

# 이동 에이전트를 이용한 대규모 분산 시스템 관리를 위한 이동 모델 설계

유 응구\*, 이 금석  
동국대학교 컴퓨터 공학과

## A Design of Migration Model for Large-Scale Distributed System Management using Mobile Agent

EungGu You\*, KeumSuk Lee  
Department of Computer Engineering, Dongguk University

### 요 약

최근 들어 통신 기술이 급속하게 발전해감에 따라 대규모 분산 시스템에 대한 요구가 증가하였다. 이런 시스템들은 구성 요소들이 이질적이고, 각 구성 요소들이 다양한 정보를 생성하기 때문에 이들의 관리에는 많은 양의 사건 전송량(event traffic)을 생성한다. 따라서 이런 시스템들을 효율적으로 관리하기 위해서는 발생한 사건들에 대한 여과(filtering)가 필요하다. 또한 관리 서비스를 동적으로 변경할 수 있고, 확장성을 제공할 수 있으며, 네트워크 부하를 줄일 수 있어야 한다.

이러한 목적으로 분산 시스템 관리에 이동 에이전트를 사용하는 방법이 연구되어 왔다. 그러나 관리해야 할 관리 대상의 수가 많고 복잡하기 때문에 기존의 이동 에이전트 방법으로는 효율적인 관리가 어렵다. 따라서 본 논문에서는 효율적인 이동 에이전트의 이동 모델을 설계하고, 기존의 이동 에이전트의 이동 방법과 제안한 방법을 응답시간 관점에서 비교한다.

### 1. 개 요

오늘날 인터넷, 인트라넷 등과 같은 통신 기술이 발전해감에 따라, 대규모 분산 시스템에 대한 요구가 증가하였다. 대규모 분산 시스템은 높은 가용성, 확장성, 생산성, 자원 공유 등의 많은 장점을 제공한다[1]. 그러나 대규모 분산 시스템의 분산된 성질과 다수의 상호작용 때문에 신뢰성과 성능이 중요한 문제로 제시되고 있으므로 대규모 분산 시스템의 신뢰성과 성능 향상을 위해 효율적인 관리가 필요하다.

지금까지의 분산 시스템 관리 방법은 크게 두 가지로 분류된다. 첫 번째로는 분산 시스템의 관리 노드에 위치하고 있는 관리 응용이 SNMP(Simple Network Management Protocol)나 CMIP(Commin Management Information Protocol)와 같은 관리 프로토콜을 기반으로 하여 관리 대상에 대한 정보를 폴링(polling)방식으로 수집하고, 처리하는 중앙집중형 관리이다. 관리 노드에 위치한 관리 응용에게 모든 기능이 집중되기 때문에 확장과 관리 서비스의 동적인 변경이 어렵다. 두 번째로는 이러한 중앙집중형 관리의 문제점을 보완하기 위하여 이동 에이전트 적용하는 방법이 있다[2][3]. 이동 에이전트를 적용한 분산 시스템 관리의 관리 노드에 위치한 관리 응용의 작업을 이동 에이전트를 통하여 관리 대상 노드에 위임할 수 있어서 관리 응용에게로 집중되는 작업 부하를 분산하여 처리할 수 있다. 또한 관리 작업이 관리 대상 노드에서 처리되어 결과만이 전송되기 때문에 네트워크 부하 감소, 관리 서비스의 동적인 변경, 이기종 네트워크간의 관리 등의 장점을 제공한다[2][3].

그러나 대규모 분산 시스템 관리는 이동 에이전트의 처리 능력뿐 아니라 이동 방법에 따라 많은 성능의 차이가 발생하므로 이동 에이전트의 이동 방법에 대한 연구가 진행되어 왔다[3][4]. 따라서 본 논문에서는 이동 에이전트의 효율적인 이동을 통하여 향상된 성능을 갖는 분산 시스템 관리 모델을 설계하고, 기존의 모델과 응답시간 측면에서 비교한다.

본 논문의 2장에서는 분산 시스템 관리 방법에 대한 관련 연구를 살펴보고, 3장에서는 제안한 이동 모델을 설명하며, 4장에서는 제안한 이동 모델의 성능을 평가해본다. 5장에서는 결론 및 향후 연구를 살펴본다.

### 2. 관련 연구

이동 에이전트 기반 관리에서는 이동 에이전트가 관리 응용으로부터 권한을 위임받아 관리 대상 노드에 이동하여 관리 작업을 수행한다. 따라서 관리 작업이 지역적으로 처리됨으로써 네트워크 부하가 감소되고, 관리 대상들이 추가되면 이동 에이전트가 관리 대상 노드로 이동하여 관리하기 때문에 확장이 쉽고, 이동 에이전트의 관리 서비스 변경을 통해 관리 대상 노드의 관리 방법을 손쉽게 변경할 수 있다.

이러한 이동 에이전트가 관리 대상 노드들을 이동하는 방법은 다음의 (그림1)과 같다[3]. 첫째는 (그림1.A)에서 보는 바와 같이 중앙의 한 관리 응용에서 각각의 측정 대상으로, 다시 측정 대상에서 중앙으로 이동 에이전트가 이동하는 방법이다. 둘째는 (그림1.B)처럼 중

양의 관리 응용에서 시작하여 링을 구성하면서 이동 에이전트가 이동하는 방법이다.

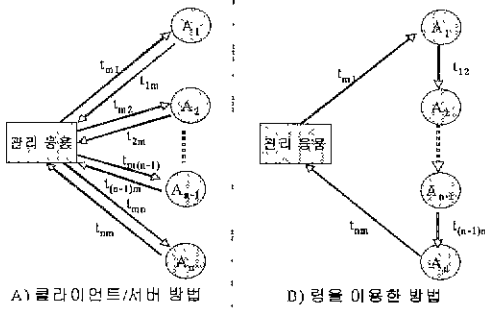


그림1 기존의 이동 에이전트 이동 방법

각각의 응답 시간은 측정 대상 사이의 네트워크 지연시간이 같다고 가정하였을 경우 다음과 같다[3].

$$R_{C/S} = 2nt + na + \beta$$

$$R_{Ring} = (n+1)t + a + \beta$$

여기서  $a$ 는 관리 응용의 지역 계산 시간,  $\beta$ 는 에이전트들이 소요한 작업시간,  $t$ 는 분산 노드간 네트워크 지연시간,  $n$ 은 전체 관리 노드 수를 의미한다

제시된 두 모델의 성능을 비교하면  $R_{C/S}$ 의 성능보다  $R_{Ring}$ 이 우수하다고 한다[3]. 이동 에이전트들은 관리 대상 노드에서 관리 작업을 수행하고, 이동할 때 작업 결과를 유지해야 한다. 그러나 (그림 1.B)의 이동 에이전트는 관리 대상 노드들을 이동할 때 결과의 누적을 고려하지 않으며, 분산된 노드간 네트워크 지연시간을 같다고 가정하였다. 정확한 성능을 비교하기 위해서는 이동 에이전트의 실제 이동량과 이동량에 따른 네트워크 지연시간이 고려되어야 한다

### 3. 제안 모델의 구성

실제로 관리 대상 노드 수가 많아지면 링 방식의 이동 에이전트는 이동할 때 유지해야 할 관리 작업 결과의 누적으로 노드간을 같은 네트워크 지연시간으로 이동할 수 없다. 그러므로 위의 두 모델  $R_{C/S}$ 와  $R_{Ring}$ 에서 네트워크 지연시간  $t$ 는 동일하지 않기 때문에  $R_{Ring}$ 의 정확한 응답시간을 구하려면 작업 결과의 누적으로 발생하는  $t$ 의 변동을 고려해야 한다. 관리 작업 결과의 응답을 다음 관리 노드로 이동하기 전에 매번 전송하는 혼합 모델이 (그림2 A)에 제시되고 있다. 이 모델에서는 빈번한 응답으로 인하여 성능의 저하가 발생할 수도 있다.

본 논문에서는 혼합 모델이 작업 결과를 에이전트가 이동하기 전에 관리 응용에게 매번 결과 응답을 보냄으로써 빈번한 응답을 발생시키고, 전체적인 응답 시간도 증가시킴으로 (그림2 B)처럼 결과 응답 누적의 임계값을 설정하고 임계값까지 응답을 누적하였다가 전송하는 방법을 제안한다.

이동 에이전트의 성능 모델에서 이동 에이전트의 작업부하  $B_{Agent}$ 는 다음과 같다

$$B_{Agent} = B_{Code} + B_{Data} + B_{State}$$

$B_{Agent}$ 가 대상 노드  $L_1$ 에서  $L_2$ 로 이동하는 경우 네트워크 부하(network load)  $B_{Mobile}$ 과 이동시간  $T_{Mobile}$ 은 다음과 같다[4].

$$B_{Mobile}(L_1, L_2, B_{Agent}) =$$

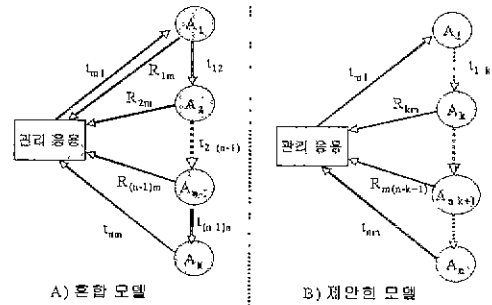


그림2 혼합 모델과 제안 모델의 이동 방법

$$T_{Mobile}(L_1, L_2, B_{Agent}) = \delta(L_1, L_2) + \frac{B_{Mobile}(L_1, L_2, B_{Agent})}{\alpha(L_1, L_2)} + \left( \begin{array}{l} 0, \text{ if } L_1 = L_2 \\ B_{Code} + B_{Data} + B_{State}, \text{ else} \end{array} \right)$$

여기서  $\delta$ 는 노드간 네트워크 지연시간,  $\tau$ 는 노드간 네트워크 처리율,  $\mu$ 는 요청, 응답 처리율이다

노드 이동마다 생성되는 관리 작업 결과의 평균 응답 누적값을  $\epsilon$  이라고 하고, 응답 누적의 임계값을  $\lambda$ 라고 하면 관리 작업 결과의 누적값이 임계값과 같아질 때까지 이동하는 노드의 수는  $\frac{\lambda}{\epsilon}$ 가 된다. 전체 관리 대상 노드수를  $n$ 이라고 할 때 모든 노드를 방문하려면 한번 응답할 때까지의 방문 작업을  $\frac{\epsilon}{\lambda} n$ 번만큼 수행하면 된다

혼합 모델과 제안한 이동 모델은 에이전트 코드 부분의 크기와 상대 정보의 크기가 같으므로 성능 모델에서 그러한 요소를 제거하고, 전체 응답시간에서 공통적인 에이전트의 지역 작업시간을 제외한 관리 작업 결과 누적의 응답 시간과 에이전트 이동시간을 합한 에이전트 실행시간을 각각의 성능으로 한다.

혼합 모델의 관리 작업 결과 누적의 응답 시간  $TR_{혼합}$ 과 혼합 모델의 에이전트 이동시간  $TM_{혼합}$ 은 다음과 같다

$$TR_{혼합} = (n-1) \left[ \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) \epsilon \right\} + \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) R \right\} \right]$$

$$TM_{혼합} = n \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) B_{Code} \right\} + \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) (B_{Code} + \lambda) \right\}$$

여기서  $R$ 은 결과 응답에 대한 ACK의 크기이다.

제안한 모델의 관리 작업 결과 누적의 응답시간  $TR_{제안}$ 과 제안한 모델의 에이전트 이동시간  $TM_{제안}$ 은 다음과 같다.

$$TR_{제안} = \left( n \frac{\epsilon}{\lambda} - 1 \right) \left[ \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) \lambda \right\} + \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) R \right\} \right]$$

$$TM_{제안} = n \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) \left\{ B_{Code} + \frac{\lambda - \epsilon}{2} \right\} \right\} + \left\{ \delta + \left( \frac{1}{\tau} + 2\mu \right) (B_{Code} + \lambda) \right\}$$

혼합 모델의 에이전트 실행시간  $T_{혼합}$ 과 제안한 이동 모델의 에이전트 실행시간  $T_{제안}$ 은 다음과 같다.

$$T_{혼합} = TR_{혼합} + TM_{혼합}$$

$$T_{제안} = TR_{제안} + TM_{제안}$$

#### 4. 성능 평가

성능을 비교하기 위해서 이동 에이전트가 관리 응용에서 시작하여 관리 대상 노드들을 모두 방문하고, 결과에 대한 응답을 받을 때까지의 시간, 즉 응답 시간에서 공통적인 에이전트의 지역 작업시간을 제외한 에이전트 실행시간을 계산하였다.

임의의 두 노드간의 지연시간  $\delta$ 는 15ms, 네트워크 처리율  $r$ 는 400Kbytes/sec,  $\mu$ 는 0.0885ms/Kbytes, 에이전트 코드의 크기  $B_{Code}$ 는 10Kbytes, 결과 응답에 대한 ACK의 크기  $R$ 은 1Kbytes,

전체 관리 대상 노드 수  $n$ 은  $\frac{\lambda}{\epsilon}$ 의 정수 배로 가정한다.

혼합 모델과 제안한 이동 모델의 성능 비교를 위해서  $T_{혼합} \geq T_{제안}$  부등식을 전개하면 다음과 같다

$$\frac{4\delta}{\left\{\frac{1}{r} + 2\mu\right\}} + 2R \geq \lambda \quad (식1)$$

(식1)에 값을 대입하여 계산을 하면 제안한 이동 모델은  $\lambda$ 가 24.413Kbytes보다 적은 경우 성능 향상을 기대할 수 있다.

(그림3)은  $n=1000$ ,  $\lambda=24.4$ Kbytes,  $\epsilon$ 을 변화시킨 결과이다.  $\epsilon$ 이  $\lambda$ 보다 적은 경우 기존의 모델보다 제안한 이동 모델의 성능이 우수하였다.

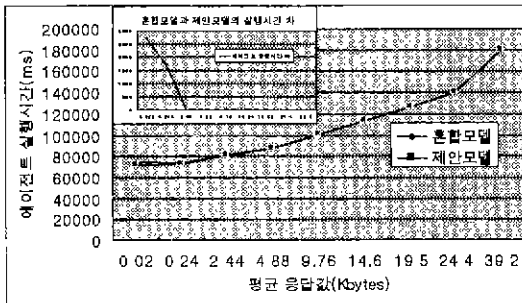


그림3. 평균 응답값과 에이전트 실행시간

(그림4)는  $n=1000$ ,  $\epsilon=2.44$ Kbytes이고,  $\lambda$ 의 값을 변화시킨 결과이다.  $\lambda$ 가  $\epsilon$ 의 값보다 크고, (식1)의 결과 값인 24.413Kbytes보다 적은 경우 기존의 모델보다 제안한 이동 모델의 성능이 우수하였다

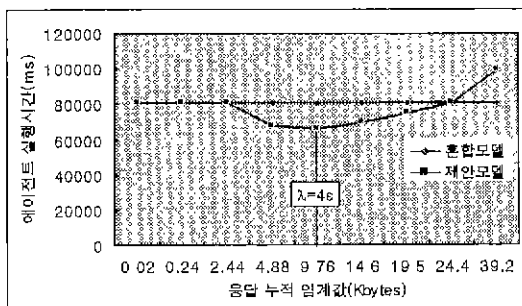


그림4. 응답 누적 임계값과 에이전트 실행시간

$$(K-1) \left[ \frac{4\delta + 2\left(\frac{1}{r} + 2\mu\right)}{K} - \epsilon \left(\frac{1}{r} + 2\mu\right) \right] \quad (식2)$$

(식2)에서  $K$ 는  $\frac{\lambda}{\epsilon}$ 이다.

실험 결과에 의하면 시스템의 성능은  $\lambda$ 값이 (식1)을 만족하는 값이면 성능이 향상되고, (식2)를 최대로 만드는 값  $K$ 가  $K\epsilon \leq \lambda$ 를 만족시킬 때  $\lambda$ 를  $K\epsilon$ 으로 하면 성능을 최대로 향상시킬 수 있다

가정에 의하여  $\frac{\lambda}{\epsilon}$ 이 정수이고,  $n$ 이  $\frac{\lambda}{\epsilon}$ 의 정수배이므로, (그림4)에서  $K$ 의 값은 4이다. 즉  $\lambda$ 를  $4\epsilon$ 으로 하면 성능이 최대로 향상된다.

#### 5. 결론 및 향후 연구

대규모 분산 시스템을 효율적으로 관리하기 위해서는 관리 서비스 동적 변경과 확장성이 제공되어야 한다. 이런 요구를 충족시키기 위해 이동 에이전트가 해결책으로 제시되어 왔다. 하지만 기존의 이동 에이전트를 이용하여 대규모 분산 시스템을 관리하기 위해서는 이동 에이전트의 결과응답의 누적 현상으로 네트워크 지연시간이 증가하는 문제와 빈번한 응답으로 인하여 발생하는 전체적인 응답 시간이 증가하는 문제가 해결되어야 한다.

본 논문에서는 이러한 문제점들을 해결하고자 결과 응답 누적 임계값을 주어 임계값이 될 때까지 응답을 지연하여 빈번한 응답으로 발생하는 전체적인 응답 시간의 증가 문제를 해결하였다. 응답 누적 임계값  $\lambda$ 가 (식1)을 만족하는 값이면 제안한 이동 모델은 기존의 모델보다 우수한 성능을 나타냈다. 또한 (식2)를 최대로 만드는 값  $K$ 가  $K\epsilon \leq \lambda$ 를 만족하는 경우,  $\lambda$ 를  $K\epsilon$ 으로 하면 성능을 최대로 향상시킬 수 있다. 즉, 제안한 이동 모델은 관리 대상 노드들을 이동하면서 작업을 수행하는 이동 에이전트 시스템에서 빠른 관리 결과의 응답을 제공할 수 있다

향후 연구로는 제안한 이동 모델을 검증해 보아야한다. 또한 본 논문에서는  $\epsilon$ 이 일정하고, 관리 대상 노드의 수를  $\frac{\lambda}{\epsilon}$ 의 정수 배로 가정하였지만, 그렇지 않을 경우에 대한 성능 평가가 필요하다.

#### [참고 문헌]

- [1] Jeff Kramer, "Distributed Systems," Network and Distributed Systems Management, Addison-Wesley, 1994 pp47-66.
- [2] Hosoon Ku, et al., "Network Management Agents Supported By Java Environment," IFIP/IEEE International Symposium on Integrated Network Management (IM'97), San Diego CA May 12-16, 1997.
- [3] Hosoon Ku, et al., "An Intelligent Mobile Agent Framework for Distributed Network Management," Globecom'97 Phoenix, AZ. Nov 3-8, 1997.
- [4] Markus Straßer and Markus Schwem, "A Performance Model for Mobile Agent Systems," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'97, 1997