분산 시스템 개발을 위한 수행 패턴 결정 알고리즘

유우종, 권혁찬*

대전보건대학 전산정보처리과
*충남대학교 컴퓨터과학과
{wjyoo,hckwon}@nspplab.cs.cnu.ac.kr

A Decision Algorithm of Migration Pattern for Developing Distributed Application

Woo-Jong Yoo, Hyeok-Chan Kwon*
Dept. of Computer Information Processing, Taejon Health Sciences College
*Dept. of Computer Science, Chungnam National University

요 약

분산 시스템 개발을 위해 사용되는 패러다임(paradigm)의 수행능력은 여러 요소들을 종합하여 고려하여 평가해야 한다. 분산시스템 개발 시 사용되는 대표적인 패러다임으로 클라이언트/서버(client-server) 구조의 RPC(Remote Procedure Call)가 있다. 또한 최근 들어서는 이동성(mobility)과 지능성(intelligence) 이라는 특성을 갖고 네트워크 부하(network load)를 감소시킬 수 있는 이동 에이전트에 대한 요구도 증가하고 있다. 그러나 이동 에이전트를 이용하여 개발한 분산 시스템이 기존의 접근 방식에 비해 성능이 좋은 지의 여부는 아직도 의견이 분분하다. 또한 분산 시스템의 성능은 어떤 패러다임을 쓰는가 뿐 아니라, 선택된 패러다임의 수행 패턴에 의해서도 많은 영향을 받는다. 본 논문에서는 RPC 와 이동 에이전트 그리고 locker 패턴이 적용된 이동에이전트의 수행을 평가하기 위한 수행 평가 모델과, 이 모델을 기초로 하는 분산 시스템 개발을 위한 수행 패턴 결정 알고리즘을 제시하고자 한다.

1. 서론

분산 시스템 개발을 위해 사용되는 패러다임(paradigm)의 수행능력은 여러 요소들을 종합하여 고려하여 평가해야 한다. 최근 들어서는 이동성(mobility)과 지능성(intelligence) 이라는 특성을 갖고 네트워크 부하(network load)를 감소시킬 수 있는 이동 에이전트에 대한 요구도 증가하고 있는 실정이다. 그러나 이동 에이전트를 이용하여 개발한 분산 시스템이 기존의 접근 방식에 비해 성능이 좋은 지의 여부는 아직도 의견이 분분하다.

본 논문은 클라이언트/서버(client-server) 구조의 RPC (Remote Procedure Call)와 이동 에이전트의 수행을 평가하기 위한 수행 평가 모델을 제안하고, 이 모델을 기초로 분산 시스템 개발을 위한 수행 패턴 결정 알고리즘을 제시하고자 한다. 대상 애플리케이션으로는 간단한 데이터 마이닝(data mining)을 가정하였다. 분산 시스템의 성능은 네트워크 소요 시간, 각 노드에서의 자원 소요, 보안등의 문제를 함께 고려하여 평가되어야 하지만, 본 연구에서는 네트워크 소요시간만을 고려하였다. 본 논문에서 고려하는 패러다임(paradigm)은 RPC, 이동 에이전트, 그리고 Locker 패턴을 적용한 이동 에이전트이다. 이동 에이전트는 에이전트 자신이 직접 서버로 이동하여 주어진 작업을 수행하는 구조를 갖는다. Locker 패턴[6]은 이동 에이전트의 수행패턴 중 하나로, 에이전트가 방문하는 각각의 노드에서 획득한 결과를 임의의 영역에 보관시키고 다음의 노드로 이동하는 구조이다. 이 때 결과는 또 다른 이동 에이전트를 파견하여 수집하거나, RPC 방식으로 클라이언트에게 전달하게 된다. 본 논문에서는 후자의 경우로 가정하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 수행 평가 모델을 제시한다. 3장에서 이 모델을 기초로 분산 시스템 개발을 위한 수행 패턴 결정 알고리즘을 제안한다. 4장에서는 알고리즘을 적용하여 시뮬레이션 한 결과를 제시하고, 마지막 결론을 5장에서 맺는다.

2. 수행 평가 모델

본 장에서는 3가지 패러다임의 수행 평가모델을 제시한다. 수행 환경은 균등 네트워크(uniform network) 환경으로 가정하였고, 모든 요구 메시지(request message)

는 동일한 크기를 갖는다고 가정하였다. 각각의 노드에 존재하는 다큐먼트의 수와 크기는 다양하다고 가정하였다. <표 1>에서 모델링에 사용되는 파라미터들을 볼 수 있다.

<표 1> 모델링에 사용된 파라미터

파라미터	설명
N	네트워크 노드(node)의 수
M_j	j 노드에 존재하는 다큐먼트의 수
r	document에 찾고자 하는 정보가 있을 확률
req	request 메시지의 크기
rep_{hj}	j 노드에서 반환되는 하나의 헤더의 크기
rep_{dj}	j 노드에서 반환되는 하나의 다큐먼트의 크기
δ	네트워크 지연시간
β	평균 네트워크 대역폭(network bandwidth)
Ma	이동 에이전트의 크기
UR	Unique Rate
NUR	Non Unique Rate
TD	Total Document
VD	Visited Document

TD는 전체 노드에 분포되어 있는 다큐먼트의 총 합을 말한다. VD는 이전에 발견한 다큐먼트의 합이다. UR은 전체 데이터 중 중복된 데이터를 제거한 데이터의 확률이다. 본 논문에서는 노드 내의 중복 데이터는 없는 것으로 가정하였다. j 노드에서 제거되어야 할 중복데이터의 평균적 수는 식 (1)과 같다.

$$NM_j = 0 \quad \text{if } j=1 \qquad (1)$$
$$M_j \times \{VD/(TD-M_j) \times (2 \times NUR)\} \text{ if } j > 1$$

중복 데이터를 판별하기 위해서는 실제 다큐먼트의 텍스트 자체를 비교하거나, 검색하여 수행해야 하지만, 본 논문에서는 헤더의 정보만으로 판별하도록 가정하였다. 만약 다큐먼트 내의 텍스트 내용에 의해 중복 데이터를 판별한다면, 모든 데이터를 갖고 이동하는 이동 에이전트의 경우에 제거시키는 중복 데이터가 더 많을 것이다.

본 논문에서 고려하는 3가지 패러다임 각각의 중복 데이터 제거 방법은 다음과 같다. 먼저 RPC 방식의 경우, 클라이언트 노드에서 서버 노드로부터 받은 헤더 정보를 계속 보관, 유지하므로 중복 데이터의 여부를 판별한다. 서버로부터 헤더 정보를 받으면 먼저 필요로 하는 데이터를 가려낸 후, 이전에 보관한 헤더와 비교하여 중복되는 데이터를 확인한 후에 서버로 필요한 다큐먼트를 요청한다. 이동 에이전트의 경우, 수집된 다큐먼트를 갖고 이동하기 때문에 각 노드에서 중복 데이터의 여부를 판별 할 수 있다. locker pattern의 경우엔, 데이터의 중복 여부를 판별하기 위해, 노드들을 방문할 때 수집된 다큐먼트는 RPC 방식으로 서버로 전송하고, 그 수집된 다큐먼트의 헤더는 이동 에이전트의 데이터 영역에 계속 누적시키며 이동한다.

RPC의 경우 j 노드에서의 네트워크 부하(network load)는 식(2), 네트워크 소요시간(network execution time)은 식 (3)과 같다. UM_j는 중복데이터를 제거한 후의 다큐먼트 수이다.

$$L_{RPC-S} = (1 + rUM_j)req + UM_j(rep_h + rrep_{dj}) \qquad (2)$$

$$T_{RPC-S} = \frac{L_{RPC-S}}{\beta} + (2 + 2UM_j r)\delta \qquad (3)$$

이동 에이전트의 경우 j 노드에서의 네트워크 부하는 식(4), 네트워크 소요시간은 식(5)와 같다.

$$L_{MA} = 2Ma + \sum_{k=1}^{t}(rUM_k rep_{dk}) + rrep_{dj}UM_j \qquad (4)$$

(단, t는 mobile agent가 현재까지 방문한 노드 수)

$$T_{MA} = 2\delta + \frac{L_{MA}}{\beta} \qquad (5)$$

locker 패턴이 적용된 이동 에이전트의 네트워크 부하는 식(6), 네트워크 소요시간은 식 (7)과 같다.

$$L_{MA(L)} = MaH + rUM_j rep_{dj} \qquad (6)$$

$$MaH = Ma + \sum_{k=1}^{t}(rUM_k rep_{hk})$$

(단, t는 locker pattern으로 방문한 노드 수)

$$T_{MA(L)} = \frac{L_{MA(L)}}{\beta} + (1 + UM_j r)\delta \qquad (7)$$

3. 수행 패턴 결정 알고리즘

각각의 패러다임은 파라미터(parameter) 값에 의해서 성능의 좋고 나쁨이 결정된다. 주된 영향을 미치는 파라미터로 네트워크 대역폭, 누적되는 데이터 량, 서버와의 상호작용(interaction) 횟수, 이동 에이전트의 크기를 들 수 있다. 이동 에이전트의 경우 식(4),(5)에서 볼 수 있듯이, 네트워크 연결을 처리하기 위한 네트워크 지연시간은 적은 반면, 이동 에이전트와 누적되는 데이터 량이 많은 문제가 있다. 반면 RPC의 경우는 식(2),(3)에서 볼 수 있듯이, 글로벌 통신 횟수가 많으므로 네트워크 연결을 처리하기 위한 네트워크 지연시간은 많이 소요되는 반면, 네트워크 상에서 이동하는 데이터 량이 적다는 장점이 있다. 따라서 네트워크 대역폭이 좋을수록 이동 에이전트가 RPC에 비해 유리하며, 글로벌 통신 횟수가 적을수록 RPC가 유리할 것이다. 본 논문에서 고려하는 데이터 마이닝에서의 가정에 의하면 글로벌 통신 횟수는 각 노드에 존재하는 다큐먼트의 수에 비례하여 증가한다. 이동 에이전트의 크기는 이동 에이전트와 locker 패턴 모두의 성능에 영향을 미치는 요소이다.

RPC와 이동 에이전트의 경우 다음의 조건이 만족되는 경우 이동 에이전트를 이동하는 것이 효과적일 것이다. 다음의 수식들은 네트워크 delay 시간을 기준으로 정리한 것이며 2장의 수식으로부터 유도된 것이다.

$$T_{RPC-S} > T_{MA}$$
$$rUM_j\delta > \frac{2Ma + AD(t) - (rreq + rep_h)UM_j - req}{\beta}$$

RPC 와 locker 패턴이 적용된 이동 에이전트의 경우 다음의 조건이 만족되는 경우 locker 패턴이 적용된 이동 에이전트를 이동하는 것이 효과적일 것이다.

$$T_{RPC-S} > T_{MA(L)}$$

$$(1+rUM_j)\delta > \frac{MaH-(rreq+rep_h)UM_j-req}{\beta}$$

이동 에이전트와 locker 패턴이 적용된 이동 에이전트의 경우 다음의 조건이 만족되는 경우 이동 에이전트를 이동하는 것이 효과적일 것이다.

$$T_{MA} > T_{MA(L)}$$

$$\frac{AD(t)-Ma}{\beta} > (rUM_j-1)\delta$$

실제 분산 응용 시스템 개발 시 적합한 수행 패턴을 결정하기 위한 알고리즘은 다음과 같다. 노드의 방문 순서는 고정 된 것으로 가정한다. 다음의 알고리즘의 migration function은 2장의 수행 평가 모델과 비교 수식에 의한 함수이다.

```
input : node list, other parameters ....
output : selected_paradigm_list, agent_position_list
set current_agent_position as client_node
for node# from 1 to last_node
  choose paradigm which has minimum execution time by
     migration function
        Add selected_paradigm into selected_ paradigm_ list
          If selected_paradigm =  locker pattern
                update accmulated_header value
        Set current_agent_position as node#
  If selected_paradigm =  MA
                update AD(t) value
       set current_agent_position as node#
End loop;
```

만약 노드 방문 순서가 고정되지 않은 경우, 방문 순서를 변경함으로서 성능을 향상시킬 수 있을 것이다. 노드를 방문 시 총 다큐먼트의 크기가 적은 순서부터 큰 순서로 방문한다면 성능 향상이 가능할 것이다.

4. 실험

본 장에서는 실제 알고리즘을 적용하여 simulation한 결과를 보인다. 실험을 위한 도구로는 VC++ 6.0 을 이용하였다.

시뮬레이션을 위한 파라미터로, 전체 노드의 수 N은 8개, request 메시지의 크기는 50bytes, 헤더를 반환(reply)하는 메시지의 크기는 60bytes, 네트워크 지연시간 δ는 20ms, 네트워크 대역폭은 300kbytes/s, 필요로 하는 정보가 방문하는 노드에 존재할 확률 r은 40%, UR은 80%, 이동 에이전트의 크기는 15KB로 가정하였다. 각 노드에서의 평균 다큐먼트 크기와 다큐먼트의 수는 <표 2> 과 같다.

<표 2> 파라미터 값

node # / parameter	1	2	3	4	5	6	7	8
doc. size (B)	2,000	5,000	10,000	3,000	5,000	15,000	1,500	300
# of doc.	6	7	2	30	50	5	20	40

<표 3>에서는 가정된 시나리오 상에서 RPC만을 이용한 경우와 이동 에이전트만을 이용한 경우, locker 패턴만을 이용한 경우, 알고리즘을 적용한 경우와 역으로 적용한 경우의 총 네트워크 상에서 소요되는 시간을 비교하였다. <표 3>의 결과를 통해, 현재 가정한 시나리오 상에서 알고리즘을 적용하여 결정된 수행패턴이 최소의 네트워크 소요시간을 갖는다는 점을 확인할 수 있다.

<표 3> 네트워크 소요시간 (단위 : ms)

수행패턴	Only Locker	Only MA	Only RPC	algorithm 적용	algorithm 역적용
네트워크 소요시간	2187	3154	2972	2978	1769

5. 결론

본 논문에서는 분산 패러다임의 수행 평가 모델과, 이 모델을 기반으로 하는 분산 시스템 개발을 위한 수행 패턴 결정 알고리즘을 제시하였다. 본 논문에서 제시한 평가모델은 분산 응용 시스템 개발 시 적합한 수행 패턴을 결정하는데 도움이 될 수 있을 것이다. 본 논문에서는 균등한 네트워크(uniform network) 환경만을 고려하였다. 현재 균등하지 않은 네트워크(non-uniform network) 환경을 고려한 모델링에 대한 연구가 진행중이다.

참고문헌

[1] Bic, L. F., M. Fukuda, and M. B. Dillencourt, "Distributed computing using autonomous objects," IEEE Computer," Aug. 1996.

[2] Wooldridge, M. and N. R. Jennings, Agent Theories, Architectures and Languages: A Survey, In Michael Jl. Wooldridge and Nicolas R.Jennings, editor, Intelligent Agent, pp.1-39, Springer-Verlag, Germany, 1995.

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

[4] Carzaniga, A., G. P. Picco, G. Vigna, "Designing Distributed Applications with Mobile Code Paradigms," Proceedings of the 19th International Conference on Software Engineering, Boston, 1997.

[5] Strasser, M., M. Schwehm, "A Performance Model for Mobile Agent Systems," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'97, Volume II, pp. 1132-1140, 1997.

[6] Yariv, A., D. B. Lange, "Agent Design Patterns : Elements of Agent Application Design," Second International Conference on Autonomous Agents (Agents 98), 1998.