



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0061393
(43) 공개일자 2020년06월02일

- (51) 국제특허분류(Int. Cl.)
G06F 9/455 (2018.01) G06F 8/658 (2018.01)
- (52) CPC특허분류
G06F 9/45558 (2013.01)
G06F 8/658 (2018.02)
- (21) 출원번호 10-2020-7012439
- (22) 출원일자(국제) 2018년09월25일
심사청구일자 2020년04월28일
- (85) 번역문제출일자 2020년04월28일
- (86) 국제출원번호 PCT/CN2018/107275
- (87) 국제공개번호 WO 2019/062699
국제공개일자 2019년04월04일
- (30) 우선권주장
201710911639.9 2017년09월29일 중국(CN)

- (71) 출원인
텐센트 테크놀로지(셴젠) 컴퍼니 리미티드
중국 518057 광둥 셴젠 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이-테크 파크 커지중이 로드 텐센트 빌딩 35층
- (72) 발명자
왕, 민
중국 518057 광둥 셴젠 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이테크 파크 커지중이 로드 텐센트 빌딩 35층
천, 위둥
중국 518057 광둥 셴젠 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이테크 파크 커지중이 로드 텐센트 빌딩 35층
(뒷면에 계속)
- (74) 대리인
양영준, 백만기

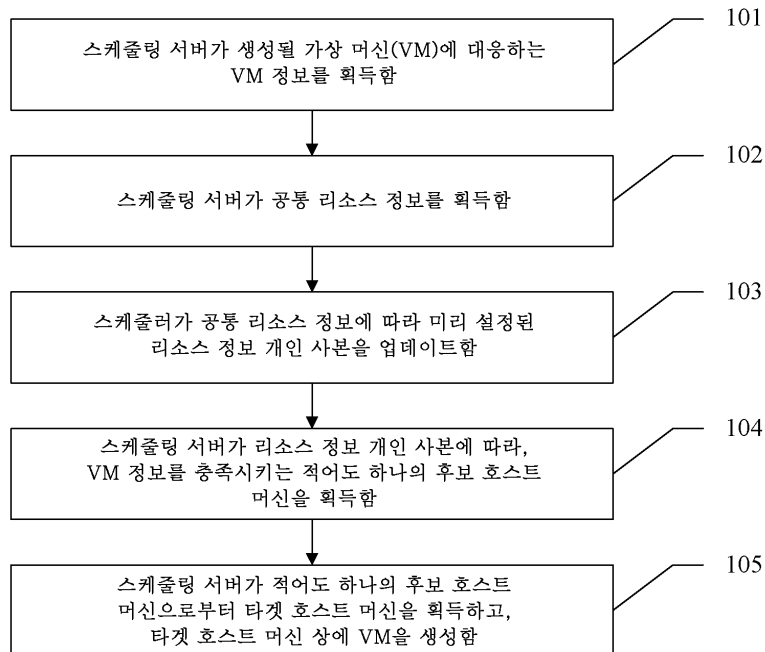
전체 청구항 수 : 총 19 항

(54) 발명의 명칭 리소스 스케줄링 방법, 스케줄링 서버, 클라우드 컴퓨팅 시스템, 및 저장 매체

(57) 요약

리소스 스케줄링 방법, 스케줄링 서버, 클라우드 컴퓨팅 시스템 및 저장 매체가 개시된다. 본 출원의 실시예들에서, 이 방법은 생성될 가상 머신에 관한 가상 머신 정보를 획득하는 단계 및 공중 리소스 정보를 획득하는 단계; 이어서, 공중 리소스 정보에 따라 리소스 정보의 미리 설정된 개인 사본을 업데이트하는 단계 - 이 리소스 (뒷면에 계속)

대표도 - 도2



정보의 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -; 및 마지막으로, 리소스 정보의 개인 사본에 따라, 가상 머신 정보를 만족시키는 적어도 하나의 후보 호스트 머신을 획득하고, 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 가상 머신을 생성하는 단계를 포함한다. 이 해결책에서, 리소스 스케줄링이 수행되기 전에, 리소스 정보의 개인 사본은 리소스 정보의 개인 사본과 공중 리소스 정보 사이의 동기화를 보장하기 위해 적시에 업데이트될 수 있어서, 리소스 스케줄링 결과가 비교적 양호하고, 리소스 스케줄링 동안 대기할 필요가 없어서 리소스 스케줄링 효율을 향상시킨다.

(52) CPC특허분류

G06F 2009/4557 (2019.08)

(72) 발명자

리, 더카이

중국 518057 광둥 선전 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이테크 파크 커지중이 로드 텐센트 빌딩 35층

리, 리

중국 518057 광둥 선전 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이테크 파크 커지중이 로드 텐센트 빌딩 35층

장, 진둥

중국 518057 광둥 선전 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이테크 파크 커지중이 로드 텐센트 빌딩 35층

팡, 보

중국 518057 광둥 선전 난산 디스트릭트 미드웨스트 디스트릭트 오브 하이테크 파크 커지중이 로드 텐센트 빌딩 35층

명세서

청구범위

청구항 1

리소스 스케줄링 방법으로서,

스케줄링 서버에 의해, 생성될 가상 머신(VM)에 대응하는 VM 정보를 획득하는 단계;

상기 스케줄링 서버에 의해, 공통 리소스 정보를 획득하는 단계 - 상기 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -;

상기 스케줄링 서버에 의해, 상기 공통 리소스 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계 - 상기 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -;

상기 리소스 정보 개인 사본에 따라 상기 스케줄링 서버에 의해, 상기 VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 단계; 및

상기 스케줄링 서버에 의해, 상기 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 상기 타겟 호스트 머신 상에 상기 VM을 생성하는 단계

를 포함하는, 리소스 스케줄링 방법.

청구항 2

제1항에 있어서,

상기 스케줄링 서버에 의해, 상기 공통 리소스 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계는,

상기 스케줄링 서버에 의해, 상기 리소스 정보 개인 사본을 상기 공통 리소스 정보와 동기화하는 단계를 포함하는, 리소스 스케줄링 방법.

청구항 3

제1항에 있어서,

상기 스케줄링 서버에 의해, 상기 공통 리소스 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계는,

상기 스케줄링 서버에 의해, 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 상기 타겟 호스트 머신 정보를 상기 미리 설정된 리소스 정보 개인 사본에 추가하는 단계를 포함하는, 리소스 스케줄링 방법.

청구항 4

제3항에 있어서,

상기 스케줄링 서버에 의해, 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 상기 타겟 호스트 머신 정보를 상기 미리 설정된 리소스 정보 개인 사본에 추가하는 단계는,

상기 스케줄링 서버에 의해, 상기 리소스 정보 개인 사본이 처음으로 업데이트되는 경우에 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 상기 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하는 단계;

상기 스케줄링 서버에 의해, 상기 타겟 호스트 머신 정보를 상기 리소스 정보 개인 사본에 복사하는 단계;

상기 스케줄링 서버에 의해, 현재의 타임스탬프를 획득하고, 상기 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로 설정하는 단계; 및

상기 스케줄링 서버에 의해, 상기 전체 업데이트 타임스탬프 및 상기 증분 업데이트 타임스탬프를 상기 리소스 정보 개인 사본에 저장하는 단계

를 포함하는, 리소스 스케줄링 방법.

청구항 5

제3항에 있어서,

상기 스케줄링 서버에 의해, 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 상기 타겟 호스트 머신 정보를 상기 미리 설정된 리소스 정보 개인 사본에 추가하는 단계는,

상기 스케줄링 서버에 의해, 상기 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 경우에 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득하는 단계;

상기 스케줄링 서버에 의해, 상기 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정하는 단계;

상기 스케줄링 서버에 의해, 상기 시간 차이가 상기 전체 업데이트 임계값보다 큰 경우에 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하는 단계;

상기 스케줄링 서버에 의해, 상기 제1 타겟 호스트 머신 정보를 상기 리소스 정보 개인 사본에 복사하는 단계;

상기 스케줄링 서버에 의해, 상기 현재의 타임스탬프를 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정하는 단계; 및

상기 스케줄링 서버에 의해, 상기 제1 전체 업데이트 타임스탬프 및 상기 제1 증분 업데이트 타임스탬프를 상기 리소스 정보 개인 사본에 저장하는 단계

를 포함하는, 리소스 스케줄링 방법.

청구항 6

제5항에 있어서,

상기 스케줄링 서버에 의해, 상기 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정하는 단계 후에, 상기 방법은,

상기 스케줄링 서버에 의해, 상기 시간 차이가 상기 전체 업데이트 임계값 이하인 경우에 상기 공통 리소스 정보로부터 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 업데이트된 호스트 머신 정보를 획득하는 단계;

상기 스케줄링 서버에 의해, 상기 호스트 머신 정보로부터 상기 VM 정보를 충족시키는 제2 타겟 호스트 머신 정보를 획득하는 단계;

상기 스케줄링 서버에 의해, 상기 제2 타겟 호스트 머신 정보를 상기 리소스 정보 개인 사본에 추가하는 단계;

상기 스케줄링 서버에 의해, 상기 현재의 타임스탬프를 제2 증분 업데이트 타임스탬프로 설정하는 단계; 및

상기 스케줄링 서버에 의해, 상기 제2 증분 업데이트 타임스탬프를 상기 리소스 정보 개인 사본에 저장하는 단계

를 더 포함하는, 리소스 스케줄링 방법.

청구항 7

제1항에 있어서,

상기 스케줄링 서버에 의해, 상기 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하는 단계는,

상기 스케줄링 서버에 의해, 각각의 차원들에서 상기 적어도 하나의 후보 호스트 머신의 각각의 호스트 머신의 우선순위 값들을 결정하는 단계;

상기 스케줄링 서버에 의해, 상기 차원들에서 상기 우선순위 값들의 내림차순으로 순위화를 수행하여 후보 호스트 머신 리스트를 생성하는 단계;

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 상기 후보 호스트 머신들로부터 상기 생성될 VM의 리소스 요건을 하나씩 차감하여, 상기 후보 호스트 머신들의 나머지 리소스들을 획득하고, 상기 후보 호스트 머신들 중 하나의 후보 호스트 머신의 나머지 리소스가 성공적으로 제출될 때까지 상기 나머지 리소스들을 상기 공통 리소스 정보에 제출하는 단계 - 상기 리소스 요건은 상기 VM에 대응하는 상기 VM 정보에 포함됨 -; 및

상기 후보 호스트 머신들 중 하나의 후보 호스트 머신에 대응하는 나머지 리소스가 성공적으로 제출되는 경우에 상기 스케줄링 서버에 의해, 그 제출이 성공한 상기 후보 호스트 머신을 상기 타겟 호스트 머신으로 설정하는 단계

를 포함하는, 리소스 스케줄링 방법.

청구항 8

제7항에 있어서,

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 상기 후보 호스트 머신들로부터 상기 생성될 VM의 리소스 요건을 하나씩 차감하여, 상기 후보 호스트 머신들의 나머지 리소스들을 획득하는 단계 전에, 상기 방법은 상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하여 후보 호스트 머신 타겟 리스트를 생성하는 단계를 더 포함하며,

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 상기 후보 호스트 머신들로부터 상기 생성될 VM의 리소스 요건을 하나씩 차감하여, 상기 후보 호스트 머신들의 나머지 리소스들을 획득하는 단계는, 상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 타겟 리스트에서의 상기 후보 호스트 머신들의 순위들에 따라 상기 후보 호스트 머신 타겟 리스트에서의 상기 후보 호스트 머신들로부터 상기 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하는 단계를 포함하는, 리소스 스케줄링 방법.

청구항 9

제8항에 있어서,

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하여 후보 호스트 머신 타겟 리스트를 생성하는 단계는,

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 상기 최상위 후보 호스트 머신들의 미리 설정된 양을 랜덤하게 재순위화하여 상기 후보 호스트 머신 타겟 리스트를 획득하는 단계를 포함하는, 리소스 스케줄링 방법.

청구항 10

제1항에 있어서,

상기 리소스 정보 개인 사본에 따라 상기 스케줄링 서버에 의해, 상기 VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 단계는,

상기 스케줄링 서버에 의해, 상기 VM 정보 내의 리소스 요건에 따라 상기 리소스 정보 개인 사본 내의 상기 호스트 머신을 필터링하여 상기 적어도 하나의 후보 호스트 머신을 획득하는 단계를 포함하는, 리소스 스케줄링 방법.

청구항 11

제7항에 있어서,

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신들 중 하나의 후보 호스트 머신의 나머지 리소스가 성공적으로 제출될 때까지 상기 나머지 리소스들을 상기 공통 리소스 정보에 제출하는 단계는,

하나의 후보 호스트 머신의 나머지 리소스를 제출하지 못하는 경우에 상기 스케줄링 서버에 의해, 상기 후보 호스트 머신의 나머지 리소스의 제출 횟수들의 양이 스케줄링 충돌 임계값을 초과하는지를 결정하는 단계;

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신의 최신 리소스 데이터를 획득하고, 상기 후보 호스트 머신의 나머지 리소스의 제출 횟수들의 양이 상기 스케줄링 충돌 임계값을 초과하지 않는 경우에 상기 후보 호스트 머신에 대응하는 나머지 리소스를 재결정하는 단계; 및

상기 스케줄링 서버에 의해, 상기 후보 호스트 머신 리스트에서의 다음 후보 호스트 머신으로부터 상기 생성된 VM의 상기 리소스 요건을 차감하여 대응하는 나머지 리소스를 획득하고, 상기 후보 호스트 머신의 나머지 리소스의 제출 횟수들의 양이 상기 스케줄링 충돌 임계값을 초과하는 경우에 상기 대응하는 나머지 리소스를 제출하는 단계

를 포함하는, 리소스 스케줄링 방법.

청구항 12

스케줄링 서버로서,

프로세서;

메모리; 및

상기 메모리에 저장되고, 상기 프로세서에 의해 실행될 때, 상기 프로세서로 하여금 다음의 동작들을 수행하게 하는 하나 이상의 프로그램

을 포함하며,

상기 동작들은,

생성된 VM에 대응하는 VM 정보를 획득하는 동작;

공통 리소스 정보를 획득하는 동작 - 상기 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -;

상기 공통 리소스 정보 및 상기 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 동작 - 상기 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -;

상기 리소스 정보 개인 사본에 따라, 상기 VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 동작; 및

상기 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 상기 타겟 호스트 머신 상에 상기 VM을 생성하는 동작인, 스케줄링 서버.

청구항 13

제12항에 있어서,

상기 프로세서는 추가로,

상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 상기 타겟 호스트 머신 정보를 상기 미리 설정된 리소스 정보 개인 사본에 추가하는 동작을 수행하는, 스케줄링 서버.

청구항 14

제13항에 있어서,

상기 프로세서는 추가로,

상기 리소스 정보 개인 사본이 처음으로 업데이트되는 경우에 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하는 동작;

상기 타겟 호스트 머신 정보를 상기 리소스 정보 개인 사본에 복사하는 동작;

현재의 타임스탬프를 획득하고, 상기 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로 설정하는 동작; 및

상기 전체 업데이트 타임스탬프 및 상기 증분 업데이트 타임스탬프를 상기 리소스 정보 개인 사본에 저장하는 동작

을 수행하는, 스케줄링 서버.

청구항 15

제13항에 있어서,

상기 프로세서는 추가로,

상기 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 경우에 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득하는 동작;

상기 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정하는 동작;

상기 시간 차이가 상기 전체 업데이트 임계값보다 큰 경우에 상기 공통 리소스 정보로부터 상기 VM 정보를 충족시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하는 동작;

상기 제1 타겟 호스트 머신 정보를 상기 리소스 정보 개인 사본에 복사하는 동작;

상기 현재의 타임스탬프를 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정하는 동작; 및

상기 제1 전체 업데이트 타임스탬프 및 상기 제1 증분 업데이트 타임스탬프를 상기 리소스 정보 개인 사본에 저장하는 동작

을 수행하는, 스케줄링 서버.

청구항 16

제15항에 있어서,

상기 프로세서는 추가로,

상기 시간 차이가 상기 전체 업데이트 임계값 이하인 경우에 상기 공통 리소스 정보로부터 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 업데이트된 호스트 머신 정보를 획득하는 동작;

상기 호스트 머신 정보로부터 상기 VM 정보를 충족시키는 제2 타겟 호스트 머신 정보를 획득하는 동작;

상기 제2 타겟 호스트 머신 정보를 상기 리소스 정보 개인 사본에 추가하는 동작;

상기 현재의 타임스탬프를 제2 증분 업데이트 타임스탬프로 설정하는 동작; 및

상기 제2 증분 업데이트 타임스탬프를 상기 리소스 정보 개인 사본에 저장하는 동작

을 수행하는, 스케줄링 서버.

청구항 17

제12항에 있어서,

상기 프로세서는 추가로,

각각의 차원들에서 상기 적어도 하나의 후보 호스트 머신의 각각의 호스트 머신의 우선순위 값들을 결정하는 동작;

상기 차원들에서 상기 우선순위 값들의 내림차순으로 순위화를 수행하여 후보 호스트 머신 리스트를 생성하는 동작;

상기 후보 호스트 머신 리스트에서의 상기 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터 상기 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하는 동작 - 상기 리소스 요건은 상기 VM에

대응하는 상기 VM 정보에 포함됨 -;

상기 나머지 리소스들을 상기 공통 리소스 정보에 제출하는 동작; 및

하나의 후보 호스트 머신에 대응하는 나머지 리소스가 성공적으로 제출되는 경우에, 그 제출이 성공한 상기 후보 호스트 머신을 상기 타겟 호스트 머신으로 설정하고, 상기 타겟 호스트 머신 상에 상기 VM을 생성하는 동작을 수행하는, 스케줄링 서버.

청구항 18

제12항 내지 제17항 중 어느 한 항에 따른 적어도 하나의 스케줄링 서버 및 적어도 하나의 호스트 머신을 포함하는 클라우드 컴퓨팅 시스템으로서,

상기 호스트 머신은 적어도 하나의 VM을 생성하도록 구성되는, 클라우드 컴퓨팅 시스템.

청구항 19

복수의 명령어들을 저장하는 저장 매체로서,

상기 명령어들은 프로세서에 의해 로딩되어 제1항 내지 제11항 중 어느 한 항에 따른 리소스 스케줄링 방법을 수행하도록 구성되는 저장 매체.

발명의 설명

기술 분야

관련 출원

[0001]

[0002] 본 출원은 "리소스 스케줄링 방법 및 장치, 클라우드 컴퓨팅 시스템, 및 저장 매체(RESOURCE SCHEDULING METHOD AND APPARATUS, CLOUD COMPUTING SYSTEM, AND STORAGE MEDIUM)"라는 명칭으로 2017년 9월 29일자로 중국 특허청에 출원된 중국 특허 출원 제2017093000870880호에 대한 우선권을 주장하며, 이는 그 전체가 참조로 포함된다.

[0003]

[0003] 본 출원은 클라우드 기술들의 분야에 관한 것으로, 구체적으로는, 리소스 스케줄링 방법, 스케줄링 서버, 클라우드 컴퓨팅 시스템, 및 저장 매체에 관한 것이다.

배경 기술

[0004]

[0004] 클라우드 컴퓨팅 시스템은 일반적으로 스케줄러들, 호스트 머신들 등을 포함한다. 스케줄러는 가상 머신(VM) 소프트웨어를 이용하여 호스트 머신 상에 하나 이상의 VM을 생성할 수 있다. 클라우드 컴퓨팅 시스템 내에는 많은 양의 호스트 머신들이 존재한다. VM이 생성될 때, VM이 생성되는 호스트 머신을 선택하는 프로세스는 리소스 스케줄링이라고 불린다. 리소스 스케줄링을 수행하는 구성요소는 스케줄러라고 불린다.

[0005]

[0005] 일부 실시예들에서, 스케줄러가 리소스 스케줄링을 수행하고 있고 호스트 머신들의 스케일 및 리소스 스케줄링 요청들이 어느 정도 증가하는 경우, 스케줄러가 단일 리소스 스케줄링 요청을 처리하는 시간은 명백하게 증가한다. 또한, 후속 요청들은 처리되도록 항상 줄지어 대기한다. 예를 들어, 2-레벨 스케줄링 아키텍처가 이용되며, 여기서 중앙 리소스 관리자는 먼저 리소스 초대를 이용하여 스케줄러에 호스트 머신 리소스를 제공하고, 스케줄러는 리소스 초대의 범위 내에서 호스트 머신을 선택하여, 리소스 스케줄링 프로세스를 완료한다. 또한, 스케줄러가 리소스 스케줄링을 완료한 후에만, 중앙 리소스 관리자는 리소스 초대를 이용하여 호스트 머신 리소스를 다른 스케줄러에 제공한다. 예를 들어, 클라우드 컴퓨팅 시스템의 데이터 센터 전체는 호스트 머신 A, 호스트 머신 B, 호스트 머신 C, 호스트 머신 D, 호스트 머신 E 등을 포함한다. 중앙 리소스 관리자는 리소스 초대를 이용하여 호스트 머신 A 및 호스트 머신 B를 스케줄러 1에 제공한다. 이 경우에, 스케줄러 1은 호스트 머신 A 및 호스트 머신 B로부터만 하나의 호스트 머신을 선택할 수 있다. 스케줄러 1이 리소스 스케줄링을 완료한 후에만, 중앙 리소스 관리자는 리소스 초대를 이용하여 호스트 머신 리소스를 스케줄러 2에 제공한다.

[0006]

[0006] 진술한 실시예들은 다음과 같은 문제점들을 갖는다: 한편으로는, 스케줄러는 단지 리소스 초대의 범위 내에서 호스트 머신을 선택할 수 있고, 이는 선택가능한 호스트 머신들의 스케일을 감소시키고, 그 결과 더 나은 리소스 스케줄링 결과가 보장될 수 없다. 다른 한편으로는, 중앙 리소스 관리자는 먼저 스케줄러에 호스트 머신 리소스를 제공하고, 스케줄러가 리소스 스케줄링을 완료한 후에, 중앙 리소스 관리자는 호스트 머신 리소스를 다

른 스케줄러에 제공한다. 그 결과, 스케줄러는 리소스 스케줄링을 수행할 때 대기할 필요가 있어서, 리소스 스케줄링의 높은 레이턴시 및 낮은 처리율을 초래한다.

발명의 내용

- [0007] 본 출원의 실시예들은, 더 나은 리소스 스케줄링 결과를 달성하고 리소스 스케줄링의 효율을 향상시킬 수 있는, 리소스 스케줄링 방법, 스케줄링 서버, 클라우드 컴퓨팅 시스템, 및 저장 매체를 제공한다.
- [0008] 전술한 기술적 문제점을 해결하기 위해, 본 출원의 실시예들은 다음과 같은 기술적 해결책들을 제공한다:
- [0009] 리소스 스케줄링 방법은,
- [0010] 스케줄링 서버에 의해, 생성될 가상 머신(VM)에 대응하는 VM 정보를 획득하는 단계;
- [0011] 스케줄링 서버에 의해, 공통 리소스 정보를 획득하는 단계 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -;
- [0012] 스케줄링 서버에 의해, 공통 리소스 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -;
- [0013] 리소스 정보 개인 사본에 따라 스케줄링 서버에 의해, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 단계; 및
- [0014] 스케줄링 서버에 의해, 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하는 단계
- [0015] 를 포함한다.
- [0016] 스케줄링 서버는, 프로세서; 메모리; 및 메모리에 저장되고, 프로세서에 의해 실행될 때, 프로세서로 하여금 다음의 동작들을 수행하게 하는 하나 이상의 프로그램을 포함하며, 이러한 동작들은,
- [0017] 생성될 VM에 대응하는 VM 정보를 획득하는 동작;
- [0018] 공통 리소스 정보를 획득하는 동작 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -;
- [0019] 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 동작 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -;
- [0020] 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 동작; 및
- [0021] 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하는 동작이다.
- [0022] 클라우드 컴퓨팅 시스템은 본 출원에 따른 적어도 하나의 스케줄링 서버 및 적어도 하나의 호스트 머신을 포함하고, 호스트 머신은 적어도 하나의 VM을 생성하도록 구성된다.
- [0023] 저장 매체는 복수의 명령어들을 저장하고, 이러한 명령어들은 프로세서에 의해 로딩되어 전술한 리소스 스케줄링 방법을 수행하도록 구성된다.
- [0024] 본 출원의 실시예들에서, 생성될 VM에 대응하는 VM 정보가 획득되고, 공통 리소스 정보가 획득된다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함한다. 다음으로, 미리 설정된 리소스 정보 개인 사본은 공통 리소스 정보 및 VM 정보에 따라 업데이트된다. 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 마지막으로, 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신이 획득되고, 타겟 호스트 머신이 적어도 하나의 후보 호스트 머신으로부터 획득되고, VM이 타겟 호스트 머신 상에 생성된다. 이러한 해결책에서, 리소스 정보 개인 사본은 리소스 스케줄링이 수행되기 전에 제시간에 업데이트될 수 있고, 이는 리소스 정보 개인 사본과 공통 리소스 정보의 동기화를 보장하여, 더 나은 리소스 스케줄링 결과가 달성되고, 리소스 스케줄링 동안 대기할 필요가 없고, 리소스 스케줄링의 효율을 향상시킨다.

도면의 간단한 설명

[0025] 본 출원의 실시예들의 기술적 해결책들을 더 명확하게 설명하기 위해, 다음에서는 이러한 실시예들을 설명하는 데 요구되는 첨부 도면들을 간단히 소개한다. 명백하게, 다음의 설명에서의 첨부 도면들은 본 출원의 일부 실시예들만을 도시하지만, 관련 기술분야의 통상의 기술자는 창의적 노력 없이도 이들 첨부 도면들로부터 다른 도면들을 여전히 도출할 수 있다.

- 도 1은 본 출원의 실시예에 따른 리소스 스케줄링 시스템의 시나리오의 개략도이다.
- 도 2는 본 출원의 실시예에 따른 리소스 스케줄링 방법의 개략적인 흐름도이다.
- 도 3은 본 출원의 실시예에 따른 리소스 스케줄링 방법의 개략적인 흐름도이다.
- 도 4는 본 출원의 실시예에 따른 리소스 스케줄링 방법의 개략적인 흐름도이다.
- 도 5는 본 출원의 실시예에 따른 리소스 스케줄링 방법의 개략적인 흐름도이다.
- 도 6은 본 출원의 실시예에 따른 리소스 스케줄링 방법에서 리소스 정보 개인 사본을 업데이트하는 개략적인 흐름도이다.
- 도 7은 본 출원의 실시예에 따른 리소스 스케줄링 방법에서 타겟 호스트 머신을 획득하는 개략적인 흐름도이다.
- 도 8은 본 출원의 실시예에 따른 리소스 스케줄링 장치의 개략적인 구조도이다.
- 도 9는 본 출원의 실시예에 따른 리소스 스케줄링 장치의 개략적인 구조도이다.
- 도 10은 본 출원의 실시예에 따른 리소스 스케줄링 장치의 개략적인 구조도이다.
- 도 11은 본 출원의 실시예에 따른 리소스 스케줄링 장치의 개략적인 구조도이다.
- 도 12는 본 출원의 실시예에 따른 스케줄링 서버의 개략적인 구조도이다.

발명을 실시하기 위한 구체적인 내용

[0026] 다음은 본 출원의 실시예들에서의 첨부 도면들을 참조하여 본 출원의 실시예들에서의 기술적 해결책들을 명확하고 완전히 설명한다. 명백히, 설명된 실시예들은 실시예들의 전부라기보다는 본 출원의 실시예들 중 일부일 뿐이다. 창의적 노력 없이 본 출원의 실시예들에 기반하여 관련 기술분야의 통상의 기술자에 의해 획득되는 모든 다른 실시예들은 본 출원의 보호 범위 내에 들 것이다.

[0027] 이하의 설명에서, 본 출원의 구체적인 실시예들은 달리 명시되지 않는 한, 하나 이상의 컴퓨터에 의해 수행되는 동작들의 단계들 및 심볼 표현들을 참조하여 설명될 것이다. 이와 같이, 컴퓨터에 의해 실행되는 것으로서 때때로 참조되는 이러한 단계들 및 동작들은 구조적 형태로 데이터를 나타내는 전기 신호들의 컴퓨터의 처리 유닛에 의한 조작을 포함한다는 것이 이해될 것이다. 이러한 조작은 데이터를 변환하거나, 관련 기술분야의 통상의 기술자가 잘 이해하는 방식으로 컴퓨터의 동작을 재구성하거나 다른 방식으로 변경하는, 컴퓨터의 메모리 시스템 내의 위치들에 데이터를 유지시킨다. 데이터가 유지되는 데이터 구조들은 데이터의 포맷에 의해 정의되는 특정한 특성들을 갖는 메모리의 물리적 위치들이다. 그러나, 본 출원의 원리가 전술한 텍스트에서 설명되어 있지만, 이것은 관련 기술분야의 통상의 기술자가 이해하는 바와 같이 이하에서 설명되는 다양한 단계들 및 동작들이 하드웨어로 구현될 수 있는 것으로 제한하려는 것은 아니다.

[0028] 본 출원의 실시예들은 리소스 스케줄링 방법, 스케줄링 서버, 클라우드 컴퓨팅 시스템 및 저장 매체를 제공한다.

[0029] 도 1은 본 출원의 실시예에 따른 리소스 스케줄링 시스템의 시나리오의 개략도이다. 리소스 스케줄링 시스템은 클라우드 컴퓨팅 시스템일 수 있다. 클라우드 컴퓨팅 시스템은 리소스 스케줄링 장치를 포함할 수 있다. 리소스 스케줄링 장치는 구체적으로 스케줄링 서버에 통합될 수 있다. 스케줄링 서버는 클라우드 컴퓨팅 시스템 내의 스케줄러일 수 있다. 다시 말해, 각각의 스케줄러는 리소스 스케줄링 장치를 포함할 수 있다. 클라우드 컴퓨팅 시스템에는 하나 이상의 스케줄러, 예를 들어, 도 1에 도시된 스케줄러 1, 스케줄러 2, ..., 및 스케줄러 n이 있을 수 있다. 스케줄러는 생성될 VM에 대응하는 VM 정보를 획득하고 공통 리소스 정보를 획득하도록 구성된다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템 내의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함할 수 있다. 그 후, 스케줄러는 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트할 수 있다. 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 스케줄러에 의해 유지되는 리소스 정보 개인 사본은 생성될 VM에 대해 적절한 호스트 머신을 선택하는

데 이용될 수 있다. 스케줄러는 또한, 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 것, 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하는 것, 타겟 호스트 머신 상에 VM을 생성하는 것 등을 행할 수 있다.

- [0030] 또한, 클라우드 컴퓨팅 시스템은 하나 이상의 호스트 머신, 예를 들어, 도 1에 도시된 호스트 머신 A, 호스트 머신 B, ..., 및 호스트 머신 n을 더 포함할 수 있다. 호스트 머신은 스케줄러에 의해 전송된 VM을 생성하라는 요청을 수신하고, 그 요청에 따라 하나 이상의 VM을 생성하도록 구성될 수 있다.
- [0031] 클라우드 컴퓨팅 시스템은 수신기를 더 포함할 수 있다. 수신기는 주로 하나 이상의 단말기에 의해 전송된 VM 생성 요청을 수신하도록 구성된다. VM 생성 요청은 VM 정보를 운반할 수 있다. 그 후, 수신기는 각각의 수신된 VM 생성 요청을 메시지 내로 캡슐화하고, 메시지를 미리 설정된 메시지 큐에 저장할 수 있고, 따라서 스케줄러는 메시지 큐로부터 메시지를 획득하고 그 메시지로부터 생성될 VM의 VM 정보를 추출할 수 있다. 복수의 메시지가 메시지 큐에 존재할 때, 복수의 스케줄러가 메시지들을 동시에 획득하고 상이한 생성될 VM들에 대한 호스트 머신들을 선택할 수 있다.
- [0032] 클라우드 컴퓨팅 시스템은 리소스 관리자를 더 포함할 수 있다. 리소스 관리자는 주로 공통 리소스 정보를 유지하도록 구성되어, 스케줄러가 리소스 관리자로부터 공통 리소스 정보를 획득할 수 있게 한다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템 내의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 저장 및 유지하는데 이용된다.
- [0033] 도 1에 도시된 리소스 스케줄링 시스템의 시나리오의 개략도는 단지 예이다. 본 출원의 실시예들에서 설명되는 리소스 스케줄링 시스템 및 시나리오는 본 출원의 실시예들에서의 기술적 해결책들을 더 명확하게 설명하기 위한 것이고, 본 출원의 실시예들에서 제공되는 기술적 해결책들에 대한 제한을 구성하지 않는다. 관련 기술분야의 통상의 기술자는, 클라우드 컴퓨팅 시스템의 진화 및 새로운 서비스 시나리오의 출현에 따라, 본 출원의 실시예들에서 제공되는 기술적 해결책들이 유사한 기술적 문제에도 적용된다는 것을 알 수 있다.
- [0034] 상세한 설명들이 아래에 각각 행해진다.
- [0035] 이 실시예에서, 리소스 스케줄링 장치의 관점에서 설명들이 제공된다. 리소스 스케줄링 장치는 구체적으로 스케줄링 서버 또는 게이트웨이와 같은 네트워크 디바이스에 통합될 수 있다.
- [0036] 스케줄링 서버에 의해 리소스 스케줄링 방법이 수행될 수 있다. 리소스 스케줄링 방법은, 생성될 VM에 대응하는 VM 정보를 획득하는 단계; 공통 리소스 정보를 획득하는 단계 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -; 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -; 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 단계; 및 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하는 단계를 포함할 수 있다.
- [0037] 도 2는 본 출원의 제1 실시예에 따른 리소스 스케줄링 방법의 개략적인 흐름도이다. 도 2에 도시된 리소스 스케줄링 방법은 예를 들어 스케줄링 서버에 의해 수행될 수 있다. 리소스 스케줄링 방법은 다음의 단계들을 포함한다:
- [0038] 단계 101에서, 스케줄링 서버는 생성될 VM에 대응하는 VM 정보를 획득한다.
- [0039] 예를 들어, 스케줄링 서버는 클라우드 컴퓨팅 시스템에서의 미리 설정된 메시지 큐로부터 생성될 VM에 대응하는 VM 정보를 획득할 수 있다. VM은 소프트웨어를 통해 시뮬레이션되고 하드웨어 시스템 기능을 가지며 격리된 환경에서 실행되는 전체 컴퓨터 시스템이다.
- [0040] 메시지 큐 내의 메시지는 비동기 작업일 수 있다. 메시지 큐 내의 메시지는 메시지 큐에서 흐른다. 클라우드 컴퓨팅 시스템 내의 각각의 구성요소는 생성자 또는 소비자의 아이덴티티를 갖는다. 생성자는 메시지를 메시지 큐에 전송한다. 소비자는 메시지 큐로부터 메시지를 수신한다. 예를 들어, 수신기는 클라우드 컴퓨팅 시스템에서 생성자로서 이용된다. 단말기에 의해 전송된 VM 정보를 운반하는 VM 생성 요청을 수신한 후, 수신기는 VM 정보를 메시지 내에 캡슐화하고, 메시지를 메시지 큐에 저장한다. VM 정보는 스케줄링 서버가 리소스 스케줄링을 수행하기를 대기하기 위해 메시지 큐에 큐잉된다. 소비자로서, 스케줄링 서버는 메시지 큐로부터 VM 정보를 추출하고, VM 정보에 대응하는 생성될 VM에 대한 호스트 머신을 선택한다.
- [0041] VM 정보는 생성될 VM에 의해 요구되는 프로세서 정보, 내부 메모리 정보, 및 디스크 정보를 포함할 수 있다.

VM 정보는 솔리드 스테이트 드라이브, 클라우드 디스크, 개인 네트워크, 네트워크 어댑터, 라우터, 네트워크 구조, 데이터 센터, 랙, 호스트 머신 유형, 호스트 머신이 속하는 리소스 풀, 및 VM 유형에 대한 생성될 VM의 리소스 요건들 중 적어도 하나를 포함할 수 있다. VM 정보의 유형이 실제 요건들에 따라 유연하게 설정될 수 있고, 특정 콘텐츠로 본 명세서에서 제한되지 않는다는 점이 이해될 수 있다.

- [0042] 단계 102에서, 스케줄링 서버는 공통 리소스 정보를 획득한다.
- [0043] 공통 리소스 정보는 클라우드 컴퓨팅 시스템에 의해 저장되고 유지되는 전체 데이터 센터의 리소스 데이터이다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함한다. 공통 리소스 정보는 리소스 관리자에 의해 유지될 수 있거나, 또는 클라우드 컴퓨팅 시스템 내의 다른 디바이스에 의해 유지될 수 있다는 점이 이해될 수 있다. 스케줄링 서버는 공통 리소스 정보를 유지하는 디바이스로부터 공통 리소스 정보를 획득할 수 있다. 예를 들어, 공통 리소스 정보가 클라우드 컴퓨팅 시스템에서 리소스 관리자에 의해 유지되는 경우, 스케줄링 서버는 리소스 관리자로부터 공통 리소스 정보를 획득할 수 있다. 설명의 용이함을 위해, 본 출원의 실시예들에서, 공통 리소스 정보를 유지하는 리소스 관리자가 설명을 위한 예로서 이용된다.
- [0044] 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서 전역적으로 고유하고, 논리 데이터 테이블의 형태로 유지될 수 있다. 예를 들어, 데이터 테이블 내의 각각의 행은 호스트 머신을 나타내고, 데이터 테이블 내의 각각의 열은 호스트 머신에 포함된 리소스의 유형을 나타낸다. 리소스들은 호스트 머신의 인터넷 프로토콜(IP) 어드레스, 이용가능한 중앙 처리 유닛(CPU), 이용가능한 내부 메모리, 및 이용가능한 디스크를 포함할 수 있다. 유지 프로세스에서, 추가, 판독, 업데이트, 또는 삭제와 같은 동작이 공통 리소스 정보에 대해 수행될 수 있다. 이러한 동작은 데이터 센터에서의 호스트 머신 리소스의 온라인 획득, 변경, 또는 오프라인 획득과 같은 프로세스에 대응한다.
- [0045] 호스트 머신은 VM 소프트웨어를 설치하도록 구성된 물리적 컴퓨터이다. VM 소프트웨어를 통해 하나 이상의 VM 이 하나의 호스트 머신 상에서 생성될 수 있다.
- [0046] 일부 실시예들에서, 호스트 머신 정보는 호스트 머신 식별자, IP 어드레스, 머신 유형, 운영 체제 이름, 커널 버전, CPU 유형, 가상화 방식, 논리적 영역 번호, 인터넷 데이터 센터 번호, 호스트 머신의 상태, 클라우드 디스크를 지원할지 여부, 솔리드 스테이트 드라이브(SSD)를 지원할지 여부, 네트워크 버전, CPU 노드 리소스 할당량, 이용가능한 내부 메모리 리소스 및 이용가능한 디스크 리소스를 포함할 수 있고, 특정 콘텐츠로 본 명세서에서 제한되지는 않는다.
- [0047] 호스트 머신이 클라우드 컴퓨팅 시스템에 진입할 때, 필요한 초기화가 호스트 머신 도입 프로세스를 통해 수행될 필요가 있다. 호스트 머신은 그 호스트 머신 정보를 리소스 관리자에 보고하고, 리소스 관리자는 호스트 머신 정보를 공통 리소스 정보에 저장할 수 있다. 일부 실시예들에서, 호스트 머신은 대안적으로 그 호스트 머신 정보를 스케줄링 서버에 보고할 수 있다. 스케줄링 서버는 호스트 머신 정보를 리소스 정보 개인 사본에 저장할 수 있다. 일부 실시예들에서, 스케줄링 서버는 실시간으로 또는 미리 설정된 간격으로 리소스 관리자에게 정보 획득 요청을 전송하고, 정보 획득 요청에 기반하여 리소스 관리자에 의해 피드백된 공통 리소스 정보를 수신하고, 공통 리소스 정보에 따라 리소스 정보 개인 사본을 업데이트할 수 있다. 일부 실시예들에서, 공통 리소스 정보가 업데이트될 때, 리소스 관리자는 업데이트된 호스트 머신 정보를 스케줄링 서버에 능동적으로 전송한다.
- [0048] 단계 103에서, 스케줄링 서버는 공통 리소스 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트한다.
- [0049] 스케줄링 서버는 스케줄링 서버의 내부 메모리 내의 공통 리소스 정보인 리소스 정보 개인 사본을 버퍼 형태로 유지한다. 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 미리 설정된 호스트 머신은 클라우드 컴퓨팅 시스템 내의 호스트 머신들의 일부 또는 모든 호스트 머신들일 수 있다. 더 나은 리소스 스케줄링 결과를 달성하기 위해, 리소스 스케줄링을 수행하기 전에, 스케줄링 서버는 리소스 정보 개인 사본을 공통 리소스 정보와 동기화할 필요가 있고, 공통 리소스 정보를 획득하고 공통 리소스 정보에 따라 리소스 정보 개인 사본을 업데이트하여, 리소스 정보 개인 사본과 공통 리소스 정보 사이의 일관성을 보장할 수 있다.
- [0050] 업데이트 시간을 감소시키기 위해, 리소스 정보 개인 사본은 공통 리소스 정보 및 VM 정보에 따라 업데이트될 수 있다. 일부 실시예들에서, 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계는 공통 리소스 정보로부터 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리

닝을 수행하고, 타겟 호스트 머신 정보를 미리 설정된 리소스 정보 개인 사본에 추가하는 단계를 포함한다.

- [0051] 일부 실시예들에서, 공통 리소스 정보로부터 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 타겟 호스트 머신 정보를 미리 설정된 리소스 정보 개인 사본에 추가하는 단계는 다음의 단계들을 포함한다:
- [0052] 단계 S1. 스케줄링 서버는 리소스 정보 개인 사본이 처음으로 업데이트되는 경우에 공통 리소스 정보로부터 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다.
- [0053] 단계 S2. 스케줄링 서버는 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다.
- [0054] 단계 S3. 스케줄링 서버는 현재의 타임스탬프를 획득하고, 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로 설정한다.
- [0055] 단계 S4. 스케줄링 서버는 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장한다.
- [0056] 구체적으로, 스케줄링 서버는 먼저 리소스 정보 개인 사본이 처음으로 업데이트되는지를 결정하고, 예를 들어, 리소스 정보 개인 사본이 비어 있는지를 결정할 수 있다. 그렇다면, 리소스 정보 개인 사본이 처음으로 업데이트되는 것으로 결정되고; 그렇지 않다면, 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 것으로 결정된다. 리소스 정보 개인 사본이 처음으로 업데이트되는 경우, 현재 리소스 정보 개인 사본이 비어 있으면, 전체 업데이트가 수행될 필요가 있으며, 즉, 공통 리소스 정보로부터 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다. 공통 리소스 정보는 대안적으로 리소스 정보 개인 사본으로서 직접 전체 복사될 수 있다.
- [0057] 타겟 호스트 머신 정보가 리소스 정보 개인 사본에 복사된 후, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프가 기록될 필요가 있다. 일부 실시예들에서, 스케줄링 서버는 현재의 타임스탬프를 획득하고, 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로 설정한다. 예를 들어, 현재의 타임스탬프는 2017년 9월 15일 오전 11:00이고, 이 경우, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프 둘 다는 2017년 9월 15일 오전 11:00이다. 그 후, 현재 업데이트를 완료하기 위해, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프가 리소스 정보 개인 사본에 저장된다.
- [0058] 일부 실시예들에서, 공통 리소스 정보로부터 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 타겟 호스트 머신 정보를 미리 설정된 리소스 정보 개인 사본에 추가하는 단계는 다음의 단계들을 포함한다:
- [0059] 단계 S11. 스케줄링 서버는 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 경우에 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득한다.
- [0060] 단계 S12. 스케줄링 서버는 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정한다.
- [0061] 단계 S13. 스케줄링 서버는 시간 차이가 전체 업데이트 임계값보다 큰 경우에 공통 리소스 정보로부터 VM 정보를 충족시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다.
- [0062] 단계 S14. 스케줄링 서버는 제1 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다.
- [0063] 단계 S15. 스케줄링 서버는 현재의 타임스탬프를 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정한다.
- [0064] 단계 S16. 스케줄링 서버는 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장한다.
- [0065] 구체적으로는, 단계 S11에서, 스케줄링 서버는 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 경우에 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득하고, 여기서 이전의 전체 업데이트 타임스탬프는 현재의 타임스탬프에 대해 이전에 기록된 전체 업데이트 타임스탬프이다.
- [0066] 그 후, 단계 S12에서, 스케줄링 서버는 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정한다. 전체 업데이트 임계값은 실제 요건들에 따라 유연하게 설정될 수 있고, 특정 값으로 본 명세서에서 제한되지는 않는다. 예를 들어, 비교적 낮은 전체 업데이트 빈도를 달성하기 위해, 전체 업데이트 임계값은 비교적 큰 값으로 설정될 수 있다.

- [0067] 단계 S13에서, 스케줄링 서버는 시간 차이가 전체 업데이트 임계값보다 큰 경우에 공통 리소스 정보로부터 VM 정보를 충족시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다. 전체 업데이트를 수행하기 위해, 스케줄링 서버는 단계 S14를 수행하고, 즉, 제1 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사할 수 있다. 현재 공통 리소스 정보는 대안적으로 리소스 정보 개인 사본으로서 전체 복사될 수 있다.
- [0068] 그 후, 단계 S15에서, 스케줄링 서버는 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프를 정확하게 기록하기 위해 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프를 기록한다. 일부 실시예들에서, 현재의 타임스탬프는 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정될 수 있다. 예를 들어, 현재의 타임스탬프는 2017년 9월 16일 오전 8:00이고, 이 경우, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프 둘 다는 2017년 9월 16일 오전 8:00이다. 그 후, 단계 S16에서, 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프는 현재 업데이트를 완료하기 위해 리소스 정보 개인 사본에 저장된다.
- [0069] 또한, 시간 차이가 미리 설정된 전체 업데이트 임계값보다 큰지를 결정하는 단계 S12 이후에, 스케줄링 서버는 단계 S17을 수행할 수 있으며, 즉, 시간 차이가 전체 업데이트 임계값보다 작거나 같은 경우에, 업데이트 타임스탬프가 공통 리소스 정보로부터의 이전의 증분 업데이트 타임스탬프보다 큰 업데이트된 호스트 머신 정보를 획득할 수 있다.
- [0070] 단계 S18에서, 스케줄링 서버는 호스트 머신 정보로부터 VM 정보를 충족시키는 제2 타겟 호스트 머신 정보를 획득한다.
- [0071] 단계 S19에서, 스케줄링 서버는 제2 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 추가한다.
- [0072] 단계 S20에서, 스케줄링 서버는 현재의 타임스탬프를 제2 증분 업데이트 타임스탬프로 설정한다.
- [0073] 단계 S21에서, 스케줄링 서버는 제2 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장한다.
- [0074] 클라우드 컴퓨팅 시스템에서의 호스트 머신들의 스케일은 비교적 크고, 예를 들어, 호스트 머신들의 스케일은 10000의 자릿수를 가지며, 스케줄링 서버들의 스케일은 100의 자릿수를 갖는다. 따라서, 전체 업데이트가 수행되는 경우, 즉, 공통 리소스 정보가 리소스 정보 개인 사본이 업데이트될 때마다 리소스 정보 개인 사본으로서 전체 복사되는 경우, 많은 시간이 소비된다. 업데이트에 의해 소비되는 시간을 감소시키기 위해, 증분 업데이트가 리소스 정보에 대해 수행될 수 있으며, 즉, 공통 리소스 정보의 일부 리소스 정보가 리소스 정보 개인 사본에 추가된다.
- [0075] 구체적으로, 공통 리소스 정보 내의 호스트 머신 정보가 업데이트될 때마다, 정확한 업데이트 시간을 획득하기 위해 업데이트 타임스탬프가 기록된다. 스케줄링 서버가 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득한 후에, 시간 차이가 전체 업데이트 임계값보다 작거나 같다면, 스케줄링 서버는 공통 리소스 정보로부터 업데이트된 호스트 머신 정보를 획득하고, 호스트 머신 정보의 업데이트된 업데이트 타임스탬프들을 획득하고, 이어서 업데이트된 호스트 머신 정보로부터 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다. 즉, 공통 리소스 정보는 이전의 증분 업데이트 타임스탬프에 따라 질의되어, 업데이트 타임스탬프들이 이전의 증분 업데이트 타임스탬프보다 크거나 같은 모든 호스트 머신 정보를 획득한다. 이전의 증분 업데이트 타임스탬프는 현재의 타임스탬프에 대한 마지막 증분 업데이트 타임스탬프이다.
- [0076] 업데이트되고 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 호스트 머신 정보가 공통 리소스 정보로부터 획득된 후에, 호스트 머신 정보가 스크리닝된다. VM 정보를 충족시키는 제2 타겟 호스트 머신 정보는 호스트 머신 정보로부터 획득되고, 리소스 정보 개인 사본에 추가된다.
- [0077] 그 후, 증분 업데이트 타임스탬프를 정확하게 기록하기 위해, 제2 증분 업데이트 타임스탬프가 기록된다. 즉, 현재의 타임스탬프는 제2 증분 업데이트 타임스탬프로 설정되고, 제2 증분 업데이트 타임스탬프는 리소스 정보 개인 사본에 저장되어 현재 업데이트를 완료한다. 리소스 정보 개인 사본은 실질적으로 내부 메모리에서 실행되는 버퍼이다. 대부분의 경우들에서, 스케줄링 서버는 단지 증분 업데이트를 수행할 필요가 있다. 업데이트의 데이터 볼륨 및 소비된 시간은 둘 다 더 적다. 일부 실시예들에서, 현재 증분 업데이트 이전에 오프라인이 되는 호스트 머신은 리소스 정보 개인 사본으로부터 삭제된다.
- [0078] 제1 타겟 호스트 머신 정보, 제2 타겟 호스트 머신 정보, 제1 전체 업데이트 타임스탬프, 제1 증분 업데이트 타임스탬프, 및 제2 증분 업데이트 타임스탬프에서의 "제1" 및 "제2"는 타겟 호스트 머신 정보, 증분 업데이트 타임스탬프들, 또는 전체 업데이트 타임스탬프들을 구별하는데 이용되는 것에 불과하고, 그 중요도의 제한을 의미

하는 것은 아니다.

- [0079] 단계 104에서, 스케줄링 서버는, 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득한다.
- [0080] 리소스 정보 개인 사본을 업데이트한 후에, 스케줄링 서버는 리소스 정보 개인 사본에 기반하여 생성될 VM에 대한 호스트 머신들을 스크리닝한다. 스케줄링 서버는 리소스 정보 개인 사본 내의 호스트 머신 정보 및 VM 정보에서의 리소스 요건에 따라 리소스 정보 개인 사본 내의 호스트 머신들의 세트를 필터링하고, 자격을 갖는 호스트 머신을 예비하고, 자격을 갖지 않는 호스트 머신을 제거하여, 적어도 하나의 후보 호스트 머신을 획득할 수 있다.
- [0081] 일부 실시예들에서, 스케줄링 서버는 호스트 머신 정보에 기반하여 확립된 직렬 접속된 필터들 및 VM 정보에 따라, 복수의 호스트 머신들에 대한 대응하는 필터링을 수행하도록 각각의 필터를 제어하여, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득할 수 있다.
- [0082] 일부 실시예들에서, 스케줄링 서버는 후보 호스트 머신들로부터 생성될 VM의 리소스 요건을 차감하여 후보 호스트 머신들의 나머지 리소스들을 획득하고, 후보 호스트 머신들 중 하나의 후보 호스트 머신의 나머지 리소스들이 성공적으로 제출될 때까지 공통 리소스 정보에 나머지 리소스들을 제출할 수 있다. 스케줄링 서버는 후보 호스트 머신들 중 하나의 후보 호스트 머신에 대응하는 나머지 리소스들이 성공적으로 제출되는 경우에, 그 제출이 성공한 후보 호스트 머신을 타겟 호스트 머신으로 설정한다.
- [0083] 호스트 머신 정보에 따라 직렬 필터링 모델이 확립될 수 있다. 직렬 필터링 모델은 복수의 직렬 접속된 필터들을 포함한다. 각각의 필터는 하나의 필터링 기능을 담당하고, VM 정보에 따라 대응하는 판단을 행하고, 이어서 대응하는 필터링을 수행한다. 예를 들어, 제1 직렬 접속된 필터는, VM에 의해 요구되는 CPU 리소스, 내부 메모리 리소스, 및 디스크 리소스와 같은 리소스 요건에 따라 리소스 요건을 충족시키지 못하는 호스트 머신을 필터링하는 것을 담당하고, 리소스 요건을 충족시키는 호스트 머신만을 예비한다. 제2 직렬 접속된 필터는 SSD를 지원하거나 지원하지 않는 호스트 머신을 필터링하는 것을 담당한다. 호스트 머신이 리소스 요건에서 SSD를 지원하도록 요구되는 경우, 제2 필터는 리소스 요건에 따라 SSD를 지원하지 않는 호스트 머신을 필터링하고, SSD를 지원하는 호스트 머신을 예비한다.
- [0084] 다른 필터들의 기능들의 경우, 다른 필터들이 다른 필터링 기능들을 담당하는 것을 제외하고는 이해를 위해 이러한 필터를 참조할 수 있다는 점이 이해될 수 있다. 특정 필터링 모델에서의 필터들의 기능들은 생성될 VM의 리소스 요건에 따라 설정될 수 있고, 특정 콘텐츠로 본 명세서에서 제한되지는 않는다.
- [0085] 단계 105에서, 스케줄링 서버는 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성한다.
- [0086] 적어도 하나의 후보 호스트 머신을 획득한 후, 스케줄링 서버는 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신을 결정할 수 있다. 타겟 호스트 머신은 랜덤하게 선택될 수 있다. 물론, 가장 적절한 타겟 호스트 머신을 선택하기 위해, 복수의 호스트 머신들이 적어도 하나의 후보 호스트 머신을 획득하도록 필터링된 후에, 각각의 차원들에서의 각각의 후보 호스트 머신의 우선순위 값들이 결정될 수 있고, 적어도 하나의 후보 호스트 머신은 우선순위 값들에 따라 순위화된다.
- [0087] 일부 실시예들에서, 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하는 단계는 다음의 단계들을 포함할 수 있다:
- [0088] 단계 S31. 스케줄링 서버는 각각의 차원들에서 적어도 하나의 후보 호스트 머신의 각각의 호스트 머신의 우선순위 값들을 결정한다.
- [0089] 단계 S32. 스케줄링 서버는 차원들에서 우선순위 값들의 내림차순으로 순위화를 수행하여 후보 호스트 머신 리스트를 생성한다.
- [0090] 단계 S33. 스케줄링 서버는 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여, 나머지 리소스들을 획득하고, 리소스 요건은 VM에 대응하는 VM 정보에 포함된다.
- [0091] 단계 S34. 스케줄링 서버는 나머지 리소스들을 공통 리소스 정보에 제출한다.
- [0092] 단계 S35. 스케줄링 서버는, 하나의 후보 호스트 머신에 대응하는 나머지 리소스들이 성공적으로 제출되는 경우

에, 그 제출이 성공한 후보 호스트 머신을 타겟 호스트 머신으로 설정한다.

- [0093] 구체적으로, 차원들에서의 각각의 호스트 머신의 우선순위 값들이 먼저 계산된다. 차원들에서의 우선순위들은 개인 네트워크 우선순위(vpcPrio), 동시 작업 우선순위(taskPrio), 이력 에러 우선순위(errorPrio), 반-친화도 우선순위(affinityPrio), 유휴 우선순위(freePrio), 이미지 캐시 우선순위(imgPrio), 단편화 비율 우선순위(fragmentPrio) 등을 포함할 수 있다.
- [0094] 개인 네트워크 우선순위의 값의 경우, vpcPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 개인 네트워크 우선순위의 값은 각각의 호스트 머신에 대해 결정될 수 있다. 예를 들어, 0은 개인 네트워크를 지원하는 호스트 머신의 vpcPrio에 할당될 수 있고, 1은 개인 네트워크를 지원하지 않는 호스트 머신의 vpcPrio에 할당될 수 있고; 대안적으로, 1은 개인 네트워크를 지원하는 호스트 머신의 vpcPrio에 할당될 수 있고, 0은 개인 네트워크를 지원하지 않는 호스트 머신의 vpcPrio에 할당될 수 있고, 다른 것들도 그런 식이다. 설명의 용이함을 위해, 본 출원의 실시예들에서, 개인 네트워크를 지원하는 호스트 머신의 vpcPrio에 0이 할당되고, 개인 네트워크를 지원하지 않는 호스트 머신의 vpcPrio에 1이 할당되는 예가 설명을 위해 이용된다.
- [0095] 개인 네트워크의 IP 어드레스는 물리적 네트워크 토폴로지에 의존하지 않고, 미래에 이동하기가 더 용이하다. 개인 네트워크 우선순위의 효과는 개인 네트워크를 지원하는 호스트 머신 상에 VM을 우선적으로 배치하는 것이어서, 미래의 이동이 더 편리하고 유연하게 된다.
- [0096] 동시 작업 우선순위의 경우, taskPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 각각의 호스트 머신에 대해 동시 작업 우선순위가 결정될 수 있다. 호스트 머신 상의 현재 작업들의 양이 결정되고, 작업들의 양이 미리 설정된 임계값 미만인 경우, 0이 taskPrio에 할당되고; 그렇지 않으면, 현재 작업들의 양이 taskPrio에 할당된다. 예를 들어, 미리 설정된 임계값은 디폴트로 3이다. 물론, 미리 설정된 임계값은 실제 요건들에 따라 유연하게 구성될 수 있다.
- [0097] 호스트 머신 상의 동시 작업들은 동시에 수행될 수 있고, 동시성은 과도하게 높다. 따라서, VM의 생성은 느리거나 심지어 실패할 수 있다. 동시 작업 우선순위의 효과는 보다 적은 양의 동시 작업들을 갖는 호스트 머신 상에 VM을 우선적으로 배치하여 VM을 생성하는 성공률 및 실시간 성능을 보장하는 것이다.
- [0098] 이력 에러 우선순위의 경우, errorPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 이력 에러 우선순위는 각각의 호스트 머신에 대해 계산될 수 있다. 마지막 기간(예를 들어, 디폴트로 24시간) 내의 호스트 머신의 이력 에러들의 양이 결정된다. 이력 에러들의 양이 에러 임계값보다 작으면, 0이 errorPrio에 할당되고; 그렇지 않은 경우, 이력 에러들의 양이 errorPrio에 할당된다. 예를 들어, 에러 임계값은 디폴트로 5일 수 있고, 에러 임계값은 실제 요건들에 따라 유연하게 구성될 수 있다.
- [0099] 짧은 기간 내에 복수의 에러가 발생하는 호스트 머신은 어떤 유형의 결함을 겪을 수 있고, 호스트 머신이 처리되기 전에, VM이 호스트 머신 상에 계속 배치되는 경우, VM을 생성하는 작업은 실패할 수 있고, 성공률은 감소된다. 이력 에러 우선순위의 효과는 과도한 에러가 발생하지 않는 호스트 머신 상에 VM을 우선적으로 배치하는 것에 의해, VM을 생성하는 성공률을 향상시키는 것이다.
- [0100] 반-친화도 우선순위의 경우, affinityPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 각각의 호스트 머신에 대해 반-친화도 우선순위가 계산될 수 있다. 호스트 머신 상의 VM 사용자의 기존의 VM들의 양이 결정되고, VM들의 양이 친화도 임계값보다 작으면, 0이 affinityPrio에 할당되고; 그렇지 않으면, 기존의 VM들의 양이 affinityPrio에 할당된다. 예를 들어, 친화도 임계값은 3이고, 물론, 친화도 임계값은 실제 요건들에 따라 유연하게 구성될 수 있다.
- [0101] 대규모 분산 시스템에서는, 에러들을 완전히 방지하는 것은 거의 불가능하다. 사용자의 VM들이 하나 이상의 호스트 머신 상에서 클러스터링되는 경우, 호스트 머신들이 결함이 있을 때 호스트 머신들 상의 VM들은 모두 영향을 받고, 사용자의 전체 애플리케이션의 이용가능성에 심각하게 영향을 미친다. 반-친화도 우선순위의 효과는 동일한 사용자의 VM들이 상이한 호스트 머신들 상에 널리 분산되게 하여, 분산된 재난 복구를 수행하게 함으로써, 단일 호스트 머신의 결함이 전체 애플리케이션의 이용가능성에 비교적 제한된 영향을 미치게 하고, 이에 의해 사용자의 애플리케이션의 안정성 및 이용가능성을 향상시키는 것이다.
- [0102] 유휴 우선순위의 경우, freePrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 예를 들어, 각각의 호스트 머신이 완전히 유휴인지가 결정될 수 있다. 각각의 호스트 머신이 완전히 유휴인 경우, 0이 freePrio에 할당되고; 그렇지 않으면, 1이 freePrio에 할당된다. 대안적으로, 각각의 호스트 머신이 완전히 유휴인 경우, 1이 freePrio에 할당되고; 그렇지 않으면, 0이 freePrio에 할당되며, 다른 것들도 그런 식이다. 설명의 용이함을

을 위해, 본 출원의 실시예들에서, 각각의 호스트 머신이 완전히 유희인 경우, 0이 freePrio에 할당되고; 그렇지 않으면, 1이 freePrio에 할당되는 예가 설명을 위해 이용된다.

- [0103] 유희 우선순위의 효과는 유희 호스트 머신을 보호하고, 비-유희 호스트 머신 상에 VM을 우선적으로 배치하여 비-유희 호스트 머신 상의 리소스들을 완전히 이용함으로써, 리소스 이용을 향상시키는 것이다.
- [0104] 이미지 캐시 우선순위의 경우, imgPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 각각의 호스트 머신이 VM에 의해 요구되는 이미지 파일을 갖는지를 결정하기 위해 후보 호스트 머신 리스트가 순회될 수 있다. 예를 들어, 각각의 호스트 머신이 이미지 파일을 갖는다면, 0이 imgPrio에 할당되고; 그렇지 않으면, 1이 imgPrio에 할당된다.
- [0105] 이미지 캐시 우선순위의 효과는 VM 이미지를 저장한 호스트 머신 상에 VM을 우선적으로 배치하는 것이고, 따라서 VM이 실제로 생성될 때 이미지 다운로드의 단계가 생략되어, 생성 시간을 감소시키고, VM 생성 프로세스의 실시간 성능을 향상시킬 수 있다.
- [0106] 단편화 비율 우선순위의 경우, fragmentPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 단편화 비율 우선 순위는 각각의 호스트 머신에 대해 계산될 수 있다. 우선 순위는 2개의 차원, 즉 CPU 및 내부 메모리에서의 리소스들에 따라 계산된다. 디폴트 경우에, 단편화 비율 우선 순위는 내부 메모리의 단편화 비율로 곱해진 CPU의 단편화 비율과 동일하다.
- [0107] 단편화 비율 우선순위의 효과는 호스트 머신의 리소스들을 완전히 이용하여 리소스 단편화 비율을 감소시키고, 또한 리소스 이용을 향상시키고 동작 비용들을 감소시키는 것이다.
- [0108] 차원들에서의 각각의 후보 호스트 머신의 우선 순위 값들이 결정된 후, 호스트 머신들은 차원들에서 우선 순위 값들의 내림차순으로 순위화되어 후보 호스트 머신 리스트를 획득하고, 후보 호스트 머신 리스트에서의 가장 적절한 후보 호스트 머신이 타겟 호스트 머신으로서 결정될 수 있다. 특정 결정 프로세스는 다음과 같을 수 있다:
- [0109] 호스트 머신들은 차원들의 우선 순위들의 내림차순에 기반하여 동일한 차원에서 우선 순위 값들의 내림차순으로 순위화된다. 하나의 차원에서 동일한 우선 순위 값을 갖는 호스트 머신들은 다음 차원에서의 각각의 우선 순위 값들에 따라 다시 순위화되고, 따라서 호스트 머신들이 우선 순위들의 내림차순으로 순위화되는 후보 호스트 머신 리스트를 획득한다.
- [0110] 후보 호스트 머신들이 순위화된 후에, 첫 번째로 순위화된 후보 호스트 머신이 최적이다. 다른 생성 작업이 최적의 후보 호스트 머신에 존재할 수 있다. 따라서, 생성의 성공이 반드시 보장될 수 없고, 타겟 호스트 머신은 리소스들의 차감을 시뮬레이션하는 방식으로 결정될 필요가 있다.
- [0111] 타겟 호스트 머신을 결정하는 프로세스에서, 먼저, 각각의 후보 호스트 머신에 대응하는 나머지 리소스를 획득하기 위해, 후보 호스트 머신 리스트에서의 순위화된 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건이 하나씩 차감되어, 호스트 머신에 대한 리소스들의 차감 시뮬레이션이 완료된다. VM의 리소스 요건은 VM에 대응하는 VM 정보에 포함된다. 즉, 생성될 VM의 리소스 요건은 순위화된 후보 호스트 머신들의 내림차순으로 첫 번째로 순위화된 후보 호스트 머신으로부터 시작하여 후보 호스트 머신들로부터 하나씩 차감된다. 리소스 요건이 후보 호스트 머신으로부터 성공적으로 차감될 때, 그 차감이 성공한 후보 호스트 머신이 타겟 호스트 머신으로서 이용된다.
- [0112] 순위는 미리 설정될 수 있다. 예를 들어, 디폴트 순위 표준, 즉 우선 순위들의 내림차순은 개인 네트워크 우선 순위(vpcPrio), 동시 작업 우선 순위(taskPrio), 이력 에러 우선 순위(errorPrio), 반-친화도 우선 순위(affinityPrio), 유희 우선 순위(freePrio), 이미지 캐시 우선 순위(imgPrio), 및 단편화 비율 우선 순위(fragmentPrio)일 수 있다. 순위 표준은 또한 디폴트 표준 범위에서 선택될 수 있고, 그 순서는 실제 요건들에 따라 유연하게 조정되고 구성될 수 있다.
- [0113] 다음으로, 후보 호스트 머신의 나머지 리소스가 획득된 후에, 나머지 리소스는 공통 리소스 정보에 제출된다. 하나의 후보 호스트 머신에 대응하는 나머지 리소스가 성공적으로 제출되는 경우, 그 제출이 성공한 후보 호스트 머신이 타겟 호스트 머신으로 설정된다. 대안적으로, 후보 호스트 머신의 나머지 리소스는 각각의 호스트 머신의 나머지 리소스 데이터를 유지하는데 이용되는 데이터베이스에 기입된다. 기입이 성공할 때, 그 기입이 성공한 후보 호스트 머신이 타겟 호스트 머신으로서 이용된다.
- [0114] 타겟 호스트 머신이 결정된 후에, VM은 타겟 호스트 머신 상에 생성될 수 있다. 일부 실시예들에서, VM의 생성이 타겟 호스트 머신의 리소스를 소비할 필요가 있기 때문에, 리소스 정보 개인 사본 내의 호스트 머신 정보는

업데이트될 수 있고, 클라우드 컴퓨팅 시스템은 공통 리소스 정보를 업데이트하도록 추가로 지시받을 수 있다. 후보 호스트 머신에 대응하는 나머지 리소스가 제출되지 못하는 경우, 이것은 스케줄링 충돌이 아마 발생하는 것, 즉, 복수의 리소스 스케줄링 프로세스들이 동일한 호스트 머신을 동시에 선택한다는 것을 나타낸다. 이 경우, 제출 횟수들의 양이 양 임계값을 초과하는지가 결정된다. 그렇지 않다면, 후보 호스트 머신의 호스트 머신 정보가 다시 획득되고, 생성될 VM의 리소스 요건이 차감되어, 나머지 리소스를 다시 획득한다. 이어서, 다시 획득된 나머지 리소스가 공통 리소스 정보에 다시 제출된다. 제출 횟수들의 양이 양 임계값을 초과하는 경우, 리소스의 차감 시뮬레이션을 수행하기 위해 후보 호스트 머신 리스트로부터 다음 후보 호스트 머신이 획득된다.

[0115] 제1 호스트 머신에 대해 차감이 성공할 때, 제1 호스트 머신은 제2 호스트 머신에 대해 차감 동작을 수행하지 않고 타겟 호스트 머신으로서 직접 결정될 수 있다. 제1 호스트 머신에 대해 차감이 성공적이지 않을 때, 그 차감 동작은 호스트 머신에 대한 차감이 성공할 때까지 제2 호스트 머신에 대해 수행될 필요가 있다.

[0116] 또한, 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하는 단계 전에, 이 방법은 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하여 후보 호스트 머신 타겟 리스트를 생성하는 단계를 더 포함할 수 있다.

[0117] 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하는 단계는, 후보 호스트 머신 타겟 리스트에서의 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신 타겟 리스트에서의 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하는 단계를 포함한다.

[0118] 미리 설정된 양은 실제 요건들에 따라 유연하게 설정될 수 있다. 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양은 랜덤하게 재순위화될 수 있거나, 미리 설정된 규칙에 따라 재순위화될 수 있다. 특정 재순위화 방식은 본 명세서에서 제한되지 않는다. 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하는 목적은 재순위화를 위해 후보 호스트 머신 리스트에서 제1 k(k는 미리 설정된 양)개의 후보 호스트 머신을 랜덤하게 분산시키고, 후보 호스트 머신 타겟 리스트를 획득하여, 복수의 리소스 스케줄링 프로세스들이 동시 시나리오에서 동일한 타겟 호스트 머신을 선택하는 것을 방지할 수 있어서, 스케줄링 충돌들을 효과적으로 피하게 하기 위한 것이다.

[0119] 전술한 것으로부터 알 수 있듯이, 본 출원의 이 실시예에서, 생성될 VM에 대응하는 VM 정보가 획득되고, 공통 리소스 정보가 획득된다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함한다. 다음으로, 미리 설정된 리소스 정보 개인 사본은 공통 리소스 정보 및 VM 정보에 따라 업데이트된다. 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 마지막으로, 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신이 획득되고, 타겟 호스트 머신이 적어도 하나의 후보 호스트 머신으로부터 획득되고, VM이 타겟 호스트 머신 상에 생성된다. 이러한 해결책에서, 리소스 정보 개인 사본은 리소스 스케줄링이 수행되기 전에 제시간에 업데이트될 수 있고, 이는 리소스 정보 개인 사본과 공통 리소스 정보의 동기화를 보장하여, 더 나은 리소스 스케줄링 결과가 달성되고, 리소스 스케줄링 동안 대기할 필요가 없고, 리소스 스케줄링의 효율을 향상시킨다.

[0120] 전술한 실시예에서 설명된 방법에 따라, 다음에서는 예를 이용하여 상세한 설명들이 추가로 제공된다.

[0121] 스케줄러가 예로서 이용된다. 복수의 VM 생성 요청이 있을 때, 각각의 VM 생성 요청에 대응하는 메시지가 메시지 큐에 저장된다. 복수의 스케줄러는 메시지 큐 내의 메시지에서부터 VM 생성 요청을 동시에 추출하고, VM 생성 요청에서 운반된 VM 정보에 따라 각각의 VM 생성 요청에 적절한 호스트 머신을 선택하고, 호스트 머신 상에 대응하는 VM을 생성할 수 있다. 복수의 스케줄러들이 동시에 리소스 스케줄링을 수행하기 때문에, 호스트 머신들은 상이한 VM들에 대해 동시에 선택될 수 있고, 따라서 클라우드 컴퓨팅 시스템의 처리량 및 동시성 능력이 향상되어 스케줄러가 전체 시스템의 성능 병목이 되는 것을 방지할 수 있다.

[0122] 도 3은 본 출원의 실시예에 따른 리소스 스케줄링 방법에서 리소스 스케줄링을 복수의 스케줄러들에 의해 동시에 수행하는 개략적인 흐름도이다. 클라우드 컴퓨팅 시스템이 3개의 스케줄러 및 3개의 호스트 머신을 갖는 것으로 가정되지만, 물론, 이러한 클라우드 컴퓨팅 시스템이 3개의 스케줄러 및 3개의 호스트 머신만을 갖는다는 것은 불가능하다. 이 실시예는 단지 주어진 예를 설명하는 용이함을 위한 것이지만, 스케줄러들의 양 및 호스트 머신들의 양을 제한하는 것으로 이해되어서는 안 된다. 그러나, 스케줄러들의 양과 호스트 머신들의 양에 관계없이, 리소스 스케줄링 프로세스들은 유사하고, 이 예에 따라 모두 이해될 수 있다.

- [0123] 이 방법 프로세스는 다음의 단계들을 포함할 수 있다:
- [0124] 201. 단말기는 VM 정보를 운반하는 VM 생성 요청을 클라우드 컴퓨팅 시스템 내의 수신기에 전송한다.
- [0125] 클라우드 컴퓨팅 시스템의 데이터 센터의 스케일의 연속적인 확장으로 인해, 호스트 머신들의 유형들 및 양이 증가하고 있고, 리소스 스케줄링의 논리가 더 복잡해지고 있다. 서비스 요건들의 급속한 성장에 의해, 시간 당 위당 처리될 필요가 있는 VM 생성 요청들의 양이 빠르게 증가하고, 이는 리소스 스케줄링의 응답 레이턴시를 더 증가시킨다. 종래의 리소스 스케줄링 해결책의 높은 응답 레이턴시 및 낮은 처리율의 문제점들을 해결하기 위해, 복수의 스케줄러들은 이 실시예에서 동시에 리소스 스케줄링을 수행한다.
- [0126] 먼저, 단말기는 VM 정보를 운반하는 VM 생성 요청을 클라우드 컴퓨팅 시스템 내의 수신기로 전송한다. 단말기는 VM을 생성하는 것을 담당하는 사용자에게 의해 이용되는 단말기 디바이스일 수 있다. VM 정보는 위에서 설명된 VM 정보와 유사하고, 본 명세서에서 설명되지 않는다.
- [0127] 202. 수신기는 수신된 VM 생성 요청을 메시지로 캡슐화하고, 이 메시지를 메시지 큐에 저장한다.
- [0128] 수신기는 메시지 큐를 유지한다. VM 생성 요청을 수신한 후, 수신기는 VM 생성 요청을 메시지로 캡슐화한다. 메시지의 포맷은 실제 요건들, 예를 들어, 키-값 또는 제이슨(json) 메시지 포맷에 따라 유연하게 설정될 수 있다. 그 후, 수신기는 메시지 큐에 메시지를 저장하고, 따라서 스케줄러는 메시지 큐로부터, 생성될 VM에 대응하는 VM 정보를 추출할 수 있다. 수신기가 특정 기간 내에 많은 양의 VM 생성 요청들을 수신할 때, 메시지 큐는 복수의 메시지를 저장할 수 있다. 도 4에 도시된 바와 같이, 메시지 큐는 도 4의 메시지 VM1을 메시지 VMn에 저장하고, 여기서 n은 양의 정수이고 n>1이다.
- [0129] 203. 스케줄러 1은 VM 정보 1을 획득한다.
- [0130] 204. 스케줄러 2는 VM 정보 2를 획득한다.
- [0131] 205. 스케줄러 3은 VM 정보 3을 획득한다.
- [0132] 단계 203 내지 단계 205는 동시에 수행될 수 있거나, 순서대로 수행될 수 있다. 대안적으로, 단계 205가 먼저 수행될 수 있고, 이후 단계 203이 수행될 수 있고, 단계 204가 최종적으로 수행된다. 대안적으로, 단계 204가 먼저 수행될 수 있고, 이후 단계 205가 수행될 수 있고, 단계 203이 최종적으로 수행되거나, 또는 다른 것들도 그런 식이다. 단계들의 특정 순서는 본 명세서에서 제한되지 않는다.
- [0133] 리소스 스케줄링을 수행하기 위한 구성요소로서, 스케줄러는 VM에 대한 적절한 호스트 머신을 선택한다. 스케줄러가 기능 블랙 박스로서 고려되면, 그 입력은 호스트 머신 정보 및 VM 정보일 수 있고, 그 출력은 VM ID와 호스트 머신 ID, 즉 <VM ID, HOST ID>의 매칭 쌍일 수 있다.
- [0134] 스케줄러 1, 스케줄러 2, 및 스케줄러 3은 동시에 리소스 스케줄링을 수행할 수 있고, 대응하는 VM 정보를 각각 획득하여, 시스템의 동시성 레벨 및 처리율을 향상시킬 수 있다.
- [0135] 스케줄러 1이 VM 정보 1을 획득하는 방식은 다음을 포함할 수 있다: 방식 1: 스케줄러 1은 VM 정보 1을 획득하라는 요청을 수신기에 전송하고, 수신기는 VM 정보 1을 획득하라는 수신된 요청에 따라 VM 정보 1을 운반하는 메시지 1을 스케줄러 1에 전송하고, 여기서 그 요청은 스케줄러 1에 의해 전송되고, 스케줄러 1은 수신된 메시지 1로부터 VM 정보 1을 추출할 수 있다. 방식 2: 스케줄러 1은 메시지 큐에 저장된 메시지를 직접 판독하고, 그 메시지에서 VM 정보 1을 획득한다.
- [0136] 스케줄러 2가 VM 정보 2를 획득하는 방식 및 스케줄러 3이 VM 정보 3을 획득하는 방식은 스케줄러 1이 VM 정보 1을 획득하는 방식과 유사하고, 본 명세서에서 설명되지 않는다.
- [0137] 206. 스케줄러 1은 공통 리소스 정보를 획득한다.
- [0138] 207. 스케줄러 2는 공통 리소스 정보를 획득한다.
- [0139] 208. 스케줄러 3은 공통 리소스 정보를 획득한다.
- [0140] 단계 206 내지 단계 208은 동시에 수행될 수 있거나, 순서대로 수행될 수 있다. 대안적으로, 단계 208이 먼저 수행될 수 있고, 다음으로 단계 206이 수행될 수 있고, 단계 207이 최종적으로 수행되거나, 또는 다른 것들도 그런 식이다. 단계들의 특정 순서는 본 명세서에서 제한되지 않는다.
- [0141] 클라우드 컴퓨팅 시스템 내의 리소스 관리자는 공통 리소스 정보를 유지하고, 공통 리소스 정보는 클라우드 컴

퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함한다. 호스트 머신이 클라우드 컴퓨팅 시스템에 진입할 때, 필요한 초기화가 호스트 머신 도입 프로세스를 통해 수행될 필요가 있다. 호스트 머신은 그 호스트 머신 정보를 리소스 관리자에 보고하고, 리소스 관리자는 호스트 머신 정보를 공통 리소스 정보에 저장할 수 있다. 호스트 머신 정보가 업데이트될 때, 리소스 관리자는 또한 제시간에 공통 리소스 정보를 업데이트할 수 있다.

- [0142] 스케줄러 1이 공통 리소스 정보를 획득하는 방식은 다음 방식들을 포함할 수 있다: 방식 1: 스케줄러 1은 공통 리소스 정보 획득 요청을 리소스 관리자에게 전송하고, 리소스 관리자는 스케줄러 1에 의해 전송된 수신된 공통 리소스 정보 획득 요청에 따라 대응하는 공통 리소스 정보 정보를 스케줄러 1에 전송한다. 방식 2: 스케줄러 1은 공통 리소스 정보를 저장하는 데이터베이스를 직접 판독하고, 데이터베이스로부터 공통 리소스 정보를 획득한다.
- [0143] 스케줄러 2가 공통 리소스 정보를 획득하는 방식 및 스케줄러 3이 공통 리소스 정보를 획득하는 방식은 스케줄러 1이 공통 리소스 정보를 획득하는 방식과 유사하고, 본 명세서에서 설명되지 않는다. 스케줄러 1, 스케줄러 2, 및 스케줄러 3은 공통 리소스 정보를 동시에 또는 미리 설정된 간격으로 획득할 수 있다.
- [0144] 209. 스케줄러 1은 수신된 공통 리소스 정보 및 VM 정보 1에 따라, 자체적으로 유지되는 리소스 정보 개인 사본(1)을 업데이트한다.
- [0145] 210. 스케줄러 2는 수신된 공통 리소스 정보 및 VM 정보 2에 따라, 자체적으로 유지되는 리소스 정보 개인 사본(2)을 업데이트한다.
- [0146] 211. 스케줄러 3은 수신된 공통 리소스 정보 및 VM 정보 3에 따라, 자체적으로 유지되는 리소스 정보 개인 사본(3)을 업데이트한다.
- [0147] 단계 209 내지 단계 210이 수행되는 순서는 실제 상황들에 따라 유연하게 설정될 수 있고, 단계들의 특정 순서는 본 명세서에서 제한되지 않는다.
- [0148] 스케줄러 1은 리소스 정보 개인 사본(1)을 자체적으로 유지하고, 스케줄러 2는 리소스 정보 개인 사본(2)을 자체적으로 유지하고, 스케줄러 3은 리소스 정보 개인 사본(3)을 자체적으로 유지한다. 리소스 정보 개인 사본(1), 리소스 정보 개인 사본(2), 및 리소스 정보 개인 사본(3)은 각각 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 미리 설정된 호스트 머신은 클라우드 컴퓨팅 시스템 내의 호스트 머신들의 일부 또는 모든 호스트 머신들일 수 있다. 리소스 정보 개인 사본은 리소스 정보의 일관성 및 정보의 정확도를 보장하기 위해 공통 리소스 정보를 통해 업데이트된다.
- [0149] 또한, 도 6에 도시된 바와 같이, 스케줄러 1에 의해, 수신된 공통 리소스 정보 및 VM 정보 1에 따라 자체적으로 유지되는 리소스 정보 개인 사본(1)을 업데이트하는 단계는 다음의 단계들을 포함할 수 있다:
- [0150] 301. 스케줄러 1은 리소스 정보 개인 사본이 처음으로 업데이트되는지를 결정한다. 만약 그렇다면, 단계 302 내지 단계 304는 순서대로 수행되고; 그렇지 않다면, 단계 305가 수행된다.
- [0151] 302. 스케줄러 1은 공통 리소스 정보로부터 VM 정보를 추출시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다.
- [0152] 303. 스케줄러 1은 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다.
- [0153] 304. 스케줄러 1은 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프를 기록한다.
- [0154] 스케줄러 1은 현재의 타임스탬프를 획득할 수 있고, 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로 설정하고; 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장한다.
- [0155] 305. 스케줄러 1은 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득한다.
- [0156] 306. 스케줄러 1은 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정한다. 만약 그렇다면, 단계 307 내지 단계 309는 순서대로 수행되고; 그렇지 않다면, 단계 310이 수행된다.
- [0157] 307. 스케줄러 1은 공통 리소스 정보로부터 VM 정보를 추출시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다.

- [0158] 308. 스케줄러 1은 제1 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다.
- [0159] 309. 스케줄러 1은 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프를 기록한다.
- [0160] 스케줄러 1은 현재의 타임스탬프를 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정할 수 있고; 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장한다.
- [0161] 310. 스케줄러 1은 공통 리소스 정보로부터 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 업데이트된 호스트 머신 정보를 획득한다.
- [0162] 311. 스케줄러 1은 VM 정보를 충족시키는 제2 타겟 호스트 머신 정보를 호스트 머신 정보로부터 획득한다.
- [0163] 312. 스케줄러 1은 제2 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 추가한다.
- [0164] 313. 스케줄러 1은 제2 증분 업데이트 타임스탬프를 기록한다.
- [0165] 스케줄러 1은 현재의 타임스탬프를 제2 증분 업데이트 타임스탬프로 설정할 수 있고; 제2 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장한다. 리소스 정보 동기화의 버퍼링 메커니즘에서, 리소스 정보 동기화에 의해 소비되는 시간을 상당히 감소시키기 위해, 리소스 정보 개인 사본에 대해 증분 업데이트가 수행될 수 있다.
- [0166] 스케줄러 2는 공통 리소스 정보 및 VM 정보 2에 따라 리소스 정보 개인 사본(2)을 업데이트하고, 스케줄러 3은 공통 리소스 정보 및 VM 정보 3에 따라 리소스 정보 개인 사본(3)을 업데이트한다. 스케줄러 2 및 스케줄러 3의 업데이트 방식들은 스케줄러 1의 업데이트 방식과 유사하고, 본 명세서에서 설명되지 않는다.
- [0167] 도 4에 도시된 바와 같이, 스케줄러들의 양이 제한되지 않을 때, n (n 은 1보다 큰 양의 정수임)개의 스케줄러는 메시지 큐 내의 메시지들로부터 VM 정보를 동시에 추출할 수 있고, 공통 리소스 정보에 따라, 스케줄러들에 의해 유지되는 리소스 정보 개인 사본들을 각각 업데이트할 수 있다.
- [0168] 212. 스케줄러 1은 리소스 정보 개인 사본(1)에 따라 적어도 하나의 후보 호스트 머신을 결정하고, 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신 A를 결정한다.
- [0169] 213. 스케줄러 2는 리소스 정보 개인 사본(2)에 따라 적어도 하나의 후보 호스트 머신을 결정하고, 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신 B를 결정한다.
- [0170] 214. 스케줄러 3은 리소스 정보 개인 사본(3)에 따라 적어도 하나의 후보 호스트 머신을 결정하고, 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신 C를 결정한다.
- [0171] 단계 212 내지 단계 214가 수행되는 순서는 실제 상황들에 따라 유연하게 설정될 수 있고, 단계들의 특정 순서는 본 명세서에서 제한되지 않는다.
- [0172] 스케줄러 1은 리소스 정보 개인 사본 내의 VM 정보 및 호스트 머신 정보에서의 리소스 요건에 따라 리소스 정보 개인 사본 내의 호스트 머신들의 세트를 필터링하고, 자격을 갖춘 호스트 머신을 예비하고, 자격을 갖지 않는 호스트 머신을 제거하여, 적어도 하나의 후보 호스트 머신을 획득할 수 있다.
- [0173] 일부 실시예들에서, 스케줄러 1은 확립된 직렬 접속된 필터들에 따라, 복수의 호스트 머신들에 대한 대응하는 필터링을 수행하도록 각각의 필터를 제어하여, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득할 수 있다. 8개의 호스트 머신이 예로서 이용된다. 8개의 호스트 머신은 호스트 머신 정보 및 생성될 VM의 VM 정보에 따라 필터링된다. 호스트 머신들이 필터 A를 통과하고, 필터 A가 VM에 의해 요구되는 CPU 리소스들, 내부 메모리 리소스들, 및 디스크 리소스들과 같이, 호스트 머신들에 의해 제공될 수 있는 하드웨어 리소스 요건들을 충족시키지 않는 호스트 머신들을 필터링하는 것을 담당할 때, 리소스 요건들을 충족시키는 호스트 머신들을 예비하기 위해, 2개의 호스트 머신이 필터링 라운드를 통해 필터링된다. 이어서, 나머지 6개의 호스트 머신이 필터 B에 진입한다. 필터 B가 VM 정보에 따라 SSD를 지원하지 않는 호스트 머신을 필터링하는 것을 담당하는 경우, 하나 이상의 호스트 머신이 필터 B에 의해 수행되는 필터링 후에 필터링되고, 5개의 호스트 머신이 남는다. 그 다음, 5개의 호스트 머신은 다른 필터들을 순차적으로 통과한다. 다른 필터들은 클라우드 디스크를 지원하지 않는 호스트 머신을 필터링하고, 개인 네트워크를 지원하지 않는 호스트 머신을 필터링하는 것 등을 담당할 수 있다. 마지막으로, 나머지 호스트 머신들이 필터 X에 도달하면, 필터 X는 가상화 유형을 충족시키지 않는 하나의 호스트 머신을 필터링하는 것을 담당하고, 예를 들어, 호스트 머신의 가상화 유형이 KVM일 필요가

있다면, 그 가상화 유형이 Xen인 호스트 머신이 필터링된다. 필터링이 종료된 후, 3개의 후보 호스트 머신, 즉, 후보 호스트 머신 A, 후보 호스트 머신 B, 및 후보 호스트 머신 C가 획득된다.

- [0174] 또한, 도 7에 도시된 바와 같이, 적어도 하나의 후보 호스트 머신을 획득한 후에, 스케줄러 1은 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신을 결정할 수 있고, 특정 실행 단계들은 다음의 단계들을 포함할 수 있다:
- [0175] 401. 스케줄러 1은 각각의 차원들에서 적어도 하나의 후보 호스트 머신의 각각의 호스트 머신의 우선순위 값들을 결정한다.
- [0176] 차원들에서의 각각의 호스트 머신의 우선순위 값들을 계산하는 방식은 전술한 실시예에서 설명된 우선순위 값들을 계산하는 방식과 유사하고, 본 명세서에서 설명되지 않는다.
- [0177] 예를 들어, 3개의 획득된 후보 호스트 머신이 후보 호스트 머신 A, 후보 호스트 머신 B, 및 후보 호스트 머신 C 일 때, 차원들에서의 후보 호스트 머신 A, 후보 호스트 머신 B, 및 후보 호스트 머신 C의 우선순위 값들이 개별적으로 계산될 수 있다.
- [0178] 402. 스케줄러 1은 차원들에서 우선순위 값들의 내림차순으로 순위화를 수행하여 후보 호스트 머신 리스트를 생성한다.
- [0179] 순위화 방식은 전술한 실시예에서 설명된 순위화 방식과 유사하고, 본 명세서에서 설명되지 않는다.
- [0180] 예를 들어, 스케줄러 1에 의해 생성된 후보 호스트 머신 리스트 1에서, 후보 호스트 머신들은 후보 호스트 머신 A, 후보 호스트 머신 B 및 후보 호스트 머신 C로서 순차적으로 우선순위 값들의 내림차순으로 순위화된다.
- [0181] 403. 스케줄러 1은 순위에 따라 후보 호스트 머신 리스트로부터 호스트 머신을 선택한다.
- [0182] 예를 들어, 스케줄러 1은 생성된 후보 호스트 머신 리스트 1로부터 후보 호스트 머신 A를 선택한다.
- [0183] 404. 스케줄러 1은, VM의 리소스 요건에 따라, 차감이 수행되는 호스트 머신의 나머지 리소스를 계산한다.
- [0184] 예를 들어, VM의 리소스 요건은 8-코어 및 8G의 리소스들이고, 후보 호스트 머신 A의 호스트 머신 정보는 10-코어 및 10G의 리소스들을 포함하고, 8-코어 및 8G의 리소스들은 VM을 생성하도록 차감될 필요가 있어서, 차감 후 후보 호스트 머신 A의 나머지 리소스들은 2-코어 및 2G의 리소스들이다.
- [0185] 405. 스케줄러 1은 나머지 리소스를 공통 리소스 정보에 제출한다.
- [0186] 406. 스케줄러 1은 그 제출이 성공했는지를 결정한다. 만약 그렇다면, 단계 407 및 단계 408은 순서대로 수행되고; 그렇지 않다면, 단계 409가 수행된다.
- [0187] 407. 스케줄러 1은 호스트 머신이 타겟 호스트 머신인 것으로 결정한다.
- [0188] 그 제출이 성공한 경우, 이것은 공통 리소스 정보 내의 나머지 리소스의 값에 대한 수정이 성공한다는 것을 나타낸다. 스케줄러 1은 하나의 후보 호스트 머신에 대응하는 나머지 리소스가 성공적으로 제출되는 경우에, 그 제출이 성공한 후보 호스트 머신을 타겟 호스트 머신으로 설정한다.
- [0189] 408. 스케줄러 1은 리소스 정보 개인 사본을 업데이트한다.
- [0190] 즉, 스케줄러 1은 리소스 정보 개인 사본(1)을 업데이트한다.
- [0191] 409. 스케줄러 1은 제출 횟수들의 양이 스케줄링 충돌 임계값을 초과하는지를 결정한다. 그렇지 않다면, 단계 410이 수행되고; 만약 그렇다면, 단계 411이 수행된다.
- [0192] 그 제출이 성공한 경우, 그 이유는 복수의 스케줄러가 동일한 호스트 머신에 리소스들을 동시에 제출하려고 시도하는 것일 수 있다. 이 경우, 하나의 스케줄러만이 리소스들을 성공적으로 제출할 수 있고, 다른 스케줄러들은 리소스들을 제출하지 못한다.
- [0193] 스케줄링 충돌 임계값은 실제 상황들에 따라 유연하게 설정될 수 있다. 일부 실시예들에서, 스케줄링 충돌 임계값은, 복수의 VM이 짧은 시간에 동일한 호스트 머신 상에서 생성되어, 그 호스트 머신에 대해 과도하게 큰 압력을 야기하는 경우를 피하기 위해, 3회 내지 8회의 범위로 설정될 수 있다.
- [0194] 410. 스케줄러 1은 호스트 머신의 리소스 데이터를 다시 획득하고, 단계 404를 수행한다.
- [0195] 스케줄링 충돌은 복수의 스케줄러가 동시에 나머지 리소스를 제출하려고 동작할 때 발생할 수 있다. 따라서,

스케줄러는 호스트 머신의 최신 리소스 데이터에 대해 질의할 수 있고, 나머지 리소스들의 차감 및 제출과 같은 동작들의 새로운 라운드를 개시할 수 있다. 동일한 호스트 머신에 대한 스케줄링 충돌들의 양이 임계값을 초과하는 경우, 후보 호스트 머신 리스트에서의 다음 호스트 머신, 즉 차선의 호스트 머신이 선택된다. 이러한 방식으로, 스케줄링 충돌의 비용이 더 낮고, 스케줄링 충돌이 효과적으로 처리된다.

- [0196] 411. 스케줄러 1은 여전히 임의의 선택되지 않은 후보 호스트 머신이 있는지를 결정한다. 그렇다면, 단계 403이 수행되고; 그렇지 않다면, 단계 412가 수행된다.
- [0197] 스케줄러 1은 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하여 후보 호스트 머신 타겟 리스트를 획득하고, 후보 호스트 머신 타겟 리스트에서의 후보 호스트 머신들의 순위들에 따라 호스트 머신을 선택하고, VM의 리소스 요건에 따라 차감 후의 호스트 머신의 나머지 리소스를 계산할 수 있어서, 복수의 스케줄러들이 리소스 스케줄링이 동시에 수행되는 시나리오에서 동일한 호스트 머신을 선택하는 것이 방지될 수 있고, 이는 스케줄링 충돌들을 효과적으로 피할 수 있다.
- [0198] 일부 실시예들에서, 스케줄링 서버는 후보 호스트 머신의 나머지 리소스를 제출하지 못하는 경우, 후보 호스트 머신의 나머지 리소스의 제출 횟수들의 양이 스케줄링 충돌 임계값을 초과하는지를 결정할 수 있다.
- [0199] 스케줄링 서버는 후보 호스트 머신의 최신 리소스 데이터를 획득하고, 후보 호스트 머신의 나머지 리소스의 제출 횟수들의 양이 스케줄링 충돌 임계값을 초과하지 않는다고 결정하는 경우에 후보 호스트 머신에 대응하는 나머지 리소스를 재결정한다.
- [0200] 스케줄링 서버는 후보 호스트 머신 리스트에서의 다음 후보 호스트 머신으로부터, 생성될 VM의 리소스 요건을 차감하여 대응하는 나머지 리소스를 획득하고, 후보 호스트 머신의 나머지 리소스의 제출 횟수들의 양이 스케줄링 충돌 임계값을 초과하는 것으로 결정하는 경우에 대응하는 나머지 리소스를 제출할 수 있다.
- [0201] 412. 스케줄러 1은 선택되지 않은 호스트 머신이 없다고 결정한다.
- [0202] 스케줄러 2는 리소스 정보 개인 사본(2)에 따라 적어도 하나의 후보 호스트 머신을 결정하고, 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신 B를 결정한다. 스케줄러 3은 리소스 정보 개인 사본(3)에 따라 적어도 하나의 후보 호스트 머신을 결정하고, 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신 C를 결정한다. 2개의 스케줄러의 결정 방식들은 스케줄러 1의 결정 방식과 유사하고, 본 명세서에서 설명되지 않는다.
- [0203] 219. 스케줄러 1은 타겟 호스트 머신 A를 결정한 후에 VM1을 생성하라는 요청을 호스트 머신 A에 전송한다.
- [0204] 220. 스케줄러 2는 타겟 호스트 머신 B를 결정한 후에 VM2를 생성하라는 요청을 호스트 머신 B에 전송한다.
- [0205] 221. 스케줄러 3은 타겟 호스트 머신 C를 결정한 후에 VM3을 생성하라는 요청을 호스트 머신 C에 전송한다.
- [0206] 222. 호스트 머신 A는 VM1을 생성하라는 요청을 수신한 후에 VM1을 생성한다.
- [0207] 223. 호스트 머신 B는 VM2를 생성하라는 요청을 수신한 후에 VM2를 생성한다.
- [0208] 224. 호스트 머신 C는 VM3을 생성하라는 요청을 수신한 후에 VM3을 생성한다.
- [0209] 예들이 이하에서 설명을 위해 이용된다. 도 5에 도시된 바와 같이, 스케줄러들이 리소스 스케줄링을 동시에 수행하는 것으로 가정하면, 후보 호스트 머신들은 스케줄러 1에 의해 획득된 후보 호스트 머신 리스트 1에서의 우선순위 값들의 내림차순으로 순차적으로 후보 호스트 머신 A, 후보 호스트 머신 B, 및 후보 호스트 머신 C이고; 후보 호스트 머신들은 스케줄러 2에 의해 획득된 후보 호스트 머신 리스트 2에서의 우선순위 값들의 내림차순으로 순차적으로 후보 호스트 머신 A, 후보 호스트 머신 C, 및 후보 호스트 머신 B이고; 후보 호스트 머신들은 스케줄러 3에 의해 획득된 후보 호스트 머신 리스트 3에서의 우선순위 값들의 내림차순으로 순차적으로 후보 호스트 머신 B, 후보 호스트 머신 A, 및 후보 호스트 머신 C이다.
- [0210] 이 경우에, 스케줄러 1은 후보 호스트 머신 리스트 1로부터 첫 번째로 순위화된 후보 호스트 머신 A를 선택하고, 차감이 수행된 후에 VM1의 리소스 요건에 따라 후보 호스트 머신 A의 나머지 리소스를 계산하고, 그 나머지 리소스를 공통 리소스 정보에 제출하며, 그 제출이 성공한 경우 후보 호스트 머신 A가 타겟 호스트 머신인 것으로 결정될 수 있다. 스케줄러 1은 VM1을 생성하라는 요청을 호스트 머신 A에 전송하고, 호스트 머신 A는 그 요청을 수신한 후에 VM1을 생성한다.
- [0211] 스케줄러 2는 후보 호스트 머신 리스트 2로부터 첫 번째로 순위화된 후보 호스트 머신 A를 선택하고, 차감이 수행된 후에 VM2의 리소스 요건에 따라 후보 호스트 머신 A의 나머지 리소스를 계산하고, 그 나머지 리소스를 공

통 리소스 정보에 제출한다. 그 제출이 실패하고 제출 횟수들의 양이 임계값에 도달하면, 스케줄러 2는 후보 호스트 머신 리스트 2로부터 두 번째로 순위화된 후보 호스트 머신 C를 선택할 필요가 있고, 차감이 수행된 후에 VM2의 리소스 요건에 따라 후보 호스트 머신 C의 나머지 리소스를 계산하고, 그 나머지 리소스를 공통 리소스 정보에 제출하며, 그 제출이 성공한 경우 후보 호스트 머신 C가 타겟 호스트 머신인 것으로 결정될 수 있다. 스케줄러 2는 VM2를 생성하라는 요청을 호스트 머신 C에 전송하고, 호스트 머신 C는 그 요청을 수신한 후에 VM2를 생성한다.

[0212] 스케줄러 3은 후보 호스트 머신 리스트 3으로부터 첫 번째로 순위화된 후보 호스트 머신 B를 선택하고, 차감이 수행된 후에 VM3의 리소스 요건에 따라 후보 호스트 머신 A의 나머지 리소스를 계산하고, 그 나머지 리소스를 공통 리소스 정보에 제출하며, 그 제출이 성공한 경우 후보 호스트 머신 B가 타겟 호스트 머신인 것으로 결정될 수 있다. 스케줄러 3은 VM3을 생성하라는 요청을 호스트 머신 B에 전송하고, 호스트 머신 B는 그 요청을 수신한 후에 VM3을 생성한다.

[0213] 도 5는 단지 예이다. 도 5에 설명된 리소스 스케줄링 시나리오는 본 출원의 실시예들의 기술적 해결책들을 더 명확하게 설명하기 위한 것이고, 본 출원의 실시예들에서 제공되는 기술적 해결책들에 대한 제한을 구성하지 않는다. 예를 들어, 도 5는 스케줄러 4를 더 포함할 수 있고, 후보 호스트 머신들은 스케줄러 4에 의해 획득된 후보 호스트 머신 리스트 4에서의 우선순위 값들의 내림차순으로 순차적으로 후보 호스트 머신 Y, 후보 호스트 머신 X, 및 후보 호스트 머신 D이다. 스케줄러 4는 후보 호스트 머신 리스트 4로부터 첫 번째로 순위화된 후보 호스트 머신 B를 선택하고, 차감이 수행된 후에 VM3의 리소스 요건에 따라 후보 호스트 머신 Y의 나머지 리소스를 계산하고, 그 나머지 리소스를 공통 리소스 정보에 제출하며, 그 제출이 성공한 경우 스케줄러 4가 후보 호스트 머신 Y 상에 VM4를 생성할 수 있다고 결정될 수 있다.

[0214] 전술한 것으로부터 알 수 있듯이, 이 실시예에서, 복수의 스케줄러들은 메시지 큐 내의 메시지로부터 VM 생성 요청을 동시에 추출하고, VM 생성 요청에서 운반된 VM 정보에 따라 각각의 VM 생성 요청에 대한 적절한 타겟 호스트 머신을 선택하고, 타겟 호스트 머신 상에 대응하는 VM을 생성할 수 있다. 공통 리소스 정보 및 리소스 정보 개인 사본을 설정하는 것을 포함하여, 리소스 정보에 대한 버퍼링 메커니즘을 설정함으로써, 복수의 스케줄러들은 동시에 리소스 스케줄링을 수행할 수 있고, 이는 리소스 스케줄링의 동시성 레벨 및 처리율을 효과적으로 향상시키고, 스케줄링 레이턴시를 감소시키고, 전체 시스템의 확장성을 향상시킬 수 있다.

[0215] 본 출원의 실시예들에서 제공되는 리소스 스케줄링 방법을 더 잘 구현하기 위해, 본 출원의 실시예는 전술한 리소스 스케줄링 방법에 기반한 장치를 추가로 제공한다. 용어들은 전술한 리소스 스케줄링 방법에서의 용어들과 동일한 의미를 가지며, 특정 구현 상세들의 경우, 방법 실시예들에서의 설명이 참조될 수 있다.

[0216] 도 8은 본 출원의 실시예에 따른 리소스 스케줄링 장치의 개략적인 구조도이다. 리소스 스케줄링 장치는 제1 획득 유닛(501), 제2 획득 유닛(502), 업데이트 유닛(503), 제3 획득 유닛(504), 및 생성 유닛(505)을 포함할 수 있다.

[0217] 제1 획득 유닛(501)은 생성될 VM에 대응하는 VM 정보를 획득하도록 구성된다.

[0218] 예를 들어, 제1 획득 유닛(501)은 클라우드 컴퓨팅 시스템 내의 미리 설정된 메시지 큐로부터, 생성될 VM에 대응하는 VM 정보를 획득할 수 있다. VM은 소프트웨어를 통해 시뮬레이션되고 하드웨어 시스템 기능을 갖고 격리된 환경에서 실행되는 전체 컴퓨터 시스템이다.

[0219] 메시지 큐 내의 메시지는 비동기 작업일 수 있고, 메시지 큐 내의 메시지는 메시지 큐에서 흐르며, 클라우드 컴퓨팅 시스템 내의 각각의 구성요소는 생성자 또는 소비자의 아이덴티티를 갖는다. 생성자는 메시지를 메시지 큐에 전송하고, 소비자는 메시지 큐로부터 메시지를 수신한다. 예를 들어, 수신기는 클라우드 컴퓨팅 시스템에서 생성자로서 이용된다. 단말기에 의해 전송된 VM 정보를 운반하는 VM 생성 요청을 수신한 후, 수신기는 VM 정보를 메시지 내에 캡슐화하고, 메시지를 메시지 큐에 저장한다. VM 정보는 제1 획득 유닛(501)이 리소스 스케줄링을 수행하기를 대기하기 위해 메시지 큐에 큐잉된다. 소비자로서, 제1 획득 유닛(501)은 메시지 큐로부터 VM 정보를 추출하고, VM 정보에 대응하는 생성될 VM에 대한 호스트 머신을 선택한다.

[0220] VM 정보는 생성될 VM에 의해 요구되는 프로세서 정보, 내부 메모리 정보, 및 디스크 정보, 및 솔리드 스테이트 드라이브, 클라우드 디스크, 개인 네트워크, 네트워크 어댑터, 라우터, 네트워크 구조, 데이터 센터, 랙, 호스트 머신 유형, 호스트 머신이 속하는 리소스 풀, 및 VM 유형에 대한 생성될 VM의 리소스 요건들 중 적어도 하나를 포함할 수 있다. VM 정보의 유형이 실제 요건들에 따라 유연하게 설정될 수 있고, 특정 콘텐츠로 본 명세서에서 제한되지 않는다는 점이 이해될 수 있다.

- [0221] 제2 획득 유닛(502)은 공통 리소스 정보를 획득하도록 구성된다.
- [0222] 공통 리소스 정보는 클라우드 컴퓨팅 시스템에 의해 저장되고 유지되는 전체 데이터 센터의 리소스 데이터이다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함한다. 공통 리소스 정보는 리소스 관리자에 의해 유지될 수 있거나, 또는 클라우드 컴퓨팅 시스템 내의 다른 디바이스에 의해 유지될 수 있다는 점이 이해될 수 있다. 그 후, 제2 획득 유닛(502)은 공통 리소스 정보를 유지하는 디바이스로부터 공통 리소스 정보를 획득한다. 예를 들어, 공통 리소스 정보가 클라우드 컴퓨팅 시스템에서 리소스 관리자에 의해 유지되는 경우, 제2 획득 유닛(502)은 리소스 관리자로부터 공통 리소스 정보를 획득할 수 있다. 설명의 용이함을 위해, 본 출원의 실시예들에서, 공통 리소스 정보를 유지하는 리소스 관리자가 설명을 위한 예로서 이용된다.
- [0223] 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서 전역적으로 고유하고, 논리 데이터 테이블의 형태로 유지될 수 있다. 예를 들어, 데이터 테이블 내의 각각의 행은 호스트 머신을 나타내고, 데이터 테이블 내의 열들은 호스트 머신에 포함된 상이한 유형들의 리소스들을 나타낸다. 상이한 유형들의 리소스들은 호스트 머신의 인터넷 프로토콜(IP) 어드레스, 이용가능한 중앙 처리 유닛(CPU), 이용가능한 내부 메모리, 및 이용가능한 디스크를 포함할 수 있다. 유지 프로세스에서, 추가, 판독, 업데이트, 또는 삭제와 같은 동작이 공통 리소스 정보에 대해 수행될 수 있다. 이러한 동작은 데이터 센터에서의 호스트 머신 리소스의 온라인 획득, 변경, 또는 오프라인 획득과 같은 프로세스에 대응한다.
- [0224] 호스트 머신은 VM 소프트웨어를 설치하도록 구성된 물리적 컴퓨터이다. VM 소프트웨어를 통해 하나 이상의 VM이 하나의 호스트 머신 상에서 생성될 수 있다.
- [0225] 일부 실시예들에서, 호스트 머신 정보는 호스트 머신 식별자, IP 어드레스, 머신 유형, 운영 체제 이름, 커널 버전, CPU 유형, 가상화 방식, 논리적 영역 번호, 인터넷 데이터 센터 번호, 호스트 머신의 상태, 클라우드 디스크를 지원할지 여부, 솔리드 스테이트 드라이브(SSD)를 지원할지 여부, 네트워크 버전, CPU 노드 리소스 할당량, 이용가능한 내부 메모리 리소스 및 이용가능한 디스크 리소스를 포함할 수 있고, 특정 콘텐츠로 본 명세서에서 제한되지는 않는다.
- [0226] 호스트 머신이 클라우드 컴퓨팅 시스템에 진입할 때, 필요한 초기화가 호스트 머신 도입 프로세스를 통해 수행될 필요가 있다. 호스트 머신은 그 호스트 머신 정보를 리소스 관리자에 보고하고, 리소스 관리자는 호스트 머신 정보를 공통 리소스 정보에 저장할 수 있다. 호스트 머신은 호스트 머신 정보를 리소스 스케줄링 장치에 보고할 수 있고, 리소스 스케줄링 장치는 호스트 머신 정보를 미리 설정된 리소스 정보 개인 사본에 저장할 수 있다. 대안적으로, 리소스 스케줄링 장치는 실시간으로 또는 미리 설정된 간격으로 리소스 관리자에게 정보 획득 요청을 전송하고, 정보 획득 요청에 기반하여 리소스 관리자에 의해 피드백되는 공통 리소스 정보를 수신하고, 공통 리소스 정보에 따라 리소스 정보 개인 사본을 업데이트한다. 대안적으로, 리소스 관리자는 공통 리소스 정보가 업데이트되는 경우에, 업데이트된 호스트 머신 정보를 리소스 스케줄링 장치에 능동적으로 전송한다.
- [0227] 업데이트 유닛(503)은 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하도록 구성된다.
- [0228] 리소스 스케줄링 장치는 리소스 스케줄링 장치의 내부 메모리 내의 공통 리소스 정보인 리소스 정보 개인 사본을 버퍼의 형태로 유지한다. 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 미리 설정된 호스트 머신은 클라우드 컴퓨팅 시스템 내의 호스트 머신들의 일부 또는 모든 호스트 머신들일 수 있다. 더 나은 리소스 스케줄링 결과를 달성하기 위해, 리소스 스케줄링을 수행하기 전에, 업데이트 유닛(503)은 리소스 정보 개인 사본을 공통 리소스 정보와 동기화할 필요가 있고, 공통 리소스 정보를 획득하고 공통 리소스 정보에 따라 리소스 정보 개인 사본을 업데이트하여, 리소스 정보 개인 사본과 공통 리소스 정보 사이의 일관성을 보장할 수 있다.
- [0229] 업데이트 시간을 감소시키기 위해, 업데이트 유닛(503)은 공통 리소스 정보 및 VM 정보에 따라 리소스 정보 개인 사본을 업데이트할 수 있다. 일부 실시예들에서, 도 9에 도시된 바와 같이, 업데이트 유닛(503)은 공통 리소스 정보로부터 VM 정보를 추출시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 타겟 호스트 머신 정보를 미리 설정된 리소스 정보 개인 사본에 추가하도록 구성된 업데이트 서브유닛(5031)을 포함한다.
- [0230] 일부 실시예들에서, 업데이트 서브유닛(5031)은 구체적으로, 리소스 정보 개인 사본이 처음으로 업데이트되는 경우에 공통 리소스 정보로부터 VM 정보를 추출시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행

하고; 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사하고; 현재의 타임스탬프를 획득하고, 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로 설정하고; 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장하도록 구성된다.

- [0231] 구체적으로, 먼저, 업데이트 서버유닛(5031)은 리소스 정보 개인 사본이 처음으로 업데이트되는지를 결정하고, 예를 들어, 리소스 정보 개인 사본이 비어 있는지를 결정할 수 있다. 그렇다면, 리소스 정보 개인 사본이 처음으로 업데이트되는 것으로 결정되고; 그렇지 않다면, 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 것으로 결정된다. 리소스 정보 개인 사본이 처음으로 업데이트되는 경우, 현재 리소스 정보 개인 사본이 비어 있고, 이 경우, 전체 업데이트가 수행될 필요가 있으며, 즉, 공통 리소스 정보로부터 VM 정보를 충족시키는 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다. 공통 리소스 정보는 대안적으로 리소스 정보 개인 사본으로서 직접 전체 복사될 수 있다.
- [0232] 타겟 호스트 머신 정보가 리소스 정보 개인 사본에 복사된 후, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프가 기록될 필요가 있다. 일부 실시예들에서, 업데이트 서버유닛(5031)은 현재의 타임스탬프를 획득하고, 현재의 타임스탬프를 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프로서 설정한다. 예를 들어, 현재의 타임스탬프는 2017년 9월 15일 오전 11:00이고, 이 경우, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프 둘 다는 2017년 9월 15일 오전 11:00이다. 그 후, 현재 업데이트를 완료하기 위해, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프가 리소스 정보 개인 사본에 저장된다.
- [0233] 일부 실시예들에서, 업데이트 서버유닛(5031)은 구체적으로, 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 경우에 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득하고; 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정하고; 시간 차이가 전체 업데이트 임계값보다 큰 경우에 공통 리소스 정보로부터 VM 정보를 충족시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고; 제1 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사하고; 현재의 타임스탬프를 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정하고; 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장하도록 구성된다.
- [0234] 구체적으로, 업데이트 서버유닛(5031)은 리소스 정보 개인 사본이 처음으로 업데이트되지 않는 경우에 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득하고, 여기서 이전의 전체 업데이트 타임스탬프는 현재의 타임스탬프에 대해 이전에 기록된 전체 업데이트 타임스탬프이다.
- [0235] 그 후, 업데이트 서버유닛(5031)은 시간 차이가 미리 설정된 전체 업데이트 임계값보다 크지를 결정한다. 전체 업데이트 임계값은 실제 요건들에 따라 유연하게 설정될 수 있고, 특정 값으로 본 명세서에서 제한되지 않는다. 예를 들어, 비교적 낮은 전체 업데이트 빈도를 달성하기 위해, 전체 업데이트 임계값은 비교적 큰 값으로 설정될 수 있다.
- [0236] 시간 차이가 전체 업데이트 임계값보다 큰 경우, 전체 업데이트가 수행되고, 즉, 공통 리소스 정보로부터 VM 정보를 충족시키는 제1 타겟 호스트 머신 정보를 획득하기 위한 스크리닝을 수행하고, 제1 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 복사한다. 현재의 공통 리소스 정보는 대안적으로 리소스 정보 개인 사본으로서 전체 복사될 수 있다.
- [0237] 그 후, 업데이트 서버유닛(5031)은 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프를 정확하게 기록하기 위해 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프를 기록한다. 일부 실시예들에서, 현재의 타임스탬프는 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프로 설정될 수 있다. 예를 들어, 현재의 타임스탬프는 2017년 9월 16일 오전 8:00이고, 이 경우, 전체 업데이트 타임스탬프 및 증분 업데이트 타임스탬프 둘 다는 2017년 9월 16일 오전 8:00이다. 그 후, 제1 전체 업데이트 타임스탬프 및 제1 증분 업데이트 타임스탬프는 리소스 정보 개인 사본에 저장되어 현재 업데이트를 완료한다.
- [0238] 또한, 업데이트 서버유닛(5031)은 추가로 구체적으로, 시간 차이가 전체 업데이트 임계값보다 작거나 같은 경우에, 공통 리소스 정보로부터 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 업데이트된 호스트 머신 정보를 획득하고; 호스트 머신 정보로부터 VM 정보를 충족시키는 제2 타겟 호스트 머신 정보를 획득하고; 제2 타겟 호스트 머신 정보를 리소스 정보 개인 사본에 추가하고; 현재의 타임스탬프를 제2 증분 업데이트 타임스탬프로 설정하고; 제2 증분 업데이트 타임스탬프를 리소스 정보 개인 사본에 저장하도록 구성된다.
- [0239] 클라우드 컴퓨팅 시스템에서의 호스트 머신들의 스케일은 비교적 크고, 예를 들어, 호스트 머신들의 스케일은 10000의 자릿수를 가지며, 리소스 스케줄링 장치들의 스케일은 100의 자릿수를 갖는다. 따라서, 전체 업데이트

가 수행되는 경우, 즉, 공통 리소스 정보가 리소스 정보 개인 사본이 업데이트될 때마다 리소스 정보 개인 사본으로서 전체 복사되는 경우, 많은 시간이 소비된다. 업데이트에 의해 소비되는 시간을 감소시키기 위해, 증분 업데이트가 리소스 정보에 대해 수행될 수 있으며, 즉, 공통 리소스 정보의 일부 리소스 정보가 리소스 정보 개인 사본에 추가된다.

- [0240] 구체적으로, 공통 리소스 정보 내의 호스트 머신 정보가 업데이트될 때마다, 정확한 업데이트 시간을 획득하기 위해 업데이트 타임스탬프가 기록된다. 업데이트 서버유닛(5031)이 현재의 타임스탬프와 이전의 전체 업데이트 타임스탬프 사이의 시간 차이를 획득한 후에, 시간 차이가 전체 업데이트 임계값보다 작거나 같다면, 업데이트 서버유닛(5031)은 공통 리소스 정보로부터 업데이트된 호스트 머신 정보를 획득하고, 호스트 머신 정보의 업데이트된 업데이트 타임스탬프들을 획득하고, 이어서 업데이트된 호스트 머신 정보로부터 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 호스트 머신 정보를 획득하기 위한 스크리닝을 수행한다. 즉, 공통 리소스 정보는 이전의 증분 업데이트 타임스탬프에 따라 질의되어, 업데이트 타임스탬프들이 이전의 증분 업데이트 타임스탬프보다 크거나 같은 모든 호스트 머신 정보를 획득한다. 이전의 증분 업데이트 타임스탬프는 현재의 타임스탬프에 대한 마지막 증분 업데이트 타임스탬프이다.
- [0241] 업데이트되고 업데이트 타임스탬프가 이전의 증분 업데이트 타임스탬프보다 큰 호스트 머신 정보가 공통 리소스 정보로부터 획득된 후에, 호스트 머신 정보가 스크리닝된다. VM 정보를 충족시키는 제2 타겟 호스트 머신 정보는 호스트 머신 정보로부터 획득되고, 리소스 정보 개인 사본에 추가된다.
- [0242] 그 후, 업데이트 서버유닛(5031)은 증분 업데이트 타임스탬프를 정확하게 기록하기 위해, 제2 증분 업데이트 타임스탬프를 기록한다. 즉, 현재의 타임스탬프는 제2 증분 업데이트 타임스탬프로 설정되고, 제2 증분 업데이트 타임스탬프는 리소스 정보 개인 사본에 저장되어 현재 업데이트를 완료한다. 리소스 정보 개인 사본은 실질적으로 내부 메모리에서 실행되는 버퍼이다. 대부분의 경우들에서, 업데이트 서버유닛(5031)은 단지 증분 업데이트를 수행할 필요가 있다. 업데이트의 데이터 볼륨 및 소비된 시간은 둘 다 더 적다. 일부 실시예들에서, 현재 증분 업데이트 이전에 오프라인이 되는 호스트 머신은 리소스 정보 개인 사본으로부터 삭제된다.
- [0243] 제1 타겟 호스트 머신 정보, 제2 타겟 호스트 머신 정보, 제1 전체 업데이트 타임스탬프, 제1 증분 업데이트 타임스탬프, 및 제2 증분 업데이트 타임스탬프에서의 "제1" 및 "제2"는 타겟 호스트 머신 정보, 증분 업데이트 타임스탬프들, 또는 전체 업데이트 타임스탬프들을 구별하는데 이용되는 것에 불과하고, 그 중요도의 제한을 의미하는 것은 아니다.
- [0244] 제3 획득 유닛(504)은, 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하도록 구성된다.
- [0245] 리소스 정보 개인 사본을 업데이트한 후에, 제3 획득 유닛(504)은 리소스 정보 개인 사본에 기반하여 생성될 VM에 대한 호스트 머신들을 스크리닝한다. 제3 획득 유닛(504)은 리소스 정보 개인 사본 내의 호스트 머신 정보 및 VM 정보에서의 리소스 요건에 따라 리소스 정보 개인 사본 내의 호스트 머신들의 세트를 필터링하고, 자격을 갖는 호스트 머신을 예비하고, 자격을 갖지 않는 호스트 머신을 제거하여, 적어도 하나의 후보 호스트 머신을 획득할 수 있다.
- [0246] 일부 실시예들에서, 제3 획득 유닛(504)은 호스트 머신 정보에 기반하여 확립된 직렬 접속된 필터들 및 VM 정보에 따라, 복수의 호스트 머신들에 대한 대응하는 필터링을 수행하도록 각각의 필터를 제어하여, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득할 수 있다.
- [0247] 호스트 머신 정보에 따라 직렬 필터링 모델이 확립될 수 있다. 직렬 필터링 모델은 복수의 직렬 접속된 필터들을 포함한다. 각각의 필터는 하나의 필터링 기능을 담당하고, VM 정보에 따라 대응하는 판단을 행하고, 이어서 대응하는 필터링을 수행한다. 예를 들어, 제1 직렬 접속된 필터는, VM에 의해 요구되는 CPU 리소스, 내부 메모리 리소스, 및 디스크 리소스와 같은 리소스 요건에 따라 리소스 요건을 충족시키지 못하는 호스트 머신을 필터링하는 것을 담당하고, 리소스 요건을 충족시키는 호스트 머신만을 예비한다. 제2 직렬 접속된 필터는 SSD를 지원하거나 지원하지 않는 호스트 머신을 필터링하는 것을 담당한다. 호스트 머신이 리소스 요건에서 SSD를 지원하도록 요구되는 경우, 제2 필터는 리소스 요건에 따라 SSD를 지원하지 않는 호스트 머신을 필터링하고, SSD를 지원하는 호스트 머신을 예비한다.
- [0248] 다른 필터들의 기능들의 경우, 다른 필터들이 다른 필터링 기능들을 담당하는 것을 제외하고는 이해를 위해 이러한 필터를 참조할 수 있다는 점이 이해될 수 있다. 특정 필터링 모델에서의 필터들의 기능들은 생성될 VM의 리소스 요건에 따라 설정될 수 있고, 특정 콘텐츠로 본 명세서에서 제한되지는 않는다.

- [0249] 생성 유닛(505)은 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하도록 구성된다.
- [0250] 적어도 하나의 후보 호스트 머신을 획득한 후에, 생성 유닛(505)은 적어도 하나의 후보 호스트 머신에서 타겟 호스트 머신을 결정할 수 있다. 타겟 호스트 머신은 랜덤하게 선택될 수 있다. 물론, 가장 적절한 타겟 호스트 머신을 선택하기 위해, 복수의 호스트 머신들이 적어도 하나의 후보 호스트 머신을 획득하도록 필터링된 후에, 각각의 차원들에서의 각각의 후보 호스트 머신의 우선순위 값들이 결정될 수 있고, 적어도 하나의 후보 호스트 머신은 우선순위 값들에 따라 순위화된다.
- [0251] 일부 실시예들에서, 도 10에 도시된 바와 같이, 생성 유닛(505)은,
- [0252] 각각의 차원들에서 적어도 하나의 후보 호스트 머신의 각각의 호스트 머신의 우선순위 값들을 결정하도록 구성된 결정 서브유닛(5051);
- [0253] 차원들에서 우선순위 값들의 내림차순으로 순위화를 수행하여 후보 호스트 머신 리스트를 생성하도록 구성된 생성 서브유닛(5052);
- [0254] 후보 호스트 머신 리스트에서의 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하도록 구성된 차감 서브유닛(5053) - 리소스 요건은 VM에 대응하는 VM 정보에 포함됨 -;
- [0255] 나머지 리소스들을 공통 리소스 정보에 제출하도록 구성된 제출 서브유닛(5054); 및
- [0256] 하나의 후보 호스트 머신에 대응하는 나머지 리소스가 성공적으로 제출되는 경우, 그 제출이 성공한 후보 호스트 머신을 타겟 호스트 머신으로 설정하고, 타겟 호스트 머신 상에 VM을 생성하도록 구성된 생성 서브유닛(5055)
- [0257] 을 포함한다.
- [0258] 구체적으로, 먼저, 결정 서브유닛(5051)은 차원들에서 각각의 호스트 머신의 우선순위 값들을 계산한다. 차원들에서의 우선순위들은 개인 네트워크 우선순위(vpcPrio), 동시 작업 우선순위(taskPrio), 이력 에러 우선순위(errorPrio), 반-친화도 우선순위(affinityPrio), 유희 우선순위(freePrio), 이미지 캐시 우선순위(imgPrio), 단편화 비율 우선순위(fragmentPrio) 등을 포함할 수 있다.
- [0259] 개인 네트워크 우선순위의 값의 경우, vpcPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 개인 네트워크 우선순위의 값은 각각의 호스트 머신에 대해 결정될 수 있다. 예를 들어, 0은 개인 네트워크를 지원하는 호스트 머신의 vpcPrio에 할당될 수 있고, 1은 개인 네트워크를 지원하지 않는 호스트 머신의 vpcPrio에 할당될 수 있고; 대안적으로, 1은 개인 네트워크를 지원하는 호스트 머신의 vpcPrio에 할당될 수 있고, 0은 개인 네트워크를 지원하지 않는 호스트 머신의 vpcPrio에 할당될 수 있고, 다른 것들도 그런 식이다. 설명의 용이함을 위해, 본 출원의 실시예들에서, 개인 네트워크를 지원하는 호스트 머신의 vpcPrio에 0이 할당되고, 개인 네트워크를 지원하지 않는 호스트 머신의 vpcPrio에 1이 할당되는 예가 설명을 위해 이용된다.
- [0260] 개인 네트워크의 IP 어드레스는 물리적 네트워크 토폴로지에 의존하지 않고, 미래에 이동하기가 더 용이하다. 개인 네트워크 우선순위의 효과는 개인 네트워크를 지원하는 호스트 머신 상에 VM을 우선적으로 배치하는 것이어서, 미래의 이동이 더 편리하고 유연하게 된다.
- [0261] 동시 작업 우선순위의 경우, taskPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 각각의 호스트 머신에 대해 동시 작업 우선순위가 결정될 수 있다. 호스트 머신 상의 현재 작업들의 양이 결정되고, 작업들의 양이 미리 설정된 임계값 미만인 경우, 0이 taskPrio에 할당되고; 그렇지 않으면, 현재 작업들의 양이 taskPrio에 할당된다. 예를 들어, 미리 설정된 임계값은 디폴트로 3이다. 물론, 미리 설정된 임계값은 실제 요건들에 따라 유연하게 구성될 수 있다.
- [0262] 호스트 머신 상의 동시 작업들은 동시에 수행될 수 있고, 동시성은 과도하게 높다. 따라서, VM의 생성은 느리거나 심지어 실패할 수 있다. 동시 작업 우선순위의 효과는 보다 적은 양의 동시 작업들을 갖는 호스트 머신 상에 VM을 우선적으로 배치하여 VM을 생성하는 성공률 및 실시간 성능을 보장하는 것이다.
- [0263] 이력 에러 우선순위의 경우, errorPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 이력 에러 우선순위는 각각의 호스트 머신에 대해 계산될 수 있다. 마지막 기간(예를 들어, 디폴트로 24시간) 내의 호스트 머신의 이력 에러들의 양이 결정된다. 이력 에러들의 양이 에러 임계값보다 작으면, 0이 errorPrio에 할당되고; 그렇지

많은 경우, 이력 에러들의 양이 errorPrio에 할당된다. 예를 들어, 에러 임계값은 디폴트로 5일 수 있고, 에러 임계값은 실제 요건들에 따라 유연하게 구성될 수 있다.

- [0264] 짧은 기간 내에 복수의 에러가 발생하는 호스트 머신은 어떤 유형의 결함을 겪을 수 있고, 호스트 머신이 처리되기 전에, VM이 호스트 머신 상에 계속 배치되는 경우, VM을 생성하는 작업은 실패할 수 있고, 성공률은 감소된다. 이력 에러 우선순위의 효과는 과도한 에러가 발생하지 않는 호스트 머신 상에 VM을 우선적으로 배치하는 것에 의해, VM을 생성하는 성공률을 향상시키는 것이다.
- [0265] 반-친화도 우선순위의 경우, affinityPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 각각의 호스트 머신에 대해 반-친화도 우선순위가 계산될 수 있다. 호스트 머신 상의 VM 사용자의 기존의 VM들의 양이 결정되고, VM들의 양이 친화도 임계값보다 작으면, 0이 affinityPrio에 할당되고; 그렇지 않으면, 기존의 VM들의 양이 affinityPrio에 할당된다. 예를 들어, 친화도 임계값은 3이고, 물론, 친화도 임계값은 실제 요건들에 따라 유연하게 구성될 수 있다.
- [0266] 대규모 분산 시스템에서는, 에러들을 완전히 방지하는 것은 거의 불가능하다. 사용자의 VM들이 하나 이상의 호스트 머신 상에서 클러스터링되는 경우, 호스트 머신들이 결함이 있을 때 호스트 머신들 상의 VM들은 모두 영향을 받고, 사용자의 전체 애플리케이션의 이용가능성에 심각하게 영향을 미친다. 반-친화도 우선순위의 효과는 동일한 사용자의 VM들이 상이한 호스트 머신들 상에 널리 분산되게 하여, 분산된 재난 복구를 수행하게 함으로써, 단일 호스트 머신의 결함이 전체 애플리케이션의 이용가능성에 비교적 제한된 영향을 미치게 하고, 이에 의해 사용자의 애플리케이션의 안정성 및 이용가능성을 향상시키는 것이다.
- [0267] 유휴 우선순위의 경우, freePrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 예를 들어, 각각의 호스트 머신이 완전히 유휴인지가 결정될 수 있다. 각각의 호스트 머신이 완전히 유휴인 경우, 0이 freePrio에 할당되고; 그렇지 않으면, 1이 freePrio에 할당된다. 대안적으로, 각각의 호스트 머신이 완전히 유휴인 경우, 1이 freePrio에 할당되고; 그렇지 않으면, 0이 freePrio에 할당되며, 다른 것들도 그런 식이다. 설명의 용이함을 위해, 본 출원의 실시예들에서, 각각의 호스트 머신이 완전히 유휴인 경우, 0이 freePrio에 할당되고; 그렇지 않으면, 1이 freePrio에 할당되는 예가 설명을 위해 이용된다.
- [0268] 유휴 우선순위의 효과는 유휴 호스트 머신을 보호하고, 비-유휴 호스트 머신 상에 VM을 우선적으로 배치하여 비-유휴 호스트 머신 상의 리소스들을 완전히 이용함으로써, 리소스 이용을 향상시키는 것이다.
- [0269] 이미지 캐시 우선순위의 경우, imgPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 각각의 호스트 머신이 VM에 의해 요구되는 이미지 파일을 갖는지를 결정하기 위해 후보 호스트 머신 리스트가 순회될 수 있다. 예를 들어, 각각의 호스트 머신이 이미지 파일을 갖는다면, 0이 imgPrio에 할당되고; 그렇지 않으면, 1이 imgPrio에 할당된다.
- [0270] 이미지 캐시 우선순위의 효과는 VM 이미지를 저장한 호스트 머신 상에 VM을 우선적으로 배치하는 것이고, 따라서 VM이 실제로 생성될 때 이미지 다운로드의 단계가 생략되어, 생성 시간을 감소시키고, VM 생성 프로세스의 실시간 성능을 향상시킬 수 있다.
- [0271] 단편화 비율 우선순위의 경우, fragmentPrio의 더 작은 값은 더 높은 우선순위를 나타낸다. 단편화 비율 우선순위는 각각의 호스트 머신에 대해 계산될 수 있다. 우선순위는 2개의 차원, 즉 CPU 및 내부 메모리에서의 리소스들에 따라 계산된다. 디폴트 경우에, 단편화 비율 우선순위는 내부 메모리의 단편화 비율로 곱해진 CPU의 단편화 비율과 동일하다.
- [0272] 단편화 비율 우선순위의 효과는 호스트 머신의 리소스들을 완전히 이용하여 리소스 단편화 비율을 감소시키고, 또한 리소스 이용을 향상시키고 동작 비용들을 감소시키는 것이다.
- [0273] 결정 서버유닛(5051)이 차원들에서 각각의 후보 호스트 머신의 우선순위 값들을 결정한 후에, 생성 서버유닛(5052)은 차원들에서의 우선순위 값들의 내림차순으로 호스트 머신들을 순위화하여 후보 호스트 머신 리스트를 획득하고, 생성 서버유닛(5055)은 타겟 호스트 머신으로서 후보 호스트 머신 리스트에서의 가장 적절한 후보 호스트 머신을 결정한다. 특정 결정 프로세스는 다음과 같을 수 있다:
- [0274] 생성 서버유닛(5052)은 차원들의 우선순위들의 내림차순에 기반하여 동일한 차원에서의 우선순위 값들의 내림차순으로 호스트 머신들을 순위화하고, 다음 차원에서의 각각의 우선순위 값들에 따라 하나의 차원에서 동일한 우선순위 값을 갖는 호스트 머신들을 다시 순위화하여, 호스트 머신들이 우선순위들의 내림차순으로 순위화되는 후보 호스트 머신 리스트를 획득할 수 있다.

- [0275] 후보 호스트 머신들이 순위화된 후에, 첫 번째로 순위화된 후보 호스트 머신이 최적이다. 다른 생성 작업이 최적의 후보 호스트 머신에 존재할 수 있다. 따라서, 생성의 성공이 반드시 보장될 수 없고, 타겟 호스트 머신은 리소스들의 차감을 시뮬레이션하는 방식으로 결정될 필요가 있다.
- [0276] 타겟 호스트 머신을 결정하는 프로세스에서, 먼저, 차감 서브유닛(5053)은 후보 호스트 머신 리스트에서의 순위화된 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여 각각의 후보 호스트 머신에 대응하는 나머지 리소스를 획득하고, 이에 의해 호스트 머신에 대한 리소스들의 차감 시뮬레이션을 완료한다. VM의 리소스 요건은 VM에 대응하는 VM 정보에 포함된다. 즉, 생성될 VM의 리소스 요건은 순위화된 후보 호스트 머신들의 내림차순으로 첫 번째로 순위화된 후보 호스트 머신으로부터 시작하여 후보 호스트 머신들로부터 하나씩 차감된다. 리소스 요건이 후보 호스트 머신으로부터 성공적으로 차감될 때, 그 차감이 성공한 후보 호스트 머신이 타겟 호스트 머신으로서 이용된다.
- [0277] 순위는 미리 설정될 수 있다. 예를 들어, 디폴트 순위 표준 및 우선순위들의 내림차순은 개인 네트워크 우선순위(vpcPrio), 동시 작업 우선순위(taskPrio), 이력 에러 우선순위(errorPrio), 반-친화도 우선순위(affinityPrio), 유휴 우선순위(freePrio), 이미지 캐시 우선순위(imgPrio), 및 단편화 비율 우선순위(fragmentPrio)일 수 있다. 순위 표준은 또한 디폴트 표준 범위에서 선택될 수 있고, 그 순서는 실제 요건들에 따라 유연하게 조정되고 구성될 수 있다.
- [0278] 다음으로, 후보 호스트 머신의 나머지 리소스를 획득한 후에, 제출 서브유닛(5054)은 나머지 리소스를 공통 리소스 정보에 제출하고, 하나의 후보 호스트 머신에 대응하는 나머지 리소스가 성공적으로 제출되는 경우, 그 제출이 성공한 후보 호스트 머신이 타겟 호스트 머신으로 설정된다. 대안적으로, 후보 호스트 머신의 나머지 리소스는 각각의 호스트 머신의 나머지 리소스 데이터를 유지하는데 이용되는 데이터베이스에 기입된다. 기입이 성공할 때, 그 기입이 성공한 후보 호스트 머신이 타겟 호스트 머신으로서 이용된다.
- [0279] 타겟 호스트 머신을 결정한 후에, 생성 서브유닛(5055)은 타겟 호스트 머신 상에 VM을 생성할 수 있다. 일부 실시예들에서, VM의 생성이 타겟 호스트 머신의 리소스를 소비할 필요가 있기 때문에, 리소스 정보 개인 사본 내의 호스트 머신 정보는 업데이트될 수 있고, 클라우드 컴퓨팅 시스템은 공통 리소스 정보를 업데이트하도록 추가로 지시받을 수 있다.
- [0280] 후보 호스트 머신에 대응하는 나머지 리소스가 제출되지 못하는 경우, 이것은 스케줄링 충돌이 아마 발생하는 것, 즉, 복수의 리소스 스케줄링 프로세스들이 동일한 호스트 머신을 동시에 선택한다는 것을 나타낸다. 이 경우, 제출 횟수들의 양이 양 임계값을 초과하는지가 결정된다. 그렇지 않다면, 후보 호스트 머신의 호스트 머신 정보가 다시 획득되고, 생성될 VM의 리소스 요건이 차감되어, 나머지 리소스를 다시 획득한다. 이어서, 다시 획득된 나머지 리소스가 공통 리소스 정보에 다시 제출된다. 제출 횟수들의 양이 양 임계값을 초과하는 경우, 리소스의 차감 시뮬레이션을 수행하기 위해 후보 호스트 머신 리스트로부터 다음 후보 호스트 머신이 획득된다.
- [0281] 제1 호스트 머신에 대해 차감이 성공할 때, 제1 호스트 머신은 제2 호스트 머신에 대한 차감 동작을 수행하지 않고 타겟 호스트 머신으로서 직접 결정될 수 있다. 제1 호스트 머신에 대해 차감이 성공적이지 않을 때, 차감 동작은 호스트 머신에 대해 차감이 성공할 때까지 제2 호스트 머신에 대해 수행될 필요가 있다.
- [0282] 또한, 도 11에 도시된 바와 같이, 생성 유닛(505)은,
- [0283] 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하여 후보 호스트 머신 타겟 리스트를 생성하도록 구성된 순위화 서브유닛(5056)을 더 포함하고,
- [0284] 차감 서브유닛(5053)은 구체적으로, 후보 호스트 머신 타겟 리스트에서의 후보 호스트 머신들의 순위들에 따라 후보 호스트 머신들로부터, 생성될 VM의 리소스 요건을 하나씩 차감하여 나머지 리소스들을 획득하도록 구성된다.
- [0285] 미리 설정된 양은 실제 요건들에 따라 유연하게 설정될 수 있다. 순위화 서브유닛(5056)은 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 랜덤하게 재순위화할 수 있거나, 미리 설정된 규칙에 따라 후보 호스트 머신 리스트에서의 상위 후보 호스트 머신들의 미리 설정된 양을 재순위화할 수 있다. 특정 재순위화 방식은 본 명세서에서 제한되지 않는다. 후보 호스트 머신 리스트에서의 최상위 후보 호스트 머신들의 미리 설정된 양을 재순위화하는 목적은 재순위화를 위해 후보 호스트 머신 리스트에서 제1 k(k는 미리 설정된 양)개의 후보 호스트 머신을 랜덤하게 분산시키고, 후보 호스트 머신 타겟 리스트를 획득하여, 복수의 리소스 스케줄링 프로세스들이 동시 시나리오에서 동일한 타겟 호스트 머신을 선택하는 것을 방지할 수 있어서,

스케줄링 충돌들을 효과적으로 피하게 하기 위한 것이다.

- [0286] 전술한 것으로부터 알 수 있는 바와 같이, 본 출원의 이 실시예에서, 제1 획득 유닛(501)은 생성될 VM에 대응하는 VM 정보를 획득하고, 제2 획득 유닛(502)은 공통 리소스 정보를 획득한다. 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함한다. 그 후, 업데이트 유닛(503)은 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트한다. 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함한다. 마지막으로, 제3 획득 유닛(504)은 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하고, 생성 유닛(505)은 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성한다. 이러한 해결책에서, 리소스 정보 개인 사본은 리소스 스케줄링이 수행되기 전에 제시간에 업데이트될 수 있고, 이는 리소스 정보 개인 사본과 공통 리소스 정보의 동기화를 보장하여, 더 나은 리소스 스케줄링 결과가 달성되고, 리소스 스케줄링 동안 대기할 필요가 없고, 리소스 스케줄링의 효율을 향상시킨다.
- [0287] 이에 대응하여, 본 출원의 실시예는 적어도 하나의 스케줄러 및 적어도 하나의 호스트 머신을 포함하는 클라우드 컴퓨팅 시스템을 추가로 제공한다. 스케줄러는 본 출원의 실시예들에서 제공되는 임의의 리소스 스케줄링 장치를 포함한다. 상세들을 위해, 전술한 리소스 스케줄링 장치에 대응하는 실시예가 참조될 수 있다. 리소스 스케줄링 장치는 구체적으로 스케줄링 서버에 통합될 수 있다. 예로서, 상세들이 다음과 같을 수 있다:
- [0288] 스케줄링 서버는, 생성될 VM에 대응하는 VM 정보를 획득하고; 공통 리소스 정보를 획득하고 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -; 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하고 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -; 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하고; 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하도록 구성된다.
- [0289] 전술한 디바이스들의 특정 구현들에 대해서는, 전술한 실시예들이 참조될 수 있다. 상세들이 본 명세서에 다시 설명되지 않는다.
- [0290] 클라우드 컴퓨팅 시스템은 본 출원의 실시예들에서 제공되는 임의의 리소스 스케줄링 장치를 포함할 수 있고, 따라서, 본 출원의 실시예들에서 제공되는 임의의 클라우드 컴퓨팅 시스템에 의해 달성될 수 있는 유익한 효과들을 달성할 수 있다. 상세들을 위해, 전술한 실시예들을 참조할 수 있다. 상세들이 본 명세서에 다시 설명되지 않는다.
- [0291] 본 출원의 실시예는 스케줄링 서버를 추가로 제공한다. 스케줄링 서버는 본 출원의 실시예들에서 설명된 임의의 스케줄러일 수 있다. 도 12는 본 출원의 실시예들에 관련된 스케줄링 서버의 개략적인 구조도이다. 상세들은 다음과 같다:
- [0292] 스케줄링 서버는 하나 이상의 처리 코어를 포함하는 프로세서(601), 하나 이상의 컴퓨터 판독가능한 저장 매체를 포함하는 메모리(602), 전원(603), 및 입력 유닛(604)과 같은 구성요소들을 포함할 수 있다. 관련 기술분야의 통상의 기술자는 도 12에 도시된 스케줄링 서버의 구조가 스케줄링 서버에 대한 제한을 구성하지 않는다는 것을 이해할 수 있다. 스케줄링 서버는 도면에 도시된 것들보다 더 많거나 더 적은 부분들을 포함할 수 있거나, 일부 부분들을 조합할 수 있거나, 상이한 부분 배열들을 가질 수 있다.
- [0293] 프로세서(601)는 스케줄링 서버의 제어 중심이고, 전체 스케줄링 서버의 다양한 부분들에 다양한 인터페이스들 및 라인들을 이용하여 접속된다. 메모리(602)에 저장된 소프트웨어 프로그램 및/또는 모듈을 구동 또는 실행하고 메모리(602)에 저장된 데이터를 호출함으로써, 프로세서(601)는 스케줄링 서버의 다양한 기능들을 실행하고, 데이터 처리를 수행하며, 이에 의해 전체 스케줄링 서버를 모니터링한다. 일부 실시예들에서, 프로세서(601)는 하나 이상의 처리 코어를 포함할 수 있다. 프로세서(601)는 애플리케이션 프로세서 및 모뎀 프로세서를 통합할 수 있다. 애플리케이션 프로세서는 운영 체제, 사용자 인터페이스, 애플리케이션 프로그램 등을 주로 처리한다. 모뎀 프로세서는 주로 무선 통신을 처리한다. 전술한 모뎀 프로세서는 대안적으로 프로세서(601)에 통합되지 않을 수 있다는 것을 이해할 수 있다.
- [0294] 메모리(602)는 소프트웨어 프로그램 및 모듈을 저장하도록 구성될 수 있다. 프로세서(601)는 메모리(602)에 저장된 하나 이상의 프로그램 및 모듈을 실행하여, 다양한 기능 애플리케이션들 및 데이터 처리를 실행한다. 메모리(602)는 주로 프로그램 저장 영역 및 데이터 저장 영역을 포함할 수 있다. 프로그램 저장 영역은 운영 체제, 적어도 하나의 기능(예컨대, 음향 재생 기능 및 이미지 표시 기능)에 필요한 애플리케이션 프로그램 등을

저장할 수 있다. 데이터 저장 영역은 스케줄링 서버 등의 이용에 따라 생성된 데이터를 저장할 수 있다. 또한, 메모리(602)는 고속 랜덤 액세스 메모리를 포함할 수 있고, 적어도 하나의 자기 디스크 저장 디바이스, 플래시 메모리와 같은, 비휘발성 메모리, 또는 다른 휘발성 솔리드 스테이트 저장 디바이스를 추가로 포함할 수 있다. 이에 대응하여, 메모리(602)는 메모리(602)에 대한 프로세서(601)의 액세스를 제공하도록 메모리 제어기를 추가로 포함할 수 있다.

[0295] 스케줄링 서버는 구성요소들에 전력을 공급하는 전원(603)을 추가로 포함한다. 전원(603)은 전력 관리 시스템을 이용하여 프로세서(601)에 논리적으로 접속될 수 있고, 이에 의해, 전력 관리 시스템을 이용하여 충전, 방전 및 전력 소비 관리와 같은 기능들을 구현한다. 전원(603)은, 직류 또는 교류 전원, 재충전 시스템, 전원 장애 검출 회로, 전력 변환기 또는 인버터, 전력 상태 표시기, 및 임의의 다른 구성요소들 중 하나 이상을 추가로 포함할 수 있다.

[0296] 스케줄링 서버는 입력 숫자 또는 문자 정보를 수신하고, 사용자 설정 및 기능 제어와 관련된 키보드, 마우스, 조이스틱, 광학, 또는 트랙볼 신호 입력을 생성하도록 구성된 입력 유닛(604)을 추가로 포함할 수 있다.

[0297] 도면에 도시되지는 않았지만, 스케줄링 서버는 디스플레이 유닛을 추가로 포함할 수 있다. 상세들은 본 명세서에 추가로 설명되지 않는다. 구체적으로, 이 실시예에서, 스케줄링 서버 내의 프로세서(601)는 다음의 명령어들에 따라 하나 이상의 애플리케이션 프로그램의 프로세스들에 대응하는 실행가능한 파일들을 메모리(602)에 로딩할 수 있고, 메모리(602)에 저장된 애플리케이션 프로그램들을 실행하여, 다음과 같은 다양한 기능들을 구현한다:

[0298] 생성될 VM에 대응하는 VM 정보를 획득하는 것; 공통 리소스 정보를 획득하는 것 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -; 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 것 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -; 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 것; 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하는 것.

[0299] 진술한 동작들 각각의 특정 구현에 대해서는, 진술한 실시예들이 참조될 수 있다. 이것은 본 명세서에서 다시 설명되지 않는다.

[0300] 진술한 것으로부터 알 수 있는 바와 같이, 본 출원의 이 실시예에서, 스케줄링 서버는 생성될 VM에 대응하는 VM 정보를 획득하고 공통 리소스 정보를 획득하고 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -; 그 다음, 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하고 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -; 마지막으로, 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하고, 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성할 수 있다. 이러한 해결책에서, 리소스 정보 개인 사본은 리소스 스케줄링이 수행되기 전에 제시기에 업데이트될 수 있고, 이는 리소스 정보 개인 사본과 공통 리소스 정보의 동기화를 보장하여, 더 나은 리소스 스케줄링 결과가 달성되고, 리소스 스케줄링 동안 대기할 필요가 없고, 리소스 스케줄링의 효율을 향상시킨다.

[0301] 관련 기술분야의 통상의 기술자는 실시예들에서의 다양한 방법들의 모든 또는 일부 단계들이 명령어들을 통해 구현될 수 있거나, 관련 하드웨어를 제어하는 명령어들을 통해 구현될 수 있고, 명령어들이 컴퓨터 판독가능한 저장 매체에 저장되고 프로세서에 의해 로딩 및 실행될 수 있다는 것을 이해할 수 있다.

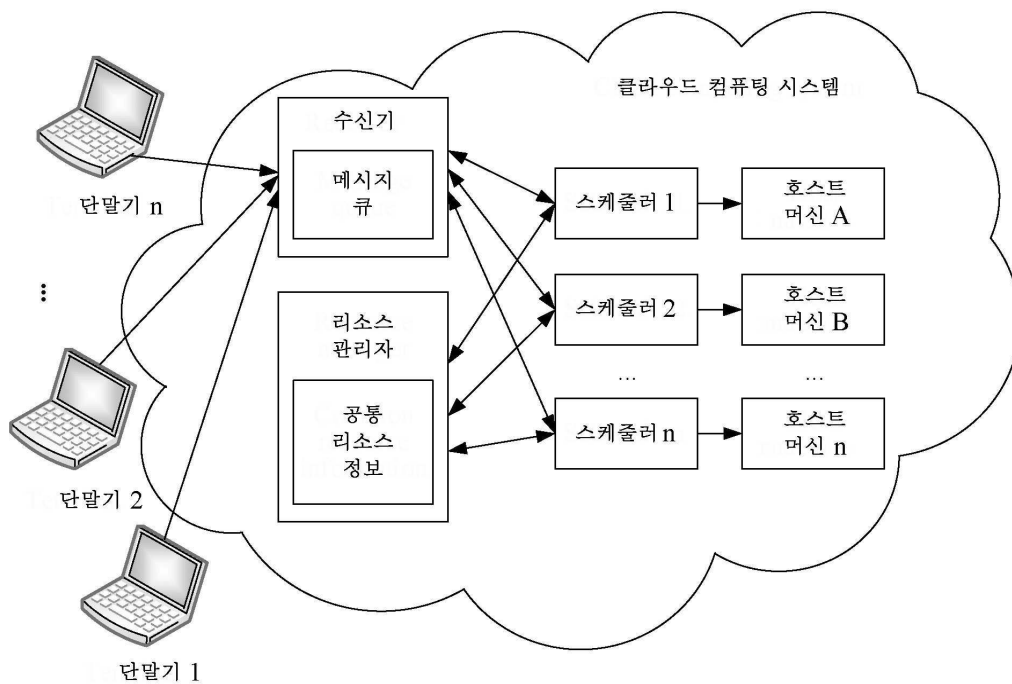
[0302] 따라서, 본 출원의 실시예는 본 출원의 실시예들에 따른 임의의 리소스 스케줄링 방법의 단계들을 수행하기 위해, 프로세서에 의해 로딩되는데 적합한 복수의 명령어들을 저장하는 저장 매체를 제공한다. 예를 들어, 이러한 명령어들은 다음의 단계들을 수행할 수 있다:

[0303] 생성될 VM에 대응하는 VM 정보를 획득하는 단계; 공통 리소스 정보를 획득하는 단계 - 공통 리소스 정보는 클라우드 컴퓨팅 시스템에서의 모든 호스트 머신들에 대응하는 호스트 머신 정보를 포함함 -; 공통 리소스 정보 및 VM 정보에 따라 미리 설정된 리소스 정보 개인 사본을 업데이트하는 단계 - 리소스 정보 개인 사본은 미리 설정된 호스트 머신에 대응하는 호스트 머신 정보를 포함함 -; 리소스 정보 개인 사본에 따라, VM 정보를 충족시키는 적어도 하나의 후보 호스트 머신을 획득하는 단계; 적어도 하나의 후보 호스트 머신으로부터 타겟 호스트 머신을 획득하고, 타겟 호스트 머신 상에 VM을 생성하는 단계.

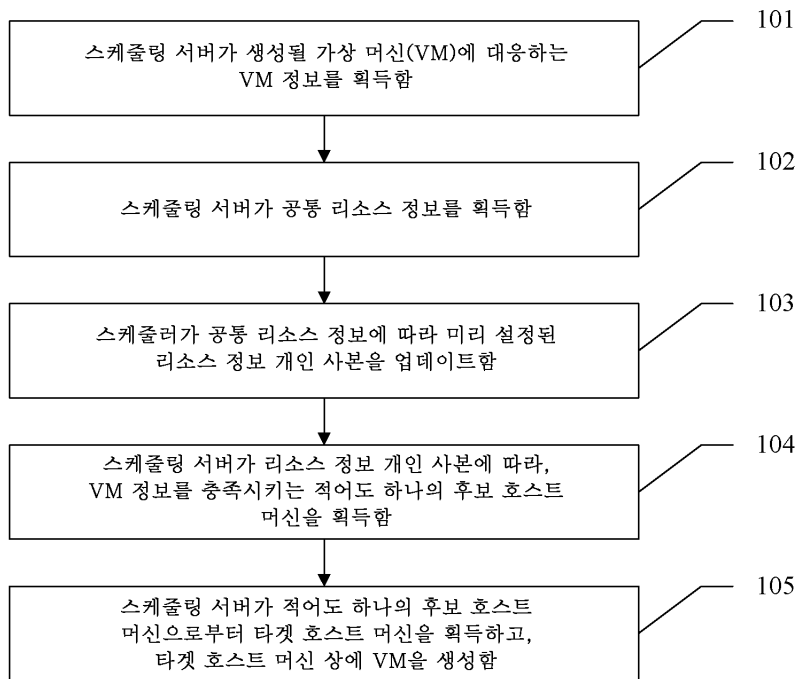
- [0304] 전술한 동작들 각각의 특정 구현에 대해서는, 전술한 실시예들이 참조될 수 있다. 이것은 본 명세서에서 다시 설명되지 않는다.
- [0305] 이러한 저장 매체는, ROM(read-only memory), RAM(random access memory), 자기 디스크, 광학 디스크 등을 포함할 수 있다.
- [0306] 저장 매체에 저장된 명령어들은 본 출원의 실시예들에서의 임의의 리소스 스케줄링 방법의 단계들을 수행할 수 있으므로, 이러한 명령어들은 본 출원의 실시예들에서의 임의의 리소스 스케줄링 방법에 의해 달성될 수 있는 유익한 효과들을 달성할 수 있다. 상세들을 위해, 전술한 실시예들을 참조할 수 있다. 상세들은 본 명세서에 추가로 설명되지 않는다.
- [0307] 본 출원의 실시예들에서 제공되는 리소스 스케줄링 방법 및 장치, 클라우드 컴퓨팅 시스템, 및 저장 매체는 위에서 상세히 설명된다. 본 출원의 원리 및 구현들은 특정 예들을 이용하여 본 명세서에서 설명된다. 전술한 실시예들의 설명들은 단지 본 출원의 방법 및 핵심 아이디어들을 이해하는데 도움을 주기 위해 이용된다. 또한, 관련 기술분야의 통상의 기술자는 본 출원의 아이디어들에 따라 특정 구현들 및 애플리케이션 범위들의 관점에서 본 출원에 대한 변형들을 행할 수 있다. 따라서, 본 명세서의 내용은 본 출원에 대한 제한으로서 해석되어서는 안 된다.

도면

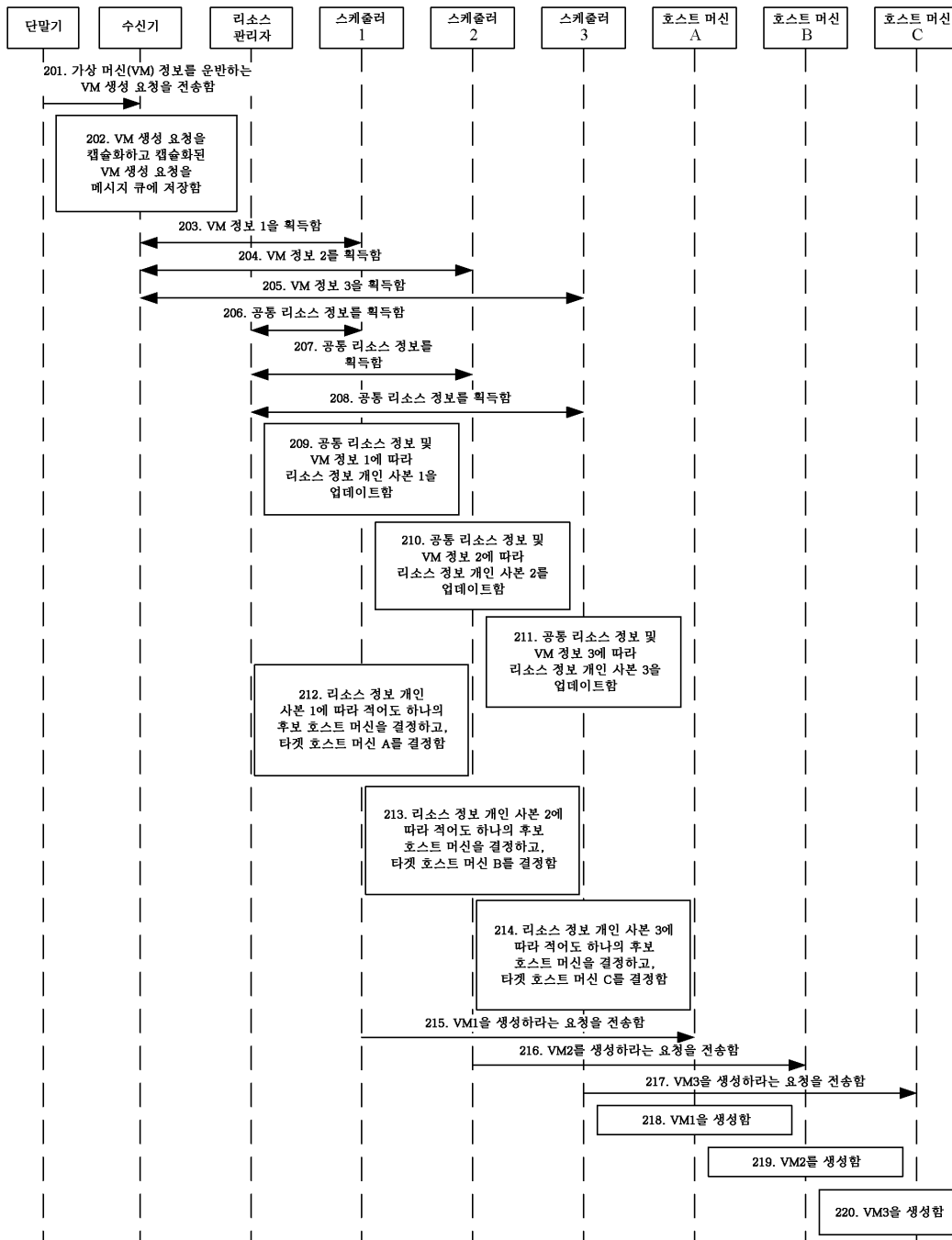
도면1



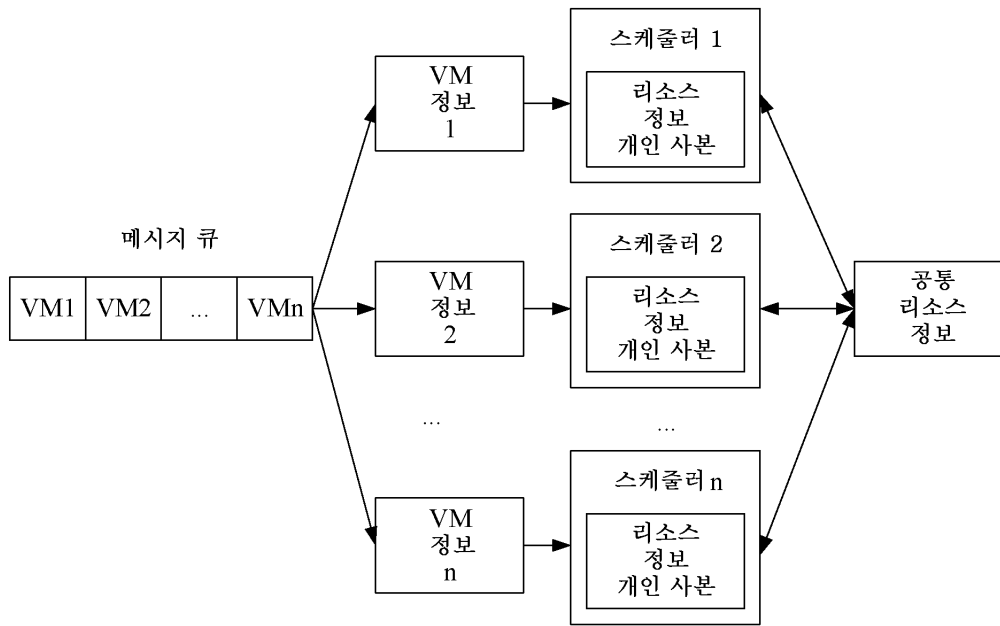
도면2



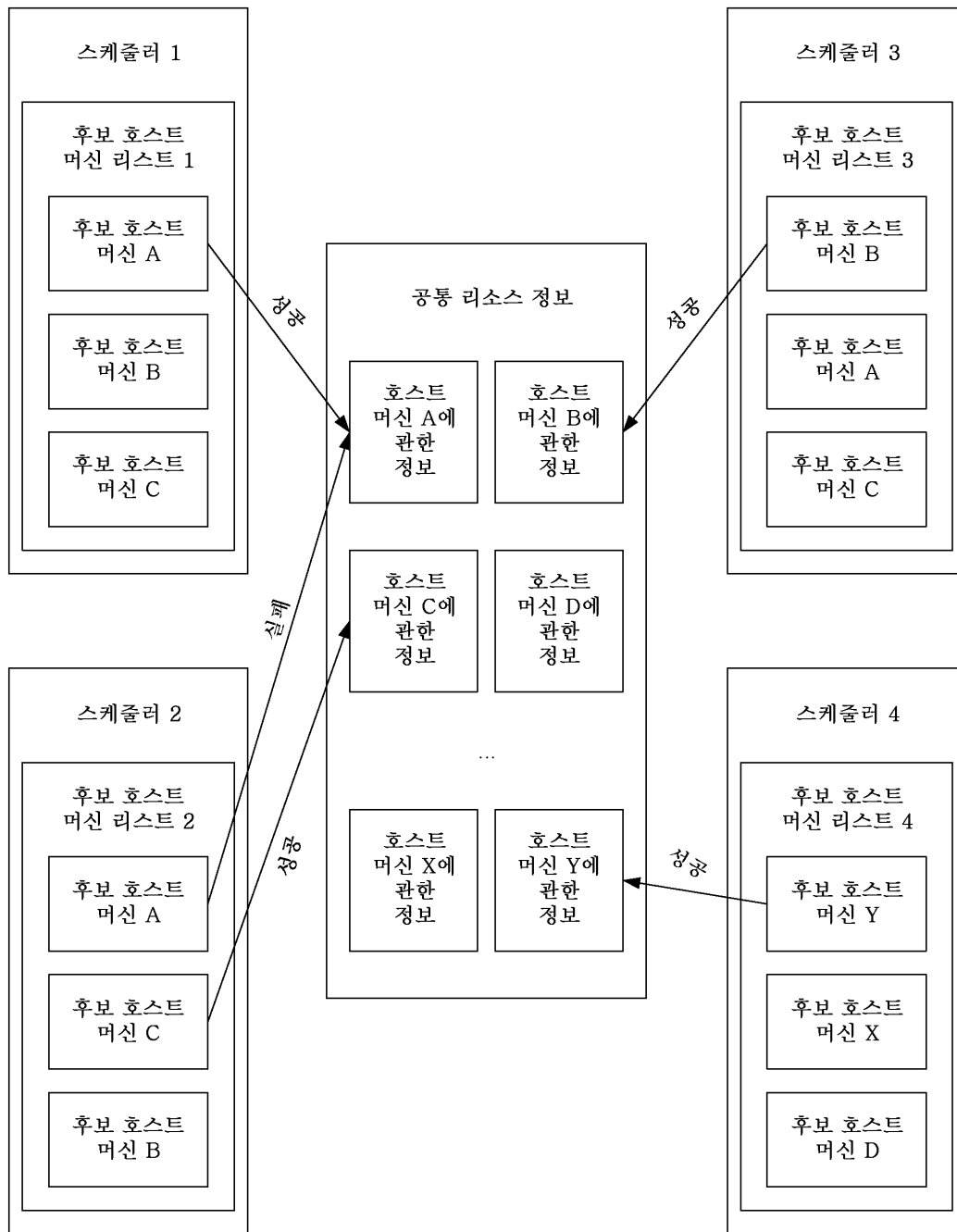
도면3



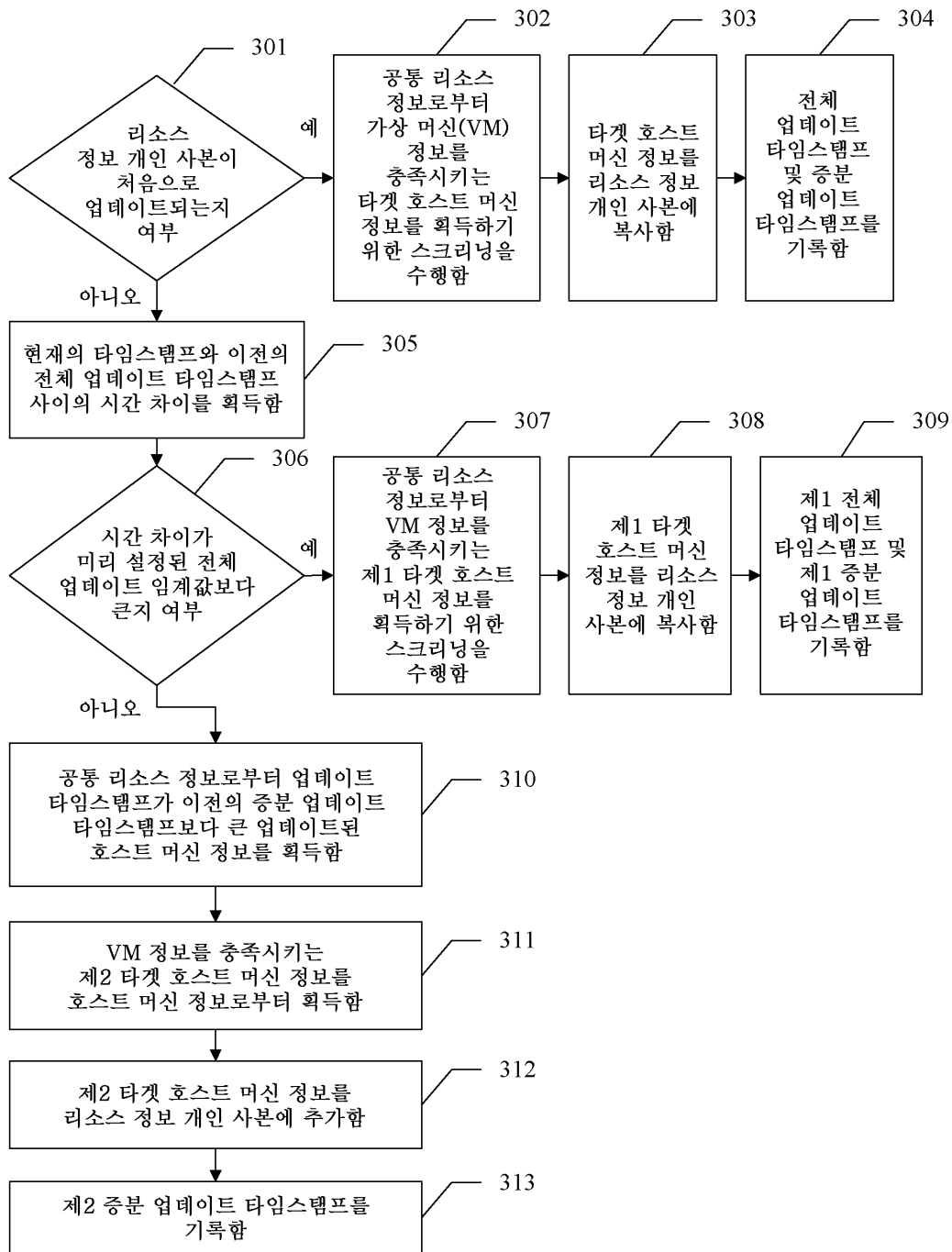
도면4



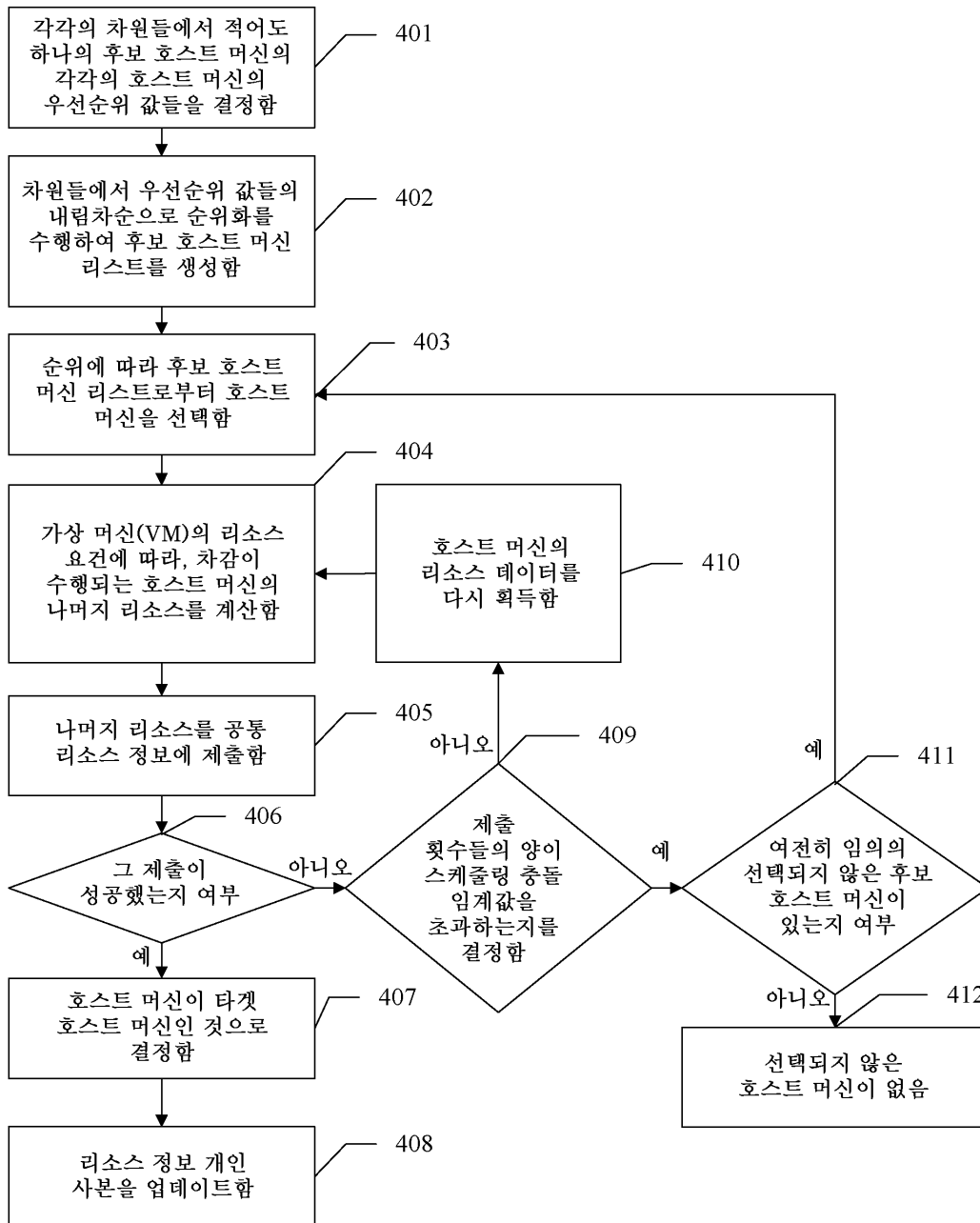
도면5



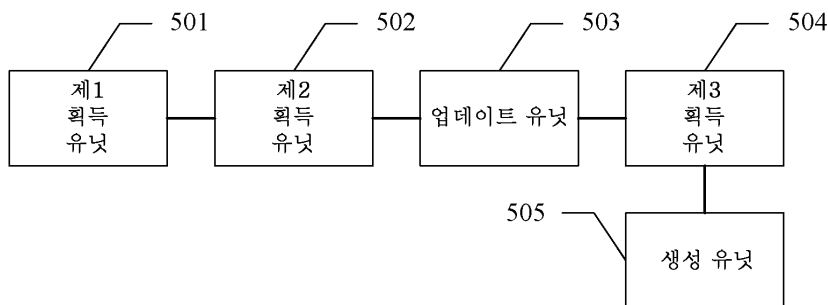
도면6



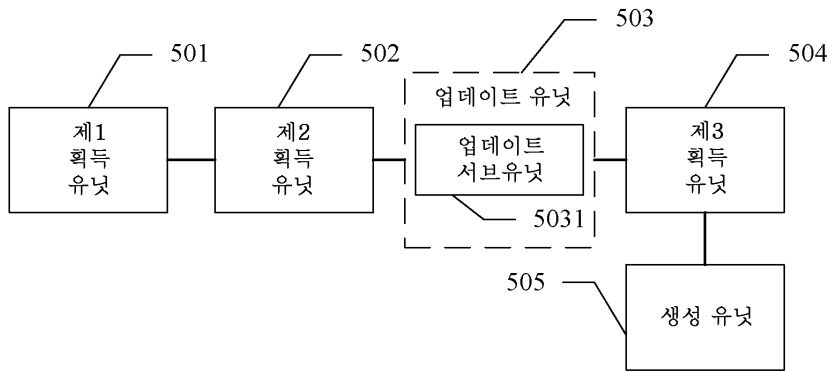
도면7



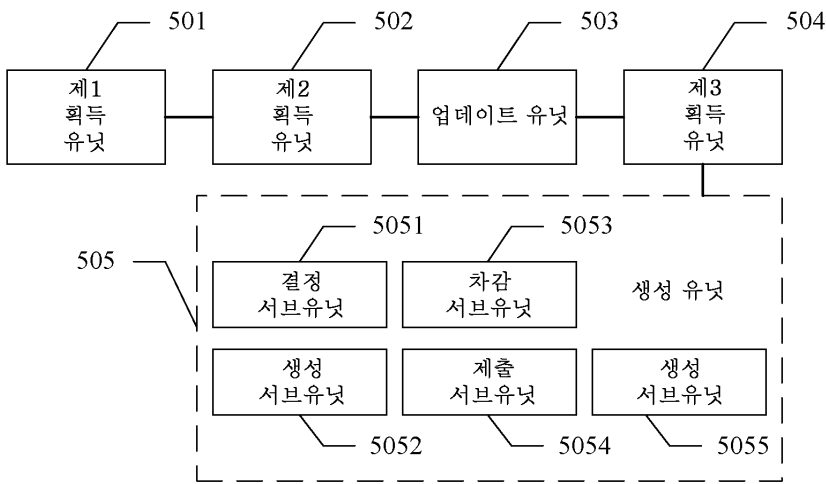
도면8



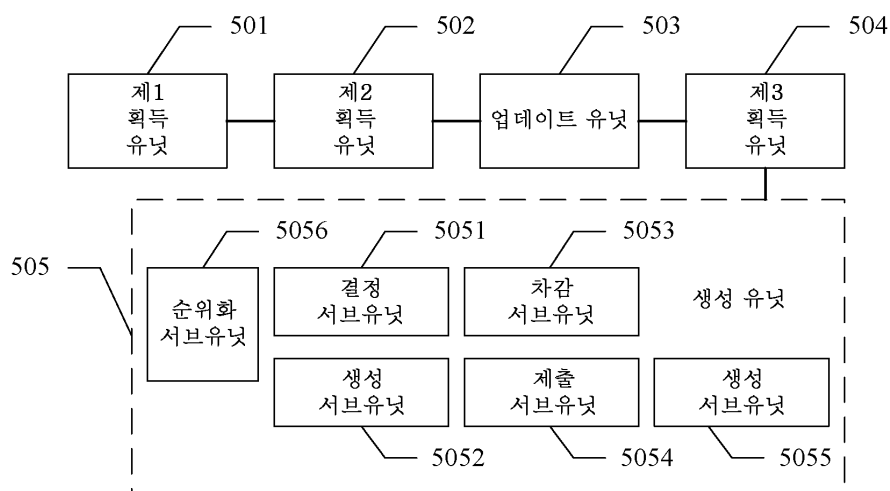
도면9



도면10



도면11



도면12

