분산 시스템 관리를 위한 효율적인 에이전트 파견 기법

천영석[0] 이금석
동 국 대 학 교 컴 퓨 터 공 학 과
{pookoo[0], kslee}@dgu.ac.kr

An Efficient Strategy of Dispatching Agent for Distributed Systems Management

Young-Suk Chun[0] Keum-Suk Lee
Dept. of Computer Engineering, Dongguk University

요 약

분산 시스템은 원거리에 위치한 자원들을 하나의 시스템으로 통합하여 여러 가지 장점을 제공한다. 하지만 시스템의 규모가 커짐에 따라서 관리자 노드와 관리 대상 노드간의 빈번한 상호작용에 의해서 전체적인 시스템의 성능저하가 발생할 수 있다. 에이전트를 사용한 방법으로 중앙 집중형 관리 방법의 폴링(polling)에 따른 문제점을 해결하였으나 에이전트 사용에 있어서도 에이전트가 이동하는 파견방법에 따라서 성능의 차이가 발생할 수 있다. 즉, 에이전트를 파견하는 관리 대상 노드의 지역 계산 시간에 따라서 관리자 노드에서의 전체적인 응답시간과 네트워크의 부하가 달라질 수 있다. 따라서, 본 논문에서는 에이전트가 관리 대상 노드에서의 지역 계산 시간에 따라 현재 작업중인 노드의 다음 노드로 새로운 에이전트를 파견하여 관리자 노드에서의 응답시간과 네트워크 부하를 줄일 수 있는 방법을 제시하고 기존의 모델과의 평가를 응답시간과 네트워크 부하 측면에서 비교, 평가하였다.

1. 서론

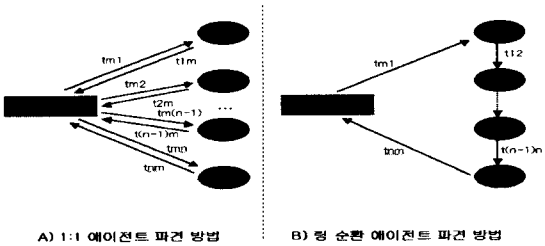
분산 시스템은 원거리에 위치한 자원들을 하나의 시스템으로 통합하여 확장성, 생산성,자원의 공유 등의 많은 장점을 제공한다. 하지만 시스템의 규모가 커짐에 따라서 분산된 시스템간의 빈번한 상호작용에 의해서 성능 저하가 발생하거나 고장을 감지하고 이에 대한 적절한 처리를 해주어야 하는 문제점이 발생 가능 하므로 이를 위한 적절하고 효율적인 관리가 필요하다. 이러한 문제점들을 해결하기 위한 연구가 진행되어 왔으며 크게 두가지 관리방법으로 나눌 수 가 있다. 하나는 SNMP(Simple Network Management Protocol) 나 CMIP(Common Management Information Protocol)같은 관리 규약을 기반으로 해서 관리 대상노드의 정보를 얻기 위하여 폴링 방식을 이용하는 중앙 집중형 관리방식이 있다[1].중앙 집중형 관리방식은 관리대상노드의 관리를 위한 기능이 관리자 노드로 집중되어 확장성 및 관리 서비스의 동적인 변경을 하기 어려운 단점이 있다.두 번째는 이러한 중앙 집중형 관리방식의 문제점을 보완하기 위한 방법으로 이동 에이전트를 이용한 방법이 있다[1][4].관리자 노드에 집중된 관리기능의 일부를 에이전트에게 위임하여 관리대상 노드에서 수집된 관리정보들을 에이전트의 지능성을 이용해서 관리 대상 노드에서 처리한 후 결과를 관리자 노드에게 전송하여 전체적으로 관리자 노드의 부하를 줄여주게 된다.하지만 이러한 이동 에이전트 기반의 방법에서도 에이전트가 관리 대상 노드에서 처리한 작업의 결과를 관리자 노드로 가져오게 되면서 결과의 누적량에 따라서 성능의 차이가 발생할 수 있으며 또한 에이전트의 파견 방법에 따라서 성능의 차이가 발생할 수 있다[1][2][3]. 따라서 본 논문에서는 이동 에이전트의 효율적인 이동을 통한 효율적인 분산시스템 관리 모델을 설계하고,기존의 모델과 응답시간 측면에서 비교한다. 본 논문의 2장에서는 분산 시스템 관리 방법에 대한 관련 연구들을 살펴보고, 3장에서는 제안된 이동 모델을 설명한다. 4장에서는 제안된 모델과 기존 모델의 성능을 비교한다.5장에서는 결론 및 향후 연구를 살펴본다.

2.관련 연구

이동 에이전트를 기반으로 하는 분산 시스템 관리에서 에이전트는 관리자 노드로부터 관리에 대한 기능을 위임받아 관리 대상 노드에서 작업을 하고 그 결과를 가져오게 되므로 네트워크 부하가 감소하고, 관리 대상 노드에 대한 관리자 노드의 관리 서비스가 변경되더라도 에이전트에게 관리 정책을 변경하여 위임하면 되므로 관리 대상 노드에 대한 관리 서비스의 변경이 용이하게 된다.또한 관리 대상 노드가 추가 되더라도 에이전트의 관리 순서만 변경하면 되므로 확장성 측면에서도 장점이 있다.

이렇게 이동 에이전트가 관리 대상 노드들을 방문하는 방법은 (그림1)과 같다[1]. 첫번째 방법은 (그림1.A)와 같이 중앙의 관리자 노드에서 각각의 관리 대상 노드로 에이전트를 파견하고 파견된 에이전트들은 중앙의 관리자 노드로 복귀하는 방법이다.두 번째 방법은 (그림1.B)와 같이 중앙의 관리자 노드에서 각 관리 대상 노드로 논리적인 링을 구성하면서 에이전트가 이동하는 방

법이다.



A) 1:1 에이전트 파견 방법 B) 링 순환 에이전트 파견 방법

그림 1. 기존의 이동 에이전트 이동 방법

(그림1)의 1:1 에이전트 파견 방법과 링 순환 에이전트 파견 방법의 경우 관리자 노드와 관리 대상 노드들 사이의 네트워크 지연시간이 같다고 가정하였을 때 응답시간은 다음과 같다.

$R_{(1:1)} = 2nt + n\alpha + \beta$

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

α = 관리자 노드에서의 지역 계산 시간.

β = 에이전트들이 소요한 작업 시간.

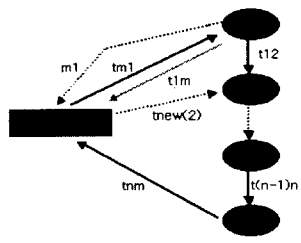
N = 관리해야 할 전체 노드 수.

제시된 두 모델의 성능을 비교하면 $R_{(1:1)}$이 $R_{(Ring)}$ 보다 응답시간이 더 우수하다[1]. 하지만 이동 에이전트들이 관리 대상 노드를 링 방식으로 순환 시 각각의 관리 대상 노드에서 작업한 결과가 누적이 되므로 관리자 노드로 복귀할 때 까지의 결과값의 크기가 변하게 되고 이 크기에 따라서 응답시간의 변화가 생기게 된다. 에이전트가 다음 번 방문할 노드로 이동하기 전에 매번 작업한 결과값을 관리자 노드로 전송하는 방법과 결과값의 임계값을 두어 임계값을 넘으면 전송하는 방법 중 후자의 방법이 더 우수하다[3]. 그리고 에이전트가 링 방식으로 순환을 할 경우 방문한 관리 대상 노드에서 지역 계산이 오래 걸릴 경우 나머지 관리 대상 노드들을 방문해서 작업을 할 시간이 그만큼 지연되어 관리자 노드에서 보면 전체적인 응답시간은 떨어지게 된다. 실질적인 이동 에이전트를 이용한 응용에서는 모든 관리대상 노드들의 작업시간이 동일하다고 가정하는 것은 적합하지 않으므로 관리 대상 노드들의 작업 시간이 다양할 수 있는 상황을 생각해 볼 수 있다.

3.제안 모델의 구성

이동 에이전트가 링 방식으로 이동 할 경우 관리 대상 노드에서 지역 계산 시간이 오래 걸릴 경우 나머지 방문해야 할 관리 대상 노드에서는 에이전트가 올 때 까지 기다리고 있어야 한다. 하지만 관리자 노드에서 새로운 에이전트를 생성해서 지역 계산이 지연되고 있는 관리 대상 노드의 다음 노드로 새로운 에이전트를 파견한다면 관리자 노드에서 볼 때의 전체적인 응답시간은 줄어들게 된다. 이때 고려해야 할 사항은 새로운 에이전트를 파견하는데 따른 추가적인 부하를 줄일 수 있는 경우에 새로운 에이전트를 파견해야 한다는 것이다.

(그림2)에서 n개의 관리 대상노드가 존재할 때 링 방식의 순환을 가정하면 첫번째 방문한 관리 대상노드 A_1에서 작업시간이 T시간 동안 지연되면 관리자 노드에게 새로운 에이전트를 다음 방문할 A_2노드에 파견하라고 메시지 m1을 보내게 된다. 그 후 A_1노드에서는 자신의 관리 대상노드의 순환 계획을 변경하여 작업이 끝나면 관리자 노드로 결과를 갖고 이동한 후 관리자 노드에게 작업결과를 보고하고 자신은 소멸되게 된다. 관리자 노드에서는 메시지를 보내온 관리 대상 노드의 다음 번 방문 대상노드로 새로운 에이전트를 생성하여 파견하게 된다. 이때 A_1에서의 작업시간



m1	관리 대상노드에서 작업이 지연되어 관리자 노드에게 새로운Agent를 파견하라는 요청.
tnew(2)	m1을 전송받은 후 새로운 Agent를 메시지가 온 다음 노드로 전송.
t1m	관리자 노드에게 새로운 Agent파견 요청후 현재 작업 중인 노드에서 관리작업이 끝나면 관리자노드로 결과와 함께 이동.

그림 2. 제안한 모델에서의 이동 에이전트 이동 방법

T를 어떻게 정할 것인지가 중요한 문제가 된다. T시간이 너무 짧으면 새로운 에이전트를 빈번히 파견하게 되며 T시간이 너무 길면 새로운 에이전트를 파견하지 못하고 결국 링 방식의 순환을 하게 된다. 작업중인 관리대상노드에서 불 때 새로운 에이전트를 보내달라는 메시지 m1과 관리자 노드에서 관리 대상노드로 새로운 에이전트를 파견하는데 걸리는 시간 tnew(2)값을 더하여 임계값으로 설정한다. 이때 메시지 메시지 m1을 보내는데 걸리는 시간은 새로운 에이전트를 보내는데 걸리는 시간 tnew(2)보다는 항상 작다고 가정한다.

즉, T = m1 + tnew(2)가 된다..

LAN 환경에서 동일한 네크워크 지연시간을 갖는다고 가정하면 메시지 전송시간과 노드간 새로운 에이전트 파견시간은 측정 가능 하므로 에이전트가 도착하면 측정된 T값 만큼을 기다리는 쓰레드가 생성이 된다. 이때 작업 대상 노드에서 에이전트의 작업이 끝나게 되면 새로운 에이전트는 파견되지 않고 다음 대상 노드로 에이전트가 이동하게 된다. 하지만 쓰레드가 깨어났을 때 에이전트의 작업이 끝나지 않았다면 새로운 에이전트를 파견하게 되며 이때 에이전트의 남은 지역 계산 작업이 T보다 크면 클수록 관리자 측면에서의 전체적인 응답시간은 좋아지게 되지만 T보다 작을 경우는 추가적인 부하가 발생하게 된다.

4.성능 평가

제안 모델의 경우 관리 대상 노드의 지역 작업시간에 따라서 새로운 에이전트의 파견을 결정하므로 관리 대상 노드마다 같은 에이전트가 동일한 작업을 하되 상황에 따라서 지역 작업시간의 편차가 큰 경우에 적용할 수 있다. 즉, 초기에는 링 방식의 순환을 하지만 작업량이 많은 관리 대상 노드에서는 응답시간의 단축을 위하여 새로운 에이전트를 파견하는 것이다. 이렇게 작업량의 편차가 큰 경우는 동일한 작업을 할 때 작업의 자료양이 많을 경우 발생 가능하다. 예를 들어,관리 대상 노드들이 웹 서비스를 하고 있을 경우 시간에 따라서 에러 로그가 발생할 수 있는데 관리 대상 노드의 상황에 따라서 에러 로그는 발생 하지 않을 수도 있거나 아주 많은 양이 발생 할 수도 있다. 성능 평가를 위한 실험에서는 이러한 웹 서비스 에러 로그를 관리 대상 노드마다 정해진 양 만큼이 발생 한다고 가정한다.에이전트의 작업은 관리할 모든 대상 노드를 방문하여 각 노드에서 에러 로그를 분석한 후 에러의 종류와 발생 시각, 횟수 등을 에이전트가 관리자 노드로 가져오게 된다.

성능 평가는 제안모델과 비교모델과의 관리자 노드에서의 전체적인 응답시간과 네트워크 부하를 측정한다. 제안 모델과의 비교를 위한 모델은 관리 대상 노드를 한 개의 에이전트가 방문한 후 다시 관리자 노드로 돌아오고 다시 새로운 관리 대상 노드로 방문하는 방법인 1:1클라이언트 서버방법,$1:1_{C/S}$와 기존 모델인 링 방법(그림1.B) 그리고 모든 관리 대상 노드의 수에 따라서 에이전트를 관리자 노드에서 한꺼번에 생성하고 모든 관리 대상 노드에 멀티 캐스트 방법으로 파견한 후 파견된 모든 에이전트가 관리 대상 노드에서 작업을 마치고 돌아와서 관리자 노드에 결과를 출력하는 방법,$1:1_{multi}$와 비교를 한다. 응답시간의 측정 방법은 4개의 모델 모두 생성되는 에이전트의 수에 상관없이 가장 처음

에이전트를 관리자 노드에서 만든 시간이 작업의 시작 시간이고 모든 관리 대상 노드의 작업을 마치고 온 에이전트가 관리자 노드에 결과를 보여주고 작업을 마치는 시간을 작업의 종료 시간으로 하여 측정, 비교한다.

네트워크 부하는 관리자 노드와 관리 대상 노드간의 에이전트 및 메시지의 전송 그리고 에이전트의 작업 결과를 모두 합한 것으로 측정한다. 성능 평가를 위한 실험환경은 10Mbps 이더넷을 기반으로 LAN을 구성하고 총 7대의 컴퓨터(관리자 노드1대 + 관리 대상 노드 6대)로 구성하였으며 이동 에이전트의 개발환경으로는 IBM사의 Aglets 소프트웨어 개발 키트(Aglets Software Development Kit : ASDK) 버전1.1Beta3를 사용하였고 자바 컴파일러와 자바 가상 기계는 JDK1.1.8버전을 사용하였다.

실험에 사용한 매개변수는 표1과 같으며 실험 결과값은 실험을 총 10번 반복하여 평균값을 이용한다..

관리 대상 노드 수	6대
평균 메시지 전송시간	30(ms)
에이전트 크기	7(Kbytes)
에이전트 평균 전송시간	165(ms)
에러 로그 크기	1,100,250,500, 750,1000(Kbytes)
네크워크 처리율	400Kbytes/sec
노드간 지연시간	15(ms)

표1. 실험에 사용한 매개 변수

첫번째 응답시간의 비교는 관리 대상 노드의 에러 로그 크기가 모두 같은 경우이다(그림3).

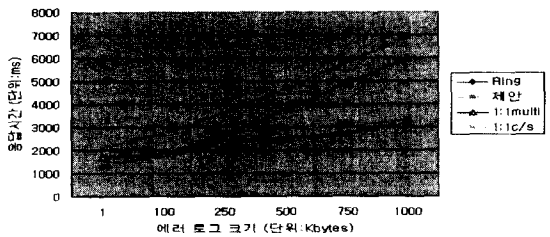


그림3.관리 대상 노드의 에러 로그 크기가 모두 같은 경우

첫번째 실험에서는 에러 로그 크기가 250Kb보다 클 경우 제안 모델에서는 새로운 에이전트가 파견되어 링 방식 보다 응답시간이 좋아지게 된다.

두 번째 실험은 관리 대상 노드의 에러 로그의 크기를 노드의 수에 따라 각각 1000Kb로 증가시키고 나머지 노드는 100Kb로 고정 시켰을 경우이다(그림4).

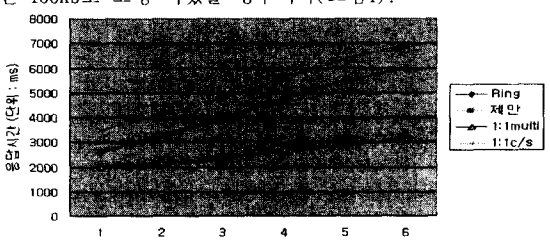


그림4.관리 대상 노드의 에러 로그 크기가 1000Kb인 노드를 증가 시켰을 때의 응답시간

두 번째 실험에서는 에러 로그 크기가 1000Kb인 관리 대상 노드수가 2개까지일 때 $1:1_{multi}$와 비슷한 응답시간을 보였다.

세 번째 실험은 두 번째 실험과 동일한 방법으로 네트워크 부하를 측정한 결과이다(그림5).

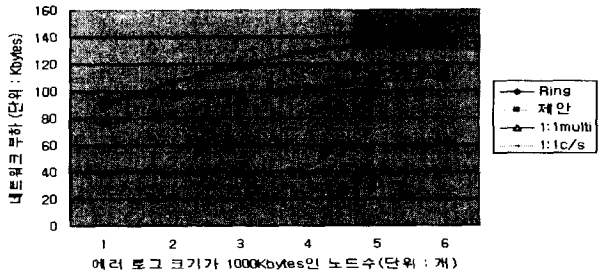


그림5.관리 대상 노드의 에러 로그 크기가 1000Kb인 노드를 증가 시켰을 때의 네트워크 부하

세 번째 실험결과는 에러 로그의 크기가 1000Kb인 노드수가 1개 이하일 때 제안 모델에서의 네트워크 부하가 가장 적었다. 그리고 $1:1_{C/S}$와 $1:1_{multi}$의 네트워크 부하는 동일하였다.

결국, $1:1_{multi}$의 경우 응답시간 면에서는 가장 좋은 결과를 보였지만 지역 계산 시간을 예측할 수 없는 상황에서, 항상 관리 대상 노드 수 만큼 에이전트를 파견해야 하기 때문에 많은 네트워크 부하가 생김을 알 수 있었다. 따라서, 제안 모델의 경우 전체 관리 대상 노드에서 에이전트가 링 방식의 순환을 할 때 지역 계산 시간이 오래 걸리는 노드가 발생할 경우(50%이하) 새로운 에이전트를 파견하는데 따른 부하를 줄이면서 응답시간을 개선하는 장점과 관리 대상 노드에 에이전트가 파견된 후 네트워크 단절 같은 상황이 발생하여 응답시간이 무한대로 증가하는 최악의 상황에서도 나머지 노드의 작업을 계속 할 수 있는 장점이 있음을 알 수 있다.

5.결론 및 향후 연구.

제안 모델의 경우 에이전트가 관리 대상 노드로 파견 되었을 때 관리 대상 노드에 따라서 작업량의 크기가 서로 다른 경우 지역 관리 계산 시간의 편차가 커지게 되어 관리자 노드에서의 전체적인 응답시간이 증가하게 되는 것을 새로운 에이전트를 파견함으로써 줄일 수 있었다. 하지만 제안 모델의 경우 모든 노드에 지역 계산 시간이 클 경우는 성능이 감소함을 알 수 있었다. 제안 모델의 경우 LAN 환경에서 실험을 하였지만 인터넷 환경 하에서의 분산 시스템의 경우 네트워크 부하에 따라서 응답시간에 큰 영향을 미치게 되므로 네트워크 부하가 적은 제안 모델의 경우 인터넷 환경에서 더 좋은 성능을 보일 것으로 예상된다. 향후 연구로는 관리 대상 노드의 수를 증가시켜서 성능을 평가해보고 인터넷 환경에서의 성능 평가도 필요하며 에이전트가 관리자 노드로 메시지를 보낸 후 관리자 노드에서 새로운 에이전트를 파견하는 것이 아닌 파견된 에이전트가 새로운 에이전트를 생성하여 다음 관리 대상 노드로 파견하여 메시지 전송시간을 단축할 수 있는 방법도 연구해야 할 것이다.

6.참고 문헌

[1] Hosoon Ku, et al, " An Intelligent Mobile Agent Framework for Distributed Network Management," Globecom' 97 Phoenix, AZ. Nov 3-8, 1997.

[2] Markus Straβer and Markus Schwehm, " A Performance Model for Mobile Agent Systems," Proc. of the Int. Conf on Parallel and Distributed Processing Techniques and Applications PDPTA' 97, 1997.

[3] 유응구, 이금석, " 이동 에이전트를 이용한 대규모 분산시스템 관리를 위한 이동 모델 설계," 한국정보과학회 *98 봄 학술발표 논문집, 제25권, 제2호, 한국정보과학회, 1998, pp.586-588.

[4] D. Hagimont, L. Ismail, " A Performance Evaluation of the Mobile Agent Paradigm," Proc. OOPSLA'99, Int. Conf. on Object-Oriented Programming, Systems and Applications, November 1999.