



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0015460
(43) 공개일자 2025년02월03일

(51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01) G06F 9/48 (2018.01)
(52) CPC특허분류
G06F 9/5077 (2013.01)
G06F 9/4806 (2013.01)
(21) 출원번호 10-2023-0096899
(22) 출원일자 2023년07월25일
심사청구일자 2023년07월25일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
정선화
서울특별시 노원구 동일로176길 25 (공릉동)
반효경
서울특별시 관악구 관악로30길 12, 107동 2001호 (봉천동, 관악우성아파트)
조경운
서울특별시 서초구 방배천로2안길 118, 3층 (방배동)
(74) 대리인
특허법인 무한

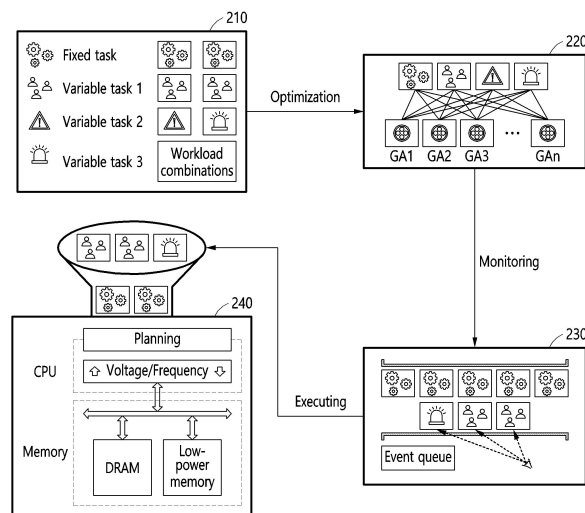
전체 청구항 수 : 총 19 항

(54) 발명의 명칭 자원 할당 방법 및 상기 방법을 수행하는 컴퓨팅 장치

(57) 요약

자원 할당 방법이 개시된다. 자원 할당 방법은 고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 단계; 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원의 할당과 관련된 자원 계획을 결정하는 단계; 상기 고정 태스크로 구성된 제1 워크로드(Workload)를 처리하기 위한 자원 할당을 수행하는 단계; 및 상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 적어도 하나 이상의 태스크 조합에 대해 결정된 자원 계획을 이용함으로써, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당을 재수행하는 단계를 포함할 수 있다.

대표도



(52) CPC특허분류

G06F 9/5016 (2013.01)

G06F 9/5027 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711179344
과제번호	00155966
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	인공지능융합혁신인재양성
연구과제명	인공지능융합혁신인재양성(이화여자대학교)
기 여 율	1/2
과제수행기관명	이화여자대학교산학협력단
연구기간	2023.01.01 ~ 2023.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711193622
과제번호	2021-0-02068-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성
연구과제명	인공지능 혁신 허브 연구 개발
기 여 율	1/2
과제수행기관명	이화여자대학교산학협력단
연구기간	2023.01.01 ~ 2023.12.31

명세서

청구범위

청구항 1

고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 단계;

상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원의 할당과 관련된 자원 계획을 결정하는 단계;

상기 고정 태스크로 구성된 제1 워크로드(Workload)를 처리하기 위한 자원 할당을 수행하는 단계; 및

상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 적어도 하나 이상의 태스크 조합에 대해 결정된 자원 계획을 이용함으로써, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당을 재수행하는 단계

를 포함하는 자원 할당 방법.

청구항 2

제1항에 있어서,

상기 자원 계획을 결정하는 단계는,

상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원을 할당하기 위한 복수의 솔루션들로 구성된 솔루션 집합을 생성하는 단계;

상기 생성된 솔루션 집합에서 효용 값에 따라 2 개의 부모 솔루션을 선택하는 단계;

상기 선택된 2개의 부모 솔루션에 미리 설정된 연산을 수행함으로써 자식 솔루션을 생성하는 단계; 및

상기 생성된 자식 솔루션을 이용하여 상기 솔루션 집합을 재구성하는 단계

를 포함하는 자원 할당 방법.

청구항 3

제2항에 있어서,

상기 복수의 솔루션들 각각은,

상기 태스크 조합에 포함된 각각의 태스크들을 처리하기 위한 CPU(central processing unit) 전압/주파수 모드 및 메모리 종류에 대한 정보를 포함하는 자원 할당 방법.

청구항 4

제2항에 있어서,

상기 부모 솔루션을 선택하는 단계는,

상기 복수의 솔루션들 각각의 효용 값에 따라 확률적으로 수행되는 자원 할당 방법.

청구항 5

제2항에 있어서,

상기 자식 솔루션을 생성하는 단계는,

상기 선택된 2개의 부모 솔루션에 교차 연산 및 돌연변이 연산을 수행하여 자식 솔루션을 생성하는 자원 할당 방법.

청구항 6

제2항에 있어서,

상기 솔루션 집합을 재구성하는 단계는,

상기 솔루션 집합을 구성하는 복수의 솔루션들 중 효용 값이 가장 작은 솔루션을 상기 생성된 자식 솔루션으로 대체하는 자원 할당 방법.

청구항 7

제2항에 있어서,

상기 자원 계획을 결정하는 단계는,

상기 재구성된 솔루션 집합을 구성하는 솔루션들의 효용 값이 미리 설정된 차이 이하가 되도록 반복하여 상기 솔루션 집합을 재구성하는 자원 할당 방법.

청구항 8

제1항에 있어서,

상기 자원 할당을 재수행하는 단계는,

상기 제2 워크로드의 제1 패턴과 상기 적어도 하나 이상의 태스크 조합 각각에 대한 워크로드의 제2 패턴을 비교하는 단계; 및

비교 결과 상기 제1 패턴과 제2 패턴 사이의 차이가 가장 작은 태스크 조합에 결정된 자원 계획에 따라 상기 자원 할당을 재수행하는 단계

를 포함하는 자원 할당 방법.

청구항 9

고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 단계;

상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원을 할당하기 위한 복수의 솔루션들로 구성된 솔루션 집합을 생성하는 단계;

상기 생성된 솔루션 집합에서 상기 복수의 솔루션들 각각의 효용 값에 따라 확률적으로 2 개의 부모 솔루션을 선택하는 단계;

상기 선택된 2개의 부모 솔루션에 교차 연산 및 돌연변이 연산을 수행함으로써 자식 솔루션을 생성하는 단계; 및

상기 솔루션 집합을 구성하는 복수의 솔루션들 중 효용 값이 가장 작은 솔루션을 상기 생성된 자식 솔루션으로 대체함으로써 상기 솔루션 집합을 재구성하는 단계

를 포함하고,

상기 솔루션 집합은,

상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당에 이용되는 자원 할당 방법.

청구항 10

제9항에 있어서,

상기 솔루션 집합은,

상기 복수의 솔루션들의 효용 값이 미리 설정된 차이 이하가 되도록 반복하여 재구성되는 자원 할당 방법.

청구항 11

제9항에 있어서,

상기 복수의 솔루션들 각각은,

상기 태스크 조합에 포함된 각각의 태스크들을 처리하기 위한 CPU(central processing unit) 전압/주파수 모드 및 메모리 종류에 대한 정보를 포함하는 자원 할당 방법.

청구항 12

컴퓨팅 장치에 있어서,

하나 이상의 프로세서; 및

상기 프로세서에 의해 실행되는 프로그램을 로드(load)하거나 저장하는 메모리를 포함하되,

상기 프로그램은,

고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 동작, 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원의 할당과 관련된 자원 계획을 결정하는 동작, 상기 고정 태스크로 구성된 제1 워크로드(Workload)를 처리하기 위한 자원 할당을 수행하는 동작 및 상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 적어도 하나 이상의 태스크 조합에 대해 결정된 자원 계획을 이용함으로써, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당을 재수행하는 동작을 수행하도록 하는 인스트럭션들을 포함하는 컴퓨팅 장치.

청구항 13

제12항에 있어서,

상기 프로세서는,

상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원을 할당하기 위한 복수의 솔루션들로 구성된 솔루션 집합을 생성하고, 상기 생성된 솔루션 집합에서 효율 값에 따라 2 개의 부모 솔루션을 선택하며, 상기 선택된 2개의 부모 솔루션에 미리 설정된 연산을 수행함으로써 자식 솔루션을 생성하고, 상기 생성된 자식 솔루션을 이용하여 상기 솔루션 집합을 재구성함으로써 상기 자원 계획을 설정하는 컴퓨팅 장치.

청구항 14

제13항에 있어서,

상기 복수의 솔루션들 각각은,

상기 태스크 조합에 포함된 각각의 태스크들을 처리하기 위한 CPU(central processing unit) 전압/주파수 모드 및 메모리 종류에 대한 정보를 포함하는 컴퓨팅 장치

청구항 15

제13항에 있어서,

상기 프로세서는,

상기 복수의 솔루션들 각각의 효율 값에 따라 확률적으로 부모 솔루션을 선택하는 컴퓨팅 장치.

청구항 16

제13항에 있어서,

상기 프로세서는,

상기 선택된 2개의 부모 솔루션에 교차 연산 및 돌연변이 연산을 수행하여 자식 솔루션을 생성하는 컴퓨팅

장치.

청구항 17

제13항에 있어서,

상기 프로세서는,

상기 솔루션 집합을 구성하는 복수의 솔루션들 중 효용 값이 가장 작은 솔루션을 상기 생성된 자식 솔루션으로 대체함으로써 상기 솔루션 집합을 재구성하는 컴퓨팅 장치.

청구항 18

제13항에 있어서,

상기 프로세서는,

상기 재구성된 솔루션 집합을 구성하는 솔루션들의 효용 값이 미리 설정된 차이 이하가 되도록 반복하여 상기 솔루션 집합을 재구성함으로써 상기 자원 계획을 결정하는 컴퓨팅 장치.

청구항 19

제12항에 있어서,

상기 프로세서는,

상기 제2 워크로드의 제1 패턴과 상기 적어도 하나 이상의 태스크 조합 각각에 대한 워크로드의 제2 패턴을 비교하고, 비교 결과 상기 제1 패턴과 제2 패턴 사이의 차이가 가장 작은 태스크 조합에 결정된 자원 계획에 따라 상기 자원 할당을 재수행하는 컴퓨팅 장치.

발명의 설명

기술 분야

[0001] 본 발명은 자원 할당 방법에 관한 것으로, 보다 구체적으로는 시간에 따라 변동하는 워크로드를 처리하기 위한 적응형 자원 할당 방법 및 장치에 관한 것이다.

배경 기술

[0002] 현대의 실시간 시스템에서 AI(Artificial Intelligence) 워크로드를 실행하는 비중이 증가하고 있는 추세이다. 이러한 AI 워크로드의 실시간 태스크 세트는 일정하지 않고, 시간이 지남에 따라 달라진다. 예를 들어, 자율 주행의 태스크 세트는 교통 및 도로 상황에 따라 달라질 수 있다.

[0003] 기존의 자원 관리 정책은 고정된 태스크 세트로 구성된 워크로드를 처리하는데 이용될 수 있으나, AI 워크로드를 처리하는 경우, AI 워크로드의 각 태스크 세트에 대한 마감시간을 지키는 것이 어려울 수 있다.

[0004] 따라서, 실시간 시스템에서 고정 태스크와 가변 태스크로 구성된 AI 워크로드를 처리하기 위한 새로운 자원 관리 정책이 요구되고 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 고정 태스크와 가변 태스크로 구성된 태스크 조합을 AI 워크로드의 특성을 고려하여 스케줄링 하는 자원 할당 방법 및 장치를 제공하고자 한다.

[0006] 다만, 기술적 과제는 상술한 기술적 과제로 한정되는 것은 아니며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0007] 본 발명의 일실시예에 따른 자원 할당 방법은 고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성

된 적어도 하나 이상의 태스크 조합을 생성하는 단계; 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원의 할당과 관련된 자원 계획을 결정하는 단계; 상기 고정 태스크로 구성된 제1 워크로드(Workload)를 처리하기 위한 자원 할당을 수행하는 단계; 및 상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 적어도 하나 이상의 태스크 조합에 대해 결정된 자원 계획을 이용함으로써, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당을 재수행하는 단계를 포함할 수 있다.

[0008] 상기 자원 계획을 결정하는 단계는 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원을 할당하기 위한 복수의 솔루션들로 구성된 솔루션 집합을 생성하는 단계; 상기 생성된 솔루션 집합에서 효율 값에 따라 2 개의 부모 솔루션을 선택하는 단계; 상기 선택된 2개의 부모 솔루션에 미리 설정된 연산을 수행함으로써 자식 솔루션을 생성하는 단계; 및 상기 생성된 자식 솔루션을 이용하여 상기 솔루션 집합을 재구성하는 단계를 포함할 수 있다.

[0009] 상기 복수의 솔루션들 각각은 상기 태스크 조합에 포함된 각각의 태스크들을 처리하기 위한 CPU(central processing unit) 전압/주파수 모드 및 메모리 종류에 대한 정보를 포함할 수 있다.

[0010] 상기 부모 솔루션을 선택하는 단계는 상기 복수의 솔루션들 각각의 효율 값에 따라 확률적으로 수행될 수 있다.

[0011] 상기 자식 솔루션을 생성하는 단계는 상기 선택된 2개의 부모 솔루션에 교차 연산 및 돌연변이 연산을 수행하여 자식 솔루션을 생성할 수 있다.

[0012] 상기 솔루션 집합을 재구성하는 단계는 상기 솔루션 집합을 구성하는 복수의 솔루션들 중 효율 값이 가장 작은 솔루션을 상기 생성된 자식 솔루션으로 대체할 수 있다.

[0013] 상기 자원 계획을 결정하는 단계는 상기 재구성된 솔루션 집합을 구성하는 솔루션들의 효율 값이 미리 설정된 차이 이하가 되도록 반복하여 상기 솔루션 집합을 재구성할 수 있다.

[0014] 상기 자원 할당을 재수행하는 단계는 상기 제2 워크로드의 제1 패턴과 상기 적어도 하나 이상의 태스크 조합 각각에 대한 워크로드의 제2 패턴을 비교하는 단계; 및 비교 결과 상기 제1 패턴과 제2 패턴 사이의 차이가 가장 작은 태스크 조합에 결정된 자원 계획에 따라 상기 자원 할당을 재수행하는 단계를 포함할 수 있다.

[0015] 본 발명의 일실시예에 따른 자원 할당 방법은 고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 단계; 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원을 할당하기 위한 복수의 솔루션들로 구성된 솔루션 집합을 생성하는 단계; 상기 생성된 솔루션 집합에서 상기 복수의 솔루션들 각각의 효율 값에 따라 확률적으로 2 개의 부모 솔루션을 선택하는 단계; 상기 선택된 2개의 부모 솔루션에 교차 연산 및 돌연변이 연산을 수행함으로써 자식 솔루션을 생성하는 단계; 및 상기 솔루션 집합을 구성하는 복수의 솔루션들 중 효율 값이 가장 작은 솔루션을 상기 생성된 자식 솔루션으로 대체함으로써 상기 솔루션 집합을 재구성하는 단계를 포함하고, 상기 솔루션 집합은 상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당에 이용될 수 있다.

[0016] 상기 솔루션 집합은 상기 복수의 솔루션들의 효율 값이 미리 설정된 차이 이하가 되도록 반복하여 재구성될 수 있다.

[0017] 상기 복수의 솔루션들 각각은 상기 태스크 조합에 포함된 각각의 태스크들을 처리하기 위한 CPU(central processing unit) 전압/주파수 모드 및 메모리 종류에 대한 정보를 포함할 수 있다.

[0018] 본 발명의 일실시예에 따른 컴퓨팅 장치는 하나 이상의 프로세서; 및 상기 프로세서에 의해 실행되는 프로그램을 로드(load)하거나 저장하는 메모리를 포함하되, 상기 프로그램은 고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 동작, 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원의 할당과 관련된 자원 계획을 결정하는 동작, 상기 고정 태스크로 구성된 제1 워크로드(Workload)를 처리하기 위한 자원 할당을 수행하는 동작 및 상기 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 상기 가변 태스크가 활성화된 경우, 상기 적어도 하나 이상의 태스크 조합에 대해 결정된 자원 계획을 이용함으로써, 상기 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당을 재수행하는 동작을 수행하도록 하는 인스트럭션들을 포함할 수 있다.

[0019] 상기 프로세서는 상기 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원을 할당하기 위한 복수의 솔루션들로 구성된 솔루션 집합을 생성하고, 상기 생성된 솔루션 집합에서 효율 값에 따라 2

개의 부모 솔루션을 선택하며, 상기 선택된 2개의 부모 솔루션에 미리 설정된 연산을 수행함으로써 자식 솔루션을 생성하고, 상기 생성된 자식 솔루션을 이용하여 상기 솔루션 집합을 재구성함으로써 상기 자원 계획을 설정할 수 있다.

[0020] 상기 복수의 솔루션들 각각은 상기 태스크 조합에 포함된 각각의 태스크들을 처리하기 위한 CPU(central processing unit) 전압/주파수 모드 및 메모리 종류에 대한 정보를 포함할 수 있다.

[0021] 상기 프로세서는 상기 복수의 솔루션들 각각의 효용 값에 따라 확률적으로 부모 솔루션을 선택할 수 있다.

[0022] 상기 프로세서는 상기 선택된 2개의 부모 솔루션에 교차 연산 및 돌연변이 연산을 수행하여 자식 솔루션을 생성할 수 있다.

[0023] 상기 프로세서는 상기 솔루션 집합을 구성하는 복수의 솔루션들 중 효용 값이 가장 작은 솔루션을 상기 생성된 자식 솔루션으로 대체함으로써 상기 솔루션 집합을 재구성할 수 있다.

[0024] 상기 프로세서는 상기 재구성된 솔루션 집합을 구성하는 솔루션들의 효용 값이 미리 설정된 차이 이하가 되도록 반복하여 상기 솔루션 집합을 재구성함으로써 상기 자원 계획을 결정할 수 있다.

[0025] 상기 프로세서는 상기 제2 워크로드의 제1 패턴과 상기 적어도 하나 이상의 태스크 조합 각각에 대한 워크로드의 제2 패턴을 비교하고, 비교 결과 상기 제1 패턴과 제2 패턴 사이의 차이가 가장 작은 태스크 조합에 결정된 자원 계획에 따라 상기 자원 할당을 재수행할 수 있다.

발명의 효과

[0026] 본 발명의 일실시예에 의하면, 고정 태스크와 가변 태스크로 구성된 태스크 조합을 AI 워크로드의 특성을 고려하여 스케줄링 할 수 있다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 일실시예에 따른 컴퓨팅 장치의 구성도를 도시한 도면이다.

도 2는 본 발명의 일실시예에 따른 자원 할당 방법을 간략하게 나타낸 도면이다.

도 3은 본 발명의 일실시예에 따른 적어도 하나 이상의 태스크 조합 각각에 대한 자원 계획 결정 방법을 나타낸 도면이다.

도 4는 본 발명의 일실시예에 따른 유전 알고리즘을 통한 반복 횟수에 따라 솔루션 집합의 효용 값이 변화하는 양상을 나타낸 도면이다.

도 5a 내지 도 5c는 본 발명의 일실시예에 따른 서로 다른 자원 계획들의 시뮬레이션 결과를 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0028] 실시예들에 대한 특정한 구조적 또는 기능적 설명들은 단지 예시를 위한 목적으로 개시된 것으로서, 다양한 형태로 변경되어 구현될 수 있다. 따라서, 실제 구현되는 형태는 개시된 특정 실시예에만 한정되는 것이 아니며, 본 명세서의 범위는 실시예들로 설명한 기술적 사상에 포함되는 변경, 균등물, 또는 대체물을 포함한다.

[0029] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 해석되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.

[0030] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다.

[0031] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 문서에서, "A 또는 B", "A 및 B 중 적어도 하나", "A 또는 B 중 적어도 하나", "A, B 또는 C", "A, B 및 C 중 적어도 하나", 및 "A, B, 또는 C 중 적어도 하나"와 같은 문구들 각각은 그 문구들 중 해당하는 문구에 함께 나열된 항목들 중 어느 하나, 또는 그들의 모든 가능한 조합을 포함할 수 있다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 설명된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는

부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

- [0032] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 해당 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0033] 이하, 실시예들을 첨부된 도면들을 참조하여 상세하게 설명한다. 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고, 이에 대한 중복되는 설명은 생략하기로 한다.
- [0035] 도 1은 본 발명의 일실시예에 따른 컴퓨팅 장치의 구성도를 도시한 도면이다.
- [0036] 도 1에 도시된 바와 같이, 컴퓨팅 장치(100)는 하나 이상의 프로세서(110), 프로세서(110)에 의하여 수행되는 프로그램(130)을 로드(load)하거나 저장하는 메모리(120)를 포함할 수 있다. 도 1의 컴퓨팅 장치(100)에 포함된 구성 요소는 일례에 불과하고, 본 발명이 속한 기술분야의 통상의 기술자라면 도 1에 도시된 구성 요소들 외에 다른 범용적인 구성 요소들이 더 포함될 수 있음을 알 수 있다.
- [0037] 프로세서(110)는 컴퓨팅 장치(100)의 각 구성의 전반적인 동작을 제어한다. 프로세서(110)는 CPU(Central Processing Unit), MPU(Micro Processor Unit), MCU(Micro Controller Unit), GPU(Graphic Processing Unit), NPU(Neural Processing Unit), DSP(Digital Signal Processor) 또는 본 발명의 기술 분야에 잘 알려진 임의의 형태의 프로세서 중 적어도 하나를 포함하여 구성될 수 있다. 또한, 프로세서(110)는 본 발명의 다양한 실시예들에 따른 방법/동작을 실행하기 위한 적어도 하나의 애플리케이션 또는 프로그램에 대한 연산을 수행할 수 있다. 컴퓨팅 장치(100)는 하나 이상의 프로세서를 구비할 수 있다.
- [0038] 메모리(120)는 컴퓨팅 장치(100)에 포함된 구성요소(예를 들어, 프로세서(110))에 의해 사용되는 다양한 데이터, 명령 및 정보 중 하나 또는 둘 이상의 조합을 저장한다. 메모리(120)는 휘발성 메모리 및/또는 비휘발성 메모리를 포함할 수 있다.
- [0039] 프로그램(130)은 본 발명의 다양한 실시예들에 따른 방법/동작들이 구현된 하나 이상의 동작(action)들을 포함할 수 있으며, 메모리(120)에 소프트웨어 형태로 저장될 수 있다. 여기서, 동작은 프로그램(130)에서 실현되는 명령어에 대응한다. 예를 들어, 프로그램(130)은 고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 적어도 하나 이상의 태스크 조합을 생성하는 동작, 생성된 적어도 하나 이상의 태스크 조합 각각에 대해 컴퓨팅 자원 및 메모리 자원의 할당과 관련된 자원 계획을 설정하는 동작, 고정 태스크로 구성된 제1 워크로드(Workload)를 처리하기 위한 자원 할당을 수행하는 동작 및 고정 태스크로 구성된 제1 워크로드가 처리되는 과정에서 가변 태스크가 활성화된 경우, 적어도 하나 이상의 태스크 조합에 대해 설정된 자원 계획을 이용함으로써, 활성화된 가변 태스크가 추가적으로 포함된 제2 워크로드를 처리하기 위한 자원 할당을 재수행하는 동작을 수행하도록 하는 인스트럭션들을 포함할 수 있다.
- [0040] 프로그램(130)이 메모리(120)에 로드 되면, 프로세서(110)는 프로그램(130)을 구현하기 위한 복수의 동작들을 실행시킴으로써 본 발명의 다양한 실시예들에 따른 방법/동작들을 수행할 수 있다.
- [0041] 프로그램(130)의 실행 화면은 디스플레이(140)를 통해 표시될 수 있다. 도 1의 경우, 디스플레이(140)는 컴퓨팅 장치(100)와 연결되는 별도의 장치로 표현되나, 스마트폰, 태블릿 등 사용자가 휴대할 수 있는 단말기와 같은 컴퓨팅 장치(100)의 경우 디스플레이(140)가 컴퓨팅 장치(100)의 구성 요소로 될 수 있다. 디스플레이(140)에 표현되는 화면은 프로그램에 정보를 입력하기 전이나 프로그램의 실행 결과일 수 있다.
- [0043] 도 2는 본 발명의 일실시예에 따른 자원 할당 방법을 간략하게 나타낸 도면이다.
- [0044] 도 2에 도시된 자원 할당 방법은 도 1에 제시된 컴퓨팅 장치(100)의 프로세서(110)에 의해 수행된다. 프로세서(110)는 실시간 시스템에서 새롭게 등장하는 AI 워크로드의 특성을 고려하여 고정 태스크(Fixed task) 및 가변 태스크(Variable task)로 구성된 태스크 조합을 스케줄링 하는 새로운 자원 할당 방법을 제공할 수 있다.
- [0045] 고정 태스크는 시스템 상황과 관계없이 항상 존재하는 기본 작업이고, 가변 태스크는 시스템이 특정 상황(예를 들어, 자율 주행에서 사고로 인한 교통 체증이 발생한 경우)이나 구성 변경(예를 들어, 스마트 팩토리에서 생산

라인을 재구성하는 경우)을 처리할 때 일시적으로 추가될 수 있는 작업을 의미할 수 있다.

[0046] 본 발명에서 제안된 워크로드 모델에서 고정 태스크 세트는 $W = \{w_1, w_2, \dots, w_n\}$ 로 정의되고, 각각의 고정 태스크 w_i 는 $\langle WCET_{w_i}, Period_{w_i}, Size_{w_i} \rangle$ 의 특징을 포함할 수 있다. 여기서, $WCET_{w_i}$ 는 고정 태스크 w_i 의 최악의 실행 시간, $Period_{w_i}$ 는 고정 태스크 w_i 의 주기, $Size_{w_i}$ 는 고정 태스크 w_i 의 메모리 크기를 나타낼 수 있다.

[0047] 한편, 고정 태스크와 함께 가변 태스크를 수용하기 위해, 프로세서(110)는 가변 태스크 세트를 $V = \{v_1, v_2, \dots, v_m\}$ 로 정의할 수 있다. 이때, 각각의 가변 태스크 v_i 는 고정 태스크와 유사하게 $\langle WCET_{v_i}, Period_{v_i}, Size_{v_i} \rangle$ 의 특징을 포함할 수 있다. 여기서, $WCET_{v_i}$ 는 가변 태스크 v_i 의 최악의 실행 시간, $Period_{v_i}$ 는 가변 태스크 v_i 의 주기, $Size_{v_i}$ 는 가변 태스크 v_i 의 메모리 크기를 나타낼 수 있다.

[0048] 프로세서(110)는 AI 워크로드의 변화에 적응하여 실시간 시스템의 에너지 소비를 최소화하고, 처리되는 태스크의 마감 시간을 보장하도록 하는 자원 할당을 목표로 할 수 있다.

[0049] 이를 위해 프로세서(110)는 실시간 시스템을 구성하는 컴퓨팅 자원과 메모리 자원을 함께 고려하여 자원 할당을 수행할 수 있다. 일례로, 프로세서(110)는 CPU(central processing unit) 전압/주파수 스케일링을 통해 CPU에 공급되는 전압을 제어함으로써 컴퓨팅 자원의 에너지를 절약할 수 있으며, 컴퓨팅 자원의 처리 성능과의 사이에서 유연성을 제공할 수 있다.

[0050] 다른 일례로, 프로세서(110)는 비휘발성 메모리(Non-Volatile Memory, NVM) 또는 영구 메모리(Persistent Memory, PMem)와 같은 저전력 메모리를 DRAM(Dynamic Random Access Memory)과 같은 휘발성 메모리와 함께 사용하여 자원 할당을 수행할 수 있다. 이때, 프로세서(110)는 메모리 할당 기법을 통해 휘발성 메모리가 아닌 저전력 메모리에 태스크를 할당함으로써 메모리 자원의 에너지를 절약할 수 있다.

[0051] 이와 같은 CPU 전압/주파수 스케일링 및 메모리 할당 기법을 본 발명의 자원 할당 방법에 적용하기 위하여 프로세서(110)는 먼저 고정 태스크 세트에 대해 아래의 식 1과 같은 타당성 테스트를 수행할 수 있다. 이때, 아래의 식 1은 두 기법을 사용한 후 최악의 실행 시간이 아래 부등식을 만족해야 함을 의미할 수 있다.

[0052] <식 1>

$$\sum_{i=1}^n \frac{Resource_{plan}(WCET_{w_i})}{Period_{w_i}} \leq 1$$

[0054] 여기서, 자원 계획 $Resource_{plan}(WCET_{w_i})$ 는 CPU 전압/주파수 스케일링 및 메모리 할당 기법을 활용한 후의 고정 태스크 w_i 에 대한 최악의 실행 시간을 의미할 수 있다.

[0055] 상기의 식 1이 만족될 경우, 프로세서(110)는 EDF(Earliest Deadline First) 스케줄링을 통해 주어진 고정 태스크 세트에 대한 실행 가능한 스케줄을 찾아 자원 할당을 수행할 수 있다.

[0056] 이후 가변 태스크 세트가 추가되어 AI 워크로드가 변경된 경우, 프로세서(110)는 아래의 식 2와 같은 타당성 테스트를 추가적으로 수행할 수 있다.

[0057] <식 2>

$$\sum_{i=1}^n \frac{Resource_{plan}(WCET_{w_i})}{Period_{w_i}} + \sum_{j=1}^m \frac{Resource_{plan}(WCET_{v_j})}{Period_{v_j}} \leq 1$$

[0059] 프로세서(110)는 상기의 식 1 및 식 2에 따른 제약 조건에서 고정 태스크 및 가변 태스크로 구성된 적어도 하나 이상의 모든 태스크 조합 각각에 대해 CPU 전압/주파수 모드 및 메모리 할당을 결정할 수 있다.

[0060] 이를 위해 프로세서(110)는 고정 태스크 및 가변 태스크로 구성된 적어도 하나 이상의 태스크 조합을 생성(210)하고, 생성된 모든 태스크 조합 각각에 대해 CPU 및 메모리의 에너지 소비를 최적화하는 것을 목표로 하는 유전 알고리즘(Genetic Algorithm, GA)을 활용함으로써 CPU 전압/주파수 모드 및 메모리 할당에 대한 최적의 솔루션을 도출(220)할 수 있다.

[0061] 고정 태스크로 구성된 워크로드가 처리되는 과정에서 모니터링을 통해 가변 태스크가 활성화된 것이 식별(230)된 경우, 프로세서(110)는 모든 태스크 조합 각각에 도출된 최적의 솔루션을 통해 가변 태스크가 포함된 워크로드를 처리하기 위한 자원 할당을 재수행(240)할 수 있다.

[0062] 한편, 프로세서(110)를 통해 생성되는 태스크 조합의 구성은 실시간 시스템에서 발생하는 실제 태스크들의 구성과 비교하여 차이가 존재할 수 있다. 따라서, 프로세서(110)는 실시간 시스템에서 실제 태스크들이 발생되었을 경우, 모든 태스크 조합 각각에 대해 도출된 솔루션들 중 어느 태스크 조합의 솔루션을 이용하여 자원 할당을 할 것인지를 결정해야 한다.

[0063] 이때, 프로세서(110)는 모니터링을 통해 식별된 가변 태스크가 포함된 워크로드의 제1 패턴과 모든 태스크 조합 각각에 대한 워크로드의 제2 패턴을 비교하여, 제1 패턴과 제2 패턴 사이의 차이가 가장 작은 태스크 조합에 대해 도출된 솔루션을 이용하여 자원 할당을 재수행할 수 있다.

[0065] 도 3은 본 발명의 일실시예에 따른 적어도 하나 이상의 태스크 조합 각각에 대한 자원 계획 결정 방법을 나타낸 도면이다.

[0066] 위에서 언급한 바와 같이 프로세서(110)는 고정 태스크 및 가변 태스크로 구성된 적어도 하나 이상의 모든 태스크 조합 각각에 대해 유전 알고리즘을 활용함으로써 CPU 전압/주파수 모드 및 메모리 할당에 대한 최적의 솔루션을 도출할 수 있다.

[0067] 유전 알고리즘은 제한된 시간 예산으로 넓은 검색 공간을 탐색하여 집단 유전학과 유사한 고품질 솔루션을 찾을 수 있다. 본 발명에서 검색 공간은 CPU 전압/주파수 모드와 워크로드 내 모든 태스크의 메모리 할당이다.

[0068] 본 발명에서 제공하는 유전 알고리즘의 목표는 마감 시간 제약 조건에서 CPU와 메모리의 에너지를 최대한 절약하는 솔루션을 찾는 것이며, 본 발명의 프로세서(110)는 이를 위해 충분히 높은 품질의 솔루션을 찾을 때까지 고정된 수의 솔루션을 반복적으로 진화시킬 수 있다.

[0069] 단계(S310)에서 프로세서(110)는, 모든 태스크 조합 각각에 대해 무작위로 미리 설정된 개수의 솔루션들을 생성하여 솔루션 집합을 도출할 수 있다. 이때, 솔루션 집합에 포함된 솔루션들 각각은 주어진 태스크 조합에 대해서로 다른 CPU 및 메모리 자원 계획을 나타낼 수 있다.

[0070] 프로세서(110)는 두 개의 선형 문자열을 사용하여 각 솔루션을 표현할 수 있다. 첫 번째 선형 문자열은 주어진 태스크 조합 내 각 태스크를 실행하기 위한 CPU 전압/주파수 모드를 나타내고, 두 번째 선형 문자열은 메모리 할당을 나타낼 수 있다. 각각의 선형 문자열은 n + m 개의 항목이 있으며, 여기서 n과 m은 각각 주어진 태스크 조합 내에 포함된 고정 태스크 및 가변 태스크의 개수이다.

[0071] 이때, 첫 번째 선형 문자열의 항목은 0에서 p-1 사이의 값을 가질 수 있으며, 여기서 p는 설정 가능한 CPU 전압/주파수 모드의 개수를 나타낸다. 마찬가지로, 두 번째 선형 문자열의 항목은 본 발명이 두 가지 유형의 메모리(예를 들어, 저전력 메모리와 DRAM)를 사용하므로 0 또는 1의 값을 가질 수 있다.

[0072] 단계(S320)에서, 프로세서(110)는 생성된 솔루션 집합에서 2개의 부모 솔루션을 선택할 수 있다. 본 발명의 유전 알고리즘은 각 솔루션의 우수성을 측정하기 위해 효용 함수를 사용할 수 있다. 이때, 각 솔루션은 CPU 전압/주파수 모드 및 메모리 할당을 나타내므로, 프로세서(110)는 각 솔루션의 자원 계획에 따라 워크로드가 실행될

때 발생하는 CPU와 메모리의 에너지 절감 효과를 기준으로 솔루션¹의 효용 함수 $Utility(i)$ 를 측정할 수 있다.

[0073] 또한, 효용 함수 $Utility(i)$ 의 최적화 문제는 마감 시간 제약이 있기 때문에 각 솔루션의 자원 계획이 마감 시간을 지키지 못할 경우, 아래의 식 3과 같이 일정한 패널티 값이 주어질 수 있다.

[0074] <식 3>

[0075] $Utility(i) = Energy_Saving(i) - Penalty(i)$

[0076] 여기서, $Energy_Saving(i)$ 는 기본 CPU 전압/주파수 및 DRAM 전용 메모리를 사용하는 경우와 비교하여 솔루션 i의 자원 계획에 의해 절약되는 에너지를 나타내고, $Penalty(i)$ 는 가용한 자원 용량의 한계를 초과하는 태스크 부하를 의미한다.

[0077] 보다 구체적으로 본 발명의 프로세서(110)는 솔루션 집합에 포함된 복수의 솔루션들 각각의 효용 가치에 따라 확률적으로 부모 솔루션을 선택할 수 있다. 즉, 프로세서(110)는 효용 함수 $Utility(i)$ 를 통해 측정된 효용 값을 이용하여 복수의 솔루션들 각각에 대해 순위에 선형적으로 비례하는 선택 확률을 부여할 수 있고, 부여된 선택 확률에 따라 2개의 부모 솔루션을 선택할 수 있다.

[0078] 일례로, 프로세서(110)는 유전 알고리즘의 표준 정규화 방식에 따라 제1 순위의 솔루션에 대해 제100 순위의 솔루션 보다 4배 높은 선택 확률을 부여할 수 있다. 다만, 이러한 선택 확률의 부여 방법은 하나의 예시일 뿐 상기의 예에 한정되지 않는다.

[0079] 단계(S330)에서, 프로세서(110)는 선택된 2개의 부모 솔루션에 미리 설정된 연산을 수행함으로써 자식 솔루션을 생성할 수 있다. 일례로, 프로세서(110)는 선택된 2개의 부모 솔루션에 크로스오버 연산과 돌연변이 연산을 수행하여 자식 솔루션을 생성할 수 있다. 다만, 자식 솔루션을 생성하기 위해 이용되는 연산의 종류는 하나의 예시일 뿐 상기의 예에 한정되지 않는다.

[0080] 먼저, 크로스오버 연산은 2개의 부모 솔루션들 각각의 특성을 부분적으로 상속하여 두 부모 솔루션을 결합하는 연산이다. 일례로, 1점 크로스오버 연산을 활용하는 경우, 프로세서(110)는 선형 문자열 내에서 크로스오버 지점을 무작위로 선택하고, 제1 부모 솔루션에서 크로스오버 지점의 왼쪽 부분 문자열을, 제2 부모 솔루션에서 크로스오버 지점의 오른쪽 부분 문자열을 각각 복사하여 새로운 자식 솔루션을 생성할 수 있다.

[0081] 이때, 생성된 자식 솔루션은 돌연변이 연산자에 의해 무작위로 교란될 수 있는데, 일례로, 특정 확률로 자식 솔루션의 문자열 일부 항목이 변경될 수 있다. 이와 같이 돌연변이가 연산자를 통해 현재 솔루션 집합에 없는 새로운 솔루션이 생성되어 주입됨으로써 솔루션 집합의 수렴이 지연될 수 있다. 그러나 이러한 과정을 통해 유전 알고리즘이 전체 문제 공간을 보다 균등하게 탐색할 수 있으므로, 솔루션 집합이 일부 제한된 공간만 탐색하는 것을 방지할 수 있다.

[0082] 단계(S340)에서, 프로세서(110)는 생성된 자식 솔루션을 이용하여 솔루션 집합을 재구성할 수 있다. 보다 구체적으로 프로세서(110)는 솔루션 집합 내에서 희생 솔루션을 선택하고, 선택된 희생 솔루션을 새로 생성된 자식 솔루션으로 대체함으로써 새로운 솔루션 집합을 생성할 수 있다. 일례로, 프로세서(110)는 효용 값이 가장 작은 솔루션을 희생 솔루션으로 선택할 수 있으나, 이러한 희생 솔루션 선택 방법은 하나의 예시일 뿐 상기의 예에 한정되지 않는다.

[0083] 한편, 유전 알고리즘은 최종 솔루션을 찾기 위해 반복하여 솔루션 집합을 재구성할 수 있다. 본 발명의 프로세서(110)는 일정한 반복 횟수에 따라 솔루션 집합을 재구성하는 대신 솔루션 집합 내 솔루션들의 효용 값을 확인하여 솔루션 집합이 수렴하는지 여부를 모니터링 할 수 있다. 즉, 프로세서(110)는 솔루션 집합 내 모든 솔루션들의 효용 값이 미리 설정된 차이 이하가 되도록 반복하여 솔루션 집합을 재구성할 수 있다.

[0084] 도 4는 본 발명의 일실시예에 따른 유전 알고리즘을 통한 반복 횟수에 따라 솔루션 집합의 효용 값이 변화하는 양상을 나타낸 도면이다. 도 4는 시간에 따른 솔루션 집합 내 제1 순위(best) 솔루션의 효용 값, 제100 순위(worst) 솔루션의 효용 값 및 모든 솔루션들의 평균(average) 효용 값 변화를 보여준다. 도 4를 참고하면, 반복 횟수가 증가함에 따라 솔루션 집합의 효용 가치가 향상되고, 모든 솔루션들 효용 값이 최종적으로 수렴하는 것을 확인할 수 있다.

[0086] 도 5a 내지 도 5c는 본 발명의 일실시예에 따른 서로 다른 자원 계획들의 시뮬레이션 결과를 나타낸 도면이다.

[0087] 프로세서(110)는 아래의 표 1에 제시된 4가지 워크로드 시나리오를 서로 다른 자원 계획들로 시뮬레이션 하였다.

[0088] <표 1>

Scenario	The sequence of workload (CPU load)	Execution time of workloads
1	Fixed (0.2)-variable1 (0.2)-variable2 (0.6)	50%-20%-30%
2	Fixed (0.3)-variable1 (0.5)	30%-70%
3	Fixed (0.1)-variable1 (0.4)-variable2 (0.7)	25%-50%-25%
4	Fixed (0.4)-variable1 (0.1)-variable2 (0.3)	30%-40%-30%

[0089]

[0090] 먼저, 서로 다른 자원 계획들 중 ARP(Adaptive Resource Planning)은 본 발명에서 제공하는 자원 계획으로 고정 태스크와 가변 태스크로 구성된 태스크 조합을 AI 워크로드의 특성을 고려하여 스케줄링 할 수 있다.

[0091] FRP(fixed task-based Resource Planning)는 유전 알고리즘과 DVFS(Dynamic Voltage Frequency Scaling)를 이용한 자원 최적화를 실행하지만, 워크로드가 실행되는 동안 자원 계획을 변경하지 않는다.

[0092] MRP(maximal task-based Resource Planning)는 고정 태스크와 가변 태스크를 최대한으로 실행하는 가정하에 유전 알고리즘과 DVFS를 이용한 자원 최적화를 실행하지만, FRP와 마찬가지로 워크로드를 실행하는 동안 자원 계획을 변경하지 않는다.

[0093] 표 1을 참고하면, 고정 태스크는 CPU에 0.1 에서 0.4까지 로드되며, 가변 태스크는 CPU에 0.1에서 0.7까지 로드 될 수 있다. 시나리오 1의 경우, 워크로드 총실행시간의 50%가 경과될 때까지 고정 태스크들만 실행되고 CPU 로드(load)는 0.2인 상태이다. 가변 태스크 1이 총실행시간의 그 다음 20%동안 CPU 로드 0.2를 차지하면서 활성화 된다. 이때 고정 태스크는 여전히 실행 중이며, CPU load는 이 시간 동안 0.4로 증가된다. 마지막으로 남은 워크로드 총 실행 시간의 30%동안 가변 태스크 2가 CPU 로드 0.6을 갖고 활성화된다. 총실행시간의 70%가 지날 때에 CPU 로드는 0.8이 된다.

[0094] 시나리오 2를 사용하여 서로 다른 자원 계획들의 시뮬레이션 결과를 비교한 설명은 아래와 같다. 워크로드의 총 실행시간의 30%까지는 고정 태스크만 실행되고, 이때 CPU 로드는 0.3이다. 총실행시간이 30%를 넘어서면 가변 태스크 1이 활성화되어 총 CPU 로드는 0.3에서 0.8로 변화된다.

[0095] 이때, ARP는 도 5a와 같이 워크로드의 변화에 대응하여 즉각적으로 가용 자원을 사용하였으며, 이로써 CPU 로드가 그에 따라 증가하여 에너지소비도 급증하는 것을 볼 수 있다.

[0096] 그에 반해, FRP는 워크로드 변화에 대응하는 자원 계획이 없으며, 도 5b와 같이 총실행시간의 30%가 경과한 후 가변 태스크 1이 활성화됨에 따라 마감 시간을 어기는 경우가 발생하였다.

[0097] MRP 역시 워크로드 변화에 대응하는 자원 계획이 없으며, 도 5c와 같이 최대 워크로드에 대한 자원 계획을 여전히 유지함으로써 상대적으로 많은 에너지를 소모한 것을 확인 할 수 있다.

[0099] 이상에서 설명된 실시예들은 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치, 방법 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 소프트웨어 애플리케이션

이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0100] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치로 저장될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0101] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 저장할 수 있으며 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0102] 위에서 설명한 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 또는 복수의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0103] 이상과 같이 실시예들이 비록 한정된 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 이를 기초로 다양한 기술적 수정 및 변형을 적용할 수 있다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

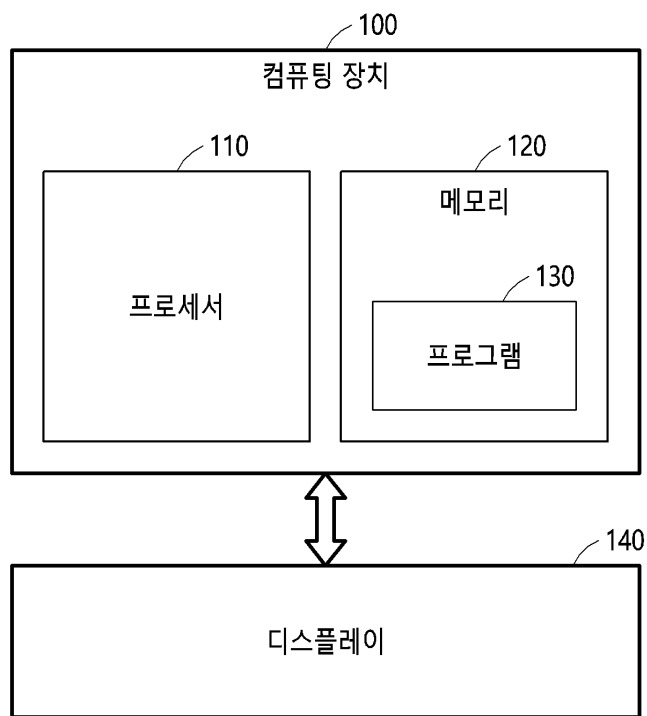
[0104] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

부호의 설명

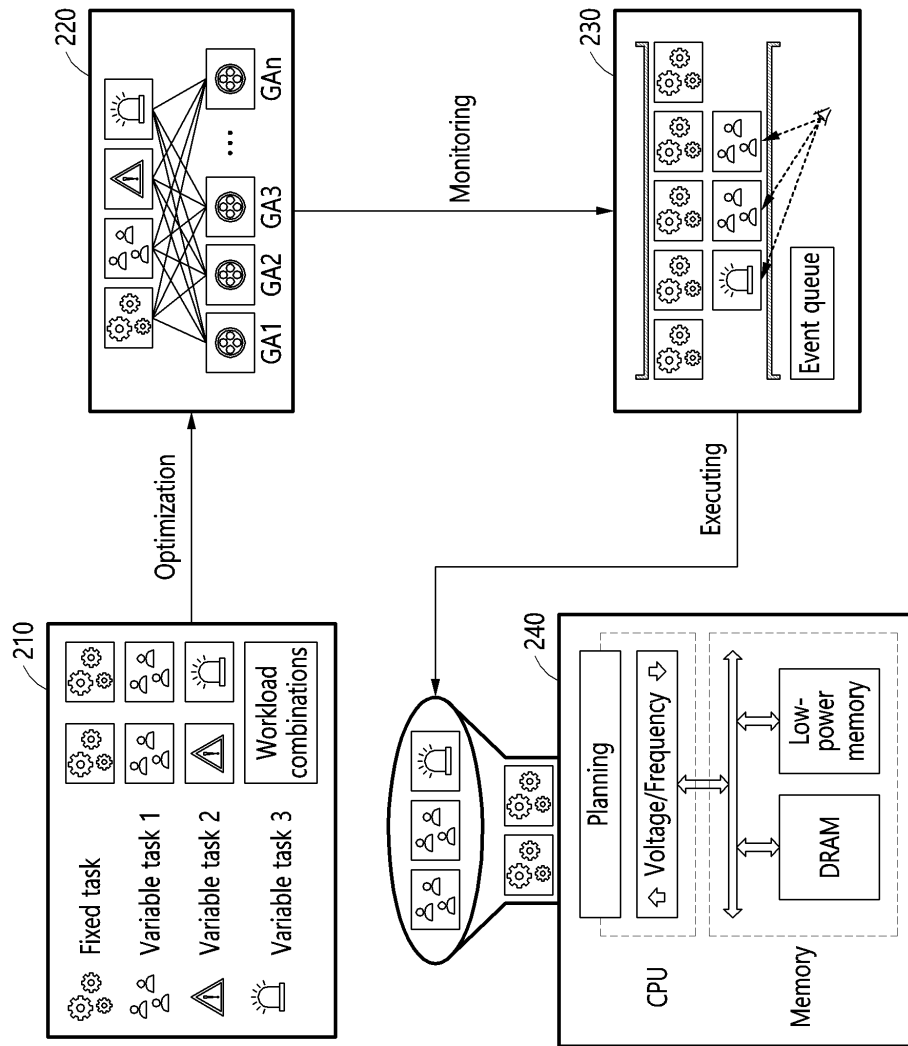
[0106] 100 : 컴퓨팅 장치
110 : 프로세서
120 : 메모리
130 : 프로그램
140 : 디스플레이

도면

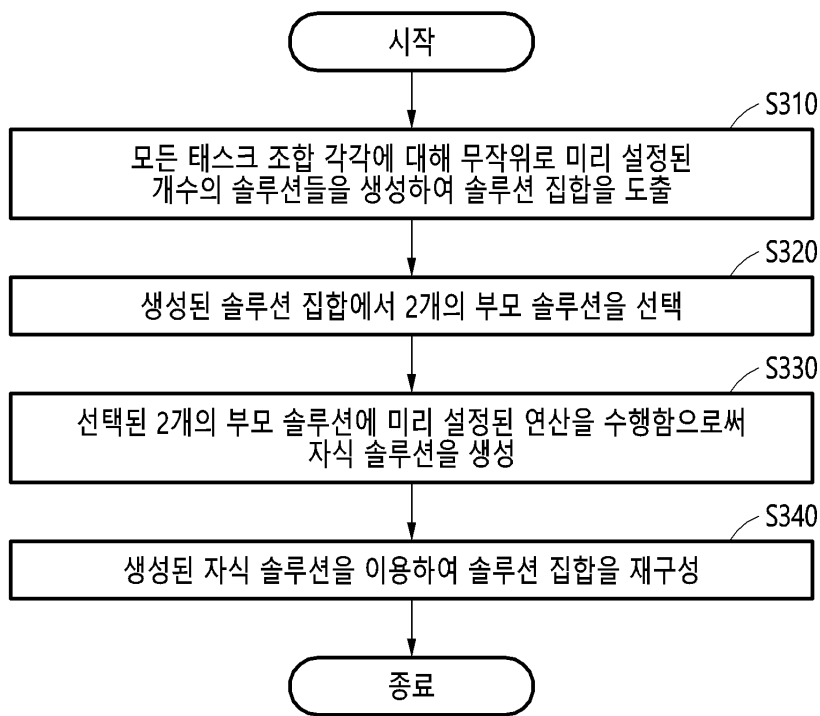
도면1



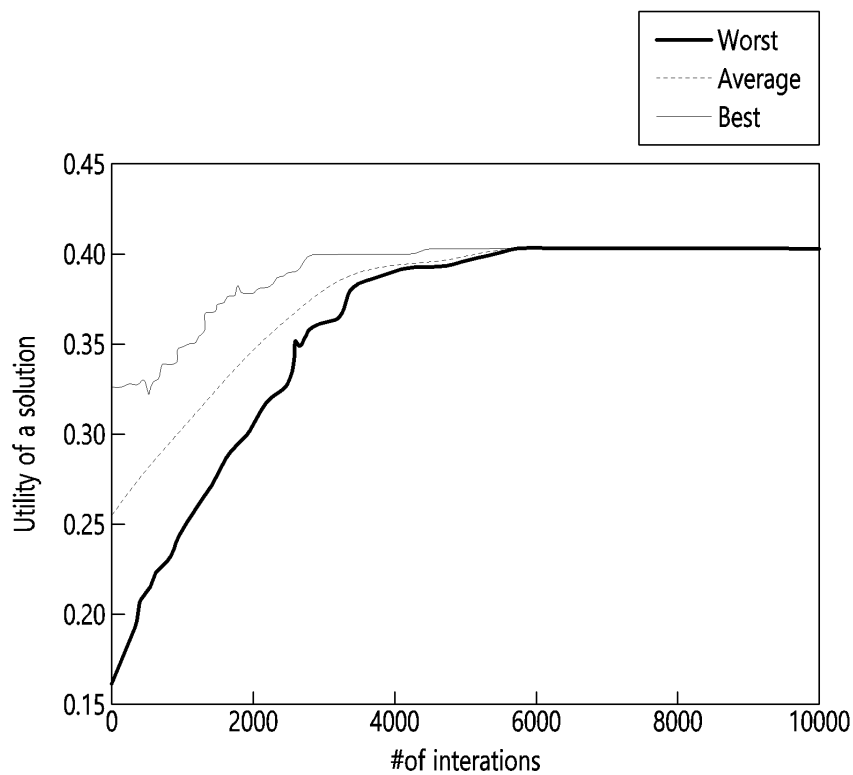
도면2



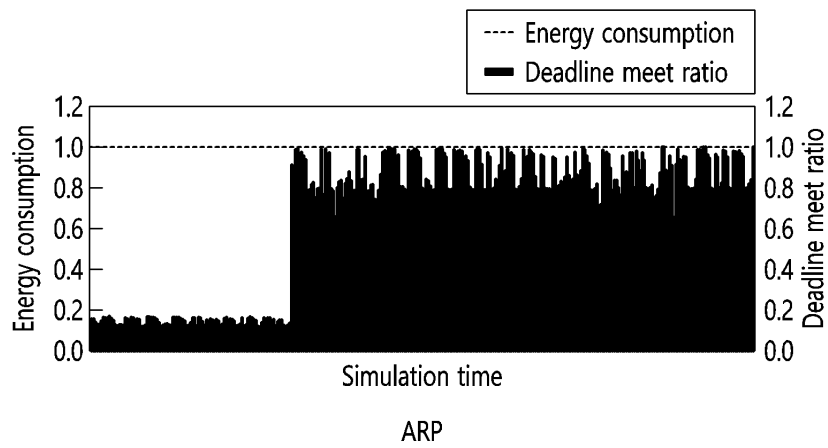
도면3



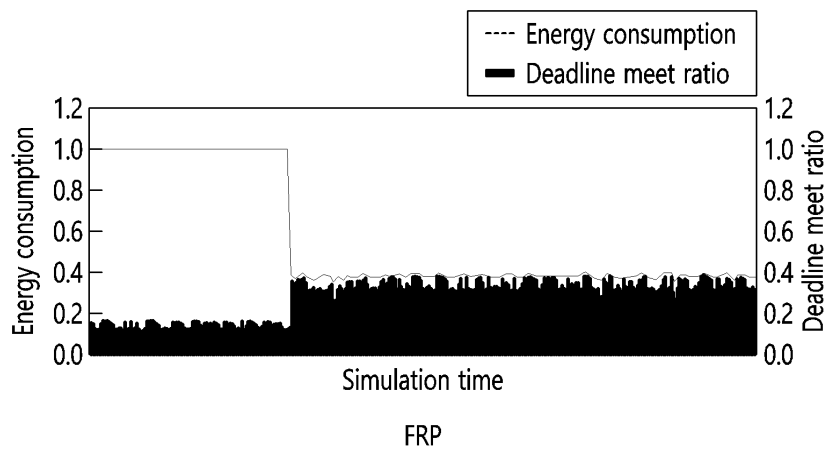
도면4



도면5a



도면5b



도면5c

