



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월12일
(11) 등록번호 10-1957533
(24) 등록일자 2019년03월06일

(51) 국제특허분류(Int. Cl.)
G06F 9/44 (2018.01) G06F 9/46 (2006.01)
(21) 출원번호 10-2013-7015588
(22) 출원일자(국제) 2011년12월14일
심사청구일자 2016년11월15일
(85) 번역문제출일자 2013년06월17일
(65) 공개번호 10-2013-0126938
(43) 공개일자 2013년11월21일
(86) 국제출원번호 PCT/US2011/064754
(87) 국제공개번호 WO 2012/082811
국제공개일자 2012년06월21일
(30) 우선권주장
12/972,424 2010년12월17일 미국(US)
(56) 선행기술조사문헌
WO2010038307 A1*
Byung-Gon Chun외 1명. Augmented Smartphone
Applications Through Clone Cloud Execution.
2009.5.18.
US7664626 B1
US20080163210 A1
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
(72) 발명자
스프라드린 제레미아 씨
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이턴즈 마
이크로소프트 코포레이션
포티어 도미니크
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이턴즈 마
이크로소프트 코포레이션
(74) 대리인
제일특허법인(유)

전체 청구항 수 : 총 18 항

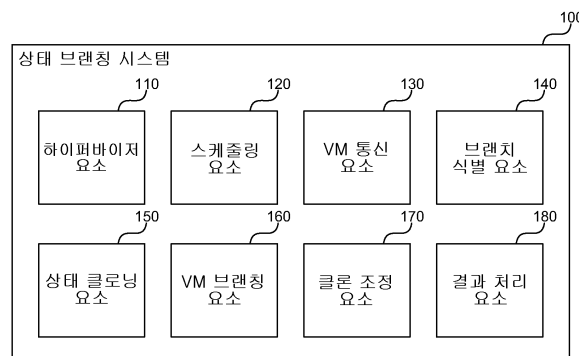
심사관 : 최정권

(54) 발명의 명칭 가상 머신 브랜칭 및 병렬 실행 기법

(57) 요약

시작 상태를 설정하는 데 소비되는 시간을 레버리징하면서 복합 상태 전환의 병렬 실행을 허용하는 상태 브랜칭 시스템이 본 명세서에 설명된다. 가상 머신 레벨에서의 브랜칭을 허용함으로써, 상태 브랜칭 시스템은 가상 머신에서의 특정 조건 또는 상태의 설정과, 상이하고 가능한 후속 상태를 탐색하기 위해 가상 머신의 병렬 인스턴스를 복사 및 브랜칭하는 것을 허용한다. 알려지지 않은 출력을 갖는 대규모 상태 변화를 검출할 시에, 상태 브랜칭 시스템은 하이퍼바이저에게 실행되는 가상 머신을 하나 이상의 별개의 가상 머신으로 복사할 것을 지시한다. 그러면 시스템은 동일한 시작점으로부터 상이한 상태를 탐색하기 위해 2개 이상의 가상 머신 사이에서 분기하는 브랜칭을 허용한다. 실행이 다음 상태에 도달되면, 시스템은 어떠한 사본이 실행을 계속할 것인지를 결정하도록 조정한다. 따라서, 상태 브랜칭 시스템은 복합 상태 변화의 보다 신속한 탐색을 허용한다.

대표도



명세서

청구범위

청구항 1

원래의 가상 머신(original virtual machine)으로부터 하나 이상의 클론(clone) 가상 머신으로 브랜칭하여 분기 상태를 고찰하기 위한, 컴퓨터로 구현되는 방법으로서,

상기 원래의 가상 머신 상에서 실행하기 위한 어플리케이션 코드를 수신하는 단계 -상기 어플리케이션 코드는 상기 가상 머신의 2개 이상의 가능한 상태를 가져오는 적어도 하나의 브랜치를 포함함- 와,

상기 수신된 어플리케이션 코드를 포함하는 원래의 가상 머신을 실행하는 단계와,

상기 어플리케이션 코드로부터 상기 어플리케이션이 복수의 가능한 상태로 분기할 위치를 나타내는 정보를 획득함으로써, 상기 가상 머신의 2개 이상의 가능한 상태를 가져오는 상기 원래의 가상 머신 상에서 실행되는 어플리케이션 코드 내의 브랜치를 식별하는 단계와,

하나 이상의 클론 가상 머신을 생성하는 단계 - 상기 하나 이상의 클론 가상 머신은 상기 원래의 가상 머신의 현재 상태를 복사하고, 그 후 각각이 상기 브랜치의 하나의 경로를 처리하여 상기 생성된 클론 가상 머신에서의 후속 상태를 탐색함 - 와,

상기 식별된 브랜치의 상이한 경로를 실행하도록 각각의 클론 가상 머신을 설정하는 단계와,

상기 식별된 브랜치의 적어도 2개의 경로를 병렬로 탐색하도록 상기 원래의 가상 머신과 클로닝된 가상 머신을 실행하는 단계와,

상기 클로닝된 가상 머신으로부터 적어도 하나의 실행 결과를 수신하는 단계

를 포함하고,

상기 단계들은 적어도 하나의 프로세서에 의해 수행되는

방법.

청구항 2

제 1 항에 있어서,

상기 어플리케이션 코드를 수신하는 단계는, 머신의 상태를 변화시키고 런타임으로 평가된 조건에 기초한 브랜치를 포함하는 어플리케이션 코드를 수신하는 단계를 포함하는

방법.

청구항 3

제 1 항에 있어서,

상기 원래의 가상 머신을 실행하는 단계는, 하이퍼바이저(hypervisor)가 동일한 물리적 리소스를 공유하는 복수의 가상 머신의 실행을 지시하는 단계를 포함하는

방법.

청구항 4

제 1 항에 있어서,

상기 원래의 가상 머신을 실행하는 단계는, 미리 정해진 임계값을 넘어서 가상 머신 상태를 잠재적으로 변화시키는 브랜치가 식별될 때까지 상기 원래의 가상 머신을 실행하는 단계와, 상기 클로닝된 가상 머신이 원래의 가상 머신 인스턴스의 외부의 임의의 리소스에 영향을 미칠 것인지 여부를 판정하는 단계를 포함하는

방법.

청구항 5

제 1 항에 있어서,

상기 브랜치를 식별하는 단계는 코드 분석 툴을 사용하여 상기 브랜치를 자동으로 식별하는 단계를 포함하는 방법.

청구항 6

제 1 항에 있어서,

상기 브랜치를 식별하는 단계는 2개 이상의 브랜치 경로를 식별하는 어플리케이션 프로그래밍 인터페이스(API: application programming interface)의 호출을 수신하는 단계 -상기 API에 의해 시스템이 가상 머신을 클로닝할 수 있으며 경로를 병렬로 실행할 수 있음- 를 포함하는

방법.

청구항 7

제 1 항에 있어서,

상기 클론 가상 머신을 생성하는 단계는 가상 하드웨어를 식별하는 하나 이상의 가상 머신 사양을 설정하는 단계, 상기 원래의 가상 머신의 메모리 상태를 복사하는 단계 및 상기 원래의 가상 머신에 부착된 하나 이상의 디스크를 복사하는 단계를 포함하는

방법.

청구항 8

제 1 항에 있어서,

상기 클론 가상 머신을 생성하는 단계는 상기 원래의 가상 머신의 실행을 중지하는 단계와 가상 머신과 연관된 하나 이상의 물리적 머신 파일을 복사하는 단계를 포함하는

방법.

청구항 9

제 1 항에 있어서,

상기 각각의 클론을 설정하는 단계는 각각의 클론이 어떠한 경로에 대한 탐색을 담당하는지에 대한 정보를 이용하여 각각의 클론을 설정하는 단계와, 그 후 각각의 클론과 상기 원래의 가상 머신을 실행하여 각각이 그 할당된 경로를 탐색할 수 있게 하는 단계를 포함하는

방법.

청구항 10

제 1 항에 있어서,

상기 각각의 클론을 설정하는 단계는 상기 클론 가상 머신이 실행할 소프트웨어 코드를 식별하는 단계와, 특정 위치에 대한 상기 클론 가상 머신의 명령 포인터 및 레지스터 상태를 설정하는 단계를 포함하는

방법.

청구항 11

제 1 항에 있어서,

상기 적어도 하나의 실행 결과를 수신하는 단계는 상기 클로닝된 가상 머신에 의해 도달된 상태의 표시를 수신하는 단계를 포함하는

방법.

청구항 12

가상 머신 브랜칭 및 병렬 실행을 위한 컴퓨터 시스템으로서,

후속하는 컴포넌트들 내에서 구현되는 소프트웨어 명령을 실행하도록 구성된 프로세서 및 메모리와,

2개 이상의 가상 머신 사이에서 물리적 컴퓨팅 디바이스의 리소스를 공유하고 가상 머신을 클로닝하기 위한 지원을 제공하는 하이퍼바이저 컴포넌트와,

원래의 가상 머신과 클로닝된 가상 머신을 포함한 2개 이상의 가상 머신 사이의 리소스 격리 및 물리적 머신 상에서의 리소스 사용의 스케줄링을 제공하는 스케줄링 컴포넌트와,

2개 이상의 가상 머신 사이에 하나 이상의 통신 채널을 제공하는 VM(Virtual Machine) 통신 컴포넌트와,

하나 이상의 클론 가상 머신에서의 후속 상태를 평가하기 위해 소프트웨어 코드를 실행하는 원래의 가상 머신을 브랜칭하기 위한 후보인 소프트웨어 코드의 현재 상태 및 하나 이상의 후속 상태를 식별하는 브랜치 식별 컴포넌트와,

유사한 상태를 갖는 하나 이상의 클론 가상 머신을 생성하기 위해 상기 원래의 가상 머신으로부터 상태 정보를 복사하는 상태 클로닝 컴포넌트와,

각각의 생성된 클론 가상 머신을 시작하고 추적할 각각의 클론에 대한 실행의 브랜치를 식별함으로써 상기 브랜치를 실행하는 VM 브랜칭 컴포넌트와,

불필요한 브랜치의 폐기 또는 브랜치 결과의 병합을 허용하기 위해 각각의 클론 가상 머신 및 상기 원래의 가상 머신의 액션을 조정하는 클론 조정 컴포넌트와,

각각의 가상 머신에 의해 생성된 결과를 처리하고 상기 결과를 상기 클론 조정 컴포넌트에 제공하는 결과 처리 컴포넌트를 포함하는

컴퓨터 시스템.

청구항 13

제 12 항에 있어서,

상기 VM 통신 컴포넌트는 상기 원래의 가상 머신으로부터 조정 정보를 수신하고 조정 명령을 하나 이상의 클론 가상 머신에 제공하는

컴퓨터 시스템.

청구항 14

제 12 항에 있어서,

상기 브랜치 식별 컴포넌트는 상기 소프트웨어 코드를 분석함으로써 상기 하이퍼바이저 컴포넌트를 통해 잠재적인 브랜치들을 자동으로 식별하는

컴퓨터 시스템.

청구항 15

제 12 항에 있어서,

상기 상태 클로닝 컴포넌트는 각각의 클론이 추적할 분기 상태를 탐색하는데 사용되는 상기 원래의 가상 머신의 상태 정보의 서브셋을 선택하고, 상기 선택된 서브셋을 복사하는

컴퓨터 시스템.

청구항 16

제 12 항에 있어서,

상기 VM 브랜칭 컴포넌트는 각각의 클론 가상 머신에게 자신이 고찰할 상기 브랜치를 통보하고, 그 후에 각각의 가상 머신이 분기 상태를 탐색하도록 실행하게 하는

컴퓨터 시스템.

청구항 17

제 12 항에 있어서,

상기 클론 조정 컴포넌트는 각각의 클론 가상 머신의 결과를 고찰하고, 상기 상태의 탐색이 완료된 후에 실행을 다시 원래의 가상 머신으로 전달하는

컴퓨터 시스템.

청구항 18

제 12 항에 있어서,

상기 결과 처리 컴포넌트는 각각의 클론 가상 머신의 상태 변화 정보를 상기 원래의 가상 머신에 복사하기 위해 상기 상태 변화 정보를 캡처하는

컴퓨터 시스템.

발명의 설명

기술 분야

배경 기술

[0001] 브랜칭은 실행이 복수의 잠재적 경로를 따라 하향 진행하는 컴퓨터 프로그램에서의 임의의 점을 지칭할 수 있다. 컴퓨터 과학에는 다양한 유형의 브랜칭이 존재한다. 머신 레벨에서, 소프트웨어 어셈블리 언어 명령은 종종 조건부 건너뛰기(conditional jump)를 포함한다. 조건이 성공적이면, 프로세서는 건너 뛰기에 의해 특정된 위치에서 코드를 실행할 것이다. 조건이 실패하면, 프로세서는 건너 뛰기에 후속하는 코드를 계속 실행할 것이다. 현재 과학계 파이프라인화된 멀티-코어 프로세서는 흔히 브랜치의 양쪽 경로를 추측 실행하기 시작한다. 실행이 브랜치 조건에 도달하고, 조건이 평가되면, 프로세서는 취해진 브랜치를 보유하고, 다른 브랜치를 추측 실행하는 중간 결과를 플러싱한다(flush). 또한, 브랜칭은 프로세스 레벨에서 발생한다. UNIX와 같은 운영 체제는 프로세스의 사본을 생성하고 새로운 프로세스에서의 실행을 계속하는 fork()와 같은 어플리케이션 프로그래밍 인터페이스(API: application programming interfaces)를 포함한다. 이는 프로세스의 멀티스레딩(multithreading) 또는 프로세스의 분기하는 잠재적 경로를 추적하는 데 사용될 수 있다.

[0002] 복합 시스템을 테스트하는 것은 시스템을 특정 조건으로 하고 다양한 자극에 대한 시스템의 반응을 테스트하기 위한 상당한 양의 설정을 종종 포함한다. 자극은 퍼즈(fuzz) 테스트 양식의 다양한 입력을 제공하는 것, 시스템의 반응을 테스트하기 위해 접속된 하드웨어를 조작하는 것 등을 포함할 수 있다. 시스템이 일부 기능은 실질적인 상태 변화를 야기할 수 있어, 이전 조건으로 복귀시키는 것이 곤란하다. 다른 예에서, 복합 시스템은 단지 데이터가 필요하지 않았는지 또는 데이터를 중요하지 않게 하는 브랜치가 취해질 것인지를 알아내기 위해 데이터의 브랜치를 처리하는 데 긴 시간을 소비할 수 있다.

[0003] 브랜칭은 현재 이러한 상황에서 도움이 되는 상당히 낮은 레벨에서 적용된다. 복합 시스템 및 다른 필드를 테스트하는 것은, 너무 많은 시간을 소비하지 않고 또한 어려운 설정을 필요로 하지 않고, 알려진 상태에서부터 시작하고 복수의 잠재적인 장애의 상태를 탐색하는 방식을 필요로 한다. 통상적으로 현재 테스트 장치는 복수의 가능한 테스트 경로를 직렬로 실행할 것이며, 또는 보다 효율적인 경우에는 복수의 머신을 레버리징하여 테스트를 병렬로 실행할 수도 있다. 이는, 특히 테스트 장치가 각 머신을 복수의 테스트를 시작하기 위한 일반적인 상태로 하는 경우에 여전히 시간 소모적일 수 있다. 복합 상태 평가에 있어서, 관련 시간은 소프트웨어 개발자가 원하는 것만큼 많은 변경 조건에서 시스템을, 그리고 결과적으로 소프트웨어 제품의 보다 많은 에러를 테스트하기 위한 보다 적은 시간을 의미할 수 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0004] 본 명세서에는 시작 상태를 설정하는 데 소비되는 시간을 레버리징하면서 복합 상태 전환(transition)의 병렬 실행을 허용하는 상태 브랜칭 시스템이 설명된다. 시스템은 하이퍼바이저(hypervisor)를 개조하여 가상 머신 레벨에서 브랜칭을 허용함으로써 동작한다. 가상 머신 레벨에서 브랜칭을 허용함으로써, 상태 브랜칭 시스템은 특정 조건 또는 가상 머신에서의 상태의 설정을 허용하고, 가상 머신의 병렬 인스턴스로 복사 및 브랜칭하여 상이하고 가능한 후속 상태를 탐색한다. 보다 크고 보다 복합적인 평가에 대한 문제점은, 특히 머신의 상태가 이러한 평가에 의해 상당히 영향을 받는 경우에 종종 발생한다. 알려지지 않은 결과(들)를 갖는 큰 상태 변화를 검출 또는 통지받을 시에, 상태 브랜칭 시스템은 하이퍼바이저에게 실행 가상 머신을 하나 이상의 별개의 가상 머신으로 복사할 것을 지시한다. 그러면 시스템은 2개 이상의 가상 머신(원본 및 사본) 사이에 분기 브랜칭을 허용하여 유사한 시작점으로부터 상이한 상태들을 탐색한다. 실행이 다음 상태에 도달되면, 시스템은 어떠한 사본 또는 사본들이 실행을 계속할 것인지를 판정하기 위해 조정한다. 따라서, 상태 브랜칭 시스템은 컴퓨터 시스템의 상태를 실질적으로 개조하는 복합 상태 변화의 보다 빠른 탐색을 허용한다.

[0005] 이러한 개요는 상세한 설명에서 추가적으로 후술되는 단순화된 양식의 개념들의 선택을 안내하기 위해 제공된다. 이러한 개요는 청구되는 청구물의 중요한 특징 또는 필수적인 특징을 식별하고자 하는 것도 아니며, 청구되는 청구물의 범위를 제한하는 데 사용되고자 하는 것도 아니다.

도면의 간단한 설명

[0006] 도 1은 일 실시예에서 상태 브랜칭 시스템의 요소들을 나타내는 블록도이다.

도 2는 일 실시예에서 원래의 가상 머신으로부터 하나 이상의 클론(clone) 가상 머신으로 브랜칭하여 분기 상태를 고찰하는 상태 브랜칭 시스템의 처리를 나타내는 흐름도이다.

도 3은 일 실시예에서 하나 이상의 클론 가상 머신을 종료시키고 실행을 다시 원래의 가상 머신으로 집중시키는 상태 브랜칭 시스템의 처리를 나타내는 흐름도이다.

도 4는 일 실시예에서 상태 브랜칭 시스템에 의한 브랜치 경로의 병렬 처리를 나타내는 타임라인 도면이다.

발명을 실시하기 위한 구체적인 내용

[0007] 본 명세서에는 시작 상태를 설정하기 위해 소비되는 시간을 레버리징하면서 복합 상태 전환의 병렬 실행을 허용하는 상태 브랜칭 시스템이 설명된다. 시스템은 하이퍼바이저를 개조하여 가상 머신 레벨에서 브랜칭을 허용함으로써 동작한다. 하이퍼바이저는 물리적인 머신의 리소스(예를 들어, 프로세서, 메모리, 디스크 및 네트워크)를 공유하는 하나 이상의 가상 머신 프로세스를 관리한다. 가상 머신 모두가 동일한 물리적 머신 상에서 실행되어도, 일반적으로 각각은 다른 가상 머신을 인지하지 못하고 이와 격리된다. 가상 머신 레벨에서의 브랜칭을 허용함으로써 상태 브랜칭 시스템은 가상 머신에서 특정 조건 또는 상태를 설정하고, 가상 머신의 병렬 인스턴스를 복사 및 브랜칭하여 상이하고 가능한 후속 상태를 탐색하는 것을 허용한다. 보다 크고 보다 복잡한 평가에 있어서, 특히 머신의 상태가 이러한 평가에 의해 상당히 영향을 받는 경우에 문제점이 종종 발생한다. 하나의 예는 가상 머신 내의 전체 운영 체제를 브랜칭할 수 있는 퍼징(fuzzing)이며, 소프트웨어 개발자가 퍼징되는 다음 반복을 설정하는 데 소비되는 시간을 현저하게 감소시킬 수 있게 하며 퍼징되는 가능한 실행 경로의 보다 빠른 탐색을 허용한다.

[0008] 알려지지 않은 결과(들)를 갖는 큰 상태 변화를 검출 또는 통지받을 시에, 상태 브랜칭 시스템은 하이퍼바이저에게 실행 가상 머신을 하나 이상의 별개의 가상 머신으로 복사할 것을 지시한다. 그러면, 시스템은 2개 이상의 가상 머신(원본과 사본) 사이의 분기 브랜칭을 허용하여 동일한 시작점으로부터 상이한 상태를 탐색하게 한다. 일부 실시예들에서, 어플리케이션이 하이퍼바이저로 호출할 수 있어 하이퍼바이저가 현존하는 가상 머신을 산출하게 하고 복사된 사본 또는 사본들을 새로운 가상 머신으로 산출하게 한다. 그리고 시스템은 사본들을 그 결정 트리에서 조정되게 하여 이들은 중첩되지 않는다. 실행이 다음 상태에 도달하면, 시스템은 어떠한 사본 또는 사본들이 실행을 계속할 것인지를 판정하기 위해 조정한다. 일부 경우에, 시스템은 복수의 가상 머신으로 간략하게 브랜칭하고, 몇몇 상태를 탐색하고, 그 후 다음의 주요한 상태 변화가 프로세스를 반복할 때까지 결과를 원래의 가상 머신으로 다시 결합할 수 있다.

[0009] 일부 실시예들에서, 상태 브랜칭 시스템은 하이퍼바이저와 운영 체제 개조 양쪽에 따라 가상 머신 레벨 브랜칭을 허용한다. 운영 체제는 브랜치가 바람직한 때를 검출하는 성능을 포함한다. 이는 제 1 기회 예외 통지, 어플리케이션 요구, 사용자 모드 제 2 기회 예외 등과 같은 다수의 잠재적인 트리거를 포함한다. 또한 운영은 어떠한 결정점이 브랜칭 인스턴스 당 개조되어야 하는지를 어플리케이션이 특정하기 위한 API 또는 다른 수단을 포함할 수 있다. 하이퍼바이저는 커맨드를 노출시키기 위해 개조되어 호스트 또는 게스트 운영체제가 현존하는 게스트 운영 체제 인스턴스의 사본을 요청할 수 있게 한다. 또한, 하이퍼바이저는 게스트 운영 체제의 실행을 프리즈(freeze)할 수 있어, 임의의 하드웨어 인터럽트가 게스트 운영 체제 상에서의 실행을 유발하는 것을 방지한다. 또한, 하이퍼바이저는 메모리 관리 유닛(MMU)에게 새로운 브랜치 인스턴스를 위한 전체 게스트 운영 메모리 할당을 복사할 것을 지시한다. 일부 경우에, 시스템은 전체 메모리를 복사하지 않고, 하이퍼바이저가 제공하는 브랜칭 지원의 레벨에 따라 일부 데이터 페이지만을 복사하고 아마도 실행가능한 페이지를 복사하지 않을 수 있다. 이러한 방식으로, 호스트 또는 게스트 운영은 분기 경로의 병렬 평가로부터 이득을 갖는, 결정점에 도달된 때를 결정하고, 하이퍼바이저 및/또는 운영 체제로부터 가상 머신 레벨에서의 브랜치를 요구한다. 따라서, 상태 브랜칭 시스템은 실질적으로 컴퓨터 시스템의 상태를 개조하는 복합 상태 변화의 보다 신속한 탐색을 허용한다.

[0010] 도 1은 일 실시예에서 상태 브랜칭 시스템의 요소를 나타내는 블록도이다. 시스템(100)은 하이퍼바이저 요소(110), 스케줄링 요소(120), VM 통신 요소(130), 브랜치 식별 요소(140), 상태 클로닝 요소(150), VM 브랜칭 요소(160), 클론 조정 요소(170) 및 결과 처리 요소(180)를 포함한다. 이러한 요소들 각각에 대해 더욱 상세하게 설명한다.

[0011] 하이퍼바이저 요소(110)는 2개 이상의 가상 컴퓨팅 디바이스 사이에서 물리적 컴퓨팅 디바이스의 리소스를 공유한다. 현재 다수의 데이터 센터는 가상 머신을 사용하여 복수의 생산 어플리케이션이 각각 가상 환경 내에서 단일 서버 상에서 실행될 수 있게 하여, 각 어플리케이션은 머신의 배타적인 사용권을 갖고 있는 것으로

고려한다. 대신, 통상적으로 어플리케이션은 가상 머신의 배타적인 사용권을 갖지만, 물리적인 리소스는 공유한다. 가상 머신은 중앙 처리 장치(CPU) 속도, 메모리 사이즈, 디스크 용량, 네트워크 대역폭 등과 같은 보장된 양의 하드웨어 리소스를 어플리케이션에 제공한다. 하이퍼바이저는 동일한 컴퓨터(예를 들어, Xen, Hyper-V 및 VMWare) 상에서 나란히 복수의 가상 머신을 실행할 수 있게 하는 제품 하드웨어에 대해 폭넓게 이용가능하다. 하이퍼바이저는 CPU, 메모리, 디스크 및 네트워크와 같은 물리적 리소스에 대한 액세스를 다중화(때로는 스케줄링)한다. 하이퍼바이저는 2개 이상의 가상 머신 사이의 모든 리소스의 고정된 파티션을 제공할 수 있는 CPU 및 I/O 리소스 양쪽에 대한 스케줄러를 제공한다. 이는 예를 들어 하드(hard) 실시간 스케줄링 알고리즘을 사용하여 다수의 방식으로 수행될 수 있다.

[0012] 스케줄링 요소(120)는 물리적 머신 상의 리소스 사용 및 원래의 가상 머신 및 클로닝된 가상 머신을 포함하여 2개 이상의 가상 머신 사이의 리소스 격리의 스케줄링을 제공한다. 스케줄링 요소(120)는 하이퍼바이저 내에서 동작할 수 있고 가상 머신 격리를 제공할 수 있어, 각각의 가상 머신은 리소스 가용성을 위해 각 가상 머신에 제공된 임의의 보장을 만족시키는 방식으로 물리적 머신 리소스를 사용할 수 있다. 예를 들어, 물리적 머신이 2GHz 프로세서를 갖고, 2개의 가상 머신 각각이 프로세서 시간의 동등한 특정 부분을 보장받았다면, 스케줄링 요소(120)는, 각 가상 머신이 물리적 머신의 프로세서 시간이 절반을 사용하게 되도록 보장할 수 있다. 스케줄링 요소(120)는 물리적 머신 메모리, 디스크 공간, 네트워크 대역폭 및 다른 리소스의 유사한 분할을 제공할 수 있다.

[0013] VM 통신 요소(130)는 2개 이상의 가상 머신 사이에서 하나 이상의 통신 채널을 제공한다. 공유된 물리적 메모리, 하이퍼바이저로 호출하는 하이퍼 호출, 물리적 디스크 상의 공통 위치에 파일을 저장하는 것 등을 포함하여 크로스-VM 통신을 위한 다양한 기술들이 존재한다. VM 통신 요소(130)는 원래의 가상 머신으로부터 조정 정보를 수신할 수 있고 하나 이상의 클론 가상 머신으로 조정 지시를 제공할 수 있다. 조정 정보는, 각 가상 머신이 다른 브랜치를 따라 복수의 가능한 상태를 평가하는 것을 보장할 수 있다. 하이퍼바이저는 공유된 통신 채널을 관리할 수 있고 가상 머신 상에 보안 또는 다른 제한을 강제할 수 있다. 원래의 가상 머신은 이들 사이에서 클론을 생성하고 활동을 조정하기 위하여 하이퍼바이저 내에서 지원을 레버리징할 수 있다.

[0014] 브랜치 식별 요소(140)는 현재 상태와, 하나 이상의 클론 가상 머신에서 후속 상태를 평가하기 위해 소프트웨어 코드를 실행하는 원래의 가상 머신을 브랜칭하기 위한 후보인 소프트웨어 코드의 하나 이상의 후속 상태를 식별한다. 요소(140)는 소프트웨어 코드를 분석함으로써 자동으로 잠재적인 브랜치를 식별할 수 있거나, 브랜칭된 어플리케이션에 의해 안내될 수 있다. 브랜치의 자동 식별은 현재 CPU와 유사하게 발생하며, 분석 엔진은 복수의 단계들을 다가오는 지시 스트림으로 바라볼 수 있고, 실질적으로 머신의 상태를 변경시킬 지시, 기능 또는 다른 동작을 식별할 수 있다. 수동 식별은 특정 위치에서 브랜칭을 요구하는 테스트 또는 다른 어플리케이션을 프로그램하는 소프트웨어 개발자에 의해 발생할 수 있다. 예를 들어, 테스트는 몇몇 테스트에 공통인 상태를 설정할 수 있으며, 각각의 테스트가 추적하는 분기 경로를 실행할 것을 브랜치에게 요구하도록 호스트 운영 체제에게 통지할 수 있다.

[0015] 상태 클로닝 요소(150)는 유사한 상태를 갖는 하나 이상의 클론 가상 머신을 생성하기 위해 원래의 가상 머신으로부터 상태 정보를 복사한다. 복사는 전체 메모리 및 원래의 가상 머신의 다른 가상 하드웨어를 복사할 수 있거나, 각각의 클론이 추적하는 분기 상태를 탐색하는 데 사용되는 서브셋만을 선택할 수 있다. 예를 들어, 테스트는 개방하기에 긴 시간이 걸리는 다양한 유형의 많은 운영 체제 핸들을 폐쇄할 수 있다. 원래의 가상 머신은 핸들 모두를 개방하는 작업을 수행할 수 있으며, 각각의 클론 가상 머신은 다양한 조건을 테스트하기 위해 특정 유형의 핸들을 폐쇄한다. 본 예에서, 각 클론 가상 머신은 별개의 상태를 추적하지만, 모든 상태 정보 또는 다른 어플리케이션보다는, 단지 핸들에 관련된 상태의 복사를 필요로 할 수 있거나 보다 일반적으로 가상 머신 상에서 실행되는 테스트 어플리케이션에 관련된 상태의 복사를 필요로 할 수 있다.

[0016] VM 브랜칭 요소(160)는 각각 생성된 클론 가상 머신을 시작하고, 추적하는 각 클론에 대한 실행의 브랜치를 식별함으로써 브랜치를 실행한다. 예를 들어, 브랜치 식별 요소(140)가 실질적으로 다른 상태로 각각 도출되는 소프트웨어 코드에서 다가오는 5개의 가능한 경로를 식별한다면, 테스트 어플리케이션은 상태 클로닝 요소(150)가 4개의 클론을 생성하도록 지시할 수 있으며, 4개의 클론 가상 머신이 나머지 4개 상태를 추적하면서, 원래의 가상 머신은 상태 중 하나를 추적할 것이다. VM 브랜칭 요소(160)는 지시 포인터 및 다른 머신 상태를 브랜치를 둘러싸는 위치 및 상태로 직접 설정하는 것 등에 의해, 각 클론 가상 머신에 고찰할 브랜치를 통지한다. VM 브랜칭 요소(160)는 각 가상 머신이 실행되어 분기 상태를 탐색할 수 있게 한다.

[0017] 클론 조정 요소(170)는 클론 가상 머신과 원래의 가상 머신의 각각의 액션을 조정하여 필요하지 않은 브랜치를

폐기하거나 브랜치 결과를 병합할 수 있게 한다. 상이한 브랜치를 실행시키는 목적은 실행이 종료되는 위치를 발견하는 것이다. 많은 경우에, 소프트웨어 코드는 몇몇 브랜치들 중 하나를 추적하고, 취해진 브랜치에 기초하여 상이한 결과를 갖는 공통 위치에서 종료될 수 있다. 따라서, 브랜칭 후에 단기간 동안 폭넓게 분기하는 코드를 실행하지만, 후에 공통 위치에서 수렴하여 결과를 고찰하는 것이 통상적이다. 클론 조정 요소(170)는 테스트 또는 다른 어플리케이션 코드가 복수의 클론 가상 머신을 오프로 송신하게 하여 다양한 상태를 고찰하기 위해 일부 작업을 하지만, 각 클론 가상 머신의 결과를 고찰함으로써 상태의 탐색이 완료되었을 때 원래의 가상 머신으로 다시 실행을 전송하는 것에 의해 서클을 완료한다. 이는, 현재 CPU가 이론적으로 몇몇 브랜치를 실행할 수 있고 취해진 채로 종료하지 않는 것을 플러싱할 수 있는 방식과 유사하지만, 가상 머신 레벨에서 보다 넓은 스케일로 상태 브랜칭 시스템(100)에 의해 적용된다.

[0018] 결과 처리 요소(180)는 각 가상 머신에 의해 생성된 결과를 처리하고 그 결과를 클론 조정 요소(170)에 제공한다. 각 가상 머신과 원본과 클론은 가상 머신에 발생하는 추가적인 상태 변화와 같은 단순한 수 또는 텍스트 결과 또는 복합 결과를 포함하여 다양한 결과를 생성할 수 있다. 일례로서, 안티바이러스 프로그램은, 코드 모듈이 컴퓨팅 디바이스에 해로운 무엇인가를 행하는지를 판정하기 위해 악성 코드에 대해 스캐닝되고 있는 컴퓨팅 디바이스 상에서 발견된 몇몇 식별된 소프트웨어 코드 모듈을 실행하기를 원할 수 있다. 안티바이러스 프로그램은 원래의 컴퓨팅 디바이스의 특성 전부를 갖는 클론 가상 머신 내에 각 소프트웨어 코드 모듈을 산출할 수 있으며, 그 결과는 소프트웨어 코드가 클론에 할 수 있는 임의의 변화이다. 안티바이러스 프로그램은 임의의 것이 유해한지 여부를 판정하기 위해 결과를 스캐닝할 수 있고(예를 들어, 중요한 파일 또는 스파밍 접촉을 삭제), 유해한 결과가 발견되었는지 여부를 나타내는 불리언(Boolean) 결과를 원래 어플리케이션에 제공할 수 있다. 이는 각각의 클론이 몇몇 방식으로 클론 가상 머신의 상태를 잠재적으로 메쉬업(mesh up)할 수 있게 하지만, 특정 결과가 얻어진 후에 안티바이러스 프로그램에 의해 폐기될 수 있게 한다. 다른 예로서, 시스템을 사용하는 맵핑형 디바이스가, 몇몇 복합 액션을 수행해야 하는지 그리고 어떻게 수행해야 하는지를 결정하기 위해 네트워크를 핑(ping)하도록 설계될 수 있다. 이러한 브랜칭은, 디바이스가 고대기(high latency) 요구가 이루어지면서 물리적인 디바이스의 휴지(idle) 처리 전력을 사용할 수 있게 할 것이다. 아마도, 이는 센서 또는 안테나 어레이를 재측정하는 계산을 포함하며, 맵핑형 디바이스는 네트워크 응답의 수신 전에 가능한 계산을 시작할 수 있다.

[0019] 상태 브랜칭 시스템이 구현되는 컴퓨팅 디바이스는 중앙 처리 장치, 메모리 입력 디바이스(예를 들어, 키보드 및 포인팅 디바이스), 출력 디바이스(예를 들어, 표시 디바이스) 및 저장 디바이스(예를 들어, 디스크 드라이브 또는 다른 비휘발성 저장 매체)를 포함할 수 있다. 메모리 및 저장 디바이스는 시스템을 구현하거나 인에이블링하는 컴퓨터 실행가능 명령(예를 들어, 소프트웨어)으로 인코딩될 수 있는 컴퓨터 판독가능 저장 매체이다. 또한, 데이터 구조 및 메시지 구조는 저장되거나 통신 링크 상의 신호와 같은 데이터 전송 매체를 통해 전송될 수 있다. 인터넷, 근거리 네트워크, 원거리 네트워크, 포인트-투-포인트 다이얼-업(point-to-point dial-up) 접속, 셀 폰 네트워크 등과 같은 다양한 통신 링크가 사용될 수 있다.

[0020] 시스템의 실시예들은, 개인용 컴퓨터, 서버 컴퓨터, 핸드헬드 또는 랩톱 디바이스, 멀티프로세서 시스템, 마이크로프로세서 기반 시스템, 프로그램가능 소비자 전자 장치, 디지털 카메라, 네트워크 PC들, 미니 컴퓨터, 메인 프레임 컴퓨터, 상술한 시스템 또는 디바이스 중 임의의 것을 포함하는 분산형 컴퓨팅 환경, 셋톱 박스, 칩 상의 시스템(SOC) 등을 포함하는 다양한 운영 환경에서 구현될 수 있다. 컴퓨터 시스템은 셀 폰, 개인용 디지털 기기, 스마트 폰, 개인용 컴퓨터, 프로그램 가능 소비자 전자 장치, 디지털 카메라 등일 수 있다.

[0021] 시스템은 하나 이상의 컴퓨터들 또는 다른 디바이스에 의해 실행되는, 프로그램 모듈과 같은 컴퓨터 실행가능 명령의 일반적인 관계에서 설명될 수 있다. 일반적으로, 프로그램 모듈들은 특정 직무를 수행하거나 특정의 추상적 데이터 유형을 구현하는 루틴, 프로그램, 객체, 요소, 데이터 구조 등을 포함한다. 통상적으로, 프로그램 모듈들의 기능은 다양한 실시예에서 원하는 바에 따라 결합되거나 분산될 수 있다.

[0022] 도 2는 일 실시예에서 원래의 가상 머신으로부터 하나 이상의 클론 가상 머신으로 브랜칭하여 분기 상태를 고찰하는 상태 브랜칭 시스템의 처리를 나타내는 흐름도이다. 블록 210에서 시작하며, 시스템은 원래의 가상 머신 상의 실행을 위한 어플리케이션 코드를 수신하며, 어플리케이션 코드는 가상 머신의 2개 이상의 가능한 상태로 귀결되는 적어도 하나의 브랜치를 포함한다. 시스템은 머신의 상태를 변화시키는 테스트 또는 다른 어플리케이션을 수신할 수 있으며, 브랜칭은 런타임에서 평가된 조건에 기초하여 발생될 수 있다. 시스템은 원래의 가상 머신의 상태를 실질적으로 변경시키는 브랜치를 식별하며 브랜치로부터 나오는 경로를 평가하는 것과 병행하여 하나 이상의 클론 가상 머신을 실행한다.

[0023] 블록 220에서 계속하여, 시스템은 수신된 어플리케이션 코드를 포함하는 원래의 가상 머신을 실행한다. 예를 들

어, 시스템은 하이퍼바이저 내에서 동작할 수 있고 동일한 물리적 리소스를 공유하는 복수의 가상 머신의 실행을 지시할 수 있다. 시스템은, 가상 머신 상태를 잠재적으로 상당히 변화시키는 브랜치가 식별될 때까지 원래의 가상 머신을 실행한다. 예를 들어, 브랜치는 파일을 삭제하고, 긴 연산을 수행하고, 대량의 데이터를 생성하거나 다른 상태 변화 동작을 수행할 수 있다.

[0024] 블록 230에서 계속하여, 시스템은 가상 머신의 2개 이상의 가능한 상태로 귀결되는 원래의 가상 머신 상에서 실행되는 어플리케이션 코드 내의 브랜치를 식별한다. 예를 들어, 시스템은 본 기술 분야에 공지된 코드 분석 툴을 사용하여 브랜치를 자동으로 식별하여 소프트웨어 코드 내의 분기 경로를 발견할 수 있다. 다수의 정적인 그리고 런타임 분석 툴이 바이너리 및/또는 소스 코드 레벨에서 소프트웨어 코드를 검사하고, 어플리케이션 상태에서 상당한 분기의 위치를 식별한다. 일부 실시예에서, 시스템은 어플리케이션이 복수의 가능한 상태로 브랜칭할 위치를 어플리케이션이 나타낼 수 있게 한다. 예를 들어, 테스트 어플리케이션은 시스템이 가상 머신을 클로닝할 수 있고 경로를 병렬로 실행할 수 있도록 2개 이상의 브랜치 경로를 식별하는 운영 체제 API를 호출할 수 있다.

[0025] 블록 240에서 계속하여, 시스템은 원래의 가상 머신의 현재 상태를 복사하고 브랜치의 하나의 경로를 처리하여 생성된 클론 가상 머신에서 후속 상태를 탐색하는 하나 이상의 클론 가상 머신을 생성할 수 있다. 클로닝은 가상 하드웨어를 식별하는 가상 머신 사양을 설정하는 것, 원래의 가상 머신의 메모리 상태를 복사하는 것, 부착된 디스크를 원래의 가상 머신에 복사하는 것 등을 포함할 수 있다. 몇몇 가상화 소프트웨어는 임의의 매립형 파일 시스템(예를 들어, MICROSOFT™ Virtual PC에 의해 사용되는 VHD 파일 및 Hyper-V)을 포함하여 단일 파일 내에 가상 머신을 저장한다. 시스템은 원래의 가상 머신의 실행을 중지하고 가상 머신과 연관된 하나 이상의 파일을 복사함으로써 가상 머신을 클로닝할 수 있다.

[0026] 블록 250에서 계속하여, 시스템은 각 클론 가상 머신을 설정하여 식별된 브랜치의 상이한 경로를 실행한다. 클론들이 원본의 완벽한 클론들이라면, 이들은 각각 다음 원본과 동일한 것을 행할 것이다. 하지만, 상태 브랜칭 시스템의 하나의 목적은 각각의 클론이 브랜치의 상이한 경로를 병렬로 추적할 수 있게 하는 것이다. 따라서, 시스템은 어떠한 경로가 탐색을 담당하는지에 대한 정보를 갖는 각 클론을 설정하고, 그 후 각 클론과 원본을 실행시켜 그 할당된 경로를 탐색할 수 있게 한다. 각 클론을 설정하는 것은 가상 머신이 실행될 소프트웨어 코드를 식별하는 것과 특정 위치에 대한 명령 포인터 및 레지스터 상태를 설정하는 것 등을 포함할 수 있다.

[0027] 블록 260에서 계속하여, 시스템은 원래의 가상 머신과 클로닝된 가상 머신을 실행하여 식별된 브랜치의 적어도 2개의 경로를 병렬로 탐색한다. 브랜치의 특정 경로가 머신 상태를 상당히 변경시킨다면, 그 브랜치를 실행하는 가상 머신만이 영향을 받을 것이다. 한편, 원래의 가상 머신은 클로닝된 가상 머신의 실행의 영향을 결정하기 위해 대기하여 원래의 가상 머신 상태에 영향을 주지 않고 브랜치 경로에 대한 정보를 수집할 수 있다. 어플리케이션이, 조건이 브랜치 경로를 취하기에 충족되지 않는 것으로 결정하면, 그 경로에 관련된 가상 머신(들)은 단순히 폐기될 수 있으며, 원래의 가상 머신은 계속될 수 있다. 일부 경우에, 시스템은 메인 가상 머신으로서 계속되는 클론 가상 머신을 선택할 수 있고(즉, 원래의 가상 머신의 역할을 인수함), 원래의 가상 머신은 종료될 수 있다.

[0028] 블록 270에서 계속하여, 시스템은 적어도 하나의 실행 결과를 클로닝된 가상 머신으로부터 수신한다. 결과는 클로닝된 가상 머신에 의해 도달된 상태, 클로닝된 가상 머신으로부터 크로스-VM 통신 채널을 통해 전달된 정보, 클로닝된 가상 머신에 의해 실행된 브랜치 경로를 실행함으로써 생성된 출력 등을 나타낼 수 있다. 원래의 가상 머신은 다음에 무엇을 할지 결정하거나 그 결과를 원래의 가상 머신이 계속하여 처리할 것인 가상 머신을 선택하기 위해 실행 결과를 사용할 수 있다. 클로닝된 가상 머신이 병렬로 실행되므로, 시스템은 원래의 가상 머신 내에서 각 브랜치의 가능한 시도였던 것보다 더 빠른 결과에 도달할 것이다. 블록 270 후에, 이러한 단계들은 종료된다.

[0029] 도 3은 일 실시예에서 하나 이상의 클론 가상 머신을 종료(exit)시키고 실행을 다시 원래의 가상 머신으로 집중시키는 상태 브랜칭 시스템의 처리를 나타내는 흐름도이다. 블록 310에서 시작하며, 시스템은 원래의 가상 머신에서의 어플리케이션 코드의 브랜치를 사전 검출하고 클로닝된 가상 머신을 산출함으로써 제 1 클로닝된 가상 머신을 선택하여 각 브랜치 경로를 실행한다. 후속하는 반복에서, 시스템은 다음의 클로닝된 가상 머신을 선택한다.

[0030] 블록 320에서 계속하여, 시스템은, 선택된 클로닝된 가상 머신이 할당된 브랜치 경로의 실행을 완료했는지를 검출한다. 몇몇 경우에, 브랜치 경로는 코드의 보다 큰 본체의 작은 서브섹션을 나타낼 수 있으며, 서브섹션은, 코드의 보다 큰 본체의 외부의 그 결과를 결정하기 위해 브랜치의 실행을 격리시키는 데 유용한 방식으로 결과

또는 변화 상태를 생성한다. 클로닝된 가상 머신의 설정 동안, 시스템은, 클로닝된 가상 머신이 브랜치 경로의 실행을 완료했을 시점에서 명령(예를 들어, 중지 또는 인터럽트) 또는 다른 표시를 삽입할 수 있다. 하이퍼바이저 또는 다른 코드는 클로닝된 가상 머신의 완료를 검출하기 위해 명령을 감시하고 검출할 수 있다.

[0031] 블록 330에서 계속하여, 시스템은 클로닝된 가상 머신의 실행의 결과 상태를 식별한다. 결과는 수치 결과, 클로닝된 가상 머신에 의해 생성된 데이터, 클로닝된 가상 머신에 대한 상태 변환 등을 포함할 수 있다. 시스템은 클로닝된 가상 머신을 원래의 가상 머신과 비교하여 클로닝된 가상 머신에 의해 실행된 브랜치 경로를 탐색함으로써 생성된 변화를 식별할 수 있다. 원래의 가상 머신이 브랜치에 의해 이론적으로 실행된 경로를 취하기로 결정했다면, 원래의 가상 머신은 선택된 클로닝된 가상 머신의 상태 차이 또는 식별된 결과를 복사할 수 있다.

[0032] 결정 블록 340에서 계속하여, 시스템은, 현재 브랜치와 연관된 더 많은 클로닝된 가상 머신이 존재하는지 여부를 결정한다. 그렇다면, 시스템은 블록 310으로 되돌아가서 다음의 클로닝된 가상 머신을 선택하고, 그렇지 않으면 각각의 클로닝된 가상 머신이 처리된 후에 시스템은 블록 350에서 계속된다.

[0033] 블록 350에서 계속하여, 시스템은 원래의 가상 머신 상에서 실행되는 어플리케이션 코드 내의 하나 이상의 조건에 기초하여 취할 브랜치를 선택한다. 일부 실시예들에서, 클로닝된 가상 머신은, 원래의 가상 머신이 취할 수 있는 이론적으로 브랜치 경로를 실행시키기 위한 이론적인 실행 엔진으로서 동작한다. 원래의 가상 머신이, 모든 조건이 어떤 경로가 취해질 지에 대해 알기 위해 평가된 시점까지 실행한 후에, 원래의 가상 머신은 올바른 브랜치로서 브랜치들 중 하나를 선택할 수 있고 다른 것들은 폐기할 수 있다. 다른 실시예들에서, 클로닝된 가상 머신은 그 모두가 원래의 가상 머신에 의해 사용되는 누적 결과를 나타낼 수 있지만, 이들은 병렬로 실행되어 결과에 더 신속하게 도달한다. 이러한 경우에, 원래의 가상 머신은 각각의 클로닝된 가상 머신으로부터 결과를 수신하고, 수신된 결과에 대한 임의의 추가적인 처리를 수행하고, 클로닝된 가상 머신이 수행된 후에 계속된다.

[0034] 블록 360에서 계속하여, 시스템은 브랜치를 선택한 것을 실행한 클로닝된 가상 머신으로부터 식별된 결과 상태를 복사한다. 이러한 방식으로, 원래의 가상 머신은 개별적인 클로닝된 가상 머신에 의해 수행된 병렬 실행으로부터 이득을 얻는다. 어떠한 브랜치가 취해지든, 원래의 가상 머신은 각 브랜치의 테스트에 의해 방해받지 않고, 선행 코드가 완료된 후에 브랜치를 직렬로 실행하기 위해 대기할 필요가 없다. 오히려, 선행 코드가 원래의 가상 머신에서 완료되면서, 클로닝된 가상 머신이 각각 브랜치들 중 하나를 실행하고, 선택된 가상 머신이 그 결과를 준비되자마자 (그리고, 원래의 가상 머신이 그 결과 자체를 달성할 수 있기 전에) 원래의 가상 머신에 제공한다.

[0035] 블록 370에서 계속하여, 시스템은 클로닝된 가상 머신으로부터의 복사된 결과를 사용하여 원래의 가상 머신의 실행을 계속한다. 일부 실시예들에서, 시스템은 원래의 가상 머신을 폐기할 수 있고 선택된 클로닝된 가상 머신을 통해 실행을 계속할 수 있다. 시스템은 취해지지 않은 브랜치 경로를 나타내는 임의의 클로닝된 가상 머신을 폐기 또는 "플러싱(flush)"할 수 있다. 블록 370에서, 이러한 단계들은 종료된다.

[0036] 도 4는 일 실시예에서 상태 브랜칭 시스템에 의한 브랜치 경로의 병렬 처리를 나타내는 타임라인 도면이다. 타임라인의 제 1 행(410)은 본 명세서에 설명된 원래의 가상 머신의 처리를 나타낸다. 통상적인 시스템에서, 원래의 가상 머신 또는 물리적인 머신은 이러한 처리 모두를 수행할 것이다. 원래의 가상 머신은, 원래의 가상 머신이 실행하고 있는 어플리케이션 코드에서 브랜치(450)를 직면하기 전에 하나 이상의 작업(440)을 처리한다. 브랜치 전의 일부 점에서, 원래의 가상 머신은, 브랜치를 식별하고 브랜치의 잠재적인 경로를 병렬로 실행하는 2개의 클로닝된 가상 머신을 산출한다. 제 2 행(420)은 제 1 클로닝된 가상 머신의 처리를 나타내고, 제 3 행(430)은 제 2 클로닝된 가상 머신의 처리를 나타낸다. 원래의 가상 머신이 사전-브랜치 작업(440)을 완료하는 동안, 제 1 클로닝된 가상 머신은 제 1 브랜치 경로(460)와 연관된 코드를 실행하고, 제 2 클로닝된 가상 머신은 제 2 브랜치 경로(470)와 연관된 코드를 병렬로 실행한다. 원래의 가상 머신이 브랜치(450)에 도달할 때, 원래의 가상 머신과 브랜치(450)의 임의의 조건에 기초하여 취할 브랜치의 경로를 결정한다. 원래의 가상 머신은 승리한 브랜치를 선택하고 결과(480)를 원래의 가상 머신에 복사한다. 그 후, 원래의 가상 머신은 실행을 계속하고 클로닝된 가상 머신을 폐기한다. 이러한 방식으로, 클로닝된 가상 머신은 원래의 가상 머신의 장래 상태를 병렬로 준비하였고, 각 브랜치 경로를 평가하는 것에 포함될 수 있었던 침입 상태 변화로부터 원래의 가상 머신을 격리하였다.

[0037] 일부 실시예들에서, 상태 브랜칭 시스템은 디바이스 드라이버 테스트를 위한 프레임워크를 제공한다. 디바이스 드라이버 테스트는, 올바르게 없게 취급되는 경우에 운영 체제의 커널 공간에서 교착 상태를 만들 수 있는 물리적인 하드웨어 및 상태 변화와 연관되므로 종종 곤란하다. 상태 브랜칭 시스템은 물리적 하드웨어의 상태 변화

를 가상화를 허용하여, 병렬로 동작하는 복수의 클론 가상 머신이 다양한 잠재적인 하드웨어 입력 및 대응하는 드라이버 입력의 결과를 평가할 수 있다. 이는 디바이스 드라이버 소프트웨어 코드의 보다 빠르고 보다 강인한 개발에 편의를 제공한다.

[0038] 일부 실시예들에서, 상태 브랜칭 시스템은 브랜치 경로를 탐색하기 위한 추가적인 가상 머신을 산출하기 위한 임계값을 결정한다. 가상 머신의 생성 및 설정은 시간 및 리소스 사용의 관점에서 일정한 비용을 초래한다. 소규모 브랜치에 있어서, 클로닝된 가상 머신을 설정하는 것과 연관된 시간은 이들로부터 얻어진 이점의 가치가 없을 수 있다. 따라서, 시스템은 클로닝된 가상 머신을 사용하는 이점과 연관된 임계값을 결정할 수 있고, 트레이드오프가 궁극적으로 실행 시간을 절약할 것이거나 다른 긍정적인 결과를 제공하는 경우에만 새로운 클로닝된 가상 머신을 산출할 수 있다. 어플리케이션은 어플리케이션-특정 고려사항에 기초하여 임계값을 설정하거나 조절할 수 있다. 마찬가지로, 일부 실시예들에서, 상태 브랜칭 시스템은, 브랜치 경로 탐색을 위해 클로닝된 가상 머신을 사용하는 이점을 평가할 때, 클로닝된 가상 머신이 클로닝된 가상 머신의 외부의 임의의 리소스에 영향을 미칠 것인지 여부(예를 들어, 외부 또는 링크된 물리적 디바이스에 커맨드를 발송하는 것, 네트워크를 통해 패킷을 전송하는 것 등)를 고려할 수 있다.

[0039] 일부 실시예들에서, 상태 브랜칭 시스템은 원래의 가상 머신과는 다른 물리적 머신 상의 클로닝된 가상 머신을 산출한다. 가상 머신은 물리적인 머신들 사이의 용이한 이동에 매우 적합하며, 시스템은 별개의 물리적 머신 상의 클로닝된 가상 머신의 일부 또는 전부를 실행함으로써 다른 물리적 머신에 대한 다양한 브랜치 경로의 탐색을 분담할 수 있다. 클로닝된 가상 머신은 필요하다면 물리적 머신들 사이의 네트워크 또는 다른 통신 채널을 통해 원래의 가상 머신과 통신할 수 있다.

[0040] 일부 실시예들에서, 상태 브랜칭 시스템은 브랜치 경로를 탐색하기 위해 클로닝된 가상 머신을 사용하기 위한 기회를 검출할 시에 하이퍼바이저로부터의 통지를 제공한다. 하이퍼바이저는 가상 머신에서 각각의 실행 프로세스를 감시할 수 있고, 별개의 가상 머신이 브랜치 경로를 병렬로 준비할 수 있는 인스턴스를 자동으로 검출할 수 있다. 이 경우에, 하이퍼바이저는 어플리케이션이 등록할 수 있고, 어플리케이션이 하이퍼바이저가 클로닝된 가상 머신을 산출하기를 원하는지를 나타내기 위해 응답할 수 있는 통지를 어플리케이션에 제공할 수 있다.

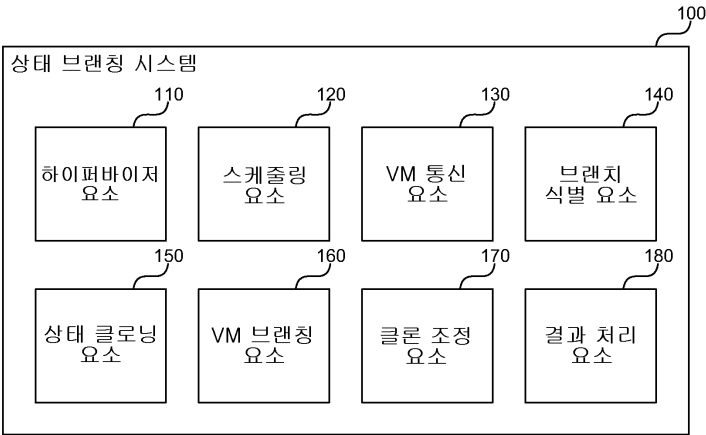
[0041] 일부 실시예들에서, 상태 브랜칭 시스템은 가상 머신 상에서 실행되는 어플리케이션의 다양한 상태에 대한 시간의 트레이블링 백(traveling back)의 효과를 제공하는 새로운 형태의 디버깅을 제공한다. 시스템은 클로닝된 가상 머신의 형태로 어플리케이션의 실행에서 다양한 점에서 체크포인트를 생성한다. 각각의 클로닝된 가상 머신은 특정 시점에서 어플리케이션의 상태를 나타내고 어플리케이션에 발생된 임의의 후속 변화로부터 격리된다. 이는 소프트웨어 개발자가 디버깅을 배속할 수 있게 하고 문제가 발생되기 전에 또는 그 동안 하나 이상의 시점에서 어플리케이션 상태를 조사할 수 있게 한다.

[0042] 일부 실시예들에서, 상태 브랜칭 시스템은 다양한 작업을 수행하기 위한 유사한 알고리즘을 비교하기 위한 A-B 테스트를 제공한다. 예를 들어, 시스템은 관리할 프로세스의 유사한 세트를 갖는 별개의 클로닝된 가상 머신 내에서 각각 설정하고, 그 후 보다 빠른 실행 또는 다른 요인 중 어느 하나의 관점에서 어떤 알고리즘이 더 양호한 결과를 제공하는지를 결정하기 위해 각각의 클로닝된 가상 머신을 병렬로 실행함으로써 운영 체제 페이징 또는 스케줄링 알고리즘을 테스트할 수 있다. 현재 이러한 테스트를 직렬로 재실행하는 것은 새로운 상태를 도입한다. 테스트 장치가 동일한 상태를 매번 설정하기를 시도한다고 해도, 어떠한 불변의 변화는 비교를 곤란하게 한다. 상태 브랜칭 시스템을 사용하면, 각각의 테스트는 알려진 클로닝된 상태로 시작하며, 결과는 예측되지 않은 상태 변화로부터의 보다 적은 영향과 비교될 수 있다.

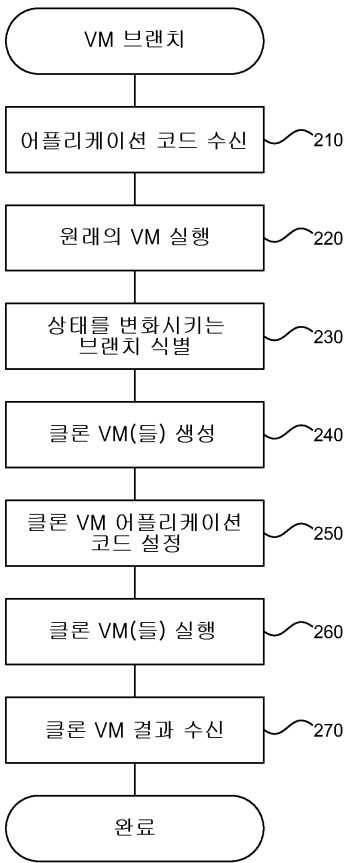
[0043] 전술한 바로부터, 상태 브랜칭 시스템의 특정 실시예들이 예시의 목적으로 본 명세서에 설명되었지만, 다양한 수정이 본 발명의 사상 및 범위를 벗어나지 않고도 이루어질 수 있다는 것이 이해될 것이다. 따라서, 본 발명은 첨부된 청구항에 의한 것을 제외하고는 한정되지 않는다.

도면

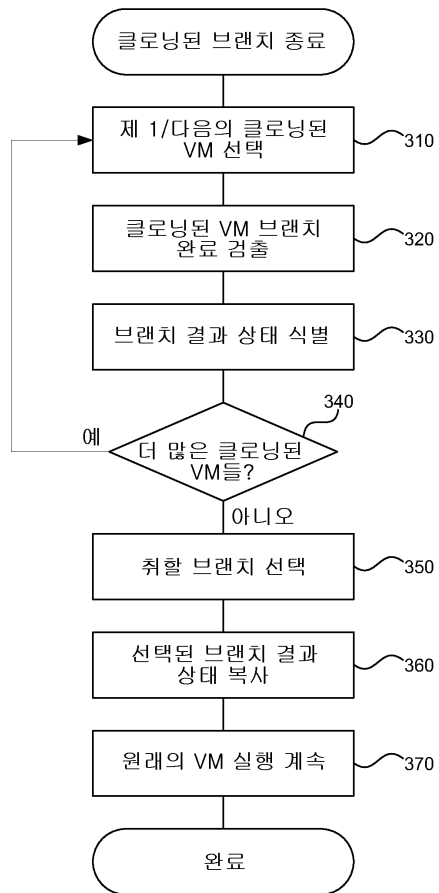
도면1



도면2



도면3



도면4

