

# 분산 이동 객체 데이터베이스를 위한 실시간 모니터링 시스템

김상우\*, 전세길\*\*, 나연목\*

\*단국대학교 전자컴퓨터공학과

\*\*한국도로공사 도로교통기술원

e-mail : \*{swkim, ymnah}@dbl.dankook.ac.kr

\*\*sgjeon@freeway.co.kr

## Real-time Monitoring System for Distributed Moving Object Databases

Sangwoo Kim\*, Segil Jeon\*\*, Yunmook Nah\*

\*Dept. of Electronics and Computer Engineering, Dankook University

\*\*Highway & Transportation Technology Institute

### 요 약

최근의 위치 측위 기술과 무선 통신 기술의 발전에 따라 위치 기반 서비스에 대한 관심이 크게 증가하고 있다. 휴대폰 사용자와 같은 대용량의 객체를 처리하기 위해서 기존의 단일 노드 기반 시스템으로는 어려움이 있어, 클러스터 기반 분산 컴퓨팅 구조로 GALIS 아키텍처가 제안되었다. 본 논문에서 제안하는 실시간 모니터링 시스템은 클러스터 기반 분산 컴퓨팅 구조로 제안된 GALIS 구현에 있어 기존 명령어 인터페이스를 GUI 로 개선하여 질의 처리 과정에 대한 직관성을 높이고, 각 노드의 부하를 모니터링 함으로써 각 노드의 역할을 동적으로 조정할 수 있게 각 노드별 이동 객체 처리 상황을 관리한다.

### 1. 서론

최근의 GPS 로 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 이루어져 휴대폰 사용자 위치 추적과 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 분산

시켜 대용량의 데이터를 처리할 수 있으며, Time-zone 이라는 개념을 도입하여 오래 된 데이터 일수록 큰 정밀도를 요구하지 않는다는 전제 하에 각 시간대 별로 다른 정밀도를 설정하여 과거 위치 정보를 필터링 하여 저장 공간 활용의 효율성 증대와 빠른 검색이 가능하도록 하였다[1].

본 논문에서는 클러스터 분산 컴퓨팅 구조로 제안된 GALIS 구현에 있어 기존 명령어 인터페이스를 GUI(Graphical User Interface)로 개선하여 질의 처리에 과정에 대한 직관성을 높이고, 이동객체 데이터 처리 시 분산 노드별 부하를 관리하여 상호간 역할을 조정할 수 있게 하는 실시간 모니터링 시스템을 제안한다.

본 논문의 구성은 다음과 같다. GALIS 의 구조에 대해 2 장에서 기술하고, 3 장에서는 본 논문에서 제안한 실시간 모니터링 시스템을 설명하고, 4 장에서는 질의 인터페이스 예를 제시하고, 5 장에서는 결론 및 향후 연구 과제에 대해 설명한다.

† 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성지원사업의 연구결과로 수행되었음

2. GALIS 개요

2.1 구조

GALIS 는 그림 1 과 같이 현재 위치 정보를 처리하는 SLDS(Short-term Location Data Subsystem)와 과거 위치 정보를 처리하는 LLDS(Long-term Location Data Subsystem)가 주요한 구성요소로서 이동 객체 데이터를 보고하는 이동객체 시뮬레이터와 질의처리로 구성된다.

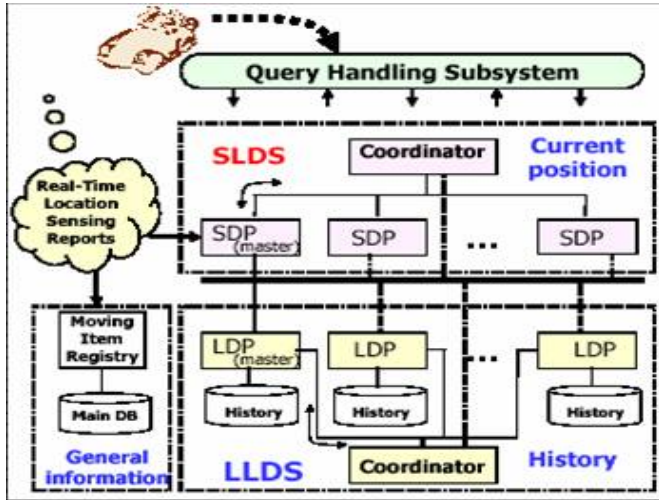


그림 1. GALIS 구조

2.1.1 이동객체 시뮬레이터

이동객체 시뮬레이터인 Network-based Generator 를 사용하여 이동객체 수, 보고시간, 이동경로 등의 조건을 “서울시내에서 운전중인 차량기준으로 이동객체 수 300 개에 대해 5 초마다 보고하라.”와 같이 주고 이동객체의 위치정보를 생성하여 SLDS 로 보고한다[8].

2.1.2 SLDS

현재 위치정보만을 다루기 위해 메인 메모리 데이터베이스를 사용함으로써 빈번하게 보고되는 위치정보를 관리하고 질의수행 시 빠른 응답시간을 가지며, SDP(Short-term Data Processor)라고 하는 노드들로 구성된다. 이동객체 시뮬레이터로부터 보고 받은 위치정보를 관리구역에 따라 다른 노드로 전송하는 SDP master 와 이를 수신한 위치정보를 처리하는 SDP worker 들로 구성되고 노드 부하 크기에 따라 유휴 노드를 활용한 관리구역 분할 및 합병을 수행한다.

2.1.3 LLDS

SLDS 와 유사하게 다수의 LDP(Long-term Data Processor)로 구성되며 디스크 기반 데이터베이스를 사용한다. 과거 위치정보를 SLDS 로부터 받는 LDP master 와 LDP worker 로 구성되어 Time-zone shifting 기법에 따라 대용량의 위치정보를 효율적으로 다룬다.

2.1.4 모니터

이동객체 시뮬레이터로부터 보고되는 위치정보를 GUI 환경에 나타내고 이동객체에 대한 질의를 입력

받아 SDPs 와 LDPs 로 전달하고 그 결과를 출력한다. 실시간으로 SLDS 와 LLDS 노드들의 CPU 사용율과 메모리 사용량 그리고 관리하는 이동객체 수를 나타낸다.

2.2 TMO scheme

본 시스템에서는 실시간 분산 처리를 위하여 TMO(Time-triggered Message-triggered Object) scheme 을 사용하였다. TMO 구조는 실시간 분산 처리에 대해 신뢰성을 제공하며 개발자에게 분산 처리 구현에 대한 부담을 적게 해주며 노드 확장을 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) : SpM 과 SvM 이 동시에 내부 데이터에 접근시 SpM 에 우선순위를 둔다.
- 각 TMO 는 독립적이며 다른 TMO 와 통신할 수 있는 RMMC(Real-time Multicast and Memory replication Channel)를 제공한다.

3. GALIS 모니터링 시스템

본 논문에서 구현한 GALIS 모니터링 시스템은 이동객체 위치표시, 상황보고, 질의인터페이스 기능들을 MMC(Microsoft Management Console) 인터페이스에 맞춘 각각의 ActiveX Control 로 구성되어 있다[9].

3.1 이동객체 위치표시

이동객체 시뮬레이터에서 보고된 위치정보는 SDP master 를 통해 전달받아 GALIS 모니터로 보고한다. 그림 2 와 같이 보고 받은 위치정보를 서울시 지도 위에 나타내어 직관적으로 이동객체의 움직임을 식별할 수 있으며 관심 있는 영역을 확대해서 볼 수 있다.

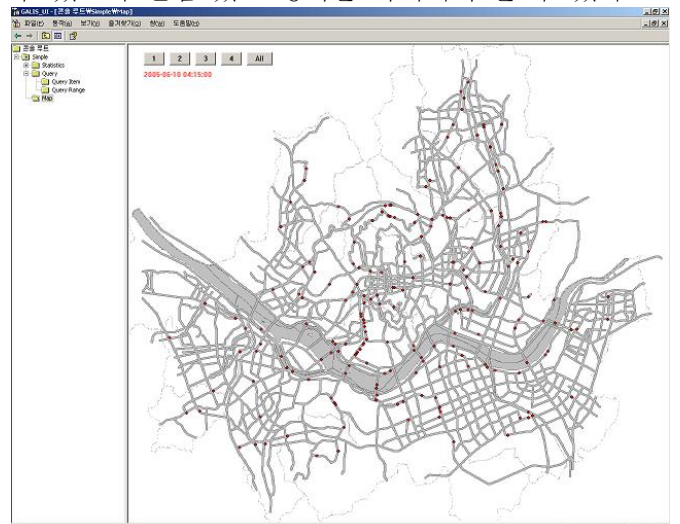


그림 2. 이동객체 위치표시

### 3.2 성능측정

각각의 노드가 편리하게 자신의 CPU 사용율, 메모리 사용량을 측정할 수 있는 라이브러리를 TMO 노드에서 사용한다. 이와 함께 각 노드는 관리하는 객체 수를 GALIS 모니터에 보고함으로써 GALIS의 상태정보를 실시간으로 살펴볼 수 있다. 그림 3은 GALIS 모니터링 시스템에 사용된 성능측정 라이브러리의 클래스 다이어그램이다.

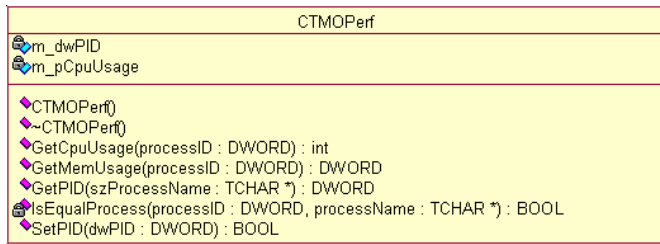


그림 3. TMOPerf 클래스 다이어그램

### 3.3 모니터링 정보 수집

질의 요청에 의해 다수의 노드에서 대용량의 이동객체 정보가 보고될 때와 실시간 상황보고가 수행될 때 실시간 분산 환경을 제공하는 TMO의 통신채널인 RMMC의 안정성을 위해 GALIS 보고용 서버를 사용한다. 질의 종류 및 상황보고를 수행하는 노드마다 각각의 포트를 할당하였고 보고되는 데이터 양에 유동적으로 대처할 수 있게 하여 통신부하를 줄여 실시간성을 높였다. 그림 4와 같이 각각의 노드는 성능측정 라이브러리를 사용하여 모니터링 시스템에 속해 있는 GALIS 보고용 서버로 직접 보고를 한다.

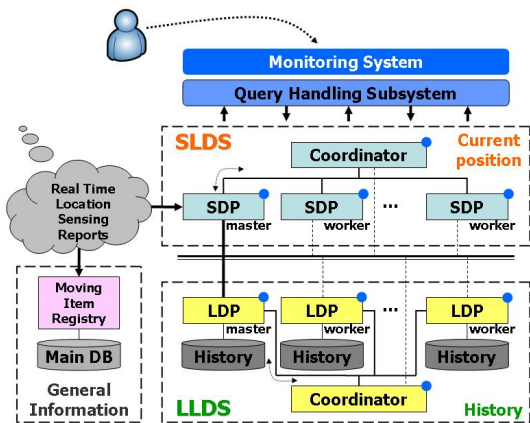


그림 4. GALIS 모니터링 시스템 아키텍처

### 3.4 상황보고

상황보고는 LLDS의 그림 5와 같이 LDP master와 LDP worker의 상태정보를 CPU 사용율, 메모리 사용량, 관리하는 이동객체 수를 나타내고 있다. 시간 경과흐름에 따라 배경의 격자가 이동하면서 현재 상태정보와 지난 상태정보들을 겹쳐서 나타내고 있다.

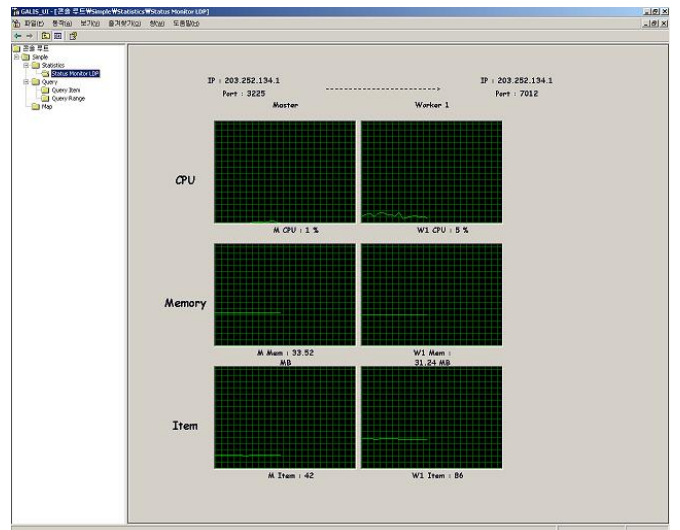


그림 5. 상황보고

넓은 영역에 대한 범위 질의를 수행 중이거나 관리 노드 변경으로 인한 대량의 이동객체 데이터가 노드 간 이동될 때 직관적으로 노드별 부하 크기변화를 알아볼 수 있다. SLDS의 경우 노드 분할 및 합병 시 노드수가 변화하는데 상황보고 창이 동적으로 변화한다. 이는 위치정보 관리기법에 노드간 역할 조정 시점을 판단하는 계수로 사용하여 동적 역할 조정을 가능하게 할 수 있다.

## 4. 질의 인터페이스

질의 인터페이스는 사용자와 상호작용하는 GUI 환경으로 시간 조건을 포함한 아이템 질의, 범위 질의, 최근접 질의를 지원한다.

### 4.1 범위 질의

그림 6은 범위 질의를 나타내고 있다. 질의 결과는 각 OID(Object ID)마다 색을 부여하여 나타내고 있다. 질의 범위를 설정하는 다각형은 최대 20 각형까지 할 수 있다. 범위 질의를 하는 과정은 시간조건을 설정하고 범위를 나타낼 다각형수를 설정한 후, 지도상에서 범위를 설정해 질의를 요청한다. 해당 노드들로부터 질의결과가 받아 이동객체의 시공간 위치정보를 살펴볼 수 있다. 또 각 노드에서 반환된 위치정보 수를 알 수 있다.

### 4.2 아이템 질의

OID와 시간조건을 포함해 질의하여 각각의 노드들로부터 보고된 이동객체의 수와 위치정보를 나타낸다. 그림 7에서는 아이템 질의로써 “OID가 113인 이동객체의 2002년 1월 1일 1시 1분 1초부터 2006년 1월 1일 1시 1분 1초까지 발생한 위치정보를 보고하라”라고 질의한 예이다.

참고문헌

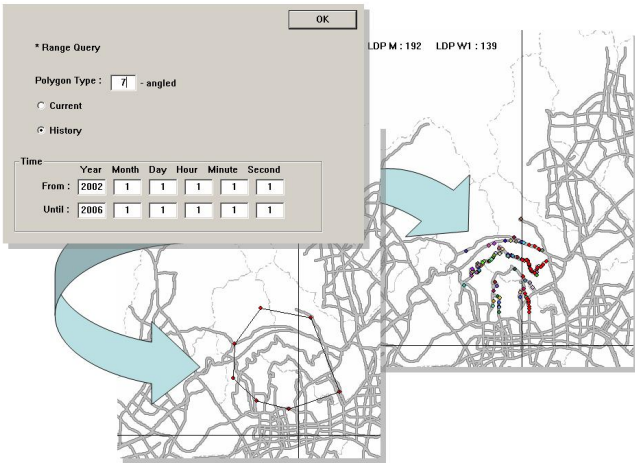


그림 6. 범위 질의

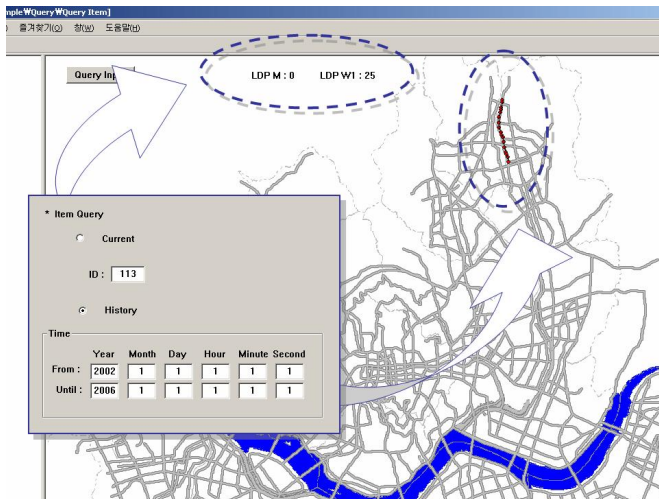


그림 7. 아이템 질의

5. 결론 및 향후 연구

본 논문에서는 분산 이동객체 데이터베이스를 성능 측정 라이브러리를 사용한 실시간 모니터링 기법을 제안한다. 또한 질의 종류에 따른 GUI 인터페이스를 구현하고 이동객체위치표시를 하여 모니터링 상태를 살펴볼 수 있게 하였다.

향후 특정 노드에 집중된 부하를 분산시키기 위해 본 논문에서 제안한 구조를 기반으로 각 노드별 사용 가능한 자원인 유휴 CPU 사용율과 유휴 메모리 사용량을 모니터링하여 노드별 관리하는 이동객체를 동적으로 조정함으로써 GALIS 시스템의 성능을 개선시키는 연구가 필요하다.

- [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, 나연목, "TMO 기반 분산 이동 객체 데이터베이스의 설계 및 구현", 한국정보과학회 학술 발표 논문집, VOL.30 NO.01, pp.764~766, 2003.04.
- [7] 고영균, 나연목, "GALIS 의 디스크 기반 위치 정보 관리기의 설계 및 구현", 한국정보과학회 학술 발표 논문집, VOL.30 NO.2-2, pp.85~87, 2003.10.
- [8] Network-based Generator of Moving Objects, <http://www.fh-oow.de/institute/iapg/personen/brinkhoff/generator/>
- [9] MSDN Library, <http://msdn.microsoft.com/library>