

대용량 위치 데이터 관리를 위한 GALIS의 SLDS 프로토타입 구현

이운주*, 이준우, 나연목
 단국대학교 전자컴퓨터공학과
 wjlee*@dbl.dankook.ac.kr

Implementation of GALIS SLDS prototype for managing large volumes of location data

Woon-Joo Lee*, Joon-Woo 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. 기존 연구의 단일 노드 기반 시스템으로는 휴대폰 사용자와 같은 대용량의 객체를 처리하는데 어려움이 있다. 본 논문에서는 대용량 이동 객체의 시공간 정보를 관리하기 위해 클러스터 기반 분산 컴퓨팅 구조로 제안된 GALIS(Gracefully Aging Location Information System)의 아키텍처 중 객체의 현재 위치 정보를 관리하는 SLDS(Short-term Location Data Subsystem)의 프로토타입을 개발하였다. 본 논문에서 구현한 시스템은 메인 메모리 데이터베이스를 사용하여 디스크 접근 시간이 없고 현재 정보와 과거 정보를 분리하여 빠른 검색이 가능하기 때문에 대용량 이동 객체를 관리하며 빠른 응답을 필요로 하는 상황에 효과적으로 대응할 수 있는 이점이 있다.

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 오래 된 데이터 일수록 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].

본 논문에서는 GALIS에서 객체의 현재 위치 정보를 담당하는 SLDS(Short-term Location Data Subsystem)의 프로토타입을 개발하였다. GALIS의 구조에 대해 2장에서 설명하고, 3장에서는 구현된 SLDS 프로토타입의 동작 방식에 대해 설명한다. 4장에서는 시스템의 구현에 대해 기술하고 5장에서는 결론과 향후 연구 과제에 대해 설명한다.

2. 관련연구

2.1 GALIS의 구조

GALIS의 전체적인 구조는 그림 1과 같다. Moving Item Registry는 main DB에 이동 객체의 프로파일 등록하게 된다. GALIS는 크게 객체의 현재 위치 정보를 처리하는 SLDS와 과거 위치 정보를 처리하는 LLDS(Long-term Location Data Subsystem)으로 나눌 수 있다.

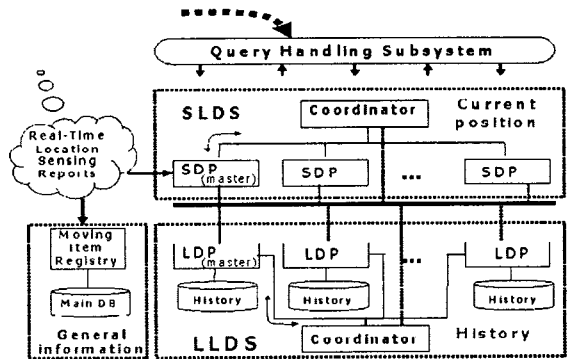


그림 1. GALIS의 전체 구조[2]

① SLDS

SLDS는 메인 메모리 데이터베이스를 사용함으로써 디스크 기반 데이터베이스에 비해 빠른 처리시간으로 대용량 객체의 빈번한 위치정보 갱신에 대응할 수 있다. SLDS내의 각 노드를 SDP(Short-term Data Processor)라고 한다. SDP는 macro-cell로 명명된 일정 지역 내의 객체의 정보를 담당하며, 다수의 SDP가 다른 객체들의 현재 위치 정보를 갱신하기 위해 동시에 동작하게 된다. SDP master는 실시간으로 측위된 이동 객체의 위치 정보를 받아 다른 SDP worker 및 LDP(Long-term Data Processor) master로 전달하는 역할을 한다. coordinator node는 각 노드에 할당된 객체 수를 모니터링하여 한 노드에 객체가 편중되지 않도록 노드들을 분할하거나 합병함으로써 동적 부하 균형을 가능하도록 한다.

② LLDS

LLDS는 SLDS와 유사하게 다수의 LDP들로 구성된다. LLDS는 디스크 기반 데이터베이스를 사용하여 객체의 과거 위치 정보를 경과된 시간대에 따라 네 개의 시간 존(time-zone)에 나누어 유지하게 된다. 만약 한 객체가 100m x 100m의 micro-cell 내에서 이동을 했다면 SLDS에서는 현재 위치 정보를 갱신하는 반면 LLDS에서는 갱신하지 않는다. 이는 SLDS에 비해 L

LDS에서의 위치 정보 갱신 빈도가 낮음을 의미한다.

2.2 TMO scheme

구현된 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 메서드를 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 프로토타입

본 논문에서 구현한 SLDS에서는 기존 GALIS 설계에서 객체 정보를 우선 SDP master가 받아 각 SDP/LDP 노드에 전달한 방식과 달리 동시에 전체 노드에 전달하고 각 노드 스스로가 자신의 영역에 해당하는 데이터인지 판단하여 저장하도록 하였다. 이는 SDP master의 부하를 줄여주기 위한 조치였다.

실현에 사용된 위치 정보는 별도의 Generator를 두어 이동 객체들의 위치 가상 데이터를 실시간으로 생성하게 하였다.

3.1 위치 정보의 저장과 질의 처리 과정

SDP는 위치 데이터 생성기에서 생성한 데이터를 RMMC를 통해 전달받아 현재 SDP가 관리하는 영역에 속하는 객체 위치 정보만을 선별하여 main-memory DB에 기존의 정보를 모두 삭제하고 새로 전달받은 내용을 삽입하는 방식으로 SDP가 관리하는 영역 내에 객체의 현재 위치 정보를 갱신하게 된다.

질의의 경우 SDP는 UI를 통해 입력받은 질의를 main-memory DB를 통해 질의를 수행하여 그 결과를 돌려받아 UI로 보내고 UI는 결과를 종합하여 사용자에게 보여준다. SLDS에서 정보의 저장 과정과 질의 처리 과정은 그림 2와 같다.

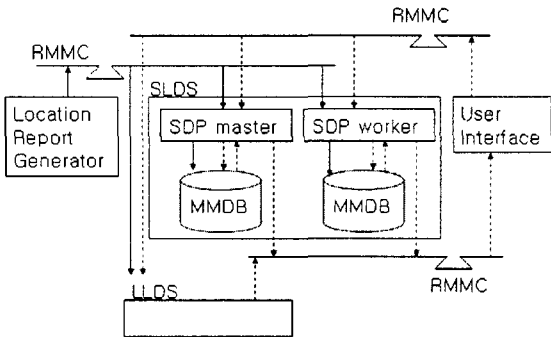
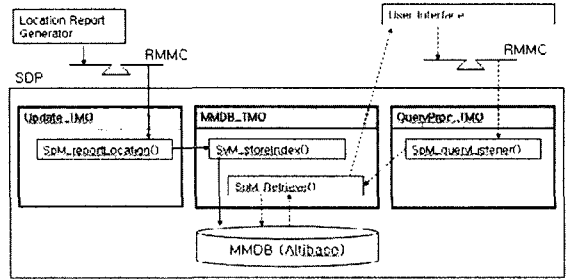


그림 2. 구현된 시스템에서 SLDS의 전체 구조

3.2 SDP의 내부 구조

그림 3에서 SDP를 구성하고 있는 세 개의 TMO object와 각 메서드들의 동작에 대해 나타내고 있다.



→ 위치 정보의 저장 과정
 질의 처리 과정

그림 3. SDP의 구조

Location Report Generator는 생성된 데이터를 RMMC를 통해 각 SDP로 보내게 된다.

SpM_reportLocation()메서드는 SpM으로써 일정 주기마다 스스로 실행되어 RMMC를 통해 전달되어지는 데이터를 MMDB_TMO로 전달하고 SvM_storeIndex()를 호출하여 DB에 삽입하게 된다.

SvM_StoreIndex()메서드는 전달받은 객체 위치 데이터 중 예 현재 SDP가 관리하는 영역의 데이터만 DB에 저장하게 된다.

SvM_Retrieve()메서드는 QueryProc_TMO에서 전달된 질의를 DB에 보내 수행하여 결과를 User Interface로 돌려주게 된다.

SpM_QueryListener()메서드는 SpM으로써 주기적으로 UI에서 질의가 전달되는지를 검사하여 질의가 입력된 경우 MMDB_TMO의 SvM_Retrieve()를 호출하여 질의를 수행한 후 그 결과를 받아 다시 UI로 되돌려준다.

이렇게 TMO를 기반으로 구성되어진 시스템은 개발자의 분산 처리 구현에 대한 부담을 덜어주며, 비교적 적은 노력으로 노드의 확장을 가능하게 한다. 또한, TMO간의 통신 채널(RMMC)을 별도로 제공하므로 개발자는 시스템의 본래 역할의 구현에만 충실하도록 하는 장점이 있다.

4. 구현

개발된 프로토타입은 세 대의 컴퓨터로 구성하였다. 한 대의 컴퓨터는 실현에 사용될 위치 데이터를 생성시키며, 전체 지역의 현재 객체 위치를 표시하고 사용자에게 질의를 받아 결과를 표시해주는 User Interface를 제공한다. 다른 두 대의 컴퓨터는 각각 하나의 SDP와 하나의 LDP를 동작시키며 SDP가 알고 있는 지역의 현재 객체 정보를 표시하는 노드 모니터를 제공한다. 즉 두 개의 SDP와 두 개의 LDP가 동작하게 된다. SDP는 main-memory DBMS인 AltiBase 3.5.3을 이용하여 객체들의 현재 위치 정보를 저장한다. 실현에 사용된 각 PC의 사양은 표 1과 같다.

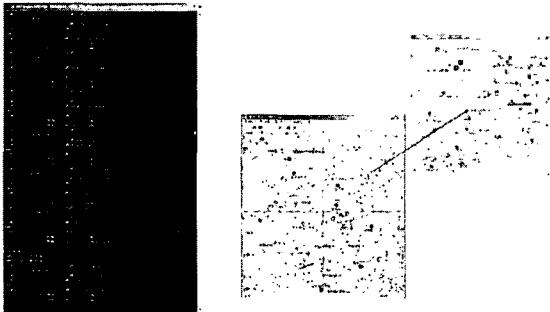
표 1. 실현에 사용된 PC 사양

	PC 1	PC 2	PC 3
역할	가상 데이터 생성 및 UI	Master SDP/LDP	Worker SDP/LDP
CPU	Pentium IV 2.6	Pentium IV 2.4	Pentium IV 2.8
RAM	1Gbyte	1Gbyte	1Gbyte
OS	Windows 2003	Windows 2003	Windows 2003

이동 객체가 움직이는 전체 영역의 범위는 1600m x 1600m로 하였으며 전체 영역을 4등분하여 좌상단과 우상단 노드에

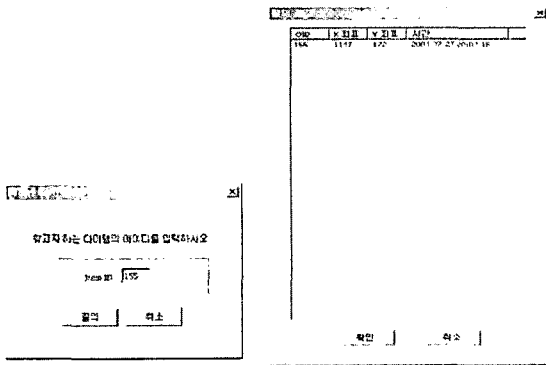
각각 한 대씩의 컴퓨터를 할당하였다. 이동 객체의 수는 300개로 정하였고 매 5초 마다 임의의 좌표 정보를 생성하여 객체의 이동을 표현하였다.

그림 4에서 4등분된 전체 영역의 현재 객체 정보를 실시간으로 파악할 수 있는 모니터와 하나의 노드 상태를 보여주는 모니터 및 객체를 생성하고 저장하는 콘솔 화면을 볼 수 있다.



(a) 시스템 동작 콘솔 (b) 전체 영역과 한 노드의 모니터
그림 4. 프로토타입 실행 화면

그림 5에서는 객체의 ID로 현재 위치를 검색하고 결과를 표시하는 화면을 나타내었다.



(a) 현재 위치 질의 화면 (b) 질의 결과 화면
그림 5. 질의 실행 화면

구현된 시스템의 SLDS에는 메인 메모리 데이터베이스를 사용하고 두 개의 서브시스템을 두어 현재와 과거의 정보를 분리하여 관리하고 있기 때문에 현재 질의에 대한 빠른 응답을 기대할 수 있다. 예를 들어 재난 복구 응용에서 긴급 환자에 대한 신고를 받았을 때 '현재 신고 지점에서 가장 가까운 구급차를 찾아라'라는 질의의 처리 등에 활용될 수 있다.

5. 결 론

GALIS는 최소 백만 단위 이상의 대용량 객체의 현재와 과거의 위치 정보를 관리하기 위해 설계된 시스템이다. 본 논문에서는 GALIS의 구조 중 현재 위치 정보를 관리하는 SLDS의 프로토타입을 TMO 아키텍처와 Altibase라는 메인 메모리 데이터베이스를 사용하여 구현하였다.

본 논문에서 제안한 시스템은 메인 메모리 데이터베이스를 사용하여 디스크 접근 시간이 없고, 과거 정보와 현재 정보를 분리하여 현재 정보 검색 시 검색해야 할 데이터의 양이 현저히 줄어드는 만큼 매우 빠른 검색을 가능하게 한다. 그 결과 다양

한 응용 분야, 특히 휴대폰 사용자와 같이 이동 객체의 수가 매우 많고 현재 위치 질의가 빈번한 분야에 효과적으로 적용될 수 있다. 다만 현재와 과거의 정보 분리에서 얻는 이점보다 분산처리와 네트워크 통신에 드는 비용이 더 큰 경우가 있을 수 있기 때문에 차후 이에 대한 고려가 필요하다.

본 논문의 향후 과제로는 LLDS에서 오래된 데이터 중 짧은 거리의 움직임을 필터링하는 시간 존의 구현, 노드간의 부하 균형을 위한 coordinator 노드의 구현이 있다.

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 SIG DB, 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, Joonwo 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, 나연욱, "TMO 기반 분산 이동 객체 데이터베이스의 설계 및 구현", 한국정보과학회 학술 발표 논문집 VOL.30 NO.01 pp.764~766 2003.04
- [7]고영균, 나연욱, "GALIS의 디스크 기반 위치 정보 관리기의 설계 및 구현", 한국정보과학회 학술 발표 논문집 VOL.30 NO.2-2 pp.85~87 2003.10