(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년05월06일
(11) 등록번호 10-0826833
(24) 등록일자 2008년04월25일

(51) Int. Cl.
G06F 9/46 (2006.01) *G06F 9/00* (2006.01)
(21) 출원번호 10-2005-7014631
(22) 출원일자 2005년08월09일
심사청구일자 2006년03월10일
번역문제출일자 2005년08월09일
(65) 공개번호 10-2006-0023951
(43) 공개일자 2006년03월15일
(86) 국제출원번호 PCT/US2003/027304
국제출원일자 2003년08월29일
(87) 국제공개번호 WO 2004/081789
국제공개일자 2004년09월23일
(30) 우선권주장
10/384,973 2003년03월10일 미국(US)
(56) 선행기술조사문헌
US20020091746A1
US20020152305 A
전체 청구항 수 : 총 10 항

(73) 특허권자
인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드
(72) 발명자
콜맨, 데이비드, 윌리
미국 06812 코넥티커트주 뉴 페어필드 콜드 스프링드라이브 8
프로에리치, 스티븐, 이.
미국 06810 코넥티커트주 댄버리 마크 로드 10
(뒷면에 계속)
(74) 대리인
장수길, 주성민

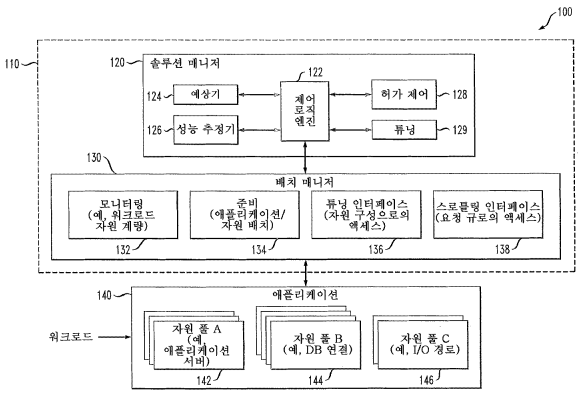
심사관 : 김견수

(54) 가변 워크로드의 존재시 컴퓨팅 배치 관리 방법 및 장치

(57) 요 약

워크로드 레벨을 가변시키는 것에 기반해서 컴퓨팅 환경 내의 하나 이상의 자원의 배치를 관리하기 위한 자동화 또는 자율적인 기술. 이 자동화된 기술은 컴퓨팅 환경에 연관된 데이터에 기반해 장래 워크로드 레벨을 예측하는 것을 포함할 수 있다. 그 후, 현재 자원 배치가 장래 워크로드 레벨을 만족시키기에 불충분한지, 충분한지 또는 너무 충분하지를 판단하기 위한 추정이 실행된다. 그 후, 현재 자원 배치가 장래 워크로드 레벨을 만족시키기에 불충분 또는 과도하게 충분하다고 추정되는 때에는 하나 이상의 동작들이 취해진다. 동작은 자원 준비(resource provisioning), 자원 튜닝(resource tuning) 및/또는 허가 제어(admission control)를 포함할 수 있다.

대표도



(72) 발명자

헬러스틴, 조셉, 엘.

미국 10562 뉴욕주 오씨닝 올덴 로드 41

시엉, 로렌스, 에스.

미국 94024 캘리포니아수 로스 알토스 오크 스코트 브라이트 1340

라쎄트리, 에드윈, 리치에

미국 95032 캘리포니아주 로스 가토스 빈야드 코트 123

머머트, 토드, 윌리암

미국 10512 뉴욕주 카멜 글렌뷰 드라이브 넘버6 90

라가바채리, 머컨드

미국 10505 뉴욕주 밸드윈 플레이스 루미스 드라이브 17

러셀, 랜스, 워렌

미국 95023 캘리포니아주 홀리스터 버샐리스 1290

슈렌드라, 마헤스와란

미국 10520 뉴욕주 크로튼 온 허드슨 그랜드 스트리트 64

와이다, 노셔, 카바스

미국 95037 캘리포니아주 모간 힐 화이트 클라우드 코트14711

예, 펭

미국 94538 캘리포니아주 프리몬트 세다 우드 드라이브43278

특허청구의 범위

청구항 1

워크로드 레벨을 가변하는 것에 기반해서 컴퓨팅 환경 내의 복수의 서버들의 배치(deployment)를 관리하는 자동화된 방법에 있어서,

상기 컴퓨팅 환경에 연관된 데이터에 기반해 장래 워크로드 레벨을 예측하는 단계 - 상기 예측하는 단계는 예상 범위(forecast horizon)에 기반해서 예상하는 단계를 포함함- ;

상기 서버들의 현재 배치가 상기 장래 워크로드 레벨을 충족시키기에 불충분한지(insufficient), 충분한지(sufficient) 또는 너무 충분한지(overly sufficient)를 추정하는 단계; 및

상기 서버들의 현재 배치가 상기 장래 워크로드 레벨을 충족시키기에 불충분하거나 너무 충분한 것으로 추정되면, 하나 이상의 동작들이 취해지도록 하는 단계

를 포함하며,

상기 동작이 취해지도록 하는 단계는 상기 서버들의 현재 배치가 불충분하다고 추정되면, 하나 이상의 서버를 추가함으로써 상기 장래 워크로드 레벨에 대처하도록 하는 단계를 더 포함하고,

상기 동작이 취해지도록 하는 단계는 상기 서버들의 현재 배치가 너무 충분하다고 추정되면, 하나 이상의 서버를 제거함으로써 상기 장래 워크로드 레벨에 대처하도록 하는 단계를 더 포함하며,

상기 예상 범위는, 상기 서버들의 현재 배치가 상기 장래 워크로드 레벨을 충족시키기에 불충분하거나 너무 충분한 것으로 추정되는 경우, 상기 하나 이상의 동작들 중 적어도 하나를 실행하기 위해 필요한 시간의 함수에 따라 선택적으로 변경될 수 있는 방법.

청구항 2

제1항에 있어서,

상기 하나 이상의 서버의 모니터링을 통해서, 상기 장래 워크로드 레벨 예측 단계에 의해 사용되는, 상기 컴퓨팅 환경과 연관된 상기 데이터를 획득하는 단계를 더 포함하는 방법.

청구항 3

삭제

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 동작이 취해지도록 하는 단계는 상기 서버들과 연관된 하나 이상의 구성 파라미터들의 튜닝을 일으키는 단계를 더 포함하는 방법.

청구항 6

제1항에 있어서,

상기 동작이 취해지도록 하는 단계는 상기 서버들 상의 허가 큐들(admission queues)의 조작을 일으키는 단계를 더 포함하는 방법.

청구항 7

제1항에 있어서,

상기 추정하는 단계는 하나 이상의 서비스 목표들에 기반해서, 상기 서버들의 현재 배치가 불충분한지, 충분한

지 또는 너무 충분한지를 추정하는 단계를 더 포함하는 방법.

청구항 8

삭제

청구항 9

워크로드 레벨들을 가변시키는 것에 기반해서 컴퓨팅 환경 내의 복수의 서버들의 배치를 관리하기 위한 장치에 있어서,

메모리; 및

상기 메모리에 연결되어서 청구항 제1항, 제2항, 제5항, 제6항 및 제7항 중 어느 한 항에 기재된 방법을 수행하도록 작동되는 적어도 하나의 프로세서

를 포함하는 장치.

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

워크로드 레벨들을 가변시키는 것에 기반해서 컴퓨팅 환경 내의 복수의 서버들의 배치를 관리하기 위한 기계 판독 가능 매체에 있어서, 실행시에 청구항 제1항, 제2항, 제5항, 제6항 및 제7항 중 어느 한 항에 기재된 방법을 구현하는 하나 이상의 프로그램들을 포함하는 기계 판독 가능 매체.

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

워크로드 레벨들을 가변하는 것에 기반해서 컴퓨팅 환경 내의 복수의 서버들의 배치를 관리하기 위한 자동화된 시스템에서,

(i) 상기 컴퓨팅 환경에 연관된 데이터에 기반해 장래 워크로드 레벨을 예측하고,

(ii) 상기 서버들의 현재 배치가 상기 장래 워크로드 레벨을 충족시키기에 불충분한지, 충분한지 및 너무 충분한지를 추정하도록 동작하고, 메모리 및 그에 연결된 적어도 하나의 프로세서를 포함하는 솔루션 매니저(solution manager) 모듈; 및

(i) 상기 컴퓨팅 환경에 연관된 상기 데이터를 상기 솔루션 매니저 모듈에 제공하고,

(ii) 상기 서버들의 현재 배치가 상기 장래 워크로드 레벨을 충족시키기에 불충분하거나 너무 충분한 것으로 상기 솔루션 매니저 모듈에 의해 추정되면, 상기 솔루션 매니저 모듈에 응답해서, 하나 이상의 동작이 취해지도록 작동되며, 메모리 및 그에 연결된 적어도 하나의 프로세서를 포함하는 배치 매니저(deployment manager) 모듈

을 포함하고,

상기 취해지는 하나 이상의 동작은, 상기 서버들의 현재 배치가 불충분하다고 추정되면, 하나 이상의 서버를 추가함으로써 상기 장래 워크로드 레벨에 대처하도록 하는 것을 포함하고,

상기 취해지는 하나 이상의 동작은, 상기 서버들의 현재 배치가 너무 충분하다고 추정되면, 하나 이상의 서버를 제거함으로써 상기 장래 워크로드 레벨에 대처하도록 하는 것을 포함하며,

상기 배치 매니저 모듈은,

(i) 워크로드 데이터로의 액세스를 제공하기 위한 모니터링 모듈;

(ii) 자원 준비를 수행하기 위한 준비 모듈(provisioning module);

(iii) 상기 서버들 중 하나 이상과 연관되는 하나 이상의 구성 파라미터를 변경하기 위한 튜닝 인터페이스 모듈; 및

(iv) 상기 서버들 중 하나 이상에서, 하나 이상의 허가 큐의 조작을 일으키기 위한 스로틀링 인터페이스 모듈(throttling interface module)

을 포함하는 시스템.

청구항 26

청구항 26은(는) 설정등록료 납부시 포기되었습니다.

제25항에 있어서,

상기 서버들이 애플리케이션의 실행을 구현하기 위해 배치될 수 있는 시스템.

청구항 27

제25항에 있어서,

상기 솔루션 매니저 및 상기 배치 매니저 중 적어도 하나가 자율적으로 동작하는 시스템.

청구항 28

제25항에 있어서,

상기 하나 이상의 동작은 서버 준비(server provisioning), 서버 튜닝(server tuning) 및 허가 제어(admission control) 중 적어도 하나를 포함하는 시스템.

청구항 29

청구항 29은(는) 설정등록료 납부시 포기되었습니다.

제25항에 있어서,

상기 솔루션 매니저는 하나 이상의 서비스 목표들에 기반해서, 서버들의 현재 배치가 불충분한지, 충분한지 또는 너무 충분한지를 추정하는 시스템.

청구항 30

삭제

명 세 서

기 술 분 야

<1> 본 출원은 대리인 포대 번호 YOR920030050US1의 제목 "Methods and Apparatus for Performing Adaptive and Robust Prediction"의 본 명세서와 동시에 출원되고 참조로서 본 명세서에 합체되는 미국 특허출원에 관련된다.

<2> 본 발명은 일반적으로 컴퓨팅 시스템 또는 네트워크에 관련되고 더 구체적으로, 가변 워크로드(workload)의 존재시 그 시스템 또는 네트워크와 연관된 컴퓨팅 배치(deployment)를 관리하기 위한 기술에 관한 것이다.

배 경 기 술

<3> 컴퓨팅 시스템 및 네트워크 내의 컴퓨팅 자원의 배치를 관리하는 것에 있어서 문제는 가변 트래픽을 다루는 것이다. 예컨대, 월드 와이드 웹 또는 인터넷과 연관된 컴퓨팅 시스템 또는 컴퓨팅 자원에서, 웹 사이트가 이 웹 사이트와 관련되어 실행되는 하나 이상의 애플리케이션의 사용에 관해 과중한 로드가 걸릴 때에도, 최종 사용자에 문제(예컨대, 늦은 응답 시간)가 되지 않을 정도로 충분한 컴퓨팅 자원(예컨대, 웹 서버, 애플리케이션 서버, 트랜잭션/데이터베이스 서버)을 가지는 것이 중요하다. 알려진 바와 같이, 애플리케이션은 일반적으로 하나 이상의 특정 기능(예컨대, 공급망 관리(supply chain management))을 수행하기 위해 설계된 하나 이상의 컴퓨터 프로그램을 가리킨다.

<4> 특정 애플리케이션을 지원하는 배치의 크기를 조정하기 위한 하나의 접근 방식은 기대되는 워크로드 트래픽 패턴을 추정하고 수용량 계획 접근 방식을 사용해서 최대 기대 로드(peak anticipated load)를 수용하기에 충분한 자원을 사용하는 것이다. 이러한 정적인 배열은 중요한 자원이 사용되지 않는 결과를 초래할 수 있는데, 이는, 대부분 워크로드 트래픽이(예컨대, 현저히 날마다, 주마다 등의 패턴으로) 상당히 동적이기 때문이다.

<5> 상기 접근 방식을 개선하여 다듬은 방식이 있는데, 이는 예상 트래픽의 장기간(예컨대, 하루에서 여러 날까지) 예상에 기반해서 자원 재배치를 스케줄하거나 또는 계획하는 것이다. 이 접근방식도 (예컨대, 판매 프로모션의 성공을 과소평가하기도 하는) 장기간 예상의 정확성에 기반하고 예상하지 못한 사건(예컨대, 9.11 테러시 CNN 웹사이트에서 경험한 것과 같은 뉴스 웹사이트의 트래픽 서지)에 영향을 받을 수도 있으므로 가끔 부적절한 경우가 있다.

<6> 기존 컴퓨팅 배치 접근 방식들의 다른 중요한 단점은 자원 불균형을 조절하기 위해 일반적으로 (예컨대, 전문 오퍼레이터에 의해) 소정 형태의 인간의 관여를 필요로 한다는 것이다.

<7> 따라서, 기존 접근방식보다 더 효율적이고 효과적으로 가변 워크로드를 다루기 위한 컴퓨팅 시스템 또는 네트워크에 연관된, 컴퓨팅 배치를 관리하기 위한 자동적 또는 자치적인 기술을 가지는 것이 바람직할 것이다.

발명의 상세한 설명

<8> 본 발명은 기존 접근방식보다 더 효율적이고 효과적으로 가변 워크로드를 다루기 위한 컴퓨팅 시스템 또는 네트워크에 연관된, 컴퓨팅 배치를 관리하기 위한 자동적(automated) 또는 자율적인(autonomic) 기술을 제공한다.

<9> 발명의 일 실시예에서, 변하는 워크로드 레벨에 기반해서 컴퓨팅 환경 내의 하나 이상의 자원의 배치를 관리하기 위한 기술이 제공된다. 이 기술은 컴퓨팅 환경에 연관된 데이터에 기반해 장래 워크로드 레벨을 예측하는 것을 포함할 수 있다. 그 후, 현재 자원 배치가 장래 워크로드 레벨을 만족시키기에 불충분(insufficient)한지, 충분(sufficient)한지 또는 너무 충분(overly sufficient)한지를 판단하기 위한 추정이 실행된다. 그 후, 현재 자원 배치가 장래 워크로드 레벨을 만족시키기에 불충분하거나 너무 충분한 때에는 하나 이상의 동작들이 취해진다. 동작은 자원 준비(resource provisioning), 자원 튜닝(resource tuning) 및/또는 허가 제어(admission control)를 포함한다.

<10> 다행스럽게도, 본 발명은 가변 워크로드에 직면해서 컴퓨팅 배치를 위한 응답 시간과 같은 서비스 레벨 목표를 사전 관리하는 것이 제공된다. 특히 자동 또는 자율 형식(automated or autonomic fashion)으로 컴퓨팅 배치를 변화시키는 것에 의해, 본 발명에 의해 채용된 기술들은 기대하지못한 워크로드를 수용하는 것에 효과적이다.

<11> 본 발명은 또한, 다행히도 애플리케이션 소유자가 서비스 제공자에 의해 호스트된 애플리케이션의 실행과 연관된 하나 이상의 서비스 목표들의 충족을 확신하도록 하는 방법이 제공된다. 이것은 애플리케이션을 호스트하고 본 명세서에서 제공된 것과 같은 컴퓨팅 배치 관리 시스템을 구현하기 위해 서비스 제공자와 계약한 애플리케이션 소유자에 의해 달성될 수 있다.

<12> 본 발명의 여러 목표, 특징 및 장점이 도시된 실시예의 다음의 설명 및 첨부된 도면으로부터 명백해 질 것이다.

실 시 예

<17> 본 발명은 예시적인 웹 기반 컴퓨팅 네트워크 환경의 문맥으로 아래에서 설명될 것이다. 즉, 관리되는 컴퓨팅 자원(예컨대, 애플리케이션 서버, 데이터 베이스 연결, 입력/출력 경로 등)은 하나 이상의 웹 사이트와 연관된다. 그러나 본 발명은 이런 특정 환경에 한정되는 것은 아니다. 오히려, 본 발명은 더 일반적으로 자동 또는 자율적으로 가변 워크로드 시에 지원 배치를 관리 및 계산하는데 더 바람직한 임의 컴퓨팅 환경에 적용가능하다.

<18> 알려진 대로, "자율적(autonomic)" 컴퓨팅은 일반적으로, 사람의 간섭을 최소화하고 자기 관리 컴퓨팅 시스템으로의 포괄적이고 전체론적인 접근방식을 말하는 것으로, 본 명세서에 참조로서 통합된 피. 혼의 2001년 10월의 IBM research의 "Autonomic Computing: IBM's Perspective on the State of Information Technology"을 참조하라. 이 용어는 의식적인 인지 또는 몰두 없이 중요 기능을 제어하는 신체의 자율 신경계에서 따온 것이다. 더 구체적으로, 자율적 컴퓨팅의 목적중 하나는 전문 오퍼레이터 또는 관리자가 전형적으로 수행해야 하는 일부 또는 모든 일들을 자동화하는 것이다. 그래서, 명세서에 제시된 발명의 원리로부터 인식될 수 있듯이, 본 발명의 컴퓨팅 배치 기술은 자동적 또는 자율적으로 동작할 수 있게 한다.

<19> 시작으로 도 1을 참조해서, 블록도가 본 발명의 실시예에 따른 컴퓨팅 배치 관리 시스템 및 그 시스템이 동작할 수 있는 전체 환경을 도시한다. 도시된 것과 같이, 환경(100)은 컴퓨팅 배치 관리 시스템(100)을 포함한다. 컴퓨팅 배치 관리 시스템(100)은 자체에 솔루션 매니저(120) 및 배치 매니저(130)를 포함한다. 솔루션 매니저(120)는 자체에 제어 로직 엔진(control logic engine)(122), 예상기 모듈(124), 성능 추정기 모듈(126), 허가 제어 모듈(128) 및 튜닝 모듈(129)을 포함한다. 배치 매니저(130)는 자체에 모니터링 모듈(132), 준비 모듈(provisioning module)(134), 튜닝 인터페이스 모듈(136) 및 스로틀링 인터페이스 모듈(throttling interface module)(138)을 포함한다.

<20> 나아가, 도시된 바와 같이 환경(100)은 애플리케이션 레벨(140)을 포함한다. 애플리케이션 레벨, 자체 내에 자원 풀(142)(자원 풀 A는 예컨대 애플리케이션 서버를 포함), 자원 풀(144)(자원 풀 B는 예컨대 데이터베이스 연결을 포함) 및 자원 풀(146)(자원 풀 C는 예컨대 입력/출력 경로를 포함)을 포함한다.

<21> 따라서, 도 1에 도시된 아키텍처는 3가지 레벨로 구성되는데, (a) 애플리케이션 레벨(140으로 표시됨) 및 상기 애플리케이션이 배치할 수 있는 연관된 자원들, (b) 자원의 연결 및 제어를 제공하는 배치 관리 레벨(130으로 표시됨) 및 (c) 서비스 레벨 목표를 관리하기 위해 필요한 동작들을 개시하는데 필요한 계획 및 실시가 분석을

수행하는 솔루션 관리 레벨(120으로 표시됨)이 그것이다.

<22> 애플리케이션 배치는 전형적으로 다양한 유형의 자원(예컨대 HTTP 서버, 애플리케이션 서버, 데이터베이스 서버, 저장소(storage), 연결, I/O 경로 등)의 혼합을 요구한다. 전형적인 컴퓨팅 배치에서(예컨대, 데이터 센터), 이 자원들은 관리되는 풀로부터 이용가능할 수 있다. 도 1은 그런 관리되는 풀 3개(142, 144 및 146)를 도시한다. 애플리케이션에 따라서, 기대되는 워크로드를 충족시키기에 충분한 미리 정해진 수의 각 자원이 관리되는 자원 풀에서의 사용을 위해 이용가능하다. 도 1 및 상술한 기술이 소정의 자원을 언급함에도 본 발명은 임의의 특정 자원에 한정되는 것은 아님이 더 이해되어야 한다. 오히려, 본 발명이 하드웨어 컴포넌트, 소프트웨어 컴포넌트 및 이들의 조합을 포함하는 임의의 모든 유형을 자원을 관리할 수 있는데, 이에 한정되는 것은 아니다. 자원은 애플리케이션 그 자체 또는 이것의 일부일 수도 있다.

<23> 배치 매니저(130)는 애플리케이션 레벨(140)의 관련 자원들과 인터페이스해서 측정/구성 데이터를 (예컨대, 응답 시간 프로브, 유닉스와 같은 운영체제에서 vmstat 데이터, IBM 코퍼레이션의 DB2와 같은 데이터베이스로부터의 스냅 샷 데이터와 같은 자원 의존 센서를 통해서나 공통 정보 모델(Common Information Model)을 사용해 구현된 사용자 지정 인터페이스 또는 표준 인터페이스를 통해서) 모니터링하고 (예컨대, IBM 코퍼레이션의 DB2와 같은 데이터베이스 내의 구성 파라미터를 변경하기 위한 애플리케이션 프로그래밍 인터페이스, IBM 코퍼레이션의 웹스피어 애플리케이션 서버와 같은 애플리케이션 서버상의 노드 에이전트와 같은 자원 의존 효과기(effectuator)를 통해서) 자원들을 제어한다. 그래서, 배치 매니저는 (준비 모듈(134)을 통해서) 자원 준비를 수행할 수 있는데, 예컨대 일부 하드웨어에 대해서, (i) 설치된 운영체제(예컨대, 윈도즈 또는 리눅스가 설치될 수 있는 x86 시스템) 없이 컴퓨터상에서 운영체제를 배치하는 것 또는 새 운영 체제로 컴퓨터상의 존재하는 운영체제를 교체하는 것, (ii) 운영 체제상에 적절한 미들웨어를 배치하는 것, (iii) 데이터와 연관된 애플리케이션을 배치하는 것 및 (iv) 부가된 자원이 애플리케이션을 지원하도록 하는 관련 클러스터 관리/동맹(federation)을 수행하는 것으로부터 시작될 수 있다. 본 준비 능력의 유리한 특성은 예컨대 기대하지 못한 워크로드 서지에 반응과 같이 필요시에 자원을 신속하고 자동으로 추가하는 것을 포함할 뿐만 아니라, 더 이상 필요하지 않을 때 자원을 제거해서 부가 자원의 고비용을 최소화한다.

<24> 추가로, 배치 매니저(130)는 (튜닝 인터페이스 모듈(136)을 통해서) 자원 튜닝에 중요한 자원 구성 파라미터(예컨대, 데이터베이스에 관한 버퍼 풀과 같은 메모리 풀 사이즈, 애플리케이션 서버 내의 ORB(object request broker) 쓰레드(thread) 풀 크기)를 리셋한다. 자원 튜닝은 일반적으로 응답 시간을 최소화하거나 처리량을 최대화하는 것과 같은 목표를 달성하는데 도움이 되는 방식으로 자원과 연관된 하나 이상의 구성 파라미터를 변화시키는 기술을 일반적으로 말한다. 배치 매니저(130)는 또한 (스로틀링 인터페이스 모듈(138)을 통해서) (예컨대, 허가 제어/요청 스로틀링에 관해) 자원 상의 허가 큐(admission queue)를 조작할 수도 있다. 스로틀링은 일반적으로 예컨대, 요청의 유형(예컨대, URL(uniform resource locator)으로 구분될 수 있는 전자 상거래 사이트에서의 구매 대 브라우즈), 오리진(orgin)(선호 고객들) 등과 같은 서비스 클래스를 식별하는 소정의 정책에 기반해 들어오는 요청을 거절하는 것을 말한다. 그런 요청들을 거절하는 것에 의해, 컴퓨팅 배치로 들어오는 로드가 관리 레벨로 줄어든다. 배치 매니저는 (모니터링 모듈(132)을 통해서) 워크로드 데이터(예컨대, 처리량, 응답 시간 등)로의 액세스도 제공한다.

<25> 솔루션 매니저(120)는 특정 애플리케이션 배치에 관한 서비스 목표를 관리하는 책임이 있다. "서비스 목표"는 요청 및/또는 SLA(service level agreement)에 따라 특정된 선호(preference)를 참조할 수 있다. 즉, 예로서, 그런 서비스 목표는 최종 사용자의 만족을 소정 레벨로 보장하면서 어떻게 서비스 애플리케이션이 제3 하부구조(infrastructure)에 호스트 되는지 처리할 수 있다. 알려진 대로, 회사들은 점증적으로 "서비스 제공자(service provider)"로 일반적으로 언급되는 제3자에 의해 제공되는 하부 구조(예컨대, 서버, 네트워크 연결)를 사용해 그들의 애플리케이션을 운용한다. IBM 클로벌 서비스(IBM Global Services)와 같은 많은 회사가 웹 사이트를 호스트하고/호스트하거나 다른 컴퓨터 호스팅 서비스를 제공한다. SLA는 서비스 제공자의 기대가 고객과 협상될 수 있는 수단을 제공한다. 애플리케이션 소유자와 서비스 제공자 사이의 SLA는 이 호스팅 서비스의 기간(term) 및 조건을 정의한다. 예컨대, SLA는 기대 응답 시간(expected response time), 네트워크 및/또는 서버에서 처리량의 밴드폭, 디스크 공간 사용, 사용가능성 즉, 고장 시 양호 회복 시간 같은 네트워크 및 서버 자원의 업 타임(up-time) 및 서비스의 다양한 레벨에 대한 대가를 포함할 수 있다. 그러나 서비스 목표는 전형적으로 정당한 결과를 가지는 SLA로부터 나와서는 안 된다는 것을 인식되어야 한다. 서비스 레벨 목표는 엔터프라이즈(enterprise) 내에서 (예컨대, 정보 기술(IT) 부서 및 온라인 구매 주문 시스템을 배치하고 있을 수도 있는 구매 부서 사이에) 자주 교섭될 수 있다. 또한, 사용자의 좋은 경험을 위해서 전자 상거래 사이트 또는 심지어 구글(Google) 같은 곳에서도 응답 시간과 같은 소정의 것에 관해 양질의 서비스 레벨을 유지하기를 원할

수 있다.

<26> 따라서, 특정 서비스 목표를 충분히 유지하기 위해서, 솔루션 매니저(120)는 제어 로직 엔진(122)을 따라서 (i) 언제 동작이 취해질 필요가 있는지 및 (ii) 어떤 동작이 취해질지를 결정한다. 제어 로직 엔진(122)은 후술하는 것과 같이 예상기 모듈(124), 성능 추정기 모듈(126), 허가 제어 모듈(128) 및 튜닝 모듈(129)을 따라서 이런 일들을 완료한다. 즉, 제어 로직 엔진(CLE)(122)은 솔루션 매니저(120) 내의 다른 모듈에 의해 제공되는 기능들을 위한 제어기로서 역할하는 것이 이해되어야 한다. 그러나 블록(120)에 도시된 기능적인 배열은 본질적으로 도시적인 것이고 그래서 솔루션 매니저에 의해 제공되는 기능을 제어하기 위한 다른 배열도 본 발명의 원리의 범주 내에서 채용될 수 있다는 것을 이해해야한다.

<27> 솔루션 매니저의 일 장점인 특성은 서비스 목표를 미리 관리하는 능력인데, 그래서 가능한 위반을 최소화/회피한다. 이것은 예측의 서비스 레벨위반이 예상되는지를 확인하기 위해 (배치 매니저 내의 모니터링 컴포넌트로부터 획득된) 워크로드 상에서의 (예상기 모듈(124)을 통한) 예상과 (성능 추정기(126)를 통한) 성능 추정과의 조합으로 완수될 수 있다. 바람직한 실시예에서, 제어 로직 컴포넌트(122)는 본질적으로 규칙 기반 컴포넌트로서 예상 및 성능 추정 결과에 기반해, 어떤 동작 또는 동작들이 취해질지를 결정한다. 취해질 수 있는 동작들의 예들은 준비(자원의 부가/제거), 허가 제어 및 자원 튜닝을 포함한다.

<28> 자원 튜닝은 워크로드가 변경시에 효과적일 수 있다. 예컨대, 전자 상거래 사이트에서, 상대적인 구매 대 브라우즈의 비율이 변경될 수 있고 이런 트랜스액션은 데이터 베이스 내의 어느 정도 다른 자원에 의존하거나 (예컨대 다른 버퍼 풀을 사용과 같음) 또는 더 많은 정렬(sorting)을 요청함으로, 상대적 버퍼 풀 크기 또는 정렬 힙(sort heap)과 같은 데이터베이스 구성 파라미터를 변경해서 전체 시스템 반응성이 개선될 수 있다. 이것은 튜닝 인터페이스(136)를 통해 튜닝 모듈(129)에 의해 결정되고 달성된다.

<29> 허가 제어는 실제로 소정의 정책(예컨대, 선호 고객만 또는 구매 트랜스액션만 허가함)에 기반해 들어오는 요청의 일부를 거절해서 시스템이 과부하가 되지 않도록 한다. 이런 종류의 동작은 준비 또는 자원 튜닝과 같은 다른 동작들이 원하는 결과를 달성하지 못하면 취해질 수 있다.

<30> 임의의 경우에, 동작의 범위(예컨대, 얼마나 많은 서버가 부가되고 구성 파라미터는 얼마나 변경되는지)는 성능 추정기 능력에 의해 정해진다. 이런 동작들은 별개로 또는 조합해서 취해질 수 있다.

<31> 다음은 본 발명에 따른 컴퓨팅 배치 관리 시스템의 기능적인 동작의 예시적인 기술이다. 이 예에서, 제어 동작의 한 유형, 즉 활성 클러스터로의 추가/이로부터의 제거로 서버에 의해 달성되는 애플리케이션 준비가 고려된다.

<32> 도 2를 참조해, 블록도는 본 발명의 실시예에 따른 컴퓨팅 배치 관리 방법(200)을 도시한다.

<33> 솔루션 매니저(120)는 제어 로직 엔진(CLE)(122)가 배치 매니저(130) 내의 모니터링 모듈(132)로부터 획득한 워크로드(트랜잭션 속도), 서비스 목표(응답 시간) 및 구성(활성, 아이들(idle), 전이 상태인 자원)을 보관한다(단계(202)).

<34> 트랜잭션 속도 이력과 같은 모니터링된 데이터에 기반해, 솔루션 매니저(120)는 예상기 모듈(124)을 사용해 예상되는 워크로드를 장래에 투영한다(스텝(204)). 예상은 장기간(예컨대, 시/일) 또는 단기간(예컨대, 초/분)일 수 있다. 워크로드(예컨대, 보통 일마다/주마다 이외 등) 변화 내의 예상하지 못한 변화를 다루기 위해서, 예상기(forecaster)는 적응해서 최근 이력으로부터 빨리 학습하는 것이 선호된다. 예상 범위는 길 필요가 없는데, 예컨대, 범위는 대략 자원 동작(이 경우에서는 애플리케이션 준비)이 효과를 가지기 위해 필요한 시간 정도일 수 있다.

<35> 선호되는 실시예에서, 예상기 모듈(124)은 대리인 포대번호 YOR920030050US1의 제목 "Methods and Apparatus for Performing Adaptive and Robust Prediction"의 본 명세서와 동시에 출원되고 참조로서 본 명세서에 합체되는 미국 특허출원에 기술된 기술을 사용해서 구현될 수 있다. 예상기 모듈(124)은 예컨대, 본 명세서에 개시가 참조로서 합체가 되는 지. 이. 피. 복스 등의 "Time series Analysis: Forecasting and Control" (개정판)이나 하이페이론(Hyperion)로부터 이용가능한 하이페리온 포리캐스팅 수트(Hyperion Forecasting Suite)와 같은 다른 예상 기술을 채용할 수 있다. 본 발명은 임의의 특정 예상 기술에 한정되지 않음이 이해되어야 한다.

<36> 이후, 솔루션 매니저(120)는 서비스 목표 임계치(예컨대, 응답 시간)를 현재 또는 예상된 워크로드 트래픽에 기반해서 유지하기 위해 성능 추정기(126)를 사용해서 현재 자원 배치가 불충분한지, 충분한지(즉, 적절한지) 또는 너무 충분한지(즉, 과잉인지)를 검사한다. 본 발명은 임의의 특정 성능 추정 기술에 한정되는 것은 아니지

만, 채택될 수 있는 그런 기술의 예들은 브라이언 엘 옹의 "Configuration and Capacity Planning for Solaris Servers", 다니엘 에이. 메나스 등의 "Capacity Planning for Web Services: Metrics, Models, and Methods" 및/또는 BMC 소프트웨어로부터의 PATROL Perform와 PATROL Predict 및 IBM 사의 웹스피어에 관한 고 볼륨 웹 사이트 성능 시뮬레이터(High Volume Web Site Performance Simulator)의 기술된 것을 포함한다. 성능 추정기의 추천과 관련 자원 풀 내의 서버 수 및 상태의 지식에 기반해, 솔루션 매니저(120)는 요청을 배치 매니저(130) 내의 애플리케이션 준비기(134)에 보내어서 적절한 수의 서버를 필요시에 증감한다(단계(208)). 즉, CLE(122)가 현재 자원이 충분하다고 판단되면, 이 시점에서 더 이상의 준비가 필요 없을 수 있다.

<37> 준비가 필요하면, 준비기(134)는 아이들 상태로부터 신속히 서버를 애플리케이션을 운용하는 활성 클러스터에 추가한다(스텝(210)). 신속한 추가(예컨대, 수 분에서 1분 이하까지)는 저장된 애플리케이션 이미지를 추가된 서버 인스턴스(instancee) 상에 로딩하는 것, 서버를 시작하는 것 및 클러스터 매니저가 요청을 그것에 송신하도록 하는 것들의 조합으로 달성될 수 있다. 서버가 애플리케이션 이미지를 미리 로딩하였으면, 자원 추가 속도는 증가가 될 수 있음을 주목하라. 그래서, 미리 로딩된 서버는 오직 시작되고 클러스터 내에서 활성화되는 것을 필요한다. 활성 클러스터 내로부터 서버를 제거하기 위해, 준비기(134)는 존재하는 일은 계속되게 하고 클러스터 매니저로부터 서버로 새롭게 들어오는 요청을 디스에이블한다. 이것은 그 서버상에서 일을 자연스럽게 휴지시키고, 그 후 멈춘다.

<38> 컴퓨팅 자원에 관해서 "불충분" 및 "너무 충분"(및 유사하게, "부적절" 및 "과잉") 구절의 본 명세서에서의 사용에 의해 충분하지 않은(부족한) 자원이 배치되었다거나 너무 많은(너무 충분한) 자원이 배치되었다는 것을 일반적으로 의미하는 것뿐만 아니라, 하나 이상의 자원이 기대되는 워크로드를 충족하기에(예컨대, 워크로드 혼합을 변경하는 것에 의해 등) 부적절하게 구성되었거나 허가되어서는 안 될 소정의 허가 요청이 허가되었다는 것을 의미한다. 그래서, 본 발명에 따라서, 자원 배치는 자원을 추가 또는 제거해서뿐만 아니라 하나 이상의 자원을 튜닝 및/또는 소정의 허가 요청을 스로틀링을 해서 "충분"(또는 유사하게 "적절")하게 만들어질 수 있다.

<39> 다음 기술은 본 명세서에 기술된 관리 시스템의 예시적인 구현으로부터의 소정의 결과를 제공한다. 고려된 예시적인 배치에서, 애플리케이션은 공급망 관리이다. 이 애플리케이션은 백-앤드(back-end) 내의 데이터베이스 서버로 웹 애플리케이션 서버의 클러스터 상에서 운용된다. 이 예에서, 시스템은 2초의 응답 시간 서비스 목표 아래로 유지하려고 노력하고 있음에도, 트랜잭션 요청(보통 로드의 20배까지 매 분마다 배가됨)의 기대하지 못한 서지에 직면하고 있다. 단기간 예상기의 예측 범위는 1분인데, 왜냐하면, 준비기의 신속한 애플리케이션 배치 능력이 약 30에서 40초이기 때문이다.

<40> 도 3에서는 시스템의 성능이 도시된다. 트랜잭션 속도의 예측(도 3의 상부 패널의 선(302))을 처음에는 실제 트랜잭션 속도(도 3의 상부 패널의 선(304))에 뒤지나, 매우 빨리 따라잡고 기대되는 트랜잭션 속도에 대해 유용한 안내를 제공한다. 다음에 이것은 도 1의 성능 추정기(126)를 사용해서 필수적인 애플리케이션 서버의 수를 계산하는데 사용될 수 있고, 서버는 도 1의 애플리케이션 준비기(도 1의 134)의 신속한 준비 능력으로 신속히 활성(도 3의 중간 패널 라인 306) 듀티(duty)로 간다. 아이들에서 활성으로 가면서 서버가 전이 시작 상태(도 3의 중간 패널의 선(308))에 있는 시간은 단지 약 30초이다. 솔루션 매니저 내의 성능 추정 및 예상의 조합은 배치 매니저의 신속한 애플리케이션 제공 능력과 함께 시스템이 극도로 공격적인 서지에 직면에서 2초 응답 시간 타깃(도 3의 아래 패널의 선(310)) 이하로 있도록 한다. 서지가 끝난 후에, 서버는 풀로 반환된다.

<41> 도 4를 참조해서, 블록도는 본 발명에 따른 컴퓨팅 배치 관리 시스템을 구현하기에 적절한 컴퓨터 시스템의 일반화된 하드웨어 아키텍처를 도시한다. 예컨대, 솔루션 매니저(120) 및 배치 매니저(130)에 관한 도 1에 도시된 기능적 컴포넌트는 도 4에 도시된 유형의 하나 이상의 컴퓨터 시스템상에서 구현될 수 있다. 물론, 별개의 기능적 컴포넌트들이 자신들의 전용 컴퓨터 시스템상에서 구현될 수 있다. 그러나 본 발명의 컴퓨팅 배치 관리 시스템이 임의의 특정 컴퓨터 플랫폼, 배열 또는 구현에 한정하려는 것은 아니다.

<42> 예시적인 구현(400)에서, 본 발명의 관리 방법 및 기능을 구현하기 위한 프로세서(402)는 버스(408)이나 다른 연결 배열에 의해 메모리(404) 및 입/출력(I/O)장치(406)에 작동하도록 연결된다. 본 명세서에 사용된 "프로세서" 용어는 예컨대, 중앙처리장치(CPU) 및/또는 다른 처리 회로(예컨대, DSP(digital signal processing), 마이크로프로세서 등)와 같은 임의의 처리 장치를 의도한다는 것을 인식해야한다. 추가로 "프로세서"라는 용어는 하나 이상의 처리 장치를 참조할 수 있고 처리 장치와 연관된 다양한 요소가 다른 처리 장치들에 의해 공유될 수 있다.

<43> 본 명세서에 사용된 "메모리"라는 용어는 예컨대, RAM(random access memory), ROM(read only memory), 고정 저장 매체(예컨대, 하드 드라이브), 이동 저장 매체(예컨대, 디스켓), 플래시 메모리 등과 같이 메모리 또는 프

로세서 또는 CPU와 연관된 다른 컴퓨터 판독 가능 매체의 포함을 의도한다. 바람직하게 메모리는 본 발명과 연관된 데이터 및 컴퓨터 프로그램을 저장하는데 사용될 수 있다.

<44> 추가해, 본 명세서에 사용된 용어 "I/O 장치"는 데이터를 처리 유닛에 입력하기 위한 하나 이상의 입력 장치(예컨대, 키보드, 마우스)뿐만 아니라 처리 유닛과 연관된 결과를 제공하기 위한 하나 이상의 출력 장치(예컨대, CRT 디스플레이)의 포함하는 것을 의도한다.

<45> 본 발명의 발명은 컴퓨터 판독 매체의 형태로 구현될 수 있음을 인식해야 한다. 본 명세서에 사용된 용어 "컴퓨터 판독가능 매체"는 전송 유형 매체뿐만 아니라 예컨대, 플로피 디스크, 하드 디스크 드라이브, RAM, CD(compact disk) ROM 등과 같은 기록 유형 매체의 포함을 의도한다.

<46> 따라서, 본 명세서에 기술된 바와 같이, 본 발명의 방법을 수행하기 위한 명령어 또는 코드를 포함하는 하나 이상의 컴퓨터 프로그램 또는 이들의 소프트웨어 컴포넌트는 하나 이상의 연관된 저장 매체(예컨대, ROM, 고정 또는 이동 저장소)에 저장될 수 있고, 이용될 준비가 되면, 전체 또는 부분으로 (예컨대, RAM으로) 로딩이 되어 프로세서(402)에 의해 실행될 수 있다.

<47> 임의의 경우에, 본 명세서에 기술되고 첨부된 도면에 도시된, 본 발명의 기술은 예컨대 메모리, 구현 특정 집적 회로, 기능 회로 등과 연관된 하나 이상의 프로그램된 범용 목적의 디지털 컴퓨터 같은 하드웨어, 소프트웨어, 또는 이들의 다양한 결합으로부터 구현될 수 있다. 본 명세서에서 제공된 본 발명의 소정의 기술로 당업자는 본 발명의 기술의 다른 구현을 구상할 수 있을 것이다.

<48> 따라서, 본 명세서에 자세히 설명된 바와 같이, 본 발명은 가변 워크로드 또는 기대하지 못한 워크로드 가변성에 직면했을 때, 시스템의 자동화된 사전 관리를 가능하게 하는 아키텍처를 제공한다. 즉, 필요한 기대하지 못한 가변성을 경험하는 컴퓨터 애플리케이션의 사전 관리에 관한 시스템이 제공되는데, 이는 예상, 성능 모델링, 제어 및 재구성을 위한 컴포넌트를 포함할 수 있다. 예상은 장래 워크로드를 예상하고 동작이 충분히 신속하게 취해져서 예상된 워크로드를 수용하도록 하면서, 서비스 레벨 목표의 잠재적인 위반에의 영향받는 것을 감소시킨다. 과도하게 계속적인 자원 구성의 변화(즉, 진동(oscillation))를 회피하면서, 미사용 자원을 최소화하는 비용대비 효과적인 방식으로 동작은 취해진다. 취해진 동작은 기대되는 워크로드 및 서비스 목표의 고려와 같이 이용 가능한 자원에 기반한 성능의 모델/추정에 기반한다. 동작은 자원 준비 및/또는 자원 튜닝 및/또는 허가 제어를 포함할 수 있다.

<49> 발명의 아키텍처는 (a) 관리되는 애플리케이션, (b) 제어를 감지하고 효과를 주기 위한 시스템에 일반 인터페이스를 제공하는 배치 매니저 및 (c) 모니터링에 일반 인터페이스에 기반해 언제 동작이 취해질지를 결정하고 일반 제어 인터페이스를 통해서 필요한 동작들을 요청하는 솔루션 매니저로 구조화될 수 있다. 배치 매니저는 다중 구성을 관리할 수 있는데, (a) 특정 자원 노드, 특정 구성 또는 특정 기능에 고유한 동작들을 취하고 (b) 애플리케이션에 적절한 데이터베이스 경로 및 파라미터를 자동으로 설정한 것을 할 수 있는 결과로 자원의 신속한 분배 또는 자원의 재배치가 된다.

<50> 더 명백한 것은, 발명은 나아가 (a) 워크로드 및 서비스 레벨 메트릭을 수집하고, (b) 워크로드 측정의 추후 값들을 예상하고, (c) 예상된 값들 및 원하는 서비스 레벨 목표가 적절한 서비스 레벨을 확보하는데 필요한 동작들을 결정하는데 사용할 수 있고 (d) 진동을 최소화하고 즉시 반응하는 방식으로 동작을 스스로 개시하는 적응 시스템을 동작시키는 기술을 제공한다.

<51> 본 발명의 예시적인 실시예가 본 명세서에서 첨부된 도면을 참조해 기술되었음에도, 본 발명은 이 정확한 실시예에 한정되는 것이 아니고 다양한 다른 변화 및 변형이 당업자에 의해 본 발명의 범위 또는 사상을 벗어나지 않고 만들어질 수 있음이 이해되어야 한다.

도면의 간단한 설명

<13> 도 1은 본 발명의 실시예에 따른 컴퓨팅 배치 관리 시스템 및 그런 시스템이 동작할 수 있는 전체 환경을 도시하는 블록도.

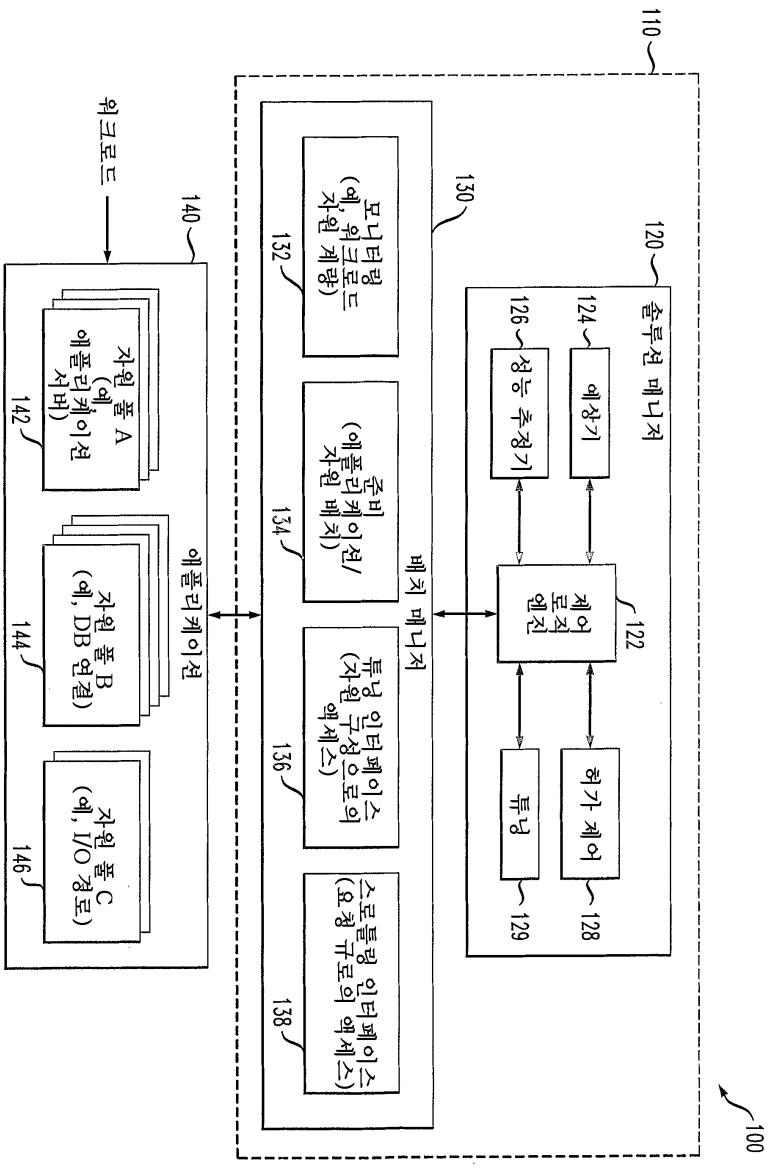
<14> 도 2는 본 발명의 실시예에 따라 컴퓨팅 배치 관리 방법을 도시하는 블록도.

<15> 도 3은 본 발명의 원리에 따라 컴퓨팅 시스템 또는 네트워크의 성능을 도시하는 도면.

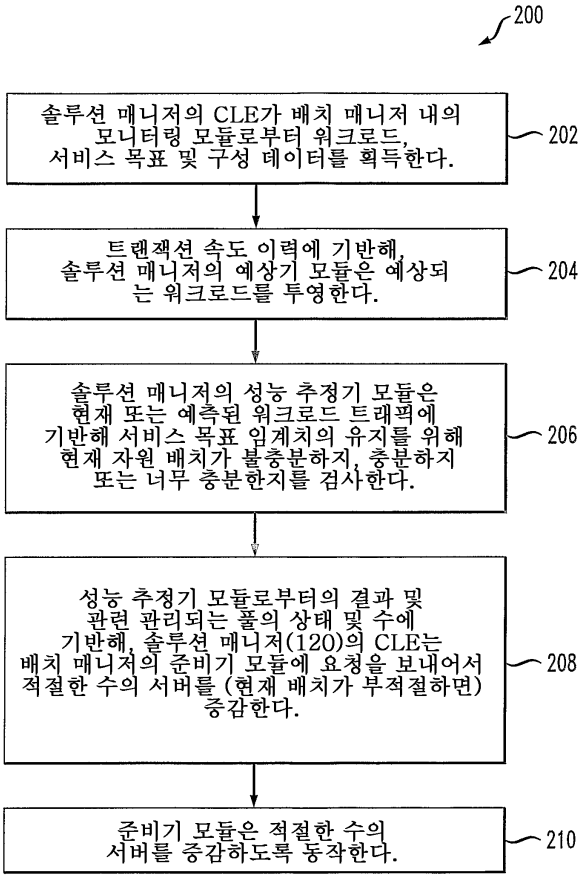
<16> 도 4는 본 발명에 따른 컴퓨팅 배치 관리 시스템을 구현하기에 적절한 컴퓨터 시스템의 일반화된 하드웨어 아키텍처를 도시하는 블록도.

도면

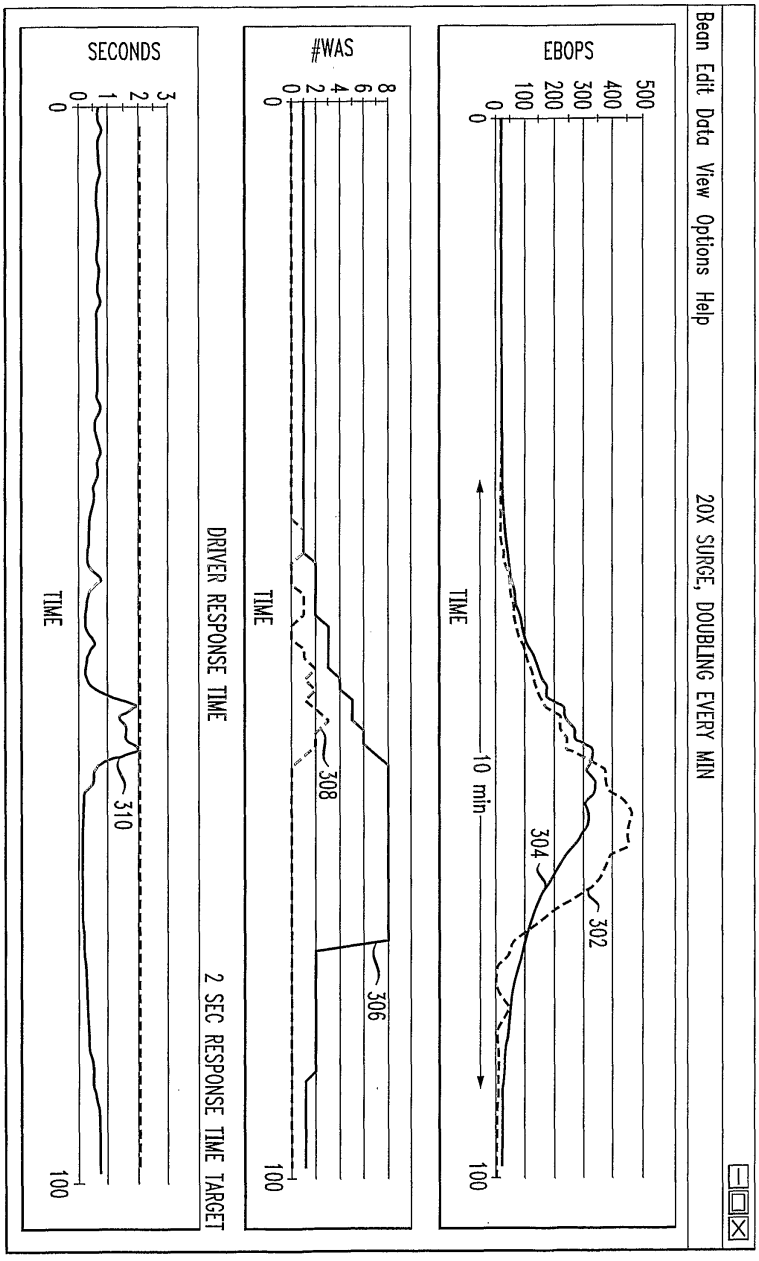
도면1


110
120
솔루션 매니저
124
예상기
126
성능 추정기
122
제어 로직 엔진
128
허가 제어
129
튜닝
130
배치 매니저
132
모니터링 (예, 워크로드 자원 계량)
134
준비 (애플리케이션/ 자원 배치)
136
튜닝 인터페이스 (자원 구성으로의 액세스)
138
스로틀링 인터페이스 (요청 규로의 액세스)
100
워크로드
140
애플리케이션
142
자원 풀 A (예 애플리케이션 서버)
144
자원 풀 B (예, DB 연결)
146
자원 풀 C (예, I/O 경로)

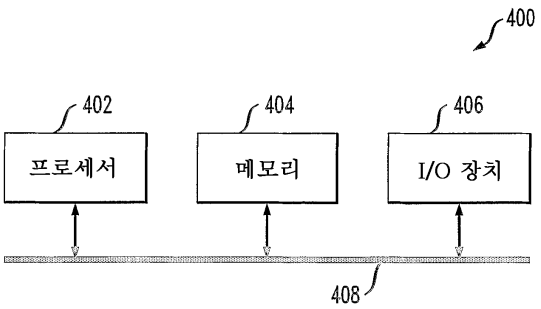
도면2


200
솔루션 매니저의 CLE가 배치 매니저 내의 모니터링 모듈로부터 워크로드, 서비스 목표 및 구성 데이터를 획득한다.
202
트랜잭션 속도 이력에 기반해, 솔루션 매니저의 예상기 모듈은 예상되는 워크로드를 투영한다.
204
솔루션 매니저의 성능 추정기 모듈은 현재 또는 예측된 워크로드 트래픽에 기반해 서비스 목표 임계치의 유지를 위해 현재 자원 배치가 불충분하지, 충분하지 또는 너무 충분한지를 검사한다.
206
성능 추정기 모듈로부터의 결과 및 관련 관리되는 풀의 상태 및 수에 기반해, 솔루션 매니저(120)의 CLE는 배치 매니저의 준비기 모듈에 요청을 보내어서 적절한 수의 서버를 (현재 배치가 부적절하면) 증감한다.
208
준비기 모듈은 적절한 수의 서버를 증감하도록 동작한다.
210

도면3


Bean Edit Data View Options Help
20X SURGE, DOUBLING EVERY MIN
EBOPS
500
400
300
200
100
0
TIME
10 min
100
304
302
#WAS
8
6
4
2
0
TIME
308
306
100
SECONDS
3
2
1
0
TIME
310
100
DRIVER RESPONSE TIME
2 SEC RESPONSE TIME TARGET

도면4


400
402
404
406
프로세서
메모리
I/O 장치
408