



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0017173
(43) 공개일자 2009년02월18일

(51) Int. Cl.

G06F 9/46 (2006.01) G06F 9/00 (2006.01)

(21) 출원번호 10-2007-0081718

(22) 출원일자 2007년08월14일

심사청구일자 2007년08월14일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

이상근

서울 동대문구 제기2동 벽산아파트 105동 1001호

변은정

서울 성북구 안암동 고려대학교 정보통신대학 컴퓨터학과

(뒷면에 계속)

(74) 대리인

김인한, 김희곤

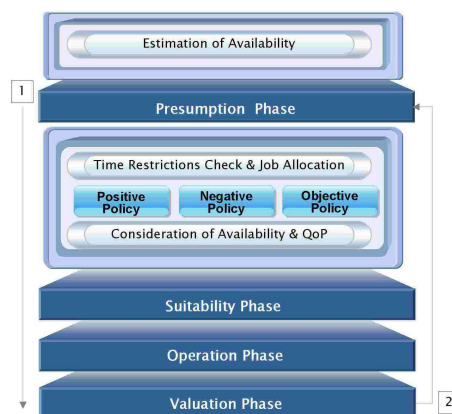
전체 청구항 수 : 총 12 항

(54) 작업 완료 시간 제약에 따라 가용성을 차등적으로 적용하는유연하고 안정적인 스케줄링 시스템 및 방법

(57) 요약

본 발명은 과거의 자원 기여 시간의 비율을 나타낸 기존의 가용성에 양질의 연산 정도를 고려함으로써 자원제공자의 환경에 부합하는 가용성을 새롭게 정의하고, 이를 기반으로 주어진 작업부하의 시간 제약 (Time Restriction, Deadline)에 따라 알맞은 자원제공자를 찾기 위해 각각 가용성을 차등화하여 적용한 스케줄링 기법에 관한 것이다. 이를 위하여 본 발명은 자원 제공자의 양질의 연산 수행 (QoP; Quality of Provision)와 대표견실도 (RC; Representative Confidence)를 이용하여 동적인 자원제공자 컴퓨터 환경에 부합하는 가용성을 추정하는 추정단계(Presumption Phase)와 상기 자원제공자들의 가용성을 기반으로 작업부하량 및 대표견실도의 값을 비교하여 자원제공자를 선정하고 작업부하를 할당하는 적합단계(Suitability Phase)와 상기 적합단계에서 할당된 작업부하를 수행하고 작업부하의 수행이 종료되면 결과값을 반환하는 실행단계(Operation Phase) 및 상기 실행 단계에서 실행된 자원제공자의 연산수행의 유형을 분석하여 예측의 정확성을 평가하는 평가단계(Valuation Phase)를 포함하여 이루어지는 것을 특징으로 한다.

대표도 - 도7



(72) 발명자

최성진

서울 성북구 안암동 고려대학교 정보통신대학 컴퓨
터학과

김홍수

서울 성북구 안암동 고려대학교 정보통신대학 컴퓨
터학과

특허청구의 범위

청구항 1

자원제공자의 양질의 연산 수행 (QoP)와 대표건실도 (RC)를 이용하여 자원제공자 컴퓨팅환경에 부합하는 가용성을 추정하는 추정 모듈과;

상기 추정모듈로부터 추정된 자원 제공자의 가용성, 작업부하량 및 대표건실도 값을 비교하여 자원제공자를 선정하고 작업부하를 할당하는 적합 판단 모듈과;

상기 적합 판단 모듈로부터 할당된 작업 부하를 수행하고 작업부하의 수행이 종료되면 결과값을 반환하는 실행 모듈과;

상기 실행모듈에 의해 수행된 자원제공자의 연산수행의 유형을 분석하여 예측의 정확성을 평가하는 평가모듈을 포함하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

청구항 2

청구항 1에 있어서, 상기 추정모듈은,

양질의 연산 수행 (QoP)와 정확도(PA)를 고려한 하기의 식에 의해 가용성(A_{RP})을 추정하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

$$A_{RP} = \frac{Avg(OnTime)}{Avg(OffTime) + Avg(OnTime)} \cdot QoP \cdot PA, \text{ where } QoP = \frac{Avg(DedicationTime)}{Avg(OnTime)}$$

청구항 3

청구항 1에 있어서, 상기 적합 판단 모듈은,

자원제공자의 가용성을 기준으로 작업부하를 할당하되 각 작업부하에 대해 작업부하의 수행을 위한 요구시간요소(TR ; Time Required)와 상기 자원제공자의 가용성의 평균값($Avg(A_{RP})$)을 비교하여, 상기 요구시간요소(TR)보다 자원제공자의 가용성의 평균값($Avg(A_{RP})$)이 큰 경우($Avg(A_{RP}) > TR$)의 자원제공자를 선정하고 작업부하를 할당하는 포지티브 방식에 의해 수행하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

청구항 4

청구항 1에 있어서, 상기 적합 판단 모듈은,

자원제공자의 가용성의 평균값($Avg(A_{RP})$)에 비견실적인 자원제공자에 대한 패널티인 대표건실도(RC)를 고려한 값이 요구시간요소(TR)보다 큰 경우($Avg(A_{RP}) \cdot RC > TR$)의 자원 제공자를 선정하고 작업부하를 할당하는 네거티브 방식(Negative Policy)에 의해 수행하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

청구항 5

청구항 4에 있어서, 상기 대표건실도 (RC)는,

하기의 식에 의해 산출되는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

$$RC = \sum_{i=1}^n P_i \log P_i \cdot \frac{NCE}{NTE}$$

청구항 6

청구항 1에 있어서, 상기 적합판단모듈은,

최초의 자원 할당시에는 상기 청구항 3에 의한 포지티브 방식에 의해 자원을 선출하여 할당하고, 일정 기준시간(Timeout)이 경과한 후에 작업이 완료되지 않는 경우에는, 청구항 4에 의한 네거티브방식(Negative Policy)에 의해 자원을 선출하여 할당하는 객관적 방식(Objective Policy)에 의해 수행하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 기법.

청구항 7

청구항 1에 있어서, 상기 평가모듈은,

자원제공자의 연산 수행의 유형을 분석하여 하기의 식에 의해 산출된 예측의 정확성(PA)을 평가하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

$$PA = AP / (IP + AP)$$

(여기서, 상기 AP는 정확한 예측의 횟수이며, IP는 부정확한 예측의 횟수이다.)

청구항 8

대표견실도와 상태전이 확률을 이용하여 자원제공자 컴퓨팅 환경에 부합하는 가용성을 추정하는 추정단계(Presumption Phase);

상기 자원제공자들의 가용성을 기반으로 작업부하량 및 대표견실도의 값을 비교하여 자원제공자를 선정하고 작업부하를 할당하는 적합단계(Suitability Phase);

상기 적합단계에서 할당된 작업부하를 수행하고 작업부하의 수행이 종료되면 결과값을 반환하는 실행단계(Operation Phase); 및

상기 실행단계에서 실행된 자원제공자의 연산수행의 유형을 분석하여 예측의 정확성을 평가하는 평가단계(Valuation Phase);를 포함하여 이루어지는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법.

청구항 9

청구항 8에 있어서,

상기 추정단계(Presumption Phase)는 청구항 2의 식에 의해 도출된 값에 의해 가용성을 추정하는 단계인 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법.

청구항 10

청구항 9에 있어서, 상기 적합단계(Suitability Phase)는,

자원제공자의 가용성을 기준으로 작업부하를 할당하되 각 작업부하에 대해 작업부하의 수행을 위한 요구시간요소(TR)와 상기 자원제공자의 가용성의 평균값을 비교하여,

상기 요구시간요소보다 자원제공자의 가용성의 평균값이 큰 경우의 자원제공자를 선정하는 포지티브 방식(Positive Policy)에 의해 수행되는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법.

청구항 11

청구항 10에 있어서, 상기 적합단계(Suitability Phase)는,

상기 자원제공자의 가용성을 기준으로 작업부하를 할당함에 있어서, 견실하지 않은 자원제공자에 대한 패널티값인 대표견실도(RC; Representative Confidence)를 고려한 평균 가용성 값이 요구시간요소(TR)보다 더 큰 자원제공자를 선정하는 네거티브 방식(Negative Policy)을 더 포함하여 이루어지는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법.

청구항 12

청구항 11에 있어서, 상기 적합단계(Suitability Phase)는,

최초의 자원할당시에는 상기 포저티브 방식(Positive Policy)에 의해 자원을 선출하여 할당하고, 일정 기준시간(Timeout)이 경과한 후에 작업이 완료되지 않는 경우에는 상기 네거티브 방식(Negative Policy)을 이용하여 자원제공자를 선출하여 할당하는 객관적 방식(Objective Policy)을 더 포함하여 이루어지는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 시스템.

명 세 서

발명의 상세한 설명

기술 분야

- <1> 본 발명에서는 자원제공자의 연산 수행 패턴, 즉 가용성을 예측하기 위해 기존의 연구들이 사용하였던 통계적 모델에서 추출된 통계치를 사용하지 않고, 작업 완료 시간의 제약에 따라 자원제공자의 가용성을 그 신뢰 정도 즉, 양질의 자원 제공과 함께 차등적으로 적용함으로써, 보다 정확한 자원제공자의 가용성을 제공하고, 자원제공자의 가용성에 대표 견실도를 고려함으로써, 작업부하(Workload)의 실행의 신뢰성을 향상시키는 스케줄링 시스템에 관한 것이다. 이로써 연산 실패율을 저하시키는 물론 연산 실패로 인한 지연 시간을 감소시키고, 연산의 완료성을 향상시킨다. 자원제공자의 가용성을 보다 정확하게 분석하여 작업부하를 할당하므로 참여와 이탈이 자유로운 자원제공자 컴퓨팅의 동적 상태에 능동적으로 대처하는 안정성이 높은 자원제공자컴퓨팅 시스템을 구축하도록 지원한다.

배 경 기 술

- <2> 자원제공자 컴퓨팅 환경에서는 자원제공자의 자율적 연산 참여와 탈퇴를 허용하기 때문에 기존 자원제공자 컴퓨팅 시스템들에서는 연산이 수행 도중 임의로 중단되거나, 예고 없이 재수행되는 등 불안정적인 연산 수행 양상을 보여왔다. 이러한 연산 도중의 비예측적이고 잦은 중단으로 인해 연산의 완료 시간이 지연되는 것은 물론, 연산 수행의 신뢰성이 저하되며 연산의 완료를 보장하지 못하고 있다. 기존의 자원제공자 컴퓨팅 시스템들은 이러한 자원제공자의 동적 특성을 반영하지 못하고 있을 뿐만 아니라 자원제공자의 연산 수행 양식을 고려하지 않아 시스템의 안정성과 성능이 저하되었다. 신뢰성 높은 연산 수행을 지원하고, 자원제공자의 비예측적 연산 수행 속성을 극복하기 위해서는 자원제공자의 동적인 특성에 부합하는 보다 정확한 가용성(Availability)을 고려한 스케줄링이 필요하다.
- <3> 기존의 자원제공자 컴퓨팅 시스템들의 스케줄링 기법은 크게 FCFS (First Come First Serve) 스케줄링 기법, 선형처리자 우선할당 스케줄링 (Eager Scheduling) 기법, 자원제공자의 성능과 연산 참여 시간을 고려한 스케줄링 기법, 확률적 스케줄링 (Stochastic Scheduling) 기법 등 네 가지로 분류된다.
- <4> 첫 째, FCFS 기법은 먼저 온 자원제공자에 먼저 작업부하를 할당하는 방식으로 XtremWeb, Korea@Home등에서 이용되었다. XtremWeb에서는 단순한 FCFS 기법을 기반으로 하여 작업부하 가로채기 기법으로 스케줄링하며, 자원제공자에 결함이 발생하여 작업부하가 중단되면 중단된 작업부하를 재스케줄링한다. 그러나 FCFS 기법은 자원제공자 컴퓨팅의 다양하고 동적인 특성을 고려한 연산 할당 기법 및 연산 수행 모델이 없어 비효율적이다.
- <5> 둘 째, 선형처리자 우선할당 스케줄링 기법은 FCFS의 변형된 형태로 작업부하를 먼저 종료한 자원제공자에게 미완료 작업부하를 재할당하는 방식이다. 선형처리자 우선할당 기법은 적응적 병렬성을 위한 단순한 기법으로 자주 이용되어 왔으며 Bayanihan, Charlotte, Javelin 등 많은 시스템에 적용되었다. 선형처리자 우선할당 스케줄링 기법은 자원제공자의 불안정한 자원 제공 및 일시적인 연산 간섭이나 네트워크 단절 등과 같은 자원제공자 컴퓨팅의 동적인 특성을 고려하지 않고, 단순히 작업부하를 먼저 종료한 자원제공자에게 추가적으로 작업부하를 할당하고 있어 자원제공자가 연산 수행에 실패할 경우, 연산 지연 시간이 증가되고, 연산의 완료를 보장하지 못하며, 처리할 수 있는 응용이 제한적인 단점이 있다. 선형처리자 우선할당 기법에서는 자원제공자의 연산 수행 실패로 인한 반복적 재할당으로 인해 안정적 연산 수행을 보장하지 못하고, 전체 연산 수행 시간이 지연된다. Javelin++, Javelin 3 등은 선형처리자 우선할당 기법을 개선하여 진보된 형태로 적용하였으나, 부가적인 오버헤드가 크고, 작업부하 관리의 복잡도가 증가되는 단점이 발생하고, 연산 지연 및 전체 연산 수행 시간 증가 문제도 여전히 남아 있다.
- <6> 셋 째, 자원제공자의 성능과 연산 참여 시간을 고려한 기법은 WBC(Web-Based Computing) 등에서 이용되었다. WBC에서는 자원제공자의 성능과 자원제공자의 연산 참여 시간을 자원제공자의 작업부하량과 연관시켰다. 그러나

자원제공자와 작업부하 간의 연관은 단순히 자원제공자의 속도만을 반영하여 연산의 의존성, 지역성, 가용성 등을 반영하지 못하는 한계점을 가지고 있으며 연산 중단이나 결함이 발생할 경우 재스케줄링 및 재연관에 대한 기법은 제시하고 있지 않다.

- <7> 상기에서 살펴본 대부분의 기존 연구들은 불안정한 자원제공자의 연산 수행 유형 및 일시적인 연산 간섭이나 네트워크 단절 등으로 인한 휘발성과 같은 자원제공자 컴퓨팅의 동적인 특성을 반영하거나 이를 해결하기 위해 가용성 등을 고려한 스케줄링 기법 등이 거의 없었다. 따라서 지속적인 연산 및 연산의 완료성을 보장하지 못하고, 지연 시간의 증가에 따른 전체 연산 수행 시간의 증가로 시스템의 성능이 저하되는 문제점을 해결하지 못하고 있다.

발명의 내용

해결 하고자하는 과제

- <8> 본 발명은 상술한 문제점을 해결하기 위하여 안출된 것으로, 본 발명에서는 과거의 자원 기여 시간의 비율을 나타낸 기존의 가용성에 양질의 연산 정도를 고려함으로써 자원제공자의 환경에 부합하는 가용성을 새롭게 정의하고, 이를 기반으로 주어진 작업부하의 시간 제약 (Time Restriction, Deadline)에 따라 알맞은 자원제공자를 찾기 위해 각각 가용성을 차등화하여 적용한 스케줄링 기법을 제안한다.
- <9> 본 발명에서의 자원제공자의 가용성 A_{rp} 는 자원제공자의 연산의 질을 고려한 새로운 가용성으로 연산 수행을 안정적으로 잘 수행할 수 있는 자원제공자를 선정하여 우선적으로 작업을 할당함으로써 연산 수행에 대한 시스템의 신뢰성과 성능을 향상시킨다.

과제 해결수단

- <10> 본 발명에서의 스케줄링 방식은 추정 단계, 적합 단계, 실행 단계, 평가 단계를 거치며 체계적인 스케줄링을 제공한다. 추정 단계에서는 과거의 경험을 바탕으로 각 연산 수행의 질을 고려하여 자원제공자들의 가용성을 추정한다. 적합 단계에서는 이러한 가용성을 기반으로, 작업 완료에 대한 시간제약이 있는 자원제공자 컴퓨팅 환경에서 포지티브 방식(Positive Policy), 네거티브 방식(Negative Policy), 객관적 방식(Objective Policy) 등을 적용하여 시간 제약에 따라 적절한 자원 제공자들을 선정한다. 실행 단계에서는 각 자원제공자가 작업을 수행하고 그 결과를 되돌려주며, 평가 단계에서는 주어진 결과 값의 평가를 통해 스케줄링 방식의 신뢰성을 부가하고 있다.

효 과

- <11> 본 발명에 따르면, 자원제공자의 불안정한 자원제공과 연산 중단 현상을 완화하여, 전체 연산 수행 시간을 단축시키고, 결과적으로 연산 수행에 대한 완료성과 신뢰성을 향상시킬 수 있으며, 작업부하의 시간 제약 여부에 따라 각각 다른 방식을 제공함으로써 시간과 비용을 고려한 스케줄링을 차등적으로 적용할 수 있는 효과가 있다.

발명의 실시를 위한 구체적인 내용

- <12> 본 발명은 대표견실도와 상태 전이 확률을 이용하여 자원제공자 컴퓨팅환경에 부합하는 가용성을 추정하는 추정 모듈과; 상기 추정모듈로부터 추정된 자원 제공자의 가용성, 작업부하량 및 대표견실도 값을 비교하여 자원제공자를 선정하고 작업부하를 할당하는 적합 판단 모듈과; 상기 적합 판단 모듈로부터 할당된 작업 부하를 수행하고 작업부하의 수행이 종료되면 결과값을 반환하는 실행모듈과; 상기 실행모듈에 의해 수행된 자원제공자의 연산수행의 유형을 분석하여 예측의 정확성을 평가하는 평가모듈을 포함하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공하여 자원제공자의 불안정한 자원제공과 연산 중단 현상을 완화하여, 전체 연산 수행 시간을 단축시키고, 결과적으로 연산 수행에 대한 완료성과 신뢰성을 향상시킬 수 있도록 한다.
- <13> 또한, 본 발명은, 상기 추정모듈은, 양질의 연산 수행(QoP)와 정확도(PA)를 고려한 하기의 식에 의해 가용성(A_{rp})을 추정하는 것을 특징으로 하는 시각 제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공함이 바람직하다.

$$A_{RP} = \frac{Avg(OnTime)}{Avg(OffTime) + Avg(OnTime)} \cdot QoP \cdot PA, \text{ where } QoP = \frac{Avg(DedicationTime)}{Avg(OnTime)}$$

<14>

<15>

또한, 본 발명은, 상기 적합 판단 모듈을, 자원제공자의 가용성을 기준으로 작업부하를 할당하되 각 작업부하에 대해 작업부하의 수행을 위한 요구시간요소(TR ; Time Required)와 상기 자원제공자의 가용성의 평균값($Avg(A_{RP})$)을 비교하여, 상기 요구시간요소(TR)보다 자원제공자의 가용성의 평균값($Avg(A_{RP})$)이 큰 경우($Avg(A_{RP}) > TR$)의 자원제공자를 선정하고 작업부하를 할당하는 포지티브 방식에 의해 수행하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공한다.

<16>

또한, 본 발명은 상기 적합 판단 모듈은, 자원제공자의 가용성의 평균값($Avg(A_{RP})$)에 비견실적인 자원제공자에 대한 패널티인 대표견실도(RC)를 고려한 값이 요구시간요소(TR)보다 큰 경우($Avg(A_{RP}) \cdot RC > TR$)의 자원 제공자를 선정하고 작업부하를 할당하는 네거티브 방식(Negative Policy)에 의해 수행하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공하여 작업부하의 시간 제약 여부에 따라 각각 다른 방식을 제공함으로써 시간과 비용을 고려한 스케줄링을 차등적으로 적용할 수 있도록 한다.

<17>

또한, 본 발명은 상기 대표견실도(RC)는, 하기의 식에 의해 산출되는 것을 특징으로 하며, 시간제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공함이 바람직하다.

$$RC = \sum_{i=1}^n P_i \log P_i \cdot \frac{NCE}{NTE}$$

<18>

<19>

또한, 본 발명은 상기 적합판단모듈을 최초의 자원 할당 시에는 상기 청구항 3에 의한 포지티브 방식에 의해 자원을 선출하여 할당하고, 일정 기준시간(Time out)이 경과한 후에 작업이 완료되지 않는 경우에는, 청구항 4에 의한 네거티브방식(Negative Policy)에 의해 자원을 선출하여 할당하는 객관적 방식(Objective Policy)에 의해 수행하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 기법을 포함함이 더욱 바람직하다.

<20>

또한, 본 발명은 상기 평가모듈은, 자원제공자의 연산 수행의 유형을 분석하여 하기의 식에 의해 산출된 예측의 정확성(PA)을 평가하는 것을 특징으로 하는 시간 제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공한다.

<21>

$$PA = AP / (IP + AP)$$

<22>

(여기서, 상기 AP 는 정확한 예측의 횟수이며, IP 는 부정확한 예측의 횟수이다.)

<23>

또한, 본 발명은 대표견실도와 상태전이 확률을 이용하여 자원제공자 컴퓨터환경에 부합하는 가용성을 추정하는 추정단계(Presumption Phase); 상기 자원제공자들의 가용성을 기반으로 작업부하량 및 대표견실도의 값을 비교하여 자원제공자를 선정하고 작업부하를 할당하는 적합단계(Suitability Phase); 상기 적합단계에서 할당된 작업부하를 수행하고 작업부하의 수행이 종료되면 결과값을 반환하는 실행단계(Operation Phase); 및 상기 실행 단계에서 실행된 자원제공자의 연산수행의 유형을 분석하여 예측의 정확성을 평가하는 평가단계(Valuation Phase);를 포함하여 이루어지는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법을 제공하여 자원제공자의 불안정한 자원제공과 연산 중단 현상을 완화하여, 전체 연산 수행 시간을 단축시키고, 결과적으로 연산 수행에 대한 완료성과 신뢰성을 향상시킬 수 있도록 한다.

<24>

또한, 본 발명은 상기 추정단계(Presumption Phase)는 청구항 2의 식에 의해 도출된 값에 의해 가용성을 추정하는 단계인 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법을 제공함이 바람직하다.

<25>

또한, 본 발명은 상기 적합단계(Suitability Phase)는, 자원제공자의 가용성을 기준으로 작업부하를 할당하되 각 작업부하에 대해 작업부하의 수행을 위한 요구시간요소(TR)의 확률값과 상기 자원제공자의 가용성의 평균값을 비교하여, 상기 요구시간요소보다 자원제공자의 가용성의 평균값이 큰 경우의 자원제공자를 선정하는 포지티브 방식(Positive Policy)에 의해 수행되는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법을 제공하여 자원선택과 작업부하의 할당의 제약을 그다지 받지 않으면서도 안정적인 스케줄링이 가능하도록 한다.

- <26> 또한, 본 발명은 상기 적합단계(Suitability Phase)는, 상기 자원제공자의 가용성을 기준으로 작업부하를 할당함에 있어서, 비 건설적인 자원제공자에 대한 패널티값인 대표견실도(RC; Representative Confidence)를 고려한 평균가용성값이 요구시간요소(TR)보다 더 큰 자원제공자를 선정하는 네거티브 방식(Negative Policy)을 더 포함하여 이루어지는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 방법을 제공하여 강력한 시간제약이 있는 경우에 응용이 가능하도록 하게 할 수 있다.
- <27> 또한, 본 발명은 상기 적합단계(Suitability Phase)는, 최초의 자원할당시에는 상기 포지티브 방식(Positive Policy)에 의해 자원을 선출하여 할당하고, 일정 기준시간(timeout)이 경과한 후에 작업이 완료되지 않는 경우에는 상기 네거티브 방식(Negative Policy)을 이용하여 자원제공자를 선출하여 할당하는 객관적 방식(Objective Policy)을 더 포함하여 이루어지는 것을 특징으로 하는 시간제약에 따라 가용성을 차등적용하는 스케줄링 시스템을 제공하여 부분적, 제한적으로 시간제약이나 성능제약이 적용되는 경우에 이용할 수 있도록 한다.
- <28> 이하 첨부한 도면을 참고하여 본 발명의 구성과 작용을 구체적으로 설명한다.
- <29> 도 1을 참조하면, 본 발명에 따른 시간제약에 따라 가용성을 차등적용하는 스케줄링 시스템은 자원제공자를 선정하고 작업부하를 할당함에 있어서, 자원제공자의 컴퓨팅환경에 부합하는 가용성을 추정하는 추정모듈(10)과 상기 추정모듈로 부터 추정된 자원제공자의 가용성, 작업부하량, 대표견실도의 값을 비교하며, 이러한 비교데이터를 기준으로 자원제공자를 선정하고 작업의 부하를 할당하는 적합판단모듈(20), 그리고 이러한 적합판단모듈로부터 할당된 작업부하를 수행하고 수행된 작업부하가 종료되면 결과값을 반환하는 실행모듈(30), 그리고 상기 실행모듈(30)으로부터 수행된 자원제공자의 연산수행의 유형을 분석하여 예측의 정확성을 평가하는 평가모듈(40)로 구성됨이 바람직하다.
- <30> 이러한 각 모듈의 작용에 의해 수행되는 단계, 즉, 구체적으로 추정모듈에 의해 수행되는 추정단계(Presumption Phase)와 적합판단모듈에 의해 수행되는 적합단계(Suitability Phase), 실행모듈에 의해 수행되는 실행단계(Operation Phase), 그리고 평가모듈에 의해 수행되는 평가단계(Valuation Phase)에 대한 구체적인 사항을 아래에서 기술함으로써, 본 발명의 작용을 설명하기로 한다.
- <31> 자원제공자 컴퓨팅 환경에서 안정적인 시스템을 환경을 제공하고, 높은 성능을 보장하기 위해서 자원제공자의 동적이고, 비예측적 특성을 고려한 스케줄링은 매우 중요한 고려사항이다. 따라서 가용성을 기반으로 작업부하를 스케줄링하게 된다. 그러나 기존의 자원제공자 컴퓨팅 시스템들은 스케줄링 시에 자원제공자들의 가용성(availability)을 고려하지 않거나, 단순한 가용성 개념을 도입하거나, 자원제공자의 자원제공 정도를 통계적인 수치로 이용해왔다. 이로 인해 자원제공자의 연산 실패 발생 시에 연산의 완료를 보장하지 못하고, 연산 수행 패턴을 예측하지 못하였다.
- <32> 자원제공자의 자율적, 동적 연산 수행 패턴에 대한 분석이 선행된 후, 작업부하가 요구하는 가용성에 부합하는 자원제공자를 선정하여 작업부하를 할당하는 것이 필요하다. 따라서 본 발명에서는 자원제공자 컴퓨팅 환경에 부합하는 가용성을 새롭게 정의하고, 이를 이용하여 자원의 자원제공여부를 예측하고, 작업부하량을 처리할 수 있는 자원제공자에게 작업을 할당하게 된다.
- <33> 일반적으로 기존의 가용성은 다음의 식 1에 나타난 바와 같이, 자원제공자가 평균적으로 켜져 있는 시간($OnTime$)과 평균적으로 꺼져 있는 시간($OffTime$)의 합에 대해 켜져 있는 시간의 비율로서 구해진다.
- <34> {식 1}

$$A = \frac{Avg(OnTime)}{Avg(OffTime) + Avg(OnTime)}$$

- <35>
- <36> 그러나 기존 가용성은 단순히 자원제공자가 켜져 있는 시간만 고려하기 때문에 연산 도중 잦은 중단 현상이 발생하여 작업부하의 완료를 지연시키고, 이러한 작업부하 지연은 전체 연산 지연에 영향을 미치게 된다.
- <37> 따라서, 본 발명에서는 자원제공자가 켜져서 자원을 제공하는 시간에 실제로 양질의 연산을 수행하는 것을 고려하여 보다 환경에 부합하는 가용성을 추정한다.
- <38> 하기의 식 2는 기존 가용성에 양질의 연산 수행 QoP (Quality of Provision) 와 정확도 PA (Prediction Accuracy)를 고려한 새로운 자원제공자 가용성을 나타낸다. 또한 상기 QoP 는 자원제공자가 자원을 기여한 시간

중에서 자원 기여를 성실하게 수행한 시간의 비율로서 나타낸다. 아울러 자원 기여를 성실하게 수행한 평균 시간 $Avg(Dedication\ Time)$ 은 단위 작업부하 한 개를 중단 없이 수행한 시간들의 평균 시간을 의미한다.

<39> {식 2}

$$A_{RP} = \frac{Avg(OnTime)}{Avg(OffTime) + Avg(OnTime)} \cdot QoP \cdot PA, \text{ where } QoP = \frac{Avg(DedicationTime)}{Avg(OnTime)}$$

<40>

<41> 시간 제약이 있는 응용을 수행하여야 할 경우, 본 발명의 QoP 를 고려한 가용성의 차등 적용 작업부하 스케줄링 기법은 시간의 변화에 민감하게 가변하는 자원제공자의 연산 수행에 대한 불확실성, 비예측성, 휘발성 문제를 완화시키는 장점이 있다.

<42> 본 발명에 따른 가용성 차등 적용 작업부하 스케줄링 기법은 스케줄링 시에 실시간으로 예측에 대한 정확성을 반영하여 작업부하를 할당한다.

<43> 이를 지원하기 위해 스케줄링 과정은 추정 단계(Presumption Phase), 적합 단계 (Suitability Phase), 실행 단계(Operation Phase), 평가 단계(Valuation Phase) 순으로 진행되며, 각 단계가 반복되면서 이전 스케줄링의 성공 여부를 고려함으로써 적응적으로 성능을 향상시킨다.

<44> 상기 추정 단계는 대표 견실도와 상태 전이 확률을 이용하여 추정, 계산하는 단계이다. 도 2는 추정 단계의 알고리즘을 기술하고 있다.

<45> 상기 적합 단계는 작업부하량과 자원제공자의 가용성 및 대표 견실도 값을 비교하여 자원제공자를 선정하고 작업부하를 할당하는 단계이다. 도 3은 적합 단계의 알고리즘을 기술하고 있다.

<46> 상기 실행 단계는 각 자원제공자가 할당받은 작업부하를 수행하고, 작업부하의 수행이 종료되면 결과값을 반환하는 단계이다. 도 4는 실행 단계의 알고리즘을 기술하고 있다.

<47> 상기 평가 단계는 자원제공자의 연산 수행의 유형을 분석하여 예측의 정확성을 평가하는 단계이다. 도 5는 평가 단계의 알고리즘을 기술하고 있다.

<48> 도 6은 추정 단계, 적합 단계, 실행 단계, 평가 단계를 거치면서, 스케줄링하는 과정을 나타내고 있다.

<49> 도 7은 스케줄링 기법의 아키텍처를 나타내고 있다. 특히 적합 단계에서는 포지티브 방식(Positive Policy), 네거티브 방식(Negative Policy), 객관적 방식(Objective Policy)을 이용하여 시간 제약에 따라 적합한 자원을 선택하도록 하고 있다.

<50> 본 발명에 따른 스케줄링 알고리즘은 적합 단계에서 시간 제약에 따라 적절한 자원을 선택하기 위해 포지티브 방식 (Positive Policy), 네거티브 방식 (Negative Policy), 객관적 방식(Objective Policy)의 3가지 방식을 지원한다.

<51> 상기 포지티브 방식(Positive Policy)은 자원제공자가 지속적으로 연산을 수행하여 연산 도중 연산이 중단된다 할지라도 언젠가는 결국 작업부하수행을 완료할 것이라는 포지티브 전망을 가지고, 별도의 오버헤드가 없는 단순한 방식으로 작업부하를 할당하는 방식이다. 따라서 포지티브 방식(Positive Policy)을 이용할 경우, 자원 선택과 작업부하 할당의 제약이 많지 않다. 따라서 상기 포지티브 방식(Positive Policy)은 작업부하 완료에 대한 시간 제약이 거의 없는 경우에 사용되며 그 응용은 주로 다수의 작업부하 간의 의존성이 없는 독립적이고, 유사한 연산으로 구성된 작업부하이나 장기간 연산을 수행하는 대용량 작업부하 등이다. SETI@HOME과 같은 시스템은 이러한 연산 특성을 가지는 대표적인 시스템이다.

<52> 상기 네거티브 방식(Negative Policy)은 자원제공자의 불안정한 자원제공으로 인해 작업부하 수행이 중단되고, 이로 인해 연산 수행이 지연되고, 심지어 작업부하를 완료하지 못하여 시스템의 안정성과 신뢰성을 저하시킨다는 모토를 가진다. 따라서 포지티브 방식(Positive Policy)에서는 작업부하는 반드시 견실한 자원에게 할당되어야 하며, 시스템은 자원제공자의 작업부하 수행능력을 판단하는 기준으로 가용성을 이용한 제약을 통해 견실한 자원 선택하게 된다. 따라서 네거티브 방식(Negative Policy)은 강력한 시간 제약이 있는 경우로, 대량의 기업, 정부 자료, 예를 들면, 인구 조사 작업부하와 같이 작업부하의 양은 많으나 반드시 데드라인을 지켜야 하는 응용에 적합하다. 이 경우에는 작업부하의 완료시간이 더욱 중요하게 된다. 그러나 이 경우에는 고신뢰성의 자원을 선택하기 위한 오버헤드가 발생한다. 네거티브 방식(Negative Policy)은 자원제공자의 수가 작업부하보다 많

을 경우에 더 유용하다.

<53> 상기 객관적 방식(Objective Policy)은 포저티브 방식(Positive Policy)과 네거티브 방식(Negative Policy)의 하이브리드 방식으로 별도의 시간 제약이 없는 작업부하의 경우에는 포저티브 방식(Positive Policy)을 이용하여 시스템 이용성을 향상시키고, 작업부하의 데드라인이 있거나 실시간 작업부하 등을 수행하기 위해서는 네거티브 방식(Negative Policy)을 이용하여 시스템의 신뢰성을 향상시키고자 한다. 따라서 객관적 방식(Objective Policy)은 부분적, 제한적으로 시간 제약이나 성능 제약이 적용되는 경우에 이용될 수 있다. 이 방식은 시간 제약의 문제를 보다 적은 비용으로 해결할 수 있다. 객관적 방식(Objective Policy)에서는 미완료 작업부하를 재할당할 경우, 신뢰적 자원을 선정하여 할당한다.

<54> 상기 포저티브 방식(Positive Policy)은 자원제공자의 가용성을 기준으로 작업부하를 할당한다. 각 작업부하에 대해 작업부하 수행을 위한 요구시간요소 $TR(Time\ Required)$ 가 확률 값 형식으로 주어져 있는 경우, TR 보다 큰 가용성을 가지는 자원제공자를 선출할 수 있다. 식 3은 자원제공자 가용성 A_{RP} 의 평균 값 $Avg(A_{RP})$ 을 요구시간 요소 TR 과 비교하여 $Avg(A_{RP})$ 이 TR 보다 큰 자원제공자를 선출하는 식을 나타낸다.

<55> {식 3}

<56>
$$Avg(A_{RP}) > TR$$

<57> 상기 네거티브 방식(Negative Policy)은 비견실적인 자원제공자에 대한 패널티를 부과한다. 비견실적 자원제공자는 자원 제공 도중, 갑작스럽게 중단하거나 비예측적 행동을 하거나, 수행한 연산의 결과가 정확하지 않는 자원제공자를 의미한다. 네거티브 방식(Negative Policy)에서는 이러한 비신뢰적 자원제공자를 여과하기 위해 자원제공자의 불규칙적인 연산 패턴으로 인해 예측이 어렵거나 작업부하수행의 결과가 부정확할 경우에 가용성에 대한 패널티를 부과한다. 패널티는 대표 견실도 $RC (Representative\ Confidence)$ 으로 주어지게 된다.

<58> 상기 RC 는 특히 자원의 예측 정도에 대해 엔트로피를 이용하여 불확실한 정도를 측정하고, 작업부하 결과의 정확도에 대해서는 총 수행 연산수 대비 정확한 연산 수행 수를 고려한다. 특히 RC 는 식 4와 같이 계산된다. 식4에서 $NCE (Number\ of\ Correct\ Execution)$ 은 정확한 연산 수행수를 나타내고, $NTE (Number\ of\ Total\ Execution)$ 는 총 연산 수행수를 나타낸다.

<59> {식 4}

<60>
$$RC = \sum_{i=1}^n P_i \log P_i \cdot \frac{NCE}{NTE}$$

<61> 네거티브 방식(Negative Policy)의 자원 선정 기준식은 식 5과 같이 RC 를 고려한 평균 가용성이 작업부하 실행을 위한 요구 시간 요소 TR 보다 큰 자원제공자를 선택한다.

<62> {식 5}

<63>
$$Avg(A_{RP}) \wedge RC > TR$$

<64> 따라서 네거티브 방식(Negative Policy)에서는 기존의 가용성의 자원제공자의 연산 수행에 대한 과거의 신뢰성을 고려하여 작업부하 할당 여부를 결정함으로써, 보다 안정적 자원에 작업부하를 할당하고, 시스템의 성능 및 신뢰성, 안정성을 향상시킨다.

<65> 객관적 방식(Objective Policy)은 사용자의 의도나 시스템의 상황에 따라 위의 두 방식을 적절히 배합하여 이용한다. 즉, 주어진 시간 제약을 만족하기 위해 적절한 일정 기준 시간 (timeout)을 두고, 처음의 자원할당 시에는 포저티브 방식(Positive Policy)을 적용하여 자원을 선출하고 할당하며, timeout이 경과된 이후에도 작업을 완료하지 못한 작업들을 할당할 경우에는 네거티브 방식(Negative Policy)을 이용하여 자원제공자를 선출하게 된다.

<66> 종합하자면, 본 발명에 따른 스케줄링은 추정 단계(Presumption Phase), 적합 단계 (Suitability Phase), 실행 단계(Operation Phase), 평가 단계(Valuation Phase) 순으로 진행됨이 바람직하다.

- <67> 추정 단계에서는 대표 건설도와 상태 전이 확률을 이용하여 추정, 계산한다.
- <68> 적합 단계에서는 작업부하량과 자원제공자의 가용성 및 대표 건설도 RC 값을 비교하여 시간대의 상태와 작업부하의 양에 부합하는 자원제공자를 선정하고 작업부하를 할당한다.
- <69> 실행 단계에서는 각 자원제공자가 할당받은 작업부하를 수행하고, 작업부하 수행이 종료되면 결과 값을 반환한다.
- <70> 마지막 평가 단계에서는 자원제공자의 연산 수행의 유형을 분석하여 예측의 정확성을 평가하기 위해 예측의 정확성 PA (Prediction Accuracy)를 평가한다. 예측의 정확성 PA 는 식 6에 기술된 바와 같이 총 예측 횟수, 정확한 예측 AP (Accurate Prediction)과 부정확한 예측 IP (Inaccurate Prediction) 즉 정확한 예측 횟수의 비율로서 구한다. 총 예측 횟수는 정확한 예측의 횟수와 부정확한 예측 IP (Inaccurate Prediction)의 횟수의 합이다.
- <71> {식 6}

$$PA = \frac{AP}{IP + AP}$$

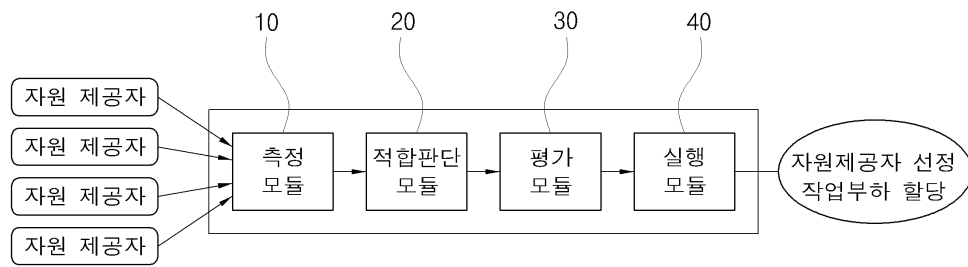
- <72>
- <73> 가용성 기반의 시간제약에 따른 차등화 작업부하 스케줄링 알고리즘은 도 2~ 도 6에 걸쳐 추정 단계 (Presumption Phase), 적합 단계 (Suitability Phase), 실행 단계 (Operation Phase), 평가 단계 (Valuation Phase) 각각의 단계가 기술되어 있다.
- <74> 전술한 바와 같은 본 발명의 상세한 설명에서는 구체적인 실시예에 관해 설명하였다. 그러나 본 발명의 범주에서 벗어나지 않는 한도 내에서는 여러 가지 변형이 가능하다. 본 발명의 기술적 사상은 본 발명의 기술한 실시예에 국한되어 정해져서는 안 되며, 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

도면의 간단한 설명

- <75> 도 1은 도 1을 참조하면, 본 발명에 따른 시간제약에 따라 가용성을 차등적으로 적용하는 스케줄링 시스템에 관한 개략도이다.
- <76> 도 2는 시간 제약에 따라 가용성을 차등 적용시키는 스케줄링 알고리즘의 추정 단계 (Presumption Phase)를 기술한 것이다.
- <77> 도 3는 시간 제약에 따라 가용성을 차등 적용시키는 스케줄링 알고리즘의 적합 단계 (Suitability Phase)를 기술한 것이다.
- <78> 도 4은 시간 제약에 따라 가용성을 차등 적용시키는 스케줄링 알고리즘의 평가 단계 (Valuation Phase)를 기술한 것이다.
- <79> 도 5는 시간 제약에 따라 가용성을 차등 적용시키는 스케줄링 알고리즘의 실행 단계 (Operation Phase)를 기술한 것이다.
- <80> 도 6는 본 발명에 따른, 추정 단계 (Presumption Phase), 적합 단계 (Suitability Phase), 실행 단계 (Operation Phase), 평가 단계 (Valuation Phase)의 4 단계를 순차적으로 도시한 개념도이다.
- <81> 도 7은 전체적인 스케줄링 아키텍처를 나타내는 도면이다.

도면

도면1



도면2

```

// Presumption Phase

For(i=1 ; n ; i++)
{
    Get A;
    Calculate_ARP ( )
    {
        ARP = A * QoP;
    }
    Grade (ResourceProvider, ARP)
}
    
```

도면3

```

// Suitability Phase

For(i=1 ; n ; i++)
{
    ARP=Avg(ARP);
    TimeOut=0;

    Case1: Positive Policy
        Compare(ARP, TtbR);
        MatchIf(ARP, TtbR)
        {
            SuitableResourceProvider = Select(ARP, ResourceProviderList);
            AllocateWorkLoad(EligibleResourceProvider);
        }
        MatchElseIf(ARP, TtbR)
        {
            ReplicableResourceProvider = Select(ARP, ResourceProviderList);
            AllocateWorkLoad(ReplicableResourceProvider);
        }
    Case2: Negative Policy
        Get RC;
        Compare(RC:ARP, TtbR);
        MatchIf(RC:ARP, TtbR)
        {
            SuitableResourceProvider = Select(RC:ARP, ResourceProviderList);
            AllocateWorkLoad(EligibleResourceProvider);
        }
        MatchElseIf(RC:ARP, TtbR)
        {
            ReplicableResourceProvider = Select(RC:ARP, ResourceProviderList);
            AllocateWorkLoad(ReplicableResourceProvider);
        }
    Case3: Objective Policy
        If (TimeOut=0)
        {
            Compare(ARP, TtbR);
            MatchIf(ARP, TtbR)
            {
                SuitableResourceProvider = Select(ARP, ResourceProviderList);
                AllocateWorkLoad(EligibleResourceProvider);
            }
            MatchElseIf(ARP, TtbR)
            {
                ReplicableResourceProvider = Select(ARP, ResourceProviderList);
                AllocateWorkLoad(ReplicableResourceProvider);
            }
        }
        If (TimeOut=1)
        {
            Compare(RC:ARP, TtbR);
            MatchIf(RC:ARP, TtbR)
            {
                SuitableResourceProvider = Select(RC:ARP, ResourceProviderList);
                AllocateWorkLoad(EligibleResourceProvider);
            }
            MatchElseIf(RC:ARP, TtbR)
            {
                ReplicableResourceProvider = Select(RC:ARP, ResourceProviderList);
                AllocateWorkLoad(ReplicableResourceProvider);
            }
        }
    }
}

```

도면4

```

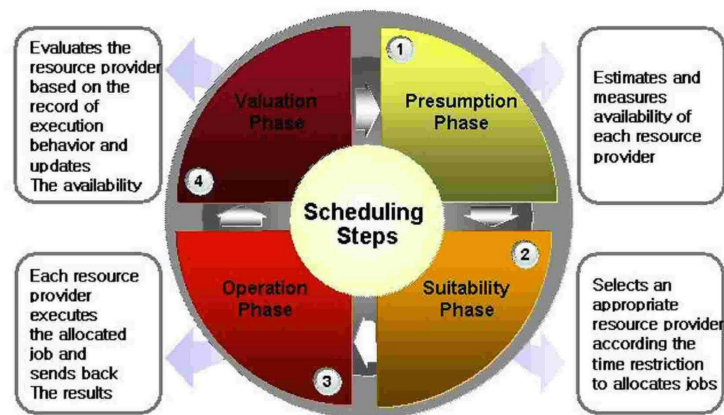
//Operation Phase
While (WorkLoad=1)
{
    OperateWorkLoad(WorkLoad);
    If(Completed) then
        Return(Results);
}

```

도면5

```
//Valuation Phase
ValuateAccuracy( )
{
    PA=0;
    TotalPrediction= AccuratePrediction + InaccuratePrediction;
    PA = AccuratePrediction / TotalPrediction;
}
UpdateMarkovModel(PA);
```

도면6



도면7

