결함 허용을 고려한 효율적인 이동 에이전트 전송방법

조 수 현° 김 영 학
금오공과대학교 대학원 컴퓨터공학과
{shcho, yhkim}@cespc1.kumoh.ac.kr

Efficient Mobile Agents Transmission Method Considering Fault-Tolerance

Soo-Hyun Cho° Young-Hak Kim
Dept. of Computer Engineering, Kumoh National University of Technology

요 약

최근에 네트워크 환경이 좋아지고 인터넷 사용이 급증함에 따라 이동 에이전트(Mobile Agent) 기술이 정보검색, 네트워크 관리, 전자상거래, 병렬/분산처리 등의 분야에 널리 활용되고 있다. 이동 에이전트 시스템에서 전체 수행시간은 이동 에이전트의 코드 크기, 이동 에이전트의 데이터 크기, 이동 에이전트의 전송 방법에 따라 좌우된다. 본 논문에서는 이동 에이전트 환경에 참여하는 모든 노드들에게 보다 빠르게 에이전트를 전송하는 방법들을 실험적으로 평가하고, 에이전트 이동시 노드의 결함(Fault)이 발생할 때 그 해결방안을 연구한다. 에이전트 전송 및 결함 허용에 대한 성능평가는 IBM's Aglets Software Development Kit 환경을 이용하였다.

1. 서론

인터넷 사용의 급증에 따라 다양한 사용자들의 요구를 만족시키기 위해 최근에 이동 에이전트 기술이 여러 분야에서 연구되고 있다. 이동 에이전트[1]는 수행되는 범위가 한 시스템에 제한되지 않으며, 스스로 네트워크에 연결된 다른 시스템으로 이동할 수 있다. 즉, 에이전트 코드 스스로가 한 시스템에서 다른 시스템으로 이동할 수 있는 능력을 갖는다. 또한 비연결성(Non-connectivity) 특성으로 인해 에이전트 이동과 결과 전송 시에만 네트워크 연결이 이루어지므로, 네트워크 부하를 크게 감소시키고 분산 시스템에서 견고한(Robust) 통신과 결함 허용(Fault Tolerance)을 지원한다.

최근에 이러한 장점들을 이용하여 이동 에이전트 기술이 정보검색, 전자상거래, 병렬/분산처리, 네트워크 관리 및 보안등의 응용분야에 널리 이용되고 있다. 이동 에이전트를 이용한 분산 시스템에서 전체 수행시간에 영향을 미치는 요소는 다음과 같다.

- o 이동 에이전트의 크기
- o 이동 에이전트의 데이터(전송 데이터 양)
- o 이동 에이전트의 전송시 네트워크 상태
- o 효율적인 이동 에이전트의 전송 방법

본 논문에서는 인터넷 환경에서 바이노미얼 트리를 이용하여 이동 에이전트를 전송하는 새로운 방법을 제안하고, 성능평가를 통하여 이 방법이 다른 방법에 비해서 효율적임을 보인다. 또한 본 논문에서는 제안된 이동 에이전트 환경에서 결함 허용을 고려한 효율적인 에이전트 전송방법을 연구한다. 이동 에이전트의 효율적인 전송방법이란 참여하는 모든 노드에 보다 빨리 이동 에이전트를 도달시키는 것이고, 결함 허용은 이동 과정에서 노드들의 결함에도 불구하고 시스템 전체에 결함이 발생하지 않게 계속해서 해당작업을 올바르게 수행하는 것을 말한다.

현재까지 자바 기반 이동 에이전트 시스템들이 많이 개발되었으며, 대표적으로 IBM's Aglets Software Development Kit (이하 Aglets), Odyssey, Concordia, Voyager 등이 있다. 본 논문에서는 성능평가를 위해 Aglets을 이용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 개괄적으로 설명하며, 3장은 이동 에이전트의 전송방법 및 성능평가를 보이고, 4장에서는 결함 허용을 고려한 이동 에이전트 전송방법과 성능평가를, 끝으로 5장에서 결론 및 향후 연구과제를 설명한다.

2. 관련 연구

[2]는 이동 에이전트가 이주할 때 발생하는 노드 혹은 호스트의 결함에 대처하기 위한 이주 정책을 제시하였다. 즉, 결함이 발생한 노드 혹은 호스트에 대해 이동 에이전트의 이주를 보장하는 경로 재조정과 후위 복구 기법을 통한 신뢰성 있는 이주정책을 제안하였다. 하지만 [2]의 결과는 이동 에이전트의 적절한 라우팅(Routing) 기법이 있다는 전제하에서의 결함 허용만을 고려한 이주정책이다. [3]은 이동 에이전트에 대한 고아(Orphan) 노드 찾기와 성공적인 종료를 위한 그림자(Shadow) 프로토콜을 제안하였는데, 그림자 프로토콜은 단지 이동 에이전트가 이동 중에 결함이 발생한 경우에 대해 이를 찾아 처리하기 위한 프로토콜이지 이동 에이전트의 결함 허용과 효율적인 전송 방법에 대해서는 고려하지 않고 있다.

본 논문에서는 참여하는 노드들에게 보다 빠르게 이동 에이전트를 전송하는 3가지 방식, 마스터/슬레이브, 이진 트리, 바이노미얼 트리의 모델을 제안하고 각각 성능순 및 무작위순 구성에 대해 성능을 평가한다. 또한 위 결과에서 우수한 성능을 보이는 방식에 결함 허용을 고려한 성능평가를 한다.

3. 이동 에이전트 전송방법

본 논문에서는 이동 에이전트를 참여한 노드로 보다 빨리 전송하기 위해 마스터/슬레이브, 이진 트리, 바이노미얼 트리 방식을 고려한다. 또한 각각의 방식에 노드들의 성능순 및 무작위순에 따라 전송시간을 평가한다(표 1참조).

표 1. 성능평가 방법

방 법	분 류	비 고
마스터/슬레이브	성능순	성능이 좋은 순으로 슬레이브 구성
	무작위	
이진 트리	성능순	성능이 좋은 순으로 완전이진 트리를 구성
	무작위	
바이노미얼 트리	성능순	성능이 좋은 순으로 바이노미얼 트리를 구성
	무작위	

3.1.1 마스터/슬레이브 방식

마스터 에이전트가 여러 슬레이브 에이전트들에게 작업들을 전송하는 방식이다. 일반적으로 로컬(Local)에서는 마스터와 슬레이브가 병렬적으로 각각 다른 작업을 수행 할 수 있다. 다시 말해, 마스터 에이전트가 리모트에서 수행할 슬레이브 에이전트들을 생성하여 보내면 각각의 슬레이브 에이전트들은 할당된 작업을 수행한다. 각 슬레이브는 결과 값을 마스터 에이전트에게 보내고, 마스터 에이전트는 슬레이브로부터 전송 받은 값들을 취합해 최종 결과 값을 얻는다. 물론 마스터 에이전트도 작업 수행에 참여 할 수 있다.

3.1.2 이진 트리(Binary Tree) 방식

그림1과 같이 작업을 수행할 노드들을 완전이진 트리로 구성한다. 마스터 에이전트는 2개의 하위 자식 노드에게 이동 에이전트를 전송하고, 그 다음에 각각의 하위 부모 노드들은 병렬적으로 자신들의 자식 노드들에게 에이전트를 전송한다. 결국 LogS 단계 후에는 전체 노드에 이동 에이전트가 전송된다.

3.1.3 바이노미얼 트리(Binomial Tree) 방식

그림2는 4비트를 할당한 바이노미얼 트리[4]의 구성 예를 보여준다. 바이노미얼 트리는 자신에 할당된 비트의 상위 비트를 세트하여 자식 노드들을 구성한다. 바이노미얼 트리에서 이동 에이전트를 전송하기 위한 방법은 이진트리의 경우와 유사하다. 그러나 바이노미얼 트리의 경우는 하나의 노드가 한 개 이상의 자식노드에게 에이전트를 전송하게 된다. 후에 실험적 결과를 통하여 이 방법이 다른 방법에 비해서 에이전트를 효율적으로 전송함을 보인다.

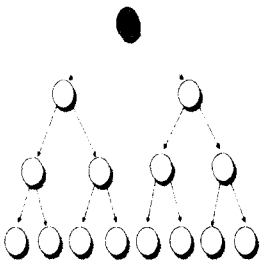

그림 1. 이진 트리

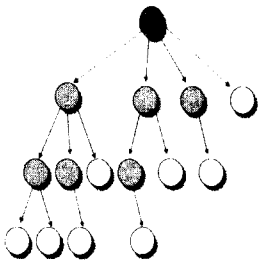

그림 2. 바이노미얼 트리

3.2 성능평가

3.2.1 실험환경

성능평가를 위한 실험 환경은 다음과 같다.

1. 운영체제 : Windows 95/98/ME
2. 소프트웨어 : JDK1.1.8, IBM Aglets1.1.b3
3. CPU/Memory : 펜티엄 III 733, 펜티엄 III 464, 펜티엄 III 451, 펜티엄 II MMX 360, 펜티엄 II MMX 262, 펜티엄 200 /, 128M, 96M, 64M, 32M

결함 허용을 고려한 성능평가에서도 같은 환경을 이용하였으며, 실험에 참여하는 노드들은 10Mbps 이더넷으로 연결된 12대의 PC로 구성되었다.

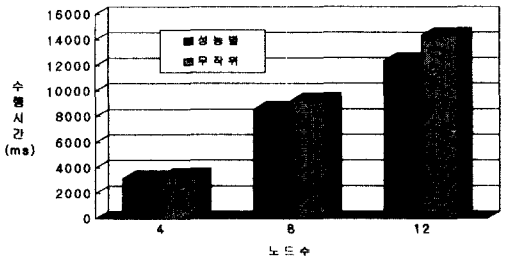

그림 3. 마스터/슬레이브 수행결과 비교

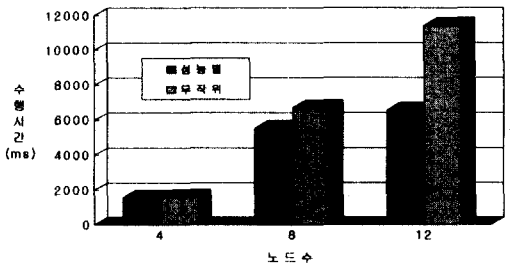

그림 4. 이진 트리 수행결과 비교

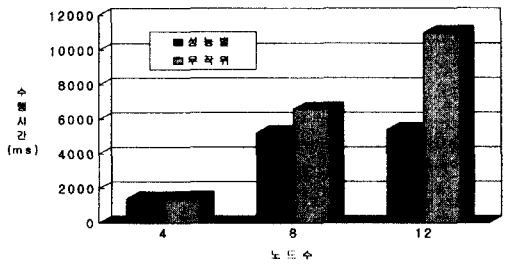

그림 5. 바이노미얼 트리 수행결과 비교

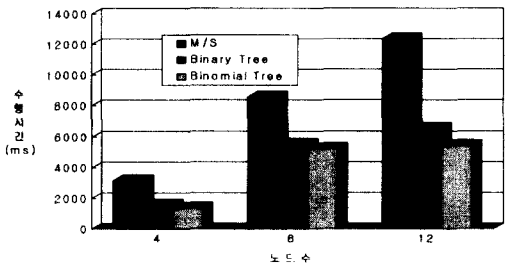

그림 6. 세 가지 방식 수행결과 비교

3.3 결과 분석

마스터/슬레이브 방식은 무작위 방법보다 성능별로 구성한 방법이 우수함을 알 수 있다. 하지만 다른 두 가지 방식에 비해 노드수 증가에 따른 성능향상이 저조했다. 또한 이진 트리와 바이노미얼 트리도 노드의 성능에 따라 구성되었을 때 결과가 향상됨을 알 수 있다.

그림6은 세 가지 방식의 성능평가 결과를 보여준다. 전체적으로 마스터/슬레이브 방식에 비해서 이진 트리와 바이노미얼 트리가 좋은 결과를 보였다. 이러한 결과는 에이전트의 분산을 여러 노드에 위임하는 효과의 결과로 볼 수 있다. 또한 바이노미얼 트리는 이진 트리에 비해서 성능이 다소 우세함을 확인할 수 있다. 이러한 결과는 매 경우 2개의 자식노드 보다는 다수개의 자식노드에게 에이전트를 전송함으로서 각 부모 노드의 유휴시간(Idle Time)을 줄인 결과로 볼 수 있다.

4. 결함 허용(Fault Tolerance)

분산 시스템에서 때때로 하나 이상의 노드 혹은 호스트의 결함이 발생할 수 있는데 이런 하드웨어 혹은 소프트웨어의 결함으로 인한 부정확한 결과 값을 나타내거나 전체 계산작업이 마치기 전 시스템이 멈출 수가 있다. 이런 문제점을 해결코자 시스템 전체에 결함이 발생하지 않게 계속해서 해당 작업을 올바르게 수행하는 것을 결함 허용이라 한다. 실시간 처리를 요구하는 교환기 및 네트워크 시스템 등에 사용되고 있다.

4.1 결함 원인 및 해결책

4.1.1 마스터 노드(Master Node) 결함

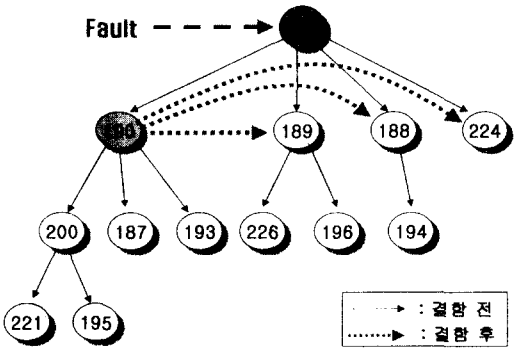

그림 7. 마스터 노드 결함 발생

자식 중에서 가장 성능이 우수한 노드가 부모노드를 대신한다. 즉, 그림7에서와 같이 190, 200, 221번 노드가 부모노드로 새롭게 구성되는 것이 아니라, 190번 노드만이 이웃하는 노드들에게 마스터가 해야할 일을 대신하게 된다.

4.1.2 부모 노드(Parent Node) 결함

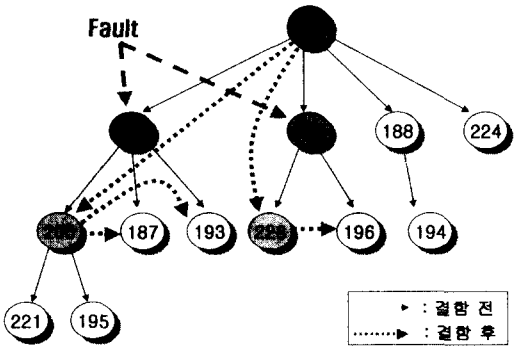

그림 8. 부모 노드 결함 발생

자식중 가장 성능이 우수한 노드가 부모노드를 대신하여 이동 에이전트를 전송한다. 그림8에서 나머지 노드들의 구성은 변하지 않고 200, 226번 노드만이 190, 189번 노드의 역할을 대신한다.

4.1.3 리프 노드(Leaf Node) 결함

결함이 발생한 노드의 부모 노드는 나머지 자식중 성능이 우수한 노드에게 다시 이동 에이전트를 전송한다. 그림7에서 195번 노드가 결함일 때 200번 부모노드는 221번 노드에게 전송한다. 만약에 자식노드 모두가 결함이 발생했다면 부모노드 자신이 작업을 수행하게 된다.

4.2 성능 평가

그림9는 4.1.2의 부모 노드의 결함에 대한 실험결과이다. 첫 번째, 결함 노드수가 1개 일 때, 결함 노드에 대한 일정시간 아무런 응답이 없으면 다른 노드로 이동 에이전트를 전송하여 수행한 결과와 이미 190번 노드가 결함임을 알고 다음 노드로 바로 전송하여 수행한 결과, 다시 말해 186번 노드에서 200번 노드로 전송되었다는 의미이다. 두 번째 실험은 참여하는 노드들이 결함이 없이 정상적으로 수행된 결과이다. 마지막으로 결함 노드수가 2개 일 때 수행된 결과를 나타낸다.

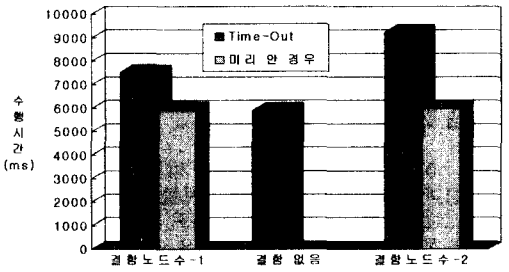

그림 9. 결함 허용 수행결과 비교

5. 결론 및 향후 연구과제

이동 에이전트를 이용한 분산 시스템은 이동 에이전트 크기, 이동 에이전트 데이터 크기, 전송시 네트워크 상태, 효율적인 에이전트 전송방법들을 고려해야 한다. 본 논문에서는 마스터/슬레이브, 이진 트리, 바이노미얼 트리 방식에 대한 성능평가를 수행하였다. 실험결과에 의하면 참여하는 노드들을 무작위로 배치하기보다는 성능순으로 구성되었을 때 더 좋은 결과를 확인할 수 있었다. 또한 이들 방법 중에서 바이노미얼 트리 방식이 가장 효율적임을 실험적으로 확인하였다.

결함 허용의 경우는 3가지 경우를 고려하였고 각각의 경우 노드들의 결함이 발생했을 때 새롭게 구성하기보다는 나머지 노드는 그대로 두고, 결함이 발생한 노드를 기준으로 자식중 성능이 우수한 노드에게 역할을 맡김으로 네트워크의 트랙픽과 부하를 감소시켜 결국 전체수행시간을 단축 시킬 수 있음을 알 수 있었다.

향후 연구과제는 실시간 처리, 다중 에이전트 시스템, 병렬/분산처리 등의 응용분야에 적용시키는 것이다.

참고문헌

[1] Lange, D.B., M.Oshima, "Programming and deploying Java Mobile Agents with Aglets", Addison Wesley Press, 1998.

[2] 전병국, 최형근, "이동 에이전트를 위한 효율적인 이주정책의 설계 및 구현", 정보처리논문지, 제6권, 제7호, Jul. 1999

[3] J.Baumann, "A Protocol for Orphan Detection and Termination in Mobile Agent Systems", TR-1997-09, Stuttgart Univ, July, 1997

[4] S.L.Johnsson and C.T.Ho, "Optimum Broadcasting and Personalized Communication in Hypercubes", IEEE Trans. Computers, Vol.38, no.9, PP. 1249-1268, Sept. 1989