

이동 에이전트를 위한 트랜잭션 모델

유정준^υ, 이동익
광주과학기술원 정보통신공학과
(jjyoo, dilee)@kjist.ac.kr

Transaction Model for Mobile Agents

Jeong-Joon Yoo^υ and Dong-Ik Lee
Department of Information and Communications
Kwang-Ju Institute of Science and Technology(K-JIST)

요 약

이동 에이전트 시스템의 상용화 시작과 더불어 이동 에이전트 시스템에 대한 신뢰성 향상의 노력이 시작되었다. 이러한 노력에 부응하여 이동 에이전트 시스템 XMAS에서는 신뢰성 향상을 위한 방법으로 트랜잭션 개념을 도입하고 있다. 트랜잭션 개념의 도입으로 이동 에이전트 수행 결과의 일관성 유지가 가능하여 데이터에 대한 신뢰성 향상을 기대할 수 있다. 이동 에이전트는 기존 트랜잭션 모델과는 다른 다양한 특성을 갖는다. 따라서 이동 에이전트가 하나의 트랜잭션으로 처리되기 위해서는 기존 트랜잭션 모델과는 다른 트랜잭션 처리방법이 고려되어야 한다. 본 논문에서는 기존 트랜잭션 모델에서 발견되지 않는 이동 에이전트의 특성을 알아보고 이러한 특성들이 트랜잭션 모델과 트랜잭션 처리방법에 미치는 영향을 살펴본다.

1. 서 론

1990년대 초반 이동 에이전트 개념의 보급과 함께 1990년대 중반 이후 Aglets[1], Mole[2], XMAS[3] 등 많은 이동 에이전트 시스템이 개발되어 왔다. 1990년대 후반 들어 이동 에이전트 패러다임은 다양한 응용분야에 적용이 시도되고 있으며, 기존 클라이언트 서버 패러다임이 갖는 많은 문제점을 해결할 수 있을 것으로 기대되고 있다.

이러한 기대와 더불어 최근 이동 에이전트 시스템의 상용화가 시작단계에 있으나 상용화 시기가 아직 이른 감이 있다. 성공적인 상용화를 위해서는 이동 에이전트가 갖는 가장 큰 문제인 보안문제 해결이 진척되어야 한다. 보안문제와 더불어 해결되어야 할 중요한 문제는 데이터의 일관성 유지문제이다. 동시에 많은 수의 이동 에이전트가 자료에 대한 변경을 수행할 경우 이를 제어해 줄 수 있는 동시성 제어관리자, 그리고 오류 발생 시 이에 대한 처리를 담당해 줄 수 있는 회복관리자에 대한 연구가 진행되어야 한다. 이러한 노력의 결과로 이동 에이전트의 동작에 신뢰성을 제공할 수 있을 것으로 기대된다. 본 논문에서는 이러한 신뢰성 향상을 위한 기반 연구로서 이동 에이전트를 위한 트랜잭션 모델을 제시하고자 한다. 이를 위해 기존 트랜잭션 모델에서 발견되지 않는 이동 에이전트의 특성을 알아보고 그러한 특성들이 트랜잭션 모델과 트랜잭션 처리방법에 미치는 영향을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 다른 이동 에이전트 시스템에서 진행중인 신뢰성 향상방법을 소개하고, 3장에서는 기존 트랜잭션 모델에서 발견되지 않는 이동 에이전트의 특성들을 제시하며, 4장에서는

이러한 특성들이 트랜잭션 모델 및 처리방법에 미치는 영향을 설명한다. 끝으로 5장에서는 이동 에이전트를 위한 트랜잭션 모델을 정리하고, 본 연구의 결과를 바탕으로 진행될 향후 연구 과제를 설명한다.

2. 관련연구

이동 에이전트 시스템의 상용화와 더불어 이동 에이전트 동작의 신뢰성 향상이 요구되었다. 표1은 각 이동 에이전트 시스템에서 신뢰성 향상을 위해 제공하고 있는 기능들을 보여주고 있다. 각 이동 에이전트 시스템들은 모두 Checkpoint 기법을 제공하여 이동 에이전트의 중간상태를 저장하고 있다. Concordia에서는 네트워크 오류에 대한 신뢰성 있는 에이전트의 전송을 위한 방법과 더불어 검색연산과 같이 여러 번 반복해도 결과가 같은 멱등연산(idempotent operation)에 대한 회복방법을 제공하고 있다. Mole에서는 Exactly Once Property를 제시하여 에이전트가 반드시 그리고 꼭 한번만 수행될 수 있는 방법을 제시하고 있다[4]. 그러나 이들은 모두 여러 이동 에이전트가 실행되며 같은 데이터를 접근할 때 발생할 수 있는 데이터의 비 일관성 문제를 해결하지 못하며 데이터에 대한 신뢰성 향상의 노력이 요구되고 있다.

3. 기존 트랜잭션 모델에서 발견되지 않는 이동 에이전트의 성질

기존 트랜잭션 모델(traditional transaction model)이라 함은 데

표1: 이동 에이전트 시스템의 신뢰성 개념 지원 현황

Mobile Agent System	Checkpoint	Reliable agent transmission with 2 PC handshaking	Recovery for idempotent operation	Exactly Once Property for Mobile Agents
Ara	O	X	X	X
MOA	O	X	X	X
Telescript	O	X	X	X
Voyager	O	X	X	X
Concordia	O	O	O	X
Mole	O	X	X	O

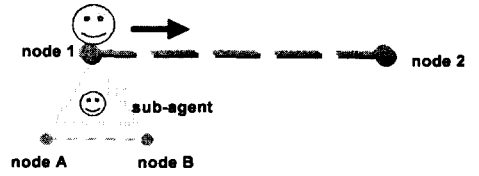


그림2: 이동성의 영향

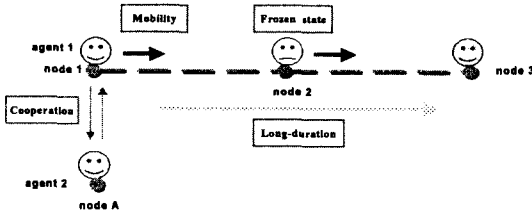


그림1: 이동에이전트의 특성

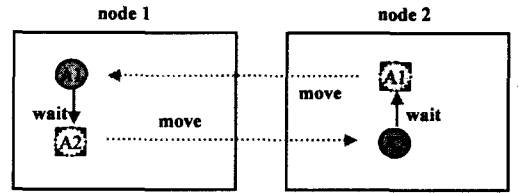


그림3: 전역 무한대기의 발생

이터의 일관성 유지를 위한 트랜잭션이 지켜야 할 제약사항인 원자성(Atomicity), 일관성(Consistency), 고립성(Isolation), 지속성(Durability)¹⁾, 성질을 모두 준수하는 트랜잭션 모델을 의미한다.

그림1과 같이 기존 트랜잭션과는 달리 이동에이전트는 트랜잭션 자체가 이동하며(Mobility), 트랜잭션이 수행도중 다른 트랜잭션들과 서로 통신함으로써 자료를 공유한다(Cooperation). 또한 이동에이전트는 시스템의 종료나 메모리 사용의 절약을 위해 하드디스크로 저장시키는 냉동상태(Frozen state)로 상태변화가 가능하며 이동성, 협동, 냉동상태 등으로 인해 전체적으로 트랜잭션 수행시간이 길어진다(Long-duration).

4. 이동에이전트를 위한 트랜잭션모델 및 트랜잭션 처리방법

3장에서 제시된 이동에이전트의 특징들이 트랜잭션 모델에 미치는 영향을 알아봄으로써 이동에이전트를 위한 트랜잭션 모델을 제시하고자 한다. 이와 더불어 이동에이전트의 특징을 이용한 효율적인 트랜잭션 처리방법도 제시한다.

4.1 이동성(Mobility)의 영향

그림2와 같이 트랜잭션(에이전트)은 노드1에서 지역적 트랜잭션(local transaction)을 수행하거나 원격 트랜잭션 수행을 위해 서브에이전트를 생성하여 서브트랜잭션을 위임할 수 있다. 트랜잭션 자체가 이동함으로써 분산트랜잭션간에 생기는 메시지 전송량을 줄일 수 있는 반면, 결과반환을 위해서는 이동하는 트랜잭션의 위치를 서로 알고있어야 하는 오버헤드가 발생한다. 반면 트랜잭션 이동에 의해 분산트랜잭션간의 메시지 전송량이 줄어들고, 안정적이지 못한 네트워크에 대해 오류발생 빈도를 줄여, 결과적으로 신뢰성 있는 트랜잭션 처리방법을 제공할 수 있다. 부모트랜잭션은 지역적 트랜잭션 또는 서브트랜잭션이 완료될 때까지 기다리지만, 지역적 또는 서브트랜잭션 내부에 포함된 연산과 다음 이동할 노드에서 수행될 연산이 서로 충돌관계(conflict relationship)에 있지 않다면, 그들의 완료를

기다리지 않고 다음 노드로 이동하여 연산을 수행하는 것이 전체 트랜잭션 수행시간을 단축시킬 수 있다. 지역적 트랜잭션 또는 서브트랜잭션의 수행결과는 이동한 새로운 노드에서 받게 수 있다. 분산트랜잭션의 연속철회(cascade rollback)을 방지 위해 사용되는 2단계 완료(2-phase commit)정책은 이동에이전트의 long-duration 성질 때문에 사용될 수 없다. 장기간 분산자원에 로크를 소유하는 것은 트랜잭션의 동시성을 떨어뜨리기 때문이다. 또한 자원에 대해 로크를 소유하고 이동하는 것은 그림3과 같은 전역 무한대기를 발생시킬 수 있다. 즉, 노드2의 자원에 로크를 소유하고 노드1으로 이동한 에이전트 A1은 에이전트 A2가 또 다른 자원에 대한 로크의 해제를 기다리고 있으며, 에이전트 A2 또한 노드1내의 자원에 대한 로크를 소유한 상태로 노드2로 이동하여 에이전트 A1이 소유하고 있는 자원에 대한 로크의 해제를 기다리고 있다. 이러한 무한대기의 해결책으로는 이동에이전트의 장기수행(long-duration) 특성 때문에 모든 자원에 대한 로크를 미리 소유하는 무한대기 방지(deadlock prevention)방법보다는 무한대기 발견 및 회복(deadlock detection and recovery)방법이 트랜잭션의 동시성을 높일 수 있다.

4.2 협력(Cooperation)의 영향

서로 협동하는 트랜잭션들의 수행결과가 일관성을 유지하는지 검사하는 정확성 판단기준(correctness criteria)은 협동의 의미가 다양하듯 그에 따른 판단기준이 다양해질 수밖에 없다. 그러나, 대체로 서로 협동하는 트랜잭션들을 그림4와 같이 하나의 트랜잭션 그룹으로 묶어, 그들간의 정확성 판단기준을 마련한다. 이러한 판단기준은 트랜잭션들간의 협력순서를 정의함으로써 가능하다. 즉 트랜잭션들간의 내부 연산호출 순서를 미리 정의하고 트랜잭션들이 이를 잘 준수하는지 검사하면 된다. 실제 협동트랜잭션 정확성 판단기준은 응용에 따라 모두 달라므로 응용에 맞게 사용자가 정해야 한다[5]. 한 트랜잭션이 진행중인 다른 트랜잭션의 내부상태를 읽을 수 있기 때문에 기존 트랜잭션의 제약사항중 하나인 고립성의 성질은 비고립성(non-isolation)으로 변경되어야 한다.

1) ACID properties

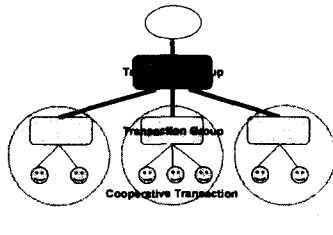


그림4: 협동의 영향

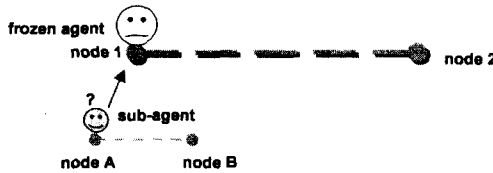


그림5: 에이전트 냉동상태의 영향

4.3 냉동상태의 영향

그림5는 이동에이전트의 냉동상태를 갖는 특성이 트랜잭션 처리에 미치는 영향을 설명하고 있다. 노드A에 존재하는 트랜잭션(sub-agent; 트랜잭션1이라 하자)이 노드1에 존재하는 트랜잭션(냉동상태의 에이전트; 트랜잭션2라 하자)의 내부변수를 접근하려는 경우를 살펴보자. 기존 트랜잭션 처리방법을 적용할 경우 트랜잭션1은 냉동상태에 있어 응답할 수 없는 트랜잭션2의 내부변수를 접근하기 위해 대기하고 있다. 냉동상태에서 다시 활성상태로 바뀌는 시간을 예측할 수 없고 심지어 수 시간이 소요될 수 있으므로 트랜잭션1의 처리시간 향상을 위해 낙관적(optimistic)방법을 사용한다. 여기서 낙관적 방법이란 접근하고자 하는 트랜잭션의 내부변수를 접근할 수 없는 상황일 때, 변수의 최근의 값을 가지고 일(연산1이라 하자)을 진행한다. 만약 이로 인해 나중에 두 트랜잭션간의 연산충돌(operation conflict)이 발생하면 연산1을 철회(rollback)하고, 충돌이 발생하지 않으면 연산1의 결과를 트랜잭션2의 내부에 반영한다. 이는 냉동상태에 있는 트랜잭션 내부변수 값의 최근상태를 알 수 있을 때 가능하다. 낙관적 방법을 사용함으로써 인해 잠시나마 데이터의 비 일관적인 상태가 존재하지만(momentary inconsistency) 최후에는 일관성을 유지시키며 트랜잭션 처리속도의 향상을 기대할 수 있다는 장점이 있다.

4.4 장기수행(Long-Duration)의 영향

장기수행이 트랜잭션 수행에 미치는 영향은 워크플로우 응용에서 많이 연구되었다. Long-duration을 갖는 트랜잭션의 전체 철회시 재수행 비용이 비싸기 때문에 전체 철회보다는 부분철회 방법을 사용하고 있다(non-atomicity). 이를 위해 트랜잭션은 그림6과 같이 nested-structure를 가지며 이미 완료된 서브트랜잭션의 철회(semantically undo; non-durability)를 위해 보상연산(compensating operation)을 제공한다[6]. 이러한 워크플로우를 위한 트랜잭션의 성질은 long-duration을 갖는 이동에이전트에 똑같이 적용될 수 있다.

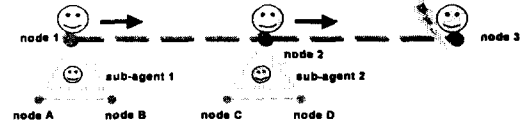


그림6: 장기 수행의 영향

5. 결론 및 향후 연구과제

광주과학기술원에서 개발중인 이동에이전트 시스템 XMAS에서는 신뢰성 있는 이동에이전트의 동작을 지원하기 위해 트랜잭션 개념을 도입하고 있다. 본 연구는 이동에이전트 패러다임에 트랜잭션 개념을 도입하기 위해서는 기존 트랜잭션 모델과 트랜잭션 처리방법을 그대로 사용할 수 없음을 보여주고 있다. 이동에이전트의 성질인 이동성, 협력, 냉동상태, 그리고 장기수행의 특성으로 인해 이동에이전트를 위한 트랜잭션 모델과 처리방법은 달라져야 함을 보였다. 이동에이전트 패러다임을 위한 트랜잭션 모델이 가져야할 특성은,

- 비 원자성(non-Atomicity),
- 일시적 비 일관성(momentary Inconsistency),
- 비 고립성(non-Isolation),
- 비 지속성(non-Durability) 이다.

그리고, 이동에이전트를 위한 트랜잭션의 처리방법으로, 에이전트의 이동시간, 로크소유 후 이동시 발생할 수 있는 전역대기 문제, 에이전트 수행결과와 일관성 유지를 위한 정확성 판단기준, 트랜잭션의 동시성 향상을 위한 데이터의 비 일관적 상태의 필요성, 그리고 트랜잭션 고장시 처리정책 등에 대해 정리하였다. 이러한 트랜잭션 모델과 처리방법은 앞으로 이동에이전트의 신뢰성 향상을 위해 필요한 이동에이전트의 동시성 제어 및 회복정책의 결정에 기본이 된다.

감사의 글

이 연구는 일부 한국과학재단(KOSEF)의 특정기초연구(98-0102-11-01-3)에 의해 지원되었습니다.

참고 문헌

- [1] Danny Lange and Mitsuru Oshima, Programming and Deploying Java Mobile Agents with Aglets, Published by Addison-Wesley, 1998.
- [2] J. Baumann, F. Hohl, K. Rothermel and M. StraBer, Mole- Concepts of a Mobile Agent System, International Journal of World Wide Web, Vol. 1, No. 3, pp. 123-137, 1998.
- [3] 유정준, 백주성, 박종열, 이동익, Java 기반 이동 에이전트 시스템: X-MAS, 한국통신학회 춘계학술대회, 1998.
- [4] K. Rothermel, M. StraBer, A Fault-Tolerant Protocol for Providing the Exactly-Once Property of Mobile Agents, In Proc. 17th IEEE Symposium on Reliable Distributed Systems 1998.
- [5] Ahmed K. Elmagarmid, Database Transaction Models for Advanced Applications, Published by Morgan Kaufmann Publishers, 1992.
- [6] Marek Rusinkiewicz and Amit Sheth, Specification and Execution of Transactional Workflows, In Modern Database Systems edited by Won Kim, Published by Addison-Wesley, 1994.