

이동 컴퓨팅 시스템에서 노드의 위치 정보 복제 메커니즘

박종선*, 오수열**

* 목포과학대학 컴퓨터정보과

**목포대학교 컴퓨터공학과

e-mail : pajosu@korea.com

Location Information Replication Mechanism of Node in Mobile Computing System

Chong-Sun Park*, Soo-Yeol Oh**

* Dept. of Computer Information, Mok-Po Science College

**Dept. of Computer Engineering, Mok-Po University

요 약

이동 객체의 현재 위치를 갱신하고, 이를 탐색하는 기법은 이동 객체의 수가 많아 이동이 자주 발생 할 때 많은 시간과 비용이 소요된다. 본 논문은 노드의 계층적인 위치 관리 구조상에서 갱신 시간은 다소 증가 하나 탐색 시간은 감소되어 효율적으로 위치 정보를 관리할 수 있는 기법을 제시한다.

본 논문에서 제시하는 위치 기법은 고정 호스트에 이동 객체의 위치를 기록하는 사용자 프로파일을 호출 대 이동 비율에 따라 복제하여 유지함으로써 위치 갱신과 위치 탐색을 효율적으로 수행하는 노드의 계층적 위치 정보를 이용한 복제본 위치 탐색(RLS : Replication Location Search) 기법이다. 이 기법은 자주 이동하는 객체의 위치 정보를 복제함으로써 위치 탐색시에 기존 기법보다 적은 비용으로 보다 빠른 시간 안에 원하는 이동 객체의 현재 위치를 탐색할 수 있는 새로운 위치 관리 기법에 대한 연구이다.

1. 서 론

최근 들어 언제 어디서나 자신이 원하는 정보에 접근하고 상호 통신을 요구하는 정보 처리 대상 업무가 증가하고 있다. 따라서 소형화, 경량화 된 하드웨어와 유·무선 통신 기술을 통합하여 시스템의 물리적 위치에 관계없이 지속적으로 업무를 처리할 수 있는 이동 컴퓨팅 개념이 등장하였다[1].

이동 컴퓨팅은 Radio, 무선 다중 접속 및 이동성 관리 기술의 성숙으로 새로운 정보처리 모델인 이동 컴퓨팅에 집중적인 연구가 진행되고 있다[2]. 본 논문은 이동 컴퓨팅 시스템에서 이동 객체를 지원하는 노드의 계층적 위치 정보 관리의 이동 객체 즉, 이동하는 사용자 또는 클라이언트, 이동 호스트, 휴대용 컴퓨터, PDA 등을 모두 포함하여 이동 객체로 통칭한다[3]. 따라서 이동 객체의 위치 정보를 여러 노드에 복제함으로써 이동 객체를 호출하는 피 호출자가 빠른 시간 내에 용이하게 원하는 호출자의 현재 위치를 발견할 수 있도록 하는 효율적 기법을 제시한다. 본 논문에서는 계층적 구조를 갖는 위치 서버들의 구조에서는 확장 탐색이 기본적인 위치 탐색 기법이 된다. Badrinath 등은 이러한 확장 위치 탐색(ELS : Extend Location Search)기법을 기본으로 분할 정보와 사용자 프로파일 등을 사용함으로써 위치 탐색 비용을 줄이고자 하였다[4].

Weng 등은 그룹 확장 위치 탐색(GELS : Group Extend Location Search)기법으로 셀에서 셀들의 그룹 개념으로 확장 해서 다른 그룹으로 들어갈 때마다 node-ty-node로 탐색 작업을 수행한다[5]. 이 기법은 상당히 갱신 비용을 줄일 수 있으나 여전히 위치 탐색 비용이 높다.

개선된 위치 탐색 기법은 위치 탐색 기법에 의한 피 호출자의 위치 발견은 피 호출자에 관한 정보를 찾기 위하여 계층적인 구조를 가지고 있는 위치 서버를 차례로 탐색하였다[6]. 본 논문에서 기법도 사용자의 프로파일을 복제하여 탐색 시간을 줄이도록 하였다. 그러나 호출자가 위치한 곳에 사용자 프로파일을 복제하기보다는 호출자의 이동 형태에 따라 계층적 구조를 갖는 위치 서버간에 차별화 된 정보를 복제하므로 네트워크의 어느 호출자나 위치 탐색을 빠르게 할 수 있다.

2. 노드의 계층적 복제본 위치 관리 모델

본 논문에서는 객체의 이동성을 고려한 새로운 위치 관리 기법을 제시한다. 본 논문의 위치 관리 기법은 네트워크상의 위치 서버들을 계층적 구조로 연결하며, 계층적 구조를 가진 경우에 데이터를 적절히 복제함으로써 위치 관리를 위한 통신 비용과 탐색 시간을 줄일 수 있다. 이 방식에서 각 위치 데이

터 베이스는 현재 그 노드 밑에 있는 모든 가입자에 대한 위치 정보를 가진다. 그림 1은 위치 서버의 계층적 노드 구조 모델이다.

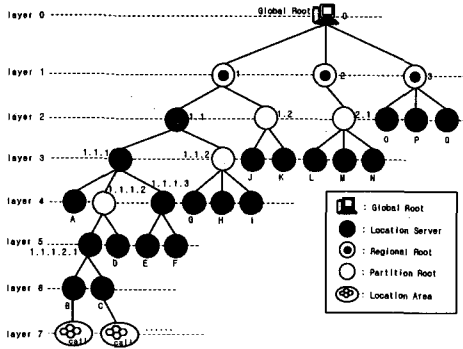


그림 1. 위치 서버의 계층적 노드 구조 모델

본 논문에서 제시하는 복제본 위치 탐색(RLS) 기법은 분할 정보를 사용하는데, 여기서 제시하는 분할 루트(partition root)는 확장 탐색 시 불필요한 위치 서버들을 미리 제외시키는 역할을 할뿐만 아니라 이 분할 루트가 상위 계층의 위치 서버에서 객체에 대한 주소로 복제되어 저장된다. 상위 계층에서 분할 루트에 관한 정보만을 기억하므로 객체들이 분할 내에서 이동한 경우 상위 계층의 위치 서버에서는 갱신될 필요가 없고, 위치 탐색 시에는 객체가 속한 분할 루트의 정보를 이용함에 따라 적은 위치 정보 복제 비용으로 위치 탐색 비용을 감소시킬 수 있다. 계층 구조의 각 노드들은 위치 서버라 불리며, 이들의 계층 구조는 고정 호스트간의 유선 연결로 이루어져 있다. 이들 위치 서버들간에는 지역 루트(regional root)라고 불리는 특수 서버를 지정한다. 그리고 분할 루트가 존재하는데, 이는 각각의 이동 객체에 대해 정의 될 수 있으며, 이동 객체가 자주 움직이는 지역의 대표 위치 서버이다. 만약 어떤 객체가 특정 위치 서버들 사이를 자주 움직이게 되면 이들 위에 가장 낮게 위치한 최소 공통 부모(LCA:Least Common Ancestor)위치 서버들이 분할 루트가 된다.

본 논문이 제시하는 위치 정보 복제의 기본 원칙은 높은 CMR(Call to Mobility Ratio)을 갖는 객체의 정확한 현재 위치 서버가 속한 분할 루트로 복제된다. 낮은 CMR을 갖는 객체의 경우는 현재 위치 정보가 단지 분할 루트에만 존재하고, 분할 루트의 주소는 이전 위치 서버와 현재 위치 서버의 가장 낮은 단계의 공통 서버 사이에만 존재한다. 본 논문이 제안하는 위치 갱신은 위치 주소의 복제로 인하여 저장 장소, 갱신 시간 및 교환 메시지 측면에서 다소 비용이 높아지나 이로 인해 위치 탐색 시간이 감소되어 전체적으로는 기존의 기법보다 낮은 비용으로 위치 정보를 관리할 수 있다.

3. 제안한 계층적 위치 관리 알고리즘

3.1 위치 갱신 알고리즘

위치 갱신 알고리즘은 새로운 위치 서버는 CMR과 이동 유

형에 따라 갱신 절차를 시작한다. 이때 이동 형태는 내부 이동과 외부 이동으로 분류할 수 있으며, 내부 이동은 객체가 현재 속하고 있는 분할 내부에서 움직이는 경우이고, 외부 이동은 분할 밖으로 움직인 경우이다. 위치 갱신 알고리즘은 그림 2와 같다.

```

location_update( );
{
if 자주 호출되는 피 호출자인 경우
/* 즉, high CMR을 갖고 mobility가 적을 때 */
if(old partition root == new partition root)
{
새 위치 서버의 완전한 주소를 분할 루트에
도달할 때까지 부모 위치 서버에 복제;
분할 루트에서 이전 위치 서버에 도달할 때까지
등록 취소 메시지 전달;
}
else
{
두 분할 루트 사이의 LCA를 구함;
새로운 분할 루트에 도달할 때까지 부모에게
새로운 주소를 복제;
LCA에 도달할 때까지 새로운 분할 루트의
주소를 복제;
LCA로부터 이전 분할 루트를 거쳐 이전
위치 서버에
도달할 때까지 등록 취소 메시지 전송;
}
else
/* 즉, low CMR을 갖는 객체인 경우 */
if (old partition root == new partition root)
새 주소를 분할 루트에 복제;
else
{
두 분할 루트의 LCA를 구함;
새 주소를 분할 루트에 복제;
새 분할 루트의 주소를 LCA에 복제;
이전 분할 루트에 등록 취소 메시지를 전송;
}
}
    
```

그림 2. 위치 갱신 알고리즘

3.2 위치 탐색 알고리즘

본 논문이 제안하는 위치 탐색 알고리즘은 그림 3과 같다.

```

location_search( );
{
/* LS=Location Server를 표시 */
if (호출자의 LS == 피 호출자의 홈(home) LS or
가장 최근에 알려진 피 호출자의 LS의 주소)
호출자의 LS 탐색;
if(피 호출자의 주소 탐색 실패)
부모 위치 서버들을 향해 위로 탐색 시작;
if(피 호출자의 주소 탐색 실패)
지역 루트 탐색;
if(피 호출자의 주소 탐색 실패)
전체 루트 탐색;
else
호출자의 LS로부터 부모 위치 서버를 향해 탐색;
if(피 호출자의 주소 탐색 실패)
지역 루트 탐색;
if(피 호출자의 주소 탐색 실패)
전체 루트 탐색;
if(피 호출자의 주소 탐색 실패)
호출자의 홈 LS를 향해
호출자의 LS와 피 호출자의 홈 LS의
LCA에서 탐색 시작;
}
    
```

그림 3. 위치 탐색 알고리즘

본 논문의 특징은 우선 호출자가 현재의 위치 서버와 피 호출자의 홈 위치 서버를 비교한다. 만일 피 호출자의 이전 위치 서버 주소를 호출자가 알고 있다면 이 주소와 호출자의 현재 위치 서버 주소를 비교한다. 비교된 두 개의 위치 서버가 동일하면, 호 지역성으로 인해 호출자와 피 호출자가 가까이 존재할 확률이 있다. 따라서 이때 피 호출자의 위치 탐색은 호출자가 현재 속한 위치 서버부터 시작을 하여 탐색 실패 시에 위 계층의 서버를 탐색한다. 이동 객체가 새로운 위치로 이동할 때 이전 위치 서버와 현재 위치 서버의 LCA노드에 주소가 복제되고, 이 LCA노드는 이전 위치 서버와 루트까지의 경로 사이에 위치가하기 때문이다.

4. 성능 평가 및 검토

본 논문에서 제안한 RLS를 이용한 위치 관리 알고리즘의 비용을 분석하고, Badrinath의 분할을 도입한 단순한 ELS 기법과 Weng 등이 제안한 GELS 기법과 비교하여 성능 평가를 수행한다. 성능 평가에서 사용된 시스템으로는 인텔 펜티엄-IV 2.53GHz이며, 사용된 소프트웨어는 MATLAB 5.3을 이용하여 시뮬레이션 하였다. 사용된 언어는 MATLAB C++을 이용하였다. 성능 평가시 위치 관리의 비용은 위치 데이터 갱신 및 탐색을 위해 접근하는 노드의 수로 측정한다. 성능 평가를 위해 사용한 매개변수의 리스트들은 표 1과 같다.

표 1. 성능 평가를 위해 사용한 매개변수 리스트

TM _i	주어진 시간동안 이동 객체 i가 이동한 횟수
TC _i	주어진 시간동안 이동 객체 i가 받은 호출 횟수
CMR _i	객체 i의 호출 대 이동비
PR _{i, j}	객체 i의 번째 분할 루트
l(s)	노드(위치 서버) s 의 level(root level : 0)
lca(s, t)	노드 s와 t의 최소 공통 분모
MAX(PR _{i, j})	최하위 노드에서 객체 i의 분할 루트 j까지 거리 중 가장 큰 값
h	위치 서버의 트리 높이
α	각 노드 내 데이터 베이스에서의 위치 탐색 비용
β	각 노드 내 데이터 베이스에서의 위치 갱신 비용
δ(s, t)	노드 s에서 노드 t까지의 메시지 전송을 위한 네트워크 비용

본 논문이 제안한 위치 갱신 작업은 위치 정보의 복제로 인하여 기존의 다른 기법에 비해 조금 더 복잡한 위치 등록과 취소 절차를 거치게 된다. 위치 갱신 알고리즘은 객체의 이동이 분할 내부 이동인지 여부에 따라 각 각 다른 갱신 비용을 갖는다. 복제가 이루어지는 내부 이동의 경우 갱신 비용 Update_{cost}는 새 위치 등록 비용 Registration_{cost}와 이전 위치 등록 취소 비용의 합으로 식 (1)과 같다.

$$Update_{cost} = Registration_{cost} + Deregistration_{cost} \quad (1)$$

Registration_{cost}은 현재의 위치 서버와 분할 루트의 거리에 따라서 결정된다. β는 하나의 위치 서버에서 위치 데이터 갱신 비용, n은 새로운 위치 서버, l(n)은 새로운 위치 서버의 레벨, o는 이전 위치 서버, l(o)는 이전에 속해 있던 위치 서버의

레벨, l(PR_{i, j})는 피 호출자 i의 새로운 분할 루트 j의 레벨인 경우에 등록과 취소 비용은 식 (2)와 같다.

$$Registration_{cost} = \beta \cdot \{l(n) - l(PR_i, j)\} + \delta(PR_i, j)$$

$$Deregistration_{cost} = \beta \cdot \{l(o) - l(PR_i, j)\} + \delta(PR_i, j, o) \quad (2)$$

객체의 이동이 외부 이동인 경우 새 위치 등록 비용과 이전 위치 등록 취소 비용은 식 (3)과 같다.

$$Registration_{cost} = \beta \cdot \{l(n) - l(lca(n, o))\} + \delta(n, lca(n, o))$$

$$Deregistration_{cost} = \beta \cdot \{l(o) - l(lca(n, o))\} + \delta(n, lca(n, o)) \quad (3)$$

따라서 최하위 노드는 leaf 이고, 루트 노드는 root일 때 식 (4)와 같이 최악의 갱신 비용을 바운드 할 수 있다.

$$Update_{cost} \leq 2 \cdot \beta \cdot h + 2 \cdot \delta(\text{leaf}, \text{root}) \quad (4)$$

본 논문이 제안하는 위치 탐색은 복제본을 사용함으로써 비교적 빨리 이루어질 수 있다. 호출자가 피 호출자의 분할 루트 하위에 위치 서버가 위치하는 경우 탐색 비용은 피 호출자의 위치 정보가 호출자의 위치 서버에 복제된 경우 통신비용 없이 한 번의 탐색만으로 피 호출자의 주소 탐색은 Search_{cost} = α이다.

피 호출자의 위치 정보가 중복되어 있지 않고 분할 루트에 서만 탐색할 경우 분할 루트까지의 통신비용과 그 사이 노드들의 탐색 비용이 필요하다. 중간 노드에서 탐색이 필요한 이유는 호출자가 피 호출자의 분할 루트에 관한 정보를 가지고 있지 않기 때문에 어떤 노드가 위치 정보가 복제된 피 호출자의 분할 노드인지 모르기 때문이다. 따라서 caller가 호출자의 위치 서버 PR_{i, j}가 피 호출자의 현재 분할 루트라고 할 때, 탐색 비용은 식 (5)와 같다.

$$Search_{cost} = \alpha \cdot \{l(\text{caller}) - l(PR_i, j)\} + \delta(\text{caller}, PR_i, j) \quad (5)$$

따라서 호출자가 피 호출자의 분할 루트 하위에 속하는 위치 서버에 위치하는 경우 탐색 비용은 식 (6)과 같다.

$$Search_{cost} = [\alpha, \alpha \cdot \{l(\text{caller}) - l(PR_i, j)\} + \delta(\text{caller}, PR_i, j)] \quad (6)$$

호출자 위치 서버와 피 호출자의 홈 위치 서버가 같은 지역 루트에 속한 경우 탐색 비용은 식 (7)과 같다.

$$Search_{cost} \leq \alpha \cdot l(\text{caller}) + \delta(\text{caller}, \text{root}) \quad (7)$$

상기 식은 지역 루트까지 탐색하는 동안에 피 호출자의 위치 정보 복제본을 발견하기 때문이다. 또한 호출자 위치 서버(caller)와 피 호출자의 홈 위치 서버(p_caller)가 같은 지역 루트(regional root)에 속하지 않는 경우 탐색 비용의 바운드는 식 (8)과 같다.

$$\begin{aligned}
 Search_{cost} &\leq a \cdot l(\text{caller}) + l(\text{p_caller}) \\
 &- 2 \cdot l(\alpha(\text{caller}, \text{p_caller})) \\
 &+ \delta(\text{caller}, \alpha(\text{caller}, \text{p_caller})) \\
 &+ \delta(\text{p_caller}, \alpha(\text{caller}, \text{p_caller}))
 \end{aligned} \quad (8)$$

본 논문이 제시하는 기법은 전체 위치 서버의 개수에 관계 없이 위치 서버 트리의 높이에 바운드 되어 트리의 높이보다 넓이가 훨씬 큰 경우에 더욱 효율적으로 동작된다. CMR이 높은 이동 객체의 경우에 중간 노드에 복제가 이루어진 경우 평균 위치 갱신 비용과 탐색 비용을 성능 평가하는데 있어서 위치 정보 복제를 하지 않은 경우와 비교한 것이다. 본 논문에서 제시한 비교 기법은 기존에 Badrinath가 제안한 ELS기법과 Weng 등이 제안한 GLES 기법 등이 그 대상이다. 이 기법들은 위치 서버를 분할하고 복제는 하지 않는 기법이다.

성능 평가를 위하여 어떤 객체에 대한 분할 루트의 집합이 이미 주어 졌다고 가정하고, 이때 분할 루트는 최하위 위치 서버로부터 거리는 3을 넘지 않고, 위치 서버 트리의 최대 높이 (h)는 10을 넘지 않는다고 가정하였다. 이러한 가정을 만족하도록 하고, 전체 위치 서버의 크기를 하나씩 증가시키면서 객체가 한번 이동 시 모든 위치 서버에서 이동한 객체에 대한 호출이 발생하도록 하여 성능 평가를 하였다.

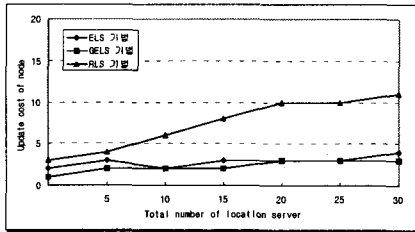


그림 4. 위치 갱신시 노드의 평균 개수

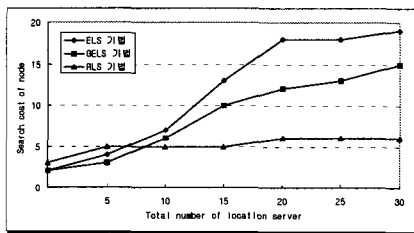


그림 5. 위치 탐색시 노드의 평균 개수

위치 갱신 비용은 객체가 한번 이동하여 복제본이 생성될 때 새로운 복제본을 포함하여 위치 갱신을 위해 접근하는 노드의 수 즉, 위치 서버의 개수를 측정한 갱신 비용의 결과는 그림 4와 같고, 탐색 비용은 이 객체에 관한 위치 정보를 발견하기까지 탐색해야 하는 노드의 수를 측정한 결과는 그림 5와 같다. 본 논문의 위치 관리 기법에서 복제 여부를 결정하기 위한 객체의 CMR 값에 따라 위치 서버 트리의 상위 계층에 위치 정보를 복제한다. 자주 호출되는 객체의 위치 정보는 보다

여러 곳에 복제하여 탐색을 빨리 하도록 하면 전체 비용 면에서 유리하고, 자주 호출되지 않는 객체의 위치 정보는 보다 덜 복제하여 갱신 비용을 줄이는 것이 유리하다.

이동 객체의 주소를 빠르게 탐색하기 위해 계층 구조에서 위치 서버들간에 주소를 복제하는 문제에 관하여 다루었으나, 여기서 제시된 기법은 객체가 자주 이동하는 위치 서버의 집합을 분할하여 이들 분할 루트를 복제함으로써 불필요한 위치 갱신 메시지를 줄이고, 적은 횟수의 노드 탐색으로 객체의 위치에 관한 정보를 찾을 수 있도록 한 것이다.

5. 결론

본 논문에서는 이동 컴퓨팅 환경 하에서 위치 탐색을 위한 메시지의 교환을 줄이기 위해 CMR 값을 근거하여 노드의 계층적 사용자의 위치 정보 복제 관리 기법을 제시하였다. 이동 객체의 주소를 빠르게 탐색하기 위해 계층 구조에서 위치 노드간에 주소를 복제하는 기법을 제시하였다. 본 논문이 제시하는 기법은 불필요한 위치 갱신 메시지를 줄이고, 적은 횟수의 노드 탐색으로 객체의 위치에 관한 정보를 찾을 수 있게 하였다. 향후 연구 방향은 이동 객체가 이동성 의한 여러 가지 제약 조건을 가지고 있으므로 이동 트랜잭션을 처리를 위한 데이터의 일관성을 보장하는 노드간의 데이터 복제본 관리에 대한 연구를 수행하고자 한다.

참고문헌

- [1] G. H. Forman, and J. Zahorjan, "The Challenges of Mobile Computing," *IEEE Computer Magazine*, 27(4), pp.38-47, April 1994.
- [2] Chander Dhawan, *Mobile Computing : A system Integrators Handbook*, McGraw-Hill : International Edition, 1997.
- [3] S. Y. Yi, H. Shin, "A new locating scheme for mobile objects based on their mobile characteristics and replication of location information," *Proceedings of the 2nd International Mobile Computing Conference*, pp. 123-130, March 1996.
- [4] B. R. Badrinath, T. Imielinski, A. Virmani, "Locating Strategies for Personal Communication Networks," *Workshop on Networking of Personal Communications Applications*, December 1992,
- [5] C. M. Weng, P. W. Huang, "Modified group method for mobility management computer communications," 23, pp.115-122, 2000.
- [6] D. Lam, J. Jannink, D. Cox, J. Widom, "Modeling location Management in personal Communication Services." *Proceedings of the International Conference on Universal Personal Communications*, September 1996.