

DB성능관리 툴 MAXGAUGE: 웹 버전 설계 및 구현

김동균¹ 구홍민¹ 박락빈² 이상원¹
성균관대학교¹, ㈜EXEM²

{ dongqsarang⁰, hong3499}@hotmail.com, lbpark@ex-em.com, swlee@acm.org,

DB Performance Management Tool MAXGAUGE: Design and Implementation of its Web Version

Donggeun Kim⁰, Hongmin Ku, Larkbin Park, Sang-Won Lee
Sungkyunkwan University and EXEM Co. Ltd.

요 약

대량의 정보를 빠르게 처리하기 위해 DBMS의 성능관리는 필수적이고 이를 뒷받침하기 위해 많은 툴들이 있다. 이들 제품 중의 하나인 국산 Maxgauge는 오라클 데이터베이스의 성능 현황이 어떤지 조회하고, 설정된 임계 값을 초과하는지 감시하고, 성능 문제 발생시에 원인을 추적하는 기능을 제공한다. 하지만 이는 설치의 번거로움과 국한된 환경에서만 서비스를 받을 수 있다는 한계가 있기 때문에 우리는 웹을 통해서도 시간과 장소에 구애 받지 않고 성능을 관리할 수 있게끔 클라이언트/서버 버전을 웹 버전으로 설계 및 구현 하였다.

1) 서론 2) MAXGAUGE 소개 3) Web버전 설계 및 구현 4) 결론 및 향후 구현방향 5) 참고문헌

1. 서론

최근 대량의 정보에 대한 신속한 응답속도를 요구하는 기업들이 급속히 증가하는 추세지만, 실제로 많은 성능저하 요인으로 인해 최적의 서비스를 제공하지 못하고 있는 실정이다. 이에 성능관리가 필요하지만 실시간에 문제 진단과 처방이 이루어져야 한다는 것과 같은 현실적 한계를 느끼게 된다. 그러나 DBA가 하루 24시간 항상 단말기 앞에서 실시간 감시 및 진단 업무를 수행할 수 없고 실시간 감시 중에 성능문제가 발생하더라도 DB의 재 구성, 세션의 강제 종료 등의 어려움이 있다. 이러한 이유로 하나의 전산시스템을 구성하고 있는 DB, 시스템, 어플리케이션 등 각각의 컴포넌트 툴에 대한 장애 및 성능관리를 효과적으로 지원하기 위해 다양한 툴을 제공하고 있다. 대표적인 국산 제품으로 ㈜EXEM의 MaxGauge라는 제품이 있다. 현재 MaxGauge는 Client-Server기반으로 동작하는데 설치의 번거로움과 이용 환경의 제약사항을 극복하고자 웹 버전을 개발하게 되었다.

2. MAXGAUGE 소개

2.1 주요기능

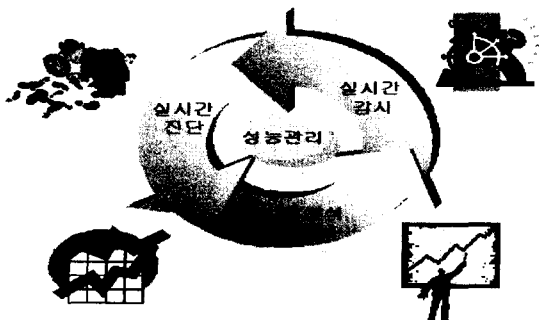


그림 1. MaxGauge 제품 기본 개념

2.1.1 실시간 감시: 복수 개의 오라클 인스턴스를 한 화면에서 모니터링 함으로써 산재되어 있는 오라클 성능문제를 손쉽게 인지하고 대처할 수 있다. 또한 하나의 지표에 대해 여러 인스턴스를 비교해서 모니터링 할 수 있다.

2.1.2 실시간 진단: 하나의 오라클 인스턴스에서 발생된 성능문제의 원인을 유발시킨 세션을 마우스 클릭에 의해 손쉽게 추적할 수 있을 뿐만 아니라 특정 세션에 대해 현재 일량 및 대기, SQL 정보를 한 화면에서 유기적으로 모니터링 함으로써 세션의 성능 현황을 실시간에 파악할 수 있다.

또한 하나의 인스턴스에 접속된 여러 개의 세션들을 조건에 의해 검색하고 조건에 맞는 세션 집합들을 동시에 감시한다. 하나의 세션에 대해서는 일량 정보, 대기정보, 현행 SQL 문장, 접속 정보 등을 조회할 수 있다.

2.1.3 사후분석 및 튜닝: 오라클 인스턴스에서 발생하는 Performance Statistics, Waiting Event 등의 성능정보와 각 시점 별 세션 정보와 SQL 정보를 제공함으로써 과거의 성능 문제를 정확히 분석하고 진단함으로써 최선의 성능을 보장한다.

그 외 락(Lock) 추적, 병렬수행(PQ) 추적, 원격 세션 추적, 지표별 세션 추적, SQL 검색, SQL 플랜 조회, 오브젝트 조회, DBA 지원 등의 기능을 제공한다.

2.2 구조

구조는 그림 2와 같고 각각의 세부기능은 다음과 같다.

2.2.1 RTAD(Real-Time Access Daemon): DB 서버 내에서 수행되는 프로세스로서 SGA(Shared Global Area) 메모리를 직접 액세스하여 오라클의 성능정보를 수집하고 O/S kernel로부터 운영체계의 성능정보를 실시간 수집하는 프로세스다.

2.2.2 LOGD(Logging Daemon): DB 서버 내에서 수행되는 프로세스로서 RTAD와 동일한 방식으로 성능정보들을 수집하여 수집된 정보를 파일에 기록한다.

2.2.3 Client: RTAD와 소켓을 통하여 접속하여 실시간 정보를 조회함으로써 오라클 DB에 대한 모니터링 및 진단을 수행한다.

2.2.4 Log Controller: LOGD와 소켓을 통하여 접속하여 현재 LOGD의 현황을 감시할 수 있고, 각종 로깅 제어 작업을 수행한다.

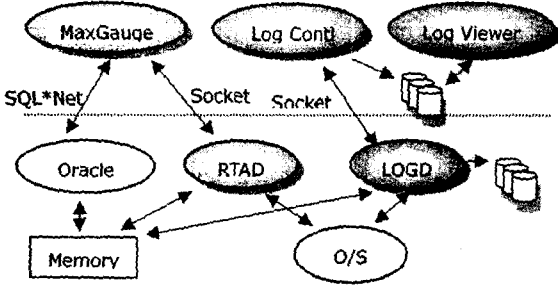


그림 2. MaxGauge 수행 구조

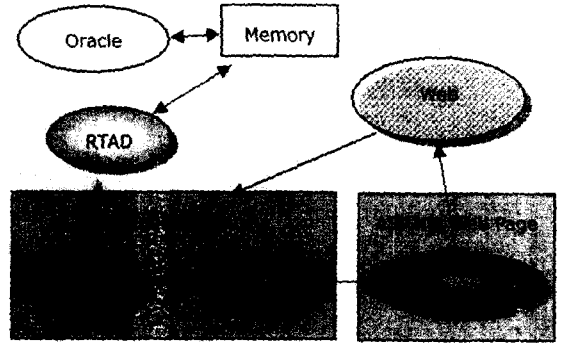


그림 4. 윈도우 환경에서의 MaxGauge 웹 버전 아키텍처

3. MAXGAUGE 웹 버전

3.1 목적 및 기능

기존의 서버/클라이언트 방식은 회사내의 클라이언트 프로그램이 설치된 PC에 한하여 모니터링을 할 수 있었던 반면 회사외부에서도 손쉽게 웹을 이용하여 모니터링을 할 수 있게 해보자는 취지에서 제작하였다. 주요 기능으로서는 앞서 소개한 MAXGAUGE에서 사후처리 및 튜닝을 제외한 모니터링에 중점을 두어 개발하였다.

3.2 구조

일반적인 Web환경을 구성하기 위해 다음의 구조를 생각한다. 그림 3과 같은 Unix서버에 jsp 언어를 사용하는 것과 그림 4의 IIS서버 기반에 Asp언어를 사용하는 것이다. 이 두 구조 중 관리측면, 비용측면 등등의 요소를 저울질 한 결과 서비스 대상이 소수 몇 명인 점과 별도의 관리부담이 없다는 점을 고려하여 기존의 window가 설치된 일반 PC환경에 IIS를 이용하여 서비스 하기로 했다. IIS기반의 자세한 구조는 다음과 같다.

Stier구조를 사용하여 사용자가 ASP.Net에 접속시 서버측에서 C#으로 구성된 DLL을 통하여 오라클 서버측 모듈(RTAD)과 소켓 통신으로 지정된 시간(default 3초)마다 데이터를 사용자측에 Java script를 통하여 뿌려주게 된다.

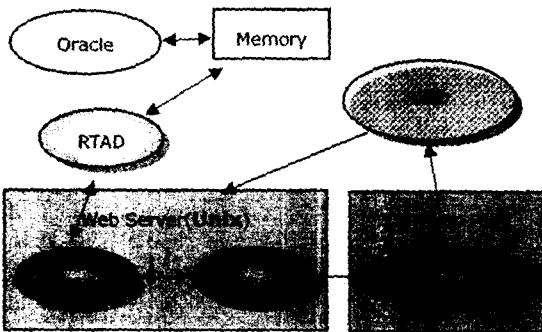


그림 3. Unix 환경에서의 MaxGauge 웹 버전 아키텍처

3.3 설계 및 구현

본 시스템은 다음과 같은 설계원칙에 따라 설계하였다. 기본적인 구성으로 서버는 IIS를 언어는 ASP.Net을 사용하고 클라이언트 측 사용자들에게 보여질 웹 페이지는 Java Script, VB Script, Dhtml, Vml 을 이용하여 보여지게 된다.

3.3.1 서버

RTAD와 직접 소켓을 통하여 데이터를 주고 받는 형식으로 다음과 같이 구현된다.

3.3.1.1 서버측 부하는 최소로 한다.

보통 3초시간 간격으로 데이터를 요청하게 되는데 구조가 복잡하고 메모리를 많이 사용하게 된다면 연산시간이나 응답속 면에서 상당한 낭비가 있다. 단순히 RTAD로부터 받은 데이터를 바로 클라이언트에 넘겨주는 역할만 담당한다.

3.3.1.2 별도의 추가 구성 비용이 들지 않는다.

오라클이 설치된 서버와는 별도로 RTAD와 접속할 윈도우 기반의 개인용 PC를 서버(IIS)로 활용한다.

3.2.1.3 C#기반의 ASP.NET사용으로 재사용성 극대

소켓연결부분을 C#기반의 dll(Dynamic Link Library)을 이용해서 처리하므로 소스의 유지, 보수가 용이하다.

3.3.2 클라이언트

서버로부터 데이터를 받아 화면상에 실시간으로 그래프와 지표표를 뿌려주는 역할을 한다. 실시간 그래프에서 가장 중요한 점인 속도측면에서 최상의 결과를 얻게 하기 위해 다음과 같은 방식을 채택하였다.

3.3.2.1 서버와의 접속횟수를 최소로 한다.

최초 한번 필요한 데이터(예를 들어 각각의 지표들의 이름)는 처음에 받아서 따로 저장함으로써 걸리는 시간을 최소화한다.

3.3.2.2 Xml기반의 GUI제공

웹 페이지에 그래프를 그리는 방법으로 크게 두 가지를 고려했다. 첫째, 서버측에서 그린 그래프를 이미지로 만들어 클라이언트에 실시간으로 전송하는 방법. 둘째, vbscript를 사용한 DOM을 사용하여 Xml기반의 웹 그리기 도구인 Vml(Vector

Markup Language)을 이용하는 방법이다. 기존에 웹에서 사용하는 증권, 날씨 같은 이미지 저장방식은 서버측에서 이미지를 만들어 전송시 부하가 상당히 커지고 또한 클라이언트측 이벤트시에 서버측에 다시 접속을 유도하게 된다. 즉 전체적인 성능에서 결함을 보인다.

3.3.2.3 javascript를 이용한 클라이언트측 모듈

ASP.NET에서는 크게 두가지 구분을 둔다. 서버컨트롤과 클라이언트 컨트롤이 그것이다. 처음에는 서버컨트롤을 이용하여 RTAD로부터 받은 데이터를 저장하고 웹으로 표현하기 위해 Asp.net behind code(C#)에서 접속, 저장 및 전송을 담당하였으나 이는 Asp.net이 서버측에서 돌아가는 특징 때문에 멀티유저 접속시 서버측에 상당한 부하를 줌으로써 서버가 죽어버리는 문제를 발생시켰다. 또한 사용자 PC에 framework를 요구한다. 대안으로 클라이언트컨트롤을 이용하여 사용자들에게 서버의 부담을 나누는 방식으로 클라이언트에서 동작하는 javascript를 이용하여 데이터를 가공하고 표현한다.

3.3.2.4 히든(hidden) 프레임을 이용한 데이터 갱신

또한 데이터를 그래프로 가공하여 화면에 뿌려줄 때 걸리는 시간은 엄청나다. 시행착오의 결과 보통 3초의 시간마다 갱신을 하는데 데이터를 받아오는 시간을 1이라 하면 표현하는데 드는 시간은 2라고 할수 있을 정도다. 결국 3초마다 갱신되는 것이 아닌 좀더 지연된 4-5초의 시간이 걸리게 된다. 이러한 문제점을 개선하기 위해 히든 프레임을 둔다. 바로 이곳에서 데이터를 전송받아 가공하여 저장하고 그래프에 뿌려주기 바로 직전 단계의 일을 수행한다. 나머지 나타나는 프레임은 단지 뿌려주기만 하는 역할을 함으로써 갱신에 걸리는 부하를 상당부분 감소시킬수 있다.

3.3.2.5 cookie를 이용한 사용자 관리

사용자마다 달리 요구하는 지표를 cookie를 통해 관리한다.

3.4 구현

그림 5는 서버측에서 받은 데이터의 변화값을 계산하여 그래프를 통해 시각화해서 표현한 화면이고 그림 6은 각각의 지표를 클릭했을시 특정 세션에 대해 현재 일할 및 대기, SQL 정보를 나타내는 화면이다.

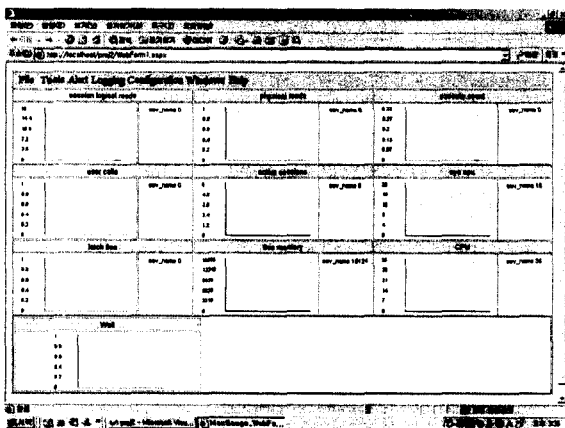


그림 5. 멀티 모니터링 화면

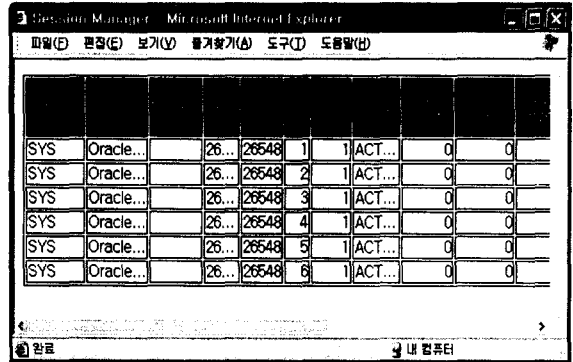


그림 6. Session Detail 화면

4. 결론 및 향후 구현 방향

4.1 결론

DB, 시스템, 애플리케이션 등 각각의 컴포넌트 둘에 대한 장애 및 성능관리를 효과적으로 지원하기 위한 성능관리 도구인 MaxGauge를 주 5일 근무의 확산, VPN의 보급 등으로 인한 재택근무가 활성화되고 또 출장업무가 많은 DBA에게 시간과 공간 제약 없이 모니터링할 수 있게 웹 버전으로 개발하였다. 이 제품이 MaxGauge와 함께 Package화 되어 많은 DBA들에게 보다 편한 서비스를 제공하기를 기대한다.

4.2 향후 구현 방향

보다 상세한 Session Detail(세션 추적기능, 세션 모니터링 기능 담당)화면 DHTML을 이용한 event 및 tree를 구현해야 하고 현재 MaxGauge는 다중 RTAD와 통신이 가능 하나 웹 버전은 현재의 구조대로라면 1개의 RTAD만 통신 가능하다. 이를 dll에서 thread를 통해 동시에 여러 개의 RTAD와 통신이 가능하게끔 구현하거나 또는 IIS 내부적으로 port를 달리하여 port별로 서로 다른 RTAD 관리하는 방안이 필요하다.

5. 참고문헌

- [1] ㈜엑셈, " User Guide of MAX GAUGE" , pp2-6, 2002.
- [2] 김순근, " ASP.Net programming Bible" , 영진, 2002.
- [3] Sean Dillon, " Beginning Oracle Programming" , 정보문화사, 2002.
- [4] IT포탈사이트(데브피아), <http://www.devipia.com/>
- [5] Microsoft MSDN <http://msdn.microsoft.com/>
- [6] 오치훈 외 1명, " 최신 JavaScript 사전 " , 성안당 2003

** Acknowledgement: 본 연구는 김동근, 구홍민이 2004년 겨울 방학 동안에 ㈜엑셈에 인턴십 학생으로 파견된 동안 이루어졌다. 이러한 기회를 제공해주신 ㈜엑셈측에 감사함을 표한다.