다중 이동 에이전트 제어를 위한 위치 관리 기법*

윤준원[O] 강석준[†] 최성진[†] 손진곤[†] 황종선[†]
고려대학교 컴퓨터학과 분산시스템 연구실, 한국방송대학 컴퓨터과학과[†]
{zebra[O], kangsa[†],lotieye[†],hwang[†]}@disys.korea.ac.kr, jgshon@mail.knou.ac.kr[†]

Location Management Scheme for the Control of Multiple Mobile Agents

JunWeon Yoon[O], SeokJun Kang, SungJin Choi, ChongSun Hwang
Distributed System Lab. Dept. of Computer Science & Engineering, Korea University
JinGon Shon
Dept. of Computer Science, Korea National Open University

요 약

여러 지역으로 구성된 다중 이동 에이전트 컴퓨팅 환경에서 에이전트들을 관리하기 위한 위치 관리 기법은 이동 에이전트 시스템을 개발하는데 있어서 중요한 고려 사항이다. 그러나, 기존의 연구에서는 단지 하나의 이동 에이전트를 통제하기 위한 위치 관리만을 고려할 뿐 같은 작업을 수행하는 다중 이동 에이전트들을 통제하는 위치 관리 기법은 다루지 않고 있다. 따라서, 이전 기법을 다중 이동 에이전트를 관리하는데 적용하면 다중 이동 에이전트들을 관리하는 비용은 증가하게 된다.

본 논문에서는 여러 지역으로 구성된 이동 에이전트 컴퓨팅 환경에서 다중 이동 에이전트 관리를 위한 위치 관리 기법인 WSC(Workname Search and Control)기법을 제안하고자 한다. 제안된 기법은 다중 이동 에이전트 관리를 위한 모델을 제시함으로써 다중 이동 에이전트 관리를 위한 메시지 전달 부하를 감소시킨다.

1. 서 론

이동 에이전트는 사용자를 대신하여 호스트와 호스트를 자율적으로 이동하면서 수행하는 프로그램이다[1][2]. 여러 지역으로 구성된 이동 에이전트 컴퓨팅 환경에서 이동 에이전트 수행 중 이동 에이전트를 관리하기 위한 위치 관리 기법은 이동 에이전트 시스템을 개발하는데 있어서 중요한 고려사항이다. 그러나, SPC[3]와 같은 기존 연구에서는 단지 하나의 이동 에이전트를 제어하기 위한 위치 관리를 고려할 뿐 같은 작업을 하는 다중 이동 에이전트들의 통제를 위한 위치 관리 기법은 다루지 않고 있다. 이로 인해 같은 작업을 수행하는 다중 이동 에이전트를 관리하기 위한 메시지 전달 비용이 증가하게 된다.

본 논문에서는 여러 지역으로 구성된 다중 지역 이동 에이전트 환경에서 같은 작업을 하는 다중 이동 에이전트들을 관리하기 위한 위치 관리 기법인 WSC 기법을 제안하고자 한다. 제안된 기법은 기존 SPC와 달리 작업 식별자를 이동 에이전트 네이밍 생성시 부여하여 작업 식별자로 다중 이동 에이전트를 관리하는 모델을 제시하며, 다중 이동 에이전트 관리에서 발생하는 부하를 줄이기 위해 지역 서버들간 위치 정보를 교환하는 새로운 위치 관리 기법을 제안한다.

2. 관련 연구

이동 에이전트 컴퓨팅 환경에서 이동 에이전트에 대한 위치 관리 기법은 크게 홈 프락시(Home-Proxy), 추적 프락시(Follower-Proxy), 브로드캐스트(Broadcast)기법으로 분류된다[5].

홈 프락시 기법[5]에서는 위치 정보 저장소를 이동 에이전트가 생성된 홈 호스트에 위치시키고 이동 에이전트 이주 시마다 그 위치 정보를 홈 호스트의 위치 정보 저장소에 저장한다. 위치 탐색 시 이 정보를 이용하여 이동 에이전트를 찾아 메시지를 전달하는 기법이다. 추적 프락시 기법[5]은 위치 정보를 방문한 호스트들에 저장한 다음 경로 프락시(Path proxies)를 따라가서 이동 에이전트를 찾거나 메시지를 경로 프락시를 따라 계속해서 포워딩(Forwarding)하는 기법이다. 브로드캐스트 기법[5]은 정해진 지역이나 이동계획(Itinerary)의 모든 호스트들에게 메시지를 보내어 위치를 찾거나 메시지를 전달하는 기법이다. 위 세 가지의 기법은 단일 지역으로 구성된 이동 에이전트 컴퓨팅 환경에 기반하고 있다. 따라서, 다중 지역에 적용되었을 경우 많은 메시지 갱신 비용을 초래하게 된다.

* 본 연구는 한국과학재단 목적기초연구(R01-2002-000-00235-0)지원으로 수행되었음.

여러 지역으로 구성된 이동 에이전트 컴퓨팅 환경에서의 위치 관리 기법으로는 SPC[3] 프로토콜이 있다. 이 기법에서 이동 에이전트는 한 호스트에서 생성되고 그 지역 서버는 지역내의 이동 에이전트의 정보를 가지고 있게 된다. 이동 에이전트는 다중 지역 즉, 여러 지역을 이주하게 되므로 많은 갱신 메시지 비용이 발생하게 되며 위치 관리 시에는 갱신 메시지의 분실로 인해 호스트의 링크를 따라가는 많은 탐색 비용을 초래할 수 있다. 또한 이동 에이전트 생성 시 등록되는 이동 에이전트 이름은 이동 에이전트 식별자와 지역주소로 표현되나 이동 에이전트 이름에는 이동 에이전트의 작업에 관한 정보가 없어 위치 탐색시 같은 작업을 하는 다중 이동 에이전트를 제어하는데 많은 메시지 전달 비용이 발생한다.

3. 제안 기법

본 논문에서는 여러 지역으로 구성된 다중 이동 에이전트 컴퓨팅 환경에서 같은 작업을 수행하는 다중 이동 에이전트들을 관리하기 위해 이동 에이전트에 작업 식별자를 부여한 WSC 기법을 제안한다.

3.1. 이동 에이전트 컴퓨팅 환경

본 논문에서 가정하는 이동 에이전트 컴퓨팅 환경은 이동 에이전트 a_i, 실행장소 p_i로 구성된다. 이동 에이전트 a_i는 각 호스트의 실행 결과에 따라 다음 호스트를 동적으로 선택하여 이동하거나, 이동계획(itinerary)에 따라 호스트와 호스트사이를 옮겨다니며 작업을 수행한다. 이때 호스트에서는 이동 에이전트가 수행할 수 있는 실행장소 p_i를 제공한다[2]. 실행장소는 이동 에이전트에게 특정 서비스를 제공한다. 한 호스트에는 여러 실행장소가 있을 수 있다.

이동 에이전트 시스템들 중에서 같은 권한을 가진 이동 에이전트 시스템들의 집합을 지역 R_i 이라고 한다[2]. 본 논문에서는 이동 에이전트가 여러 지역으로 구성된 다중 지역 이동 에이전트 컴퓨팅 환경에서 이동하면서 작업을 수행하는 걸로 가정한다.

각 지역에는 지역의 권한을 책임지는 지역 서버(RS:Region Server)가 존재한다[2]. 그리고 다중 지역 이동 에이전트 컴퓨팅 환경에서 전역적인 네이밍 서비스를 제공하는 룩업서비스 서버(LSS:Lookup Service Server)가 존재한다.

본 논문에서 가정하는 이동 에이전트 모델은 10개미만의 지역의 수가 제한된 경우를 가정하며 이 지역 내에서 이동 에이전트는 특정 작업을 수행하며 이주하게 된다. 그림 1은 본 논문에서 가정하는 다중 이동 에이전트 컴퓨팅 환경을 나타내고 있다.

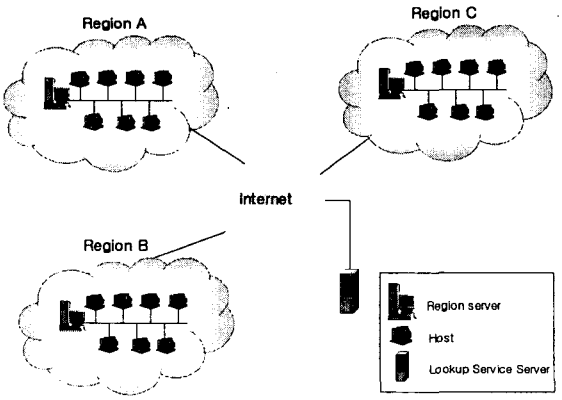

그림 1. 다중 지역 이동 에이전트 컴퓨팅 환경

3.2. WSC 기법

WSC 기법은 다음과 같이 크게 세 과정 즉, 네이밍&생성, 이주, 위치관리과정으로 구분된다.

- **네이밍&생성(Naming&Creation)**
  이동 에이전트가 생성되면 식별되어지기 위한 네이밍이 부여되어야 하며 또한 위치 관리를 위해 정보를 등록해야 한다. 같은 작업을 하는 다중 이동 에이전트를 관리하기 위해 LSS는 이동 에이전트 생성시 이동 에이전트에 작업 식별자를 부여하는 네이밍 서비스를 제공한다. 이동 에이전트가 생성되면 생성된 지역서버에 이동 에이전트의 식별자와 생성된 호스트의 주소({AgentID, HostLocation})를 등록하고 또한 LSS에 이동 에이전트 식별자, 작업 식별자, 생성된 지역 주소({AgentID, WorkID, HRSLocation})를 등록한다.
- **이주(Migration)**
  이동 에이전트는 호스트와 호스트를 옮겨다니는 이주를 통해 작업을 수행하게 된다. 이동 에이전트의 이주 과정은 크게 같은 지역으로 이주하는 지역 내이주(Intra Migration)와 다른 지역으로 이주하는 지역외 이주(Inter Migration)로 나누어진다. 지역내 이주시 갱신정보는 단지 지역내 서버에게만 알리게 되고, 다른 지역으로 이주하는 경우 이동 에이전트는 이주 후 새로운 지역 서버에게 갱신 메시지를 전달하고 그 지역서버는 갱신된 메시지를 다시 모든 지역 서버에게 이동 에이전트 이주에 대한 갱신 메시지를 멀티캐스트 한다. 그림 2는 지역내 이주와 다른 지역으로 이주를 포함한 전체 이주과정 알고리즘이다.

```
if ai : Hi -> Hi+1 then  // 에이전트가 다음 호스트(H)로 이주
    if Compare(RS of Hi, RS of Hi+1) = 1 then
        /* Intra Migration*/
        UpdateMsg(RS of Hi+1, HA of ai);
    fi;
    if Compare(RS of Hi, RS of Hi+1) = 0 then
        /* Inter Migration*/
        UpdateMsg(RS of Hi+1, HostLocation of ai);
        UpdateMsg(All Rs except RS of Hi+1,
                RSLocation of ai);  /*Multicast*/
    fi;
fi;
```

그림 2. WSC 기법의 이주과정 알고리즘

- **위치관리(Location Management)**
  이동 에이전트들에게 메시지를 전달하기 위해 다음과 같은 위치관리 절차를 갖는다.

① 메시지를 전달하고자 하는 사용자는 수행하고자 하는 작업 식별자 즉, WorkID를 가지고 LSS에 질의한다.
② LSS는 작업 식별자에 해당하는 이동 에이전트들의 정보를 사용자에게 제공한다.
③ 작업 식별자로 얻은 정보를 바탕으로 해당하는 이동 에이전트가 생성된 각 지역 서버에 메시지를 전달한다.
④ 이동 에이전트가 홈 지역 내에서 이주한 경우 직접 메시지를 전달한다. 홈 지역 외로 이주한 경우에는 각 지역 서버는 이주된 이동 에이전트가 존재하는 지역 서버로부터 멀티캐스트를 이용하여 이주 정보를 받았으므로 이동 에이전트가 존재하는 지역서버로 메시지를 전달하고 지역서버는 호스트로 메시지를 전달한다.

그림 3은 메시지 전달 과정에 대한 알고리즘이다.

```
Set DBConnection LSS
DBConnection.open
sql="Select * from LSS Where WokrID = 'WorkName'"
ReS = Excute(sql) /*get recoreds AgentID, WokrID,
        HRSLocation correspond to WorkName form LSS*/
Do while ReS.EndOfFile = False
    if Search(ReS.HRSLocation, ai) = 1 then  // Intra Migration
        UpdateMsg(Update, HostLocation of ai);
    fi;
    if Search(ReS.HRSLocation, ai) = 0 then  // Inter Migration
        SendMsg(Update, DRS(DifferentRS)Location of ai);
        UpdateMsg(Update, HostLocation of ai);
    fi;
    ReS.NextRecord
Loop
```

그림 3. WSC기법에서 위치 관리 알고리즘

4. 시나리오

본 논문의 기법이 어떻게 동작하는지 예를 통해 알아보자. 그림 2에서와 같이 이동 에이전트 컴퓨팅 환경은 4개의 지역 korea.ac.kr, kisti.re.kr, chunan.ac.kr, gongju.ac.kr로 구성되어 있다.

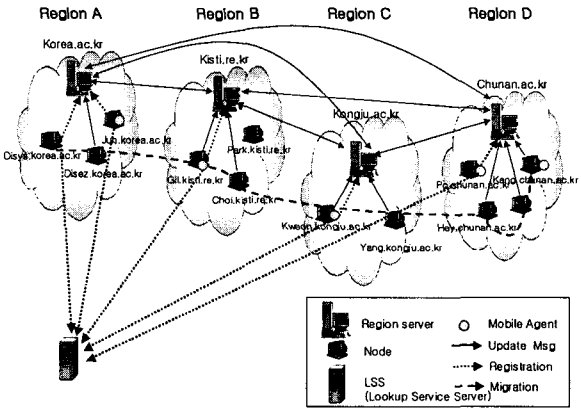

그림 4. 네이밍&생성과 이주과정

각 지역마다 특정 작업에 대한 이동 에이전트를 생성하고, 생성된 지역의 지역서버와 전역적인 이동 에이전트 정보를 제공하는 LSS에 아래 표1과 같이 이동 에이전트 정보를 등록하게 된다.

AgentID	WorkID	HRSLocation
1495	Reverse	Chunan.ac.kr
2174	Upgrade	Chunan.ac.kr
2542	Upgrade	Korea.ac.kr
3487	Change	Kisti.re.kr
4784	Change	Kongju.ac.kr
5199	Upgrade	Korea.ac.kr
6247	Upgrade	Kisti.re.kr

표 1. LSS 등록정보

표1에 있는 생성된 모든 이동 에이전트는 같은 지역 또는 다른 지역으로 각각 이주하게 된다. 예를 들어, 그림 4의 지역 A에서 생성된 이동 에이전트는 disys.korea.ac.kr → Disez.korea.ac.kr → Gil.kisti.re.kr → Choi.kisti.re.kr →Kweon.kongju.ac.kr → yang.kongju.ac.kr → hey.chunan.ac.kr → kang.chunan.ac.kr 의 이주 경로를 갖는다. 지역내 이주 disys.korea.ac.kr → Disez.korea.ac.kr와 같은 동일 지역이 경우 korea.ac.kr의 지역서버만을 갱신하고, Disez.korea.ac.kr → Gil.kisti.re.kr 와 같은 다른 지역으로 이주 시 이주한 새로운 지역 즉, kisti.re.kr의 지역 서버를 갱신한 후 다른 모든

(korea.ac.kr, kongju.ac.kr, chunan.ac.kr) 지역 서버에 갱신 메시지를 멀티캐스트 한다. 따라서, 각 지역서버는 이동 에이전트의 이주정보를 가지고 있게 된다.

위치 관리시 사용자는 현재 동작중인 이동 에이전트의 제어(예를 들어, 이동 에이전트 삭제, 정지, 변경 등과 같은 작업)를 위해, 작업 식별자를 명시하고 있는 WorkID에 Upgrade란 키워드로 LSS에 질의하게 된다. 사용자는 LSS로부터 Upgrade와 일치하는 이동 에이전트의 정보에 대한 결과를 얻게 된다. 그 결과는 표 2와 같이 얻어진다.

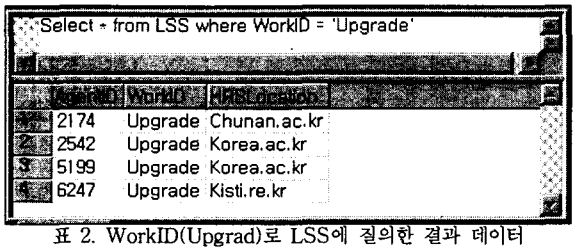

표 2. WorkID(Upgrad)로 LSS에 질의한 결과 데이터

표 2와 같이 LSS로부터 얻은 결과에는 질의한 작업에 대한 이동 에이전트들의 식별자와 홈RS 주소 얻을 수 있다. AgentID가 2174인 이동 에이전트인 경우 우선 홈RS인 Chunan.ac.kr로 메시지를 전달한다. 홈RS는 이동 에이전트가 다른 지역으로 이주했음을 알고 있으므로 이동 에이전트가 존재하는 Kongju.ac.kr의 RS에 메시지를 다시 전달한다. Kongju.ac.kr의 RS는 이동 에이전트가 존재하는 호스트 Kweon.kongju.ac.kr로 메시지를 보내게된다. AgentID 2542와 5199는 홈RS의 주소가 같으므로 두 개의 이동 에이전트에 대해 하나의 메시지로 홈RS에게 보낸다. AgentID 5199는 지역내에서 이주를 하였으므로 Disez.korea.ac.kr로 바로 메시지를 전달하고, AgentID 2542는 지역외 이주를 하였으므로 이주한 Chunan.ac.kr로 메시지를 전달한 후 바로 지역내 hey.chunan.ac.kr에 메시지가 전달된다. AgentID 6247은 같은 지역에서 이주하였으므로 Kisti.re.kr지역서버에서 Gil.kisti.re.kr 호스트로 메시지를 전달한다.

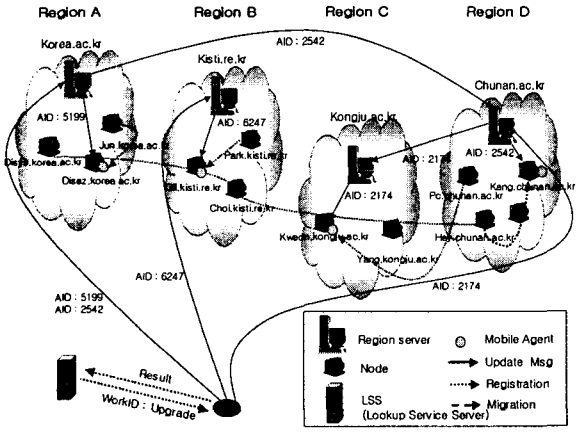

그림 5. WSC 위치 관리 과정

5. 분석 및 토론

SPC기법과 WSC 기법의 위치 갱신 메시지 비용 및 메시지 전달 비용을 비교해보자.

같은 지역내에 호스트들 사이에서의 연결은 다른 지역 사이의 호스트들 사이보다 더 빠르고 신뢰성이 있다고 가정한다[5]. 이 가정은 분산 환경에 제한 없이 적용된다. 일반적으로 같은 지역내에 호스트들은 LAN(10~100 MB/s)에 의해 연결되어져 있고 지역과 지역사이의 연결은 WAN으로 더 낮은 속도(64Kb/s~2Mb/s)로 동작한다[6]. LAN에서의 지연시간은 1~10ms를 WAN에서는 100~500ms를 갖는다.

메시지가 전달되어야 하는 지역이 R개 존재하는 이동에이전트 컴퓨팅 환경에서 홈 지역내 이주가 m개, 다른 지역으로 이주가 n개있을 경우 그 비용은 다음 표와 같다.

	SPC	WSC
메시지 전달 비용	2m+3n	m+2n+R

표 3. SPC 기법과 WSC기법 메시지 비용 비교

SPC 기법에서는 메시지 전달시 하나의 에이전트에 대해 수행하게 된다. 그러나 WSC 기법에서는 작업 식별자로 선택된 다중 에이전트들에게 메시지를 전달할 수 있다. 특히, 본 기법에서는 같은 지역에서 생성된 이동 에이전트에 대해 단지 하나의 메시지를 지역서버에 전달함으로써, 그 지역에서 생성된 이동 에이전트 모두에게 메시지를 전달 할 수 있다. 이주에 따른 메시지 비용을 측정한 결과는 다음과 같았다.

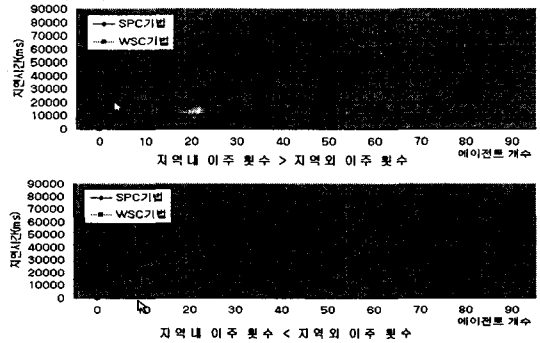

그림 6. SPC기법과 WSC기법의 메시지 전달 비용 비교

현재 제안된 기법의 환경이 특정 작업을 다룬 10개미만의 지역을 다루었으며 따라서, 이주 후 지역서버가 각 지역서버로 보내는 멀티캐스트 비용이 상대적으로 작았다. 그러나, 에이전트가 수행되는 지역의 수가 10개 이상의 많은 지역이 존재 할 경우 멀티캐스트 비용이 상당히 증가하게 된다. 따라서, 새로운 에이전트의 작업을 다룬 위치 관리 기법에 연구가 필요하다.

6. 결론 및 향후 연구

본 논문은 여러 지역으로 구성된 이동 에이전트 컴퓨팅 환경에서 같은 작업을 수행하는 다중 이동 에이전트들에 대한 관리를 위한 위치 관리 기법을 제안하였다. 작업 식별자를 바탕으로 한 다중 이동 에이전트의 위치 관리 기법은 다중 이동 에이전트 관리 시 발생하는 메시지 전달 부하를 감소시킨다.

제안된 본 논문의 기법은 현재 진행중인 "ODDUGI" 프로젝트에 구현하고, 현재 연구 중인 보안 기법과 결합하여 안전한 위치 관리 및 메시지 전달 모델을 수행중에 있다.

참 고 문 헌

[1] Neeran Karnik and Anand Tripathi, "Design issues in Mobile Agent Programing Systems", IEEE Concurrency, pp 52-61, July-Sep 1998.

[2] Dejan Milojicic, Markus Breugst, Ingo Busse, John Campbell, Stefan Covaci, Barry Friedman, Kazuya Kosaka, Danny Lange, Kouichi Ono, Mitsuru Oshima, Cynthia Tham, Sankar Virdhagriswaran and Jim White, "MASIF : The OMG Mobile Agent System Interoperability Facility", In Proceedings of the Second International Workshop on Mobile Agents (MA'98), LNCS 1477, pp 14-15. springer Verlag, September 1998.

[3] Antonella Di Stefano, A. Lo Bello, L; Santoro, C.; "Naming and Locating Mobile Agents in an Internet Environment", Enterprise Distributed Object Computing Conference, 1999. EDOC '99. Proceedings. Third International , 27-30, Page(s): 153 -161 9. 1999

[4] Antonella Di Stefano and Corrado Santoro, "Locating Mobile Agents in a Wide Distributed Environment", IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 8, 2002.

[5] Dwight Deugo, "Mobile Agent Messaging Models", In Proc. 5th International Symposium on Autonomous Decentralized Systems, pp. 278 -286, 2001.

[6] D.B. Lange and M. Oshima, "Programming Mobile Agents in Java - with hte Java Aglet API". Addison-Wesley, 1998.