

? xa_start=-9의 원인과 추적 방법

제 품	Tuxedo
사용 OS	ALL
이슈사항	<p>이 에러는 XA 트랜잭션을 새롭게 시작하고자 할 때, 정상적으로 XA 트랜잭션을 시작하지 못할 때 발생을 하는데, 그 주요 원인을 보면 다음과 같다.</p> <p>Oracle 8 부터는 하나의 Proc*C 프로세스에서 Local Transaction과 XA기반의 Global Transaction을 동시에 사용할 수 있게 되어있는데, xa_start = -9 (XAER_OUTSIDE)는 해당 프로세스가 Local Transaction mode에서 Global Transaction을 수행하고자 할 때 발생한다.</p> <p>이런 현상이 주로 발생하는 경우는 XA 모드에서 SQL을 수행하였을 때, timeout에 의해 SQL 에러가 발생하면 에러 처리를 해서 에러로 리턴 해야 하는데, 그렇지 않고 다음 SQL을 수행하면 다음 SQL은 Local Transaction으로 수행이 된다. 즉, SQL1 SQL2 이 두 문장을 하나의 XA Transaction으로 묶었을 때, SQL1에서 Timeout 이 되거나 SQL 에러가 발생하면 이 Global Transaction은 실패하고 Rollback으로 mark가 된다. Timeout이 난 후에 만약 SQL2를 수행하게 되면, SQL2는 이미 Global Transaction이 실패하였기 때문에 에러를 발생시킨다.</p> <p>이 상태에서 해당 프로세스가 다시 request를 받아서 tpbegin으로 global transaction을 시작하면 이미 앞에서 시작된 Transaction이 Local mode로 진행중이기 때문에 새로운 Global Transaction을 수행할 수 없게 되고 결과적으로 xa_start =-9 에러를 발생시키게 된다.</p> <p>다른 유사한 사항을 살펴보자.</p> <pre> --- client에서 ----- tpbegin(10,0); <=== XA tran 시작, time-out이 10초임에 유의 tpcall (...) ----- --- server에서 ----- EXEC SQL UPDATE ; <=== lock 1 sleep(20); <=== time out 유발 (TMS에 의해서 tpabort(0)가 호출됨.) </pre>

이슈사항	<pre> EXEC SQL UPDATE ... ; <=== lock 2 EXEC SQL UPDATE ... ; <=== lock 3 tpreturn (서비스 종료) ----- --- 서비스 종료 후 client에서 ----- tpcommit(0); <=== XA tran 종료 ----- tpbegin에서 time-out을 10초로 주고 수행을 했기 때문에, 정상적으로 lock1은 수행이 되겠지만, sleep(20)후에, 이 Transaction의 Timeout 시간은 10초이기 때문에 TMS에 의해서 timeout에 의한 tpabort가 호출된다. 그 다음에 lock2에 해당하는 SQL을 수행하게 되면, 트랜잭션이 없기 때문에 트랜잭션이 존재하지 않는다는 에러 24761 이 발생한다. 그러나 그 다음 SQL인 lock 3를 실행할 때에는 이미 lock2에서 에러가 처리되었기 때문에 (Oracle 입장에서는 에러가 처리된 것이다) 마치 에러가 없는것처럼 작동을 하고 SQL3는 XA 상태가 아닌 상태(local transaction)로 수행이 되고, tpcommit (혹은 tpabort)에서는 에러가 발생한다. 이 후에 또 서비스 요청이 오면, local transaction 상태이기 때문에 서비스 진입시에 xa_start error -9를 만나게 된다. </pre>
해 결 안	<p>이를 해결하기 위해서는 SQL1에서 에러가 났을 때 다음 SQL을 수행하지 말고, 바로 SQL error code와 함께 return을 해서 Global Transaction이 끝나게 하고, 새롭게 Local Transaction이 시작되지 않도록 해야한다.</p> <p>해당 에러가 발생할 경우, ULOG에서는 해당 process만을 알 수 있지 어떤 서비스인지를 알수가 없다. 이를 해결하기 위해서는 에러를 발생 시키는 server에 tadmin을 이용해서 trace를 걸면 어느 service에서 에러가 발생하는지를 찾아낼 수 있다.</p> <p>tadmin에서 changetrace -i 서버ID -g 그룹명 "atmi+iatmi+xa+trace:ulog:dye"으로 써 주면 해당 Server가 동작할 때 마다, ULOG에서 ATMI 와 XA에 대한 호출 로그를 남기기 때문에 xa_start=-9가 나오기 전에 어떤 서비스가 호출되었는지를 찾으면 문제가 있는 서비스를 찾아낼 수 있다.</p> <p>그 후에 소스코드 상에서 Oracle SQL문장 수행 후에 SQL 에러처리를 해서 다음 SQL을 수행하지 않도록 return하는 코드가 있는지를 검증해야 한다.</p> <p>이를 위해서는 단순히 소스를 검증하는 방법도 있고, 또는 SQL 수행시마다 ULOG에 SQL 에러코드와 함께 수행된 SQL문장을 출력해보면 XA 에러 후에 계속 SQL이 수행되었는지를 찾아볼 수 있다.</p>
추가정보	<ul style="list-style-type: none"> • 또는 이를 예방하는 방법으로 Oracle 9.2이상을 사용할 때 Tuxedo 서버 GROUP의 OPENINFO에 NoLocal=T로 주면 Local Transaction으로 빠지는 현상을 예방할 수 있다.