WAS성능 개요 및 Oracle9*i* Application Server의 성능 튜닝

글 | 장성우, 이진호 (한국오라클 iPlatform BD 본부) sungwoo.chang@oracle.com, jinho.lee@oracle.com

최근 WAS(Web Application Server)가 기업 전산 시스템의 중요 플랫폼으로서 인식되고 있다. WAS는 기업의 기간 정보 시스템과 웹 전산환경 사이에 위치한 일종의 미들웨어로서, 웹에 기반한 분산 시스템을 쉽게 개발할 수 있도록 도와주고 안정적인 트랜잭션 처리를 보장해 주는 역할을 수행한다 . 이 글에서는 WAS의 개발과 운용에 있어 주요 요소인 WAS의 성능 측정 방안에 대해 개략적으로 설명한 후, 오라클의 WAS 제품인 Oracle9*i* Application Server 활용시 적용 가능한 성능 튜닝 방안을 소개한다.

일반적으로 WAS는 웹 브라우저와 기업 내 데이타베이스 시스템의 중간에 위치해서 웹 기반의 3계층 클라이언트/서버 시스템의 한 구성요소를 이루며, 웹 브라우저를 통해 들어오는 사용자의 각종 데이타 요청에 응답하고 비즈니스 로직 등을 실행한다. 이런 점에서 최근 WAS는 대규모 트랜잭션을 처리해야 하는 기업의 웹 전산 환경에서 필수적인 개발 플랫폼으로 급부상하고 있다.

일반적인 WAS 기반 웹 시스템은 접속한 사용자가 적어서 단위 시간에 WAS가 처리할 수 있는 것보다 적은 양의 요청이 들어오면 사용자는 아주 빠르게 요청에 대한 응답을 받을 수 있다. 하지만 같은 시스템에 대해 사용자가 몰려서 단위 시간에 WAS가 처리할 수 있는 용량 이상의 요청이 한꺼번에 몰리게 되면 WAS의 처리 가능 용량 이상의 초과 요청은 앞의 요청이 처리될 때까지 보류(대기)되었다가, 앞의 요청들이 처리되면 순차적으로 나중에 처리되므로 사용자 입장에서는 응답을 지루하게 기다리게 되는 상황이 발생하게 된다. 이러한 상황이 심해지면 사용자는 해당 시스템에 대해 불평을 하게 되고 최악의 경우에는 아예 접근조차 않는 상황이 발생하게 된다. 이러한 상황을 어떻게 이해해야 할까? 또한 이러한 상황에서는 어떻게 대처해야 할까? 이러한 문제에 대한 답을 구하는 것이 이 글의 핵심이다.

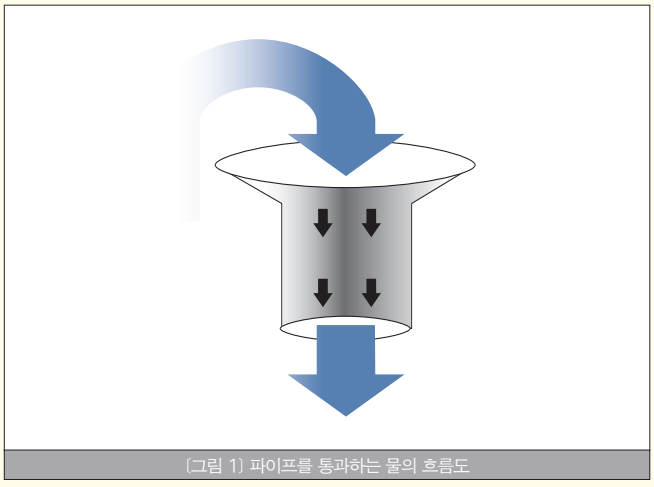
WAS 성능 개요

WAS의 성능 원리

앞서 잠시 언급한 것처럼, WAS는 사용자와 데이타베이스 사이에서 사용자의 요청을 받아 요청 자료를 넘겨 주거나 데이타베이스를 이용하는 특정 비즈니스 로직을 실행하여 수행 결과를 되돌려 주는 역할을 수행한다. 따라서 WAS를 가운데 두고 지속적으로 사용자와 데이타베이스 사이에 데이타가 흘러다니는 상황이 발생하게 된다. 이러한 상황을 가장 잘 표현할 수 있는 방법은 무엇일까? 그것은 바로 파이프 모델이다.

〔그림 1〕에서 수도 파이프가 처리할 수 있는 양보다 적은 양의 물이 흘러 들어오면 당연히 물은 곧 바로 흘러 내려가게 된다. 수도 파이프가 처리할 수 있는 양과 같은 양만큼의 물이 흘러 들어와도 물은 곧 바로 흘러 내려갈 수 있다. 하지만 파이프를 통해 통과할 수 있는 양보다 더 많은 물이 흘러 들어오면 처리량 보다 많은 양만큼은 깔때기 위에 쌓이게 된다. 지속적으로 처리량보다 많은 양의 물이 흘러 들어오게 되면 단위 시간당 유입량에서 파이프가 처리할 수 있는 양을 뺀 만큼의 초과분은 계속적으로 이전의 초과분 위에 쌓이게 되어 결과적으로 깔때기 너머로 물이 넘치게 된다.

앞서 제시한 사용자 폭주 시의 지루한 응답 대기 상황은 이러한 파이프 모델을 통해 설명될 수 있다. 즉, WAS가 단위 시간당 처리 가능한 작업의 양보다 많은 양의 요청이 몰리게 되면 초과 요청들은 지속적으로 쌓이게 되고, 앞의 요청이 처리될 때까지 기다린 후에야 비로소 요청이 처리되므로 사용자들은 많은 시간을 기다리게 되는 것이다. 따라서, 사용자 폭주시의 대응 방안은 바로 지름이 큰 수도 파이프를 사용하여 단위 시간당 물의 통과량을 늘리듯이 단위 시간당 처리량을 증가시켜 줄 수 있는 방안을 찾아내는 것이다. 즉, 보다 성능이 좋아지도록 WAS 및 관련 시스템의 운용 환경을 개선하여 주는 것이다.



〔그림 1〕 파이프를 통과하는 물의 흐름도

WAS 성능 척도

그렇다면 WAS의 성능이 좋아졌음을 어떻게 알아볼 수 있을까? 일반적으로 WAS의 성능 척도로서는 보통 TPS(Transaction per Second)와 ARS(Average Response Time)를 활용하고 있다. TPS는 단위 시간당 최대 처리 건수를 나타내며, ARS는 요청 후로부터 결과가 도착할 때까지의 평균적인 대기 시간을 나타낸다. TPS와 ARS는 〔그림 2〕와 같은 상관관계를 갖는다.

〔그림 2〕에서 보면 액티브 클라이언트의 수가 증가함에 따라, 즉, 사용자 요청이 증가함에 따라 초기부터 일정 시점까지는 TPS가 증가하지만 일정 시점 이후로는 TPS가 어느 수준에서 머무르게 된다. 이 수준이 바로 WAS가 단위 시간에 처리할 수 있는 업무 건의 최대값을 나타내게 되므로, 바로 WAS의 성능을 나타내는 최대 TPS가 된다. 최대 TPS에 이르기까지 ARS는 일정하게 나타난다. 앞서 파이프 모델을 견주어 설명하면 파이프의 처리 용량에 다다르기까지는 물의 양이 증가하더라도 다 곧바로 파이프를 통과할 수 있기 때문이다. 하지만, 최대 TPS를 통과하면, 즉, 파이프의 처리 용량 이상의 물이 유입되면 초과분만큼은 나중에 처리되므로 결과적으로 대기시간이 발생하게 되고, 따라서 평균 응답 시간인 ARS의 값은 상승하게 된다. 이것이 WAS의 성능을 나타내는 두 요소인 TPS와 ARS의 변화 추이 특성이다.

그렇다면 두 개의 WAS 시스템 A, B를 비교했을 때 어느 특정 시점에서 A 시스템의 TPS가 B 시스템의 TPS보다 높고 또한 A 시스템의 ARS가 B 시스템의 ARS보다 낮다고 해서 A 시스템이 B 시스템 보다 성능이 좋다고 말할 수 있을까? 여기서 결코 간과해서는 안 되는 사항은 바로 이러한 값들은 액티브 클라이언트, 즉, 현재 시스템에 접속하고 있는 사용자의 수에 따라 변할 수 있고, 또한 그 값이 달리 해석될 수 있는 가변

치라는 점이다. 이것은 WAS의 성능을 측정함에 있어 절대 간과해서는 안 되는 중요한 포인트인데, 이 부분에 대한 구체적인 근거의 언급 없이 어느 시점의 값만을 보고서 A 시스템의 성능이 좋다고 기술하는 것은 심각한 문제를 야기할 수 있다. 이러한 배경을 이해하기 위해 Java 전문가들 사이에서 인정 받고 있는 Java 전문가인 이원영씨의 성능 관련 자료를 인용해 보도록 하겠다.

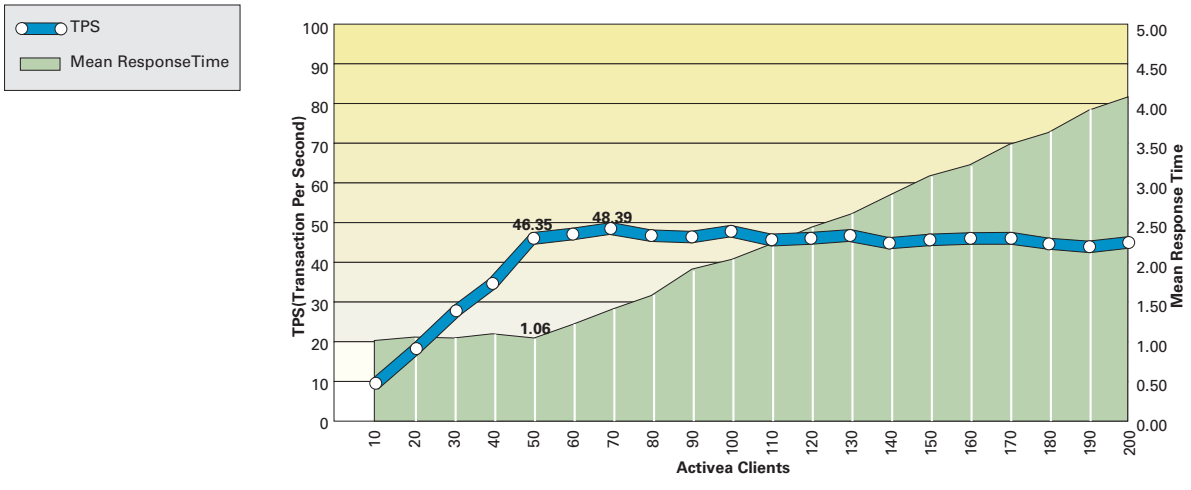
[그림 3]의 그래프는 A, B 두 시스템의 동시 사용자 숫자의 증가에 따른 TPS와 ARS의 변화를 보여주는 예제이다. 여기서 중요한 것은 액티브 클라이언트의 숫자가 어디 있을 때 TPS와 ARS를 보느냐에 따라 전체적인 해석이 달라질 수 있다는 점이다. 만약 'Active Clients = 50'일 때만을 놓고 본다면, 당연히 A 시스템이 처리량도 많고 평균 응답 시간도 빠르기 때문에 훨씬 좋은 시스템이라고 할 수 있다. 하지만, 'Active Clients = 100'이라고 봤을 때는 해석이 완전히 바뀔 수 있다. 이 경우에는 당연히 B 시스템이 TPS와 ARS 측면에서 더욱 좋은 성능을 보이고 있음을 알 수 있다. 이렇듯 WAS 시스템의 성능은 접속된 동시 사용자의 수의 변화에 따라 다른 양상을 보일 수 있기 때문에 어느 한 시점에서의 값만을 놓고 자의적으로 해석해서는 완전히 다른 결과를 보일 수 있다. 따라서, 다양한 성능 측정 결과를 놓고 분석을 해야만 그 우열을 가릴 수 있게 된다. 이는 WAS를 이용하는 사람이라면 꼭 알고 있어야 하는 중요한 사항이다.

WAS 성능 튜닝 방안

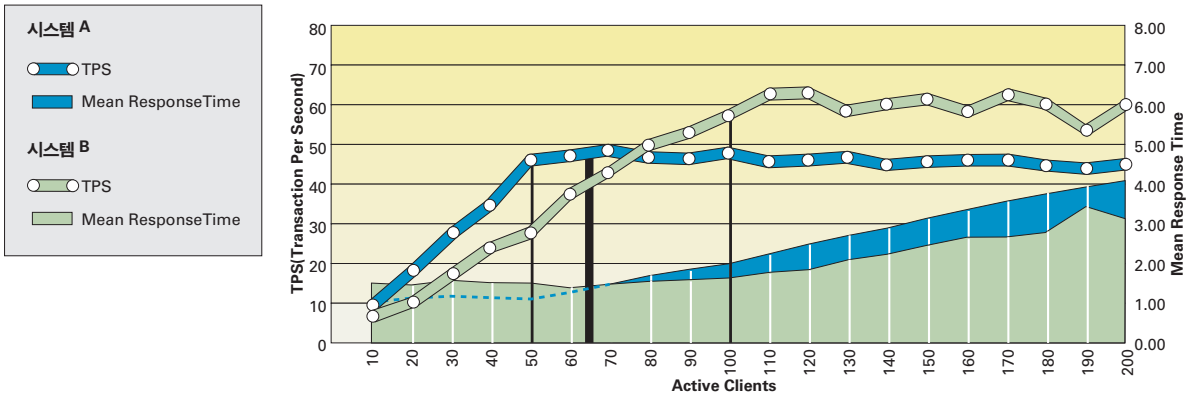
WAS의 성능에 영향을 주는 요인들

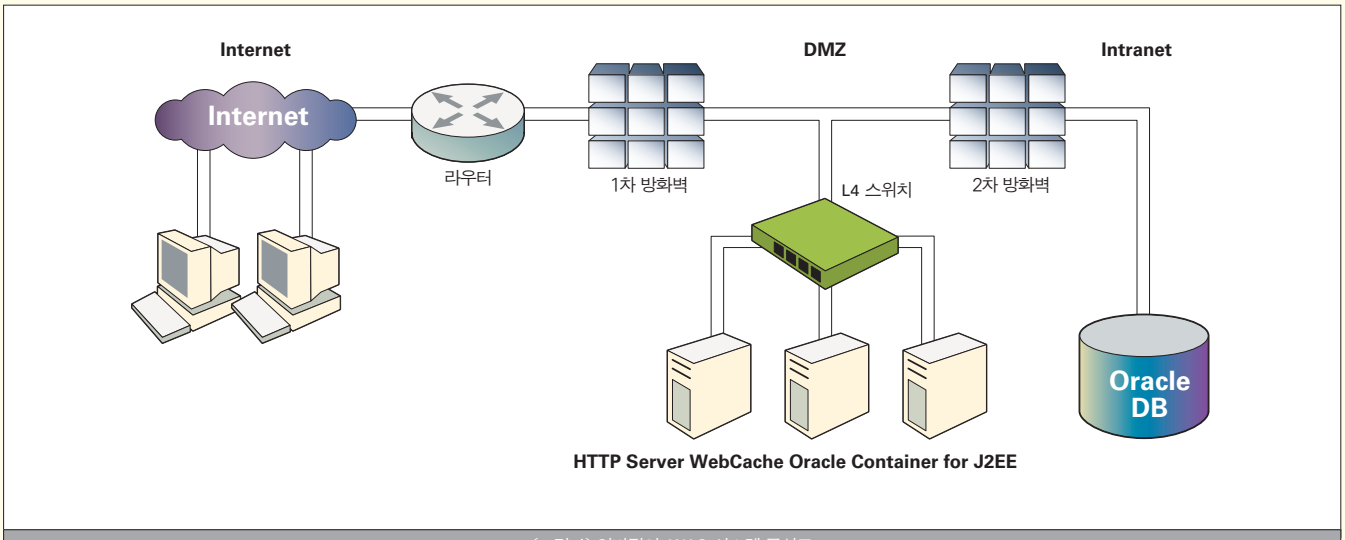
WAS 기반의 웹 전산 시스템의 구축에는 L4 스위치, 방화벽, WAS와 DB의 운용을 위한 하드웨어 시스템 등 다양한 요소들이 필요하다. 이러한 요

[그림 2] TPS와 ARS의 상관관계 그래프



[그림 3] 두 시스템 A, B의 액티브 클라이언트의 변화에 따른 TPS/ARS 변화 그래프





〔그림 4〕 일반적인 WAS 시스템 구성도

소들로 이루어진 WAS를 중심으로 한 일반적인 웹 시스템의 구성도는 〔그림 4〕와 같다.
그러면, 이러한 시스템 구성에서 WAS 성능에 영향을 주는 요인들을 정리해 보자.

◉ **네트워크 용량** | 〔그림 4〕의 시스템 구성요소인 각 노드들간의 네트워크 대역폭에 의해 전체 WAS의 수행 성능이 큰 영향을 받는다. 연결 네트워크는 다음과 같다.

- 사용자와 라우터
- 라우터와 방화벽
- 방화벽과 L4 스위치
- L4 스위치와 웹 서버
- 웹 서버와 WAS 서버
- WAS 서버와 DB 서버

이들 중 한 곳이라도 네트워크 연결에 문제가 있는 경우, 즉, 네트워크의 대역폭이 충분하지 못한 경우에는 심각한 성능 장애로 이어질 수 있다. 이것은 앞서 파이프 이론에 견주어 살펴보면, 대형 송수관끼리 연결되어 있더라도 중간에 적은 용량의 파이프가 하나 끼어 있다면 이 작은 파이프 때문에 전체의 물 흐름에 문제가 생기는 것과 마찬가지 이치이다.

◉ **라우터 성능** | 라우터는 사용자의 모든 트래픽을 WAS 시스템이 있는 지역으로 경로 배정해 주는 노드이다. 주로 UNIX 시스템 운영체제가 사용될 수 있으며, 성능을 위해 전용 하드웨어가 사용되기도 한다. 대부분 인트라넷에서는 전체 회사에서 사용하는 네트워크 용량을 산정하여 매우 큰 성능으로 설계되어 있기 때문에 큰 문제가 되지는 않는다.

◉ **방화벽** | 방화벽은 외부 네트워크에서 WAS 시스템으로 오는 모든 패킷을 조사하여 불법 침입 등을 감시하는 기능을 수행한다. 보통 여기에서 가장 많은 부하가 걸린다. 소프트웨어 방화벽이 많이 쓰이며 각 제품마다 최대 대역폭을 제시한다. 아주 대용량인 경우, 방화벽을 병렬로 쓰기도 하며, 때로는 전용 하드웨어 방화벽을 사용하기도 한다. 소프트웨어 방화벽과 하드웨어 방화벽의 가격차는 5~10배 정도 나지만, 소프트웨어 방화벽의 경우 별도의 시스템을 구입하는 비용도 계산해야 한다.

◉ **L4 스위치의 용량** | L4 스위치의 역할은 사용자의 트래픽을 동일한 기능을 수행하는 여러 가지 서버에 분배하는 기능을 수행한다. 보통 한 대의 L4 장비에서 최대 10~20대의 장비까지 분배해 준다. 보통 포탈 사이트는 수십 대의 웹 서버를 사용하며, 이 경우 L4 장비를 트리 구조의 다단계로 연결하여 구성한다. L4 스위치의 분배 방식은 Round-Robin 방식과 Hash Map 방식을 들 수 있다. RR 방식은 사용자의 요청을 순차적으로 각 WAS에 나눠주는 방식이며, Hash 방식은 내부 Hash Map을 통해 특정 IP 값에 대한 Hash 값에 따라 분배하는 방식이다. 즉 Hash 방식인 경우, 특정 IP가 WAS A로 연결되면, 이 후에는 무조건 WAS A에 연결되는 방식이다. 두 가지 모두 장단점이 존재하며, 이는 WAS 애플리케이션의 성격에 따라 결정된다.

◉ **WAS 시스템 장비의 성능** | WAS 장비와 WAS 제품 자체의 성능 그리고 DB 장비 및 DB 제품 자체의 성능에 따라 성능 차이가 난다.

WAS에서의 트래픽 제어 방안

최적의 성능을 제공하는 WAS 시스템을 구성하는 기본적인 규칙은 바로 '70% 규칙'이다. 이는 다중 시스템의 설계에서 많이 사용되는 것으로,

연계된 모든 시스템들이 70%의 성능을 내도록 설계함으로써, 병목 현상 없이 최적의 성능을 낼 수 있도록 하는 방법이다. 가령 10Mbps의 LAN 라인 하나가 낼 수 있는 최대 성능은 보통 6~7Mbps로 잡는다. 만약 10Mbps가 온전히 나올 것으로 생각하고 설계된 시스템이 있다면, LAN 선이 병목 현상의 원인이 되는 상황이 발생할 수도 있다.

[그림 5]는 WAS 시스템과 DB, 그리고 네트워크간의 관계를 확실하게 보여준다. WAS 시스템은 주로 데이타가 대부분 DB에서 WAS를 거쳐 네트워크를 통해 대량으로 방출되는 구조로 구성된다. 즉 DB에 부어넣은 물이 WAS를 거쳐 네트워크에 잘 흐를 수 있는 구조로 설계해야 한다. 이를 위해서는 [그림 6]과 같이 모든 서버 및 네트워크의 부하가 70%가 되는 시스템의 최적 구성도를 찾아야 한다.

최적 시스템을 찾기 위한 평가 도구로는 HTTP Stress Tool이 있다. 주의할 점은, Stress Node는 UNIX 환경에서 운영하도록 권고하며, 가능하면 WAS와 동급이고 같은 수의 장비로 테스트하는 것이 좋다. 이때 각 Stress Node 역시 70%의 CPU 이용률을 보여야 효율적인 성능 테스트를 수행할 수 있다.

앞서 최적의 구성도를 찾는 방법은 모든 서버에 걸리는 부하가 70%가 되는 환경을 찾아야 한다고 언급했는데, 만일 데이타베이스는 부하가 많이 걸리고 WAS는 상대적으로 놀고 있는 경우라면 데이타베이스를 증설해야 한다. 만일, 스트레스 노드 머신에 걸리는 부하가 많고 상대적으로 WAS가 놀고 있는 경우라면 스트레스 노드 수를 늘려야 한다. 대부분은 WAS 개수와 DB 개수 간에 최적의 조합을 찾는 것이 1차 목표이다.

WAS 내부 튜닝 방안

대부분의 WAS 시스템들은 내부적으로 실제로 비즈니스 로직을 실행하는 스레드들과 외부적으로 DB와의 연결을 관리하는 DB 커넥션들을 풀(pool)의 형태로 관리한다. WAS 시스템의 튜닝에서 가장 중요한 요소는 이러한 스레드 풀의 스레드들의 개수와 DB 커넥션 풀 내의 커넥션들의 개수를 조절하는 문제이다. 기본적으로 이 두 가지 요소의 값을 어떻게 설정하느냐가 WAS 전체의 성능에 지대한 영향을 주게 된다.

◉ **스레드 풀 설정 방안** | 먼저 스레드 풀 내의 스레드 개수를 상대적으로 작게 설정한다고 가정하자. 그러면, 다음과 같은 결과를 얻을 것이다.

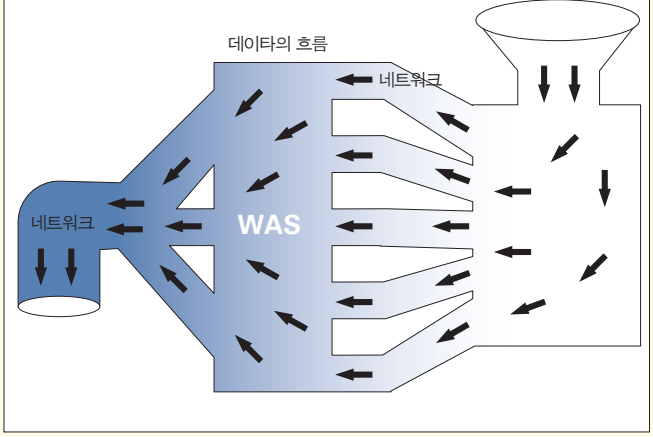
- 최소/평균 응답 시간이 늘어남
- 최대 응답 시간이 줄어듦

즉, 평균 응답시간은 안 좋아지지만 일정한 시간 내(평균 응답 시간의 20~50%)에 모든 처리가 종료되는 안정성을 보장하게 된다. 반대로, 스레드 풀 내의 스레드 개수를 크게 설정하면,

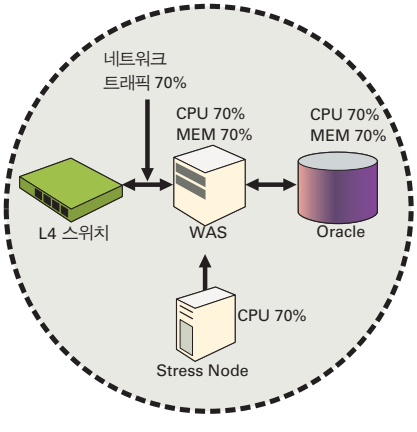
- 최소/평균 응답 시간이 줄어듦
- 최대 응답 시간 폭주(타임아웃 가능성)

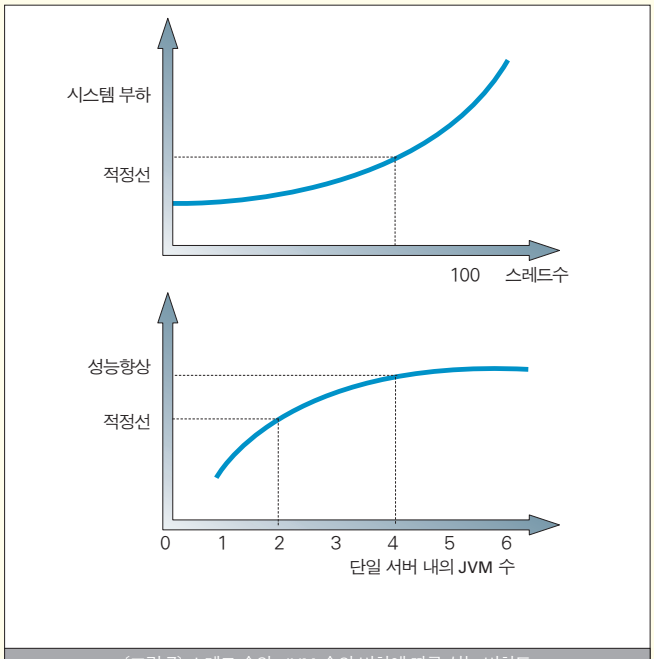
의 결과를 얻게 된다. 즉, 평균 응답 시간은 좋아지지만, 최대 응답 시간이 평균 응답 시간의 10~20배까지 걸리는 결과가 발생한다. 스레드의 크기가 늘어남에 따라 JVM이 이를 처리하기 위한 부하는 급격하게 늘어난다. 따라서, JVM은 많은 양의 스레드를 처리하기에는 부적합한 구조로 되어 있다고 할 수 있다. Sun사는 이러한 이유 때문에 약 100개 이하가 적정선이라고 권고하고 있다. 실제로 테스트를 해보면, 같은 시스템이라도 여러 개의 JVM을 동시에 띄워서 사용하는 것이 하나의 JVM에서 스레드의 개수를 많게 하는 것보다 성능이 더 좋아지는 경우를 볼 수 있다. [그림 7]은 이러한 결과를 보여준다. 일반적으로 JVM 개수의 증설은 기계상의 CPU 개수에 비례해서 이루어지며, CPU 개수를 초과하는 일정 개수 이상은 더 이상의 성능 향상을 기대할 수 없다. 또한 메모리의 사용량이 JVM의 개수에 비례해

[그림 5] WAS 주변의 데이타 흐름도



[그림 6] WAS 시스템 최적의 튜닝 방안





〔그림 7〕 스레드 수와 JVM 수의 변화에 따른 성능 변화도

서 늘어나는 단점이 있다. 그러므로 시스템의 CPU 개수 및 메모리의 양에 따라 업무에 맞는 적정한 JVM의 개수를 선택해야한다.
스레드 개수는 시스템 운용시에는 40개 이상 정도로 해 주는 것이 좋다. 운영 모드에서는 동시에 다양한 작업이 수행되기 때문에 풀의 개수가 너무 작으면 궁핍 상태(starvation)가 발생할 수 있다.

◉ **DB 커넥션 풀의 설정 방안** | 일반적으로 WAS와 DB 사이의 지속적인 연결의 설정과 삭제의 반복 작업으로 인한 과부하를 줄이기 위해, WAS로부터 데이타베이스로의 연결을 미리 생성하여 커넥션 풀을 구성한 후, 사용자가 물리적 연결 과정 없이 바로 DB와 접속하여 사용할 수 있게 방안이다. 한 번 커넥션이 생성되면, WAS가 죽이지 않는 한 계속 라이브 상태로 유지된다. 이로 인해 DB 재연결로 인한 부하(20~30%)를 해결할 수 있게 된다. 이러한 작업은 J2EE 엔진에서 자동으로 해주므로 간편한 설정과 API로 쉽게 사용할 수 있다. 일반적인 커넥션 풀의 구성과 사용 방법은 〔그림 8〕과 같다. DB 커넥션 풀의 크기는 DB를 이용하는 애플리케이션의 특성에 따라서 값을 달리하여 주는 것이 좋다. 즉, 애플리케이션 내에서 DB에 접근하는 SQL 문장의 트랜잭션 속성에 따라서 값을 달리 주는 것이 좋다. 이를 정리하면 다음과 같다.

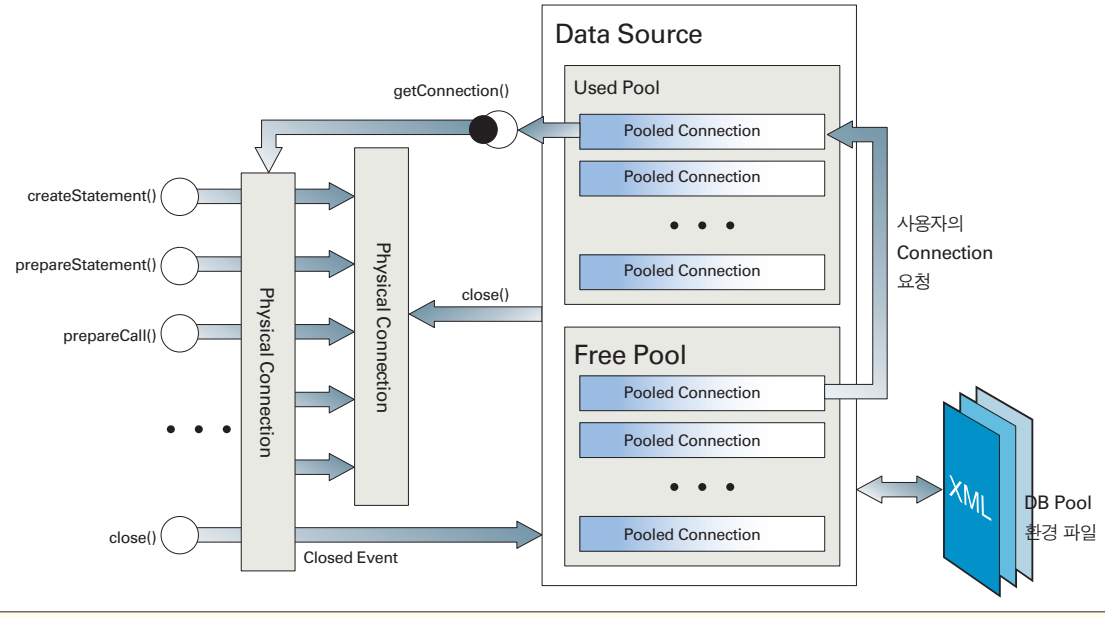
- Short term SQL : 5개 미만
- Long term SQL : 동시에 작업을 수행하는 평균 사용자 수 이상

하지만, 실제 운용시에는 DB 풀의 개수는 40개 이상으로 설정하도록 한다. 너무 작은 값을 일반 운영 환경에 적용하면, 가끔 Long Term Job이 이를 다 점유하는 경우가 발생하며, 이는 결국 다른 DB 처리 문장이 DB에 접근할 수 없게 만들어 모든 서비스가 불가능한 상황이 발생할 수도 있다.

데이타베이스 튜닝과 WAS 튜닝의 차이점

데이타베이스 튜닝과 WAS 튜닝의 차이점을 간략히 정리하면, 다음과 같다.

〔그림 8〕 DB 커넥션 풀 구성도



◉ **데이타베이스 튜닝** | 주로 파라미터 및 SQL 튜닝을 통해, 외부로 나가는 데이타 양보다는 내부적으로 처리하는 데이타 양을 조절하면서 동시에 다양한 로직(SQL)이 얼마나 빨리 끝나는지에 초점을 맞춘다.

◉ **WEB 및 WAS 튜닝** | 데이타베이스 튜닝과 달리 WAS의 경우에는 시스템 전반에 대 한 튜닝이 필요하다. 즉, 네트워크, 게이트웨이, 방화벽, L4 스위치, WAS 및 DB 서버 등 다양하고 전반적인 기술들을 두루 필요로 한다. 또한 실제 문제가 되는 부분을 찾기가 매우 힘들다. 또한, 내부적인 처리량보다는 외부로 나가는 데이타 양에 대한 트래픽 제어가 필수적(우선)이다.
그 이유는 대부분의 WAS 시스템이 DB에서 질의한 결과를 HTML로 포장하여 전송하기 때문에 평균적으로 DB에서 오는 데이타 양에 2~10배의 데이타가 붙어서 사용자에게 전송되기 때문이다.

기타 WAS 시스템 구성시 고려사항

WAS 시스템의 성능은 시스템을 구성하는 방법에 따라 성능이 달라질 수 있다. 시스템 구성시 선택할 수 있는 구성 옵션은 무엇인지 알아보도록 하자.

◉ **웹 서버를 사용할 것인가?** | 현재 추세는 WAS 서버 프론트엔드에 Apache와 같은 웹 서버를 두는 방식을 사용한다. 프론트엔드의 웹 서버에서 HTML과 이미지 등 정적인 자료를 서비스하고, WAS는 애플리케이션 로직을 전담시키자는 목적이다. 그러나 WAS 서버도 웹 서버와 동일한 기능(HTTP 서비스)을 수행할 수 있기 때문에 서버의 여유가 없는 경우에는 굳이 두 시스템 모두를 사용할 필요는 없다.

◉ **웹 서버와 WAS 서버의 물리적 분리** | 웹과 WAS를 동시에 사용하는 경우에는 이들의 배치 문제가 생긴다. 즉 두 서버를 같은 하드웨어 노드에 배치하는 경우에는 시스템 부하나 리소스 등에 문제가 발생할 여지가 많아진다. 주로 이 두 시스템을 독립된 서버로 분리하는 경우가 대부분이다. 그러나 이 또한 두 서버간의 네트워크 트래픽 문제가 발생할 수도 있다.

◉ **웹 서버개수와 서버개수는?** | 또한 웹 서버 몇 개에 WAS 서버 몇 개를 붙일 것인가도 역시 고려 대상이다. 적정 비율이 아닌 경우 한쪽에 부하가 몰릴 위험이 있다.

◉ **WAS 와 데이타베이스 서버와의 분리** | WAS와 데이타베이스는 반드시 분리해야 한다. 같은 서버에서 두 서비스를 운용하는 경우에는 엄청난 성능 저하가 발생하게 된다.

◉ **애플리케이션별 분리** | 동일한 기능을 수행하는 WAS를 여러 개 놓는 것보다 때에 따라서는 운용되는 애플리케이션별로 따로 WAS 시스템을 구동하는 것이 성능을 높이는 방안이 될 수도 있다.

Oracle9*i* Application Server 성능 튜닝 방안

그러면, 이제 앞서 언급한 다양한 성능 요소들의 조정 방법을 오라클의 WAS 제품인 Oracle9*i* Application Server에서는 어떤 형태로 지원하고 있는지 살펴보자.

스레드 풀 조절 방법

Oracle9*i* Application Server 최신 버전 9.0.2.1부터는 J2EE 엔진에서 동시에 수행 가능한 스레드의 크기를 유동적으로 조절할 수 있는 기능을 제공한다. 최소/최대뿐만 아니라 운영 중에 사용 안 되는 스레드를 자동으로 줄여주는 기능까지 제공한다. Oracle9*i* Application Server 내의 J2EE 엔진인 Oracle Container for J2EE에서 스레드 풀 값을 조절하기 위해서는 Oracle Container for J2EE 구성 파일인 server.xml에 다음과 같은 요소를 추가하면 된다.

```
<global-thread-pool min="30" max="30" queue="60" keepAlive="-1"
debug="true" />
```

min, max 값은 스레드 풀 내의 스레드 개수의 최소값과 최대값을 나타낸다. 보통 min=max로 설정해 주고, queue=max ×2로 설정해 줄 것을 권장하고 있다. KeepAlive=-1은 스레드 수행시의 타임아웃을 억지시켜 준다는 의미이다. “Debug=true” 스레드 풀 관련 디버깅 메시지를 로그 파일에 쓰도록 설정한다는 의미이다.

DB 커넥션 풀 조절 방법

Oracle Container for J2EE에서 DB 커넥션 풀 조정 작업은 데이타소스 인터페이스를 통해 지원된다. 이는 사용자가 DB 커넥션을 획득할 수 있는 인터페이스로서 DB 커넥션을 관리하는 팩토리 역할을 수행하며, 사용자가 지정한 환경 정보(주로 XML)를 통해 해당 커넥션을 생성 관리하는 일을 담당한다. 그 결과로서 논리적인 DB 커넥션을 제공하게 되는데, 주로 JNDI에 특정 이름으로 등록한 후, 이 이름을 통해 사용자들이 JNDI Lookup을 통해 사용할 수 있다. DB 커넥션 풀 값의 조절은 data-sources.xml 설정 파일에서 [리스트 1]과 같이 기술한다.
다음은 [리스트 1]의 속성들에 대한 설명이다.

- Pool Name : JNDI 통해 입출력 이름으로 사용되는 풀의 이름
- DataSource Emulator class : DataSource 기능을 수행하는 클래스 이름
- Connection Driver : JDBC Driver(Oracle, Sybase, MS SQL, ...)
- User Name, Password : DB 사용자 이름과 패스워드
- JDBC Connection URL : “jdbc:oracle:thin:@myserver:1521:ora92”
- Min Connection : 처음 구동시 미리 연결되는 연결의 수
- Max Connection : 최대로 연결 가능한 연결의 수
- Inactivity Timeout : 주어진 시간 동안 사용하지 않는 연결을 강제 종료

Oracle Container for J2EE에서 커넥션 풀의 튜닝시에 꼭 주의를 기울여야 할 부분이 바로 max-connections의 값을 스레드 풀 내의 스레드 개수 값보다 크거나 같도록 설정해야 한다는 점이다.

커넥션 풀 사용시에 현실적으로 가장 문제가 되는 경우는 프로그램상에서 사용자가 커넥션을 사용한 후에 커넥션을 종료하지 않는 경우이다. 이 때, Max 값이 지정된 경우에는 더 이상 사용할 풀이 없으면 모든 서비스가 중단된다('stack trace' 하면 모든 스레드가 getConnection()에 멈춰 있음). 만약, 무한대로 설정한 경우에는 DB 커넥션이 계속 증가하다가 더 이상 DB에 연결될 수 없으면 에러가 발생하게 된다. 이의 해결 방법은 프로그램상에서 애플리케이션 코드를 규격화하는 방법 뿐이다. 즉, try{ } finally{}를 통해 무조건 사용 후 close()하도록 프레임워크를 설계해 주어야 한다. Oracle Container for J2EE의 경우에는 객체 참조(object reference) 없이 일정 시간(수십 초)동안 유지되는 커넥션은 자동으로 close() 해 주도록 되어 있다.

HTTP 연결 조절 방법

Oracle9*i* Application Server는 웹 서버로서 Apache 기반의 Oracle HTTP Server를 사용한다. Oracle HTTP Server는 mod_oc4j를 통해 Oracle Container for J2EE와 연결된다. 전체적인 HTTP 연결은 httpd.conf 파일 안에서 다음과 같이 설정함으로써 조절할 수 있다.

```
MaxClients 150
```

일반적으로는 HTTP 연결의 개수를 증가시킬수록 전체적인 시스템 성능이 좋아질 것으로 생각할 수도 있다. 하지만, 도리어 HTTP 연결 개수를 너무 크지 않도록 줄여줌으로써 성능 향상을 이룰 수 있다. [그림 9]는 HTTP 연결 개수의 증가에 따른 성능 변화를 보여 준다.

만약 Oracle Container for J2EE를 스탠드얼론으로 사용하는 경우에는 server.xml 설정 파일 안에 다음과 같은 요소를 추가함으로써 HTTP 연결을 조절할 수 있 다. value 속성 값은 최대로 연결될 수 있는 HTTP 연결 개수이다.

```
<max-http-connections max-connections-queue-timeout="120"
      socket-backlog="50"
      value="1000">
        http://etc.us.oracle.com/redirect/login.jsp
</max-http-connections>
```

네트워크 튜닝 방법

앞서 WAS 기반 시스템의 성능 향상을 위해서는 네트워크 자체에 병목이 없어야 한다고 언급했었다. Oracle9*i* Application Server를 운용할 때도 효율적인 운용을 위해 권장되는 네트워크 인자값 집합이 있다. 여러 UNIX 시스템 중 Solaris를 위한 주요 네트워크 파라미터 설정 권장값은 다음과 같다.

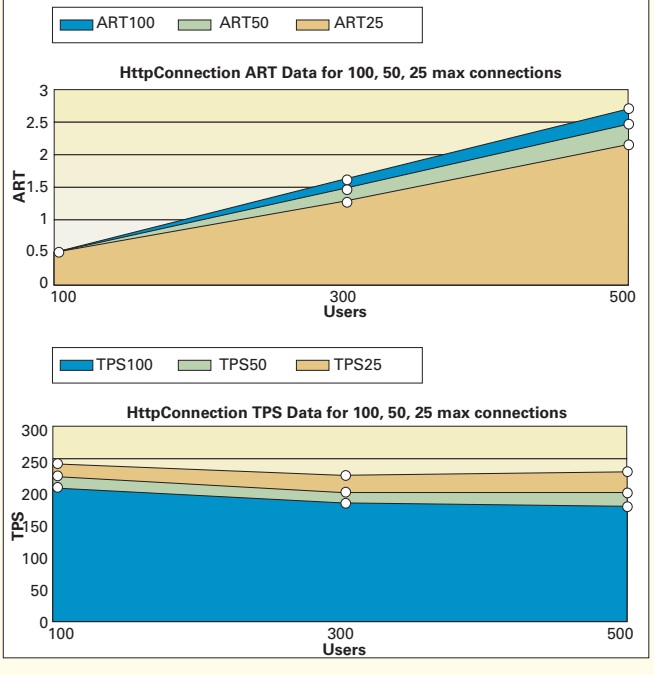
- tcp_conn_hash_size 32768
- tcp_conn_req_max_q 1024
- tcp_conn_req_max_q0 1024

[리스트 1] data-source.xml 설정 파일에서 DB 커넥션 풀 값의 조절

```
<data-source
   class="com.evermind.sql.DriverManagerDataSource"
   name="OracleDS"
   location="jdbc/OracleCoreDS"
   xa-location="jdbc/xa/OracleXADS"
   ejb-location="jdbc/OracleDS"
   connection-driver="oracle.jdbc.driver.OracleDriver"
   username="scott" password="tiger"
   url="jdbc:oracle:thin:@krservers1.kr.oracle.com:
   1521:oracle8"
   inactivity-timeout="300"
   max-connections="40"
   min-connections="10"

/>a
```

[그림 9] HTTP 연결 개수에 따른 WAS 성능 변화



- tcp_recv_hiwat 32768
- tcp_slow_start_initial 2
- tcp_close_wait_interval 60000
- tcp_time_wait_interval 60000
- tcp_xmit_hiwat 32768

위 값은 /etc/system 파일에 저장되어야 한다. 이 값은 시스템 리부팅 때 반영된다. 또 현재 설정값은 ndd 프로그램을 통해서 볼 수 있다

```
ndd /dev/tcp
```

를 수행하면 TCP 파라미터 값을 검색하거나 설정할 수 있다.

운영체제 튜닝 방법

대부분의 UNIX 시스템은 단일 프로세스가 동시에 열 수 있는 최대 FD(File Descriptor)의 수를 제한한다. 여기에는 다음과 같은 3가지 종류의 FD가 있다.

- Soft limit : 일반적인 애플리케이션에 적용되는 FD의 제한 값(rlim_fd_cur)
- Hard limit : 시스템 애플리케이션에 적용되는 FD의 제한 값(rlim_fd_max)
- System limit : 하드웨어 플랫폼 또는 운영체제 자체의 최대 값

이 값들 사이에는 다음과 같은 관계가 존재한다.

```
soft limit <= hard limit <= system limit
```

이러한 FD 값들은 ulimit을 통해서 확인 및 설정할 수 있다. 하지만, 이것은 셸에서만 일시적으로 변경 가능한 것이다. Ulimit은 셸에 따라 동작이 다를 수 있으며 시스템의 기본 설정 값 내에서만 변경할 수 있다.

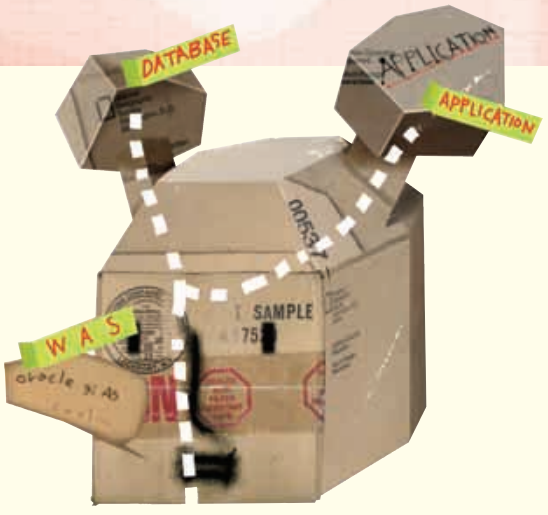
- $ ulimit -Sn 1024 : Soft 제한 값을 설정
- $ ulimit -Hn 2048 : Hard 제한 값을 설정
- $ ulimit -n 2048 : Soft/Hard 제한 값을 설정

영구적인 설정을 위해서는 /etc/system에 다음 명령 줄 추가 후 재부팅(Solaris)을 해야 한다.

- set rlim_fd_cur=1024
- set rlim_fd_max=2048

HP는 sam utility로 확인 후 설정한다.

UNIX 시스템에서는 대부분의 리소스들이 FD 값에 의존적인데, 동시에 연결 가능한 TCP 커넥션 값을 위해서는 충분히 큰 Hard Limit이



보장되어야 한다. Apache 역시 Soft Limit에 영향을 받는다. Apache 프로세스는 다중 프로세스로 운용되는데, 보통 기본 Soft Limit(64~512)으로도 충분하다. J2EE 엔진은 단일 프로세스(JVM)로 되어 있는데, 많은 양의 FD를 사용한다. 따라서, Soft Limit에 많은 영향을 받으며, 이 값의 확장이 필수적이다. 보통 512에서 많게는 4096 이상이 사용된다.

기타 튜닝 방안

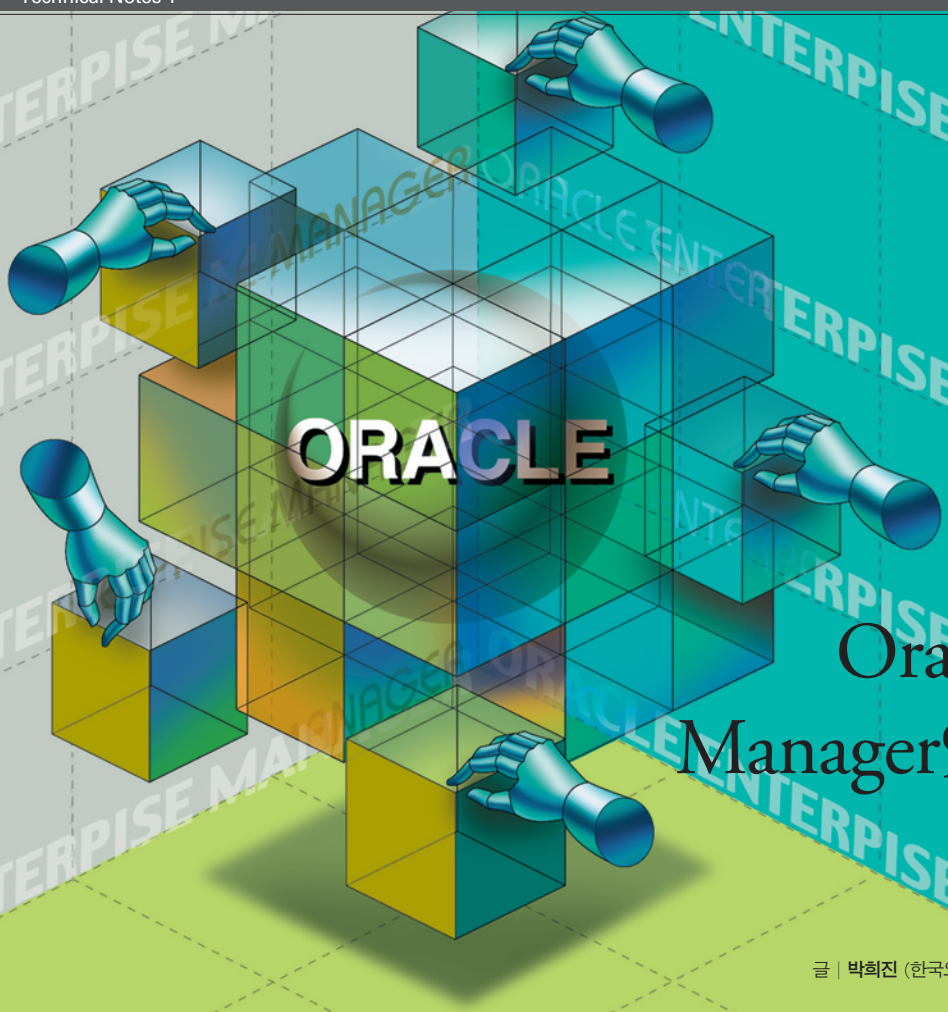
이 밖에도 Oracle9*i* Application Server상에서 수행 성능을 높이기 위한 다음과 같은 다양한 튜닝 방안들이 있다.

- JSP/Servlet 활용시의 튜닝 방안
- EJB 활용시의 튜닝 방안
- JVM 자체 튜닝 방안

지면 관계상 이러한 요소들에 대한 튜닝 방안은 생략하기로 한다. 위의 요소들에 대한 상세한 튜닝 방안은 'Oracle9*i* Application Server Performance Guide' 자료를 참조하기 바란다.

다양한 요소들간의 조정이 필수

지금까지 최근 기업 전산 시스템의 핵심 플랫폼으로서 그 중요성이 증대되고 있는 WAS 시스템의 개발과 운용에 있어서 반드시 고려해야 할 중요한 요소로서 WAS의 성능 측정 방안에 대한 기본적인 이론에 대해 설명하고, 오라클의 WAS 제품인 Oracle9*i* Application Server의 활용시에 적용 가능한 성능 튜닝 방안에 대하여 설명하였다. 계속 언급했다시피 WAS 시스템 튜닝은 내부 로직의 최적화를 통한 방법이 아니라 도리어 WAS를 구성하는 다양한 요소들 사이의 데이타의 흐름에 병목이 없도록 조정하는 작업을 통해서 이룰 수 있다. 독자들이 이러한 WAS 시스템의 성능 특성을 잘 이해함으로써 WAS 기반의 웹 시스템의 구축 및 활용 시에 효율적인 시스템 최적화를 이룰 수 있게 되기를 바란다.



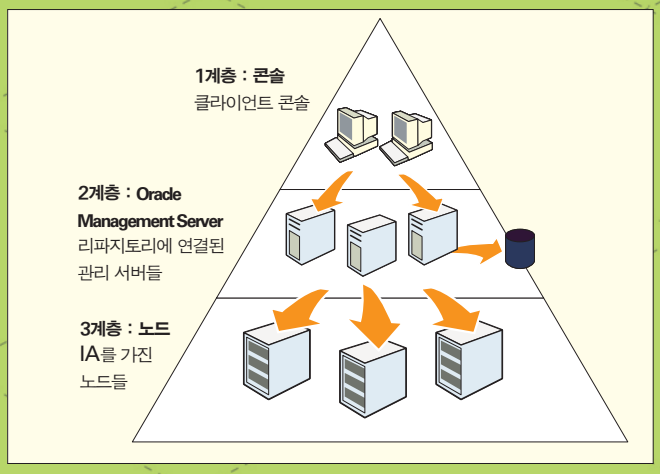
Oracle Enterprise Manager9*i* Release 2의 구성과 설정

글 | 박희진 (한국오라클 교육컨설팅팀) heejin.park@oracle.com

Oracle Enterprise Manager는 데이타베이스 리스너, 노드, Oracle9*i* Application Server 등을 관리, 모니터링하는 데이타베이스 관리자 전용 툴로서, 현재 버전 Oracle Enterprise Manager9*i* Release 2까지 나와 있다. 보다 많은 사람들이 좀 더 쉽게 오라클 관련 모든 작업을 Oracle Enterprise Manager를 통해서 관리, 운영하기를 바라는 의도에서 작성된 이 글에서는 Oracle Enterprise Manager의 기본 개념과 설정 사항들을 Oracle Enterprise Manager9*i* Release 2 버전을 중심으로 설명한다.

Oracle Enterprise Manager는 글로벌 기업 환경에서 다양한 리소스, 즉 데이타베이스, 리스너, 노드, Oracle9*i* Application Server를 관리, 모니터링하도록 통합된 환경을 제공하는 GUI 툴이다. Oracle Enterprise Manager는 다양한 데이타베이스 툴과 마법사, 데이타베이스 애플리케이션들과 특정 목적의 툴들을 묶어서 팩으로 제공하고 있다. 이 팩에는 진단 도구 팩(Diagnostics Pack), 튜닝 팩(Tuning Pack), 변경 관리 팩(Change Management Pack), 애플리케이션 관리 팩(Application Management Pack) 등이 있다.

Oracle Enterprise Manager는 1.x 버전에서 2.x 버전, 그리고 Oracle Enterprise Manager9*i*, Oracle Enterprise Manager9*i* Release 2로 발전하여 왔다. [그림 1]의 아키텍처 구성은 Oracle Enterprise Manager 2.x부터의 아키텍처 구성이며 Oracle Enterprise Manager9*i* Release 2도 이러한 구성을 따르고 있다.



[그림 1] Oracle Enterprise Manager 의 아키텍처

Oracle Enterprise Manager9*i* Release 2의 아키텍처

Oracle Enterprise Manager9*i* Release 2의 아키텍처 구성은 크게 3계층으로 이루어져 있다.

- 1계층 : 클라이언트 콘솔

그래픽 사용자 인터페이스를 제공하는 콘솔 클라이언트이다.

- 2계층 : 매니지먼트 서버, 리파지토리

Oracle Management Server는 Oracle Enterprise Management 전용 계정을 제공하며, JOB, EVENT 등의 관리 작업을 담당하는 Oracle Enterprise Management 의 핵심으로 리파지토리를 필요로 한다. 리파지토리는 관리 대상 노드에 대한 다양한 정보와 관리용 팩들(Management Packs)에 대한 정보를 저장하고 있는 별도의 테이블들의 집합을 의미한다.

- 3 계층 : 관리 대상 노드

관리 대상 데이타베이스나 다른 관리 대상을 가진 노드이다. 각 노드는 하나 이상의 데이타베이스로 구성될 수 있으며 반드시 하나의 노드에 하나의 IA(Intelligent Agent)가 있어야 한다. IA는 Oracle Managemant Server와 통신하며 콘솔이나 클라이언트 애플리케이션에서 요청한 작업을 수행하는 프로세스이다.

이와 같은 3계층 구조로 관리 대상을 관리하게 되면 여러 데이타베이스 관리자가 하나의 Oracle Management Server를 통해서 여러 노드에 흩어져 있는 다양한 리소스를 통합된 환경에서 관리할 수 있다. 또한 Oracle Management Server를 하나 이상 구성하여 중단이나 이상 없이 관리 대상을 관리할 수 있는 환경을 구현할 수 있다.

이러한 3계층 구조로 Oracle Enterprise Manager를 구성하여 데이타베이스를 관리할 수도 있고, 2계층 또는 클라이언트/서버 구성으로 Oracle Enterprise Manager를 구성하여 관리 대상 데이타베이스를 관리할 수 있다. 2계층 구성을 '스탠드얼론 모드' 또는 '클라이언트/서버 모드'라고 부른다. 이것은 이전 버전의 DBA Studio를 대체한 것으로 Oracle Enterprise Manager9*i*부터 등장했으며, DBA Studio보다는 좀 더 기능이 다양하다. 이 경우는 Oracle Management Server, 리파지토리, IA가 필요 없으며, 단일 데이타베이스 관리자가 단일 데이타베이스를 관리할 때 유용하다. 단 스탠드얼론 모드에서는 다음과 같은 일을 할 수 없다.

- 비-오라클 데이타베이스 관리
- 여러 관리자간의 관리 데이타 공유
- (Event 서비스를 이용한) 미리 정의된 문제 상황에 대한 알림 기능
- (Job 서비스를 이용한) 반복적인 관리 작업 수행
- 웹 브라우저를 통한 클라이언트 콘솔 수행

달리 말해, 위 작업을 수행하기 위해서는 반드시 Oracle Management Server가 있는 3계층 구조로 Oracle Enterprise Manager를 구성해서 관리 대상을 관리해야 한다.

Oracle Enterprise Manager9*i* Release 2의 설치

Oracle Enterprise Manager는 Oracle9*i* Database를 Universal Installer를 통해 설치하면서 함께 설치할 수 있다.

- 엔터프라이즈 에디션(Enterprise Edition) : 콘솔, Oracle Management Server, IA, 오라클 유틸리티들, 모든 Oracle Enterprise Manager 관련 팩들, 온라인 도큐먼트
- 스탠더드 에디션(Standard Edition) : 콘솔, Oracle Management Server, IA, 오라클 유틸 리티들
- 퍼스널 에디션(Personal Edition) : 엔터프라이즈 에디션과 동일하나 단일 사용자만 지원
- 커스텀(Custom) : 기본 공통 사항과 사용자가 선택한 구성요소

그러면, 스탠드얼론 모드와 3계층 아키텍처로 구분해 Oracle Enterprise Manager의 설정 방법을 알아보도록 하자.

Oracle Enterprise Manager9*i* Release 2의 스탠드얼론 모드 설정

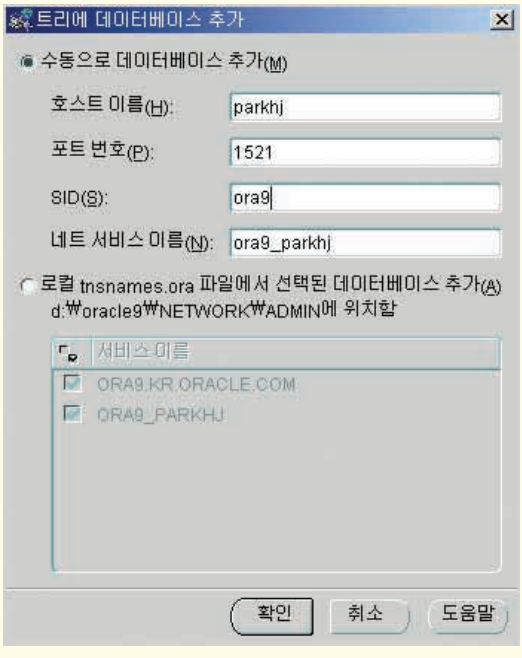
앞에서 설명했듯이, Oracle Enterprise Manager9*i* Release 2의 스탠드얼론 모드는 2계층 아키텍처로, Oracle Management Server 없이 콘솔을 이용하여 관리 대상 데이타베이스에 직접 접속하는 방식을 의미한다. 별도의 Oracle Management Server, 리파지토리, IA에 대한 설정이 필요 없으며, 반드시 하나의 데이타베이스에만 접속하여 관리할 수 있으며, 기능이 제한되어 있다.

이 모드로 데이타베이스에 접속할 때는 콘솔 화면(화면 1)에서 '독립적으로 실행'을 선택한 후 '확인'을 클릭한다. 관리 대상 데이타베이스에 대한 정보를 메뉴의 네비게이터의 '트리에 데이타베이스 추가'를 이용하여 입력한다(화면 2).

관리 대상 데이타베이스가 등록되면 더블클릭하여 접속에 필요한 사용자 정보를 입력하여 정상적으로 접속한다(화면 3). 데이타베이스에 관련된 일반적인 모든 업무를 이 모드를 통해서 수행할 수 있다(화면 4).

- 인스턴스(Instance):인스턴스에 대한 기동과 종료, 세션에 대한 강제 종료, 리소스 관리에 대한 설정 작업 등을 할 수 있다.
- 스키마(Schema): 스키마 생성 및 관리 작업을 할 수 있다.
- 보안(Security):사용자, 롤, 프로파일의 생성과 삭제 및 관리 작업을 할 수 있다.

(화면 1) Oracle Enterprise manager Consol 로그인



(화면 2) 트리에 데이터베이스 추가

- 저장 영역(Storage): 테이블스페이스, 데이타파일, 롤백 세그먼트 및 리두로그 그룹의 생성 및 관리 작업을 할 수 있다.
- 분산됨(Replication): 데이타베이스 링크 생성 및 데이타 복제 및 메시징 기술을 지원한다.
- 웨어하우스(Warehouse): 웨어하우스 관리를 위한 구체화된 뷰(Materialized View), 차원(Dimension)에 대한 생성 및 관리 작업을 할 수 있다.
- 작업 영역(Workspace): 하나 이상의 사용자가 데이타베이스의 데이타를 변경 관리하기 위해 공유하는 가상 작업 영역을 생성 및 관리할 수 있다.

일반적으로 스탠드얼론 모드는 리파지토리를 필요로 하지 않는다. 그러나, 스탠드얼론 모드를 통해서 다음과 같은 애플리케션 작업을 하기 위해서는 스탠드얼론 리파지토리가 필요하다.

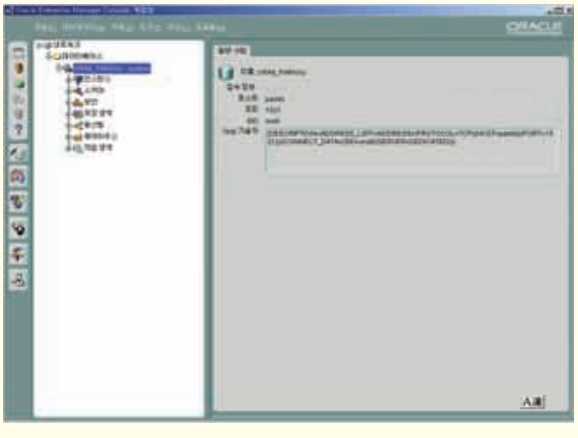
- Change Manager
- Oracle Expert
- Oracle SQL Analyze
- Oracle Index Tuning Wizard
- Oracle Tablespace Manager

스탠드얼론 리파지토리는 Oracle Management Server가 사용하는 리파지토리와는 다르며 공유해서 사용할 수도 없다. 약 64M 정도의 공간을 차지하며, 스탠드얼론 모드에서 위 애플리케이션을 처음으로 실행할 때 스탠드얼론 리파지토리 접속에 필요한 로그인 화면이 나타난다(화면 5).

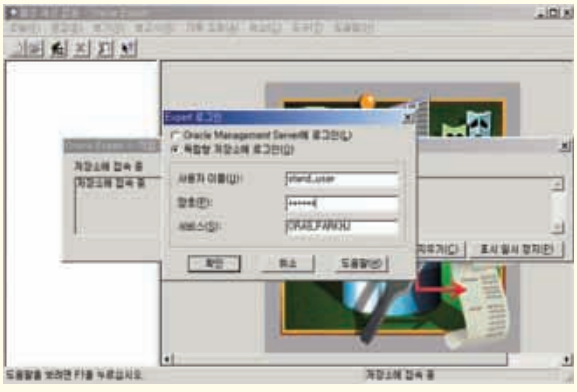
이때 중요한 것은 스탠드얼론 리파지토리용 사용자 계정은 미리 생성되어 있어야 한다는 것이다. 최초로 접속하면서 필요한 테이블이 접속 시점에 생성되며 사용자는 생성하지 않는다. 따라서 반드시 스탠드얼론 리파지토리의 소유 사용자를 다음과 같은 권한을 갖도록 생성하여 두어야 한다.

"create procedure, create trigger, create type, execute any procedure, execute any type, select any table, select any dictionary, connect, select_catalog_role"

(화면 3) 관리 대상 데이타베이스 접속 정보



(화면 4) 스탠드얼론 모드



〔화면 5〕 스탠드얼론 리파지토리 로그인

Oracle Enterprise Manager9*i* Release 2의 3계층

일반적으로 Oracle Enterprise Manager를 사용하는 구성으로 Oracle Management Server에 대한 설정과 리파지토리 생성에 대한 작업이 필요하다. Oracle Management Server는 리파지토리 생성 후 정상적인 연결이 되어야 Oracle Enterprise Manager를 사용할 수 있는 상태로 준비된다.

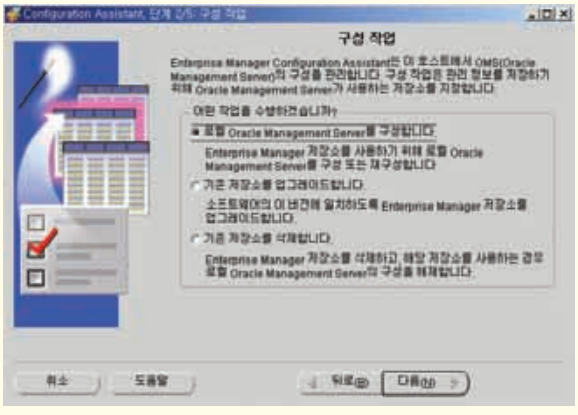
콘솔은 별도의 컴퓨터에 Windows 계열의 운영체제를 사용하는 어떠한 버전에서도 설치할 수 있다. Oracle Management Server만 별도로 설치하려는 경우는 Windows NT/2000이나 Solaris 2.6 이상에만 Oracle Management Server가 설치되므로 주의하여야 한다. 리파지토리용 데이타베이스는 관리 대상 데이타베이스 외에 별도로 준비되어 있어야 한다. 관리 대상 데이타베이스 내 별도의 스키마에 리파지토리를 생성하여 사용할 수 있지만, 이것은 관리 대상 데이타베이스에 이상이 생기면 리파지토리까지 이상이 생겨 Oracle Enterprise Manager를 사용할 수 없게 된다. 따라서 별도의 데이타베이스에 리파지토리를 구성하는 것이 좋다. 설치에 필요한 소프트웨어 하드웨어에 대한 권장사항은 〔표 1〕을 참조하기 바란다.

이 글에서는 Oracle Management Server와 리파지토리가 Windows 2000에 설치되어 있는 것으로 가정하고 설정에 대한 설명을 진행한다. 콘솔 또한 Oracle Management Server, 리파지토리와 함께 설치된 구조로 가정한다. 이미 앞에서 설명했듯이 Oracle Database9*i* Release 2를 엔터프라이즈 에디션으로 설치하면 Oracle Enterprise Manager에 대한 모든 것이 설치된다. 이때 Oracle Management Server에 대한 설정과 리파지토리에 대한 생성 작업을 완료해야 Oracle Enterprise Manager를 3계층 구조로 사용할 수 있다. 엔터프라이즈 에디션을 Windows 계열 운영체제에 설치한 후 바로 제어판의 관리 도구의 '서비스'를 살펴보면, Oracle Management Server에 대한 서비스(*OracleOracleHome*-Management Server)가 생성되어 있지만, 완전한 구성으로 생성된 것이 아니고 서비스의 이름만 서비스에 포함되어 있는 것이므로 이를 기동한다고 Oracle Enterprise Manager를 사용할 수 있는 것은 아니다.

리파지토리를 생성하면서 Oracle Management Server 접속에 대한 정보가 수정되고 Oracle Management Server에 대한 제어판의 서비스가 자동으로 기동된다.

STEP1. Oracle Management Server에 대한 구성 작업 선택

Windows의 시작 → 프로그램 → Oracle-ORACLEHOME → Configuration and Migration Tools → Enterprise Manager Configuration Assistant를 선택한다. 'Welcome' 화면에서 '다음'을 클릭한 후 '구성 작업' 화면에서 '로컬 Oracle Management Server를 구성합니다'를 선택한다. 〔화면 6〕에서 이미 생성되어 있는 리파지토리의 삭제나 리파지토리에 대한 업그레이드 작업 등도 수행한다.



〔화면 6〕 **STEP 1.** Oracle Management Server에 대한 구성 작업 선택

STEP 2. 리파지토리 생성 선택

〔화면 7〕에서는 리파지토리가 미리 생성되어 있지 않다면 '새 저장소 생성'을 선택한 후 '다음'을 클릭한다.

〔화면 7〕 **STEP 2.** 리파지토리 생성 선택

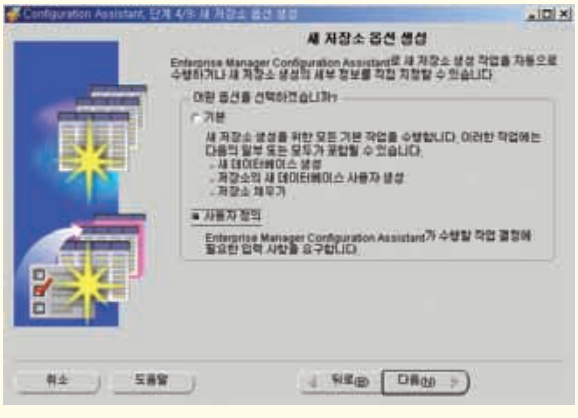
STEP 3. 새 리파지토리 옵션 생성

〔화면 8〕에서는 새로운 리파지토리용 데이타베이스가 없다면, 새롭게 데이타베이스까지 생성하면서 리파지토리를 생성하려면 '기본'을 선택하고, 이미 리파지토리용 데이타베이스가 있다면 '사용자 정의'를 선택

Component	Disk Space	Minimal Process/RAM	Recommended Processor/RAM
Console	510MB	Pentium 166/64MB SPARC 20/128MB	Pentium I 266/128MB SPARC Ultra 1/128MB
Oracle Diagnostics Pack	515MB	Pentium 166/64MB SPARC 20/128MB	Pentium I 266/128MB SPARC Ultra 1/128MB
Oracle Tuning Pack	511MB	Pentium 166/64MB SPARC 20/128MB	Pentium I 266/128MB SPARC Ultra 1/128MB
Oracle Change Management Pack	625MB	Pentium 166/64MB SPARC 20/128MB	Pentium I 266/128MB SPARC Ultra 1/128MB
Oracle Management Pack for Oracle Applications	511MB	Pentium 166/64MB SPARC 20/128MB	Pentium I 266/128MB SPARC Ultra 1/128MB
Oracle Standard Management Pack	625MB	Pentium 166/64MB SPARC 20/128MB	Pentium I 266/128MB SPARC Ultra 1/128MB
Management Server	730MB	Pentium 266/128MB SPARC Ultra 1/128MB	Pentium III 300/256MB SPARC Ultra 1/256MB
Enterprise Manager Web Site	820MB	depends on web server	depends on web server

〔표 1〕 **Oracle Enterprise Manager9*i* Release 2** 의 3계층 설치를 위한 소프트웨어/하드웨어 권장사항

한다. 여기서는 이미 리파지토리가 생성될 데이타베이스가 있으므로 '사용자 정의' 를 선택한다.



〔화면 8〕 **STEP 3.** 새 리파지토리 옵션 생성

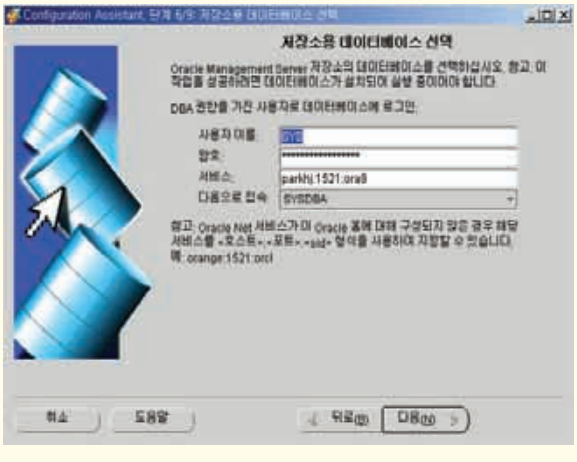
STEP 4. 데이타베이스 위치 선택

기존의 데이타베이스를 이용하는 경우는 '다른 기존 데이타베이스' 를 선택한 후 '다음' 을 클릭한다[화면9].

SETP 5. 리파지토리용 데이타베이스 선택

이미 리파지토리가 생성될 데이타베이스가 있으므로 접속에 필요한 정보를 입력한다. DBA 롤을 가진 계정으로 접속해야 한다. 일반적으로 SYS나 SYSTEM 계정으로 접속하면 된다. 이때 접속할 데이타베이스에 대한 오라클 SID와 리스너 포트에 대한 정보를 서비스 칸에 데이타베이스가 생성된 노드명 : 리스너 포트번호 : 오라클 SID 순으로 입력하면 된다[화면 10].

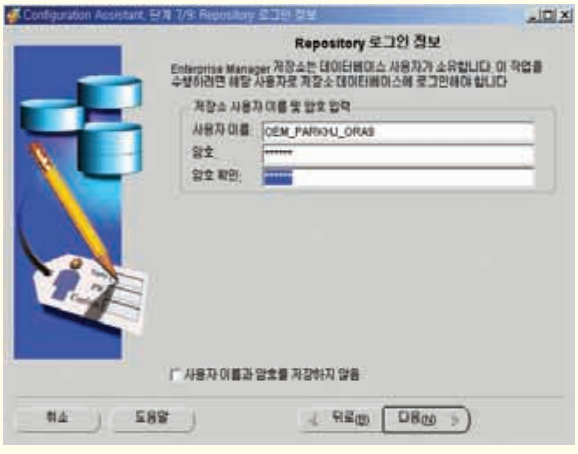
〔화면 9〕 **STEP 4.** 데이타베이스 위치 선택



〔화면 10〕 **STEP 5.** 리파지토리용 데이타베이스 선택

STEP 6. 리파지토리 로그인 정보

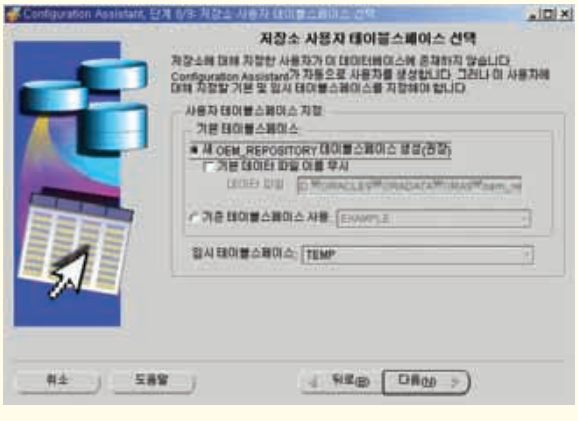
리파지토리란 결국 특정 사용자가 가진 Oracle Enterprise Manager에서 필요한 정보를 저장한 테이블의 집합이다. 따라서 여기서는 리파지토리의 소유자가 될 데이타베이스 계정을 하나 생성하도록 계정명을 지정하는 것이다. 계정명의 기본값은 OEM_노드명_SID 형태로 만들어지는데, 사용자 계정 생성 규칙에 맞게 계정명을 지정한 후 '다음'을 클릭한다[화면 11].



[화면 11] **STEP 6.** 리파지토리 로그인 정보

STEP 7. 리파지토리용 사용자에 대한 정보 선택

오라클에서 사용자를 생성하기 위해서는 기본 테이블스페이스, 임시 테이블스페이스에 대한 지정을 해야 한다. [화면 12]에서는 리파지토리 계정이 사용할 테이블스페이스에 대한 정보를 선택하는데, 이미 존재하는 테이블스페이스를 이용하거나 새로운 Oracle Enterprise Manager 리파지토리용 테이블스페이스를 생성해도 된다. 별도의 Oracle Enterprise Manager 전용의 테이블스페이스를 생성할 것을 권장한다.



[화면 12] **STEP 7.** 리파지토리용 사용자에 대한 정보 선택

STEP 8. 리파지토리 생성 요약 정보

리파지토리 생성에 대한 요약 정보를 확인한 후 '완료'를 클릭하면 이제 리파지토리가 지정된 옵션대로 생성된다. 생성이 정상적으로 완료되면 Oracle Management Server에 대한 서비스까지 자동으로 기동된다 [화면 13]. 3계층 구성에 대한 작업이 완료되었으면, Oracle Enterprise Manager 콘솔을 실행한다[화면 14].

[화면 13] **STEP 8.** 리파지토리 생성 요약정보

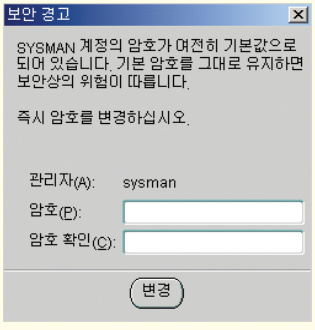
[화면 14] Oracle Enterprise Manager 콘솔

콘솔 화면에서 특정 Oracle Management Server를 선택할 때는 [화면 14]의 Management Server의 선택 버튼을 클릭하면 [화면 15]와 같이 Oracle Management Server 선택 화면이 나타난다. 이때 접속을 원하는 Oracle Management Server를 추가, 삭제하여 선택한 후 접속을 한다. 이때 사용하는 계정은 Oracle Enterprise Manager의 관리 계정으로, 기본적으로 사용자명은 SYSMAN, 암호는 OEM_TEMP이다. 첫번째 로그인 시 OEM_TEMP라는 암호는 변경해야 한다. 그렇지 않으면 매번 접속 시 암호 변경에 대한 팝업이 나타난다 [화면 16]. Oracle Enterprise Manager 관리 계정은 SYSMAN 외에 추가로 생성할 수 있다.

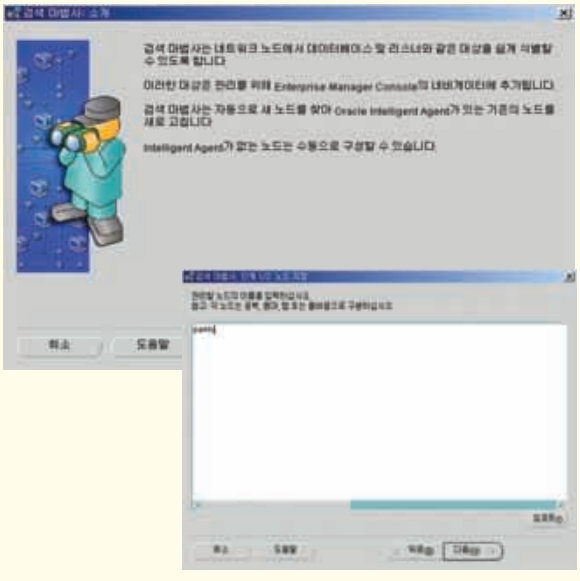
Oracle Enterprise Manager가 정상적으로 실행되면 관리 대상 노드 검색 화면이 나타난다. [화면 17]은 검색 마법사이다.

관리 대상 노드를 자동으로 검색하기 위해서는 관리 대상 노드에 IA라는 프로세스가 기동되어 있어야 한다. IA는 관리 대상 노드에 관리 대상

〔화면 15〕 Oracle Management Server 선택 화면



〔화면 16〕 SYSMAN 암호 변경 화면



〔화면 17〕 검색 마법사(메뉴의 객체→노드 검색 클릭:노드명 또는 노드 IP입력)

데이타베이스를 설치할 때 함께 설치되므로 별도의 설치를 필요로 하지 않는다. 단 관리 대상 노드의 하나 이상의 데이타베이스에 대한 모니터링과 콘솔을 통해 등록된 여러 애플리케이션의 작업을 수행하기 위해서는 관리 대상 노드의 IA가 관리 대상 데이타베이스를 찾아낼 수 있도록 약간의 설정 작업이 필요하다.

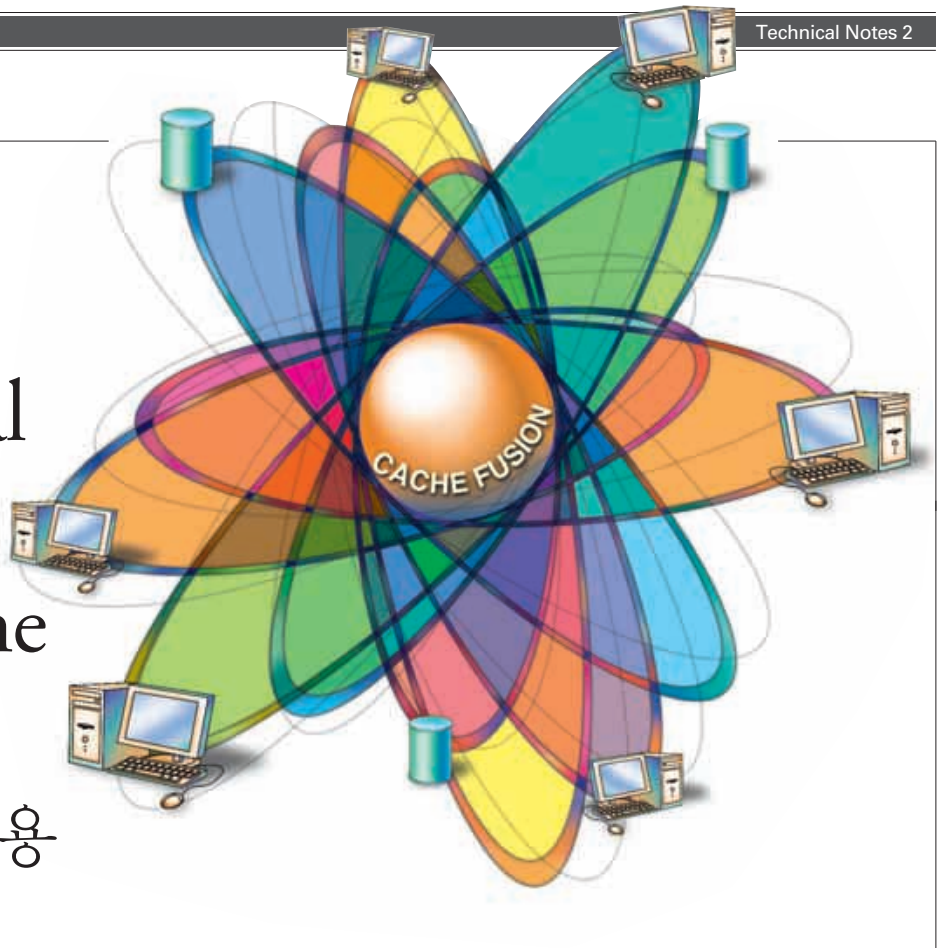
관리 대상 데이타베이스에 대한 검색 작업까지 완료되면 이제 Oracle Enterprise Manager를 통해 데이타베이스를 관리, 모니터링할 수 있는 모든 준비 과정에 대한 설정이 완료된 것이다[화면 18].

〔화면 18〕 관리 및 모니터링 준비가 완료된 Oracle Enterprise Manager 콘솔

Oracle Enterprise Manager 사용을 위한 기초 토대

스탠드얼론 모드나 3계층 구성으로 Oracle Enterprise Manager를 사용할 때 이 정도의 기본 설정만 알고 있어도 Oracle Enterprise Manager를 처음으로 사용하는 사람은 쉽게 Oracle Enterprise Manager에 접근할 수 있다. 그러나 실제로 Oracle Enterprise Manager 설정이 끝난 지금부터가 Oracle Enterprise Manager를 이용하여 데이타베이스를 관리하기 위한 이용의 시작이다. Oracle Enterprise Manager를 이용하려면 콘솔에서 추가적인 설정 작업도 필요하고 Oracle Enterprise Manager의 핵심 기능인 JOB, EVENT 서비스를 이용하여 관리 대상 데이타베이스의 관리를 자동화시킬 수 있는 기능에 대한 설정도 필요하다. 또한 Oracle Enterprise Manager의 다양한 팩들을 이용한 다양한 기능에 대한 사용법을 익힐 필요도 있다. 그러나, 지면의 한계상 Oracle Enterprise Manager사용의 가장 기본인 설정 작업에 대한 설명으로 이 글을 마무리하고 나머지 다양한 Oracle Enterprise Manager 기능에 대한 설명과 설정은 Oracle University와 여러분의 몫으로 남기려 한다.

마지막으로 팁 하나만 소개하면서 끝맺으려 한다. Oracle Enterprise Manager의 관리 계정인 SYSMAN의 암호는 첫 번째 로그인 시 반드시 변경해서 사용해야 한다고 이미 설명했다. 이때 SYSMAN의 암호를 변경 후 해당 암호를 잊어버리는 경우 다시 SYSMAN의 암호를 초기 설정 상태인 OEM_TEMP로 되돌리기 위한 팁이다. Oracle Enterprise Manager 2.x 는 오라클이 설치된 오라클 홈디렉토리 아래 sysman 폴더 밑에 admin 폴더 아래에 있는(%OracleHome%\sysman\admin) vduResetSysman.sql 파일을 리파지토리 계정으로 접속하여 수행하면 OEM_TEMP로 다시 재설정된다. Oracle Enterprise Manager9*i*는 이 방법 외에 리파지토리 계정으로 접속하여 smp_maintenace.reset_sysman() 프로시저를 실행해도 동일한 기능을 얻을 수 있다.



Oracle9*i* Real Application Cluster-Cache Fusion의 작동원리와 활용

글 | **이상범** (한국오라클 Advanced Support Team) sangbum.lee@oracle.com

Oracle9*i* Real Application Cluster는 뛰어난 HA 솔루션이자 확장성 솔루션이다. 이제까지 싱글 인스턴스에서 운영하던 애플리케이션은 간단하게 Real Application Cluster 환경으로 전환할 수 있으며, Oracle E-Business Suite나 SAP와 같은 패키지 애플리케이션들도 Real Application Cluster로 구성할 수 있다.이는 바로 Oracle9*i* Real Application Cluster에서 더욱 향상된 캐시 퓨전이 작동하기 때문이다. 이 글에서는 바로 이 캐시 퓨전이 어떻게 작동하는지, 그리고 이를 잘 활용하기 위한 준비사항에는 어떤 것들이 있는지 알아본다.

2001년 여름, 베를린에서 열린 Oracle9*i* 론치 행사장, 오라클의 CEO Larry Ellison은 특유의 쇼맨십을 연출하면서 청중들에게 질문을 던진다. "SAP를 IBM DB2 클러스터, 또는 MS SQL 서버 클러스터에서 실행할 수 있습니까? Oracle9*i* Real Application Cluster에서는 실행할 수 있습니다!" 한 시간 남짓 이어진 Larry의 기조연설을 인터넷으로 들으면서, 필자는 '도대체 Oracle9*i*를 론치하는 거야 아니면 Real Application Cluster를 론치하는 거야?' 라는 생각까지 들 정도였다.

이제까지 Oracle Parallel Server는 HA(High Availability) 솔루션으로만 알려져 왔는데, 그 이유는 노드간 버퍼 캐시의 동기화(synchronization)가 디스크를 통하여 이루어지기 때문이다. Oracle8*i* Oracle Parallel Server에서 캐시 퓨전(Cache Fusion)이 최초로 도입되어 이 문제가 일부 해결되었지만, 쓰기/쓰기 경합의 경우에는 여전히 디스크를 통한 동기화가 이루어지므로 성능 및 확장성에서 문제가 남아 있었다.

하지만, Oracle9*i* Real Application Cluster에서는 이 문제를 완전히 해소하면서 버퍼 캐시의 동기화는 100% 디스크를 거치지 않고 이루어진다. 따라서 기존에 Oracle Parallel Server를 도입할 때 항상 디자인 단계에서부터 철저하게 고려해야만 했던 노드간의 애플리케이션 분할은 이제 필요가 없게 되었고, Oracle9*i* Real Application Cluster는 HA 솔루션으로서 뿐만 아니라 확장성 솔루션으로서도 거듭나게 되었다. 다시 말하자면, 이제까지 싱글 인스턴스에서 운영하던 애플리케이션은 간단하게 Real Application Cluster 환경으로 전환할 수 있다. 따라서 Oracle E-Business Suite나 SAP와 같은 패키지 애플리케이션들도 Real Application Cluster로 구성할 수 있게 되었다.

이는 바로 Oracle9*i* Real Application Cluster에서 더욱 향상된 캐시 퓨전이 작동하기 때문이다. 이 글에서는 바로 이 캐시 퓨전이 어떻게 작동하는지, 그리고 이를 잘 활용하기 위한 준비사항에는 어떤 것들이 있는지 알아본다.

Gobal Enqueue Service & Global Cache Service

싱글 인스턴스 환경과 비교하여 Real Application Cluster에서는 노드간의 동기화를 위하여 부가적인 구성요소가 필요하다. 참고로 이는 Oracle Parallel Server에서는 DLM이라고 말하던 부분이며, Real Application Cluster에서는 캐시 퓨전 기능이 추가되어 GES(Global Enqueue Service)와 GCS(Global Cache Service)의 두 가지 서비스로 나누어진다.

Global Enqueue Service

GES는 사용자가 어느 노드에 접속되었는지에 상관 없이 트랜잭션에 대한 록(lock)을 조정하는 서비스로서, 예를 들면 1번 노드에서 특정 테이블의 특정 레코드에 대하여 DML을 수행하고 커밋 또는 롤백을 하지 않았다면 싱글 인스턴스 환경에서와 마찬가지로 2번 노드에서도 해당 레코드에 대하여 DML을 수행할 수 없다. GES는 오라클 백그라운드 프로세스 중에 LMD가 수행한다.

Global Cache Service

GCS는 캐시 퓨전과 함께 작동하는 서비스로서, 예를 들면 자신의 노드에 있는 캐시에서 원하는 블록을 찾지 못했을 때 다른 노드의 캐시를 확인하여 이미 다른 노드의 캐시에 원하는 블록이 올라와 있다면 디스크에 접근할 필요 없이 인터커넥트를 통하여 블록을 전달해 준다(인터커넥트에 대해서는 뒤에서 자세히 설명하기로 한다.) 이때 데이타의 일관성을 위하여 캐시에 올라온 블록에 대해서 GCS는 리소스를 할당하여 상태(모드)를 지속적으로 관리한다. GCS 리소스의 상태는 다음과 같이 3가지가 있다.

- Null(N) 모드 : 현재 해당 블록에 대하여 쓰기 또는 읽기를 수행중인 세션이 없는 상태이다.
- Share(S) 모드 : 현재 해당 블록을 읽는 세션이 적어도 하나 있는 상태이다.
- Exclusive(X) 모드 : 현재 해당 블록에 대하여 쓰기를 수행중인 세션이 적어도 하나 있는 상태이다.

이 중 S 모드는 특정 시점에 서로 다른 두 노드에서 공존할 수 있으나, X 모드는 특정 시점에 반드시 하나의 노드에서만 존재할 수 있다. 즉, 한 노드에서 특정 블록에 대하여 쓰기를 하고 싶다면 반드시 X 모드로 GCS 리소스를 할당 받아야 하며, 반대로 그 이전에 X 모드로 잡고 있던 노드에서는 N 모드로 다운그레이드가 된다.

캐시에 올라온 각각의 블록에 대하여 GCS 리소스는 하나의 노드가 마스터가 되어 관리하며, 어느 노드가 어느 블록을 관리할 것인가는 인스턴스를 시작할 때 오라클 내부적으로 Hashing 알고리즘에 따라서 결정되므로 동적으로 할당된다.

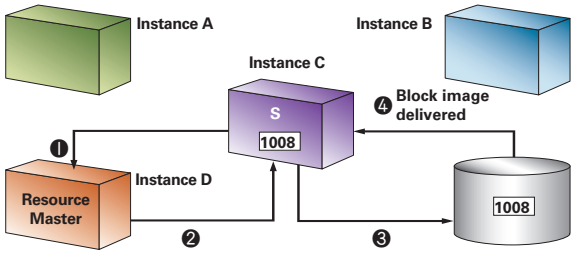
GCS는 오라클 백그라운드 프로세스 중에 LMS가 수행한다.

〔예제 1〕 Read with No Transfer

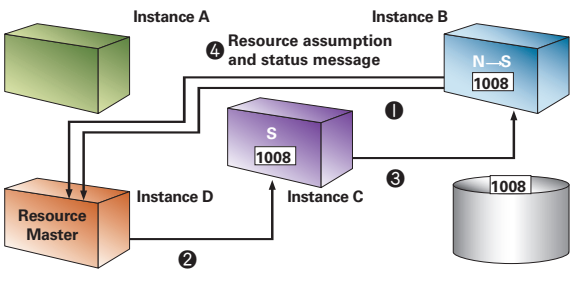
그러면, 지금부터 예제를 통해 캐시 퓨전이 어떻게 이루어지는지 살펴보도록 하겠다. 〔그림 1〕은 4노드 Real Application Cluster 구성을 간단하게 나타낸 그림으로, 이해를 돕기 위해 최초에 디스크에 있던 하나의 블록이 캐시 퓨전을 통해 어떻게 움직이는지 따라가보도록 하겠다. 최초에 디스크에 있던 블록의 버전을 나타내는 SCN(System Change Number)은 1008번이다.

1. 인스턴스 C에서 블록을 읽으려고 한다. 이때 자신의 캐시에 없으므로 이 블록을 마스터하고 있는 인스턴스에게 해당 블록의 상태를 확인한다. 여기서 우리가 살펴볼 블록의 마스터는 인스턴스 D라고 가정한다.
2. 마스터 노드인 인스턴스 D는 현재 어느 노드의 캐시에도 해당 블록이 올라와 있지 않으므로, 그 사실을 알려주면서 요구하는 인스턴스 C에게 GCS 리소스를 S 모드로 할당한다.
3. 인스턴스 C는 S 모드로 GCS 리소스를 할당 받은 후 디스크에 접근한다.
4. 이제 자신의 캐시로 해당 블록이 올라와 있으며, 읽기만 했기 때문에 SCN은 계속 1008번이 된다.

〔그림 1〕 예제 1 : Read with No Transfer



〔그림 2〕 예제 2 : Read to Read Transfer



〔예제 2〕 Read to Read Transfer

상황은 〔예제 1〕에서 계속 이어지며, 이제 또 다른 인스턴스 B가 같은 블록을 읽으려고 한다.

1. 〔예제 1〕과 마찬가지로 일단 자신의 캐시에 원하는 블록이 없으므로 마스터 노드에게 가서 해당 블록의 상태를 확인한다.
2. 마스터 노드인 인스턴스 D는 이미 인스턴스 C의 캐시에 해당 블록이 올라와 있기 때문에 인스턴스 C에게 현재 해당 블록을 요구하는 인스턴스 B로 블록을 보내주도록 지시한다.
3. 인스턴스 C는 인스턴스 B에게 블록을 보내주며, 이때 바로 읽기/읽기 캐시 퓨전이 일어나게 된다.
4. 인스턴스 B는 이제 원하는 블록을 캐시에 받고, GCS 리소스 모드 또한 읽기를 위해 S 모드로 할당 받는다.

이때 주목해야 할 점은, 이전 Oracle Parallel Server에서는 이와 같은 경우에 인스턴스 B 는 마스터 노드로 확인하는 과정 없이 바로 〔예제 1〕과 같이 디스크로부터 블록을 찾았다는 점이다. 핑(Ping)은 일어나지 않지만, 디스크를 접근하므로 성능은 Real Application Cluster에 비해 상대적으로 떨어진다.

〔예제 3〕 Read to Write Transfer

〔예제 2〕에서 계속 이어져, 이제 블록을 캐시로 전달 받은 인스턴스 B가 같은 블록에 대하여 업데이트를 하려 한다.

1. 인스턴스 B가 블록을 업데이트하려면 일단 GCS 리소스를 X 모드로 할당 받아야 하므로 다시 마스터 노드인 인스턴스 D로 가서 리소스 모드 업그레이드를 요청한다.
2. 인스턴스 B, C 둘 다 이전에는 S 모드로 가지고 있었으므로, 인스턴스 D가 X 모드로 업그레이드하기 위해서는 인스턴스 C가 N 모드로 다운그레이드되어야 한다. 마스터 노드는 인스턴스 C에게 다운그레이드를 지시한다.
3. 인스턴스 C가 다운그레이드를 완료하면 이제 인스턴스 B에게 알려주어 S에서 X 모드로 업그레이드를 할 수 있다.
4. 인스턴스 B는 이제 원하는 X 모드로 GCS 리소스를 받았다는 사실을 마스터 노드에게 알려준다. 이때 인스턴스 C도 N 모드로 다운그레이드되었다는 사실까지 함께 알려 준다. 이것이 완료되면 비로소 인스턴스 B는 해당 블록을 업데이트할 수 있으며, 〔그림 3〕과 같이 SCN이 1009번으로 바뀐 것으로 나타난다.

디스크에는 여전히 SCN이 1008번으로 남아 있으며, 따라서 현재 가장 최근의 블록 이미지는 인스턴스 B의 캐시에 SCN 1009번으로 놓여 있다는 사실에 주목하자.

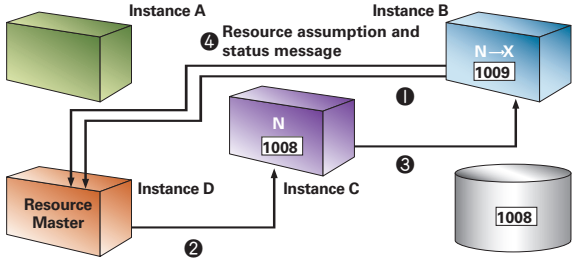
〔예제 4〕 Write to Write Transfer

〔예제 3〕에서 계속 이어져, 이제 인스턴스 A가 또다시 해당 블록에 대하여 업데이트를 하려고 한다.

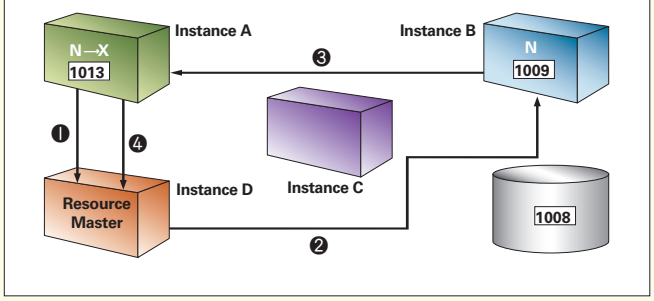
1. 인스턴스 A는 자신의 캐시에 해당 블록이 없으므로 마스터 노드에 정보를 요구한다.
2. 마스터 노드인 인스턴스 D는 인스턴스 B가 X 모드로 GCS를 잡고 있으므로 가장 최근의 이미지를 가지고 있다는 사실을 알고 있다. 따라서 인스턴스 B에게 요구하는 인스턴스 A로 블록을 보내주도록 지시한다.
3. 인스턴스 B는 우선 GCS 리소스를 N 모드로 다운그레이드한 후 요구하는 인스턴스 A로 블록을 보낸다. 이때 바로 쓰기/쓰기 캐시 퓨전이 일어난다.
4. 인스턴스 A는 GCS 리소스를 X 모드로 업그레이드한 후 마스터 노드로 그 정보를 보내 준다. 이제 비로소 인스턴스 A는 해당 블록에 업데이트할 수 있으며, 가장 최근의 블록 이미지는 인스턴스 A에 있다. 〔그림 4〕에서는 SCN이 1013번으로 바뀐 것을 볼 수 있다.

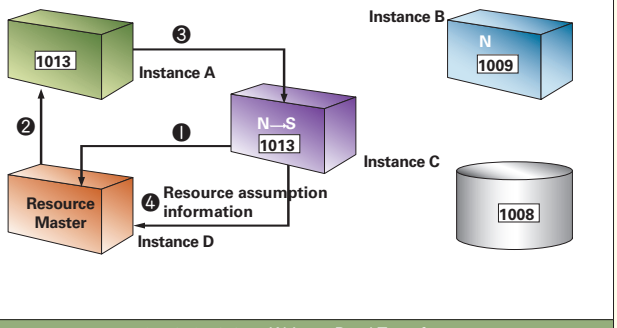
Oracle Parallel Server에서는 이와 같은 시나리오에서 인스턴스 B는

〔그림 3〕 예제 3 : Read to Write Transfer



〔그림 4〕 예제 4 : Write to Write Transfer





〔그림 5〕 예제 5 : Write to Read Transfer

일단 디스크에 블록을 쓰고, 인스턴스 A에서는 다시 디스크에서 읽는 과정을 거쳐야만 했으며, 이 디스크를 거쳐가는 동기화 과정을 '핑(Ping)'이라고 불렀다. Real Application Cluster에서는 쓰기/쓰기 캐시 퓨전이 가능해지면서 핑이 완전히 사라져, 성능이 눈에 띄게 향상되면서 확장이 용이하게 되었다. 즉 현재 싱글 인스턴스 환경은 언제든 Real Application Cluster 환경으로 전환할 수 있다.

〔예제 5〕 Write to Read Transfer

〔예제 4〕에서 계속 이어져, 이제 인스턴스 C에서 다시 해당 블록을 읽고자 한다. 〔예제 3〕에서 인스턴스 C는 해당 블록에 대하여 N 모드로 GCS 리소스를 다운그레이드했음을 기억하자.

1. 인스턴스 C는 해당 블록에 대하여 N 모드로 GCS 리소스를 가지고 있었으므로, 다시 읽기를 위하여 S 모드로 업그레이드하기 위해 마스터 노드에 정보를 확인한다.
2. 마스터 노드는 인스턴스 A가 가장 최근 이미지를 X 모드로 가지고 있다는 사실을 알고 있으므로 인스턴스 A에게 요구하는 인스턴스 C로 블록을 보내라고 지시한다.
3. 인스턴스 A는 X에서 S로 GCS 리소스를 다운그레이드하고 인스턴스 C로 블록을 보내 준다. 이때 바로 쓰기/읽기 캐시 퓨전이 일어난다. 참고로 이 기능은 Oracle8*i* Oracle Parallel Server에서부터 있던 기능이다.
4. 인스턴스 C는 N에서 S로 GCS 리소스를 업그레이드하고 다시 마스터 노드에게 이 사실을 알려 준다. 이제 SCN 1013번의 블록 이미지가 인스턴스 A와 인스턴스 C의 캐시에 동시에 놓여 있으며, 이제 인스턴스 C는 원하는 대로 블록을 읽을 수 있다.

이때 디스크에는 여전히 SCN 1008번이 놓여 있으며, 인스턴스 B에는 디스크에 있는 이미지보다는 새로운 이미지지만, 인스턴스 A 또는 C보다는 오래된 이미지인 SCN 1009번으로 캐시에 블록을 가지고 있다. 하지만 GCS 리소스가 N 모드이기 때문에 차후에 읽기 또는 쓰기를 할 때는 캐시에 해당 블록을 가지고 있더라도 반드시 마스터 노드에게 정보를 확인하게 되며, 유효하지 않은 SCN 1009번 이미지를 읽을 경우는 없다. 이때 시나리오는 〔예제 2〕(read) 또는 〔예제 3〕(write)와 같이 이루어지게 된다.

고속 인터커넥트

캐시 퓨전이 없었던 Oracle8*i* 이전 버전의 Oracle Parallel Server에서는 캐시에서 캐시로 블록을 전송하지 않았으므로, 노드간 동기화를 위한 정보는 GCS/GES 메시지(이전에는 DLM 메시지)밖에 없었다. 이 메시지의 양은 그렇게 사이즈가 크지 않았기 때문에(메시지당 약 200바이트), Oracle8*i* 이전 버전의 Oracle Parallel Server에서는 노드간의 통신을 위한 전용 네트워크를 특별히 별도로 마련하지 않아도 큰 문제 없이 운영할 수가 있었다.

하지만, 캐시 퓨전이 나오면서, 블록 자체를 네트워크를 통해 상대 노드의 캐시로 직접 전송하게 되면서, 노드간의 통신량은 다음 두 가지 변수에 크게 의존하게 된다.

- DB 블록 크기
- 노드간의 데이타에 대한 동시 접근, 즉 디스크상의 같은 블록을 서로 다른 노드에서 얼마나 동시에 접근할 수 있는가.

특히 최근에 시스템의 I/O 성능이 향상되고 테이블의 레코드 크기가 커지는 추세이므로, SQL 성능 향상을 위하여 과거에 비하여 상대적으로 큰 블록 크기(8k 또는 16k)를 선호하면서 결과적으로 캐시 퓨전에 의한 노드간 통신량은 기존의 공용 네트워크로는 감당하기가 힘든 상태가 되었다.

따라서 Real Application Cluster는 설치를 위한 필수 사항으로 노드간의 전용 네트워크를 확보하도록 하는데, 이것을 '고속 인터커넥트(High-Speed Interconnect)' 또는 '인터커넥트' 라고 부른다. 즉, Real Application Cluster를 설치하기 위해서는 기존에 클라이언트를 서비스하기 위한 공용 네트워크 이외에 별도로 전용 네트워크 인터페이스가 반드시 필요하다.

오라클은 대부분의 UNIX 플랫폼에서는 이 노드간 통신을 위한 인터커넥트의 프로토콜로 TCP보다 성능이 우수한 UDP(User Datagram Protocol)를 사용하지만, 특정 플랫폼에서는 플랫폼 벤더가 제공하는 특수한 프로토콜을 사용하기도 한다. 물론 이것들을 사용하기 위해서는 오라클의 인증이 필요하며, 현재까지 인증을 받은 프로토콜은 Tru64의 RDG와 HP-UX의 HMP가 있다.

자신이 설정한 인터커넥트가 올바르게 설정되었는지를 확인하기 위해서는 다음과 같이 oradebug를 사용할 수 있다.

```
SQL> oradebug setmypid
SQL> oradebug ipc
```

이렇게 하면, user_dump_dest 아래에 trace가 생기며, 아래와 같이 인터커넥트가 사용하는 IP를 확인할 수 있다.

```
SSKGXPT 0x2ab25bc flags info for network 0
socket no 10 IP 204.152.65.33 UDP 49197
sflags SSKGXPT_UP
```

보통 오라클은 특별한 설정 없이 전용 네트워크를 자동으로 발견하여 인터커넥트로 사용하지만, 만일 확인한 IP가 전용 네트워크로 올바르게 설정되지 않은 경우에는 cluster_interconnects 파라미터에 원하는 IP를 강제로 설정할 수 있다.

단, RDG 또는 HMP를 사용하는 경우에는 oradebug ipc trace 안에 인터커넥트 IP 정보가 나오지 않으며, 인스턴스 시동시 나오는 다음과 같은 메시지를 보고 확인할 수 있다.

```
'cluster interconnect IPC version:Oracle RDG'
'cluster interconnect IPC version:Oracle using HP-HMP'
```

인터커넥트 장애 발생시 어떤 일이 일어나는가

이렇게 확보해 놓은 노드간의 인터커넥트(전용 네트워크)는 캐시 퓨전에 의한 블록 이미지가 전달되는 통로로 쓰이며, 또한 앞서 설명한 GES/GCS가 수반하는 메시지가 지나가는 통로가 된다.

그렇다면, 만일 이 인터커넥트에 장애가 생긴다면 어떻게 될까? 물론 GES, GCS는 정상적으로 작동하지 못할 것이며, 따라서 오라클 인스턴스는 모든 것이 멈추는 '중단(hang)' 상태로 빠지게 된다. 이 상황을 우리는 'Split-Brain' 현상이라고 부르며, 사람의 머리에 해당하는 오라클 인스턴스가 마치 좌뇌와 우뇌 사이를 연결하는 신경이 끊어져서 식물인간이 된 상태에 비유할 수 있겠다.

이 Split-Brain현상은 다음과 같은 두 가지 시나리오에 의해서 해결될 수 있다. 우선은 2노드 Real Application Cluster 환경을 생각해 보자. 물론 Real Application Cluster 환경을 구성하기 위해서는 운영체제의 클러스터 매니저를 먼저 구성해야 함을 기억하자.

- 만일 오라클 인터커넥트가 운영체제 클러스터 매니저의 하트비트가 사용하는 네트워크를 공유하는 경우에는 오라클 입장에서 뿐만 아니라 운영체제 클러스터 매니저의 입장에서도 또한 Split-Brain 현상이 되므로 운영체제 클러스터 매니저의 하트비트 타임아웃에 걸려서 두 노드 중 하나를 강제로 죽이게 된다. 물론 어느 노드가 죽는가는 운영체제 클러스터 매니저의 결정에 따르므로 오라클 입장에서는 예측할 수가 없다.
- 만일 오라클 인터커넥트가 운영체제 클러스터 매니저의 하트비트가 사용하는 네트워크와 별도의 네트워크를 사용한다면, 이제는 Split-Brain현상은 오라클만의 문제가 된다. 운영체제 클러스터 하트비트는 올바르게 작동하고 있기 때문에 운영체제 입장에서는 오라클 인터커넥트에 장애가 발생한 상황에 대하여 책임을 지지 않는다.

이때는 오라클이 설정해 놓은 GES/GCS 타임아웃에 걸려 이 현상을 자체적으로 해결해야만 하는 상황이다. 즉, 오라클이 두 인스턴스 중 어느 인스턴스를 죽일 것인가를 결정하여 한 노드만을 살려두게 된다. 오라클 내부적으로 설정된 타임아웃 값은 5분이며, 플랫폼마다 약간씩 차이가 있을 수 있다. 이러한 경우에 죽는 인스턴스의 alert.log에는 ORA-29740 에러가 아래와 같이 기록된다.

```
Mon Nov  4 10:11:25 2002
Errors in file
/oracle/app/oracle/product/9.2/admin/RAC02/bdump/rac02_
lmon_4382770.trc
:
ORA-29740: evicted by member 0, group incarnation 3
```

또한 살아 남는 인스턴스의 alert.log에는 아래와 같이 기록된다.

```
Mon Nov  4 10:11:25 2002
Evicting instance 2 from cluster
```

이는 Real Application Cluster의 새로운 기능인 IMR(Instance Membership Recovery)이 작동하기 때문이다.

그런데, 여기서 한 가지 의문점이 생길 수 있다. IMR이 작동할 때 과연 어느 인스턴스가 살아남을지 예상할 수 있는가? 여기서는 2노드 Real Application Cluster을 예로 들었지만, 만일 3노드 Real Application Cluster 구성에서 2:1로 인터커넥트 장애가 일어났다면? 다시 말해서 인스턴스 A와 B 사이의 인터커넥트에는 문제가 없는데, 인스턴스 C와의 인터커넥트가 장애가 났다면? 이때 인스턴스 C가 살아 남고 인스턴스 A와 B를 둘 다 죽여버린다면 불공평한 상황이 된다. 우리는 Split-Brain 상황에서 언제나 가장 많은 수의 인스턴스를 가지고 있는 그룹이 살아 남기를 원하기 때문이다.

IMR은 이런 문제에서 우리가 원하는 대로 항상 가장 많은 수의 인스턴스가 속한 그룹을 살리기 때문에 언제나 공평한 결과가 된다. 즉 위와 같은 상황에서는 항상 인스턴스 C가 죽게 된다.

그렇다면 다시 떠오르는 의문점은? 만일 2노드 환경으로 1:1 Split-Brain 상황이라면? 이때는 서로 대등한 관계이기 때문에 IMR은 오라클 내부의 Heuristic 알고리즘에 따라서 임의로 한 인스턴스를 선택하게 되며, 따라서 우리는 어떤 인스턴스가 살아남을지 예상할 수 없다.

인터커넥트의 HA 구성

Real Application Cluster를 안정적으로 운영하기 위해서는 인터커넥트를 올바르게 설정하고 장애가 없도록 잘 관리하는 것이 무엇보다도 중요하다.

Real Application Cluster가 페일오버 기능으로 가용성을 보장하듯이, 최근에는 이 인터커넥트에 대해서도 플랫폼 벤더에서는 가용성을 위하여 HA(High Availability) 솔루션을 제공하고 있는데, 주로 인터커넥트를 이중으로 구성하여 액티브/스탠바이 형태로 운영하는 환경이다. 즉, 액티브 인터커넥트에 장애가 발생하면 스탠바이 인터커넥트가 같은 IP를 가지고 페일오버하므로 오라클 입장에서는 일단 페일오버가 완료되면 장애 이전 상황으로 완전히 돌아가게 된다.

Real Application Cluster를 구성할 때는 가능하면 이렇게 인터커넥트를 HA로 구성하기를 권장하며, 실제 어떻게 적용하는가는 각 플랫폼 벤더에게 문의하기 바란다.

인터커넥트의 통신량 확인

그렇다면 인터커넥트를 통해 흐르고 있는 통신량(traffic)이 얼마나 되는지 어떻게 확인할 수 있을까? 물론 운영체제에서 제공하는 유틸리티를 이용할 수 있겠지만, 오라클에서 제공하는 Statspack으로도 어느 정도까지는 계산할 수 있다(Statspack Report의 사용법에 대해서는 Oracle Megazine 2000년 11/12월호(http://www.oracle.com/oramag/oracle/00-nov/o60toc.html)를 참조하기 바란다.)

인터커넥트의 통신량은 크게 앞서 설명한 GES/GCS 메시지와 캐시 퓨전에 의한 블록 이미지의 두 가지로 이루어진다. 계산 공식은 아래와 같다.

```
Interconnect traffic = GES/GCS messags + cache fusion messages
 =( ('gcs messages sent' + 'ges messages sent') * 200 bytes
+ ('global cache cr blocks served' + 'global cache current blocks served') * DB_BLOCK_SIZE)   * 1.2
```

여기서 GES/GCS 메시지에 곱하는 200바이트는 메시지의 평균값이며, 마지막 1.2는 네트워크 패킷의 헤더를 감안한 오버헤드가 된다. 따라서 여기서 계산된 값은 우리가 어느 정도의 통신량이 있는지 추정치가 될 뿐이며, 정확한 값은 아니라는 점에 유의하기 바란다.

이 계산 공식에 사용되는 변수는 Statspack Report를 통해 볼 수 있다. Statspack Report 중에는 'Instance Activity Stats' 라는 부분이 있으며, 여기에는 리포트를 생성한 타임 인터벌 내에 이루어진 인스턴스의 활동에 대한 여러 가지 통계치가 기록되는 곳이다. [표 1]은 Statspack Report에서 발췌한 내용이다.

인터벌은 17:26에서 17:31까지 5분(300초)으로 두고 측정한 결과이며, 이 수치를 가지고 공식을 적용해 본다. db_block_size는 8K이다.

```
Interconnect traffic = GES/GCS messags + cache fusion messages =( (20048 + 987) * 200 bytes
+ (7918 + 8224) * 8192)   * 1.2
= 163730716.8 bytes
```

Snap Id	Snap Time	Sessions	Curs/Sess
Begin Snap:4	2002-11-29 17:26	65	82.5
End Snap:5	2002-11-29 17:31	65	99.4

Instance Activity Stats for DB : OCFS **Instance :** ocfs2 **Snaps:** 4-5

Statistic	Total	per Second	per Trans
gcs messages sent	20,048	67.5	2.5
ges messages sent	987	3.3	0.1
global cache cr blocks served	7,918	26.7	1
global cache current blocks serve	8,224	27.7	1

[표 1]

즉 300초 동안 약 156MB의 통신량이 있었으며, 이것을 초당으로 나누면 약 532KB가 된다. 물론 Statspack에 이미 계산된 값인 'per Second' 값을 사용해도 같은 값이 나올 것이다.

단, 여기서 계산된 값은 현재 상대방 인스턴스로 보낸 일방적인 통신량만을 구한 것이며, 만일 애플리케이션이 분할되지 않았다면 전체 통신량은 Real Application Cluster를 구성하는 노드 수만큼 곱해주면 될 것이다. 여기서는 2노드 Real Application Cluster이기 때문에 두 배를 곱해주면, 인터커넥트의 대역폭을 어느 정도로 확보해야 하는지 추정할 수 있다.

하지만, 최근에는 인터커넥트의 대역폭은 보통 100Mbyte/sec 이상, 특히 하이엔드급 서버에는 최소 기가비트 이더넷을 사용하는 것이 보통이므로 대역폭이 작아서 생기는 병목현상은 거의 없다고 보는 것이 옳다. 오히려 Real Application Cluster의 성능이 갑자기 떨어지는 경우에는 각 노드내의 상태를 점검하여 인스턴스 자체의 성능에 초점을 맞추어 분석하는 것이 올바른 접근방법이라 할 수 있겠다.

캐시 퓨전에 의한 놀라운 확장성

이제까지 설명한 대로 Real Application Cluster를 구성하기 위한 요구사항만 갖추어진다면, 캐시 퓨전으로 가능하게 된 놀라운 확장성을 경험할 수 있다. 이제까지 싱글 인스턴스 환경으로 운영하면서 언제나 가용성/확장성에 대하여 고민하던 독자들은 오라클이 야심차게 내놓은 확실한 솔루션인 Real Application Cluster를 가장 먼저 고려해보길 바란다.

Oracle9*i* 론칭 때 Larry Ellison의 말을 인용하며 글을 끝맺고자 한다.

"비즈니스를 시작하면서 DBMS를 도입할 때, 일단은 값싼 하드웨어에서 싱글 인스턴스로 운영하십시오. 이후 비즈니스가 성장하여 DBMS 시스템을 확장하고 싶을 때는, 비싼 하드웨어로 교체할 필요 없이, 그냥 똑같이 값싼 하드웨어를 하나 더 도입하여 기존의 싱글 인스턴스에 플러그인하여 Real Application Cluster를 사용하면 됩니다. 이때 비로소 애플리케이션의 분할이 필요 없는 Real Application Cluster의 확장성을 경험할 수 있습니다."

추가 Database Administrator Fundamentals 인증

글 | **Jim Dilanni** (Oracle Certification Program의 인증 책임자) James.Dilanni@oracle.com

이 칼럼은 Oracle9*i*: Oracle9*i* Database Administrator Oracle Certified Professinal 시리즈 내 Database Administrator Fundamentals I 시험에서 다루고 있는 Oracle9*i* Database 기술에 초점을 맞추고 있다. 이 칼럼은 시험에 출제될 것으로 예상되는 기술을 반영한 추가 샘플 질문에 대해서도 집중적으로 다룬다. 이번호에서는 리두 로그 파일 및 테이블 스페이스 관리에 대해 알아본다.

리두 로그 파일

Oracle9*i* Database Administrator가 수행해야 하는 가장 중요한 작업 2가지는 데이타베이스 가용성을 유지하는 것과 데이타베이스 장애시 데이타 손실을 예방하는 것이다. 리두 로그 파일은 중요한 Oracle9*i* Database 구조로서 Database Administrator가 성공적으로 이러한 작업들을 수행할 수 있도록 보장해 준다. Oracle9*i* Database 기능은 리두 로그 파일 및 이들에게 영향을 미치는 작동에 대한 관리의 용이성을 강화했다.

Q . 리두 로그 파일의 목적을 가장 적절하게 서술한 문장은?

a . 리두 로그 파일은 로그 스위치가 효율적으로 수행되는 것을 보장한다.

b . 리두 로그 파일은 비동기식으로 기록된 데이타베이스의 변경을 지원한다.

c. 리두 로그 파일은 데이타베이스 장애시 트랜잭션 리두 방법을 제공한다.

d. 리두 로그 파일은 아직 수행하지 않은 변경 내용을 기록한다.

A, B, D는 리두 로그 파일에 해당되는 내용이기는 하지만, 리두 로그 파일은 데이타베이스 장애 복구가 필요 없는 경우에는 요구되지 않는다. 그러므로 정답은 C이다.

Oracle9*i* Database는 리두 로그 파일에 기록된 트랜잭션을 자동 관리한다. 그러나, 더욱 중요한 것은 Oracle9*i* Database Administrator가 리두 로그 파일의 작동을 제대로 파악하여 Database Administrator가 데이타베이스 환경을 적절하게 구성함으로써 장애 발생시 데이타베이스를 완벽하게 복구할 수 있도록 지원한다는 점이다.

Q : Oracle Database 인스턴스에 요구되는 최소한의 리두 로그 파일 그룹의 수는?

a. 1개

b. 2개

c. 3개

d. 4개

정답은 B이다. 데이타베이스가 로그 파일에 데이타를 작성하기 위해서는 최소 2개의 로그 파일 그룹이 필요하다. 데이타베이스가 SGA의 리두 로그 버퍼에서 데이타를 가져와 리두 로그 파일에 기록할 경우, 동일한 리두 로그 그룹의 모든 파일에 데이타가 기록된다. 'Current' 파일 그룹 내에 리두 로그 파일이 꽉 차면 로그 스위치가 일어나서 다음 리두 로그 파일 그룹이 'Current' 그룹이 된다. 가장 좋은 방법은 서로 다른 디바이스에 위치해 있는 각 그룹의 멤버 파일을 포함하고 있는 리두 로그 파일 그룹을 2개 이상 보유함으로써 장애 발생시의 데이타 손실을 예방하는 것이다.

Q : 리두 로그 그룹 내 리두 로그 파일 멤버 가운데 하나가 손실된 경우의 프로세스를 가장 잘 설명한 것은?

a. 데이타베이스는 작동을 멈추며 트랜잭션은 더 이상 처리되지 않는다.

b. 데이타베이스는 유효한 리두 로그 그룹 멤버에 계속해서 쓰기 작업을 하며 트랜잭션은 계속해서 처리된다.

c. 데이타베이스는 정지되며 Database Administrator에게 문제 해결을 요구한다.

d. 로그 스위치는 발생하지 않는다.

리두 로그 그룹의 멤버를 이용할 수 없게 된 경우, Oracle9*i* Database는 그룹이 가득 찰 때까지 현재 그룹 내의 유효한 멤버에 계속해서 쓰기 작업을 하게 되고, 이어서 로그 스위치가 발생한다. 따라서 정답은 B이다. 로그 그룹 멤버를 이용할 수 없는 경우, 데이타베이스는 alert.log 파일에 오류 메시지를 기록한다. 다른 디바이스에 위치해 있는 동일한 리두 로그 그룹의 멤버를 보유함으로써 손실된 리두 로그 파일로 인한 데이타베이스 장애를 막을 수 있다.

동일한 리두 로그 그룹의 모든 멤버를 포함하고 있는 디바이스에 장애가 발생한 경우, Database Administrator는 복구를 수행해야 한다. 복구가 완료되기 전에는 데이타베이스를 이용할 수 없다. 가장 좋은 방법은 Database Administrator가 초기에 데이타베이스를 구성해서 이러한 상황이 발생하지 않도록 하는 것이다. 또한, 문제가 있다 하더라도 데이타베이스 운영이 지속되는지 여부를 파악하기 위해 규칙적으로 alert.log 파일을 검사하는 것이 좋다.

앞에서 언급했듯이, 하나의 리두 로그 그룹에서 다른 그룹으로의 스위칭 프로세스를 로그 스위치라고 한다. Oracle9*i* Database 환경에서 Log Writer 프로세스는 리두 로그 버퍼에서 데이타를 가져와 리두 로그 파일에 기록한다. 뿐만 아니라, 로그 스위치가 일어나면 체크포인트도 작동한다.

Q : 체크포인트에 대해 올바른 설명 3가지를 고르시오.

a. 체크포인트는 자동 로그 스위치가 수행되는 경우 작동된다.

b. 체크포인트는 데이타베이스가 Normal, Immediate 또는 Transactional 옵션으로 종료되는 경우 작동한다.

c. Database Administrator는 체크포인트를 강제로 작동할 수 없다.

d. 체크포인트는 Database Administrator가 수동 로그 스위치를 수행하는 경우 자동으로 작동된다.

e. 체크포인트는 기본적으로 alert.log 파일에 작성된다.

정답은 A, B, D이다. Database Administrator는 ALTER SYSTEM CHECKPOINT 명령어를 통해 체크포인트를 강제로 작동할 수 있으며, 파라미터 LOG_CHECKPOINT_TO_ALERT가 TRUE로 설정되는 경우 체크포인트는 alert.log 파일에서만 작성된다는 점에서 C와 E는 올바른 설명이 아니다.

리두 로그는 중요한 데이타베이스 구조이기 때문에 Oracle9*i* Database Administrator Fundamental I 시험을 치르기에 앞서 리두 로그 파일과 관련된 개념, 운영, 명령어 및 파라미터에 대해 완벽하게 숙지해야 한다. 이러한 개념은 Oracle9*i* Database Administrator Fundamental II 및 Oracle9*i* Performance Tuning 시험에도 포함되는 내용이기 때문에, 그에 대해서는 이후 칼럼에서 보다 자세하게 설명할 것이다.

테이블스페이스 관리

Oracle9*i* Database는 여러 유형의 테이블스페이스(tablespace)를 지원한다. Database Administrator가 서로 다른 유형의 테이블스페이스로 데이타베이스를 구성함으로써 정렬 작업, 백업 및 복구 기능을 향상시키는 것은 물론, 서로 다른 액세스 방식 및 데이타 관리를 책임질 수 있도록 테이블 및 인덱스 같은 세그먼트를 분리할 수 있다는 이점이 있다. 테이블스페이스는 SYSTEM 또는 비(non) SYSTEM 테이블스페이스로 카테고리를 만들 수 있다. 또한, Oracle9*i* Database의 테이블스페이스는 로컬에서 관리되는 테이블스페이스 또는 딕셔너리 관리 테이블스페이스로 변경할 수 있다. Oracle9*i*는 로컬 관리 테이블스페이스를 생성할 수 있도록 기본으로 설정되어 있다.

Next Steps

- Oracle9*i* 다운로드
  otn.oracle.com/software/products/oracle9i
- Oracle9*i* Database 구입
  Oraclestore.oracle.com
- 오라클 인증
  oracle.com/education/certification

Q : 로컬 관리 테이블스페이스의 특징 3가지를 고르시오.

a. 확장 영역(extent)은 데이타 딕셔너리에 의해 관리된다.

b. 데이타파일의 비트맵은 데이타 파일의 블록 상태(프리 또는 사용 중)를 추적한다.

c. 테이블스페이스에 저장된 각 세그먼트는 서로 다른 스토리지 구문(clause)을 가질 수 있다.

d. 결합(coalescing)은 필요하지 않다.

e. 확장 영역의 할당 및 재할당이 수행될 때 UNDO는 생성되지 않는다.

로컬 관리 테이블스페이스에 대한 올바른 설명은 B, D, E이다. A와 C는 딕셔너리 관리 테이블 스페이스에 대한 내용이다. SYSTEM 테이블스페이스는 딕셔너리 관리 테이블 스페이스의 하나이다.

다른 유형의 로컬 관리 테이블 스페이스 및 딕셔너리 관리 테이블 스페이스는 다양한 목적으로 사용된다.

Q : 정렬 작업에 가장 적합한 테이블스페이스 유형은?

a. UNDO 테이블스페이스

b. SYSTEM 테이블스페이스

c. 임시 테이블스페이스

d. 영구 테이블스페이스

이 문제의 정답은 C라는 것을 어렵지 않게 알 수 있을 것이다. Oracle9*i*에 도입된 UNDO 테이블스페이스는 UNDO 관리에 사용되는데, UNDO 관리는 자동 수행된다는 이점을 가지고 있다. Oracle9*i* 이전 제품의 경우, UNDO 세그먼트 관리는 Database Administrator가 롤백 세그먼트를 구성 및 튜닝해야 하는 수동 프로세스였다. 영구 테이블스페이스는 테이블 및 인덱스 같은 영구적인 객체를 위해 사용해야 하는 반면, SYSTEM 테이블스페이스는 항상 딕셔너리 객체를 위해 비워 두어야 한다는 점에서 B와 D는 올바른 설명이 아니다.

Database Administrator의 작업을 단순화해 주는 또 하나의 Oracle9*i* Database 기능으로는 디폴트 임시 테이블스페이스가 있다. 이전의 Oracle Database 버전의 경우, 새롭게 생성된 사용자를 위한 디폴트 임시 테이블스페이스는 SYSTEM이었다. 따라서, Database Administrator는 디폴트 임시 테이블스페이스가 SYSTEM이 아닌 다른 것으로 변경되도록 사용자를 변경(ALTER) 해야 했다.

Q : 새롭게 생성된 데이타베이스 사용자를 DBA_EXAM이라는 임시 테이블스페이스에 자동 할당하도록 지원하는 명령어는?

a. ALTER TABLESPACE DBA_EXAM DEFAULT TEMPORARY TABLESPACE;

b. ALTER TABLESPACE DEFAULT TEMPORARY TABLESPACE DBA_EXAM;

c. ALTER DATABASE TEMPORARY TABLESPACE DBA_EXAM;

d. ALTER DATABASE DEFAULT TEMPORARY TABLESPACE DBA_EXAM;

정답은 D이다. Database Administrator는 이 명령어를 통해 데이타베이스 내에 디폴트 임시 테이블스페이스를 설정할 수 있다. 이를 확인하고 싶은 경우에는 디폴트 임시 테이블스페이스 및 사용자를 생성해서 DBA_USERS 뷰를 질의할 수 있다. 이로써 TEMPORARY_TABLESPACE 컬럼에서 해당 사용자에 대한 디폴트 임시 테이블스페이스 이름을 확인할 수 있다.

Oracle Certified Database Administrator Professional 자격증을 획득하려면

- Introduction to Oracle9*i* : SQL
  (본지 2002년 가을호에서 설명)
- Oracle9*i* : Database Administrator Fundamentals I
  (본지 2002년 겨울호에서 설명)
- Oracle9*i* : Database Administrator Fundamentals II
- Oracle9*i* Database : Performance Tuning

의 4개 시험을 통과해야 한다.

시험 준비에 도움이 되길 바라며

이번호에서는 리두 로그 파일 및 테이블 스페이스 관리를 중점으로 설명했는데, 이 주제는 Oracle9*i* : Database Administrator Fundamentals I 시험에서 큰 비중을 차지한다. 여기에서 제공되는 샘플 질문들은 여러분이 시험을 준비하는 데 많은 도움이 될 것이다. Introduction to Oracle9*i* : SQL 및 Oracle9*i* : Database Administrator Fundamentals I 시험을 통과하면 Oracle Certified Associate 자격증을 취득하게 된다.

다음호에서는 Oracle9*i* : Database Administrator Fundamentals II 시험과 관련된 내용을 다룰 것이다. 이 시험은 Oracle9*i* Database Administrator certification path : Oracle9*i* Database Administrator OCP(Oracle Certified Professional)에서 다음 레벨의 자격을 획득하기 위해 필요하다.

이 글의 원문은 http://otn.oracle.com/oramag/oracle/02-nov/o62ocp.html을 참조하기 바랍니다.

CHANNEL VENDOR NEWS

Mongoose PortalStudio에서도 Oracle9*i* Application Server 지원

엔터프라이즈 포탈 솔루션 업체인 Mongoose Technology는 Oracle9*i* Application Server에서 포탈 라이프 사이클을 관리할 수 있는 기능이 포함된 Mongoose PortalStudio 버전 3.0을 출시했다. 이 새로운 버전의 제품에는 Oracle9*i* Application Server를 위한 여러 기능들이 새롭게 추가되었다.

- **검증 및 자동 배포** : PortalStudio 마법사를 사용하여 Oracle9*i* Application Server에서 포탈 및 웹 서비스를 검사하고 배포할 수 있다.
- **스테이징 및 반복 개발** : 개발자들은 이 기능과 함께 마법사를 사용하여 PC에서 개발된 포탈을 개발 팀 서버에 재배포할 수 있다.
- **단계적 포탈** : PortalStudio는 '포탈 안의 포탈'을 지원하기 때문에, PortalStudio에서 개발된 엔터프라이즈 포탈에는 다양한 업체들의 포틀릿, 웹 애플리케이션, 웹 서비스를 통합할 수 있다.

PortalStudio에는 오래된 웹 애플리케이션과 포탈을 Oracle9*i* Application Server로 이전하는 데 필요한 툴도 포함되어 있다. 즉, ▲ LDAP, Active Directory, JDBC 데이타베이스에 대한 로그인 모듈 및 JAAS를 사용하여 기존 액세스 제어 메커니즘을 사용할 수 있도록 해주는 보안 기능 ▲ JDBC 커넥터를 사용하는 테이블 및 트리 포틀릿을 통해 오라클과 다른 벤더의 데이타베이스를 통합하여 주는 기능 ▲ 많은 엔터프라이즈 애플리케이션의 통합을 가능하게 해주는 J2EE 커넥터 아키텍처(JCA) 리소스 어댑터 및 XML 커넥터 ▲ 지식관리 및 협력 툴 등이 포함되어 있다.

www.mongoosetech.com

Transenigma Forms*Facelift를 사용하여 기존 Oracle Forms 이전 가능

Transenigma는 기존 Oracle Forms 애플리케이션에 사용할 수 있는 자동화 이전 서비스인 Forms*Facelift를 발표했다. Forms*Facelift는 오라클 DBA들이 Oracle Forms의 모든 버전으로부터 현재 문자 모드에서 실행되는 폼, 최신 6*i*, Oracle9*i* 버전으로 애플리케이션을 이전할 수 있도록 도와 준다.

Forms*Facelift를 사용하면 새로운 GUI 기능을 추가하거나 새로운 색상 체계를 적용할 수 있을 뿐 아니라, 그래픽, 로고, 마우스 이동 기능, 다국어 기능을 추가하는 일을 할 수도 있다. 이 밖에 Forms*Facelift는 보고 툴과 메뉴를 최신 버전으로 향상시키는 역할도 수행한다.

Transenigma의 이전 서비스는 클라이언트/서버, 씬-클라이언트, 웹 지원 Oracle Forms 환경으로의 이전을 지원한다.

www.transenigma.co.uk

핵심 데이타 복원에 용이한 Lumigent의 Log Explorer for Oracle

Lumigent Technologies는 최근에 Log Explorer for Oracle을 출시하였다. Log Explorer는 데이타 손상의 원인을 찾아내고 문제가 발생할 때 핵심 데이타를 복원하는 수단을 제공함으로써, 엔터프라이즈 정보를 위한 지속적인 보안 액세스를 보장한다.

Log Explorer에서는 오라클 트랜잭션 로그를 직접 사용할 수 있는데, 이렇게 하면 교묘한 애플리케이션 문제와 사용자 문제를 해결할 수도 있고, 데이타베이스 변경의 원인을 찾아낼 수도 있고, 데이타를 온라인상에서 빠르고 선택적으로 복원할 수도 있고, 전통적인 복원 기법들이 실패하는 경우에도 데이타를 복원할 수 있다. 또한 Log Explorer가 지속적인 운영을 위해 데이타베이스의 가용성을 연속적으로 유지해 주기 때문에, 사용자들은 다운타임 없이도 데이타베이스 문제를 수정하고 보안 문제를 해결할 수 있다.

Log Explorer for Oracle을 사용하면, 애플리케이션과 데이타베이스의 상호작용을 이해하고, 모든 트랜잭션의 세부사항을 검토하고, 로우들이 시간에 따라 어떻게 변화하는지를 살펴볼 수 있으며, 보고 및 분석을 위해서도 로그 데이타를 익스포트하거나 또한 전체 백업을 재로드하지 않고서도 선택적으로 변경 사항을 백업할 수도 있다. 또한 이 제품은 사용자가 개별 트랜잭션이나 트랜잭션 집합을 재실행할 수 있도록 도와줄 뿐 아니라, 삭제되거나 잘려진 테이블의 복원도 지원한다.

www.lumigent.com

PDA 피부에 이식되는 식별장치, 강력한 패드, 핸드헬드 데이타 전송

어깨에 이식된 칩 : Applied Digital Solutions의 VeriChip

사생활 옹호자들의 모골이 송연해질 이야기지만, Applied Digital Solutions의 VeriChip은 피부 밑에 이식되는 12 x 2.1mm 고주파 장치로서, 독점 외부 스캐너에 의해 활성화되면 고유한 확인 번호를 전송한다. 이 번호는 스캐너에 의해 표시되어 보안 데이타 저장 사이트로 중계된다. Applied Digital Solutions은 원자력 발전소와 같은 1급 보안 지역에서 이 장치를 사용할 것을 권장하지만, ID 절도를 막을 수 있도록 고객에게 꼬리표를 붙이면 금융기관에도 효과가 있을 것이라고 제안하고 있다.

www.adsx.com/prodservpart/verichip.html

무선 전력의 범용 충전 플랫폼 : Splashpower

영국 캠브리지에 위치한 Splashpower Ltd.는 플러그나 크레들 없이 휴대용 전자 장치를 충전하는 시스템을 디자인했다. 이 프로토타입 시스템은 두 부분으로 이루어지는데, 그 중 첫 번째인 SplashPad는 범용 충전 플랫폼으로서 전기 콘센트에 꽂는 것이다. 그리고 SplashModule은 수신기로, PDA 또는 휴대 전화와 같은 호스트 장치에 삽입된다. 호환성 있는 장치를 충전하려면 그 장치를 SplashPad 위에 놓기만 하면 된다.

www.splashpower.com

피부간 데이타 전송 : NTT DoCoMo

옛날에는 서로 악수를 하는 것만으로도 계약이 체결되었다. 이제 Nippon Telegraph and Telephone과 그 자회사인 NTT DoCoMo의 연구가 결실을 얻게 되면, 그러한 악수 한 번으로도 계약 체결 뿐만 아니라 양측 PDA간에 전자 계약서를 전송하는 것이 가능하게 된다. 이러한 기술은 사람 피부의 전도성(또는 의류의 전도성)을 기초로 한 시스템을 통해 손을 잡는 두 사람 사이에 최고 10메가비트 속도로 데이타가 전송되도록 하는 것이다. 피부간 데이타 전송은 새로운 기술은 아니지만, 광대역 인터넷 연결에 필적할 만한 속도를 제공하는 기술로는 최초이다.

기사 출처 : NewScientist.com
(www.newscientist.com/hottopics/tech/article.jsp?id=ns99992891)