

? TUXEDO- Transaction timeout 이후의 TMS_ORACLE waiting

제 품	TUXEDO
사용 OS	모든 플랫폼(platform)
작 성 일	2003/07/16
이슈사항	Application 서비스 진행중 transaction timeout이 발생한 경우 TMS_ORACLE process는 rollback을 시도한다. 이때 waiting이 발생할 수 있다.
해 결 안	위와 같은 상황에서 TMS_ORACLE process는 OPENINFO상의 SesTm값만큼 waiting을 한 후 rollback성공 요청을 받게된다. 그러나 SesTm전에 서비스가 수행이 끝나게 되면 이와 함께 TMS_ORACLE의 rollback 작업요청도 끝나게 된다. 위와 같은 action은 Oracle의 XA 구현에 따른 것으로 보이므로 Oracle version에 따라 상이한 상황이 생길 수도 있을 것으로 보인다.
추가정보	

? URESUMESIGS(), UDEFERSIGS() 를 이용한 signal 처리

제 품	TUXEDO 6.5
사용 OS	모든 플랫폼(platform)
작 성 일	2003/08/14
이슈사항	TUXEDO 6.5 patchlev 410 이상에서는 SVCTIMEOUT 발생 시에 환경변수 TM_SVCTIMEOUT_SIGTERM 이 설정되어 있으면, signal을 지연 처리하지 않고 즉시 처리되도록 하고 있다. 그러나 이는 프로세스가 BB critical section에서 작업 중인 때에 이러한 signal을 만나서 종료되는 경우에는 BB의 내부 데이터를 손상된 채로 남겨둘 수 있는 위험성이 있으므로 사용에 신중을 기하여야 한다. 따라서 전체 서비스 루틴 중에서 외부의 요인에 의해 처리가 지연될 수 있는 부분들에는 별도의 timeout 처리를 하여야 하며 SVCTIMEOUT 을 사용하지 않은 방안이 필요하다.
해 결 안	이에 대해서는 지난 호에 signal()을 사용한 방안이 제공되었었다.(BEST 통권 2호 p109~p112) 여기에서는 URESUMESIGS(), UDEFERSIGS()를 사용한 또 다른 방법을 제시하고자 한다. UDEFERSIGS() 는 signal을 지연 처리하도록 지시하는 MACRO이고, URESUMESIGS() 는 이러한 지연 처리를 해제하도록 하는 MACRO이다. 프로그램에서 이 둘을 사용할 때에는 아래의 예제에서처럼 반드시 쌍으로 사용하여야 한다. <pre>#include <stdio.h> #include <string.h> #include <ctype.h> #include <unistd.h> #include <errno.h> #include <signal.h> #include <Usignal.h> #include <atmi.h> #include <Usyslds.h> #include <fml.h> #include <userlog.h> EXEC SQL INCLUDE SQLCA; void EMP_UPDATE(rqst) TPSVCINFO* rqst;</pre>

해 결 안

```
{
EXEC SQL BEGIN DECLARE SECTION:
  short  empno;
  varchar ename[11];
EXEC SQL END DECLARE SECTION;

  FBFR*  transf;
  long   len;

  transf = (FBFR*)rqst->data;

  ename.len = 0;
  Fget(transf, Fldid("EMPNO"), 0, (char*)&empno, 0);
  Fget(transf, Fldid("ENAME"), 0, (char*)&ename.arr, &ename.len);

  URESUMESIGS();
  alarm(5);
  EXEC SQL UPDATE EMP
    SET      ENAME = :ename
    WHERE   EMPNO = :empno;
    userlog("sqlcode = %d, %s", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
  alarm(0);
  UDEFERSIGS();

  if (sqlca.sqlcode != 0) {
    userlog("sqlcode = %d, %s", sqlca.sqlcode, sqlca.sqlerrm.sqlerrmc);
    tpreturn(TPFALL, sqlca.sqlcode, (char*)NULL, 0L, 0);
  }

  tpreturn(TPSUCCESS, 0, (char*)transf, 0L, 0);
}
```

위의 예제에서 보드시피 전체 서비스 루틴이 아닌 DB 작업에만 이러한 timeout 처리를 하게 되면 보다 안전하며, 지난 호에서의 signal() 처리보다 많이 단순해진 것을 알 수 있다.

추가정보



TUXEDO – WSNAT_CAT:1175: WARN: Disconnecting listener connection to workstation, request timed out

제 품	TUXEDO
사용 OS	모든 플랫폼(platform)
작 성 일	2003/07/21
이슈사항	TUXEDO client에서의 tpinit과 관련 다음과 같은 Warning 메시지가 서버상의 ULOG에 발생하는 경우가 있다. ==== WSNAT_CAT:1175: WARN: Disconnecting listener connection to workstation, request timed out. ====
해 결 안	/WS client는 처음에 WSL과 접속한다. 이후 WSL은 가용한 WSH정보를 client에게 전달하고 client는 이 정보를 가지고 다시 WSH와 최종적으로 접속하여 tpinit작업을 마무리한다. 따라서 이때 가용한 WSH가 없으면(이미 모든 WSH가 client와 전부 session을 맺고 있는 등으로 인해) client는 일정시간(tpinit에 대한 timeout) 후에 tpinit 작업이 실패하게 된다. 위와 같은 메시지가 발생하는 빈도가 매우 적다면 순간적인 부하 등의 문제로 무시해도 별 상관이 없어 보이 나 빈번하다면 이와 관련된 ubb parameter(MAXACCESSERS, MAXWSCLIENTS, WSH관련 -m, -M, -x 등) 값의 조정을 고려해 보아야 한다.
추가정보	