = "예"), 컴퓨팅 디바이스는 설명한 바와 같이 블록 (904) 에서 경량 커널을 단순 태스크로서 실행할 수도 있다.

[0079] 커널 반복 부분이 실행을 위해 남아있지 않다고 결정하는 것에 응답하여 (즉, 결정 블록 (912) = "아니오"), 컴퓨팅 디바이스는 결정 블록 (914) 에서 커널 반복 부분에 대한 단순 태스크 출력이 다른 커널 반복 부분에 대한 다른 단순 태스크 출력에 독립적인지 여부를 결정할 수도 있다. 이 방식으로, 컴퓨팅 디바이스는 경량 커널 실행들의 단순 동기화를 관리할 수도 있다. 예를 들어, 컴퓨팅 디바이스는 단일 출력을 초래하는 경량 커널 실행들이 병렬로 실행될 수도 있고 그들의 출력들이 실행들의 완료 후에 함께 조인될 수도 있도록 포크-조인 패

러다임을 구현할 수도 있다.

- [0080] 커널 반복 부분에 대한 단순 태스크 출력이 다른 커널 반복 부분에 대한 다른 단순 태스크 출력에 독립적이지 않다고 결정하는 것에 응답하여 (즉, 결정 블록 (914) = "아니오"), 컴퓨팅 디바이스는 블록 (916) 에서 의존적인 단순 태스크 출력들을 결합할 수도 있다.
- [0081] 커널 반복 부분에 대한 단순 태스크 출력이 다른 커널 반복 부분에 대한 다른 단순 태스크 출력에 독립적이라고 결정하는 것에 응답하여 (즉, 결정 블록 (914) = "예") 또는 컴퓨팅 디바이스가 블록 (916) 에서 의존적인 단순 태스크 출력들을 결합한 후에, 컴퓨팅 디바이스는 블록 (918) 에서 결합된 단순 태스크 출력을 출력할 수도 있다. 블록 (920) 에서, 컴퓨팅 디바이스는 추출된 경량 커널과 연관된 단순 태스크에 대한 엔트리를 단순 태스크 테이블에서 무효화하거나 또는 단순 태스크 테이블로부터 클리어할 수도 있다.
- [0082] 다양한 양태들 (도 1 내지 도 9 를 참조하여 상기 논의된 양태들을 포함하지만, 이들에 제한되지는 않음) 은 도 10 에 예시된 다양한 양태들에의 이용에 적합한 일 예의 모바일 컴퓨팅 디바이스를 포함할 수도 있는, 매우 다양한 컴퓨팅 시스템들에서 구현될 수도 있다. 모바일 컴퓨팅 디바이스 (1000) 는 내부 메모리 (1006) 및 터치스크린 제어기 (1004) 에 커플링된 프로세서 (1002) 를 포함할 수도 있다. 프로세서 (1002) 는 일반 또는 특정 프로세싱 태스크들에 대해 지정된 하나 이상의 멀티코어 집적 회로들일 수도 있다. 내부 메모리 (1006) 는 휘발성 또는 비휘발성 메모리일 수도 있고, 또한 보안 및/또는 암호화된 메모리, 또는 비보안 및/또는 비암호화된 메모리, 또는 그 임의의 조합일 수도 있다. 레버리징될 수 있는 메모리 타입들의 예들은 DDR, LPDDR, GDDR, WIDEIO, RAM, SRAM, DRAM, P-RAM, R-RAM, M-RAM, STT-RAM, 및 임베딩된 DRAM 을 포함하지만 이들에 제한되지는 않는다. 터치스크린 제어기 (1004) 및 프로세서 (1002) 는 또한 터치스크린 패널 (1012), 이를 테면 레지스트리브-센싱 터치스크린, 커패시티브-센싱 터치스크린, 적외선 센싱 터치스크린 등에 커플링될 수도 있다. 추가적으로, 컴퓨팅 디바이스 (1000) 의 디스플레이는 터치 스크린 능력을 가질 필요는 없다.
- [0083] 모바일 컴퓨팅 디바이스 (1000) 는 하나 이상의 무선 신호 트랜시버들 (1008) (예를 들어, 피넛 (Peanut), 블루투스, 지그비, Wi-Fi, RF 라디오 등) 및 서로에 및/또는 프로세서 (1002) 에 커플링된, 통신물들을 전송 및 수신하기 위한 안테나 (1010) 를 가질 수도 있다. 트랜시버들 (1008) 및 안테나 (1010) 는 다양한 무선 송신 프로토콜 스택들 및 인터페이스들을 구현하기 위해 상기 언급된 회로부와 함께 이용될 수도 있다. 모바일 컴퓨팅 디바이스 (1000) 는 셀룰러 네트워크를 통해 통신을 가능하게 하고 프로세서에 커플링되는 셀룰러 네트워크 무선 모듈 칩 (1016) 을 포함할 수도 있다.
- [0084] 모바일 컴퓨팅 디바이스 (1000) 는 프로세서 (1002) 에 커플링된 주변 디바이스 접속 인터페이스 (1018) 를 포함할 수도 있다. 주변 디바이스 접속 인터페이스 (1018) 는 하나의 타입의 접속을 수락하도록 단수로 구성될 수도 있거나, 또는 USB, FireWire, Thunderbolt, 또는 PCIe 와 같이, 공통 또는 사유의, 다양한 타입들의 물리적 및 통신 접속들을 수락하도록 구성될 수도 있다. 주변 디바이스 접속 인터페이스 (1018) 는 또한 유사하게 구성된 주변 디바이스 접속 포트 (미도시) 에 커플링될 수도 있다.
- [0085] 모바일 컴퓨팅 디바이스 (1000) 는 오디오 출력들을 제공하기 위한 스피커들 (1014) 을 또한 포함할 수도 있다. 모바일 컴퓨팅 디바이스 (1000) 는 또한, 본 명세서에서 논의된 컴포넌트들의 전부 또는 일부를 포함하기 위한, 플라스틱, 금속, 또는 재료들의 조합으로 구성된, 하우징 (1020) 을 포함할 수도 있다. 모바일 컴퓨팅 디바이스 (1000) 는 일회용 또는 재충전가능한 배터리와 같이, 프로세서 (1002) 에 커플링된 전력 소스 (power source) (1022) 를 포함할 수도 있다. 재충전가능한 배터리는 또한 모바일 컴퓨팅 디바이스 (1000) 외부의 소스로부터 충전 전류를 수신하기 위해 주변 디바이스 접속 포트에 커플링될 수도 있다. 모바일 컴퓨팅 디바이스 (1000) 는 또한 사용자 입력들을 수신하기 위한 물리적 버튼 (1024) 을 포함할 수도 있다. 모바일 컴퓨팅 디바이스 (1000) 는 또한 모바일 컴퓨팅 디바이스 (1000) 를 턴 온 및 턴 오프하기 위한 전원 버튼 (1026) 을 포함할 수도 있다.
- [0086] 다양한 양태들 (도 1 내지 도 9 를 참조하여 상기 논의된 양태들을 포함하지만, 이들에 제한되지는 않음) 은, 도 11 에 예시된 랩톱 컴퓨터 (1100) 와 같은 다양한 모바일 컴퓨팅 디바이스들을 포함할 수도 있는 매우 다양한 컴퓨팅 시스템들에서 구현될 수도 있다. 많은 랩톱 컴퓨터들은 컴퓨터의 포인팅 디바이스로서 서빙하는 터치패드 터치 표면 (1117) 을 포함하고, 따라서 상기 설명되고 터치 스크린 디스플레이가 구비된 컴퓨팅 디바이스들 상에서 구현된 것들과 유사하게 드래그, 스크롤, 및 플릭 제스처 (flick gesture) 들을 수신할 수도 있다. 랩톱 컴퓨터 (1100) 는 통상적으로 휘발성 메모리 (1112) 및 대용량 비휘발성 메모리, 이를 테면 플래시 메모리의 디스크 드라이브 (1113) 에 커플링된 프로세서 (1111) 를 포함할 것이다. 추가적으로, 컴퓨터

(1100) 는 무선 데이터 링크에 접속될 수도 있는 전자기 방사를 전송 및 수신하기 위한 하나 이상의 안테나 (1108) 및/또는 프로세서 (1111) 에 커플링된 셀룰러 전화기 트랜시버 (1116) 를 가질 수도 있다. 컴퓨터 (1100) 는 또한 프로세서 (1111) 에 커플링된 플로피 디스크 드라이브 (1114) 및 콤팩트 디스크 (CD) 드라이브 (1115) 를 포함할 수도 있다. 노트북 구성에서, 컴퓨터 하우징은 전부가 프로세서 (1111) 에 커플링되는 터치패드 (1117), 키보드 (1118), 및 디스플레이 (1119) 를 포함한다. 컴퓨팅 디바이스의 다른 구성들은, 다양한 양태들과 함께 또한 이용될 수도 있는, 잘 알려져 있는 바와 같은 (예를 들어, 범용 시리얼 버스 (USB) 입력력을 통해) 프로세서에 커플링된 컴퓨터 마우스 또는 트랙볼을 포함할 수도 있다.

[0087] 다양한 양태들 (도 1 내지 도 9 를 참조하여 상기 논의된 양태들을 포함하지만, 이들에 제한되지는 않음) 은 서버 캐시 메모리에서 데이터를 압축하기 위한 다양한 상업적으로 입수가능한 서버들 중 임의의 것을 포함할 수도 있는, 매우 다양한 컴퓨팅 시스템들에서 구현될 수도 있다. 일 예의 서버 (1200) 가 도 12 에 예시된다.

이러한 서버 (1200) 는 통상적으로 휘발성 메모리 (1202) 및 대용량 비휘발성 메모리, 이를 테면 디스크 드라이브 (1204) 에 커플링된 하나 이상의 멀티-코어 프로세서 어셈블리들 (1201) 을 포함한다. 도 12 에 예시한 바와 같이, 멀티-코어 프로세서 어셈블리들 (1201) 은 그들을 어셈블리의 랙들에 삽입함으로써 서버 (1200) 에 부가될 수도 있다. 서버 (1200) 는 또한 프로세서 (1201) 에 커플링된 플로피 디스크 드라이브, 콤팩트 디스크 (CD) 또는 DVD 디스크 드라이브 (1206) 를 포함할 수도 있다. 서버 (1200) 는 네트워크 (1205), 이를 테면 다른 브로드캐스트 시스템 컴퓨터들 및 서버들에 커플링된 로컬 영역 네트워크, 인터넷, 공중 교환된 전화기 네트워크, 및/또는 셀룰러 데이터 네트워크 (예를 들어, CDMA, TDMA, GSM, PCS, 3G, 4G, LTE, 또는 임의의 다른 타입의 셀룰러 데이터 네트워크) 와 네트워크 인터페이스 접속들을 확립하기 위한 멀티-코어 프로세서 어셈블리들 (1201) 에 커플링된 네트워크 액세스 포트들 (1203) 을 포함할 수도 있다.

[0088] 다양한 양태들의 동작들을 수행하기 위한 프로그래밍가능 프로세서 상에서의 실행을 위한 컴퓨터 프로그램 코드 또는 "프로그램 코드" 는 고급 프로그래밍 언어, 이를 테면, C, C++, C#, 스몰토크 (Smalltalk), 자바 (Java), 자바스크립트 (JavaScript), 비주얼 베이직 (Visual Basic), 구조화 질의 언어 (Structured Query Language) (예를 들어, Transact-SQL), 펄 (Perl) 로, 또는 다양한 다른 프로그래밍 언어들로 기입될 수도 있다. 본 출원에서 사용되는 바와 같은 컴퓨터 판독가능 저장 매체 상에 저장된 프로그램 코드 또는 프로그램들은 포맷이 프로세서에 의해 이해가능한 기계 언어 코드 (이를 테면 오브젝트 코드) 를 지칭할 수도 있다.

[0089] 시스템 커널들을 동작시키는 많은 컴퓨팅 디바이스들은 사용자 공간 (비-특권부여된 코드가 실행) 및 커널 공간 (특권부여된 코드가 실행) 으로 조직화된다. 이 분리는 안드로이드 및 다른 GPL (general public license) 환경들에서 특히 중요한데, 그 환경들에서 커널 공간의 일부인 코드는 GPL 라이선싱되어야 하는 한편, 사용자-공간에서 실행되는 코드는 GPL 라이선싱되지 않을 수도 있다. 여기에 논의된 다양한 소프트웨어 컴포넌트들/모듈들은 명확히 다르게 언급하지 않는 한, 커널 공간 또는 사용자 공간 중 어느 하나에서 구현될 수도 있는 것으로 이해되어야 한다.

[0090] 전술한 방법 설명들 및 프로세스 플로우 다이어그램들은 단순히 예시적인 예들로서만 제공될 뿐이며 다양한 양태들이 동작들이 제시된 순서로 수행되어야 한다는 것을 요구하거나 또는 의미하도록 의도되지 않는다. 당업자들에 의해 인식될 바와 같이, 전술한 양태들에서의 동작들의 순서는 임의의 순서로 수행될 수도 있다. "그 후에", "그 후", "다음에", 등과 같은 단어들은 동작들의 순서를 제한하도록 의도되지 않는다; 이들 단어들은 단순히 방법들의 설명을 통하여 독자를 안내하는데 사용된다. 게다가, 예를 들어, 관사들 "a", "an" 또는 "the" 를 이용하여, 단수로의 청구항 엘리먼트들에 대한 임의의 참조는 엘리먼트를 단수로 제한하는 것으로 해석되어서는 안된다.

[0091] 다양한 양태들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 회로들, 및 알고리즘 동작들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양자의 조합들로서 구현될 수도 있다. 하드웨어와 소프트웨어의 이 상호교환가능성을 명확히 예시하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 동작들은 그들의 기능성의 관점에서 일반적으로 상기 설명되었다. 이러한 기능성이 하드웨어로서 구현되는지 소프트웨어로서 구현되는지 여부는 전체 시스템에 부과된 설계 제약들 및 특정한 애플리케이션에 의존한다. 당업자들은 각각의 특정한 애플리케이션에 대해 다양한 방식으로 설명된 기능성을 구현할 수도 있지만, 이러한 구현 관점들은 본 발명의 범위로부터 벗어남을 야기하는 것으로서 해석되어서는 안된다.

[0092] 본 명세서에서 개시된 양태들과 관련하여 설명된 다양한 예시적인 로직들, 논리 블록들, 모듈들, 및 회로들을 구현하는데 이용되는 하드웨어는 범용 프로세서, 디지털 신호 프로세서 (DSP), 주문형 집적 회로 (ASIC), 필드 프로그래밍가능 게이트 어레이 (FPGA) 또는 다른 프로그래밍가능 로직 디바이스, 이산 게이트 또는 트랜지스터

로직, 이산 하드웨어 컴포넌트들, 또는 본 명세서에서 설명된 기능들을 수행하도록 설계된 그 임의의 조합으로 구현 또는 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안으로, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예를 들어, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합된 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 대안적으로, 일부 동작들 또는 방법들은 주어진 함수에 특징적인 회로부에 의해 수행될 수도 있다.

[0093]

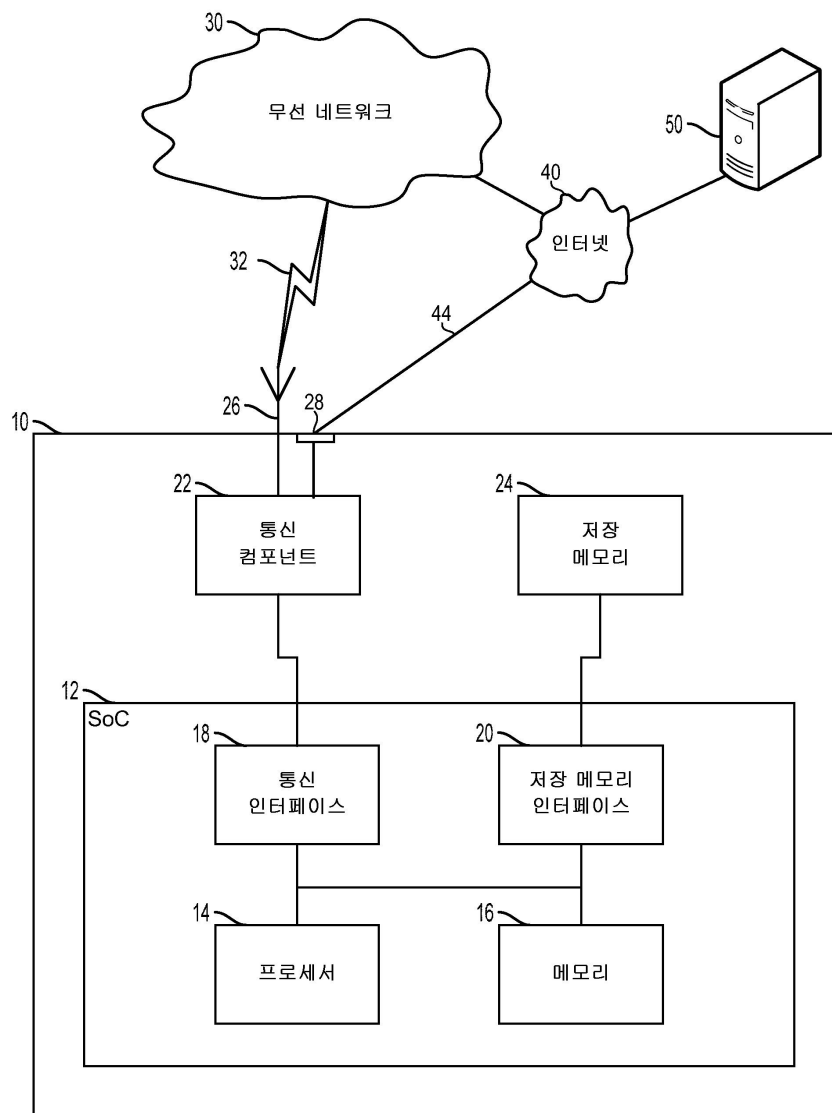
하나 이상의 양태들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 기능들은 비일시적 컴퓨터 판독가능 매체 또는 비일시적 프로세서 판독가능 매체 상에 하나 이상의 명령들 또는 코드로서 저장될 수도 있다. 본 명세서에서 개시된 방법 또는 알고리즘의 동작들은 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체 상에 상주할 수도 있는 프로세서 실행가능 소프트웨어 모듈에서 구현될 수도 있다. 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체들은 컴퓨터 또는 프로세서에 의해 액세스될 수도 있는 임의의 저장 매체들일 수도 있다. 제한이 아닌 일 예로, 이러한 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 매체들은 RAM, ROM, EEPROM, FLASH 메모리, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 저장 디바이스들, 또는 명령들 또는 데이터 구조들의 형태로 원하는 프로그램 코드를 저장하는데 이용될 수도 있고 컴퓨터에 의해 액세스될 수도 있는 임의의 다른 매체를 포함할 수도 있다. 디스크 (disk) 및 디스크 (disc) 는, 본 명세서에서 사용한 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루-레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 보통 데이터를 자기적으로 재생하는 한편, 디스크 (disc) 들은 레이저들로 데이터를 광학적으로 재생한다. 상기의 조합들이 또한 비일시적 컴퓨터 판독가능 및 프로세서 판독가능 매체들의 범위 내에 포함된다. 추가적으로, 방법 또는 알고리즘의 동작들은 컴퓨터 프로그램 제품에 통합될 수도 있는, 비일시적 프로세서 판독가능 매체 및/또는 컴퓨터 판독가능 매체 상에 코드들 및/또는 명령들의 하나 또는 임의의 조합 또는 세트로서 상주할 수도 있다.

[0094]

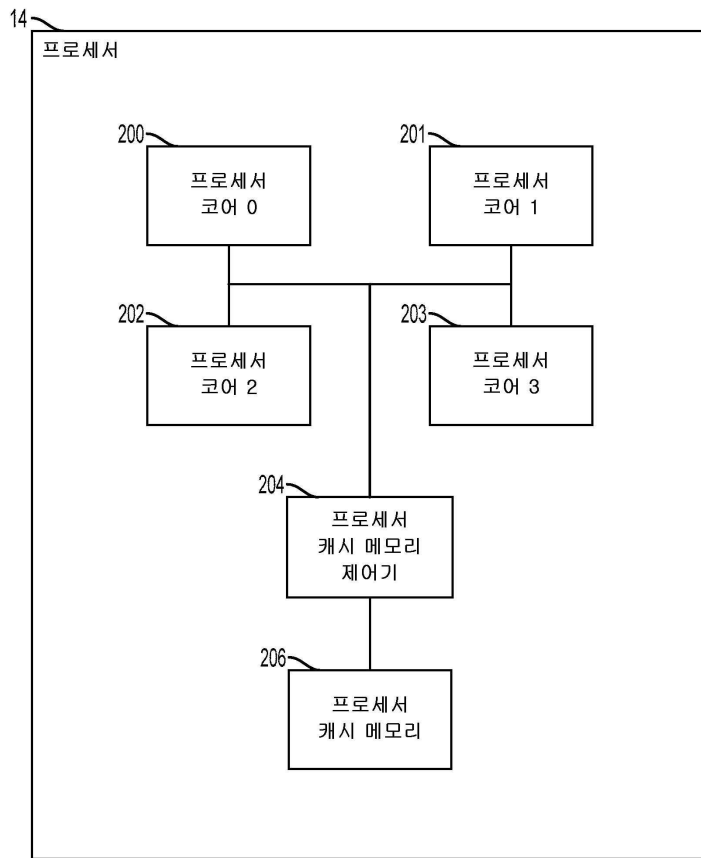
개시된 양태들의 기술한 설명은 임의의 당업자가 본 발명을 제조 또는 이용하는 것을 가능하게 하기 위해 제공된다. 이들 양태들에 대한 다양한 변경들은 당업자들에게 용이하게 명백할 것이며, 본 명세서에서 정의된 일반적인 원리들은 본 발명의 사상 또는 범위로부터 벗어남 없이 다른 양태들에 적용될 수도 있다. 따라서, 본 발명은 본 명세서에서 도시된 양태들에 제한되도록 의도되지 않고 다음의 청구항들 및 본 명세서에서 개시된 원리들 및 신규한 피쳐들에 부합하는 최광의 범위를 부여받게 하려는 것이다.

도면

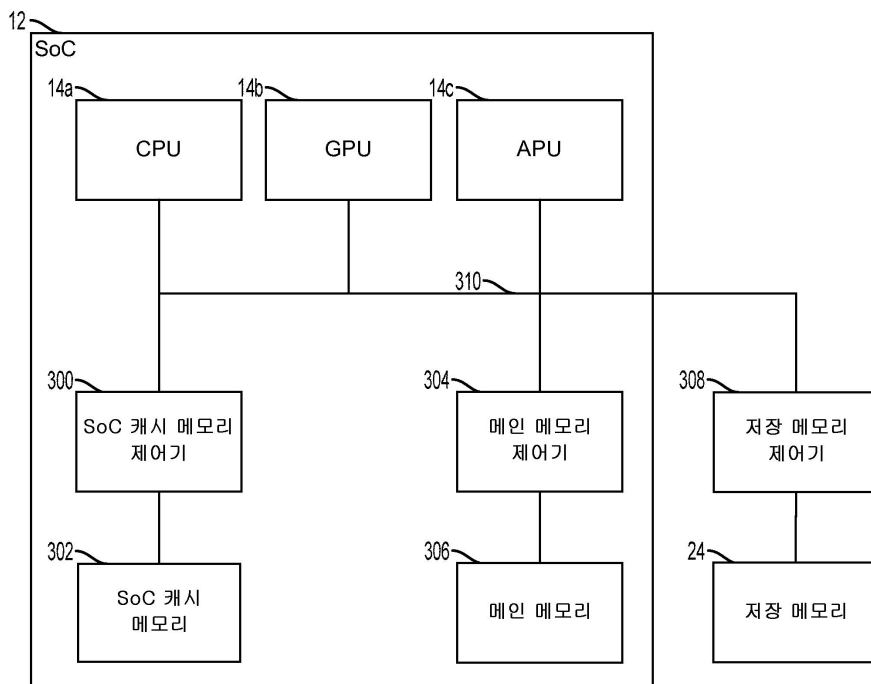
도면1



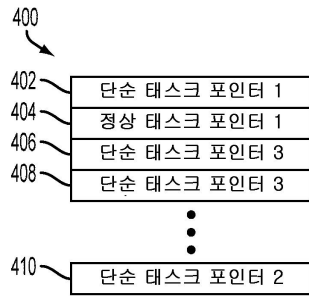
도면2



도면3



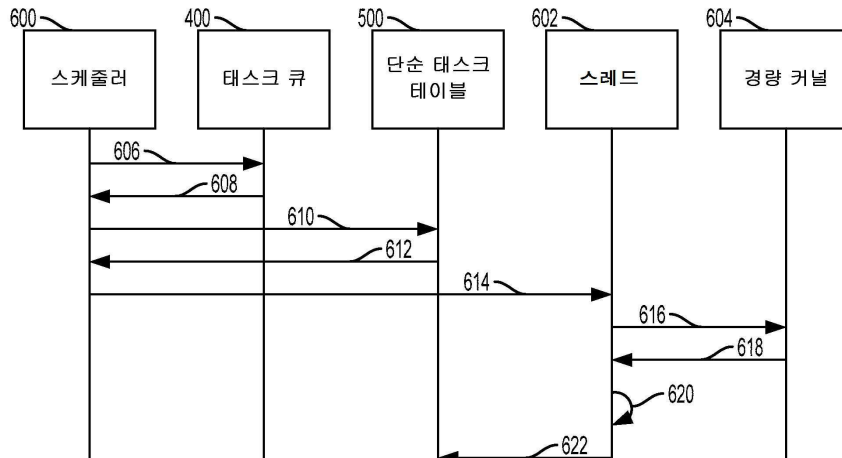
도면4



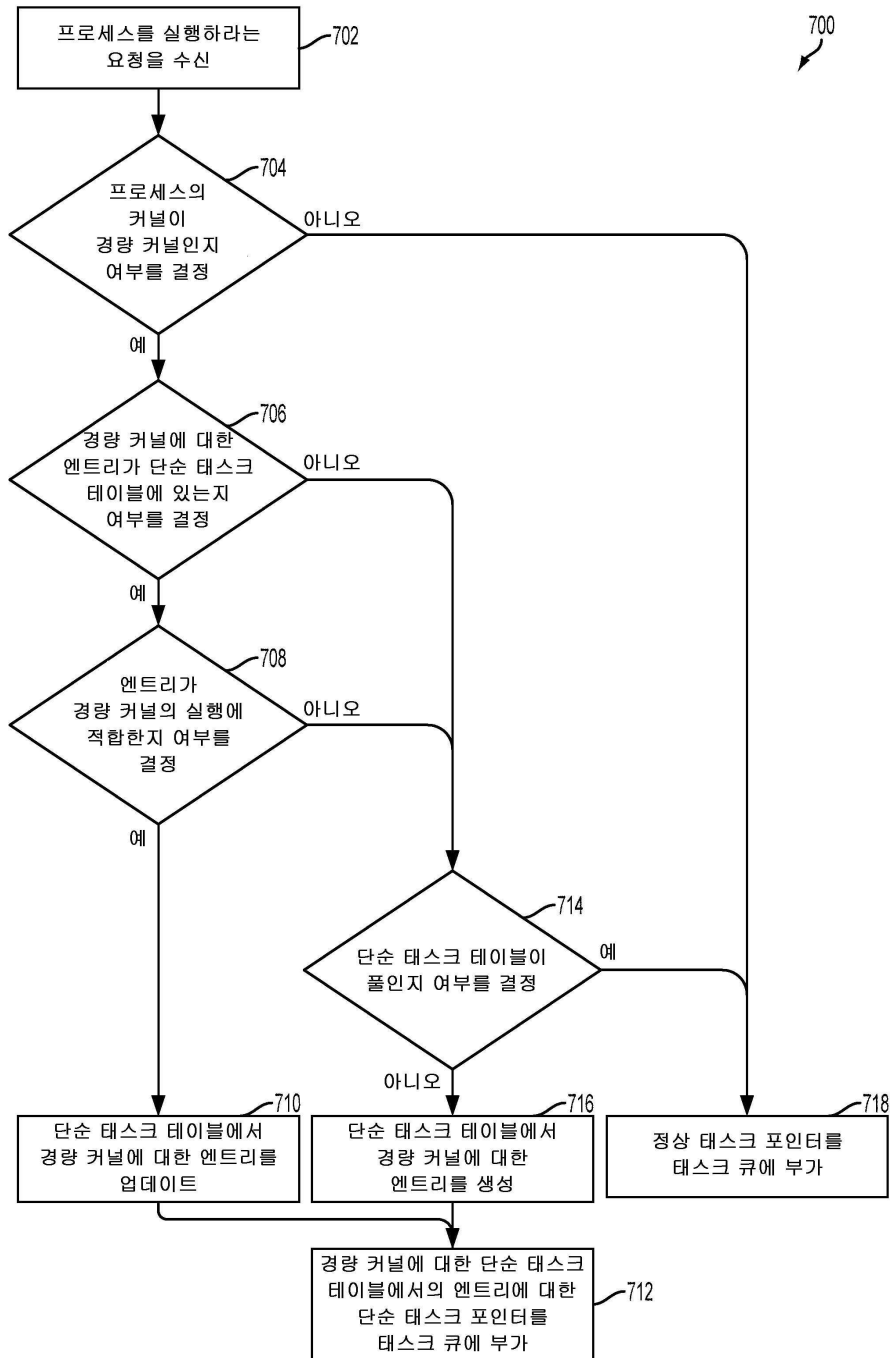
도면5

502	504	506
커널 ID	커널 포인터	커널 반복 카운터
508 1	커널 포인터 1	500,000
510 2	커널 포인터 2	1
512 3	커널 포인터 3	750
⋮		
516 N	커널 포인터 N	62,000

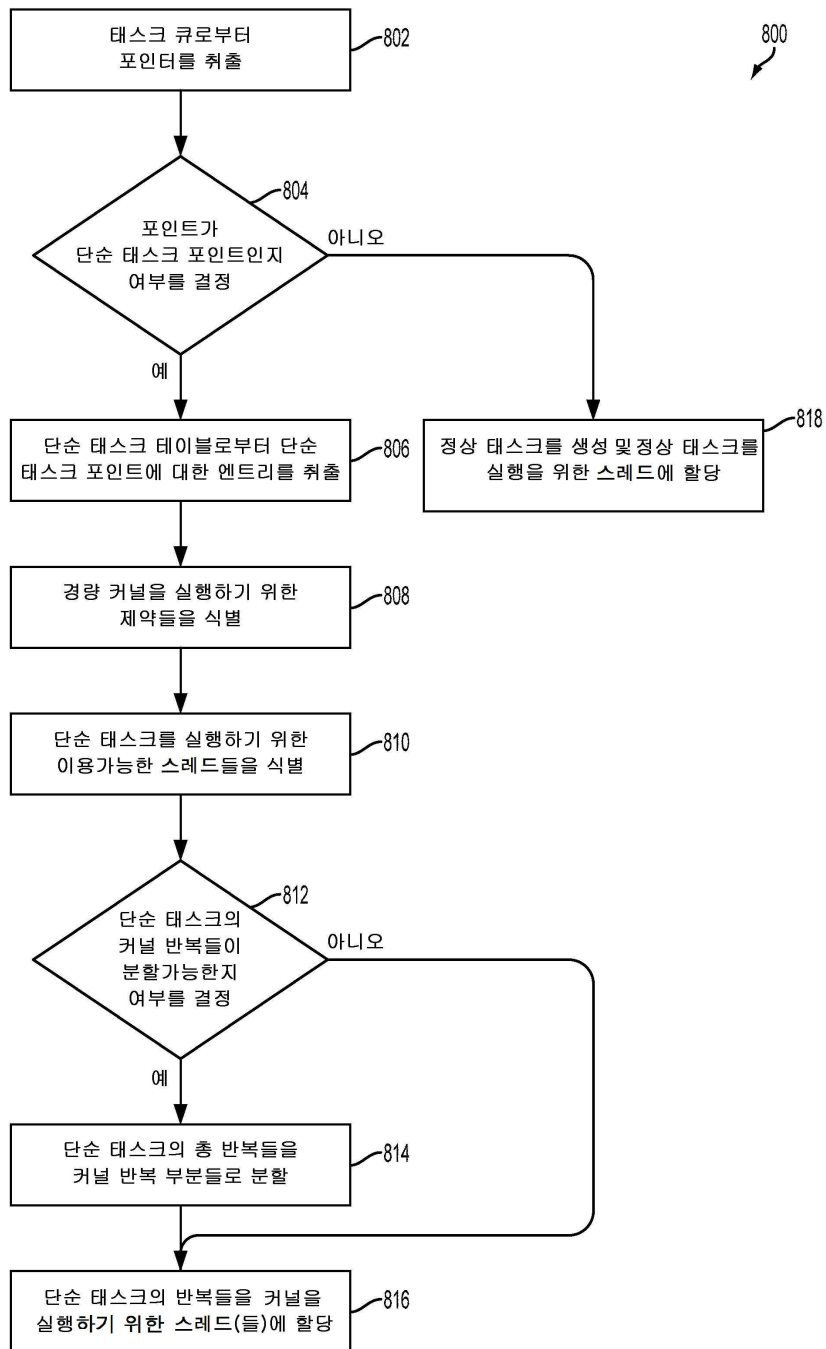
도면6



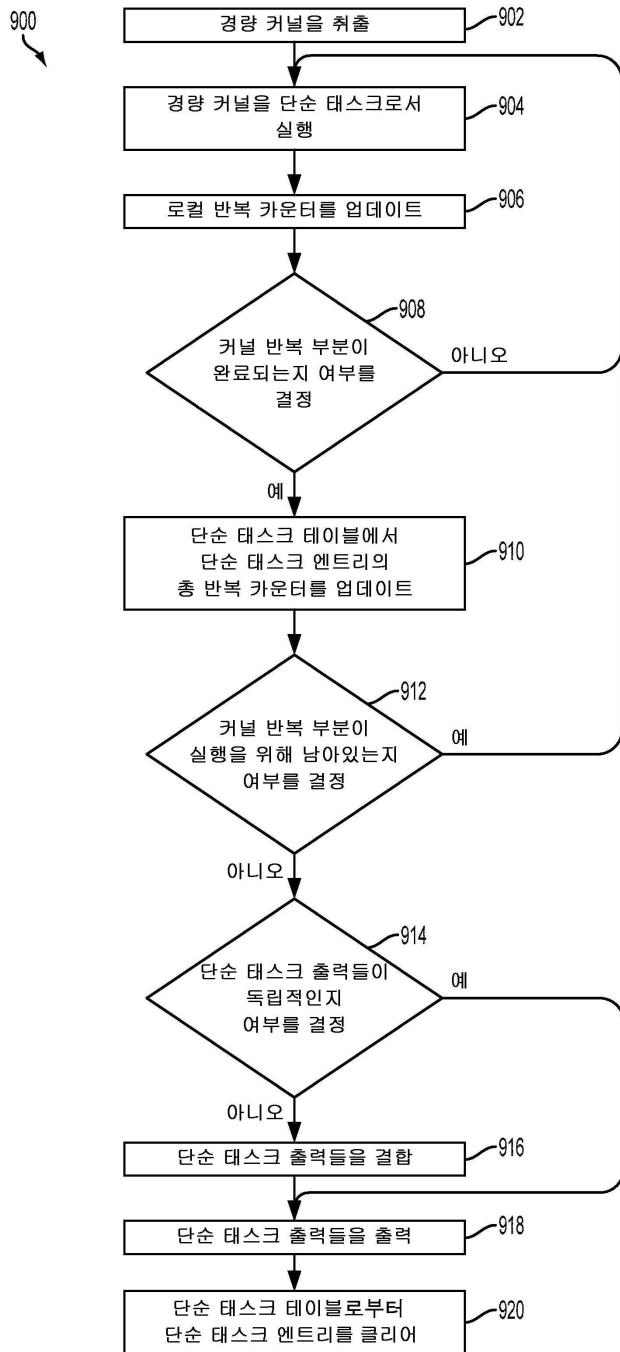
도면7



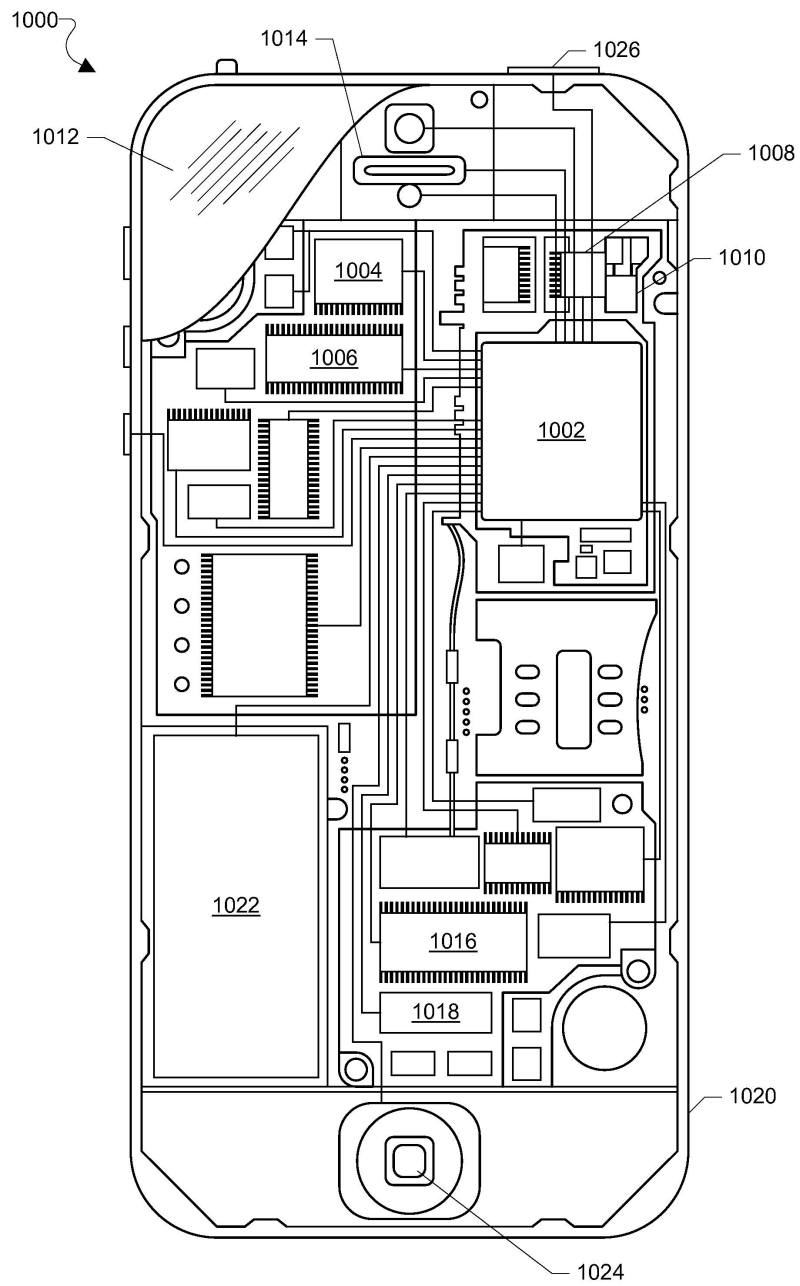
도면8



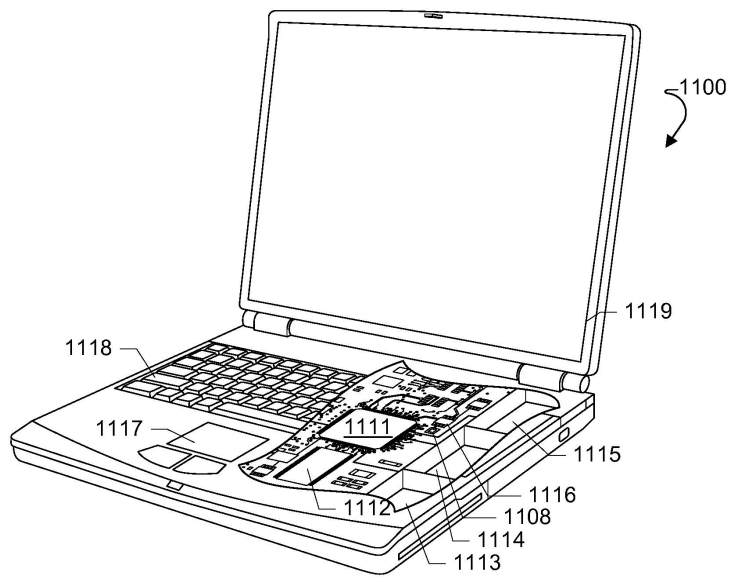
도면9



도면10



도면11



도면12

