
Goodus 기술노트 [20 회]

DataPump

Author	한대영, 김형미
Creation Date	2007-05-28
Last Updated	2007-05-28
Version	1.0
Copyright(C) 2004 Goodus Inc. All Rights Reserved	

Version	변경일자	변경자(작성자)	주요내용
1	2007-05-28	한대영, 김형미	문서 최초 작성
2			
3			

Contents

1. Data Pump.....	3
1.1. Oracle Data pump란?	3
1.2. Data pump Key features	3
1.2.1. Fast Performance.....	3
1.2.2. Imporved Management Restart.....	3
1.2.3. Fine-Grained Object Selection	3
1.2.4. Monitoring and Estimating Capability.....	4
1.2.5. Network Mode	4
1.3. Data pump overview.....	5
1.3.1. Data Pump Overview	5
1.3.2. Data Pump Benefit	5
1.3.3. Data Pump File Locations	5
1.3.4. Data Pump File Naming and size.....	6
1.3.5. Data Pump Filtering	6
1.3.6. Data Pump Job Monitoring.....	7
1.3.7. Data Pump Export and Import.....	7
1.3.8. Data Pump restarting	8
1.4. Data pump 실습	10
1.4.1. 전체 데이터베이스 export 실습.....	10
1.4.2. 특정 스키마 DDL 스크립트 생성 실습	13
1.4.3. 존재하는 table import	15
부록 – A expdp parameter	16
부록 – B impdp parameter	17
2. Pro-Active Tuning Service	19
2.1. 실제 사용자(End-User) 관점의 응답시간 튜닝	19
2.2. 최상의 성능 상태로 비즈니스 고가용성을 유지	19
2.3. Knowledge Transfer.....	20
2.4. Tuning 범위 확대	20
2.5. 기대효과	21
2.5.1. 재무적 관점.....	21
2.5.2. 서비스 관점.....	21
2.5.3. 사용자 관점.....	22
2.5.4. 혁신적 관점.....	22

1. Data Pump

1.1. Oracle Data pump란?

Oracle Data Pump 는 Oracle Database 10g 버전에서 제공되는 Utility 로 향상된 데이터 이동을 가능하게 한다.

이전 버전의 오라클을 설치한 홈 디렉토리에는 'imp','exp'라는 실행 파일이 있다. 이는 오라클에서 제공하는 backup 및 recovery 에 사용되는 도구 이다.

Exp 는 데이터베이스에 저장되어 있는 데이터들을 OS 의 바이너리 파일로 전환하는 도구이고, imp 는 바이너리 파일을 데이터베이스 안의 데이터로 전환하는 도구이다. 새로 등장한 Data Pump 는 exp 와 imp 를 대체하기 위하여 오라클 10g 버전부터 제공되는 유틸리티로 Exp / Imp 와 유사한 동작을 하지만 data pump 가 훨씬 효율적으로 동작한다.

Exp/Imp 와 비교하여 그 효율성을 예를 들자면 exp 시 single thread 에서 2 배가 빠르고 imp 시 15~45 배 빠르므로 데이터베이스간의 bulk data 와 meta data 의 전송시간을 줄이는데 효율적으로 사용될 수 있다.

1.2. Data pump Key features

1.2.1. Fast Performance

앞에서 말한 것과 같이 Data Pump Export and Import 유틸리티는 기존의 Export and Import 유틸리티보다 훨씬 빠르다. Data Pump Export 에서 direct path method 를 사용시 single stream data unload 에서 기존의 export 보다 2 배가 빠르다. 이는 direct path API 가 더 효과적으로 수정 되었기 때문이다. Parallelism 의 level 에 따라서는 더욱 향상된 performance 를 보일 수 있다.

Data pump import 에서는 single stream 의 data load 시 기존의 import 보다 15~45 배가 빠르다. 이는 기존의 import 에서 단순히 export dump 파일에서 레코드를 읽고 일반적인 insert into 명령을 사용해서 대상 테이블에 삽입 하는 대신에 Data pump import 는 Direct path method loading 을 사용하기 때문이다.

1.2.2. Improved Management Restart

모든 Data Pump operation 은 Data Pump job 을 실행하는 스키마에 만들어진 master table 을 가지고 있다. Master table 은 현재 수행중인 모든 export 또는 import 시 객체의 상태정보와 dump file set 에서의 위치정보를 가지고 있다. 이는 갑작스런 job 의 중단에도 job 의 성공적인 종료에 상관 없이 어떤 object 의 작업이 진행 중이었는지 알 수 있게 해준다. 그래서 master table 과 dump file set 이 있는 한 모든 정지된 data pump job 은 데이터 손실 없이 다시 시작할 수 있다.

1.2.3. Fine-Grained Object Selection

Data Pump job 은 거의 모든 type 의 object 를 exclude 또는 include 시킬 수 있다. 아래의 parameter 가 사용된다.

- * EXCLUDE - 특정 객체 유형을 제외한다. (예: EXCLUDE=TABLE:EMP)
- * INCLUDE - 특정 객체 유형을 포함한다. (예: INCLUDE=TABLE_DATA)
- * CONTENT - 로드를 취소할 데이터를 지정한다.
적합한 키: (ALL), DATA_ONLY 및 METADATA_ONLY.
- * QUERY - 테이블의 부분 집합을 익스포트하기 위해 사용되는 술어 절이다.

1.2.4. Monitoring and Estimating Capability

Data Pump 는 Standard progress , error message 를 log file 에 기록할 뿐만 아니라 현재 operation 의 상태를 대화식모드 'command line'으로 보여 준다. Job 의 completion percentage 를 측정하여 보여주며 초 단위의 지정한 time period 에 따라 자동으로 update 하여 표시한다.

1 개 이상의 client 가 running job 에 attach 수 있기 때문에 업무환경에서 job 을 실행하고, detach 한 후 집에 가서 job 을 reattach 하여 끊임 없이 모든 job 을 모니터링 할 수 있다.

모든 export job이 시작할 때 대략적인 전체 unload 양을 측정해 준다. 이는 사용자가 dump file set을 위한 충분한 양의 disk space를 할당할 수 있게 한다.

1.2.5. Network Mode

Data Pump Export and Import는 job의 source가 리모트 인스턴스 일 경우를 위한 network mode를 지원한다.

Network을 통해 import를 할 때 source가 dump file set이 아닌 다른 database에 있기 때문에 dump file이 없다.

Network 를 통해 export 를 할 때 source 가 다른 시스템에 있는 read-only database 일 수 있다. Dumpfile 은 local(non-networked)export 처럼 local 시스템에 쓰이게 된다.

1.3. Data pump overview

1.3.1. Data Pump Overview

- expdp/impdp로 제공 되어 진다.
- exp/imp의 superset 이다.
- Data 와 metadata를 매우 빠른 속도로 load/unload 하는 Server-based facility이다.
==> dump file sets은 Server에 생성
- DBMS_DATAPUMP PL/SQL Package를 이용하여 사용 가능 하다.
- Web-based interface <--access from EM Database Control이 가능하다.
- Data Pump job을 실행하는 schema에 master table(MT)이 만들어 진다.
MT는 해당 job의 모든 것(aspects)을 관리하며 data pump(expdp)의 마지막 단계에서 pump file sets에 기록된다.
file based import 작업(impdp)시 dump file에 있는 MT 내용을 current user의 schema에 제일먼저 loading한다.
계획 또는 예상치 못한 job 의 중단 시 재가동수 있게 하는 Data Pump 의 핵심이 MT 이다.

```
expdp client -->Data Pump Job -- DB(Master Table:Job소유자) -->Dump file sets(MT내용 기록)
      ^
      |
      |_____Remote의 Server Process
```

Client process는 Data Pump API를 call한다.

-여러 개의 clients가 모니터링하고 control하기 위해서 job을 attach/detach 한다.

1.3.2. Data Pump Benefit

- Data Access Methods : Direct Path, External Tables
- Detach from, reattach to log-running jobs
- Restart Data Pump Jobs
- Find-grained object selection <-- 원하는 rows만(EXCLUDE, INCLUDE, CONTENT)
- Explicit database version specification
- Parallel execution
- Estimate export job space <--ESTIMATE_ONLY
- Network Mode에서는 Remote의 server process가 DB link를 이용하여 Local에 dump file을 직접 만들어 준다..
- Import 과정에서 target data file name, schema, tablespace 을 변경할 수 있다.

1.3.3. Data Pump File Locations

- Data pump file 종류
 - DUMP file : data와 metadata를 포함한다.
 - LOG file : operation과 관련된 message를 기록한다.
 - SQL file : impdp에서 SQLFILE operation의 결과를 기록한다.
 - Data Pump는 server-based 이므로 Oracle directory path를 통해서 Data Pump file에 access한다.
Absolute path는 보안상 지원되지 않는다.
- Order of precedence of file locations
 - 1) per-file directory
 - dump file, log file, sql file 마다 지정될 수 있다. 콜론(:)으로 directory 와 file name 을 구분한다.
 - 예) dumpfile=AA:A.dmp
 - 2) DIRECTORY parameter

-
- directory object를 사용한다.
Create Directory DIR_PJH as '/home/oracle10g/test/';
Grant read, write On Directory DIR_PJH to SCOTT;
Directory=AA
Dumpfile=A.dmp

3) DATA_PUMP_DIR 환경변수

- DIRECTORY Parameter를 대신하여 directory object name을 설정한다.
export DATA_PUMP_DIR=AA
Dumpfile=A.dmp
- 위의 모든 경우에 시도하려는 operation에 대해 directory object에 대해 적절한 access privs가 있어야 한다.
Export할 경우 모든 file에 대해 write access가 필요하다.
Import할 경우 dump file에 대해 read access, log file과 sql file에 대해 write access가 필요하다.

1.3.4. Data Pump File Naming and size

(1) DUMPFILE

- file list는 , 로 분리한다.
- %U template --> two-character, fix-width, 01부터 증가하는 integer 를 가진다.
- DUMPFILE 이 지정되어 있지 않으면 expdat.dmp 가 default로 사용된다. Default는 autoextensible이다.

(2) FILESIZE

- FILESIZE 가 지정되어 있으면 각 file은 FILESIZE안으로 만들어지고 늘어날 수 없다. dump 공간이 더 필요하고 template %U가 지정되었다면, 새로운 파일이 생성된다. 그렇지 않으면 사용자는 new file을 add하라는 메시지를 받는다.

(3) PARALLEL

- %U가 지정되면 PARALLEL parameter의 개수만큼 초기에 file이 생성된다.
- 기존에 존재하는 file과 이름이 중복될 경우 overwrite하지 않고 에러를 발생시키고 job이 abort된다.
- 복수개의 dump file template가 제공되면 round-robin fashion으로 dump file을 생성하는 데 사용한다.

1.3.5. Data Pump Filtering

(1) Find-grained object selection

- 기존의 exp/imp는 index, trigger, grant, constraint를 포함하거나 제외하는 것이 있으나 data pump는 virtually any type of object를 포함하거나 제외할 수 있다.
- EXCLUDE 와 INCLUDE는 mutually exclusive 하다.
- INCLUDE = object_type[:"name_expr"]
- EXCLUDE = object_type[:"name_expt"]
- 모든 view, 모든 package, EMP로 시작하는 Index만 제외한다.
EXCLUDE=view
EXCLUDE=package
EXCLUDE=INDEX:"LIKE 'EMP%' "

(2) Data Selection

- CONTENT = ALL(def) | METADATA_ONLY | DATA_ONLY
- QUERY = [Schema.][table_name:] "query_clause"
- CONTENT에 data_only가 사용되면 EXCLUDE 와 INCLUDE를 사용할 수 없다.

```
QUERY=hr.employees:"WHERE department_id in (10,20) and salary < 1600 ORDER BY department_id"
```

<--특정 table 을 지정해서 해당 table 로 한정. imp 시에도 적용.

1.3.6. Data Pump Job Monitoring

- 데이터베이스 뷰에서 실행되는 Data Pump 작업에 관해서도 자세한 정보를 확인할 수 있다.
- DBA_DATAPUMP_JOBS - 작업에서 실행되는 작업자 프로세스(DEGREE 열)의 수를 확인할 수 있다.
- DBA_DATAPUMP_SESSIONS -이전 뷰 및 V\$SESSION과 조인하여 foreground 프로세스 세션의 SID확인 할 수 있다.

```
select sid, serial#
from v$session s, dba_datapump_sessions d
where s.saddr = d.saddr;
```

- V\$SESSION_LONGOPS - 작업 완료에 걸리는 시간을 예측하는 또 다른 유용한 정보를 얻을 수 있다.

```
select sid, serial#, sofar, totalwork
from v$session_longops
where opname = 'CASES_EXPORT' and sofar != totalwork;
```

totalwork 열에는 총 작업량이 표시되는데, 이 중 현재까지 sofar 작업량을 완료했으므로 이를 통해 얼마나 더 시간이 걸릴지 예측할 수 있다.

1.3.7. Data Pump Export and Import

1) Parallel Full Export and Import

```
> expdp system/manager full=y parallel=4
      dumpfile=DATADIR1:full1%U.dat,
            DATADIR2:full2%U.dat,
            DATADIR3:full3%U.dat,
            DATADIR4:full4%U.dat
      filesize=2G
```

<--4개의 work process를 가진 full export,
Pump file은 DATADIR1, DATADIR2, DATADIR3, DATADIR4 네 곳에 라운드로빈 방식으로 생성된다.
2G를 넘지 않으면서 최소4개 생성.
Job 과 master table 이름은 default로 SYSTEM_EXPORT_FULL_01 를 가진다.

```
>impdp system/manager directory= NET_STORGAE_1 parallel=4
      dumpfile= full1%U.dat,
            full2%U.dat,
            full3%U.dat,
            full4%U.dat
```

<--expdp로 받은 dump file을 network를 통해 NET_STORAGE_1 이라는 directory object위치로 보내졌다. Default import는 dump set 전체를 import하는 것이므로 Full=y 는 필요 없다.
Job 과 master table 이름은 default로 SYSTEM_IMPORT_FULL_01 를 가진다.

2) Limited Schema Export (fine-grained)

```
> expdp system/manager schemas=hr,oe
directory=USR_DATA
dumpfile=schema_hr_oe.dat
parfile=exp_par.txt <-----
include=function
include=procedure
include=pacakge
include=type
include=view:"like 'PRODUCT%'"
```

<--HR, OE schema에서 모든 func, prod, pkg, user-defined type, PRODUCT로 시작하는 view를 export한다.
Schema definition과 system priv graints는 export되지 않는다.

```
> impdp system/manager directory=USR_DATA
dumpfile=schema_hr_oe.dat
sqlfile=schema_hr_oe.dat
```

<--실제 import는 하지 않고 dmp file에서 DDL 문장만 뽑아낸다.

3) Network Mode Import (DB Link)

```
> impdp system/manager schemas=hr,sh,payroll
parfile=imp_par.txt <-----
network_link=finance.hq.com <--db link
remap_schema=payroll:finance
```

<--Source DB에 dblink로 붙어서 hr, sh, payroll schema를 가져온 다음 imp 한다.
이때 payroll schema로 finance schema로 만들어 진다.
SYSTEM은 IMPORT_FULL_DATABASE role을 가지고 있고 Source DB에 대해서는
EXPORT_FULL_DATABASE role을 가지므로 Target DB에 해당 schema definition이 없으면
만들어진다.
flashback_time은 예전의 CONSISTENT와 동일하다.

4) Data-Only Unload

```
> expdp hr/hr parfile=exp_par.txt dumpfile=expdat.dmp content=data_only
include=table:"in ('DEPARTMENTS','DEPARTMENTS_HIST','EMPLOYEES','EMP_HIST')"
query="where DEPARTMENT_ID != 30 order by DEPARTMENT_ID"
```

1.3.8. Data Pump restarting

1) Attaching to Existing Job

```
> expdp system/manager attach=EXP_TS1
<--Job name(MT name) :dba_datapump_jobs
job: EXP_TS1          해당 스키마에 active export job 이 하나만 있을 경우 안 적어도 된다..
owner: SYSTEM
mode:
status:
Export> STOP_JOB
<--중지.
```

Attach session은 terminate 되고 실행되던 job은 controlled fashion으로 run down 된다.

해당 Job은 dump file과 SYSREM.EXP_TS1 table이 disturbed 되지 않는 한 startable 하다.

2) Restarting Stopped Job

> expdp system/manager attach=exp_ts1

<--같은 schema안에 여러 개의 outstanding job이 있으면 job

name지정한다.

Export> parallel=4

Export> start_job

Export> status =600 <--10분

<-- detailed per-work process가 10분 단위로 regular status message를 보여준다.

KILL_JOB로 job을 kill한다.

<--status, status=600(초)

stop_job,

start_job,

continue_client: attach한 session이 계속 받아서 expdp 실행한다.(logging mode로 전환)

exit_client: Attach를 빠져 나옴. expdp는 background로 실행한다.

parallel=4,

1.4. Data pump 실습

1.4.1. 전체 데이터베이스 export 실습

```
SQL> conn /as sysdba
연결되었습니다.
SQL> create directory dump as 'C:\Woracle\backup';    ->directory 를 생성한다.

디렉토리가 생성되었습니다.
SQL> grant read ,write on directory dump to public;    -> directory 에 권한을 부여한다.

권한이 부여되었습니다.
SQL> host
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985–2001 Microsoft Corp.
C:\Woracle>expdp system/oracle dumpfile=full.dmp directory=dump full=y job_name=Lucie

Export: Release 10.2.0.1.0 – Production on 화요일, 29 5 월, 2007 17:17:41

Copyright (c) 2003, 2005, Oracle. All rights reserved.

접속 대상: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 – Production
With the OLAP and Data Mining options
"SYSTEM"."LUCIE" 시작 중: system/***** dumpfile=full.dmp directory=dump full=y job_name=Lucie
BLOCKS 메소드를 사용하여 예측 진행 중...
객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA 처리 중
BLOCKS 메소드를 사용한 총 예측: 66.56 MB ->대략적인 dmp 파일 size 를 예측할 수 있다.
객체 유형 DATABASE_EXPORT/TABLESPACE 처리 중
객체 유형 DATABASE_EXPORT/SYS_USER/USER 처리 중
객체 유형 DATABASE_EXPORT/SCHEMA/USER 처리 중
객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/POST_INSTANCE/PROCDEPOBJ 처리 중
.
.
.
.. "SCOTT"."DEPT"                                48.00 MB 2097152 행이 익스포트됨
.. "SYSMAN"."MGMT_JOB_CRED_PARAMS"                11.70 KB    18 행이 익스포트됨
.. "SYSMAN"."MGMT_JOB_PROP_PARAMS"                8.820 KB    12 행이 익스포트됨
.. "SYSMAN"."MGMT_JOB_STEP_PARAMS"               127.3 KB   1128 행이 익스포트됨
.

Control + c ->중간에 끊어도 job 이 끊기지 않고 명령모드로 들어간다.

Export>status
작업: LUCIE
작업: EXPORT
모드: FULL
상태: EXECUTING
처리된 바이트: 50,337,376
완료율: 84 -> 진행률을 알 수 있다.
현재 병렬도: 1
작업 오류 수: 0
덤프 파일: C:\WORACLEWBACKUPWFULL.DMP
기록된 바이트: 55,226,368

작업자 1 상태:
상태: EXECUTING
객체 스키마: SYSMAN
객체 이름: MGMT_JOB_EXECUTION
객체 유형: DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA

완료된 객체: 45
```

총 객체: 408
작업자 병렬도: 1

Export> stop_job -> job 을 정지시킨다.
이 작업을 정지하겠습니까([예]/[아니오]):

C:\Woracle>
C:\Woracle>sqlplus "/as sysdba"

SQL*Plus: Release 10.2.0.1.0 - Production on 화 5 월 29 17:32:59 2007

Copyright (c) 1982, 2005, Oracle. All rights reserved.
다음에 접속됨:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the OLAP and Data Mining options

SQL>select owner_name,job_name,operation,job_mode,state from dba_datapump_jobs;
OWNER_NAME JOB_NAME OPERATION JOB_MODE STATE

OWNER_NAME	JOB_NAME	OPERATION	JOB_MODE	STATE
SYSTEM	LUCIE	EXPORT	FULL	NOT RUNNING

->job 상태를 확인할 수 있다.

SQL> exit

C:\Woracle>expdp system/oracle attach=lucie ->job 을 다시 attach 한다.

Export: Release 10.2.0.1.0 - Production on 화요일, 29 5 월, 2007 17:35:54

Copyright (c) 2003, 2005, Oracle. All rights reserved.

접속 대상: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the OLAP and Data Mining options

작업: LUCIE
소유자: SYSTEM
작업: EXPORT
생성자 권한: FALSE
GUID: 18405C1B820C4ABB9B30C4948E0D356F
시작 시간: 화요일, 29 5 월, 2007 17:35:56
모드: FULL
인스턴스: ora10
최대 병렬도: 1
EXPORT 작업 매개변수:
매개변수 이름 매개변수 값:
CLIENT_COMMAND system/***** dumpfile=full.dmp directory=dump full=y
job_name=Lucie
상태: IDLING
처리된 바이트: 51,646,000
완료율: 99
현재 병렬도: 1
작업 오류 수: 0
덤프 파일: C:\Woracle/backupWfull.dmp
기록된 바이트: 55,914,496

작업자 1 상태:
상태: UNDEFINED

SQL>select owner_name,job_name,operation,job_mode,state from dba_datapump_jobs;
OWNER_NAME JOB_NAME OPERATION JOB_MODE STATE

OWNER_NAME	JOB_NAME	OPERATION	JOB_MODE	STATE
SYSTEM	LUCIE	EXPORT	FULL	IDLING

->job 상태를 확인할 수 있다.

```
Export> start_job ->Job 을 다시 restar 한다.
```

```
Export> status
```

```
작업: LUCIE
  작업: EXPORT
  모드: FULL
  상태: COMPLETING
  처리된 바이트: 51,646,001
  완료율: 100
  현재 병렬도: 1
  작업 오류 수: 0
  덤프 파일: C:\Woracle\backupWfull.dmp
  기록된 바이트: 64,684,032
```

```
작업자 1 상태:
  상태: WORK WAITING
```

```
C:\Woracle>
```

Logfile 확인

지정한 directory 위치에 “export” log file 을 확인 한다. 파일의 끝부분을 보면 성공적으로 완료됨을 확인할 수 있다.

```
.
.
.
"SYSTEM"."LUCIE" 작업이 17:18:56 에서 사용자 요청에 의해 정지됨
LUCIE 작업이 화요일, 29 5 월, 2007 17:35 에서 다시 열림      ->작업을 정지했다 다시 시작한
                                                              것을 확인 할 수 있음
"SYSTEM"."LUCIE" 재시작 중: system/***** dumpfile=full.dmp directory=dump full=y
job_name=Lucie
마스터 테이블 "SYSTEM"."LUCIE"이(가) 성공적으로 로드됨/로드 취소됨
*****
SYSTEM.LUCIE 에 대해 설정된 덤프 파일:
  C:\Woracle\backupWfull.dmp
"SYSTEM"."LUCIE" 작업이 17:37:12 에서 성공적으로 완료됨
```

1.4.2. 특정 스키마 DDL 스크립트 생성 실습

```
SQL> conn /as sysdba
```

연결되었습니다.

```
SQL> create directory dump as 'C:\Oracle\backup'; -->directory 를 생성한다.
```

디렉토리가 생성되었습니다.

```
SQL> grant read ,write on directory dump to public; --> directory 에 권한을 부여한다.
```

```
C:\Oracle>impdp system/oracle directory=dump dumpfile=full.dmp schemas=scott sqlfile=ddl_scott.sql
```

<이 명령은 dump 로 지정된 디렉터리에 ddl_scott.sql 로 명명된 파일을 생성하며 익스포트 덤프 파일 내의 객체 스크립트를 생성한다.>

Import: Release 10.2.0.1.0 - Production on 화요일, 29 5 월, 2007 19:12:13

Copyright (c) 2003, 2005, Oracle. All rights reserved.

접속 대상: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production

With the OLAP and Data Mining options

마스터 테이블 "SYSTEM"."SYS_SQL_FILE_SCHEMA_01"이(가) 성공적으로 로드됨/로드 취소됨

"SYSTEM"."SYS_SQL_FILE_SCHEMA_01" 시작 중: system/***** directory=dump dumpfile=full.dmp schemas=

scott sqlfile=ddl_scott.sql

객체 유형 DATABASE_EXPORT/SCHEMA/USER 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/ROLE_GRANT 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/PROCACT_SCHEMA 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/TABLE 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/STATISTICS/TABLE_STATISTICS 처리 중

"SYSTEM"."SYS_SQL_FILE_SCHEMA_01" 작업이 19:12:22 에서 성공적으로 완료됨

Logfile 확인

- dump 로 지정된 C:\Oracle\backup 에 ddl_scott.sql 파일이 생성된다.

```
-- CONNECT SYSTEM
```

```
-- new object type path is: DATABASE_EXPORT/SCHEMA/USER
```

```
CREATE USER "SCOTT" IDENTIFIED BY VALUES 'F894844C34402B67'
```

```
DEFAULT TABLESPACE "USERS"
```

```
TEMPORARY TABLESPACE "TEMP";
```

```
-- new object type path is: DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT
```

```
GRANT UNLIMITED TABLESPACE TO "SCOTT";
```

```
GRANT CREATE SESSION TO "SCOTT";
```

```
-- new object type path is: DATABASE_EXPORT/SCHEMA/ROLE_GRANT
```

```
GRANT "RESOURCE" TO "SCOTT";
```

```
-- new object type path is: DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE
```

```
ALTER USER "SCOTT" DEFAULT ROLE ALL;
```

```
-- new object type path is: DATABASE_EXPORT/SCHEMA/PROCACT_SCHEMA
```

```
-- CONNECT SCOTT
```

```
BEGIN
```

```
sys.dbms_logrep_imp.instantiate_schema(schema_name=>SYS_CONTEXT('USERENV','CURRENT_SCHEMA'), export_db_name=>'ORA10', inst_scn=>'283762');
```

```
COMMIT;
```

```
END;
```

```
/
```

```
-- new object type path is: DATABASE_EXPORT/SCHEMA/TABLE/TABLE
```

```
-- CONNECT SYSTEM
```

```
CREATE TABLE "SCOTT"."EMP"  
  (  
    "EMPNO" NUMBER(4,0) NOT NULL ENABLE,  
    "ENAME" VARCHAR2(10),  
    "JOB" VARCHAR2(9),  
    "MGR" NUMBER(4,0),  
    "HIREDATE" DATE,  
    "SAL" NUMBER(7,2),  
    "COMM" NUMBER(7,2),  
    "DEPTNO" NUMBER(2,0)  
  ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING  
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
    PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)  
  TABLESPACE "USERS" ;  
CREATE TABLE "SCOTT"."DEPT"  
  (  
    "DEPTNO" NUMBER(2,0),  
    "DNAME" VARCHAR2(14),  
    "LOC" VARCHAR2(13)  
  ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING  
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
    PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)  
  TABLESPACE "USERS" ;
```

```
.  
.  
.
```

1.4.3. 존재하는 table import

1) content=data_only 포함한 경우

```
C:\Woracle>Impdp system/oracle dumpfile=full.dmp directory= dump content=data_only  
job_name=data_import logfile=table_log tables=scott.dept
```

Import: Release 10.2.0.1.0 – Production on 화요일, 29 5 월, 2007 19:42:15

Copyright (c) 2003, 2005, Oracle. All rights reserved.

접속 대상: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 – Production
With the OLAP and Data Mining options

마스터 테이블 "SYSTEM"."DATA_IMPORT"이(가) 성공적으로 로드됨/로드 취소됨

"SYSTEM"."DATA_IMPORT" 시작 중: system/***** dumpfile=full.dmp directory= dump
content=data_only

job_name=data_import logfile=table_log tables=scott.dept

객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA 처리 중

. . "SCOTT"."DEPT" 48.00 MB 2097152 행이 임포트됨

"SYSTEM"."DATA_IMPORT" 작업이 19:42:52 에서 성공적으로 완료됨

2) content=data_only 포함하지 않은 경우

```
C:\Woracle>Impdp system/oracle dumpfile=full.dmp directory= dump job_name=data_import  
logfile=table_log tables=scott.dept
```

Import: Release 10.2.0.1.0 – Production on 화요일, 29 5 월, 2007 19:41:02

Copyright (c) 2003, 2005, Oracle. All rights reserved.

접속 대상: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 – Production
With the OLAP and Data Mining options

마스터 테이블 "SYSTEM"."DATA_IMPORT"이(가) 성공적으로 로드됨/로드 취소됨

"SYSTEM"."DATA_IMPORT" 시작 중: system/***** dumpfile=full.dmp directory= dump
job_name=data_imp

rt logfile=table_log tables=scott.dept

객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/TABLE 처리 중

**ORA-39151: "SCOTT"."DEPT" 테이블이 존재합니다. 건너 뛰기 table_exists_action 으로 인해 모든 종속
메타 데이터 및 데이터를 건너 뛩니다.**

객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA 처리 중

객체 유형 DATABASE_EXPORT/SCHEMA/TABLE/STATISTICS/TABLE_STATISTICS 처리 중

"SYSTEM"."DATA_IMPORT" 작업이 1 오류와 함께 19:41:09 에서 완료됨

* 임포트 프로세스의 기본 작업 방식은 테이블 및 연관된 모든 객체를 생성하고 테이블이 있는 상태에서 오류를 만들어 낸다.

부록 – A expdp parameter

키워드	설명(기본값)
ATTACH	기존 작업에 연결합니다. 예: ATTACH [=작업 이름]
CONTENT	로드할 데이터를 지정합니다. 적합한 키: (ALL), DATA_ONLY 및 METADATA_ONLY.
DIRECTORY	덤프, 로그 및 sql 파일에 사용할 디렉토리 객체입니다.
DUMPFILE	임포트할 덤프 파일(expdat.dmp) 목록입니다. 예: DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp.
ENCRYPTION_PASSWORD	암호화된 열 데이터에 액세스하기 위한 암호 키입니다. 이 매개변수는 네트워크 임포트 작업에 대해 적합하지 않습니다.
ESTIMATE	작업 예측을 계산합니다. 적합한 키워드: (BLOCKS) 및 STATISTICS 입니다.
EXCLUDE	특정 객체 유형을 제외합니다. (예: EXCLUDE=TABLE:EMP)
FLASHBACK_SCN	세션 스냅샷을 다시 설정하기 위해 사용되는 SCN 입니다.
FLASHBACK_TIME	지정된 시간에 가장 가까운 SCN 을 가져오기 위해 사용되는 시간입니다.
FULL	소스에서 모든 것을 임포트합니다. (Y)
HELP	도움말 메시지를 표시합니다. (N)
INCLUDE	특정 객체 유형을 포함합니다. (예: INCLUDE=TABLE_DATA)
JOB_NAME	생성할 임포트 작업 이름입니다.
LOGFILE	로그 파일 이름(import.log)입니다.
NETWORK_LINK	소스 시스템에 대한 원격 데이터베이스 링크 이름입니다.
NOLOGFILE	로그 파일을 작성하지 않습니다. (N)
PARALLEL	현재 작업에 대한 활성 작업자 수를 변경합니다.
PARFILE	매개변수 파일을 지정합니다.
QUERY	테이블의 부분 집합을 임포트하기 위해 사용되는 술어 절입니다.
REMAP_DATAFILE	모든 DDL 문의 데이터 파일 참조를 재정의합니다.
REMAP_SCHEMA	한 스키마의 객체를 다른 스키마에 로드합니다.
REMAP_TABLESPACE	테이블스페이스 객체가 다른 테이블스페이스에 재매핑됩니다.
REUSE_DATAFILES	테이블스페이스가 존재하는 경우 초기화됩니다. (N)
SCHEMAS	임포트할 스키마 목록입니다.
SKIP_UNUSABLE_INDEXES	사용할 수 없는 인덱스 상태로 설정된 인덱스를 건너 뛩니다.
SQLFILE	모든 SQL DDL 을 지정된 파일에 씁니다.
STATUS	작업 상태가 모니터되는 빈도(초)입니다. 기본값(0)은 사용 가능할 때 새 상태를 표시합니다.
STREAMS_CONFIGURATION	스트림 메타 데이터의 로드 활성화
TABLE_EXISTS_ACTION	임포트된 객체가 존재하는 경우 수행할 작업입니다. 적합한 키워드: (SKIP), APPEND, REPLACE 및 TRUNCATE.
TABLES	임포트할 테이블 목록을 식별합니다.
TABLESPACES	임포트할 테이블스페이스 목록을 식별합니다
TRANSFORM	해당 객체에 적용할 메타 데이터 변환입니다. 적합한 변환 키워드: SEGMENT_ATTRIBUTES, STORAGE OID 및 PCTSPACE 입니다.
TRANSPORT_DATAFILES	전송 가능 모드로 임포트할 데이터 파일 목록입니다. 메타 데이터가 로드될 테이블스페이스 목록입니다.
TRANSPORT_FULL_CHECK	모든 테이블의 저장 영역 세그먼트를 확인합니다. (N)
TRANSPORT_TABLESPACES	NETWORK_LINK 모드 임포트 작업에서만 적합합니다.
VERSION	엑스포트할 객체의 버전입니다. 적합한 키워드: (COMPATIBLE), LATEST 또는 임의의 적합한 데이터베이스 버전. NETWORK_LINK 및 SQLFILE 에 대해서만 적합합니다.

다음 명령은 대화형 모드에서 적합합니다.

참고: 약어도 허용됨

명령	설명(기본값)
CONTINUE_CLIENT	로깅 모드로 돌아옵니다. 유휴 상태인 경우 작업이 재시작됩니다.
EXIT_CLIENT	덤프, 로그 및 sql 파일에 사용할 디렉토리 객체입니다.
HELP	대화형 명령을 요약합니다.

KILL_JOB	작업을 분리하고 삭제합니다.
PARALLEL	현재 작업에 대한 활성 작업자 수를 변경합니다. PARALLEL=<작업자 수>.
START_JOB	현재 작업을 시작/재개합니다. START_JOB=SKIP_CURRENT 는 작업 정지 시 진행 중이던 작업을 건너 뛴 후 해당 작업을 시작합니다.
STATUS	작업 상태가 모니터되는 빈도(초)입니다. 기본값(0)은 사용 가능할 때 새 상태를 표시합니다. STATUS[=간격]
STOP_JOB	작업 실행을 순서대로 종료하고 클라이언트를 종료합니다. STOP_JOB=IMMEDIATE 는 데이터 펌프 작업의 데이터 펌프 작업.

부록 - B impdp parameter

키워드	설명(기본값)
ATTACH	기존 작업에 연결합니다. 예: ATTACH [=작업 이름]
CONTENT	로드할 데이터를 지정합니다. 적합한 키: (ALL), DATA_ONLY 및 METADATA_ONLY.
DIRECTORY	덤프, 로그 및 sql 파일에 사용할 디렉토리 객체입니다.
DUMPFIL	임포트할 덤프 파일(expdat.dmp) 목록입니다. 예: DUMPFIL=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp.
ENCRYPTION_PASSWORD	암호화된 열 데이터에 액세스하기 위한 암호 키입니다. 이 매개변수는 네트워크 임포트 작업에 대해 적합하지 않습니다.
ESTIMATE	작업 예측을 계산합니다. 적합한 키워드: (BLOCKS) 및 STATISTICS 입니다.
EXCLUDE	특정 객체 유형을 제외합니다. (예: EXCLUDE=TABLE:EMP)
FLASHBACK_SCN	세션 스냅샷을 다시 설정하기 위해 사용되는 SCN 입니다.
FLASHBACK_TIME	지정된 시간에 가장 가까운 SCN 을 가져오기 위해 사용되는 시간입니다.
FULL	소스에서 모든 것을 임포트합니다. (Y)
HELP	도움말 메시지를 표시합니다. (N)
INCLUDE	특정 객체 유형을 포함합니다. (예: INCLUDE=TABLE_DATA)
JOB_NAME	생성할 임포트 작업 이름입니다.
LOGFILE	로그 파일 이름(import.log)입니다.
NETWORK_LINK	소스 시스템에 대한 원격 데이터베이스 링크 이름입니다.
NOLOGFILE	로그 파일을 작성하지 않습니다. (N)
PARALLEL	현재 작업에 대한 활성 작업자 수를 변경합니다.
PARFILE	매개변수 파일을 지정합니다.
QUERY	테이블의 부분 집합을 임포트하기 위해 사용되는 술어 절입니다.
REMAP_DATAFILE	모든 DDL 문의 데이터 파일 참조를 재정의합니다.
REMAP_SCHEMA	한 스키마의 객체를 다른 스키마에 로드합니다.
REMAP_TABLESPACE	테이블스페이스 객체가 다른 테이블스페이스에 재매핑됩니다.
REUSE_DATAFILES	테이블스페이스가 존재하는 경우 초기화됩니다. (N)
SCHEMAS	임포트할 스키마 목록입니다.
SKIP_UNUSABLE_INDEXES	사용할 수 없는 인덱스 상태로 설정된 인덱스를 건너 뛩니다.
SQLFILE	모든 SQL DDL 을 지정된 파일에 씁니다.
STATUS	작업 상태가 모니터되는 빈도(초)입니다. 기본값(0)은 사용 가능할 때 새 상태를 표시합니다.
STREAMS_CONFIGURATION	스트림 메타 데이터의 로드 활성화
TABLE_EXISTS_ACTION	임포트된 객체가 존재하는 경우 수행할 작업입니다. 적합한 키워드: (SKIP), APPEND, REPLACE 및 TRUNCATE.
TABLES	임포트할 테이블 목록을 식별합니다.
TABLESPACES	임포트할 테이블스페이스 목록을 식별합니다
TRANSFORM	해당 객체에 적용할 메타 데이터 변환입니다.

	적합한 변환 키워드: SEGMENT_ATTRIBUTES, STORAGE OID 및 PCTSPACE 입니다.
TRANSPORT_DATAFILES	전송 가능 모드로 임포트할 데이터 파일 목록입니다.
TRANSPORT_FULL_CHECK	모든 테이블의 저장 영역 세그먼트를 확인합니다. (N)
TRANSPORT_TABLESPACES	메타 데이터가 로드될 테이블스페이스 목록입니다. NETWORK_LINK 모드 임포트 작업에서만 적합합니다.
VERSION	엑스포트할 객체의 버전입니다. 적합한 키워드: (COMPATIBLE), LATEST 또는 임의의 적합한 데이터베이스 버전. NETWORK_LINK 및 SQLFILE 에 대해서만 적합합니다.

다음 명령은 대화형 모드에서 적합합니다.

참고: 약어도 허용됨

명령	설명(기본값)
CONTINUE_CLIENT	로깅 모드로 돌아옵니다. 유휴 상태인 경우 작업이 재시작됩니다.
EXIT_CLIENT	클라이언트 세션을 종료하고 작업은 실행되도록 둡니다.
HELP	대화형 명령을 요약합니다.
KILL_JOB	작업을 분리하고 삭제합니다.
PARALLEL	현재 작업에 대한 활성 작업자 수를 변경합니다 PARALLEL=<작업자 수>.
START_JOB	현재 작업을 시작/재개합니다. START_JOB=SKIP_CURRENT 는 작업 정지 시 진행 중이던 작업을 건너 뛴 후 해당 작업을 시작합니다.
STATUS	작업 상태가 모니터되는 빈도(초)입니다. 기본값(0)은 사용 가능할 때 새 상태를 표시합니다. STATUS[=간격]
STOP_JOB	작업 실행을 순서대로 종료하고 클라이언트를 종료합니다. STOP_JOB=IMMEDIATE 는 데이터 펌프 작업의 데이터 펌프 작업.

참조문서 -

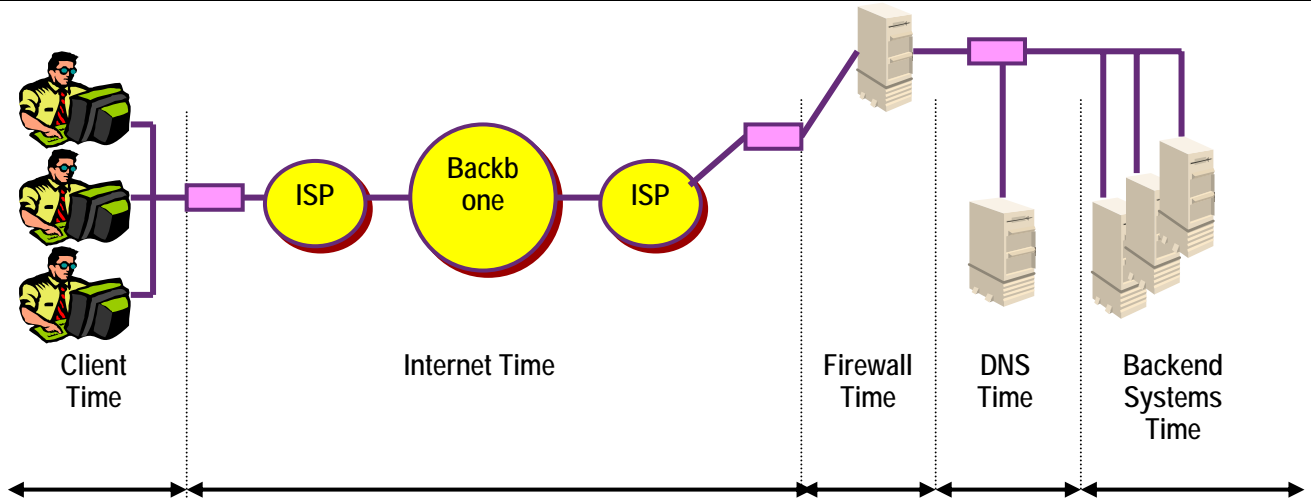
http://www.oracle.com/technology/products/database/utilities/htdocs/data_pump_overview.html

- oracle database 10g:new features for administrators student guide
- kr.forums.oracle.com

2. Pro-Active Tuning Service

2.1. 실제 사용자(End-User) 관점의 응답시간 튜닝

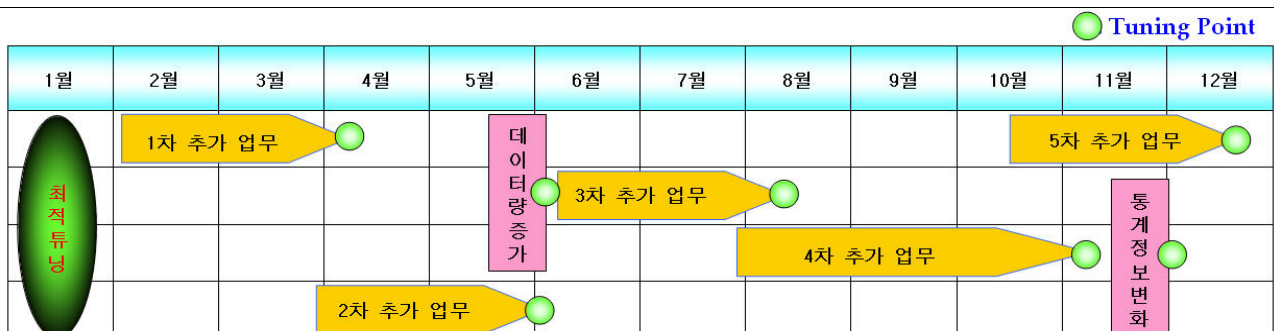
Pro-active tuning service 는 사용자 관점의 모니터링 및 분석을 통하여 실제 End-User 가 느끼는 응답시간(Response Time)을 튜닝합니다. APM(Application Performance Management) 툴을 이용하여 End-User 의 Request 결과를 반환 받기까지의 모든 구간(Client PC, Internet 구간, FireWall, DNS, Web Server, WAS, DBMS) 을 분석하여 가장 Delay Time 이 많이 소요된 구간을 찾아 냅니다.



2.2. 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. !!

2.3. 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.

2.4. Tuning 범위 확대

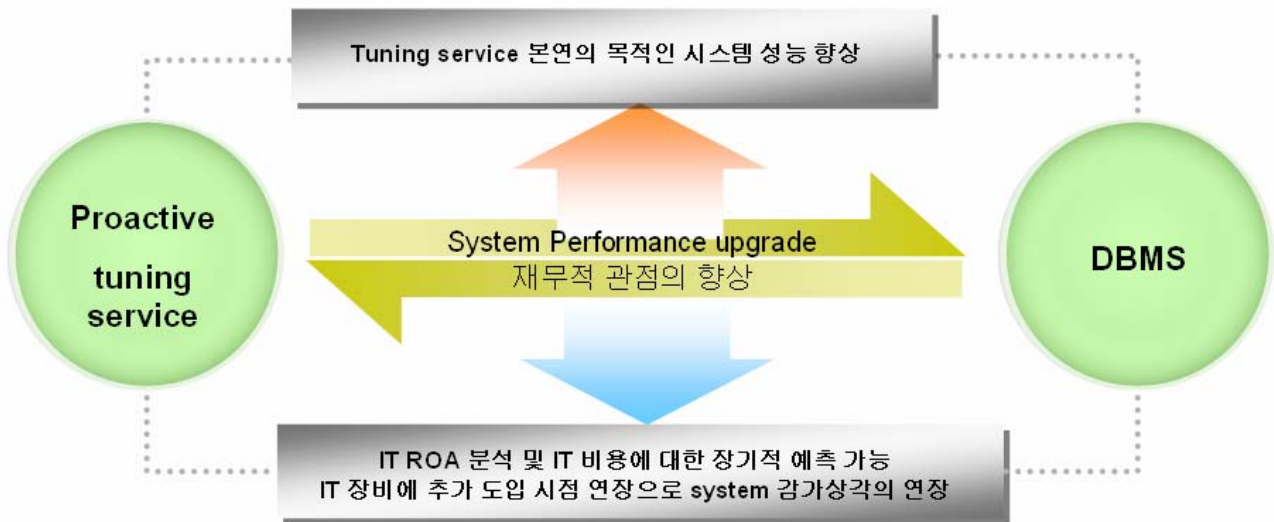
Pro-active tuning service 의 제공 범위는 제한된 Database, 제한된 Server 에 국한 되지 않으며, 고객사 전체의 Server+DB 를 대상으로 그 범위를 확대 함으로써 고객사 전체 Performance 향상에 기여 합니다.



2.5. 기대효과

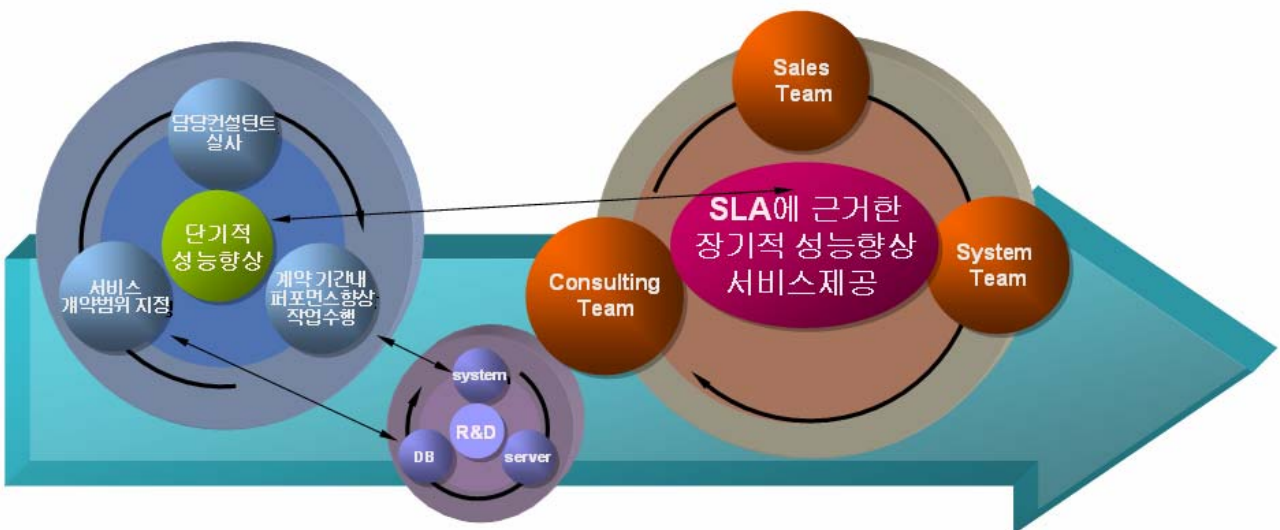
2.5.1. 재무적 관점

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.



2.5.2. 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.



2.5.3. 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.

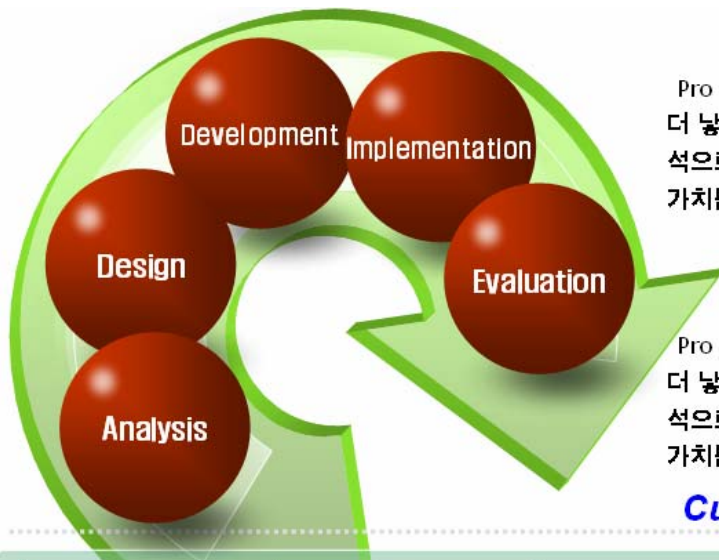
· 사용자 관점의 기술 전수 서비스 제공

계약 종료 시점 일정기간 획일적이고 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



2.5.4. 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



"Breathing Life into Technology"

굿어스㈜는 IT 프로세스 컨설팅 역량과 운영관리 기술을 갖춘 IT 전문 서비스 기업입니다.
다양한 고객들에게 IT 진단부터 운영관리 컨설팅, ITSM 구축 그리고 IT 인프라 매니지먼트 서비스를 제공합니다.
국내 유수의 기업들이 굿어스의 고부가가치 IT 서비스를 통해 비즈니스 성과 향상을 경험하고 있습니다.

