



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0080794
(43) 공개일자 2014년07월01일

(51) 국제특허분류(Int. Cl.)
G06F 1/26 (2006.01) G06F 1/28 (2006.01)
G06F 1/32 (2006.01)
(21) 출원번호 10-2012-0148586
(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
임현중
서울 강동구 상암로 343, 1동 302호 (상일동, 효성빌라)
(74) 대리인
특허법인충현

전체 청구항 수 : 총 16 항

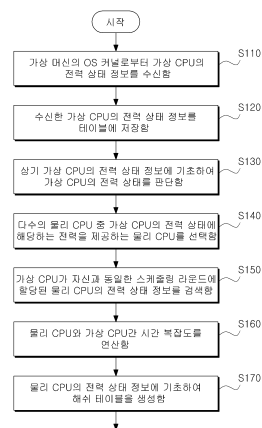
(54) 발명의 명칭 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템

(57) 요약

본 발명은 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템에 관한 것으로, 보다 구체적으로는 물리(Physical) CPU가 가상머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신하는 단계; 상기 물리 CPU가 수신한 가상 CPU의 전력 상태 정보에 기초하여 상기 가상 CPU의 전력 상태를 판단하는 단계; CPU선택부가 다수의 물리 CPU 중 상기 가상 CPU의 전력 상태에 해당하는 전력을 제공하는 적어도 하나의 물리 CPU를 선택하는 단계; 및 CPU매칭부가 선택된 상기 물리 CPU와 상기 가상 CPU를 상호 매칭시키는 단계;를 포함한다.

이러한 구성에 의해, 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템은 다수의 가상 CPU가 하나의 물리 CPU에 매칭되는 가상화 환경 내에서 서로 다른 전력량을 요구하는 가상 CPU가 해당 전력량을 제공할 수 있는 물리 CPU로 매칭됨에 따라, 시스템의 성능이 저하되거나, 불필요한 전력이 사용되는 것을 방지함으로써, 전력을 효율적으로 사용할 수 있는 효과가 있다.

대표도 - 도1a



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

물리(Physical) CPU가 가상머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신하는 단계;

상기 물리 CPU가 수신한 가상 CPU의 전력 상태 정보에 기초하여 상기 가상CPU의 전력 상태를 판단하는 단계;

CPU선택부가 다수의 물리 CPU 중 상기 가상 CPU의 전력 상태에 해당하는 전력을 제공하는 적어도 하나의 물리 CPU를 선택하는 단계; 및

CPU매칭부가 선택된 상기 물리 CPU와 상기 가상 CPU를 상호 매칭시키는 단계;

를 포함하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 2

제1항에 있어서,

상기 가상 CPU의 전력 상태 정보를 수신하는 단계는

상기 물리 CPU가 상기 가상 CPU의 전력 상태 정보를 수신한 후, 테이블에 저장하는 과정을 더 포함하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 3

제1항에 있어서,

상기 가상 CPU의 전력 상태를 판단하는 단계는

상기 가상 CPU의 전력 상태 정보 중 전압 또는 주파수가 기설정된 임계값을 초과하는 경우에는 상기 가상 CPU가 구동중이라고 판단하거나,

상기 가상 CPU의 전력 상태 정보 중 전압 또는 주파수가 기설정된 임계값 보다 같거나 작은 경우에는 상기 가상 CPU가 아이들(idle) 상태라고 판단하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 4

제1항에 있어서,

상기 물리 CPU를 선택하는 단계는

상기 가상 CPU가 자신과 동일한 스케줄링 라운드에 할당된 적어도 하나의 물리 CPU의 전력 상태 정보를 검색하는 과정; 및

상기 가상 CPU의 현재 전력 상태에 해당하는 전력을 제공할 수 있는 물리 CPU를 판단하도록 상기 물리 CPU와 가상 CPU간 시간 복잡도를 연산하는 과정;

을 포함하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 5

제4항에 있어서,

상기 물리 CPU와 가상 CPU간 시간 복잡도를 연산하는 과정은

상기 물리 CPU와 가상 CPU간 연산한 시간 복잡도를 해쉬 테이블을 이용하여 최저 시간 복잡도에 근사화시키는 과정을 더 포함하는 것을 특징으로 하는 가상환경 내 중앙 처리 장치의 전력 스케줄링 방법.

청구항 6

제5항에 있어서,

상기 물리 CPU와 가상 CPU간 시간 복잡도를 연산하는 과정은

상기 물리 CPU의 전력 상태 정보에 기초하여 해쉬 테이블을 생성하는 과정;

기설정된 역치값에 기초하여 상기 해쉬 테이블 내 해쉬 버킷에 물리 CPU를 매칭하는 과정; 및

상기 가상 CPU가 상기 물리 CPU를 해쉬 버킷과 연결시킨 해쉬함수와 동일한 함수를 갖는 물리 CPU를 검색하는 과정;

을 포함하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 7

제6항에 있어서,

상기 해쉬 버킷의 수는

현재 구동중인 물리 CPU의 개수 중 1/2로 지정되는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 8

제1항에 있어서,

상기 물리 CPU의 전력 상태 정보에 따라 가상 CPU를 상호 매칭시키는 단계는

적어도 하나의 가상 CPU의 전력 상태 정보에 대한 평균값을 연산하는 과정;

각 물리 CPU의 전력 상태 정보를 상기 가상 CPU의 전력 상태 정보의 평균값과 비교하여 상기 물리 CPU의 전력 상태 정보를 조정하는 과정; 및

전력 상태 정보가 조정된 상기 물리 CPU와 가상 CPU를 상호 매칭하는 과정;

을 포함하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 9

제8항에 있어서,

상기 물리 CPU의 전력 상태 정보를 조정하는 과정은

상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 기설정된 값보다 작은 경우에는, 상기 물리 CPU의 현재 전력 상태 정보를 유지하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 10

제8항에 있어서,

상기 물리 CPU의 전력 상태 정보를 조정하는 과정은

상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 기설정된 값보다 큰 경우에는 상기 물리 CPU의 현재 전력 상태 정보를 역치만큼 증가시키는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 11

제8항에 있어서,

상기 물리 CPU의 전력 상태 정보를 조정하는 과정은

상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 상기 기설정된 값의 마이너스 값보다 큰 경우에는 상기 물리 CPU의 현재 전력 상태 정보를 역치만큼 감소시키는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 12

제8항에 있어서,

상기 물리 CPU와 가상 CPU간 상호 매칭 후, 상기 물리 CPU의 전력 상태 정보를 재조정하는 과정;

을 더 포함하는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 13

제12항에 있어서,

상기 물리 CPU의 전력 상태 정보를 재조정하는 과정은

상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 값이 0 보다 큰 경우에는 상기 물리 CPU의 전력 상태 정보를 역치만큼 증가시키는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 14

제12항에 있어서,

상기 물리 CPU의 전력 상태 정보를 재조정하는 과정은

상기 가상 CPU의 전력 상태 정보의 평균값에서, 상기 물리 CPU의 현재 전력 상태 정보를 차감한 값이 0 보다 작은 경우에는 상기 물리 CPU의 전력 상태를 역치만큼 감소시키는 것을 특징으로 하는 가상환경 내 중앙처리장치의 전력 스케줄링 방법.

청구항 15

제1항 내지 제14항 중 어느 한 항에 따른 방법을 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체.

청구항 16

가상 머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신하고, 수신한 가상 CPU의 전력 상태 정보에 기초하여 가상 CPU의 전력 상태를 판단하는 물리(Physical) CPU;

다수의 물리 CPU 중 가상 CPU의 전력 상태에 해당하는 전력을 제공하는 적어도 하나의 물리 CPU를 선택하는 CPU 선택부; 및

선택된 물리 CPU와 가상 CPU를 상호 매칭하는 CPU 매칭부;

를 포함하는 가상환경 내 중앙처리장치의 전력 스케줄링 시스템.

명세서

기술분야

[0001] 본 발명은 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템에 관한 것으로, 특히 가상 환경에서의 다수의 가상 CPU에 대한 전력 스케줄링을 효율적으로 수행할 수 있는 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템에 관한 것이다.

배경기술

[0002] 오늘날 컴퓨터 산업의 주요한 경향 중 하나로서, 그린 컴퓨팅이 많은 관심을 받고 있다. 이러한 그린 컴퓨팅(Green Computing)이란, 컴퓨팅 작업에 소모되는 에너지를 줄여서 전력 생산으로 인해 발생하는 환경 오염 및 온실 가스 배출을 억제하고자 하는 것을 나타낸다. 이러한 그린 컴퓨팅을 구현하기 위해, CPU 및 GPU 등 컴퓨팅에 관련된 각종 프로세서들을 재설계하는 등 전력을 효율적으로 사용하기 위한 기술들이 연구되고 있다.

[0003] 이처럼, 그린 컴퓨팅을 구현하기 위해, 하드웨어 설계 쪽에서는 제작 공정의 미세화, 다중 코어 구조로의 설계, 뿐만 아니라, 전력 관리 모듈이라는 구성이 추가 되어야하고, 또한 소프트웨어 설계 쪽에서는 기존 플랫폼을 가상화 플랫폼으로 전환 및 이전하는 방법이 주로 연구되고 있다.

[0004] 먼저, 제작 공정 미세화는 다이 사이즈(Die Size)를 최소화하여 발열을 억제하고, 발열로 인해 발생하는 에너지 손실을 최소화 하면서도 성능을 유지하고자 하는 접근 방식이다. 공정방식이 미세공정으로 변화함에 따라, 프로세서 구동에 필요한 전압이 낮춰지게 되고, 이에 따라, $p=cv^2f$ 로 표현되는 프로세서의 전력 소모량 공식에서 공급수로 표현되는 전압이 감소하게 되므로, 발열 및 전력 사용량을 감소시킬 수 있다.

[0005] 또한 다중 코어 구조 설계 역시 전력 소모량을 고려한 결과를 나타낸다. $p=cv^2f$ 로 표현되는 전력 소모량 공식에서 하나의 프로세서에 전압이 물리게 되면, 전압의 제곱에 비례하여 전력 소모량(P)이 증가하는 것을 알 수 있다. 예를 들면, 2개의 코어에서 각각 절반 가량의 전압을 통해 일을 절반으로 나눠서 처리하는 경우, 다시 말해, 코어에 걸리는 부하가 1/2이 되는 경우, 전체 소모되는 전력량은 기존의 소모되는 전력량에 비해 약 1/2이 된다. 따라서, 이상적인 다중 코어 환경에서는 단일 코어 환경과 대비하여 전력 소모량을 급감시킬 수 있다.

[0006] 한편, 소프트웨어 측면에서는 기존 시스템을 클라우드 및 가상화 플랫폼으로 전환, 즉 가상화 하는 것이 연구되고 있다.

[0007] 최근에는, 컴퓨팅 환경이 다중 코어 환경으로 전환됨에 따라, 컴퓨터가 예전처럼 시분할 방식을 이용하지 않으면서도 여러 가지 작업을 동시에 수행할 수 있고, 또한 다수의 운영체제를 동시에 구동할 수 있다. 이에 따라 복수 개의 PC에서 구동되는 작업을 가상 머신 환경을 통해 하나의 고성능 서버에서 구동함으로써, 상기 작업 수행 시, 소모되는 전력을 최소화 할 수 있다. 특히, 아마존에서 제공하는 AWS(Amazon Web Services), 구글에서 제공하는 구글 클라우드(Google Cloud) 등의 가상 데스크탑 서비스들은 이런 가상화 플랫폼 및 플랫폼 구성을 위한 대표적인 서비스이다.

[0008] 이러한 전력 효율 향상 방법은 단일 OS만을 고려한 방식일 뿐 가상화 환경, 즉 다수의 가상 CPU가 하나의 물리 CPU에 매핑(Mapping)되는 환경은 고려하지 않는 방식이다.

[0009] 결과적으로, 서로 다른 요구 전력을 갖는 가상 CPU가 동일한 전력량이 유지되는 물리 CPU에 매핑되는 경우, 성

능이 저하되고, 불필요한 전력이 사용되어, 전력이 낭비되는 문제점이 발생했다.

[0010] 상술한 바와 같이, 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템을 살펴보면 다음과 같다.

[0011] 선행기술 1은 한국등록특허공보 제10-1084876호(2011.11.11)로서, 운영서버에 구성된 CPU의 전체전력소모를 절감하는 방법에 관한 것이다. 이러한 선행기술 1은 (a) 메인메모리에서 어느 하나의 프로그램의 프로세스 처리를 요청받으면 상기 멀티코어 중 어느 하나의 코어로만 프로세스를 패치하여 처리하는 단계; (b) 메인메모리에서 다른 하나의 프로그램의 프로세스 처리를 요청받으면 상기 패치된 코어의 처리용량을 상기 운영체제가 보유하고 있는 CPU Management library로 스캔하는 단계; 및 (c) 상기 스캔된 코어의 처리용량이 상기 운영체제 상에 기 설정값 초과여부에 따라 상기 다른 하나의 프로그램의 프로세서를 처리하기 위한 코어가 추가로 패치되는 단계;를 포함한다. 본 발명을 통해 다중 스레드환경의 운영체제하에서 운영서버에 구성된 CPU의 멀티코어를 프로세스의 처리용량에 따라 코어단위로 조절하여 현재 프로세스를 처리하는데 필요한 적정한 코어만이 동작하도록 하여 운영서버의 CPU의 소비전력을 최소화할 수 있는 효과가 있다.

[0012] 또한, 선행기술 2는 한국공개특허공보 제10-2004-0058788호(2004.07.05)로서, CPU의 전력 제어방법에 관한 것이다.

[0013] 이러한 선행기술 2는 운영체제 고유의 스케줄에 의한 유휴시간이 특정 임계값보다 큰가를 판단하는 제1 단계와; 상기 판단결과, 유휴시간이 특정 임계값보다 크면 가(假) 취침모드로 변환하는 제2 단계와; 상기 가(假) 취침모드시 인터럽트 신호가 발생하는지를 판단하여 인터럽트 신호가 발생하면 상기 가(假) 취침모드를 해제하는 제3 단계와; 상기 판단결과, 상기 가(假) 취침모드시 인터럽트 신호가 발생하지 않으면 그 가(假) 취침모드를 계속 유지하며 실시간 시간 인터럽트를 이용하여 설정된 시간에 상기 가(假) 취침모드를 해제한다.

발명의 내용

해결하려는 과제

[0014] 상기와 같은 종래 기술의 문제점을 해결하기 위해, 본 발명은 가상 CPU의 전력 상태에 따라 해당 전력을 제공할 수 있는 물리 CPU를 상기 가상 CPU와 매칭시켜 전력을 효율적으로 사용할 수 있는 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템을 제공하고자 한다.

과제의 해결 수단

[0015] 위와 같은 과제를 해결하기 위한 본 발명의 한 실시 예에 따른 가상환경 내 중앙처리장치의 전력 스케줄링 방법은 물리(Physical) CPU가 가상머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신하는 단계; 상기 물리 CPU가 수신한 가상 CPU의 전력 상태 정보에 기초하여 상기 가상CPU의 전력 상태를 판단하는 단계; CPU선택부가 다수의 물리 CPU 중 상기 가상 CPU의 전력 상태에 해당하는 전력을 제공하는 적어도 하나의 물리 CPU를 선택하는 단계; 및 CPU매칭부가 선택된 상기 물리 CPU와 상기 가상 CPU를 상호 매칭시키는 단계;를 포함한다.

[0016] 특히, 상기 물리 CPU가 상기 가상 CPU의 전력 상태 정보를 수신한 후, 테이블에 저장하는 과정을 더 포함하는 가상 CPU의 전력 상태 정보를 수신하는 단계를 포함할 수 있다.

[0017] 보다 바람직하게는 상기 가상 CPU의 전력 상태 정보 중 전압 또는 주파수가 기설정된 임계값을 초과하는 경우에는 상기 가상 CPU가 구동중이라고 판단하거나, 상기 가상 CPU의 전력 상태 정보 중 전압 또는 주파수가 기설정된 임계값 보다 같거나 작은 경우에는 상기 가상 CPU가 아이들(idle) 상태라고 판단하는 가상 CPU의 전력 상태를 판단하는 단계를 포함할 수 있다.

[0018] 보다 바람직하게는 상기 가상 CPU가 자신과 동일한 스케줄링 라운드에 할당된 적어도 하나의 물리 CPU의 전력 상태 정보를 검색하는 과정; 및 상기 가상 CPU의 현재 전력 상태에 해당하는 전력을 제공할 수 있는 물리 CPU를 판단하도록 상기 물리 CPU와 가상 CPU간 시간 복잡도를 연산하는 과정;을 포함하는 물리 CPU를 선택하는 단계를 포함할 수 있다.

[0019] 특히, 상기 물리 CPU와 가상 CPU간 연산한 시간 복잡도를 해쉬 테이블을 이용하여 최저 시간 복잡도에 근사화시

키는 과정을 더 포함하는 물리 CPU와 가상 CPU간 시간 복잡도를 연산하는 과정을 포함할 수 있다.

- [0020] 보다 바람직하게는 상기 물리 CPU의 전력 상태 정보에 기초하여 해쉬 테이블을 생성하는 과정; 기설정된 역치값에 기초하여 상기 해쉬 테이블 내 해쉬 버킷에 물리 CPU를 매칭하는 과정; 및 상기 가상 CPU가 상기 물리 CPU를 해쉬 버킷과 연결시킨 해쉬함수와 동일한 함수를 갖는 물리 CPU를 검색하는 과정; 을 포함하는 물리 CPU와 가상 CPU간 시간 복잡도를 연산하는 과정을 포함할 수 있다.
- [0021] 특히, 현재 구동중인 물리 CPU의 개수 중 1/2로 지정되는 해쉬 버킷의 수를 포함할 수 있다.
- [0022] 보다 바람직하게는 적어도 하나의 가상 CPU의 전력 상태 정보에 대한 평균값을 연산하는 과정; 각 물리 CPU의 전력 상태 정보를 상기 가상 CPU의 전력 상태 정보의 평균값과 비교하여 상기 물리 CPU의 전력 상태 정보를 조정하는 과정; 및 전력 상태 정보가 조정된 상기 물리 CPU와 가상 CPU를 상호 매칭하는 과정; 을 포함하는 물리 CPU의 전력 상태 정보에 따라 가상 CPU를 상호 매칭시키는 단계를 포함할 수 있다.
- [0023] 특히, 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 기설정된 값보다 작은 경우에는, 상기 물리 CPU의 현재 전력 상태 정보를 유지하는 물리 CPU의 전력 상태 정보를 조정하는 과정을 포함할 수 있다.
- [0024] 특히, 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 기설정된 값보다 큰 경우에는 상기 물리 CPU의 현재 전력 상태 정보를 역치만큼 증가시키는 물리 CPU의 전력 상태 정보를 조정하는 과정을 포함할 수 있다.
- [0025] 특히, 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 상기 기설정된 값의 마이너스 값보다 큰 경우에는 상기 물리 CPU의 현재 전력 상태 정보를 역치만큼 감소시키는 물리 CPU의 전력 상태 정보를 조정하는 과정을 포함할 수 있다.
- [0026] 보다 바람직하게는 상기 물리 CPU와 가상 CPU간 상호 매칭 후, 상기 물리 CPU의 전력 상태 정보를 재조정하는 과정;을 더 포함할 수 있다.
- [0027] 특히, 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 값이 0 보다 큰 경우에는 상기 물리 CPU의 전력 상태 정보를 역치만큼 증가시키는 물리 CPU의 전력 상태 정보를 재조정하는 과정을 포함할 수 있다.
- [0028] 특히, 상기 가상 CPU의 전력 상태 정보의 평균값에서, 상기 물리 CPU의 현재 전력 상태 정보를 차감한 값이 0 보다 작은 경우에는 상기 물리 CPU의 전력 상태를 역치만큼 감소시키는 물리 CPU의 전력 상태 정보를 재조정하는 과정을 포함할 수 있다.
- [0029] 위와 같은 과제를 해결하기 위한 본 발명의 다른 실시 예에 따른 가상환경 내 중앙처리장치의 전력 스케줄링 시스템은 가상 머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신하고, 수신한 가상 CPU의 전력 상태 정보에 기초하여 가상 CPU의 전력 상태를 판단하는 물리(Physical) CPU; 다수의 물리 CPU 중 가상 CPU의 전력 상태에 해당하는 전력을 제공하는 적어도 하나의 물리 CPU를 선택하는 CPU 선택부; 및 선택된 물리 CPU와 가상 CPU를 상호 매칭하는 CPU 매칭부;를 포함한다.

발명의 효과

- [0030] 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템은 다수의 가상 CPU가 하나의 물리 CPU에 매칭되는 가상화 환경 내에서 서로 다른 전력량을 요구하는 가상 CPU가 해당 전력량을 제공할 수 있는 물리 CPU로 매칭됨에 따라, 시스템의 성능이 저하되거나, 불필요한 전력이 사용되는 것을 방지함으로써, 전력을 효율적으로 사용할 수 있는 효과가 있다.
- [0031] 또한 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템은 하나의 물리 CPU에 과부하가 걸리는 것을 방지함으로써, 전력 사용량이 급증하는 것을 방지할 수 있는 효과가 있다.

도면의 간단한 설명

- [0032] 도 1은 본 발명의 일 실시 예에 따른 가상환경 내 중앙처리장치의 전력 스케줄링 방법을 나타낸 순서도이다.

도 2는 가상 CPU의 전력 상태를 판단하는 기준을 나타낸 도면이다.

도 3은 가상 환경 내 가상 머신의 가상 CPU와 물리 CPU간 상관관계를 나타낸 도면이다.

도 4는 해쉬 테이블에 기초하여 가상 CPU가 적합한 물리 CPU를 검색하는 과정을 나타낸 도면이다.

도 5는 Xen을 기반으로 하는 부하 분산구조를 나타낸 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0033] 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.
- [0034] 먼저, 본 발명을 설명하기에 앞서, 본 발명이 구현되는 가상환경에 대하여 간략히 설명하도록 한다.
- [0035] 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.
- [0036] 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.
- [0037] 이러한 가상환경에서 구현되는 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법에 대하여 자세히 살펴보도록 한다.
- [0038] 도 1은 본 발명의 일 실시 예에 따른 가상환경 내 중앙처리장치의 전력 스케줄링 방법을 나타낸 순서도이다.
- [0039] 도 1에 도시된 바와 같이, 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법은 물리 컴퓨터의 중앙처리장치인 물리(Physical) CPU가 가상머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신한다(S110).
- [0040] 상기 물리 CPU는 상기 가상 CPU의 전력 상태 정보를 수신한 후, 수신한 가상 CPU의 전력 상태 정보를 테이블에 저장한다(S120).
- [0041] 상기 물리 CPU가 수신한 가상 CPU의 전력 상태 정보에 기초하여 상기 가상 CPU의 전력 상태를 판단한다(S130).
- [0042] 도 2는 가상 CPU의 전력 상태를 판단하는 기준을 나타낸 도면이다.
- [0043] 도 2에 도시된 바와 같이, 상기 가상 CPU의 전력 상태 정보 중 전압 또는 주파수가 기설정된 임계값을 초과하는 경우에는 상기 가상 CPU가 구동중이라고 판단하거나, 상기 가상 CPU의 전력 상태 정보 중 전압 또는 주파수가 기설정된 임계값 보다 같거나 작은 경우에는 상기 가상 CPU가 아이들(idle) 즉, 유휴 상태라고 판단한다.
- [0044] 이처럼, 도 2를 통해 알 수 있듯이, 결과적으로, 가상 CPU가 높은 전압과 주파수를 가질수록, 다시말해, 많은 전력을 사용할수록 더 작은 상태 값을 갖는 것을 알 수 있다. 예를 들면, 도 2에 도시된 P0는 P1보다 더 많은 전력을 소모하는 것을 알 수 있다.
- [0045] 다시 도 1로 돌아가서, CPU선택부가 다수의 물리 CPU 중 앞서서, 상기 과정 S130을 통해 판단한 상기 가상 CPU의 전력 상태에 해당하는 전력을 제공할 수 있는 적어도 하나의 물리 CPU를 선택한다(S140).
- [0046] 가상 머신 입장에서 살펴보면, 가상 머신을 구동시키는 가상 CPU는 자신의 물리 CPU와 동일한 것이나 마찬가지로, 이를 통해 가상 머신에서 구동중인 리눅스 커널이 DVFS(Dynamic Voltage and Frequency Scailing) 모듈을 이용하여, 자신의 물리 CPU의 상태를 변경시키는 것이 가상 CPU가 요구하는 전력 상태를 변경하는 것과 동일하다는 것을 알 수 있다. 다시 말해, 가상 머신의 커널이 변경시키는 가상 CPU의 전력 상태를 파악하여 상기 가상 CPU를 해당 상태의 전력 소모량을 가장 낮은 값으로 만족시키는 물리 CPU에 스케줄링 함으로써, 낭비되는 전력량을 최소화 할 수 있다.
- [0047] Xen의 성능 측면에서 공평성을 위해 사용되는 크레딧 스케줄러(Credit Scheduler)가 이미 존재함에 따라, 기존의 상기 크레딧 스케줄러와 전력 관리를 위한 스케줄러가 공존하기 위해, 가상 CPU를 물리 CPU의 실행 큐

(Queue)에 넣어주는 과정을 전력값을 기준으로 하여 구현하는 것이 바람직하다.

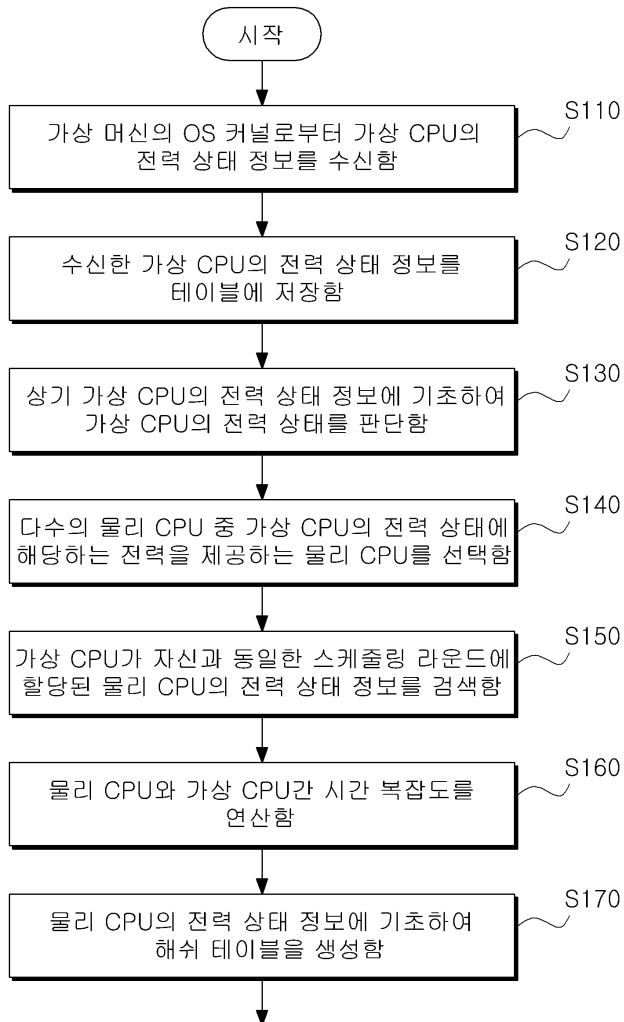
- [0048] 도 3은 가상 환경 내 가상 머신의 가상 CPU와 물리 CPU간 상관관계를 나타낸 도면이다.
- [0049] 도 3에 도시된 바와 같이, 물리 CPU를 선택하는 과정은 가상 CPU가 자신과 동일한 스케줄링 라운드에 할당된 적어도 하나의 물리 CPU의 전력 상태 정보를 모두 검색한다(S150).
- [0050] 특히, 상기 가상 CPU의 현재 전력 상태에 해당하는 전력을 물리 CPU가 제공할 수 있는지 여부를 판단하기 위해, 상기 물리 CPU와 가상 CPU간 시간 복잡도를 연산한다(S160). 예를 들어, 물리 CPU의 개수를 n 개(이때, 상기 n 은 자연수이다.)라고 하면, 같은 스케줄링 라운드에 존재하는 가상 CPU의 개수 역시 n 개이고, n 개의 가상 CPU가 n 개의 물리 CPU에 대한 전력 상태 정보를 모두 검색해야 하므로, 이러한 경우의 가상 CPU와 물리 CPU간 시간 복잡도는 $O(n^2)$ 이 된다.
- [0051] 이러한, 시간 복잡도의 연산과정은 Xen의 크레딧 스케줄러가 10ms 간격으로 스케줄링을 수행할 때마다 발생하게 되는데, 특히 앞서 연산한 시간 복잡도 $O(n^2)$ 는 성능 면에 있어서 악영향을 줄 가능성이 높기 때문에, 시간 복잡도가 보다 낮은 쪽에 가까운 값을 찾는 과정이 수행되어야 한다. 이에 따라, 앞서 과정 S160에서 연산한 상기 물리 CPU와 가상 CPU간 연산한 시간 복잡도를 해쉬 테이블(hash table)을 이용하여 최저 시간 복잡도에 근사화시킨다.
- [0052] 이처럼, 연산한 시간 복잡도를 최저 시간 복잡도에 근사화를 시키기 위해, 상기 물리 CPU의 전력 상태 정보에 기초하여 해쉬 테이블을 생성한다(S170). 즉, 상기 물리 CPU가 구동 중인 상태일 때의 전력 상태 정보를 이용하여 해쉬 테이블을 생성하는 것이 바람직하다.
- [0053] 이어서, 기설정된 역치값(T)을 기준으로 하여, 각각의 해쉬 버킷에 물리 CPU를 매칭시킨다(S180). 이때, 상기 해쉬 버킷의 수는 현재 구동 중인 물리 CPU의 전체 개수 중 1/2로 지정되는 것이 바람직하다.
- [0054] 이후, 가상 CPU가 물리 CPU를 상기 해쉬 버킷과 연결시킨 해쉬함수와 동일한 함수를 갖는 물리 CPU를 검색한다(S190). 이러한, 검색 결과, 상기 가상 CPU와 검색된 물리 CPU간 시간 복잡도는 상수 시간이 된다.
- [0055] 도 4는 해쉬 테이블에 기초하여 가상 CPU가 적합한 물리 CPU를 검색하는 과정을 나타낸 도면이다.
- [0056] 도 4에 도시된 바와 같이, 물리 CPU가 구동 중인 전력 상태 정보를 기초로 하여 작성된 해쉬 테이블로부터 가상 CPU가 자신이 필요한 전력을 제공받을 수 있는 물리 CPU를 검색하는 과정에 있어서, 실제 물리 CPU의 실행 큐에 들어가는 과정과 동일한 것을 알 수 있다.
- [0057] 특히, 하나의 물리 CPU에 전력이 지속적으로 집중되면, 해당 물리 CPU에 걸리는 부하가 커지게 되고, 뿐만 아니라, 단일 코어에서 작업을 돌리는 것과 유사한 형태가 되기 때문에 다중 코어 환경에서 얻을 수 있는 전력 효율이 반감된다. 따라서 하나의 물리 CPU에 전력이 편중되지 않도록 부하를 적절히 분산 해주는 것이 전력 효율성 측면에서 중요하다.
- [0058] 이처럼, 가상 CPU의 요구 전력 상태와 물리 CPU의 전력 상태간 차이로 인해 야기되는 전력 낭비 및 성능 감소를 최소화하기 위해, 모든 가상 CPU의 전력 상태 정보에 대한 평균값을 연산한다(S200).
- [0059] 이후, 각 물리 CPU의 전력 상태 정보를 상기 가상 CPU의 전력 상태 정보의 평균값과 비교하여 상기 물리 CPU의 전력 상태 정보를 조정한다(S210).
- [0060] 이때, 상기 물리 CPU의 전력 상태 정보의 조정과정은 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 기설정된 값보다 작은 경우에, 상기 물리 CPU의 현재 전력 상태 정보를 유지한다.
- [0061] 또는 상술한 바와 달리, 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 기설정된 값보다 큰 경우에는 상기 물리 CPU의 현재 전력 상태 정보를 역치만큼 증가시킨다.
- [0062] 또는 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 절대값이 상기 기설정된 값의 마이너스 값보다 큰 경우에는 상기 물리 CPU의 현재 전력 상태 정보를 역치만큼 감소시킨다.
- [0063] 이처럼, 전력 상태 정보가 조정된 상기 물리 CPU와 가상 CPU를 상호 매칭한다(S220).
- [0064] 하지만, 물리 CPU의 전력 상태 정보의 조정 과정에 있어서, 일괄적으로 전력 상태가 조정된 물리 CPU는 지나치

게 높은 전력 상태를 유지하는 경우에는 낮은 전력 상태인 가상 CPU의 성능이나 전력 효율을 감소시킬 수 있으므로, 가상 CPU와 매칭된 물리 CPU에 대하여 물리 CPU의 전력 상태를 재조정할 필요가 있다.

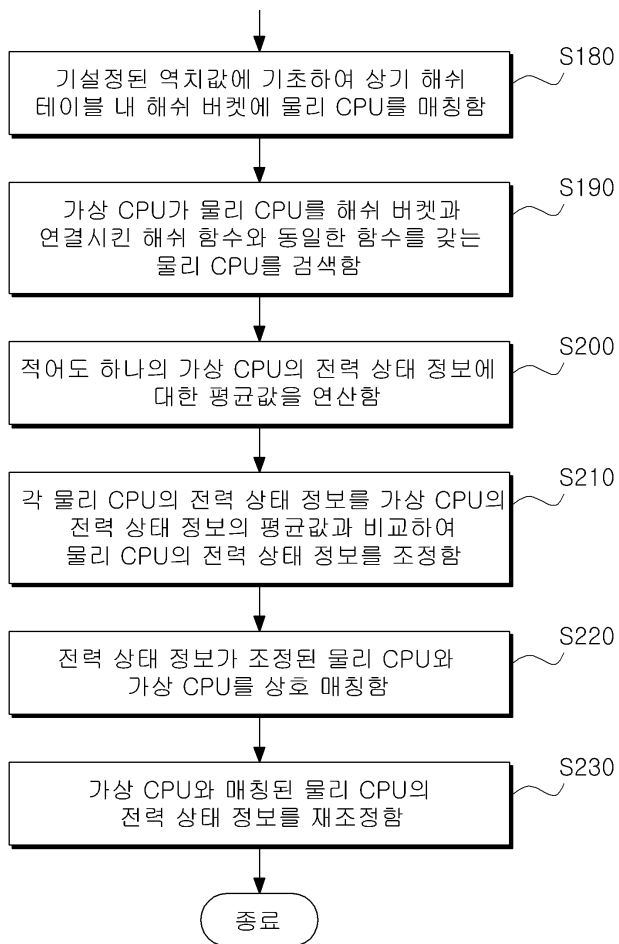
- [0065] 이에 따라, 가상 CPU와 매칭된 상기 물리 CPU에 대하여, 상기 물리 CPU의 전력 상태 정보를 재조정한다(S230). 예를 들어, 상기 가상 CPU의 전력 상태 정보의 평균값에서 상기 물리 CPU의 현재 전력 상태 정보를 차감한 값이 0 보다 큰 경우에는 상기 물리 CPU의 전력 상태 정보를 역치만큼 증가시킨다.
- [0066] 또는 상기 가상 CPU의 전력 상태 정보의 평균값에서, 상기 물리 CPU의 현재 전력 상태 정보를 차감한 값이 0 보다 작은 경우에는 상기 물리 CPU의 전력 상태를 역치만큼 감소시킨다.
- [0067] 앞서 수행된 조정을 통해서 이미 실행 큐에 들어와 있는, 조정된 물리 CPU에 적합하지 않은 가상 CPU에 매칭되는 경우, 물리 CPU의 상태를 재조정하여 상기 가상 CPU의 요구에 최대한 가까운 상태로 맞춰줄 수 있게 된다. 이러한 재조정 과정은 매 10ms의 스케줄링 라운드마다 다음 가상 CPU의 요구 전력 상태를 기준으로 진행된다.
- [0068] 이때, 상기 해쉬 테이블을 구성하는 키 값은 전력 상태를 기준으로 하여 구성되기 때문에 물리 CPU의 전력 부하가 분산되는 전력 기준 부하 분산과정이 이루어 질 때 마다 갱신되어야만 한다.
- [0069] 도 5는 Xen을 기반으로 하는 부하 분산구조를 나타낸 블록도이다.
- [0070] 도 5에 도시된 바와 같이, 가상머신의 커널로부터 수신한 가상 CPU의 전력 상태 정보를 DOM 0의 테이블에 저장한다. 저장된 테이블 내 전력 상태 정보를 이용하여 하이퍼바이저에서 상기 가상 CPU의 전력을 스케줄링한다.
- [0071] 이후, 부하분산기가 상기 가상 CPU의 스케줄링 결과를 바탕으로 물리 CPU를 모니터링하다가, 상기 가상 CPU의 스케줄링 타이밍에 맞춰, 상기 물리 CPU의 전력 상태를 조정한다. 즉, 물리 CPU의 부하 분산기가 Xen Credit 스케줄러의 Credit이 재계산되는 30ms에 맞춰서 실행한다.
- [0072] 또한 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 시스템은 물리 CPU, CPU 선택부 및 CPU 매칭부를 포함한다.
- [0073] 물리 CPU는 가상 머신의 OS(Operating System) 커널로부터 가상(Virtual) CPU의 전력 상태 정보를 수신하고, 수신한 가상 CPU의 전력 상태 정보에 기초하여 가상 CPU의 전력 상태를 판단한다.
- [0074] CPU 선택부는 다수의 물리 CPU 중 가상 CPU의 전력 상태에 해당하는 전력을 제공하는 적어도 하나의 물리 CPU를 선택한다.
- [0075] CPU 매칭부는 선택된 물리 CPU와 가상 CPU를 상호 매칭한다.
- [0076] 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템은 다수의 가상 CPU가 하나의 물리 CPU에 매칭되는 가상화 환경 내에서 서로 다른 전력량을 요구하는 가상 CPU가 해당 전력량을 제공할 수 있는 물리 CPU로 매칭됨에 따라, 시스템의 성능이 저하되거나, 불필요한 전력이 사용되는 것을 방지함으로써, 전력을 효율적으로 사용할 수 있는 효과가 있다.
- [0077] 또한 본 발명의 가상환경 내 중앙처리장치의 전력 스케줄링 방법 및 시스템은 하나의 물리 CPU에 과부하가 걸리는 것을 방지함으로써, 전력 사용량이 급증하는 것을 방지할 수 있는 효과가 있다.
- [0078] 상기에서는 본 발명의 바람직한 실시 예에 대하여 설명하였지만, 본 발명은 이에 한정되는 것이 아니고 본 발명의 기술 사상 범위 내에서 여러 가지로 변형하여 실시하는 것이 가능하고 이 또한 첨부된 특허청구범위에 속하는 것은 당연하다.

도면

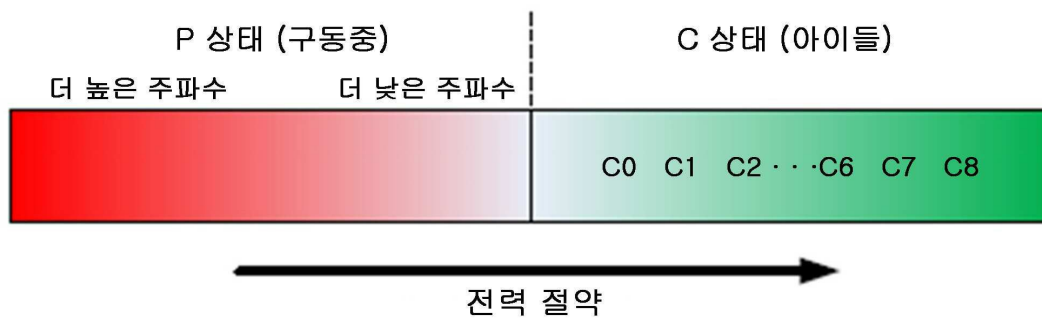
도면1a



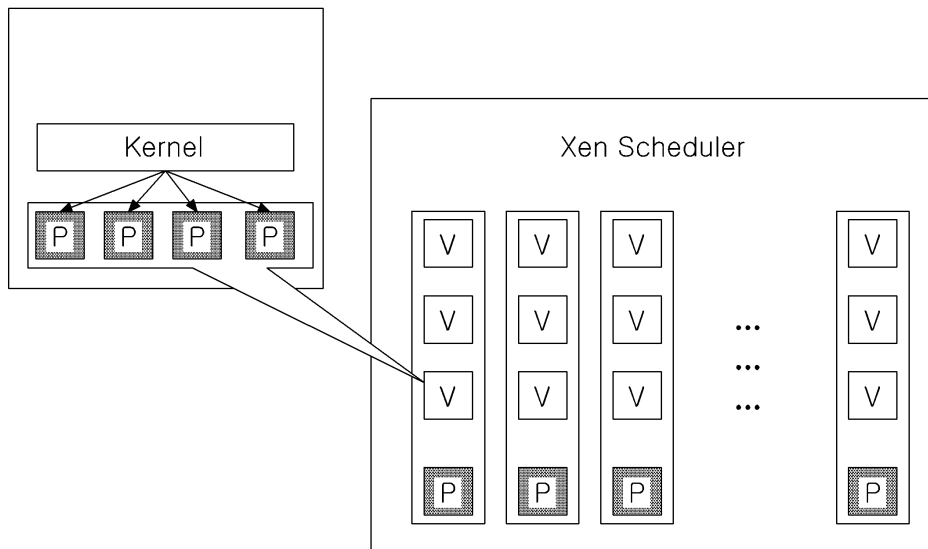
도면1b



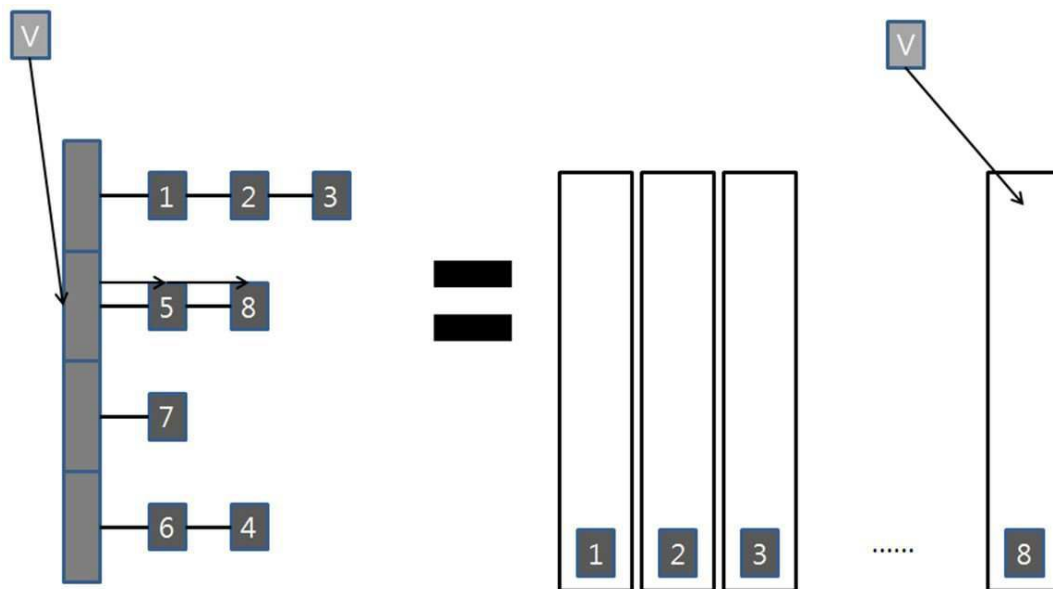
도면2



도면3



도면4



도면5

