

# SDP 마스터 이중화를 지원하는 확장 SLDS 설계 및 구현† Design and Implementation of the Extended SLDS Supporting SDP Master Replication

신인수\*, 강홍구, 홍동숙, 한기준

In-Su Shin\*, Hong-Koo Kang, Dong-Suk Hong, Ki-Joon Han

건국대학교 컴퓨터·정보통신공학과

{isshin\*, hkkang, dshong, kjhan}@db.konkuk.ac.kr

## 요약

최근 이동체의 위치 데이터를 활용한 위치 기반 서비스에 대한 관심이 높아지면서, 보다 효율적인 이동체 위치 데이터 관리 시스템으로 클러스터 기반 분산 컴퓨팅 구조인 GALIS(Gracefully Aging Location Information System)가 제시되었다. 그러나 GALIS의 서버 시스템인 SLDS(Short-term Location Data Subsystem)는 다수의 SDP(Short-term Data Processor) 노드들이 처리한 질의 결과를 SDP 마스터에서 취합하여 클라이언트로 보내는 구조이기 때문에 SDP 마스터에 장애가 발생할 경우 서비스가 중지되고, SDP 마스터에 부하가 집중될 경우 클라이언트의 응답 시간이 길어지는 문제가 있다.

따라서, 본 논문에서는 기존 SLDS에 이중화를 지원하기 위해 SDP 마스터를 추가하여 기존 SLDS의 안정성을 높이고 이동체 위치 데이터의 질의 처리 성능을 향상시킨 확장 SLDS를 설계 및 구현하였다. 확장 SLDS에서는 두 대의 SDP 마스터를 가동함으로써 한 SDP 마스터에 장애가 발생하더라도 다른 SDP 마스터가 서비스를 계속 제공함으로써 시스템의 안정성을 보장한다. 또한, 확장 SLDS는 두 대의 SDP 마스터가 질의 처리를 분산하여 수행하기 때문에 클라이언트의 응답 시간을 줄일 수 있다. 마지막으로 확장 SLDS의 장애 테스트와 질의 처리 성능을 실험하였으며, 이러한 실험을 통해 확장 SLDS의 고신뢰성 및 고가용성을 검증하였다.

## 1. 서론

최근 개인용 휴대 단말기가 대중화됨에 따라 이동체의 위치 데이터를 이용한 GPS, GIS, 텔레매틱스 등의 서비스와 관련된 위치 기반 서비스(LBS) 분야의 기술이 활발히 연구되고 있다. 이러한 다양한 위치 기반 서비스를 제공하기 위해서는 대용량의 이동체 위치 데이터를 효율적으로 관리할 수 있고, 실시간으로 저장 및 검색

할 수 있는 위치 데이터 관리 시스템이 필요하다[1,2].

이동체의 위치 데이터는 시간에 따라 그 위치 정보가 계속 변하는 대용량의 시공간 데이터이므로, 효율적인 관리를 위한 위치 데이터 서버가 요구된다. 이러한 이유로 대용량 이동체 데이터의 관리를 위한 시스템 아키텍처인 GALIS가 제안되었다[3,4,5]

† 본 논문은 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음.

GALIS는 여러 노드로 구성된 분산 컴퓨팅 구조로 되어있으며, 그 서브시스템인 SLDS는 여러 개의 SDP 노드들로 구성된다. SLDS에서는 클라이언트로부터 들어온 질의를 각 SDP 노드들이 처리한 후, 그 결과를 SDP 마스터에서 취합하여 다시 클라이언트로 보내게 된다. 그러나 이러한 SLDS의 정책은 질의 결과 취합 및 전송을 담당하는 SDP 마스터에 시스템 오류 등으로 인한 장애가 발생할 경우 서비스가 중단되는 문제가 존재한다. 또한, 다수의 질의 처리 시 하나의 SDP 마스터에서 모두 처리해야 하므로 SDP 마스터에 부하가 집중되어 클라이언트의 응답시간이 길어지게 되는 문제가 있다.

이러한 문제들을 해결하기 위한 방법으로 본 논문에서는 이중화된 SDP 마스터를 추가함으로써 기존 SLDS의 안정성을 높이고 이동체 위치 데이터의 질의 처리 성능을 향상시킨 확장 SLDS를 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 GALIS와 TMO 프로그래밍의 구조에 대해 설명한다. 3장에서는 확장 SLDS 구조에 대하여 기술하고, 4장에서는 확장 SLDS 구현 및 테스트를 기술한다. 마지막으로, 5장에서는 결론에 대하여 언급한다.

## 2. 관련연구

### 2.1. TMO 프로그래밍 구조

TMO(Time-triggered Message-triggered Object)는 기존 객체 모델을 실시간 분산 컴퓨팅이 가능하도록 확장한 모델이다[6]. 그림 2는 TMO 프로그래밍 구조를 보여주고 있다.

EAC(Environment Access Capability)는 네트워크 상의 다른 TMO와 통신하기 위한 방법으로, 메시지 기반 원격 메소드 호출 인터페이스인 Gate와 메모리 복

제 기반의 통신 채널인 RMMC(Real-time Multicast and Memory-Replication Channel)로 구성된다.

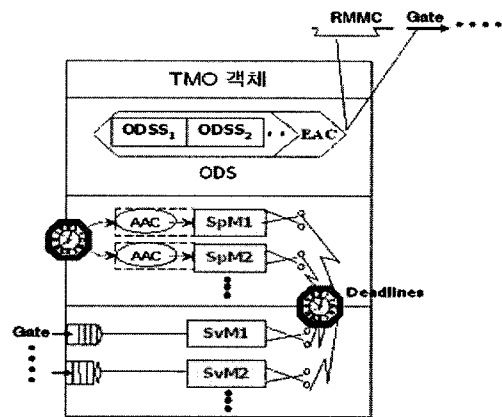


그림 1. TMO 프로그래밍 구조

TMO의 주요 메서드로 SvM(message-triggered Service Method)과 SpM(time-triggered Spontaneous Method)이 제공된다. SvM은 외부 TMO로부터 전달받은 메시지에 의해서 수행되는 메소드이다. 다중 노드 상에 분산되어 있는 TMO들은 Gate를 이용하여 다른 TMO의 서비스 메소드를 수행할 수 있다. SvM과는 다른 새로운 유형의 시간구동 메소드인 SpM은 주기성을 띄거나 시간성을 갖는 서비스를 수행하기 위한 메소드이다. SpM은 시간 속성을 정의하기 위한 명세인 AAC(Autonomous Activation Condition)에 의해서 실행된다.

### 2.2. GALIS

이동체의 데이터의 관리를 위해 만들어진 GALIS는 다중 데이터 프로세서로 구성되며, 클러스터 기반 분산 컴퓨팅 구조로서 각 프로세서가 서로 다른 특정 영역 내에 있는 이동체의 움직임에 대한 처리를 수행한다.

GALIS는 현재 위치 데이터를 처리하는 SLDS와 과거 위치 데이터를 처리하는 LL





SDP 마스터는 코디네이터에게 일정 시간(SEND\_ALIVE\_TIME)마다 alive 메시지를 보냄으로써 자신의 동작 상태를 알리게 된다. 만약 SDP 마스터로부터 alive 메시지를 받지 못할 경우 수신 대기하게 되고, alive 메시지를 일정 횟수(Threshold) 만큼 받지 못할 경우에 장애로 판단하게 된다. 코디네이터가 장애라고 판단하게 되면, 장애가 발생한 SDP 마스터의 상태를 FAIL로 변경한 후 정상 동작 중인 SDP 마스터에게 역할을 대행하도록 takeover 메시지를 보낸다.

takeover 메시지를 받은 SDP 마스터는 장애가 발생한 SDP 마스터 대신하여 SDP 워커들로부터 질의 결과를 취합하여 클라이언트에게 전송한다. 그리고 장애가 발생한 SDP 마스터의 복구가 완료되면 코디네이터는 장애 복구가 완료된 SDP 마스터를 등록하고, 두 대의 SDP 마스터는 정상적인 서비스를 시작한다.

#### 4. 확장 SLDS 구현 및 테스트

본 장에서는 기존 SLDS의 안전성을 향상시키고, 효율적인 질의 처리를 할 수 있도록 SDP 마스터의 이중화를 지원하는 확장 SLDS를 구현 및 테스트 하였다. 테스트는 두 대의 SDP 마스터, 한 대의 SD

P 워커, 코디네이터, 클라이언트로 수행하였고, 장애 테스트 시 SDP 마스터의 장애를 판단하는 데 사용되는 Threshold 값은 10으로 설정하였다.

그림 6은 SDP 마스터의 장애 시 질의 처리 테스트 화면을 보여준다.

그림 6에서와 같이 코디네이터는 SDP 마스터로부터 질의 QID1을 수신하였다는 메시지를 받은 후 IDLE 상태인 SDP Master1에게 질의 처리를 요청한다. SDP Master1이 질의를 처리 중에 장애가 발생하면, 코디네이터는 SDP Master1로부터 alive 메시지를 받지 못하므로 계속 failcount를 증가시킨다. failcount가 10(Threshold)이 되면 코디네이터는 SDP Master1을 장애로 판단하여 SDP Master1이 처리하던 질의 QID1을 SDP Master2가 대신 처리하도록 takeover 메시지를 보낸다. takeover 메시지를 받은 SDP Master2는 질의 QID1을 처리하여 결과를 클라이언트에게 전송한다.

그림 7은 질의 처리 테스트 화면을 보여준다.

그림 7에서와 같이 SDP Master1이 질의 QID1을 처리하는 동안 클라이언트로부터 질의 QID2가 들어오면, 코디네이터는 SDP 마스터의 상태를 확인한다. 이 때 S



그림 6. 장애 시 TAKEOVER 테스트



그림 7. 질의 처리 테스트

DP Master1이 BUSY 상태이므로, IDLE 상태인 SDP Master2에게 질의 QID2의 처리를 요청한다. 따라서, SDP Master1은 질의 QID1의 처리를 담당하고, SDP Master2는 질의 QID2의 처리를 담당하게 된다.

## 5. 결론

개인 휴대 단말기가 보편화되고 위치 기반 서비스가 늘어남에 따라 이동체의 위치 데이터를 실시간으로 분산 처리할 수 있는 시스템이 요구되고 있다. 이러한 요구에 따라 대용량 이동체 데이터를 관리하기 위한 GALIS 아키텍처가 제시되었다. 그러나, GALIS의 서브시스템인 SLDS는 SDP 마스터에 장애가 발생하여 서비스가 중지되거나, SDP 마스터에 부하가 집중될 경우 클라이언트의 응답 시간이 길어지는 문제가 존재한다.

따라서, 본 논문에서는 SDP 마스터의 이중화를 지원하여 기존 SLDS의 안정성을 높이고 질의 처리 성능을 향상시킨 확장 SLDS를 설계 및 구현하였다. 확장 SLDS에서는 기존의 SLDS에 이중화된 SDP 마스터를 추가함으로써 SDP 마스터에 장애가 발생하더라도 지속적인 서비스를 제공할 수 있도록 하고, 또한 질의를 두 대의 SDP 마스터에서 분산 처리하도록 하여 질의 처리 과정도 효율적으로 개선하였다.

## 참고문헌

[1] C.S. Jensen, A. Friis-Christensen, T.B. Pedersen, D. Pfoser, S. Altenis, N. Tryfona, "Location-Based Services: A Database Perspective," Proc. of the 8th Scandinavian Research Conf. on Geographical Information Science,

2001, pp.59-68.

[2] 박상미, 진희채, 안병익, "위치기반정보서비스를 지원하는 시스템 구조 및 소프트웨어 기술동향 분석," 개방형지리정보시스템학회 학술회의 논문집, 2001, pp.145-160.

[3] 나연목, K.H. Kim, 왕태형, 김문회, 이종훈, 양영규, "GALIS: LBS 시스템의 클러스터 기반 신축성소유 아키텍처," 한국정보과학회 데이터베이스 연구회지, 18권 4호, 2002, pp.66-80.

[4] M.H. Kim, K.H. Kim, Y.M. Nah, J. W. Lee, T.H. Wang, J.H. Lee, Y.K. Yang, "Distributed Adaptive Architecture for Managing Large Volumes of Moving Items," In Proc. of the 7th Biennial World Conf. on Integrated Design and Process Technology, 2003, pp.737-744.

[5] Y.M. Nah, K.H. Kim, T.H. Wang, M.H. Kim, J.H. Lee, Y.K. Yang, "A Cluster-based TMO-structured Scalable Approach for Location Information Systems," Proc. of the 9th IEEE Int. Workshop on Object-oriented Real-time Dependable Systems, 2003, pp.225-233.

[6] K.H. Kim, "Object-Oriented Real-Time Distributed Programming and Support Middleware," Proc. of the 7th Int. Conf. on Parallel & Distributed Systems, 2000, pp.10-20.

[7] 이승원, 강홍구, 홍동숙, 한기준, "실시간 위치 기반 서비스를 위한 확장 SLDS 설계 및 구현," 한국공간정보시스템학회 논문지, 7권2호, 2005, pp.47-56.

[8] 김성기, 정관식, 민병준, 김문회, "실시간 객체의 이중화 구조 설계," 한국정보과학회, 한국정보과학회 학술발표논문집, 24권 2호(1), 1997, pp.547-550.