

# GALIS 프로토타입을 위한 위치 데이터 관리 기법

이 호\*, 이준우, 나연욱  
단국대학교 전자컴퓨터공학과  
{hlee\*,jwlee,ymnah}@dmlab.dankook.ac.kr

## An Improved Location Data Management Scheme for GALIS Prototype

Ho Lee\*, Joonwoo Lee, Yunmook Nah  
Dept. of Electronics & Computer Engineering, Dankook University

### 요 약

최근의 위치 측위 기술과 무선 통신 기술의 발전에 따라 위치 기반 서비스에 대한 관심이 크게 증가하고 있다. 기존의 단일 노드 기반 시스템으로는 처리하기 힘든 휴대폰 사용자와 같은 대용량의 객체를 처리하기 위해 제시된 클러스터 기반 분산 컴퓨팅 구조로 GALIS 아키텍처가 제안되었다. 본 논문에서는 제시한 프로토타입은 그동안 구현된 기존 GALIS 프로토타입보다 개선된 구조로 SLDS에서 Global Cell\_ID를 적용하여 노드의 분할 합병 시 발생할 수 있는 처리 비용을 감소시켰다. 또한 LLDS에서는 필터링을 통해 손실될 수 있는 위치 데이터 정보를 보다 신뢰할 수 있는 데이터로 만들기 위한 기능을 추가하여 질의 시 발생할 수 있는 여러 가지 상황을 대비할 수 있게 되었다.

### 1. 서 론

최근의 GPS로 대표되는 위치 측위 기술과 무선 통신 기술의 비약적인 발전, 그리고 이동 단말기의 대중화로 인하여 위치 기반 서비스(LBS: Location Based Service)에 대한 관심이 크게 증가하고 있다. 이에 대한 많은 연구가 수행되었지만, 대부분의 연구가 단일 노드를 대상으로 하여 휴대폰 사용자 위치 추적 같은 위치 정보 갱신이 빈번하면서 최소 백만 단위 이상의 대용량 이동 객체를 처리하는데 어려움이 있었다.

이러한 문제를 해결하기 위하여 제안된 GALIS(Gracefully Aging Location Information System) 아키텍처는 클러스터 기반 분산 컴퓨팅 구조로 설계되어 각 지역별 데이터를 여러 노드에 저장함으로써 위치 정보의 저장과 갱신 및 질의에 대한 부하를 분산시켜 대용량의 데이터를 처리할 수 있으며, Time-zone 이라는 개념을 도입하여 오래된 데이터일수록 큰 정밀도를 요구하지 않는다는 전제 하에 각 시간대 별로 다른 정밀도를 설정하여 과거 위치 정보를 필터링하여 저장 공간 활용의 효율성 증대와 빠른 검색이 가능하도록 하였다[1,6,7].

본 논문에서는 현재의 GALIS 구조에서 데이터베이스에서 필터링 개념 적용한 뒤 다양한 질의 시 발생할 수 있는 필요 이상의 프로세싱 과정을 줄이고자 저장구조를 개선하여 설계하였고, 처리 과정을 디자인 하였다.

GALIS의 구조에 대해 2장에서 설명하고, 3장에서는 SLDS, 4장에서는 LLDS에서의 위치 데이터 저장과정을 기술하고 5장에서는 결론과 향후 연구 과제에 대해 설명한다.

### 2. 관련연구

#### 2.1 GALIS의 구조

GALIS는 크게 객체의 현재 위치 정보를 처리하는 SLDS(Short-term Location Data Subsystem)와 과거 위치

정보를 처리하는 LLDS(Long-term Location Data Subsystem)으로 나누어 진다.

#### 2.1.1 GALIS 아키텍처의 공간 인덱싱

GALIS에서는 공간 영역에 매크로 셀(macro-cell)과 마이크로 셀(micro-cell)로 불리는 2단계의 격자 구조(2-level grid structure)를 사용하고 있다[2,5].

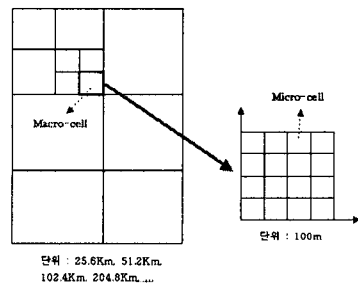


그림 1. 위치 데이터 인덱싱을 위한 2단계 격자 구조

1단계의 매크로 셀은 한 변의 기본 길이 단위가 25.6Km 이고 그의 배수로 늘어날 수 있는 정사각형의 셀이다. 하나의 매크로 셀에는 하나의 프로세서가 할당되어 셀 내의 이동 객체 정보를 관리하게 된다. 각 매크로 셀들은 마이크로 셀들로 분할된다. 2 단계의 마이크로 셀은 한 변의 기본 길이가 100m로 고정되어 균등한 크기를 갖는 정사각형의 셀이다.

#### 2.1.2 SLDS

SLDS내의 각 노드를 SDP(Short-term Data Processor)라고 하며 현재의 위치 정보만을 다루게 된다. SLDS는 메인 메모리 데이터베이스를 사용함으로써 현재와 관련된 질의 수행 시 보다 빠르게 응답할 수 있고 빈번한 위치정보 갱신에 효과적으로 대응할 수 있다. 다수의 SDP가 다른 객체를

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 지원사업의 연구결과로 수행되었음

의 현재 위치 정보를 갱신하기 위해 동시에 동작하게 된다. SDP master는 실시간으로 축위된 이동 객체의 위치 정보를 받아 다른 SDP worker로 전달하고, 새로운 위치정보가 들어오게 되면 과거위치 정보는 LDP(Long-term Data Processor) master로 전달하는 역할을 한다. LDP master로 과거 위치정보를 전달할 때에는 이동객체의 마이크로 셀을 벗어난 과거의 위치정보를 전달한다.

2.1.3 LLDS

LLDS는 SLDS와 유사하게 다수의 LDP들로 구성된다. LLDS는 디스크 기반 데이터베이스를 사용하여 객체의 과거 위치 정보를 경과된 시간대에 따라 네 개의 Time-zone에 나누어 유지하게 된다. 만약 한 객체가 100m x 100m의 마이크로 셀 내에서 이동을 했다면 SLDS에서는 현재 위치 정보를 갱신하는 반면 LLDS에서는 갱신하지 않는다. 이는 SLDS에 비해 LLDS에서의 위치 정보 갱신 빈도가 낮음을 의미한다.

2.2 TMO scheme

제안된 시스템에서는 실시간 분산 처리를 위하여 TMO (Time-triggered Message-triggered Object) scheme을 사용하였다. TMO 구조는 실시간 분산 처리에 대해 신뢰성을 제공하며 개발자에게 분산 처리 구현에 대한 부담을 적게 해주며 노드의 확장을 비교적 간단하게 처리할 수 있다 [3,4]. TMO 구조의 특징을 간단히 살펴보면 다음과 같다.

- 미리 지정한 시간 조건에 의해 구동하는 SpM (Spontaneous Method)와 메시지 전달에 의해 구동되는 SvM(Service Method)의 두 가지 메소드를 제공한다.
- BCC(Basic Concurrency Control) : SpM과 SvM이 동시에 내부 데이터에 접근시 SpM에 우선 순위를 둔다.
- 각 TMO는 독립적이며 다른 TMO와 통신할 수 있는 RMMC(Real-time Multicast and Memory replication Channel)를 제공한다.

3. SLDS 위치정보 처리

3.1 전역 마이크로 셀 ID를 사용한 공간 인덱싱

Location Sensor로 부터 전달 받은 위치 정보는 그림 2와 같은 Z-Ordering 알고리즘을 수행하여 이동 객체의 좌표에 해당하는 마이크로 셀 ID를 할당 받는다.

0	1	4	5	16	17	20	21
2	3	6	7	18	19	22	23
8	9	12	13	24	25	28	29
10	11	14	15	26	27	30	31
32	33	36	37	48	49	52	53
34	35	38	39	50	51	54	55
40	41	44	45	56	57	60	61
42	43	46	47	58	59	62	63

그림 2. Z-Ordering을 통한 셀 ID 부여

이때 특정 매크로 셀마다 지역적인 마이크로 셀 ID를 부여할 수도 있고, GALIS 시스템이 적용해야 하는 전체 지역을 기준으로 하는 전역 마이크로 셀 ID를 할당할 수도 있다. GALIS는 특정 매크로 셀을 담당하는 하나의 프로세스에 부하가 몰리게 되면 분할을 시도하는데, 이때 지역 마이

크로 셀 ID를 적용했다면, 분할과 동시에 모든 마이크로 셀 ID를 갱신해야 하는 상황이 발생하는데 이는 데이터베이스 및 전체 시스템에 큰 부하를 불러일으킬 수 있다. 이 때문에 전체 지역을 Z-Ordering을 통해 전역 마이크로 셀 ID를 부여하면 노드의 분할/합병시 저장된 레코드들을 갱신하지 않아도 되게 된다.

3.2 SDP에서의 현재 위치정보 처리과정

축위된 위치정보는 SDP master에게 전달되어 공간 인덱싱을 통해 자신이 속한 셀의 ID를 부여 받고 좌표와 함께 MMDB에 저장하게 된다. 이를 위한 테이블 스키마는 표 1과 같다.

표 1. SLDS 이동 객체 테이블 스키마

OID	X_position	Y_Position	Time	Cell_ID
Int	Float	Float	Datetime	Int

기 저장된 위치 정보는 다음 주기에 들어오는 위치정보로 갱신되는데 과거 위치정보를 관리하는 LLDS의 LDP master에게 마이크로 셀 영역을 벗어난 정보만을 필터링하여 전달한다. 이러한 필터링을 통해 회사원이나 주부와 같이 이동 범위가 작은 패턴을 갖는 객체의 데이터를 주기적으로 전부 저장하지 않고, 일부만 저장하기 때문에 저장 데이터의 수를 줄일 수 있다.

4. LLDS 위치정보 처리과정

SLDS는 전술한 바와 같이 마이크로 셀 영역을 벗어난 이동에 대해서만 LLDS에게 보고하게 되는데 LLDS는 필터링된 데이터를 보고 받음으로 인하여 몇 가지 문제점이 발생하게 된다.

그림 3에서 같이 LLDS의 1번 매크로 셀을 담당하는 1번 노드(독립된 프로세서로서 LDP master나 worker)가 OID(Object ID) 3번 이동 객체의 위치를 한시간 전에 보고 받고 현재 다시 보고 받았다고 했을때, 실제 OID 3번 이동 객체의 움직임은 다음 두 가지 경우로 요약할 수 있다.

- ① 1시간 전부터 직전까지 마이크로 셀 영역을 벗어나지 않았다가 현재 보고받은 위치로 변경된 경우(그림 3(b)).
- ② 1시간 전에 보고받은 위치에서 다른 매크로 셀의 영역으로 이탈하였다가, 현재시점에서 다시 담당하는 매크로 셀 영역으로 진입한 경우(그림 3(a)).

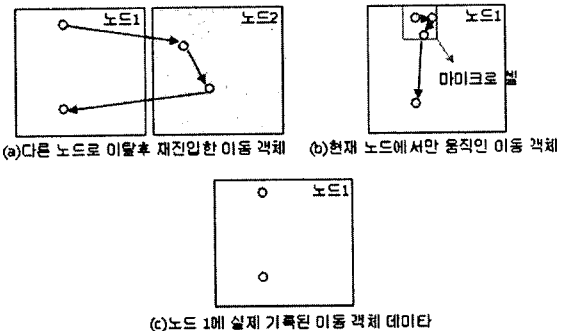


그림 3. 객체의 이동상황 및 기록 결과

매크로 셀을 담당하는 각각의 노드는 다른 노드와 협업하

지 않는 이상 OID 3번 객체의 움직임을 파악할 수 없다. 그림 3(c)와 같이 저장되어 관리되는 경우 이동 객체가 현재 노드를 벗어나서 다른 노드로 이동한 시간을 알 수 없으며 이는 여러 가지 다양한 질의에 신뢰성 있는 결과를 반환하기 위해 많은 비용이 발생할 수 있는 요인이 된다.

예를 들면 노드1의 특정지역에 30분 전에 위치했던 이동 객체들을 질의하면 이달 시간이 관리 되지 않는 이상 OID 3번 이동객체가 질의한 특정지역에 위치했는지 아닌지를 다른 노드들과 협업해야만 정확한 결과를 얻어낼 수 있게 된다.

이 문제를 해결하기 위해 LLDS의 매크로 셀을 담당하는 노드들은 보고받은 위치 정보에 진입시간과 이탈시간을 표 2와 같이 추가적으로 저장하여 관리함으로써 전술한 문제점을 상대적으로 적은 비용으로 해결 할 수 있다.

표 2. LLDS 이동 객체 테이블 스키마

OID	X_position	Y_posiotion	Cell_ID	Start_time	End_time
Int	Float	Float	Int	Datetime	Datetime

LLDS에서 진입, 이탈시간을 저장 관리하기 위해서는 각각의 노드들은 자신이 관리하는 매크로 셀에 관계된 이동 객체뿐만 아니라 자신의 매크로 셀에서 벗어나는 이동 객체에 대하여서도 추가적인 작업을 해야 한다. 즉, 자신의 노드에 위치했던 이동 객체가 다른 노드로 이동했는지를 판단하여, 자신의 노드에서 이탈한 시간을 기록해야 한다.

LDP master는 보고 받은 위치 정보를 worker들에게 브로드캐스팅하게 되는데, 만약 매크로 셀 단위의 이동이 있다면 브로드캐스팅할 전송 레코드에 매크로 셀의 변화 여부 플래그를 활성화 하여 worker들로 전송한다. 따라서 LDP master는 이전 주기 전체 이동 객체에 대한 매크로 셀 ID를 유지해야 하며 추가적인 메모리 자원이 필요하게 된다(그림 4).

```

0: // MO (Moving Object)
1: recieve_location_report()
2: while( every MO in location report ) {
3:   caculate_macro_cell_ID(MO)
4:   if ( is_macro_cell_ID_changed(MO) == true )
5:     MO.macro_cell_changed_flag = true
6: }
7: broadcast_location_eport()
    
```

그림 4. LDP master 의사코드

```

0: // MO (Moving Object)
1: receive_location_report()
2: while( every MO in location report ) {
4:   if ( MO.macro_cell_changed_flag == true ) {
5:     if ( is_exist_buffer(MO) ) {
7:       insert_database(MO,end_time)
8:       remove_buffer(MO)
9:     }
11:  } else {
12:    if ( is_in_my_macro_cell(MO) ) {
13:      if ( is_exist_buffer(MO) ) {
14:        insert_database(MO,end_time)
15:        update_buffer(MO)
16:      } else
17:        add_buffer(MO)
18:    }
19:  }
20: }
21: }
    
```

그림 5. LDP worker 의사코드

worker 노드들은 master로부터 브로드캐스팅된 위치 정

보를 수신할 때 자신의 매크로 셀 영역에 해당하는 이동 객체뿐만 아니라, 신규 보고된 이동 객체의 위치가 자신의 매크로 셀에 해당하지 않더라도 마지막에 자신의 매크로 셀에 해당했다면 이탈정보를 추가로 기록해야 한다.

노드에 진입한 이동 객체를 데이터베이스에 기록할 때 이탈 시간을 알 수 없으므로, 메모리 버퍼상에 기록한다. 추후 노드를 이탈하는 시점에 데이터베이스에 기록하는 기법을 사용함으로써 하나의 이동 객체를 기록하기 위해서 중복되는 Disk IO를 감소시킬 수 있다(그림 5).

### 5. 결론 및 향후 연구

본 논문은 GALIS구조에서 발생할 수 있는 몇 가지 문제를 해결하기 위해 인덱스 구조에서 마이크로 셀 적용을 지역적으로 적용하는 것이 아니라 GALIS 전체에 전역으로 적용시키는 방법을 제안하여, 노드의 분할 및 합병 시 발생할 수 있는 시스템의 부하를 줄일 수 있도록 하였다. 또한 이동 객체에 몇 가지 데이터를 추가하여 필터링을 적용하면서도 정확한 질의 결과를 반환하도록 할 수 있도록 하였다.

향후 본 논문에서 제안한 구조를 기존의 GALIS 구조에 적용시켜 성능상의 개선 정도를 확인하는 연구가 필요하다.

### 참고문헌

- [1]Yunmook Nah, K.H. (Kane) Kim, Taehyung Wang, Moon Hae Kim, Jonghoon Lee, Young Kyu Yang, "GALIS: A Cluster-based Scalable Architecture for Location-Based Service Systems," Database Research, 18(4), KISS SIGDB, December 2002, pp.66-80.
- [2]Yunmook Nah, Moon Hae Kim, Taehyung Wang, K.H. (Kane) Kim and Young Kyu Yang, "TMO-structured Cluster-based Real-time Management of Location Data on Massive Volume of Moving Items," in Proc. WSTFES 2003, IEEE Press, Hakodate, Japan, May 2003, pp.89-92.
- [3]Yunmook Nah, K.H.(Kane) Kim, Taehyung Wang, Moon Hae Kim, Jonghoon Lee, Young Kyu Yang, "A Cluster-based TMO-structured Scalable Approach for Location Information Systems," in Proc. WORDS 2003 Fall, IEEE CS Press, October 2003, Capri Island, Italy.
- [4]Moon Hae Kim, K.H.(Kane) Kim, Yunmook Nah, Joonwoo Lee, Taehyung Wang, Jonghoon Lee, Young Kyu Yang, "Distributed Adaptive Architecture for Managing Large Volumes of Moving Items," IDPT-Vol.2, 2003, pp.737-744.
- [5]Yunmook Nah, Moon Hae Kim, and Ki-Joon Han, "Distributed Scalable Approach for Managing Large Volumes of Location Data," in Proc. US-Korea Conference 2004, August 12-14, 2004, Research Triangle Park, North Carolina, USA.
- [6]이준우, 전세길, 나연록, "TMO 기반 분산 이동 객체 데이터베이스의 설계 및 구현", 한국정보과학회 학술 발표 논문집 VOL.30 NO.01, pp.764-766, 2003.04.
- [7]고영균, 나연록, "GALIS의 디스크 기반 위치 정보 관리 기의 설계 및 구현", 한국정보과학회 학술 발표 논문집 VOL.30 NO.2-2, pp.85-87, 2003.10.