(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
H04L 12/28 (2006.01)

(45) 공고일자 2006년11월14일
(11) 등록번호 10-0645829
(24) 등록일자 2006년11월07일

(21) 출원번호 10-2005-0075963
(22) 출원일자 2005년08월18일

(65) 공개번호
(43) 공개일자

(73) 특허권자 고려대학교 산학협력단
서울 성북구 안암동5가1 고려대학교 내

(72) 발명자 황종선
서울 성북구 안암동5가 고려대학교 아산이학관 216호

최봉대
서울 성북구 안암동5가 고려대학교 아산이학관 522호

백맹순
서울 성북구 안암동5가 고려대학교 아산이학관 226호 분산시스템연구실

최성진
서울 성북구 안암동5가 고려대학교 226호 분산시스템 연구실

김홍수
서울 성북구 안암동5가 고려대학교 아산이학과 226호 분산시스템연구실

(74) 대리인 현종철

심사관 : 장대근

(54) 이동 에이전트 결함포용 방법 및 시스템과 이를 이용한다중지역에서의 이동에이전트 결함포용 방법 및 시스템

요약

이동 에이전트 결함포용 방법 및 시스템과 이를 이용한 다중지역에서의 이동에이전트 결함포용 방법 및 시스템이 개시된다. 그 결함포용 방법은 적어도 둘 이상의 호스트를 구비하는 네트워크에서의 이동 에이전트 결함포용방법에 있어서, 이동 에이전트가 자신의 이주경로를 따라 이주할 때, 이전 단계의 제1호스트가 이동에이전트를 복사한 후 현재 단계의 제2호스트로 이주시키는 과정; 제2호스트는 제1호스트에게 단일수행 보장특성이 보장되는지 확인하는 과정; 제1호스트가 현재 단계의 고장을 잘못 검출하여 이동에이전트를 제3호스트로도 이동 에이전트를 이주시켰다면, 제1호스트는 제2호스트 또는 제3호스트 중 하나에게 이동 에이전트 취소 메시지를 보내는 과정; 및 제1호스트가 이동에이전트를 상기 제2호스트에게 이주시키고 그 외의 다른 호스트에게 이동 에이전트를 이주시키지 않았다면, 제1호스트는 제2호스트에게 이동 에이전트의 계속적인 실행을 지시하는 계속 메시지를 보내는 과정을 포함함을 특징으로 한다.

본 발명에 의하면, one-free 단일 수행 보장 property에 기반하여 결함 포용적 이동 에이전트 수행을 가능하게 한다. 또한 다중 이동 에이전트 컴퓨팅 환경에서 발생하는 확장성 문제를 해결한다.

대표도

도 5

명세서

도면의 간단한 설명

도 1은 다중 지역 이동 에이전트 컴퓨팅 환경을 도시한 것이다.

도 2는 이동 에이전트 연산 수행 모델을 도시한 것이다.

도 3은 고아단계와 단일수행 보장 특성 위배를 설명하기 위한 개념도이다.

도 4는 본 발명에 의한 이동에이전트 결함포용 시스템의 구성을 블록도로 도시한 것이다.

도 5는 본 발명에 의한 이동 에이전트의 결함포용 방법을 설명하기 위한 개념도이다.

도 6은 본 발명에 의한 다중지역에서의 이동 에이전트 결함포용 시스템의 구성을 블록도로 도시한 것이다.

도 7은 본 발명에 의한 다중 지역에서의 이동에이전트 결함포용 방법을 설명하는 개념도이다.

도 8은 본 발명에 의한 낙관적 시간적 복제 기법(OTRBA)의 성능 평가에서 정상적인 상태에서의 메시지 비용을 도시한 것이다.

도 9는 본 발명에 의한 낙관적 시간적 복제 기법(OTRBA)의 성능 평가에서 잘못된 결함 검출로 인해 발생하는 메시지 비용을 도시한 것이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 이동에이전트에 관한 것으로서, 특히 이동에이전트의 결함포용 방법 및 시스템과 그를 이용한 다중 지역에서의 이동 에이전트 결함 포용 방법 및 시스템에 관한 것이다.

이동 에이전트(mobile agent)는 이질적인 네트워크 환경에서 노드들을 옮겨 다니면서 사용자를 대신하여 자율적으로 주어진 일을 처리하는 프로그램이다. 이동 에이전트 수행 중 통신 고장이나 노드고장이 일어나면 이동 에이전트의 연산은 부분적으로 혹은 전체적으로 손실되거나 수행이 봉쇄되는 결과를 초래한다. 따라서, 이동 에이전트 시스템(mobile agent system)을 개발하는데 있어서 결함 포용(fault-tolerance)은 기본적이면서도 중요한 고려사항이다.

이동 에이전트 환경에서 결함포용 기법은 크게 Byzantine Failure Apporach (BFA), Checkpoint Based Approach (CBA), 시간적 복제방법(Temporal Replication Based Approach : 이하 TRBA라 한다), 공간적 복제방법(Spatial Replication Based Approach : 이하 SRBA라 한다)로 구분된다.

상기 BFA는 이동 에이전트의 결함 포용적 연산 수행을 보장하는 기초적인 기법으로 비잔틴(Byzantine) 고장 모델에서 발생하는 악의의 이동 에이전트에 대한 공격을 복제 방법을 사용하여 해결한다. 특히 정지 고장 (Crash failure)에 대한 비봉쇄(non-bolcking)문제와 단일 수행 보장 특성을 해결한다.

상기 CBA에서는 검사점에 기반한 롤백 기법과 트랜잭션 기반의 통신 기법을 사용하여 이동 에이전트의 결함 포용적 연산 수행을 보장한다. 이 기법에서는 노드는 이동 에이전트를 수행하기 전에 안정적 저장 공간에 이주한 이동 에이전트를 저장한 다음 이동 에이전트에 대한 연산을 수행한다. 그리고 연산 수행 중간 중간에 정기적으로 이동 에이전트의 연산 중간 상태를 안정적인 저장 공간에 저장한다. 만약 현재 노드에서 고장이 발생한다면 이동 에이전트는 안정적인 저장 공간에 저장된 정보를 이용하여 그 시점부터 다시 이동 에이전트를 다시 수행한다. 그러나 이 기법은 회복 과정이 완료될 때까지 이동 에이전트의 연산 수행이 봉쇄되는 문제점을 가지고 있다.

상기 TRBA는 이전 단계가 에이전트를 저장하여 현재 단계가 고장이 나더라도 이전 단계에서 미리 저장되어 있는 에이전트를 다시 보내어 연산을 수행하는 방법이다. 이를 보다 자세히 설명하면, TRBA는 이동 에이전트의 이동 경로 상에서 현재 이동 에이전트가 수행되고 있는 이전 단계에 있는 노드가 이동 에이전트와 상태를 저장하고 있다가 현재 단계에 있는 이동 에이전트가 고장이 나면 이전 단계에 저장되어 있는 이동 에이전트와 상태들을 다시 보내어 연산을 재수행하여 결함을 해결하는 기법이다. TRBA 기법들 중에서 Mohindra는 이전 단계에 있는 실행장소가 다음번에 있는 실행장소를 탐지하는 고장 탐지 기법을 제안하였다. 이 기법에서는 모든 실행장소는 자신으로부터 일정 거리에 있는 이전 실행장소들에게 하트비트(heartbeat) 메시지를 보내게 된다. 만약 자신의 다음번에 있는 실행장소로부터 해당 메시지를 일정 기간동안 받지 못한다면 현 단계의 실행장소는 다음번에 있는 실행장소가 고장이 났다고 간주를 하고 다른 실행장소로 이동 에이전트를 보내게 된다. 그러나 이 기법에서는 실행장소가 다음 단계 있는 실행장소의 고장을 통신 지연으로 인하여 잘못 판단한 경우에는 단일 수행 보장 특성을 위배하는 문제점을 가지고 있다. Silva는 중복 이동 에이전트 수행을 막기 위해 결함 포용적 룩업 디렉토리를 제안하였다. 룩업 디렉토리는 복제되고 이들 간에는 동기화가 이루어진다. 만약에 이동 에이전트가 이주를 하면은 해당 실행장소는 룩업 디렉토리를 접근하여 해당 이동 에이전트가 다른 실행장소에서 수행되고 있는지를 조사한다. 해당 실행장소에서 이동 에이전트에 대한 모든 연산을 종료하면 해당 실행장소는 룩업 디렉토리에 접근하여 연산 종료 상태를 반영시킨다. 그러나 이 기법에서는 룩업 디렉토리를 설치해야 하는 추가 통신 비용과 복제된 룩업 디렉토리들에 대한 일관성 문제등이 발생한다.

상기 SRBA는 이동 에이전트가 수행되는 각 단계마다 여러 실행장소를 두어 결함 포용을 해결하는 방법이다. 현 단계에 있는 실행장소는 다음 단계로 이동 에이전트를 전송할때 다음 단계에 있는 모든 실행장소들에게 이동 에이전트를 복제하여 이주시킨다. 만약 다음 단계에 있는 실행장소들 중에서 하나의 실행장소가 고장이 나더라도 다른 실행장소들이 고장난 이동 에이전트를 가지고 있어 이동 에이전트의 고장을 극복할 수 있다. SRBA의 가장 큰 단점은 다음 단계에 있는 실행장소들에게 이동 에이전트를 이주시킬 때 추가 통신비용이 발생하여 낮은 통신 대역폭을 가진 인터넷 환경에서는 적당하지 않다. 그러나 TRBA는 SRBA와 같이 여러 실행장소로 이동 에이전트를 보내지 않기 때문에 SRBA에서 발생하는 추가 오버헤드는 발생하지 않는다. 상기 TRBA와 SRBA는 여러 지역으로 구성된 다중 이동 에이전트 컴퓨팅 환경에 적용할 때 확장성 문제를 가지고 있다.

발명이 이루고자 하는 기술적 과제

본 발명이 이루고자 하는 기술적 과제는 1-단계 단일 수행 보장 특성 보장에 기반한, 이동 에이전트의 결함포용 방법 및 시스템을 제공하는 것이다.

본 발명이 이루고자 하는 다른 기술적 과제는 상기 이동 에이전트 결함 포용 방법 및 시스템을 이용하여 다중 지역 이동 에이전트 컴퓨팅 환경에서 이동에이전트 결함포용 방법 및 시스템을 제공하는 것이다.

발명의 구성 및 작용

상기 기술적 과제를 이루기 위한 본 발명에 의한 이동 에이전트의 결함포용 방법은, 적어도 둘 이상의 호스트를 구비하는 네트워크에서의 이동 에이전트 결함포용방법에 있어서, (a) 상기 이동에이전트가 자신의 이주경로를 따라 이주할 때, 이전 단계의 제1호스트가 이동에이전트를 복사한 후 현재 단계의 제2호스트로 이주시키는 과정; (b) 상기 제2호스트는 상기 제1호스트에게 단일수행 보장특성이 보장되는지 확인하는 과정; (c) 상기 제1호스트가 현재 단계의 고장을 잘못 검출하여 이동에이전트를 제3호스트로도 상기 이동 에이전트를 이주시켰다면, 상기 제1호스트는 상기 제2호스트 또는 제3호스트 중 하나에게 상기 이동 에이전트 취소 메시지를 보내는 과정; 및 (d) 상기 제1호스트가 이동에이전트를 상기 제2호스트에게 이주시키고 그 외의 다른 호스트에게 이동 에이전트를 이주시키지 않았다면, 상기 제1호스트는 상기 제2호스트에게 이동 에이전트의 계속적인 실행을 지시하는 계속 메시지를 보내는 과정을 포함함을 특징으로 한다.

상기 (c)과정은 제1호스트로부터 취소 메시지를 받은 호스트는 현재 수행한 이동 에이전트 연산의 결과를 모두 취소하고 상기 이동 에이전트를 제거하는 과정을 더 구비하고, 상기 (d)과정은 제1호스트로부터 계속 메시지를 받으면, 이동 에이전트를 새로운 단계로 이주시키는 과정을 더 구비함이 바람직하다.

상기 기술적 과제를 이루기 위한 본 발명에 의한 이동 에이전트의 결함포용 시스템은, 적어도 둘 이상의 호스트를 구비하는 네트워크에서의 이동 에이전트 결함포용시스템에 있어서, 이동 에이전트가 자신의 이주경로를 따라 이주할 때, 이동에이전트를 복사한 후 다음 단계의 호스트로 이주시키고, 이주된 이동에이전트의 호스트로부터 확인 메시지를 받으면 고장을 잘못 검출하여 이동에이전트를 다른 호스트로 이주시켰으면 취소메시지를 보내고 그렇지 않으면 실행계속을 나타내는 계속메시지를 보내는 제1호스트; 및 상기 제1호스트로부터 이주한 이동에이전트가 수행되며, 수행결과를 반영하기 위해 상기 제1호스트에게 단일수행 보장특성이 보장되는지 확인하고, 확인결과 취소메시지를 받으면 이동에이전트 및 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트의 실행을 계속하게 하는 제2호스트를 포함함을 특징으로 한다.

상기 다른 기술적 과제를 이루기 위한 본 발명에 의한 다중 지역에서의 이동 에이전트의 결함포용 방법은, 네트워크로 연결되는 적어도 두 개의 지역을 포함하며 상기 지역 각각은 적어도 하나의 호스트 및 지역서버를 구비하는, 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 방법에 있어서, 이동 에이전트가 동일 지역 내에 위치한 다른 호스트로 이주하는 지역내 이주과정; 및 이동 에이전트가 다른 지역에 위치한 호스트로 이주하는 지역간 이주과정을 포함함을 특징으로 하고, 상기 지역내 이주과정은 만일 이동 에이전트에 고장이 발생할 경우, 이전 단계의 호스트가 고장난 이동 에이전트를 다른 곳으로 이주시키고, 상기 지역간 이주과정은 이동 에이전트가 다른 지역의 호스트로 이주할 때 호스트가 속하는 지역서버간에도 이동에이전트를 이주하여 이동 에이전트 결함포용을 수행한다. 바람직하게는, 상기 지역 내 이주과정은 (a) 상기 이동에이전트가 자신의 이주경로를 따라 한 단계에서 다음 단계로 이주할 때, 다음 단계의 제2호스트에서 발생할 수 있는 고장에 대비하기 위해 이동 에이전트를 현재 단계의 제1호스트에 복사하는 과정; (b) 제1호스트에 이동 에이전트가 복사된 후, 이동 에이전트는 이동경로를 통해 다음 단계의 제2호스트로 이주하는 과정; (c) 이동 에이전트를 수행하고 있는 제2호스트는 이전 단계의 제1호스트에게 단일수행 보장특성이 보장되는지 확인하는 과정; (d) 상기 제1호스트가 현재 단계의 고장을 잘못 검출하여 이동에이전트를 제2호스트 외의 제3호스트로도 상기 이동 에이전트를 이주시켰다면, 상기 제1호스트는 상기 제2호스트 또는 제3호스트 중 하나에게 상기 이동 에이전트 취소 메시지를 보내는 과정; 및 (e) 상기 제1호스트가 이동에이전트를 상기 제2호스트에게 이주시키고 그 외의 다른 제3호스트에게 이동 에이전트를 이주시키지 않았다면, 상기 제1호스트는 상기 제2호스트에게 이동 에이전트의 계속적인 실행을 지시하는 계속 메시지를 보내는 과정을 구비한다.

바람직하게는 상기 (d)과정은 제1호스트로부터 취소 메시지를 받은 호스트는 현재 수행한 이동 에이전트 연산의 결과를 모두 취소하고 상기 이동 에이전트를 제거하는 과정을 더 구비하고, 상기 (e)과정은 제1호스트로부터 계속 메시지를 받으면, 이동 에이전트를 새로운 단계로 이주시키는 과정을 더 구비한다.

바람직하게는 상기 지역 간 이주과정은 이동 에이전트가 다른 지역의 제2호스트로 이주하기 전에 상기 제2호스트의 고장을 대비하기 위해 현재 단계의 제1호스트에 상기 이동에이전트를 복사하는 과정; 이동 에이전트를 다른 지역의 제2호스트로 이주시키는 과정; 현재 단계의 제1호스트는 같은 지역의 제1지역서버로 상기 이동 에이전트를 이동시키고, 상기 제1지역서버는 상기 이동에이전트가 다른 지역으로 이주한 지역의 제2지역서버로 이동에이전트를 이주시키는 과정; 이동 에이전트를 수행한 제2호스트는 상기 제2호스트가 속한 지역의 제2지역서버에게 확인 요청 메시지를 보내는 과정; 제2호스트가 제2지역서버로부터 다른 호스트로 이동 에이전트를 이주시키지 않았다는 확인 메시지를 받으면, 상기 제2호스트는 이동 에이전트 수행 결과를 반영하고, 이동 에이전트를 다음 호스트로 이주시키는 과정; 및 만일 상기 제2지역서버가 현재 수행중인 호스트의 고장을 잘못 검출하여 다른 호스트로 이동 에이전트를 보냈다면 상기 제2지역서버는 제2호스트에게 취소 메시지를 보내는 과정을 구비한다.

상기 다른 기술적 과제를 이루기 위한 본 발명에 의한 다중 지역에서의 이동 에이전트의 결함포용 시스템은, 네트워크로 연결되는 적어도 두 개의 지역을 포함하며 상기 지역 각각은 적어도 하나의 호스트 및 지역서버를 구비하는, 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 시스템에 있어서, 이동 에이전트가 자신의 이주경로를 따라 이주할 때, 지역내 이동을 처리하는 지역내 이동처리부 및 지역간 이동을 처리하는 지역간 이동처리부를 구비하는 제1호스트; 이동에이전트의 지역내 이동일 경우, 상기 제1호스트로부터 이주한 이동에이전트가 수행되며, 수행결과를 반영하기 위해 상기 제1호스트에게 단일수행 보장특성이 보장되는지 확인하고, 확인결과 취소메시지를 받으면 이동에이전트 및 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트의 실행을 계속하게 하는 제2호스트; 이동 에이전트의 지역간 이동일 경우, 상기 제1호스트로부터 복사된 이동 에이전트를 저장하고 있으며, 상기 이동에이전트가 상기 제3호스트로 이동할 때 상기 다른 지역의 지역서버(제2지역서버)에 상기 이동에이전트를 이주시키는 제1지역서버; 상기 제1지역서버가 이주시킨 이동에이전트를 저장하고 있으며, 같은 지역의 제3호스트의 확인요청 메시지를 받으면 다른 호스트로 이동 에이전트를 이

주시켰는지 여부를 체크하여 이주 시켰으면 제3호스트에게 취소메시지를 보내고 그렇지 않으면 계속메시지를 보내는 제2지역서버; 및 상기 제2지역서버의 지역에 존재하며, 상기 이동 에이전트가 수행되어 결과를 반영하고자 할 때 상기 제2지역서버에게 확인요청 메시지를 보내어 취소메시지를 받으면 이동에이전트와 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트가 계속 실행되게 하는 제3호스트를 포함함을 특징으로 하고, 상기 제1호스트의 지역내 이동처리부는 지역내 이동일 경우, 이동에이전트를 복사한 후 다음 단계의 호스트로 이주시키고 이주된 이동에이전트의 호스트로부터 확인 메시지를 받으면 고장을 잘못 검출하여 이동에이전트를 다른 호스트로 이주시켰으면 취소메시지를 보내고 그렇지 않으면 실행계속을 나타내는 계속메시지를 보내며, 상기 제1호스트의 지역간 이동처리부는 지역간 이동일 경우, 이동 에이전트가 다른 지역의 호스트로 이주할 때, 이동에이전트를 복사한 후 상기 다른 지역의 호스트(제3호스트)로 이주시키고 자기가 속한 지역의 지역서버(제1지역서버)로 상기 이동 에이전트를 이동시킨다.

그리고 상기 기재된 발명을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

이하, 첨부된 도면들을 참조하여 본 발명에 따른 이동 에이전트의 결함포용 방법 및 시스템과 이를 이용한 다중지역에서의 이동에이전트의 결함포용 방법 및 시스템에 대해 상세히 설명한다.

다중 지역 이동 에이전트 컴퓨팅 환경은 크게 이동 에이전트, 노드(또는 실행장소(Place)), 지역(Region), 지역 서버(Region server : RS), 룩업서버(Lookup Server : LS)로 구성된다. 도 1은 다중 지역 이동 에이전트 컴퓨팅 환경을 나타내며, 이동에이전트(Mobile Agent), 노드 또는 실행장소(Place), 지역(Region)과 지역서버(Region Server), 룩업서버(Lookup Server:LS)를 구비한다.

상기 이동 에이전트(Mobile Agent)를 설명하면, 이동 에이전트는 코드, 데이터, 수행 상태, 이동 메타데이터로 구성된 이동 객체이다. 이동 에이전트는 각 노드의 수행 결과에 따라 다음 노드를 동적으로 선택하여 이동하거나, 이동계획(itinerary)에 따라 노드와 노드사이를 이동하며 주어진 작업을 자치적으로 수행한다. 이동 에이전트가 이동을 시작한 출발지에서부터 목적지까지를 이동에이전트의 이주 경로(Migration Path)라 한다.

상기 노드 또는 실행장소(Place)를 설명하면, 노드는 이동 에이전트가 수행할 수 있는 수행환경인 실행장소(place)를 제공한다. 다시 말하면, 노드에는 이동 에이전트 시스템이 설치되어 있어 이동 에이전트를 생성, 수행, 이주, 종료시킬 수 있는 기능을 제공한다. 이동 에이전트를 맨 처음 생성한 노드를 홈 노드(Home node : HN)이라고 한다. 본 발명에서는 노드와 호스트(Host)가 동일 개념으로 사용된다.

상기 지역과 지역서버에 대해 설명하면, 같은 권한을 가진 노드들의 집합을 지역이라고 한다. 그리고 각 지역의 권한을 책임지는 서버를 지역 서버(Mobile Agent Region Server : MARS)라 한다. 지역 서버는 자신의 지역에서 생성된 이동 에이전트에 대해서 룩업 서버와 협력하여 네이밍 서비스를 제공한다. 또한 자신의 지역에 위치한 이동 에이전트들에 대한 위치 관리 및 메시지 전달을 수행한다.

상기 룩업서버를 설명하면, 룩업 서버(Lookup server : LS)는 다중 지역 이동 에이전트 컴퓨팅 환경에서 이동 에이전트의 전역 네이밍 서비스를 제공하며 또한 이동 에이전트에 대한 룩업 서비스를 제공한다. 룩업 서버는 모든 지역에서 생성된 이동 에이전트에 대한 위치 관리 정보를 유지 관리한다.

상기 이동 에이전트 연산수행 모델을 설명하기로 한다. 이동 에이전트 컴퓨팅 환경에서 이동 에이전트 연산은 일련의 노드들(실행장소들)에서 수행된다. 이동 에이전트는 각 노드의 수행 결과에 따라 다음 노드를 동적으로 선택하여 이동하거나, 이동계획(itinerary)에 따라 노드와 노드사이를 이동하며 주어진 작업을 자치적으로 수행한다. 이동 에이전트가 이동을 시작한 출발지에서부터 목적지까지를 이동에이전트의 이주 경로(Migration Path)라 한다. 하나의 실행장소에서 이동 에이전트가 수행되는 논리적 연산 수행을 단계(Stage)라고 한다. 이동 에이전트는 한 단계에서 현재 실행장소에서 수행해야할 연산을 수행하게 된다. 하나의 단계에서 수행된 이동 에이전트는 새로운 내부 상태와 또한 새로운 실행장소의 상태를 초래하게 된다. 도 2는 이동 에이전트 연산 수행 모델을 도시한 것이다.

본 발명에 의한 이동 에이전트 결함포용 방법(이를 낙관적 결함포용방법이라고도 함)을 설명하기 위해, 먼저 단일수행 보장 특성에 대해 설명한다. 이동 에이전트 컴퓨팅 시스템에서 이동 에이전트의 수행은 단일수행 보장 특성을 만족해야한다. 단일수행 보장 특성이란 이동 에이전트에 의해 수행되는 단계에서의 수행이 정확히 한번만 수행이 되어야한다는 것을 말한다. 단일수행 보장 특성을 표현하기 위해서 이동 에이전트의 연산 수행에 대한 다음과 같은 구성요소를 고려한다.

이동 에이전트(mobile agent) : $\overrightarrow{ma_i}$

실행장소(place) : p_i

이동 에이전트 $\overrightarrow{ma_i}$ 가 수행하는 장소 p_i : $[\overrightarrow{ma_i} \circlearrowright p_i]$

단계(stage) : S_i

사이트(site) : $site_i$

사이트, $site_i$ 에 위치한 장소, p_i 에서 이동 에이전트, $\overrightarrow{ma_i}$ 의 수행하는 단계 : $[\frac{\circlearrowright S_i}{\overrightarrow{ma_i} \circlearrowright p_i}]$

사이트, $site_i$ 에서 단계군, S_i 의 수행하는 단계 : $[\frac{site_i \circlearrowright S_i}{\overrightarrow{ma_i} \circlearrowright p_i}]$

이동 에이전트, $\overrightarrow{ma_i}$ 의 두 번째 이주 : $\overleftarrow{ma_i}$

이주하는 이동 에이전트, $\overrightarrow{ma_i}$ 의 단계 : $\widehat{S_i}$

이주한 이동 에이전트, $\overrightarrow{ma_i}$ 의 단계 : $\widehat{S_i}$

단계 S_i 로부터 단계 S_j 로 이동에이전트, $\overrightarrow{ma_i}$ 가 이주하는 단계 : $\widehat{S_i} \rightsquigarrow \widehat{S_j}$

고아 단계 : $\not\exists S_i$

이동 에이전트 $\overrightarrow{ma_i}$ 는 이동 에이전트 생성자자 위탁한 작업(task)을 수행할 수 있는 프로그램이다. 실행장소 p_i 는 이동 에이전트가 각 사이트에 의해 수행되고 제공될 수 있는 논리적인 환경이다. $[\overrightarrow{ma_i} \circlearrowright p_i]$ 는 이주한 이동 에이전트 $\overrightarrow{ma_i}$ 를 현재 수행하는 실행장소 p_i의 상태이다. 단계 S_i 는 동일한 작업을 수행하는 이동 에이전트 $\overrightarrow{ma_i}$ 가 수행하는 실행장소 p_i의 집합이다. 연산 수행 상태 $[\frac{\circlearrowright S_i}{\overrightarrow{ma_i} \circlearrowright p_i}]$ 는 동일 이동 에이전트 $\overrightarrow{ma_i}$ 에 의해 수행되고 있는 한 단계 내에 존재하는 실행장소들의 집합이다. 단계 $\widehat{S_i}$ 는 해당 단계에서 수행한 이전 이동 에이전트 $\overrightarrow{ma_{i-1}}$ 의 결과를 다음 단계로 새로운 이동 에이전트 $\overrightarrow{ma_i}$ 를 전송하는 단계를 나타낸다. 단계 $\widehat{S_i}$ 는 이전 단계로부터 이동 에이전트를 수신하는 단계를 나타낸다. 마지막으로, 고아 단계 $\not\exists S_i$는 또 다른 단계가 동일한 이동 에이전트를 수행하고 있는 경우 자신의 단계는 유효하지 못하게 되는 무효한 상태를 나타낸다.

위에서 살펴본 연산수행 구성요소에 따라 단일수행 보장 특성에 대한 새로운 조건을 생성한다.

Condition 1 $[\frac{\circlearrowright S_m}{\overrightarrow{ma_i} \circlearrowright p_j}]$ 과 $[\frac{\circlearrowright S_m}{\overrightarrow{ma_i} \circlearrowright p_i}]$ *이 동시에 존재하지 않을 경우 이동 에이전트* $\overrightarrow{ma_i}$*의 연산 수행은 단일 수행 보장 특성(property)을 보장한다.*

상기 Condition 1은 만약 같은 단계에서 다른 두 개 또는 그 이상의 실행장소에서 수행중인 동일한 이동 에이전트가 공존하지 않는다면 단일수행 보장 특성을 만족하는 이동 에이전트의 수행을 의미한다. 그러나, Condition 1은 이동 에이전트 수행을 위한 결함포용 특성을 정의하기에는 불완전하다. 따라서 새롭게 고아단계 $^{\Phi}S_i$를 정의한다. 고아단계에 대한 정의는 다음과 같다.

Definition 1 $[\frac{H_l \bigcirc S_l}{\overrightarrow{ma}_i \bigcirc p_k}]$ 과 $[\frac{H_j \bigcirc S_j}{\overrightarrow{ma}_i \bigcirc p_m}]$ *이 동시에 존재할 때* $\overset{\frown}{S_l} \looparrowright \overset{\frown}{S_{l+1}}$ *을 만족한다면 이동 에이전트* $\overrightarrow{ma}_i$*의 단계* S_j*는 고아 단계* $^{\Phi}S_j$*라고 정의한다.*

상기 Definition 1는 이전 단계 S_l이 현재 단계 S_j의 결함을 잘못 발견하고 다른 호스트 H_l로 이동 에이전트 $\overrightarrow{ma}_i$를 전송하고 동일한 이동 에이전트 $\overrightarrow{ma}_i$를 수신한 새로운 단계 S_l가 다음 단계 S_{l+1}으로 이동 에이전트를 전송한 경우에 고아 단계 $^{\Phi}S_j$ 가 존재한다는 것을 나타낸다. Definition 1에서 $\overset{\frown}{S_l} \looparrowright \overset{\frown}{S_{l+1}}$ 는 현재 단계 S_l이 다음 단계 S_{l+1}로 이동 에이전트를 전송하는 것을 나타내고 $\looparrowright$는 이동 에이전트가 이주하는 것을 의미한다.

결함 포용적인 이동 에이전트 연산수행 특성과 관련된 종래 기술들은 단일 수행 보장 특성에만 초점을 맞추었다. 그러나 본 발명에서의 고아 단계는 신뢰적인 이동 에이전트 수행을 위해 고려되어야 할 중요한 문제이다. 왜냐하면 고아 단계가 이동 에이전트 수행 단계에서 존재할지라도 단일수행 보장 특성을 항상 위반하지는 않기 때문이다. 도 3은 고아단계와 단일수행 보장 특성 위배를 설명하기 위한 개념도이다. 도 3에서 호스트 H_2에서 단계 S_1는 고아 단계이고, 단계 S_1과 S_1'는 이동 에이전트 수행하는데 있어서 단일수행 보장 특성을 위반한다. 그러나 만약 고아 단계 $^{\Phi}S_1$이 호스트 H_2에서 이동 에이전트의 결과에 영향을 미치지 않으면서 호스트 H_2에서 취소되고 또한 만약 단계 S_1로부터 이동 에이전트가 단계 S_3에 의해 호스트 H_3에서 전송되지 않거나 취소되지 않는다면, 전체 이동 에이전트 수행에 대한 단일수행 보장 특성은 위배되지 않는다. 고아 단계의 발생은 단일수행 보장 특성을 위배할 수도 있지만, 항상 위배되지는 않는다. 이러한 특징을 반영하기 위해 이동 에이전트 수행 상태에 대한 두 가지 조건을 새롭게 제시한다.

Condition 2 *불안정한 상태(Precarious state) : 만약 고아 단계* $^{\Phi}S_i$*가 전체 이동 에이전트 수행* $\sum_{i=0}^{n}[\frac{H_i \bigcirc S_i}{\overrightarrow{ma}_i \bigcirc p_i}]$ *에서 존재한다면 이동 에이전트 수행 상태* $Ex(\overrightarrow{ma}_i)$*는 불안한 상태이다.*

이동 에이전트가 자신의 작업 수행 중 전체 작업 수행 단계에서 고아 단계가 존재하면, 해당 이동 에이전트의 전체 작업 상태는 불안정한 상태(Precarious state)가 된다. 그러나, 불안정한 상태가 단일수행 보장 특성을 항상 위배하지 않는다. 단일수행 보장 특성의 위배 상태들과 불안정한 이동 에이전트 수행 상태를 구별하기 위하여, 이동 에이전트 수행에서 위배된 상태들에 대한 조건을 정의한다.

Condition 3 *위배 상태(Violating state) :* 만약 고아 단계 $^{\Phi}S_i$가 전체 이동 에이전트 수행 $\sum_{i=0}^{n}[\frac{H_i \bigcirc S_i}{\overrightarrow{ma}_i \bigcirc p_i}]$ 에 존재하고 고아 단계 $Re(^{\Phi}S_i)$의 결과가 다음 단계 S_{i+1}의 시작에 영향을 미친다면 이동 에이전트의 수행 상태 $Ex(\overrightarrow{ma}_i)$는 위배 상태가 된다.

도 3에서 이동 에이전트 수행의 불안정한 상태는 단계 S_1과 S_1'이 공존한다는 것을 의미하고 위배 상태는 S_1, S_1'의 결과인 이동 에이전트 $\overrightarrow{ma}_i$가 단계 S_2의 시작에 영향을 미친다는 것을 의미한다. 이동 에이전트의 결함 포용을 제공하기 위해, 전체 이동 에이전트 수행의 상태를 두 가지 정의를 제시한다.

Definition 2 *강건한 상태 (Roust state) : 만약 어떠한 고아 단계도 전체 이동 에이전트 수행에서 발생되지 않는다면 이동 에이전트 수행 상태는 강건(robust)한 상태라고 정의한다.*

Definition 3 *회복 가능 상태 (Recoverable state) : 이동 에이전트의 작업 수행 중 이동 에이전트의 작업 상태가 불안정한 상태로 전환됐다 할지라도 고아 단계의 작업 결과가 해당 호스트에서 취소되고, 고아 스테이지에서 발생된 이동 에이전트가 새로이 이주된 호스트에서 종료된다면 이러한 이동 에이전트의 작업 상태는 회복 가능 상태라 정의한다.*

종래에는 전체 이동 에이전트 수행이 강건하도록 하는 비관적(pessimistic) 기법을 연구해 왔다. 그러나 본 발명에서는 전체 이동 에이전트 수행이 회복가능한 상태로 되게 하는 낙관적(optimistic) 기법에 초점을 맞춘다. 낙관적 기법에서는 고아 단계의 발생을 허용한다. 본 발명에서는 전체 이동 에이전트 수행이 회복 가능한 상태가 되도록 하는 낙관적 시간적 복제 기법(OTRBA)을 제안한다. 단일수행 보장 특성을 다음과 같이 세 가지로 구분한다. 즉 이동 에이전트 실행을 위한 결함 포용 속성은 세 가지 유형으로 분류된다. 이러한 분류는 이동 에이전트 실행에 대한 고장을 보다 유연하게 다룰 수 있게 한다.

Definition 4 *이동 에이전트의 수행 중 어느 수행 스테이지도 고아 단계가 되지 않는다면 해당 이동 에이전트의 수행은 엄격한 단일 수행 보장 특성 (strictly exactly-once property) 을 만족한다고 정의한다.*

상기 Definition 4의 엄격한 단일수행 특성은 Definition 2 의 강건한 상태를 보장한다. 엄격한 단일수행 보장 특성은 기존 연구에서 말하는 단일수행 보장 특성과 동일하다. 그러나 엄격한 단일수행 보장 특성이 유연성을 제공하지 않기 때문에 OTRBA는 이 속성에 기반을 두지 않는다. 다음의 정의는 OTRBA가 기반을 두고 있는 속성을 나타낸다.

Definition 5 *이동 에이전트의 전체 수행 중 동일 작업에 대한 2개 이상의 고아 단계가 동시에 존재하지 않는다면 해당 이동 에이전트의 수행은 1-단계 자유 단일 수행 보장 특성(one-free exactly-once property)을 만족한다고 정의한다.*

상기 Definition 5의 1-단계 자유 단일수행 보장 특성은 Definition 3의 회복 가능 상태와 연관된다. 그러나 1-단계 자유 단일수행 보장 특성은 항상 복구 가능한 상태를 보장하는 것은 아니다. OTRBA에서는 1-단계 자유 단일수행 보장 특성을 가지는 이동 에이전트의 연산 수행이 회복 가능한 상태가 되도록 하는 방법을 제안한다.

Definition 6 *이동 에이전트의 전체 수행 중 동일 작업에 대한 K+1개 이상의 스테이지가 동시에 존재하지 않는다면 해당 이동 에이전트의 수행은 K-단계 자유 단일 수행 보장 특성(k-free exactly once property)을 만족한다고 정의한다.*

K-단계 자유 단일수행 보장 특성은 다른 속성에 비해 가장 느슨한 속성이다. K-단계 자유 단일수행 보장 특성에서는 회복 가능 상태로 전이하는 비용이 너무 높기 때문에 본 발명에서는 고려하지 않는다.

한편, 본 발명에 의한 이동에이전트 결함포용 방법(낙관적 시간적 복제 방법: OTRBA) 및 시스템에 대해 설명한다.

도 4는 본 발명에 의한 이동에이전트 결함포용 시스템의 구성을 블록도로 도시한 것으로서, 적어도 둘 이상의 호스트를 구비하는 네트워크에서 최소한의 구성요소로 제1호스트(400) 및 제2호스트(450)로 이루어진다.

상기 제1호스트(400)는 이동 에이전트가 자신의 이주경로를 따라 이주할 때, 이동에이전트를 복사한 후 다음 단계의 호스트로 이주시키고, 이주된 이동에이전트의 호스트로부터 확인 메시지를 받으면 고장을 잘못 검출하여 이동에이전트를 다른 호스트로 이주시켰으면 취소메시지를 보내고 그렇지 않으면 실행계속을 나타내는 계속메시지를 보낸다. 상기 제2호스트(450)는 상기 제1호스트(400)로부터 이주한 이동에이전트가 수행되며, 수행결과를 반영하기 위해 상기 제1호스트(400)에게 단일수행 보장특성이 보장되는지 확인하고, 확인결과 취소메시지를 받으면 이동에이전트 및 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트의 실행을 계속하게 한다.

도 5는 본 발명에 의한 이동 에이전트의 결함포용 방법을 설명하기 위한 개념도로서, 도 5를 참조하여 이동 에이전트 결함포용 방법을 설명하기로 한다.

상기 낙관적 시간적 복제기법(OTRBA)에서 이동 에이전트는 자신의 이주 경로를 따라 한 단계에서 다음 단계로 이주한다. (??) 생존 기간 동안 이동 에이전트는 호스트 고장, 실행장소 고장, 이동 에이전트 고장, 그리고 통신 고장과 같은 다양한 고장들을 접할 수 있다. 이러한 고장의 영향으로 인해 이동 에이전트는 자신의 연산 수행을 지속할 수 없게 된다. OTRBA는 고장을 처리하기 위해서 이전 호스트에 의해서 고장 단계가 복구되기를 기다리는 대신 이전 단계의 호스트가 고장난 이동 에이전트를 다른 곳으로 이주시키는 방법을 사용한다.

만약 이전 단계에 있는 호스트가 현재 단계의 고장을 잘못 검출한다면(??) 고아 단계는 다른 단계에서 동일 이동 에이전트를 수신을 시작하는 동안에도 자신의 연산을 계속해서 수행한다. OTRBA에서는 고아 단계를 처리하는 책임은 이전 단계와 고아 단계에 있다. 현재 단계가 고장이 났다고 판단한 이전 단계가 두 번째 이주를 실행한 후(??), 고아 단계의 실행장소는 이전 실행장소에게 단일 수행 보장 특성이 보장되는지 확인하도록 요청한다.(??) 이전 단계에 있는 실행장소가 고아 단계의 실행장소로부터 이 요구 메시지를 받으면, 이전 단계의 실행장소는 확인 메시지(confirm message)를 보냈는지 보내

지 않았는지를 결정한다. 만약 실행 장소가 동일 이동 에이전트를 다른 호스트에 이주시켰다면 이전 단계는 현재 단계가 고아 단계라고 판단하고 제거 메시지(remove message, 또는 discard message)를 보내게 된다.(??) 그렇지 않다면, 이전 단계는 이동 에이전트를 계속해서 실행하라는 의미에서 계속 메시지 (continuity message)를 현재 단계에 보내게 된다.

만약 ??과정에서 현재 단계의 실행장소가 계속 메시지를 받게되면(??), 현재 단계는 고아 단계가 발생하지 않았다고 간주를 하고 이동 에이전트를 새로운 단계로 이주시킨다.(??)

만약 ??과정에서 현재 단계의 실행장소가 제거 메시지를 받으면 현재 단계의 실행자소는 자신의 단계가 고아 단계라고 판단을 한 후 현재 수행한 이동 에이전트 연산의 결과를 모두 취소하고 제거한다.(??)

다음으로, 본 발명에 의한 다중 지역에서의 이동 에이전트 결함포용 방법을 설명하기로 한다. 즉 이동 에이전트를 위한 결함포용 기법의 확장성을 높이고자 다중 지역으로 구성되는 이동 에이전트 환경에서의 이동 에이전트 지역서버(Mobile Agent Region Server: MARS)와 연동할 수 있는 낙관적 결함 포용 기법에 대해서 설명한다. 한 지역 안에서의 이동 에이전트가 이주하는 지역 내 이주(intra-region 이주)의 경우, 이동 에이전트의 이전 호스트가 현재의 이동 에이전트의 결함포용에 대한 책임을 진다. 이동 에이전트가 지역 간 이주(inter-region 이주)를 수행하는 경우, 이동 에이전트의 복사본이 해당 지역의 MARS에게 복사되어 해당 지역의 이동 에이전트의 최초 호스트의 수행에 대한 결함 포용성에 대한 책임을 진다. 본 발명에 의한 다중 지역에서의 이동에이전트 결함포용 방법(낙관적 결함 포용 기법)은 다중 지역 이동 에이전트 컴퓨팅 환경에 적용되어 이동 에이전트의 결함 포용성을 제공하며, 또한 이동 에이전트 수행에 대한 확장성을 높인다. 다중 지역 이동 에이전트 컴퓨팅 환경에서의 낙관적 결함포용 기법은 이동 에이전트의 두 가지 이주 형태, 즉 같은 지역 내에서의 이주(intra-region 이주)와 다른 지역으로의 이주(inter-region 이주)에 따라 다르게 적용된다.

도 6은 본 발명에 의한 다중지역에서의 이동 에이전트 결함포용 시스템의 구성을 블록도로 도시한 것으로서, 상기 시스템은 네트워크로 연결되는 적어도 두 개의 지역을 포함하며 상기 지역 각각은 적어도 하나의 호스트 및 지역서버를 구비하며, 구체적으로 제1호스트(600), 제2호스트(620), 제1지역서버(640), 제2지역서버(660), 제3호스트(680)를 최소한도로 구비한다.

상기 제1호스트(600)는 이동 에이전트가 자신의 이주경로를 따라 이주할 때, 지역내 이동을 처리하는 지역내 이동처리부(602) 및 지역간 이동을 처리하는 지역간 이동처리부(604)를 구비한다.

상기 지역내 이동처리부(602)는 지역내 이동일 경우, 이동에이전트를 복사한 후 다음 단계의 호스트로 이주시키고 이주된 이동에이전트의 호스트로부터 확인 메시지를 받으면 고장을 잘못 검출하여 이동에이전트를 다른 호스트로 이주시켰으면 취소메시지를 보내고 그렇지 않으면 실행계속을 나타내는 계속메시지를 보낸다. 상기 지역간 이동처리부(604)는 지역간 이동일 경우, 이동 에이전트가 다른 지역의 호스트로 이주할 때, 이동에이전트를 복사한 후 상기 다른 지역의 호스트(제3호스트)로 이주시키고 자기가 속한 지역의 지역서버(제1지역서버)로 상기 이동 에이전트를 이동시킨다.

상기 제2호스트(620)는 이동에이전트의 지역내 이동일 경우, 상기 제1호스트로(600)부터 이주한 이동에이전트가 수행되며, 수행결과를 반영하기 위해 상기 제1호스트(600)에게 단일수행 보장특성이 보장되는지 확인하고, 확인결과 취소메시지를 받으면 이동에이전트 및 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트의 실행을 계속하게 한다.

상기 제1지역서버(640)는 이동 에이전트의 지역간 이동일 경우, 상기 제1호스트(600)로부터 복사된 이동 에이전트를 저장하고 있으며, 상기 이동에이전트가 상기 제3호스트(680)로 이동할 때 상기 다른 지역의 지역서버(제2지역서버)에 상기 이동에이전트를 이주시킨다.

상기 제2지역서버(660)는 상기 제1지역서버(640)가 이주시킨 이동에이전트를 저장하고 있으며, 같은 지역의 제3호스트(680)의 확인요청 메시지를 받으면 다른 호스트로 이동 에이전트를 이주시켰는지 여부를 체크하여 이주 시켰으면 제3호스트(680)에게 취소메시지를 보내고 그렇지 않으면 계속메시지를 보낸다.

상기 제3호스트(680)는 상기 제2지역서버(660)의 지역에 존재하며, 상기 이동 에이전트가 수행되어 결과를 반영하고자 할 때 상기 제2지역서버(660)에게 확인요청 메시지를 보내어 취소메시지를 받으면 이동에이전트와 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트가 계속 실행되게 한다.

본 발명에 의한 다중 지역에서의 이동 에이전트의 결함포용 방법은 이동 에이전트가 동일 지역 내에 위치한 다른 호스트로 이주하는 지역내 이주과정 및 이동 에이전트가 다른 지역에 위치한 호스트로 이주하는 지역간 이주과정을 포함한다.

먼저, 같은 지역 내에서의 이주(intra-region 이주)를 설명하기로 한다. 이동 에이전트가 동일 지역(region)내에 위치한 다른 호스트로 이주하는 것을 같은 지역내에서의 이주라고 한다. 이동 에이전트가 동일 지역내의 호스트로 이주할 경우에 결함 포용 기법은 세 가지 순서로 이루어지게 된다. 세 가지 순서는 다음과 같다.

1. 이동 에이전트 복사 : 다음 호스트에서 발생할 수 있는 고장(failure)에 대비하기 위해 이동 에이전트는 다음 호스트로 이주되기 전에 현재의 호스트에 복사된다.

2. 이동 에이전트 이주 : 현재의 호스트에 이동 에이전트가 복사된 후, 이동 에이전트는 이동 경로를 통해 다음 호스트로 이주하게 된다.

3. 확인 요청(confirm request) : 이동 에이전트를 수행하고 있는 호스트는 확인 요청 메시지(confirmation request message)를 이전 호스트에 보내어 이동 에이전트의 수행을 확인시켜준다. 현재의 호스트에서는 이전 호스트가 이동 에이전트를 다른 호스트로 이주시키지 않았다는 의미를 지닌 확인 메시지(confirm message)를 받게 되면, 호스트는 이동 에이전트의 수행을 반영하고 이동 에이전트는 다음 호스트로 이주하게 된다.

도 7은 본 발명에 의한 다중 지역에서의 이동에이전트 결함포용 방법을 설명하는 개념도이다. 도 7을 참조하면, 이동 에이전트 $\overrightarrow{ma}_i$는 호스트 $R_0.H_0$에 의해 만들어지고 처음으로 $R_0.H_1$으로 이주하게 된다. 하지만 호스트 $R_0.H_0$는 호스트 $R_0.H_1$의 고장을 잘못 검출하고 다시 이동 에이전트 $\overrightarrow{ma}_i$를 호스트 $R_0.H_3$로 이주시킨다. 호스트 $R_0.H_0$가 확인 요청 메시지를 호스트 $R_0.H_1$으로부터 받게 되면, 호스트는 이미 이동 에이전트 $\overrightarrow{ma}_i$를 또 다른 호스트 $R_0.H_3$로 보낸 상태이기 때문에 취소 메시지(discard message)를 보내게 된다. 만일 호스트 $R_0.H_0$가 이동 에이전트 $\overrightarrow{ma}_i$를 다른 호스트로 보내지 않았다면 호스트는 확인 메시지를 보내고 호스트 $R_0.H_1$은 이동 에이전트 수행 결과를 반영하게 된다.

다음으로 다른 지역으로의 이주(inter-region 아주)에 대해 설명한다. 만약 이동 에이전트가 다른 지역에 위치한 호스트로 이주하는 경우 다른 지역으로의 이주라고 부른다. 이동 에이전트가 다른 지역의 호스트로 이주했을 때, 결함 포용 기법은 4가지의 절차로 구성된다. 그 절차는 다음과 같다.

1. 이동 에이전트 복사: 이동 에이전트는 다른 지역의 다음 호스트로 이주하기 전에 다음 호스트의 고장을 대비하기 위해 현재 호스트에 복사된다.

2. 이동 에이전트 이주: 현재 호스트가 이동 에이전트를 복사한 후에, 이동 에이전트는 다른 지역의 다음 호스트로 이주한다.

3. MARS로 이동: 현재 호스트는 같은 지역의 MARS로 이동 에이전트를 이동시키고, 이 MARS는 이주한 호스트 지역의 MARS로 이동 에이전트를 이동 시킨다.

4. 확인 요청: 이동 에이전트를 수행한 호스트는 호스트의 MARS에게 확인 요청 메시지를 보냄으로써 수행의 확인을 요청한다. MARS가 다른 호스트로 이동 에이전트를 이주시키지 않았다는 확인 메시지를 호스트가 받는다면, 호스트는 이동 에이전트 수행의 결과를 반영한고, 이동 에이전트는 다음 호스트로 이주되게 된다. 만약 MARS가 현재 수행중인 호스트의 고장을 잘못 검출하여 다른 호스트로 이동 에이전트를 보냈다면 MARS는 호스트에게 취소 (discard) 메시지를 보내게 된다. 이 경우 MARS로부터 취소 메시지를 받은 호스트는 이동 에이전트 수행 결과를 버리게 된다.

도 7을 참조하면, 이동 에이전트 $\overrightarrow{ma}_i$는 호스트 $R_0.H_3$ 에서 $R_1.H_2$ 로 이주되고 region0의 $MARS_0$로 이주된다. $MARS_0$은 $MARS_1$로 이동 에이전트 $\overrightarrow{ma}_i$를 보낸다. 그러나 $MARS_1$은 호스트 $R_1.H_2$의 고장을 잘못 탐지하고 이동 에이전트를 호스트 $R_1.H_3$로 새롭게 이주시킨다. 이러한 상황에서 만약 $MARS_1$가 호스트 $R_1.H_2$로부터 확인 요청 메시지를 받은 경우, $MARS_1$는 $R_1.H_3$로 이동 에이전트를 보냈기 때문에 취소 메시지를 해당 호스트에게 보내게 된다. 만약 $MARS_1$이 이동 에이전트 $\overrightarrow{ma}_i$를 다른 호스트로 이주시키지 않는다면, $MARS_1$은 확인 메시지를 보내고 호스트 $R_1.H_2$는 이동 에이전트 수행의 결과를 반영한다.

한편, 본 발명에 의한 다중지역에서의 이동 에이전트 결함포용 방법 및 시스템의 정당성을 증명하고 성능을 평가하기로 한다.

먼저, 본 발명에 의한 이동에이전트의 결함포용 방법(낙관적 시간적 복제 기법)에 대한 정당성 증명은 다음과 같다. 낙관적 시간적 복제 기법 (OTRBA)의 정당성은 Theorem 1에 의해 증명된다.

Theorem 1 *1-단계 자유 단일 수행 보장 특성을 따르는 OTRBA 프로토콜은 회복가능 상태를 보장하고, 전체 이동 에이전트 실행에 대한 단일 수행 보장 특성을 위반하지 않는다.*

증명. 이전 단계 S_{i-1} 의 잘못된 고장 검출로 인해 고아 상태 ΦS_i 가 발생하고 다음 단계 S_{i+1} 가 고아 상태 ΦS_i 로부터 새로운 이동 에이전트 $\overrightarrow{ma_i}$를 받는다고 가정하자. OTRBA가 1-단계 자유 단일 수행 보장 특성을 따르기 때문에 고아 단계 ΦS_i 가 발생할 수 있다. 그러나 OTRBA에서 고아 단계 ΦS_i 는 결코 계속 메시지를 받을 수 없다. 고아 단계 $[\frac{H_i \bigcirc \Phi S_i}{\overrightarrow{ma_i} \bigcirc p_l}]$ 와 이전 단계 S_{i-1} $[\frac{H_i \bigcirc \Phi S_i}{\overrightarrow{ma_i} \bigcirc p_l}]$ 로부터 동일 이동 에이전트 $\overrightarrow{ma_i}$를 받은 다른 단계가 공존한다 할지라도 다음 단계 S_{i+1}은 이전 단계 S_{i-1}로부터 계속 메시지를 받은 S_k 단계로부터 새로운 이동 에이전트 $\overrightarrow{ma_{i+1}}$를 받게된다. 또한, 고아 단계 ΦS_i 의 결과는 이전 단계 S_{i-1}로부터 제거 메시지 때문에 호스트 H_i에서 폐기된다. 그러므로 비록 고아 단계 ΦS_i 가 발생하더라도 단일 수행 보장 특성에 대한 위배가 전체 이동 에이전트수행 $\sum_{i=0}^{n}[\frac{H_i \bigcirc S_i}{\overrightarrow{ma_i} \bigcirc p_l}]$ 에서 발생하지 않는다.

다음으로 다중지역 이동 에이전트 컴퓨팅 환경에서의 낙관적 결함포용 기법의 정당성을 증명한다. 다중 지역 이동 에이전트 컴퓨팅 환경에서 적용한 낙관적 결함 포용 기법의 정당성은 **Theorem 2**에 의해 증명된다.

Theorem 2 *제안된 기법은 전체 이동 에이전트 연산 수행에 대한 단일 수행 보장 특성을 보장한다.*

증명. 전체 이동 에이전트 연산 수행 단계에서 고아 단계 ΦS_j 와 수행 단계 $[\frac{H_l \bigcirc S_l}{\overrightarrow{ma_i} \bigcirc p_k}]$와 $[\frac{H_j \bigcirc S_j}{\overrightarrow{ma_i} \bigcirc p_m}]$, 그리고 $\overset{\frown}{S_l} \trianglerighteq \overset{\frown}{S_{l+1}}$ 가 있다고 가정하자. 만약 단계 S_j에서 이동 에이전트 $\overrightarrow{ma_i}$의 결과가 호스트 H_j로 반영된다면, 그때 단일 수행 보장 특성은 위배된다. 그러나 제안된 기법에서는 만약 이전의 호스트 H_l로부터 확인 메시지를 받지 않는다면, 호스트 H_j는 이동 에이전트 수행의 결과를 반영시키지 않는다. 따라서 비록 고아 단계 ΦS_j가 전체 이동 에이전트 수행에서 존재할지라도, 고아 단계 ΦS_j의 결과는 호스트 H_j에서 취소되어, 전체 이동 에이전트 연산 수행에 대한 단일 수행 보장 특징은 만족하게 된다.

한편, 본 발명에 대한 성능을 평가하면 다음과 같다. 낙관적 시간적 복제 기법 (OTRBA)의 성능을 평가하기 위해 본 발명에서는 메시지 비용, 이동 에이전트의 이주 비용, 잘못된 결함 검출, 동일 이동 에이전트에 대한 두 번째 이주, 단일 수행 보장 특성을 보장하기 위해 사용되는 메시지들의 비용을 평가한다. 정상적인 상태에서의 메시지 비용은 도 8에 나타나 있고 잘못된 결함 검출로 인해 발생하는 메시지 비용은 도 9에 나타나 있다. 이 평가에서 보여주듯이 메시지 비용은 지역의 수가 증가할 때 높게 증가함을 알 수 있다.

본 발명은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터(정보 처리 기능을 갖는 장치를 모두 포함한다)가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 장치의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등이 있다.

본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

발명의 효과

본 발명에 의한 이동에이전트 결함포용 방법 및 장치와 그를 이용한 다중지역에서의 이동에이전트 결함포용 방법 및 장치에 의하면, one-free 단일 수행 보장 property에 기반하여 결함 포용적 이동 에이전트 수행을 가능하게 한다. 또한 다중 이동 에이전트 컴퓨팅 환경에서 발생하는 확장성 문제를 해결한다.

(57) 청구의 범위

청구항 1.

적어도 둘 이상의 호스트를 구비하는 네트워크에서의 이동 에이전트 결함포용방법에 있어서,

(a) 상기 이동에이전트가 자신의 이주경로를 따라 이주할 때, 이전 단계의 제1호스트가 이동에이전트를 복사한 후 현재 단계의 제2호스트로 이주시키는 과정;

(b) 상기 제2호스트는 상기 제1호스트에게 단일수행 보장특성이 보장되는지 확인하는 과정;

(c) 상기 제1호스트가 현재 단계의 고장을 잘못 검출하여 이동에이전트를 제3호스트로도 상기 이동 에이전트를 이주시켰다면, 상기 제1호스트는 상기 제2호스트 또는 제3호스트 중 하나에게 상기 이동 에이전트 취소 메시지를 보내는 과정; 및

(d) 상기 제1호스트가 이동에이전트를 상기 제2호스트에게 이주시키고 그 외의 다른 호스트에게 이동 에이전트를 이주시키지 않았다면, 상기 제1호스트는 상기 제2호스트에게 이동 에이전트의 계속적인 실행을 지시하는 계속 메시지를 보내는 과정을 포함함을 특징으로 하는 이동 에이전트 결함 포용 방법.

청구항 2.

제1항에 있어서, 상기 (c)과정은

제1호스트로부터 취소 메시지를 받은 호스트는 현재 수행한 이동 에이전트 연산의 결과를 모두 취소하고 상기 이동 에이전트를 제거하는 과정을 더 구비하고,

상기 (d)과정은

제1호스트로부터 계속 메시지를 받으면, 이동 에이전트를 새로운 단계로 이주시키는 과정을 더 구비함을 특징으로 하는 이동 에이전트 결함 포용 방법.

청구항 3.

적어도 둘 이상의 호스트를 구비하는 네트워크에서의 이동 에이전트 결함포용시스템에 있어서,

이동 에이전트가 자신의 이주경로를 따라 이주할 때, 이동에이전트를 복사한 후 다음 단계의 호스트로 이주시키고, 이주된 이동에이전트의 호스트로부터 확인 메시지를 받으면 고장을 잘못 검출하여 이동에이전트를 다른 호스트로 이주시켰으면 취소메시지를 보내고 그렇지 않으면 실행계속을 나타내는 계속메시지를 보내는 제1호스트; 및

상기 제1호스트로부터 이주한 이동에이전트가 수행되며, 수행결과를 반영하기 위해 상기 제1호스트에게 단일수행 보장특성이 보장되는지 확인하고, 확인결과 취소메시지를 받으면 이동에이전트 및 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트의 실행을 계속하게 하는 제2호스트를 포함함을 특징으로 하는 이동 에이전트 결함 포용 시스템.

청구항 4.

네트워크로 연결되는 적어도 두 개의 지역을 포함하며 상기 지역 각각은 적어도 하나의 호스트 및 지역서버를 구비하는, 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 방법에 있어서,

이동 에이전트가 동일 지역 내에 위치한 다른 호스트로 이주하는 지역내 이주과정; 및

이동 에이전트가 다른 지역에 위치한 호스트로 이주하는 지역간 이주과정을 포함함을 특징으로 하고,

상기 지역내 이주과정은

만일 이동 에이전트에 고장이 발생할 경우, 이전 단계의 호스트가 고장난 이동 에이전트를 다른 곳으로 이주시키고,

상기 지역간 이주과정은

이동 에이전트가 다른 지역의 호스트로 이주할 때 호스트가 속하는 지역서버간에도 이동에이전트를 이주하여 이동 에이전트 결함포용을 수행하는, 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 방법.

청구항 5.

제4항에 있어서, 상기 지역 내 이주과정은

(a) 상기 이동에이전트가 자신의 이주경로를 따라 한 단계에서 다음 단계로 이주할 때, 다음 단계의 제2호스트에서 발생할 수 있는 고장에 대비하기 위해 이동 에이전트를 현재 단계의 제1호스트에 복사하는 과정;

(b) 제1호스트에 이동 에이전트가 복사된 후, 이동 에이전트는 이동경로를 통해 다음 단계의 제2호스트로 이주하는 과정;

(c) 이동 에이전트를 수행하고 있는 제2호스트는 이전 단계의 제1호스트에게 단일수행 보장특성이 보장되는지 확인하는 과정;

(d) 상기 제1호스트가 현재 단계의 고장을 잘못 검출하여 이동에이전트를 제2호스트 외의 제3호스트로도 상기 이동 에이전트를 이주시켰다면, 상기 제1호스트는 상기 제2호스트 또는 제3호스트 중 하나에게 상기 이동 에이전트 취소 메시지를 보내는 과정; 및

(e) 상기 제1호스트가 이동에이전트를 상기 제2호스트에게 이주시키고 그 외의 다른 제3호스트에게 이동 에이전트를 이주시키지 않았다면, 상기 제1호스트는 상기 제2호스트에게 이동 에이전트의 계속적인 실행을 지시하는 계속 메시지를 보내는 과정을 구비함을 특징으로 하는 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 방법.

청구항 6.

제5항에 있어서, 상기 (d)과정은

제1호스트로부터 취소 메시지를 받은 호스트는 현재 수행한 이동 에이전트 연산의 결과를 모두 취소하고 상기 이동 에이전트를 제거하는 과정을 더 구비하고,

상기 (e)과정은

제1호스트로부터 계속 메시지를 받으면, 이동 에이전트를 새로운 단계로 이주시키는 과정을 더 구비함을 특징으로 하는 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 방법.

청구항 7.

제4항에 있어서, 상기 지역 간 이주과정은

이동 에이전트가 다른 지역의 제2호스트로 이주하기 전에 상기 제2호스트의 고장을 대비하기 위해 현재 단계의 제1호스트에 상기 이동에이전트를 복사하는 과정;

이동 에이전트를 다른 지역의 제2호스트로 이주시키는 과정;

현재 단계의 제1호스트는 같은 지역의 제1지역서버로 상기 이동 에이전트를 이동시키고, 상기 제1지역서버는 상기 이동 에이전트가 다른 지역으로 이주한 지역의 제2지역서버로 이동에이전트를 이주시키는 과정;

이동 에이전트를 수행한 제2호스트는 상기 제2호스트가 속한 지역의 제2지역서버에게 확인 요청 메시지를 보내는 과정;

제2호스트가 제2지역서버로부터 다른 호스트로 이동 에이전트를 이주시키지 않았다는 확인 메시지를 받으면, 상기 제2호스트는 이동 에이전트 수행 결과를 반영하고, 이동 에이전트를 다음 호스트로 이주시키는 과정; 및

만일 상기 제2지역서버가 현재 수행중인 호스트의 고장을 잘못 검출하여 다른 호스트로 이동 에이전트를 보냈다면 상기 제2지역서버는 제2호스트에게 취소 메시지를 보내는 과정을 구비함을 특징으로 하는 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 방법

청구항 8.

네트워크로 연결되는 적어도 두 개의 지역을 포함하며 상기 지역 각각은 적어도 하나의 호스트 및 지역서버를 구비하는, 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 시스템에 있어서,

이동 에이전트가 자신의 이주경로를 따라 이주할 때, 지역내 이동을 처리하는 지역내 이동처리부 및 지역간 이동을 처리하는 지역간 이동처리부를 구비하는 제1호스트;

이동에이전트의 지역내 이동일 경우, 상기 제1호스트로부터 이주한 이동에이전트가 수행되며, 수행결과를 반영하기 위해 상기 제1호스트에게 단일수행 보장특성이 보장되는지 확인하고, 확인결과 취소메시지를 받으면 이동에이전트 및 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트의 실행을 계속하게 하는 제2호스트;

이동 에이전트의 지역간 이동일 경우, 상기 제1호스트로부터 복사된 이동 에이전트를 저장하고 있으며, 상기 이동에이전트가 상기 제3호스트로 이동할 때 상기 다른 지역의 지역서버(제2지역서버)에 상기 이동에이전트를 이주시키는 제1지역서버;

상기 제1지역서버가 이주시킨 이동에이전트를 저장하고 있으며, 같은 지역의 제3호스트의 확인요청 메시지를 받으면 다른 호스트로 이동 에이전트를 이주시켰는지 여부를 체크하여 이주 시켰으면 제3호스트에게 취소메시지를 보내고 그렇지 않으면 계속메시지를 보내는 제2지역서버; 및

상기 제2지역서버의 지역에 존재하며, 상기 이동 에이전트가 수행되어 결과를 반영하고자 할 때 상기 제2지역서버에게 확인요청 메시지를 보내어 취소메시지를 받으면 이동에이전트와 그 수행결과를 삭제하고, 계속메시지를 받으면 이동에이전트가 계속 실행되게 하는 제3호스트를 포함함을 특징으로 하고,

상기 제1호스트의 지역내 이동처리부는

지역내 이동일 경우, 이동에이전트를 복사한 후 다음 단계의 호스트로 이주시키고 이주된 이동에이전트의 호스트로부터 확인 메시지를 받으면 고장을 잘못 검출하여 이동에이전트를 다른 호스트로 이주시켰으면 취소메시지를 보내고 그렇지 않으면 실행계속을 나타내는 계속메시지를 보내며,

상기 제1호스트의 지역간 이동처리부는

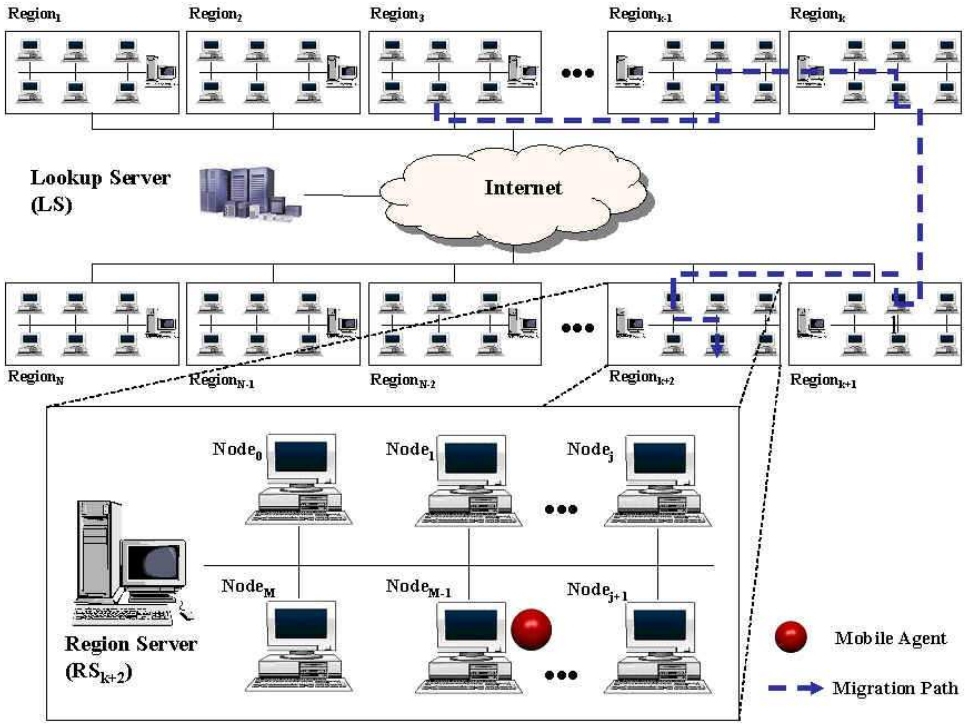
지역간 이동일 경우, 이동 에이전트가 다른 지역의 호스트로 이주할 때, 이동에이전트를 복사한 후 상기 다른 지역의 호스트(제3호스트)로 이주시키고 자기가 속한 지역의 지역서버(제1지역서버)로 상기 이동 에이전트를 이동시키는, 다중 지역 이동 에이전트 컴퓨팅 환경에서의 이동 에이전트 결함 포용 시스템.

청구항 9.

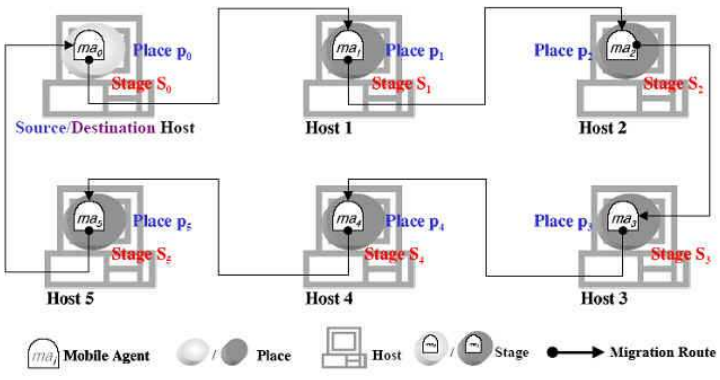
제1항 내지 제6항 중 어느 한 항에 기재된 발명을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

도면

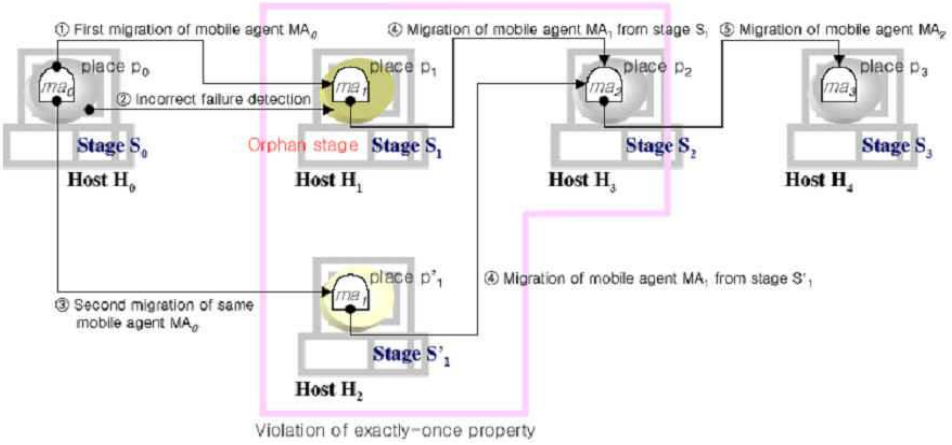
도면1



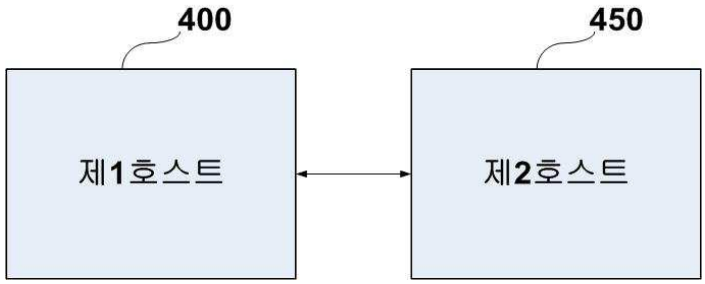
도면2



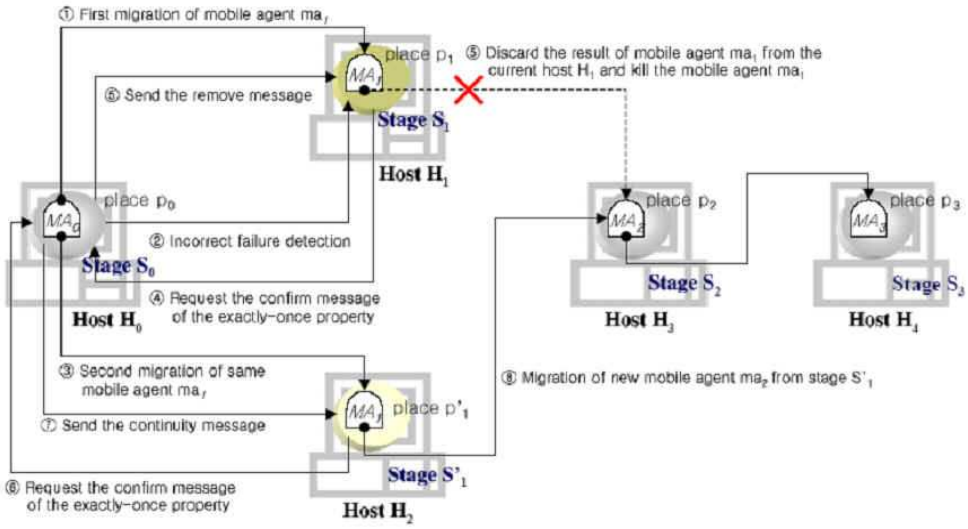
도면3


① First migration of mobile agent MA0
④ Migration of mobile agent MA1 from stage S1
⑤ Migration of mobile agent MA2
place p0
place p1
place p2
place p3
② Incorrect failure detection
Stage S0
Orphan stage
Stage S1
Stage S2
Stage S3
Host H0
Host H1
Host H3
Host H4
place p'1
④ Migration of mobile agent MA1 from stage S'1
③ Second migration of same mobile agent MA0
Stage S'1
Host H2
Violation of exactly-once property

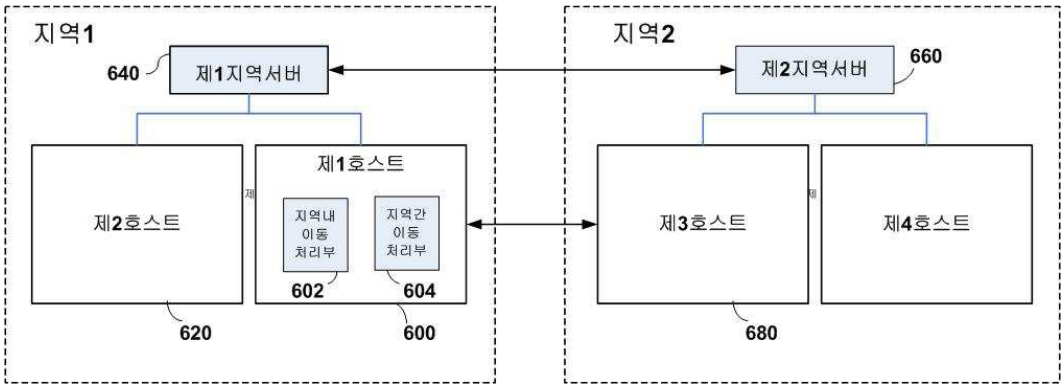
도면4


400
450
제1호스트
제2호스트

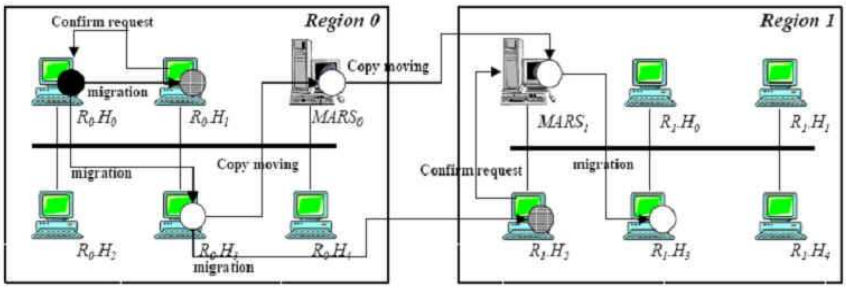
도면5


① First migration of mobile agent ma1
place p1
⑤ Discard the result of mobile agent ma1 from the current host H1 and kill the mobile agent ma1
⑥ Send the remove message
Stage S1
Host H1
place p0
place p2
place p3
② Incorrect failure detection
Stage S0
Stage S2
Stage S3
④ Request the confirm message of the exactly-once property
Host H0
Host H3
Host H4
③ Second migration of same mobile agent ma1
⑧ Migration of new mobile agent ma2 from stage S'1
⑦ Send the continuity message
place p'1
Stage S'1
⑧ Request the confirm message of the exactly-once property
Host H2

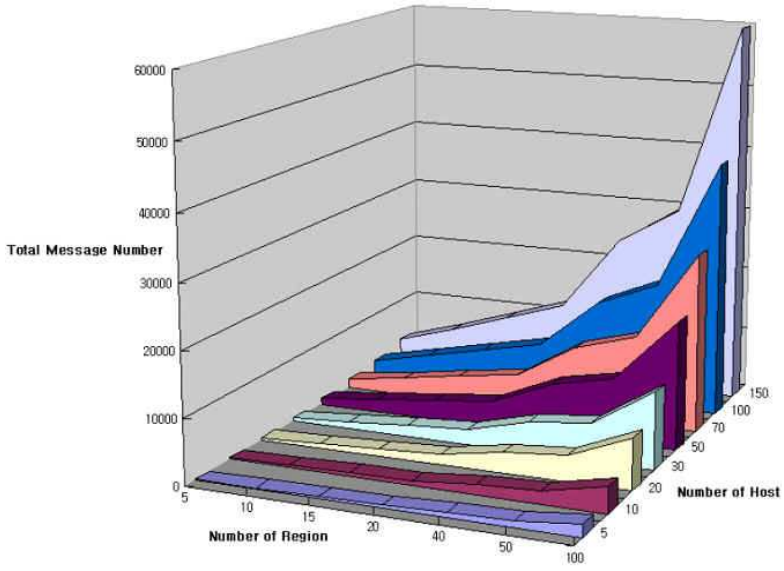
도면6


지역1
640
제1지역서버
지역2
660
제2지역서버
제1호스트
제2호스트
지역내 이동 처리부
지역간 이동 처리부
602
604
620
600
제3호스트
제4호스트
680

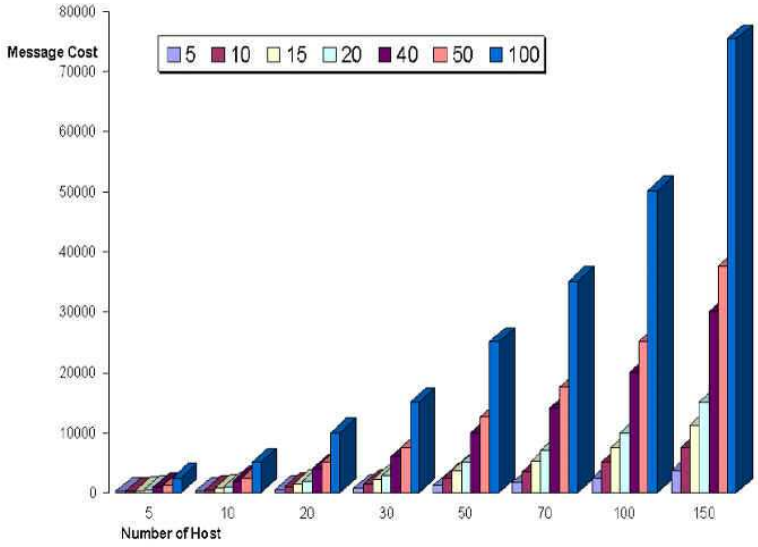
도면7


Confirm request
Region 0
Region 1
migration
Copy moving
R_0H_0
R_0H_1
$MARS_0$
$MARS_1$
R_1H_0
R_1H_1
R_0H_2
R_0H_3
R_0H_4
R_1H_2
R_1H_3
R_1H_4

도면8


Total Message Number
60000
50000
40000
30000
20000
10000
0
Number of Region
5
10
15
20
40
50
100
Number of Host
150
100
70
50
30
20
10
5

도면9


Message Cost
80000
70000
60000
50000
40000
30000
20000
10000
0
5
10
15
20
40
50
100
5
10
20
30
50
70
100
150
Number of Host