



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년02월11일
(11) 등록번호 10-2362045
(24) 등록일자 2022년02월08일

(51) 국제특허분류(Int. Cl.)

G06F 9/455 (2018.01)

(52) CPC특허분류

G06F 9/45558 (2013.01)

G06F 2009/4557 (2019.08)

(21) 출원번호 10-2018-0057320

(22) 출원일자 2018년05월18일

심사청구일자 2021년04월14일

(65) 공개번호 10-2019-0009694

(43) 공개일자 2019년01월29일

(30) 우선권주장

62/534,647 2017년07월19일 미국(US)

15/887,980 2018년02월02일 미국(US)

(56) 선행기술조사문헌

US20140108763 A1

(뒷면에 계속)

(73) 특허권자

삼성전자주식회사

경기도 수원시 영통구 삼성로 129 (매탄동)

(72) 발명자

양, 정위

미국 매사추세츠주 02148 물든 플로렌스 스트리트 99 #820

에반스, 티. 데이비드

미국 캘리포니아주 92122 샌디에고 코스타 베르데 블라바드 8875 #1410

(뒷면에 계속)

(74) 대리인

특허법인 고려

전체 청구항 수 : 총 20 항

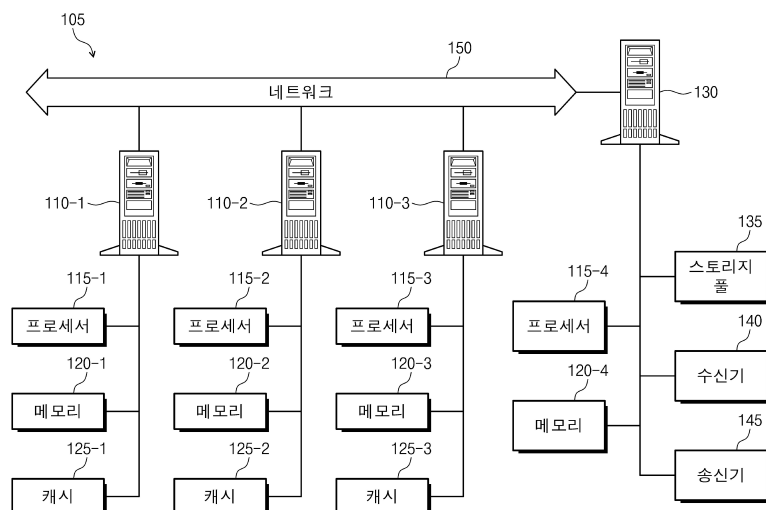
심사관 : 유진태

(54) 발명의 명칭 멀티-티어 올-플래시 데이터센터의 자동 데이터 배치 관리자

(57) 요약

오토-티어링(auto-tiering)을 수행하기 위한 시스템이 개시된다. 시스템은 복수의 리소스들을 제공하고 복수의 스토리지 디바이스들로 구성될 수 있다. 스토리지 디바이스들은 가상 머신들에 대한 데이터를 저장할 수 있다. 수신기는 가상 머신들에 대한 입력/출력 커맨드들 및 성능 데이터를 수신할 수 있다. 송신기는 입력/출력 커맨드들에 대한 응답을 전송할 수 있다. 오토-티어링 컨트롤러는 가상 머신에 대한 데이터를 저장할 스토리지 티어를 선택할 수 있으며 성능 데이터에 따라 스토리지 티어들 간에 데이터를 마이그레이션할 수 있다. 가상 머신의 성능 변화와 스토리지 티어간에 데이터를 마이그레이션하는 데 드는 마이그레이션 비용을 고려하여 스토리지 티어를 선택함으로써, 모든 스토리지 티어에서 모든 가상 머신의 성능을 최적화할 수 있다.

대표도



(72) 발명자 앤드류스, 알렌 미국 캘리포니아주 92019 엘카혼 비아 볼사 11646 마이어스, 클레이 미국 캘리포니아주 92130 샌디에고 페더먼 레인 4267 볼트, 토마스 로리 미국 캘리포니아주 92130 샌디에고 쿡캐넌 코트 4972	(56) 선행기술조사문헌 US20130326183 A1 US20130074092 A1 US20160224257 A1 S. Byan et al. 'Mercury: Host-side flash caching for the data center'. IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST), 2012.04.
---	--

명세서

청구범위

청구항 1

복수의 스토리지 tier들로 구성되어 제1 가상 머신에 대한 제1 데이터 및 제2 가상 머신에 대한 제2 데이터를 저장하고, 복수의 리소스들을 제공하는 복수의 스토리지 디바이스들;

상기 제1 가상 머신으로부터 제1 입력/출력 커맨드, 상기 제2 가상 머신으로부터 제2 입력/출력 커맨드, 상기 복수의 스토리지 tier들에서 상기 제1 가상 머신의 성능을 모델링하는 제1 성능 데이터, 및 상기 복수의 스토리지 tier들에서 상기 제2 가상 머신의 성능을 모델링하는 제2 성능 데이터를 수신하는 수신기;

상기 제1 입력/출력 커맨드에 대한 제1 응답을 상기 제1 가상 머신에 송신하고 상기 제2 입력/출력 커맨드에 대한 제2 응답을 상기 제2 가상 머신에 송신하기 위한 송신기; 및

상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하기 위한 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하기 위한 제2 스토리지 tier를 선택하고, 상기 제1 성능 데이터 및 상기 제2 성능 데이터에 응답하여 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 제1 스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 마이그레이션하기 위한 오토-tier링 컨트롤러를 포함하되,

상기 오토-tier링 컨트롤러는, 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하기 위한 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하기 위한 상기 제2 스토리지 tier를 선택하여, 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하도록 동작하고,

상기 오토-tier링 컨트롤러는, 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 이동시킴으로써 초래되는 성능 변화 및 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로, 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 이동시키는 마이그레이션 비용을 고려하도록 동작하는 시스템.

청구항 2

제 1 항에 있어서,

상기 오토-tier링 컨트롤러는,

적어도 하나의 프로세서를 이용하여 구현되고, 상기 제1 성능 데이터 및 상기 제2 성능 데이터에 응답하여, 상기 제1 가상 머신에 대한 상기 제1 스토리지 tier와 상기 제2 가상 머신에 대한 상기 제2 스토리지 tier를 선택하는 tier 선택 모듈; 및

상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로, 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 마이그레이션하기 위한 마이그레이션 모듈을 포함하는 시스템.

청구항 3

제 1 항에 있어서,

상기 오토-tier링 컨트롤러는, 상기 제1 성능 데이터와 상기 제2 성능 데이터를 사용하여, 상기 제1 가상 머신 및 상기 제2 가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 리소스 매트릭스 생성기를 포함하되,

상기 리소스 매트릭스 생성기는 적어도 하나의 프로세서를 이용하여 구현되는 시스템.

청구항 4

제 3 항에 있어서,

상기 리소스 매트릭스 생성기는, 상기 복수의 스토리지 티어를 각각에서 상기 제1 가상 머신에 대한 제1 성능 메트릭을 상기 복수의 리소스들의 제2 리소스에 따라 상기 복수의 스토리지 티어들 각각의 상기 제1 가상 머신에 대한 제2 성능 메트릭으로 변환하는 메트릭 컨버터를 포함하되,

상기 메트릭 컨버터는 적어도 하나의 프로세서를 이용하여 구현되는 시스템.

청구항 5

제 4 항에 있어서,

상기 제1 성능 메트릭은 상기 제1 가상 머신에 대한 제1 평균 입력/출력 레이턴시 데이터를 포함하는 시스템.

청구항 6

제 1 항에 있어서,

상기 오토-티어링 컨트롤러는, 상기 제1 성능 데이터에 대한 제1 선형 회귀를 수행하고 상기 제2 성능 데이터에 대한 제2 선형 회귀를 수행하는 선형 회귀 모듈을 포함하며; 및

상기 오토-티어링 컨트롤러는, 상기 제1 선형 회귀 및 상기 제2 선형 회귀에 응답하여, 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하기 위한 상기 제1 스토리지 티어를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하기 위한 상기 제2 스토리지 티어를 선택하는 시스템.

청구항 7

제 6 항에 있어서,

상기 오토-티어링 컨트롤러는, 상기 제1 선형 회귀 및 상기 제2 선형 회귀에 응답하여, 상기 제1 가상 머신과 상기 제2 가상 머신에 대한 예측된 평균 입력/출력 레이턴시들을 사용함으로써 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하기 위한 상기 제1 스토리지 티어를 선택하고 상기 제2 가상 머신에 대한 제2 데이터를 저장하기 위한 상기 제2 스토리지 티어를 선택하는 시스템.

청구항 8

제 1 항에 있어서,

상기 오토-티어링 컨트롤러는, 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하기 위한 제1 스토리지 티어를 선택하고, 제2 가상 머신에 대한 제2 데이터를 저장하기 위한 제2 스토리지 티어를 선택하도록 동작하고, 및 하나의 마이그레이션 에포크 당 상기 제1 성능 데이터와 상기 제2 성능 데이터에 응답하여 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 티어로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 티어로 마이그레이션하도록 동작하는 시스템.

청구항 9

제 8 항에 있어서,

상기 제1 성능 데이터는 상기 마이그레이션 에포크 내의 복수의 모니터링 에포크들로부터의 제1 성능 데이터를 포함하고; 및

상기 제2 성능 데이터는 상기 마이그레이션 에포크 내의 상기 복수의 모니터링 에포크들로부터의 제2 성능 데이터를 포함하는 시스템.

청구항 10

제 1 항에 있어서,

상기 제1 가상 머신을 저장하는 제1 가상 머신 서버; 및

상기 제2 가상 머신을 저장하는 제2 가상 머신 서버를 더 포함하는 시스템.

청구항 11

제 10 항에 있어서,

상기 제1 가상 머신 서버는 상기 복수의 스토리지 tier들에서 상기 제1 가상 머신의 성능을 모델링하는 상기 제1 성능 데이터를 생성하기 위한 입력/출력 필터를 포함하는 시스템.

청구항 12

제 11 항에 있어서,

상기 입력/출력 필터는 상기 제1 가상 머신으로부터의 상기 제1 입력/출력 커맨드를 포함하는 복수의 입력/출력 커맨드들에 레이턴시를 주입하고, 상기 주입된 레이턴시에 응답하여 상기 제1 가상 머신에 대한 평균 입력/출력 레이턴시를 결정하도록 동작하는 시스템.

청구항 13

수신기는 복수의 스토리지 tier들에서 제1 가상 머신의 성능을 모델링하는 제1 성능 데이터를 수신하는 단계;

상기 수신기는 상기 복수의 스토리지 tier들에서 제2 가상 머신의 성능을 모델링하는 제2 성능 데이터를 수신하는 단계;

오토-tier링 컨트롤러는 상기 제1 성능 데이터와 상기 제2 성능 데이터를 사용하여, 상기 제1 가상 머신 및 상기 제2 가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계;

상기 오토-tier링 컨트롤러는 상기 리소스 매트릭스를 사용하여 상기 제1 가상 머신에 대한 제1 데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중 제1 스토리지 tier를 선택하는 단계;

상기 오토-tier링 컨트롤러는 상기 리소스 매트릭스를 사용하여 상기 제2 가상 머신에 대한 제2 데이터를 저장하기 위해 상기 복수의 스토리지 tier들의 제2 스토리지 tier를 선택하는 단계; 및

상기 오토-tier링 컨트롤러는 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 마이그레이션하는 단계를 포함하되,

상기 오토-tier링 컨트롤러는 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하도록 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하도록 상기 제2 스토리지 tier를 선택하는 것은 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하고,

상기 오토-tier링 컨트롤러는 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하도록 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하도록 상기 제2 스토리지 tier를 선택하는 것은 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 이동시킴으로써 초래되는 성능 변화를 고려하고,

상기 오토-tier링 컨트롤러는 상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하도록 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하도록 상기 제2 스토리지 tier를 선택하는 것은 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로, 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 마이그레이션하는 마이그레이션 비용을 고려하는 방법.

청구항 14

제 13 항에 있어서,

상기 오토-tier링 컨트롤러는 상기 리소스 매트릭스를 사용하여 상기 제1 가상 머신에 대한 제1 데이터를 저장하기 위해 제1 스토리지 tier를 상기 복수의 스토리지 tier들로부터 선택하는 단계는, 상기 복수의 스토리지 tier들 각각에 대한 상기 리소스 매트릭스 및 전문성 정보를 사용하여 상기 제1 가상 머신에 대한 제1 데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중에서 상기 제1 스토리지 tier를 선택하는 단계를 포함하고; 및

상기 오토-티어링 컨트롤러는 상기 리소스 매트릭스를 사용하여 상기 제2 가상 머신에 대한 제2 데이터를 저장하기 위한 제2 스토리지 티어를 상기 복수의 스토리지 티어들로부터 선택하는 단계는, 상기 복수의 스토리지 티어들 각각에 대한 상기 리소스 매트릭스 및 전문성 정보를 사용하여 상기 제2 가상 머신에 대한 제2 데이터를 저장하기 위해 상기 복수의 스토리지 티어들 중에서 상기 제2 스토리지 티어를 선택하는 단계를 포함하는 방법.

청구항 15

제 14 항에 있어서,

상기 복수의 스토리지 티어들 각각에 대한 상기 전문성 정보는 상기 복수의 스토리지 티어들 각각이 상기 복수의 리소스들 각각을 지원하는지 여부를 식별하는 방법.

청구항 16

제 15 항에 있어서,

상기 복수의 스토리지 티어들에서 상기 제1 가상 머신의 성능을 모델링하는 상기 제1 성능 데이터는, 상기 제1 가상 머신에 대한 평균 입력/출력 레이턴시와 주입된 레이턴시들을 상호 관련시키는 복수의 데이터 포인트들을 포함하는 방법.

청구항 17

제 13 항에 있어서,

상기 오토-티어링 컨트롤러는 상기 복수의 스토리지 티어들의 제1 가상 머신의 성능을 모델링하는 제1 성능 데이터를 수신하는 단계는,

입력/출력 필터에서 상기 가상 머신으로부터 복수의 제1 입력/출력 커맨드들을 수신하는 단계;

상기 복수의 제1 입력/출력 커맨드들에 제1 레이턴시를 주입하는 단계; 및

상기 복수의 제1 입력/출력 커맨드들에 대한 상기 가상 머신에 대한 제1 평균 입력/출력 레이턴시를 계산하는 단계를 포함하는 방법.

청구항 18

제 13 항에 있어서,

상기 제1 가상 머신에 대한 상기 제1 데이터의 적어도 하나를 상기 제1 스토리지 티어로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 티어로 마이그레이션하는 비용은, 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 티어로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 티어로 마이그레이션시키는 데 필요한 마이그레이션 시간을 포함하는 방법.

청구항 19

머신에 의해 실행될 때

복수의 스토리지 티어들에서 제1 가상 머신의 성능을 모델링하는 제1 성능 데이터를 수신하는 단계;

상기 복수의 스토리지 티어들에서 제2 가상 머신의 성능을 모델링하는 제2 성능 데이터를 수신하는 단계;

상기 제1 성능 데이터와 상기 제2 성능 데이터를 사용하여, 상기 제1 가상 머신 및 상기 제2 가상 머신을 상기 복수의 스토리지 티어들과 상기 복수의 스토리지 티어들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계;

상기 리소스 매트릭스를 사용하여 상기 제1 가상 머신에 대한 제1 데이터를 저장하기 위해 상기 복수의 스토리지 티어들 중 제1 스토리지 티어를 선택하는 단계;

상기 리소스 매트릭스를 사용하여 상기 제2 가상 머신에 대한 제2 데이터를 저장하기 위해 상기 복수의 스토리지 티어들의 제2 스토리지 티어를 선택하는 단계; 및

상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 티어로 또는 상기 제2 가상

머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 마이그레이션하는 단계를 야기하고,

상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하도록 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하도록 상기 제2 스토리지 tier를 선택하는 것은 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하고,

상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하도록 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하도록 상기 제2 스토리지 tier를 선택하는 것은 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 이동시킴으로써 초래되는 성능 변화를 고려하고,

상기 제1 가상 머신에 대한 상기 제1 데이터를 저장하도록 상기 제1 스토리지 tier를 선택하고, 상기 제2 가상 머신에 대한 상기 제2 데이터를 저장하도록 상기 제2 스토리지 tier를 선택하는 것은 상기 제1 가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1 스토리지 tier로, 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2 스토리지 tier로 마이그레이션하는 마이그레이션 비용을 고려하는 것을 수행하는 커맨드들을 저장하는 비-일시적 기계 판독 가능 저장매체.

청구항 20

제 19 항에 있어서,

상기 복수의 스토리지 tier들의 제1 가상 머신의 성능을 모델링하는 제1 성능 데이터를 수신하는 단계는,

입력/출력 필터에서 상기 가상 머신으로부터 제1 입력/출력 커맨드를 수신하는 단계;

상기 제1 입력/출력 커맨드에 제1 레이턴시를 주입하는 단계; 및

상기 제1 입력/출력 커맨드에 대한 상기 가상 머신에 대한 제1 평균 입력/출력 레이턴시를 계산하는 단계를 포함하는 비-일시적 기계 판독 가능 저장매체.

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 데이터 스토리지 디바이스에 관한 것으로, 더욱 상세하게는 어떤 스토리지 tier 가상 머신 데이터가 저장되는지를 자동으로 관리하는 것에 관한 것이다.

배경 기술

[0002] 기존의 데이터 스토리지 센터들은 하드 디스크 드라이브 및 이전 형식의 기타 데이터 스토리지에 의존해 왔다. 하드 디스크 드라이브 및 이전 형식의 기타 데이터 스토리지는 상대적으로 긴 레이턴시를 갖기 때문에 가상 머신은 자주 액세스하는 데이터를 로컬 스토리지에 캐시할 수 있다. 대개 데이터 캐싱을 사용함으로써, 데이터 액세스 시간이 10 배 이상 증가하는 상대적 이점이 있다.

[0003] 그러나 플래시 스토리지가 보다 보편화되고 저렴해짐에 따라 데이터 스토리지와 캐시 간의 액세스 시간 차이가 감소하고 있다. 이러한 캐싱의 감소된 이점은 데이터가 데이터 센터에만 독점적으로 저장되고 전혀 캐시되지 않는다는 것을 의미할 수 있다. 플래시 스토리지에 데이터의 중복적인 사본(캐시 및 데이터 센터 모두에)을 저장하는 것이 비용이 많이 들기 때문에 데이터를 가장 효율적으로 데이터 스토리지 센터에 저장하는 것이 중요하다.

[0004] 또한 하드 디스크 드라이브와 같은 기존의 스토리지와 달리 플래시 스토리지는 다른 특수화를 거칠 수 있다. 기존의 스토리지는 디바이스 특수화를 고려하지 않기 때문에, 기존의 기술을 사용하여 데이터 스토리지를 위한 임의의 tier를 선택하는 것이 차선이 될 수 있다.

[0005] 만족스러운 액세스 시간을 제공하고 다른 스토리지 tier들이 제공하는 특수화를 활용하는 플래시 스토리지를 사용하여, 데이터를 데이터 스토리지 센터에 보다 최적으로 저장하는 방법이 필요하다.

발명의 내용

해결하려는 과제

- [0006] 만족스러운 액세스 시간을 제공하고 다른 스토리지 tier들이 제공하는 특수화를 활용하는 플래시 스토리지를 사용하여, 데이터를 데이터 스토리지 센터에 보다 최적으로 저장하는 방법이 개시될 수 있다.
- [0007] 또한, 상기 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공하는데 있다. 본 실시 예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 이하의 실시 예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제의 해결 수단

- [0008] 시스템은, 복수의 리소스들을 제공하는 복수의 스토리지 디바이스들, 상기 복수의 스토리지 디바이스들은 복수의 스토리지 tier들로 구성되어 제1가상 머신에 대한 제1 데이터 및 제2가상 머신에 대한 제2데이터를 저장하고, 상기 제1가상 머신으로부터 제1입력/출력 커맨드, 상기 제2가상 머신으로부터 제2입력/출력 커맨드, 상기 복수의 스토리지 tier들에서 상기 제1가상 머신의 성능을 모델링하는 제1성능 데이터, 및 상기 복수의 스토리지 tier들에서 상기 제2가상 머신의 성능을 모델링하는 제2성능 데이터를 수신하는 수신기, 상기 제1 입력/출력 커맨드에 대한 제1응답을 상기 제1가상 머신에 송신하고 상기 제2 입력/출력 커맨드에 대한 제2응답을 상기 제2가상 머신에 송신하기 위한 송신기, 및 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위한 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하기 위한 제2스토리지 tier를 선택하고, 상기 제1성능 데이터 및 상기 제2성능 데이터에 응답하여 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션하기 위한 오토-tier링 컨트롤러, 상기 오토-tier링 컨트롤러는, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위한 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하기 위한 상기 제2스토리지 tier를 선택하여, 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하고, 상기 오토-tier링 컨트롤러는, 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2스토리지 tier로 이동시킴으로써 초래되는 성능 변화 및 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로, 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 이동시키는 마이그레이션 비용을 고려하도록 동작한다.
- [0009] 방법은, 복수의 스토리지 tier들에서 제1가상 머신의 성능을 모델링하는 제1성능 데이터를 수신하는 단계, 상기 복수의 스토리지 tier들에서 제2가상 머신의 성능을 모델링하는 제2성능 데이터를 수신하는 단계, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계, 상기 리소스 매트릭스를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중 제1스토리지 tier를 선택하는 단계, 상기 리소스 매트릭스를 사용하여 상기 제2가상 머신에 대한 제2데이터를 저장하기 위해 상기 복수의 스토리지 tier들의 제2스토리지 tier를 선택하는 단계, 및 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션하는 단계, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하고, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2 가상 머신에 대한 상기 제2 데이터를 상기 제2스토리지 tier로 이동시킴으로써 초래되는 성능 변화를 고려하고, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로, 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션하는 마이그레이션 비용을 고려할 수 있다.

도면의 간단한 설명

- [0010] 도1은 본 발명의 실시 예에 따라, 가상 머신을 호스팅하고 오토-tier링을 수행하도록 구비된 스토리지 풀을 사

용할 수 있는 네트워크화된 머신들을 나타낸다.

도2는 도 1의 머신들의 추가적인 세부 사항을 도시한다.

도3은 본 발명의 실시 예에 따라, 도 1의 스토리지 풀과 오토-티어링으로 상호 작용하는 도1의 서버의 부가적인 세부 사항을 도시한다.

도4는 상이한 스토리지 디바이스들에 의해 제공되는 상이한 리소스들을 도시한다.

도5는 하나의 스토리지에서 다른 스토리지로 마이그레이션되는 도3의 가상 머신의 파일들을 나타낸다.

도6a 및 도6b는 상이한 시점에서의 스토리지 tier들에 대한 가상 머신들의 최적의 할당들을 도시한다.

도7은 도3의 가상 머신에 대한 레이턴시 주입 및 성능 데이터 결정을 수행하는 도3의 입력/출력 필터를 도시한다.

도8은 도3의 오토-티어링 컨트롤러의 세부 사항을 도시한다.

도9는 도7의 성능 데이터를 가상 머신들, 스토리지 tier들, 및 스토리지 tier들의 리소스들을 상관시키는 매트릭스로 변환하는 것을 도시한다.

도10은 도7의 성능 데이터의 선형 회귀를 도시한다.

도11은 도3의 스토리지 tier들에 대한 도8의 전문성 정보를 도시한다.

도12는 도3의 가상 머신들을 도3의 다른 스토리지 tier들로 마이그레이션시키는 데 사용되는 스코어 매트릭스의 계산을 도시한다.

도13은 모니터링 에포크와 마이그레이션 에포크 사이의 관계를 보여준다.

도14는 일 실시 예에 따라, 도7의 성능 데이터를 생성하기 위해 도7의 입력/출력 커맨드들로 레이턴시를 주입하기 위한 도3의 입력/출력 필터에 대한 예시적인 절차의 흐름도를 도시한다.

도15a-15b는 일 실시 예에 따라, 스토리지 tier들을 통한 가상 머신들의 마이그레이션을 관리하기 위한 도3의 오토-티어링 컨트롤러에 대한 예시적인 절차의 흐름도를 도시한다.

도16은 일 실시 예에 따라, 도9의 리소스 매트릭스를 생성하기 위한, 도3의 오토-티어링 컨트롤러에 대한 예시적인 절차의 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0011] 본 발명의 개념에 대한 상세한 설명이 첨부된 도면에 예시되어있다. 다음의 상세한 설명에서, 본 발명의 개념을 완전히 이해할 수 있도록 수많은 특정 세부 사항이 개시될 수 있다. 그러나, 당업자는 이러한 특정 세부 사항 없이 본 발명의 개념을 실행할 수 있음을 이해하여야 한다. 다른 예들에서, 공지된 방법들, 절차들, 구성 요소들, 회로들 및 네트워크들은 실시 예들의 태양들을 불필요하게 불명료하게하지 않기 위해 상세하게 설명되지 않았다.
- [0012] 제1, 제2 등의 용어가 본 명세서에서 다양한 구성 요소를 설명하기 위해 사용될 수 있지만, 이들 구성 요소는 이들 용어에 의해 제한되지 않음을 이해할 것이다. 이러한 용어는 하나의 요소를 다른 요소와 구별하기 위해서만 사용된다. 예를 들어, 제1 모듈은 제2 모듈로 지칭될 수 있고, 유사하게, 제2 모듈은 본 발명의 개념의 범위를 벗어나지 않으면서 제1 모듈로 지칭될 수 있다.
- [0013] 본 발명의 개념의 설명에 사용된 용어는 특정 실시 예만을 설명하기 위한 것이며, 본 발명의 개념을 제한하려는 것은 아니다. 본 발명의 개념 및 첨부된 청구 범위의 설명에 사용되는 바와 같이, 단수 형태 "a", "an", 및 "the"는 문맥 상 달리 명시하지 않는 한 복수 형태를 포함하는 것으로 의도된다. 본 명세서에서 사용된 "및/또는"이라는 용어는 하나 이상으로 열거된 관련 항목의 임의의 그리고 모든 가능한 조합을 포함할 수 있다. 본 명세서에서 사용되는 "포함한다" 및/또는 "포함하는"이라는 용어는 명시된 특징, 정수, 단계, 동작, 구성 요소 및/또는 엘리먼트의 존재를 나타내지만, 하나 이상의 다른 특징, 정수, 단계, 동작, 엘리먼트, 구성 요소 및/또는 그룹의 존재 또는 추가를 배제하지 않는다. 도면의 구성 요소 및 특징은 반드시 일정한 비율로 그려지는 것은 아니다.
- [0014] SSD(Solid State Drive) 가격이 계속 하락하고 SSD 용량이 계속 증가함에 따라 가까운 미래에 모든 플래시 스토

리지 시스템이 엔터프라이즈 데이터 센터에 채택될 것이다. 결과적으로 하드 디스크 드라이브(HDD)의 장점은 점차 희석되어 멀티-레벨 셀(MLC) SSD들과 같은 대용량의 로우 엔드 SSD들로 대체될 수 있다. 따라서, 비용 효율적인 설계는, 티어들 각각이 잠재적으로 고속, 대용량, 읽기 입력/출력(I/O) 커맨드에 강함, 긴 수명 등과 같은 고유한 특수 기능을 가지는 멀티-티어 SSD를 구현할 수 있다.

- [0015] 한편, 일반적인 엔터프라이즈 데이터 센터 사용 사례는 하이퍼바이저(Hypervisor)-가상 머신(Virtual Machine, VM) 구조를 사용한다. VM의 입력/출력 워크로드 패턴들은 다를 수 있으며 서비스의 상위 티어로 그들을 업그레이드 할 때 볼 수 있는 성능 향상은 동일하지 않다. 이를 통해 "오토 티어링(Auto-tiering)"이라는 전체 솔루션이 올-플래시 멀티-티어 데이터 센터에서 VM 할당 및 마이그레이션을 처리하여 스토리지 리소스를 최대한 활용하고 성능을 최적화하며 마이그레이션 오버 헤드를 감소시킬 수 있다. 오토 티어링은 또한 과거 및 예측된 성능 요소들을 고려할 수 있다.
- [0016] VM들은 리눅스(Linux) 및 윈도우(Windows)와 같은 게스트 운영 체제(OS) 및 응용 프로그램을 실행할 수 있다. VM들은 서로 분리된다. 클라우드 서비스 공급 업체는 이러한 VM들을 사용자들에게 "렌트"한다. VM들은 사용자 응용 프로그램에 기초하여 다른 워크로드 특성을 가질 수 있으므로, 스토리지 디바이스 속도에 대해 다른 레벨의 감도를 가질 수 있다.
- [0017] Xen, KVM, 및 VMware와 같은 VM 하이퍼바이저 소프트웨어는 단일 서버 내에 여러 가상 머신들(VM)을 호스팅할 수 있다. VM 하이퍼바이저 소프트웨어는 스케줄링, 리소스 관리, 시스템 소프트웨어 API(Application Programming Interfaces), 및 하드웨어 가상화를 담당한다. 본 발명의 실시 예는 VM 하이퍼바이저 레벨에 설치된 오토 티어링 데몬(AutoTiering Daemon)을 포함할 수 있으며, 그것의 역할은 오토 티어링 입력/출력 필터 및 오토 티어링 컨트롤러(AutoTiering Controller)와 통신하는 것이다.
- [0018] 모든 VM 서버들은 비 휘발성 메모리 익스프레스(NVMe) SSD, 3D XPoint Non-Volatile Memory(NVM) SS, 및 MLC/트리플 레벨 셀들(TLC)/쿼드 레벨 셀들(QLC) SSD들과 같은 장치를 포함할 수 있는 백엔드 올-플래시(backend all-flash) 멀티-티어 SSD 디스크들을 공유하기 위해 중앙 집중식 스토리지 풀에 연결될 수 있다. 각 티어에는 빠른 속도, 대용량 저장 용량 등과 같은 다른 전문 분야가 있을 수 있다.
- [0019] 오토 티어링은 오토 티어링 입력/출력 필터, 오토 티어링 데몬(Daemon), 및 오토 티어링 컨트롤러의 세 가지 구성 요소를 포함할 수 있다.
- [0020] 오토 티어링 입력/출력 필터 드라이버는 관리되는 모든 VM(즉, VMDK(Virtual Machine Disk) 파일)에 첨부될 수 있다. 오토 티어링 입력/출력 필터 드라이버는 입력/출력 관련 통계를 수집하고 모든 VMDK에 대해 특별한 레이턴시 테스트를 실행할 수도 있다. 데이터는 샘플 주기율로 수집될 수 있으며 결과는 모든 VMDK로부터 모든 데이터를 수집하는 호스트 시스템의 오토 티어링 데몬으로 전송될 수 있다.
- [0021] 오토 티어링 데몬은 모든 물리적 머신들의 VM 하이퍼바이저 상에서 실행될 수 있다. 오토 티어링 데몬은 VM 서버의 워크로드 변경(입력/출력 액세스 패턴 변경)을 추적하고 오토 티어링 입력/출력 필터로부터 주입된 레이턴시 테스트 결과를 수집하여 오토 티어링 컨트롤러에 전송한다.
- [0022] 오토 티어링 컨트롤러는 전용 서버 또는 스토리지 시스템의 임베디드 시스템에서 실행될 수 있다. VM이 대응하는 마이그레이션 오버 헤드 뿐만 아니라 다른 티어로 마이그레이션되는 경우, 오토 티어링 컨트롤러는 예측된 VM 성능을 기반으로 마이그레이션을 트리거하도록 결정할 수 있다.
- [0023] 각 VM 관점에서, 특정 VM이 한 티어에서 다른 티어로 마이그레이션되는 이유는 VM이 마이그레이션 이후에 더 잘 수행될 수 있다는 것이다(보다 적은 평균 입력/출력 레이턴시, 높은 IOPS(I/O Operations Per Second), 높은 처리량(throughput), 등). 그러나 이와 동시에 이와 대응하는 마이그레이션 비용이 있다. 이는 하나의 티어에서 다른 티어로 VM을 마이그레이션하는 것이 상대적으로 비용이 많이 들며 무시할 수 없기 때문이다.
- [0024] 글로벌 최적화 관점에서, 리소스 제약 조건과 대응하는 SLA(Service Level Agreements)로 인해 동시에 가장 좋아하는 티어들로 마이그레이션 할 모든 VM 요청들을 충족시키기는 어렵다. 글로벌 최적화는 이전의 마이그레이션의 영향은 물론 시간 경과에 따른 VM들의 변경을 고려하여야 한다. 예를 들어, 현재 최상의 할당 솔루션은 런타임 중에 VM 동작을 변경함으로써 향후에 나쁜 상황을 초래할 수 있다. 따라서, 글로벌 최적화는 잠재적인 이점과 불리한 점, 마이그레이션 오버 헤드, 각 티어의 VM들에 대한 과거 및 예측된 성능, SLA (service level agreement) 등을 고려하여야 한다. 이러한 정보를 기반으로, 오토 티어링은 식1-6를 사용하여 멀티-티어 올-플래시 스토리지 시스템을 위한 최적의 솔루션에 도달하려고 시도할 수 있다.

표 1

표기	의미
v	VM id, $v \in [1, v_{max}]$, v_{max} 는 마지막 VM.
t	티어id, $t \in [1, t_{max}]$, t_{max} 는 마지막 티어.
$t_{v,t}$	에포크 τ 동안 VM v 의 호스팅 티어
k	IOPS, 처리량, 스토리지 크기, 등과 같은 서로 다른 타입들의 리소스들, $k \in [1, k_{max}]$
τ	시간적인 에포크 id, $\tau \in [0, \infty)$.
α_k	리소스의 웨이트 k .
β	마이그레이션 비용의 웨이트.
$r(v, t_{v,t}, \tau, k)$	티어 $t_{v,t}$ 에서 실행중인 VM v 의 예측된 k 리소스 사용 타입
$m(v, t_{v,t-1}, t_{v,t})$	티어 $t_{v,t-1}$ 에서부터 티어 $t_{v,t}$ 까지의 에포크 τ 동안의 VM v 의 마이그레이션 비용.
k_{stg}	스토리지 크기.
$Pr(v, t, \tau)$, $Pw(v, t, \tau)$	에포크 τ 동안의 티어 t 상의 VM v 의 읽기 또는 쓰기 커맨드 처리량.
$Pr(A, t, \tau)$, $Pw(A, t, \tau)$	에포크 τ 동안의 티어 t 상의 남은 가용 읽기 또는 쓰기 처리량.
$W_{v,t}$	VM v 의 웨이트.
Γ_k	사용되는 각 리소스 타입의 상한(백분율).
$R(t_{v,t}, k)$	티어 $t_{v,t}$ 에서 k 번째 타입의 총 용량.

[0025]

[0026] <표기들>

표 2

$$\sum_{\forall v, \forall t, \forall \tau} w_{v,t} \cdot [\sum_k \alpha_k \cdot r(v, t_{v,t}, \tau, k) - \beta \cdot m(v, t_{v,t-1}, t_{v,t})] \quad (1)$$

[0027]

[0028] <최소화>

표 3

$r(v, t_{v,\tau}, \tau, k) = r_{prd}(v, t_{v,\tau-1}, t_{v,\tau}, \tau-1, k) \geq 0, \varphi$	$\forall v, \forall t, \forall \tau, \varphi$	(2) φ
$r(v, t_{v,\tau_1}, \tau_1, k_s) \equiv r(v, t_{v,\tau_2}, \tau_2, k_s) \geq 0, \varphi$	$\forall \tau_1, \tau_2 \in [0, +\infty)$	(3) φ
$m(v, t_{v,\tau-1}, t_{v,\tau}) = \frac{r(v, t_{v,\tau-1}, \tau-1, k_s)}{\min(Pr(\Lambda, t_{v,\tau-1}, \tau-1) + Pr(v, t_{v,\tau-1}, \tau-1), Pw(\Lambda, t_{v,\tau-1}, \tau-1))}, \varphi$		(4) φ
$t_{v,\tau} \neq \emptyset, \varphi$	$\forall v, \forall \tau, \varphi$	(5) φ
$\sum_{\forall v} r(v, t_{v,\tau}, \tau, k) \leq \Gamma_k \cdot R(t_{v,\tau}, k), \varphi$	$\forall t, \forall k, \forall \tau, \varphi$	(6) φ

[0029]

[0030]

<대상>

[0031]

식 1은 목적 함수를 나타낸다. 주요 아이디어는 전체 성능 이득에서 마이그레이션 비용을 뺀 값을 최대화하는 것이다: "성능 이득 - 성능 페널티". 내부 "sum" 연산자는 v를 티어($t_{\tau-1}$)에서 티어(t_τ)로 마이그레이션한다고 가정할 때, 각 VM의 모든 타입의 리소스 사용량(IOPS, 처리량, 및 스토리지 크기 등)의 가중치 합계를 수행할 수 있다. 여기서 마이그레이션은 하나의 호스트 서버에서 다른 호스트 서버로 VM을 마이그레이션하지 않는다: 오로지 백엔드(backend) VMDK 파일들이 하나의 SSD 티어에서 다른 SSD 티어로 마이그레이션된다. 따라서, CPU, 메모리 등과 같은 비-디스크-I/O 관련 리소스가 고려될 필요가 없다. 외부 'sum' 연산자는 가능한 모든 마이그레이션 사례들을 반복한다. 가중치 매개 변수 $w_{v,\tau}$ 는 각 VM의 SLA를 반영할 수 있다.

[0032]

식2는 성능 이득을 예측하기 위해 예측 모델 함수가 이용될 수 있음을 나타낸다. 식3은 각 마이그레이션 전후에 스토리지 크기(즉, VMDK 크기)가 변경되지 않도록 할 수 있다. 식4는 일시적인 마이그레이션 오버헤드를 예측한다. 상세히는, 일시적인 마이그레이션 오버헤드는 각 티어들에 데이터를 읽고 쓰는 것에 의해 야기되는 병목으로 나눈 마이그레이션 될 VM의 크기이다. 상세히는, $Pr(\Lambda, t_{v,\tau-1}, \tau-1)$ 은 사용 가능한 잔여 속도를 나타내고, VM은 마이그레이션 중에 정지될 수 있기 때문에, 이러한 VM에 의해 사용된 처리량도 사용가능하다(즉, $Pr(\Lambda, t_{v,\tau-1}, \tau-1)$ 가 다시 더해질 수 있다.) $Pw(\Lambda, t_{v,\tau-1}, \tau-1)$ 는 마이그레이션하는 쓰기 속도를 나타낸다. 식 5는 오직 하나의 디스크 티어에 의해서 VM이 호스팅되어야 할 필요성을 반영한다. 마지막으로, 식6은 리소스 제약 조건을 나타내는데, 여기서 Γ_k 는 사용될 수 있는 각 리소스 타입의 미리 설정된 상한 (백분율)이다.

[0033]

다음과 같은 문제로 인해 런타임 중에 모든 에포크들에 대해 글로벌 최적화를 수행하기가 어렵다.

[0034]

1) 시스템은 미래의 워크로드 패턴에 대한 정보가 거의 없기 때문에 런타임 동안 모든 τ 시간 동안 글로벌 최적화를 수행하기 어렵다.

[0035]

2) 글로벌 최적화를 수행할 때, 각 에포크에 대한 결정은 독립적이지 않다. 즉, 기존의 서브-최적화-기반의 동적 프로그래밍 기법은 각 에포크의 목표 함수 값인 "성능 이득 - 성능 페널티"를 최적화함으로써 문제를 해결하는 데에 사용되지 못할 수 있음을 의미한다.

[0036]

3) 성능 예측 모델(마이그레이션 없이 다른 티어들에서의 VM의 성능을 예측함)의 복잡성에 따라, 최적화 문제가 쉽게 NP-하드(NP-hard)가 될 수 있다.

[0037]

따라서 수용 가능한 시간에 이러한 문제를 해결하려면 몇 가지 제약 사항이 완화될 필요가 있다.

[0038]

모든 시간 동안 글로벌 최적화를 수행하는 대신, 각 에포크(즉, 런타임 그리디)에 대해서만 최적화를 시도할 수 있다.

[0039]

문제는 가중 배낭 문제와 유사한다. 더욱이, 높은 IOPS, 높은 처리량, 큰 공간, 작은 WAF(Write Amplification Factor) 함수, 대규모 오버-프로비저닝(over-provisioning), 및 큰 프로그램/삭제(P/E) 주기 등과 같은 각 티어의 성능 "전문성"이 미리 공지되기 때문에, 전문성 리소스 차원(예를 들어, 티어 1의 IOPS 및 처리량)에 중점을 둔 VM들 각각의 예상 성능에 대해 "점수"를 계산할 수 있으며, 대응하는 마이그레이션 오버헤드의 예측과 함

게 성능-전문성 점수 순위에 기초하여, 마이그레이션 결정을 내릴 수 있다.

- [0040] 2 개의 상이한 에포크가 스케줄 될 수 있다 : 모니터링 에포크, 성능을 평가하기 위한 빈도를 나타내고 성능 예측 모델을 회귀하고, 그것은 임의의 원하는 간격으로 설정될 수 있다: 예를 들어, 15 분; 마이그레이션 에포크, 하나의 티어에서 다른 티어로의 VM 이동이 수행될 수 있는 빈도를 나타낸다. 이것은 임의의 원하는 간격으로 설정될 수 있다: 예를 들어, 하루 또는 일주일. 모니터링 에포크와 마이그레이션 에포크는 반드시 고정되어있는 것은 아니며 런타임 중에 적응적으로 크기를 조정할 수 있는 함수를 사용하여 결정될 수 있다. 또한 마이그레이션 에포크는 모니터링 에포크의 정수배일 필요는 없다. 이러한 창 크기들이 작아질수록 스토리지 시스템이 모니터링되고 측정되고 VM들이 마이그레이션되는 빈도가 높아진다. 시스템 관리자는 배치(deployment)전에 민감도 분석을 수행하여 정확성과 마이그레이션 비용 사이의 트레이드-오프의 균형을 맞출 수 있다.
- [0041] 마이그레이션 없이 다른 티어들의 VM 성능을 평가하기 위해, 각 VM의 입력/출력 커맨드에 추가 레이턴시들을 수동으로 주입하고 오토 티어링 입력/출력 필터 API(즉, 입력/출력 필터링을 위한 VMware vSphere API)를 호출하여 전체 입력/출력 레이턴시에 미치는 영향을 측정함으로써, 다른 티어의 VM 성능을 "에뮬레이트(emulate)"할 수 있다.
- [0042] 다른 티어들의 VM 성능의 변화는 선형 커브를 사용하여 모델링될 수 있다. 평균 입력/출력 레이턴시와 주입된 레이턴시와의 관계를 나타내는 그래프는 선형 방정식으로 회귀될 수 있다. 라인의 기울기는 그때 실행되고 있는 응용 프로그램이 동기식 또는 비동기식인지 따라 다를 수 있다. 동기식 애플리케이션과 비동기식 애플리케이션은 동작 방식이 서로 다르고 라인의 기울기에 반영될 수 있다. 특히, 동기 응용 프로그램이 입력/출력 커맨드들을 전송할 때, 동기 응용 프로그램들은 입력/출력 커맨드가 완료될 때까지 대기하며, 통신이 진행되는 동안 응용 프로그램의 진행을 차단하고 시스템 리소스를 유향 상태로 둘 수 있다. 대조적으로 비동기 응용 프로그램은 입력/출력 커맨드들을 전송할 수 있으며 전송이 완료되기 전에 다른 처리를 수행할 수 있다. 이러한 사실들은, 비동기 응용 프로그램이 입력/출력 커맨드의 레이턴시에 의해 덜 영향을 받는 경향이 있지만 그러한 개선은 비용을 야기시킴을 의미한다 : 비동기 응용 프로그램은 응답이 전송될 시점을 알지 못하므로 언제든지 응답을 수신할 수 있어야 한다.
- [0043] 티어 속도에 대한 VM 민감도는 다르다. 동기식 응용 프로그램이 있는 VM은 비동기식 응용 프로그램이 있는 VM보다 티어 속도 변화에 더 민감하다. 주된 이유는 비동기 응용 프로그램이 일반적으로 플러시(flush) 할 입력/출력 커맨드의 빈도가 적고 입력/출력 타이밍에 대한 의존도가 낮기 때문이다.
- [0044] 이러한 관측들에 의해 동기 부여 된 "TSSCS(tier speed sensitivity calibration session)"는 실제 마이그레이션 없이 VM을 상위 (또는, 하위) 속도의 티어로 마이그레이션함으로써 VM이 얼마나 많은 양을 얻게 될지 (또는 불이익을 받을 것인지) 예측하기 위해 사용될 수 있다. 상세히는, TSSCS는 다음과 같은 특성을 갖는다 :
- [0045] 1) 경량 : 오토 티어링 입력/출력 필터에서 실행되는 TSSCS는 각 VM에 대한 입력/출력 커맨드들에 레이턴시를 주입하며, 현재 호스트 워크로드에 영향을 미치지 않도록 매우 짧은 시간이 소요된다.
- [0046] 2) TSSCS 당 레이턴시 당 다중 샘플들 : 각 티어에서 각 VM의 성능에 대한 에뮬레이션 정확도를 향상시키려면 각 티어의 동일한 주입된 레이턴시로 획득된 결과들이 평균될 수 있다.
- [0047] 3) TSSCS 당 다중 레이턴시 : 회귀 분석을 개선하기 위해 여러 가지 레이턴시들이 주입될 수 있다. 또한, 주입된 레이턴시들은 현재 스토리지 풀의 사용 가능한 티어들에만 국한될 필요는 없다. 대신, 회귀 정확도를 높이기 위해 추가적인 레이턴시 데이터 포인트들이 사용될 수 있다.
- [0048] 4) 런타임 동안 다중 TSSCS : 곡선 및 회귀 함수를 업데이트하기 위해 TSSCS가 주기적으로 트리거될 수 있다.
- [0049] 대응하는 성능을 기반으로, 성능 이득 대 티어 속도 사이의 상관 관계는 디스크 속도 향상 또는 감소에 대한 각 VM의 민감도를 반영하는 선형 회귀에 의해 결정될 수 있다. 곡선이 다항식 또는 비선형 함수로 회귀될 수 있지만 회귀를 변경해도 오토 티어링이 동작하는 방식에는 영향을 미치지 않는다. 현재 순간의 각 VM에 대해 주입된 레이턴시 곡선이 주어지면 각 티어의 각 VM에 대한 IOPS, 처리량, 스토리지 크기 등이 계산될 수 있다. IOPS, 처리량, 및 스토리지 크기가 중점을 두고 있지만 다른 타입의 리소스도 고려될 수 있다.
- [0050] 표4는 하위 섹션에서 사용된 표기법을 요약하고 표5는 세 가지 다른 티어들에서 세 개의 VM을 사용하는 예에 대한 예상 평균 입력/출력 레이턴시의 결과를 나타낸다.

표 4

표기	의미
L_t	티어 t 의 원래의 평균 입력/출력 레이턴시(주입된 레이턴시 없이)
b_v, m_v	VM v 의 회귀된 티어 속도 민감도 상관 모델, 즉, $y = mx + b$ 에 대한 두 파라미터들, y 는 평균 입력/출력 레이턴시 및 x 는 주입된 레이턴시.
s_v	VM v 의 평균 I/O 크기.
$S_v, VM[v].size$	VM v 의 스토리지 크기.
$maxIOPS[t], maxMBPS[t], maxSize[t]$	티어 t 의 사용 가능한 리소스의 캡들을 미리 설정, 여기서 리소스의 각 타입의 총 용량의 90%가 되도록 설정.

<표기들>

표 5

	티어1	티어2	티어3
V1 Avg I/O Latency (μs)	b_1	$m_1(L_2 - L_1) + b_1$	$m_1(L_3 - L_1) + b_1$
V2 Avg I/O Latency (μs)	$m_2(L_1 - L_2) + b_2$	b_2	$m_2(L_3 - L_2) + b_2$
V3 Avg I/O Latency (μs)	$m_3(L_1 - L_3) + b_3$	$m_3(L_2 - L_3) + b_3$	b_3

<마이크레이션 없이 평균 입력/출력 레이턴시들을 예측한 예>

TSSCS(Tier speed sensitivity calibration session)은 스토리지 시스템의 각 VM 및 각 티어에 대해 반복될 수 있다. 그것은 주입된 각 레이턴시에 대한 샘플로부터의 입력/출력 레이턴시들의 계수 변화(즉, CV)를 계산할 수 있다. 그런 다음 주입된 각 레이턴시에 대한 입력/출력 레이턴시의 평균값을 계산할 수 있다. 그런 다음 주입된 모든 레이턴시의 CV 값을 평균화하고, 결과에 근거하여, 오토 티어링은 각 VM에 대한 선형 회귀를 생성하는 데 사용된 주입된 레이턴시에 대한 신뢰도를 정의하고 이에 따라 선형 회귀 자체에 대한 신뢰도를 정의할 수 있다. CV가 낮을수록 이 VM에 대한 신뢰도가 높아진다. CV가 1보다 큰 경우, 신뢰도는 하한(예를 들어, 0.05)으로 설정될 수 있으며, 하한은 시스템 관리자의 요구 사항에 따라 조정될 수 있다. 마지막으로, TSSCS는 대응하는 m 및 b 값을 계산하기 위해 회귀 함수를 호출할 수 있다. 본 발명의 실시 예는 리턴된 튜플(tuple)에서 단지 두 개의 파라미터들을 갖는 선형 회귀 모델을 사용할 수 있지만, 시스템 관리자는 선택된 회귀 모델에 기초하여 리턴된 튜플에 더 많은 항목들을 추가할 수 있다.

오토 티어링은 IOPS(초당 입력/출력 커맨드), 처리량(초당 메가 바이트 또는 MBPS), 및 스토리지 크기(바이트)와 같은 세 가지 타입의 리소스를 기록하는 세 개의 2 차원 매트릭스를 가질 수 있다. 스토리지 시스템의 다른 타입의 리소스에 대한 추가 매트릭스가 원하는대로 추가될 수 있다. IOPS 및 처리량과 달리, 스토리지 크기에 대한 매트릭스는 각 VM이 차지하는 스토리지 크기를 측정하기 위해 하이퍼바이저 API를 호출하여 획득하는 것이 상대적으로 간단하다.

오토 티어링은 회귀된 M 및 B 값을 기반으로 하는 선형 예측을 사용하여 IOPS를 예측할 수 있다. 예상 평균 입력/출력 레이턴시가 대응하는 IOPS를 계산하고 다른 티어의 각 VM의 처리량이 계산되면 IOPS 및 처리량은 다음 방정식을 사용하여 계산될 수 있다.

$$IOPS = \frac{avgIOSize(v) \times 10^6}{avgIOLatency(v,t)} \quad (7)$$

$$Throughput = avgIOSize \times IOPS \quad (8)$$

서로 다른 타입의 리소스는 서로 다른 단위를 가지고 있기 때문에, 하나의 다차원 매트릭스에 이들을 조합하는 것은 어려울 수 있다. 따라서 각 리소스는 각 티어의 사용 가능한 총 리소스 용량의 백분율로 변환될 수 있다. 이러한 정규화의 예가 표 6에 나타난다.

표 6

	IOPS	처리량	스토리지
VM1	10^6 $b_1 \times maxIOPS[t]$	$s_1 \times 10^6$ $b_1 \times maxMBPS[t]$	S_1 $maxSize[t]$
VM2	10^6 $[m_2(L_1 - L_2) + b_2] \times maxIOPS[t]$	$S_2 \times 10^6$ $[m_2(L_1 - L_2) + b_2] \times maxMBPS[t]$	S_2 $maxSize[t]$
VM3	10^6 $[m_3(L_1 - L_3) + b_3] \times maxIOPS[t]$	$s_3 \times 10^6$ $[m_3(L_1 - L_3) + b_3] \times maxMBPS[t]$	S_3 $maxSize[t]$

<3개의 VM 들이 티어1을 실행 중인 경우의 정규화된 성능 매트릭스의 예>

계산된 점수는 다음 사실들을 반영해야 한다.

1) 티어와 VM의 특성 : 각 티어에서 실행 중인 경우 각 티어의 특성과 각 VM의 워크로드 특성을 점수가 반영해야 한다. 따라서 각 티어에서 각 VM의 점수는 개별적으로 계산될 수 있다.

2) SLA 가중치 : 식1에 나타나는 바와 같이, VM들은 SLA 가중치가 다르기 때문에 같지 않다.

3) 예측 신뢰도 : 성능 매트릭스에서 계산된 계수 변화는 예측의 신뢰를 반영하기 위해 사용될 수 있다.

4) 히스토리 및 마이그레이션 비용 : 컨볼루션 에이징 팩터는 히스토리 점수를 점수 계산에 포함시키는데 사용될 수 있다. 또한 마이그레이션 비용을 산정하여 점수 계산에 포함시킬 수도 있다.

사용자 컨피규레이션 또는 기본 컨피규레이션을 기반으로, 각 티어에서 제공되는 리소스는 미리 알려져 있다. 오토 티어링은 주어진 티어에 대한 리소스들의 임의의 개수에 대해 최적화를 수행할 수 있다. 전문성을 반영하기 위해, 티어 전문성 매트릭스가 사용될 수 있다. 예시적인 티어 전문성 매트릭스는 표 7에 도시되어있다. 표 7에서, 전문성 컨피규레이션에 대한 전부 또는 일부의 입상도(granularity)가 사용된다. 1이 있는 셀은 현재 티어에 해당 타입의 리소스에 대한 전문성이 있음을 의미한다. 그렇지 않으면 해당 타입의 리소스에 대해 티어가 최적화되지 않는다. 그러나 본 발명의 개념의 다른 실시 예에서, 이 매트릭스의 값은 반드시 0 및 1 일 필요는 없다. 시스템 관리자는 전문성을 제어하기 위해, 보다 세부적인 입상도(granularity)를 원하면 0 및 1 대신에 임의의 가중치를 설정할 수 있다. 시스템 관리자는 더 많은 타입의 리소스(즉, 더 많은 행)를 추가하여 이러한 매트릭스를 확장할 수도 있다.

표 7

	IOPS	Throughput (MBPS)	Storage Size (bytes)
Tier 1	1	1	0
Tier 2	0	1	0
Tier 3	0	0	1

<티어 전문성 매트릭스의 예>

티어들이 기본적으로, 가장 비싼 티어에서 가장 싼 티어의 순서로 정렬되어 있으면 유용하고 이 순서가 마이그레이션 결정을 내리는 기간 동안에 우선 순위로 간주될 수 있다. 시스템 관리자는 각 티어의 선호도에 근거하여 이 순서를 변경할 수도 있다.

다음 질문은 "각 티어의 전문성을 반영하면서 각 티어에 각 VM의 성능을 반영하는 방법"이다. "성능"이라는 용어는 여러 차원 또는 리소스의 타입 (IOPS, 처리량 및 스토리지 크기)을 고려하기 때문에, 각 티어의 전문성을

반영하기 위해 가중합이 적용될 수 있다. "전문성(specialty)" 매트릭스 및 VM 용량 이용률 매트릭스의 per-VM-per-tier 곱셈 연산인 "직교 매치(orthogonal match)"가 식 9 에 나타나는 바와 같이 사용될 수 있다 :

$$\begin{bmatrix} VMCapRateMat[t][v].IOPS \\ VMCapRateMat[t][v].MBPS \\ VMCapRateMat[t][v].size \end{bmatrix} \times [specialty[t].IOPS, specialty[t].MBPS, specialty[t].size] \quad (9)$$

또한, 오토 티어링은 시스템 관리자가 각 타입의 리소스에 대해 서로 다른 가중치를 구성할 수 있게 한다.

티어1에 대한 용량 점수를 계산하는 예가 표8에 나타난다. 첫 번째 행("capacityRate")의 (IOPS%, MBPS%, 크기%)의 튜플은 이 티어의 각 리소스 타입에 대한 예상 VM 사용률이다. 이러한 예에서, 이들 값 중 어느 것도 1보다 크지 않으므로, 세 개 모두가 제한 조건 테스트를 통과할 수 있다. 마지막으로 전문성 매트릭스가 표 7에 도식한 것과 같다고 가정한다.

표 8

	VM1	VM2	VM3
capacityRate	(0.5, 0.6, 0.3)	(0.2, 0.3, 0.3)	(0.1, 0.5, 0.4)
capacityConstraint	Pass	Pass	Pass
currCapScore	$(0.5 + 0.6)/2 = 0.55$	$(0.2 + 0.3)/2 = 0.25$	$(0.1 + 0.5)/2 = 0.30$

<3개의 VM 들이 티어1을 실행 중인 경우, 3개의 VM들의 용량 비율, 제약 조건, 및 용량 점수 값들>

최종 스코어는 식10에 나타나는 바와 같이, 히스토리 스코어, 현재 스코어, 및 마이그레이션 비용의 컨벌루션 합이다 :

$$totalScore = agingFactor \times historicalScore + currCapacityScore - weightMig \times migrationCost \quad (10)$$

이 최종 점수에는 세 가지 구성 요소가 있다.

1) historicalScore : 입력/출력 스파이크(spike) 또는 버스트(burst)를 포함할 수 있는 최근 에포크를 기반으로 한 의사 결정으로 일부 VM이 두 티어간에 자주 마이그레이션되고 처리되는 것을 방지하기 위해, 오토 티어링은 사전 설정된 agingFactor와 함께 historicalScore를 고려하여 오래된 점수들을 페이드 아웃시킬 수 있다. 초기에는, 특정 티어에 현재 할당되지 않은 VM은 0 점을 받는다.

2) currCapScore : 현재 용량 스코어는 식 (9)와 표8을 참조하여 상술한 바와 같이 계산된다. 다시, 특정 티어에 할당되지 않은 VM은 모두 0 점을 받는다.

3) migrationCost : 오토 티어링은 마이그레이션-아웃과 마이그레이션-인 가용 처리량의 작은 값을 선택함으로써 마이그레이션 속도의 병목 현상을 계산한다. 오토 티어링은 마이그레이션 속도로 나눈 VM 크기의 결과를 마이그레이션 시간 비용으로 반환할 수 있다.

각 티어에 대해 점수를 계산하기 위해, 오토 티어링은 VM들을 호스팅하는 것을 반복하고 현재 VM이 현재 티어 t의 리소스들의 모든 타입의 상한을 초과하는지 확인할 수 있다. VM가 리소스 용량 상한을 초과하면 해당 VM 용량 사용률이 0으로 설정되고 티어 t의 VM v의 점수가 -1로 설정된다. 그렇지 않으면 리소스 사용이 퍼센트 용량으로 변환된다. 총 점수는 위와 같이 계산된다. 오토 티어링을 사용하면, 시스템 관리자가 리소스의 다양한 가중치를 조정함으로써 가중치들과 각 타입의 리소스를 제어할 수 있다.

오토 티어링은 "장기적인 관점"에서 마이그레이션을 수행한다. 즉, 마이그레이션 작업이 비교적 비용이 비싸기 때문에 오토 티어링은 마이그레이션 빈도가 비교적 적다.

오토 티어링은 마이그레이션 작업이 필요한지 여부를 확인하고 필요한 경우 해당하는 마이그레이션을 트리거할 수 있다. 오토 티어링은 각 티어에 대해 반복 수행 할 수 있으며 가장 바람직하게는 가장 비싼 티어로부터 가장 싼 티어의 순서로 반복할 수 있다. 시스템 관리자는 각 티어의 선호도에 따라 이러한 순서를 변경할 수 있다. 오토 티어링은 티어 점수를 내림차순으로 정렬한다. 이러한 정렬 목록에 있는 각 VM에 대해 오토 티어링은 다음과 같은 경우 VM을 스킵할 수 있다.

1) 현재 반복된 VM이 이번 타입에 이미 할당(마이그레이션)되었거나

- [0089] 2) 이 티어의 용량 제한을 초과하므로 이 티어의 이 VM에 대한 점수가 없다.
- [0090] 3) 할당하는 동안, 이 VM v에 대해 티어 t에 남아있는 리소스 용량이 충분하지 않다.
- [0091] 그렇지 않은 경우, 오토 티어링은 가장 많은 점수의 VM에서부터 가장 작은 점수의 VM들을 이 티어에 하나씩 할당할 수 있다. VM이 이미 타겟 티어에 있는 경우, 이 할당은 VM을 이동시키지 않는다. VM이 다른 티어에 할당되면, VM에 마이그레이션을 위한 플래그가 지정될 수 있다. 여기에는 데이터(VMDK 파일)를 원본 스토리지 티어에서 타겟 스토리지 티어로 복사하는 작업이 포함된다.
- [0092] 기존 데이터 센터는 HDD를 사용하여 데이터를 저장한다. HDD는 일반적으로 서로 많은 면에서 유사하나 주로 기가 바이트 당 비용, 읽기/쓰기 속도 및 전체 용량이 다르다. 예측되는 바와 같이, 용량이 클수록 읽기/쓰기 속도가 빠를수록 HDD가 가장 비싸다. 그러나 이러한 차이를 고려하더라도 HDD는 기가 바이트 당 평균 비용이 0.25 달러이며 평균 읽기/쓰기 속도는 10 밀리 초이다. 그러나 컴퓨터 작업에서 10 밀리 초는 대기하는 데 오랜 시간이 될 수 있으므로 일반적으로 액세스 시간을 향상시키기 위해 일부 데이터가 로컬로 캐시될 수 있다.
- [0093] 반면에, SSD는 다양한 전문성을 제공할 수 있다. 일부 SSD는 낮은 읽기/쓰기 레이턴시를 가지며, 다른 SSD는 높은 대역폭, 큰 용량, 낮은 쓰기 증폭 인자(write amplification factor)를 가질 수 있다. 본 명세서에서, "리소스"이라는 용어는 개별 스토리지가 더 크게 또는 더 작게 제공할 수도 있는 전문성을 설명하기 위해 사용된다. 또 다른 SSD는 이러한 리소스들의 조합을 제공할 수 있다. 이러한 추가적인 전문성은 이러한 리소스를 활용할 수 있는 응용 프로그램에 이점을 제공할 수 있지만, 특정 응용 프로그램에 가장 적합한 SSD를 결정하는 프로세스를 복잡하게 만들 수 있다.
- [0094] HDD와 달리 SSD는 일반적으로 포맷에 따라 기가 바이트 당 가격이 \$ 0.28 ~ \$ 4.68에 이르는 등 보다 비싸다. 그러나 SSD는 일반적으로 훨씬 더 즉각적으로 반응할 수 있다: 읽기/쓰기 시간은 사용되는 SSD 포맷에 따라 1-2ms에서 마이크로 초(μ s) 또는 심지어 나노 초(ns)까지 다양하다.
- [0095] HDD를 통해 SSD가 제공하는 추가 리소스는 SSD에 응용 프로그램을 가장 잘 할당하기 위한 방법에 있어서 중요한 요소이지만, 응용 프로그램을 SSD에 할당하는 방법에 가장 큰 영향을 미치는 것은 액세스 시간이다. 기존의 캐싱 알고리즘은 스토리지 레이어들 간에 성능이 상당히 향상 될 것으로 가정한다. 일반적으로, HDD 스토리지와 비교하여 로컬 캐시 레이어로부터 10 배의 성능 향상이 기대된다. 또한, SSD는 HDD보다 데이터를 저장하는 데 기가 바이트 당 더 많은 비용이 소요되므로 데이터를 캐싱하면 약간의 비용으로 데이터가 두 번 저장될 수 있다. 따라서, 기존의 티어링 및 캐싱 알고리즘은 HDD가 아닌 스토리지 디바이스를 사용할 때 차선의 결과를 생성한다.
- [0096] 도 1은 본 발명의 실시 예에 따라, 가상 머신을 호스팅하고 오토-티어링을 수행하도록 구비된 스토리지 풀을 사용할 수 있는 네트워크화된 머신들을 나타낸다. 도1을 참조하면, 시스템(105)이 도시된다. 시스템(105)은 가상 머신 서버들(110-1, 110-2, 및 110-3)을 포함할 수 있다. 한편, 도 1은 세 개의 서버들을 도시하지만, 본 발명의 실시 예는 임의의 수의 가상 머신 서버들을 지원할 수 있다.
- [0097] 각각의 가상 머신 서버들(110-1, 110-2, 110-3)은 프로세서들(115-1, 115-2, 115-3) 및 메모리들(120-1, 120-2, 120-3)을 각각 포함할 수 있다. 다른 용도 중에서, 프로세서들(115-1, 115-2 및 115-3)은 가상 머신 서버들(110-1, 110-2 및 110-3)에서 실행중인 가상 머신들에 의해 사용되는 애플리케이션과 같은 애플리케이션을 실행할 수 있으며, 메모리(120-1, 120-2, 120-3)에 저장될 수 있다. 프로세서들(115-1, 115-2, 및 115-3)은 임의의 종류의 프로세서, 예를 들어 Intel Xeon, Celeron, Itanium 또는 Atom 프로세서, AMD Opteron 프로세서, ARM 프로세서 등일 수 있다. 도 1은 각각의 가상 머신 서버들(110-1, 110-2 및 110-3)의 단일 프로세서를 도시하나, 가상 머신 서버들(110-1, 110-2 및 110-3)은 각각 다수의 프로세서를 포함할 수 있으며, 프로세서 각각은 단일 코어 또는 멀티 코어 프로세서 일 수 있으며 원하는 조합으로 혼합될 수 있다. 메모리들(120-1, 120-2, 및 120-3) 각각은 각각 플래시 메모리, Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), Persistent Random Access Memory, Ferroelectric Random Access Memory (FRAM), 또는 Magnetoresistive Random Access Memory(MRAM) 등과 같은 비-휘발성 랜덤 액세스 메모리(NVRAM)와 같은 다양한 메모리일 수 있다. 메모리들(120-1, 120-2, 및 120-3)은 각각 상이한 메모리 타입들의 임의의 원하는 조합일 수 있다. 메모리들(120-1, 120-2, 및 120-3)은 또한 가상 머신 서버들 (110-1, 110-2, 및 110-3)의 일부인 메모리 컨트롤러 (도 1에 도시되지 않음)에 의해 제어될 수 있다.
- [0098] 가상 머신 서버들(110-1, 110-2, 및 110-3)은 각각 로컬 스토리지 디바이스들을 또한 포함할 수 있다. 예를 들어, 가상 머신 서버들(110-1, 110-2 및 110-3)은 캐시들(125-1, 125-2 및 125-3)을 각각 포함할 수 있다. 캐시

들(125-1, 125-2, 및 125-3)와 같은 로컬 스토리지 디바이스들은 하드 디스크 드라이브(HDDs) 또는 솔리드 스테이트 드라이브(SSDs)와 같은 임의의 원하는 형태를 취할 수 있고, Peripheral Component Interconnect(PCI), PCI Express(PCIe), Serial AT Attachment(SATA), Non-Volatile Memory Express (NVMe), NVMe over Fabric(NVMeoF), M.2, 또는 기타 원하는 연결과 같은 임의의 원하는 커넥터를 사용할 수 있다. 본 발명의 개념의 일부 실시 예에서, 가상 머신 서버들(110-1, 110-2 및 110-3)은 데이터 센터의 저장에만 의존하는 캐시들(125-1, 125-2 및 125-3)을 생략할 수 있다. 캐시들(125-1, 125-2, 및 125-3)을 포함하는 본 발명의 개념들의 실시 예들에서, 본 발명의 개념들의 실시 예들은 미국 특허 출원 일련 번호 _____에 개시된 발명 개념들의 실시 예들과 결합될 수 있으며, 상기 특허는 모두 모든 목적을 위해 본원에 참고로 인용된 2017년 11월 29일자로 출원된 미국 특허 출원 제 62 / 592,355 호로부터의 우선권을 주장한다.

[0099] 시스템(105)은 또한 데이터 센터 서버(130)를 포함할 수 있다. 가상 머신 서버들(110-1, 110-2, 110-3)과 마찬가지로, 데이터 센터 서버(130)는 프로세서(115-4) 및 메모리(120-4)를 포함할 수 있다. 그러나 데이터 센터 서버(130)는 또한 스토리지 풀(storage pool, 135), 수신기(140), 및 송신기(145)를 포함할 수 있다. 스토리지 풀(135)은 스토리지 티어들로 구성된 스토리지 디바이스들의 풀일 수 있다. 본 발명의 개념의 일부 실시 예에서, 스토리지 풀(135)은 SSD 또는 다른 플래시 메모리 디바이스들만을 포함할 수 있다. 본 발명의 개념의 다른 실시 예에서, 스토리지 풀(135)은 다른 스토리지 포맷들을 사용하는 스토리지 디바이스들을 포함할 수 있다. 수신기(140)는 가상 머신 서버들(110-1, 110-2, 및 110-3)로부터 입력/출력 커맨드 및 다른 데이터를 수신할 수 있고, 송신기(145)는 가상 머신 서버들(110-1, 110-2, 및 110-3)로 데이터를 전송할 수 있다.

[0100] 가상 머신 서버들(110-1, 110-2, 110-3) 및 데이터 센터 서버(130)는 네트워크(150)를 통해 접속된 것으로 도시된다. 네트워크(150)를 사용하여, 가상 머신 서버들(110-1, 110-2, 110-3)은 데이터 센터 서버(130)에 입력/출력 커맨드를 전송하고, 데이터 센터 서버(130)로부터 응답을 수신할 수 있다. 가상 머신 서버들(110-1, 110-2 및 110-3) 및 데이터 센터 서버(130)는 모두 근거리 통신망(LAN)의 일부일 수 있거나, WAN(Wide Area Network) 또는 인터넷과 같은 글로벌 네트워크와 같은 다른 네트워크 타입을 포함할 수 있다. 네트워크(150)는 임의의 원하는 조합으로 다수의 그러한 네트워크들의 조합일 수 있다.

[0101] 또한, 도 1은 서버로서(또한 각각 독립형 또는 랙(rack) 서버일 수 있는) 가상 머신 서버들(110-1, 110-2, 및 110-3) 및 데이터 센터 서버(130)를 도시하지만, 본 발명의 개념은 어떠한 원하는 타입의 가상 머신 서버들(110-1, 110-2, 및 110-3) 및 데이터 센터 서버(130)를 제한 없이 포함할 수 있다. 예를 들어, 도 1에 도시된 서버들은 각각 데스크톱 또는 랩톱 컴퓨터 또는 본 발명의 개념의 실시 예로부터 이익을 얻을 수 있는 임의의 다른 장치로 각각 대체될 수 있다. 가상 머신 서버들(110-1, 110-2, 110-3) 및 데이터 센터 서버(130)는 각각 특수화된 휴대용 컴퓨팅 디바이스들, 태블릿 컴퓨터들, 스마트 폰들 및 다른 컴퓨팅 디바이스들을 또한 포함할 수 있다. 설명의 목적으로, 가상 머신 서버들(110-1, 110-2, 및 110-3)은 호스팅 가상 머신들로 설명되지만, 본 발명의 개념은 가상 머신 서버들(110-1, 110-2, 및 110-3)에서 실행중인 가상 머신들이 아닌 머신들을 포함할 수 있다.

[0102] 도 2는 도 1의 서버들(110-1, 110-2, 110-3, 및 130)의 추가적인 세부 사항을 도시한다. 도 2에서, 통상적으로, 서버(110)(도 1의 서버들(110-1, 105-2 또는 110-3) 중 어느 하나를 나타낼 수 있음) 및 서버(130)는 메모리 컨트롤러(205) 및 클럭(210)을 포함할 수 있는 하나 이상의 프로세서들(115)을 포함할 수 있으며, 프로세서들(115)은 서버들(110, 130)의 구성 요소들의 동작들을 조작하기 위해 사용될 수 있다. 프로세서들(115)은 예로서, 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM) 또는 다른 상태 보존 매체를 포함하는 메모리들(120)에 결합될 수 있다. 서버(130)에서, 프로세서(115)는 또한 스토리지 디바이스(135)에 결합될 수 있다(서버(110)는 또한 국부적으로 부착된 스토리지를 포함할 수 있지만, 필수적인 것은 아니다). 서버들(110 및 130)에서, 프로세서들(115)은 또한 예를 들어, 이더넷 커넥터 또는 무선 커넥터일 수 있는 네트워크 커넥터들(215)에 결합될 수 있다. 프로세서들(115)은 버스들(220)에 접속될 수 있으며, 다른 구성요소들 중에서 입력/출력 엔진들(230)을 사용하여 관리될 수 있는 사용자 인터페이스(225) 및 입력/출력 인터페이스 포트들은 버스들(220)에 부착될 수 있다.

[0103] 도 3은 본 발명의 실시 예에 따라, 도 1의 스토리지 풀(135)과 오토-티어링으로 상호 작용하는 도 1의 가상 머신 서버(110)의 부가적인 세부 사항을 도시한다. 도 3에서, 가상 머신 서버(110)는 3개의 가상 머신들(305-1, 305-2 및 305-3)을 호스팅하는 것으로 도시되어 있지만, 본 발명의 개념은 가상 머신 서버(110)에 의해 호스팅되는 임의의 수의 가상 머신들을 포함할 수 있다. 가상 머신(305-3)만이 이하에서 더 상세히 설명되지만, 본 발명의 개념의 실시 예는 유사하게 동작하는 가상 머신(305-1 및 305-2)과 같은 다른 가상 머신들을 포함할 수 있다. 가상 머신(305-3)은 운영 체제(315)에서 동작하고 있는 애플리케이션(310)을 포함하는 것으로 도시된다. 하

하이퍼바이저(320)는 가상 머신 서버(110) 상에서 동작하는 오퍼레이션 가상 머신들(305-1, 305-2, 및 305-3)을 관리 할 수 있다.

[0104] 입력/출력 필터(325)는 애플리케이션(310)으로부터 커맨드를 수신하고 실행을 위해 이들을 하이퍼바이저(320)에 전송하는 통과 필터로서 동작할 수 있다. 입력/출력 필터(325)는 입력/출력 필터링 용 VMware vSphere API와 같은 임의의 원하는 입력/출력 필터일 수 있다. 그러나 입력/출력 필터(325)는 일반적인 동작 이외에, 가상 머신(305-3)으로부터 오는 일부 입력/출력 커맨드들에 레이턴시를 주입할 수 있으며, 이는 다른 스토리지 티어들에 의해 지원될 때 VM 성능을 예측하는 방법에 대한 정보를 제공할 수 있다. 입력/출력 필터(325)의 동작은 도7을 참조하여 더 설명된다.

[0105] 오토 티어링 데몬(330)은 입력/출력 필터(325)로부터 성능 데이터를 수신하여 오토 티어링 컨트롤러(335)로 전송할 수 있다. 오토 티어링 데몬(330)은 하이퍼바이저(320)의 일부이거나 별개의 구성 요소일 수 있다. 또한, 오토 티어링 데몬(330)은 입력/출력 필터(325)로부터 데이터를 수신하여 이를 오토 티어링 컨트롤러(335)로 전송하는 단순한 통과 장치일 수도 있지만, 가상 머신들(305-1, 305-2, 및 305-3)에 관한 성능 데이터에 대한 선행 회귀를 수행하는 동작과 같이 오토-티어링 컨트롤러(335)에게 추후 제공될 몇몇 기능들을 포함할 수 있다. 본 발명의 개념의 일부 실시 예에서, 오토-티어링 데몬(330)가 성능 데이터를 수집하고 전송하는 것을 허용하기 보다는 입력/출력 필터(325)가 오토-티어링 컨트롤러(335)와 직접 통신함으로써, 오토-티어링 데몬(330)은 완전히 생략될 수 있다.

[0106] 오토-티어링 컨트롤러(335)는 오토-티어링 데몬(330)으로부터 수신된 성능 데이터를 취하여 그 데이터를 사용하여 가상 머신(305-3)이 각각의 스토리지 티어에서 어떻게 수행하는지를 예측하고, 그 후에 오토-티어링 컨트롤러(335)가 가상 머신(305-3)에 대한 데이터를 다른 스토리지 티어로 적절하게 마이그레이션할 수 있다. 오토-티어링 컨트롤러(335)에 의해 관리되는 스토리지 풀(135)은 임의의 개수의 스토리지 티어들을 포함할 수 있다. 도 3은 세 개의 티어들(340-1, 340-2, 및 340-3)을 도시한다. 각각의 티어 내에는 임의의 개수의 스토리지 디바이스들(345)이 있을 수 있다.

[0107] 위에서 언급한바와 같이, 각 스토리지 티어는 해당 스토리지 티어에서 제공하는 특정 리소스를 활용하는 특성화 기능을 제공할 수 있다. 예를 들어, 스토리지 티어(340-1)는 빠른 읽기/쓰기 시간을 제공하지만, 스토리지 티어(340-3)는 낮은 쓰기 증폭 인자를 제공할 수 있다. 스토리지 티어들(340-1, 340-2, 및 340-3)이 특성화를 제공하기 위해서는 스토리지 티어 내의 스토리지 디바이스들(345)이 모두 특성화를 제공해야 한다. 따라서, 본 발명의 개념의 일부 실시 예에서, 주어진 스토리지 티어 내의 스토리지 디바이스(345)는 하나 이상의 전문 분야에 대해 상대적으로 또는 완전히 균일(homogeneous)하다: 어떤 경우에는 주어진 스토리지 티어 내의 스토리지 디바이스들(345)은 모든 스토리지 디바이스(345)가 동일한 스토리지 디바이스 모델이어야 한다.

[0108] 본 발명의 개념의 일부 실시 예에서, 오토-티어링 컨트롤러(335)는 도 1의 데이터 센터 서버(130)의 일부로서 구현될 수 있다. 본 발명 개념의 다른 실시 예에서, 오토-티어링 컨트롤러(335)는 가상 머신으로서 구현될 수 있으며, 가상 머신 서버들(110-1, 110-2, 및 110-3) 중 하나에서 실행될 수 있다.

[0109] 도 4는 상이한 스토리지 디바이스들에 의해 제공되는 가용한 상이한 리소스들을 도시한다. 도4에는 리소스(405)들이 도시되어있다. 리소스(405)는 1 초와 같은 주어진 시간에 수행되는 입력/출력 커맨드들의 개수(일반적으로 초당 입력/출력 동작으로 기술되는, 또는 IOPS)를 나타내는 처리량(410), 주어진 시간 단위에 얼마나 많은 데이터가 전송될 수 있는지를 나타내는 대역폭(415)(일반적으로, 초 당 메가바이트 단위로 기술되는, 또는 MBPS), 저장되는 데이터의 양을 반영하는 용량(420), 및 얼마나 많은 추가적인 데이터가 가비지 컬렉션 동작의 결과로서 기록되는지를 나타내는 쓰기 증폭 인자(425)를 포함할 수 있다. 리소스들(405)은 도1의 스토리지 풀(135)의 스토리지 디바이스들에 의해 제공되는 임의의 다른 원하는 리소스를 포함하도록 쉽게 확장될 수 있다.

[0110] 도5는 하나의 스토리지에서 다른 스토리지로 마이그레이션되는 도3의 가상 머신 파일들을 나타낸다. 도5에서, 가상 머신 파일(505)은 가상 머신에 대한 데이터를 저장한다. 가상 머신 파일(505)은 현재 스토리지 티어(340-3) 상에 상주하고, 제2가상 머신을 위한 가상 머신 파일(510)은 현재 스토리지 티어(340-2) 상에 상주한다. 화살표(515)에 의해 도시된 바와 같이, 가상 머신 파일(505)은, 도3의 오토-티어링 컨트롤러(335)가 어떠한 마이그레이션이 도 1의 시스템(105) 전체에게 최고의 향상을 가져오느냐를 고려함으로써, 스토리지 티어(340-2 또는 340-1) 중 하나로 마이그레이션될 수 있다. 도3의 오토-티어링 컨트롤러(335)가 가상 머신 파일(510)이 스토리지 티어(340-2) 상에서 마이그레이션되지 않은 채 남아 있다는 사실에 의해 도시된 바와 같이, 가상 머신을 또 다른 스토리지 티어로 마이그레이션하지 않기로 결정할 수 있다.

- [0111] 변하지 않는 스토리지 tier에 가상 머신을 할당하는 방법에 대한 최적의 솔루션이 있는 것이 좋지만(적어도 가상 머신들의 집합이 변경되지 않는 한), 실제로 가상 머신들은 워크로드의 변화를 경험할 수 있으며, 시간이 지남에 따라 요구 사항이 다를 수 있는 다른 VM들과 리소스에 대해 경쟁할 수도 있다. 이러한 워크로드의 변화 및 경쟁은 특정 가상 머신에 대한 가장 적합한 스토리지 tier에 영향을 줄 수 있다. 예를 들어, 임의의 시점에서, 하나의 가상 머신은 매우 바쁠 수 있고(busy 상태) 도 1의 스토리지 풀(135)로부터 빠른 응답 시간을 요구할 수 있다. 하지만 다른 시점에서, 동일한 가상 머신이 활동이 적어 고속 스토리지 tier를 필요로 하지 않을 수도 있다. 따라서 항상 모든 가상 머신에 가장 적합한 단일한 글로벌 최적화는 존재하지 않을 수 있다.
- [0112] 따라서 단일한 글로벌 최적화를 찾는 대신 일정 기간 동안 작동하는 최적화를 찾는 것이 더 좋을 수 있다. 그 후에는 다른 최적화가 바람직할 수 있다. 예를 들어, 도 6a 및 도 6b는 상이한 시점에서의 스토리지 tier들에서의 가상 머신들의 최적의 할당들을 도시한다. 예를 들어, 도 6A에 도시된 바와 같이, 시간 T_0 에서, 가상 머신 파일(505)은 스토리지 tier(340-3)에 최적이고 가상 머신 파일(510)은 스토리지 tier(340-1)에 최적이다. 대조적으로, 도 6b에 도시된 바와 같이, 시간 T_1 에서, 가상 머신 파일들(505 및 510)은 모두 스토리지 tier(340-1)에 최적이다.
- [0113] 도 7은 도 3의 가상 머신에 대한 레이턴시 주입 및 성능 데이터 결정하는 도 3의 입력/출력 필터(325)를 도시한다. 도 7에서, 입력/출력 필터(325)는 도 3의 가상 머신(305-3)의 애플리케이션(310)으로부터 입력/출력 커맨드(705)를 수신할 수 있다. 입력/출력 필터(325)는 입력/출력 커맨드(705)에 레이턴시(710)를 주입할 수 있다. 주입된 레이턴시는 가상 머신에 대해 상이한 평균 입력/출력 레이턴시를 생성하는 저 성능 스토리지 tier를 시뮬레이션할 수 있다.
- [0114] 주입된 레이턴시(710)를 사용함으로써, 입력/출력 필터(325)는 입력/출력 커맨드(705)를 데이터 센터 서버(130)로 전달할 수 있다. 데이터 센터 서버(130)는 입력/출력 커맨드(705)를 처리하고 응답(715)을 반환할 수 있다. 입력/출력 필터(325)는 응답(715)을 애플리케이션(310)에 반환하고 주입된 레이턴시(710)를 사용하여 도 3의 가상 머신(305-3)의 평균 입력/출력 레이턴시를 측정하고, 성능 데이터(720)를 생성할 수 있다. 도 3의 오토-tier링 컨트롤러(335)는 성능 데이터(720)를 사용하여, 가상 머신을 다른 스토리지 tier로 마이그레이션하는 이점이 있는지 여부를 판별할 수 있다(더 빠르거나 더 느리거나, 가상 머신을 더 느린 스토리지 tier로 마이그레이션하면 성능의 보상 개선을 갖는 또 다른 가상 머신을 위한 공간을 확보할 수 있으므로).
- [0115] 한편, 도 7은 성능 데이터(710)를 생성하기 위해 주입된 레이턴시(710)를 사용하는 입력/출력 필터(325)를 도시하고, 주입된 레이턴시(710)는 자체 오버 헤드를 갖는다. 주입된 레이턴시(710)는 성능 데이터(710)가 생성될 수 있다면 원하는 경량 접근법으로 대체될 수 있다. 예를 들어, 입력/출력 필터(325)는 가상 머신 별로 아직 처리되지 않은 입력/출력 커맨드들의 개수를 추적할 수 있고, 그 정보를 사용하여 아직 처리되지 않은 입력/출력 커맨드들의 개수가 계속 높거나 계속하여 증가하면 다른 스토리지 tier로 가상 머신을 마이그레이션하는 것을 제안할 수 있다.
- [0116] 도 8은 도 3의 오토-tier링 컨트롤러(335)의 세부 사항을 도시한다. 도 8에서, 오토-tier링 컨트롤러(335)는 선형 회귀 모듈(805), 리소스 매트릭스 생성기(810), 메트릭 컨버터(metric converter, 815), 전문성 정보(820), tier 선택 모듈(825), 및 마이그레이션 모듈(830)을 포함할 수 있다. 선형 회귀 모듈(805)은 도 7의 성능 데이터(720)를 취하고, 데이터에 선형 회귀를 수행하여 주어진 스토리지 tier의 레이턴시에 대한 가상 머신의 평균 입력/출력 레이턴시를 예측하는 선형 방정식을 생성한다. 도 10을 참조하여 후술하는 바와 같이, 주어진 주입된 레이턴시들에 대해, 주어진 가상 머신의 평균 입력/출력 레이턴시들이 결정될 수 있으며 이러한 데이터 포인트들은 그래프로 그려질 수 있다. 본 발명의 개념의 일부 실시 예에서, 이들 데이터 포인트들은 대략적으로 선형이며, 선형 회귀를 수행함으로써 주어진 주입된 레이턴시에 대한 평균 입력/출력 레이턴시를 근사화하는 방정식을 생성할 수 있다. 그런 다음 각 스토리지 tier의 레이턴시를 파악하면, 각 스토리지 tier들에 대해 주어진 가상 머신의 평균 입력/출력 레이턴시를 예측할 수 있다.
- [0117] 리소스 매트릭스 생성기(810)는 각각의 가상 머신을 각 스토리지 tier 및 각 스토리지 tier에 의해 제공되는 리소스와 상관(correlate)시키기 위해 선형 회귀로부터 생성될 수 있는 예측치를 취할 수 있다. 예를 들어, 처리량은 가상 머신의 평균 입력/출력 크기와 가상 머신의 평균 입력/출력 레이턴시로부터 계산될 수 있으며, 대역폭은 VM로부터의 입력/출력 커맨드들에 대한 평균 입력/출력 크기 및 VM에 의해 발행된 IOPS로부터 계산될 수 있다. 리소스 매트릭스 생성기(810)는 메트릭 컨버터(815)를 사용하여 처리량과 같은 하나의 메트릭에 대한 성능 데이터를 대역폭과 같은 다른 메트릭에 대한 성능 데이터로 변환할 수 있다.

- [0118] 전문성 정보(820)는 각 스토리지 tier에 의해 제공되는 리소스를 나타낼 수 있다. 전문성 정보는 도11을 참조하여 아래에서 더 논의된다.
- [0119] tier 선택 모듈(825)은 리소스 매트릭스 생성기(810)에 의해 생성된 리소스 매트릭스 및 전문성 정보(820)를 취할 수 있고, 가상 머신들을 위한 스토리지 tier들을 선택할 수 있다. 그런 다음, 가상 머신이 하나의 스토리지 tier에서 다른 스토리지 tier로 마이그레이션되는 경우, 마이그레이션 모듈 (830)은 해당 마이그레이션을 수행할 수 있다.
- [0120] 도9는 도7의 성능 데이터(720)를 가상 머신들, 스토리지 tier들, 및 스토리지 tier들의 리소스들을 상관시키는 매트릭스로 변환하는 것을 도시한다. 도 9에서, 성능 데이터(720)는 주어진 스토리지 tier를 사용할 때 각각의 가상 머신에 대한 평균 입력/출력 레이턴시를 나타내는 매트릭스로서 도시된다. 따라서, 예를 들어, 가상 머신 1은 스토리지 tier 1을 사용하는 경우 800 μ s, 스토리지 tier 2를 사용하는 경우 500 μ s, 스토리지 tier 3을 사용할 경우 300 μ s의 평균 레이턴시를 가지는 것으로 예측되며, 가상 머신 2 및 3에 대한 성능 데이터는 유사한 정보를 제공할 수 있다. 도8에 도시된 번호는 단순한 예일 뿐이며 실제 성능은 특정 가상 머신에 의해 실행되는 어플리케이션과 동일한 스토리지 풀로부터 데이터에 액세스하는 다른 가상 머신의 영향에 크게 좌우되므로 스토리지 tier들의 모든 가상 머신들의 실제 성능 값을 반드시 반영하지는 않는다.
- [0121] 성능 데이터(720)의 일부 값은 예측치가 아니라 실제 값이다. 특히, 각각의 가상 머신은 자신의 데이터를 도1의 스토리지 풀(135)의 일부 스토리지 tier에 저장한다. 따라서, 가상 머신에 대한 데이터를 실제로 저장하는 스토리지 tier의 경우, 성능 데이터(720)의 평균 입력/출력 레이턴시는 예측치가 아닌 해당 스토리지 tier의 가상 머신에 대한 실제 평균 입력/출력 레이턴시를 나타낼 수 있다.
- [0122] 각 스토리지 tier의 각 가상 머신에 대한 예상 평균 레이턴시가 주어지면, 가상 머신들, 스토리지 tier들, 및 리소스들을 상호 관련시키는 다차원 매트릭스가 생성될 수 있다. 이러한 다차원 매트릭스는 도9의 리소스 매트릭스(905)로서 도시된다. 다차원 매트릭스를 종이에 나타내기 어렵기 때문에, 도9는 리소스 매트릭스(905)를 개별적인 스토리지 tier에 대한 데이터를 저장하는 각각의 2 차원 매트릭스를 갖는 몇몇의 별개의 2 차원 매트릭스들(910, 915, 및 920)로서 도시하지만, 이는 단지 시각화 목적을 위한 것이다.
- [0123] 스토리지 tier에 대한 2차원 매트릭스(910)에서, 성능 데이터(720)로부터의 평균 입력/출력 레이턴시는 다음
- $$\frac{10^6}{AverageIOlatency * maxIOPS[t]}$$
- 공식을 사용하여 처리량으로 변환될 수 있다. 즉, $10^6(1,000,000)$ 을 당해 스토리지 tier를 사용하는 방정식에서 가상 머신의 평균 입력/출력 레이턴시와 당해 스토리지 tier에서 지원할 수 있는 최대 IOPS(이 값은 시스템 관리자가 스토리지 tier에서 제공하는 실제 최대 값보다 작은 값으로 설정할 수 있다(예 : 실제 최대 값의 90%))의 곱으로 나눌 수 있다. 레이턴시는 일반적으로 마이크로 초(1 마이크로 초 = $\frac{1}{1,000,000}$)로 표현되기 때문에 값 10^6 가 사용될 수 있다. 따라서 12 μ s의 레이턴시가 0.000012 초로 변환되지 않으면 평균 입력/출력 레이턴시의 정확한 측정은 방정식에서 발생하는 단위의 변화를 반영하여야 한다. 처리량은 처리량에 가상 머신에서 발행한 입력/출력 커맨드의 평균 쓰기 크기를 곱하여 대역폭으로 변환될 수 있다. 용량은 가상 머신 파일의 스토리지 크기를 스토리지 tier에 의해 제공되는 최대 스토리지로 나눈 값으로부터 계산할 수 있다(또한, 시스템 관리자는 이 값을 실제의 최대 값보다 작은 값으로 설정할 수도 있음). 성능 데이터(720)로부터 리소스 매트릭스(905)에 저장된 값들의 이러한 변환은 도8의 메트릭 컨버터(815)를 사용하여 계산될 수 있다.
- [0124] 도10은 도7의 성능 데이터(720)의 선형 회귀를 도시한다. 도10에서, 두 개의 상이한 가상 머신들에 대한 성능 데이터가 도시되어있다. 하나의 가상 머신의 경우, 그래프상의 데이터 포인트들은 선형 회귀(1005)에 의해 근사될 수 있다. 다른 가상 머신의 경우, 그래프 상의 데이터 포인트들은 선형 회귀 (1010)에 의해 근사화될 수 있다. 도 10은 단지 두 개의 가상 머신에 대한 데이터 포인트들을 도시하지만, 본 발명의 실시 예는 선형 회귀가 계산 될 수 있는 임의의 수의 가상 머신들을 지원할 수 있다.
- [0125] 각 데이터 포인트는 주입된 특정 레이턴시를 가상 머신의 평균 입력/출력 레이턴시와 상호 연관시킬 수 있다. 예를 들어, 주입된 레이턴시가 없는 경우, 첫 번째 가상 머신은 평균 입력/출력 레이턴시가 약 3000 μ s 인 반면 두 번째 가상 머신은 평균 입력/출력 레이턴시가 약 700 μ s이다.
- [0126] 주입된 레이턴시는 특히 가상 머신이 더 낮은 스토리지 tier(즉, 낮은 수준의 성능을 가진 스토리지 tier)을 사

용하여 적절하게 작동하는지 여부를 판별할 때 유용하다. 결국, 주입된 레이턴시의 양은 정확히 현재 스토리지 tier와 타겟 스토리지 tier 간의 차이일 수 있다. 하지만 주입된 레이턴시를 사용하면 더 빠른 스토리지 tier를 에뮬레이트할 수 없다. 결국, 네거티브 레이턴시를 주입할 수 없으며 실제로 스토리지 tier가 실제로 응답할 수 있는 것보다 빠르게 응답하지 않을 수 있다. 이러한 솔루션은 선형 회귀(1010)와 같이 나타난다. 가상 머신이 느린 스토리지 tier에 어떻게 반응하는지에 대한 정보를 사용하면 선형 회귀 모델을 사용하여 더 빠른 스토리지 tier의 성능을 예측할 수 있다. 기술적으로, 제공된 데이터의 범위를 벗어나는 값을 예측하는 것은 데이터 범위 내부의 값을 예측하는 "보간법(interpolation)"과 비교하여 "외삽법(extrapolation)"이라고 한다. 그러나 선형 회귀(1010)와 같은 선형 회귀를 사용하는 경우, 계산은 보간과 외삽에 대해 동일하다. 따라서, 선형 회귀(1010)를 사용하면 더 빠른 스토리지 tier를 사용할 때 가상 머신이 어떻게 반응할지 예측할 수 있다.

[0127] 스토리지 tier들 사이의 레이턴시의 실제적인 차이에 대응하는 데이터 포인트들만을 사용할 수 있지만, 보다 많은 데이터 포인트가 사용 가능할수록, 선형 회귀 모델(805)에 의해 제공되는 예측치가 더 양호해질 수 있다. 따라서, 다양한 상이한 레이턴시들이 도3의 입력/출력 필터(325)를 사용하여 주입됨으로써 원하는 만큼의 많은 데이터 포인트들을 제공할 수 있다.

[0128] 물론 주입된 레이턴시를 사용하여 가상 머신의 모든 입력/출력 커맨드가 느려지면 할당된 스토리지 tier의 진정한 이점을 가상 머신이 받지 못한다. 따라서 주입된 레이턴시는 가상 머신에서 발행한 입력/출력 커맨드의 하위 집합에만 사용될 수 있다. 주입된 레이턴시에 대해 선택되는 입력/출력 커맨드가 무작위로 선택되면, 도7의 성능 데이터(720)가 의미있는 선형 회귀(1005 및 1010)를 제공하여야 한다. 데이터 포인트에 대한 변동 계수(및 신뢰도)를 결정하기 위해 알려진 통계 기법이 사용될 수 있다. 데이터 포인트의 신뢰도가 충분히 높으면, 선형 회귀(1005 및 1010)는 정확한 예측치로 신뢰될 수 있다. 원하는 신뢰 수준이 지정될 수 있다 (예 : 90 % 또는 95 %). 주입된 레이턴시에 대해 입력/출력 커맨드가 선택되는 방법은 도13을 참조하여 아래에서 더 설명된다.

[0129] 선형 회귀(1005 및 1010)는 매우 다른 데이터 세트를 나타낸다 : 선형 회귀(1005)는 상당히 가파른 기울기를 포함하는 반면, 선형 회귀(1010)는 상당히 완만한 기울기를 포함한다. 모든 가상 머신이 입력/출력 커맨드들의 레이턴시의 변화에 동일한 방식으로 응답하는 것은 아니다. 일부 가상 머신은 다른 것들보다 민감하다. 예를 들어, 선형 회귀(1005)를 초래하는 데이터 포인트들을 생성한 가상 머신은 동기 어플리케이션을 실행할 수 있으며, 레이턴시의 작은 변화는 어플리케이션의 성능에 중요한 영향을 미칠 수 있다. 한편, 선형 회귀(1010)를 초래하는 데이터 포인트들을 생성한 가상 머신은 레이턴시의 변화가 더 작은 영향을 미치는 비동기 어플리케이션을 실행할 수 있다. 또는 선형 회귀(1005)는 높은 처리량을 요구하는 가상 머신에 대한 것일 수 있지만, 선형 회귀(1010)는 보다 낮은 처리량을 요구하는 가상 머신에 대한 것일 수 있다. 비슷한 방식으로 가상 머신의 민감도가 선형 회귀에 영향을 미칠 수 있다. 더 민감한 가상 머신은 더 가파른 경사를 가지며 덜 민감한 가상 머신은 더 완만한 경사를 가진다. 민감도에 대한 비슷한 개념은 가상 머신이 밀접하게 결합되어 있는지 여부이다 ("밀접하게 결합"이란, 동기식 및 비동기식 어플리케이션의 비교와 유사하게 가상 머신이 스토리지 tier의 응답 시간에 얼마나 민감하거나 종속적인지를 의미함). 가상 머신이 더 밀접하게 결합될수록 선형 회귀의 기울기가 더 가팔라진다. 일반적으로 선형 회귀(1005)와 같이 가파른 기울기를 갖는 선형 회귀는 더 빠른 스토리지 tier에서 보다 잘 수행하는 가상 머신을 나타내지만 선형 회귀(1010)와 같이 완만한 기울기를 가진 선형 회귀는 느린 스토리지 tier에서 적절하게 수행하는 가상 머신을 나타낸다.

[0130] 도11은 도3의 스토리지 tier들(340-1, 340-2, 340-3)에 대한 도8의 전문성 정보(820)를 도시한다. 도11에서, 전문성 정보(820)는 매트릭스로서 도시된다. 각 스토리지 tier와 각 리소스의 공통점은 해당 스토리지 tier가 해당 리소스를 제공하는지 여부(또는, 그 정도)를 반영하는 값을 지정할 수 있다는 것이다. 본 발명의 개념의 일부 실시 예에서, 도11에 도시된 바와 같이, 전문성 정보(830)는 이진 정보(0 vs 1, 예 vs 아니오 또는 임의의 다른 값들의 쌍)를 사용할 수 있는데, 여기서 하나의 값은 스토리지 tier가 해당 리소스를 제공함을 가리키고, 다른 값은 스토리지 tier가 해당 리소스를 제공하지 않음을 가리킨다. 따라서, 예를 들어, 전문성 정보(820)는 스토리지 tier 1이 높은 처리량 및 높은 대역폭을 제공하고, 스토리지 tier 2가 높은 대역폭을 제공하고, 스토리지 tier 3이 높은 용량을 제공한다는 것을 나타내는 것으로서 판독될 수 있다.

[0131] 본 발명의 개념의 다른 실시 예에서, 전문성 정보(820) 내의 값은 이진 선택에 제한될 필요는 없다. 예를 들어, 값의 범위는 0에서 1까지이며, 0은 스토리지 tier에서 해당 리소스를 제공하지 않음을 나타내고 1은 스토리지 tier에서 해당 리소스를 완전히 제공함을 나타내고 중간에 있는 값은 부분 지원을 나타낼 수 있다. 따라서 0.75의 값은 스토리지 tier가 리소스를 어느 정도까지 제공하지만 "최상" 스토리지 tier만큼은 아니라는 것을 나타낼 수 있다. 반면 0.25의 값은 스토리지 tier에서 해당 리소스를 제공하지만 "최악" 스토리지 tier("최상" 및

"최악"은 상대적 용어임)보다 조금 나은 정도를 나타낼 수 있다.

[0132] 도12는 도3의 가상 머신들(305-1, 305-2 및 305-3)을 도3의 다른 스토리지 티어들로 마이그레이션시키는 데 사용되는 스코어 매트릭스의 계산을 도시한다. 전문성 정보(820)에 리소스 매트릭스(905)를 곱함으로써, 각 티어가 각 가상 머신을 얼마나 잘 지원 하는지를 반영하는 스코어 매트릭스(1205)가 생성 될 수 있다. 예를 들어, 가상 머신 1에 값들이 제공될 수 있다. 이 값들은 스토리지 티어 1이 55 % 일치, 스토리지 티어 2가 25 % 일치, 스토리지 티어 3이 30 % 일치함을 나타낼 수 있다. (특정 스토리지 티어에 특정 가상 머신의 저장량을 표시하는 것이 아니라 각 스토리지 티어가 가상 머신을 얼마나 잘 지원 하는지를 나타내기 때문에 값들은 100%까지 추가 되지 않는다.) 다른 가상 머신들에 대한 엔트리들은 유사하게 해석될 수 있다.

[0133] 스코어 매트릭스(1205)가 생성되면, 도 8의 티어 선택(825)은 각 가상 머신에 적합한 티어를 선택할 수 있다. 이러한 선택은 티어 및 가상 머신의 각 조합에 대해 하기 값을 계산함으로써 수행될 수 있다.

[0134] $FinalScore = AgingFactor \times HistoricalFinalScore + CurrPerfGain - CurrPerfPenalty$

[0135] AgingFactor는 가상 머신과 스토리지 티어의 조합에 대해 더 오래된 최종 점수를 보존하는 데 사용되는 요소일 수 있다(이전 티어 선택이 결과에 지나치게 편향되는 것을 방지하기 위해). CurrPerfGain은 가상 머신과 티어의 조합을 위해 도12의 스코어 매트릭스(1205)로부터 추출된 값일 수 있다. CurrPerfPenalty는 하나의 스토리지 티어에서 다른 스토리지 티어로 데이터를 마이그레이션하는 비용을 나타낸다.

[0136] 가상 머신은 도1의 스토리지 풀(135)을 포함하는 서버와 다른 서버 상에 상주할 수 있기 때문에, CurrPerfPenalty는 데이터 자체의 이동 비용 만 반영하고 가상 머신의 작동에 대한 어떤 영향도 고려할 필요가 없다. 달리 말하면, 가상 머신은 도 1의 가상 머신 서버(110-1) 상에 있기 때문에, 데이터 마이그레이션을 수행하기 위해 가상 머신을 호스팅하는 서버상의 처리 능력이 필요하지 않다. 모든 처리는 도1의 데이터 센터 서버(130)에서 수행된다. CurrPerfPenalty는 마이그레이션해야 하는 가상 머신 파일의 크기를 원본 스토리지 티어의 읽기 처리 속도와 타겟 스토리지 티어의 쓰기 처리 속도 중 작은 값으로 나눔으로써 계산될 수 있다. 즉, 원본 스토리지 티어에서 데이터를 읽기 위해 필요한 시간과 타겟 스토리지 티어에 데이터를 쓰기 위해 필요한 시간을 비교하고 더 큰 값이 마이그레이션을 수행하는 데 필요한 시간으로 선택될 수 있다. 왜냐하면, 더 느린 것은(원본 스토리지 티어 읽기 또는 타겟 스토리지 티어 쓰기) 데이터 마이그레이션의 병목 현상을 나타내기 때문이다.

[0137] 가상 머신 및 스토리지 티어의 각 조합에 대한 최종 점수가 주어지면, 티어 선택 모듈(825)은, 도1의 시스템(105)의 성능의 가장 큰 향상을 가져오는 가상 머신의 스토리지 티어들로의 할당을 식별함으로써, 전체 시스템 성능을 전체적으로 최적화하도록 각 가상 머신에 대한 스토리지 티어들을 쉽게 선택할 수 있다. 이 때, 성능 향상과 성능 감소(가상 머신이 덜 강력한 스토리지 티어로 마이그레이션 됨) 및 마이그레이션 비용이 모두 고려되어야 한다.

[0138] 위에서 암시된 것처럼, 모든 시간 동안에 단일한 글로벌 최적화를 수행하면 각 가상 머신의 워크로드가 변화하지 않고 가상 머신들의 집합 자체가 변경되지 않는다고 가정한다(새 가상 머신이 추가되지 않고 기존 가상 머신이 제거되지 않음). 실제로 이러한 조건들은 비현실적이다. 따라서, 도1의 시스템 (105)이 한 번만 최적화되었다면, 시간이 지남에 따라 스토리지 티어에 대한 가상 머신 할당이 더 이상 최적이지 아닐 수도 있다. 이러한 문제에 대한 해결책은 최적화를 한 번만 수행하기보다는 일정한 간격으로 수행하는 것이다. 그러나 지속적으로 최적화를 수행하는 것은 효율적이지 않을 수 있다. 예를 들어 가상 머신 입력/출력 커맨드는 지속적으로 주입된 레이턴시를 거쳐야한다. 이는, 가상 머신이 해당 스토리지 티어의 모든 이점을 받지 못함을 의미할 수 있다. 따라서, 연속적인 최적화를 수행하는 대신, 최적화는 주기적으로 수행될 수 있다. 이를 위해 마이그레이션 에포크와 모니터링 에포크의 개념이 도입된다.

[0139] 모니터링 에포크는 도3의 입력/출력 필터(325)가 가상 머신이 다른 스토리지 티어에서 현재 수행할 수 있는 방법을 판별하기 위해 일부 입력/출력 커맨드들에 레이턴시를 주입하는 시점을 나타낼 수 있다. 에포크의 모니터링이 시작될 때, 도3의 입력/출력 필터(325)는 선택된 수의 입력/출력들에 레이턴시를 주입할 수 있다. 이러한 입력/출력 커맨드들의 수는 고정된 수로서 미리 결정될 수 있거나, 이전 모니터링 에포크 동안 발행된 입력/출력 커맨드의 수의 퍼센테이지(예를 들어, 1 %)로 설정되거나, 또는 둘 다(예를 들어, 두 값들 중 더 큰 값)일 수 있다. 본 발명의 실시 예가 불규칙한 모니터링 에포크를 지원할지라도, 모니터링 에포크는 매 15 분마다, 매 시간 또는 매일과 같이 규칙적인 간격으로 발생할 수 있다. 모니터링 에포크가 시작될 때 가상 머신의 실제 워크로드를 예측하는 간단한 방법은 없다. 따라서, 도3의 입력/출력 필터(325)를 트리거링하여 모니터링 에포크의 시작 단계에서 입력/출력 커맨드들에 레이턴시를 주입하는 것은 임의의 입력/출력 커맨드 세트를 선택하고 도 7

의 의미있는 성능 데이터(720)를 제공할 것이다. 도7의 성능 데이터(720)가 주어진 모니터링 에포크에서 가상 머신들에 대해 일단 생성되고 나면, 도3의 오토-티어링 데몬(330)은 도7의 성능 데이터(720)를 도3의 오토-티어링 컨트롤러(335)로 전송할 수 있다. 대안적으로, 도 3의 오토-티어링 데몬(330)은 마이그레이션 에포크가 시작될 때까지 대기하고 도7의 성능 데이터(720)를 그 때 전송할 수 있다.

[0140] 마이그레이션 에포크(Migration epoch)는 도3의 오토-티어링 컨트롤러(335)가 가상 머신을 하나의 스토리지 티어에서 다른 스토리지 티어로 마이그레이션해야하는지 여부를 결정하는 시점을 나타낼 수 있다. 모니터링 에포크와 마찬가지로, 마이그레이션 에포크는 매일 또는 매주와 같이 정기적 또는 불규칙적인 간격으로 발생하도록 설정될 수 있다. 모니터링 에포크가 발생할 때, 도3의 오토-티어링 컨트롤러(335)는 도7의 성능 데이터(720)를 분석하고, 도9의 리소스 매트릭스(905)를 생성하고, 최종 점수를 계산하고, 각 가상 머신에 대한 스토리지 티어들을 선택하고, 및 필요한 경우 가상 머신 데이터를 마이그레이션할 수 있다.

[0141] 도13은 모니터링 에포크와 마이그레이션 에포크 사이의 관계를 보여준다. 도13에서, 마이그레이션 에포크(1305) 및 모니터링 에포크(1310)가 도시되어 있다. 도 13에 도시된 바와 같이, 본 발명의 실시 예에서, 마이그레이션 에포크(1305)는 모니터링 에포크(1310)와 동일하거나 더 짧을 수 있지만, 마이그레이션 에포크(1305)는 모니터링 에포크(1310)보다 전형적으로 더 길다. 또한, 마이그레이션 에포크(1305)와 모니터링 에포크(1310)는 정렬될 수 있지만 정렬될 필요는 없다.

[0142] 도14는 일 실시 예에 따라, 도7의 성능 데이터(3615)를 생성하기 위해 도7의 입력/출력(Input/Output) 커맨드들(705)로 레이턴시를 주입하기 위한 도3의 입력/출력 필터(325)에 대한 예시적인 절차의 흐름도를 도시한다. 도 14의 블록(1405)에서, 도3의 입력/출력 필터(325)는 도13의 새로운 모니터링 에포크(1310)가 시작되었는지 판별할 수 있다. 시작되지 않았다면, 도3의 입력/출력 필터(325)는 기다릴 수 있다. 시작되었다면, 블록(1410)에서 도3의 입력/출력 필터(325)는 도3의 어플리케이션(310)로부터 도7의 몇몇 입력/출력 커맨드들을 수신할 수 있다. 이러한 방식으로 도3의 입력/출력 필터(325)에 의해 얼마나 많은 커맨드들이 처리를 위해 선택되는지는 시스템 구성에 따라 달라질 수 있다. 블록(1415)에서, 도3의 입력/출력 필터(325)는 도7의 입력/출력 커맨드들로 도7의 레이턴시들을 주입할 수 있다. 블록(1420)에서, 도3의 입력/출력 필터(325)는 도1의 데이터 센터 서버(130)로 커맨드들을 전송할 수 있다. 블록(1425)에서, 도3의 입력/출력 필터(325)는 도1의 데이터 센터 서버(130)로부터 도7의 응답(715)을 수신할 수 있다. 블록(1425)에서, 도3의 입력/출력 필터(325)는 도7의 응답(715)을 도3의 어플리케이션(310)으로 전달할 수 있다. 최종적으로, 블록(1430)에서, 도3의 입력/출력 필터(325)는 가상 머신에 대한 평균 입력/출력 레이턴시들을 사용하여 도7의 성능 데이터(720)를 계산할 수 있다.

[0143] 도15a-15b는 일 실시 예에 따라, 스토리지 티어들을 통한 가상 머신들의 마이그레이션을 관리하기 위한 도3의 오토-티어링 컨트롤러(335)에 대한 예시적인 절차의 흐름도를 도시한다. 도15a의 블록(1505)에서, 도3의 오토-티어링 컨트롤러(335)는 도13의 새로운 모니터링 에포크(1310)가 시작되었는지 판별할 수 있다. 시작되지 않았다면, 도3의 오토-티어링 컨트롤러(335)는 기다릴 수 있다. 시작되었다면, 블록(1510)에서, 도3의 오토-티어링 컨트롤러(335)는 도3의 오토-티어링 데몬(330)으로부터 도7의 성능 데이터(720)를 수신할 수 있다. 블록(1515)에서, 도3의 오토-티어링 컨트롤러(335)는 도13의 새로운 마이그레이션 에포크(1305)가 시작되었는지 판별할 수 있다. 시작되지 않았다면, 도3의 오토-티어링 컨트롤러(335)는 기다릴 수 있다.

[0144] 시작되었다면, 블록(1520)에서, 도8의 리소스 매트릭스 생성기(810)는 도9의 리소스 매트릭스를 생성할 수 있다. (다른 구성 요소들 중에서, 도8의 선행 회귀 모듈(805) 및 도8의 메트릭 컨버터(815)를 사용하는). 블록(1525)에서, 티어 선택 모듈(825)은 리소스 매트릭스(905)를 사용하여 스토리지 티어들에 가상 머신들을 할당할 수 있다. 블록(1530)에서, 도3의 오토-티어링 컨트롤러(335)는 하나의 스토리지 티어에서 다른 스토리지 티어로 가상 머신이 마이그레이션되는지 확인할 수 있다. 그렇다면, 블록(1535)에서, 마이그레이션 모듈(830)은 도5의 가상 머신 파일(505)을 하나의 스토리지 티어에서 다른 스토리지 티어로 마이그레이션할 수 있다.

[0145] 도16은 일 실시 예에 따라, 도9의 리소스 매트릭스(905)를 생성하기 위한, 도3의 오토-티어링 컨트롤러(335)에 대한 예시적인 절차의 흐름도를 도시한다. 도 16을 참조하면, 블록(1605)에서, 도8의 선행 회귀 모듈(805)은 도7의 성능 데이터(720)에 대해 선행 회귀를 수행하여 각 스토리지 티어에서 가상 머신의 성능을 평가하는 수식을 생성할 수 있다. 블록(1610)에서, 도3의 오토-티어링 컨트롤러(335)는 도10의 선행 회귀(1005 및 1010)를 사용하여 각 가상 머신이 각 스토리지 티어에서 얼마나 잘 수행되는지(즉, 각 스토리지 티어 상의 가상 머신의 성능)를 예측할 수 있다. 블록(1615)에서, 리소스 매트릭스 생성기(810)는 각각의 리소스에 따라 각 스토리지 티어에서 가상 머신에 대한 성능 메트릭들을 생성할 수 있다. 블록(1620)에 도시된 바와 같이, 이는 메트릭 변환기(815)를 사용하여 처리량과 같은 하나의 메트릭을 대역폭과 같은 다른 메트릭으로 변환하는 것을 포함할 수

있다. 점선(1625)으로 도시된 바와 같이, 블록(1620)은 생략될 수 있다.

- [0146] 상술한 논의로부터 알 수 있는 바와 같이, 본 발명의 개념의 실시 예는 종래의 티어링 및 캐싱 솔루션에 비해 중요한 이점을 제공한다. 첫 번째, 본 발명의 개념의 실시 예는 스토리지 tier의 다중 리소스를 고려하여 지원한다. 또한, 리소스의 수가 변할 수 있기 때문에, 리소스의 수는 1로 설정될 수 있으며, 종래의 HDD 스토리지 풀과 함께 사용되기 위한 역 호환성을 제공할 수 있다. 두 번째, 본 발명의 실시 예는 종래의 티어링 및 캐싱 솔루션이 무시하는 플래시 저장 시스템의 보다 빠른 성능을 설명한다. 세 번째로, 본 발명의 실시 예는 글로벌 최적화를 찾고, 기존 가상 머신들의 집합 및 워크로드 스파이크(workload spike)들과 같은 가상 머신들의 워크로드에서의 변화가 발생함에 따라 시간 경과에 따른 글로벌 최적화를 조정하는 것을 지원할 수 있다. 이는 일반적으로 시스템 전체를 최적화하는 것이 아니라 개별 가상 머신에 대한 로컬 솔루션을 찾는 기존 솔루션에 비해 유리하다.
- [0147] 도14-16을 참조하면, 본 발명의 개념의 일부 실시 예가 도시된다. 그러나 당업자는 블록의 순서를 변경하거나, 블록을 생략하거나, 또는 도면에 도시되지 않은 링크를 포함함으로써, 본 발명의 개념의 다른 실시 예도 가능하다는 것을 인식할 것이다. 흐름도의 이러한 모든 변형은 명시적으로 기술되었든 아니든간에 본 발명의 개념의 실시 예인 것으로 간주된다.
- [0148] 다음의 설명은 본 발명의 개념의 특정 실시 예가 구현될 수 있는 적합한 머신 또는 장치에 대한 간략하고 일반적인 설명을 제공하기 위한 것이다. 머신(Machine) 또는 머신들은 키보드, 마우스 등과 같은 종래의 입력 디바이스뿐만 아니라 다른 머신으로부터 수신된 지시들, 가상 현실(VR) 환경과의 상호 작용, 생체 인식(biometric) 피드백, 또는 다른 입력 신호에 의해 적어도 부분적으로 제어될 수 있다. 본 명세서에 사용된 바와 같이, "머신"이라는 용어는 단일 머신, 가상 머신, 또는 통신적으로 결합된 머신들의 시스템, 가상 머신들, 또는 함께 동작하는 디바이스들을 광범위하게 포함하는 것으로 의도된다. 예시적인 머신은 퍼스널 컴퓨터, 워크 스테이션, 서버, 휴대용 컴퓨터, 핸드 헬드(handheld) 디바이스, 전화, 태블릿 등과 같은 컴퓨팅 디바이스뿐만 아니라 자동차, 기차, 택시 등과 같은 사설 또는 대중 교통과 같은 운송 디바이스를 포함할 수 있다.
- [0149] 머신 또는 머신들은 프로그램 가능 또는 비-프로그래밍 가능 로직 디바이스 또는 어레이, ASIC(Application Specific Integrated Modules), 임베디드 컴퓨터, 스마트 카드 등과 같은 내장형 컨트롤러를 포함할 수 있다. 머신 또는 머신들은 가령 네트워크 인터페이스, 모뎀 또는 다른 통신 커풀링을 통하는 하나 이상의 원격 머신들에 대한 하나 이상의 연결을 활용할 수 있다. 머신은 인트라넷, 인터넷, 로컬 영역 네트워크, 광역 네트워크 등과 같은 물리적 및/또는 논리적 네트워크에 의해 상호 접속될 수 있다. 당업자는 네트워크 통신은, 라디오 주파수 (RF), 위성, 마이크로 웨이브, IEEE(Institute of Electrical and Electronics Engineers) 802.11, Bluetooth®, 광학, 적외선, 케이블, 레이저 등을 포함한 다양한 유선 및/또는 무선의 근거리 통신망이나 장거리 통신 사업자 및 프로토콜을 활용할 수 있다는 것을 이해할 것이다.
- [0150] 본 발명의 개념의 실시 예는 머신에 의해 액세스될 때 머신이 태스크를 수행하거나 추상 데이터 타입을 정의하거나 로우-레벨 하드웨어 컨텍스트들을 정의하는 것과 같은 기능, 절차, 데이터 구조, 애플리케이션 프로그램 등을 포함하는 관련 데이터를 참조하여 또는 관련하여 기술될 수 있다. 관련 데이터는 예를 들어, RAM, ROM 등과 같은 휘발성 및/또는 비 휘발성 메모리, 또는 하드 드라이브, 플로피 디스크, 광학 스토리지, 테이프, 플래시 메모리, 메모리 스틱, 디지털 비디오 디스크, 생물학적 스토리지 등을 포함하는 다른 스토리지 및 이들과 연관된 저장 매체에 저장될 수 있다. 관련 데이터는 물리적 및/또는 논리적 네트워크를 포함하는 전송 환경을 통해 패킷, 시리얼 데이터, 병렬 데이터, 전파되는 신호 등의 형태로 전달될 수 있으며, 압축 또는 암호화 된 포맷으로 사용될 수 있다. 관련 데이터는 분산 환경에서 사용될 수 있으며 머신 액세스를 위해 로컬 및/또는 원격으로 저장될 수 있다.
- [0151] 본 발명의 개념의 실시 예는 하나 이상의 프로세서에 의해 실행 가능한 커맨드들을 포함하는 실질적인 비-일시적 기계 판독 가능 매체를 포함할 수 있으며, 상기 커맨드들은 본 명세서에서 설명된 바와 같은 발명 개념의 요소를 수행하기 위한 커맨드들을 포함한다.
- [0152] 전술한 방법의 다양한 동작은 다양한 하드웨어 및/또는 소프트웨어 컴포넌트(들), 모듈 및/또는 모듈(들)과 같은 동작을 수행할 수 있는 임의의 적절한 수단에 의해 수행될 수 있다. 소프트웨어는 논리적 기능들을 구현하기 위한 실행 가능 커맨드의 순서화된 목록을 포함할 수 있으며, 싱글 또는 멀티-코어 프로세서 또는 프로세서를 포함하는 시스템과 같은 커맨드 실행 시스템, 장치, 또는 디바이스에 의해 또는 그것과 관련된 임의의 "프로세서-판독 가능 매체"에서 구현될 수 있다.

- [0153] 본 명세서에 개시된 실시 예와 관련하여 설명된 방법 또는 알고리즘 및 기능의 블록 또는 단계는 하드웨어, 프로세서에 의해 실행되는 소프트웨어 모듈, 또는 이들의 조합으로 직접 구현될 수 있다. 소프트웨어로 구현되는 경우, 기능은 비-일시적 컴퓨터 판독 가능 매체 상에 하나 이상의 커맨드 또는 코드로서 저장되거나 전송될 수 있다. 소프트웨어 모듈은 RAM(random access memory), 플래시 메모리, ROM(Read Only Memory), EPROM(Electrically Programmable ROM), EEPROM(Electrically Erasable Programmable ROM), 레지스터, 하드 디스크, 이동식 디스크, CD ROM, 또는 당 업계에 공지된 임의의 다른 형태의 저장 매체에 상주할 수 있다.
- [0154] 설명된 실시 예를 참조하여 본 발명의 개념의 원리를 설명하고 예시 하였지만, 설명된 실시 예는 이러한 원리를 벗어나지 않고 배열 및 세부 사항을 수정할 수 있으며 임의의 바람직한 방식으로 조합될 수 있음을 알 것이다. 또한, 전술한 설명은 특정 실시 예에 초점을 맞추었지만, 다른 구성도 고려될 수 있다. 특히, "발명의 개념의 실시 예에 따른" 등과 같은 표현이 본 명세서에서 사용 되더라도, 이러한 문구는 일반적으로 실시 예의 가능성을 참조하는 것으로 의도되며, 본 발명의 개념을 특정 실시 예의 구성으로 제한하려는 것은 아니다. 본 명세서에 사용된 바와 같이, 이들 용어는 다른 실시 예와 조합 가능한 동일 또는 상이한 실시 예를 나타낼 수 있다.
- [0155] 전술한 예시적인 실시 예는 본 발명의 개념을 제한하는 것으로 해석되어서는 안된다. 비록 일부 실시 예가 설명 되었지만, 당업자는 본 개시의 신규한 교시 및 이점으로부터 실질적으로 벗어나지 않고도 이들 실시 예에 많은 수정이 가능하다는 것을 용이하게 이해할 것이다. 따라서, 이러한 모든 수정은 청구항에 정의된 본 발명의 개념의 범위 내에 포함되는 것으로 의도된다.
- [0156] 본 발명의 개념의 실시 예는 다음의 설명으로 제한 없이 확장될 수 있다 :
- [0158] Statement 1. 본 발명의 개념의 실시 예는,
- [0159] 복수의 리소스들을 제공하는 복수의 스토리지 디바이스들, 상기 복수의 스토리지 디바이스들은 복수의 스토리지 티어들로 구성되고 제1가상 머신에 대한 제1 데이터 및 제2가상 머신에 대한 제2데이터를 저장하고;
- [0160] 상기 제1가상 머신으로부터 제1입력/출력(I/O) 커맨드, 상기 제2가상 머신으로부터의 제2입력/출력 커맨드, 상기 복수의 스토리지 티어에서 상기 제1가상 머신의 성능을 모델링하는 제1성능 데이터, 및 상기 복수의 스토리지 티어에서 상기 제2가상 머신의 성능을 모델링하는 제2성능 데이터를 수신하는 수신기;
- [0161] 상기 제1 입력/출력 커맨드에 대한 제1응답을 상기 제1가상 머신에 송신하고 상기 제2 입력/출력 커맨드에 대한 제2응답을 상기 제2가상 머신에 송신하기 위한 송신기; 및
- [0162] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위한 제1스토리지 티어를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하기 위한 제2스토리지 티어를 선택하고, 상기 제1성능 데이터 및 상기 제2성능 데이터에 응답하여 상기 제1가상 머신에 대한 적어도 하나의 상기 제1데이터를 제1스토리지 티어로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 티어로 마이그레이션하기 위한 오토-티어링 컨트롤러를 포함하되,
- [0163] 상기 오토-티어링 컨트롤러는, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 티어를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 티어를 선택하여, 상기 복수의 스토리지 티어들 전반의 모든 가상 머신들의 성능을 최적화하도록 동작하고,
- [0164] 상기 오토-티어링 컨트롤러는, 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 티어로 또는 상기 제2가상 머신에 대한 상기 제2 데이터를 상기 제2스토리지 티어로 이동시킴으로써 초래되는 성능 변화 및 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 티어로, 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 티어로 이동시키는 이동 비용을 고려하도록 동작하는 시스템.
- [0166] Statement 2. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며,
- [0167] 상기 시스템은 프로세서를 더 포함하고; 및
- [0168] 오토-티어링 컨트롤러는 프로세서에서 실행되는 소프트웨어를 포함한다.
- [0170] Statement 3. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 오토-티어링 컨트롤러는 :
- [0171] 상기 제1성능 데이터 및 상기 제2성능 데이터에 응답하여, 상기 제1가상 머신에 대한 상기 제1스토리지 티어와

상기 제2가상 머신에 대한 상기 제2스토리지 티어를 선택하는 티어 선택 모듈; 및

- [0172] 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 티어로, 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 티어로 이동시키기 위한 마이그레이션 모듈을 포함한다.
- [0174] Statement 4. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 오토-티어링 컨트롤러는 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 티어들과 상기 복수의 스토리지 티어들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 리소스 매트릭스 생성기를 포함한다.
- [0176] Statement 5. 본 발명의 개념의 실시 예는 Statement 4에 따른 시스템을 포함하며, 상기 오토-티어링 컨트롤러는, 상기 리소스 매트릭스에 응답하여, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위한 제1스토리지 티어를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하기 위한 제2스토리지 티어를 선택한다.
- [0178] Statement 6. 본 발명의 개념의 실시 예는 Statement 5에 따른 시스템을 포함하며, 상기 오토-티어링 컨트롤러는, 상기 리소스 매트릭스 및 상기 복수의 스토리지 티어들 각각에 대한 전문성 정보에 응답하여, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위한 제1스토리지 티어를 선택하고 상기 제2가상 머신에 대한 상기 제2데이터를 저장하기 위한 제2스토리지 티어를 선택하도록 동작한다.
- [0179] Statement 7. 본 발명의 개념의 실시 예는 Statement 6에 따른 시스템을 포함하며, 상기 복수의 스토리지 티어들 각각에 대한 상기 전문성 정보는 상기 복수의 스토리지 티어들 각각이 상기 복수의 리소스 각각을 지원하는지 여부를 식별한다.
- [0181] Statement 8. 본 발명의 개념의 실시 예는 Statement 7에 따른 시스템을 포함하며, 상기 복수의 리소스들은 처리량, 대역폭, 용량, 및 쓰기 증폭을 포함한다.
- [0183] Statement 9. 본 발명의 개념의 실시 예는 Statement 4에 따른 시스템을 포함하며, 상기 리소스 매트릭스 생성기는, 상기 복수의 스토리지 티어들 각각에서 상기 제1가상 머신에 대한 제1성능 메트릭을 상기 복수의 리소스들의 제2리소스에 따라 상기 복수의 스토리지 티어들 각각의 상기 제1가상 머신에 대한 제2성능 메트릭으로 변환하는 메트릭 컨버터를 포함한다.
- [0185] Statement 10. 본 발명의 개념의 실시 예는 Statement 4에 따른 시스템을 포함하며, 상기 제1성능 메트릭은 상기 제1가상 머신에 대한 제1평균 입력/출력 레이턴시 데이터를 포함한다.
- [0187] Statement 11. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며,
- [0188] 상기 오토-티어링 컨트롤러는 상기 제1성능 데이터에 대한 제1선형 회귀를 수행하고 상기 제2성능 데이터에 대한 제2선형 회귀를 수행하는 선형 회귀 모듈을 포함하며; 및
- [0189] 상기 오토-티어링 컨트롤러는 상기 제1선형 회귀 및 상기 제2선형 회귀에 응답하여, 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위해 상기 제1스토리지 티어를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하기 위해 상기 제2스토리지 티어를 선택한다.
- [0191] Statement 12. 본 발명의 개념의 실시 예는 Statement 11에 따른 시스템을 포함하며, 상기 오토-티어링 컨트롤러는, 상기 제1선형 회귀 및 상기 제2선형 회귀에 응답하여, 상기 제1가상 머신과 상기 제2가상 머신에 대한 예측된 평균 입력/출력 레이턴시들을 사용함으로써 상기 제1가상 머신에 대한 상기 제1데이터를 저장하기 위한 상기 제1스토리지 티어를 선택하고 상기 제2가상 머신에 대한 제2데이터를 저장하기 위한 상기 제2스토리지 티어를 선택한다.
- [0193] Statement 13. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 제1성능 데이터는 제1리소스를 사용하는 제1메트릭 성능 데이터를 포함하고; 및
- [0194] 상기 오토-티어링 컨트롤러는, 상기 제1리소스를 사용하는 상기 제1메트릭 성능 데이터를 제2리소스를 사용하는 제2메트릭 성능 데이터로 변환하는 메트릭 변환기를 포함한다.
- [0196] Statement 14. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 오토-티어링 컨트롤러는, 상기 제1가상화 머신에 대한 상기 제1데이터를 저장하기 위한 제1스토리지 티어를 선택하고, 제2가상 머신에 대한 제2데이터를 저장하기 위한 제2스토리지 티어를 선택하도록 동작하고, 및 마이그레이션 에포크 당 상기 제1성능 데이터와 상기 제2성능 데이터에 응답하여 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나

를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션시키도록 동작한다.

- [0198] Statement 15. 본 발명의 개념의 실시 예는 Statement 14에 따른 시스템을 포함하며, 상기 제1성능 데이터는 상기 마이그레이션 에포크 내의 복수의 모니터링 에포크들로부터의 제1성능 데이터를 포함하고; 및 상기 제2성능 데이터는 상기 마이그레이션 에포크 내의 상기 복수의 모니터링 에포크들로부터의 제2성능 데이터를 포함한다.
- [0200] Statement 16. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 제1가상 머신을 저장하는 제1가상 머신 서버; 및 상기 제2가상 머신을 저장하는 제2가상 머신 서버를 포함한다.
- [0202] Statement 17. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 제2가상 머신 서버가 상기 제1가상 머신 서버이다.
- [0204] Statement 18. 본 발명의 개념의 실시 예는 Statement 16에 따른 시스템을 포함하며, 상기 제1가상 머신 서버는 상기 복수의 스토리지 tier들에서 상기 제1가상 머신의 성능을 모델링하는 상기 제1성능 데이터를 생성하기 위한 입력/출력 필터를 포함한다.
- [0206] Statement 19. 본 발명의 개념의 실시 예는 Statement 18에 따른 시스템을 포함하며, 상기 입력/출력 필터는 상기 제1가상 머신으로부터의 상기 제1 입력/출력 커맨드를 포함하는 복수의 입력/출력 커맨드들에 레이턴시를 주입하고, 상기 주입된 레이턴시에 응답하여 상기 제1가상 머신에 대한 평균 입력/출력 레이턴시를 결정하도록 동작할 수 있다.
- [0208] Statement 20. 본 발명의 개념의 실시 예는 Statement 19에 따른 시스템을 포함하며, 상기 입력/출력 필터는, 상기 제1가상 머신으로부터 상기 복수의 입력/출력 커맨드들로 복수의 레이턴시들을 주입하고, 상기 주입된 레이턴시들에 응답하여 상기 제1가상 머신에 대한 복수의 평균 입력/출력 레이턴시들을 결정하도록 더 동작하며; 및 상기 오토-tier링 컨트롤러는 상기 제1가상 머신에 대한 상기 복수의 평균 입력/출력 레이턴시들에 대한 제1 선형 회귀를 수행하는 선형 회귀 모듈을 포함한다.
- [0210] Statement 21. 본 발명의 개념의 실시 예는 Statement 18에 따른 시스템을 포함하며, 상기 제1가상 머신 서버는 상기 입력/출력 필터로부터 복수의 가상 머신들에 대한 상기 성능 데이터를 수집하고 상기 수집된 성능 데이터를 상기 오토-tier링 컨트롤러에 전달하는 오토-tier링 데몬을 포함한다.
- [0212] Statement 22. 본 발명의 개념의 실시 예는 Statement 16에 따른 시스템을 포함하며, 상기 제1가상 머신 서버는 상기 제1가상 머신에 대한 상기 제1데이터 중 임의의 것을 저장하기 위한 제1로컬 캐시를 포함하지 않으며; 및
- [0213] 제2가상 머신 서버는 상기 제2가상 머신에 대한 상기 제2데이터 중 임의의 것을 저장하기 위한 제2로컬 캐시를 포함하지 않는다.
- [0215] Statement 23. 본 발명의 개념의 실시 예는 Statement 1에 따른 방법을 포함하며, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2가상 머신으로 마이그레이션함으로써 야기되는 성능의 변화는, 상기 제1가상 머신의 성능의 개선을 포함한다.
- [0217] Statement 24. 본 발명의 개념의 실시 예는 Statement 1에 따른 방법을 포함하며, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2가상 머신으로 마이그레이션함으로써 야기되는 성능의 변화는, 상기 제1가상 머신의 성능의 감소를 포함한다.
- [0219] Statement 25. 본 발명의 개념의 실시 예는 Statement 1에 따른 방법을 포함하며, 상기 제1가상 머신에 대한 상기 제1데이터의 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션하는 비용은, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션시키는 데 필요한 마이그레이션 시간을 포함한다.
- [0221] Statement 26. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 상기 제2스토리지 tier가 상기 제1스토리지 tier일 수 있다.
- [0223] Statement 27. 본 발명의 개념의 실시 예는 Statement 1에 따른 시스템을 포함하며, 복수의 스토리지 tier들 중 하나의 복수의 스토리지 디바이스들은 동종(homogeneous)일 수 있다.

- [0225] Statement 28. 본 발명의 개념의 실시 예는 다음을 포함하는 방법을 포함한다 :
- [0226] 복수의 스토리지 tier들의 제1가상 머신의 성능을 모델링하는 제1성능 데이터를 수신하는 단계;
- [0227] 상기 복수의 스토리지 tier들의 제2가상 머신의 성능을 모델링하는 제2성능 데이터를 수신하는 단계;
- [0228] 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스를 각각과 관련시키는 리소스 매트릭스를 생성하는 단계;
- [0229] 상기 리소스 매트릭스를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중 제1스토리지 tier를 선택하는 단계;
- [0230] 상기 리소스 매트릭스를 사용하여 상기 제2가상 머신에 대한 제2데이터를 저장하기 위해 상기 복수의 스토리지 tier들의 제2스토리지 tier를 선택하는 단계; 및
- [0231] 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션시키는 단계를 포함하되,
- [0232] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하고,
- [0233] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2 데이터를 상기 제2스토리지 tier로 이동시킴으로써 초래되는 성능 변화를 고려하고,
- [0234] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로, 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 이동시키는 이동 비용을 고려한다.
- [0236] Statement 29. 본 발명의 개념의 실시 예는 Statement 28에 따른 방법을 포함하며, 상기 제2스토리지 tier가 상기 제1스토리지 tier일 수 있다.
- [0238] Statement 30. 본 발명의 개념의 실시 예는 Statement 28에 따른 방법을 포함하며, 제1성능 데이터를 수신하고, 제2성능 데이터를 수신하고, 리소스 매트릭스를 생성하고, 제1스토리지 tier를 선택하고, 및 제2스토리지 tier를 선택하는 것은 마이그레이션 에포크 당 한번 수행된다.
- [0240] Statement 31. 본 발명의 개념의 실시 예는 Statement 30에 따른 방법을 포함하며, 제1성능 데이터를 수신하고 제2성능 데이터를 수신하는 것이 모니터링 에포크 당 한번 수행된다.
- [0242] Statement 32. 본 발명의 개념의 실시 예는 Statement 31에 따른 방법을 포함하며, 상기 마이그레이션 에포크는 복수의 모니터링 에포크들을 포함한다.
- [0244] Statement 33. 본 발명의 개념의 실시 예는 Statement 28에 따른 방법을 포함하며, 상기 리소스 매트릭스를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 제1스토리지 tier를 상기 복수의 스토리지 tier들로부터 선택하는 단계는, 상기 복수의 스토리지 tier들 각각에 대한 상기 리소스 매트릭스 및 전문성 정보를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중에서 상기 제1 스토리지 tier를 선택하는 단계를 포함하고; 및 상기 리소스 매트릭스를 사용하여 상기 제2가상 머신에 대한 제2데이터를 저장하기 위한 제2스토리지 tier를 상기 복수의 스토리지 tier들로부터 선택하는 단계는, 상기 복수의 스토리지 tier들 각각에 대한 상기 리소스 매트릭스 및 전문성 정보를 사용하여 상기 제2가상 머신에 대한 제2데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중에서 상기 제2스토리지 tier를 선택하는 단계를 포함한다.
- [0246] Statement 34. 본 발명의 개념의 일 실시 예는 Statement 33에 따른 방법을 포함하며, 상기 복수의 스토리지 tier들 각각에 대한 상기 전문성 정보는 상기 복수의 스토리지 tier들 각각이 상기 복수의 리소스 각각을 지원하는지 여부를 식별한다.

- [0248] Statement 35. 본 발명의 개념의 실시 예는 Statement 34에 따른 방법을 포함하며, 상기 복수의 리소스들은 처리량, 대역폭, 용량 및 쓰기 증폭을 포함한다.
- [0250] Statement 36. 본 발명의 개념의 실시 예는 Statement 34에 따른 방법을 포함하며, 상기 복수의 스토리지 티어들에서 상기 제1가상 머신의 성능을 모델링하는 상기 제1성능 데이터는, 상기 제1가상 머신에 대한 평균 입력/출력 레이턴시와 주입된 레이턴시들을 상호 관련시키는 복수의 데이터 포인트들을 포함한다.
- [0252] Statement 37. 본 발명의 개념의 실시 예는 Statement 36에 따른 방법을 포함하며, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 티어들과 상기 복수의 스토리지 티어들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계는, 상기 제1가상 머신에 대한 평균 입력/출력 레이턴시들을 주입된 레이턴시들과 상호 관련시키는 상기 복수의 데이터 포인트들에 대한 선형 회귀를 수행하는 단계를 포함한다.
- [0254] Statement 38. 본 발명의 개념의 실시 예는 Statement 37에 따른 방법을 포함하며, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 티어들과 상기 복수의 스토리지 티어들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계는, 상기 선형 회귀를 사용하여 상기 복수의 스토리지 티어들 각각에 대한 상기 제1가상 머신의 성능을 예측하는 단계를 더 포함한다.
- [0256] Statement 39. 본 발명의 개념의 실시 예는 Statement 37에 따른 방법을 포함하며, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 티어들과 상기 복수의 스토리지 티어들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계는, 상기 복수의 리소스들 각각에 따라 상기 복수의 스토리지 티어들 각각에서 상기 제1가상 머신에 대한 성능 매트릭스를 결정하는 단계를 더 포함한다.
- [0258] Statement 40. 본 발명의 개념의 일 실시 예는 Statement 39에 따른 방법을 포함하며, 상기 복수의 리소스들 각각에 따라 상기 복수의 스토리지 티어들 각각에서 상기 제1가상 머신에 대한 성능 매트릭스를 결정하는 단계는, 상기 복수의 스토리지 티어들 각각에서 상기 제1가상 머신에 대한 제1성능 메트릭을 상기 복수의 리소스들의 제2리소스에 따라 상기 복수의 스토리지 티어들 각각의 상기 제1가상 머신에 대한 제2성능 메트릭으로 변환하는 단계를 더 포함한다.
- [0260] Statement 41. 본 발명의 개념의 실시 예는 Statement 41에 따른 방법을 포함하며, 상기 복수의 스토리지 티어들의 제1가상 머신의 성능을 모델링하는 제1성능 데이터를 수신하는 단계는:
- [0261] 입력/출력 필터에서 상기 가상 머신으로부터 복수의 제1 입력/출력 커맨드들을 수신하는 단계;
- [0262] 상기 복수의 제1입력/출력 커맨드들에 제1레이턴시를 주입하는 단계; 및
- [0263] 상기 복수의 제1 입력/출력 커맨드들에 대한 상기 가상 머신에 대한 제1평균 입력/출력 레이턴시를 계산하는 단계를 포함한다.
- [0265] Statement 42. 본 발명의 개념의 실시 예는 Statement 41에 따른 방법을 포함하며,
- [0266] 상기 복수의 제1입력/출력 커맨드들에 제1레이턴시를 주입하는 단계는 상기 복수의 제1입력/출력 커맨드들에 복수의 레이턴시를 주입하는 단계를 포함하며;
- [0267] 상기 복수의 제1입력/출력 커맨드들에 대한 상기 가상 머신에 대한 제1평균 입력/출력 레이턴시를 계산하는 단계는 상기 복수의 제1입력/출력 커맨드들에 대한 상기 가상 머신에 대한 복수의 평균 입력/출력 레이턴시들을 계산하는 단계를 포함하고; 및
- [0268] 상기 방법은 상기 복수의 평균 입력/출력 레이턴시들에 대한 선형 회귀를 수행하는 단계를 더 포함한다.
- [0270] Statement 43. 본 발명의 개념의 실시 예는 Statement 28에 따른 방법을 포함하며, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 티어로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2가상 머신으로 마이그레이션함으로써 야기되는 성능의 변화는, 상기 제1가상 머신의 성능의 개선을 포함한다.
- [0272] Statement 44. 본 발명의 개념의 실시 예는 Statement 28에 따른 방법을 포함하며, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 티어로 또는 상기 제2가상 머신에 대한 상기 제2데이터를

상기 제2가상 머신으로 마이그레이션함으로써 야기되는 성능의 변화는, 상기 제1가상 머신의 성능의 감소를 포함한다.

- [0274] Statement 45. 본 발명의 개념의 실시 예는 Statement 28에 따른 방법을 포함하며, 상기 제1가상 머신에 대한 상기 제1데이터의 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션하는 비용은, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션시키는 데 필요한 마이그레이션 시간을 포함한다.
- [0276] Statement 46. 본 발명의 개념의 실시 예는 비 일시적인 저장 매체를 포함하는 물품을 포함하며, 상기 머신에 의해, 복수의 스토리지 tier들의 제1가상 머신의 성능을 모델링하는 제1성능 데이터를 수신하는 단계;
- [0277] 상기 복수의 스토리지 tier들의 제2가상 머신의 성능을 모델링하는 제2성능 데이터를 수신하는 단계;
- [0278] 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계;
- [0279] 상기 리소스 매트릭스를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중 제1스토리지 tier를 선택하는 단계;
- [0280] 상기 리소스 매트릭스를 사용하여 상기 제2가상 머신에 대한 제2데이터를 저장하기 위해 상기 복수의 스토리지 tier들의 제2스토리지 tier를 선택하는 단계; 및
- [0281] 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션시키는 단계,
- [0282] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 복수의 스토리지 tier들 전반의 모든 가상 머신들의 성능을 최적화하고,
- [0283] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2 데이터를 상기 제2스토리지 tier로 이동시킴으로써 초래되는 성능 변화를 고려하고,
- [0284] 상기 제1가상 머신에 대한 상기 제1데이터를 저장하도록 상기 제1스토리지 tier를 선택하고, 상기 제2가상 머신에 대한 상기 제2데이터를 저장하도록 상기 제2스토리지 tier를 선택하는 것은 상기 제1가상 머신에 대한 상기 제1 데이터 중 적어도 하나를 상기 제1스토리지 tier로, 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 이동시키는 이동 비용을 고려한다.
- [0286] Statement 47. 본 발명의 개념의 실시 예는 Statement 46에 따른 물품을 포함하고, 상기 제2스토리지 tier가 상기 제1스토리지 tier일 수 있다.
- [0288] Statement 48. 본 발명의 개념의 실시 예는 Statement 46에 따른 물품을 포함하고, 제1성능 데이터를 수신하고, 제2성능 데이터를 수신하고, 리소스 매트릭스를 생성하고, 제1스토리지 tier를 선택하고, 및 제2스토리지 tier를 선택하는 것은 마이그레이션 에포크 당 한번 수행된다.
- [0290] Statement 49. 본 발명의 개념의 실시 예는 Statement 48에 따른 물품을 포함하고, 제1성능 데이터를 수신하고 제2성능 데이터를 수신하는 것이 모니터링 에포크 당 한번 수행된다.
- [0292] Statement 50. 본 발명의 개념의 실시 예는 Statement 49에 따른 물품을 포함하고, 상기 마이그레이션 에포크는 복수의 모니터링 에포크들을 포함한다.
- [0294] Statement 51. 본 발명의 개념의 실시 예는 Statement 46에 따른 물품을 포함하고, 상기 리소스 매트릭스를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 제1스토리지 tier를 상기 복수의 스토리지 tier들로부터 선택하는 단계는, 상기 복수의 스토리지 tier들 각각에 대한 상기 리소스 매트릭스 및 전문성 정보를 사용하여 상기 제1가상 머신에 대한 제1데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중에서 상기 제1스토리지 tier를 선택하는 단계를 포함하고; 및 상기 리소스 매트릭스를 사용하여 상기 제2가상 머신에 대한 제

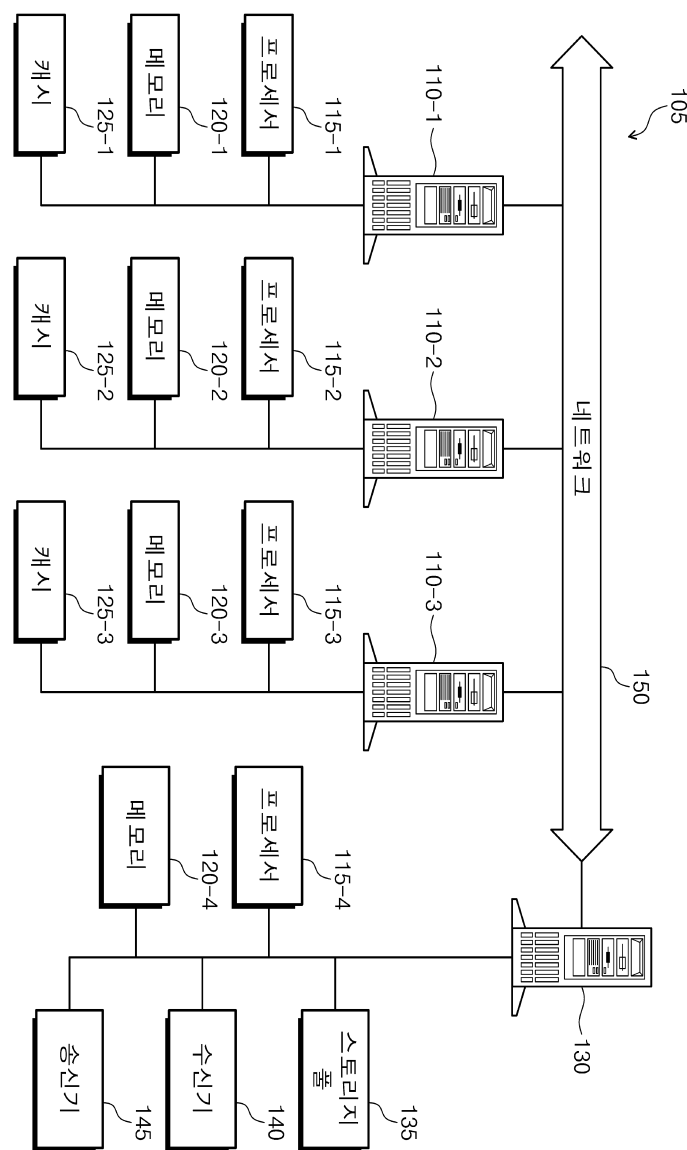
2데이터를 저장하기 위해 제2스토리지 tier를 상기 복수의 스토리지 tier들로부터 선택하는 단계는, 상기 복수의 스토리지 tier들 각각에 대한 상기 리소스 매트릭스 및 전문성 정보를 사용하여 상기 제2가상 머신에 대한 제2데이터를 저장하기 위해 상기 복수의 스토리지 tier들 중에서 상기 제2스토리지 tier를 선택하는 단계를 포함한다.

- [0296] Statement 52. 본 발명의 개념의 실시 예는 Statement 51에 따른 물품을 포함하고, 상기 복수의 스토리지 tier들 각각에 대한 상기 전문성 정보는 상기 복수의 스토리지 tier들 각각이 상기 복수의 리소스 각각을 지원하는지 여부를 식별한다.
- [0298] Statement 53. 본 발명의 개념의 실시 예는 Statement 52에 따른 물품을 포함하고, 상기 복수의 리소스들은 처리량, 대역폭, 용량 및 쓰기 증폭을 포함한다.
- [0300] Statement 54. 본 발명의 개념의 실시 예는 Statement 52에 따른 물품을 포함하고, 상기 복수의 스토리지 tier들에서 상기 제1가상 머신의 성능을 모델링하는 상기 제1성능 데이터는, 상기 제1가상 머신에 대한 평균 입력/출력 레이턴시와 주입된 레이턴시들을 상호 관련시키는 복수의 데이터 포인트들을 포함한다.
- [0302] Statement 55. 본 발명의 개념의 실시 예는 Statement 54에 따른 물품을 포함하고, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계는, 상기 제1가상 머신에 대한 평균 입력/출력 레이턴시들을 주입된 레이턴시들과 상호 관련시키는 상기 복수의 데이터 포인트들에 대한 선형 회귀를 수행하는 단계를 포함한다.
- [0304] Statement 56. 본 발명의 개념의 실시 예는 Statement 55에 따른 물품을 포함하고, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계는, 상기 선형 회귀를 사용하여 상기 복수의 스토리지 tier들 각각에 대한 상기 제1가상 머신의 성능을 예측하는 단계를 더 포함한다.
- [0306] Statement 57. 본 발명의 개념의 실시 예는 Statement 55에 따른 물품을 포함하고, 상기 제1성능 데이터와 상기 제2성능 데이터를 사용하여, 상기 제1가상 머신 및 상기 제2가상 머신을 상기 복수의 스토리지 tier들과 상기 복수의 스토리지 tier들 각각에 대한 복수의 리소스들 각각과 관련시키는 리소스 매트릭스를 생성하는 단계는, 상기 복수의 리소스들 각각에 따라 상기 복수의 스토리지 tier들 각각에서 상기 제1가상 머신에 대한 성능 메트릭들을 결정하는 단계를 더 포함한다.
- [0308] Statement 58. 본 발명의 개념의 실시 예는 Statement 57에 따른 물품을 포함하고, 상기 복수의 리소스들 각각에 따라 상기 복수의 스토리지 tier들 각각에서 상기 제1가상 머신에 대한 성능 메트릭스를 결정하는 단계는, 상기 복수의 스토리지 tier들 각각에서 상기 제1가상 머신에 대한 제1성능 메트릭을 상기 복수의 리소스들의 제2리소스에 따라 상기 복수의 스토리지 tier들 각각의 상기 제1가상 머신에 대한 제2성능 메트릭으로 변환하는 단계를 더 포함한다.
- [0310] Statement 59. 본 발명의 개념의 실시 예는 Statement 57에 따른 물품을 포함하고, 상기 복수의 스토리지 tier들의 제1가상 머신의 성능을 모델링하는 제1성능 데이터를 수신하는 단계는:
- [0311] 입력/출력 필터에서 상기 가상 머신으로부터 제1 입력/출력 커맨드를 수신하는 단계;
- [0312] 상기 제1입력/출력 커맨드에 제1레이턴시를 주입하는 단계; 및
- [0313] 상기 제1 입력/출력 커맨드에 대한 상기 가상 머신에 대한 제1평균 입력/출력 레이턴시를 계산하는 단계를 포함한다.
- [0315] Statement 60. 본 발명의 개념의 실시 예는 Statement 59에 따른 물품을 포함하고,
- [0316] 상기 복수의 제1입력/출력 커맨드들에 제1레이턴시를 주입하는 단계는 상기 복수의 제1입력/출력 커맨드들에 복수의 레이턴시를 주입하는 단계를 포함하며;
- [0317] 상기 복수의 제1입력/출력 커맨드들에 대한 상기 가상 머신에 대한 제1평균 입력/출력 레이턴시를 계산하는 단계는 상기 복수의 제1입력/출력 커맨드들에 대한 상기 가상 머신에 대한 복수의 평균 입력/출력 레이턴시들을 계산하는 단계를 포함하고; 및

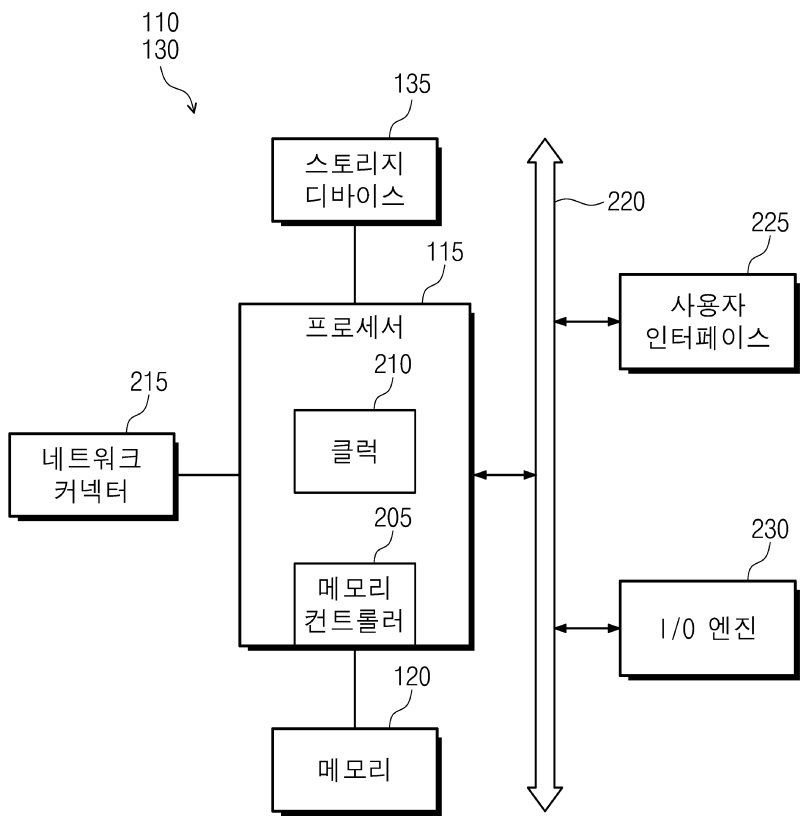
- [0318] 상기 비 일시적 저장 매체는 머신에 의해 실행될 때 상기 복수의 평균 입력/출력 레이턴시의 선형 회귀를 수행하는 결과를 초래하는 그것에 대한 저장된 추가 커맨드들을 포함한다.
- [0320] Statement 61. 본 발명의 개념의 실시 예는 Statement 46에 따른 물품을 포함하고, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2가상 머신으로 마이그레이션함으로써 야기되는 성능의 변화는, 상기 제1가상 머신의 성능의 개선을 포함한다.
- [0322] Statement 62. 본 발명의 개념의 실시 예는 Statement 46에 따른 물품을 포함하고, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2가상 머신으로 마이그레이션함으로써 야기되는 성능의 변화는, 상기 제1가상 머신의 성능의 감소를 포함한다.
- [0324] Statement 63. 본 발명의 개념의 실시 예는 Statement 46에 따른 물품을 포함하고, 상기 제1가상 머신에 대한 상기 제1데이터의 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션하는 비용은, 상기 제1가상 머신에 대한 상기 제1데이터 중 적어도 하나를 상기 제1스토리지 tier로 또는 상기 제2가상 머신에 대한 상기 제2데이터를 상기 제2스토리지 tier로 마이그레이션시키는 데 필요한 마이그레이션 시간을 포함한다.
- [0326] 결론적으로, 본 명세서에 기술된 실시 예에 대한 다양한 변경의 관점에서, 상세한 설명 및 첨부된 자료는 단지 예시적인 것으로 의도되며, 본 발명의 개념의 범위를 제한하는 것으로 간주되어서는 안된다. 그러므로, 본 발명의 개념으로서 청구되는 것은 이하의 청구 범위 및 그 균등물의 범위 및 사상 내에 있을 수 있는 모든 변형이다.

도면

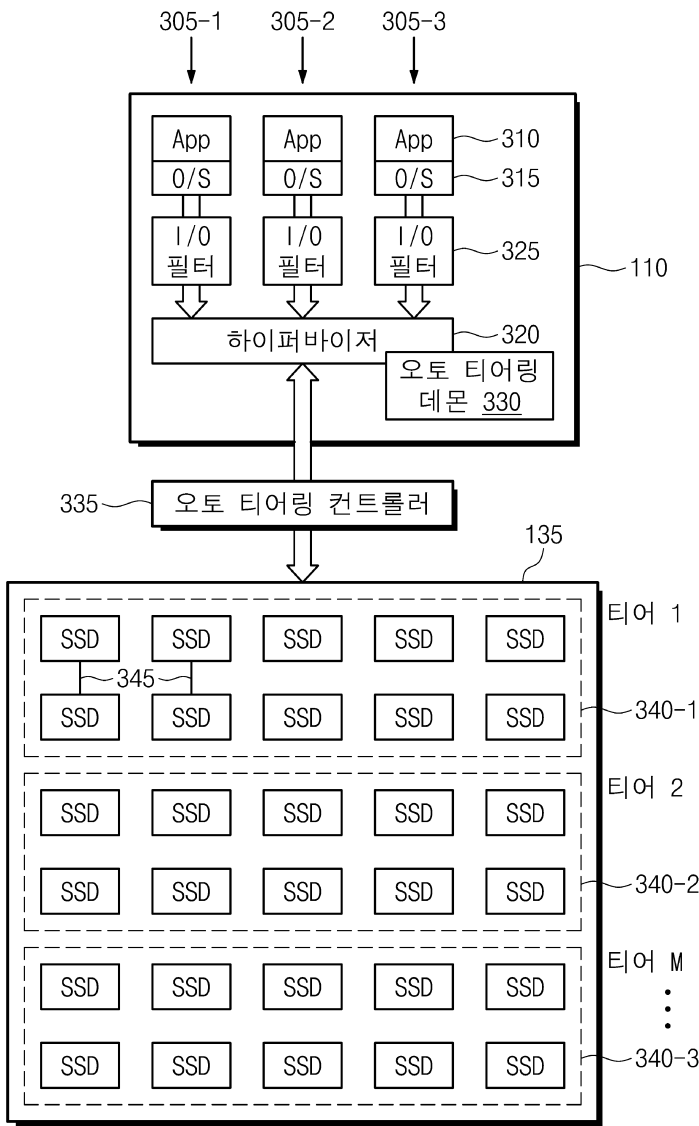
도면1



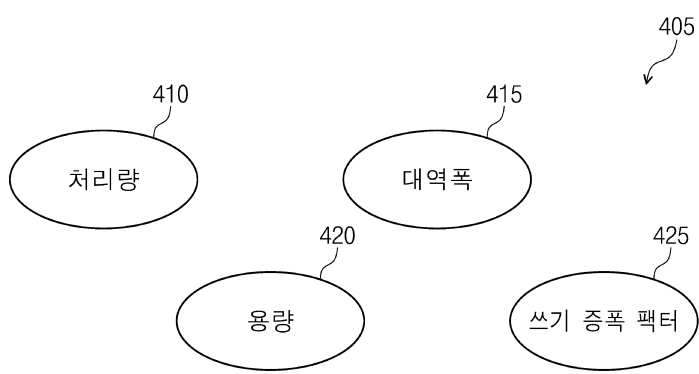
도면2



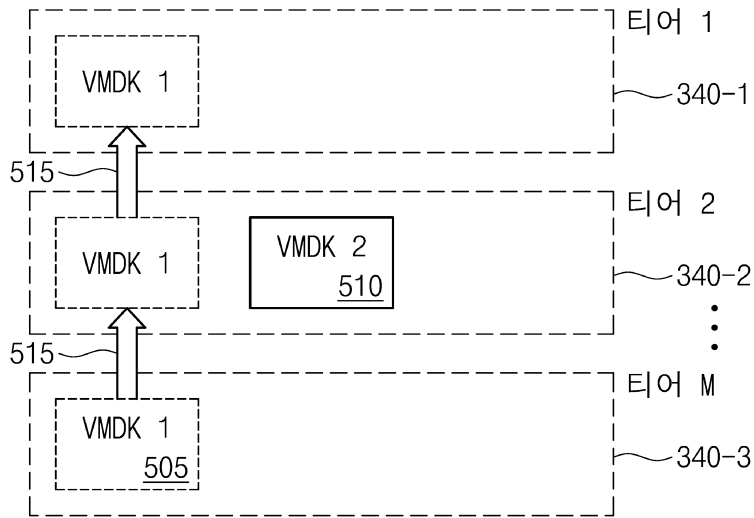
도면3



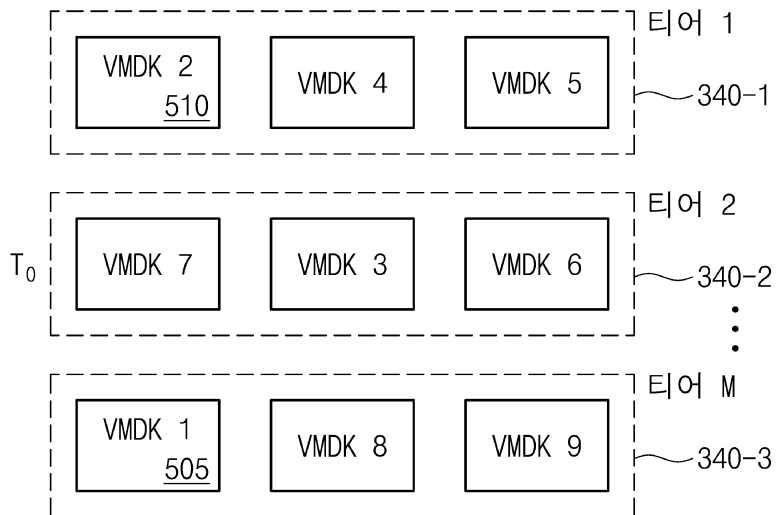
도면4



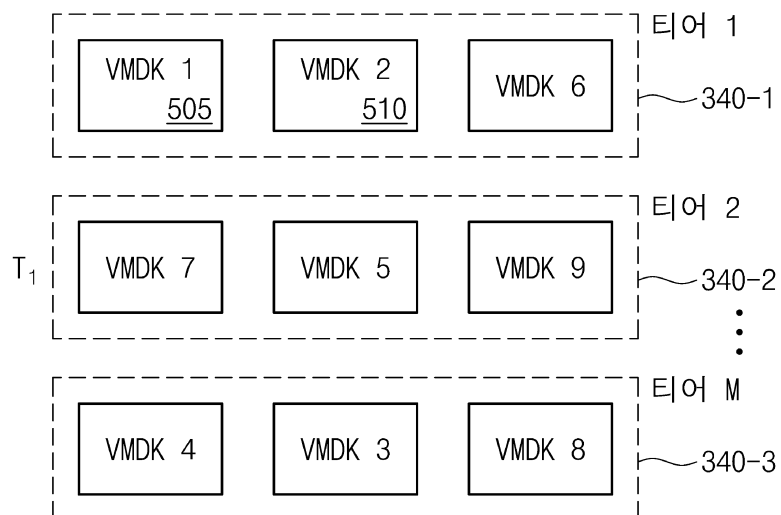
도면5



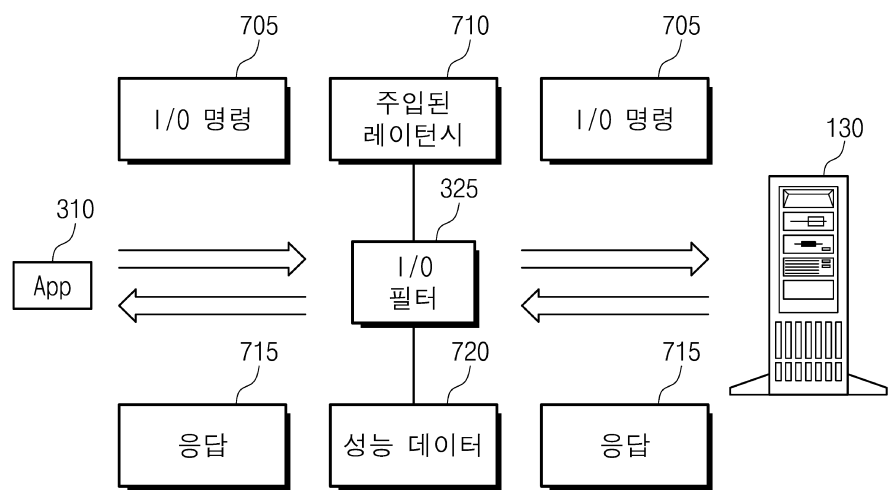
도면6a



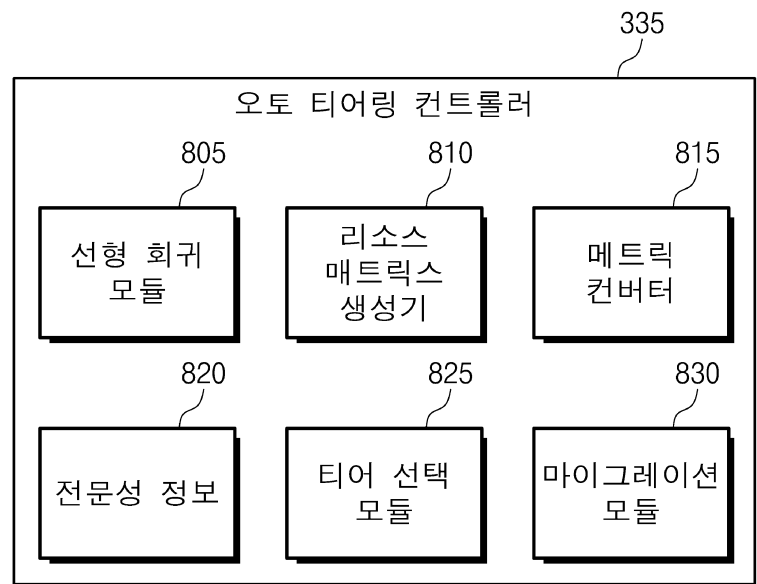
도면6b



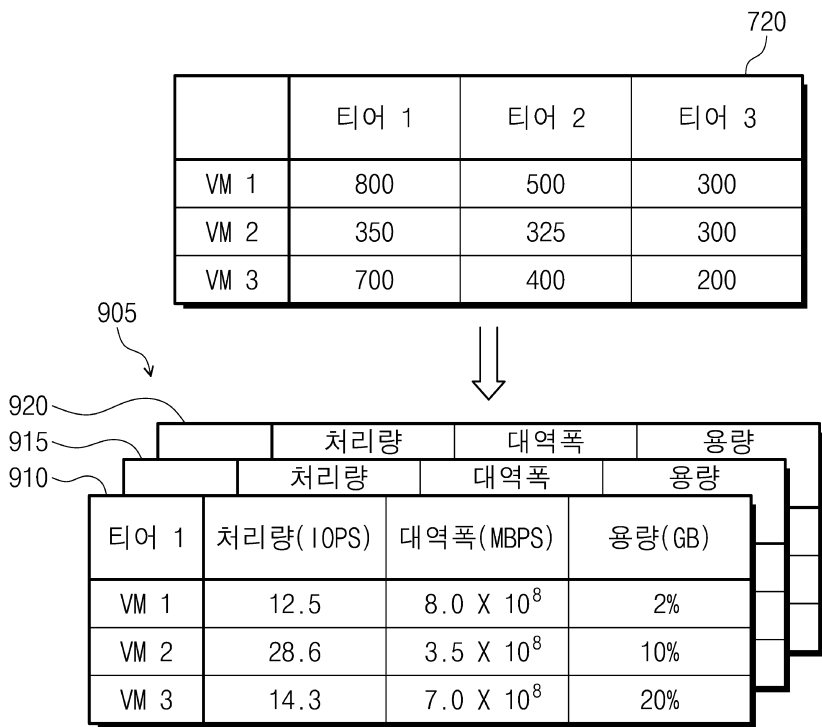
도면7



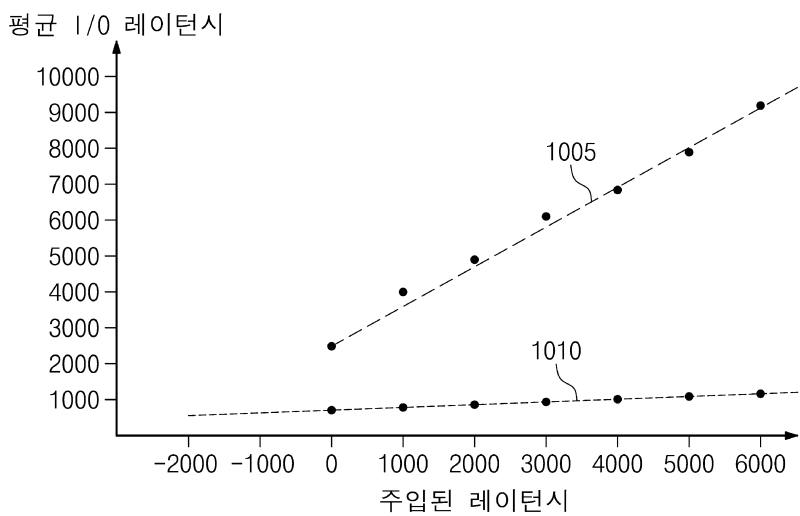
도면8



도면9



도면10



도면11

820

전문성 정보			
	처리량(IOPS)	대역폭(MBPS)	스토리지 크기(GB)
티어 1	1	1	0
티어 2	0	1	0
티어 3	0	0	1

도면12

820

전문성 정보

905

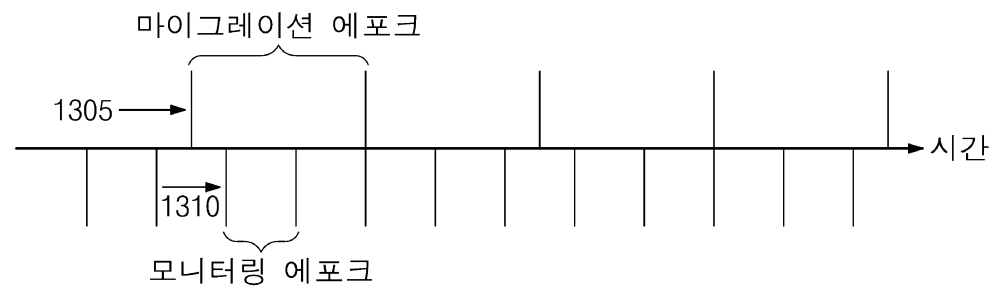
리소스 매트릭스

=

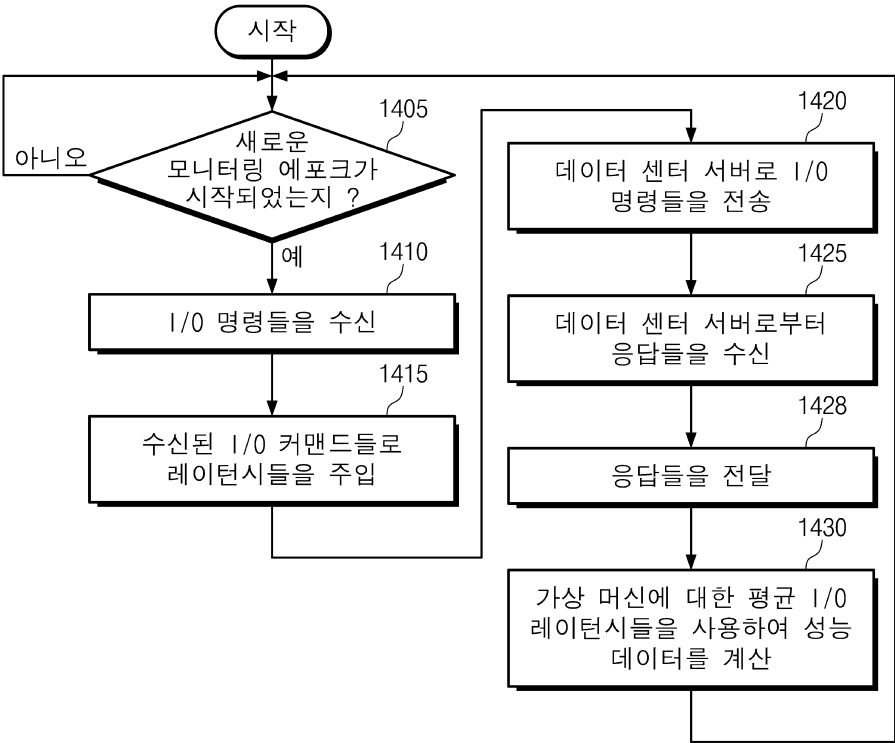
1205

현재의 성능 스코어 매트릭스			
	티어 1	티어 2	티어 3
VM 1	0.55	0.25	0.30
VM 2	...	...	...
VM 3	...	...	...

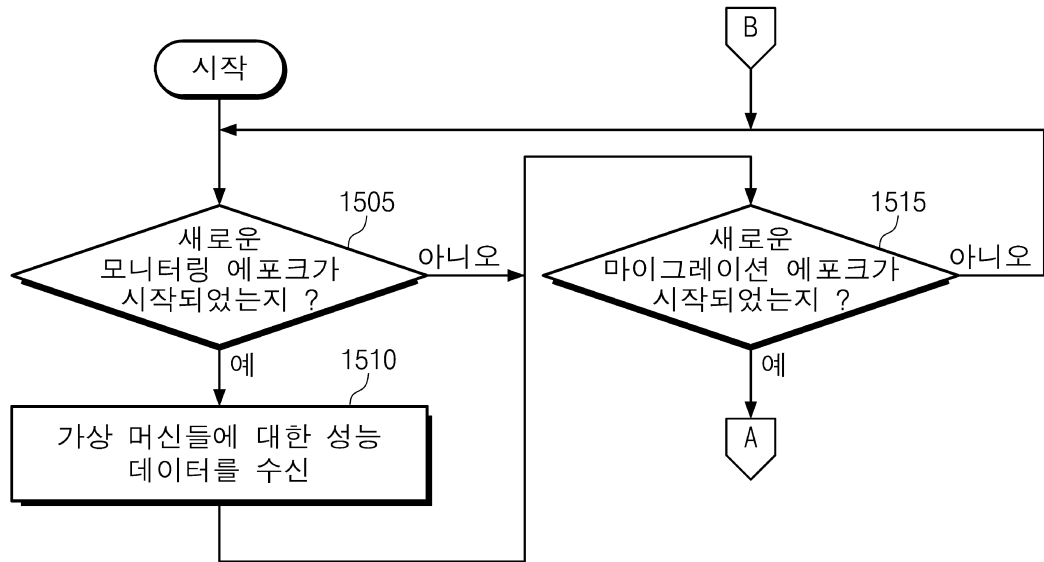
도면13



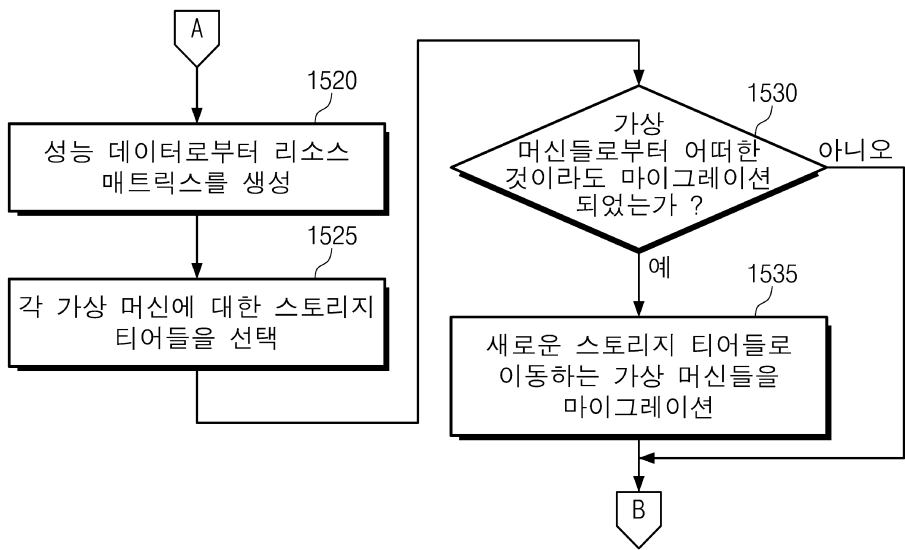
도면14



도면15a



도면15b



도면16

