(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년10월26일
(11) 등록번호 10-1669555
(24) 등록일자 2016년10월20일

(51) 국제특허분류(Int. Cl.)
G06F 9/50 (2006.01)
(52) CPC특허분류
G06F 9/50 (2013.01)
G06F 9/5038 (2013.01)
(21) 출원번호 10-2015-0106844
(22) 출원일자 2015년07월28일
심사청구일자 2015년07월28일
(56) 선행기술조사문헌
KR100788328 B1*
JP2008176646 A*
KR1020120139237 A*
KR1020120035480 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
최영리
울산광역시 울주군 언양읍 유니스트길 50, 401동 701호 (울산과학기술대학교 교수아파트)
김선태
울산광역시 울주군 언양읍 유니스트길 50 울산과학기술대학교 학생기숙사 302동 601호
황은지
경상남도 진주시 진주대로 1317 이현하이클래스웰 가아파트 105동 301호
(74) 대리인
제일특허법인

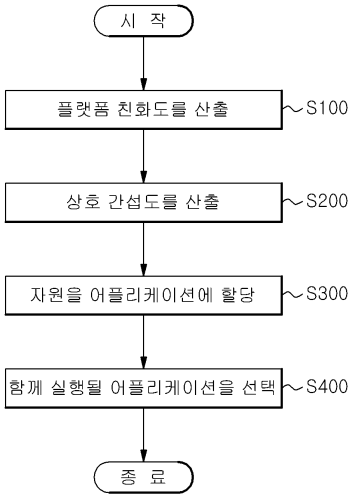
전체 청구항 수 : 총 19 항 심사관 : 유진태

(54) 발명의 명칭 **어플리케이션에 자원을 할당하는 장치 및 방법**

(57) 요 약

본 발명의 일 실시예에 따른 자원 할당 장치는 어플리케이션이 플랫폼(platform)에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출하는 플랫폼 친화도 산출부; 상기 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 상기 플랫폼에서 실행될 때, 상기 어플리케이션이 상기 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출하는 상호 간섭도 산출부; 및 상기 플랫폼 친화도를 기초로 상기 플랫폼에 포함된 자원을 상기 어플리케이션에 할당하고, 상기 상호 간섭도를 기초로 상기 적어도 하나 이상의 어플리케이션 중에서 상기 어플리케이션과 함께 실행될 어플리케이션을 선택하는 자원 할당부를 포함한다.

대 표 도 - 도5



(52) CPC특허분류
G06F 9/5083 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711015927[2012R1A1A1A1043329]
부처명 교육과학기술부
연구관리전문기관 재단법인 한국연구재단
연구사업명 신진연구
연구과제명 병렬 및 분산 애플리케이션을 위한 클라우드 컴퓨팅 기술 연구
기 여 율 1/1
주관기관 국립대학법인 울산과학기술대학교 산학협력단
연구기간 2014.09.01 ~ 2015.08.31
공지예외적용 : 있음

명 세 서

청구범위

청구항 1

어플리케이션(application)이 플랫폼(platform)에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출하는 플랫폼 친화도 산출부;

상기 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 상기 플랫폼에서 실행될 때, 상기 어플리케이션이 상기 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출하는 상호 간섭도 산출부; 및

상기 플랫폼 친화도를 기초로 상기 플랫폼에 포함된 자원을 상기 어플리케이션에 할당하고, 상기 상호 간섭도를 기초로 상기 적어도 하나 이상의 어플리케이션 중에서 상기 어플리케이션과 함께 실행될 어플리케이션을 선택하는 자원 할당부를 포함하며,

상기 플랫폼은 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)에 포함되는 복수의 플랫폼 중 어느 하나이고,

상기 복수의 플랫폼 각각은 서로 상이하며,

상기 플랫폼 친화도 산출부는,

상기 어플리케이션을 포함하는 복수의 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 복수의 어플리케이션마다 산출하고,

상기 자원 할당부는,

상기 복수의 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제1 플랫폼 친화도 평균값을 산출하며, 상기 복수의 어플리케이션 중에서 상기 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션에 상기 자원을 할당하는

자원 할당 장치.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 플랫폼 친화도 산출부는,

상기 어플리케이션이 상기 복수의 플랫폼 각각에서 실행될 때의 실행 시간을 나타내는 비교 실행 시간과, 상기 어플리케이션이 상기 복수의 플랫폼 중에서 상기 플랫폼 친화도의 산출 대상이 되는 제1 플랫폼에서 실행될 때의 실행 시간을 나타내는 기준 실행 시간을 기초로 상기 플랫폼 친화도를 산출하는

자원 할당 장치.

청구항 4

제 3 항에 있어서,

상기 비교 실행 시간은,

상기 복수의 플랫폼 각각의 노드(node)에서 실행되는 상기 어플리케이션에 포함된 작업(task)의 수가 1개인 것

을 기준으로 산출되고,

상기 기준 실행 시간은,

상기 제1 플랫폼의 노드에서 실행되는 상기 어플리케이션에 포함된 작업의 수가 1개인 것을 기준으로 산출되는

자원 할당 장치.

청구항 5

제 1 항에 있어서,

상기 상호 간섭도 산출부는,

상기 적어도 하나 이상의 어플리케이션과 상기 어플리케이션으로 구성된 조합이 복수 개일 때,

상기 조합 중 어느 하나가 상기 플랫폼에서 실행될 때의 실행 시간을 나타내는 동시 실행 시간과 상기 어플리케이션이 상기 플랫폼에서 단독으로 실행될 때의 실행 시간을 나타내는 단독 실행 시간을 기초로, 상기 플랫폼에서의 상기 조합의 상호 간섭도를 나타내는 조합 상호 간섭도를 산출하고,

상기 복수의 조합마다 산출된 조합 상호 간섭도의 평균값을 상기 상호 간섭도로 산출하는

자원 할당 장치.

청구항 6

삭제

청구항 7

제 1 항에 있어서,

상기 자원 할당부는,

상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 제1 어플리케이션에 우선적으로 상기 자원을 할당하는

자원 할당 장치.

청구항 8

제 7 항에 있어서,

상기 자원 할당부는,

상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 제1 어플리케이션에 우선적으로 상기 자원을 할당함에 있어서,

상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당하는

자원 할당 장치.

청구항 9

제 8 항에 있어서,

상기 자원 할당부는,

상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당함에 있어서,

상기 플랫폼의 자원을 상기 제1 어플리케이션 각각에 동일한 수로 할당하되, 상기 제1 어플리케이션 각각에 할당 가능한 자원의 최대 수는 상기 제1 어플리케이션 각각에 포함된 작업의 수인 것을 특징으로 하는

자원 할당 장치.

청구항 10

제 1 항에 있어서,

상기 자원 할당부는,

상기 복수의 어플리케이션 중에서 상기 자원이 할당되지 않은 작업을 포함하는 적어도 하나 이상의 제2 어플리케이션에 대하여,

상기 제2 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 제2 어플리케이션마다 산출하고,

상기 제2 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제2 플랫폼 친화도 평균값을 산출하며,

상기 제2 어플리케이션 중에서 상기 제2 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제3 어플리케이션에 상기 플랫폼에 속한 상기 자원을 할당하는

자원 할당 장치.

청구항 11

제 5 항에 있어서,

상기 자원 할당부는,

상기 상호 간섭도가 가장 높은 어플리케이션을 포함하는 조합 중에서 상기 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을 함께 실행될 어플리케이션으로 선택하는

자원 할당 장치.

청구항 12

자원 할당 장치에 의하여 수행되는 자원 할당 방법으로써,

(a) 어플리케이션이 플랫폼(platform)에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출하는 단계;

(b) 상기 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 상기 플랫폼에서 실행될 때, 상기 어플리케이션이 상기 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출하는 단계;

(c) 상기 플랫폼 친화도를 기초로 상기 플랫폼에 포함된 자원을 상기 어플리케이션에 할당하는 단계; 및

(d) 상기 상호 간섭도를 기초로 상기 적어도 하나 이상의 어플리케이션 중에서 상기 어플리케이션과 함께 실행될 어플리케이션을 선택하는 단계를 포함하며,

상기 플랫폼은 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)에 포함되는 복수의 플랫폼 중 어느 하나이고, 상기 복수의 플랫폼 각각은 서로 상이하며,

상기 (c) 단계는,

(c-1) 상기 복수의 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제1 플랫폼 친화도 평균값을 산출하는 단계; 및

(c-2) 상기 복수의 어플리케이션 중에서 상기 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션에 상기 플랫폼에 속한 상기 자원을 할당하는 단계를 포함하는

자원 할당 방법.

청구항 13

삭제

청구항 14

제 12 항에 있어서,

상기 (a) 단계는,

상기 어플리케이션이 상기 복수의 플랫폼 각각에서 실행될 때의 실행 시간을 나타내는 비교 실행 시간과 상기 어플리케이션이 상기 플랫폼 친화도의 산출 대상이 되는 제1 플랫폼에서 실행될 때의 실행 시간을 나타내는 기준 실행 시간을 기초로 상기 플랫폼 친화도를 산출하는

자원 할당 방법.

청구항 15

제 14 항에 있어서,

상기 (a) 단계는,

상기 복수의 플랫폼 각각의 노드(node)에서 실행되는 상기 어플리케이션에 포함된 작업(task)의 수가 1개인 것을 기준으로 상기 비교 실행 시간을 산출하고,

상기 제1 플랫폼의 노드에서 실행되는 상기 어플리케이션에 포함된 작업의 수가 1개인 것을 기준으로 상기 기준 실행 시간을 산출하는

자원 할당 방법.

청구항 16

제 12 항에 있어서,

상기 적어도 하나 이상의 어플리케이션과 상기 어플리케이션으로 구성된 조합이 복수 개일 때,

상기 (b) 단계는,

상기 조합 중 어느 하나가 상기 플랫폼에서 실행될 때의 실행 시간을 나타내는 동시 실행 시간과 상기 어플리케이션이 상기 플랫폼에서 단독으로 실행될 때의 실행 시간을 나타내는 단독 실행 시간을 기초로, 상기 플랫폼에서의 상기 조합의 상호 간섭도를 나타내는 조합 상호 간섭도를 산출하는 단계; 및

상기 복수의 조합마다 산출된 조합 상호 간섭도의 평균값을 상기 상호 간섭도로 산출하는 단계를 포함하는

자원 할당 방법.

청구항 17

삭제

청구항 18

제 12 항에 있어서,

상기 (c-2) 단계는,

상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 제1 어플리케이션에 우선적으로 자원을 할당하는

자원 할당 방법.

청구항 19

제 18 항에 있어서,

상기 (c-2) 단계는,

상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 제1 어플리케이션에 우선적으로 자원을 할당함에 있어서,

상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당하는

자원 할당 방법.

청구항 20

제 19 항에 있어서,

상기 (c-2) 단계는,

상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당함에 있어서,

상기 플랫폼의 자원을 상기 제1 어플리케이션 각각에 동일한 수로 할당하되, 상기 제1 어플리케이션 각각에 할당 가능한 자원의 최대 수는 상기 제1 어플리케이션 각각에 포함된 작업의 수인 것을 특징으로 하는

자원 할당 방법.

청구항 21

제 12 항에 있어서,

상기 (c) 단계는,

상기 복수의 어플리케이션 중에서 상기 자원이 할당되지 않은 작업을 포함하는 적어도 하나 이상의 제2 어플리케이션에 대하여,

상기 제2 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 제2 어플리케이션마다 산출하는 단계;

상기 제2 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제2 플랫폼 친화도 평균값을 산출하는 단계; 및

상기 제2 어플리케이션 중에서 상기 제2 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제3 어플리케이션에 상기 플랫폼에 속한 상기 자원을 할당하는 단계를 더 포함하는

자원 할당 방법.

청구항 22

제 16 항에 있어서,

상기 (d) 단계는,

상기 상호 간섭도가 가장 높은 어플리케이션을 포함하는 조합 중에서 상기 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을 함께 실행될 어플리케이션으로 선택하는

자원 할당 방법.

청구항 23

제 12 항의 자원 할당 방법의 각 단계를 수행하는 컴퓨터 프로그램이 기록된 컴퓨터로 읽을 수 있는 기록매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 어플리케이션(또는 응용, application)에 자원(resource)을 할당하는 장치 및 방법에 관한 것이다. 보다 자세하게는 복수의 플랫폼으로 구성된 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)에서, 각 플랫폼(platform)에 대한 어플리케이션의 친화 정도를 나타내는 플랫폼 친화도(platform affinity) 및 어플리케이션 상호 간의 간섭(interference) 정도를 나타내는 상호 간섭도(co-runner affinity)를 기초로 어플리케이션에 플랫폼의 자원을 할당하는 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 최근 천문학, 물리학, 약학 또는 화학 등과 같은 분야에서 많은 수(예를 들면 수만 개에서 수십만 개)의 독립된 작업(task)들을 포함하는 어플리케이션(또는 응용, application)이 등장하고 있다. 이러한 어플리케이션에 속한 작업들 각각은 다양한 실행 시간(execution time) 및 자원 사용 패턴(resource usage pattern)을 갖는다.

[0003] 전술한 어플리케이션이 복수로 존재하는 경우 이러한 복수의 어플리케이션은 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)에서 실행될 수 있다. 여기서 이종 분산 컴퓨팅 플랫폼이란 예를 들면 슈퍼컴퓨터, 그리드(grid) 또는 클라우드 등과 같은 다양한 종류의 플랫폼으로 구성된 플랫폼을 지칭한다.

[0004] 그런데, 복수의 어플리케이션이 이종 분산 컴퓨팅 플랫폼에서 실행될 때의 성능은 복수의 어플리케이션의 작업에 이종 분산 컴퓨팅 플랫폼의 자원이 할당되는 방법에 영향을 받을 수 있다. 이는 전술한 바와 같이 어플리케이션에 속한 작업들 각각이 다양한 실행 시간과 자원 사용 패턴을 갖기 때문이며, 또한 이종 분산 컴퓨팅 플랫폼을 구성하는 플랫폼 각각의 하드웨어와 시스템 소프트웨어 스택(system software stack), 미들웨어, 네트워크 또는 스토리지 설정 등이 서로 상이하기 때문이다.

[0005] 이를 기초로 살펴보면 어플리케이션의 실행 성능은 해당 어플리케이션이 실행되는 플랫폼의 특성에 의하여 영향받을 수 있다. 또한, 어플리케이션의 실행 성능은 해당 어플리케이션과 함께 실행되는 어플리케이션으로부터 영향(간섭, interference) 받을 수도 있다. 이와 관련하여, 특허문헌 1은 멀티 코어 시스템에서 각 코어 및 각 어플리케이션의 특성을 고려하여 작업을 코어에 할당하고 스케쥴링하며 관리하는 장치 및 방법에 관하여 개시하고 있다.

[0006] 그러나, 이러한 특허문헌 1을 포함한 종래의 기술들은 어플리케이션이 실행되는 플랫폼의 특성, 그리고 해당 어플리케이션과 함께 실행되는 어플리케이션으로부터 받는 영향을 고려하여 자원을 작업에 할당하는 기술에 대해서는 개시하지 않는다.

[0007] 따라서, 복수의 어플리케이션이 이종 분산 컴퓨팅 플랫폼에서 실행될 때 이종 분산 컴퓨팅 플랫폼에 속한 플랫폼의 자원을 보다 효과적으로 어플리케이션에 할당하는 기술에 대한 요구가 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 한국특허공개공보 2012-0017294 , 공개일자 2012년 02월 28일

발명의 내용

해결하려는 과제

[0009] 본 발명의 실시예들은 이종 분산 컴퓨팅 플랫폼의 자원을 복수의 어플리케이션의 작업에 할당함에 있어서 어플리케이션이 실행되는 플랫폼의 특성, 그리고 어플리케이션이 해당 어플리케이션과 함께 실행되는 어플리케이션으로부터 받는 영향을 나타내는(representing)하는 메트릭(metric)에 관하여 제안하고자 한다.

[0010] 또한, 본 발명의 실시예들은 전술한 메트릭을 기초로 이종 분산 컴퓨팅 플랫폼의 자원을 복수의 어플리케이션의 작업에 할당하는 장치 및 방법에 관하여 제안하고자 한다.

[0011] 다만, 본 발명의 해결하고자 하는 과제는 이상에서 언급한 것으로 제한되지 않으며, 언급되지 않은 또 다른 해결하고자 하는 과제는 아래의 기재로부터 본 발명이 속하는 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0012] 본 발명의 일 실시예에 따른 자원 할당 장치는 어플리케이션(application)이 플랫폼(platform)에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출하는 플랫폼 친화도 산출부; 상기 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 상기 플랫폼에서 실행될 때, 상기 어플리케이션이 상기 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출하는 상호 간섭도 산출부; 및 상기 플랫폼 친화도를 기초로 상기 플랫폼에 포함된 자원을 상기 어플리케이션에 할당하고, 상기 상호 간섭도를 기초로 상기 적어도 하나 이상의 어플리케이션 중에서 상기 어플리케이션과 함께 실행될 어플리케이션을 선택하는 자원 할당부를 포함한다.

[0013] 또한, 상기 플랫폼은 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)에 포함되는 복수의 플랫폼 중 어느 하나이고, 상기 복수의 플랫폼 각각은 서로 상이한 것을 특징으로 할 수 있다.

[0014] 또한, 상기 플랫폼 친화도 산출부는 상기 어플리케이션이 상기 복수의 플랫폼 각각에서 실행될 때의 실행 시간을 나타내는 비교 실행 시간과 상기 어플리케이션이 상기 플랫폼 친화도의 산출 대상이 되는 제1 플랫폼에서 실행될 때의 실행 시간을 나타내는 기준 실행 시간을 기초로 상기 플랫폼 친화도를 산출할 수 있다.

[0015] 또한, 상기 비교 실행 시간은 상기 복수의 플랫폼 각각의 노드(node)에서 실행되는 상기 어플리케이션에 포함된 작업(task)의 수가 1개인 것을 기준으로 산출되고, 상기 기준 실행 시간은 상기 제1 플랫폼의 노드에서 실행되는 상기 어플리케이션에 포함된 작업의 수가 1개인 것을 기준으로 산출될 수 있다.

[0016] 또한, 상기 상호 간섭도 산출부는 상기 적어도 하나 이상의 어플리케이션과 상기 어플리케이션으로 구성된 조합이 복수 개일 때, 상기 조합 중 어느 하나가 상기 플랫폼에서 실행될 때의 실행 시간을 나타내는 동시 실행 시간과 상기 어플리케이션이 상기 플랫폼에서 단독으로 실행될 때의 실행 시간을 나타내는 단독 실행 시간을 기초로, 상기 플랫폼에서의 상기 조합의 상호 간섭도를 나타내는 조합 상호 간섭도를 산출하고, 상기 복수의 조합마다 산출된 조합 상호 간섭도의 평균값을 상기 상호 간섭도로 산출할 수 있다.

[0017] 또한, 상기 플랫폼 친화도 산출부는 상기 어플리케이션을 포함하는 복수의 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 복수의 어플리케이션마다 산출하고, 상기 자원 할당부는 상기 복수의 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제1 플랫폼 친화도 평균값을 산출하며, 상기 복수의 어플리케이션 중에서 상기 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션에 상기 자원을 할당할 수 있다.

[0018] 또한, 상기 자원 할당부는 상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 어플리케이션에 우선적으로 상기 자원을 할당할 수 있다.

[0019] 또한, 상기 자원 할당부는 상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 어플리케이션에 우선적으로 상기 자원을 할당함에 있어서, 상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당할 수 있다.

[0020] 또한, 상기 자원 할당부는 상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당함에 있어서, 상기 플랫폼의 자원을 상기 제1 어플리케이션 각각에 동일한 수로 할당하되, 상기 제1 어플리케이션 각각에 할당 가능한 자원의 최대 수는 상기 제1 어플리케이션 각각에 포함된 작업의 수인 것을 특징으로 할 수 있다.

[0021] 또한, 상기 자원 할당부는 상기 복수의 어플리케이션 중에서 상기 자원이 할당되지 작업을 포함하는 적어도 하나 이상의 제2 어플리케이션에 대하여, 상기 제2 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 제2 어플리케이션마다 산출하고, 상기 제2 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제2 플랫폼 친화도 평균값을 산출하며, 상기 제2 어플리케이션 중에서 상기 제2 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제3 어플리케이션에 상기 플랫폼에 속한 상기 자원을 할당할 수 있다.

[0022] 또한, 상기 자원 할당부는 상기 상호 간섭도가 가장 높은 어플리케이션을 포함하는 조합 중에서 상기 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을 함께 실행될 어플리케이션으로 선택할 수 있다.

[0023] 본 발명의 다른 실시예에 따른 자원 할당 장치에 의하여 수행되는 자원 할당 방법으로써, (a) 어플리케이션이 플랫폼(platform)에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출하는 단계; (b) 상기 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 상기 플랫폼에서 실행될 때, 상기 어플리케이션이 상기 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출하는 단계; (c) 상기 플랫폼 친화도를 기초로 상기 플랫폼에 포함된 자원을 상기 어플리케이션에 할당하는 단계; 및 (d) 상기 상호 간섭도를 기초로 상기 적어도 하나 이상의 어플리케이션 중에서 상기 어플리케이션과 함께 실행될 어플리케이션을 선택하는 단계를 포함한다.

[0024] 또한, 상기 플랫폼은 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)에 포함되는 복수의 플랫폼 중 어느 하나이고, 상기 복수의 플랫폼 각각은 서로 상이한 것을 특징으로 할 수 있다.

[0025] 또한, 상기 (a) 단계는 상기 어플리케이션이 상기 복수의 플랫폼 각각에서 실행될 때의 실행 시간을 나타내는 비교 실행 시간과 상기 어플리케이션이 상기 플랫폼 친화도의 산출 대상이 되는 제1 플랫폼에서 실행될 때의 실행 시간을 나타내는 기준 실행 시간을 기초로 상기 플랫폼 친화도를 산출할 수 있다.

[0026] 또한, 상기 (a) 단계는 상기 복수의 플랫폼 각각의 노드(node)에서 실행되는 상기 어플리케이션에 포함된 작업(task)의 수가 1개인 것을 기준으로 상기 비교 실행 시간을 산출하고, 상기 제1 플랫폼의 노드에서 실행되는 상기 어플리케이션에 포함된 작업의 수가 1개인 것을 기준으로 상기 기준 실행 시간을 산출할 수 있다.

[0027] 또한, 상기 적어도 하나 이상의 어플리케이션과 상기 어플리케이션으로 구성된 조합이 복수 개일 때, 상기 (b) 단계는 상기 조합 중 어느 하나가 상기 플랫폼에서 실행될 때의 실행 시간을 나타내는 동시 실행 시간과 상기 어플리케이션이 상기 플랫폼에서 단독으로 실행될 때의 실행 시간을 나타내는 단독 실행 시간을 기초로, 상기 플랫폼에서의 상기 조합의 상호 간섭도를 나타내는 조합 상호 간섭도를 산출하는 단계; 및 상기 복수의 조합마다 산출된 조합 상호 간섭도의 평균값을 상기 상호 간섭도로 산출하는 단계를 포함할 수 있다.

[0028] 또한, 상기 (a) 단계는 상기 어플리케이션을 포함하는 복수의 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 복수의 어플리케이션마다 산출하고, 상기 (c) 단계는 (e) 상기 복수의 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제1 플랫폼 친화도 평균값을 산출하는 단계; 및 (f)상기 복수의 어플리케이션 중에서 상기 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션에 상기 플랫폼에 속한 상기 자원을 할당하는 단계를 포함할 수 있다.

[0029] 또한, 상기 (f) 단계는 상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 어플리케이션에 우선적으로 자원을 할당할 수 있다.

[0030] 또한, 상기 (f) 단계는 상기 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 어플리케이션에 우선적으로 자원을 할당함에 있어서, 상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당할 수 있다.

[0031] 또한, 상기 (f) 단계는 상기 제1 어플리케이션의 상기 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당함에 있어서, 상기 플랫폼의 자원을 상기 제1 어플리케이션 각각에 동일한 수로 할당하되, 상기 제1 어플리케이션 각각에 할당 가능한 자원의 최대 수는 상기 제1 어플리케이션 각각에 포함된 작업의 수인 것을 특징으로 할 수 있다.

[0032] 또한, 상기 (c) 단계는 상기 복수의 어플리케이션 중에서 상기 자원이 할당되지 작업을 포함하는 적어도 하나

이상의 제2 어플리케이션에 대하여, 상기 제2 어플리케이션의 상기 플랫폼에 대한 플랫폼 친화도를 상기 제2 어플리케이션마다 산출하는 단계; 상기 제2 어플리케이션마다 산출된 상기 플랫폼 친화도의 평균값을 나타내는 제2 플랫폼 친화도 평균값을 산출하는 단계; 및 상기 제2 어플리케이션 중에서 상기 제2 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제3 어플리케이션에 상기 플랫폼에 속한 상기 자원을 할당하는 단계를 포함할 수 있다.

[0033] 또한, 상기 (d) 단계는 상기 상호 간섭도가 가장 높은 어플리케이션을 포함하는 조합 중에서 상기 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을 함께 실행될 어플리케이션으로 선택할 수 있다.

[0034] 본 발명의 일 실시예에 따르면, 컴퓨터 프로그램이 기록된 기록매체에서 자원 할당 방법이 수행될 수 있다.

발명의 효과

[0035] 본 발명의 실시예들에 따르면, 어플리케이션의 각 플랫폼(platform)에 대한 친화 정도 및 어플리케이션 상호 간의 간섭 정도를 나타내는 메트릭을 제안할 수 있다. 아울러, 이러한 메트릭을 기초로 복수의 어플리케이션에 이종 분산 컴퓨팅 플랫폼의 자원을 할당할 수 있다. 따라서, 이종 분산 컴퓨팅 플랫폼에서 실행되는 어플리케이션의 성능을 개선시킬 수 있다.

도면의 간단한 설명

[0036] 도 1은 본 발명의 일 실시예에 따른 자원 분배 장치가 이종 분산 컴퓨팅 플랫폼의 자원을 복수의 어플리케이션에 할당하는 개념을 예시적으로 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 자원 분배 장치의 구성을 예시적으로 도시한 도면이다.

도 3은 본 발명의 일 실시예에 따른 어플리케이션을 포함하는 조합에 대하여 예시적으로 도시한 도면이다.

도 4는 본 발명의 일 실시예에 따라서 어플리케이션에 자원을 할당하는 과정을 예시적으로 도시한 도면이다.

도 5는 본 발명의 일 실시예에 따른 자원 분배 방법의 과정을 예시적으로 도시한 도면이다.

도 6은 본 발명의 일 실시예에 따른 상호 간섭도를 산출하는 과정을 예시적으로 도시한 도면이다.

도 7은 본 발명의 일 실시예에 따른 자원을 어플리케이션에 할당하는 과정을 예시적으로 도시한 도면이다.

도 8은 본 발명의 일 실시예에 따른 플랫폼의 노드에서 함께 실행될 어플리케이션을 지정하는 과정을 예시적으로 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0037] 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 서로 다른 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 발명은 청구항의 범주에 의해 정의될 뿐이다.

[0038] 본 발명의 실시예들을 설명함에 있어서 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 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.

[0039] 도 1은 본 발명의 일 실시예에 따른 자원 분배 장치가 이종 분산 컴퓨팅 플랫폼의 자원을 복수의 어플리케이션에 할당하는 개념을 예시적으로 도시한 도면이다.

[0040] 도 1을 참조하면, 본 발명의 일 실시예에 따른 자원 분배 장치(1000)는 이종 분산 컴퓨팅 플랫폼(heterogeneous distributed computing platform)(20)의 자원을 복수의 어플리케이션(또는 응용, application)(30)에 할당한다.

[0041] 여기서, 이종 분산 컴퓨팅 플랫폼(20)은 복수의 플랫폼(플랫폼1, 플랫폼 2 등)으로 구성된 시스템을 지칭한다. 이종 분산 컴퓨팅 플랫폼(20)을 구성하는 플랫폼은 복수의 슈퍼 컴퓨팅 센터(supercomputing center), 단일 컴퓨팅 플랫폼(single computing platform) 또는 클라우드 컴퓨팅 시스템(cloud computing system) 중 적어도 하

나 이상의 조합으로 구성될 수 있는데, 다만 이는 예시적인 것이므로 이에 한정되는 것은 아니다.

[0042] 각 플랫폼은 적어도 하나 이상의 노드(node)(노드1, 노드 2 등)를 포함하고, 각 노드는 적어도 하나 이상의 코어(core)(코어1, 코어 2 등)를 포함한다. 이 때, 본 발명의 일 실시예에서 코어는 어플리케이션에 할당되는 자원의 한 종류를 지칭하는 것으로 해석될 수 있다.

[0043] 이종 분산 컴퓨팅 플랫폼(20)에서는 도 1에 도시된 것과 같이 적어도 하나 이상의 어플리케이션(30)이 실행된다. 이러한 어플리케이션은 예를 들면 천문학, 물리학, 약학 또는 화학 등과 같은 다양한 분야의 어플리케이션일 수 있다. 각각의 어플리케이션은 적어도 하나 이상의 작업(작업1, 작업 2 등)을 포함한다. 이 때, 각 어플리케이션에 포함된 작업의 수는 서로 상이할 수 있다.

[0044] 어플리케이션에 포함된 적어도 하나 이상의 작업은 파일럿 잡(pilot job) 방식으로 스케쥴링 되어 실행될 수 있다. 다만, 파일럿 잡 방식에 의하여 어플리케이션의 작업이 스케쥴링 되어 실행되는 것은 이미 공지된 기술이므로 이에 관한 자세한 설명은 생략하기로 한다.

[0045] 한편, 본 발명의 일 실시예에서는 어플리케이션이 이종 분산 컴퓨팅 플랫폼(20)에서 실행되는 것으로 설명하고 있으나 본 발명의 사상이 이에 한정되는 것은 아니다. 예를 들면, 어플리케이션은 서로 동일한 플랫폼을 복수로 포함하는 분산 컴퓨팅 플랫폼에서 실행될 수 있다. 다만 이하에서는 어플리케이션은 이종 분산 컴퓨팅 플랫폼(20)에서 실행되는 것을 전제로 설명하기로 한다.

[0046] 이종 분산 컴퓨팅 플랫폼(20)에서 실행되는 어플리케이션의 성능은 해당 어플리케이션이 실행되는 플랫폼의 특성(플랫폼에 대한 어플리케이션의 플랫폼 친화도)에 의하여 영향을 받을 수 있다. 또한, 어플리케이션의 실행 성능은 해당 어플리케이션과 함께 실행되는 어플리케이션으로부터 영향(어플리케이션 상호 간의 간섭)을 받을 수도 있다.

[0047] 이에, 본 발명의 일 실시예에 따른 자원 할당 장치(1000)는 플랫폼에 대한 어플리케이션의 친화 정도를 나타내는 플랫폼 친화도(platform affinity) 및 어플리케이션 상호 간의 간섭(interference) 정도를 나타내는 상호 간섭도(co-runner affinity)를 기초로 어플리케이션에 플랫폼의 자원을 할당한다.

[0048] 이러한 자원 할당 장치(1000)는 최종 사용자 컴퓨터 시스템, 컴퓨터, 네트워크 서버 또는 서버 시스템, 모바일 컴퓨팅 디바이스(예를 들어, PDA(personal digital assistant), 모바일 전화기, 스마트폰, 랩탑, 태블릿 컴퓨터 또는 그와 유사한 것), 소비자 전자 디바이스, 또는 임의의 다른 적합한 전자 디바이스 또는 그들의 임의의 조합과 같은 하나 이상의 범용 네트워크 호스트 머신일 수 있다. 아울러, 자원 할당 장치(1000)는 적어도 일부의 소프트웨어와 하드웨어의 하이브리드 구현 방식으로 프로세서 및 프로세서에 의해 실행되는 명령어들을 저장하는 메모리를 포함하는 전자 디바이스 또는 컴퓨터 프로그램에 의해 선택적으로 활성화 또는 재구성되는 프로그래밍 가능한 머신(machine) 또는 적어도 하나 이상의 가상화된 컴퓨팅 환경(예를 들어 네트워크 컴퓨팅 클라우드 또는 그와 유사한 것)에서 구현될 수 있다. 이하에서는 이러한 자원 할당 장치(1000)에 대하여 보다 자세하게 살펴보기로 한다.

[0049] 도 2는 본 발명의 일 실시예에 따른 자원 분배 장치의 구성을 예시적으로 도시한 도면이다.

[0050] 도 2를 참조하면, 자원 분배 장치(1000)는 플랫폼 친화도 산출부(100), 상호 간섭도 산출부(200) 및 자원 할당부(300)를 포함한다. 다만, 도 2는 자원 분배 장치(1000)의 구성을 예시적으로 도시한 것이다. 따라서, 자원 분배 장치(1000)는 이외에 다른 구성을 더 포함할 수 있고 이와 달리 이 중에서 적어도 하나 이상의 구성을 포함하지 않을 수도 있다.

[0051] 플랫폼 친화도 산출부(100)는 어플리케이션이 플랫폼에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출한다. 이러한 플랫폼 친화도 산출부(100)는 예를 들면 마이크로프로세서를 포함하는 연산 장치에서 구현될 수 있다.

[0052] 구체적으로 살펴보면, 플랫폼 친화도 산출부(100)는 어플리케이션이 복수의 플랫폼 각각에서 실행될 때의 실행 시간을 나타내는 비교 실행 시간, 그리고 어플리케이션이 플랫폼 친화도의 산출 대상이 되는 제1 플랫폼에서 실행될 때의 실행 시간을 나타내는 기준 실행 시간을 기초로 플랫폼 친화도를 산출할 수 있다.

[0053] 이하에서는 플랫폼 친화도를 산출하는 방법에 대한 예시를 살펴보기로 한다. 다만, 이는 예시에 불과한 것이므로 본 발명의 사상이 이에 한정되는 것은 아니다.

[0054] 플랫폼 친화도의 산출 대상이 되는 어플리케이션과 플랫폼을 각각 어플리케이션 K 및 제1 플랫폼이라고 지칭하

고, 제1 플랫폼을 제외한 나머지 플랫폼을 P1, P2 및 P3라고 하자. 이 때, 플랫폼 친화도 산출부(100)는 제1 플랫폼 및 P1, P2, P3에서 K에 포함된 작업들이 실행되는 실행 시간의 평균값을 각각의 플랫폼마다 산출할 수 있다. 여기서, 제1 플랫폼의 실행 시간 또는 이러한 실행 시간의 평균값은 기준 실행 시간으로 지칭될 수 있다. 아울러, 나머지 플랫폼인 P1, P2 및 P3 각각에 대한 실행 시간 또는 이러한 실행 시간의 평균값은 비교 실행 시간으로 지칭될 수 있다. 다음으로, 플랫폼 친화도 산출부(100)는 P1, P2 및 P3의 각 비교 실행 시간을 기준 실행 시간으로 정규화한 값을 산출할 수 있다. 이 후, 플랫폼 친화도 산출부(100)는 정규화한 값에 대한 평균값을 산출한 뒤, 이러한 평균값을 K의 제1 플랫폼에 대한 플랫폼 친화도로 산출할 수 있다.

[0055] 예를 들면, K의 기준 실행 시간이 2이고 P1, P2 및 P3의 비교 실행 시간이 각각 5,7,9라고 하면, 플랫폼 친화도 산출부(100)는 각 비교 실행 시간을 기준 실행 시간으로 정규화하여 5/2, 7/2, 9/2를 산출할 수 있다. 다음으로, 플랫폼 친화도 산출부(100)는 5/2와 7/2와 9/2를 모두 더한 뒤 3으로 나눠서 평균값인 7/2을 산출하고, 이러한 평균값인 7/2을 K의 제1 플랫폼에 대한 플랫폼 친화도로 산출할 수 있다.

[0056] 이 때, 플랫폼 친화도 산출부(100)는 전술한 비교 실행 시간을 산출함에 있어서, 전술한 나머지 플랫폼의 노드(node)에서 실행되는 작업(task)의 수가 1개인 것을 기준으로 비교 실행 시간을 산출할 수 있다. 아울러, 플랫폼 친화도 산출부(100)는 전술한 기준 실행 시간을 산출함에 있어서, 제1 플랫폼의 노드에서 실행되는 작업의 수가 1개인 것을 기준으로 기준 실행 시간을 산출할 수 있다.

[0057] 한편, 플랫폼 친화도 산출부(100)는 어플리케이션이 복수 개인 경우 각각의 어플리케이션마다 어느 하나의 플랫폼에 대한 플랫폼 친화도를 산출할 수 있다. 아울러, 플랫폼 친화도 산출부(100)는 이종 분산 컴퓨팅 플랫폼(20)에 포함되는 복수의 플랫폼마다 플랫폼 친화도를 산출할 수 있다.

[0058] 상호 간섭도 산출부(200)는 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 특정 플랫폼에서 실행될 때, 어플리케이션이 이러한 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출한다. 이러한 상호 간섭도 산출부(200)는 예를 들면 마이크로프로세서를 포함하는 연산 장치에서 구현될 수 있다.

[0059] 보다 구체적으로 살펴보기 위하여 적어도 하나 이상의 어플리케이션 중 일부 및 상호 간섭도의 산출 대상이 되는 어플리케이션 A를 포함하는 복수의 조합을 가정하고, 어플리케이션 A가 제2 플랫폼에서 실행될 때의 상호 간섭도를 산출하는 것을 가정하자. 이 때, 상호 간섭도 산출부(200)는 이러한 복수의 조합 중 어느 하나인 제1 조합이 제2 플랫폼에서 실행될 때의 실행 시간을 나타내는 동시 실행 시간과 A가 제2 플랫폼에서 단독으로 실행될 때의 실행 시간을 나타내는 단독 실행 시간을 기초로, 제2 플랫폼에서의 제1 조합의 상호 간섭도를 나타내는 조합 상호 간섭도를 산출할 수 있다. 다음으로, 상호 간섭도 산출부(200)는 복수의 조합마다 산출된 조합 상호 간섭도의 평균값을 산출할 수 있다. 이 후, 상호 간섭도 산출부(200)는 조합 상호 간섭도의 평균값을 제1 조합에 포함된 A의 제2 플랫폼에서의 상호 간섭도로 산출할 수 있다.

[0060] 이하에서는 상호 간섭도를 산출하는 방법에 대한 예시를 살펴보기로 한다. 다만, 이하에서의 설명은 예시에 불과한 것이므로 본 발명의 사상이 이에 한정되는 것은 아니다.

[0061] 도 3은 상호 간섭도의 산출 대상이 되는 어플리케이션 A를 포함하는 제1 조합을 예시적으로 도시한 도면이다. 도 3을 참조하면, 제1 조합은 A를 포함하여 2개의 어플리케이션을 갖는 구성 {(A,B), (A,C), (A,D), (A,E)} 4개와, A를 포함하여 4개의 어플리케이션을 갖는 구성 {(A,B,C,D), (A,B,C,E), (A,B,D,E), (A,C,D,E)} 4개를 포함한다. 상호 간섭도 산출부(200)는 제1 조합의 8개의 구성 각각에 대한 동시 실행 시간에서 단독 실행 시간을 뺀 차이값을 산출할 수 있다. 다음으로, 상호 간섭도 산출부(200)는 이러한 차이값을 단독 실행 시간으로 정규화한 값을 산출할 수 있다. 이 후, 상호 간섭도 산출부(200)는 제1 조합에 포함된 각 구성에 대하여 정규화한 값 전체에 대한 평균값을 산출한 뒤, 이러한 평균값을 어플리케이션 A의 제2 플랫폼에서의 상호 간섭도로 산출할 수 있다.

[0062] 예를 들면, 제1 조합의 8개의 구성 각각에 대한 동시 실행 시간이 각각 5,6,7,8,9,10,11,12이고 A의 단독 실행 시간이 3인 경우, 상호 간섭도 산출부(200)는 각 동시 실행 시간에서 단독 실행 시간을 뺀 뒤에 단독 실행 시간으로 정규화하여 각각 2/3,1,4/3,5/3,2,7/3,8/3,3를 산출할 수 있다. 다음으로, 상호 간섭도 산출부(200)는 2/3,1,4/3,5/3,2,7/3,8/3,3를 모두 더한 뒤 8로 나눠서 평균값인 11/6을 산출하고, 이러한 평균값인 11/6를 A의 제2 플랫폼에서의 상호 간섭도로 산출할 수 있다.

[0063] 한편, 상호 간섭도 산출부(200)는 동시 실행 시간에서 단독 실행 시간을 뺀 차이값이 0보다 작은 값인 경우, 차이값을 0으로 변경한 뒤 상호 간섭도를 산출할 수 있다. 일반적으로 복수의 어플리케이션을 실행할 때의 동시

실행 시간은 한 개의 어플리케이션을 단독으로 실행할 때의 단독 실행 시간보다 길다. 따라서, 동시 실행 시간에서 단독 실행 시간을 뺀 차이값은 0보다 작은 값이 발생하지 않는다. 그러나, 예를 들면 이종 분산 컴퓨팅 플랫폼(20)에 오류가 발생하는 등의 특수한 상황이 발생한다면 차이값은 0보다 작은 값이 발생할 수 있다. 따라서, 차이값을 0으로 변경함으로써 이러한 특수한 상황에서 발생하는 차이값에 대한 보정이 가능하다.

[0064] 다시 도 2를 참조하면, 자원 할당부(300)는 플랫폼 친화도 산출부(100)에 의하여 산출된 플랫폼 친화도를 기초로 플랫폼에 포함된 자원을 어플리케이션에 할당한다. 이러한 자원 할당부(300)는 예를 들면 마이크로프로세서를 포함하는 연산 장치에서 구현될 수 있다.

[0065] 보다 구체적으로 살펴보면, 자원 할당부(300)는 친화도의 평균값을 나타내는 제1 플랫폼 친화도 평균값을 산출할 수 있다. 즉, 플랫폼 친화도 산출부(100)가 복수의 어플리케이션의 플랫폼에 대한 플랫폼 친화도를 각각의 어플리케이션마다 산출하면, 자원 할당부(300)는 복수의 어플리케이션(30)마다 산출된 플랫폼 친화도에 대한 평균값을 산출하여 이를 제1 플랫폼 친화도 평균값으로 지정할 수 있다. 예를 들면, 어플리케이션 A,B,C,D의 플랫폼 P에 대한 플랫폼 친화도가 각각 2,4,6,8인 경우, 플랫폼 P에 대한 제1 플랫폼 친화도 평균값은 5가 된다.

[0066] 이 후, 자원 할당부(300)는 어플리케이션 중에서 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션에 자원을 할당할 수 있다. 위의 예에서 제1 플랫폼 친화도 평균값인 5보다 높은 제1 어플리케이션은 C와 D가 된다.

[0067] 이 때, 자원 할당부(300)는 다음과 같은 3가지 규칙에 따라서 제1 어플리케이션에 자원을 할당할 수 있다.

[0068] 첫째, 자원 할당부(300)는 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 어플리케이션에 우선적으로 자원을 할당할 수 있다. 이를 통해 특정 어플리케이션의 작업의 수를 초과하는 자원은 해당 특정 어플리케이션보다 많은 작업의 수를 갖는 어플리케이션에 할당될 수 있다.

[0069] 둘째, 자원 할당부(300)는 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당할 수 있다.

[0070] 셋째, 자원 할당부(300)는 플랫폼의 자원을 제1 어플리케이션 각각에 동일한 수로 할당하되, 제1 어플리케이션 각각에 할당 가능한 자원의 최대 수는 제1 어플리케이션 각각에 포함된 작업의 수일 수 있다.

[0071] 이 때, 위의 3가지 규칙은 예시적인 것이므로 본 발명의 사상이 이에 한정되는 것은 아니다. 즉, 실시예에 따라서 위의 3가지 규칙 이외의 다른 규칙이 더 고려되거나 또는 이 중 적어도 하나 이상의 규칙이 고려되지 않을 수도 있다. 아울러, 실시예에 따라서 위의 3가지 규칙은 동시에 고려되거나 순차적으로 고려될 수 있는데, 순차적으로 고려될 경우 고려되는 순서는 임의로 변경 가능하다.

[0072] 이하에서는 자원 할당부(300)가 위의 3가지 규칙을 기초로 자원을 할당하는 과정의 예시를 참조하여 살펴보기로 한다. 다만, 이는 예시에 불과한 것이므로 본 발명의 사상이 이에 한정되는 것은 아니다. 도 4에서 가로는 어플리케이션(A,B,C)를 나타낸다. 각 어플리케이션 옆의 괄호 안의 숫자는 각 어플리케이션에 포함된 작업의 수이다. 도 4에서 세로는 플랫폼(X, Y)을 나타낸다. 각 플랫폼 옆의 괄호 안의 숫자는 각 플랫폼의 자원의 수를 나타낸다. 도 4에서 원 안의 숫자는 자원이 할당된 순서를 나타낸다. 도 4에서 어플리케이션 A와 B, C는 모두 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션에 속하는 것을 전제로 한다.

[0073] 어플리케이션 A의 경우 플랫폼 X에 대한 친화도가 플랫폼 Y에 대한 친화도보다 높고, 어플리케이션 B의 경우 플랫폼 X에 대한 친화도가 플랫폼 Y에 대한 친화도보다 높으며, 어플리케이션 C의 경우 플랫폼 X에 대한 친화도가 플랫폼 Y에 대한 친화도보다 낮다고 가정하자.

[0074] 첫째 규칙에 따르면, 자원 할당부(300)는 작업의 수가 적은 어플리케이션에 우선적으로 자원을 할당할 수 있다. 따라서, 자원 할당부(300)는 A->C->B의 순서로 자원을 할당할 수 있다. 이를 기초로 살펴보면, A에 자원이 할당된 순서가 (①,②), C에 자원이 할당된 순서가 (③,④), B에 자원이 할당된 순서가 (⑤,⑥)이다.

[0075] 둘째 규칙에 따르면, 자원 할당부(300)는 어플리케이션의 플랫폼에 대한 친화도가 높은 순서로 자원을 할당할 수 있다. 어플리케이션 A의 경우 플랫폼 X에 먼저 자원이 할당(①)된 후 플랫폼 Y에 자원이 할당(②)되고, 이는 다른 어플리케이션에 대해서도 마찬가지이다.

[0076] 셋째 규칙에 따르면, 자원 할당부(300)는 플랫폼의 자원을 각각의 어플리케이션에 동일한 수로 할당하되 해당 어플리케이션의 작업의 수를 최대 할당 가능한 자원의 수로 한다. 예를 들어 살펴보면, 플랫폼 X의 자원 90개는 어플리케이션 A,B,C에 동일한 수로 할당되어야 한다. 따라서, 어플리케이션 A에는 X의 자원 30개가 할당된

다(①). 다음으로, 플랫폼 Y의 자원은 90개이므로 어플리케이션 A,B,C에 동일한 수로 할당되어야 한다. 그런데, 어플리케이션 A의 작업 수는 플랫폼 X의 자원 30개의 할당으로 인하여 40개에서 10개로 줄었으므로 플랫폼 Y의 자원은 10개만 어플리케이션 A에 할당된다(②).

[0077] 다음으로, 어플리케이션 A보다 작업의 수가 그 다음으로 많은 어플리케이션 C에 자원이 할당될 수 있다. 이 때, 어플리케이션 C는 플랫폼 Y에 대한 친화도가 플랫폼 X에 대한 친화도보다 높다. 플랫폼 Y의 자원 80개(90개였으나 어플리케이션 A에 10개를 이미 할당함)는 어플리케이션 B, C에 동일한 수로 할당되어야 한다. 따라서, 어플리케이션 C에는 Y의 자원 40개가 할당된다(③). 다음으로, 플랫폼 X의 자원 60개(최초에 90개였으나 어플리케이션 A에 30개를 할당하였으므로 60개만 남음)는 어플리케이션 B,C에 동일한 수로 할당되어야 한다. 그런데, 어플리케이션 C의 작업 수는 10개로 줄었으므로 플랫폼 X의 자원은 10개만 할당된다(④).

[0078] 다음으로, 어플리케이션 B에 자원이 할당될 수 있다. 이 때, 어플리케이션 B는 플랫폼 X에 대한 친화도가 플랫폼 Y에 대한 친화도보다 높다. 따라서, 플랫폼 X의 잔여 자원 50개가 먼저 할당된 후 플랫폼 Y의 잔여 자원 40개가 할당된다.

[0079] 한편, 전술한 3가지 규칙에 의하여 자원을 할당할 경우, 포함하고 있는 작업의 수가 상대적으로 많으면서 플랫폼 친화도가 상대적으로 낮은 어플리케이션에 대해서는 자원이 할당되지 않은 작업이 존재할 수 있는데 이하에서는 이러한 어플리케이션을 제2 어플리케이션으로 지칭하기로 한다.

[0080] 자원 할당부(300)는 제2 어플리케이션에 자원을 할당하기 위하여 다음과 같은 절차를 수행할 수 있다. 먼저, 자원 할당부(300)는 제2 어플리케이션의 플랫폼에 대한 플랫폼 친화도를 산출할 수 있다(제2 어플리케이션이 복수인 경우 제2 어플리케이션마다 산출). 다음으로, 자원 할당부(300)는 제2 어플리케이션에 대하여 산출된 플랫폼 친화도의 평균값(제2 어플리케이션이 복수인 경우, 산출된 플랫폼 친화도의 평균값을 의미)을 나타내는 제2 플랫폼 친화도 평균값을 산출할 수 있다. 이 후, 자원 할당부(300)는 제2 어플리케이션 중에서 제2 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제3 어플리케이션에 자원을 할당할 수 있다.

[0081] 이를 통해, 자원 할당부(300)는 포함하고 있는 작업의 수가 상대적으로 많으면서 플랫폼 친화도가 상대적으로 낮은 어플리케이션에도 자원이 할당되도록 할 수 있다.

[0082] 이상에서 살펴본 바와 같이, 본 발명의 일 실시예에 따르면 어플리케이션의 플랫폼에 대한 친화도를 기초로 이종 분산 컴퓨팅 플랫폼의 자원을 어플리케이션에 할당할 수 있으며, 따라서 이종 분산 컴퓨팅 플랫폼에서 보다 높은 성능으로 어플리케이션을 실행시킬 수 있다.

[0083] 다음으로, 자원 할당부(300)는 상호 간섭도 산출부(200)가 산출한 상호 간섭도를 기초로 플랫폼의 노드에서 함께 실행될 어플리케이션을 선택한다.

[0084] 보다 구체적으로 살펴보면, 자원 할당부(300)는 상호 간섭도가 가장 높은 어플리케이션을 포함하는 조합 중에서 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을 함께 실행될 어플리케이션으로 선택할 수 있다.

[0085] 예를 들어 살펴보면, 자원 할당부(300)는 플랫폼 K에서 상호 간섭도가 가장 높은 어플리케이션 A를 선택할 수 있다. 다음으로, 자원 할당부(300)는 어플리케이션 A를 포함하는 조합 중에서 플랫폼 K에 대한 조합 상호 간섭도가 가장 낮은 조합을 선택할 수 있다. 이 후, 자원 할당부(300)는 플랫폼 K에 대한 조합 상호 간섭도가 가장 낮은 조합에 포함된 어플리케이션을 플랫폼 K의 노드에서 함께 실행될 어플리케이션으로 지정할 수 있다.

[0086] 도 3을 예로 들어 살펴보면, 플랫폼 K에서 상호 간섭도가 가장 높은 어플리케이션을 A라고 가정하자. 다음으로, A를 포함하는 도 3에 도시된 8개의 조합 중에서 플랫폼 K에 대한 조합 상호 간섭도가 가장 낮은 조합을 선택해야 하는데, 조합 상호 간섭도가 가장 낮은 조합은 (A,D)라고 가정하자. 그렇다면, 자원 할당부(300)는 조합 상호 간섭도가 가장 낮은 조합 (A,D)에 속한 어플리케이션 A와 어플리케이션 D를 플랫폼 K의 노드에서 함께 실행될 어플리케이션으로 지정할 수 있다.

[0087] 이상에서 살펴본 바와 같이, 본 발명의 일 실시예에 따르면 어플리케이션의 플랫폼에 대한 친화도를 기초로 이종 분산 컴퓨팅 플랫폼의 자원을 어플리케이션에 할당할 수 있으며, 또한 플랫폼의 노드에서 함께 실행될 때 간섭 받는 정도가 최소인 어플리케이션을 선택(지정)할 수 있다. 따라서 이종 분산 컴퓨팅 플랫폼에서 보다 높은 성능으로 어플리케이션이 실행되도록 할 수 있다.

[0088] 도 5는 본 발명의 일 실시예에 따른 자원 분배 방법의 과정을 예시적으로 도시한 도면이다.

[0089] 먼저, 본 발명의 일 실시예에 따른 자원을 할당하는 방법은 자원 할당 장치(1000)에 의하여 수행될 수 있다.

[0090] 도 5를 참조하면, 본 발명의 일 실시예에 따른 자원 분배 방법은 어플리케이션이 플랫폼에서 실행될 때의 성능을 나타내는 플랫폼 친화도를 산출하는 단계(S100), 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 플랫폼에서 실행될 때 어플리케이션이 적어도 하나 이상의 어플리케이션으로부터 간섭받는 정도를 나타내는 상호 간섭도를 산출하는 단계(S200), 플랫폼 친화도를 기초로 플랫폼에 포함된 자원을 어플리케이션에 할당하는 단계(S300), 상호 간섭도를 기초로 적어도 하나 이상의 어플리케이션 중에서 어플리케이션과 함께 실행될 어플리케이션을 선택하는 단계(S400)를 포함한다. 다만, 이는 예시적인 것이므로 자원 분배 방법은 이외에 다른 단계를 더 포함할 수 있고 이와 달리 이 중에서 적어도 하나 이상의 단계를 포함하지 않을 수도 있다. 아울러, 도 5에 도시된 단계는 도 5에 도시된 것과는 상이한 순서로 수행될 수 있다.

[0091] 먼저, 플랫폼 친화도(platform affinity)를 산출하는 단계(S100)는 어플리케이션이 플랫폼에서 실행될 때의 성능(performance)을 나타내는 플랫폼 친화도(platform affinity)를 산출한다. 이러한 플랫폼 친화도를 산출하는 단계(S100)는 도 2에 도시된 플랫폼 친화도 산출부(100)에 의하여 수행될 수 있다.

[0092] 플랫폼 친화도를 산출하는 단계(S100)는 어플리케이션이 복수의 플랫폼 각각에서 실행될 때의 실행 시간을 나타내는 비교 실행 시간, 그리고 어플리케이션이 플랫폼 친화도의 산출 대상이 되는 제1 플랫폼에서 실행될 때의 실행 시간을 나타내는 기준 실행 시간을 기초로 플랫폼 친화도를 산출할 수 있다.

[0093] 보다 구체적으로 살펴보면, 플랫폼 친화도의 산출 대상이 되는 어플리케이션과 플랫폼을 각각 어플리케이션 K 및 제1 플랫폼이라고 지칭하고, 제1 플랫폼을 제외한 나머지 플랫폼을 P1, P2 및 P3라고 지칭하기로 하자. 이 때, 플랫폼 친화도를 산출하는 단계(S100)는 제1 플랫폼 및 P1, P2, P3에서 K에 포함된 작업들이 실행되는 실행 시간의 평균값을 각각의 플랫폼마다 산출할 수 있다. 여기서, 제1 플랫폼의 실행 시간 또는 이러한 실행 시간의 평균값은 기준 실행 시간으로 지칭될 수 있다. 아울러, 나머지 플랫폼인 P1, P2 및 P3 각각에 대한 실행 시간 또는 이러한 실행 시간의 평균값은 비교 실행 시간으로 지칭될 수 있다. 다음으로, 플랫폼 친화도를 산출하는 단계(S100)는 P1, P2 및 P3의 각 비교 실행 시간을 기준 실행 시간으로 정규화한 값을 산출할 수 있다. 이후, 플랫폼 친화도를 산출하는 단계(S100)는 정규화한 값 전체에 대한 평균값을 산출한 뒤, 이러한 평균값을 K의 제1 플랫폼에 대한 플랫폼 친화도로 산출할 수 있다.

[0094] 이 때, 플랫폼 친화도를 산출하는 단계(S100)는 기준 실행 시간 및 비교 실행 시간을 산출함에 있어서 각 플랫폼의 노드에서 실행되는 작업의 수가 1개인 것을 기준으로 산출하는 것은 전술한 바와 같다.

[0095] 한편, 플랫폼 친화도를 산출하는 단계(S100)는 어플리케이션이 복수 개인 경우 각각의 어플리케이션마다 어느 하나의 플랫폼에 대한 플랫폼 친화도를 산출할 수 있다. 아울러, 플랫폼 친화도를 산출하는 단계(S100)는 이종 분산 컴퓨팅 플랫폼(20)에 포함되는 복수의 플랫폼마다 플랫폼 친화도를 산출할 수 있다.

[0096] 상호 간섭도를 산출하는 단계(S200)는 어플리케이션이 적어도 하나 이상의 어플리케이션과 함께 플랫폼에서 실행될 때, 어플리케이션이 이러한 적어도 하나 이상의 어플리케이션으로부터 간섭(interference)받는 정도를 나타내는 상호 간섭도(co-runner affinity)를 산출한다. 이러한 상호 간섭도를 산출하는 단계는 도 2에 도시된 상호 간섭도 산출부(200)에 의하여 수행될 수 있다.

[0097] 보다 구체적으로 살펴보기 위하여, 적어도 하나 이상의 어플리케이션 중 일부의 어플리케이션 및 상호 간섭도의 산출 대상이 되는 어플리케이션 A를 포함하는 복수의 조합을 가정하자. 아울러, 어플리케이션 A가 제2 플랫폼에서 실행될 때의 상호 간섭도를 산출하는 것을 가정하자. 이하에서는 상호 간섭도를 산출하는 과정을 예시적으로 도시하고 있는 도 6을 참조하기로 한다.

[0098] 도 6을 참조하면, 상호 간섭도를 산출하는 단계(S200)는 조합이 제2 플랫폼에서 실행될 때의 실행 시간을 나타내는 동시 실행 시간과 어플리케이션 A가 제2 플랫폼에서 단독으로 실행될 때의 실행 시간을 나타내는 단독 실행 시간의 차이값을 각 조합마다 산출하는 단계(S210), 이러한 차이값을 단독 실행 시간으로 정규화하여 각 조합에 대한 조합 상호 간섭도를 산출하는 단계(S220) 및 조합 상호 간섭도의 평균값을 어플리케이션 A의 제2 플랫폼에서의 상호 간섭도로 산출하는 단계(S230)를 포함할 수 있다.

[0099] 한편, 상호 간섭도를 산출하는 단계(S200)는 동시 실행 시간에서 단독 실행 시간을 뺀 차이값이 0보다 작은 값인 경우, 차이값을 0으로 변경한 뒤 상호 간섭도를 산출할 수 있음은 전술한 바와 같다.

[0100] 다시 도 5를 참조하면, 자원을 어플리케이션에 할당하는 단계(S300)는 플랫폼 친화도를 기초로 플랫폼의 자원을 어플리케이션에 할당한다. 이러한 자원을 어플리케이션에 할당하는 단계(S300)는 자원 할당부(300)에 의하여 수행될 수 있다.

[0101] 보다 구체적으로 살펴보기 위하여 자원을 어플리케이션에 할당하는 과정을 예시적으로 도시하고 있는 도 7을 참조하여 살펴보기로 한다. 도 7을 참조하면, 플랫폼 친화도의 평균값을 나타내는 제1 플랫폼 친화도 평균값을 산출하는 단계(S310)가 수행될 수 있다. 즉, 플랫폼 친화도를 산출하는 단계(S100)에서 복수의 어플리케이션의 플랫폼에 대한 플랫폼 친화도를 각각의 어플리케이션마다 산출하면, 단계 S310에서는 복수의 어플리케이션(30)마다 산출된 플랫폼 친화도에 대한 평균값을 산출하여 이를 제1 플랫폼 친화도 평균값으로 지정할 수 있다.

[0102] 다음으로, 어플리케이션 중에서 제1 플랫폼 친화도 평균값 이상의 플랫폼 친화도를 갖는 제1 어플리케이션을 선택하는 단계(S320)가 수행될 수 있다.

[0103] 이 후, 제1 어플리케이션에 자원을 할당하는 단계(S330)가 수행될 수 있다. 단계 S330은 다음과 같은 3가지 규칙에 따라서 제1 어플리케이션에 자원을 할당할 수 있다.

[0104] 첫째, 제1 어플리케이션 각각에 포함된 작업의 수가 상대적으로 적은 어플리케이션에 우선적으로 자원을 할당할 수 있다. 이를 통해 특정 어플리케이션의 작업의 수를 초과하는 자원은 해당 특정 어플리케이션보다 많은 작업의 수를 갖는 어플리케이션에 할당될 수 있다.

[0105] 둘째, 복수의 플랫폼 각각에 대한 플랫폼 친화도가 상대적으로 높은 플랫폼의 자원을 우선적으로 할당할 수 있다.

[0106] 셋째, 플랫폼의 자원을 제1 어플리케이션 각각에 동일한 수로 할당하되, 제1 어플리케이션 각각에 할당 가능한 자원의 최대 수는 제1 어플리케이션 각각에 포함된 작업의 수일 수 있다.

[0107] 이 때, 위의 3가지 규칙은 예시적인 것이므로 본 발명의 사상이 이에 한정되는 것은 아니다. 즉, 실시예에 따라서 위의 3가지 규칙 이외의 다른 규칙이 더 고려될 수 있으며, 이와 달리 이 중 적어도 하나 이상의 규칙이 고려되지 않을 수도 있다. 아울러, 실시예에 따라서 위의 3가지 규칙은 동시에 고려되거나 순차적으로 고려될 수 있는데, 순차적으로 고려될 경우 그 순서는 3가지 규칙에 대하여 임의로 변경 가능하다.

[0108] 한편, 전술한 3가지 규칙에 의하여 자원을 할당할 경우, 포함하고 있는 작업의 수가 상대적으로 많으면서 플랫폼 친화도가 상대적으로 낮은 어플리케이션의 경우 자원이 할당되지 않은 작업이 존재할 수 있는데, 이런 어플리케이션(제2 어플리케이션)에 자원을 할당하는 방법에 대해서는 이미 기술하였으므로 이에 대한 설명은 생략하기로 한다.

[0109] 자원을 할당하는 단계(S400)는 상호 간섭도를 기초로 플랫폼의 노드에서 함께 실행될 어플리케이션을 선택한다.

[0110] 보다 구체적으로 살펴보기 위하여 도 8을 참조하여 살펴보기로 하자. 도 8을 참조하면, 상호 간섭도가 가장 높은 어플리케이션을 선택하는 단계(S410)가 수행될 수 있다. 다음으로, 상호 간섭도가 가장 높은 어플리케이션을 포함하는 조합 중에서 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을 선택하는 단계(S420)가 수행될 수 있다. 이 후, 조합 상호 간섭도가 가장 낮은 조합에 속하는 어플리케이션을, 플랫폼의 노드에서 함께 실행될 어플리케이션으로 선택하는 단계(S430)가 수행될 수 있다.

[0111] 따라서, 본 발명의 일 실시예에 따르면 플랫폼의 노드에서 함께 실행될 때 간섭 받는 정도가 최소인 어플리케이션을 선택할 수 있다. 따라서 이종 분산 컴퓨팅 플랫폼에서 실행되는 어플리케이션을 성능을 보다 향상시킬 수 있다.

[0112] 이상에서 살펴본 바와 같이, 본 발명의 일 실시예에 따르면 어플리케이션의 각 플랫폼에 대한 친화 정도를 나타내는 플랫폼 친화도와, 어플리케이션 상호 간의 간섭 정도를 나타내는 상호 간섭도를 기초로 복수의 어플리케이션에 이종 분산 컴퓨팅 플랫폼의 자원을 할당할 수 있다. 따라서, 이종 분산 컴퓨팅 플랫폼에서 실행되는 어플리케이션의 성능을 개선시킬 수 있다.

[0113] 아울러, 본 발명의 일 실시예에 따른 자원 할당 방법은 컴퓨터 프로그램이 기록된 기록매체에 구현될 수 있다.

[0114] 본 발명에 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 인

스트럭션들은 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.

[0115] 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.

[0116] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

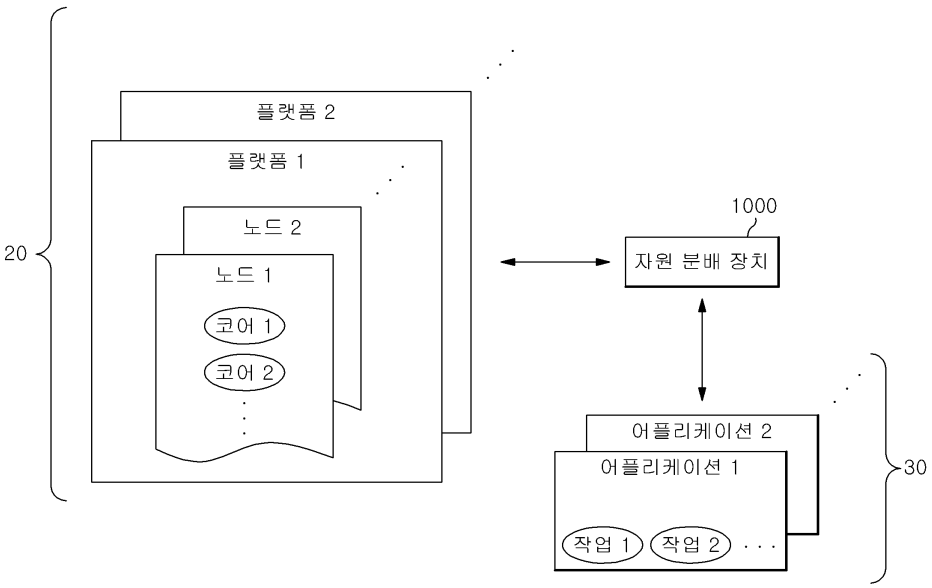
[0117] 20: 이종 분산 컴퓨팅 플랫폼

30: 복수의 어플리케이션

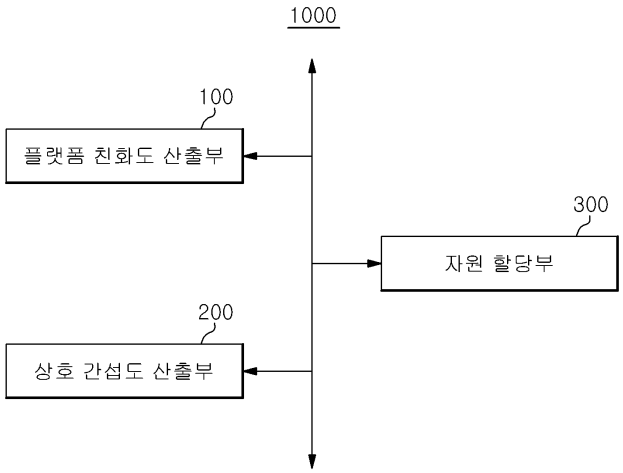
1000: 자원 분배 장치

도면

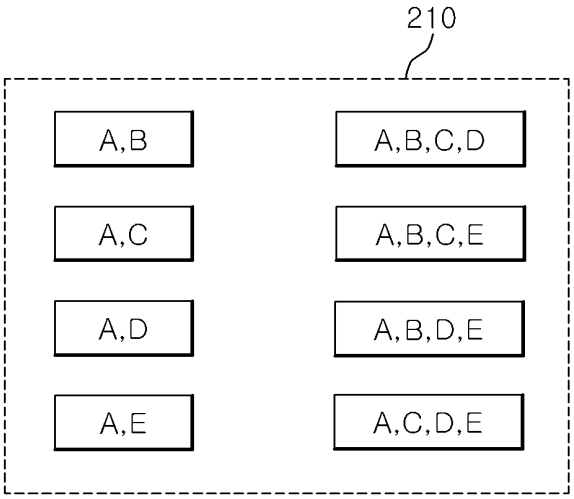
도면1



도면2



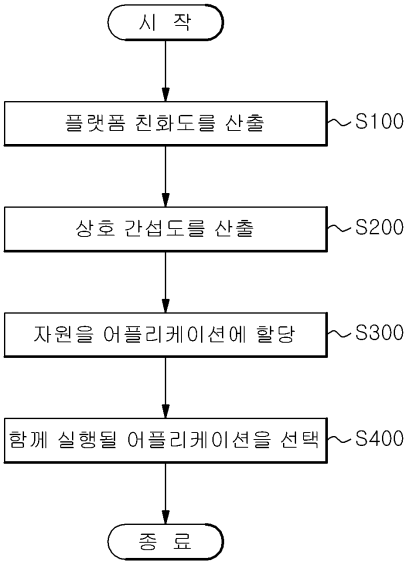
도면3



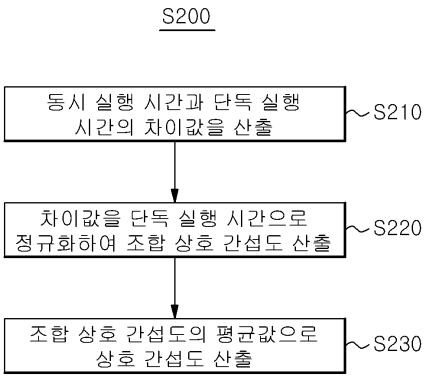
도면4

어플리케이션 / 플랫폼	A(40)	B(1000)	C(50)
X(90)	① 30	⑤ 50	④ 10
Y(90)	② 10	⑥ 40	③ 40

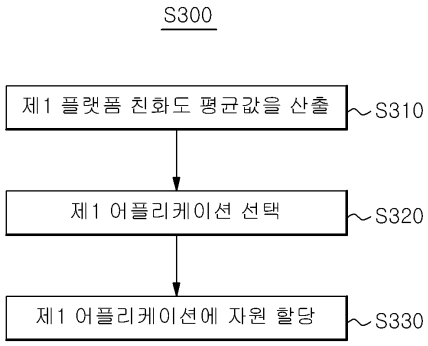
도면5


시 작
플랫폼 친화도를 산출
S100
상호 간섭도를 산출
S200
자원을 어플리케이션에 할당
S300
함께 실행될 어플리케이션을 선택
S400
종 료

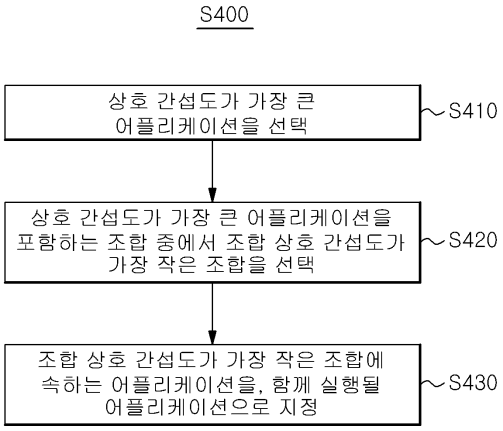
도면6


S200
동시 실행 시간과 단독 실행 시간의 차이값을 산출
S210
차이값을 단독 실행 시간으로 정규화하여 조합 상호 간섭도 산출
S220
조합 상호 간섭도의 평균값으로 상호 간섭도 산출
S230

도면7


S300
제1 플랫폼 친화도 평균값을 산출
S310
제1 어플리케이션 선택
S320
제1 어플리케이션에 자원 할당
S330

도면8


S400
상호 간섭도가 가장 큰 어플리케이션을 선택
S410
상호 간섭도가 가장 큰 어플리케이션을 포함하는 조합 중에서 조합 상호 간섭도가 가장 작은 조합을 선택
S420
조합 상호 간섭도가 가장 작은 조합에 속하는 어플리케이션을, 함께 실행될 어플리케이션으로 지정
S430