# 분산 시스템 관리를 위한 에이전트-온-디맨드

설승진*, 이금석
동국대학교 컴퓨터공학과

# Agent-On-Demand for Distributed System Management

Seungjin Sul*, Keumsuk Lee
Department of Computer Engineering, Dongguk University

요 약

분산 시스템이 널리 적용됨에 따라 효율적인 분산 시스템 관리 방안에 관한 연구가 다양하게 진행되고 있다. 그러나 SNMP나 CMIP에 바탕을 둔 클라이언트/서버 방식의 분산 시스템 관리 환경은 확장성, 상호운영성, 신뢰성, 유연성 등과 관련하여 많은 제한점을 드러내고 있다. 근래에는 이러한 단점들을 극복하기 위해 시스템 관리에 이동 에이전트(mobile agent)를 적용하려는 연구가 집중적으로 이루어지고 있으며, 이러한 연구는 자바 언어의 출현으로 가속화되고 있다.

본 논문에서는 관리자 응용(manager application)이 관리 작업을 수행하는 이동 에이전트를 관리 대상 노드에 파견(dispatch)하는 일반적인 방식에서 탈피하여 관리 대상 노드가 관리자 응용에게 특정 이동 에이전트의 파견을 요청하는 에이전트-온-디맨드(Agent-On-Demand) 방법을 제안하고 이를 위한 계층적 상태 임계값(Hierarchical State Threshold)에 대해 설명한다. 또한 성능 분석을 위해 자바 RMI와 이동 에이전트를 위한 분석 모델을 제시하고 AOD를 적용한 분산 시스템 관리 기법과 기존의 방법을 네트워크 부하와 실행 시간 관점에서 비교한다.

## 1. 개 요

분산 시스템은 효율적인 자원 공유, 확장성 및 작업 부하의 분산 등과 같은 장점으로 인해 여러 분야에 적용되고 있다. 하지만 분산 시스템의 규모가 방대해지고 더욱 다양한 서비스들이 제공됨에 따라, 뚜하지 않은 성능 저하나 자원 할당의 비효율성 그리고 물리적으로 원거리에 위치한 장치들의 고장, 보안 문제 등 심각한 문제들이 속출하고 있다. 이러한 문제들을 해결하려는 노력으로 분산 시스템 관리 방안에 관한 연구가 활발히 진행되고 있으며, 주로 SNMP(Simple Network Management Protocol)나 CMIP(Common Management Information Protocol) 등과 관련된 연구가 진행되었다[1].

그러나 SNMP나 CMIP를 기반으로 하는 중앙집중형 분산 시스템 관리는 단순한 폴링(polling)을 기반으로 하며, 중앙의 관리자 응용이 관리 대상에 대한 관리 정보의 수집 및 분석부터 제어 연산의 실행까지 모든 작업을 처리하는 것이 일반적이다. 이러한 관리 방식은 중앙 관리자 응용으로 처리 부하가 집중될 뿐만 아니라 네트워크 통신량을 상당히 증가시키는 결과를 초래한다. 더욱이 관리자의 관리 작업이나 정책 등을 쉽게 변경할 수 없기 때문에 확장성, 상호운영성, 신뢰성 및 유연성 등을 저하시키는 문제점을 갖고 있다[2].

이러한 중앙집중형 시스템 관리의 문제점을 해결하기 위해 관리자 응용의 관리 기능을 관리 대상 노드로 위임(delegation)하려는 연구와 함께 이동 에이전트 개념을 시스템 관리에 적용하기 위한 노력이 이루어지고 있으며, 주로 네트워크 관리 분야에서 많은 성과를 이루고 있다. 이동 에이전트 기반 시스템 관리는 관리상의 복잡성을 완화시켜줄 수 있을 뿐만 아니라 에이전트의 이동성 및 지능성을 이용하면 관리 시스템의 확장성 및 호환성을 증대시킬 수 있다[2][3][4].

그러나 분산 시스템 관리를 위한 이동 에이전트는 각종 정책과 기능을 포함하므로 그 크기가 상당히 커지며, 기능 면에서의 복잡성 또한 증가하므로 실행 시간도 지연된다. 이러한 대규모 이동 에이전트를 분산 시스템을 구성하는 모든 노드로 파견하는 것은 관리 대상 노드의 시스템 자원을 지나치게 낭비하게 된다.

따라서 본 논문에서는 계층적 상태 임계값(Hierarchical State Threshold ; 이하 HST)을 이용하는 에이전트-온-디맨드(Agent-On-Demand ; 이하 AOD) 방법을 제안한다. 제안한 방법은 관리자 응용이 관리 작업을 수행하는 이동 에이전트를 모든 관리 대상 노드에 파견하는 일반적인 방식에서 탈피하여 HST를 유지하고 있는 관리 대상 노드가 특정 에이전트의 파견을 요청하도록 하고, 해당 관리 대상 노드를 관리하기에 충분한 기능만을 가진 에이전트를 여행 계획(travel plan)을 이용하여 파견한다. 성능 분석을 위해서 자바 RMI와 이동 에이전트를 위한 성능 모델을 제시하고, 제안한 AOD의 성능 모델과 비교하였다.

본 논문의 구성은 2장에서 이동 에이전트의 모델, 3장에서 제안한 AOD에 관해 설명한다. 4장에서는 성능 모델을 제시하며, 5장에서 성능 평가, 6장에서 결론 및 향후 연구를 기술한다.

## 2. 이동 에이전트 모델

### 2.1 이동 에이전트의 정의

이동 에이전트는 적용 분야에 따라 약간의 차이는 있지만 다른 개체(entity)의 역할을 대행하는 소프트웨어 프로그램이라는 공통된 개념을 갖는다. 또한 이동 에이전트는 어느 정도의 자치성(autonomy)을 가지며 명세된 행위의 수행(proactivity)뿐만 아니라 발생되는 사건에 따라 적절한 반응 조치(reactivity)를 행할 수 있어야 한다.

### 2.2 이동 에이전트의 분류

이동 에이전트는 에이전트의 이동을 요청하는 주체에 따라 원격 실행(remote execution)과 코드-온-디맨드(code-on-demand) 형태로 구분되며, 이동 에이전트가 유지하는 상태 정보에 따라 강한 이주(strong migration)와 약한 이주(weak migration) 형태로 구분된다.

[5]

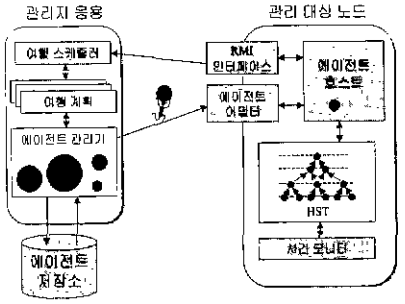
분산 시스템 관리 측면에서 살펴보면 원격 실행은 관리자 응용이 특정 이동 에이전트를 관리 대상 노드에 파견하는 방식이며, 코드-온-디맨드는 에이전트를 받아들이는 관리 대상 노드에서 에이전트의 파견을 요청하는 형태를 의미한다. 그리고 강한 이주는 이동 에이전트의 이동시 에이전트의 코드, 데이터 및 상태 정보 모두를 목적지 노드로 이전시키는 형태인 반면, 약한 이주는 단지 데이터와 관련된 상태 정보만을 이전시킨다

## 3. 에이전트-온-디맨드

제안한 AOD는 각각의 관리 대상 노드에 관련된 관리 작업 모두를 처리하기 위한 대규모 에이전트가 이동하면서 시스템 자원을 낭비할 필요가 없으며, 동시에 이동 에이전트의 관리 기능이 변경되어야 하는 조건의 발생 빈도가 비교적 낮다는 사실에 바탕을 두고 있다. 관리 대상 노드는 관리자 응용에게 더 많은 혹은 더 적은 기능을 가진 에이전트의 파견을 요청하기 위해 HST를 유지하며, 관리자 응용은 각각의 관리 대상 노드의 요청을 기반으로 파견할 에이전트의 여행 계획을 생성한 후, 해당 에이전트를 파견한다

### 3.1 AOD 시스템의 구성

(그림 1)은 AOD 시스템의 구성을 나타낸 것으로 전체 분산 시스템 관리 시스템 가운데 HST 유지와 에이전트 파견과 관련된 부분만 나타내었다. 관리자 응용의 에이전트 관리자는 시스템 관리자에 의해 작성된 관리 에이전트의 저장 및 파견을 담당한다. 여행 스케줄러는 관리 대상 노드의 요청을 바탕으로 각 에이전트에 대한 여행 계획을 조정한다. 한편 에이전트 호스트는 에이전트 관리기를 통해 파견된 에이전트를 전송 받아 관리 시스템에 적재 및 초기화하고, 사건 모니터에 의해 구성된 HST를 접근하여 다른 에이전트의 전송을 요청하는 작업을 수행한다 에이전트 호스트는 관리자 응용의 여행 스케줄러와 RMI 인터페이스를 통해 통신하게 된다.



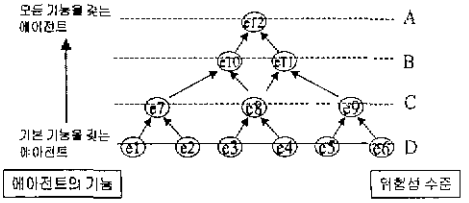

(그림 1) 에이전트-온-디맨드 시스템의 구성

### 3.2 계층적 상태 임계값

각 관리 대상 노드가 유지하는 HST는 시스템 관리자에 의해 작성된 관리 정책[6]을 바탕으로 구성된다. 관리 대상 노드의 상태 정보는 위험성 수준에 따라 4 가지 단계로 관리된다 위험성 수준은 안전-임계 시스템(safety-critical system)의 위험 수준(risk level)을 바탕으로 구성하였다[7] 이러한 등급을 바탕으로 구성된 HST는 이동 에이전트에 의해 참조되며, 명세된 사건의 발생 여부나 발생 빈도에 의해 상위 위험성 수준 혹은 하위 위험성 수준으로 조정되어 에이전트 업그레이드(agent upgrade) 혹은 에이전트 다운그레이드(downgrade)를 야기하게 된다 (그림 2)는 HST와 이동 에이전트의 기능 및 위험성 수준 사이의 관계를 나타낸다.

### 3.3 여행 계획

AOD 시스템에서 여행 계획의 초기값은 모든 관리 대상 노드를 방문하도록 구성한다. 여행 계획은 에이전트 종류별로 유지되며, 시작 노드와 일련의 방문 노드의 튜플로 구성된다 AOD 시스템에서 모든 이동 에이전트는 반드시 관리자 응용으로 복귀하는 것을 가정한다. 예를 들어, 분산 시스템이 관리자 응용 $m$과 관리 대상 노드 $a1$, $a2$, $a3$, $a4$, $a5$로 구성되었다면 일대일로 방문하는 경우의 여행 계획은 $\{(m, a1), (m, a2), (m, a3), (m, a4), (m, a5)\}$로 표현하며, 모든 관리 대상 노드를 링 형태로 방문할 경우의 여행 계획은 $\{(m, a1, a2, a3, a4, a5)\}$로 표현한다




(그림 2) HST와 위험성 수준

## 4. 성능 모델

### 4.1 자바 RMI의 성능 모델

자바 RMI는 대부분의 이동 에이전트 시스템의 구현 언어가 자바이고, AOD를 이용한 분산 시스템 관리에서 각 관리 대상 노드의 HST 상태를 관리자 응용으로 전송하기 위해 사용되기 때문에 성능 모델을 고려할 필요가 있다. 자바 RMI의 성능 모델은 기존의 RPC 성능 모델[8]과 달리 원격 메쏘드 호출시에 RMI 서버로부터의 클라이언트 스텁 전송 오버헤드를 포함해야 한다 한편 RMI 서버 바인딩 시간과 요청에 대한 실행 시간은 무시하여, 성능 모델이 통신 관련 요소만으로 구성되도록 하였다

임의의 노드 $L_1$에서 $L_2$로 각각의 크기가 $B_{req}$인 요청, $B_{rep}$인 응답, 그리고 $B_{stub}$인 스텁으로 구성된 자바 RMI 호출에 대한 네트워크 부하(바이트 크기) $B_{RMI}$는 다음과 같이 나타낼 수 있다.

$$B_{RMI}(L_1, L_2, B_{req}, B_{rep}, B_{stub}) = \begin{cases} 0, & \text{if } L_1 = L_2 \\ B_{req} + B_{rep} + B_{stub}, & else \end{cases}$$

따라서 실행 시간 $T_{RMI}$는 다음과 같다

$$T_{RMI}(L_1, L_2, B_{req}, B_{rep}, B_{stub}) = 4\delta(L_1, L_2) + \left(\frac{1}{\tau(L_1, L_2)} + 4\mu\right) B_{RMI}(L_1, L_2, B_{req}, B_{rep}, B_{stub})$$

여기서 $\delta$는 노드간 네트워크 지연 시간, $\tau$는 노드간 네트워크 처리율, $\mu$는 요청, 응답 및 클라이언트 스텁의 전송준비 처리율을 나타낸다. 앞으로의 모델에서 $B_{RMI}$와 $T_{RMI}$는 각각 상수 $C_b$와 $C_t$로 표기한다.

### 4.2 이동 에이전트의 성능 모델

이동 에이전트는 코드, 데이터, 상태 정보로 구성되므로 $B_{AGENT} = (B_{code}, B_{data}, B_{state})$로 표현할 수 있다[8] 4.1절의 자바 RMI 성능 모델과 동일한 가정 하에서 에이전트 $B_{AGENT}$가 노드 $L_1$에서 $L_2$로 이동하는 경우, 네트워크 부하는 다음과 같이 나타낼 수 있다

$$B_{MOBILE}(L_1, L_2, B_{AGENT}) = \begin{cases} 0, & \text{if } L_1 = L_2 \\ B_{code} + B_{data} + B_{state}, & else \end{cases}$$

그러므로 이동 에이전트의 실행 시간은 다음과 같다

$$T_{MOBILE}(L_1, L_2, B_{AGENT}) = \delta(L_1, L_2) + \frac{B_{MOBILE}(L_1, L_2, B_{AGENT})}{\tau(L_1, L_2)} + \begin{cases} 0, & \text{if } L_1 = L_2 \\ 2\mu(B_{code} + B_{data} + B_{state}), & else \end{cases}$$

### 4.3 AOD를 적용한 경우의 성능 모델

우선 비교 대상인 AOD를 적용하지 않은 경우와 AOD를 적용한 경우 각각에 대해 초기 상태를 살펴본다. 비교를 간소화하기 위해 이동 에이전트의 이동 방식을 링 형태의 순환 방식으로 고정하며, 모두 $n$개

의 노드로 이루어진 분산 시스템을 $m$번 순환한다고 가정한다. 또한 AOD에서 요청할 수 있는 이동 에이전트의 종류는 $B_{A1}'$와 $B_{A2}'$ 등 2가지로만 제한한다.

AOD를 적용하지 않은 경우에는 4.2절에서 정의한 네트워크 부하와 실행 시간 모델을 준수하지만, AOD를 적용할 경우 네트워크 부하와 실행 시간 모델은 다음과 같다

$$B_{AODfirst}(S,D,B_{A_1}') = \sum_{i=1}^{n}(B_{MOBILE}(D_{i-1},D_i,B_{A_1}')) + Pn \cdot C_b$$

$$T_{AODfirst}(S,D,B_{A}') = \sum_{i=1}^{n}(T_{MOBILE}(D_{i-1},D_i,B_{A_1}')) + Pn \cdot C_l$$

여기서 $S$는 관리자 응용, $D$는 여행 계획에 기술된 이동 대상 노드를 의미한다 한편 $P$는 에이전트 업그레이드 확률을 의미한다. 즉, 매 순환마다 $n$개의 노드중 $Pn$개의 노드가 에이전트 업그레이드를 요청한다는 $z$것을 의미한다

AOD를 적용한 경우, 두 번째 이동 에이전트 순환에 대한 네트워크 부하와 실행 시간 성능 모델은 다음과 같다.

$$B_{AODsecond}(S,D,B_{A_1}',B_{A_2}') = \sum_{i=1}^{(1-P)n}(B_{MOBILE}(D_{i-1},D_i,B_{A_1}')) + \sum_{j=1}^{Pn}(B_{MOBILE}(D_{j-1},D_j,B_{A_2}')) + Pn \cdot C_b$$

$$T_{AODsecond}(S,D,B_{A_1}',B_{A_2}') = \sum_{i=1}^{(1-P)n}(T_{MOBILE}(D_{i-1},D_i,B_{A_1}')) + \sum_{j=1}^{Pn}(T_{MOBILE}(D_{j-1},D_j,B_{A_2}')) + Pn \cdot C_l$$

그러므로 전체 관리 대상 노드를 $m$번 순환할 경우, 네트워크 부하와 실행 시간 모델은 다음과 같다.

$$B_{AODtotal}(S,D,B_{A_1}',B_{A_2}') = B_{AODfirst}(S,D,B_{A_1}') + \sum_{i=2}^{m}\left(\sum_{j=1}^{(n-Pn-Q_{i-1})}(B_{MOBILE}(D_{j-1},D_j,B_{A_1}')) + \sum_{k=1}^{(Pn+Q_{i-1})}(B_{MOBILE}(D_{k-1},D_k,B_{A_2}')) + Pn \cdot C_b\right)$$

$$T_{AODtotal}(S,D,B_{A_1}',B_{A_2}') = T_{AODfirst}(S,D,B_{A_1}') + \sum_{i=2}^{m}\left(\sum_{j=1}^{(n-Pn-Q_{i-1})}(T_{MOBILE}(D_{j-1},D_j,B_{A_1}')) + \sum_{k=1}^{(Pn+Q_{i-1})}T_{MOBILE}(D_{k-1},D_k,B_{A_2}')) + Pn \cdot C_l\right)$$

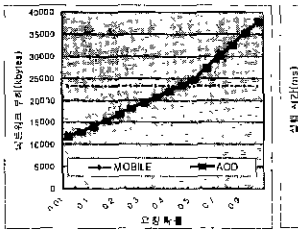
여기서 $Q_{j-1}$은 $j-1$번째 순환에서 업그레이드를 요청한 노드의 개수, $P_{j-1}n$을 의미한다

## 5. 성능 평가

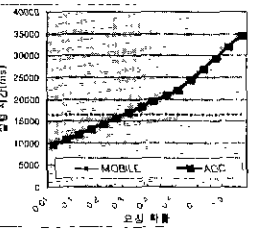
성능 평가를 위해 임의의 두 노드 사이의 네트워크 지연시간 $\delta$는 10ms, 네트워크 처리율 $\tau$는 400kbytes/s로 고정하고 이동 에이전트의 크기 $B_A$는 93kbytes로 하였다 AOD를 적용할 경우 기본 이동 에이전트의 크기인 $B_{A1}'$는 $B_A/2$인 46 5kbytes, 업그레이드된 에이전트의 크기 $B_{A2}'$는 $B_A$와 동일한 93kbytes로 하였다. 또한 자바 RMI의 요청과 응답 크기는 1kbyte. 그리고 자바 RMI 스텁의 크기는 10kbytes로 가정하였다.

우선 전체 노드 중 에이전트 업그레이드를 요청하는 비율을 나타내는 $P$와 네트워크 부하 및 실행 시간과의 관계에 대한 평가 결과는 (그림3)과 (그림4)와 같다. 그림에서와 같이 네트워크 부하의 경우 $P$=0.45, 그리고 실행 시간의 경우 $P$=0.3을 기준으로 제안한 AOD 방법의 성능이 우월함을 보여주며, $P$가 작을수록 효율성이 높아짐을 알 수 있다. 다시 말해서 업그레이드 요청 비율이 30% 미만일 경우에는 기존 방법에 비해 성능이 개선됨을 알 수 있다.

노드수의 증가에 대해서도 거의 유사한 결과를 나타내었으며, 이동 에이전트의 순환 횟수 증가에 대해서는 AOD를 적용한 경우의 성능이 월등히 나아짐을 알 수 있었다 또한 AOD에서 HST 유지 비용이라 할 수 있는 RMI 연산의 부하에 대한 평가 결과는 RMI 연산 부하가 어느 정도 증가하여도 AOD의 성능 개선이 가능함을 보여주었다

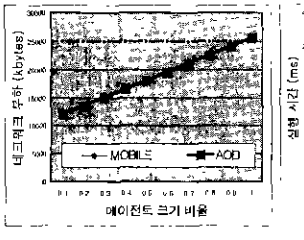



(그림 3) P와 네트워크 부하

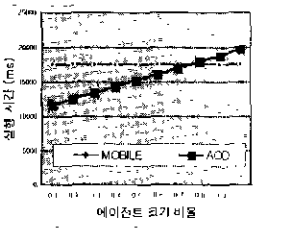



(그림 4) P와 실행 시간

한편 AOD에서 이동 에이전트의 크기 결정을 위한 평가에서 $P$를 0.2로 고정하고 이동 에이전트의 크기 비율을 살펴본 결과는 (그림5)와 (그림6)과 같다




(그림 5) 에이전트 크기 비율과 네트워크 부하




(그림 6) 에이전트 크기 비율과 실행 시간

즉, AOD를 적용할 경우, 업그레이드를 위한 에이전트의 크기를 약간만 줄이더라도 많은 성능 향상을 기대할 수 있음을 알 수 있었다

## 6. 결론 및 향후 연구

중앙집중형 분산 시스템 관리는 확장성, 상호운영성, 신뢰성 및 유연성 등의 관점에서 문제점을 드러내고 있으며, 이에 따라 이동 에이전트 개념을 분산 시스템 관리에 접목시키려는 노력이 이루어지고 있다 본 논문에서는 계층적 상태 임계값(HST)을 바탕으로한 에이전트-운-디맨드(AOD) 방법을 제안하였다. 제안한 AOD는 성능 평가 결과 네트워크 부하와 실행 시간 관점에서 성능이 개선됨을 알 수 있었다

향후 분석 모델의 증명과정이 수반되어야 하며, 성능 개선안으로 이동 에이전트로부터 행위 부분을 분리하는 방법에 대해 연구가 진행될 것이다.

[참고 문헌]

[1] Alexander Keller, "System Management with Distributed Objects: Porting SNMP Agents to a CORBA Environment," Proc. of the 4th Workshop of the OpenView University Association OVUA'97, 1997
[2] Hosoon Ku, et al, "An Intelligent Mobile Agent Framework for Distributed Network Management," Globecom'97, 1997.
[3] Gatot Susilo, et al, "Infrastructure for Advanced Network Management based on Mobile Code," Carleton Univ, 1997
[4] Gottfried Luderer, et al, "Network Management Agents Supported by a Java Environment," ISINM'97, 1997
[5] J.Baumann, et al, "Mole-Concepts of a Mobile Agent System," Technical Report TR-1997-15, Stuttgart Univ, 1997
[6] 유용국, 심승진, 이금석, "SGML과 RDBMS를 이용한 분산 시스템 관리 정책 프레임워크," 한국정보과학회 *98 봄 학술발표 논문집, 1998, pp158-160
[7] Neil Storey, "Safety-Critical Computer Systems," Addison Wesley, 1996.
[8] Markus StraBer and Markus Schwehm, "A Performance Model for Mobile Systems," Proc of the Int. Conf on Parallel and Distributed Processing Techniques and Applications PDPTA'97, 1997.