
Goodus 기술노트 [26 회]

AWR & ADDM

Author	이창수, 박덕우
Creation Date	2007-12-20
Last Updated	2007-12-20
Version	1.0
Copyright(C) 2004 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2007-12-20	이창수, 박덕우	문서 최초 작성
2			
3			

Contents

1. AWR.....	3
1.1. AWR (Automatic Workload Repository) 란?.....	3
1.2. AWR의 구성	3
1.2.1. AWR Overview	3
1.2.2. AWR 환경 설정하기	3
1.2.3. AWR 구현 원리	4
1.2.4. 통계의 활용.....	6
1.2.5. 10g OEM을 이용한 AWR 보고서 만들기	6
1.2.6. 타임 모델 (Time Model)	9
1.2.7. Active Session History	10
1.2.8. 수작업으로 스냅샷 생성하기	11
1.2.9. 베이스라인	12
2. ADDM.....	14
2.1. ADDM (AUTOMATIC DATABASE DIAGNOSTIC MONITOR) 란?	14
2.2. ADDM의 구성	14
2.2.1. ADDM Overview.....	14
2.2.2. ADDM 구성과 작동 원리	14
2.2.3. SQL Tuning Advisor를 이용한 Access 분석	19
2.2.4. SQL Tuning Advisor를 이용한 SQL 튜닝 예시.....	19
2.2.5. ADDM과 SQL Tuning Advisor의 활용	21
2.2.6. Enterprise Manager의 활용	22
2.2.7. 결론	26
3. Pro-Active Tuning Service	27
3.1. 실제 사용자(End-User) 관점의 응답시간 튜닝	27
3.2. 최상의 성능 상태로 비즈니스 고가용성을 유지	27
3.3. Knowledge Transfer.....	28
3.4. Tuning 범위 확대	28
3.5. 기대효과	29
3.5.1. 재무적 관점.....	29
3.5.2. 서비스 관점.....	29
3.5.3. 사용자 관점.....	30
3.5.4. 혁신적 관점.....	30
3.6. 제안 Package별 가격.....	31
3.6.1. DBMS Tuning.....	31
3.6.2. DBMS Tuning+ APM + NA.....	31
3.6.3. Server (H/W, OS) Tuning + Storage 재구성 컨설팅 + DBMS Tuning	31
3.7. Package 설명.....	32

1. AWR

1.1. AWR (Automatic Workload Repository) 란?

AWR(Automatic Workload Repository)은 Oracle Database 10g 의 내장 저장소이다. 오라클 데이터베이스는 일정한 주기로 모든 주요 통계와 작업 로드 정보에 대한 스냅샷을 만들어 AWR에 저장한다. 이렇게 수집된 데이터는 보고서나 뷰를 통해 표시할 수 있다. AWR 스냅샷은 기본적으로 시간당 1회 캡처되며 7일간 데이터베이스에 보존된다.

AWR을 이용하여 분석과 튜닝을 위한 데이터베이스 성능 통계정보와 메트릭을 수집하고, 데이터베이스에서 사용한 정확한 시간을 확인하거나 세션 정보를 저장할 수 있습니다.

1.2. AWR의 구성

1.2.1. AWR Overview

데이터베이스 성능에 관련한 문제가 생겼을 때, DBA로서 가장 먼저 취하는 조치는 아마도 문제에 일정한 패턴이 존재하는지 확인하는 것이 가장 일반적인 접근방법 중 하나일 것입니다. “동일한 문제가 반복되는가?”, “특정한 시간대에만 발생하는가?”, 또는 “두 가지 문제에 연관성이 있는가?” 등의 질문을 먼저 제기해 봄으로써 보다 정확한 진단을 수행할 수 있습니다.

Oracle DBA들은 데이터베이스 운영에 관련한 통계정보를 수집하거나 성능 메트릭(metric)을 추출하기 위해 써드 파티 툴, 또는 직접 개발한 툴을 사용하고 있습니다. 이렇게 수집된 정보는, 문제 발생 이전과 이후의 상태를 비교하는 데 이용됩니다. 과거에 발생했던 이벤트들을 재현해 봄으로써 현재 문제를 다양한 관점에서 분석할 수 있습니다. 이처럼 관련 통계정보들을 지속적으로 수집하는 것은 성능 분석에서 매우 중요한 작업 중의 하나입니다.

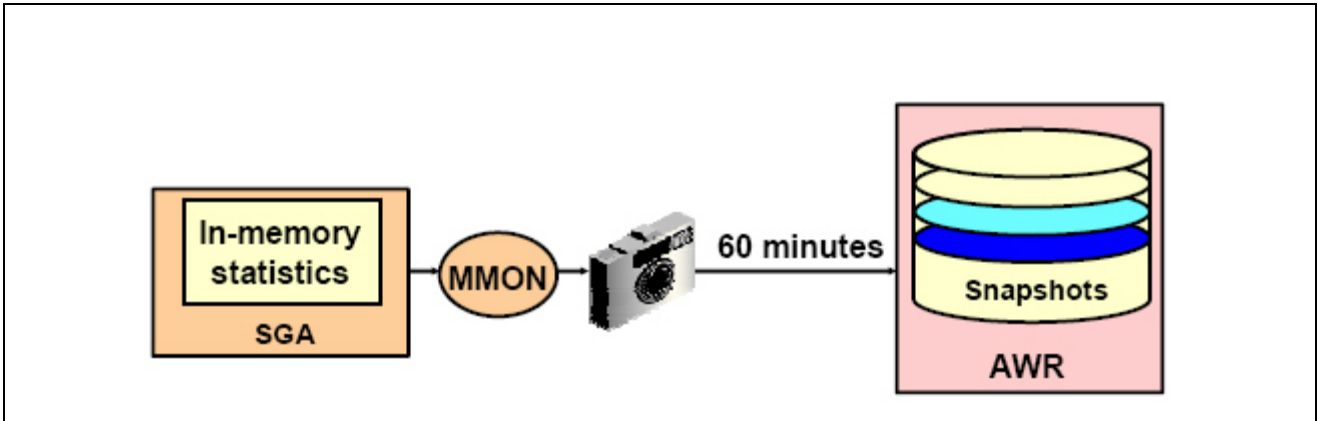
오라클은 한동안 이를 위해 Statspack이라는 이름의 빌트-인 툴을 제공하기도 했습니다. Statspack은 DB 튜닝 등에 매우 유용하게 사용되었지만, 성능 관련 트러블슈팅 과정에서 요구되는 안정성이 약간은 결여되었다는 것이 문제였다. Oracle Database 10g는 성능 통계정보의 수집과 관련하여 그 기능이 많이 향상된 Automatic Workload Repository(AWR)를 제공합니다. AWR은 데이터베이스와 함께 설치되며, 기본적인 통계정보뿐 아니라 통계정보로부터 유추된 메트릭(derived metric)도 함께 수집합니다.

1.2.2. AWR 환경 설정하기

AWR 을 이용한 새로운 기능들은 \$ORACLE_HOME/rdbms/admin 디렉토리의 awrrpt.sql 스크립트를 실행하고, 수집된 통계정보와 메트릭 정보를 바탕으로 생성된 리포트를 확인함으로써 가장 쉽게 이해할 수 있습니다. awrrpt.sql 스크립트는 Statspack 과 유사한 구조로 되어있습니다. 먼저 현재 저장된 AWR 스냅샷을 모두 표시한 후, 시간 간격 설정을 위한 입력값을 요구합니다. 출력은 두 가지 형태로 제공됩니다. 텍스트 포맷 출력은 Statspack 리포트와 유사하지만, AWR 리포트토리를 기반으로 하며 (디폴트로 제공되는) HTML 포맷을 통해 section/subsection 으로 구분된 하이퍼링크를 제공하는 등 사용자 편의성을 강화하였다는 점에서 차이를 갖습니다. 먼저 awrrpt.sql 스크립트를 실행하여 리포트를 확인해 보시고, AWR 의 기능에 대한 특성을 이해하시기 바랍니다.

1.2.3. AWR 구현 원리

이제 AWR 의 설계방식과 구조에 대해 알아보기로 합시다. AWR 은 수집된 성능관련 통계정보가 저장되며 이를 바탕으로 성능 메트릭을 제공함으로써 잠재적인 문제의 원인 추적을 가능하게 해주는 근간을 제공해 줍니다



Statspack 의 경우와 달리, Oracle10g 는 AWR 을 활용하여 새로운 MMON 백그라운드 프로세스와, 여러 개의 슬레이브 프로세스를 통해 자동적으로 매시간별 스냅샷 정보를 수집합니다. 공간 절약을 위해, 수집된 데이터는 7 일 후 자동으로 삭제됩니다. 스냅샷 빈도와 보관 주기는 사용자에게 의해 설정 가능합니다.

아래 내용은 현재 설정값을 확인하고 스냅샷주기 20 분, 보관주기 2 일로 변경하는 방법은 입니다.

```
SQL> SHOW USER
```

USER 은 "SYSTEM"입니다 ← DBA 유저로 접속

```
SQL> SELECT snap_interval , retention FROM dba_hist_wr_control;
```

SNAP_INTERVAL	RETENTION	
+00000 01:00:00.0	+00007 00:00:00.0	← SNAP_INTERVAL : 스냅샷주기, RETENTION : 보관주기
		← 스냅샷주기 = 1 시간 (Default), 보관주기 = 7 일 (Default)

```
SQL> BEGIN
```

```
    dbms_workload_repository.modify_snapshot_settings (
        interval => 20,          ← 스냅샷주기 = 20 분단위
        retention => 2*24*60     ← 보관주기 = 2 일
    );
```

```
END;
```

PL/SQL 처리가 정상적으로 완료되었습니다.

```
SQL> SELECT snap_interval , retention FROM dba_hist_wr_control;
```

SNAP_INTERVAL	RETENTION	
+00000 00:20:00.0	+00002 00:00:00.0	← 스냅샷주기 = 20 분, 보관주기 = 2 일

AWR 은 수집된 통계를 저장하기 위해 여러 개의 테이블을 사용합니다. 이 테이블들은 모두 SYS 스키마의 SYSAUX 테이블스페이스 내에 저장되어 있으며, WRM\$_* 또는 WRH\$_*의 네임 포맷을 갖습니다. WRM\$_* 테이블은 수집 대상 데이터베이스 및 스냅샷에 관련한 메타데이터 정보를, WRH\$_* 테이블은 실제 수집된 통계 정보를 저장하는데 사용됩니다. (WRH\$_*의 H 는 “Historical”, WRM\$_*의 M 은 “metadata”의 약자를 의미합니다.) 이 테이블을 기반으로 DBA_HIST_라는 prefix 를 갖는 여러 가지 뷰가 제공되고 있으며, 이 뷰들을 응용하여 자신만의 성능 분석 툴을 만들 수도 있습니다. 뷰의

이름은 테이블 이름과 직접적인 연관성을 갖습니다. 예를 들어 DBA_HIST_SYSMETRIC_SUMMARY 뷰는 WRH\$_SYSMETRIC_SUMMARY 테이블을 기반으로 합니다.

AWR 히스토리 테이블은 Statspack 에서는 수집되지 않았던 다양한 정보(테이블스페이스 사용 통계, 파일시스템 사용 통계, 운영체제 통계 등)를 제공합니다. 테이블의 전체 리스트는 아래와 같이 데이터 디렉터리 조회를 통해 확인할 수 있습니다.

```
SQL> SELECT view_name FROM user_views WHERE view_name like 'DBA_HIST_%';
```

← 10g 설치시 총 78 개가 리스트됨

```
VIEW_NAME
-----
DBA_HIST_DATABASE_INSTANCE
DBA_HIST_SNAPSHOT
DBA_HIST_SNAP_ERROR
DBA_HIST_BASELINE

--- 중략 ---

DBA_HIST_BUFFERED_SUBSCRIBERS
DBA_HIST_RULE_SET

78 rows selected.
```

DBA_HIST_METRIC_NAME 뷰는 AWR 에 수집되는 주요 메트릭과 메트릭이 속한 그룹, 그리고 수집 단위(unit) 등을 정의하고 있습니다. DBA_HIST_METRIC_NAME 뷰의 레코드에 대한 조회 결과의 예가 아래와 같습니다.

컬럼명	데이터 값	컬럼 설명
DBID	1169471561	데이터베이스 ID
GROUP_ID	2	메트릭 그룹 ID
GROUP_NAME	System Metrics Long Duration	메트릭 그룹 이름
METRIC_ID	2075	메트릭 ID
METRIC_NAME	CPU Usage Per Sec	메트릭 이름
METRIC_UNIT	CentiSeconds Per Second	측정단위

위에서는 "초당 CPU 사용량(CPU Usage Per Sec)" 메트릭이 "100 분의 1 초(CentiSeconds Per Second)" 단위로 수집되고 있으며, 이 메트릭이 "System Metrics Long Duration" 그룹에 속함을 확인할 수 있습니다. 이 레코드를 DBA_HIST_SYSMETRIC_SUMMARY 와 JOIN 하여 실제 통계를 확인할 수 있습니다.

```
SQL> SELECT begin_time, intsize, num_interval, minval, maxval, average, standard_deviation sd
FROM dba_hist_sysmetric_summary WHERE metric_id = 2075;
```

BEGIN	INTSIZE	NUM_INTERVAL	MINVAL	MAXVAL	AVERAGE	SD (표준편차)
2007/12/15 10:39:52	179916	30	0	33	3(초)	9.81553548 ← 부정확
2007/12/15 11:09:56	180023	30	21	35	28(초)	5.91543912 ← 비교적정확

위 조회 결과를 통해 백 분의 1 초 단위로 CPU 자원이 어떻게 소비되고 있는지 확인할 수 있습니다. SD(standard deviation, 표준편차) 값을 참조하면 계산된 평균 값이 실제 부하와 비교하여 얼마나 오차를 갖는지 분석 가능합니다. 첫 번째 레코드의 경우, 초당 백 분의 3 초의 CPU 시간이 소모된 것으로 계산되었지만, 표준 편차가 9.81 이나 되므로 계산된 3 의 평균값이 실제 부하를 정확하게 반영하지 못하는 것으로 볼 수 있습니다. 반면 28 의 평균값과 5.9 의 표준 편차를 갖는 두 번째 레코드가 실제 수치에 더 가깝다고 볼 수 있습니다. 이러한 트렌드 분석을 통해 성능 메트릭과 환경 변수의 상관 관계를 보다 명확하게 이해할 수 있습니다.

1.2.4. 통계의 활용

지금까지 AWR 의 수집 대상이 어떻게 정의되는지 알아보았습니다. 이번에는 수집된 데이터를 어떻게 활용할 수 있는지 설명하기로 합니다.

성능 문제는 독립적으로 존재하는 경우가 거의 없으며, 대개 다른 근본적인 문제를 암시하는 징후로서 해석되는 것이 일반적입니다. 전형적인 튜닝 과정의 예를 짚어보기로 합니다. DBA 가 시스템 성능이 저하되었음을 발견하고 wait 에 대한 진단을 수행합니다. 진단 결과 “buffer busy wait”이 매우 높게 나타나고 있음을 확인합니다. 그렇다면 문제의 원인은 무엇일까요? 여러 가지 가능성이 존재합니다. 인덱스의 크기가 감당할 수 없을 만큼 커지고 있을 수도 있고, 테이블의 조밀도(density)가 너무 높아 하나의 블록을 메모리에 읽어 오는 데 요구되는 시간이 제한된 때문일 수도 있고, 그 밖의 다른 이유가 있을 수 있습니다. 원인이 무엇이든, 문제가 되는 세그먼트를 먼저 확인해 보는 것이 필요합니다. 문제가 인덱스 세그먼트에서 발생했다면, rebuild 작업을 수행하거나 reverse key index 로 변경하거나 또는 Oracle Database 10g 에서 새로 제공하는 hash-partitioned index 로 변경해 볼 수 있을 것입니다. 문제가 테이블에서 발생했다면, 저장 관련 매개변수를 변경해서 조밀도를 낮추거나, 자동 세그먼트 공간 관리(automatic segment space management)가 설정된 테이블스페이스로 이동할 수 있을 것입니다.

DBA 가 실제로 사용하는 접근법은 일반적인 방법론, DBA 의 경험 및 지식 등을 그 바탕으로 합니다. 만일 똑같은 일을 별도의 엔진이, 메트릭을 수집하고 사전 정의된 로직을 바탕으로 적용 가능한 방법을 추론하는 엔진이 대신해 준다면 어떨까요? DBA 의 작업이 한층 쉬워지지 않을까요?

바로 이러한 엔진이 Oracle Database 10g 에 새로 추가된 Automatic Database Diagnostic Monitor (ADDM)입니다. ADDM 은 AWR 이 수집한 데이터를 사용하여 결론을 추론합니다. 위의 예의 경우, ADDM 은 buffer busy wait 이 발생하고 있음을 감지하고, 필요한 데이터를 조회하여 wait 이 실제로 발생하는 세그먼트를 확인한 후, 그 구조와 분포를 평가함으로써 최종적으로 해결책을 제시합니다. AWR 의 스냅샷 수집이 완료될 때마다, ADDM 이 자동으로 호출되어 메트릭을 점검하고 권고사항을 제시합니다. 결국 여러분은 데이터 분석 및 권고사항 제시를 담당하는 풀 타임 DBA 로봇을 하나 두고, 보다 전략적인 업무에 집중할 수 있게 된 셈입니다.

1.2.5. 10g OEM을 이용한 AWR 보고서 만들기

Enterprise Manager 10g 콘솔의 “DB Home(홈)” 페이지에서 ADDM 의 권고사항과 AWR 리포지토리 데이터를 확인할 수 있습니다. AWR 리포트를 보려면, Administration(관리탭) → Workload Repository(통계관리-자동 작업 로드 저장소) → Snapshot(스냅샷)의 순서로 메뉴를 따라가야 합니다.

① 홈 (DB Home)

ORACLE Enterprise Manager 10g Database Control

데이터베이스

SYSTEM(으)로 로그인됨

데이터베이스 인스턴스: orcl

홈 **관리** 유지 관리

페이지가 새로 고쳐진 시간 2007.12.19. 오후 3:39:26 새로 고침 데이터 보기 자동(60초)

일반 사항

상태 **작동**

작동 시작 시간 2007.12.19. 오전 10시 27분 32초 KST

인스턴스 이름 **orcl**

버전 **10.2.0.1.0**

호스트 **192.168.10.102**

리스너 **LISTENER_192.168.10.102**

[모든 속성 보기](#)

× 호스트 CPU

로드 13.00 페이징 8.99

기타 orcl

활성 세션

최대 CPU 1

대기 사용자 I/O CPU

SQL 응답 시간

기존 요소가 비어 있습니다.

[기본 요소 재설정](#)

진단 요약

ADDM 검색 결과 0

모든 정책 위반 2

경보 로그 **ORA- 오류 없음**

공간 요약

데이터베이스 크기(GB) 0.839

문제 테이블스페이스 0

세그먼트 권고자 권장 사항 0

공간 위반 0

사용된 덤프 영역(%) 31

고가용성

인스턴스 복구 시간(초) 38

마지막 백업 **해당 사항 없음**

사용 가능한 플래시 복구 영역(%) 100

플래시백 로깅 **비활성화됨**

② 관리 (Administration)

데이터베이스 인스턴스: orcl

홈 성능 **관리** 유지 관리

[관리] 탭에는 데이터베이스 객체를 관리하고 오라클 데이터베이스 내에서 데이터베이스 작업을 시작할 수 있는 링크가 표시됩니다. [유지 관리] 탭에는 오라클 데이터베이스 간 또는 오라클 데이터베이스 외부의 데이터 흐름을 제어하는 기능을 제공하는 링크가 표시됩니다.

데이터베이스 관리

저장 영역

제어 파일

데이터베이스

임시 테이블스페이스 그룹

데이터 파일

롤백 세그먼트

리우 로그 그룹

아카이브 로그

데이터베이스 구성

메모리 매개변수

실행 취소 관리

모든 초기화 매개변수

데이터베이스 기능 사용량

데이터베이스 스케줄러

작업

제어

일정

프로그램

작업 클래스

작업

작업 그룹

전역 속성

통계 관리

자동 작업 로드 저장소

작업 로드 통계 관리

데이터베이스 변경

ASM으로 이전

로컬에서 관리되는 테이블스페이스로 만들기

리소스 관리자

모니터

소비자 그룹

소비자 그룹 매핑

계획

정책

정책 라이브러리

정책 위반

③ 자동 작업 로드 저장소 (Workload Repository)

데이터베이스 인스턴스: orcl > 자동 작업 로드 저장소

SYSTEM(으)로 로그인됨

자동 작업 로드 저장소

페이지가 새로 고쳐진 시간 2007.12.19. 오후 3:41:31 새로 고침

자동 작업 로드 저장소는 성능 튜닝에 사용되는 데이터베이스 통계를 저장하는 데 사용됩니다.

일반 사항

편집

스냅샷 보존(일) 2

스냅샷 간격(분) 20

수집 레벨 TYPICAL

다음 스냅샷 캡처 레벨 2007.12.19. 오후 4:00:33

스냅샷 및 보관된 스냅샷 집합 관리

스냅샷 16

보관된 스냅샷 1

최근 스냅샷 시간 2007.12.19. 오후 3:40:33

최초 스냅샷 시간 2007.12.19. 오전 12:01:02

④ 스냅샷 (Snapshot)

스냅샷의 보고서를 생성하려면 스냅샷 리스트에서 해당 스냅샷을 선택한 후 보고서탭을 누르면 "WORKLOAD REPOSITORY 보고서"가 생성되며, html 파일로도 저장이 가능하다.

데이터베이스 인스턴스: orcl > 자동 작업 로드 저장소 > 스냅샷 SYSTEM(으)로 로그인됨

확인
스냅샷이 성공적으로 삭제됨

스냅샷
스냅샷은 단일 시점의 데이터베이스 통계 모음입니다. 스냅샷에 있는 정보를 사용하여 데이터베이스 페이지가 새로 고쳐진 시 문제를 진단할 수 있습니다. 2007. 12. 19. 오후 3:43:36 새로 고침

시작 스냅샷 선택
이동 시간: 07. 12. 19. (예: 03. 12. 15.) 오후 4:00 ▼ 실행

삭제
작업
보관된 스냅샷 집합 생성 ▼
실행
생성

선택 m	캡처 시간	수집 레벨	보관된 스냅샷 집합 내
<input checked="" type="radio"/> 10	2007. 12. 19. 오후 2:00:51	TYPICAL	
<input type="radio"/> 11	2007. 12. 19. 오후 2:20:58	TYPICAL	
<input type="radio"/> 12	2007. 12. 19. 오후 2:40:10	TYPICAL	
<input type="radio"/> 13	2007. 12. 19. 오후 3:00:12	TYPICAL	
<input type="radio"/> 14	2007. 12. 19. 오후 3:20:20	TYPICAL	
<input type="radio"/> 15	2007. 12. 19. 오후 3:26:18	TYPICAL	
<input type="radio"/> 16	2007. 12. 19. 오후 3:40:27	TYPICAL	

⑤ 보고서 생성중

데이터베이스 인스턴스: orcl > 자동 작업 로드 저장소 > 스냅샷 > 스냅샷 세부 정보 SYSTEM(으)로 로그인됨

처리 중: 보고서 보기

보고서를 생성하는 중입니다.

취소

취소

Copyright (c) 1996, 2005, Oracle. All rights reserved.
Oracle Database Manager 10g 데이터베이스 컨트롤러 정보

데이터베이스 | 설정 | 환경설정 | 도움말 | 로그아웃

⑥ AWR 보고서 생성

데이터베이스 인스턴스: orcl > 자동 작업 로드 저장소 > 스냅샷 > 스냅샷 세부 정보 SYSTEM(으)로 로그인됨

스냅샷 세부 정보

ADDM 실행 보기

파일에 저장

세부 정보 | 보고서

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
ORCL	1169471561	orcl	1	10.2.0.1.0	NO	COMPUTER

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	15	19-12월-07 15:26:18	26	12.7
End Snap:	16	19-12월-07 15:40:27	28	12.7
Elapsed:		14.15 (mins)		
DB Time:		0.74 (mins)		

Report Summary

Cache Sizes

	Begin	End		
Buffer Cache:	328M	328M	Std Block Size:	8K
Shared Pool Size:	92M	92M	Log Buffer:	2,872K

특정 조건을 기준으로 알림 메시지를 생성하도록 설정하는 것도 가능합니다. 이 기능은 Server Generated Alert 라 불리며, Advanced Queue 에 푸시(push) 형태로 저장되고 리스닝 중인 모든 클라이언트에 전달되는 형태로 관리됩니다. Enterprise Manager 10g 역시 Server Generated Alert 의 클라이언트의 하나로서 관리됩니다.

1.2.6. 타임 모델 (Time Model)

성능 문제가 발생했을 때, 응답시간을 줄이기 위한 방법으로 DBA 의 머릿속에 가장 먼저 떠오르는 것은 무엇일까요? 말할 필요도 없이, 문제의 근본원인을 찾아내어 제거하는 것이 최우선일 것입니다. 그렇다면 얼마나 많은 시간이 (대기가 아닌) 실제 작업에 사용되었는지 어떻게 확인할 수 있을까요? Oracle Database 10g 는 여러 가지 자원에 관련한 실제 사용 시간을 확인하기 위한 타임 모델(time model)을 구현하고 있습니다. 전체 시스템 관련 소요 시간 통계는 V\$SYS_TIME_MODEL 뷰에 저장됩니다. V\$SYS_TIME_MODEL 뷰에 대한 쿼리 결과의 예가 아래와 같습니다.

```
SQL> SELECT stat_name, value FROM v$sys_time_model;
```

STAT_NAME	VALUE	
DB time	791943871	← 인스턴스 시작 후 DB 사용 누적치
DB CPU	447140760	
background cpu time	299960255	
sequence load elapsed time	101608767	
parse time elapsed	1757328	
hard parse elapsed time	159813929	
sql execute elapsed time	15732352	
connection management call elapsed time	234811	
failed parse elapsed time	23	
hard parse (sharing criteria) elapsed time	234592	
hard parse (bind mismatch) elapsed time	5341	
PL/SQL execution elapsed time	23193722	
inbound PL/SQL rpc elapsed time	0	
PL/SQL compilation elapsed time	2	
Java execution elapsed time	238940	
bind/define call elapsed time	0	

위에서 DB time 이라는 통계정보는 인스턴스 시작 이후 데이터베이스가 사용한 시간의 누적치를 의미합니다. 샘플 작업을 실행한 다음 다시 뷰를 조회했을 때 표시되는 DB time 의 값과 이전 값의 차이가 바로 해당 작업을 위해 데이터베이스가 사용한 시간이 됩니다. 튜닝을 거친 후 DB time 값의 차이를 다시 분석하면 튜닝을 통해 얻어진 성능 효과를 확인할 수 있습니다.

이와 별도로 V\$SYS_TIME_MODEL 뷰를 통해 파싱(parsing) 작업 또는 PL/SQL 컴파일 작업에 소요된 시간 등을 확인할 수 있습니다. 이 뷰를 이용하면 시스템이 사용한 시간을 확인하는 것도 가능합니다. 시스템 / 데이터베이스 레벨이 아닌 세션 레벨의 통계를 원한다면 V\$SESS_TIME_MODEL 뷰를 이용할 수 있습니다. V\$SESS_TIME_MODEL 뷰는 현재 연결 중인 active/inactive 세션들의 통계를 제공합니다. 세션의 SID 값을 지정해서 개별 세션의 통계를 확인할 수 있습니다.

이전 릴리즈에서는 이러한 통계가 제공되지 않았으며, 사용자들은 여러 정보 소스를 참고해서 근사치를 추측할 수 밖에 없었습니다.

1.2.7. Active Session History

Oracle Database 10g 의 V\$SESSION 에도 개선이 이루어졌습니다. 가장 중요한 변화로 wait 이벤트와 그 지속시간에 대한 통계가 뷰에 추가되어, V\$SESSION_WAIT 를 별도로 참조할 필요가 없게 되었다는 점을 들 수 있습니다. 하지만 이 뷰가 실시간 정보를 제공하므로, 나중에 다시 조회했을 때에는 중요한 정보가 이미 사라져 버리고 없을 수 있습니다. 예를 들어 wait 상태에 있는 세션이 있음을 확인하고 이를 조회하려 하면, 이미 wait 이벤트가 종료되어 버려 아무런 정보도 얻지 못하는 경우가 있을 수 있습니다.

또 새롭게 추가된 Active Session History(ASH)는 AWR 과 마찬가지로 향후 분석 작업을 위해 세션 성능 통계를 버퍼에 저장합니다. AWR 과 다른 점은, 테이블 대신 메모리가 저장 매체로 이용되며 V\$ACTIVE_SESSION_HISTORY 등을 통해 조회된다는 사실입니다. 데이터는 1 초 단위로 수집되며, 액티브 세션만이 수집 대상이 됩니다. 버퍼는 순환적인 형태로 관리되며, 저장 메모리 용량이 가득 차는 경우 오래된 데이터부터 순서대로 삭제됩니다. 이벤트를 위해 대기 중인 세션의 수가 얼마나 되는지 확인하려면 아래와 같이 조회하면 됩니다.

```
SQL> SELECT session_id||','||session_serial# SID, n.name, wait_time, time_waited
FROM v$active_session_history a, v$event_name n
WHERE n.event# = a.event#;
```

SID	NAME	WAIT_TIME	TIME_WAITED
166,1	log file parallel write	0	288
166,1	log file parallel write	0	223
161,1	control file sequential read	0	39401
166,1	log file parallel write	0	30367
146,271	null event	0	2712
167,1	db file parallel write	0	385

--- 후략 ---

위 쿼리는 이벤트 별로 대기하는 데 얼마나 많은 시간이 사용되었는지를 알려 줍니다. 특정 wait 이벤트에 대한 드릴다운을 수행할 때에도 ASH 뷰를 이용할 수 있습니다. 예를 들어, 세션 중 하나가 buffer busy wait 상태에 있는 경우 정확히 어떤 세그먼트에 wait 이벤트가 발생했는지 확인하는 것이 가능합니다. 이때 ASH 뷰의 CURRENT_OBJ# 컬럼과 DBA_OBJECTS 뷰를 조인하면 문제가 되는 세그먼트를 확인할 수 있습니다.

ASH 뷰는 그 밖에도 병렬 쿼리 서버 세션에 대한 기록을 저장하고 있으므로, 병렬 쿼리의 wait 이벤트를 진단하는 데 유용하게 활용됩니다. 레코드가 병렬 쿼리의 slave process 로서 활용되는 경우, coordinator server session 의 SID 는 QC_SESSION_ID 컬럼으로 확인할 수 있습니다. SQL_ID 컬럼은 wait 이벤트를 발생시킨 SQL 구문의 ID 를 의미하며, 이 컬럼과 V\$SQL 뷰를 조인하여 문제를 발생시킨 SQL 구문을 찾아낼 수 있습니다. CLIENT_ID 컬럼은 웹 애플리케이션과 같은 공유 사용자 환경에서 클라이언트를 확인하는 데 유용하며, 이 값은 DBMS_SESSION.SET_IDENTIFIER 를 통해 설정 가능합니다.

ASH 뷰가 제공하는 정보의 유용성을 감안하면, AWR 과 마찬가지로 이 정보들을 영구적인 형태의 매체에 저장할 필요가 있을 수도 있습니다. AWR 테이블을 MMON 슬레이브를 통해 디스크로 flush 할 수 있으며, 이 경우 DBA_HIST_ACTIVE_SESS_HISTORY 뷰를 통해 저장된 결과를 확인할 수 있습니다.

1.2.8. 수작업으로 스냅샷 생성하기

스냅샷은 자동으로 수집되도록 디폴트 설정되어 있으며, 원하는 경우 온디맨드 형태의 실행이 가능합니다. 모든 AWR 기능은 DBMS_WORKLOAD_REPOSITORY 패키지에 구현되어 있습니다. 스냅샷을 실행하려면 아래와 같은 명령을 사용하면 됩니다.

```
SQL> SELECT sysdate FROM dual;
```

```
SYSDATE
```

```
-----  
2007/12/19 17:04:43      ← 현재 시간 확인
```

```
1 rows selected.
```

```
SQL> SELECT snap_id, begin_interval_time begin, end_interval_time end  
FROM SYS.DBA_HIST_SNAPSHOT;
```

SNAP_ID	BEGIN	END
15	2007/12/19 15:20:20.203	2007/12/19 15:26:18.484
17	2007/12/19 15:40:27.609	2007/12/19 16:00:29.781
18	2007/12/19 16:00:29.781	2007/12/19 16:20:50.640
10	2007/12/19 13:32:52.546	2007/12/19 14:00:51.546
11	2007/12/19 14:00:51.546	2007/12/19 14:20:58.562
12	2007/12/19 14:20:58.562	2007/12/19 14:40:10.468
14	2007/12/19 15:00:12.500	2007/12/19 15:20:20.203
16	2007/12/19 15:26:18.484	2007/12/19 15:40:27.609
19	2007/12/19 16:20:50.640	2007/12/19 16:40:56.546
13	2007/12/19 14:40:10.468	2007/12/19 15:00:12.500
20	2007/12/19 16:40:56.546	2007/12/19 16:55:39.968

```
11 rows selected.
```

```
SQL> execute dbms_workload_repository.create_snapshot;      ← 스냅샷 직접 생성(수동)
```

```
PL/SQL Executed.
```

```
SQL> select snap_id, begin_interval_time begin, end_interval_time end  
from SYS.DBA_HIST_SNAPSHOT;
```

SNAP_ID	BEGIN	END
15	2007/12/19 15:20:20.203	2007/12/19 15:26:18.484
17	2007/12/19 15:40:27.609	2007/12/19 16:00:29.781
18	2007/12/19 16:00:29.781	2007/12/19 16:20:50.640
10	2007/12/19 13:32:52.546	2007/12/19 14:00:51.546
11	2007/12/19 14:00:51.546	2007/12/19 14:20:58.562
12	2007/12/19 14:20:58.562	2007/12/19 14:40:10.468
14	2007/12/19 15:00:12.500	2007/12/19 15:20:20.203
16	2007/12/19 15:26:18.484	2007/12/19 15:40:27.609
19	2007/12/19 16:20:50.640	2007/12/19 16:40:56.546
13	2007/12/19 14:40:10.468	2007/12/19 15:00:12.500
21	2007/12/19 16:55:39.968	2007/12/19 17:05:22.890
20	2007/12/19 16:40:56.546	2007/12/19 16:55:39.968

← 새로 추가된 스냅샷(21)

```
12 rows selected.
```

위 명령은 스냅샷을 즉각적으로 실행하여 그 결과를 table WRM\$_SNAPSHOT 테이블에 저장합니다. 수집되는 메트릭의 수준은 TYPICAL 레벨로 설정됩니다. 더욱 자세한 통계를 원하는 경우 FLUSH_LEVEL 매개변수를 ALL 로 설정하면 됩니다. 수집된 통계는 자동으로 삭제되며, 수작업으로 삭제하려는 경우 drop_snapshot_range() 프로시저를 실행하면 됩니다.

```
SQL> SELECT snap_id, startup_time
       FROM dba_hist_snapshot
       ORDER BY 1,2;
```

SNAP_ID	STARTUP_TIME	
10	2007/12/19 10:27:32.000	← 삭제할 첫번째 스냅샷
11	2007/12/19 10:27:32.000	
12	2007/12/19 10:27:32.000	
13	2007/12/19 10:27:32.000	
14	2007/12/19 10:27:32.000	
15	2007/12/19 10:27:32.000	← 삭제할 마지막 스냅샷
16	2007/12/19 10:27:32.000	
17	2007/12/19 10:27:32.000	
18	2007/12/19 10:27:32.000	
19	2007/12/19 10:27:32.000	
20	2007/12/19 10:27:32.000	
21	2007/12/19 10:27:32.000	

12 rows selected.

```
SQL> exec dbms_workload_repository.drop_snapshot_range(10, 15); ← 삭제할 SNAP_ID 범위 지정
```

PL/SQL Executed.

```
SQL> SELECT snap_id, startup_time
       FROM dba_hist_snapshot
       ORDER BY 1,2;
```

SNAP_ID	STARTUP_TIME
16	2007/12/19 10:27:32.000
17	2007/12/19 10:27:32.000
18	2007/12/19 10:27:32.000
19	2007/12/19 10:27:32.000
20	2007/12/19 10:27:32.000
21	2007/12/19 10:27:32.000

6 rows selected.

1.2.9. 베이스라인

성능 튜닝 작업을 수행할 때에는 먼저 일련의 메트릭에 대한 베이스라인(baseline)을 수집하고 튜닝을 위한 변경 작업을 수행한 뒤, 다시 또 다른 베이스라인 셋을 수집하는 과정을 거치는 것이 일반적입니다. 이렇게 수집된 두 가지 셋을 서로 비교하여 변경 작업의 효과를 평가할 수 있습니다. AWR 에서는 기존에 수집된 스냅샷을 통해 이러한 작업이 가능합니다. 예를 들어 매우 많은 자원을 사용하는 test_baseline 라는 프로세스가 오후 3 시 40 분부터 4 시 40 분까지 실행되었고, 이 기간 동안 스냅샷 ID 17 에서 20 까지가 수집되었다고 합시다. 이 스냅샷들을 위해 test_baseline_1 이라는 이름의 베이스라인을 아래와 같이 정의할 수 있습니다.

```
SQL> SELECT snap_id, begin_interval_time begin, end_interval_time end
FROM SYS.DBA_HIST_SNAPSHOT;
```

SNAP_ID	BEGIN	END
17	2007/12/19 15:40:27.609	2007/12/19 16:00:29.781
18	2007/12/19 16:00:29.781	2007/12/19 16:20:50.640
16	2007/12/19 15:26:18.484	2007/12/19 15:40:27.609
19	2007/12/19 16:20:50.640	2007/12/19 16:40:56.546
22	2007/12/19 17:05:22.890	2007/12/19 17:20:06.281
21	2007/12/19 16:55:39.968	2007/12/19 17:05:22.890
20	2007/12/19 16:40:56.546	2007/12/19 16:55:39.968

7 rows selected.

```
SQL> SELECT dbid, baseline_id, baseline_name, start_snap_id, end_snap_id
FROM dba_hist_baseline;
```

DBID	BASELINE_ID	BASELINE_NAME	START_SNAP_ID	END_SNAP_ID
------	-------------	---------------	---------------	-------------

0 rows selected. → 현재 베이스라인이 생성되어 있지 않은 것을 확인.

```
SQL> exec dbms_workload_repository.create_baseline (17,20,'test_baseline_1');
```

→ 17 번부터 20 번까지의 'test_baseline_1' 베이스라인 생성

PL/SQL Executed.

```
SQL> SELECT dbid, baseline_id, baseline_name, start_snap_id, end_snap_id
FROM dba_hist_baseline;
```

DBID	BASELINE_ID	BASELINE_NAME	START_SNAP_ID	END_SNAP_ID
1169471561	4	test_baseline_1	17	20

1 rows selected. → 1 개의 베이스라인이 생성된 것을 확인할 수 있다.

```
SQL> exec dbms_workload_repository.drop_baseline ('test_baseline_1');
```

→ 'test_baseline_1'을 삭제함

PL/SQL Executed.

```
SQL> SELECT dbid, baseline_id, baseline_name, start_snap_id, end_snap_id
FROM dba_hist_baseline;
```

DBID	BASELINE_ID	BASELINE_NAME	START_SNAP_ID	END_SNAP_ID
------	-------------	---------------	---------------	-------------

0 rows selected. → 베이스라인이 삭제된 것을 확인

튜닝 과정을 거친 후, 또 다른 이름(예: apply_interest_2)의 베이스라인을 생성하여, 이 두 가지 베이스라인에 해당하는 스냅샷의 메트릭을 비교할 수 있습니다. 이처럼 비교 대상을 한정함으로써 성능 튜닝의 효과를 한층 향상시킬 수 있습니다. 분석이 끝나면 위의 예제와 같이 drop_baseline(); 프로시저로 베이스라인을 삭제할 수 있습니다 (이 때 스냅샷은 그대로 보존됩니다). 또 오래된 스냅샷이 삭제되는 과정에서, 베이스라인과 연결된 스냅샷은 삭제되지 않습니다.

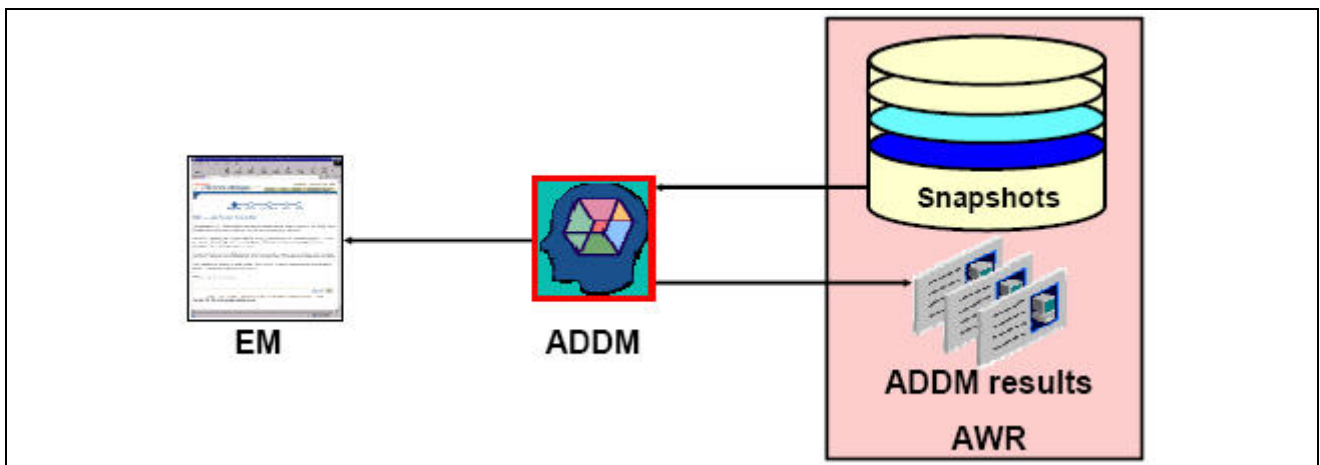
2. ADDM

2.1. ADDM (AUTOMATIC DATABASE DIAGNOSTIC MONITOR) 란?

이제 Oracle Database 가 직접 제공하는 SQL 튜닝 서비스를 활용해 보십시오! SQL Profile 를 이용하여 쿼리 성능을 향상시키고 ADDM 을 통해 일반적인 성능 문제를 쉽고 빠르게 해결하는 것이 가능합니다. DBA 는 반복적인 잡무에서 벗어나 좀 더 전략적인 과제에 집중할 수 있기를 원합니다. 그런 반복적인 업무를 대신 처리해주는 역할을 하는 것이 Oracle Database 10g 에 추가된 Automatic Database Diagnostic Monitor(ADDM) 입니다. ADDM 은 데이터베이스 성능 통계에 대한 철저한 분석을 통해 성능 병목을 확인하고, SQL 구문에 대한 분석을 통해 성능 향상을 위한 조언을 제공합니다. 또 SQL Tuning Advisor 와 연동된 기능을 제공하기도 합니다. 이번에는 ADDM 을 이용한 성능 향상 방안에 대해 설명합니다.

2.2. ADDM의 구성

2.2.1. ADDM Overview



AWR 은 데이터베이스로부터 상세한 성능 관련 지표를 주기적으로 수집하여 저장합니다. 스냅샷 생성 작업이 완료될 때마다, ADDM 이 호출되어 서로 다른 스냅샷의 데이터와 성능 지표를 비교 분석하고 성능 향상을 위한 조언을 제공합니다. 문제가 발견된 후, ADDM 은 다른 어드바이저 툴(SQL Tuning Advisor 등)을 호출하여 해결 방법을 찾아내기도 합니다.

2.2.2. ADDM 구성과 작동 원리

예를 통해 ADDM 의 기능을 설명해 보도록 하겠습니다. 원인이 확인되지 않은 성능 문제에 대한 진단 작업에 착수한 경우를 생각해봅시다. DBA 는 문제가 되는 SQL 구문이 무엇인지 확인해 둔 상태입니다. 그러나 실제 환경에서는 이와 같은 유용한 단서도 모를 경우도 있습니다.

10g 에서 진단 작업을 수행하는 과정에서, drill-down 분석(필요시 상세정보를 파악할 수 있게 해주는 방식)을 위해 적절한 시간 간격을 두고 생성된 스냅샷들을 선택할 수 있습니다. Enterprise Manager 10g 의 데이터베이스 홈 페이지에서는 아래와 같은 경로로 ADDM 을 실행합니다.

① 홈 (DB home)

홈 하단에 있는 관련 링크 중에 “중앙 권고자(advisor central)”를 클릭합니다.

홈

설계 관리 유지 관리

관련 링크

[SQL*Plus](#)
[정보 로그 내용](#)
[모니터링 단위](#)
[자문](#)
[특정 단위 기준 요소](#)

[SQL 기록](#)
[메모리 액세스 모드로 모니터](#)
[사용자 성능 측정 단위](#)
[중앙 권고자](#)
[특정 단위 구성 오류](#)

[정보 기록](#)
[모니터링 구성](#)
[임시 중단](#)
[특정 단위 관리](#)

② 중앙권고자 (advisor central)

“권고자(advisor)” 중에 ADDM 을 선택합니다.

데이터베이스 인스턴스: orcl > 중앙 권고자

SYSTEM(으)로 로그인됨

중앙 권고자

페이지가 새로 고쳐진 시간 2007.12.19. 오후 7시 48분 57초 KST

새로 고침

권고자

ADDM

MTTR 권고자

SQL 액세스 권고자

실행 취소 관리

메모리 권고자

세그먼트 권고자

권고자 작업

기본 매개변수 변경

검색

권고 유형을 선택하고 결과 집합에 표시되는 데이터를 필터링할 작업 이름을 선택적으로 입력하십시오.

권고 유형: 모든 유형, 작업 이름: , 권고자 실행: 마지막 실행, 상태: 모두

실행

기본적으로 검색은 사용자가 입력한 문자열로 시작되는 모든 대문자 일치 항목을 반환합니다. 정확하게 일치하는 검색을 실행하거나 대소문자를 구분하여 검색을 실행하려면 검색 문자열을 큰 따옴표로 묶으십시오. 큰 따옴표로 묶은 문자열에서 대체 문자 샘플(%)을 사용할 수 있습니다.

결과

결과 보기 삭제 작업 일정 변경 실행

선택	권고 유형	이름	설명	사용자	상태	시작 시간	기간 (초)	만기 시간 (일)
<input checked="" type="radio"/>	ADDM	ADDM:1169471561_1_36	ADDM auto run: snapshots [33, 36], instance 1, database id 1169471561	SYSTEM	COMPLETED	2007. 12. 19. 오후 7:46:06	0	30
<input type="radio"/>	SQL Access Advisor	SQACCESS3193334	SQL 액세스 권고자	SYSTEM	COMPLETED	2007. 12. 19. 오전 12:09:42	3	30

③ ADDM 실행 (기간 설정)

현재 인스턴스 성능을 분석할 것인지, 과거의 인스턴스 성능을 분석할지를 선택합니다.

과거 인스턴스 성능을 분석한다면 “기간 시작 시간(Period Start)”을 체크한 후 스냅샷(사진그림)을 원하는 시작 시간에 맞춰서 선택하고, “기간 종료 시간(Period End)”을 체크한 후 스냅샷 중 원하는 종료 시간에 맞춰서 선택합니다. 선택이 다 되었으면 “확인” 버튼을 눌러 ADDM 을 실행합니다.

데이터베이스 인스턴스: orcl > 중앙 권고자 > ADDM 실행

SYSTEM(으)로 로그인됨

ADDM 실행

취소 확인

☒ 현재 인스턴스 성능을 분석하려면 ADDM을 실행하십시오.
새 AWR 스냅샷을 생성하고 새 스냅샷 및 이전 스냅샷에 대해 ADDM을 실행합니다. 이 작업을 완료하려면 DBMS_WORKLOAD_REPOSITORY에 대한 EXECUTE 권한이 있어야 합니다.

☐ 과거의 인스턴스 성능을 분석하려면 ADDM을 실행하십시오.
ADDM 분석을 위해서는 분석 기간을 지정하는 한 쌍의 AWR 스냅샷이 필요합니다. [기간 시작 시간] 또는 [기간 종료 시간] 옵션을 선택하고 그래프 아래의 아이콘을 눌러 분석 기간을 지정하십시오.

☐ 기간 시작 시간 2007.12.19. 오후 4시 00분 29초 KST

☒ 기간 종료 시간 2007년 12월 19일 수요일 오후 7시 21분 05초 KST

용량 측정

4:00 4:15 4:30 4:45 5:00 5:15 5:30 5:45 6:00 6:15 6:30 6:45 7:00 7:15 7:30 7:45

2007. 12. 19.

대기 사용자 I/O CPU

확대/축소

참고 이 페이지에 사용되는 아이콘 및 기호에 대한 설명은 다음을 참조하십시오. 아이콘 키

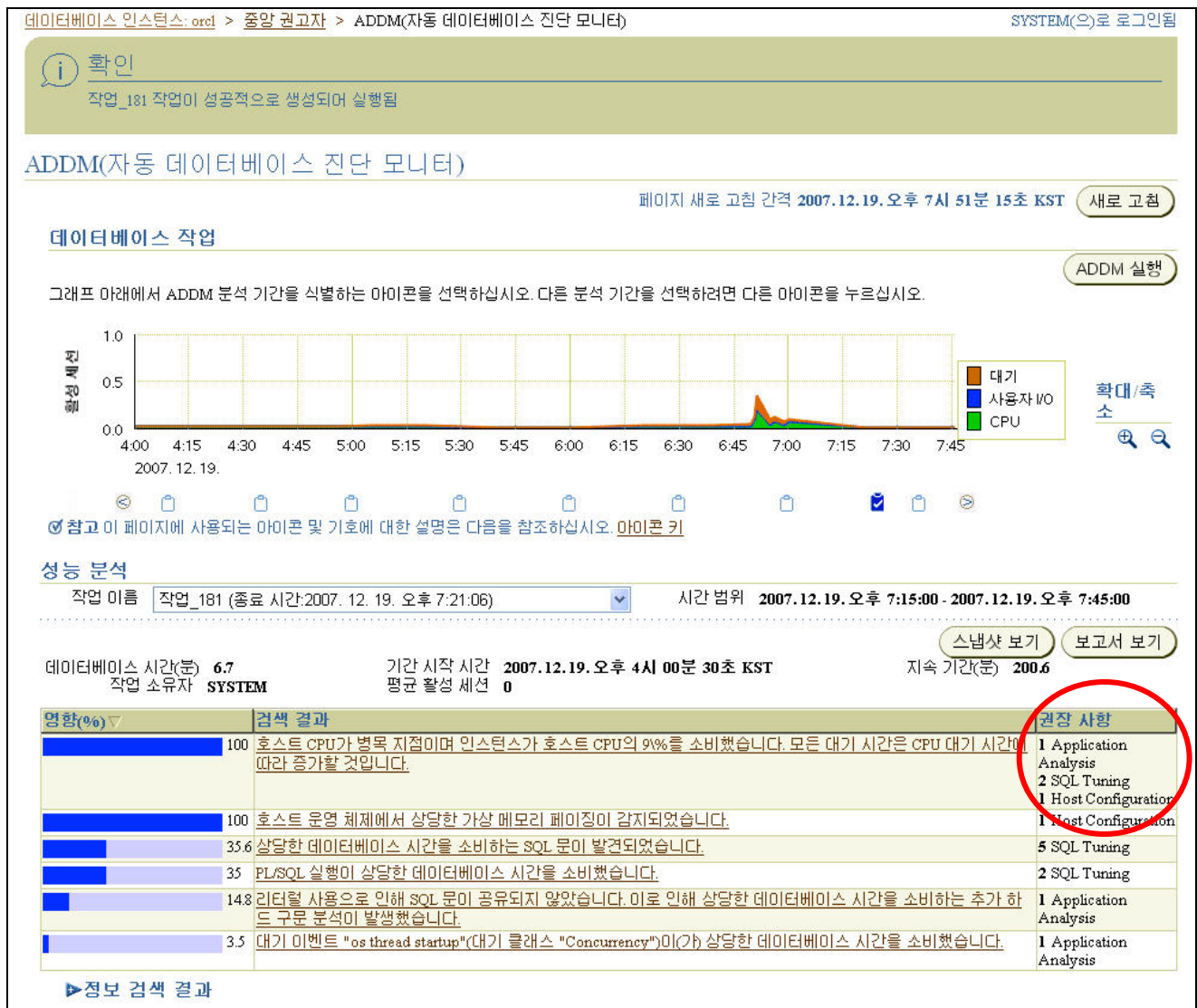
취소 확인

④ ADDM (자동 데이터베이스 진단 모니터)

아래 그림은 ADDM의 진단 모니터의 결과입니다.

12월 19일 오후 7시 15분부터 45분까지 30분동안 취합된 성능 분석 내용입니다.

ADDM은 총 6가지의 성능 문제점을 발견해냈습니다. 가장 많은 영향도를 보이고 있는 첫번째 리스트 항목에 따르면 분석 결과 현재 CPU의 병목현상으로 인해 CPU 대기 시간이 증가할 수 있으며, 가상 메모리 페이징이 일어나고 있어 이로 인해 Database의 성능이 전체적으로 저하되고 있는 것으로 나타났습니다. 발견된 내용을 근거로 ADDM은 "어플리케이션 분석, SQL 튜닝, 호스트 설정"을 수행할 것을 권고하고 있습니다.



⑤ 성능 검색 결과 세부 정보 01

더 상세한 내용을 보기 위해 첫번째 항목을 열어보면 “성능 검색 결과 세부 정보”가 나타나며, 아래 사진과 같은 “권장 사항”을 보여준다. 권장사항에서는 작업 리소스 우선순위를 변경할 것과, 호스트의 CPU 추가 등을 세부 정보로 보여주고 있다.

데이터베이스 인스턴스: orcl > 중앙 권고자 > ADDM(자동 데이터베이스 진단 모니터) > 성능 검색 결과 세부 정보 SYSTEM(으)로 로그인됨

성능 검색 결과 세부 정보

데이터베이스 시간(분) 6.7 기간 시작 시간 2007.12.19. 오후 4시 00분 30초 KST 지속 시간(분) 200.6
작업 소유자 SYSTEM 작업 이름 작업_181 평균 활성 세션 0

검색 결과 호스트 CPU가 병목 지점이며 인스턴스가 호스트 CPU의 91%를 소비했습니다. 모든 대기 시간은 CPU 대기 시간에 따라 증가할 것입니다.
영향(분) 6.7
영향(%) 100

권장 사항

모든 세부 정보 표시 | 모든 세부 정보 숨기기

세부 정보 범주

숨기기 Host Configuration 이익률(%) 100

작업 또한 오라클 데이터베이스 리소스 관리자를 사용하여 다양한 소비자 그룹에서 작업 로드 우선순위를 매기는 것을 고려하십시오.
작업 호스트에 CPU를 더 추가하거나 다른 호스트에서 데이터베이스를 제공하는 인스턴스의 수를 늘리는 것을 고려하십시오.

숨기기 SQL Tuning

이익률(%) 26.1

작업 PL/SQL 블록(SQL_ID "2b064ybzkwfly")을 튜닝합니다. Oracle "PL/SQL User's Guide and Reference"의 "Tuning PL/SQL Applications" 장을 참조하십시오.
SQL 텍스트 BEGIN EMD NOTIFICATION.QUEUE READY(1, 2, 3); END;
SQL ID 2b064ybzkwfly

근거 SQL_ID "2b064ybzkwfly"를(들) 가진 SQL 문이 391번 실행되었고 평균 경과 시간은 0.33초였습니다.
근거 실행당 사용된 평균 CPU는 0.26초였습니다.

숨기기 Application Analysis

이익률(%) 9.3

작업 SQL 문 구문 분석이 CPU를 많이 소비했습니다. 구문 분석과 관련한 자세한 내용은 이 작업의 기타 검색 결과를 참조하십시오.

숨기기 SQL Tuning

이익률(%) 5.7

작업 PL/SQL 블록(SQL_ID "044v83st298g1")을 튜닝합니다. Oracle "PL/SQL User's Guide and Reference"의 "Tuning PL/SQL Applications" 장을 참조하십시오.
SQL 텍스트 begin dbms workload repository.create_snapshot;end;
SQL ID 044v83st298g1

근거 SQL_ID "044v83st298g1"를(들) 가진 SQL 문이 5번 실행되었고 평균 경과 시간은 6.7초였습니다.
근거 실행당 사용된 평균 CPU는 4.5초였습니다.

⑥ 성능 검색 결과 세부 정보 02

아래 내용은 자동 데이터베이스 진단 결과 리스트의 3 번째인 “상당한 데이터베이스 시간을 소비하는 SQL 문이 발견되었습니다.”를 선택하여 들어온 화면이다. 총 35.6%의 영향도 중에 가장 영향도가 높은 5 개의 SQL 문을 보여주고 있다. ADDM 은 SQL 튜닝 권고자(SQL Tuning Advisor)를 이용해 이 SQL 구문에 대한 분석 작업을 수행할 것을 권고하고 있습니다. “지금 권고자 실행(Run Advisor Now)” 버튼을 클릭하면 SQL Tuning Advisor 가 호출되어 분석 작업을 시작합니다.

데이터베이스 인스턴스: orcl > 중앙 권고자 > ADDM(자동 데이터베이스 진단 모니터) > 성능 검색 결과 세부 정보 SYSTEM(으)로 로그인됨

성능 검색 결과 세부 정보

데이터베이스 시간(분) 6.7 기간 시작 시간 2007.12.19. 오후 4시 00분 30초 KST 지속 시간(분) 200.6
작업 소유자 SYSTEM 작업 이름 작업_181 평균 활성 세션 0

검색 결과 상당한 데이터베이스 시간을 소비하는 SQL 문이 발견되었습니다.
영향(분) 2.4
영향(%) 35.6

권장 사항

모두 선택 | 선택 취소 | 모든 세부 정보 표시 | 모든 세부 정보 숨기기

선택 세부 정보 범주

숨기기 SQL Tuning 이익률(%) 10

작업 SQL_ID "bunssq950snhf"의 SQL 문에 SQL 튜닝 권고자를 실행합니다. **지금 권고자 실행**
SQL 텍스트 insert into wrh\$ sga target advice (snap_id, dbid, instance number, SGA_size)
SQL ID bunssq950snhf

근거 SQL_ID "bunssq950snhf"를(들) 가진 SQL 문이 17번 실행되었고 평균 경과 시간은 3.1초였습니다.

표시	SQL Tuning	이익률(%)
<input type="checkbox"/>	SQL Tuning	8.2
<input type="checkbox"/>	SQL Tuning	6.3
<input checked="" type="checkbox"/>	SQL Tuning	4.1
<input type="checkbox"/>	SQL Tuning	3.6

검색 결과 경로

모두 확장 | 모두 축소


검색 결과 영향(%) 추가 정보

상당한 데이터베이스 시간을 소비하는 SQL 문이 발견되었습니다. 35.6

⑦ 성능 검색 결과 세부 정보 02 - SQL 튜닝 작업 생성중

위 리스트 중에서 첫번째 SQL 문에 대하여 “지금 권고자 실행(Run Advisor Now)”을 선택하면 아래와 같이 SQL 문에 대한 분석작업을 실행한다.

SQL 튜닝 권고자 작업이 제출되고 있습니다. 이 작업은 약간의 시간이 걸릴 수 있습니다. 이전 페이지로 돌아가려면 [취소]를 누르십시오. SQL 튜닝 권고자 작업이 계속 실행됩니다. [중앙 권고자] 페이지에서 상태를 확인하고 권장 사항을 볼 수 있습니다.


 새 SQL 튜닝 작업 생성
작업 실행

취소

⑧ 성능 검색 결과 세부 정보 02 - SQL ID 권장 사항

분석된 SQL 문에 대한 권장사항과 권장사항의 근거를 제시하고 있으며, “구현” 버튼을 선택하여 권장사항에 대하여 실행할 수 있다.

데이터베이스 인스턴스: orcl > 중앙 권고자 > SQL 튜닝 결과작업 182 > SQL ID 권장 사항:bunssq950snhf SYSTEM(으)로 로그인됨

SQL ID 권장 사항:bunssq950snhf

돌아가기

하나의 권장 사항만 구현되어야 합니다.

SQL 텍스트

```
insert into wrh$_sga_target_advice(snap_id,dbid,instance_number,SGA_SIZE,SGA_SIZE_FACTOR,ESTD_DB_TIME,ESTD_PHYSICAL_READS) select :snap_id,:dbid,instance_number,SGA_SIZE,SGA_SIZE_FACTOR,ESTD_DB_TIME,ESTD_PHYSICAL_READS from wrh$_sga_target_advice where snap_id=:snap_id and dbid=:dbid and instance_number=:instance_number
```

권장 사항 선택

원본 계획 설명

선택	유형	검색 결과	권장 사항	근거	이득률 (%)	새 계획명
<input checked="" type="radio"/>	통계	"SYS"."WRH\$_SGA_TARGET_ADVICE" 테이블 및 해당 인덱스에 대한 최적기 통계는 무효화되었습니다.	이 테이블에 대한 최적기 통계가 더 나은 실행 계획을 선택하기 위해서는 계를 수집하는 것을 고려하십시오.	더 나은 실행 계획을 선택하기 위해서는 최적기에 테이블에 대한 최신 통계가 필요합니다.		
<input type="radio"/>	기타	최적기는 실행 계획의 행 ID 1에서 뷰를 병합할 수 없었습니다.				

구현

돌아가기

⑨ 성능 검색 결과 세부 정보 02 - 권장 사항 구현 : 통계 수집

통계정보 수집을 위한 화면이 나타나면 “확인” 버튼을 선택하여 관련 스키마의 통계정보를 새롭게 생성할 수 있다.

데이터베이스 인스턴스: orcl > 중앙 권고자 > SQL 튜닝 결과작업 182 > SQL ID 권장 사항:bunssq950snhf > 권장 사항 구현:통계 수집 SYSTEM(으)로 로그인됨

권장 사항 구현: 통계 수집

취소 SQL 표시 확인

테이블 및/또는 인덱스에 대한 통계를 수집하기 위한 작업이 생성됩니다.

*작업 이름: GATHER_STATS_bunssq950snhf

설명:

SQL ID: bunssq950snhf

객체 목록

객체 이름: "SYS"."WRH\$_SGA_TARGET_ADVICE" 객체 유형: TABLE

일정

시간대: GMT +9:00

☒ 즉시

☐ 나중에

날짜: 2007. 12. 19. (예: 2007. 12. 19.)

시간: 7:58:00 ☐ 오전 ☒ 오후

취소 SQL 표시 확인

개별 웹 페이지 별로 권고 사항을 제시하는 것과 별도로, ADDM 은 전체 분석 결과에 대한 텍스트 리포트를 생성합니다. 텍스트 리포트는 문제가 되는 SQL 구문과 그 hash value 등의 상세한 정보를 제공합니다. 또 텍스트 리포트의 SQL ID 정보를 이용하여 SQL Tuning Advisor 또는 커맨드 라인을 통한 분석 작업을 수행할 수 있습니다.

ADDM 은 AWR 스냅샷이 생성될 때마다 호출되고, 가장 최근의 스냅샷과의 비교를 통해 권고사항을 제시합니다. 따라서 비교해야 할 두 스냅샷이 서로 인접한 경우에는 (이미 보고서가 생성되어 있으므로) 별도로 ADDM 태스크를 실행할 필요가 없으며, 스냅샷이 인접해 있지 않은 경우에만 ADDM 태스크를 실행할 필요가 있습니다.

ADDM 의 기능이 단순히 SQL 구문의 분석에 한정되지 않는다는 사실을 명심하시기 바랍니다. ADDM 은 메모리 관리, 세그먼트 관리, REDO/UNDO 등의 영역에 대한 다양한 분석 기능을 제공합니다.

2.2.3. SQL Tuning Advisor를 이용한 Access 분석

오라클 데이터베이스의 옵티마이저(optimizer)는 가능한 액세스 경로를 여럿 생성한 뒤, 오브젝트 통계정보를 기준으로 가장 적은 비용이 드는 하나를 선택하는 방식으로 runtime optimization 을 수행합니다. 하지만 옵티마이저는 (시간의 제약을 받는 만큼) SQL 구문의 튜닝이 필요한지, 통계가 정확한지, 새로운 인덱스를 생성해야 하는지 등의 여부를 판단하지 않습니다. 반면 SQL Tuning Advisor 는 일종의 “전문가 시스템”과 같은 역할을 합니다. 옵티마이저가 “현재 가능한 대안 중 최적의 결과를 얻을 수 있는 것은 무엇인가?”라는 질문에 대한 답변을 제공한다면, SQL Tuning Advisor 는 “사용자의 요구사항을 기반으로 고려했을 때, 성능을 향상시키기 위해 할 수 있는 일이 무엇인가?”라는 질문의 답을 제공합니다.

이러한 “전문가 시스템”으로서의 작업은 CPU 등의 자원을 많이 소모합니다. 이러한 이유로 SQL Tuning Advisor 는 데이터베이스가 Tuning Mode 로 설정된 경우에만 SQL 구문에 대한 분석작업을 수행합니다. Tuning Mode 는 튜닝 태스크를 생성하는 과정에서 SCOPE 및 TIME 매개변수를 설정함으로써 지정됩니다. 사용자에게 대한 영향을 최소화하려면 데이터베이스 활동이 적은 시간대를 선택하여 Tuning Mode 를 사용하는 것이 바람직합니다.

2.2.4. SQL Tuning Advisor를 이용한 SQL 튜닝 예시

아래 내용은 scott 유저의 간단한 SQL 문을 SQL Tuning Advisor 를 이용하여 튜닝하는 과정입니다.

```
SQL> CONN sys/<password> AS SYSDBA
SQL> GRANT ADVISOR TO scott;      ← scott 유저에게 advisor 권한 부여
SQL> CONN scott/tiger
SQL> SELECT ename from emp e
      WHERE job = 'SALESMAN'
      AND comm not in (SELECT comm FROM emp
                       WHERE ename=e.ename AND comm is not null);  ← 문제의 쿼리
선택된 레코드가 없습니다.
SQL> DECLARE
  2  l_task_id varchar2(30);
  3  l_sql varchar2(2000);
  4  BEGIN
  5  l_sql := 'select ename from emp e where job = :job and comm not in
  6          (select comm from emp where ename=e.ename and comm is not null)';
```

```

7  l_task_id := dbms_sqltune.create_tuning_task (          ← dbms_sqltune 패키지를 사용.
8  sql_text => l_sql,
9  user_name => 'SCOTT',          ← [주의] 유저명은 대문자로 해야함.
10 scope => 'COMPREHENSIVE',
11 time_limit => 120,
12 task_name => 'sql_advisor_test14'      ← 테스트 이름
13 );
14 dbms_sqltune.execute_tuning_task ('sql_advisor_test14');    ← 튜닝 테스트 실행
15 END;
16 /

```

PL/SQL 처리가 정상적으로 완료되었습니다.

SQL> set serveroutput on size 999999

SQL> set long 999999

SQL> select dbms_sqltune.report_tuning_task ('sql_advisor_test14') from dual; ← 생성된 테스트 보고서 보기

RECOMMENDATIONS

GENERAL INFORMATION SECTION

```

Tuning Task Name           : sql_advisor_test14
Tuning Task Owner          : SCOTT
Scope                      : COMPREHENSIVE
Time Limit(seconds)       : 120
Completion Status          : COMPLETED
Started at                 : 12/20/2007 18:00:56
Completed at               : 12/20/2007 18:00:56
Number of Statistic Findings : 1
Number of Index Findings   : 1
Number of SQL Restructure Findings : 1

```

Schema Name: SCOTT

SQL ID : 2m1h0au4m4kak

SQL Text : select ename from emp e where job = :job and comm not in
(select comm from emp where ename=e.ename and comm is not null)

RECOMMENDATIONS

FINDINGS SECTION (3 findings) ← 권고사항이 3가지가 있음

1- Statistics Finding ← 권고사항 첫번째 : 통계를 최적으로 만들어라.

"SCOTT"."EMP" 테이블 및 해당 인덱스는 분

Recommendation

-이 테이블 및 해당 인덱스에 대한 최적기 통계

```
execute dbms_stats.gather_table_stats(ownname => 'SCOTT', tablename =>
'EMP', estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE,
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);
```

Rationale

더 나은 실행 계획을 선택하기 위해서는 최적기에 테이블 및 해당

2- Index Finding (see explain plans section below)

← 권고사항 두번째 : 인덱스를 생성하라

이 명령문의 실행 계획은 하나 이상의 인덱스를

RECOMMENDATIONS

Recommendation (estimated benefit: 100%)

-Access Advisor 를 실행하여 물리적 스키마 설계를 향상시키거나 권장

```
create index SCOTT.IDX$$_008E0001 on SCOTT.EMP('ENAME');
```

-Access Advisor 를 실행하여 물리적 스키마 설계를 향상시키거나 권장

```
create index SCOTT.IDX$$_008E0002 on SCOTT.EMP('JOB');
```

----- 후락 -----

예로 든 구문 자체가 단순한 만큼, 굳이 어드바이저를 이용해서 이런 결과를 얻을 필요는 없었을 것입니다. 하지만 보다 복잡한 형태의 쿼리에 성능 문제가 발생한 경우라면, 어드바이저를 통해 수작업으로는 불가능한 성능 개선 효과를 얻을 수 있습니다.

어드바이저는 근본적인 원인을 통해 권장 내용의 근거를 설명하고, 그 결정을 DBA 의 판단에 맡깁니다.

2.2.5. ADDM과 SQL Tuning Advisor의 활용

SQL Tuning Advisor 는 쿼리에서 사용하는 오브젝트에 통계가 누락되어 있는지의 여부를 확인해 줍니다. 어드바이저는 다음과 같은 네 가지 유형의 작업을 수행합니다.

- 최적화를 위한 오브젝트들의 유효하고, 사용 가능한 통계정보를 보유하고 있는지 확인합니다.
- 성능 향상을 위해 쿼리를 재작성하는 방안을 권고합니다.
- 접근 경로를 확인하고 인덱스, MV(materialized view) 등을 추가하여 성능을 향상시킬 수 있는 방법을 조사하고 제안합니다.
- SQL Profile 을 생성하고, 이를 특정 쿼리에 연결합니다.

ADDM 과 SQL Tuning Advisor 가 유용하게 활용되는 경우를, 다음과 같은 세 가지 시나리오를 기준으로 검토해 볼 수 있습니다.

- **사후조치적 튜닝(Reactive Tuning):** 애플리케이션의 성능이 갑자기 저하됩니다. DBA 는 ADDM 을 사용하여 문제가 되는 SQL 구문을 확인합니다. ADDM 의 권고사항에 따라 SQL Tuning Advisor 를 실행하고 문제를 해결합니다.
- **사전예방적 튜닝(Proactive Tuning):** 애플리케이션은 정상적으로 동작합니다. DBA 는 유지보수를 위해 필요한 작업을 실행하고 쿼리를 좀 더 개선할 방법이 있는지 확인하고자 합니다. DBA 는 스탠드얼론 모드로 SQL Tuning Advisor 를 실행하고 가능한 대안을 검토합니다.
- **개발단계 튜닝(Development Tuning):** QA 단계 또는 운영 단계보다는, 개발 단계에서 쿼리 튜닝을 수행하는 것이 상대적으로 용이합니다. 이때 어드바이저의 커맨드 라인 버전을 사용하여 개별 SQL 구문을 튜닝할 수 있습니다.

2.2.6. Enterprise Manager의 활용

앞의 예는 SQL Tuning Advisor 를 커맨드 라인 모드에서 사용하는 방법을 기준으로 설명되었습니다. 커맨드 라인 모드는 태스크 실행을 위한 스크립트 생성이 용이하다는 장점이 있습니다. 하지만, 사용자에게 의해 이미 문제가 보고된 경우라면 Enterprise Manager 10g 를 이용하는 방법이 더 효과적일 수 있습니다.

이번에는 EM 을 활용하여 SQL 구문을 진단하고 튜닝하는 방법을 설명해 보겠습니다. 데이터베이스 홈 페이지 하단의 “Advisor Central” 링크를 클릭하면 다양한 어드바이저를 위한 메뉴를 제공하는 화면이 표시됩니다. 이 화면 상단의 “SQL Tuning Advisor”를 클릭합니다.

① 홈 (DB home)

홈 하단 관련 링크에 있는 “중앙 권고자”를 선택한다.

홈 | 설정 관리 | 유지 관리

관련 링크

- SQL*Plus
- 경보 로그 내용
- 모든 측정 단위
- 작업
- 측정 단위 기준 요소
- SQL 기록
- 메모리 액세스 모드로 모니터
- 시작사 성능 측정 단위
- 중앙 권고자**
- 측정 단위 수집 오류
- 경보 기록
- 모니터링 구성
- 일시 중단
- 측정 단위 관리

② 중앙권고자 (Advisor Central)

권고자 중 “SQL 튜닝 권고자”를 선택한다.

데이터베이스 인스턴스: orcl > 중앙 권고자 SYSTEM(으)로 로그인됨

중앙 권고자 페이지가 새로 고쳐진 시간 2007. 12. 20. 오전 11시 03분 32초 KST 새로 고침

권고자

- ADDM** (circled in red)
- SQL 튜닝 권고자
- 실행 취소 관리
- MTTR 권고자
- 메모리 권고자
- SQL 액세스 권고자
- 세그먼트 권고자

권고자 작업 기본 매개변수 변경

검색

권고 유형을 선택하고 결과 집합에 표시되는 데이터에 필터링할 작업 이름을 선택적으로 입력하십시오.

권고 유형: 작업 이름: 권고자 실행 상태:

모든 유형 마지막 실행 모두 실행

기본적으로 검색은 사용자가 입력한 문자열로 시작되는 모든 대문자 일치 항목을 반환합니다. 정확하게 일치하는 검색을 실행하거나 대소문자를 구분하여 검색을 실행하려면 검색 문자열을 큰 따옴표로 묶으십시오. 큰 따옴표로 묶은 문자열에서 대체 문자 심볼(%)을 사용할 수 있습니다.

결과

선택	권고 유형	이름	설명	사용자	상태	시작 시간	기간 (초)	만기 시간(일)
<input checked="" type="radio"/>	ADDM	ADDM:1169471561_1_76	ADDM auto run: snapshots [75, 76], instance 1, database id 1169471561	SYS	COMPLETED	2007. 12. 20. 오전 11:01:07	7	30
<input type="radio"/>	SQL Tuning Advisor	SQL_TUNING_1198115592609		SYSTEM	COMPLETED	2007. 12. 20. 오전 10:53:35	569	30
<input type="radio"/>	Segment Advisor	SYS_AUTO_SPCADV_401319122007	Auto Space Advisor	SYS	COMPLETED	2007. 12. 19. 오후 10:00:05	2	29
<input type="radio"/>	SQL Access Advisor	SQLACCESS3193334	SQL 액세스 권고자	SYSTEM	COMPLETED	2007. 12. 19. 오전 12:09:42	3	29

③ SQL 튜닝 권고자 링크 (SQL Tuning Advisor Links)

“최상위 작업(Top SQL)”을 선택한다.

데이터베이스 인스턴스: [orcl](#) > [중앙 권고자](#) > [SQL 튜닝 권고자 링크](#) SYSTEM(으)로 로그인됨

SQL 튜닝 권고자 링크

SQL 튜닝 권고자는 개별 SQL 문을 분석하며 SQL 문 성능을 향상시키기 위한 권장 사항을 만듭니다. 실행 가능한 위치는 다음과 같습니다. 다음 소스 중 하나를 누를 수 있습니다. 그러면 SQL 튜닝 권고자를 사용하여 SQL 문을 튜닝할 수 있는 데이터 소스로 이동됩니다.

최상위 작업

가장 SQL
SQL 튜닝 집합
스냅샷
보관된 스냅샷 집합

④ 최상위 작업 (Top SQL)

아래 보여지는 화면과 같이 여러 개의 그래프를 통해 다양한 wait class 에 관련된 정보를 시간대 별로 확인할 수 있습니다. 그림 상단에 있는 활성세션 그래프 중 관심구간은 회색 사각형으로 표시됩니다. 관심영역의 시간대를 변경하려면 그래프 하단에 있는 “회색바”를 드래그 하여 관심구간을 조정할 수 있습니다. 관심구간이 조정될 때마다 그림 하단에 있는 최상위 SQL 리스트가 새롭게 조정되며, 최상위 SQL 리스트는 SQL ID(붉은색 부분)를 클릭하여 해당 SQL 문의 상세정보를 알 수 있다.

데이터베이스 인스턴스: [orcl](#) > [최상위 작업](#) SYSTEM(으)로 로그인됨

최상위 작업

아래의 세부 정보 섹션에 대한 기간을 변경하려면 차트 아래의 밴드를 누르십시오.

데이터 보기 | 실시간 | 수동 새로 고침

선택한 5분 간격에 대한 세부 정보

시작 시간 2007.12.20. 오후 11시 13분 21초 KST

ASH 보고서 실행

최상위 SQL

SQL 튜닝 권고자 일정 잡기 | SQL 튜닝 집합 생성

모두 선택 | 선택 취소

선택 작업(%)	SQL ID	SQL 유형
<input type="checkbox"/> 21.43	2b0644bdc9f71v	PL/SQL EXECUTE
<input type="checkbox"/> 21.43	3dh6wrvufq41uk	SELECT
<input type="checkbox"/> 7.14	b07usvw8cn92f	SELECT
<input type="checkbox"/> 7.14	3f8tyza722jdb	SELECT
<input type="checkbox"/> 7.14	7nz5cmj5wmbm	SELECT
<input type="checkbox"/> 7.14	4ckzywfw81bk	SELECT
<input type="checkbox"/> 7.14	6bq37x2md4r	SELECT
<input type="checkbox"/> 7.14	9kzwsazsktuzv	SELECT
<input type="checkbox"/> 7.14	cp5caas42udnw	SELECT
<input type="checkbox"/> 7.14	ck2wqjm5p3h0p	PL/SQL EXECUTE

SQL 튜닝 권고자 일정 잡기 | SQL 튜닝 집합 생성

총 샘플 수: 14

최상위 세션

보기 | 최상위 세션

작업(%)	세션 ID	사용자 이름	프로그램
17.65	134	SYSMAN	OMS
17.65	143	SYSMAN	OMS
17.65	140	SYSTEM	sqlplus.exe
11.76	150	SYSTEM	ORACLE.EXE (J000)
11.76	145	DESNMP	OMS
5.88	139	DESNMP	emagent.exe
5.88	141	DESNMP	emagent.exe
5.88	165	SYS	ORACLE.EXE (CKPT)
5.88	156	SYSTEM	OMS

총 샘플 수: 17

⑤ SQL 세부 정보 (SQL Details)

SQL 세부 정보에서는 텍스트 형식으로 SQL 문의 내용을 보여주며, 활성세션 그래프가 해당 SQL 구문의 소모 자원들을 그래프로 보여준다. 세부정보의 활성세션 그래프에서도 그래프 하단의 “흰색바”를 드래그 하여 관심구간을 선택할 수 있다. 관심구간을 선택하였다면 “SQL 튜닝 권고자 일정 잡기(Run SQL Tuning Advisor)” 버튼을 선택하여 Advisor 를 실행한다.

데이터베이스 인스턴스: orcl > 최상위 작업 > SQL 세부 정보: 3dh6wvufq41uk SYSTEM(으)로 로그인됨

SQL 세부 정보: 3dh6wvufq41uk

SQL ID로 전환 실행 데이터 보기 실시간: 수동 새로 고침 새로 고침 SQL 튜닝 권고자 일정 잡기

텍스트

```
select distinct dname from scott.dept d, scott.emp e where d.deptno = e.deptno and e.sal > 2000
```

세부 정보

아래 세부 정보를 보려면 계획 해시 값을 선택하십시오. 계획 해시 값: 652980069

통계 작업 계획 튜닝 정보

요약

아래 세부 정보 섹션에 대한 기간을 변경하려면 음영 상자를 끌어 놓으십시오.

선택한 5분 간격에 대한 세부 정보

시작 시간 2007.12.20. 오후 11:13:21

작업(%)	SID	사용자	프로그램	서비스	계획 해시 값
100.00	140	SYSTEM	sqlplus.exe	SYS\$USERS	652980069

⑥ 권고자 일정 잡기 (Schedual Advisor)

권고자 일정 잡기 화면에서, 태스크의 유형과 분석의 심도를 결정할 수 있습니다. 아래 화면에서는 “포괄적모드(comprehensive)” analysis 를 선택하고 어드바이저를 바로 실행(immediately)하도록 설정하였습니다.

데이터베이스 인스턴스: orcl > 권고자 일정 잡기 SYSTEM(으)로 로그인됨

권고자 일정 잡기

권고자 실행 시작 날짜와 시간을 입력하십시오. 데이터베이스 작업이 입력한 시간에 제출됩니다. 또한 권고자 실행 시간을 제한할 수도 있습니다. 이 제한 시간에 도달하면, 권고자 실행이 중단되며 결과 중 일부만 반환됩니다. 중앙 권고자를 통해 권고자 실행 상태를 확인할 수 있습니다.

* 이름: SQL_TUNING_1198160404496
설명:

SQL 문

SQL 텍스트 스키마 구분 분석 중

```
select distinct dname from scott.dept d, scott.emp e where d.deptno = e.deptno and e.sal > 2000
```

범위

☐ 제한 모드입니다. SQL 프로파일 권장 사항 없이 분석이 수행됩니다. 한 개의 명령문을 처리하는 데 1초 정도 소요됩니다.

☒ 포괄적 모드입니다. SQL 프로파일을 포함한 분석이 수행됩니다. 상당한 시간이 소요될 수 있습니다.

동 시간 제한(분): 30

일정

시간대: GMT +9:00

☒ 즉시 ☐ 나중에

날짜: 2007. 12. 20. (해: 2007. 12. 20.)


시간: 11:20:00 ☐ 오전 ☒ 오후

⑦ SQL 권고자 실행 중

권고자의 분석범위와 심도에 따라서 분석시간이 오래 걸리기도 한다.

SYSTEM(으)로 로그인됨

SQL 튜닝 권고자 작업이 제출되고 있습니다. 이 작업은 약간의 시간이 걸릴 수 있습니다. 이전 페이지로 돌아가려면 [취소]를 누르십시오. SQL 튜닝 권고자 작업이 계속 실행됩니다. [중앙 권고자] 페이지에서 상태를 확인하고 권장 사항을 볼 수 있습니다.



➡ 새 SQL 튜닝 작업 생성
작업 실행

⑧ 권장 사항

어드바이저의 작업 실행이 완료되면 화면을 통해 어드바이저의 권고 사항을 확인할 수 있습니다.

권장 사항으로 2 개의 인덱스[scott.emp("deptno"), scott.emp("sal")]를 생성할 것을 권고하고 있다.

SYSTEM(으)로 로그인됨

데이터베이스 인스턴스: orcl > 중앙 권고자 > SQL 튜닝 결과:SQL_TUNING_1198160115465 > SQL ID 권장 사항:3dh6wvufq41uk

SQL ID 권장 사항:3dh6wvufq41uk

돌아가기


하나의 권장 사항만 구현되어야 합니다.

SQL 텍스트

```
select distinct dname from scott.dept d, scott.emp e where d.deptno = e.deptno and e.sal > 2000
```

권장 사항 선택

원본 계획 설명

선택	유형	검색 결과	권장 사항	근거	마지막 실행 (%)	새 계획 설명
<input checked="" type="radio"/>	인덱스	이 명령문의 실행 계획은 하나 이상의 인덱스를 생성하여 향상시킬 수 있습니다.	Access Advisor를 실행하여 물리적 스키마 설계를 향상시키거나 권장 인덱스를 생성하는 것을 고려하십시오. SCOTT EMP("DEPTNO") SCOTT EMP("SAL")	권장 인덱스를 생성하면 이 명령문의 실행 계획이 크게 향상됩니다. 하지만 단일 명령문 대신 대표 SQL 작업 로드를 사용하는 "Access Advisor"를 실행하는 것이 더 좋을 수 있습니다. 이렇게 하면 인덱스 유지 관리 오버헤드 및 추가 공간 사용을 고려한 포괄적인 인덱스 권장 사항을 얻을 수 있게 됩니다.	100.0	

⑨ 튜닝 결과

여러 권장 사항들 중에 인덱스 생성란에 체크표시가 되어있다.

SYSTEM(으)로 로그인됨

데이터베이스 인스턴스: orcl > 중앙 권고자 > SQL 튜닝 결과:SQL_TUNING_1198160404496

SQL 튜닝 결과:SQL_TUNING_1198160404496

페이지가 새로 고쳐진 시간 2007.12.20. 오후 11:21:52 새로 고침

상태 **COMPLETED**
SQL ID **3dh6wvufq41uk**
시간 제한(초) **1800**

시작됨 **2007.12.20. 오후 11:20:25**
완료됨 **2007.12.20. 오후 11:20:25**
실행 시간(초) **0**

권장 사항

권장 사항 보기

선택	SQL 텍스트	스키마 구분 분석 중	SQL ID	통계	SQL 프로파일	인덱스	SQL 구조 조정	기타	오류
<input checked="" type="radio"/>	select distinct dname from scott.dept d, scott.emp e where d.deptno = e.deptno and e.sal > 2000		3dh6wvufq41uk			<input checked="" type="checkbox"/>			

⑩ 인덱스 생성

DBA 가 인덱스가 필요하다고 판단하였다면 권장사항 페이지의 “구현” 버튼을 선택하여, 바로 인덱스 생성을 할 수 있다.

데이터베이스 인스턴스: orcl > 중앙 권고자 > SQL 튜닝 결과SQL TUNING 1198160404496 > SQL ID 권장 사항:3dh6wvufq41uk >

권장 사항 구현: 인덱스 생성

SYSTEM(으)로 로그인됨

취소

SQL 표시

확인

다음 새 인덱스를 생성하기 위한 작업이 생성됩니다.

* 작업 이름 CREATE_INDEXES_3dh6wvufq41uk

설명

SQL ID 3dh6wvufq41uk

새 인덱스

인덱스 이름	인덱스 유형	테이블 이름	인덱스 열
SCOTT.IDX\$\$\$_020A0001	BTREE	SCOTT.EMP	"DEPTNO"
SCOTT.IDX\$\$\$_020A0002	BTREE	SCOTT.EMP	"SAL"

일정

시간대 GMT +9:00

☒ 즉시

☐ 나중에

날짜 2007. 12. 20.
(예: 2007. 12. 20.)

시간 11 21 00 ☐ 오전 ☒ 오후

OEM 을 이용한 SQL Tuning Advisor 실행 방법은, 앞에서 설명한 커맨드 라인 버전을 이용하는 경우와 유사합니다. 하지만, 발생한 문제를 확인하여 드릴다운을 수행하고, 권고사항을 생성한 후 이를 승인하는 전체 과정이, 실제 발생한 문제를 해결하기에 편리한 형태로 구성되어 있다는 점에서 차이가 있습니다.

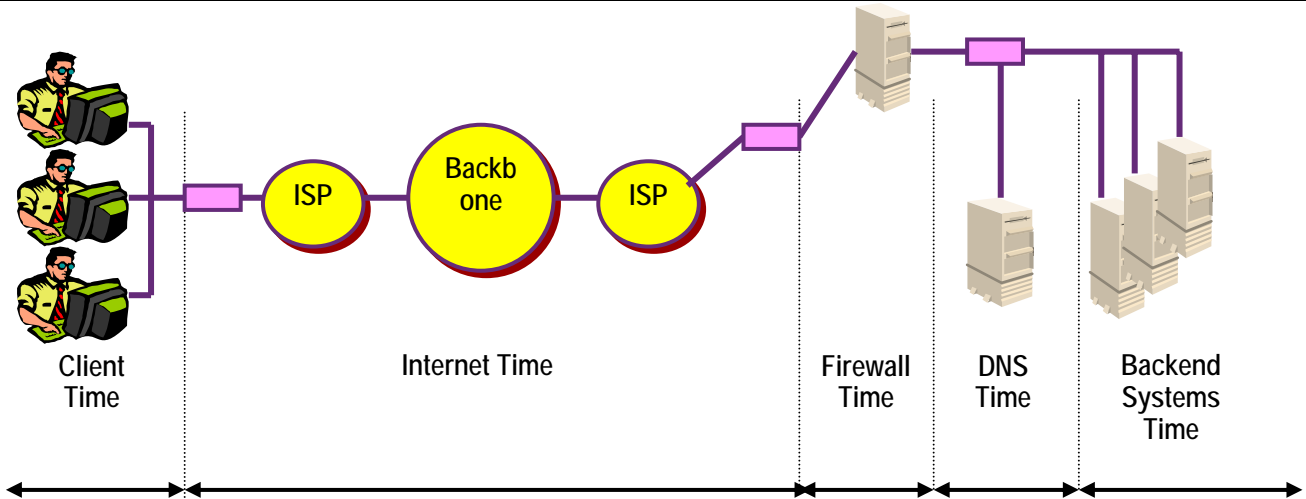
2.2.7. 결론

ADDM 은 성능 지표를 자동적으로 분석하고 오라클 전문가가 구현한 베스트 프랙티스 및 방법론을 기반으로 권고사항을 제시해 주는 강력한 성능 관리 툴입니다. 이 기능을 활용하여 발생한 문제와 그 원인을 확인할 수 있을 뿐 아니라, 취해야 할 조치에 대한 조언을 얻을 수 있습니다.

3. Pro-Active Tuning Service

3.1. 실제 사용자(End-User) 관점의 응답시간 튜닝

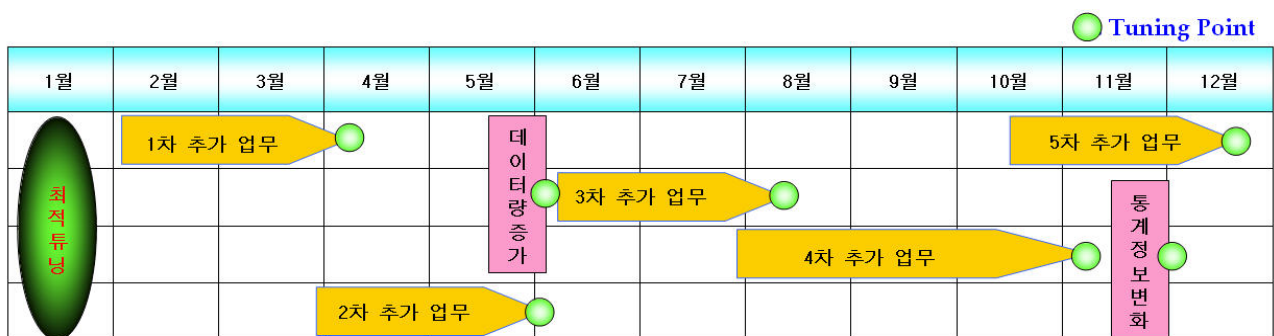
Pro-active tuning service 는 사용자 관점의 모니터링 및 분석을 통하여 실제 End-User 가 느끼는 응답시간(Response Time)을 튜닝합니다. APM(Application Performance Management) 툴을 이용하여 End-User 의 Request 결과를 반환받기까지의 모든 구간(Client PC, Internet 구간, FireWall, DNS, Web Server, WAS, DBMS) 을 분석하여 가장 Delay Time 이 많이 소요된 구간을 찾아 냅니다.



3.2. 최상의 성능 상태로 비즈니스 고가용성을 유지

Pro-Active Tuning Service

- 매 업무 단위 프로젝트 마다 참여하여 업무 적용(Open) 前 문제 요소를 분석하여 튜닝.
- 단위 업무 적용(Open) 후 매 3개월(데이터량 갱신 주기) 마다 튜닝 포인트를 설정, 성능 둔화 요소를 해결.
- 전사적으로 새롭게 추가되는 업무 단위 프로젝트의 모든 SQL 쿼리를 검토 및 튜닝.
- 다양한 대용량 데이터베이스 관리/튜닝 기법을 도입하여 최적의 DB 상태를 1년 내내 상시 유지.
- 전략적 튜닝 Factor 를 분석, 투자 대비 효율이 높은 Targeting 기법 적용. (비중도 높은 SQL 을 튜닝함)



[Pro-Active Tuning Service Schedule]

Performance Drop Point 마다 적절한 튜닝을 실시함으로써 항상 최적의 성능을 유지한다. !!

3.3. Knowledge Transfer

Pro-Active Tuning Service 는 고객의 Business Process 를 이해하고 시스템을 분석한 후 튜닝하는 것으로 완료되지 않습니다. 실제로 고객사 환경에서 튜닝한 내용을 그대로 실무자들에게 전수하여 내부 임직원의 역량을 제고시킵니다. 또한, Oracle RDBMS 신 버전의 New Features 를 교육함으로써, 이용자(관리자 및 개발자)가 스스로 개발 업무의 효율 및 생산성을 향상시킬 수 있도록 지원합니다. 이외에도 DBMS 관리자를 위한 관리 노하우(고급 Trouble-Shooting, 대용량 DB 처리, 병렬 처리 등)를 전수함으로써, 최상의 시스템을 최고의 기술로 유지할 수 있도록 지원합니다.

UAS (User Adapted Seminar) 진행 사례 및 내용 (Contents)

● 개발자를 위한 SQL 튜닝 실무 사례 세미나

G 쇼핑몰 업체 튜닝 후 실제 고객사의 튜닝 사례를 개발자들에게 전수하여 개발자들이 성능을 고려한 SQL 을 작성할 수 있도록 내부 역량을 제고시킴.

● Oracle 10g New Features 세미나

S Global 전자 기업 : Oracle 10g 버전으로 업그레이드 하기 전, 신 버전의 새로운 기능과 주의 사항을 전파함으로써, 업그레이드 후 발생할 수 있는 문제점의 사전 제거와 개발자들이 새로운 기능을 이용함으로써, 개발 생산성을 향상시킴.

● K 국가기관 DBMS 관리 노하우 세미나

내부 관리자 (DBA,SE)들을 대상으로 DBMS 관리자들이 흔히 겪을 수 있는 상황에 대한 Administration Know-How 와 고급 Trouble-Shooting 사례를 소개함으로써, 관리 기술력을 향상시킴.

3.4. Tuning 범위 확대

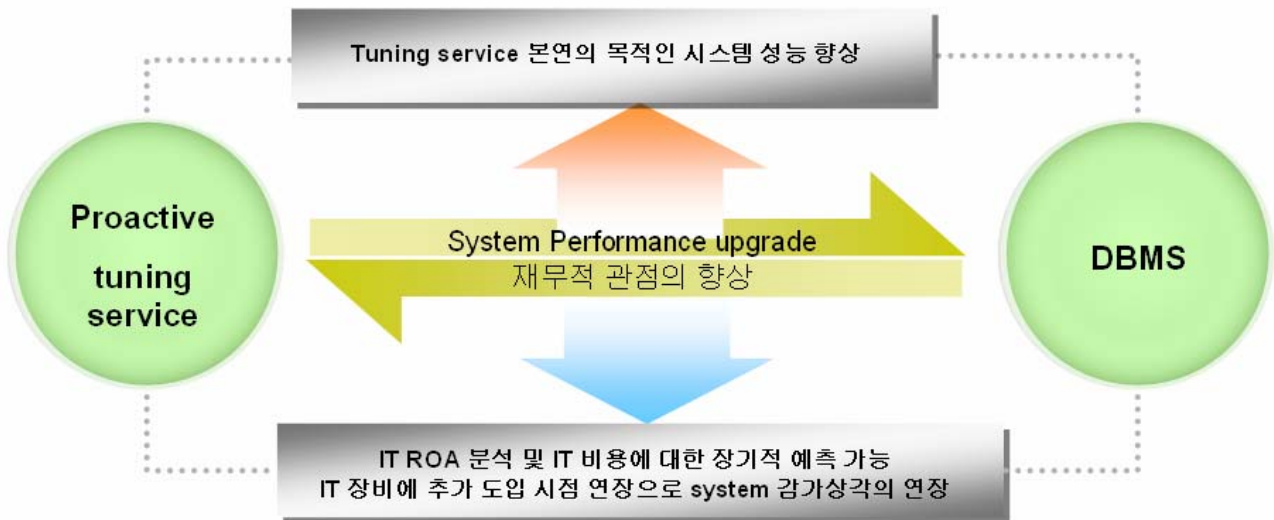
Pro-active tuning service 의 제공 범위는 제한된 Database, 제한된 Server 에 국한 되지 않으며, 고객사 전체의 Server+DB 를 대상으로 그 범위를 확대 함으로써 고객사 전체 Performance 향상에 기여 합니다.



3.5. 기대효과

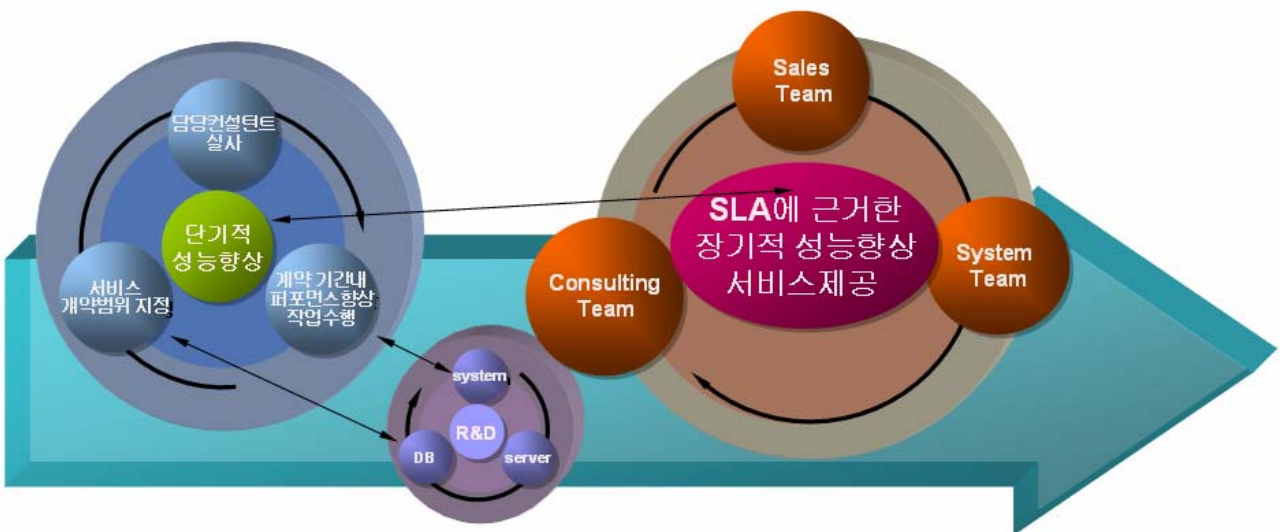
3.5.1. 재무적 관점

기존 Tuning Service 는 주로 System Performance 향상에 따른 업무 트래픽 감소에 초점이 맞춰져 있었습니다. Pro-actvice 서비스는 Tuning 작업을 통한 업무 처리 시간 단축 뿐만 아니라, 업무 처리 시간 단축으로 가져 올 수 있는 재무적 성과를 가능하게 합니다.



3.5.2. 서비스 관점

단기적 성능 향상에 맞추어진 기존 Tuning 서비스는 계약된 system 및 Database 를 서비스 대상으로 하기 때문에 전사적인 차원의 성능 향상을 기대 하기 어려웠습니다. Proactive tuning service 는 계약 기간 동안 주요 비즈니스 Factor 별로 SLA 를 정하여 Tuning consulting 을 수행함으로써 서비스 자체의 안정성을 제고 할 수 있습니다.



3.5.3. 사용자 관점

Proactive tuning service 는 계약 종료 시점 작업한 Tuning 산출물을 통한 기술전수 세미나를 진행 함으로서 고객사의 사용자가 실무에서 바로 적용 가능한 기술을 전수함과 동시에, 계약기간 종료 후에도 Proactive tuning service 를 유지 할 수 있는 방향을 제시 합니다.

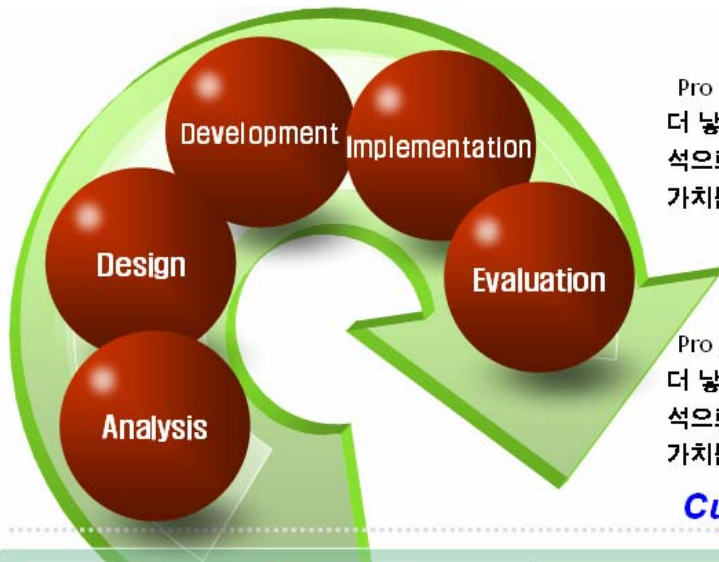
· 사용자 관점의 기술 전수 서비스 제공

계약 종료 시점 일정기간 확실적이고 형식적인 기술 전수 서비스가 아닌 고객사 각각의 사용자들의 특성에 맞추어진 차별화된 기술세미나 진행

- 계약 종료 시점에 고객사 엔지니어 대상 기술 전수 세미나 진행
- 담당 컨설턴트의 고객사 Reporting 자료를 근거로 사례 중심, Case 중심의 기술전수
- 고객사 담당자의 기술 수준에 맞추어, 실무에 적용 가능한 기술 조언
- 단위 추가 업무의 적용 전/후에 대한 지속적인 튜닝을 통한 상시 안정화 상태 제공
- IT 인프라 상황 변화에 능동적으로 대응 할 수 있는 고급 서비스의 “유지”에 초점



3.5.4. 혁신적 관점



Pro active tuning service는 단순 Tuning 업무에서 더 나아가 System, Database, server에 대한 포괄적 분석으로 궁극적으로 IT Resource에 대한 전략적 활용을 가치를 향상 시킬 수 있습니다.

Pro active tuning service는 단순 Tuning 업무에서 더 나아가 System, Database, server에 대한 포괄적 분석으로 궁극적으로 IT Resource에 대한 전략적 활용을 가치를 향상 시킬 수 있습니다.

Customer Business Flow



3.6. 제안 Package별 가격

3.6.1. DBMS Tuning

(단위 : 천원)

상 품	집중 Tuning	정기 Tuning	Total 시간	시간당	List Price	할인가	할인률
Package1	1 주(5 일) 8H*5D = 40H	1 일/분기 8H*3Q = 24H	40H + 24H = 64H (8D)	200	12,800	12,000	6.3 %
Package2	2 주(10 일) 8H*10D = 80H	1 일/월 8H*1D*11M = 88H	80H + 88H = 168H (21D)	200	33,600	30,000	10.7 %
Package3	3 주(15 일) 8H*15D = 120H	2 일/월 8H*2D*11M = 176H	120H + 176H = 296H (37D)	200	59,200	50,000	15.5 %
Package4	4 주(1 개월) 8H*20D = 160H	3 일/월 8H*3D*11M = 264H	160H + 264H = 424H (53D)	200	84,800	68,000	19.8 %

※ 집중 Tuning 이나 정기 Tuning 의 시간을 사용자교육(UAS)로 전환 하실 수 있습니다.

※ Tuning 대상 Instance 를 고정하지 않습니다.

3.6.2. DBMS Tuning+ APM + NA

(단위 : 천원)

상 품	집중 Tuning	정기 Tuning	Total 시간	시간당	List Price	할인가	할인률
Package5	1 주(5 일) 8H*5D = 40H	1 일/분기 8H*3Q = 24H	40H + 24H = 64H (8D)	250	16,000	15,000	6.3 %

※ 추가 1 일당 2,000,000 원이 책정 됩니다.

3.6.3. Server (H/W, OS) Tuning + Storage 재구성 컨설팅 + DBMS Tuning (단위 : 천원)

상 품	집중 Tuning	정기 Tuning	Total 시간	시간당	List Price	할인가	할인률
Package6 Athena	1 주(5 일) 8H*5D = 40H	1 일/분기 8H*3Q = 24H	40H + 24H = 64H (8D)	250	16,000	15,000	6.3 %

※ 추가 1 일당 2,000,000 원이 책정 됩니다.

3.7. Package 설명

내 용	설 명
집중 Tuning	Target DB 의 산정된 Tuning 기간에 따라, 성능이 저하된 DB 를 최적의 성능을 발휘 할 수 있도록 집중적인 Tuning 을 실시 합니다. Database Server Tuning, Database Instance Tuning, Database Object Tuning, Database SQL Tuning
정기 Tuning	집중 Tuning 을 통해서 향상된 최적의 성능을 지속적으로 유지하기 위하여, 정기적으로 Tuning 을 실시하여, 업무 추가/변경, Data 증가 등으로 인한 성능저하의 요인을 찾아 최적의 성능을 유지 할 수 있도록 정기적으로 Tuning 을 실시 합니다.
User Adapted Seminar	장기적 관점의 관리 운영 Know-How 전파 합니다. 내부 관리자 (DBA,SE) 대상 DBMS Administration Know-How 와 사례 소개함으로써, 관리 기술력을 향상할 수 있습니다. 개발자 대상 최적의 SQL 을 작성할 수 있는 Skill 을 전파합니다.
Package5	DBMS Tuning 뿐만이 아니라, DB 및 Network 에 대한 Advanced Trouble Shooting 을 실시 합니다. (예를 들어, 고객사의 본사에서는 속도가 빠르는데, 지사에서는 느릴 경우, 해결되지 않는 Database 의 복구)
Package6	사용중인 고가의 System 의 최적화 여부를 진단해서 튜닝을 실시 합니다. Server(Hardware, O/S), Storage 구성상태를 진단하여, 최상의 성능을 발휘 할 수 있도록 Tuning 을 실시합니다,