

이동 에이전트의 하이브리드 이주 기법 설계

이연식, 김광종
군산대학교

yslee@kunsan.ac.kr, kkjkim@kunsan.ac.kr

Hybrid Migration Method of Mobile Agent

Yon-sik Lee, Kwangjong Kim
Kunsan University

요 약

유비쿼터스 환경에서 이동 에이전트는 데이터 전송, 네트워크 지연 및 대역폭 오버헤드와 같은 네트워크 문제의 해결 대안으로 사용되어 왔으나 동적 환경변화 및 이주비용에 대한 특징을 고려하는 문제와 정적으로 결정된 순서로 노드를 이주하는 순차 구조로서 네트워크 부하, 노드 장애 및 서비스 부재와 같은 문제들은 여전히 이동 에이전트 성능에 관련된 밀접한 문제로 남아있다. 따라서 본 논문에서는 이러한 이동 에이전트의 성능에 주요한 영향을 미치는 문제를 하이브리드 이주 기법을 통해 해결하고자 하며 이를 위한 구조를 설계한다. 제안 기법은 순차와 병렬 이주 기법을 통합한 것이며 노드의 이주 스케줄 정보는 네이밍 에이전트의 메타데이터에 의해 결정된다. 이를 위해 이주 스케줄 정보의 생성 및 갱신에 대한 내용을 기술하며 이동 에이전트 복제를 통한 병렬 이주 기법과 선택적 이주 노드를 결정하는 이주 기법을 통합한 하이브리드 이주 기법을 제시한다.

1. 서 론

이동 에이전트 기술은 분산 응용에 있어서 네트워크 부하 및 대기시간을 감소시킬 수 있는 기술 중 하나이며 네트워크의 적응성이 좋아 네트워크 오류 시 다른 시스템에 비해 신뢰도가 높기 때문에 분산 환경에 적용이 가능한 기술이라고 할 수 있다 그러나 데이터의 크기 및 네트워크 상태 이주 노드 수의 증가에 따른 이주 방법은 이주비용을 증가시키므로 이동 에이전트 성능에 영향을 주는 요인으로 작용한다[1,2,3]. 또한, 동적 환경변화에 대한 특징을 고려하는 문제와 정적으로 결정된 순서로 노드를 이주하는 순차적 구조로서 네트워크 부하, 노드 장애 및 서비스 부재와 같은 문제들은 여전히 이동 에이전트 성능에 관련된 밀접한 문제로 남아있다[4,5,6].

그러므로 본 논문에서는 이동 에이전트의 성능에 주요한 영향을 미치는 문제와 동적 환경 변화에 능동적으로 대처하는 하이브리드 이주 기법을 제안하며 이를 위한 구조를 설계한다. 제안 기법은 순차와 병렬 이주 기법을 통합한 것이며 노드의 이주 스케줄 정보는 네이밍 에이전트의 메타데이터에 의해 결정된다 이를 위해 이주 스케줄의 생성 및 갱신하는 방법을 기술하며 이동 에이전트 복제를 통한 병렬 이주 기법과 선택적 이주 노드를 결정하는 이주 기법을 통합한 하이브리드 이주 기법을 제시한다. 제안 기법의 기반 구조는 멀티 에이전트 구조이며 서로 각기 다른 에이전트들 간에 상호협력을 지원하고 이동할 목적지의 위치를 관리하고 이주 우선순위 및 이동 에이전트의 복제 수 결정을 위해 분산된 노드의 서버 객체 정보들을 등록하고 관리하는 네이밍 에이전트

를 통해 각 노드의 정보나 메시지를 수신할 수 있도록 한다. 그리고 동적 환경 변화 발생 시 에이전트의 목적 노드의 조정을 지원할 수 있도록 노드 간 통신 효율성 향상을 위한 방법을 설계하여 환경 변화에 적합한 이주를 지원하도록 한다.

본 논문의 구성은 2장에서 관련 연구로서 기존의 에이전트 이주 기법에 대해 설명하고 3장에서는 제안 기법을 지원하는 시스템 구조와 각 에이전트의 상호 협력 과정 및 이주 정보에 대한 생성 및 갱신 방법과 동적 환경 변화에 따른 목적 노드의 조정에 관한 방법을 기술한다 그리고 4장에서 이동 에이전트 복제를 통한 병렬 이주 기법과 선택적 이주 노드를 결정하는 이주 기법을 통합한 하이브리드 이주 기법을 제시한다 마지막 5장에서는 결론 및 향후 연구방향에 대해 기술한다

2. 관련연구

이동 에이전트는 분산 환경의 새로운 패러다임으로 등장하면서 사용자의 요청을 대신하여 수행할 에이전트의 실행 환경을 제공하기 위한 많은 에이전트 시스템들이 개발되었다. 기존 개발된 에이전트 시스템은 표준 지원 여부에 따라 FIPA 계열과 OMG의 MASIF 계열로 구분되고 있다[5,6,7]. 하지만 현재 이동 에이전트 시스템은 Java 언어를 기반으로 하는 추세이며 이러한 이동 에이전트 시스템을 기반으로 이동 에이전트의 이주 기법은 많은 연구가 진행 중이다[10,11,12,13].

Ajanta[8]는 자바 직렬화, 반사(Reflection), RMI 기능을 이용하여 이동 에이전트를 구현하였다 에이전트의 이주는 네트워크를 통해 이동할 에이전트를 자바 직렬화 기법을 이용하여 요구한 클래스만을 이동 시키는 약 이주 기법을 사용한다 또한, 프록시 중재를 통해 시스템

자원에 접근하는 이주 기법을 제공한다 따라서 Ajanta 에서 제공하는 클래스 로딩 기법은 수행에 필요한 코드 만을 이동하는 방법을 사용하여 전송할 데이터양을 최소화함으로써 네트워크 부하를 감소시킨다 또한, 에이전트 수행에 적합한 이주 패턴을 선택하고 선택된 이주 패턴에 따라 에이전트를 수행함으로써 이주 성능을 향상시킬 수 있다. 그러나 요구한 클래스만을 이동시키는 방법을 사용함으로써 이주 데이터의 크기를 고려하지 못하는 문제점과 결합에 대한 처리를 하지 못한다는 문제점이 있다 JAMES[9]는 고성능의 네트워크 관리 및 통신에 있어 성능을 향상시키고자 개발된 이동 에이전트 시스템이다 이주 기법은 클라이언트가 서비스를 요청하게 되면 중앙 노드에서 이동 에이전트가 생성되고 이주 경로를 생성한다. 이주 정보에 따라 필요한 코드를 각 에이전트 시스템에 메시지 형태로 전송하고 메시지를 수신한 에이전트 시스템에서는 자신이 필요한 코드를 자신의 캐시메모리나 이전 에이전트 시스템의 캐시메모리 또는 자신의 보조기억장치에서 찾는다. 만약 코드가 없는 경우에는 서버에 필요한 코드를 요청하여 미리 코드를 받는다 이러한 프리패칭 기법을 이용한 JAMES 시스템은 이주할 때 발생할 수 있는 네트워크 트래픽을 개선하였다 그러나 동적인 환경변화 즉 목적지 시스템의 자원 및 네트워크의 상태를 고려하지 못한다는 점과 이주 과정에서 발생하는 결점에 대한 처리를 제공하지 못한다는 것이다 이와 같이 기존 시스템들은 목적지 시스템의 자원 및 네트워크의 상태 변화에 적절히 대응하지 못할 뿐만 아니라 순차적 이주 수행 형태의 구조로써 이주할 데이터의 크기 및 결합에 대한 처리를 고려하지 못함으로써 에이전트가 무한 대기나 고아 상태가 되어 과피 및 쓰레기로 처리될 수 있다.

3. 하이브리드 이주 기법 설계

3.1 지원 시스템 구조

제안 이주 기법을 지원하기 위한 시스템 구조는 멀티 에이전트의 수행 환경으로 구성된다 이러한 구조는 각 에이전트간의 협력 관계를 유지하며 동적 환경 변화에 능동적으로 대처하기 위해 서로 다른 에이전트들의 위치 유지 및 분산 환경에 존재하는 객체에 대한 투명성을 보장하기 위함이다. 이는 분산 객체의 통합을 유도하여 이주의 신뢰성을 제공하며 에이전트의 성능을 향상시키는 효율적 이주를 지원한다 그림 1은 이를 지원하는 시스템의 구조이다.

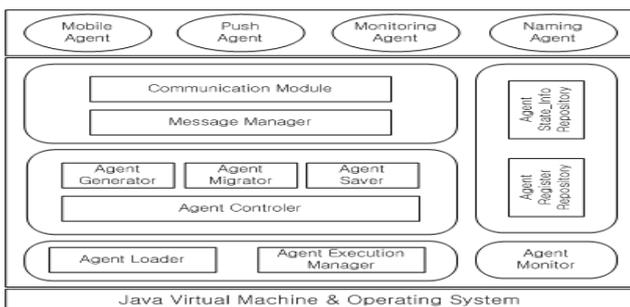


그림 1. 시스템 구조

3.2 이주 패턴과 상호협력

이주 패턴은 각 에이전트가 네이밍 에이전트를 기반으로 협력 관계가 유지되며 메타데이터의 정보를 통해 노드 이주 경로와 에이전트 복제 그리고 이주해야 할 노드 수를 결정하는 정보를 제공한다 그림 2는 이주 패턴에 관한 그림이다.

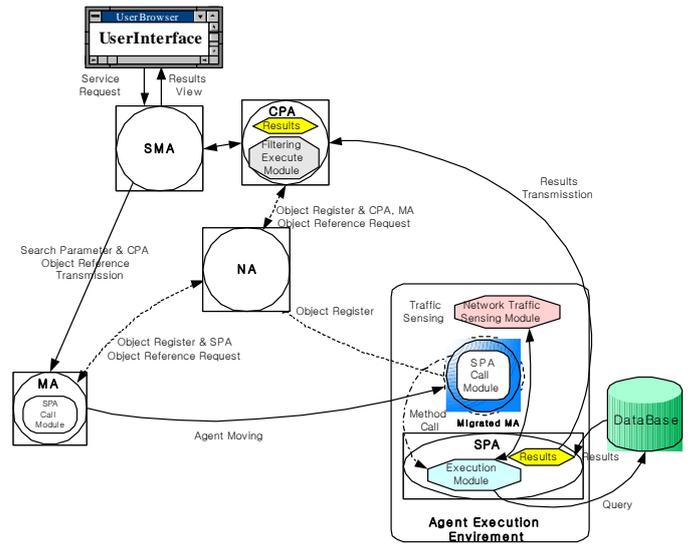


그림 2. 이주 패턴

그림 2에서와 같이 각 에이전트에서 이동 에이전트(MA)는 이주 스케줄링 정보를 통해 경로에 존재하는 서버로의 이주를 수행하여 사용자의 요구 작업을 수행한다. 푸시 에이전트는 클라이언트 측에 위치한 클라이언트 푸시 에이전트(CPA)와 서버 측에 위치한 서버 푸시 에이전트(SPA)로 구성되며 클라이언트 푸시 에이전트는 네이밍 에이전트로부터 분산된 서버의 이주 정보를 요청하여 반환된 정보를 이동 에이전트 생성복제에 필요한 일부 정보 제공 및 이주 스케줄링 결정에 사용한다 서버 푸시 에이전트는 이동 에이전트의 자원 접근 요구를 수용하며 실행 후 정보를 요청한 위치로 정보를 반환한다. 이러한 요청 결과의 반환은 서버 푸시 에이전트에 의해 생성된 임시 저장소인 해시테이블에 저장되며 결과에 대한 클라이언트 푸시 에이전트로의 전송 시기는 서버 푸시 에이전트의 판단에 의해 수행된다 네이밍 에이전트(NA)는 각 쓰레드를 생성하여 분산된 서버로부터 얻은 정보들을 관리하기 위한 테이블 형태의 네임 스페이스를 갖는다. 네임 스페이스는 각 분산된 서버로부터 얻은 서버 푸시 에이전트 클라이언트 푸시 에이전트 이동 에이전트의 정보를 중간 형태의 메타데이터로 바꾸어 보관한다. 보관된 이러한 각 에이전트에 대한 정보는 메타데이터 참조를 통해 이루어진다 그림 3은 이러한 각 에이전트들 간의 상호 협력에 대한 그림이다

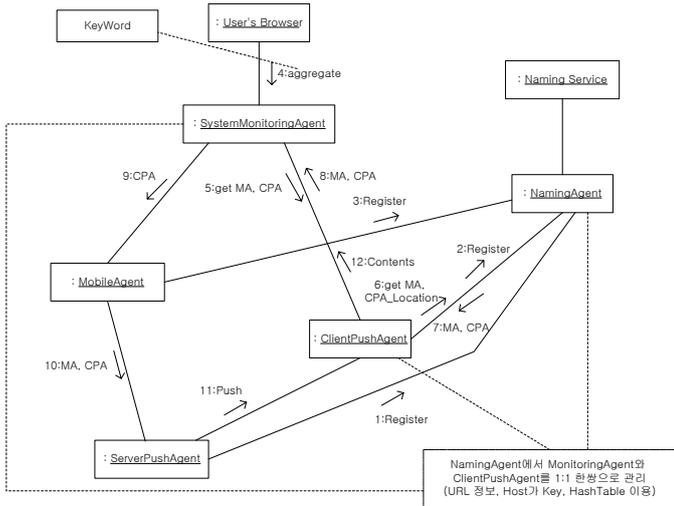


그림 3. 에이전트간 상호협력

3.3 이주 정보 생성과 갱신

각 에이전트의 위치 정보는 네이밍 에이전트가 관리함으로 에이전트 중 제일 먼저 실행되어 다른 에이전트의 요구에 따라 에이전트 이름을 등록하고 관계한 메타데이터의 각 필드를 초기화 한다 이 때 네이밍 에이전트는 같은 이름으로의 에이전트 등록을 방지하고 중복을 제거하여 분산된 서버의 위치에 대한 신뢰성을 제공해야 한다. 따라서 에이전트의 이름 충돌 시 예외 기능을 이용하여 기존 에이전트의 삭제 및 재등록 과정을 수행한다 동시에 발생하는 에이전트의 등록 요구에 대해 네이밍 에이전트는 쓰레드별로 생성한다 생성된 쓰레드들은 상호 동기화 시켜 잘못된 내용의 등록을 방지한다 등록되는 에이전트가 서버 푸시 에이전트 일 때는 키워드별 메타데이터에 해당 에이전트 객체를 등록하고 등록된 서버 푸시 에이전트에 대한 키워드별 메타데이터의 각 필드는 초기화된다. 서버 푸시 에이전트에 대한 키워드별 메타데이터 생성은 서버 푸시 에이전트가 실행모듈을 가지고 있으며 그 실행 결과에 따라 이동 에이전트의 키워드에 따른 노드 이주에 중요한 정보를 제공하기 때문이다 모든 에이전트는 고유한 이름을 갖고 이름별 메타데이터에 등록된 후 각 에이전트의 상호 접근을 제공하게 된다 갱신은 서버 측으로부터 전달된 에이전트 객체ID를 메타데이터의 정보와 비교하여 최초 등록된 에이전트 객체ID를 통해 전달된 정보를 갱신하기 전 서버 푸시 에이전트에 의해 해당 작업을 완료 후 갱신을 수행한다 그러나 에이전트 객체 ID가 존재하지 않는다면 새로운 ID를 메타데이터에 관련 정보와 함께 등록한다

3.4 목적 노드의 재조정

다중 복제 이주의 경우에는 재조정 없이 타임스탬프 방법으로 일정시간 응답이 없으면 복제된 에이전트를 재전송하는 방법으로 이루어지며 선택적 순차 이주시 노드의 장애로 인해 이미 이주하여 실행중인 에이전트가 무한 대기 상태에 빠지거나 쓰레기로 처리되는 경우에는 노드 재조정 방법을 사용하여 해결할 수 있다 노드 재조정 방법은 장애가 발생한 노드의 이전 이주 노드에서

이주 경로에 있는 다음 이주 노드로 재조정하여 에이전트를 이주시키는 방식이다 그림 4는 노드 재조정 방법에 관한 그림이다

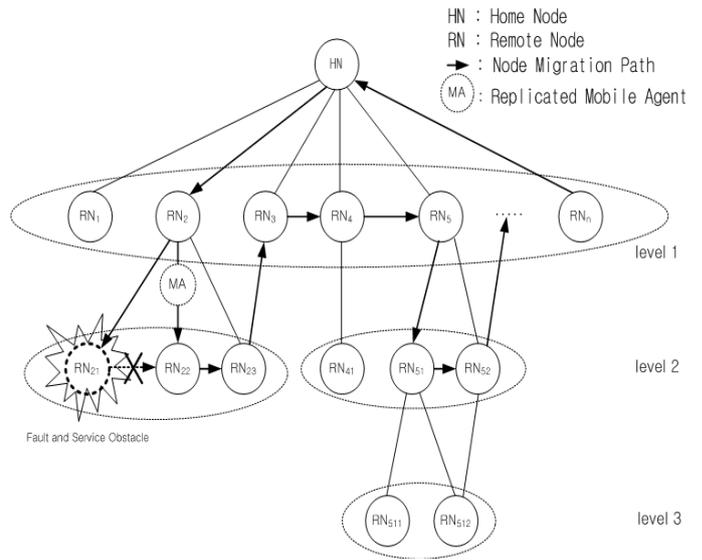


그림 4. 목적 노드 재조정

그림 4에서와 같이 노드 재조정 방식의 수행과정은 먼저, HN(Home Node) 노드로부터 이주 경로 테이블의 순서와 이주 노드 수 결정에 따라 RN₁(Remote Node), RN₂ 노드로 이주하여 작업을 처리한 후 이주 순서대로 RN₂₁ 노드로 이주를 수행한다 이 때, RN₂₁ 노드로 이주하는 이동 에이전트는 자신을 복제하여 RN₂ 노드에 남겨둔다. RN₂ 노드에 복제되어 남겨진 이동 에이전트는 RN₂₁ 노드로부터 복제 에이전트 삭제 신호가 수신될 때까지 일정한 시간동안 대기한다 만약, RN₂₁ 노드에 장애가 발생하여 RN₂ 노드로부터 이주한 에이전트가 고아 상태에 빠지거나 파괴되었을 경우 RN₂ 노드의 복제 에이전트는 정해진 대기 시간이 지나면 RN₂₁ 노드에서 장애가 발생했다고 판단하고 우선순위에 따라 RN₂₁ 노드가 아닌 다음의 노드인 RN₂₂ 노드로 에이전트를 재 이주시킨다. RN₂₂ 노드에서 장애가 발생하지 않고 이주된 이동 에이전트가 주어진 작업을 완전하게 처리하면 RN₂₂ 노드의 이동 에이전트는 다음 노드로 이주하기 전 RN₂ 노드에 남겨놓은 복제 에이전트를 삭제하도록 신호를 보낸다. RN₂ 노드는 RN₂₂ 노드로부터 복제 에이전트의 삭제 신호를 수신하고 남겨진 에이전트 객체를 삭제한다

이와 같이 노드 재조정 방법은 이주 노드들의 장애에 능동적으로 대처함으로써 보다 신뢰할 수 있도록 이동 에이전트의 이주를 보장한다 따라서 현재의 노드에 이동 에이전트를 복제하여 남겨두고 다음 노드로 이동 에이전트를 이주시킨다 이때, 이주한 노드의 이동 에이전트 시스템은 동작 대기상태로 들어가게 되는데 노드의 장애 발생으로 인해 이동 에이전트가 무한대기 상태에 빠지거나 파괴되어 더 이상의 작업 수행이 불가능한 경우를 대비하여 현재의 시간 측정 한 후 이를 저장한다 만약, 노드 장애로 인해 이주한 노드로부터 복제 에이전트 삭제 신호가 반환되지 않고 있을 때 이주 전 노드는 해당 노드에서 장애가 발생했다고 판단하고 일정한 시간

이 지난 후 다시 현재의 시간을 측정한다 그 다음 과정은 측정된 시간이 현재시간과 이전시간의 차이를 구했을 때 차이 값이 설정된 값 보다 크거나 같으면 이주 전 노드의 다음 우선순위를 갖는 노드로 에이전트를 이주시킨다. 한편, 이주한 노드에서 장애가 발생하지 않을 경우, 이동 에이전트는 작업을 실행하여 완료한 후 다른 노드로 이주하기 전에 이주 전 노드에게 복제 에이전트를 삭제하라는 신호를 전송한다 삭제 신호를 받은 노드는 복제해놓은 이동 에이전트를 삭제한다

4. 하이브리드 이주 기법 구조

4.1 하이브리드 이주 기법의 구조

아래 그림 5는 선택과 병렬[12]의 두 이주 방식을 통합한 하이브리드 이주 기법 구조이다

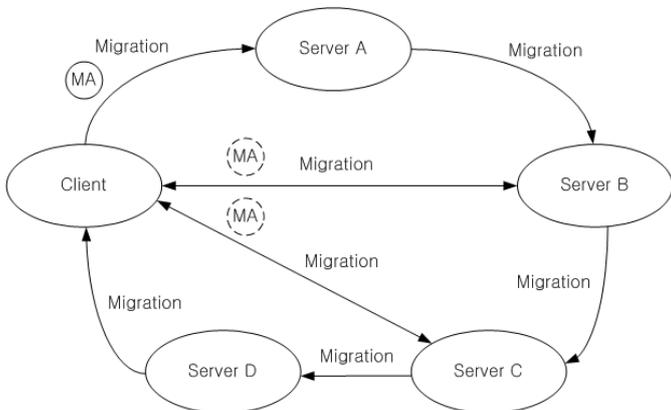


그림 5. 하이브리드 이주 기법 구조

그림 5와 같이 제안 기법은 클라이언트에서 순회 노드 수가 결정되어지면 스케줄링된 노드에 선택과 병렬 방법으로 이주를 수행한다 선택적으로 이주 할 경우 순회 노드 수의 결정에 따라 Migration Path = Server(A, B, C, D) or (A, B, C) or (A, B)와 같이 이주 경로를 선택할 수 있고 병렬로 이주를 수행할 경우 Migration Path = Server(B, C)에 동시이주를 수행한다 그러나 이러한 방식 중 선택적 방식은 원본 에이전트와 각 해당 노드에 하나의 복제본을 생성하여 이주를 수행하는 반면 병렬 방식은 순회 노드 수의 결정에 따라 해당 노드의 수만큼 이주할 에이전트를 복제한 후 복제된 에이전트가 이주를 수행한다 이와 같이 하이브리드 기법은 두 이주 방식이 통합된 새로운 이주 기법이며 이동 에이전트의 이주 문제를 최소화 할 수 있다
따라서 이러한 이주를 위한 순회 노드 수 결정 방법은 얻어진 각 노드의 적중 문건 수(Hit_Count)와 적중률(Hit_Ratio)을 기준으로 정보가득률에 따른 임계값을 설정하여 순회 노드의 수를 제한한다 그러나 제안 이주 기법의 수행을 위해서는 에이전트 복제와 선택적 순차 이주를 위해 순회 노드 수의 결정이 이루어져야 한다 먼저, 에이전트 복제 방법은 사용자의 키워드를 입력 받아 네이밍 에이전트의 메타 테이블을 검색한 후 적합한 노드 위치 및 키워드 관련 정보를 수집하여 이를 기반으로 원본 에이전트 객체를 생성한다 원본 에이전트 객체

가 생성되면 객체 참조자 리스트 정보들에 의한 순회 노드 수 결정을 참고하여 에이전트 복제 수를 결정한다 복제된 에이전트 객체들의 투명성과 일관성을 검사하여 문제 발생 시, 에이전트 객체 복제를 반복 실행하며 에이전트 객체가 투명성과 일관성 검사를 통과하게 되면 에이전트 객체는 각 노드로의 이주를 대기하게 된다 따라서 순회 노드 수 결정에 관한 방법은 각 노드의 적중 문건 수와 적중률을 기준으로 정렬된 객체 정보에서 정보 가득률에 따른 임계값을 설정하여 이주 노드의 수를 결정한다.

이주 노드 수 설정은 정렬된 객체 정보에서 전체 적중 문건 수에 대한 비율로 정보 가득률의 임계값 설정하며 설정된 임계값에 따라 이주 노드 수가 결정된다 하지만 임계값의 설정은 사용자에게 의해 조정이 가능하다고와 같이 에이전트 복제는 사용자가 요구하는 작업을 수행함에 있어 에이전트의 가용성 및 이주비용과 에이전트 크기 증가에 따른 문제를 최소화할 수 있다 그러나 노드 수의 증가에 따른 복제에 대한 비용과 부하는 증가하기 때문에 복제에 대한 제한이 필요하다 그러므로 순회 노드 수의 결정에 따른 복제를 수행함으로써 복제에 따른 비용과 부하를 감소시킬 수 있다 또한, 선택적 순차 이주일 경우에도 전체 노드의 순회를 수행하지 않고 순회 노드 수 결정에 따른 방법으로 원하는 노드만을 이주 스케줄링 함으로써 전체 노드에 따른 이주비용을 감소시킬 수 있다. 그림 6은 이러한 순회 노드 수 결정에 관한 그림이다.

내림차순 정렬 후

Object Reference	Search keyword	Hit Count	Hit Ratio
...	적합키워드	134	75%
...	적합키워드	120	55%
...	적합키워드	101	65%
...	적합키워드	98	81%
...	적합키워드	93	82%
...	적합키워드	91	58%
...	적합키워드	88	73%
...	적합키워드	85	75%
...	적합키워드	78	90%
...	적합키워드	65	95%

정보가득률에 따른 임계값 설정 후, 이주 노드 수 결정

Server Name	Server Host	Port Number	Object Key	...
-------------	-------------	-------------	------------	-----

그림 6. 순회 노드 수 결정

그림 6과 같이 이주 노드 수의 설정은 전체 953개의 적중 문건에서 임계값을 60%의 비율로 설정했다면 $953 * 0.6(60\%) = 572$ 개의 적중 문건 수를 얻을 수 있다 이때 5개 노드의 적중 문건 수의 합은 546이고 6개 노드의 적중 문건 수의 합은 643이 된다. 그러므로 여기에서 임계값을 적용한 결과를 적용하여 이주 노드 수를 결정할 때, 임계값이 적용된 결과 문서 수와의 차에 대한 절대값을 구하여 값이 가장 작은 노드 수를 순회 대상 노드 수로 결정한다. 그림 6에서 순회 노드가 5개일 경우 차이의 절대값은 26이고 6개일 경우 71이므로 순회 대상 노드 수는 5개이다. 그러므로 차가 적은 노드의 수를 선택하면 전체 10개 노드 중 5개 노드가 최종적으로 이주할 노드 수로 결정된다

그러나 만약에 결과 값에 의한 적중 문건 수의 차가 동일하거나 차가 큰 폭이 아닐 때는 하나 작은 노드 수를 선택한다. 이러한 이주 노드 설정 과정은 먼저 사용자의 키워드 입력을 통해 네이밍 에이전트를 호출할 메시지 객체를 생성하고 생성된 메시지 객체는 네이밍 에이전트 구현부의 관리자를 호출하게 된다 호출된 관리자는 네이밍 에이전트 스레드 생성 명령에 의해 해당 네임 스페이스의 메타데이터를 검색하여 해당 객체 참조자 리스트 정보를 획득한다. 이렇게 획득된 객체 참조자 리스트 정보는 정보 가득률에 따른 임계값 설정에 따라 이주 노드 수를 결정하게 된다. 이러한 순회 노드의 제한은 유효정보에 대한 최소의 이주비용으로 유효정보 확보율을 높일 수 있으므로 사용자가 원하는 유효정보에 대한 결과반환 시간과 이주비용을 감소시킬 수 있다

4.2 제안 이주 수행 방법

제안 이주 기법의 수행 방법은 순회 노드 수 결정에 의해 수행된다. 먼저, 선택적 이주와 병렬 이주의 주요한 특징은 사용자의 동일한 요청 작업에 대해 전체 노드를 순회 하지 않고 결정된 노드의 수만큼만 이주를 수행함으로써 이루어진다. 그림 7은 제안 기법의 이주 수행 방법을 나타낸 그림이다

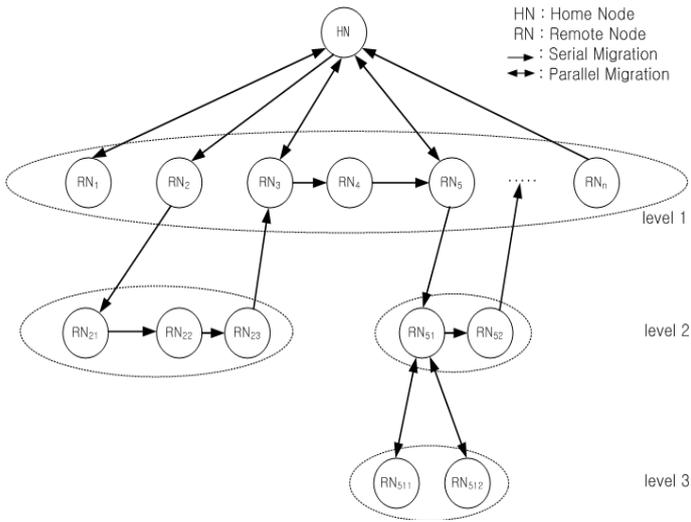


그림 7. 이주 수행

그림 7과 같이 이주 수행에 있어서 에이전트 복제에 의한 병렬 이주 수행은 네이밍 에이전트로부터 전달된 객체 참조자 리스트 정보를 기반으로 순회 노드 수가 결정되면, 원본 에이전트 객체를 생성하며 원본과 복제에 대한 식별값을 True로 설정한다. 이렇게 원본 에이전트 객체가 생성되면 원본 에이전트 객체를 순회 노드 수만큼 복제를 실행한다. 그러므로 그림 7에서와 같이 병렬 이주를 위한 각 노드는 레벨 1, 3에서의 {RN₁, RN₃, RN₅, RN₅₁₁, RN₅₁₂}와 같다.

선택적 이주 수행은 각 노드가 보유하고 있는 전체 문건에서 적중 문건 수, 적중률을 기준으로 내림차순으로 정렬된 객체 참조자 리스트를 반환 받으며 이에 따른 정보 가득률의 임계값을 설정하여 노드의 수를 제한하며 임계

값에 따라 이주 노드 수가 결정되면 원본 에이전트 객체를 생성하며 에이전트 저장소에 저장한다 저장된 에이전트는 이주 노드의 결점이나 장애시 에이전트 객체가 파괴되어 더 이상 이주를 수행 할 수 없을 때 이주 시마다 각 노드에 에이전트를 복제하여 이주를 수행한다 따라서 선택적으로 HN에서 각 노드로 이주를 수행함에 있어 {RN₂, RN₂₁, RN₂₂, RN₂₃, RN₃, RN₄, RN₅, RN₅₁, RN₅₂...RN_n}와 같이 이주를 수행한다

그러나 병렬 이주 수행은 노드 수의 증가에 따른 에이전트 복제에 대한 부하와 비용 문제를 수반하기 때문에 이러한 문제 해결이 요구되며 선택적 이주는 검색 대상이 되는 문건 수, 적중률, 노드 처리 시간 및 네트워크 지연은 노드 이주에 영향을 준다 이러한 정보들은 적중 문건의 수가 상대적으로 낮은 노드에 대한 이주가 더 효율적일 수 있는 경우를 발생시키므로 모든 노드 이주가 이루어진다면 이주비용은 증가하게 된다 이는 각 에이전트가 네이밍 에이전트에 접근하고 갱신하는 시간이 발생하기 때문이다. 그러므로 반환되는 적중 문건의 수가 많다고 하더라도 이주는 비효율적일 수 있으며 높은 응답 시간을 야기하여 성능과 신뢰성을 감소시키는 요인이 된다.

그러므로 제안 이주 기법은 병렬 이주 수행에서 발생하는 노드 수 증가에 따른 오버헤드를 선택적 이주의 수를 늘리고 복제에 의한 이주 노드 수를 줄임으로써 복제의 부하와 비용을 절감할 수 있고 선택적 이주 수행 또한 병렬과 선택 이주의 비율을 적절히 조정함으로써 접근 및 갱신 시간에 따른 문제를 해결할 수 있다 이와 같이 각 이주 수행 방법의 비율을 사용자의 결정에 따라 통합하여 수행함으로써 노드 이주 수행의 효율성과 신뢰성을 확보할 수 있다.

5. 결론 및 향후 연구과제

본 논문에서는 이동 에이전트의 하이브리드 이주 기법을 설계하였다. 이를 위해 에이전트 등록에 의한 이주 정보의 생성 및 갱신 각 에이전트간의 상호협력 과정 동적 환경 변화에 따른 목적 노드 조정 및 제안 기법의 이주 구조와 수행과정에 관한 방법을 제시하였다

이에 적중률에 대한 적중 문건 수에 따라 이동 에이전트의 순회 노드 수를 결정할 때 전체 노드를 대상으로 정보 가득률에 대한 임계값을 설정하고 제안 이주를 수행하여 사용자가 요구하는 유효정보에 대한 확보율을 높임으로써 검색의 신뢰성 및 이주비용이 향상 시킬 수 있으며 기존 이주 기법에 따른 문제를 최소화 할 수 있다 그러나 검색 대상이 되는 문건 수 적중률, 노드 처리 시간 및 네트워크 지연은 노드 이주에 영향을 주며 모든 노드 순회가 이루어진다면 이주비용은 키워드에 따른 메타데이터를 적용하지 않은 경우가 더 효율적일 것이다 이는 이동 에이전트가 순회 노드 수를 결정하여 노드 이주를 할 경우 각 에이전트가 네이밍 에이전트의 메타데이터를 접근하고 갱신하는 시간이 발생하기 때문이다 그러므로 해당 노드의 적중 문건의 수가 많다고 하더라도 접근 및 갱신의 시간이 증가한다면 비효율적일 수 있으며 응답 시간의 증가로 인하여 제안 이주 기법의 성능

을 감소시키는 요인이 된다 향후 연구 과제로는 반환되는 적중 문건의 수를 유지하면서 로드 이주 정책의 신뢰성을 보장할 수 있도록 본 연구를 보완해야 하며 제안 이주 기법에 대한 연구와 평가가 계속되어야 한다

계", 한국컴퓨터정보학회논문지 제10권, 제6호, March, pp.153-166, 2006.

참고문헌

- [1] Tino, S., Peter, B., Ryszard, K., "Towards Autonomous Mobile Agents with Emergent Migration Behaviour", AAMAS'06, pp.585-592, 2006.
- [2] T. Schlegel, P. Braun and R. Kowalczyk, "Towards Autonomous Mobile Agents with Emergent Migration Behaviour", AAMAS 06', May 8 - 12, Hakodate, Hokkaido, Japan, pp.585-592, 2006.
- [3] Y. Berbers, B. D. Decker and W. Joosen, "Infrastructure for mobile agents", ACM AGENTS 04', pp.173-180, 2004.
- [4] P. Metes, "A Pattern based development tool for mobile agents", SIGCSE 05', St.Louis, Missouri, USA, pp.72-75, 2005.
- [5] Haizheng, Z., Bruce, C.W., Brian, L., Victor, L., "A Multi-Agent Approach for Peer-to-Peer Based Information Retrieval System", AGENTS, International Conference on Autonomous Agents, pp.456-463, 2004.
- [6] T. Marian, B. Dumitriu, M. Dinsoreanu and I. Salomie, "A framework of reusable structures for mobile agent development", Mobile Computing and Communications Review, Vol.10, No.2, pp.112-116, 2004.
- [7] P. Metes, "A Pattern based development tool for mobile agents", SIGCSE 05', St.Louis, Missouri, USA, pp.72-75, 2005.
- [8] A. R. Tripathi, N. M. Karnik, T. Ahmed, R. D. Singh, A. Prakash, V. Kakani, M. K. Vora and M. Pathak, "Design of the Ajanta Systems for Mobile Agent Programming", Journal of Systems and Software, Vol.62, No.2, pp.123-140, 2002.
- [9] L. M. Silva and P. Simoes, "JAMES : A platform of Mobile Agents for the Management of Telecommunication Networks", The International Workshop on Intelligent Agents for Telecommunication Application, pp.76-95, 1999.
- [10] Todd, W., "Naming Services in Multi-Agent Systems: A Design for Agent-based White Pages", AAMAS'04, pp.1476-1477, 2004.
- [11] Steven, P.F., Martin, L.G., Reed, L., "Agent behavior architecture a MAS framework comparison", AGENTS, International Conference on Autonomous Agents, pp. 86-87, 2002.
- [12] 김광중, 김영자, 고현, 이연식, "네트워크 소요시간 최소화를 위한 이동 에이전트의 멀티캐스트 이주 모델 구현", 한국정보처리학회논문지(D) 제12-D권 제2호, April, 2005.
- [13] 김광중, 김영자, 고현, 이연식, "데이터 검색의 적중률 향상을 위한 이중 캐시의 푸시 에이전트 모델 설