

# 스마트폰 소셜 네트워크 서비스를 이용한 원격 데이터베이스 모니터링 시스템

김명주, 이언배  
한국방송통신대학교 평생대학원 정보과학과  
e-mail : imuver@gmail.com

## Remote Database Monitoring System using Smart Phone Social Network Service

Myung-Ju Kim, Un-Bae Lee  
Dept. of Computer Science, Korea National Open University Graduate School

### 요 약

본 논문에서는 스마트폰 소셜 네트워크 서비스(Social Network Service)인 트위터(Twitter)를 이용한 원격 데이터베이스 모니터링 시스템을 제안하였다. 제안한 시스템을 통해 원격 데이터베이스를 모니터링할 경우 기존 상용화된 CS(Client-Server)기반 모니터링 방식에 비해 다음과 같은 장점들이 있다. 첫째, 웹 애플리케이션 기반 사용자 인터페이스를 사용하기 때문에 프로그램 유지보수의 문제가 발생하지 않으며, 웹 브라우저를 통한 인터넷 접속이 가능한 환경이라면 언제 어디서든 시간과 공간의 제약 없이 원격 데이터베이스의 모니터링이 가능하다. 둘째, 제안한 시스템은 원격 데이터베이스 모니터링 시 중앙관리서버에 통합 저장되어 있는 원격 데이터베이스들의 상태정보를 사용하므로 관리자 접속이 증가할 경우 데이터베이스에 대한 연결세션과 질의수행 증가로 인해 데이터베이스 성능에 영향을 미칠 수 있는 CS 기반 모니터링 방식과 달리 데이터베이스 성능에 영향을 미치지 않는다. 셋째, 제안한 시스템은 관리자가 모니터링을 수행하고 있지 않은 상황에서 원격 데이터베이스에 문제가 발생할 경우 이를 트위터를 통해 관리자의 스마트폰으로 즉시 통보하므로 관리자가 해당 문제를 실시간으로 확인할 수 있다.

### 1. 서론

데이터는 모든 조직에 있어 가장 중요한 자산 중 하나이다. 따라서, 이토록 중요한 데이터가 저장되는 데이터베이스는 항상 안정적으로 운영되어야 한다.

데이터베이스가 안정적으로 운영되기 위해서는 데이터베이스의 관리자가 시간과 공간의 제약 없이 언제 어디서든 데이터베이스의 현재 상태를 모니터링하여 앞으로 발생할지 모르는 예상치 못한 상황들에 대해 미리 대비할 수 있어야 한다. 하지만, 현재 상용화 되어 있는 CS 기반 원격 데이터베이스 모니터링 방식의 경우 해당 프로그램이 설치되어 있는 PC에서만 모니터링이 가능하기 때문에 모니터링에 있어 시간과 공간상의 제약이 따르며, 관리자가 모니터링을 하고 있지 않은 상황에서 데이터베이스에 문제가 발생할 경우 이를 즉시 통보 받을 수 있는 방법 또한 없다.

본 논문에서는 이러한 CS 기반 모니터링 방식의 문제점들을 해결하고 나아가 관리자가 보다 효율적으로 원격 데이터베이스를 모니터링 할 수 있도록 스마트폰 소셜 네트워크 서비스(이하 ‘SNS’)를 이용한 원격

데이터베이스 모니터링 시스템을 제안하고 구현 및 평가하여 실제 적용 가능성을 검증하고자 한다.

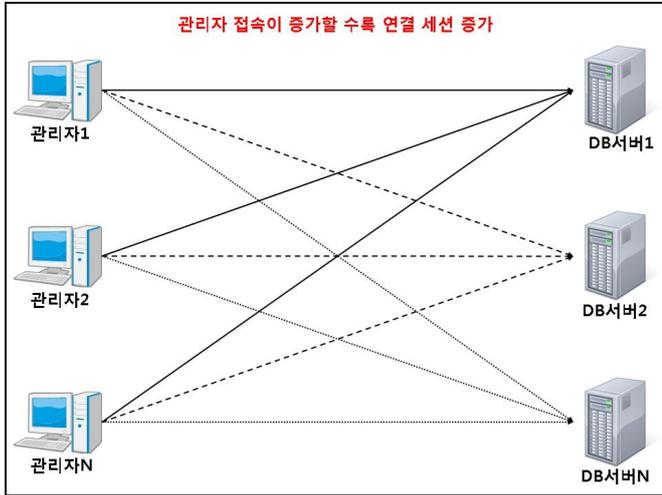
### 2. 관련연구

원격 데이터베이스 모니터링의 주된 목적은 관리자가 해당 데이터베이스의 상태를 수시로 점검하여 앞으로 발생할지 모르는 장애를 사전에 감지하고 예방하여 해당 데이터베이스가 장애 없이 안정적으로 운영될 수 있도록 하는 것이다. 따라서, 원격 데이터베이스 모니터링의 궁극적인 목적은 데이터베이스의 가용성을 증가시키는데 있다고 할 수 있다.

현재 상용화 되어있는 대표적인 원격 데이터베이스 모니터링 제품들은 CS 기반 모니터링 방식으로 클라이언트 PC에 모니터링 프로그램을 설치한 후 설치된 모니터링 프로그램을 이용해 모니터링 대상 원격 데이터베이스에 연결한 후 데이터베이스의 상태를 모니터링하는 아키텍처를 사용하고 있다.

CS 기반 모니터링 방식은 아키텍처상 해당 프로그램이 설치된 PC에서만 모니터링이 가능하고, 아래(그림 1)에서 보는 바와 같이 여러 명의 관리자가 하

나의 데이터베이스에 동시에 접속할 경우 데이터베이스의 연결세션과 질의수행이 함께 증가하여 전체 데이터베이스 성능에 영향을 미칠 수 있다. 또한, 관리자가 모니터링을 하고 있지 않은 상황에서 데이터베이스에 문제가 발생할 경우 이를 즉시 통보 받을 수 있는 방법이 없으며, CS 기반 프로그램들의 고질적인 문제인 프로그램 유지보수에 대한 문제점도 안고 있다.



(그림 1) CS 기반 시스템 구성도

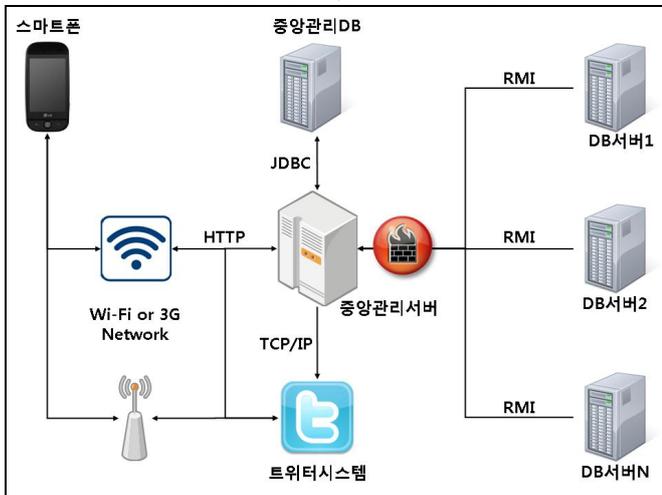
따라서, 본 논문에서는 CS 기반 모니터링 방식의 문제점들을 극복하고 보다 효율적으로 원격 데이터베이스를 관리하기 위해 웹 애플리케이션 기반 중앙관리서버와 스마트폰 SNS 인 트위터를 이용한 원격 데이터베이스 모니터링 시스템을 제안한다.

### 3. 원격 데이터베이스 모니터링 시스템 설계

본 장에서는 제안 시스템의 구성과 아키텍처에 대해 논한다.

#### 3.1 시스템 구성

아래 (그림 2)는 스마트폰 SNS 를 이용한 원격 데이터베이스 모니터링 시스템의 구성도이다.

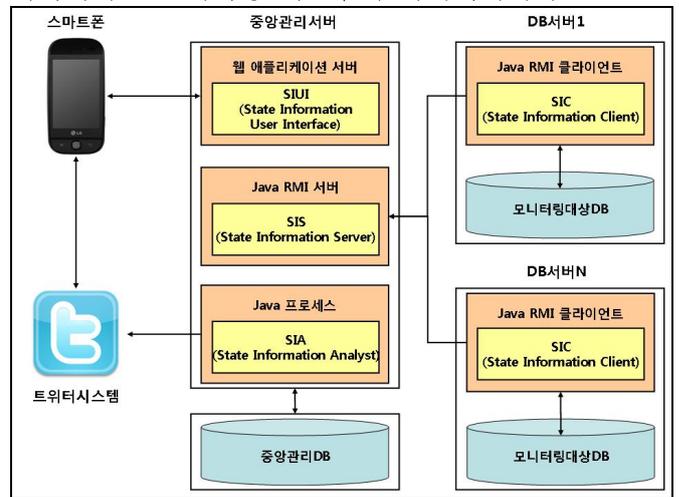


(그림 2) 시스템 구성도

(그림 2)의 정 중앙에 위치한 중앙관리서버는 (그림 2)의 우측에 위치한 모니터링 대상이 되는 DB 서버 1 ~ N 과는 Java RMI 통신을 이용하여 연결 된다. 연결 시 중앙관리서버는 시스템 운영체제에서 제공하는 방화벽 기능을 이용하여 허가 받지 않은 IP 주소와 포트에 대한 연결로부터 시스템을 안전하게 보호한다. (그림 2)의 중앙관리서버 상단에 위치한 중앙관리 DB 는 모니터링 대상이 되는 DB 서버 1 ~ N 으로부터 전송된 데이터베이스의 상태정보를 저장 및 관리하는 데이터베이스서버로써 JDBC 를 이용해 중앙관리서버와 연결된다. (그림 2)의 중앙관리서버 하단에는 DB 서버 1 ~ N 으로부터 시스템상의 문제가 발생하였을 경우 이를 관리자에게 즉시 통보하기 위한 트위터시스템이 위치한다. 중앙관리서버는 트위터시스템과의 연동을 위해 Java 언어기반의 프로그램 API 인 Twitter4J 를 이용한다. 마지막으로 (그림 2)의 좌측에는 원격 데이터베이스 관리자의 단말인 스마트폰이 위치한다. 스마트폰은 Wi-Fi 또는 3G 네트워크 망을 통해 중앙관리서버 및 트위터시스템과 실시간으로 연결되어 시간과 공간의 제약 없이 트위터시스템을 통해 중앙관리서버가 전송한 메시지를 수신하고 필요에 따라 중앙관리서버에 접속하여 데이터베이스의 현재 상태를 확인 할 수 있다.

#### 3.2 시스템 아키텍처

아래 (그림 3)은 스마트폰 SNS 를 이용한 원격 데이터베이스 모니터링 시스템의 아키텍처이다.



(그림 3) 시스템 아키텍처

(그림 3)의 중앙에 위치한 중앙관리서버는 웹 애플리케이션 기반 사용자 인터페이스인 SIUI(State Information User Interface)와 모니터링 대상이 되는 DB 서버 1 ~ N 으로부터 전송된 상태정보데이터를 수신하여 저장하기 위한 SIS(State Information Server) 그리고, 중앙관리 DB 에 저장된 DB 서버 1 ~ N 의 상태정보 데이터를 분석한 후 문제 발생 시 이를 메시지화 한 후 트위터시스템을 통해 관리자에게 전송하는 SIA(State Information Analyst)로 구성되어 있다. 또한, 모니터링 대상이 되는 DB 서버 1 ~ N 에는

모니터링대상 DB 의 상태를 주기적으로 파악한 후 이를 데이터화 하여 SIS 로 전송하는 SIC(State Information Client)가 각각 설치되어 동작한다.

전체 시스템 아키텍처는 이렇다. 모니터링 대상이 되는 DB 서버 1 ~ N 에서 Java 프로세스로 동작하고 있는 Java RMI 클라이언트 프로그램인 SIC 는 주기적으로 모니터링대상 DB 의 상태정보를 파악한 후 이를 중앙관리서버에서 Java 프로세스로 동작하고 있는 Java RMI 서버인 SIS 로 전송하게 된다. SIC 로부터 데이터를 수신한 SIS 는 데이터가 수신될 때마다 이를 분석하여 중앙관리 DB 의 상태정보 테이블에 저장하는 작업을 수행한다. 중앙관리서버에서 또 하나의 Java 프로세스로 동작하고 있는 SIA 는 주기적으로 중앙관리 DB 에 누적된 DB 서버 1 ~ N 의 상태정보 데이터를 분석하여 기준치를 초과하는 상태정보 데이터가 있을 경우 이를 즉시 메시지화 하여 트위터시스템을 통해 해당 관리자에게 전송한다. 스마트폰을 통해 트위터시스템으로부터 전송된 메시지를 수신한 관리자는 메시지를 확인한 후 그 즉시 스마트폰의 웹 브라우저를 이용하여 중앙관리서버에서 제공되는 사용자 인터페이스인 모니터링 웹 애플리케이션에 접속하여 아이디와 패스워드를 이용한 인증 절차를 걸친 후 해당 데이터베이스의 현재 상태를 확인한다.

**4. 구현 및 성능평가**

제안된 원격 데이터베이스 모니터링 시스템은 아래 <표 1>의 환경에서 구현 후 동작을 확인하였다.

<표 1> 제안 시스템 구현 환경

구분	도구 및 버전
운영체제	Windows Vista Home Premium K
데이터베이스	Oracle Database 10g Express Edition
개발언어	Java (JDK1.6.0_21)
개발도구	Eclipse Java EE IDE for Web Developers (Helios)
웹 응용 서버	apache-tomcat-6.0.29
트위터 API	twitter4j-2.1.12
차트	Google 차트 API (Google chart API)
스마트폰	LG KH-5200 (Android Froyo v2.2)

**4.1 상태정보 클라이언트의 구현**

데이터베이스의 상태정보를 주기적으로 수집하여 중앙관리서버로 전송하는 SIC 는 순수 Java 언어와 SQL 을 이용하여 구현하였다. 작성된 SIC 는 Java 프로세스로 동작하기 때문에 JRE 가 설치되어있는 환경이라면 운영체제에 관계없이 어디서든 정상적인 동작이 가능하다. SIC 는 주기적으로 데이터베이스의 상태정보를 파악해오는 작업을 수행하게 되는데, 해당 작업은 오라클에서 제공하는 동적성능뷰(Dynamic Performance Views)에 대한 SQL 질의 방법을 이용하여 구현하였다. 주의할 점은 동적성능뷰에 대한 SQL 질의 시 사용자의 계정이 동적성능뷰에 접근이 가능

한 관리자 이상의 권한이어야 한다는 점이다. 아래 (그림 4)는 SIC 가 동적성능뷰로부터 데이터베이스의 상태정보를 파악해오기 위해 사용하는 SQL 질의 중 하나이다.

```
select sum(value) || ' bytes' from v$sysstat where name like 'bytes%SQL*Net%'
```

(그림 4) 네트워크 I/O 트래픽 파악 SQL 질의

파악된 상태정보는 Java RMI 통신을 이용해 SIS 로 전송되게 된다. 아래 (그림 5)는 SIC 가 SIS 를 호출한 후 상태정보 데이터를 전송하는 프로그램 소스코드의 일부이다.

```
StateInformationServer server = null;
while (true) {
    try {
        server = (StateInformationServer)
            Naming.lookup("rmi://127.0.0.1:1099/SIS");
        server.addStateInformation(
            StateInformationCollector.getInstance().
                getNowStateInformation());
    };
    Thread.sleep(interval);
} catch (Exception e) {
    System.err.println(e.getMessage());
}
```

(그림 5) 상태정보 전송 프로그램 소스코드

**4.2 상태정보 서버의 구현**

SIC 로부터 상태정보 데이터를 수신 받아 중앙관리 DB 에 저장하는 Java RMI 서버 프로그램인 SIS 는 SIC 와 마찬가지로 순수 Java 언어와 SQL 을 이용하여 구현하였다. 아래 <표 2>는 SIC 로부터 전송되어 중앙관리 DB 에 저장되는 상태정보 데이터의 구조이다.

<표 2> 상태정보 데이터 구조

순서	항목명	값
1	일련번호	데이터베이스 관리번호
2	전송일시	YYYY-MM-DD HH24:MI:SS
3	IP 주소	xxx.xxx.xxx.xxx
4	DB 명	데이터베이스 이름
5	상태정보 1	상태정보 1 측정 값
...	...	...
N	상태정보 N	상태정보 N 측정 값

**4.3 상태정보 분석기의 구현**

SIA 는 중앙관리 DB 에 누적되어 있는 상태정보 데이터를 주기적으로 분석한 후 기준치를 초과하는 항목이 있을 경우 이를 메시지화하여 트위터를 통해 해당 관리자에게 전송하는 Java 프로세스로써 순수 Java 언어와 SQL, 그리고 트위터와의 연동을 위한 Java 언어기반 프로그램 API 인 Twitter4J 를 이용하여 구현하였다. 아래 (그림 6)은 상태정보 측정 값을 'S' 라 하고 기준치를 'L', 기준치 연속 초과 회

수를 'C', 전송메시지를 'M' 이라고 할 때 SIA 가 기준치를 초과하는 상태정보 측정값을 감지하고 이를 관리자에게 통보하는 알고리즘이다.

```

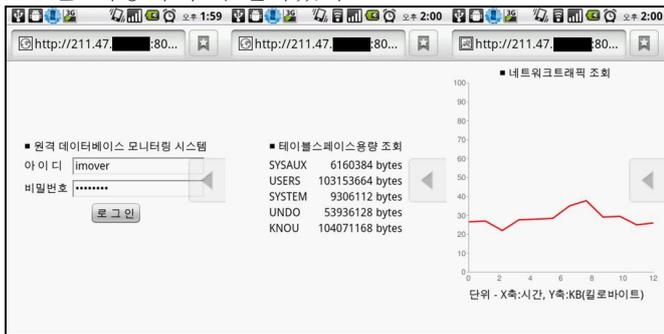
if (S1 >= L1) {
    C1++;
    M.addMessage (1, S1, C1);
} else { C1 = 0; }
...
if (Sn >= Ln) {
    Cn++;
    M.addMessage (n, Sn, Cn);
} else { Cn = 0; }

if (M.size() > 0) {
    sendMessage (M);
};
    
```

(그림 6) 상태정보 분석 알고리즘

4.4 사용자 인터페이스의 구현

웹 애플리케이션 기반 사용자 인터페이스인 SIUI 는 JSP, SQL, HTML, CSS, Java Script, Google 차트 API 를 이용하여 구현하였다.



(그림 7) 모니터링 사용자 인터페이스

위 (그림 7)은 LG-KH5200 스마트폰에 탑재된 웹 브라우저를 이용하여 실제 구현된 SIUI 에 접속한 화면을 캡처한 것으로 가장 좌측에 있는 화면은 사용자 인증을 구현한 것이며, 중앙에 위치한 화면은 관리대상 원격 데이터베이스의 테이블스페이스용량을 조회하는 화면이다. 마지막으로 가장 우측에 있는 화면은 데이터베이스의 네트워크 트래픽 양을 시간대별(X 축: 시간, Y 축: Kilobytes) 그래프로 표시한 것이다.

4.5 성능평가

기존 상용화된 CS 기반 모니터링 방식과 제안 시스템을 아래 <표 3>을 통해 비교하였다.

<표 3> 기존방식과 제안시스템의 비교

항목	기존방식	제안시스템
아키텍처	Client-Server 방식	웹 기반 중앙관리방식
단말기	PC	스마트폰, PC
프로그램	전용프로그램 (유지보수 필요)	웹 브라우저
실시간 경고	미지원	지원(트위터)

데이터베이스 연결세션	관리자 접속 수 만큼 유지	1 개만 유지
-------------	----------------	---------

위 <표 3>에서의 비교를 통해 알 수 있듯이 본 논문에서 제안한 시스템을 이용할 경우 기존 상용화된 CS 기반 모니터링 방식에 비해 보다 효율적으로 원격 데이터베이스를 모니터링할 수 있다.

5. 결론

본 논문에서는 스마트폰 SNS 를 이용한 원격 데이터베이스 모니터링 시스템을 제안하고 구현하였다. 제안한 시스템을 통해 원격 데이터베이스를 모니터링 할 경우 기존 상용화된 CS 기반 모니터링 방식에 비해 아래와 같은 장점들이 있다.

- 모니터링 시 웹 애플리케이션 기반 중앙관리서버를 이용하기 때문에 사용자 프로그램 유지보수가 필요치 않으며, 데이터베이스 성능에 영향을 미치지 않는다.
- 휴대가 편리한 스마트폰을 이용하여 시간과 공간의 제약 없이 언제 어디서든 원격 데이터베이스를 모니터링 할 수 있다.
- 신속한 전파와 편리한 공유가 장점인 트위터를 원격 데이터베이스 모니터링에 사용함으로써 데이터베이스에 문제 발생 시 관리자가 이를 즉시 통보 받을 수 있다.

향후 연구 과제로는 SNS 외에 스마트폰에서 제공되는 여러 가지 기능과 서비스를 활용한 원격 데이터베이스 관리 방법에 대한 연구가 될 것이다.

참고문헌

- [1] Toad for Oracle, "http://www.toad.co.kr/", QUEST SOFTWARE
- [2] Lab128, "http://www.lab128.com/", 128 Consulting
- [3] Twitter, "http://www.twitter.com"
- [4] Twitter4J, "http://twitter4j.org/en/index.jsp"
- [5] Oracle Database, "http://www.oracle.com/kr/index.html"
- [6] Google 차트 API, "http://code.google.com/intl/ko/apis/chart"
- [7] 조경연 외 7명, "WAP 에 기반한 원격 데이터베이스 관리 시스템", 한국정보과학회, 2008