Cheatography

Tibero Command Cheat Sheet

by TmaxTibero via cheatography.com/212635/cs/46241/

세션 관리

자기 자신의 세션번호 빠르게 조회하기

```
SELECT
  *
FROM
  VT_MYSESSID;
```

간단한 세션모니터링

```
SELECT
  TO_CHAR(sysdate, 'yyyy/mm/dd hh24:m-
i:ss') AS dt,
  sid,
  serial#,
  username,
  status,
  sql_et / 1000 AS "sec",
  machine,
  prog_name
FROM
  v$session;
```

현재 수행중인 세션 확인

```
SELECT
  sid,
  sql_id,
  prog_name,
  ipaddr
FROM
  v$session
WHERE
  status = 'RUNNING';
```

자기 세션을 제외하고 running 중인 session 에 대해 kill session 구문 생성

```
SELECT
  'ALTER SYSTEM KILL SESSION(' || sid ||
',' || serial# || ')'
FROM
  v$session
WHERE
  status = 'RUNNING'
  AND sid &amp;lt;&amp;gt; (
    SELECT
      *
    FROM
      vt_mytid
  );
```

Session Kill 시 Spinlock을 잡고있는 상태면 DB Down이 발생함. Force로 Kill 하기 전에 Spinlock 잡고 있는지 확인하는 방법

성능 관리 (cont)

```
ALTER SYSTEM FLUSH PPC
  CHILE_NUMBER 'SQL_ID';
```

autotrace 수행

```
SET AUTOT TRACE EXP;
SET AUTOT TRACE EXP PLANS STAT;
```

접속 중인 다른 세션의 sql trace 수집
예) 접속 중인 89번 세션에 sqltrace 수집하기

```
exec DBMS_MONITOR.SESSION_TRA-
CE_ENABLE(89); // 활성화
exec DBMS_MONITOR.SESSION_TRA-
CE_DISABLE(89); // 비 활성화
```

DB 백업

유저 별 Export Backup

```
$ tbexport
  username=sys
  password=tibero
  sid=<SID>
  port=<PORT>
  file=<파일명>
  log=<로그명>
  script=y
  user=<유저명>
```

Control File Backup

```
ALTER DATABASE BACKUP
CONTROLFILE
  TO TRACE AS '<저장할파일명>';
```

기타 명령어

tbsql 명령어를 통한 sql 파일 직접수행

```
$ tbsql <계정명>/'<패스워드>' @<파일명>
```

현재 접속된 유저 확인

```
ls user
```

tbsql에서 loop 구문 후 SQL 문장 작성시 주기적으로 해당 SQL문 수행
예) loop select 1 from dual;

```
SET INTERVAL <시간간격(초)>;
LOOP <반복 수행할 SQL문>;
```

로그 및 덤프 관리

로그 및 덤프 관리 (cont)

데이터파일의 특정 블럭 번호 덤프 수행

```
ALTER SYSTEM DUMP DATAFILE <데이
터파일번호>
  BLOCK <블럭번호>;
```

데이터파일의 특정 블럭 번호 구간 덤프 수행

```
ALTER SYSTEM DUMP DATAFILE <데이
터파일번호>
  BLOCK <시작블럭번호> <끝블럭번호>;
```

특정 세션이 수행한 쿼리덤프 수집(메모리정보)

```
ALTER SYSTEM DUMP MEMLOG <SID>;
```

```
ALTER SYSTEM DUMP PPC
  'SQLID' CHILD_NUMBER;
```

모든 행위에 대해 Callstack 덤프 수행

```
ALTER SYSTEM DUMP CALLSTACK;
```

아카이브 로그 파일 덤프 수행

```
ALTER SYSTEM DUMP LOGFILE
  '<아카이브파일명(절대경로포함)>';
```

DB 할당 메모리 Dump 추출

```
ALTER SYSTEM DUMP LOGFILE '<아카이
브파일명(절대경로포함)>';
```

OS 환경 관리

파일 삭제 시 티베로 프로세스가 파일을 오픈하고 있는지 확인

```
$ fuser <파일명>
```

티베로 Port를 사용해야 하는데 충돌 발생 시, 해당 Port를 오픈한 프로세스 존재 확인

```
$ lsof -i <포트번호>
```

비정상 종료 시 공유 메모리, 세마포어 일괄 정리

```
$ ipcs -m | grep tibero6 | awk '{print $2}' |
while read line ; do ipcrm -m $line; done
$ ipcs -s | grep tibero6 | awk '{print $2}' |
while read line ; do ipcrm -s $line; done
```

DB 차원이 아닌 OS 차원에서 DB에 연결된 TCP 접속 확인

```
$ netstat -antp | grep <리스너포트> | grep
LISTEN
```

```
SELECT
  inst_id,
  sid,
  serial#,
  pid,
  OS_THR_ID
FROM
  gv$session
WHERE
  sid = <SID>
  AND inst_id = <Instance>
```

위 조회 결과가 없을 시 Kill Session 수행

```
ALTER SYSTEM KILL SESSION
  '<위에 조회된sid>,<위에 조회된 serial#>'
FORCE;
```

성능 관리

TPR Snapshot 생성

```
EXEC
  dbms_tpr.create_snapshot;
```

TPR Report 생성

```
EXEC
  dbms_tpr.report_text_id(<snap_id>,
    <snap_id>,
    <inst_id>,
    '<file_name>');
```

SQL 수행 플랜 보기 쿼리 수행 후 플랜 노드별 수치 보기

```
SELECT *
FROM table(
  dbms_xplan.display_cursor(null, null,
    'IOSTATS -TEMPREAD -TEMPWRITE
LAST CARDS ROWS -COST PREDICATE
ELAPTIME'
));
```

SQL 동작 중 Physical Plan Cache 초기화

tbev -ts 파일명 eventlog 바이너리를 텍스트로 추출하고자 할 때 사용

```
$ tbev -ts <파일명>
```

tbiv -ts 파일명 iLOG 바이너리를 텍스트로 추출하고자 할 때 사용

```
$ tbiv -ts <파일명>
```

특정 세션의 callstack 수집

```
ALTER SYSTEM DUMP CALLSTACK <SI-
D>;
```

특정 에러 번호에 대한 Callstack 덤프 on/off

```
ALTER SYSTEM DUMP CALLSTACK ON
ERROR
  <에러번호> <ON/OFF>;
```

컨트롤 파일 덤프 수행

```
ALTER SYSTEM DUMP CONTROLFILE;
```