(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0080795
(43) 공개일자 2014년07월01일

(51) 국제특허분류(Int. Cl.)
G06F 9/06 (2006.01) *G06F 15/16* (2006.01)
G06F 9/44 (2006.01)
(21) 출원번호 10-2012-0148587
(22) 출원일자 2012년12월18일
심사청구일자 2012년12월18일

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
박성용
서울 송파구 중대로 24, 229동 1103호 (문정동, 올림픽훼밀리타운)
이권용
서울특별시 마포구 신수동 167번지 메트로디오빌 2704호
정대영
경기 성남시 분당구 양현로 220, 1107동 303호 (이매동, 이매촌아파트)
(74) 대리인
특허법인충현

전체 청구항 수 : 총 15 항

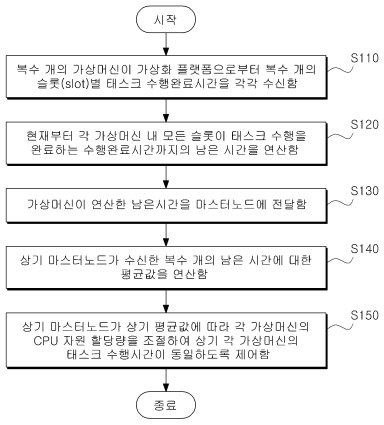
(54) 발명의 명칭 **가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템**

(57) 요 약

본 발명은 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템에 관한 것으로, 보다 구체적으로는 복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신하는 단계; 상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산하는 단계; 상기 가상머신이 연산한 남은시간을 마스터노드에 전달하는 단계; 상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산하는 단계; 및 상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계;를 포함한다.

이러한 구성에 의해, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 다수의 가상머신이 태스크 수행 시, 동일한 태스크 수행시간이 소요됨으로써, 클러스터 환경에서의 성능 향상을 기대할 수 있는 효과가 있다.

대 표 도 - 도4



이 발명을 지원한 국가연구개발사업

과제고유번호	NIPA-2012-H0301-12-3006
부처명	지식경제부
연구사업명	대학 IT 연구센터 육성지원사업
연구과제명	소셜 미디어 서비스를 위한 클라우드 플랫폼 및 응용 서비스 기술 개발
기 여 율	1/1
주관기관	건국대학교 산학협력단
연구기간	2012.01.01 ~ 2012.12.31

특허청구의 범위

청구항 1

가상화 플랫폼 내 크레딧 스케줄러의 파라미터를 변경하여 복수 개의 가상머신간에 CPU 자원 할당량을 조절하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 2

제1항에 있어서,

상기 크레딧 스케줄러는

상기 가상머신에 태스크(task)를 할당하여 할당된 태스크가 소모될 때까지 CPU 시간을 태스크의 생성 순으로 차례로 분배하여 상기 CPU를 점유하는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 3

제2항에 있어서,

상기 크래딧은

상기 가상머신에 부여된 가중치 또는 점유율 중 적어도 하나에 기초하여 연산되는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 4

제3항에 있어서,

상기 가중치는

복수 개의 가상머신간에 상대적인 값을 나타내는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 5

제3항에 있어서,

상기 점유율은

상기 가상머신이 점유할 수 있는 물리 CPU의 절대값을 백분율로 나타낸 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 6

제1항에 있어서,

복수 개의 가상머신 내 슬롯이 태스크를 수행하는 중 낙오자 태스크가 발생한 경우에는 다른 가상머신의 CPU 자원이 상기 낙오자 태스크가 발생한 가상머신에 할당되는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 7

복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신하는 단계;

상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산하는 단계;

상기 가상머신이 연산한 남은시간을 마스터노드에 전달하는 단계;

상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산하는 단계; 및

상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계;

를 포함하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 8

제7항에 있어서,

상기 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계는

상기 마스터노드가 각 가상머신 내 모든 슬롯의 태스크 수행완료시간까지의 남은 시간을 평균값과 비교하여, 상기 남은 시간이 상기 평균값보다 낮으면 해당 가상머신의 점유율을 감소시키거나,

상기 남은 시간이 상기 평균값보다 높으면 해당 가상머신의 점유율을 증가시키는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 9

제7항에 있어서,

상기 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계는

동일한 물리머신에 속하는 다수의 가상머신별 각 남은 시간이 상기 평균값과 비교하여 기설정된 값보다 크거나 작은 경우에는, 상기 가상머신의 CPU에 대한 가중치를 재할당하는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 10

제7항에 있어서,

상기 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계는

동일한 가상머신 내 다수의 슬롯이 존재하는 경우에는 상기 슬롯의 태스크 처리에 따른 프로세스의 우선순위를 변경하는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법.

청구항 11

제1항 내지 제10항 중 어느 한 항에 따른 방법을 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체.

청구항 12

복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신하는 태스크 수행완료시간 수신부;

상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산하는 남은시간 연산부;

상기 가상머신이 연산한 남은시간을 마스터노드에 전달하는 남은시간 전송부;

상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산하는 평균값 연산부; 및

상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어하는 제어부;

를 포함하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템.

청구항 13

제12항에 있어서,

상기 제어부는

상기 마스터노드가 각 가상머신 내 모든 슬롯의 태스크 수행완료시간까지의 남은 시간을 평균값과 비교하는 비교모듈; 및

상기 남은 시간이 상기 평균값보다 낮은 경우에는 해당 가상머신의 점유율을 감소시키거나, 상기 남은 시간이 상기 평균값보다 높은 경우에는 해당 가상머신의 점유율을 증가시키는 점유율 변경모듈;

을 포함하는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템.

청구항 14

제12항에 있어서,

상기 제어부는

동일한 물리머신 내 속하는 다수의 가상머신별 각 남은 시간이 상기 평균값과 비교하여 기설정된 값보다 크거나 작은 경우에는, 상기 가상머신의 CPU에 대한 가중치를 재할당하는 가중치 재할당모듈;

을 포함하는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템.

청구항 15

제12항에 있어서,

상기 제어부는

동일한 가상머신 내 다수의 슬롯이 존재하는 경우에는 상기 슬롯의 태스크 처리에 따른 프로세스의 우선순위를 변경하는 프로세스 우선순위 변경모듈;

을 포함하는 것을 특징으로 하는 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템.

명 세 서

기 술 분 야

[0001] 본 발명은 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템에 관한 것으로, 특히 복수 개의 가상머신이 태스크를 수행하는데 있어서, 수행시간이 동일하게 소요되도록 하여, 성능을 향상시킬 수 있는 가상화 환경

내 하둡 맵리듀스의 부하 분산 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 최근 들어, 가상화 기술의 발달과 성능향상 및 대용량의 하드웨어 인프라 구축 기술이 구현됨에 따라, 아마존을 비롯한 대표적인 글로벌 기업에서는 가상화 기반의 클라우드 컴퓨팅 서비스를 상용화 하고 있는 추세이다.

[0003] 이러한 클라우드 컴퓨팅을 각종 언론 매체를 통해서도 비용 절감 및 IT 자원의 효율적 활용에 적합하다고 소개되고 있다. 이와 관련된 서비스들이 전 세계의 이목을 끌고 있다.

[0004] 특히, 클라우드 환경에서는 가상화를 통해 서비스 제공자로부터 사용자가 원하는 만큼의 컴퓨팅 자원을 할당 받을 수 있으며, 사용자에게 할당된 컴퓨팅 자원에 대한 포괄적인 제어를 통해 사용자의 요구사항에 맞춰 컴퓨팅 자원 할당을 신속하게 확장하거나, 축소할 수 있다. 다시 말해, 지금까지 사용자가 컴퓨팅 자원을 직접 구매 및 관리하는 형태에서 필요한 만큼 서비스를 공급받는 형태로 변화함에 따라, 기존 방식보다 저렴하고 유연한 자원 운영이 가능하게 되었다.

[0005] 따라서, 물리머신 기반의 클러스터가 클라우드의 이점을 얻기 위해, 점차 가상화 클러스터로 구축 환경이 변화되었고, 물리 클러스터 환경에서 동작하던 분산처리 플랫폼 또한 가상화 클러스터 환경으로 변화하게 되었다.

[0006] 이러한 분산처리 플랫폼은 테라 바이트(Tera Byte)규모의 대용량 데이터를 효율적으로 저장하고 데이터-집중적(Data-intensive)인 워크로드를 정해진 시간 내 빠르게 처리할 수 있는 장점을 가지고 있다.

[0007] 특히, 대표적인 분산처리 플랫폼인 하둡(Hadoop)에서도 가상화 클러스터 구성 시, 성능 문제가 발생하며 실제로 하둡에서 분산처리를 담당하는 맵리듀스(MapReduce)의 성능을 비교해보면, 물리머신들로 구성된 클러스터 보다 그 성능이 떨어진다.

[0008] 이러한 하둡 맵리듀스는 태스크(Task)를 클러스터에 분산하여 처리할 때, 데이터 지역성(Data locality)과 구성된 클러스터 환경의 차이로 모든 태스크가 각 노드에서 동시에 끝나지 않고 물결(Wave) 형태로 종료된다.

[0009] 따라서, 모든 태스크가 완료되어야만 종료되는 맵리듀스의 특성상 심각한 성능저하를 야기시키는 문제점이 발생했다.

[0010] 상술한 바와 같이, 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템을 살펴보면 다음과 같다.

[0011] 선행기술 1은 한국공개특허공보 제2012-0041907호(2012.05.03)로서, 맵리듀스 기반의 대용량 데이터 분산 계산 방법 및 그 시스템에 관한 것이다. 이러한 선행기술 1은 a) 각 데이터 공급자로부터의 데이터들을 시큐러티 에이전트에서 수신하는 단계; b) 시큐러티 에이전트에서 수신된 데이터를 준동형 암호화 방식에 따라 암호화하는 단계; c) 상기 암호화된 데이터들을 하둡에 제공하는 단계; d) 상기 시큐러티 에이전트에서 상기 암호화된 데이터를 연산하기 위한 연산 프로그램을 상기 하둡에 제공하는 단계; e) 상기 하둡에서 상기 연산 프로그램 및 맵리듀스(MapReduce)에 따라 상기 암호화된 데이터들을 연산하여 키/값 쌍들을 발생시키고, 상기 발생된 키/값 쌍들을 연산한 후, 상기 키/값 쌍들을 준동형성을 이용하여 계산하고, 그 결과 값들을 상기 시큐러티 에이전트에 제공하는 단계; 및 f) 상기 시큐러티 에이전트에서 상기 하둡으로부터의 상기 결과 값들을 복호화하는 단계를 포함하여, 대용량 분산시스템 기술의 하나인 하둡 기반 맵리듀스 구조에서 암호화를 통해 기밀정보를 보호하며, HE연산을 통해 암호화된 통계데이터를 이용가능하게 함으로써, 맵퍼와 리듀서를 신뢰하지 않으면서 HE를 이용하여 일관된 정확도로 결과 값을 사용하여 최종적으로 end-to-end 기밀성을 보장하면서 분산계산을 얻을 수 있다.

[0012] 또한, 선행기술 2는 한국공개특허공보 제2012-0071980호(2012.07.03)로서, 맵 리듀스를 이용한 데이터 처리 장치 및 방법에 관한 것이다. 이러한 선행기술 2는 생성된 로그 데이터를 저장하는 서버에 상기 로그 데이터의 1차 변환을 지시하는 서버 관리부(Agent Manager); 및 상기 1차 변환된 로그 데이터를 수합(Reduce) 및 2차 변환하여 데이터 웨어 하우스(Data Warehouse)로 전송하는 데이터 가공부(Aggregation Agent);를 포함하며, 추가 스토리지 하드웨어나 고사양 서버의 추가 없이 기존 하드웨어와 일반 서버에서 ETL(Extraction, Transformation, Loading)작업을 수행할 수 있는 장점이 있다.

발명의 내용

해결하려는 과제

[0013] 상기와 같은 종래 기술의 문제점을 해결하기 위해, 본 발명은 가상화 클러스터 환경에서 전체 맵 리듀스 태스크가 균일한 시간 동안 처리되도록 하여 성능을 향상시킬 수 있는 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템을 제공하고자 한다.

과제의 해결 수단

[0014] 위와 같은 과제를 해결하기 위한 본 발명의 한 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법은 가상화 플랫폼 내 크레딧 스케줄러의 파라미터를 변경하여 복수 개의 가상머신간에 CPU 자원 할당량을 조절한다.

[0015] 보다 바람직하게는 상기 가상머신에 태스크(task)를 할당하여 할당된 태스크가 소모될 때까지 CPU 시간을 태스크의 생성 순으로 차례로 분배하여 상기 CPU를 점유하는 크레딧 스케줄러를 포함할 수 있다.

[0016] 특히, 상기 가상머신에 부여된 가중치 또는 점유율 중 적어도 하나에 기초하여 연산되는 크레딧을 포함할 수 있다.

[0017] 특히, 복수 개의 가상머신간에 상대적인 값을 나타내는 가중치를 포함할 수 있다.

[0018] 특히, 상기 가상머신이 점유할 수 있는 물리 CPU의 절대값을 백분율로 나타낸 점유율을 포함할 수 있다.

[0019] 보다 바람직하게는 복수 개의 가상머신 내 슬롯이 태스크를 수행하는 중 낙오자 태스크가 발생한 경우에는 다른 가상머신의 CPU 자원이 상기 낙오자 태스크가 발생한 가상머신에 할당될 수 있다.

[0020] 위와 같은 과제를 해결하기 위한 본 발명의 다른 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법은 복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신하는 단계; 상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산하는 단계; 상기 가상머신이 연산한 남은시간을 마스터노드에 전달하는 단계; 상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산하는 단계; 및 상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계;를 포함한다.

[0021] 보다 바람직하게는 상기 마스터노드가 각 가상머신 내 모든 슬롯의 태스크 수행완료시간까지의 남은 시간을 평균값과 비교하여, 상기 남은 시간이 상기 평균값보다 낮으면 해당 가상머신의 점유율을 감소시키거나, 상기 남은 시간이 상기 평균값보다 높으면 해당 가상머신의 점유율을 증가시키는 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계를 포함할 수 있다.

[0022] 특히, 동일한 물리머신에 속하는 다수의 가상머신별 각 남은 시간이 상기 평균값과 비교하여 기설정된 값보다 크거나 작은 경우에는, 상기 가상머신의 CPU에 대한 가중치를 재할당하는 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계를 포함할 수 있다.

[0023] 특히, 동일한 가상머신 내 다수의 슬롯이 존재하는 경우에는 상기 슬롯의 태스크 처리에 따른 프로세스의 우선순위를 변경하는 가상머신의 태스크 수행시간이 동일하도록 제어하는 단계를 포함할 수 있다.

[0024] 위와 같은 과제를 해결하기 위한 본 발명의 다른 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템은 복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신하는 태스크 수행완료시간 수신부; 상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산하는 남은시간 연산부; 상기 가상머신이 연산한 남은시간을 마스터노드에 전달하는 남은시간 전송부; 상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산하는 평균값 연산부; 및 상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어하는 제어부;를 포함한다.

[0025] 보다 바람직하게는 상기 마스터노드가 각 가상머신 내 모든 슬롯의 태스크 수행완료시간까지의 남은 시간을 평균값과 비교하는 비교모듈; 상기 남은 시간이 상기 평균값보다 낮은 경우에는 해당 가상머신의 점유율을 감소시키는 점유율 감소모듈; 및 상기 남은 시간이 상기 평균값보다 높은 경우에는 해당 가상머신의 점유율을 증가시

키는 점유율 증가모듈;을 포함하는 제어부를 포함할 수 있다.

[0026] 보다 바람직하게는 동일한 물리머신 내 속하는 다수의 가상머신별 각 남은 시간이 상기 평균값과 비교하여 기설정된 값보다 크거나 작은 경우에는, 상기 가상머신의 CPU에 대한 가중치를 재할당하는 가중치 재할당모듈;을 포함하는 제어부를 포함할 수 있다.

[0027] 특히, 동일한 가상머신 내 다수의 슬롯이 존재하는 경우에는 상기 슬롯의 태스크 처리에 따른 프로세스의 우선순위를 변경하는 프로세스 우선순위변경모듈;을 포함하는 제어부를 포함할 수 있다.

발명의 효과

[0028] 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 다수의 가상머신이 맵리듀스 태스크 수행 시, 균일한 태스크 수행시간이 소요되도록 함으로써, 클러스터 환경에서의 하둡 맵리듀스의 성능 향상을 기대할 수 있는 효과가 있다.

[0029] 또한, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 크레딧 스케줄러의 가중치 또는 점유율 변경을 통해 부하분산이 이루어지도록 함으로써, 하둡 플랫폼과 하이퍼바이저에 대한 수정은 이루어지지 않아, 맵 리듀스 뿐만 아니라 다양한 하둡 응용프로그램에서도 적용할 수 있어, 소요비용을 절약할 수 있는 효과가 있다.

[0030] 더불어, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 각 가상머신마다 전체 태스크 수행시간을 동일하게 되도록 제어함으로써, 낙오자 태스크의 발생을 최소화시킬 수 있는 효과가 있다.

도면의 간단한 설명

[0031] 도 1은 하둡 맵리듀스 수행절차를 나타낸 도면이다.

도 2는 각 슬롯별 태스크 수행시간을 비교한 그래프이다.

도 3은 데이터 크기 증가에 따른 완료 태스크의 수와 강제 종료된 완료 태스크의 수를 나타낸 그래프이다.

도 4는 본 발명의 일 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법을 나타낸 순서도이다.

도 5는 부하 분산 시스템의 전체 구조를 나타낸 블록도이다.

도 6은 본 발명의 다른 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템을 나타낸 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0032] 이하, 본 발명을 바람직한 실시 예와 첨부한 도면을 참고로 하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 여기에서 설명하는 실시 예에 한정되는 것은 아니다.

[0033] 먼저, 본 발명을 설명하기에 앞서, 본 발명이 구현되는 가상환경에 대하여 간략히 설명하도록 한다.

[0034] 가상환경이 구현되기 위해, 사용되는 가상머신은 특정한 프로그램을 실행하되, 하드웨어와 직접적인 통신이 이루어지지 않는 가상의 컴퓨터를 말한다. 이러한 가상머신은 시스템 가상머신과 프로세스 가상머신으로 크게 나누어지는데, 본 발명에서는 시스템 가상머신에 대한 가상환경을 중심으로 설명하도록 한다.

[0035] 이러한 시스템 가상머신은 완전한 시스템 플랫폼 즉, 완전한 운영체제(Operating System)의 실행을 지원하며, 각각의 운영체제를 실행하는 가상 머신 사이의 기초가 되는 물리 컴퓨터를 다중화(multiplex)한다. 이처럼, 가상화를 제공하는 소프트웨어 계층은 가상머신 모니터 또는 하이퍼바이저(Hypervisor)라고 하며, 상기 하이퍼바이저는 하드웨어 또는 호스트 운영체제의 위에서 실행할 수 있다.

[0036] 하둡(Hadoop)은 분산처리 플랫폼을 제공하는 아파치 오픈소스 프로젝트로 신뢰성(Reliability)과 확장성(Scalability)을 보장하는 환경을 제공하여 현재 아마존(Amazon), 야후(Yahoo!), 페이스북(Facebook) 등 많은 업체에서 사용되고 있다.

[0037] 이러한 하둡을 구성하는 중요한 요소 중 하나인 맵리듀스는 구글(Google)에서 제안한 맵리듀스 프로그래밍 모델을 하둡에 맞춰 구현한 소프트웨어 프레임워크로서, 분산환경에서 대용량 데이터를 처리하는데 있어서, 간단하고 직관적인 인터페이스를 제공한다.

[0038] 도 1은 하둡 맵리듀스 수행절차를 나타낸 도면이다.

[0039] 도 1에 도시된 바와 같이, 하둡 맵리듀스의 맵, 리듀스 함수는 사용자에 의해 작성되며, 입출력 값으로 Key-Value 쌍을 갖는다. 입력 데이터가 블록 단위로 분할되어 전달되면, 각 노드는 전달된 데이터를 입력받아 맵 함수를 수행하게 된다.

[0040] 이후, 각 노드가 상기 맵 함수의 결과값을 셔플(suffle)한 후 정렬하여 리듀스 함수로 전달하면, 리듀스 함수를 통해 정제된 데이터값을 가지고 사용자가 최종적으로 원하는 데이터가 추출된다.

[0041] 상기 각 맵과 리듀스 함수는 각 노드의 태스크 트랙커(Task Tracker)에 의해서 자식 프로세스로 수행되며, 상기 태스크 트랙커는 자신이 수행하고 있는 태스크의 상태를 주기적으로 마스터 노드에 전달하여 태스크가 종료되었을 때, 마스터 노드로부터 새로운 태스크를 할당받게 된다.

[0042] 도 2는 각 슬롯별 태스크 수행시간을 비교한 그래프이다.

[0043] 도 2에 도시된 바와 같이, 동일한 성능을 갖는 4개의 Amazon EC2 인스턴스로 맵리듀스 워드카운트(WordCount)를 수행한 결과를 확인할 수 있다. 특히, 컴퓨팅 파워가 동일한 머신임에도 불구하고, 데이터 지역성으로 인해 불균등한 수행시간을 나타내는 것을 알 수 있다. 이때, 상기 데이터 지역성(Data Locality)이란, 노드가 할당받은 태스크의 입력 데이터가 자신의 노드에 존재하지 않을 때, 다른 노드로부터 전송받아야 하는데, 이는 네트워크 오버헤드로 인해 태스크를 수행하는데 있어서 성능저하가 발생할 수 있다.

[0044] 특히, 모든 태스크가 종료되어야만 잡이 끝나는 맵리듀스의 특성상 전체 태스크 수행시간은 가장 늦게 끝나는 태스크의 처리 시간에 따라 달라지며, 이러한 특성은 맵리듀스에 심각한 성능저하를 초래한다.

[0045] 특히, 종래기술에 따른 추론적 실행은 낙오자 태스크의 잘못된 선별을 방지하기 위해, 낙오자 태스크를 다른 노드로 이전하여 수행하도록 하고, 이와 동시에 기존에 낙오자 태스크가 동작하던 노드에서도 계속 수행하게 하여 더 빨리 완료된 태스크의 결과를 수신하고, 수행속도가 느린 태스크는 강제 종료하는 매카니즘을 사용한다.

[0046] 도 3은 데이터 크기 증가에 따른 완료 태스크의 수와 강제 종료된 완료 태스크의 수를 나타낸 그래프이다.

[0047] 도 3에 도시된 바와 같이, 14개의 Amazon EC2 인스턴스에서 워드카운트 수행 시, 데이터 크기가 증가함에 따라 발생한 전체 태스크의 수를 나타낸다. 파란색 그래프와 빨간색 그래프를 비교해보면, 약 20% 정도의 강제 종료된 태스크가 발생한 것을 알 수 있다.

[0048] 즉, 전체 클러스터 자원을 고려해보면, 의미 없는 작업으로 인해 자원 낭비가 발생하였으며, 이로 인해 클라우드 서비스와 같이 사용량에 따라 과금이 책정되는 방식의 경우에는 더 비싼 비용을 지불해야만하는 문제점이 발생했다.

[0049] 이하, 도 4를 참고하여, 본 발명의 일 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법에 대하여 자세히 살펴보도록 한다.

[0050] 도 4는 본 발명의 일 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법을 나타낸 순서도이다.

[0051] 도 4에 도시된 바와 같이, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법은 먼저, 가상머신 내 태스크 수행완료시간 수신부가 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신한다(S110).

[0052] 이후, 상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산한다(S120).

[0053] 상기 가상머신이 연산한 남은시간을 상기 가상머신을 제어하는 마스터노드에 전달한다(S130).

[0054] 이에 따라, 상기 마스터노드 내 평균값 연산부가 수신한 복수 개의 남은 시간에 대하여 평균값을 연산한다(S140).

[0055] 상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 가상머신의 슬롯별 태스크 수행시간이 동일하도록 CPU 할당량을 조절한다(S150).

[0056] 이하, 상기 과정 S150에 기재된 CPU 할당량 조절 방법에 대하여 보다 자세히 살펴보도록 한다.

[0057] 마스터노드가 각 가상머신 내 모든 슬롯이 현재부터 태스크 수행완료시간까지의 남은 시간을 앞서 연산한 평균값과 비교하여, 상기 남은 시간이 상기 평균값보다 낮으면 해당 가상머신의 점유율을 감소시키고, 상기 남은 시간이 상기 평균값보다 높으면 해당 가상머신의 점유율을 증가시킨다.

[0058] 뿐만 아니라, 동일한 물리머신에 속하는 다수의 가상머신별 각각의 남은 시간이 상기 평균값과 비교하여 기설정된 값보다 크거나 작은 경우에는, 상기 가상머신의 CPU에 대한 가중치를 재할당한다.

[0059] 또한, 동일한 가상머신 내 다수의 슬롯이 존재하는 경우에는 상기 슬롯의 태스크 수행에 따른 프로세스의 우선순위를 변경하여, CPU 할당량을 조절할 수 있다.

[0060] 상술한 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법을 적용한 시스템에 대하여 살펴보도록 한다.

[0061] 대표적인 가상화 플랫폼인 Xen에서 디폴트 스케줄러인 크레딧 스케줄러(Credit Scheduler)의 파라미터를 변경하여 동적으로 가상머신 사이의 CPU 자원을 조절할 수 있다. 이때, 상기 크레딧 스케줄러는 각 가상머신에 크레딧을 할당하여 가상머신이 크레딧을 모두 소모할 때까지 라운드-로빈(Round-Robin) 방식으로 물리 CPU를 점유하게 한다.

[0062] 이때, 사용되는 라운드-로빈 방식이란, CPU 시간을 태스크의 생성 순으로 차례로 분배하는 방식을 나타내는 것으로서, 태스크가 입출력 명령 등을 내밀었을 때, CPU 시간을 포기하고 다시 대시 행렬의 맨 뒤로 되돌려진다. 이러한 라운드 로빈 방식은 각각의 프로세스에게 차례대로 일정한 단위시간 동안 물리 CPU를 차지하도록 한다. 뿐만 아니라, 상기 라운드 로빈 방식은 시분할 시스템(TSS)에서의 사용되는 알고리즘으로서, 하나의 업무를 중앙처리장치에 대하여 라운드 로빈 방식으로 할당하여 마치 각가의 업무가 CPU를 점유하고 있는 것처럼, 시분할하여 사용하는 방식을 말한다.

[0063] 이때, 상기 크레딧의 값은 각 가상머신에 부여된 가중치 또는 점유율에 의해 계산될 수 있다. 이때, 상기 가중치는 가상머신과 가상머신 간에 상대적인 값을 나타내는데, 예를 들어, 가상머신 1(VM1)에 가중치가 256이고, 가상머신 2(VM2)에 가중치가 512일 때, 상기 가상머신 2(VM2)는 상기 가상머신 1(VM1) 보다 2배 많은 크레딧을 가지게 되어 2배 많은 CPU 시간을 점유하게 된다.

[0064] 또한, 상기 점유율은 가상머신이 점유할 수 있는 CPU의 절대 값을 백분율로 나타내는 것으로서, 만약 싱글코어에서 가상머신1(VM1)의 점유율이 50이면, 상기 가상머신 1(VM1)은 단독으로 수행되어도 50% 이상의 CPU 점유율을 가질 수 없음을 나타낸다.

[0065] 이러한 상기 가중치 및 점유율은 동적으로 변경이 가능하기 때문에 이러한 두 파라미터를 이용하여 가상화 클러스터의 CPU 자원을 조절할 수 있는 것이다.

[0066] 도 5는 부하 분산 시스템의 전체 구조를 나타낸 블록도이다.

[0067] 도 5에 도시된 바와 같이, 각 가상머신은 하둡 플랫폼으로부터 태스크가 수행되고 있는 모든 슬롯의 완료시간을 수신하여 수행완료까지 남은 시간(RT)을 연산한다. 이후, 상기 가상머신은 남은시간(RT)을 마스터 노드에 전달한다.

[0068] 이에 따라, 상기 마스터 노드는 각 가상머신과 물리머신 기준으로 앞서 가상머신이 연산한 남은시간(RT)들의 평균값을 연산한다.

[0069] 상기 마스터 노드는 이와 같이, 연산한 평균값을 이용하여 태스크의 전체 수행시간을 동일하게 조정하기 위한 부하분산 과정을 수행한다.

[0070] 먼저, 첫 번째 부하분산과정을 살펴보면, 물리머신과 물리머신간에 현재부터 모든 태스크의 수행완료시간까지의 남은시간(RT)들의 평균값을 상기 남은시간과 비교하여 수행속도가 더 빠른 물리머신에 속해있는 가상머신의 점유율을 감소시키고, 수행속도가 느린 물리머신에 속해있는 가상머신의 점유율을 증가시킨다. 이때, 상기 점유율은 상기 평균값의 편차 비율만큼 백분율로 환산하여 할당된다.

[0071] 이에 따라, 동일한 CPU 자원을 소모하지만 보다 빠른 맵리듀스 결과를 도출해 낼 수 있다.

[0072] 만약 이때, 상기 평균값이 기설정된 한계값 이상으로 차이나는 경우에는 태스크의 수행속도가 느린 물리머신에 장애가 발생한 것으로 판단하고, 장애 처리를 위해 해당 물리머신에 속해 있는 모든 가상머신의 태스크를 앞서 설명한 추론적 실행을 통해 다른 물리머신으로 이주시킨다.

[0073] 이어서, 두 번째 부하분산과정은 하나의 물리머신에서 각 가상머신에 대한 남은시간의 평균값이 서로 차이나는 경우에는 평균값의 편차 비율만큼 각 가상머신의 가중치를 재할당한다. 이에 따라, 상대적으로 수행속도가 느린 가상머신은 보다 많은 CPU 자원을 할당 받게 되므로, 가상머신 사이에 부하분산이 이루어진다.

[0074] 마지막으로, 세 번째 부하 분산과정을 살펴보도록 한다.

[0075] 하둡 맵리듀스는 CPU 코어 수보다 1~2개 더 많은 수의 태스크 슬롯을 갖는 것을 권장하기 때문에, 하나의 노드당 최소 2개 이상의 태스크 슬롯을 갖는다. 이러한 경우, 가중치 또는 점유율의 조절은 가상머신의 CPU자원에만 영향을 미치므로, 하나의 가상머신에 두 개 이상의 태스크 슬롯이 존재하는 경우에는 슬롯간 부하의 균형을 맞추기 어렵다. 이에 따라, 리눅스 스케쥴러에서 프로세스의 우선순위를 변경하는 renice를 명령어를 사용함으로써, 태스크는 태스크 트랙커의 자식 프로세스로 동작하기 때문에 프로세스의 우선순위가 변경되어 슬롯간 수행시간의 균형을 맞출 수 있다.

[0076] 도 6은 본 발명의 다른 실시 예에 따른 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템을 나타낸 블록도이다.

[0077] 도 6에 도시된 바와 같이, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템을 구축하기 위해 필요한 컴포넌트 요소를 나타낸다.

[0078] 이러한 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 시스템은 태스크 수행완료시간 수신부, 남은시간 연산부, 남은시간 전송부, 평균값 연산부 및 제어부를 포함한다.

[0079] 태스크 수행완료시간 수신부는 복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신한다.

[0080] 남은시간 연산부는 상기 가상머신이 현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산한다.

[0081] 남은시간 전송부는 상기 가상머신이 연산한 남은시간을 마스터노드에 전달한다.

[0082] 평균값 연산부는 상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산한다.

[0083] 제어부는 상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어한다. 이러한 제어부는 비교모듈, 점유율 변경모듈, 재할당모듈 및 우선순위 변경모듈을 포함한다.

[0084] 비교모듈은 상기 마스터노드가 각 가상머신 내 모든 슬롯의 태스크 수행완료시간까지의 남은 시간을 평균값과 비교한다.

[0085] 점유율 변경모듈은 상기 남은 시간이 상기 평균값보다 낮은 경우에는 해당 가상머신의 점유율을 감소시키거나, 상기 남은 시간이 상기 평균값보다 높은 경우에는 해당 가상머신의 점유율을 증가시킨다.

[0086] 재할당모듈은 동일한 물리머신 내 속하는 다수의 가상머신별 각 남은 시간이 상기 평균값과 비교하여 기설정된 값보다 크거나 작은 경우에는, 상기 가상머신의 CPU에 대한 가중치를 재할당한다.

[0087] 우선순위 변경모듈은 동일한 가상머신 내 다수의 슬롯이 존재하는 경우에는 상기 슬롯의 태스크 처리에 따른 프로세스의 우선순위를 변경한다.

[0088] 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 다수의 가상머신이 맵리듀스 태스크 수행 시, 균일한 태스크 수행시간이 소요되도록 함으로써, 클러스터 환경에서의 하둡 맵리듀스의 성능 향상을 기대할 수 있는 효과가 있다.

[0089] 또한, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 크레딧 스케줄러의 가중치 또는 점유율 변경을 통해 부하분산이 이루어지도록 함으로써, 하둡 플랫폼과 하이퍼바이저에 대한 수정은 이루어지지 않아, 맵 리듀스 뿐만 아니라 다양한 하둡 응용프로그램에서도 적용할 수 있어, 소요비용을 절약할 수 있는 효과가 있다.

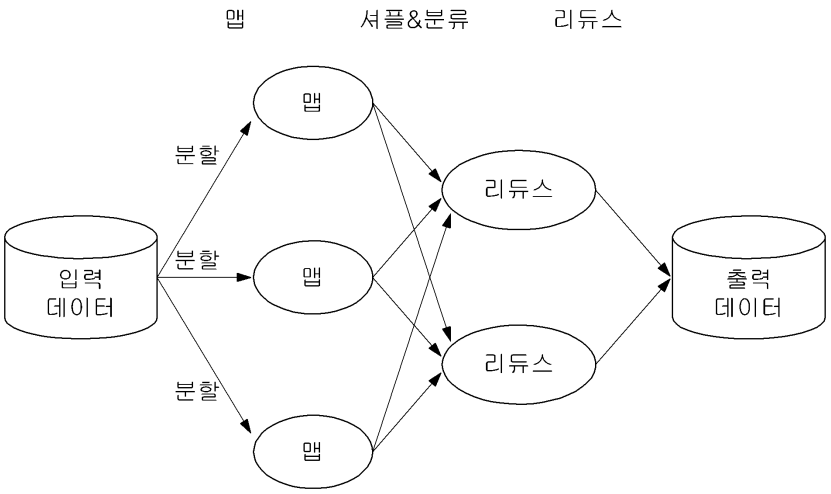
[0090] 더불어, 본 발명의 가상화 환경 내 하둡 맵리듀스의 부하 분산 방법 및 시스템은 각 가상머신마다 전체 태스크 수행시간을 동일하게 되도록 제어함으로써, 낙오자 태스크의 발생을 최소화시킬 수 있는 효과가 있다.

[0091] 상기에서는 본 발명의 바람직한 실시 예에 대하여 설명하였지만, 본 발명은 이에 한정되는 것이 아니고 본 발명

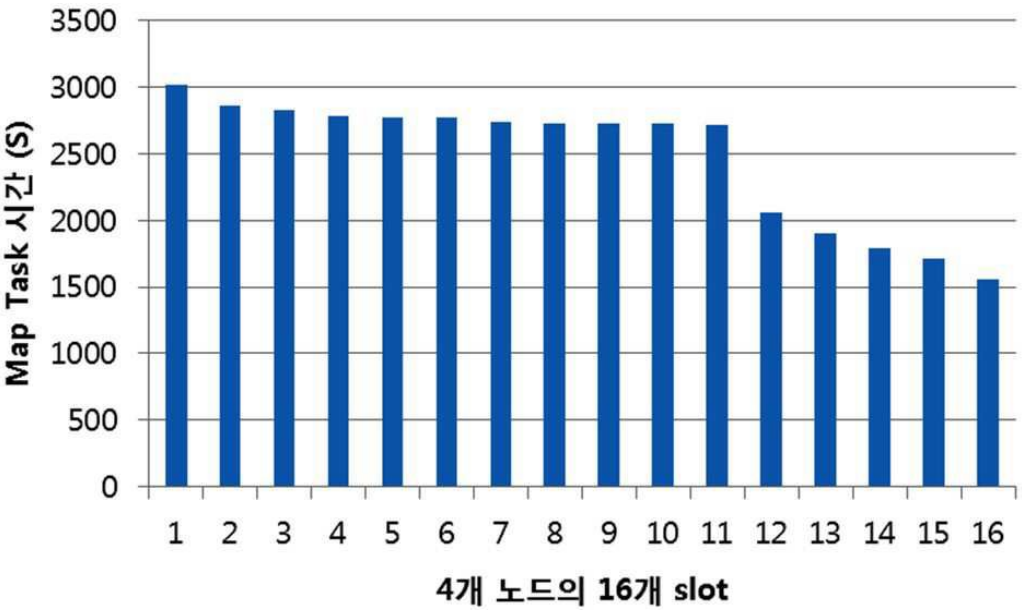
의 기술 사상 범위 내에서 여러 가지로 변형하여 실시하는 것이 가능하고 이 또한 첨부된 특허청구범위에 속하는 것은 당연하다.

도면

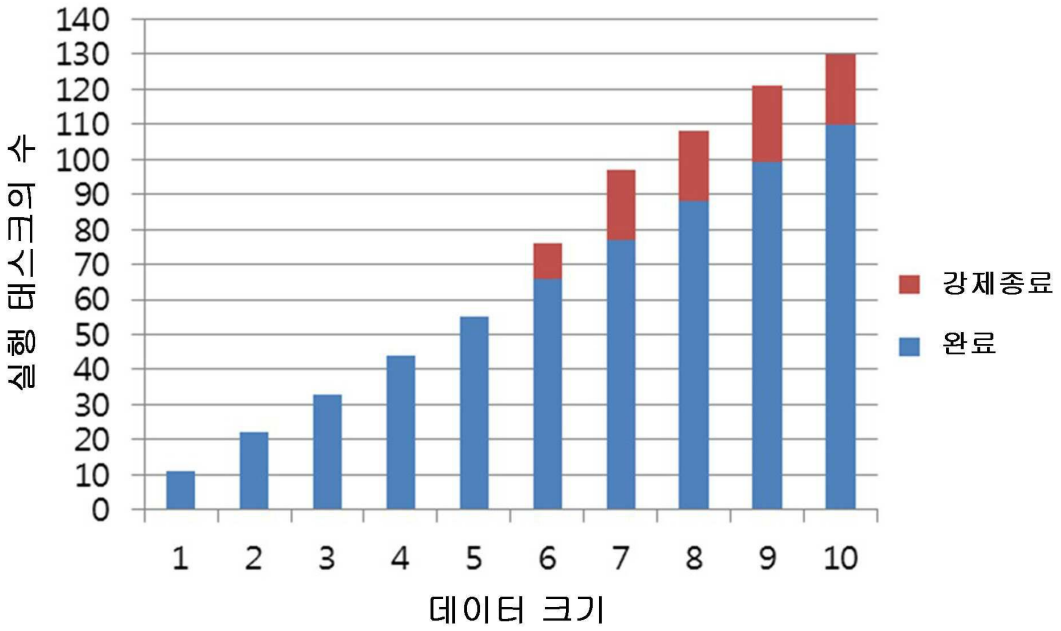
도면1



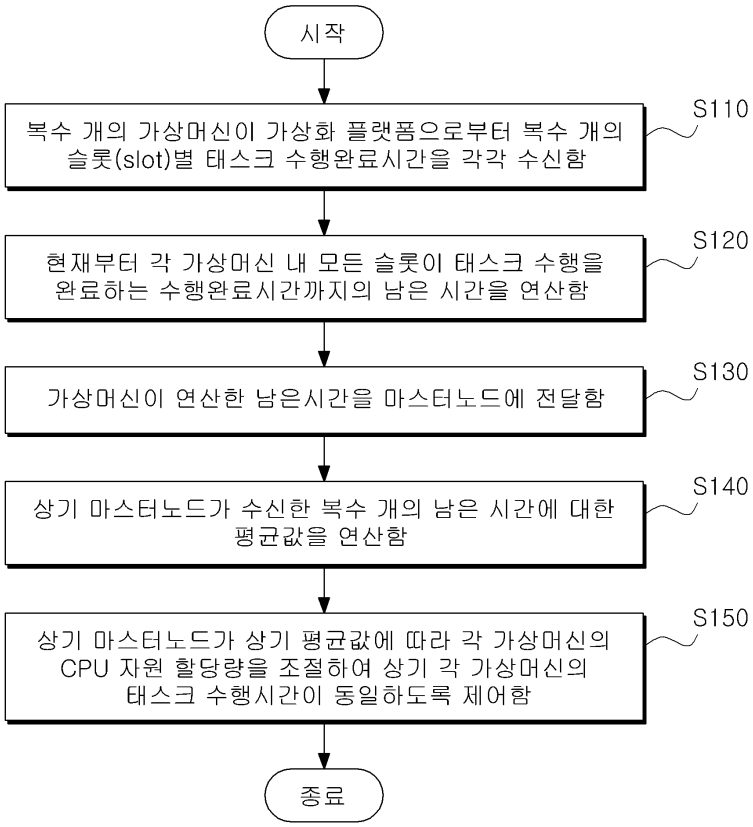
도면2



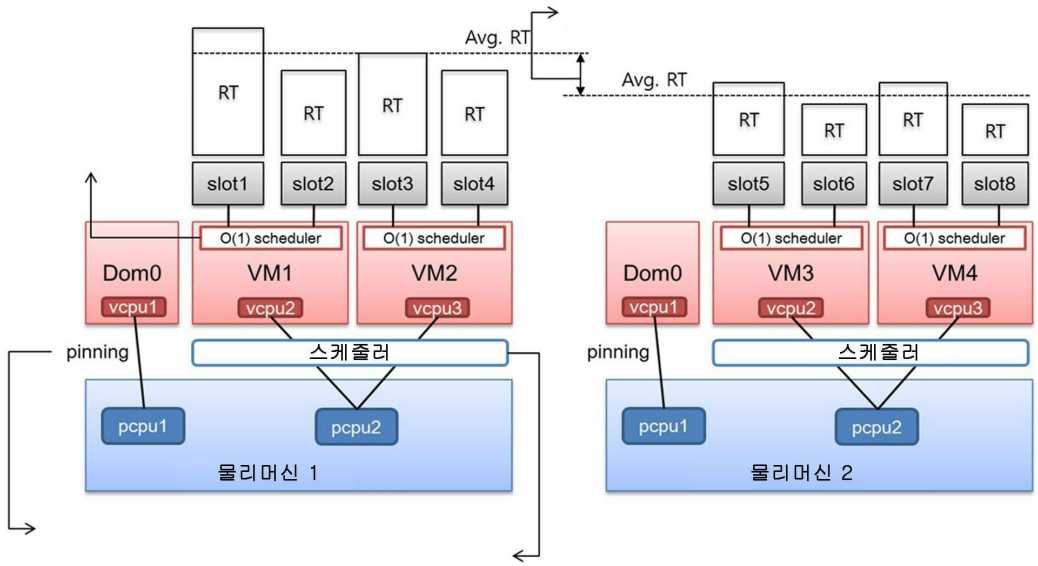
도면3


실행 태스크의 수
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
데이터 크기
강제종료
완료

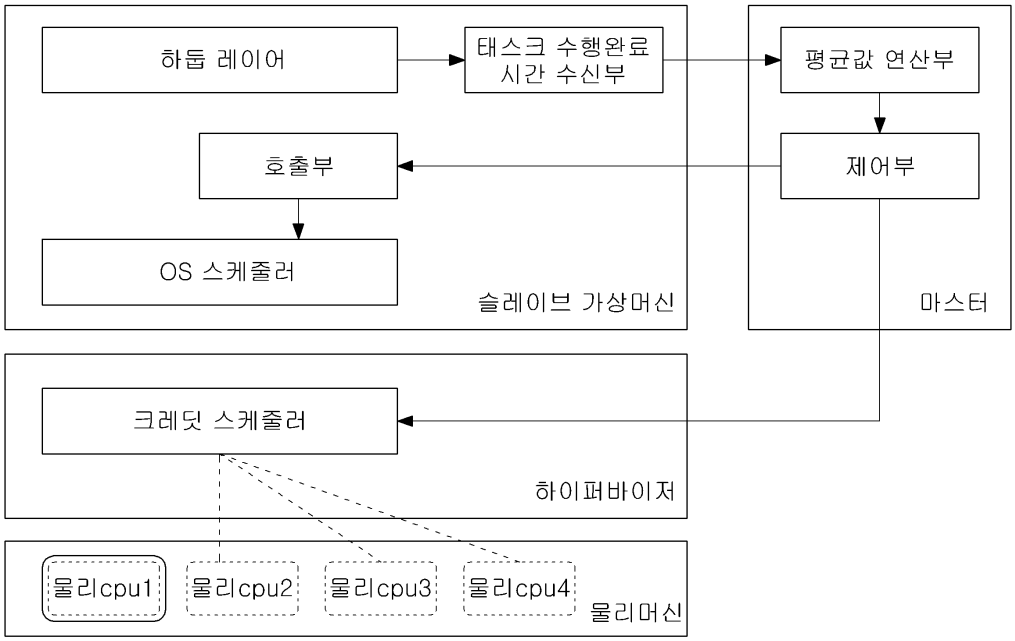
도면4


시작
복수 개의 가상머신이 가상화 플랫폼으로부터 복수 개의 슬롯(slot)별 태스크 수행완료시간을 각각 수신함
S110
현재부터 각 가상머신 내 모든 슬롯이 태스크 수행을 완료하는 수행완료시간까지의 남은 시간을 연산함
S120
가상머신이 연산한 남은시간을 마스터노드에 전달함
S130
상기 마스터노드가 수신한 복수 개의 남은 시간에 대한 평균값을 연산함
S140
상기 마스터노드가 상기 평균값에 따라 각 가상머신의 CPU 자원 할당량을 조절하여 상기 각 가상머신의 태스크 수행시간이 동일하도록 제어함
S150
종료

도면5


Avg. RT
Avg. RT
RT
RT
RT
RT
RT
RT
RT
RT
slot1
slot2
slot3
slot4
slot5
slot6
slot7
slot8
O(1) scheduler
O(1) scheduler
O(1) scheduler
O(1) scheduler
Dom0
VM1
VM2
Dom0
VM3
VM4
vcpu1
vcpu2
vcpu3
vcpu1
vcpu2
vcpu3
pinning
스케줄러
pinning
스케줄러
pcpu1
pcpu2
pcpu1
pcpu2
물리머신 1
물리머신 2

도면6


하둡 레이어
태스크 수행완료 시간 수신부
평균값 연산부
호출부
제어부
OS 스케줄러
슬레이브 가상머신
마스터
크레딧 스케줄러
하이퍼바이저
물리cpu1
물리cpu2
물리cpu3
물리cpu4
물리머신