(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년04월28일
(11) 등록번호 10-2105681
(24) 등록일자 2020년04월22일

(51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01)
(52) CPC특허분류
G06F 9/5094 (2013.01)
G06F 2209/501 (2013.01)
(21) 출원번호 10-2018-0126690
(22) 출원일자 2018년10월23일
심사청구일자 2018년10월23일
(56) 선행기술조사문헌
US20160252946 A1
(뒷면에 계속)

(73) 특허권자
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
백웅기
울산광역시 울주군 언양읍 유니스트길 50 울산과학기술대학교
현지훈
울산광역시 울주군 언양읍 유니스트길 50 울산과학기술대학교
박진수
울산광역시 울주군 언양읍 유니스트길 50 울산과학기술대학교
(74) 대리인
제일특허법인(유)

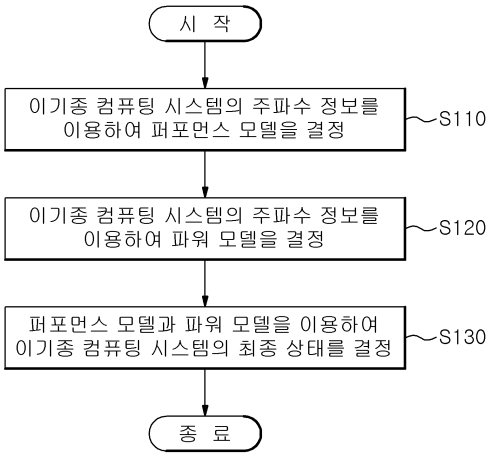
전체 청구항 수 : 총 15 항 심사관 : 유진태

(54) 발명의 명칭 **이기종 컴퓨팅 시스템의 제어 방법 및 장치**

(57) 요 약

본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템 제어 방법은, 상기 이기종 컴퓨팅 시스템의 주파수 정보를 이용하여, 상기 이기종 컴퓨팅 시스템의 작동 효율을 나타내는 퍼포먼스 모델과 상기 이기종 컴퓨팅 시스템의 에너지 소모량을 나타내는 파워 모델을 결정하는 단계와, 상기 퍼포먼스 모델과 상기 파워 모델을 이용하여 상기 이기종 컴퓨팅 시스템의 최종 상태를 결정하는 단계를 포함하고, 상기 이기종 컴퓨팅 시스템은 복수의 이기종 컴퓨팅 장치를 포함할 수 있다.

대 표 도 - 도3



(56) 선행기술조사문헌
US20170206113 A1
나갑주 외 2명. '임베디드 기기를 위한 이기종 멀티코어 프로세서 및 소프트웨어 기술 동향'. 전저통신동향분석, 제28권 제2호, 2013.4, pp.1-10.
WO2018031092 A1
WO2008073597 A1

이 발명을 지원한 국가연구개발사업
과제고유번호 1711075079
부처명 과학기술정보통신부
연구관리전문기관 한국연구재단
연구사업명 개인기초연구(과기정통부)
연구과제명 대규모 확장 가능한 고성능, 고보안, QoS 보장 빅데이터 처리를 위한 시스템 소프트웨어 기법 연구 및 개발
기 여 율 1/3
주관기관 울산과학기술원
연구기간 2018.03.01 ~ 2019.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711074286
부처명 과학기술정보통신부
연구관리전문기관 한국연구재단
연구사업명 차세대정보·컴퓨팅기술개발
연구과제명 Peta Flops 스케일의 기계학습 프레임워크 개발
기 여 율 1/3
주관기관 울산과학기술원
연구기간 2018.06.01 ~ 2019.03.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711073161
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 SW컴퓨팅산업원천기술개발(정보화)
연구과제명 (빅데이터-총괄) 빅데이터 처리 고도화 핵심 기술개발 사업 총괄 및 고성능 컴퓨팅 기술을 활용한 성능 가속화 기술 개발
기 여 율 1/3
주관기관 울산과학기술원
연구기간 2018.04.01 ~ 2019.01.31

명 세 서

청구범위

청구항 1

이기종 컴퓨팅 시스템을 제어하는 방법에 있어서,

상기 이기종 컴퓨팅 시스템의 주파수 정보를 이용하여, 상기 이기종 컴퓨팅 시스템의 작동 효율을 나타내는 퍼포먼스 모델과 상기 이기종 컴퓨팅 시스템의 에너지 소모량을 나타내는 파워 모델을 결정하는 단계와,

상기 퍼포먼스 모델을 통해 획득되는 퍼포먼스 값과 상기 파워 모델을 통해 획득되는 파워 값을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 상기 이기종 컴퓨팅 시스템의 최종 상태로 결정하는 단계를 포함하고,

상기 이기종 컴퓨팅 시스템은 복수의 이기종 컴퓨팅 장치를 포함하고,

상기 시스템 상태는 상기 복수의 이기종 컴퓨팅 장치 각각에 대한 주파수 값에 의해 결정되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 2

제1항에 있어서,

상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 및 메모리를 포함하고,

상기 주파수 정보는 상기 CPU, 상기 GPU, 및 상기 메모리 각각에 대한 주파수 값을 포함하는

이기종 컴퓨팅 시스템 제어 방법.

청구항 3

제1항에 있어서,

상기 퍼포먼스 모델은

$$\mathrm{Perf} = \alpha_C \cdot f_C + \alpha_G \cdot f_G + \alpha_M \cdot f_M + \alpha_{C,G} \cdot f_C \cdot f_G + \alpha_{G,M} \cdot f_G \cdot f_M + \alpha_{M,C} \cdot f_M \cdot f_C + \beta$$

(Perf는 상기 퍼포먼스 값, f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기 복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β는 각각 기 결정된 값)

로 구성되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 4

제3항에 있어서,

상기 α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β 각각은 상기 퍼포먼스 모델에 서로 다른 적어도 7개의 시스템 상태를 각각 대입하고, 상기 대입에 의해 획득되는 7개의 연립 방정식을 이용하여 결정된 값이며,

상기 시스템 상태는 상기 f_C, 상기 f_G, 상기 f_M 값에 따라 각각 결정되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 5

제4항에 있어서,

상기 α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β 각각은

상기 퍼포먼스 모델에 대입된 상기 시스템 상태 중 적어도 하나가 변경되면, 상기 적어도 하나의 변경된 시스템 상태가 상기 퍼포먼스 모델에 재대입됨에 기초하여 기 획득된 연립 방정식이 갱신되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 6

제1항에 있어서,

상기 파워 모델은

$$P_{D,f_D} = \epsilon_{D,f_D} \cdot U_D + \zeta_{D,f_D}$$

(P_{D,f_D}는 주파수 값으로 f_D를 가지는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 파워 값, ϵ_{D,f_D}과 ζ_{D,f_D} 각각은 기 지정된 값, U_D는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 사용률(utilization)을 나타내는 사용률 모델)

로 구성되고,

상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 또는 메모리를 포함하고,

상기 사용률은 상기 복수의 이기종 컴퓨팅 장치 중 하나가 동작할 수 있는 전체 동작 시간에 대한 실제 동작하는 시간의 비율인

이기종 컴퓨팅 시스템 제어 방법.

청구항 7

제6항에 있어서,

상기 U_D는 상기 복수의 이기종 컴퓨팅 장치 각각에 대해

$$U_D = \gamma_{D,C} \cdot f_C + \gamma_{D,G} \cdot f_G + \gamma_{D,M} \cdot f_M + \gamma_{D,CG} \cdot f_C \cdot f_G + \gamma_{D,GM} \cdot f_G \cdot f_M + \gamma_{D,MC} \cdot f_M \cdot f_C + \delta_D$$

(f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기 복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D는 각각 기 결정된 값)

를 이용하여 획득되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 8

제7항에 있어서,

상기 $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D 각각은 상기 사용률 모델에 서로 다른 적어도 7개의 시스템 상태에 따른 사용률 값을 각각 대입하고, 상기 대입에 의해 획득되는 7개의 연립 방정식을 이용하여 결정된 값이며,

상기 시스템 상태는 상기 f_C, 상기 f_G, 상기 f_M 각각의 값에 따라 결정되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 9

제8항에 있어서,

상기 $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D 각각은

상기 파워 모델에 대입된 상기 시스템 상태 중 적어도 하나가 변경되면, 상기 적어도 하나의 변경된 시스템 상태가 상기 파워 모델에 재대입됨에 기초하여 기 획득된 연립 방정식이 갱신되는

이기종 컴퓨팅 시스템 제어 방법.

청구항 10

삭제

청구항 11

제1항에 있어서,

상기 최종 상태를 결정하는 단계는

상기 기 지정된 조건이 변경되는지 여부를 식별하는 단계와,

상기 기 지정된 조건이 변경되는 것에 대응하여 상기 시스템 상태를 재결정하는 단계를 포함하는

이기종 컴퓨팅 시스템 제어 방법.

청구항 12

제1항에 있어서,

상기 결정된 최종 상태를 기초로 머신 러닝 어플리케이션의 동작을 수행하는 단계를 포함하는

이기종 컴퓨팅 시스템 제어 방법.

청구항 13

이기종 컴퓨팅 시스템의 제어 장치에 있어서,

상기 이기종 컴퓨팅 시스템의 주파수 정보를 이용하여, 상기 이기종 컴퓨팅 시스템의 작동 효율을 나타내는 퍼포먼스 모델을 결정하는 퍼포먼스 모델 결정부와,

상기 이기종 컴퓨팅 시스템의 에너지 소모량을 나타내는 파워 모델을 결정하는 파워 모델 결정부와,

상기 퍼포먼스 모델을 통해 획득되는 퍼포먼스 값과 상기 파워 모델을 통해 획득되는 파워 값을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 상기 이기종 컴퓨팅 시스템의 최종 상태로 결정하는 최종 상태 결정부를 포함하고,

상기 이기종 컴퓨팅 시스템은 복수의 이기종 컴퓨팅 장치를 포함하고,

상기 시스템 상태는 상기 복수의 이기종 컴퓨팅 장치 각각에 대한 주파수 값에 의해 결정되는

이기종 컴퓨팅 시스템의 제어 장치.

청구항 14

제13항에 있어서,

상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 및 메모리를 포함하고,

상기 주파수 정보는 상기 CPU, 상기 GPU, 및 상기 메모리 각각에 대한 주파수 값을 포함하는

이기종 컴퓨팅 시스템의 제어 장치.

청구항 15

제13항에 있어서,

상기 퍼포먼스 모델은

$$\mathrm{Perf} = \alpha_C \cdot f_C + \alpha_G \cdot f_G + \alpha_M \cdot f_M + \alpha_{C,G} \cdot f_C \cdot f_G + \alpha_{G,M} \cdot f_G \cdot f_M + \alpha_{M,C} \cdot f_M \cdot f_C + \beta$$

(Perf는 상기 퍼포먼스 값, f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기 복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β는 각각 기 결정된 값)

로 구성되는

이기종 컴퓨팅 시스템의 제어 장치.

청구항 16

제13항에 있어서,

상기 파워 모델은

$$P_{D,f_D} = \epsilon_{D,f_D} \cdot U_D + \zeta_{D,f_D}$$

(P_{D,f_D}는 주파수 값으로 f_D를 가지는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 파워 값, ϵ_{D,f_D}과 ζ_{D,f_D} 각각은 기 지정된 값, U_D는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 사용률(utilization)을 나타내는 사용률 모델)

로 구성되고,

상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 또는 메모리를 포함하고,

상기 사용률은 상기 복수의 이기종 컴퓨팅 장치 중 하나가 동작할 수 있는 전체 동작 시간에 대한 실제 동작하는 시간의 비율인

이기종 컴퓨팅 시스템의 제어 장치.

청구항 17

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 이기종 컴퓨팅 시스템을 제어하는 방법 및 장치에 관한 것이다. 보다 구체적으로, 본 발명은 이기종 컴퓨팅 시스템의 퍼포먼스와 파워에 대한 정보를 이용하여 이기종 컴퓨팅 시스템의 에너지 효율을 향상시키는 이기종 컴퓨팅 시스템의 제어 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 기존의 전자 장치에서의 컴퓨팅은 작업의 대부분을 중앙 처리 장치(CPU)에 의존하였다. 중앙 처리 장치(CPU)의 경우 복잡한 작업을 처리하는 데 최적화되어 있어 단순 반복적인 연산에서는 효율이 떨어진다. 한편, 그래픽 연산 유닛(GPU)의 경우 복잡한 연산에 대해서는 효율이 낮지만 특정한 공식에 의해 반복되는 수치 계산의 효율은 CPU보다 높다. 이에 따라, CPU와 GPU를 함께 구동하여 전자 장치의 수행 속도와 소모 전력을 개선하기 위한 이기종 컴퓨팅 시스템에 대한 연구가 활발이 진행되고 있다.

[0003] 이기종 컴퓨팅 시스템은 기존에 전자 장치를 통해 수행되는 다양한 동작들의 수행 속도 및 소모 전력을 개선할 수 있다. 이러한 측면에서, 이기종 컴퓨팅 시스템은 반복 연산과 복잡한 연산이 함께 수행되어 기존 전자 장치에서는 수행 효율이 낮은, 즉 수행 속도가 길고 소모 전력이 높은 기계 학습의 수행에도 적합하다. 그렇지만 기계 학습은 그 복잡도가 매우 높아 이기종 컴퓨팅 시스템에 의해 수행된다 하더라도 다른 동작들에 비해 수행 효율의 저하가 야기된다. 이에 따라, 보다 향상된 효율을 가지는 이기종 컴퓨팅 시스템에 대한 요구가 증가하고

있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 한국공개특허 제10-2016-0116753호 (2017년 03월 31일 공개)

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하고자 하는 과제는, 이기종 컴퓨팅 시스템의 퍼포먼스와 파워을 통합적으로 고려하여 그 상태를 제어함으로써 이기종 컴퓨팅 시스템의 에너지 효율을 개선하는 이기종 컴퓨팅 시스템의 제어 방법 및 장치를 제공하는 것이다.

[0006] 다만, 본 발명이 해결하고자 하는 과제는 이상에서 언급한 바로 제한되지 않으며, 언급되지는 않았으나 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있는 목적을 포함할 수 있다.

과제의 해결 수단

[0007] 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템 제어 방법은, 상기 이기종 컴퓨팅 시스템의 주파수 정보를 이용하여, 상기 이기종 컴퓨팅 시스템의 작동 효율을 나타내는 퍼포먼스 모델과 상기 이기종 컴퓨팅 시스템의 에너지 소모량을 나타내는 파워 모델을 결정하는 단계와, 상기 퍼포먼스 모델과 상기 파워 모델을 이용하여 상기 이기종 컴퓨팅 시스템의 최종 상태를 결정하는 단계를 포함하고, 상기 이기종 컴퓨팅 시스템은 복수의 이기종 컴퓨팅 장치를 포함할 수 있다.

[0008] 또한, 상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 및 메모리를 포함하고, 상기 주파수 정보는 상기 CPU, 상기 GPU, 및 상기 메모리 각각에 대한 주파수 값을 포함할 수 있다.

[0009] 또한, 상기 퍼포먼스 모델은 $\mathrm{Perf} = \alpha_C \cdot f_C + \alpha_G \cdot f_G + \alpha_M \cdot f_M + \alpha_{C,G} \cdot f_C \cdot f_G + \alpha_{G,M} \cdot f_G \cdot f_M + \alpha_{M,C} \cdot f_M \cdot f_C + \beta$ (Perf는 퍼포먼스 값, f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기 복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β는 각각 기 결정된 값)로 구성될 수 있다.

[0010] 또한, 상기 α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β 각각은 상기 퍼포먼스 모델에 대입된 상기 시스템 상태 중 적어도 하나가 변경되면, 상기 적어도 하나의 변경된 시스템 상태가 상기 퍼포먼스 모델에 재대입됨에 기초하여 기 획득된 연립 방정식이 갱신될 수 있다.

[0011] 또한, 상기 파워 모델은 $\mathrm{P}_{D,f_D} = \epsilon_{D,f_D} \cdot U_D + \zeta_{D,f_D}$ (P_{D,f_D}는 주파수 값으로 f_D를 가지는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 파워 값, ϵ_{D,f_D}과 ζ_{D,f_D} 각각은 기 지정된 값, U_D는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 사용률(utilization)을 나타내는 사용률 모델)로 구성되고, 상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 또는 메모리를 포함하고, 상기 사용률은 상기 복수의 이기종 컴퓨팅 장치 중 하나가 동작할 수 있는 전체 동작 시간에 대한 실제 동작하는 시간의 비율일 수 있다.

[0012] 또한, 상기 U_D는 상기 복수의 이기종 컴퓨팅 장치 각각에 대해 $\mathrm{U}_D = \gamma_{D,C} \cdot f_C + \gamma_{D,G} \cdot f_G + \gamma_{D,M} \cdot f_M + \gamma_{D,CG} \cdot f_C \cdot f_G + \gamma_{D,GM} \cdot f_G \cdot f_M + \gamma_{D,MC} \cdot f_M \cdot f_C + \delta_D$ (f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기

복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D는 각각 기 결정된 값)를 이용하여 획득될 수 있다.

[0013] 또한, 상기 $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D 각각은 상기 사용률 모델에 서로 다른 적어도 7개의 시스템 상태에 따른 사용률 값을 각각 대입하고, 상기 대입에 의해 획득되는 7개의 연립 방정식을 이용하여 결정된 값이며, 상기 시스템 상태는 상기 f_C, 상기 f_G, 상기 f_M 값에 따라 각각 결정될 수 있다.

$\gamma_{D,C}$ $\gamma_{D,G}$ $\gamma_{D,M}$ $\gamma_{D,CG}$ $\gamma_{D,GM}$ $\gamma_{D,MC}$ δ_D

[0014] 또한, 상기 , , , , , , 각각은 상기 파워 모델에 대입된 상기 시스템 상태 중 적어도 하나가 변경되면, 상기 적어도 하나의 변경된 시스템 상태가 상기 파워 모델에 재대입됨에 기초하여 기 획득된 연립 방정식이 갱신될 수 있다.

[0015] 또한, 상기 최종 상태를 결정하는 단계는 상기 퍼포먼스 모델을 통해 획득되는 퍼포먼스 값과 상기 파워 모델을 통해 획득되는 파워 값을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 상기 최종 상태로 결정하는 단계를 포함하고, 상기 시스템 상태는 상기 복수의 이기종 시스템 장치 각각에 대한 주파수 값에 의해 결정될 수 있다.

[0016] 또한, 상기 최종 상태를 결정하는 단계는 상기 기 지정된 조건이 변경되는지 여부를 식별하는 단계와, 상기 기 지정된 조건이 변경되는 것에 대응하여 상기 시스템 상태를 재결정하는 단계를 포함할 수 있다.

[0017] 또한, 상기 결정된 최종 상태를 기초로 머신 러닝 어플리케이션의 동작을 수행하는 단계를 포함할 수 있다.

[0018] 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 장치는, 상기 이기종 컴퓨팅 시스템의 주파수 정보를 이용하여, 상기 이기종 컴퓨팅 시스템의 작동 효율을 나타내는 퍼포먼스 모델을 결정하는 퍼포먼스 모델 결정부와, 상기 이기종 컴퓨팅 시스템의 에너지 소모량을 나타내는 파워 모델을 결정하는 파워 모델 결정부와, 상기 퍼포먼스 모델과 상기 파워 모델을 이용하여 상기 이기종 컴퓨팅 시스템의 최종 상태를 결정하는 최종 상태 결정부를 포함하고, 상기 이기종 컴퓨팅 시스템은 복수의 이기종 컴퓨팅 장치를 포함할 수 있다.

[0019] 또한, 상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 및 메모리를 포함하고, 상기 주파수 정보는 상기 CPU, 상기 GPU, 및 상기 메모리 각각에 대한 주파수 값을 포함할 수 있다.

[0020] 또한, 상기 퍼포먼스 모델은

$$\mathrm{Perf} = \alpha_C \cdot f_C + \alpha_G \cdot f_G + \alpha_M \cdot f_M + \alpha_{C,G} \cdot f_C \cdot f_G + \alpha_{G,M} \cdot f_G \cdot f_M + \alpha_{M,C} \cdot f_M \cdot f_C + \beta$$

(Perf는 퍼포먼스 값, f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기 복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β 는 각각 기 결정된 값)로 구성될 수 있다.

[0021] 또한, 상기 α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β 각각은 상기 퍼포먼스 모델에 서로 다른 적어도 7개의 시스템 상태를 각각 대입하고, 상기 대입에 의해 획득되는 7개의 연립 방정식을 이용하여 결정된 값이며, 상기 시스템 상태는 상기 f_C, 상기 f_G, 상기 f_M 각각의 값에 따라 결정될 수 있다.

α_C α_G α_M $\alpha_{C,G}$ $\alpha_{G,M}$ $\alpha_{M,C}$ β

[0022] 또한, 상기 , , , , , , 각각은 상기 퍼포먼스 모델에 대입된 상기 시스템 상태 중 적어도 하나가 변경되면, 상기 적어도 하나의 변경된 시스템 상태가 상기 퍼포먼스 모델에 재대입됨에 기초하여 기 획득된 연립 방정식이 갱신될 수 있다.

$$\mathrm{P}_{D,f_D} = \epsilon_{D,f_D} \cdot U_D + \zeta_{D,f_D} \quad \mathrm{P}_{D,f_D}$$

[0023] 또한, 상기 파워 모델은 (는 주파수 값으로 f_D를 가지는 상기 복수의 이기종 컴퓨팅 장치 중 하나의 파워 값, ϵ_{D,f_D} 과 ζ_{D,f_D} 각각은 기 지정된 값, U_D 는 상기 복수의 이기종 컴퓨팅 장치 중 하

나의 사용률(utilization))로 구성되고, 상기 복수의 이기종 컴퓨팅 장치는 CPU, GPU, 또는 메모리를 포함하고, 상기 사용률은 상기 구성 요소가 동작할 수 있는 전체 동작 시간에 대한 실제 동작하는 시간의 비율일 수 있다.

[0024] 또한, 상기 U_D는 상기 복수의 이기종 컴퓨팅 장치 각각에 대해

$$U_D = \gamma_{D,C} \cdot f_C + \gamma_{D,G} \cdot f_G + \gamma_{D,M} \cdot f_M + \gamma_{D,CG} \cdot f_C \cdot f_G + \gamma_{D,GM} \cdot f_G \cdot f_M + \gamma_{D,MC} \cdot f_M \cdot f_C + \delta_D$$

(f_C는 상기 복수의 이기종 컴퓨팅 장치 중 CPU의 주파수 값, f_G는 상기 복수의 이기종 컴퓨팅 장치 중 GPU의 주파수 값, f_M은 상기 복수의 이기종 컴퓨팅 장치 중 메모리의 주파수 값, $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D는 각각 기 결정된 값)를 이용하여 획득될 수 있다.

[0025] 또한, 상기 $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D 각각은 상기 파워 모델에 서로 다른 적어도 7개의 시스템 상태를 각각 대입하고, 상기 대입에 의해 획득되는 7개의 연립 방정식을 이용하여 결정된 값이며, 상기 시스템 상태는 상기 f_C, 상기 f_G, 상기 f_M 각각의 값에 따라 결정될 수 있다.

[0026] 또한, 상기 $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D 각각은 상기 파워 모델에 대입된 상기 시스템 상태 중 적어도 하나가 변경되면, 상기 적어도 하나의 변경된 시스템 상태가 상기 파워 모델에 재대입됨에 기초하여 기 획득된 연립 방정식이 갱신될 수 있다.

[0027] 또한, 상기 최종 상태 결정부는 상기 퍼포먼스 모델을 통해 획득되는 퍼포먼스 값과 상기 파워 모델을 통해 획득되는 파워 값을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 상기 최종 상태로 결정하며, 상기 시스템 상태는 상기 복수의 이기종 시스템 장치 각각에 대한 주파수 값에 의해 결정될 수 있다.

[0028] 또한, 상기 최종 상태 결정부는 상기 기 지정된 조건이 변경되는지 여부를 식별하고, 상기 기 지정된 조건이 변경되는 것에 대응하여 상기 시스템 상태를 재결정할 수 있다.

[0029] 또한, 상기 결정된 최종 상태를 기초로 머신 러닝 어플리케이션의 동작을 수행하는 기계 학습 수행부를 더 포함할 수 있다.

발명의 효과

[0030] 본 발명의 실시예에 의하면, 이기종 컴퓨팅 시스템이 최적의 퍼포먼스 값 및 최적의 파워 값을 가지도록 제어함으로써 이기종 컴퓨팅 시스템의 효율을 향상시킬 수 있다.

[0031] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0032] 도 1은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 장치의 기능적 구성을 도시한다.

도 2는 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 CEML의 기능적 구성을 도시한다.

도 3은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 방법의 각 단계의 흐름을 도시한다.

도 4는 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 최종 상태의 결정을 위한 알고리즘의 예를 도시한다.

도 5는 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 최종 상태의 결정을 위한 알고리즘의 다른 예를 도시한다.

도 6은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 최종 상태를 결정하는 단계를 구체적으로 도시한다.

도 7은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 방법에 의해 제어된 이기종 컴퓨팅 시스템의 에러율의 실험 결과의 예를 도시한다.

도 8은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 방법에 의해 제어된 이기종 컴퓨팅 시스템의 에너지 소모량의 실험 결과의 다른 예를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0033] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명의 범주는 청구항에 의해 정의될 뿐이다.

[0034] 본 발명의 실시예들을 설명함에 있어서 공지 기능 또는 구성에 대한 구체적인 설명은 본 발명의 실시예들을 설명함에 있어 실제로 필요한 경우 외에는 생략될 것이다. 그리고 후술되는 용어들은 본 발명의 실시예에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0035] 도 1은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 장치의 기능적 구성을 도시한다. 이기종 컴퓨팅 시스템(130)의 제어 장치(10)(이하 '제어 장치')의 각 구성은 하드웨어나 소프트웨어, 또는, 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0036] 도 1을 참조하면, 제어 장치(10)는 머신 러닝 어플리케이션(110), CEML(120), 이기종 컴퓨팅 시스템(130)을 포함할 수 있다. 경우에 따라, 제어 장치(10)는 이기종 컴퓨팅 시스템(130)과 별도의 장치로 구성될 수도 있으며, 이러한 경우에는 이기종 컴퓨팅 시스템(130)과 무선 연결 또는 유선 연결을 통해 이기종 컴퓨팅 시스템(130)을 제어할 수 있다.

[0037] 머신 러닝 어플리케이션(110)은 머신 러닝(기계 학습)을 수행할 수 있다. 구체적으로, 머신 러닝 어플리메이션(110)은 기계 학습을 위한 응용프로그램(또는 알고리즘)을 저장할 수 있고, 응용프로그램을 실행하여 기계 학습을 수행할 수 있다. 경우에 따라, 머신 러닝 어플리케이션(110)은 이기종 컴퓨팅 시스템(130)과의 연결에 기초하여 기계 학습을 수행할 수 있다. 구체적으로 예를 들면, 이기종 컴퓨팅 시스템(130)의 하드웨어적 구성의 제어에 의해 머신 러닝 어플리케이션(110)에 저장된 응용프로그램에 대한 기계 학습이 수행될 수 있다.

[0038] 머신 러닝 어플리케이션(110)은 공지의 다양한 기계 학습 프로그램 중 적어도 하나를 포함할 수 있다. 한편, 도 1에서는 제어 장치(10)에 의해 수행되는 응용프로그램이 기계 학습에 대한 것으로 설명하고 있으나 이에 제한되지 않는다, 즉, 머신 러닝 어플리케이션(110)이 아닌 응용프로그램이 제어 장치(10)에 포함될 수도 있고, 그에 따라 포함된 응용프로그램의 동작이 수행될 수 있음은 물론이다.

[0039] CEML(120)은 이기종 컴퓨팅 시스템에서 효과적인 기계 학습의 수행을 위한 역할을 하는 제어 장치(10)의 구성요소일 수 있다. CEML(120)은 이기종 컴퓨팅 시스템(130)의 퍼포먼스와 파워를 이용하여 이기종 컴퓨팅 시스템(130)의 최종 상태를 제어할 수 있다. 퍼포먼스는 이기종 컴퓨팅 시스템(130)의 수행 속도 또는 수행 효율을 나타내고, 파워는 이기종 컴퓨팅 시스템(130)에 의해 소요되는 전력을 나타낼 수 있다.

[0040] 한편, CEML(120)이라는 용어는 coordinated runtime system for efficient machine learning on heterogeneous computing systems의 약자로 표기하였으나 이에 제한되지 않고 다양한 명칭으로 지칭될 수 있다. CEML(120)과 관련하여서는 도 2를 통해 보다 구체적으로 설명하겠다.

[0041] 이기종 컴퓨팅 시스템(130)은 복수의 이기종 컴퓨팅 장치를 포함할 수 있다. 복수의 이기종 컴퓨팅 장치는 예를 들면, CPU(central processing unit)와 GPU(graphic processing unit), 메모리(memory)를 포함할 수 있다. 이기종 컴퓨팅 시스템(130)은 CPU와 GPU를 함께 구동하여 다양한 응용프로그램을 효과적으로 실행시킬 수 있다. 예를 들어, 이기종 컴퓨팅 시스템(130)은 CPU와 GPU를 함께 구동하여 머신 러닝 어플리케이션(110)에 대한 동작이 수행되도록 할 수 있다. 이러한 경우, 이기종 컴퓨팅 시스템(130)은 CPU와 GPU를 함께 구동시킴으로써 머신 러닝 어플리케이션(110)의 동작, 즉 기계 학습의 수행과 관련된 퍼포먼스를 개선하고, 파워를 최적화할 수 있다.

[0042] 이기종 컴퓨팅 시스템(130)는 CEML(120)에 이기종 컴퓨팅 시스템(130)에 대한 정보를 제공할 수 있다. 이기종 컴퓨팅 시스템(130)에 대한 정보는 예를 들면, CPU, GPU, 및 메모리 중 적어도 하나에 대한 정보를 포함할 수 있다. 보다 구체적으로, 이기종 컴퓨팅 시스템(130)는 CEML(120)에 CPU, GPU, 및 메모리 각각에 대한 주파수 값

에 대한 정보를 제공할 수 있다.

[0043] 이기종 컴퓨팅 시스템(130)은 CEML(120)에 의해 상태가 제어될 수 있다. 일실시예에서, 이기종 컴퓨팅 시스템(130)은 CEML(120)으로부터 상태에 대한 정보를 수신하여 수신된 정보에 상응하도록 이기종 컴퓨팅 시스템(130)의 상태를 제어할 수 있다. 다른 일실시예에서, 이기종 컴퓨팅 시스템(130)은 CEML(120)과의 연결에 기반하여 CEML(120)에 의해 상태가 제어될 수 있다. 여기서, 이기종 컴퓨팅 시스템(130)의 상태는 CPU, GPU, 및 메모리 각각의 주파수의 상태, 즉 주파수 값을 지칭하는 용어일 수 있다.

[0044] 한편, 도 1에서 이기종 컴퓨팅 시스템(130)을 제어 장치(10)에 포함되는 것으로 도시하였으나, 이에 제한되지 않고 제어 장치(10)와 별도의 구성으로 존재할 수 있다.

[0045] 도 2는 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 CEML의 기능적 구성을 도시한다. 이하 사용되는 '…부', '…기' 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어, 또는, 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0046] 도 2를 참고하면, CEML(120)은 퍼포먼스 모델 결정부(210), 파워 모델 결정부(220), 최종 상태 결정부(230)를 포함할 수 있다.

[0047] 퍼포먼스 모델 결정부(210)는 이기종 컴퓨팅 시스템(130)의 주파수 정보를 이용하여 이기종 컴퓨팅 시스템(130)의 작동 효율을 나타내는 퍼포먼스 모델을 결정할 수 있다. 구체적으로, 퍼포먼스 모델 결정부(210)는 이기종 컴퓨팅 시스템(130)으로부터 다양한 정보, 예를 들어 이기종 컴퓨팅 시스템(130)의 주파수 정보, 퍼포먼스 정보를 획득할 수 있다. 주파수 정보는 이기종 컴퓨팅 시스템에 포함되는 CPU, GPU, 및 메모리 각각의 주파수 값을 포함할 수 있다. 퍼포먼스 정보는 주파수 정보에 따른 이기종 컴퓨팅 시스템의 퍼포먼스의 값일 수 있다. 퍼포먼스 모델 결정부(210)는 하기의 수학식 1로 구성되는 퍼포먼스 모델을 결정할 수 있다.

수학식 1

[0048] $$\mathrm{Perf} = \alpha_C \cdot f_C + \alpha_G \cdot f_G + \alpha_M \cdot f_M + \alpha_{C,G} \cdot f_C \cdot f_G + \alpha_{G,M} \cdot f_G \cdot f_M + \alpha_{M,C} \cdot f_M \cdot f_C + \beta$$

[0049] 수학식 1에서, Perf는 퍼포먼스 모델에 의해 예측된 이기종 컴퓨팅 시스템(130)의 퍼포먼스 값(이하 퍼포먼스 예측 값), f_C는 CPU의 주파수 값, f_G는 GPU의 주파수 값, f_M은 메모리의 주파수 값, α_C, α_G, α_M, $\alpha_{C,G}$, $\alpha_{G,M}$, $\alpha_{M,C}$, β는 퍼포먼스 모델의 계수들이다.

[0050] 퍼포먼스 모델의 계수들 각각은 이기종 컴퓨팅 시스템(130)의 서로 다른 적어도 7개의 시스템 상태와 각각의 상태에 따른 퍼포먼스 값을 퍼포먼스 모델에 대입하여 획득되는 7개의 방정식을 이용하여 결정될 수 있다. 여기서, 시스템 상태는 이기종 컴퓨팅 시스템(130)의 (f_C, f_G, f_M)를 지칭하는 것일 수 있다. 보다 구체적으로, 퍼포먼스 모델 결정부(210)는 수학식 1을 통해 획득되는 서로 다른 7개의 방정식을 연립하여 계산함으로써 7개의 미지수인 퍼포먼스 모델의 계수들 각각의 값을 결정할 수 있다.

[0051] 한편, 시스템 상태는 이기종 컴퓨팅 시스템(130)에서 지원하는 주파수에 따라 달라질 수 있다. 7개의 시스템 상태는 이기종 컴퓨팅 시스템(130)이 지원하는 주파수 범위 내에서 보다 넓은 범위의 주파수를 나타내도록 선정될 수 있다. 파워 모델 결정부(220)는 이기종 컴퓨팅 시스템(130)의 주파수 정보를 이용하여 이기종 컴퓨팅 시스템(130)의 에너지 소모량(또는 전력 소모량)을 나타내는 파워 모델을 결정할 수 있다. 구체적으로, 파워 모델 결정부(220)는 이기종 컴퓨팅 시스템(130)으로부터 다양한 정보, 예를 들어 이기종 컴퓨팅 시스템(130)의 사용률(utilization)과 관련된 정보, 주파수 정보를 획득할 수 있다. 사용률과 관련된 데이터는 복수의 이기종 컴퓨팅 장치 각각에 대해 이기종 컴퓨팅 장치가 동작할 수 있는 전체 동작 시간에 대한 실제 동작하는 시간의 비율인 사용률과 관련한 정보일 수 있다. 주파수 정보는 이기종 컴퓨팅 시스템에 포함되는 CPU, GPU, 및 메모리 각각의 주파수 값을 포함할 수 있다. 파워 모델 결정부(220)는 하기의 수학식 2로 구성되는 파워 모델을 결정할 수 있다.

수학식 2

[0052] $$P_{D,f_D} = \epsilon_{D,f_D} \cdot U_D + \zeta_{D,f_D}$$

[0053] 수학식 2에서, P_{D,f_D} 는 주파수 값으로 f_D를 가지는 복수의 이기종 컴퓨팅 장치 중 하나의 파워 값, ϵ_{D,f_D}과 ζ_{D,f_D}는 파워 모델의 기 지정된 값, U_D는 복수의 이기종 컴퓨팅 장치 중 하나의 사용률을 나타내는 사용률 모델이다. 여기서, 파워 모델의 기 지정된 계수는 복수의 이기종 컴퓨팅 장치 각각에 대해 스트레스 벤치마크를 이용하여 실험적으로 기 결정된 회귀 계수(regression coefficient)일 수 있다.

[0054] 스트레스 벤치마크는 이기종 컴퓨팅 시스템에 포함되는 CPU, GPU 및 메모리에 대하여 사용자가 원하는 사용률을 가지도록 부하(load)를 제공하는 구성(또는 프로그램)일 수 있다.

[0055] 한편, U_D는 하기의 수학식 3과 같이 나타날 수 있다.

수학식 3

[0056] $$U_D = \gamma_{D,C} \cdot f_C + \gamma_{D,G} \cdot f_G + \gamma_{D,M} \cdot f_M + \gamma_{D,CG} \cdot f_C \cdot f_G + \gamma_{D,GM} \cdot f_G \cdot f_M + \gamma_{D,MC} \cdot f_M \cdot f_C + \delta_D$$

[0057] 수학식 3에서, $\gamma_{D,C}$, $\gamma_{D,G}$, $\gamma_{D,M}$, $\gamma_{D,CG}$, $\gamma_{D,GM}$, $\gamma_{D,MC}$, δ_D 는 사용률 모델의 계수들로서 이기종 컴퓨팅 시스템(130)의 서로 다른 적어도 7개의 시스템 상태에 따른 사용률 값을 사용률 모델에 대입하여 획득되는 7개의 방정식을 이용하여 결정될 수 있다. 파워 모델 결정부(220)는 수학식 3을 통해 획득되는 서로 다른 7개의 방정식을 연립하여 계산함으로써 7개의 미지수인 사용률 모델의 계수들 각각의 값을 결정할 수 있다. 파워 모델 결정부(220)는 퍼포먼스 모델 결정부(210)와 동일한 방식으로 사용률 모델의 계수들을 결정할 수 있다.

[0058] 파워 모델 결정부(220)는 수학식 3의 사용률 모델의 계수들이 결정되면, 이를 수학식 2에 대입하여 복수의 이기종 컴퓨팅 장치 각각의 파워 모델을 결정할 수 있다. 이기종 컴퓨팅 시스템의 파워 모델은 복수의 이기종 컴퓨팅 장치 각각의 파워 모델의 합으로 결정될 수 있다. 이를 구체적으로 나타내면 하기의 수학식 4와 같이 표현될 수 있다.

수학식 4

[0059] $$P=P_C+P_G+P_M$$

[0060] 수학식 4에서, P는 이기종 컴퓨팅 시스템(130)의 파워 모델에 의해 예측된 이기종 컴퓨팅 시스템(130)의 파워 값(이하, '파워 예측 값'), P_C는 CPU의 파워 모델에 의해 예측된 CPU의 파워 값, P_G는 GPU의 파워 모델에 의해 예측된 GPU의 파워 값, P_M는 메모리의 파워 모델에 의해 예측된 파워값이다.

[0061] 최종 상태 결정부(230)는 퍼포먼스 모델과 파워 모델을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 이기종 컴퓨팅 시스템의 최종 상태로 결정할 수 있다. 최종 상태 결정부(230)는 퍼포먼스 모델을 통해 획득되는 퍼포먼스 예측 값과 파워 모델을 통해 획득되는 파워 예측 값을 이용하여 이기종 컴퓨팅 시스템(130)의 상태(이하, '최종 상태')를 결정할 수 있다. 최종 상태는 최종 상태 결정부(230)를 통해 결정된 이기종 컴퓨팅 시스템(130)의 효율을 극대화하는 최적의 상태일 수 있다. 최종 상태 결정부(230)의 최종 상태의 결정과 관련된 구체적인 예는 도 4 또는 도 5를 참조할 수 있다.

[0062] 한편, 기 지정된 조건은 사용자의 입력에 의해 결정되며, 예를 들면 이기종 컴퓨팅 시스템(130)의 소모 전력이 최소가 되는 시스템 상태, 이기종 컴퓨팅 시스템(130)의 동작에 소요되는 시간이 소정 값 이하인 시스템 상태, 이기종 컴퓨팅 시스템(130)의 효율이 소정 값 이상인 시스템 상태일 수 있다.

[0063] 최종 상태 결정부(230)는 결정된 최종 상태가 기 지정된 조건에 부합(또는 만족)하는지 여부를 판정할 수 있다. 구체적으로, 최종 상태 결정부(230)는 최종 상태를 결정하는 시점에서, 상기 결정된 최종 상태가 기 지정된 조건에 부합하는 지 여부를 판정할 수 있다. 만약 기 지정된 조건에 부합하는 시스템 상태가 복수인 경우, 그 중에서 보다 최적인 상태를 선별할 수 있다. 최종 상태 결정부(230)는 최종 상태가 기 지정된 조건에 부합하지 않는 경우, 퍼포먼스 모델과 파워 모델을 이용하여 최종 상태를 재결정할 수 있다.

[0064] 경우에 따라, 최종 상태 결정부(230)는 기 지정된 조건이 변경되는지 여부를 식별할 수 있다. 최종 상태 결정부(230)는 기 지정된 조건이 변경되는 경우, 변경된 조건에 부합하는 최종 상태를 결정하기 위해 상술한 다양한 동작들, 예를 들면 상술한 퍼포먼스 예측 값과 파워 예측 값을 이용한 동작들을 재수행할 수 있다.

[0065] 도 3은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 방법의 각 단계의 흐름을 도시한다. 도 3에 도시된 방법의 각 단계의 적어도 일부(예: S110, S120)는 경우에 따라 도면에 도시된 바와 그 순서를 달리하여 수행될 수 있음은 물론이다. 이하에서는 도 1과 도 2와 중복되는 내용이 생략될 수 있다.

[0066] 도 3을 참조하면, 퍼포먼스 모델 결정부(210)는 이기종 컴퓨팅 시스템(130)의 주파수 정보를 이용하여 퍼포먼스 모델을 결정할 수 있다(S110). 보다 구체적으로, 퍼포먼스 모델 결정부(210)는 이기종 컴퓨팅 시스템(130)에 포함되는 복수의 이기종 컴퓨팅 장치(예: CPU, GPU, 메모리)의 주파수 정보를 획득하고, 이를 이용하여 퍼포먼스 모델을 결정할 수 있다.

[0067] 파워 모델 결정부(220)는 이기종 컴퓨팅 시스템(130)의 주파수 정보를 이용하여 파워 모델을 결정할 수 있다(S120). 보다 구체적으로, 파워 모델 결정부(220)는 이기종 컴퓨팅 시스템(130)에 포함되는 복수의 이기종 컴퓨팅 장치의 주파수 정보를 획득하고, 이를 이용하여 파워 모델을 결정할 수 있다.

[0068] 최종 상태 결정부(230)는 퍼포먼스 모델과 파워 모델을 이용하여 이기종 컴퓨팅 시스템의 최종 상태를 결정할 수 있다(S130). 보다 구체적으로 최종 상태 결정부(230)는 퍼포먼스 모델에 의해 획득되는 퍼포먼스 예측 값과 파워 모델에 의해 획득되는 파워 예측 값을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 결정할 수 있다. 최종 상태 결정부(230)는 결정된 시스템 상태를 이기종 컴퓨팅 시스템(130)에 적용할 최종 상태로 결정할 수 있다.

[0069] 최종 상태 결정부(230)의 최종 상태의 결정은 도 4와 도 5를 통해 후술되는 알고리즘을 통해 이루어질 수 있다.

[0070] 도 4는 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 최종 상태의 결정을 위한 알고리즘의 예를 도시한다. 최종 상태 결정부(230)는 도 4의 알고리즘을 통해 최적의 시스템 상태를 결정할 수 있다.

[0071] 보다 구체적으로, 도 4의 알고리즘은 최적의 시스템 상태를 찾기 위한 알고리즘의 예로서, 모든 실현 가능한 시스템의 상태를 조사하여 기 지정된 조건에 부합하는 시스템 상태를 결정할 수 있다. 예를 들어, 기 지정된 조건이 이기종 컴퓨팅 시스템(130)의 효율을 최대화하는 시스템 상태인 경우, 최종 상태 결정부(230)는 이기종 컴퓨팅 시스템(130)의 효율을 최대화하는 시스템 상태를 최종 상태로 결정할 수 있다.

[0072] 도 5는 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 최종 상태의 결정을 위한 알고리즘의 다른 예를 도시한다. 최종 상태 결정부(230)는 도 5의 알고리즘을 통해 최적의 시스템 상태를 결정할 수 있다.

[0073] 보다 구체적으로, 도 5의 알고리즘은 최적의 시스템 상태를 찾기 위한 알고리즘의 예로서, 초기의 시스템 상태에서 시작하여 이웃한 시스템 상태 각각에 대해 효율을 비교하는 방식으로 조사하여 기 지정된 조건에 부합하는 시스템 상태를 결정할 수 있다.

[0074] 도 6은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 최종 상태를 결정하는 단계를 구체적으로 도시한다.

[0075] 도 6을 참조하면, 최종 상태 결정부(230)는 퍼포먼스 모델 및 파워 모델에 기초하여 기 지정된 조건을 만족하는 시스템 상태를 최종 상태로 결정할 수 있다(S210). 보다 구체적으로, 최종 상태 결정부(230)는 퍼포먼스 모델을 통해 획득되는 퍼포먼스 값과 파워 모델을 통해 획득되는 파워 값을 이용하여 기 지정된 조건을 만족하는 시스템 상태를 최종 상태로 결정할 수 있다.

[0076] 최종 상태 결정부(230)는 기 지정된 조건이 변경되는지 여부를 식별할 수 있다(S220). 최종 상태 결정부(230)는 결정된 최종 상태와 관련된 기 지정된 조건이 변경되는지 여부를 식별할 수 있다.

[0077] 최종 상태 결정부(230)는 기 지정된 조건이 변경되는 것에 대응하여 시스템 상태를 재결정할 수 있다(S230). 최종 상태 결정부(230)는 기 지정된 조건이 변경되는 경우, 기 지정된 조건에 부합하는 시스템 상태를 결정할 수

있다. 최종 상태 결정부(230)는 기 지정된 조건이 변경되지 않은 경우, 결정된 최종 상태를 유지할 수 있다.

[0078] 최종 상태 결정부(230)는 재결정된 시스템 상태로 최종 상태를 변경할 수 있다(S240). 최종 상태 결정부(230)는 기 지정된 조건이 변경됨에 따라 재결정된 시스템 상태로 최종 상태를 변경할 수 있다.

[0079] 도 7은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 방법에 의해 제어된 이기종 컴퓨팅 시스템의 에러율의 실험 결과의 예를 도시한다.

[0080] 도 7은 머신 러닝 어플리케이션(110)의 종류(예: CF, IN, LR, MN, RB, VP, WD)에 따른 이기종 컴퓨팅 시스템(130)의 에러율을 나타낸다. 구체적으로, 에러율은 퍼포먼스 모델과 파워 모델 각각에 의해 예측된 값에 대응하도록 이기종 컴퓨팅 시스템(130)의 시스템 상태가 설정된 후 실제 이기종 컴퓨팅 시스템(130)의 퍼포먼스 값과 파워 값의 오차에 대한 것일 수 있다.

[0081] 도 7에 따르면, 본 발명의 실시예에 따른 이기종 컴퓨팅 시스템(130)은 퍼포먼스와 관련하여 약 5.3%, 파워와 관련하여 8.8%의 매우 작은 에러율을 나타내므로 뛰어난 성능을 가짐을 알 수 있다.

[0082] 도 8은 본 발명의 일 실시예에 따른 이기종 컴퓨팅 시스템의 제어 방법에 의해 제어된 이기종 컴퓨팅 시스템의 에너지 소모량의 실험 결과의 다른 예를 도시한다.

[0083] 도 8은 머신 러닝 어플리케이션(110)의 종류(예: CF, IN, LR, MN, RB, VP, WD)에 따른 이기종 컴퓨팅 시스템(130)의 에너지 소모량을 실험한 결과를 나타낸다. 도 8에서 CEML-L은 도 5의 알고리즘이 이용될 때의 이기종 컴퓨팅 시스템(130)의 에너지 소모량을 나타낸다. CEML-E은 도 4의 알고리즘이 이용될 때의 이기종 컴퓨팅 시스템(130)의 에너지 소모량을 나타낸다.

[0084] 도 8을 참조하면, 기본적인 에너지 소모량인 baseline과 최적의 에너지 소모량, 즉 최소로 에너지가 소모될 때의 소모량인 static best를 기준으로, CEML-L과 CEML-E 모두 static best에 근사한 수치로 나타남을 알 수 있다. 다시 말해, 본 발명의 실시예에 따라 이기종 컴퓨팅 시스템(130)이 제어되는 경우, 이기종 컴퓨팅 시스템(130)의 에너지 효율이 향상될 수 있다.

[0085] 본 발명의 일 실시예에 따르면, 제어 장치(10)는 이기종 컴퓨팅 시스템(130)의 주파수 정보를 이용하여 CEML(120)의 구성 요소를 통해 이기종 컴퓨팅 시스템(130)이 제어되도록 함으로써, 최적의 효율로 이기종 컴퓨팅 시스템(130)이 작동하도록 할 수 있다. 즉, 제어 장치(10)는 이기종 컴퓨팅 시스템(130)의 퍼포먼스와 파워를 개선하여 최적의 효율로 이기종 컴퓨팅 시스템(130)이 작동하도록 할 수 있다.

[0086] 본 명세서에 첨부된 블록도의 각 블록과 흐름도의 각 단계의 조합들은 컴퓨터 프로그램 인스트럭션들에 의해 수행될 수도 있다. 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서를 통해 수행되는 그 인스트럭션들이 블록도의 각 블록 또는 흐름도의 각 단계에서 설명된 기능들을 수행하는 수단을 생성하게 된다. 이들 컴퓨터 프로그램 인스트럭션들은 특정 방식으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 블록도의 각 블록 또는 흐름도 각 단계에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다. 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 블록도의 각 블록 및 흐름도의 각 단계에서 설명된 기능들을 실행하기 위한 단계들을 제공하는 것도 가능하다.

[0087] 또한, 각 블록 또는 각 단계는 특정된 논리적 기능(들)을 실행하기 위한 하나 이상의 실행 가능한 인스트럭션들을 포함하는 모듈, 세그먼트 또는 코드의 일부를 나타낼 수 있다. 또, 몇 가지 대체 실시예들에서는 블록들 또는 단계들에서 언급된 기능들이 순서를 벗어나서 발생하는 것도 가능함을 주목해야 한다. 예컨대, 잇달아 도시되어 있는 두 개의 블록들 또는 단계들은 사실 실질적으로 동시에 수행되는 것도 가능하고 또는 그 블록들 또는 단계들이 때때로 해당하는 기능에 따라 역순으로 수행되는 것도 가능하다.

[0088] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 품질에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 명세서에 개시된 실시예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명

하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 균등한 범위 내에 있는 모든 기술사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

[0089] 10: 제어 장치

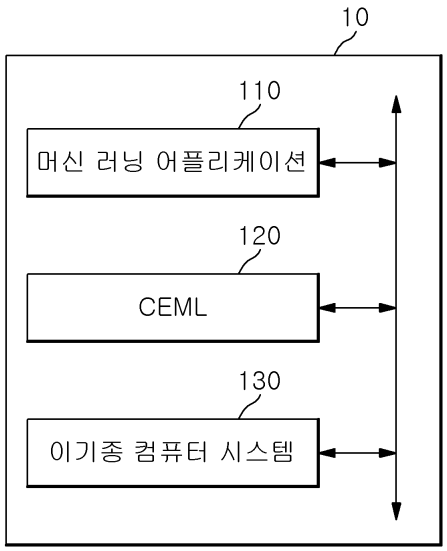
110: 머신 러닝 어플리케이션

120: CEML

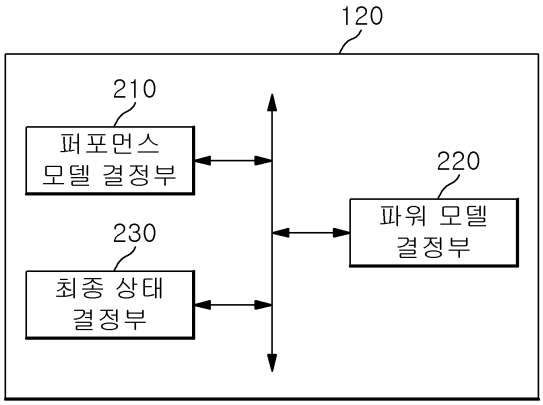
130: 이기종 컴퓨팅 시스템

도면

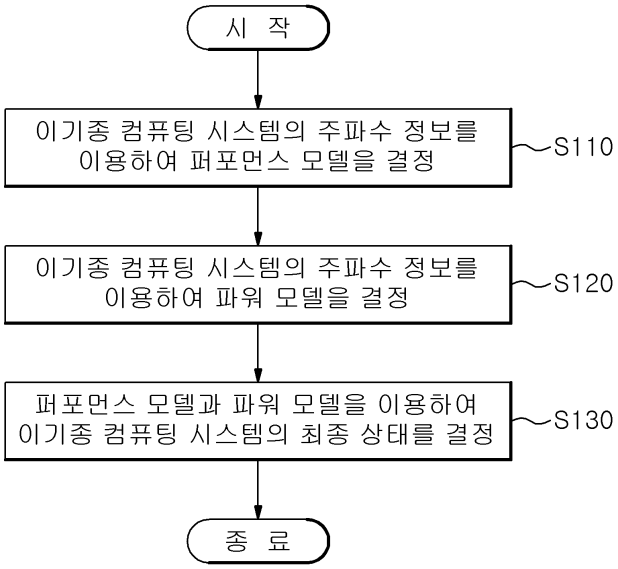
도면1



도면2



도면3



도면4

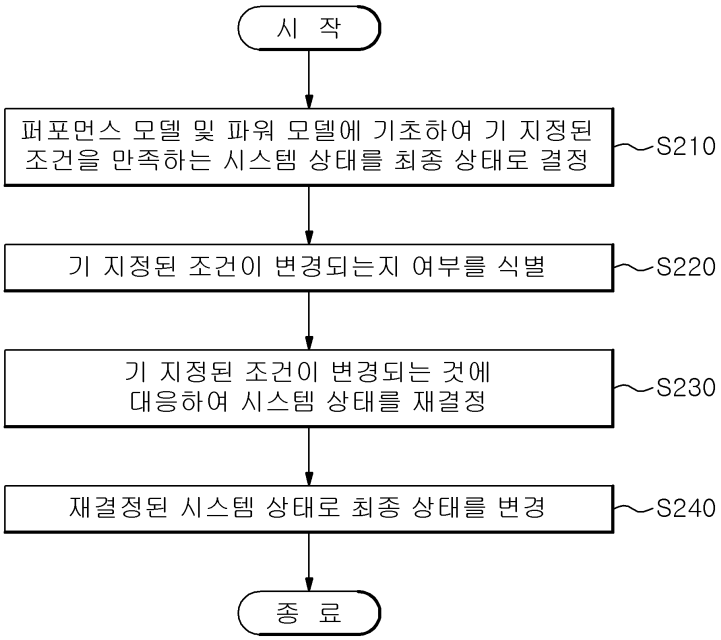
Algorithm 1 The echaustive search function

```
1: Procedure EXPLOREWITHE XHAUSTIVESEARCH
2:     bestState ← getInitialState()
3:     bestState ← estimateScore(bestState)
4:     for fC ∈ FC
5:         for fG ∈ FG
6:             for fM ∈ FM
7:                 cState ← (fC,fG,fM)
8:                 cState ← estimateScore(cState)
9:                 if cScore>bestScore ∧checkConstraint(cState)
10:                    bestState←cState
11:                    bestState←cScore
12:                end if
13:            end for
14:        end for
15:    end for
16:    setSystemState(bestState)
17: end procedure
```

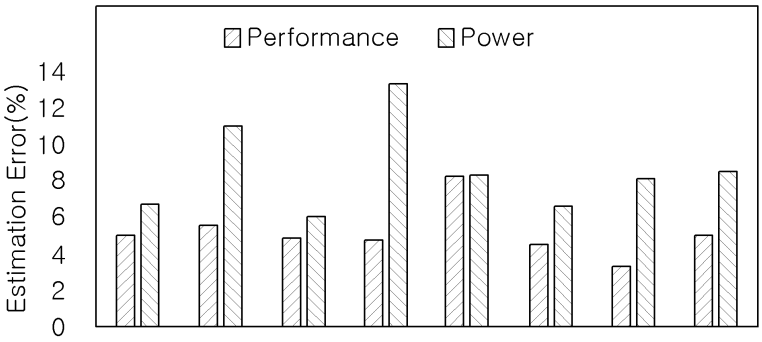
도면5

```
Algorithm 2 The local search function
 1: Procedure EXPLOREW ith LocalSearch
 2:     bestState←getInitialState()
 3:     bestState←estimateScore(bestState)
 4:     while true
 5:         cState←getBestNeighborState(bestState)
 6:         cScore←estimateScore(cState)
 7:         if cScore>bestScore
 8:             bestState←cState
 9:             bestScore←cScore
10:         else
11:             bresk
12:         end if
13:     end while
14:     setSystemState(bestState)
15: end procedure
```

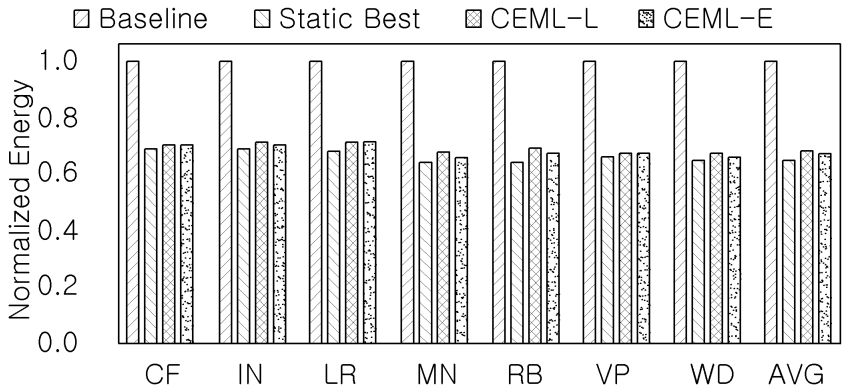
도면6



도면7


Performance
Power
Estimation Error(%)
14
12
10
8
6
4
2
0

도면8


Baseline
Static Best
CEML-L
CEML-E
Normalized Energy
1.0
0.8
0.6
0.4
0.2
0.0
CF
IN
LR
MN
RB
VP
WD
AVG